Altova DatabaseSpy 2023 Enterprise Edition

User & Reference Manual

Altova DatabaseSpy 2023 Enterprise Edition
User & Reference Manual

Published: 2022

Table of Contents

4   Connecting to a Data Source   115

5   DatabaseSpy Projects   206

6   Managing Database Objects   222

1   Introduction

Altova DatabaseSpy 2023 Enterprise Edition is a Windows database client application that simplifies querying, visualizing, managing, and designing a wide range of relational databases. It allows database administrators, database developers, and other IT professionals interact with multiple database kinds consistently from the same graphical user interface. With DatabaseSpy, you can design or query databases visually or by means of SQL, generate SQL statements [346], generate scripts [349] describing the structure of existing databases, perform data import [400] (XML, CSV) or export [406] (XML, CSV, Excel, HTML), and reuse resources [436] defined globally across multiple Altova applications, such as files or database connections. SQL syntax highlighting [359], auto-completion [369], conditional formatting [388], and text formatting [354] capabilities are available for a wide range of SQL grammars. The Enterprise edition of DatabaseSpy provides the ability to automate repetitive tasks with the help of SQL Templates [382], and includes charting [310] capabilities.

ALTOVA®
DatabaseSpy®

Last updated: 7 October 2022

1.1   File Paths

File paths given in this documentation will not be the same for all operating systems. Note the following correspondences:

- *Documents folder:* Example files are located at C:\Users\<username>\Documents\Altova.
- *Application folder:* The Application folder is the folder where your Altova application is located (by default, C:\Program Files\Altova). When DatabaseSpy 32-bit is installed on a 64-bit operating system, the path is C:\Program Files (x86)\Altova.

1.2   Support Notes

DatabaseSpy runs on the following operating systems:

- Windows 7 SP1 with Platform Update, Windows 8, Windows 10, Windows 11
- Windows Server 2008 R2 SP1 with Platform Update or newer

Both the 32-bit and 64-bit platforms are supported.

DatabaseSpy integrates with Microsoft Office products as follows:

- It can be used to query and design Access databases. For supported versions, see Supported Databases [16]
- It can export data to Excel format, see Exporting Database Data [406].

1.3   Supported Databases in DatabaseSpy

The table below lists all the supported databases. While Altova endeavors to support other databases, successful connection and data processing have only been tested with the databases listed below. If your Altova application is a 64-bit version, ensure that you have access to the 64-bit database drivers needed for the specific database you are connecting to.

Database	Notes
Firebird 2.x, 3.x	
IBM DB2 8.x, 9.x, 10.x, 11.x	
IBM Db2 for i 6.x, 7.4	Logical files are supported and shown as views.
IBM Informix 11.70 and later	
MariaDB 10 and later	
Microsoft Access 2003 and later	At the time of writing (early September 2019), there is no Microsoft Access Runtime available for Access 2019. You can connect to an Access 2019 database from Altova products only if Microsoft Access 2016 Runtime is installed and only if the database does not use the "Large Number" data type.
Microsoft Azure SQL Database	SQL Server 2016 codebase
Microsoft SQL Server 2005 and later Microsoft SQL Server on Linux	
MySQL 5 and later	
Oracle 9i and later	
PostgreSQL 8 and later	PostgreSQL connections are supported both as native connections and driver-based connections through interfaces (drivers) such as ODBC or JDBC. Native connections do not require any drivers.
Progress OpenEdge 11.6	With Progress OpenEdge databases, there may be cases where a table cannot be deleted using the SQL script generated by DatabaseSpy. The workaround is to drop the table by entering a DROP TABLE statement in the SQL Editor window, instead of running the generated Database Structure Change Script. The limitation originates in the fact that, in Progress OpenEdge, you cannot drop the first index created on the table, except by dropping the table. DatabaseSpy, on the other hand, attempts to remove all table constraints (which may be linked to the first index created on the table) before removing the actual table.
SQLite 3.x	SQLite connections are supported as native, direct connections

Database	Notes
	to the SQLite database file. No separate drivers are required.
Sybase ASE 15, 16	
Teradata 16	

2   The DatabaseSpy Interface

The main components of the DatabaseSpy graphical user interface are illustrated below.
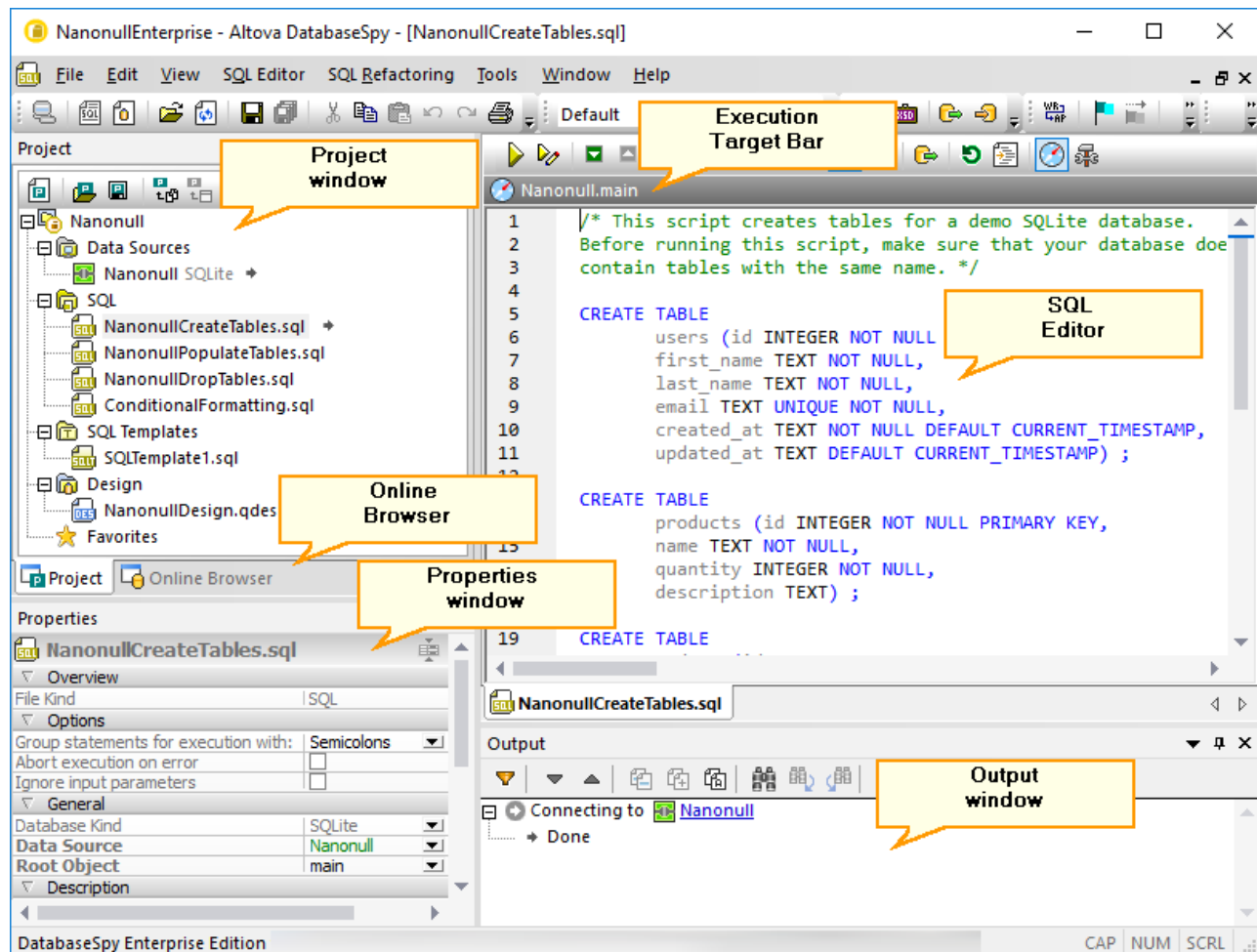
The interface consists of the following main items:

- Several helper windows which help you connect to a database, explore objects inside it, and organize files into projects. These include the Project window [20], Online Browser [24], Properties window [30], Overview window [31], Data Inspector window [32], Charts window [35], and Change Script window [37]. Some of these windows are displayed by default, while others can be triggered on demand from the **View** menu.
- The main pane is reserved for the SQL Editor [39], where you can enter, edit generate, and run SQL statements (this requires that a data source be created first, see the Tutorials [52]).
- The Output Window [33] (displayed by default in the lower part of the interface) shows various messages, warnings, and errors that may occur while you work with databases.

Use the ▼ ⊼ × buttons displayed in the upper-right corner of each window to show, hide, pin, or dock it. To restore toolbars and windows to their default state, run the menu command **Tools | Restore Toolbars and Windows**.

To quickly hide (or unhide) all windows except the main window, run the commands **View | Toggle All Side Windows** and **View | Toggle Output and Change Script Windows**.

2.1   Project Window

The Project window displays files and other items that you have added to your project, which include:

- Data sources (see [Adding Data Sources](#) [208])
- SQL files (see [Adding Files](#) [210])
- SQL templates (see [SQL Templates](#) [382])
- Design files ([Opening, Saving and Printing Diagrams](#) [232])
- External files (files external to DatabaseSpy that you can open in the application with which they are registered)
- Favorites (see [Favorites](#) [212])

Project window

The Project window provides the following commands:

Icon	Command name	Description
	Create a new project	Closes the active project in the Project window and displays a new, empty project. If a project contains unsaved changes when you click this button, a dialog box appears and you can choose to save changes in the editor window and project modifications. All editor windows are set offline.
	Open a project	Shows the Open Project dialog box where you can choose a *.qprj file to open in DatabaseSpy. If there is an unsaved project already open in the Project window, you are prompted to save this project.

Icon	Command name	Description
	Save project	Saves the project under its project name. If you save a new project for the first time, the Save As dialog box is displayed where you can specify a project name and choose a location. The project name then also appears in the Project window.
	Add files to the project	Shows the Open dialog box where you can browse for DatabaseSpy documents (SQL files, design files) that should be added to the project. Files are added to the corresponding folder in the Project window.
	Add the active file	Adds the active file to the project. If the file has not been saved yet, you are prompted to enter a name and location for the file. The file is then added to the corresponding folder in the Project window.
	Find	Finds a project item by name. Wildcards can be used in the search. For example, the search term "Pr*ject" will find an item called "Project".
	Find next	Finds the next occurrence of the specified text.
	Find previous	Find the previous occurrence of the specified text.

Depending on where you right-click into the Project window, different options are displayed in the context menu. The same context menu command may be available in multiple places (as shown in the "Applies to" column below).

Context menu command	Description	Applies to...
New Project (Ctrl+Shift+N)	Creates a new project.	Any Project
Open Project (Ctrl+Shift+O)	Opens a project.	Any Project
Save Project (Ctrl+S)	Saves the project under its current name.	Any Project
Save Project As...	Opens the Save As dialog box where you can define a new name and/or path for the project.	Any Project
Add Files to Project...	Opens the Open dialog box where you can select files to be added to the project.	Any Project SQL SQL Templates Design
Add Active File to Project	Includes the active file with the project and adds it to its corresponding folder in the Project window.	Any Project SQL SQL Templates Design

Context menu command	Description	Applies to...
Connect to all Data Sources	Connects to all data sources that are included in the project.	Data Sources
Disconnect from all Data Sources	Disconnects from all active data sources in the project.	Data Sources
Remove all Data Sources	Removes all data sources from the project.	Data Sources
**Expand	Siblings** **Expand	Children**	Expands all folders on the same level as, or all children of, the selected item.	Data Sources SQL SQL Templates Design
**Collapse	Siblings** **Collapse	Children**	Collapses all folders on the same level as, or all children of, the selected item.	Data Sources SQL SQL Templates Design
Add a new Data Source...	Opens a dialog box where you can define a new data source connection.	Data Sources
Convert all to Global Resources	Converts all data sources in the project to global resources. Note that data source names must not contain blanks.	Data Sources
Connect	Connects to the selected data source.	Any Data Source
Disconnect	Disconnects from the selected data source.	Any Data Source
Remove (Del)	Removes the selected item from the project.	Any Data Source Any Folder Any SQL Script.sql Any SQL Template.sql Any Design File.qdes
Rename (F2)	Renames the selected item. Note that you can rename data sources only if they are disconnected.	Any Data Source Any Folder
Convert to Global Resource	Converts the selected data source into a global resource.	Any Data Source
Copy Global Resource into Project	Creates a copy of the global resource and adds it to the project as a normal data source.	Any Data Source
Edit Global Resource	Opens the Global Resource dialog box where you can edit the global data source connection.	Any Data Source
Create Folder...	Creates a sub-folder for the selected item.	SQL SQL Templates Design

Context menu command	Description	Applies to...
▶ **Execute All SQL Files**	Opens all SQL files that are contained in the SQL folder in new SQL Editor windows and executes them automatically. Note that a data source must be assigned to the files.	📁 SQL
Open	Opens the selected file in its corresponding editor window, respectively.	📄 Any SQL Script.sql 📄 Any SQL Template.sql 📄 Any Design File.qdes
Locate file...	Opens Windows Explorer so that you can locate the file after it has been moved on the disk.	📄 Any SQL Script.sql 📄 Any SQL Template.sql 📄 Any Design File.qdes
▶ **Execute SQL**	Opens and executes the SQL file in an SQL Editor window. Note that a data source must be assigned to the file.	📄 Any SQL Script.sql 📄 Any SQL Template.sql
Remove all favorites	Removes all items from the Favorites folder.	⭐ Favorites

2.2 Online Browser

The Online Browser displays all database objects on the database server(s) to which you are currently connected. The structure of folders and database objects displayed in the Online Browser may slightly vary depending on the database kind you are connected to.
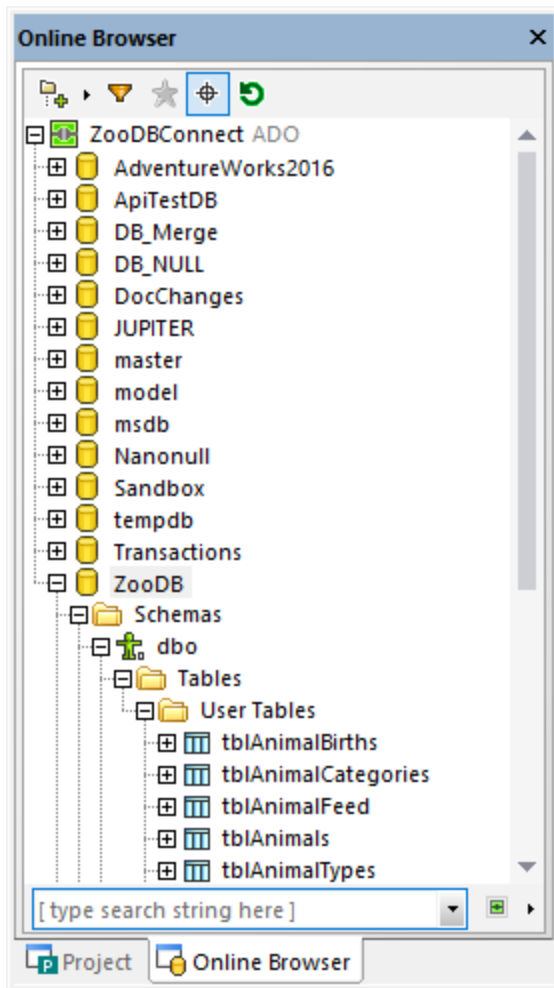
Online Browser

Note: After connecting to a PostgreSQL database server, you can explore only the currently connected database (catalog). Other databases on the server are visible in the Online Browser but not accessible unless you connect to them explicitly.

The Online Browser serves as a starting point for most of the database actions in DatabaseSpy: retrieve or edit data, create SQL statements, explore database objects, export database data, or show the row count for tables. All these actions can be started from the Online Browser with a few mouse clicks, as further described in this documentation.

Note: For Online Browser to be populated, an active (connected) data source must exist in your project. By default, disconnected data sources are not displayed in the Online Browser. If you want data sources

---

to be displayed in Online Browser regardless of whether they are connected or not, first select the data source in the Project Window [20], and then select the **Always show in Online Browser** check box in the Properties Window [30].

The following database objects may be displayed in the Online Browser:

- Databases
- Schemas
- Tables
- Views
- Procedures
- Columns
- Constraints
- Keys
- Indexes
- Data types
- XML Schemas (for DB2, SQL Server, and Oracle databases)
- Functions
- Triggers

The Online Browser's toolbar includes the following buttons:

- **Layout**
- **Filter folder contents**
- **Show favorites**
- **Object locator**
- **Refresh**

Layout

The **Layout** button controls how various database objects are displayed in the Online Browser. Note that this button changes the appearance based on the layout that is currently selected. You can select from among the following predefined layouts:

	Folders	Organizes database objects into folders based on object type in a hierarchical tree (for example, "Schemas", "Tables", and others). This is the default setting in DatabaseSpy.
	No Schemas	This layout is similar to the **Folders** layout, except that there are no schema folders; tables are therefore not categorized by schema.
	No Folders	Displays database objects in a hierarchy without using folders. For a better overview, you can select the **Show labels in browser window** check box in Online Browser options [420], so that the name of each database object appears as a prefix.
	Flat	Organizes all database objects into folders according to their type, as a flat (not hierarchical) structure. For example, there is one

		"Tables" folder that contains all tables, one "Columns" folder with all columns, one "Indexes" folder with all indexes, and so on.
	Table Dependencies	Categorizes tables according to their relationships with other tables. The categories include tables with foreign keys, tables referenced by foreign keys, tables without relations, and tables with self-referencing foreign keys.

For some databases such as SQL Server, you can optionally display user tables and system tables into separate folders, as follows:

- Right-click the "Tables" folder and select **Sort into User and System Tables** from the context menu.

Note that this function is available only when using the following layouts:

- Folders
- No Schemas
- Flat

Filter folder contents (Ctrl+Alt+F)

Use the **Filter** tool to filter database objects by name or part of a name. Objects are filtered as you type in the name. Filtering is case-insensitive by default; however, you can enable case-sensitive filtering from the General options [418].

Note: Filtering is not supported when you select the **No Folders** layout.

To filter objects:

1. Click the **Filter folder contents** button. Alternatively, select a database object and press **Ctrl+Alt+F** to activate filtering. Notice that filter icons appear next to folders.
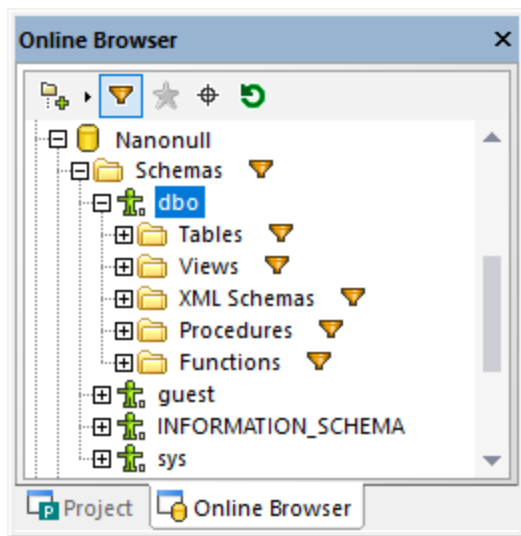
2. Click the filter icon next to the folder you want to filter and select the desired filtering option, for example, "Contains".

3. Start typing the filter text next to the [icon]. icon. The results are adjusted as you type.
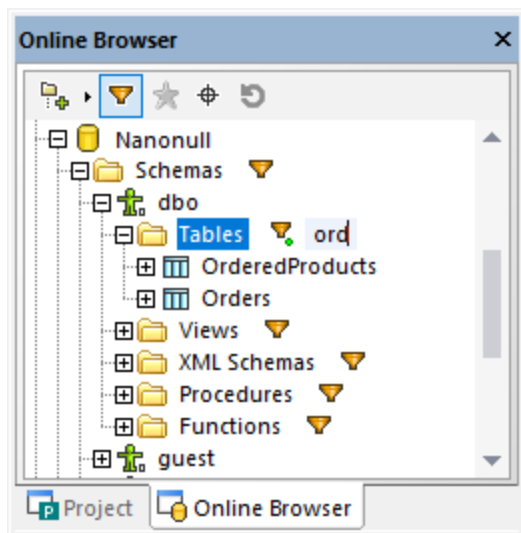
Show Favorites

You can add database objects that you use frequently to your favorites. Favorite objects appear in bold in the

Online Browser and are available in the Favorites folder of the Project window. The **Show Favorites** button lets you to switch between viewing all objects and showing only favorites.

Object Locator (Ctrl+L)

The **Object Locator** tool lets you find and filter database objects by any part of their name. Filtering is case-insensitive by default. You can enable case-sensitive filtering from the General options [418].

To find database objects using the Object Locator:

1. Click the **Object Locator** button or press **Ctrl+L**. A combo box appears at the base of the Online Browser.

2. Optionally, change the Object Locator context by clicking the arrow icon next to the combo box. You can restrict the display of objects as follows:

From current Data Source	Displays only objects from the currently selected data source.
From Focused Item	Displays only objects nested under the folder or object that is currently selected.
All	Displays all objects from all data sources that are currently connected.

3. Optionally, enter the string you want to look for (for example, "feed"). The drop-down list displays all elements that contain that string.
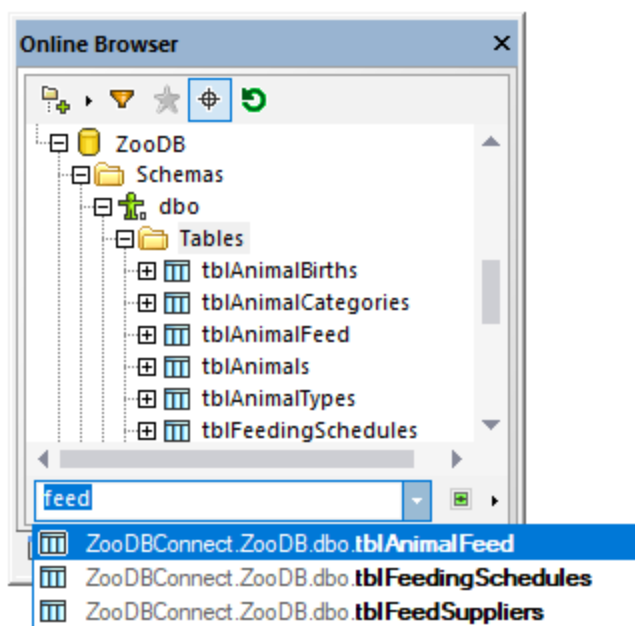
Refresh Data Source (F5)

Click the **Refresh** [icon] button to update the display of the Online Browser with the latest changes in the database structure.

Note: The Refresh functionality has a known limitation; namely, it will not update the Online Browser with the latest database structure after you have added or dropped a database. To address this limitation, disconnect from the data source and then reconnect to it.

2.3   Properties Window

The Properties window displays the properties of any selected object in the graphical user interface of DatabaseSpy. For a better overview, each element is assigned a particular color in the properties title bar. You can change these colors in the Design Editor options [434]. If no object is selected, the properties of the editor window itself will be displayed.

Properties	×
⊞ **tblVeterinarians (dbo)**	
▽ Overview	
	Update All Counts
Number of Columns	Update Count
Number of Check Constraints	Update Count
Number of Primary Keys	Update Count
Number of Foreign Keys	Update Count
Number of Unique Keys	Update Count
Number of Indexes	Update Count
Number of Triggers	Update Count
▽ General	
Schema	dbo
Unique Id	2101582525
▽ Description	
Description	

Properties window

You can customize the appearance of the Properties window by clicking one of the following buttons at the right edge of the header line:

	Hide empty properties	Hides all lines where no entry appears.
	Show empty properties	Restores all lines and displays also empty properties.

For performance reasons, information about child objects of an element is loaded only on demand, that is, if that particular element is expanded in the Online Browser or displayed in the Design Editor. To explicitly request information about an object, click the appropriate **Update Count** field in the object properties.

Some properties may have informative tooltips associated with them. To display the tooltip (if available), hover over the respective property. For more information, see Properties [214].

2.4   Overview Window

The Overview window serves as a navigator for large design files that occupy more space than is available in the Design Editor window. The Overview window shows the currently displayed area in the design and you can move the focus by clicking into the rectangle and moving it to the desired location in the design file.

To display the Overview window:

- On the **View** menu, click **Overview Window**.
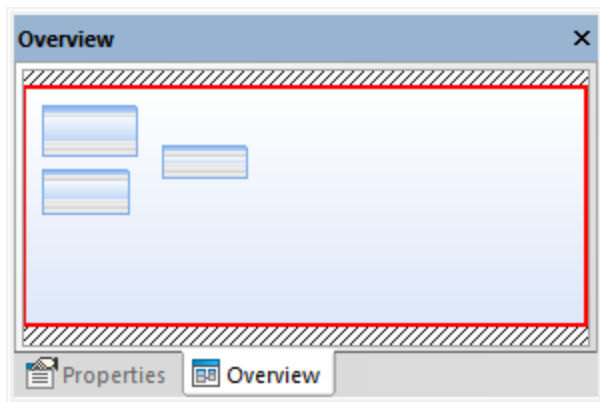
Overview window

Like other DatabaseSpy windows, the Overview window can be undocked and dragged to any position within the application, see [Arranging the Information Windows](#) [51].

2.5   Data Inspector Window

The Data Inspector Window is a convenient way to display the content of large data cells when you have only limited space to view the data directly in the result grid. The Data Inspector can be called via the **View** menu or by clicking the **Show Data Inspector window** ⬒ while one or more data cells are selected in the Result window [41].
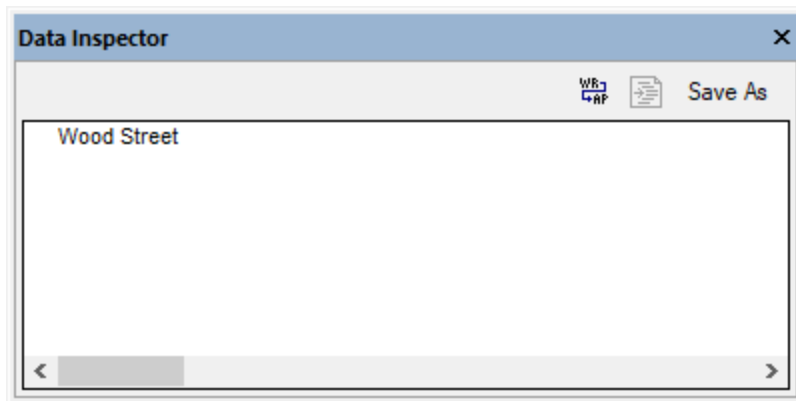
Data Inspector window

You can resize the Data Inspector window to view all the data it contains, or dock it to the application, see Arranging the Information Windows [51]. In addition, the following commands are available in the Data Inspector:

Icon	Command	Description
⬚	**Word wrap**	Automatically wraps the text in the window so as to make all the text visible. This is a toggle command, that is, word wrap is turned off again if you click the button again.
⬚	**Pretty Print**	This button is applicable only to cells that store XML data. It indents the XML tags in a consistent and more readable way.
	Save As	Opens the standard Windows Save As dialog box where you can specify a path and file name for the content of the data cell. The default file format is XML for XML columns and text for any other columns. You can also choose a different file format if you select "All Files" in the **Save as type** drop down list.

2.6   Output Window

The Output window in DatabaseSpy contains loading and error messages. Underlined parts in the Output window are hyperlinks which enable you to jump to a certain database object in the Online Browser.
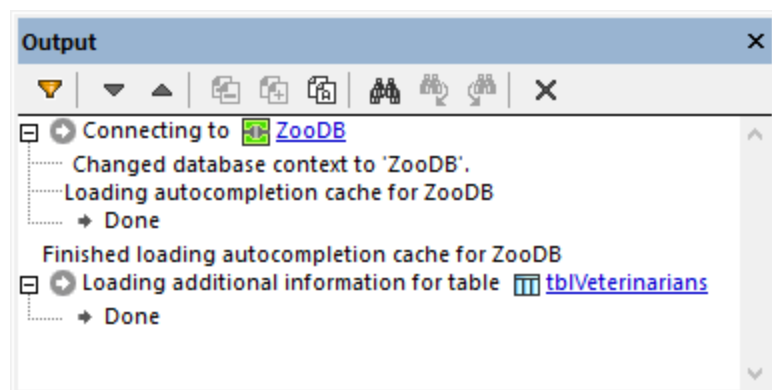
Output window

The Output window provides the following commands:

Icon	Command	Description
	Filter	Opens a context menu from where you can filter out messages by their severity (information messages, errors, warnings, and so on). Furthermore, you can check all or none of these options with a single mouse click by selecting either **Check All** or **Uncheck All** from the context menu.
	Next	Jumps to and highlights the next message.
	Previous	Jumps to and highlights the previous message.
	Copy message	Copies the selected message to the clipboard.
	Copy message with children	Copies the selected message including its children to the clipboard.
	Copy all	Copies all messages in the message window to the clipboard.
	Find	Opens the Find dialog box.
	Find previous	Jumps to the previous occurrence of the string specified in the Find dialog box.
	Find next	Jumps to the next occurrence of the string specified in the Find dialog box.
	Clear	Removes all messages from the Output window.

The Output window may show any of the following status icons accompanied by a respective message:

Icon	Description
✅	Indicates a success message. For example, it can appear after a database update query has been successfully run.
⊙	This message indicates the progress of an operation.
ⓘ	Indicates an information message.
⚠	Indicates a warning message.
❌	Indicates an error message. It can occur, for example, when a database query fails. The actual error text may be provided by the driver which was used to connect to the database.

2.7   Charts Window

The Charts window serves to display selected data from the result grid in a graphical form. It contains nine tabs so that you can produce different charts from a single result grid or from different Result windows.
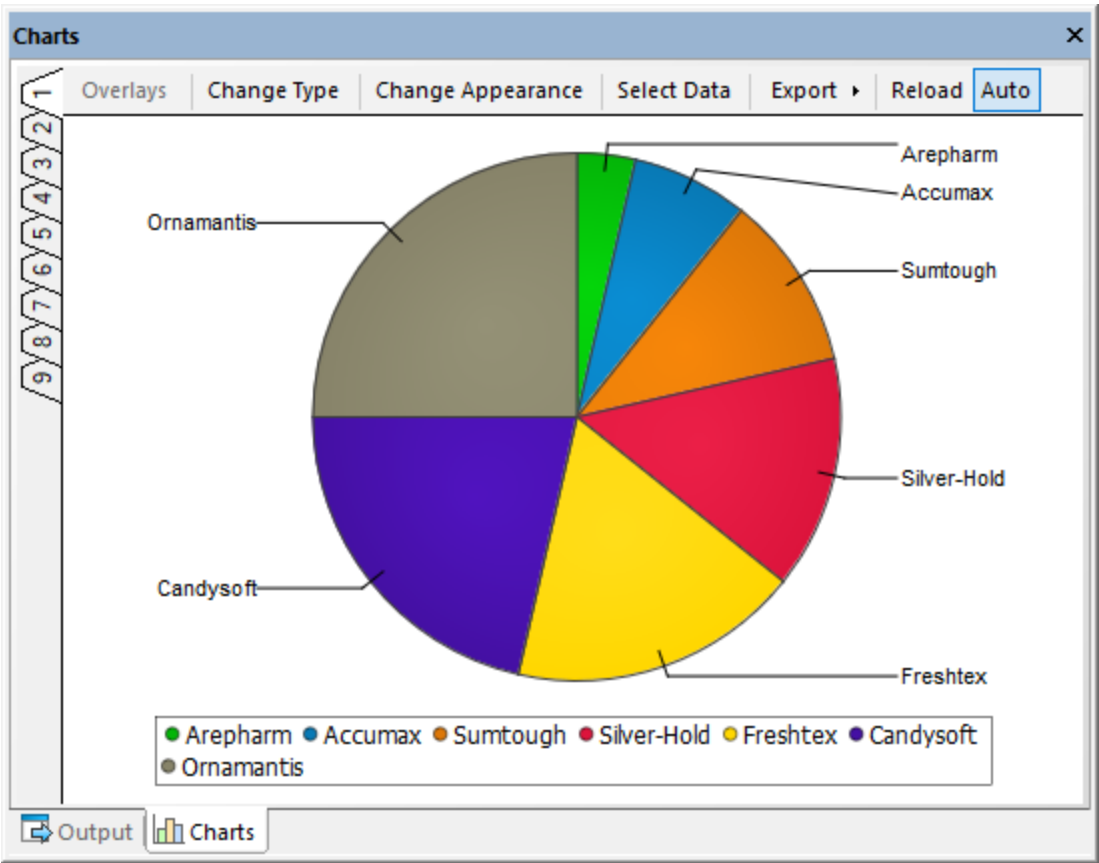
The Charts window provides the following commands:

Command	Description
Overlays	Lets you to create multiple layers in the same chart window. This button is disabled if the chart type does not support overlays, see Chart Overlays [316].
Change Type	Opens the Chart Type dialog box where you can choose between pie, bar, line, and gauge charts and select several sub-types.
Change Appearance	Opens the Change Appearance dialog box which allows you to define titles, labels, axes, sizes, fonts, and 3d features (if applicable).
Select Data	Opens the Chart Data dialog box where you can choose the columns that should be depicted on the X- and Y-axes.
Export	Opens a submenu which allows you to save the chart, copy it to the clipboard or print the chart.

Command	Description
Reload	Reloads the data from the Result window. If the **Auto** toggle is activated, data is reloaded automatically when the source data is changed.

For more information, see [Displaying Charts](#) [310].

2.8   Database Structure Change Script Window

The **Database Structure Change Script** window records any changes you make to the database structure in an SQL script. This script is either generated instantly or on demand, depending on the settings in the Change Script options [420].
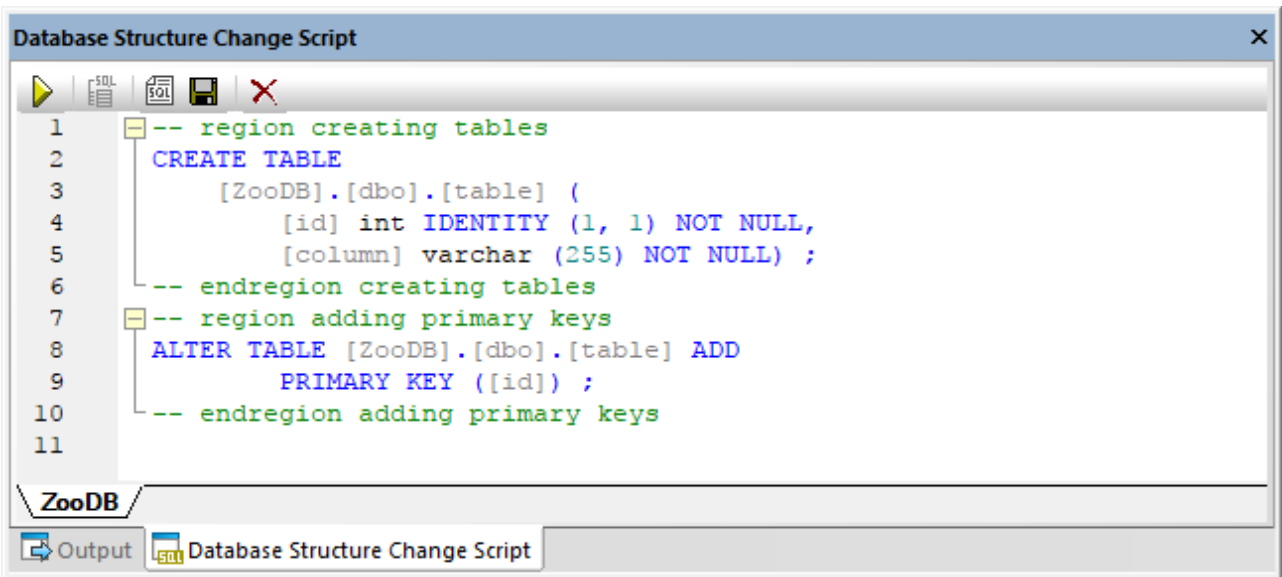
Database Structure Change Script window

In order to provide a better overview if changes are made in large scale, the change script is structured into regions, which can be collapsed or expanded when checking the script. Please note that collapsing a region will not exclude this part of the script from execution; collapsed regions are expanded automatically when the change script is being executed.

If you have several data sources connected and you change the structure in more than one of them, a separate tab is created for each data source in the Database Structure Change Script window.

When you define primary keys or check constraints, the Database Structure Change Script window also displays comments on the validity of the definition.
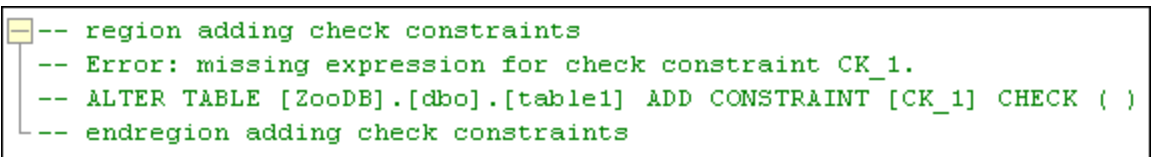
The Database Structure Change Script window provides the following commands.

Icon	Command	Description
	Execute Change Script	Executes the change script and thus implements the recorded changes in the database. In the Change Script options [420], you can define the

Icon	Command	Description
		time delay during execution of the script.
■	**Stop**	Terminates the execution of the change script, the remaining changes are not implemented in the database and the Database Structure Change Script window is cleared. This button is only visible while a change script is being executed. **Note**: When you click the **Stop** button, all changes that have not been executed so far will be lost.
SQL	**Generate Change Script**	Generates or updates, respectively, a change script that reflects all the changes you have made in the database design since the change script has last been updated. This button is only active if you have defined in the [Change Script options](#) [420] that the change script be generated on demand.
SQL	**Edit Change Script**	Opens the change script in a new SQL Editor window where you can edit it. **Note**: The script will be removed from the Database Structure Change Script window and the changes will be lost unless you execute the script in the SQL Editor.
💾	**Save Change Script**	Opens the standard Windows **Save As** dialog box where you can save the script as an SQL file.
✕	**Discard Changes**	Discards all changes recorded in the change script. The changes you made in the design will not be implemented in the database and the Database Structure Change Script window will be cleared.

Note: The Database Structure Change Script window becomes visible whenever you take an action that is about to change the structure of the database, even if you have previously closed this window.

2.9   SQL Editor

The SQL Editor is the area where you can display, edit, and execute SQL statements. It provides a Message window [41] that displays success or error messages and warnings, and a Result window [41] to display the query results, if applicable.
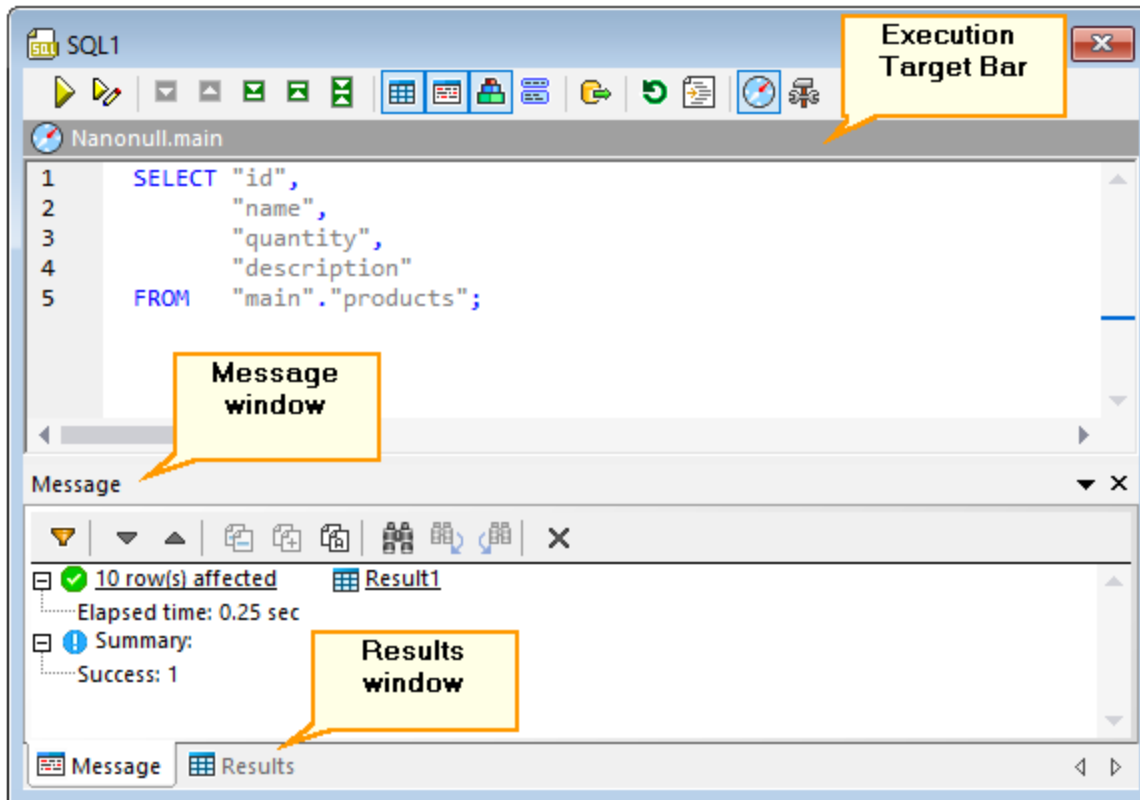
SQL Editor window

Note: A new SQL Editor window is automatically associated with the data source that is active in the Online Browser [24] or Project Window [20]. The active data source is displayed at all times in the Execution Target Bar [49]. That is, all commands issued from SQL Editor will affect this particular data source only. If there are multiple data sources, the SQL Editor belongs to the one which was last clicked before you opened the SQL Editor. Any subsequent SQL Editor windows that you open will inherit the currently active data source.

The Execution Target Bar (set to be visible by default) indicates at all times the database against which all statements in the SQL Editor will be executed. See also Execution Target Bar [49].

The SQL Editor window provides the following commands:

Icon	Command	Description
▷	**Execute (F5)**	Runs the statement (or statements) displayed in SQL Editor against the database.
▷	**Execute for data editing**	This command is applicable only for SELECT statements. It executes the SELECT statement and switches the results into editing mode. See also Retrieving and Editing Data[284].
▣	**Next Statement (Alt+Down)**	If the SQL Editor contains multiple statements, this command jumps to the next statement and selects it.
▣	**Previous Statement (Alt+Down)**	If the SQL Editor contains multiple statements, this command jumps to the previous statement and selects it.
▣	**Last Statement (Alt+End)**	If the SQL Editor contains multiple statements, this command jumps to the last statement and selects it.
▣	**First Statement (Alt+Up)**	If the SQL Editor contains multiple statements, this command jumps to the first statement and selects it.
▣	**Select Entire Statement**	Selects the SQL statement in which the cursor is currently located.
▦	**Toggle Result Window(s)**	Shows or hides the **Result** window containing the query results (if applicable). If the query returns multiple recordsets, then multiple Result tabs will be shown.
▤	**Toggle Messages Window**	Shows or hides the **Message** window.
⛁	**Show multiple results stacked**	This option is applicable if the SQL Editor contains multiple statements which would retrieve multiple recordsets after execution. If the option is on, all results are displayed in a single Result window with multiple panes for the individual queries. If this option is off, results of queries are displayed in separate, sequentially numbered tabs (Result1, Result2, etc). See also Viewing Results[288].
▤	**Show groupings for execution**	If the SQL Editor contains multiple statements, this command displays statements as standalone execution blocks, if allowed by the context. See also Showing Execution Groups[364].
⮫	**Export**	Opens the Export dialog box, see Exporting Database Data[406].
↻	**Re-parse the whole document**	Re-parses the statements in the currently active SQL Editor window. This helps correct parsing errors that may have occurred during the initial parsing of the statement.
▤	**Apply SQL formatting to the active SQL Editor**	Formats statements in the SQL Editor according to the SQL grammar defined for the current database type, see Formatting SQL[354].

Icon	Command	Description
	Show/Hide execution target bar	Shows or hides the Execution Target Bar [49]. Note that the Execution Target Bar will also be toggled off for all SQL Editor and Design Editor windows that you open subsequently.
	Show Options	Displays the Options dialog box, see DatabaseSpy Settings [417].

2.9.1   Message Window

The Message window provides a statistical overview on the SQL statements that have been executed in the SQL Editor [39] and reports errors that might have occurred during execution. The "Summary" statistic displays the total count of errors, warnings, or info messages, as applicable.
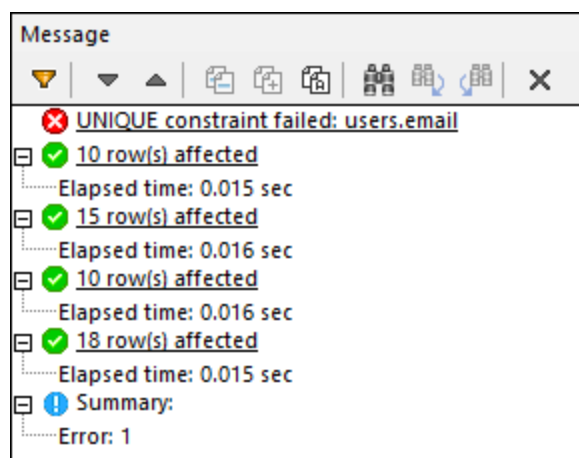
Underlined parts in the Message window are hyperlinks which enable you to jump to a certain part of the SQL script in the SQL Editor, or to the corresponding Result window [41] that holds the result of the respective SQL statement.

The commands and status icons available in this window are identical with those in the Output Window [33].

2.9.2   Result Window

The Result window of the SQL Editor shows the recordsets that were retrieved as a result of the database query. If an SQL Editor window contains more than one query, the results are displayed either in individual tabs or stacked in a single window, depending on configured options, see Viewing Results [288].

The status bar at the bottom of the window displays information on the progress of the query: whether the retrieval executed successfully, was aborted or has been stopped by the user. In addition, the number of rows and columns retrieved as well as the amount of time necessary for retrieval, and the time when the query was executed.

Note: In DatabaseSpy, a maximum number of 30 result tabs is allowed. If you execute a query that would produce more than 30 result tabs, only the first 30 result tabs are generated and a corresponding message is displayed in the **Message** window.

When you right-click anywhere in the Result window of the SQL Editor, a context menu opens which provides options for working with the results of a query. Note that the available options in the content menu change dynamically, depending on the command you used for retrieving the data in the Result window. The Result window can, therefore, be in three different states: Execution was successful, Stopped retrieval, or Editing mode.

Execution was successful

The ✅ Execution was successful message indicates that the SELECT query has been executed successfully. This enables the following commands in the toolbar and in the context menu.

📋	**Go to statement**		Jumps to the SQL Editor window and highlights the group of SQL statements that produced the respective result.
	Auto Size Columns		Adjusts the column width of all columns to provide an optimized view of text contained in the columns. Please note that the maximum column width for auto-sizing in the Result window is 250 pixels. In data cells containing a large amount of text, only the first part of the text is displayed followed by an ellipsis (...). You can, however, always drag the column border to show more text.
	Selection		Allows you to select the row or column in which the cursor is currently positioned, or the entire table. Choose the desired option from the sub-menu.

	Sorting		Opens a sub-menu to sort the table either ascending or descending and uses the column where the cursor is currently positioned as a sort key. To restore the default sort order, choose the corresponding option from the sub-menu.
⬚	**Copy selected cells**	**Ctrl+C**	Copies the selected cells to the clipboard. You can select several individual cells by holding down the **Ctrl** button and clicking the desired cells or select a consecutive number of neighboring cells by clicking a cell and pressing the **Shift** key while clicking a second cell. The content of the cells can be pasted as tabbed text into a text editor or spreadsheet calculator, or can replace existing cells in the result grid. This menu option is only available if data editing in the Result window is enabled.
	Copy selected cells with header		Copies the selected cells as well as the corresponding headers (i.e., the column names) to the clipboard.
⬚	**Show in Data Inspector Window**		Opens the [Data Inspector](#)[291] window where you can conveniently view the content of large data cells.
	Show result toolbar		Shows or hides the toolbar in the Result window of the SQL Editor window.
	Auto Hide		currently not available
	Hide		Hides the Results window. In order to show it again, click the **Toggle Result Window(s)** 🖩 button in the toolbar of the SQL Editor window or choose the menu option **SQL Editor	Result view**.

Stopped Retrieval

The ➡ Stopped Retrieval message indicates that the SELECT query has finished retrieving the first *n* rows. The query has been started by right-clicking tables, columns or views in the Online Browser and selecting **Retrieve data | First n rows** from the context menu. In this mode, the following additional options are available in the **Result** window:

▶+	**Retrieve next n rows**	Retrieves the next n rows from the query in the active SQL Editor window.
▶▶	**Retrieve outstanding rows**	Retrieves all the remaining rows from the query in the active SQL Editor window.

Editing Mode

The 🖉 Editing Mode message indicates that data in the Result window can be edited. The query has been started by executing a SELECT statement with the **Execute for Data Editing** command, or by right-clicking tables, columns or views in the Online Browser and selecting **Edit data** from the context menu. When the SQL Editor is in the Editing mode, the following additional options are available in the Result window:

	Paste	**Ctrl+V**	Pastes the content of the clipboard into the cell—and, if applicable, its neighboring cells—where the cursor is currently positioned. Note that if you use the **Edit cell** command before pasting, the entire content of the clipboard will be pasted into the selected cell as tabbed text.
	Paste as new rows	**Ctrl+Shift+V**	Adds new rows to the result grid and pastes the content of the clipboard into the new rows. Note that only content that has been copied using the **Copy selected cells** command can be pasted as new rows.
	Edit cell		Enables the data cell for editing and selects the content of the cell. This menu option is only available if no limitations with regard to editing records apply.
ø	**Set Null**		Resets the cell to the NULL value.
Def	**Set Default**		Resets the cell to the default value. This menu option is only available if a default value has been defined for the cell.
	Undo Changes for this Cell		Rejects the changes that have been made to a data cell but not yet committed to the database.
+	**Append a new row**	**Alt+Ins**	Adds a new line to the result grid, where you can add records [304] to the database table. The data must be committed to the database by clicking the Commit button. This menu option is only available if no limitations with regard to adding records apply.
×	**Delete row**		Deletes the row [307] in which the cursor is currently positioned. This menu option is only available if no limitations with regard to deleting records apply.

Other commands

In addition, the toolbar provides the following commands:

	Find	Opens a Find dialog box allowing you to search a particular string in the Result window.
	Maximize result view	Enlarges the size of the Result window to a maximum.
	Undo all changes	Rejects all changes that have been made in the result grid since the data has been selected from the database or since updated data has last been committed to the database, respectively. This toolbar option is only available if the Result window is in the Editing mode.
	Pin down	Pins the Result window so that it remains visible even if other queries are executed in the same window.

| | **Show Data in Chart Window** | Opens the Select columns dialog box in which you can choose the columns that are to be used as source for the chart. |

When the Result window is in editing mode and XML columns are present in the current grid, the following menu opens if you click the button:

Load XML Document from File...	Opens the standard Windows Open dialog box from where you can choose an XML file to load into the selected cell in the result grid.
Save XML Document to File...	Opens the standard Windows Save As dialog box where you can choose a location to save the content of the selected cell as XML file.
Assign XML Schema...	Applicable only for DB2 databases. Opens a dialog box where you can assign an XML schema [399] for validation of the XML file stored in the database field.

2.10 Design Editor

The **Design Editor** is a graphical user interface where you can view your database objects and their relations as graphical objects. A table including its columns, indexes, keys, and constraints is displayed in a so-called "table design". You can add new tables or edit existing ones or just drag a table into a Design Editor window to see what its structure looks like.
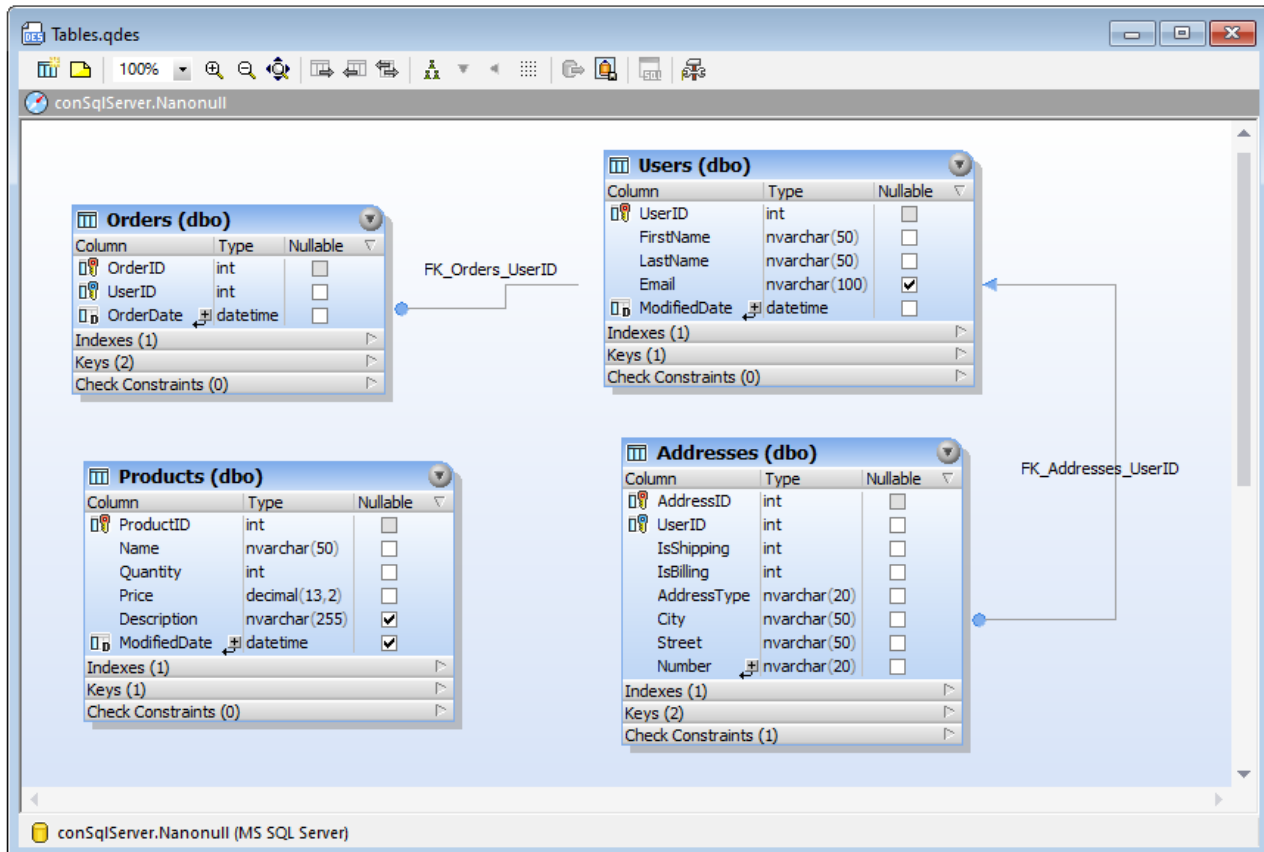
Design Editor
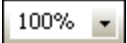
When you add new objects (tables or columns) to Design Editor or change the structure of existing ones, these changes are not immediately effective in the database. Any changes in the database design are recorded in the Database Structure Change Script 37 and can then either be executed or rejected.

When you right-click in an unoccupied area of the Design Editor, several general options are available in the context menu which are listed below. The corresponding Toolbar icons are shown to the left, and keyboard shortcuts to the right of the options if they are available.

Icon	Command	Shortcut	Description
	Create new Table	**Ctrl+T**	Creates a new table in the currently active Design Editor window and automatically assigns it to the active data

Icon	Command	Shortcut	Description
			source, that is, the data source which is selected in the Online Browser. See also Creating Tables [234].
	Add Sticky Note	**Alt+N**	Adds a Sticky Note to the Design Editor window.
	Select | Select User Tables	**Alt+T**	Selects all user tables that are contained in the active Design Editor window.
	Select | Select System Tables	**Alt+S**	Selects all system tables that are contained in the active Design Editor window.
	Auto Layout whole Diagram		Shows the design in a hierarchical layout where all relations between the tables are considered and the tables are laid out properly.
	Save Diagram as Image		Saves the currently active design as Portable Network Graphic (*.png).

In addition, the toolbar provides the following icons:

Icon	Command	Shortcut	Description
100% ▾			Changes the zoom factor of the Design Editor window; enter a value or select a zoom factor from the drop-down list.
	Zoom in	**Ctrl+Num +**	Increases the zoom factor of the Design Editor window by 10 percent. You can also zoom in by scrolling (with the scroll-wheel of the mouse) while keeping the **Ctrl** key pressed.
	Zoom Out	**Ctrl+Num -**	Decreases the zoom factor of the Design Editor window by 10 percent. You can also zoom out by scrolling (with the scroll-wheel of the mouse) while keeping the **Ctrl** key pressed.
	Zoom to Fit		Sizes the working area to include all tables in the current Design Editor window.
	Insert Referenced Tables		Inserts all related tables that are referenced by a constraint of the selected table.
	Insert Referencing Tables		Inserts all related tables that include a constraint referencing the selected table.
	Insert All Related Tables		Inserts both referenced and referencing tables.
	Expand selected Tables		Expands all tables that are selected in the current Design Editor window.

Icon	Command	Shortcut	Description
	Collapse selected Tables		Collapses all tables that are selected in the current Design Editor window.
	Snap to Grid	**Ctrl+G**	Shows/hides the grid and toggles the snap-to-grid function on and off.
	Export		Lets you export data from tables, see Exporting Database Data[406]. This command becomes enabled after you select one or more tables in Design Editor.
	Update Change Script		Starts the generation of a change script or updates it after changes occurred to the database structure. By default, the database structure change script is configured to be generated or updated automatically, so this button is disabled. To enable on demand generation of the database change script, select the **On demand** check box in Change Script[420] options.
	Show Options		Opens the Design Editor options[433] dialog box where you can customize the settings for the Design Editor.

For more information about exploring or designing the database structure in Design Editor, see Managing Database Objects[222].

2.11   Execution Target Bar

The Execution Target Bar is an optional toolbar which displays the data source and root object currently assigned to the SQL Editor window or—if the data source connection is active—the Design Editor.
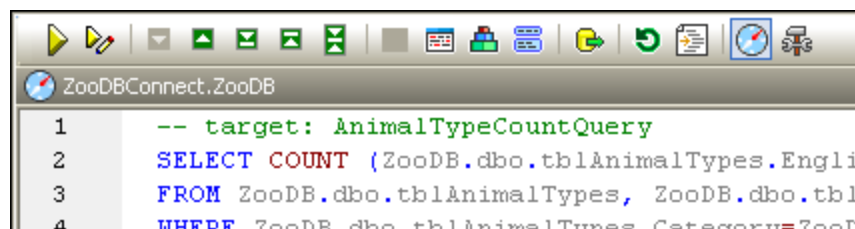
The Execution Target Bar is switched on for all SQL Editor windows by default. To hide the bar for individual

SQL Editor windows, click the **Show/Hide execution target bar** button in the SQL Editor toolbar. Note that disabling the Execution Target Bar for an SQL Editor window will deactivate it in all SQL Editor and Design Editor windows that you open subsequently.

When you move your mouse cursor over the Execution Target Bar, the different elements appear as hyperlinks, enabling you to jump the Properties window where you can change the respective element.

If the data source to which an SQL Editor window is assigned is offline, the Execution Target Bar provides a **Connect** button for conveniently establishing a connection.

For an SQL Editor window that has no data source defined, you can use the hyperlink functionality to jump to the Properties window if you want to specify a data source. Click the "Offline" text in the Execution Target Bar to access the **Data Source** drop-down list where you can select a data source for the SQL Editor window.
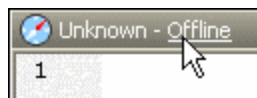
If the root object specified for the SQL Editor window differs from the root object of the data source the window is currently connected to, the execution target bar shows a tooltip providing a more detailed description of the problem when you place the mouse cursor over its icon. By clicking the **Synchronize** button you can set the SQL script's root object to the currently active root object of its data source.
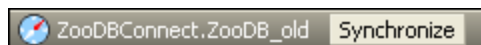
2.12   Menu Bar, Toolbars, and Status Bar

The **menu bar** contains the various application menus. The following conventions apply:

- If commands in a menu are not applicable in an editor or at a particular location in the document, they are unavailable.
- Some menu commands pop up a sub-menu with a list of additional options. Menu commands with sub-menus are indicated with a right-pointing arrowhead to the right of the command name.
- Some menu commands pop up a dialog that prompts you for further information required to carry out the selected command. Such commands are indicated with an ellipsis (...) after the name of the command.
- To access a menu command, click the menu name and then the command. If a sub-menu is indicated for a menu item, the submenu opens when you place the cursor over the menu item. Click the required sub-menu item.
- A menu can be opened from the keyboard by pressing the appropriate key combination. The key combination for each menu is **Alt+KEY**, where **KEY** is the underlined letter in the menu name. For example, the key combination for the **File** menu is **Alt+F**.
- Some menu commands can be selected directly by pressing a special shortcut key or key combination (**Ctrl+KEY**). Commands which have shortcuts associated with them are indicated with the shortcut key or key combination listed to the right of the command. For example, you can use the shortcut key combination **Ctrl+N** to open a new SQL Editor; the shortcut key **F2** to rename a database object.

Toolbars

The toolbars contain buttons that are shortcuts for commands found in the menus. The name of the command appears when you place your mouse pointer over the button. To execute the command, click the button.

Toolbar buttons are arranged in groups. In the Tools | Customize | Toolbars [476] dialog, or when right-clicking into the toolbar and selecting **Customize**, you can specify which toolbar groups are to be displayed. In the GUI, you can also drag toolbar groups by their handles (or title bars) to alternative locations on the screen. Double-clicking the handle causes the toolbar to undock and to float; double-clicking its title bar causes the toolbar to dock at its previous location.

Status Bar

The Status Bar is located at the bottom of the application window and displays (i) status information about the connection to data sources, the loading of database content, and the loading of files, and (ii) information about menu commands and command shortcuts in the toolbars when the mouse cursor is placed over these. If you are using the 64-bit version of DatabaseSpy, this is indicated in the status bar with the suffix (x64) after the application name. There is no suffix for the 32-bit version.

2.13   Arranging the Information Windows

All information windows can be docked/undocked by double-clicking the title bar. When docked, the auto-hide feature can be activated by clicking the drawing-pin icon in the title bar. When auto-hidden, the window is minimized as a tab at an edge of the application window. An auto-hidden window can be re-docked by rolling it out from the edge (by mousing over its tab) and clicking the drawing-pin icon in the title bar. When you click the little arrow that is located to the right of a data source connection or favorite item and shows the respective object in the Online Browser, DatabaseSpy will automatically hide the Project window and roll out the Online Browser to display the object.

Context Menu

The context menu can be accessed by right-clicking a window tab or title bar.
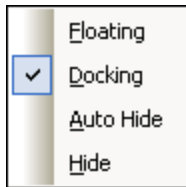
Click the required option to cause that window to float, be docked, or be auto-hidden or hidden.

Drag-and-drop

You can drag a window by its tab or title bar and place it at a desired location. Additionally, you can dock the window in another window or in the interface using placement controls that appear when you drag a window:

- When you drag a window over another window, a placement control appears. This control is divided into five placement sectors. Releasing the mouse key on any of these sectors docks the dragged window into the respective sector of the target window. The four arrow sectors dock the dragged window into the respective sides of the target window. The center button docks the dragged window as a tab of the target window. You can also dock a window as a tab in another window by dragging it to the tab bar and dropping it there.

- When you drag a window, a placement control consisting of four arrows appears. Each arrow corresponds to one side of the window. Releasing a dragged window over one of these arrows docks the window to that side.

You can also double-click the title bar of a window to toggle it between its docked and floating positions.

3   Tutorials

This section contains the following tutorials:

- "Nanonull" Database [53]: This tutorial runs you through the main features of DatabaseSpy. You do not need a running database server to follow the tutorial because it uses a file-based database (SQLite). You will learn how to create a file-based database from scratch, as well as how to perform basic data manipulation tasks from DatabaseSpy.
- "ZooDB" Database [72]: This tutorial provides more in-depth information about DatabaseSpy features. To follow this tutorial, you need access to a running SQL Server database where you have rights to create a demo schema and populate it with data.

All the required SQL scripts and other accompanying files for both tutorials are available in the **C:\Users\<username>\Documents\Altova\DatabaseSpy2023\DatabaseSpyExamples\Tutorial** folder, relative to your personal Documents [14] folder. The SQL grammar of the accompanying scripts is specific to SQLite and SQL Server, respectively. Nevertheless, the features highlighted in the tutorials are identical or similar across various database kinds supported by DatabaseSpy.

3.1   "Nanonull" Database

This tutorial shows you the basics of how to interact with a database using the DatabaseSpy graphical user interface. To follow this tutorial, no previous experience with databases is required, although basic knowledge of SQL (Structured Query Language) will be advantageous. The tutorial uses a file-based database (SQLite), for portability reasons. Most of the described procedures are either very similar or identical in other database kinds, so you can follow the tutorial even if you have little or no experience with SQLite.

The tutorial is organized into the following parts. It is recommended that you follow them sequentially.

- Create a New Database [53] - In this part, you will learn how to create a new SQLite database and how DatabaseSpy projects are useful.
- Design Database Tables [57] - This section shows you how to design database tables visually, using the Design Editor available in DatabaseSpy.
- Run SQL Scripts [60] - This section shows you how to run various SQL scripts against the database, generate SQL statements from existing tables, save scripts to files, and organize them inside the project for ease of access.
- Explore a Database [66] - This section shows you how to view the structure of objects inside a database. It also illustrates how to retrieve data from the database and edit it.
- Generate Queries From Templates [68] - This section illustrates how you can run database queries conditionally, and also how to generate SQL statements from SQL templates.

3.1.1   Create a New Database

This tutorial part shows you how to create a new SQLite database using DatabaseSpy. The steps are very similar for a Microsoft Access database, since it is also file-based. For databases which are hosted on a server, creating the database requires a different approach, see the "ZooDB" Database [72] tutorial.

To create a new SQLite database:

1. Run DatabaseSpy. The **Add a Data Source** dialog box opens. (You can open this dialog box at any time using the menu command **File | Create a Database Connection**.)
2. Select **SQLite** and click **Next**.
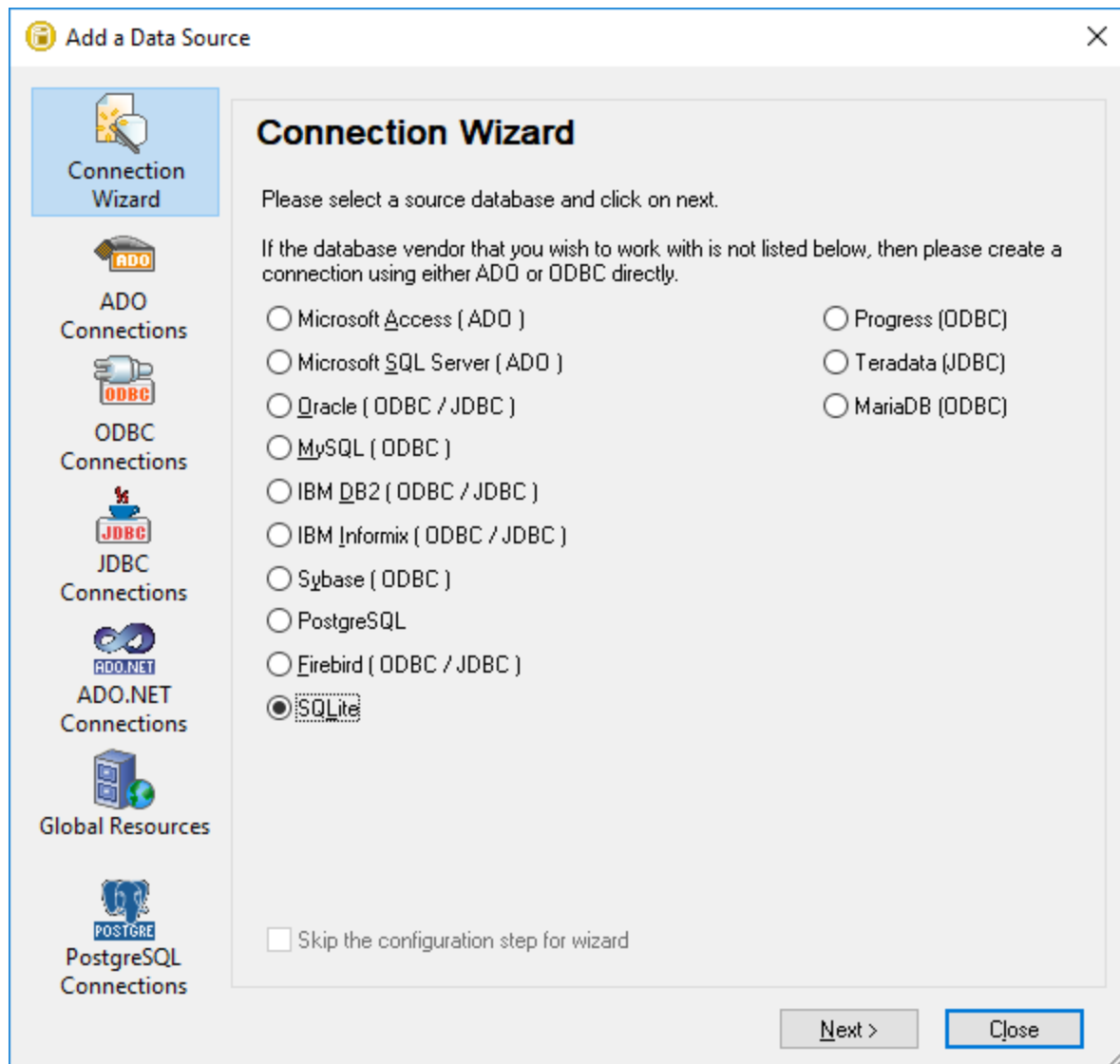
3. Select **Create a new SQLite database**.
4. Click **Browse** and select a destination path and name.
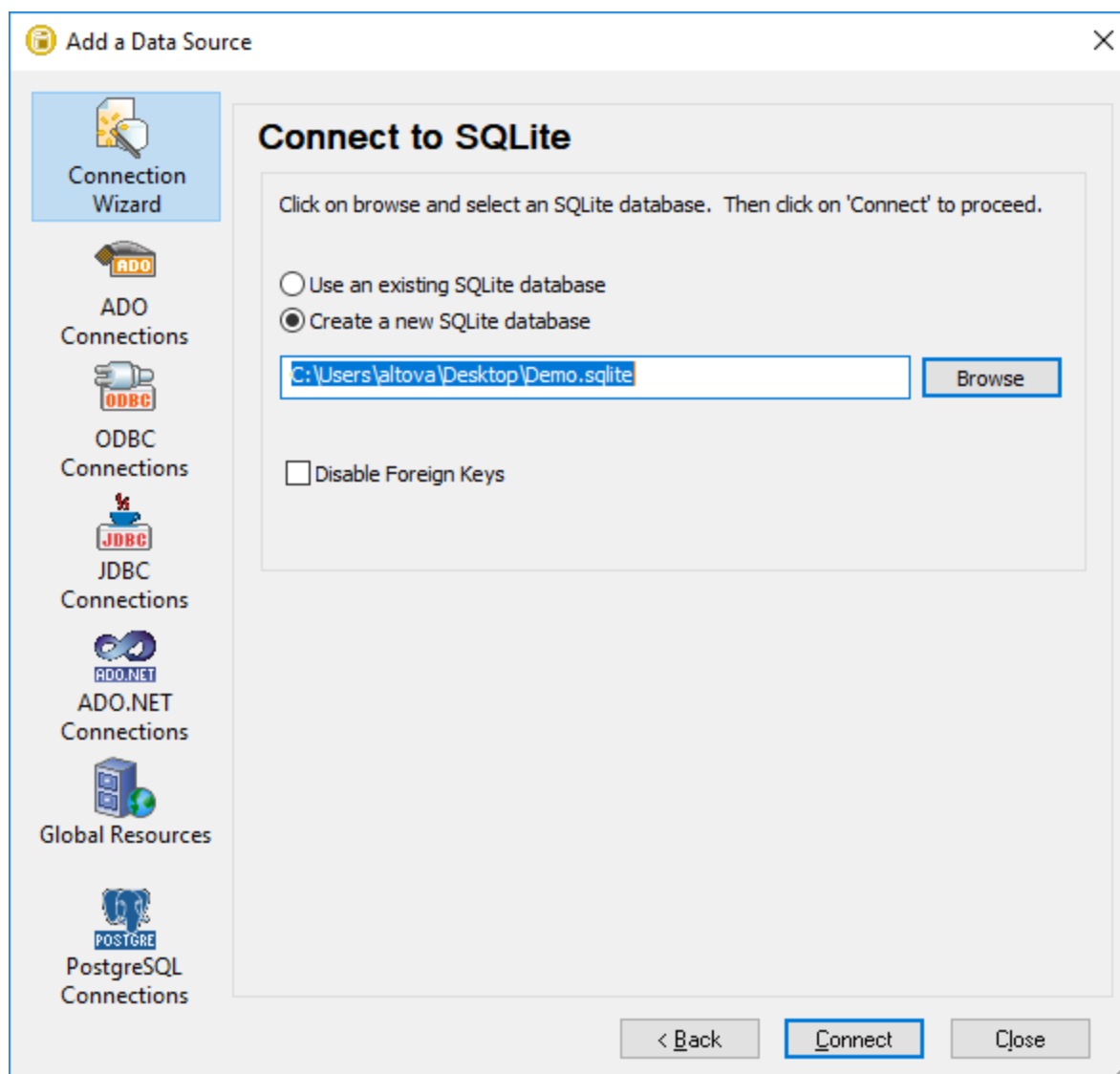
Note: There is no restriction as to what the extension of the SQLite database file should be. To recognize it easily, you might want to call the file **Demo.sqlite** , but you can also use other extensions, such as **.db** (note this documentation and the accompanying sample files use the **.sqlite** suffix by convention). For the scope of this tutorial, it is also recommended to enter a dedicated full path instead of just the database file name.

5.   Click **Connect**. You are now prompted to name the data source, for example:
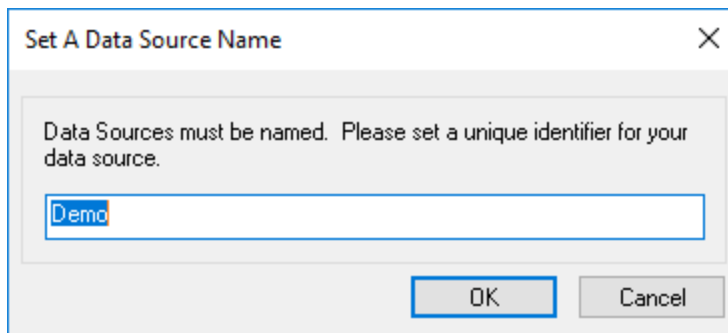
6. Leave the default value as is, and click **OK**.

A data source is now available in the Project window in the top left area of the DatabaseSpy graphical interface. Notice that the data source icon is green, which indicates that the connection to the database is currently active. To quickly connect to the database or disconnect from it, right-click the connection and select **Connect** (or **Disconnect**, respectively) from the context menu. When a data source is disconnected, its icon changes to.
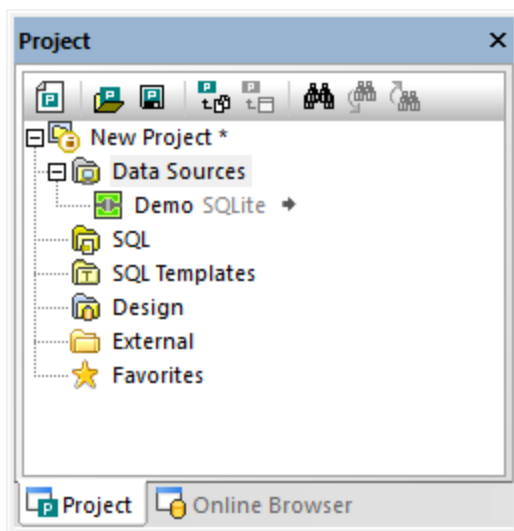
Project window

Also notice that, when a data source is selected in the Project window, its connection details are shown in the Properties window. This applies not only to data sources, but virtually to any object that you select from the user interface, including tables, columns, SQL scripts, and so on.

DatabaseSpy projects

Projects are a convenient way to organize database connections and database-related files such as SQL scripts. Saving data as a DatabaseSpy project will make it easier for you to connect to the same database in future, since you won't have to go through the database connection wizard steps again.

To save the DatabaseSpy project created so far, do one of the following:

- On the **File** menu, click **Save Project As**.

- Click the **Save Project** 🖫 button in the Project Window.

You can also conveniently add multiple database connections to the same project. To do this, right-click the "Data Source" folder in the Project Window, and select **Add a New Data Source** from the context menu.

3.1.2   Design Database Tables

So far in this tutorial, you have created an empty SQLite database. With DatabaseSpy, there are various ways to create database objects or modify existing ones. This part of the tutorial shows a simple approach to quickly create a new table. The new table will supposedly store some reports, and it should consist of the following columns:

- The identifier ("id") of the report, of type INTEGER. This column must have a primary key which ensures the value of each row is unique.
- The report name ("name"), of type TEXT.

Step 1: Create a new table

To create the new table:

1. Make sure that the data source is connected 🔛 in the Project window. Otherwise, right-click the data source, and select **Connect**.
2. Click the **Show in Online Browser** 🔵 button that appears next to the connected data source. (Alternatively, click the **Online Browser** tab.)
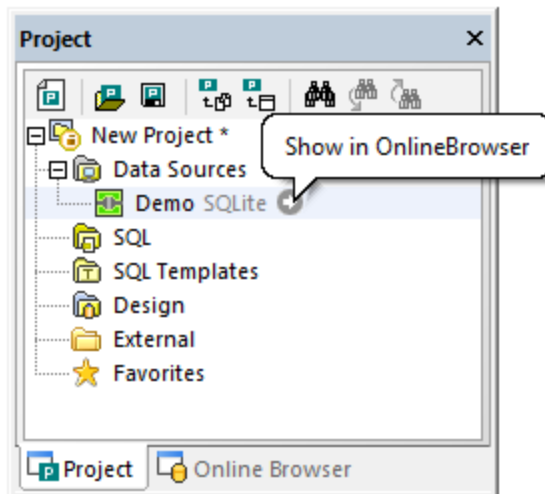
3. In the Online Browser, right-click the **main** 🟡 database, and select **Create Table** from the context menu. This action has the following outcome:

   - A dialog box appears and informs you that DatabaseSpy has created a change script for the current action. Click **OK** to dismiss the dialog box.
   - A new table called "newtable" appears with the "Tables" folder. Notice that the table has the ➕ icon overlaid, which indicates that it is a new uncommitted database object. Likewise, the database has the ➕ icon overlaid, which indicates that some of its child objects have uncommitted changes.

· No changes are committed to the database at this stage. The database change script generated so far is displayed in the [Database Structure Change Script Window](#) [37].
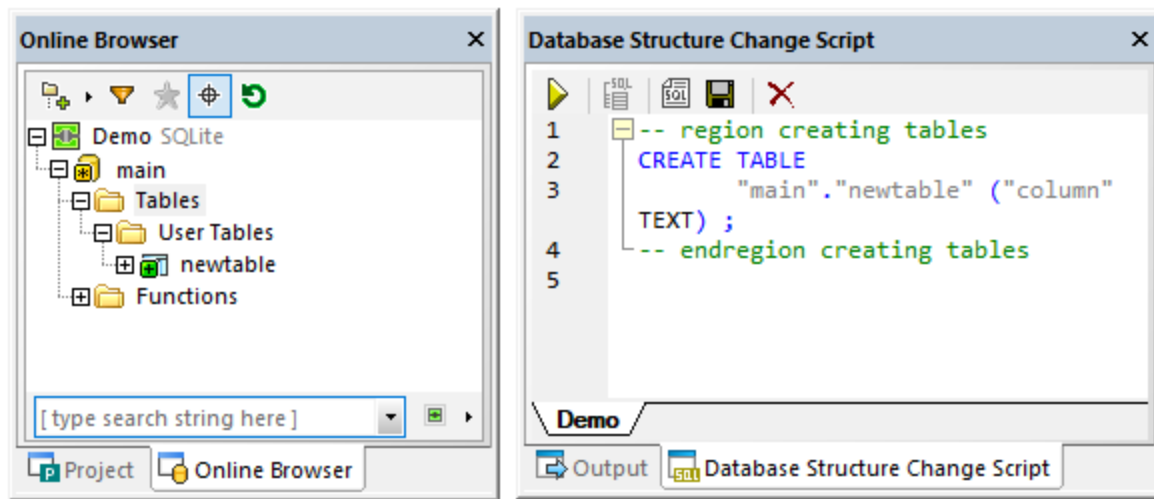
4.   Select the table and press **F2** to rename it to "reports". You can also right-click the table and select **Rename table** from the context menu. You will notice that the change script is updated accordingly.

By default, the script will keep changing as you are changing the structure of the database in subsequent steps; therefore, you might want to select the **Don't show this message again** option.

Step 2: Add columns

Each time when you create a new table from DatabaseSpy, it has one column by default. Recall that, in this example, the requirement was that the first column must be named "id" and it should be of type INTEGER. It must also be a primary key. To achieve this, take the following steps:

1.   Select the column and then press **F2** to enter the new name ("id").
2.   To change the column's type, first select the column and then change the **Type** property to INTEGER in the Properties Window.
3.   Also, make sure to clear the **Nullable** property. This makes it possible to add the primary key to the column in a subsequent step.
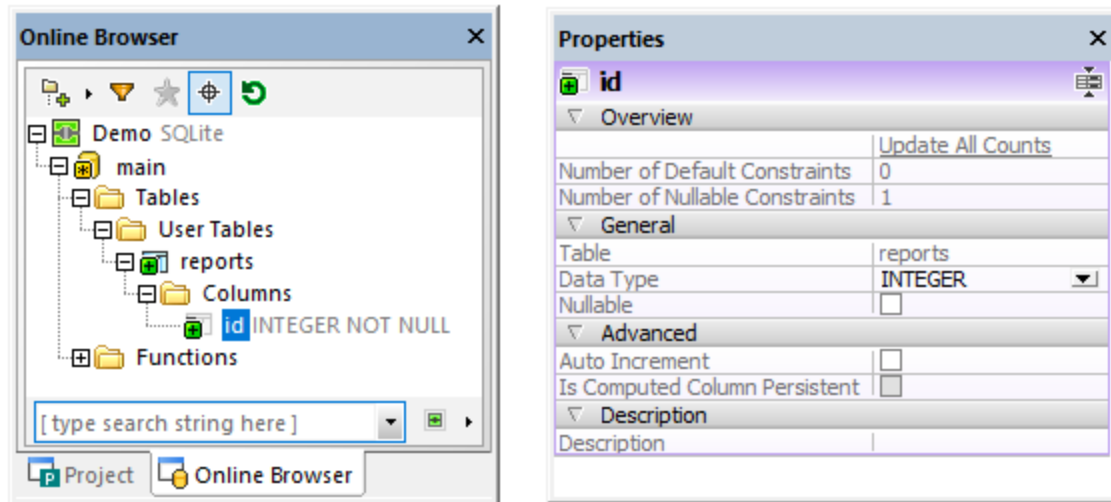
With other database types, you might want to set additional properties of the column, like **Size** or **Auto-increment**. The database used in this example is a SQLite database, so it does not need an **Auto increment** property explicitly, because the first column in the table is of type INTEGER and has a primary key.

4. To add the primary key on the "id" column, right-click the column and select **Create Primary Key** from the context menu. The outcome is that a "Keys" folder is added to the table, which stores the new primary key.
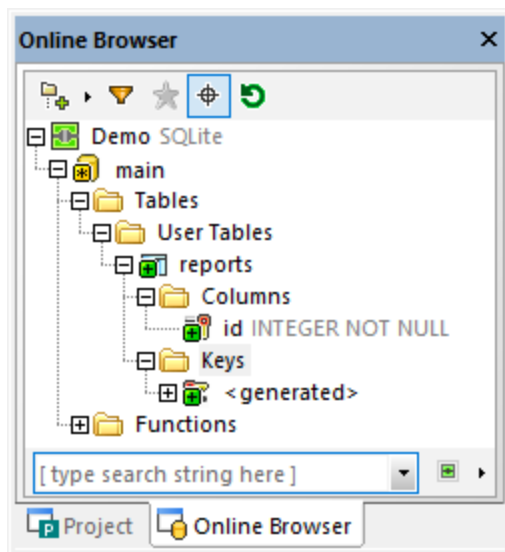
The new primary key has the name **<generated>** by default; this indicates that the key name will be generated by the database when you commit the script. Note that, with SQLite databases, the name of primary keys is always shown as **<generated>** in DatabaseSpy, even after the script is committed to the database. Other database kinds may support changing the **<generated>** name to a custom name and displaying it as such after the script was committed, see also Reference of Generated Names[277].

Using the same steps as above, add a new column called "name" of type TEXT.

Step 3: Commit the change script

The table design created so far has not been committed to the database yet. To commit the changes to the database, click the **Execute the Generated SQL Change Script** ▷ button in the Database Structure Change Script window.
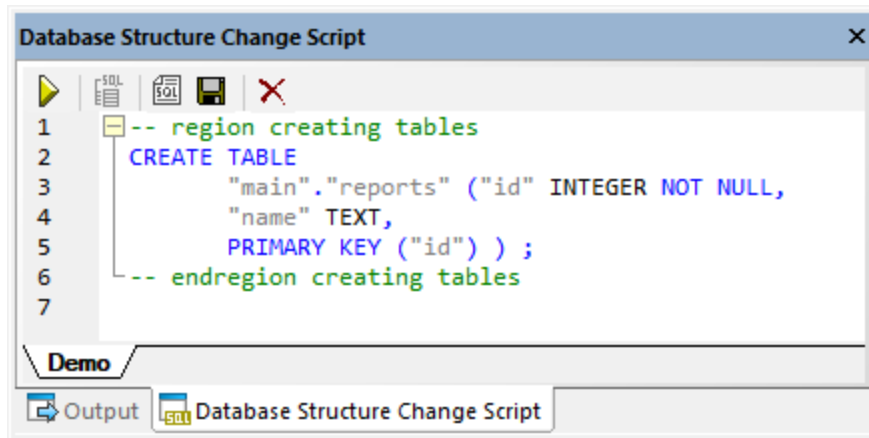
DatabaseSpy runs the script against the database, and informs you about the result. The new table is now visible in the Online Browser.
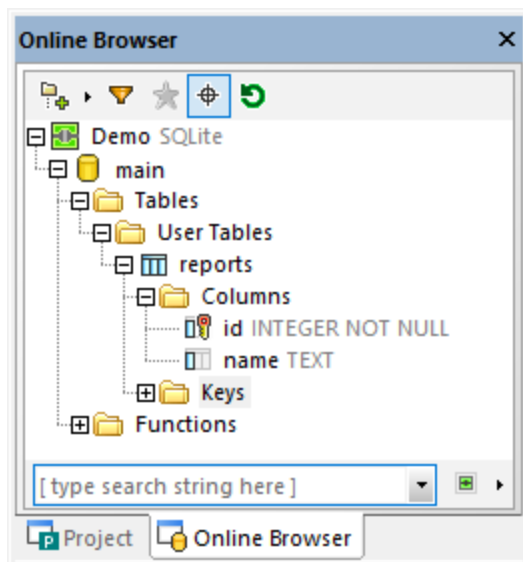
This concludes the design part of this tutorial. For more in-depth information about designing database objects, see Managing Database Objects [222].

3.1.3   **Run SQL Scripts**

In the previous part of this tutorial, you have created a table by using context menu commands in the Online Browser. An alternative way to interact with your database is by means of SQL (Structured Query Language) scripts. SQL scripts can be used to create the structure of the database, perform various operations against

the database (for example, populate it with data), and also change or delete the database structure. Below, you will learn how to run some sample SQL scripts against the demo database you created so far.
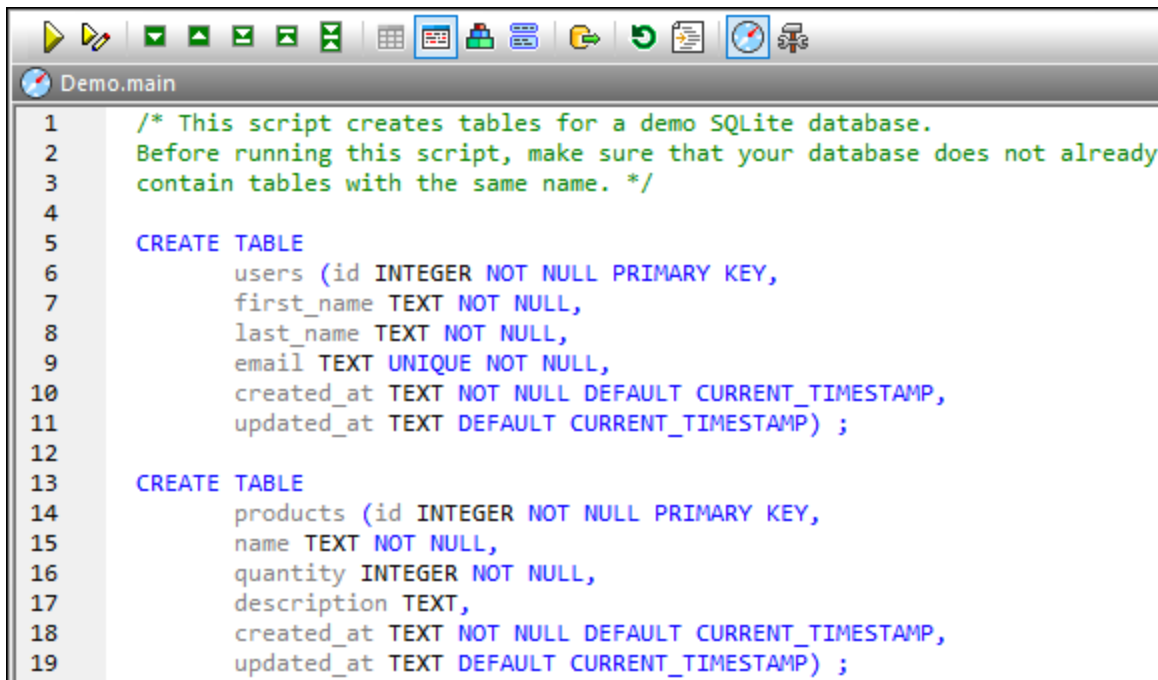
Let's first create the structure of all the tables in the database, as follows:

1. Connect to your demo SQLite database created so far.
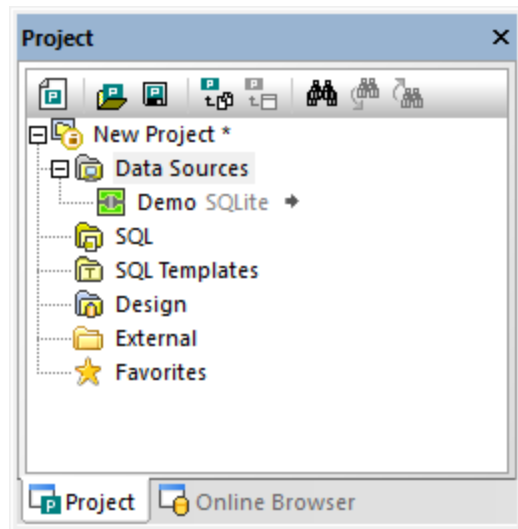
2. On the **File** menu, click **Open | Open File**, and browse for the **NanonullCreateTables.sql** file from the **C:\Users\<username>\Documents\Altova\DatabaseSpy2023\DatabaseSpyExamples\Tutorial** folder.

The scripts now opens inside the SQL Editor in the main pane.

SQL Editor

The SQL Editor is the area where you can view, edit, and execute SQL statements. The demo script illustrated above will create a few tables in the database.

Note: A new SQL Editor window is automatically associated with the data source that is active in the Online Browser [24] or Project Window [20]. The active data source is displayed at all times in the Execution Target Bar [49]. That is, all commands issued from SQL Editor will affect this particular data source only. If there are multiple data sources, the SQL Editor belongs to the one which was last clicked before you opened the SQL Editor. Any subsequent SQL Editor windows that you open will inherit the currently active data source.

To run the SQL script against the database, do one of the following:

- Click **Execute** ▷ in the toolbar at the top of SQL Editor.
- On the **SQL Editor** menu, click **Execute**.
- Press **F5**.

DatabaseSpy runs the script and informs you about the execution result in the Message window, for example:
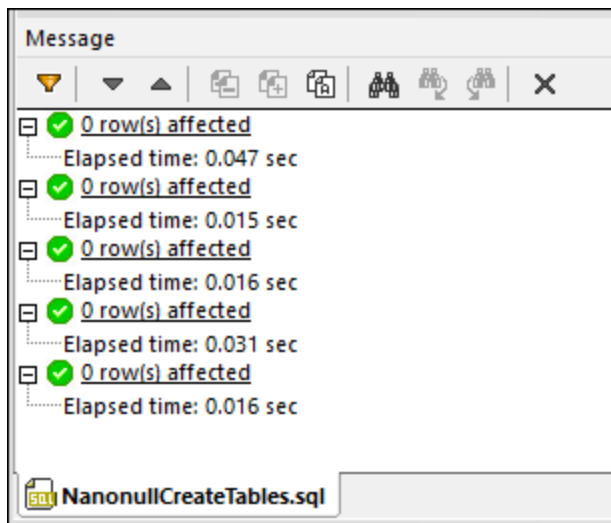
Now click the Online Browser tab in the Project window, and then click **Refresh** ↺ . You will notice that several tables have been added to the database.
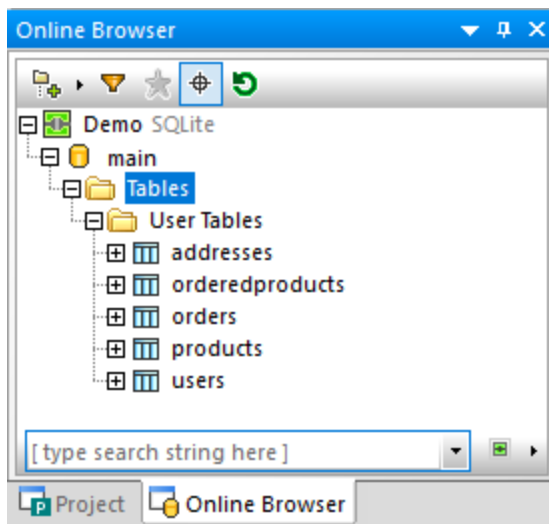
So far, you have created the database structure; however, the tables are empty. To quickly view the number of rows in each table, click the "count" link which appears when you move the mouse over any table in the Online Browser.

You can populate tables with some data as follows:

1. On the **File** menu, click **Open | Open File**, and browse for the **NanonullPopulateTables.sql** file from the **C:\Users\<username>\Documents\Altova\DatabaseSpy2023\DatabaseSpyExamples\Tutorial** folder.
2. Click **Execute** ▷.

If you go back to Online Browser now and check the count of rows in each table, you will notice that the count of rows has changed.

Let's now automatically generate a new script which retrieves some data from the database:

· In the Online Browser, locate a table, right-click it, and select **Show in new SQL Editor | SELECT** from the context menu.

An SQL statement is automatically generated based on the structure of the selected table, and displayed in the SQL Editor window. Click **Execute** ▷ to run the generated script and view the table data in a read-only grid. Click **Execute for data editing** 🗎 if you intend to modify data after retrieving it.

You can also create a SQL script manually, and add it to the current DatabaseSpy project, for example:

1. Open a new SQL Editor window. This can be done in one of the following ways:
   o On the **File** menu, click **New | SQL Editor**.
   o Click the SQL Editor toolbar button.
   o Press **Ctrl+ N**.
2. Type your script contents, for example: `SELECT * FROM users;`
3. Click inside the SQL Editor to make the current script active.
4. In the Project window, right-click the "SQL" folder and select **Add Active File to Project** from the context menu. When prompted to save the file, choose a suitable location for it (for example, the same folder as the DatabaseSpy project file).

Existing SQL scripts created outside DatabaseSpy can also be added to the project, as follows:

1. Make the Project window active.
2. Right-click on a folder inside the Project window, select **Add Files to Project**, and browse for the existing SQL scripts.

The "SQL" folder from the Project window is meant to help you easily group and access all SQL scripts pertaining to the project. However, there is no restriction that any SQL script must be specifically stored inside this folder. You can create additional folders or subfolders and relocate the scripts as necessary. To achieve this, use the commands available in the [Project Window](20).

Note: Should you need to delete all the tables from the demo database, a third script called **NanonullDropTables.sql** is available in the **C:\Users\<username>\Documents\Altova\DatabaseSpy2023\DatabaseSpyExamples\Tutorial** folder for this purpose.

3.1.4   Opening the Tutorial Project

The remainder of this tutorial uses a sample DatabaseSpy project to illustrate various database tasks. The sample project is available at the following path: **C:\Users\<username>\Documents\Altova\DatabaseSpy2023\DatabaseSpyExamples\Tutorial\NanonullEnterprise.qprj**. To open the sample project in DatabaseSpy, do one of the following:

· On the **File** menu, click **Open | Open Project**.
· Click the **Open Project** 🗁 button in the Project window.
· Double-click the project file in Windows Explorer, and close the database connection dialog box which appears.

The sample project contains a connection to a SQLite database which is by default inactive. To connect to the data source, right-click it in the Project window, and select **Connect** from the context menu.
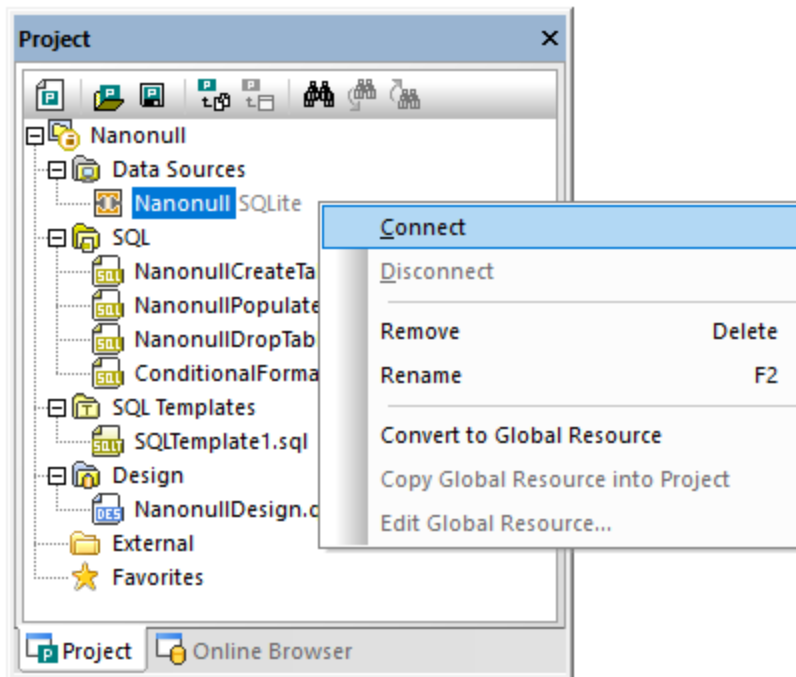
Project Window

Once you connect to the data source, the contents of the project can be viewed from the Project Window illustrated above, while the structure of the database can be viewed from the Online Browser [24]. The Online Browser is by default available as a tab of the Project window, but it can also be undocked and pinned to another location if necessary, see also Arranging the Information Windows [51].
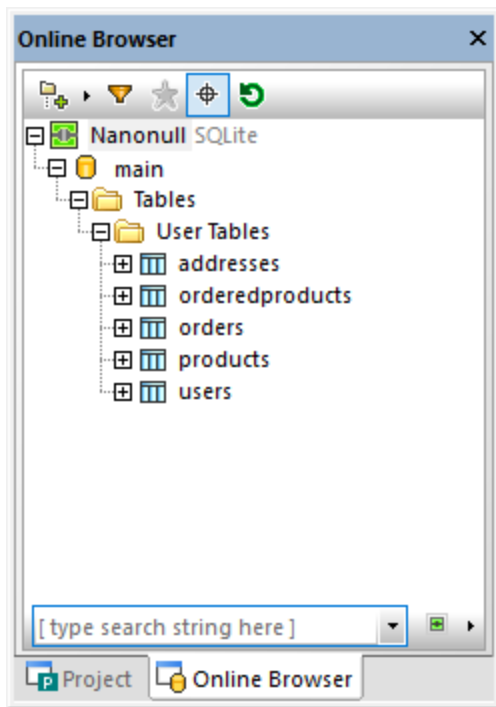
Online Browser

Note: For Online Browser to be populated, an active (connected) data source must exist in your project. By
default, disconnected data sources are not displayed in the Online Browser. If you want data sources
to be displayed in Online Browser regardless of whether they are connected or not, first select the data
source in the Project Window [20] , and then select the **Always show in Online Browser** check box in
the Properties Window [30] .

3.1.5   Explore a Database

This tutorial part shows you how to explore the contents of an existing database, and modify some data. To
this end, let's open the demo DatabaseSpy project which already includes a ready SQLite database populated
with some data, see Opening the Tutorial Project [64] .

Since the demo database is a SQLite database, the Online Browser displays a fixed database root object
called "main". Other database kinds may display multiple root objects (databases or schemas, as applicable).
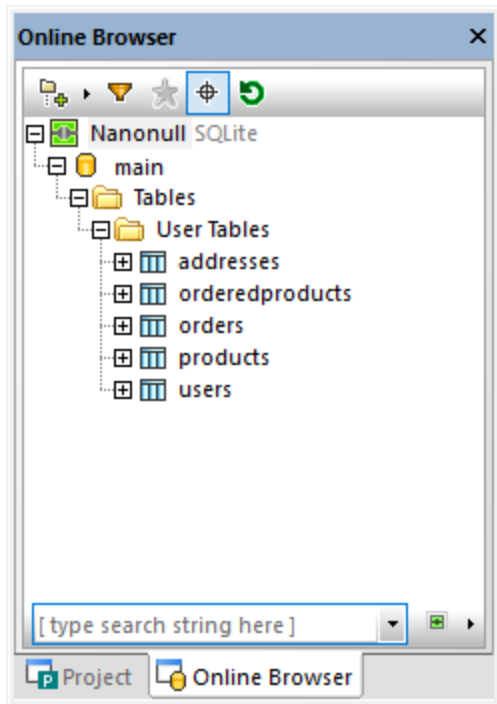
You can explore the database by expanding folders of interest in the Online Browser. The actual folder structure depends on the database kind. For the scope of this tutorial, you can expand each table in order to view its columns, as well as other objects, such as primary keys, constraints, indices, and so on. Keep in mind that the Properties window displays at all times detailed information about the currently selected object.

To view the structure of one or more tables as a diagram, open a new Design Editor (**Ctrl+D**) and drag the tables from the Online Browser into the editor area.

To view the actual data inside a table (remember this will display data as read-only):

1. Right-click the table in the Online Browser.
2. Select **Retrieve Data | All rows** or **Retrieve Data | First 200 rows** from the context menu. As a side note, the number of rows for the option **Retrieve Data | First N rows** can be configured from **Tools | Options | SQL Editor**.

If you want to open data for editing as opposed to just viewing it, right-click the table in Online Browser and select **Edit Data**. Alternatively, select **Retrieve Data | N rows**, and then run the command **Execute for data editing** . Selecting or updating data stored in the database is also possible if you run SQL queries in the SQL Editor window, see Run SQL Scripts⁶⁰.

Regardless of the approach you choose to retrieve data, the content retrieved from the database is displayed in a Result view, as a grid.

Result view

To be able edit data, make sure that you retrieved it using the **Edit Data** command (or the **Execute for data editing** command 🖉). In this case, you can use the commands available at the top of the Results grid to add new rows ➕ or delete ❌ existing ones. To modify a cell value, double-click it and enter the new value.

Note: To protect the integrity of the database, various data editing restrictions may apply, depending on the constraints defined for the table. For example, it is not possible to edit a column defined as a primary key, or enter text inside a numeric column. DatabaseSpy performs some basic data integrity validation checks as you modify data. If the integrity check fails, the cell background color changes to brick red. A second level of data integrity checks is provided by the database, when you actually commit the changes. If the changes are rejected by the database for any reason, an error is displayed in the Messages window, and data is not committed to the database.

Any changes made to the grid are not immediately reflected in the database. To update the database, click the **Commit** button, which deploys all changes to the database. In case you change your mind and would like to revert all the changes to the grid before committing them to the database, click the **Undo all** 🗑 button.

Various options applicable to the Result view, including display and data editing options, can be configured from **Tools | Options | SQL Editor | Result View**.

You have now learned the basics of retrieving and editing database data. For more information about working in the Result grid, see Retrieving and Editing Data [284].
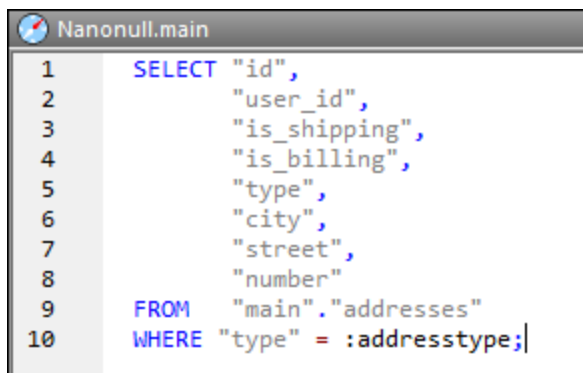
3.1.6   Generating SQL Statements

With DatabaseSpy Enterprise Edition, in addition to standard SQL statements, you can also run special kinds of SQL statements that are referred to as "SQL templates".

If you haven't done that already, first open the demo **NanonullEnterprise.qprj** project, see Opening the Tutorial Project [64]. You will find a demo SQL template called **SQLTemplate1.sql** in the SQL Templates folder inside the Project Window [20].

To run the demo SQL template:

1. Double-click the **SQLTemplate1.sql** file to open it in the SQL Editor.

```
 Nanonull.main
 1      SELECT "id",
 2             "user_id",
 3             "is_shipping",
 4             "is_billing",
 5             "type",
 6             "city",
 7             "street",
 8             "number"
 9      FROM   "main"."addresses"
10      WHERE "type" = :addresstype;
```

2. Click **Execute** ▷. A dialog box is now displayed, which prompts you to enter a parameter value.
3. Double-click the row under **:addresstype**, type "home" as parameter value, and click OK.

```
 Parameter values                                    ✕

 ╔═══════════════════════════════════════════╗  ✕
 :addresstype
 home                                          ▼

                                               OK    Cancel
```
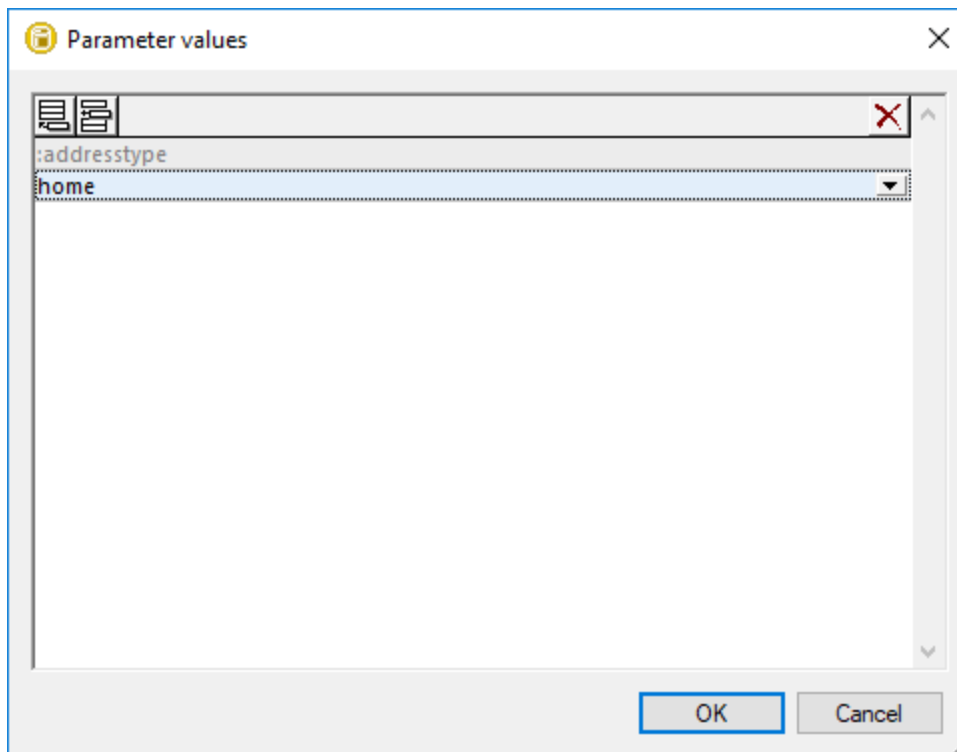
Notice that only addresses of type "home" (fifth column) are retrieved from the addresses table.

Result1

	id	user_id	is_shipping	is_billing	type	city	street	number
1	2	1	0	1	home	Bridgedell	Olive Street	6
2	3	3	1	1	home	Roseford	Evergreen Lane	34
3	5	6	1	1	home	Johnson City	Franklin Avenue	11
4	6	7	1	1	home	North Kingstown	Beach Alley	5
5	7	8	1	1	home	Merrowmeadow	Freybeach Street	85
6	9	12	1	1	home	Elfville	Creek Road	3
7	10	13	1	1	home	Roseford	Bowman Ave.	853
8	12	17	1	1	home	Bridgedell	Smith Road	7
9	13	18	1	0	home	Roseford	Wood Street	7
10	15	20	1	1	home	Mechanicsville	Vine Street	9065
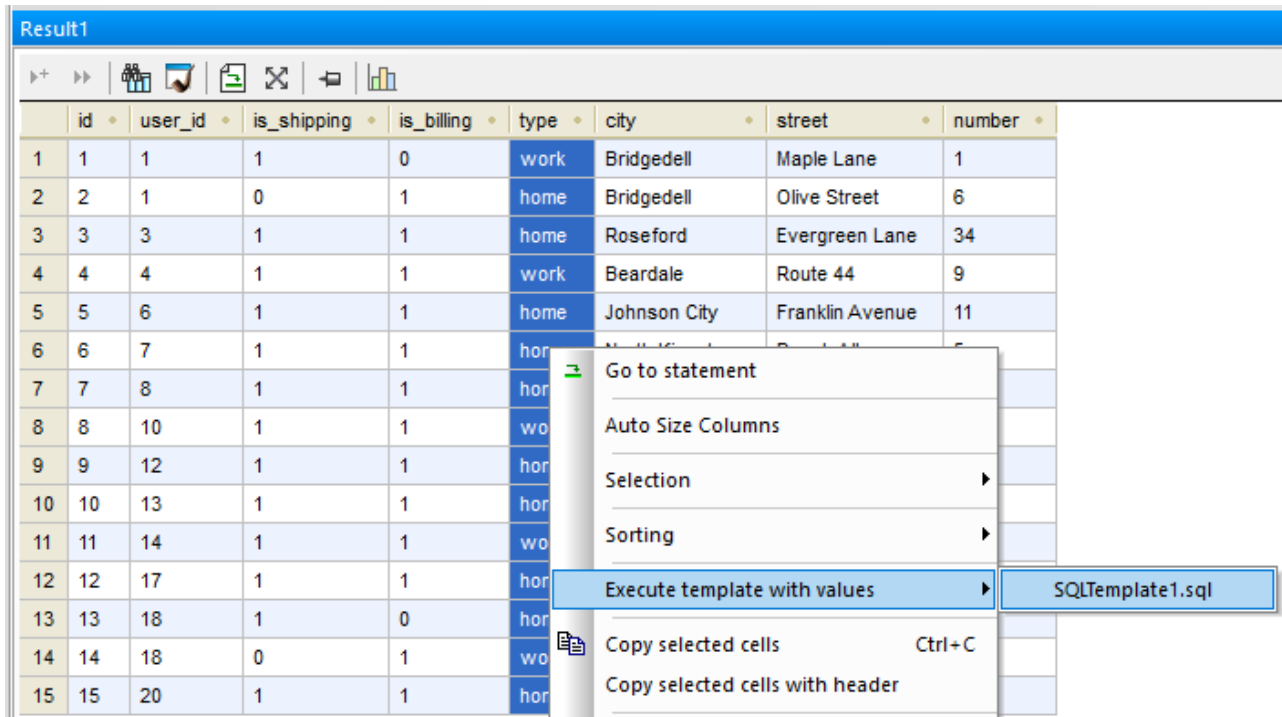
4. Click **Execute** ▶ and this time enter "work" as parameter value. You will notice that only addresses of type "work" are now retrieved.

Result2

	id	user_id	is_shipping	is_billing	type	city	street	number
1	1	1	1	0	work	Bridgedell	Maple Lane	1
2	4	4	1	1	work	Beardale	Route 44	9
3	8	10	1	1	work	Barrowedge	Penn Street	8
4	11	14	1	1	work	Beardale	Iroquois Street	98
5	14	18	0	1	work	Johnson City	Thorne Lane	9677

The steps above essentially illustrate how a typical SQL template works: it provides the option to make certain parts of the SQL query conditional. The conditional parts are called "parameters" and are marked with a colon character (see the `:addresstype` parameter inside the SQL statement). When you attempt to run such a query, DatabaseSpy will prompt you to fill in the parameter value (or multiple values, if applicable) as required.

Apart from executing SQL statements conditionally, you can also use SQL templates to generate multiple SQL statements with a click of a button. Let's try the following example:

1. Right-click the `addresses` table in the Online Browser, and select **Retrieve Data | All Rows** from the context menu.
2. In the Result grid, click the header of the `type` column to select all data in it.
3. Right-click the selection and choose **Execute template with values | SQLTemplate1.sql**.

This command calls the template described previously as many times as there are rows inside the column, and supplies the value of each row as a parameter to the SQL template. As a result, multiple SQL SELECT statements are generated and displayed in the SQL Editor. DatabaseSpy also executes all generated statements, and the result of each is visible in a separate Result grid.

The SQL Template run in this example is only meant for demo purposes. In a real life scenario, you might want to apply the same technique in order to generate more useful statements, for example, ones that insert data into tables. For more details about SQL templates, including an example of how to generate INSERT statements, see SQL Templates [382].

3.2   "ZooDB" Database

In this tutorial, you will use DatabaseSpy to interact with a SQL Server database that models a zoo. Step-by-step instructions, SQL scripts and other files are provided to help you complete this task.

The tutorial is organized into the following parts, which must be followed sequentially.

- Setting up the "ZooDB" Project [73] - Shows you how to create the project that groups together all the required script files and the database connection.
- Adding Tables to the Database [80] - Illustrates various ways to create or design the database structure.
- Defining Constraints [88] - Provides instructions for creating the following constraint kinds: unique, check, default, and foreign key constraints.
- Adding Data to the Database [94] - In this tutorial section, you learn how to populate the database with data. You can do this by importing data from XML or CSV files, by running an INSERT script, or by entering data manually.
- Querying the Database [104] - This tutorial section illustrates various ways to query the "ZooDB" database in order to view, update, or delete data. This includes running SQL scripts and generating on demand SQL statements. In addition, you can always select, update, or delete data manually.
- Exporting Database Data [112] - Shows you how to export data to other formats such as XML.

Prerequisites

Before you can start with this tutorial, a new sandbox database must be created in Microsoft SQL Server. You can accomplish this task either with Microsoft SQL Server Management Studio, or from Visual Studio. To do this, connect to a database server running Microsoft SQL Server, and follow the instructions specific to your SQL Server database administration tool to create the database. For example, the following image illustrates creating a new database on a network SQL Server from Visual Studio 2015:
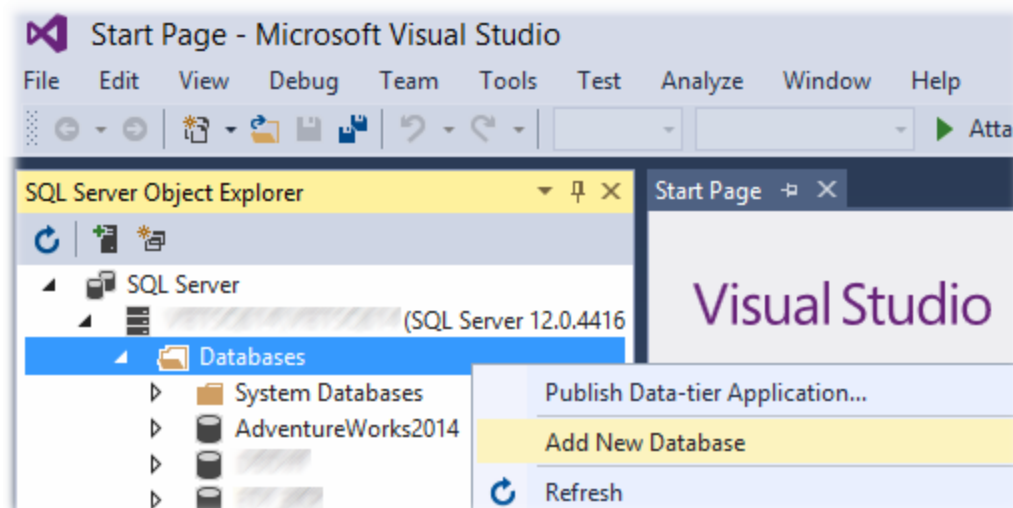
Notes:

- In this tutorial, the database name is "ZooDB".
- Your SQL Server user account requires "dbcreator" privileges to create a database. Contact your database server administrator if you do not have the respective rights.

- The new "ZooDB" database must be empty. All tables, relationships, and data will be added to this database in the subsequent steps of this tutorial.

Tutorial example files

The table below lists the files used in this tutorial. These files are available at the following path: **C: \Users\<username>\Documents\Altova\DatabaseSpy2023\DatabaseSpyExamples\Tutorial**. Most of them are scripts in SQL Server grammar that create the database structure and then populate some of the tables with data. Note that the files are listed in the order in which they are meant to be run against the database, as you will discover by following the tutorial.

File	Purpose
create_ZooDB_tables.sql	Generates multiple tables in the "ZooDB" tutorial database.
create_ZooDB_relationships.sql	Creates foreign key relationships between database tables.
tblAnimalTypes_data.sql	Contains INSERT queries that populate two of the tables with data.
animalType_queries.sql	Contains SELECT queries that retrieve data from two tables.
tblZookeepers.txt	Contains data in CSV format that can be used to populate one of the database tables with data.
dbo.tblFeedSuppliers.xml	Contains data in XML format that can be used to populate one of the database tables with data.

Several other CSV and XML files are available if you want to optionally populate all of the database tables with data, see Populating Other Tables (Optional)[100].

3.2.1   Setting up the "ZooDB" Project

In this section of the tutorial, you will learn how to create a project in DatabaseSpy and connect to a database. Specifically, you will learn how to do the following:

- Create a new project and add a data source connection
- Add SQL files to the project
- Save the project and the data source
- Define the project startup options

3.2.1.1 Connecting to the Database

As mentioned in Prerequisites[72], a new empty database called "ZooDB" must first be created in SQL Server. Once the database has been created, you can connect to it from DatabaseSpy, as shown below. The Connection Wizard allows you to establish a connection to existing databases of the most commonly used database types, such as Microsoft Access, Microsoft SQL Server, Oracle, MySQL, IBM DB2, Sybase, and PostgreSQL.

To connect to the tutorial database:
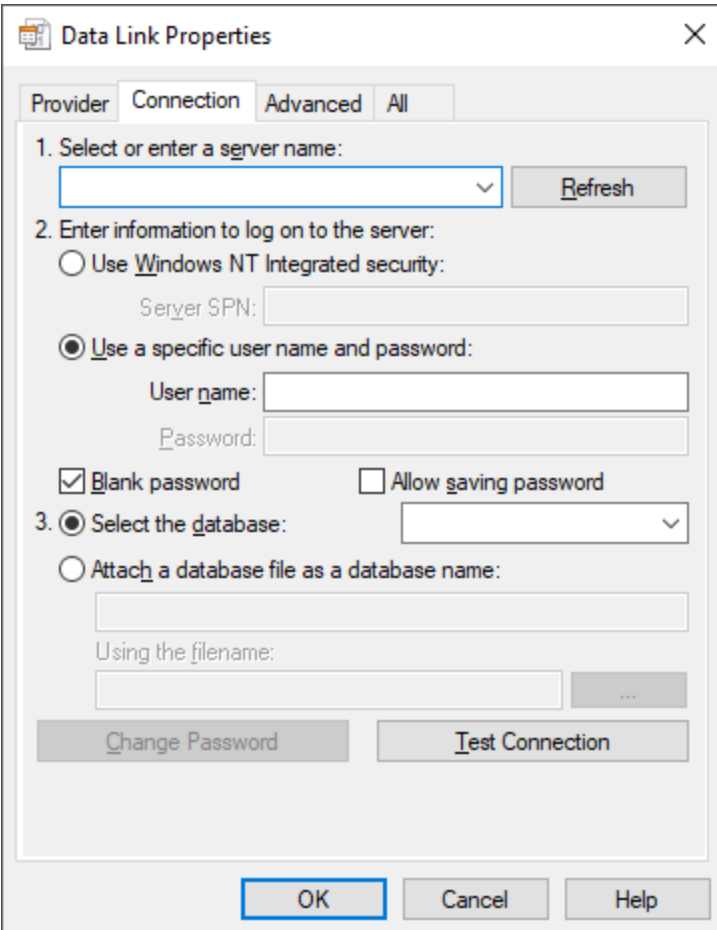
1. Do one of the following:

    *   Click the ▣ **Connect to a database** toolbar button.
    *   On the **File** menu, click **Create a Database Connection**.
    *   Press **Ctrl+Q**.
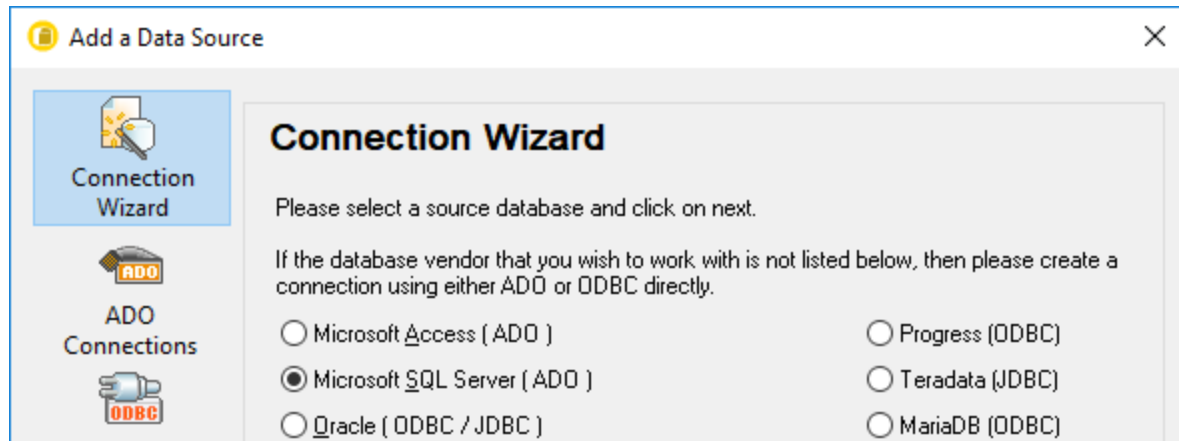
2. Select **Microsoft SQL Server (ADO)**, and then click **Next**. The list of available ADO drivers is displayed. In this example, the **SQL Server Native Client 11.0** is used as provider. You can also select the **Microsoft OLE DB Provider for SQL Server** provider.
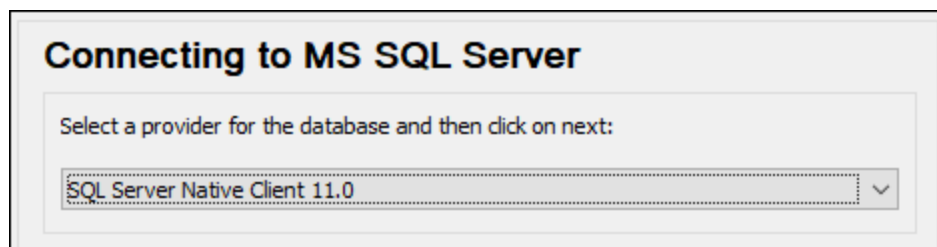
3. Click **Next**. The Data Link Properties dialog box appears.

4. Select or enter the name of the database server, for example, **SQLSERV01**. If you are connecting to a named SQL Server instance, the server name looks like **SQLSERV01\SOMEINSTANCE**. To view the list of all servers on the network, expand the drop-down list.
5. If the database server was configured to allow connections from users authenticated on the Windows domain, select **Use Windows NT integrated security**. Otherwise, select **Use a specific user name and password**, clear the **Blank password** check box, and enter the credentials in the relevant boxes.
6. Select the database to which you are connecting (in this example, **ZooDB**).

7. To test the connection at this time, click **Test Connection**. This is an optional, recommended step.

Even if the connection test is successful at this time, you must still persist the security info, see the next step.

8. Click the **All** tab, and change the **Persist Security Info** property to **True**.
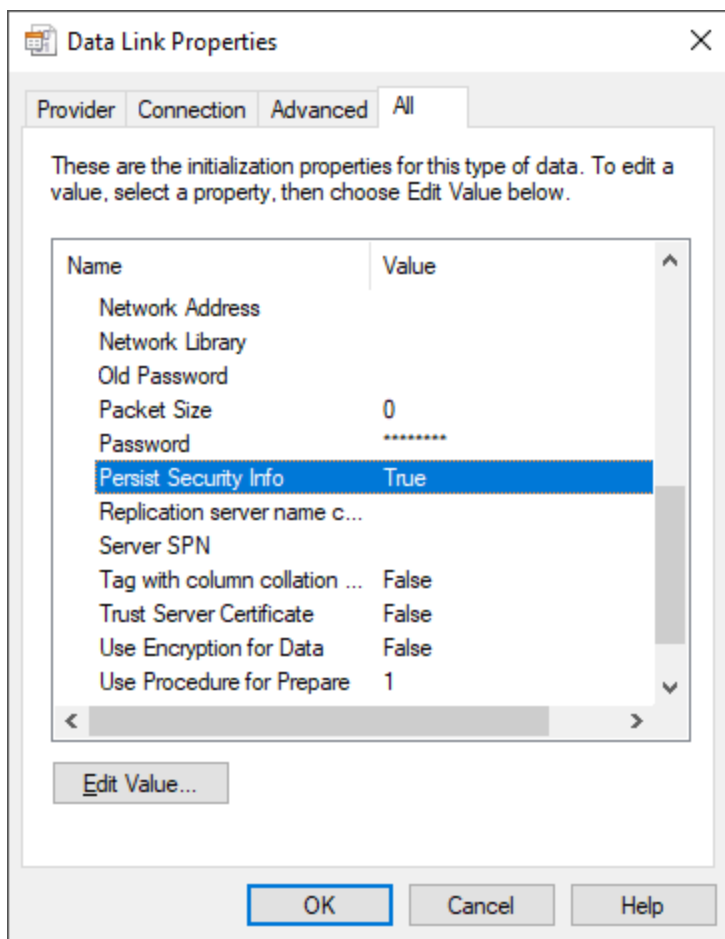
9. Click **OK**.
10. In the **Set A Data Source Name** dialog box, enter a connection name and click **OK**. In this example, the connection name is "ZooDBConnect".
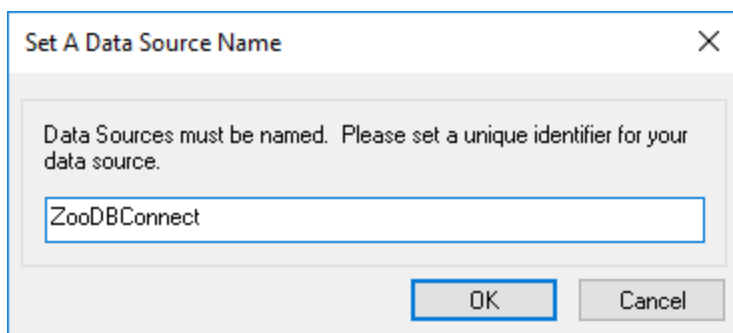
The Project window now contains a connection called "ZooDBConnect" in the Data Sources folder.
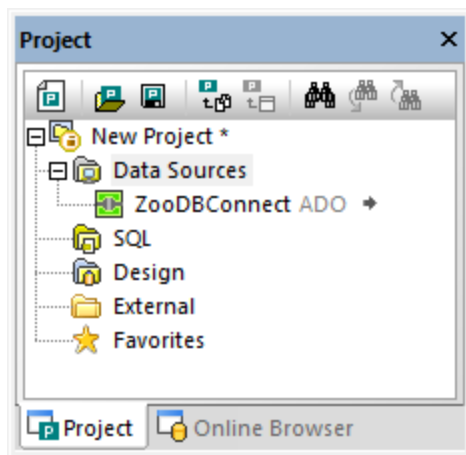
3.2.1.2 Adding SQL Files to the Project

A database project in DatabaseSpy allows you to store SQL files together with the database connection in one project file so that you have all the files and connections you need at hand when you open a project in DatabaseSpy. In this step, you add the SQL scripts to the project that you will later use to create the tables for the zoo database and to query the "Zoo" database.

To add the SQL files to the project file:

1. In the Project window, right-click the SQL folder and select **Add Files to Project...** from the context menu. The Open dialog appears.
2. Select the files to be added to the SQL folder from the tutorial directory:

   - **animalType_queries.sql**
   - **create_ZooDB_relationships.sql**
   - **create_ZooDB_tables.sql**
   - **tblAnimalTypes_data.sql**

3. Click **Open**. The SQL files appear in the SQL folder of the Project window.
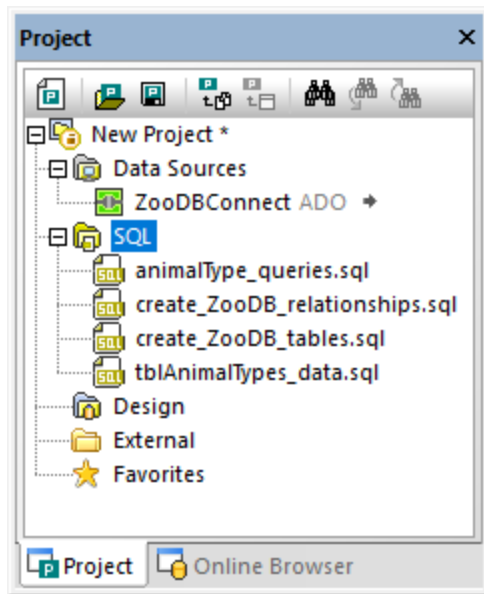
3.2.1.3 Saving the Project and the Data Source

The project created so far will be reused in the subsequent steps of this tutorial, so it must be saved to the disk.

To save the project:

1. Do one of the following:
    - Click **Save project** 🖫 in the Project window.
    - On the **File** menu, click **Save Project As**.
    - Press **Ctrl+S**.
2. In the Save As dialog box, choose a path for the project and enter "ZooDBProject" as a file name.
3. Click **Save**.

Optionally, you can also rename any of the project's data sources. To do this, the data source must first be disconnected:

1. Right-click the 🔁 **ZooDBConnect** data source in the Project window, and select **Disconnect** from the context menu. The icon now changes to "disconnected" 🔁.
2. Right-click the data source and select **Rename** from the context menu (or press **F2**), and enter the new data source name.

3.2.1.4 Defining the Project Start-up Options

You can optionally define that the last project be opened by default whenever DatabaseSpy starts, as follows:

1. On the **Tools** menu, click **Options**, and then click **General**.

2. Select the **Open last project on startup** check box.
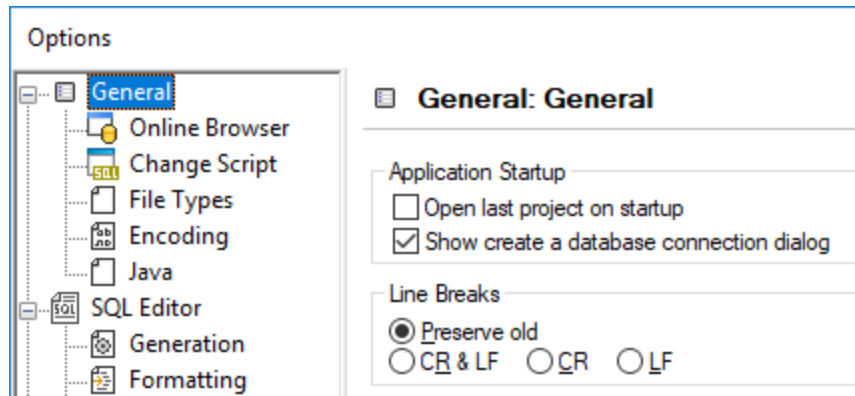
For reference to other settings, see DatabaseSpy Settings[417].

3.2.2 Adding Tables to the Database

In the previous section, you have created a database project and added a connection to a database. However, this database does not contain any tables yet. In this section of the tutorial, you will learn how to use SQL scripts and the Design Editor to create the tables that make up the "ZooDB" database. Specifically, you will learn how to do the following:

- Open and execute an SQL file
- Add a table using the Design Editor
- Generate and modify a CREATE statement
- Add a new column to a table

3.2.2.1 Opening and Executing an SQL File

In the previous section of the tutorial, you added the create_ZooDB_tables.sql script[78] to your database project. This script can now be run to add tables to the zoo database. Note that any SQL script will be executed against the currently assigned data source, as shown below.

To add tables to a database by executing an SQL script:

1. Open the "ZooDB" project[72] created previously and double-click the 🔷 **ZooDBConnect** data source to connect to it.
2. In the Project window, double-click the **create_ZooDB_tables.sql** entry in the SQL folder. The script file opens in an SQL Editor window.

Note: A new SQL Editor window is automatically associated with the data source that is active in the Online Browser[24] or Project Window[20]. The active data source is displayed at all times in the Execution Target Bar[49]. That is, all commands issued from SQL Editor will affect this particular data source only. If there are multiple data sources, the SQL Editor belongs to the one which was last clicked before you opened the SQL Editor. Any subsequent SQL Editor windows that you open will inherit the currently active data source.

3. If necessary, assign a data source to the current script. When the script has no data source assigned, the "Unknown - Offline" status appears at the top of the SQL Editor window.
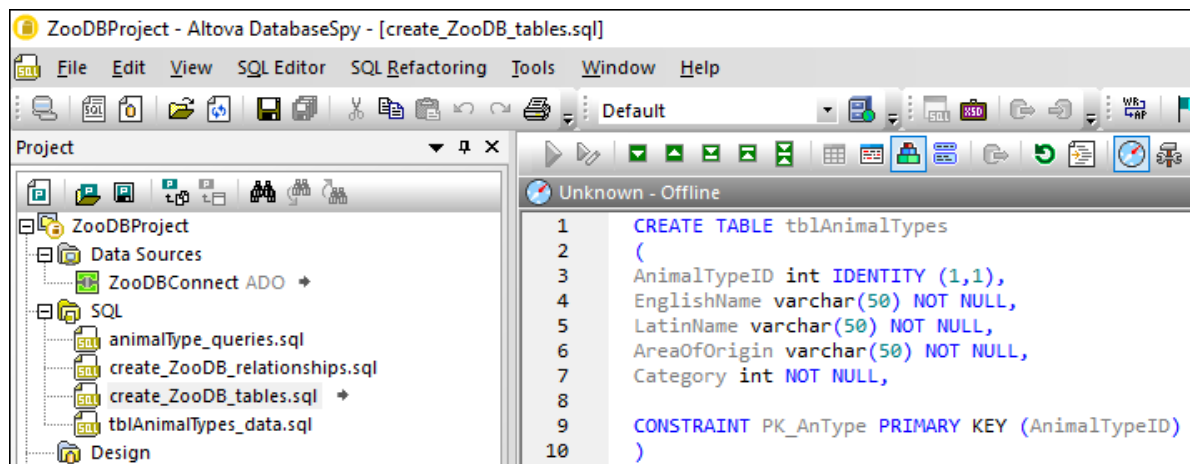
To assign a data source to the script, click the **Offline** hyperlink and, in the Properties window, select "ZooDBConnect" from the Data Source list.
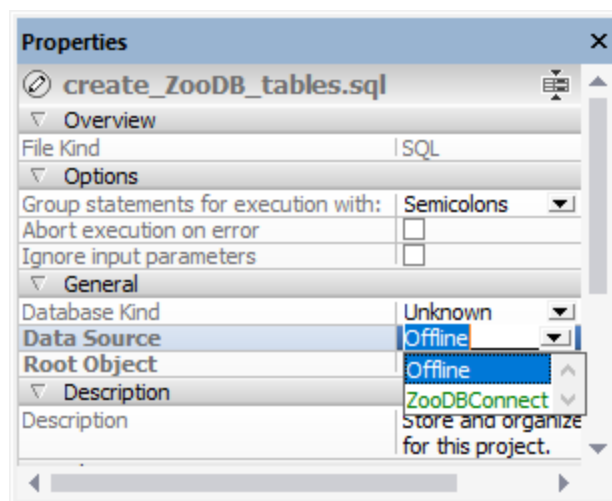
The SQL script is now connected to the **ZooDBConnect** data source, and the execution buttons in the

SQL Editor are active. Click the **Save Project** button in the Project window or press **Ctrl+S** to save this data source assignment in the project properties.

4. In the SQL Editor, click the **Execute** button or press **F5**.

If the script executed successfully, several new tables will have been added to the database:
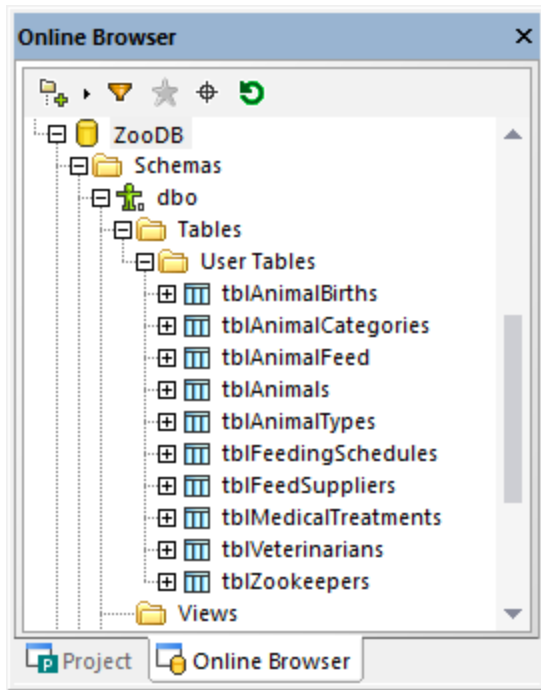
You can view the tables by clicking the **Online Browser** tab, and expanding the folder structure as shown above.

After performing changes to the database, make sure to refresh the data displayed in the Online Browser. To do this, right-click on the name of the database, "ZooDB" in this example, and select **Refresh** from the context menu or click the **Refresh** [icon] button.

3.2.2.2 Adding Tables Using Design Editor

The Design Editor provides the means to add tables to databases using a graphical user interface. Use of the Design Editor requires no knowledge of SQL syntax. To illustrate this, we will now add to the database a new table that stores zoo visitor statistics. We will call this table tblVisitorStatistics. The table will consist of the following columns:

- **StatisticID**: int; not null; primary key
- **StatisticDate**: datetime; not null;
- **VisitorCount**: int; not null

To add the table to the database using the Design Editor:

1. Open the "ZooDB" project [72] created previously and double-click the [icon] **ZooDBConnect** data source to connect to it.

2. Click the **Design Editor** [icon] toolbar button or press **Ctrl + D**. A new Design window, called Design1, is opened with the connection name and database name visible in the Status bar.
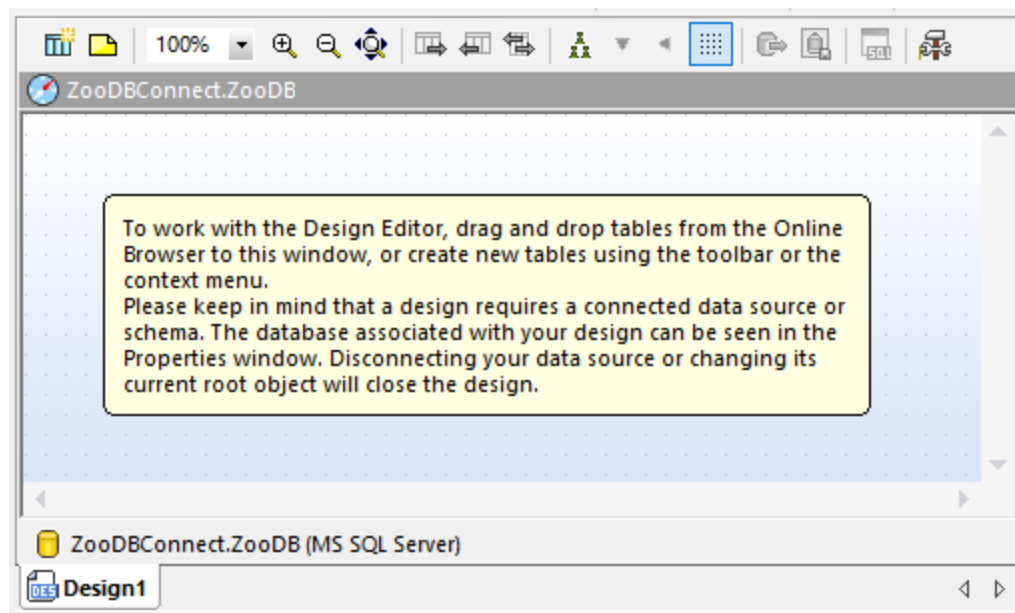
3. Click the **New Table** [icon] button. A new table appears in the Design Editor window. A dialog box appears at this point informing that a structure change script has been created. The generated SQL statements appear in the Database Structure Change Script window, as shown below.
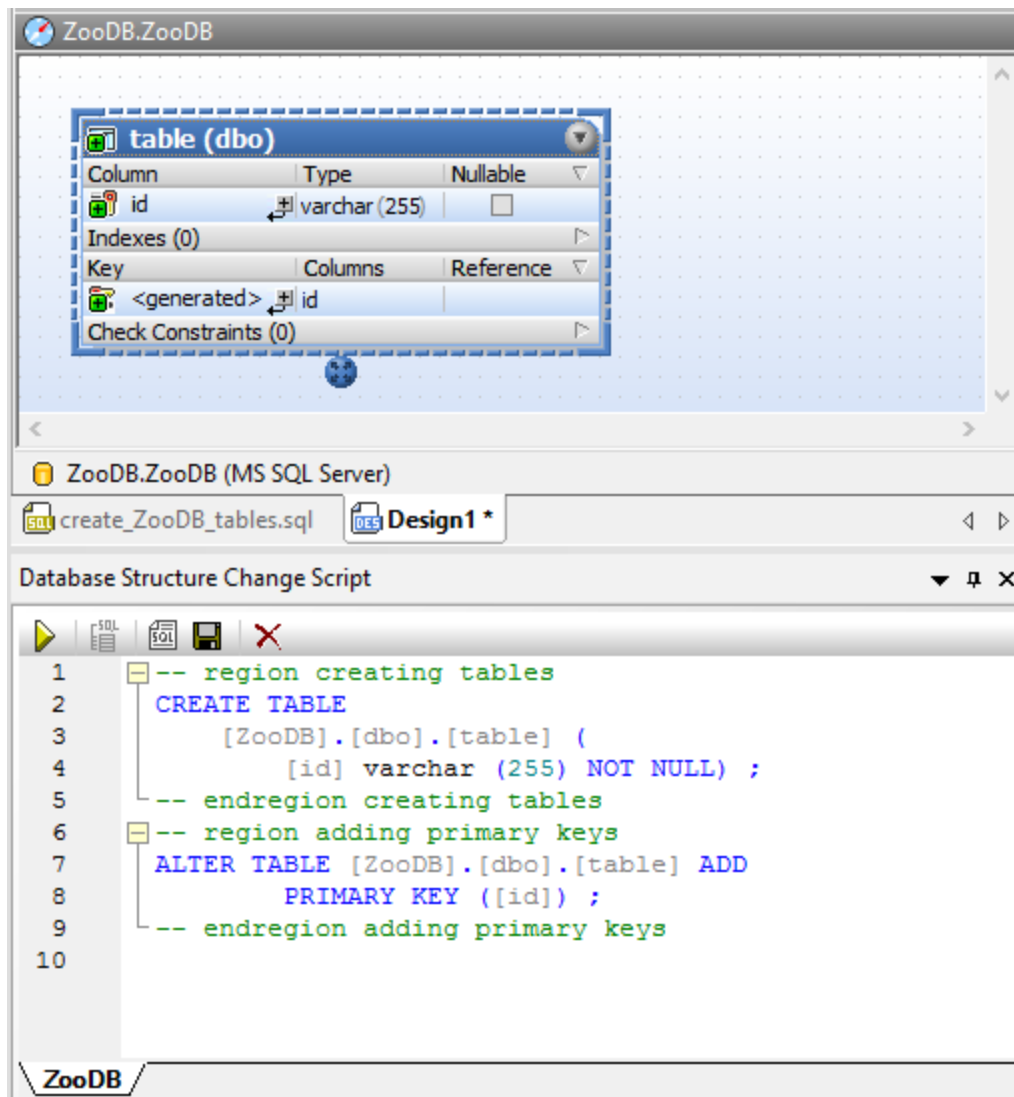
4. Double-click the table name, "table (dbo)", and enter "tblVisitorStatistics". Press **Enter**.
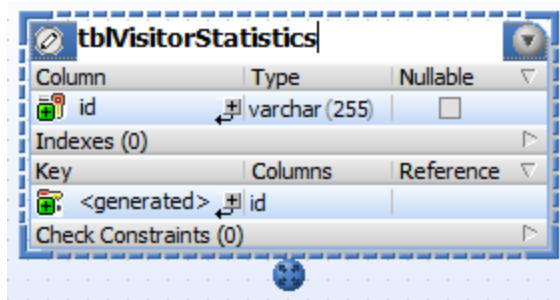
5. In the table design, double-click the "id" column. Change the column name to "StatisticID" and press **Enter**.
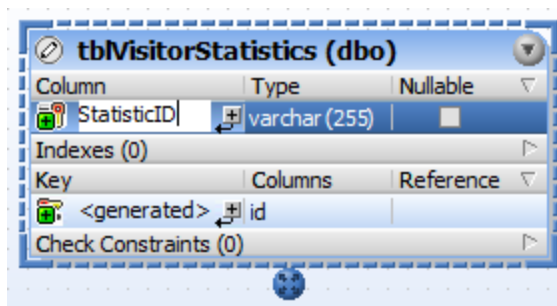
6. Likewise, double-click "varchar(255), replace it with "int", and press **Enter**. At this stage, the table has one column, and we want to add two more.
7. Click the plus icon ⊞ next to column "StatisticID". A column is added and selected for editing. Type "StatisticDate", press **Tab**, enter type as "date", and press **Enter**.
8. Right-click into the table design and choose **Insert new | Column** from the context menu. Change the column name to "VisitorCount" and press **Enter**. Using the same technique as for previous columns, change the data type of the "VisitorCount" column to "int".
9. Right-click the title bar of the Columns section and select the **Identity** option in the context menu. Then select the **Identity** check box for the "StatisticID" column.

At this stage, the table design should look as follows:
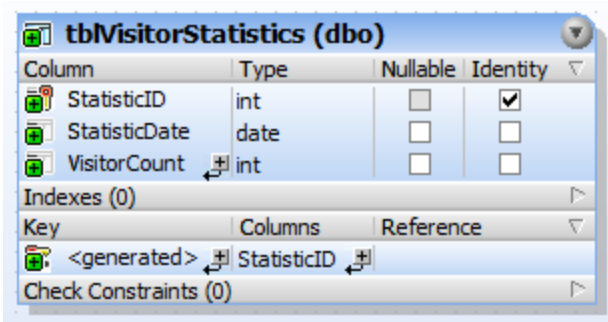
Notice that a primary key having the name <generated> has been generated automatically for the first column in the table. This indicates that the new column will have a primary key with a name automatically assigned by the database, when you run the database update script. If you prefer primary keys not to be generated automatically, you can change this behavior from **Tools | Options | Design Editor | Automatically create new primary key for new tables**.

Optionally, if you want to change the name of the primary key, replace the text <generated> with a descriptive primary key name. In this case, however, you will need to ensure that the name is not already used in the database.

You are now ready to execute the database script that will actually add the table to the database. In the

Database Structure Change Script window, click the **Execute Change Script** ▷ button to commit the changes to the database.

```
Database Structure Change Script                              ▼ ⁋ ✕

 ▷ | ▤ | ▥ ▉ | ✕
   1    ⊟-- region creating tables
   2      CREATE TABLE
   3          [ZooDB].[dbo].[tblVisitorStatistics] (
   4              [StatisticID] int IDENTITY (1, 1) NOT NULL,
   5              [StatisticDate] date NOT NULL,
   6              [VisitorCount] int NOT NULL) ;
   7      └-- endregion creating tables
   8    ⊟-- region adding primary keys
   9      ALTER TABLE [ZooDB].[dbo].[tblVisitorStatistics] ADD
  10              PRIMARY KEY ([StatisticID]) ;
  11      └-- endregion adding primary keys
  12

 ╲ ZooDB ╱
```

3.2.2.3 Generating a CREATE Statement

You can also generate SQL statements based on an existing table or column. In this step, we will generate a CREATE statement from the table tblZookeepers, edit the statement, and use it to create a new table tblEmployees.

To add a table by editing a CREATE statement generated from an existing table:

1. Open the "ZooDB" project [72] created previously and double-click the 🔼 **ZooDBConnect** data source to connect to it.
2. In the Online Browser, right-click the tblZookeepers table.
3. Select the menu option **Show in new SQL Editor | Create** from the context menu. An SQL Editor window appears, displaying the following SQL statement:

```
 🌐 ZooDB.ZooDB
   1    CREATE TABLE
   2        [ZooDB].[dbo].[tblZookeepers] (
   3            [ZookID] int IDENTITY (1, 1) NOT NULL,
   4            [FirstName] varchar (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
   5            [LastName] varchar (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
   6            [Address] varchar (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
   7            [City] varchar (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
   8            [State] varchar (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
   9            [Telephone] varchar (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
  10            [DOB] datetime NOT NULL,
  11            CONSTRAINT [PK_Zook] PRIMARY KEY ([ZookID]) ) ;
```

4. Edit the statement as shown below to create tblEmployees. The parts of the statement that need to be changed are marked using *underlined, italic text*:

```
CREATE TABLE
    [ZooDB].[dbo].[tblEmployees] (
        [EmployeeID] int IDENTITY (1, 1) NOT NULL,
        [FirstName] varchar (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
        [LastName] varchar (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
        [Address] varchar (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
        [City] varchar (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
        [State] varchar (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
        [Telephone] varchar (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
        [DOB] datetime NOT NULL,
        CONSTRAINT [PK_Employee] PRIMARY KEY ([EmployeeID]) ) ;
```

5. Click the **Execute** ▶ button or press **F5**.In the Online Browser, right-click the ZooDB database and
    select **Refresh** from the context menu, or click the **Refresh** 🔄 button.
    The newly inserted table, tblEmployees, is displayed along with the tables that you created earlier.

3.2.2.4 Adding a New Column to a Table

The instructions below show you how to add a new column to the table tblVeterinarians, using a SQL script
typed manually. The new column should be called "Email" and its data type should be varchar(50).

Note: Entering a SQL statement is just one of the ways to add columns to an existing table. For other ways
to add a column, see Creating Columns [240].

To create the new column:

1. Open the "ZooDB" project [72] created previously and double-click the 🔼 **ZooDBConnect** data source
    to connect to it.
2.   Click the **SQL Editor** 🗔 toolbar button to open a new SQL Editor window.
3. Start typing "alt". The auto-completion helper pops up.
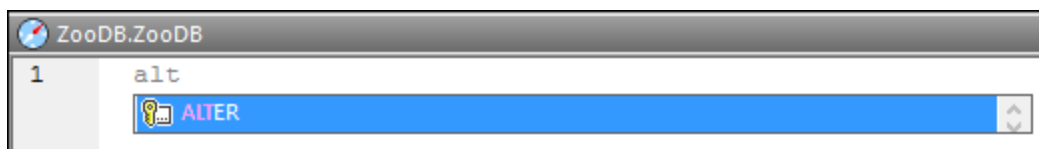
4. Press **Enter**. The word "ALTER" appears in the SQL Editor.
5. Start typing "tab" and add the word "TABLE" after "ALTER". Your SQL statement should now be
    "ALTER TABLE".
6. Press **Space**, and then press **Ctrl+Space** to manually trigger auto-completion.
7. Select "tblVeterinarians" from the list and press **Space**.
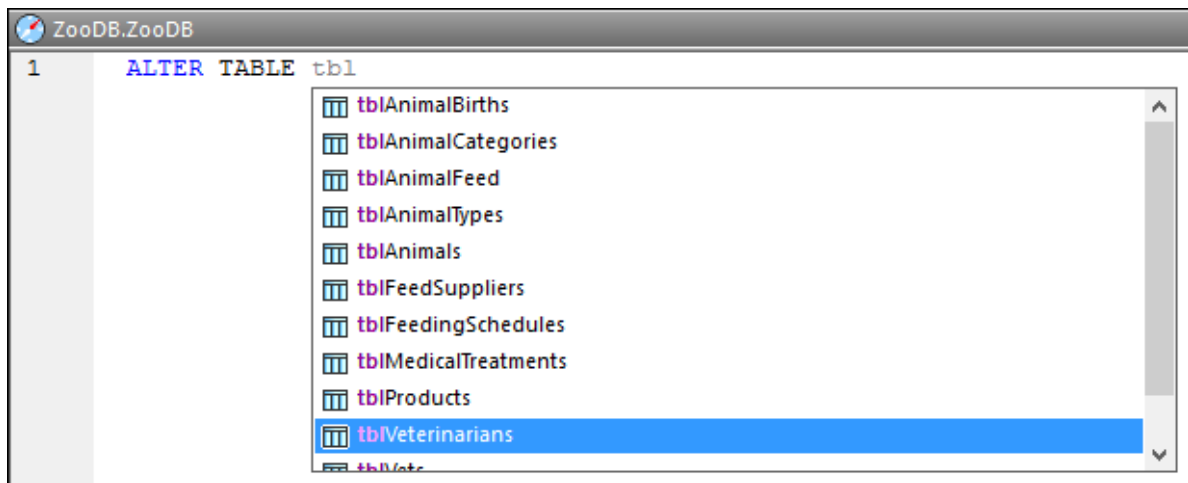
8. Continue creating the SQL statement until you have typed the following text:

```
ALTER TABLE [tblVeterinarians] ADD Email VARCHAR (50)
```

9. Execute the SQL statement by clicking the **Execute** button or pressing **F5**.

You have now created the new `Email` column. To view it in the Online Browser, click the **Refresh database** button.

3.2.3   Defining Constraints

In the previous section, you have added tables using both SQL statements and the Design Editor. In this section of the tutorial, you will learn how to create database constraints, using the Design Editor. Specifically, you will learn how to do the following:

*   Define a unique constraint for a column
*   Add check constraints on column and table level
*   Provide columns with default values
*   Manually create relationships between tables
*   Use SQL scripts to create foreign key relationships

3.2.3.1 Defining a Unique Constraint

By defining a unique key, you can prevent duplicate data from being entered into a column. In our zoo database, you will define a unique key on the `LatinName` column of table `tblAnimalTypes`. This way, each name can be added to the table only once.

To define a unique key on a column:

1. Open the "ZooDB" project [72] created previously and double-click the ZooDBConnect data source to connect to it.

2. In the Online Browser, right-click the table `tblAnimalTypes` and select **Design Editor | Show in new Design Editor** from the context menu.
3. In the table design, right-click the column `LatinName` and select **Make Unique key** from the context menu.
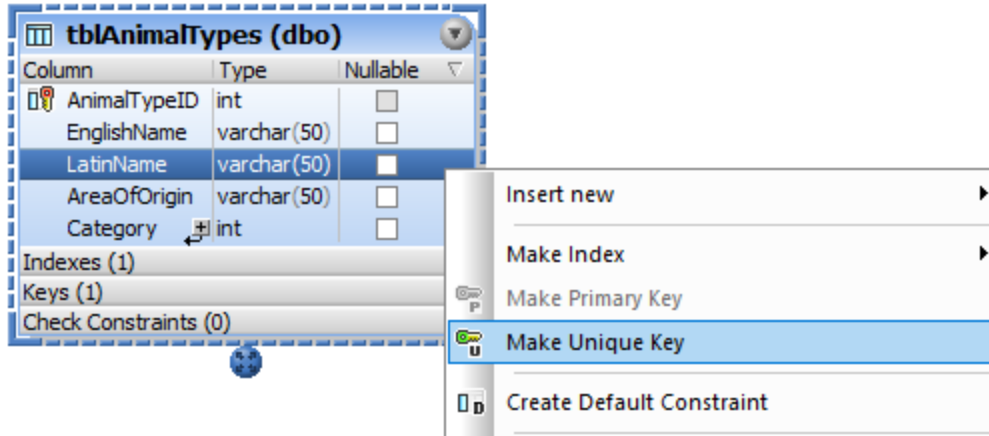
A unique key is created in the design, having the name `<generated>`, and the entry is selected for editing.
4. Optionally, change the key name from `<generated>` to "UK_LatinName" and press **Enter**. If you leave the name as `<generated>`, the key name will be automatically generated by the database.
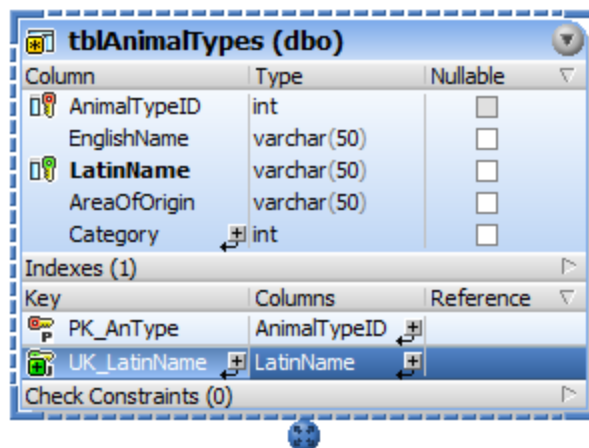
5. In the Database Structure Change Script window, click the **Execute Change Script** button to execute the change script and to commit the new unique key to the database.

3.2.3.2 Defining a Check Constraint

When a check constraint is defined, data will be validated according to the expression specified in the check constraint, before it is added to a table. Data that fails the validation will not be added.

In this section of the tutorial, you will create a check constraint that makes sure that the Mother and Father columns of `tblAnimalBirths` are not the same.

To define the check constraint:

1. Open the "ZooDB" project [72] created previously and double-click the ⬆ **ZooDBConnect** data source to connect to it.

2. In the Online Browser, right-click table `tblAnimalBirths` and select **Design Editor | Show in new Design Editor** from the context menu.

3. Right-click the table design and select **Insert new | Check Constraint** from the context menu. A check constraint is created in the design, having the name `<generated>`, and the entry is selected for editing.
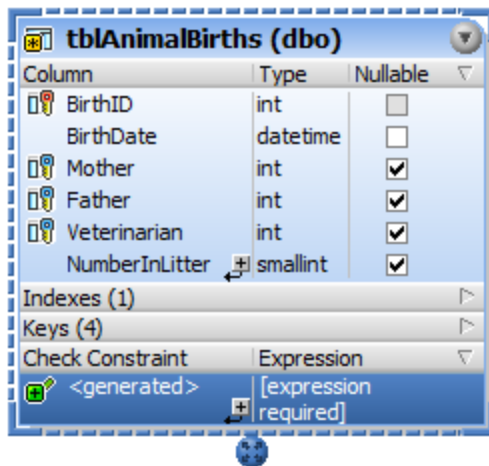
4. Optionally, change the constraint name from `<generated>` to "CK_CheckParents" and press **Enter**. If you leave the name as `<generated>`, the key name will be automatically generated by the database.

5. In the Expression column, double-click the "[expression required]" entry, enter "Mother <> Father", and press **Enter**.
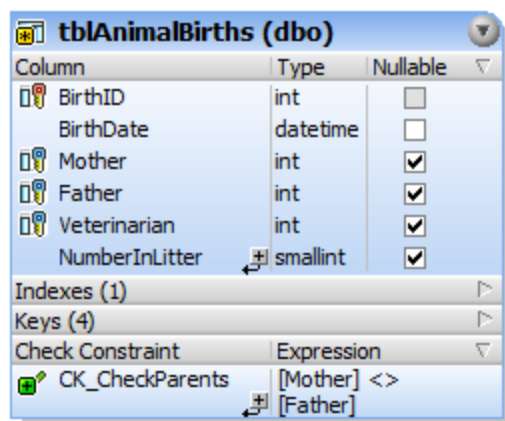
6. In the Database Structure Change Script window, click the **Execute Change Script** ▶ button to execute the change script and to commit the new check constraint to the database.

Testing the check constraint

After the check constraint has been committed to the database, you cannot enter the same animal ID into the "Mother" and "Father" columns anymore. For example, a SQL statement like the one below will result in an error because it attempts to add a record where the "Mother" and "Father" values are the same:

```
INSERT INTO [ZooDB].[dbo].[tblAnimalBirths] ([BirthDate],
                                             [Mother],
                                             [Father],
                                             [Veterinarian],
                                             [NumberInLitter])
VALUES ('2019-04-24 00:00:00', 5055, 5055, 4, 3) ;
```

3.2.3.3 Defining a Default Constraint

For some columns it may be useful to define a standard default value. When no other value is supplied explicitly, the default is inserted automatically in the respective column when you add a new record to the table.

In this section of the tutorial, you will define a default constraint on the NumberInLitter column of the tblAnimalBirths table. In other words, the NumberInLitter value must be set automatically to "1" for new rows.

To add a default constraint:

1. Open the "ZooDB" project [72] created previously and double-click the 🔁 **ZooDBConnect** data source to connect to it.
2. In the Online Browser, right-click table tblAnimalBirths and select **Design Editor | Show in new Design Editor** from the context menu.
3. In the table design, right-click column "NumberInLitter" and select **Create Default Constraint** from the context menu. The **Default Value** column appears in the table design. (If the **Default Value** column is not visible, click into the table design and it will resize properly.)
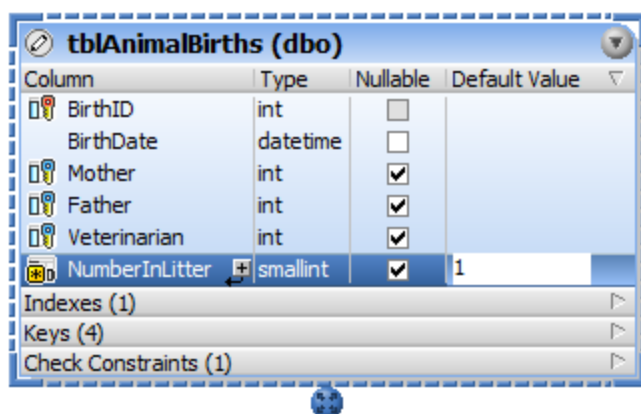
4. Double-click the field in the NumberInLitter row, type "1", and press **Enter**.

5.   In the Database Structure Change Script window, click the **Execute Change Script** ▶ button to execute the change script and to commit the new default constraint to the database.

Testing the default constraint

Before testing the default constraint you have just defined, make sure that the **Initialize cells with default values** option is enabled in the Result View [429] options.

To create a new row containing a default value:

1.   Right-click the table heading in the Design Editor and select **SQL and Data | Edit Data** from the context menu. A message box appears, click **OK**.

2.   In the Result view, click the **Append a new row** ⊞ button. The "NumberInLitter" column shows "1" in the new row.

	BirthID •	BirthDate •	Mother •	Father •	Veterinarian •	NumberInLitter •
1	0	2000-01-0...	0	0	0	1

3.2.3.4 Defining a Foreign Key Constraint

A foreign key is a relation between two tables that helps preserve the integrity of the database data. For example, when you add a record, a foreign key ensures that a field (column value) can be inserted in one table only if it exists in the second table.

In this example, let us create a foreign key constraint that achieves the following: it should be possible to add a new record in table `tblAnimalTypes` only if the record's `Category` exists as `CategoryID` in table `tblAnimalCategories`.

To define the foreign key relationship:

1.   Open the "ZooDB" project [72] created previously and double-click the 🔼 **ZooDBConnect** data source to connect to it.
2.   Click the **Design Editor** 🔲 toolbar button or press **Ctrl+D**.
3.   In the Online Browser, click the table `tblAnimalTypes` and drag it into the Design Editor window.
4.   Right-click the table design and select **Insert new | Key | Foreign Key** from the context menu. A foreign key is created in the design, having the name `<generated>`, and the entry is selected for editing.
5.   Optionally, change the key name from `<generated>` to "FK_CategoryID" and press **Enter**. If you leave the name as `<generated>`, the key name will be automatically generated by the database.
6.   In the Reference column, double-click the "[select the referenced table]" entry, and select "tblAnimalCategories" from the drop-down list that appears. The first column "CategoryID" of the referenced table appears below the table name and there is no need to change this entry.
7.   In the Columns column, double-click the "AnimalTypeID" entry that has been inserted by default, and select "Category" from the drop-down list that appears.
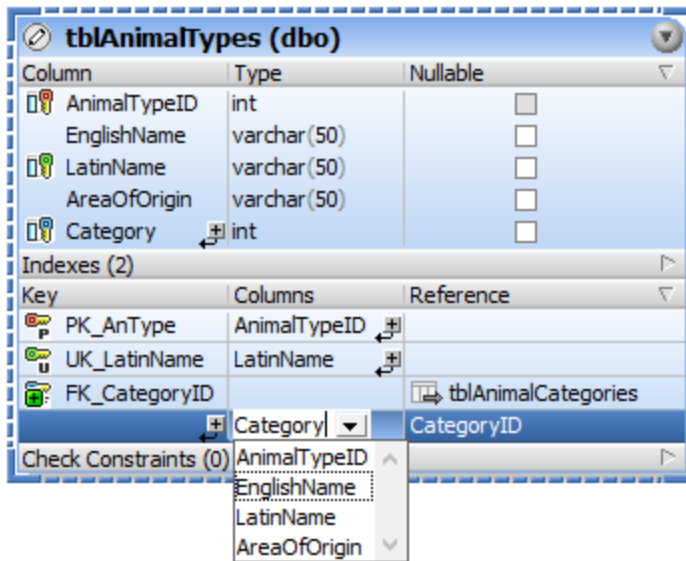
8. In the Database Structure Change Script window, click the **Execute Change Script** ▷ button to execute the change script and to commit the new foreign key to the database.

To view the new relationship in the Design Editor, right-click the "FK_CategoryID" entry in the table design and select **Add related tables | Referenced Tables** from the context menu.

3.2.3.5 Adding Foreign Keys via SQL Scripts

To speed up the creation of the remaining foreign key relationships for the zoo database, an SQL script is provided. This script has already been added to the project [78] earlier in this tutorial.

1. Open the "ZooDB" project [72] created previously and double-click the ⬆ **ZooDBConnect** data source to connect to it.
2. In the Project window, double-click the file **create_ZooDB_relationships.sql**. Notice that the top of the SQL Editor displays the data source and the database against which the script is to be executed (**ZooDBConnect.ZooDB**). For more information, see Execution Target Bar [49].
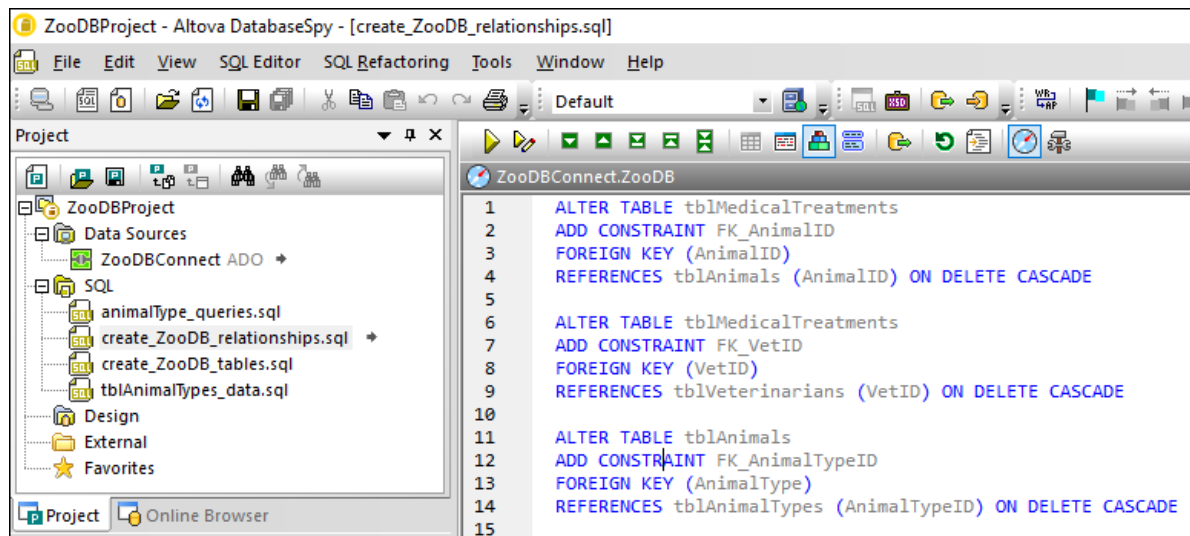
3. Click the **Execute** ▶ button or press **F5**. Foreign key relationships are created between the tables in the zoo database.

To get a graphical view of the relationships that were just created, right click the "User Tables" folder in the Online Browser, and select **Show in new Design Editor**.

3.2.4 Adding Data to the Database

The "ZooDB" database created so far has several tables, but it does not contain any data. This tutorial section shows you how to insert data in the database. Specifically, you will learn how to do the following:

- Execute an existing SQL script that populates the database with data
- Import data from a CSV file
- Import data from an XML file
- Create a new SQL INSERT script
- Add new records manually

3.2.4.1 Using an INSERT Script

Earlier in this tutorial, you added several SQL scripts to the ZooDBProject. The scripts contain SQL statements that either create the database structure or populate it with data. For example, the file **tblAnimalTypes_data.sql** contains all the INSERT statements that are necessary to add data to the `tblAnimalTypes` and `tblAnimalCategories` tables.
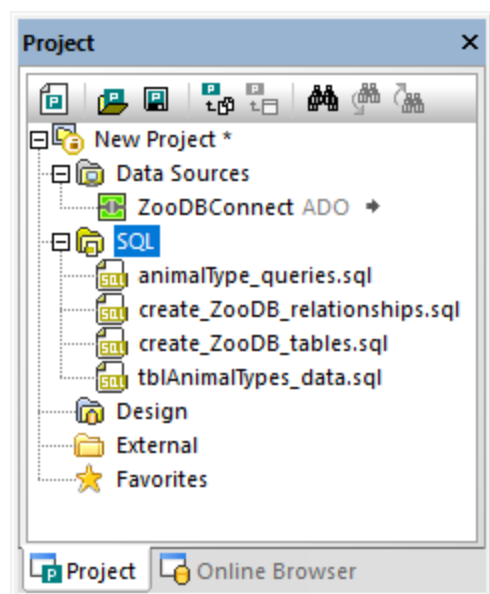
Project window

As previously mentioned in <u>Opening and Executing an SQL File</u> [80], you can change the target data source for each script in the Properties window. This is typically necessary when your project contains multiple data sources and you must ensure that you are running the script against the correct database.

It is also possible to assign the target data source of multiple scripts simultaneously. For example, in the Project window illustrated above, notice that all the script files are in the **SQL** folder. By changing the folder's properties, you can assign the same target data source to all the scripts, as follows:

1. Open the <u>"ZooDB" project</u> [72] created previously and double-click the **ZooDBConnect** data source to connect to it.
2. Click the **SQL** folder in the Project window.
3. In the Properties window, select "ZooDBConnect" from the **Data Source** list.
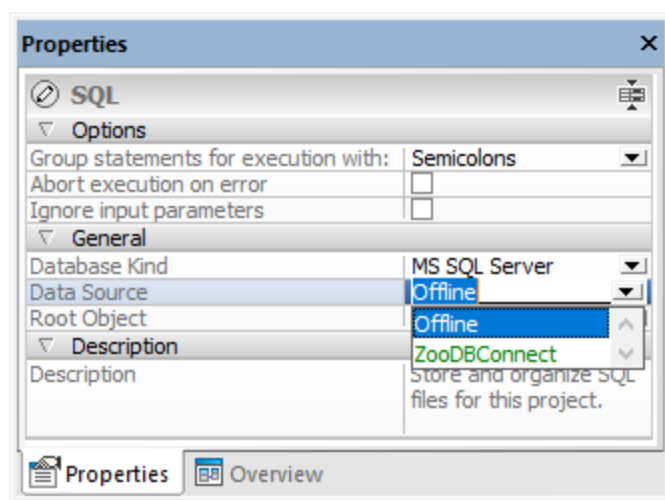
All the SQL files from the **SQL** folder should now have the same data source. Click any of the script files and observe the assigned data source in the Properties window.

Next, let's populate two of the database tables with data, by running a script:

- In the Project window, right-click the file **tblAnimalTypes_data.sql** and select **Execute SQL** [▷] from the context menu. The file opens in an SQL Editor window and executes automatically.

The tables `tblAnimalTypes` and `tblAnimalCategories` should now be populated with data. To do a quick check as to whether the tables have been successfully populated with data, right-click the "User Tables" folder and select **Row Count | Show/Update** from the context menu.
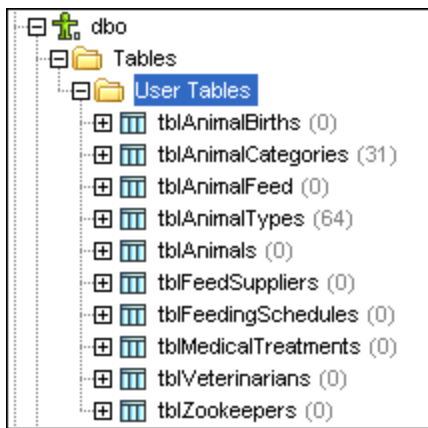
3.2.4.2 Importing Data from CSV Files

DatabaseSpy provides an import function which allows you to insert data from comma-separated value (CSV) files into a database. In this tutorial step, you will import data from the **tblZookeepers.txt** file into the database table `tblZookepers`.

To import data:

1. Open the "ZooDB" project [72] created previously and double-click the 🔼 **ZooDBConnect** data source to connect to it.
2. Do one of the following:

   - Click the **Import** [⤵] toolbar button.
   - On the **Tools** menu, click **Import data to the database**.
   - Press **Ctrl+I**.

3. A dialog box appears. Click the **CSV** button on the left side of the dialog box.

4. Click [⋯] next to the **Path**, and browse for the file **C: \Users\<username>\Documents\Altova\DatabaseSpy2023\DatabaseSpyExamples\Tutorial\tblZo okeepers.txt**.
5. Under **Database Generation Options**, select **Update existing tables**.

6. Under **Import Mode**, select **Generate SQL in SQL Editor**.

7. Click the `ZookID` column 🔲 until it changes to the symbol for a column not to be imported ❎. The column header allows you to define the field's key type, and also if it should be included during the import process. We will not import the `ZookID` column since this is an identity column and the primary key will be generated automatically during import.
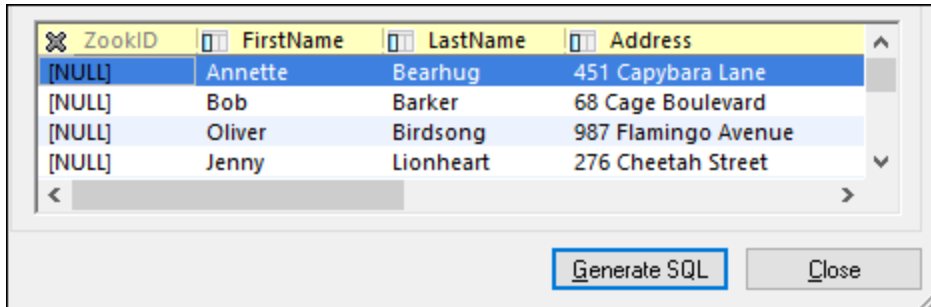
8. Click the **Generate SQL** button. This creates an SQL script which can be executed to complete the import process. Notice that the `ZookID` column is not included in the script, because it was specifically excluded in a previous step.


```
 1     INSERT INTO [ZooDB].[dbo].[tblZookeepers] ([FirstName],
 2                                                [LastName],
 3                                                [Address],
 4                                                [City],
 5                                                [State],
 6                                                [Telephone],
 7                                                [DOB])
 8     VALUES ('Annette',
 9             'Bearhug',
10             '451 Capybara Lane',
11             'Newark',
12             'NJ',
13             '(515)555-7836',
14             '1979-11-08 00:00:00.000') ;
```
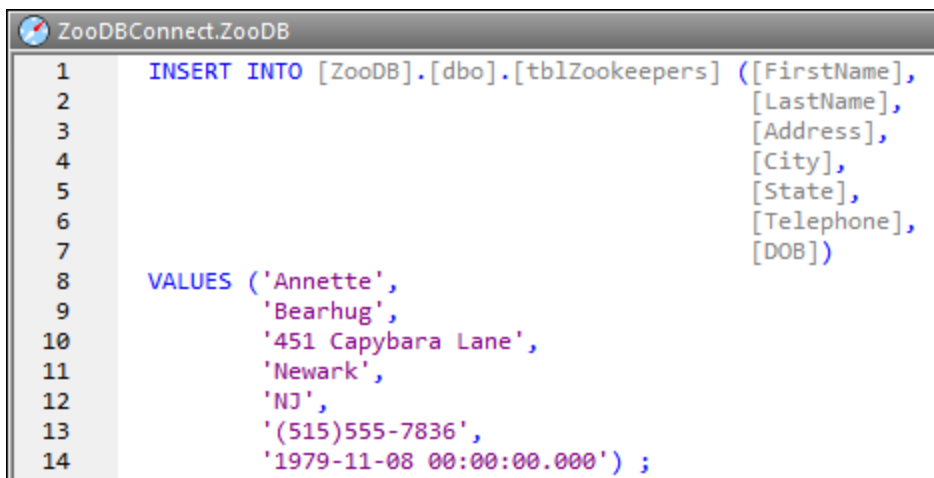
9. Click the **Execute** ▶ button or press **F5**.

To view the imported data, right-click the `tblZookeepers` table in the Online Browser and select **Retrieve data | All rows** from the context menu (alternatively, press **Ctrl+Alt+R**).

	ZookID •	FirstName •	LastName •	Address •	City •	State •	Telephone •	DOB •
1	1	Annette	Bearhug	451 Capybara Lane	Newark	NJ	(515)555-7836	1979-11-08 00:00:00.000
2	2	Bob	Barker	68 Cage Boulevard	Philadelphia	PA	(515)555-1357	1972-04-24 00:00:00.000
3	3	Oliver	Birdsong	987 Flamingo Avenue	Harrisburg	PA	(515)555-8003	1958-01-03 00:00:00.000
4	4	Jenny	Lionheart	276 Cheetah Street	Sayre	PA	(515)555-2073	1980-02-04 00:00:00.000
5	5	Piers	Swanson	56 Penguin Court	Newark	NJ	(515)555-5278	1966-11-06 00:00:00.000
6	6	Audrey	Wolf	22 Zebra Street	Philadelphia	PA	(515)555-6299	1969-07-02 00:00:00.000
7	7	Damien	Frogier	67 Sturgeon Place	Harrisburg	PA	(515)555-1198	1976-01-26 00:00:00.000

3.2.4.3 Importing Data from XML Files

In addition to importing data from CSV files, you can also import data from XML files. In this tutorial step, you will import data from an XML file into the `tblFeedSuppliers` database table.

To import data:

1. Open the "ZooDB" project [72] created previously and double-click the 🔼 **ZooDBConnect** data source to connect to it.
2. Do one of the following:

    *   Click the **Import** 🔄 toolbar button.
    *   On the **Tools** menu, click **Import data to the database**.
    *   Press **Ctrl+I**.

3. A dialog box appears. Click the **XML** button on the left side of the dialog box.
4. Click ⋯ next to the **Path**, and browse for the file **C:\Users\<username>\Documents\Altova\DatabaseSpy2023\DatabaseSpyExamples\Tutorial\dbo.tblFeedSuppliers.xml**.
5. Under **Database Generation Options**, select **Update existing tables**.
6. Under **Import Mode**, select **Generate SQL in SQL Editor**.
7. Click the **Options** tab.
8. Under **Start point of import**, in the "Starting from" list, select "Row". (This specifies the name of the XML element from which data will be imported).
9. Under **Automatic fields**, click to clear the **Create primary/foreign keys** check box.
10. Under **Other options**, select **Exclude namespace name**.
11. Under **Preview**, click the **Reload** button. This displays table data in the Preview window.
12. Click inside the "Target Table Name" column and choose **dbo.tblFeedSuppliers** from the drop-down list.
13. Click the **SupplierID** column 🔲 until it changes to the symbol for a column not to be imported ❎. The column header allows you to define whether it should be included during the import process. We will not import the **SupplierID** column since this is an identity column and the primary key will be generated automatically during import.
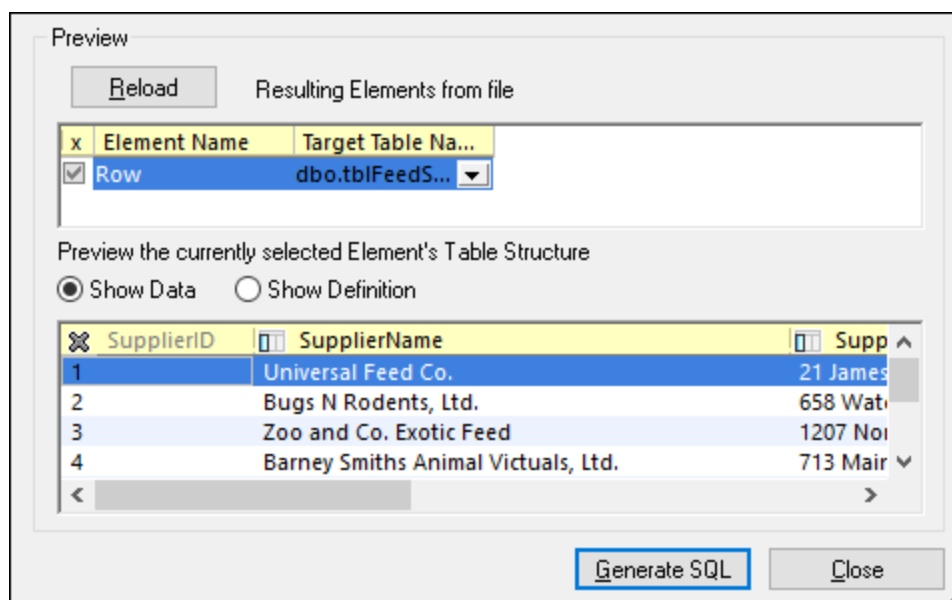
14. Click the **Generate SQL** button. This creates an SQL script which can be executed to complete the import process.
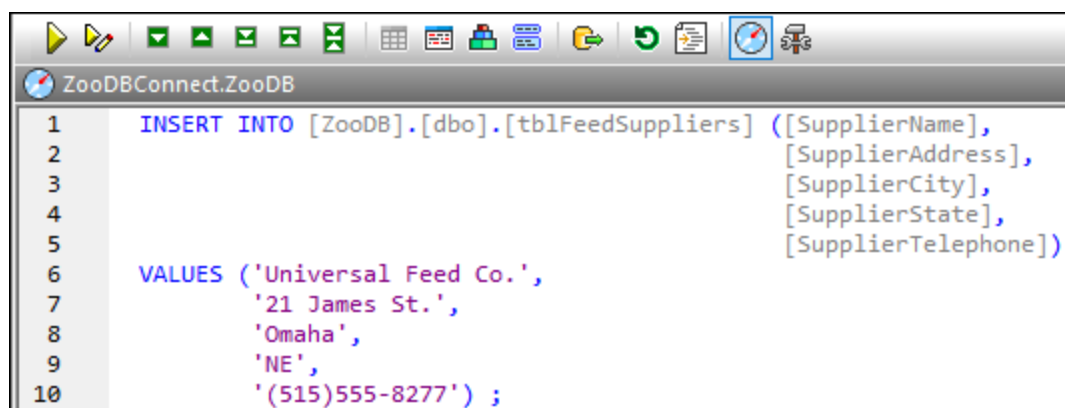
15. Click the **Execute** [icon] button or press **F5**.

To view the imported data, right-click the `tblFeedSuppliers` table in the Online Browser and select **Retrieve data | All rows** from the context menu (alternatively, press **Ctrl+Alt+R**).

3.2.4.4 Populating Other Tables (Optional)

In the previous steps of this tutorial, you have imported data into the database from a text file as well as from an XML file. This has populated the tblZookeepers and tblFeedSuppliers tables with data.

Optionally, to populate the remaining database tables with data, run an import from the following text or XML files available in the **C: \Users\<username>\Documents\Altova\DatabaseSpy2023\DatabaseSpyExamples\Tutorial** folder. For each table, you should import data from either the CSV or the XML file (not both).

CSV files	XML files
1. tblAnimalFeed.txt 2. tblFeedingSchedules.txt 3. tblAnimals.txt 4. tblVeterinarians.txt 5. tblAnimalBirths.txt 6. tblMedicalTreatments.txt	1. dbo.tblAnimalFeed.xml 2. dbo.tblFeedingSchedules.xml 3. dbo.tblAnimals.xml 4. dbo.tblVeterinarians.xml 5. dbo.tblAnimalBirths.xml 6. dbo.tblMedicalTreatments.xml

Note that the source files contain foreign key values that are supposed to exist in the target database tables at the time of import. For example, table tblAnimals must be populated before tblMedicalTreatments, because one cannot add a medical treatment record for an animal ID that does not exist yet in the database. Therefore, for the import to be successful, make sure that:

1. The files are imported in the order shown above.
2. No data has been previously added or deleted from the destination table (in other words, the table must have an identity value of **0**).

If you discover that the import fails due to foreign key constraint violations, it is likely that data has already been added or deleted in the table. To remedy this, delete all data from the table, reset the identity value to **0**, and then perform the import again. The following code listing shows how to reset the identity value of a SQL Server table to **0**:

```
USE ZooDB;
GO
DBCC CHECKIDENT ('tblVeterinarians', RESEED, 0);
GO
```

3.2.4.5 Generating an INSERT Statement

One of the ways to insert data into a table is creating an INSERT statement and running it against the database. This tutorial step shows you how to generate the INSERT statement automatically.

Assuming that the zoo in our tutorial has just hired two new zookeepers, you will create a script that adds these two new employees to the table tblZookeepers.

To generate the INSERT script:

1. Open the "ZooDB" project [72] created previously and double-click the ⬆ **ZooDBConnect** data source to connect to it.

2. Click the **SQL Editor** 🔲 toolbar button or select the menu option **File | New | SQL Editor** (**Ctrl+N**). A new SQL Editor window opens.

3. In the Online Browser, find the tblZookepers table and, holding the **Ctrl** key pressed, select all its columns except the ZookID column.
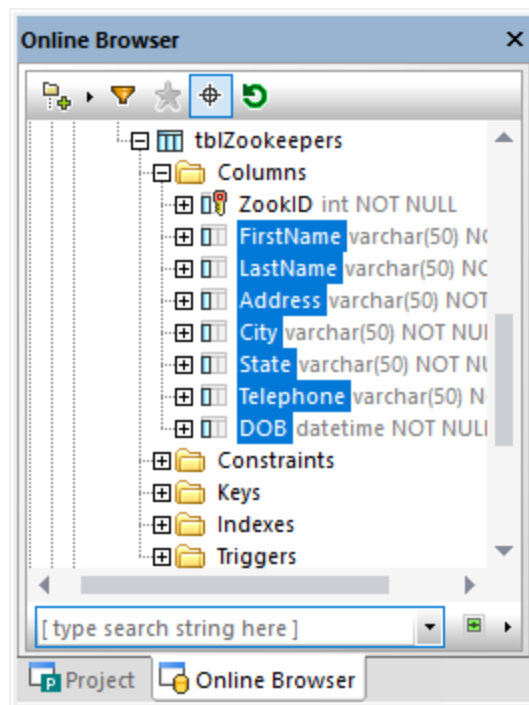
4. Using the *right* mouse button, drag and drop the selected columns from the Online Browser into the SQL Editor window. A context-sensitive menu appears.
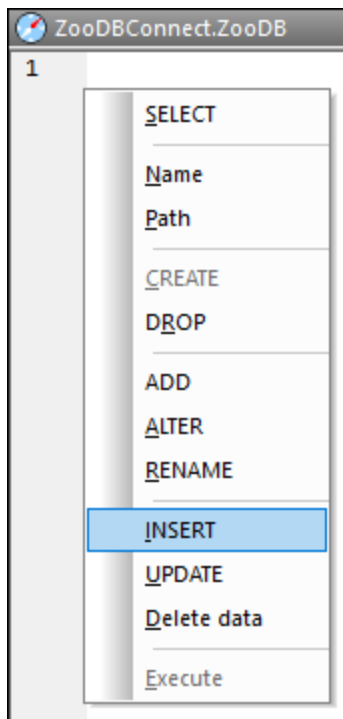
5. Select **Insert** from the context menu. The following statement appears in the SQL window:

```
INSERT INTO [ZooDB].[dbo].[tblZookeepers] ([FirstName],  [LastName], [Address],
[City], [State], [Telephone], [DOB])
VALUES (NULL, NULL, NULL, NULL, NULL, NULL, NULL) ;
```

6. Edit the statement as follows:

```
INSERT INTO [ZooDB].[dbo].[tblZookeepers] ([FirstName], [LastName], [Address],
[City], [State], [Telephone], [DOB])
VALUES ('June', 'Bug', '1345 Gorilla Highway', 'Newark', 'NJ', '(515)555-7217',
'1963-11-26');
```

7. Create a second INSERT statement and edit it as follows:

```
INSERT INTO [ZooDB].[dbo].[tblZookeepers] ([FirstName], [LastName], [Address],
[City], [State], [Telephone], [DOB])
VALUES ('Bruno', 'Katz', '87 Panther Place', 'Trenton', 'NJ', '(515)555-7667',
'1969-09-13');
```

8. Click the **Execute** ⏵ button or press **F5**.

To check the results, open a new SQL Editor window and enter:

```
SELECT * FROM tblZookeepers
```

Click **Execute** to run the statement and notice that two new rows have been inserted into the table.

	ZookID	FirstName	LastName	Address	City	State	Telephone	DOB
13	13	Sally	Crowe	42 Platypus Drive	Camden	NJ	(515)555-5587	1972-04-24 00:00:00.000
14	14	June	Bug	1345 Gorilla Highway	Newark	NJ	(515)555-7217	1963-11-26 00:00:00.000
15	15	Bruno	Katz	87 Panther Place	Trenton	NJ	(515)555-7667	1969-09-13 00:00:00.000

3.2.4.6 Inserting Data Manually

Entering data manually in a database table is another way to insert data in the database. This approach does not require you to write any SQL queries because all of them are generated by DatabaseSpy as you work.

1. Open the "ZooDB" project [72] created previously and double-click the ⬆ **ZooDBConnect** data source to connect to it.
2. In the Online Browser, right-click the tblFeedSuppliers table and select **Edit Data** from the context menu; alternatively, select the table and press **Ctrl+Alt+E**.
3. DatabaseSpy generates a SELECT statement in a new SQL Editor window, automatically executes it, and displays the table's data in the Result window for editing. A dialog box appears, informing you that editing of data is limited for this table. This warning merely indicates that the table is designed with automatically incremented columns. Such columns are automatically updated by the database upon inserting the record, so you cannot type data into such columns manually. All other columns can still be updated manually.
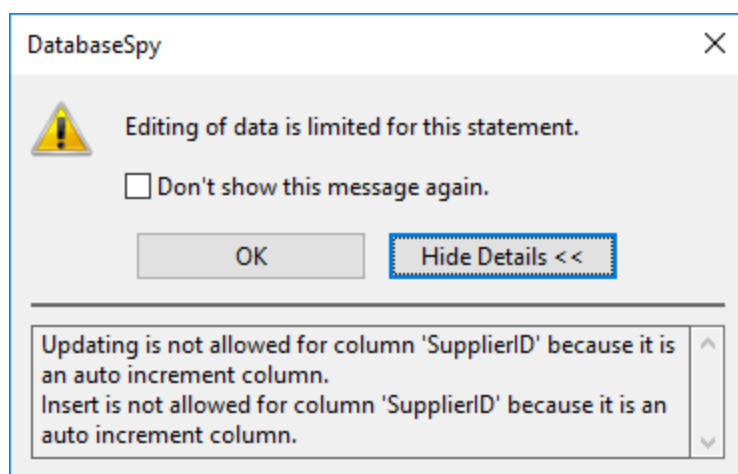
4. In the Result window, click the **Append a new row** button. Note that the cursor automatically appears in the "SupplierName" column because the value for the "SupplierID" column will be generated automatically by the database.

	SupplierID •	SupplierName	SupplierAddress •	SupplierCity •	SupplierState	SupplierTelephone •
5	5	Nemos Specialty Fish Food	26 Tower St.	NY	NY	(515)555-1887
6	6	The Feedery	410 Railroad Avenue	Minneapolis	MN	(515)555-4877
7	7	The Pampered Pet	245 Greenaway Street	Chicago	IL	(515)555-0706
8	8	ZooChow Central	3130 Ninth Avenue	NY	NY	(515)555-3345
9	0					

5.   Enter some data into the newly created row, for example:

	SupplierID •	SupplierName	SupplierAddress •	SupplierCity •	SupplierState •	SupplierTelephone •
5	5	Nemos Specialty Fish Food	26 Tower St.	NY	NY	(515)555-1887
6	6	The Feedery	410 Railroad Avenue	Minneapolis	MN	(515)555-4877
7	7	The Pampered Pet	245 Greenaway Street	Chicago	IL	(515)555-0706
8	8	ZooChow Central	3130 Ninth Avenue	NY	NY	(515)555-3345
9	0	Food4Animals	67 Ella Drive	Boston	MA	(515)555-8647

6.   Click Commit to save the changes to the database.

The edits are now committed to the database. On success, the message ✅**Execution was successful** appears at the base of the Result window.

Since the SQL Editor still displays the SELECT statement for this table, click **Execute** ▶ or **Execute for data editing** 📝 to retrieve data from the database again and display the table updated with your changes.

3.2.5   Querying the Database

This tutorial section illustrates various ways to query the "ZooDB" database in order to view, update, or delete data. This includes running SQL scripts and generating on demand SQL statements. In addition, you can always select, update, or delete data manually.

3.2.5.1 Adding Objects as Favorites

Some of the tables in the zoo database need to be queried more often than others. For example, tblMedicalTreatments needs to be updated whenever a veterinarian visits. Adding such tables as favorites provides you with quick access to these tables.

To add tables as favorites:

1.   Open the "ZooDB" project [72] created previously and double-click the 🔼 **ZooDBConnect** data source to connect to it.

2. In the Online Browser, right-click table tblAnimalFeed and select **Add to/Remove from Favorites** from the context menu. Alternatively, click the ⭐ icon next to the table, or select the table and press **Ctrl+F2**.

> To remove the object from favorites, use the same actions as for adding.

Tables that have been added as favorites are shown in bold in the Online Browser.
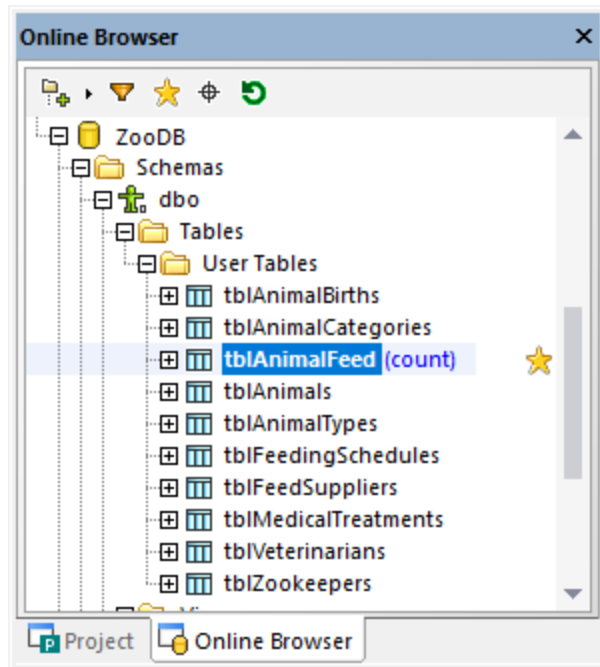
To switch to the "Favorites" view, click **Show Favorites** ⭐ at the top of the Online Browser. Now, only favorite items are shown in the Online Browser.
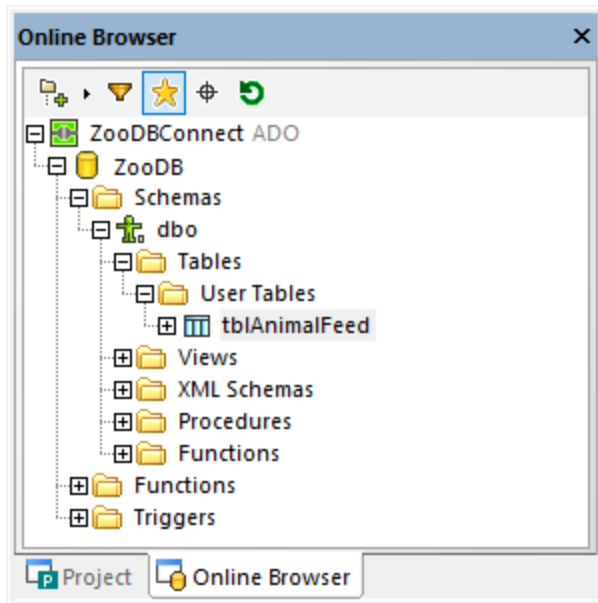
3.2.5.2 Using Drag and Drop to Create Queries

In DatabaseSpy, the easiest way to generate a query is to drag a database object like a table or column from the Online Browser into an SQL Editor window, for example:

1. Open the "ZooDB" project [72] created previously and double-click the 🔷 **ZooDBConnect** data source to connect to it.
2. Click the **SQL Editor** 🔲 toolbar button or select the menu option **File | New | SQL Editor** (**Ctrl+N**). A new SQL Editor window opens.
3. Drag and drop any table (tblZookepers, in this example) from the Online Browser into the SQL Editor.
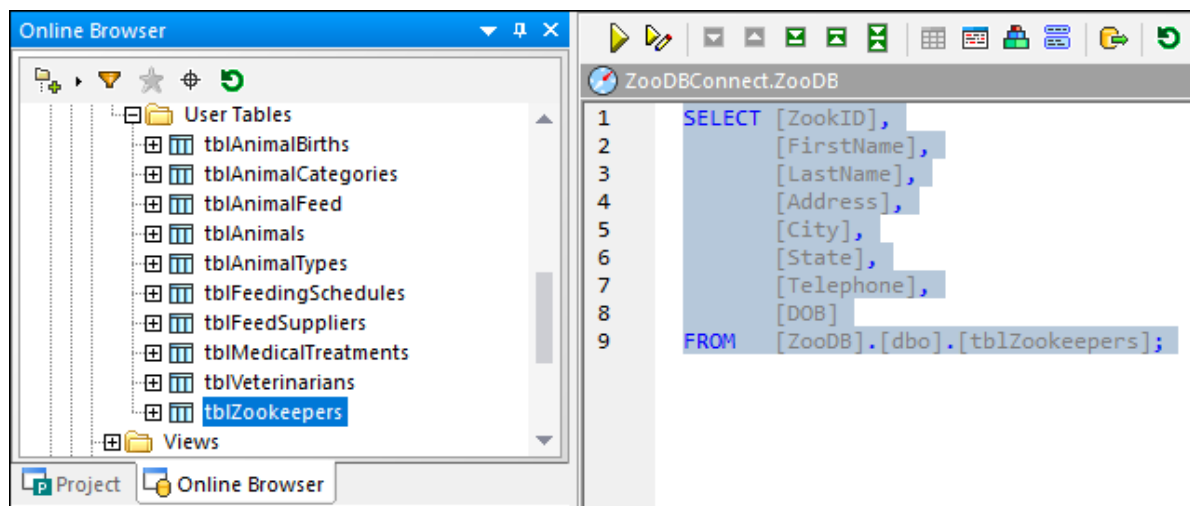
By default, this generates a SELECT statement. You can also modify the behavior above so that other statement kinds (like INSERT, CREATE, or UPDATE) are generated by default.

To configure the statement kind generated by default:

1. On the **Tools** menu, click **Options**.
2. Click the "Online Browser" page and choose the desired action next to each object (for example, **SELECT** for tables).
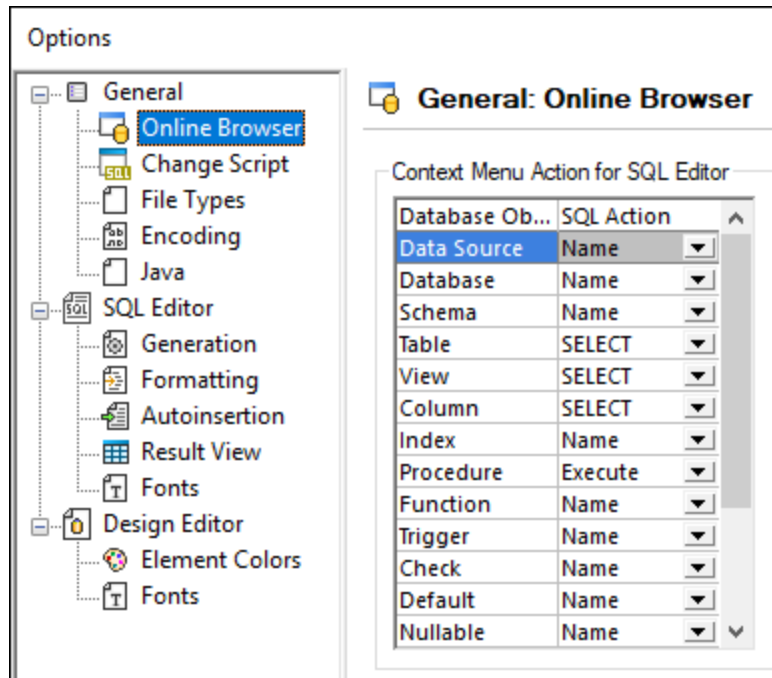
3.2.5.3 Running Multiple Named Queries

Whenever you run a query that returns some result from the database, the data is loaded in the Result window, for example:
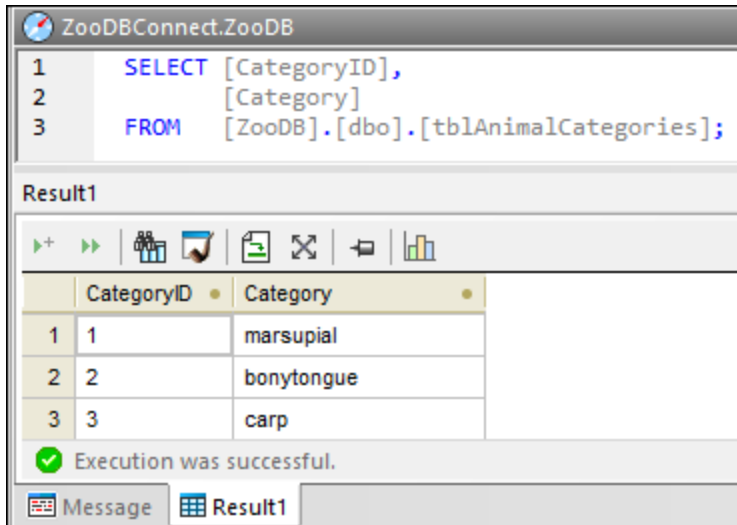
In the example above, the name of the Result window is *Result1*. This might not be informative enough when you work with multiple queries in the same project. To make the results more informative, it is possible to run multiple queries in the same script and display the results as separate *named* Result windows. For example, you may have one query that selects data from the `tblAnimalCategories` table and another one that selects data from `tblAnimalTypes` table. You can add both to the same SQL script and display their returned result sets separately. The name of each Result window is configurable.

To run multiple named queries:

1. Open the "ZooDB" project[72] created previously and double-click the 🔼 **ZooDBConnect** data source to connect to it.
2. In the "SQL" folder, double-click the file **animalType_queries.sql**. The file opens in an SQL Editor window. As shown below, it contains two SELECT queries. Importantly, each query starts with a comment like `--target: <tableName>`. You can easily create such comments either by typing them manually, or by clicking the **Define Target name** 🔲 toolbar button.
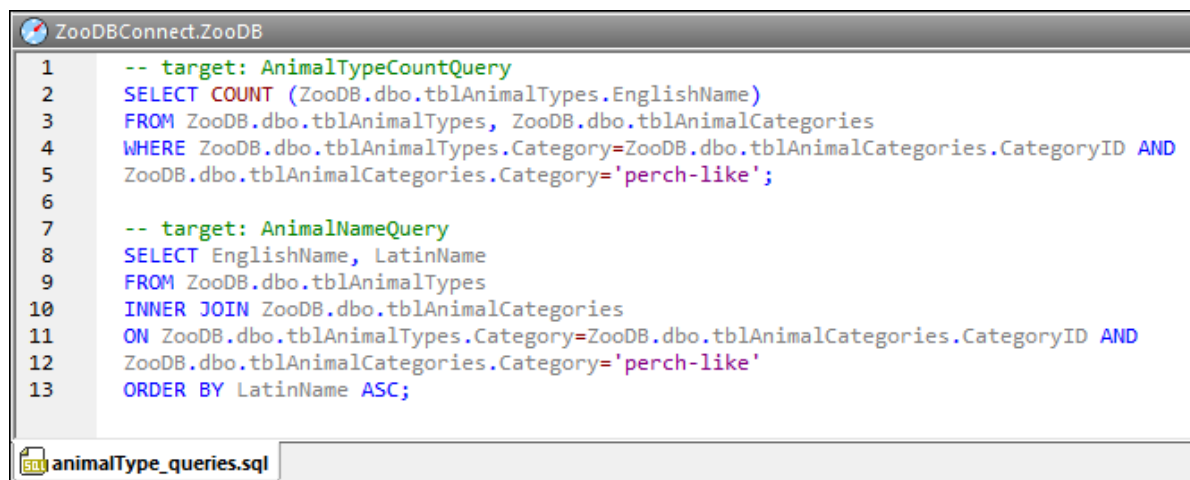
3. In the Properties window, make sure that either "Semicolons" or "SQL Grammar" is selected in the **Group statements for execution with** list.
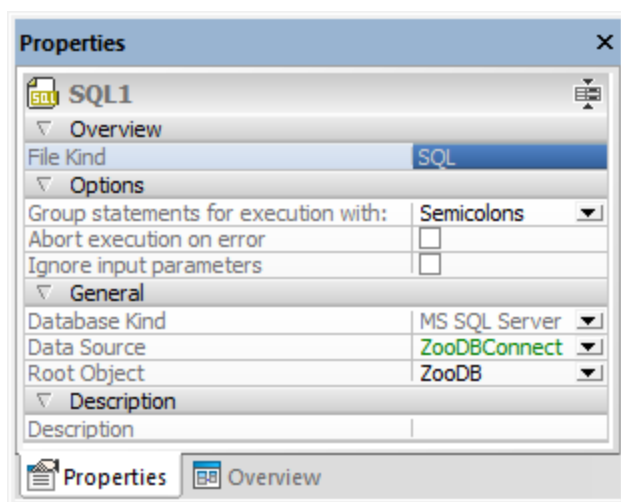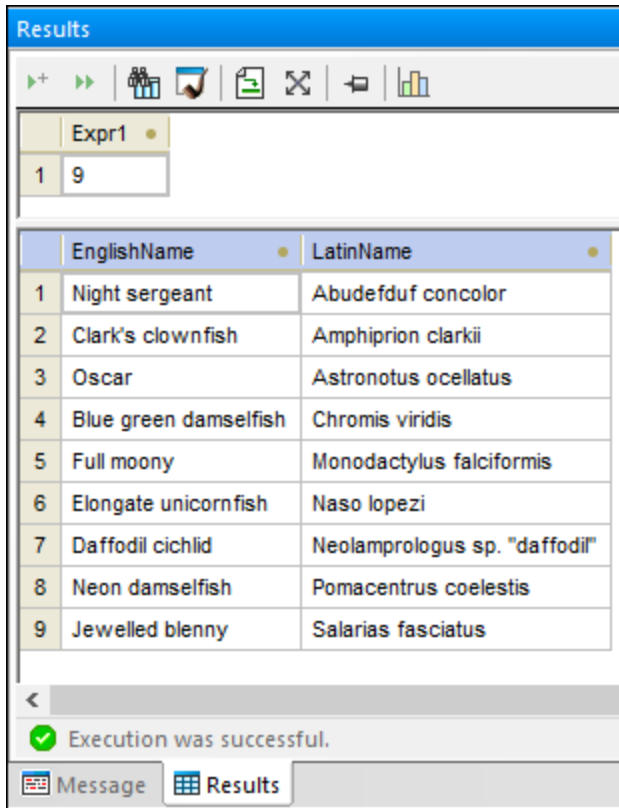
4. Click the **Execute** button or press **F5**.

The results of the two queries now appear in separate tabbed windows. As shown below, the name of each tab corresponds to the `--target` names that were set previously.
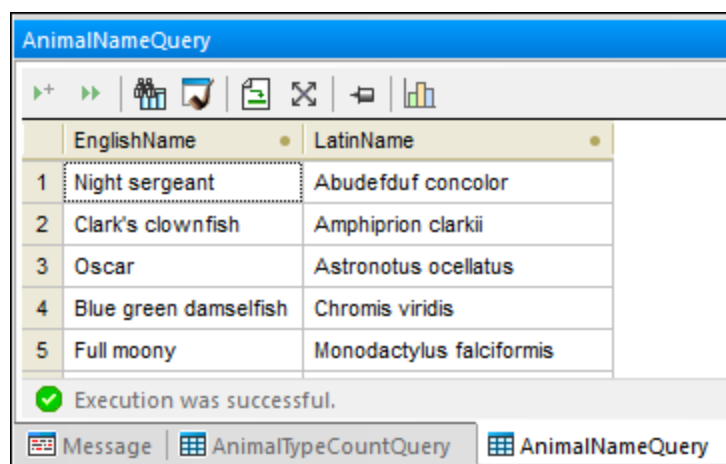
You can also stack all the result windows, for example:

Results

	Expr1 ●
1	9

	EnglishName ●	LatinName ●
1	Night sergeant	Abudefduf concolor
2	Clark's clownfish	Amphiprion clarkii
3	Oscar	Astronotus ocellatus
4	Blue green damselfish	Chromis viridis
5	Full moony	Monodactylus falciformis
6	Elongate unicornfish	Naso lopezi
7	Daffodil cichlid	Neolamprologus sp. "daffodil"
8	Neon damselfish	Pomacentrus coelestis
9	Jewelled blenny	Salarias fasciatus

✓ Execution was successful.

📄 Message   ⊞ Results

To stack all windows, click the **Show multiple results stacked** 🔳 option in the SQL Editor toolbar.

3.2.5.4 Editing Records

In the previous section of the tutorial, you have added a new column [87] for the e-mail address of the veterinarians to the tblVeterinarians table. Now you will enter the actual e-mail information into this new column.

To edit a table record:

1. Open the "ZooDB" project [72] created previously and double-click the 🔼 **ZooDBConnect** data source to connect to it.
2. In the Online Browser, right-click the tblVeterinarians table and select **Edit Data** from the context menu. Alternatively, select the table and press **Ctrl+Alt+E**. A dialog box appears, informing you that editing of data is limited for this table. This warning merely indicates that the table is designed with automatically incremented columns. Such columns are automatically updated by the database upon inserting the record, so you cannot type data into such columns manually. All other columns can still be updated manually.
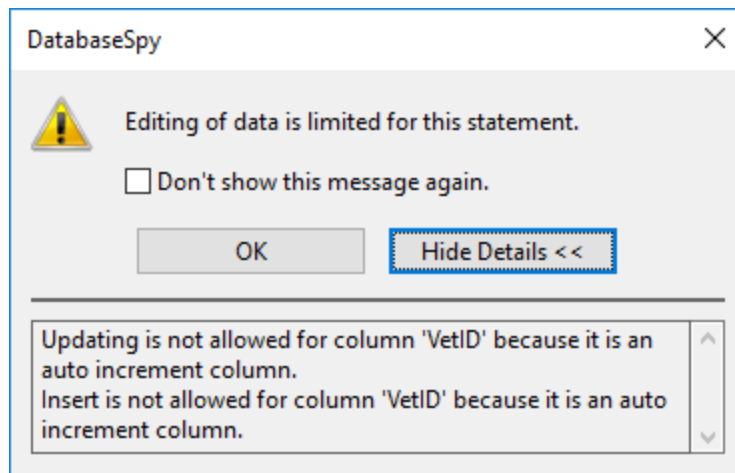
3. Double-click the "Email" cell in the first row and enter the new email value (in this example, "peggy@example.org").

4. Press the **Enter** key. The color of the updated cell changes.
5. Click the Commit button to save your changes to the database. The updated cell changes back to its default color and the update is committed to the database.

3.2.5.5 Deleting Records

One of the veterinarians will not be available for the zoo anymore and the zoo authorities want to have his records deleted from the database.

To delete the record:

1. Open the "ZooDB" project ⁷² created previously and double-click the **ZooDBConnect** data source to connect to it.
2. In the Online Browser, right-click the `tblVeterinarians` table and select **Edit Data** from the context menu. Alternatively, select the table and press **Ctrl+Alt+E**.
3. Click somewhere into the sixth row of the result grid and then click the **Delete row** button. The background color of the row as well as the text color change to indicate that this row is to be deleted from the database.

FirstName •	LastName •	Address •	City •	State •	Telephone •	DOB •	email •
Peggy	Hoofman	12 Bivalve Boulevard	Concord	NH	(515)555-2759	[NULL]	peggy@example.org
Richard	Scales	49 Whaletooth Avenue	Hartford	CT	(515)555-2843	[NULL]	[NULL]
Victoria	Wormworth	31 Marsupial Place	Newark	NJ	(515)555-9352	[NULL]	[NULL]
Danielle	Batwinger	238 Ella Funt Drive	Sacramento	CA	(515)555-7519	[NULL]	[NULL]
Robin	Fowler	183 Lionheart Drive	Albuquerque	NM	(515)555-2810	[NULL]	[NULL]
Jacob	Fisher	87 Bearpew Circle	Milwaukee	WI	(515)555-7209	[NULL]	[NULL]

The row is now marked for deletion but not actually deleted in the database. If you change your mind about deleting the record, click **Undo all changes** 🗷 to discard all changes from the grid.

4. Click the Commit button. The row is removed from the result grid and deleted from the database.

3.2.6   Exporting Database Data

In this section of the tutorial, you will learn how to export data from the "ZooDB" database to XML files.

3.2.6.1 Exporting Tables to XML

With DatabaseSpy, you can export database data to XML, CSV, HTML, and Excel formats. You can export either selected tables or the result of custom SELECT statements. The instructions below show you how to export data from selected database tables, as XML.

To export table data to XML:

1. Click the **Export** toolbar button. A dialog box opens.
2. Click the **XML** button on the left side of the dialog box.
3. Make sure that the **Source** option is set to **Table**.
4. Browse the database structure and select the tables that are to be exported (in this example, tblAnimalCategories). You can optionally limit the amount of displayed data by using filtering tools such as Filters [26] 🔽, Favorites [28] ⭐, or Object Locator [28] ⊕. To hide objects not selected for export, select the **Show checked objects only** ☑ check box. To preview the table before exporting it, first select one of the tables to be exported, and then click the **Preview** button.
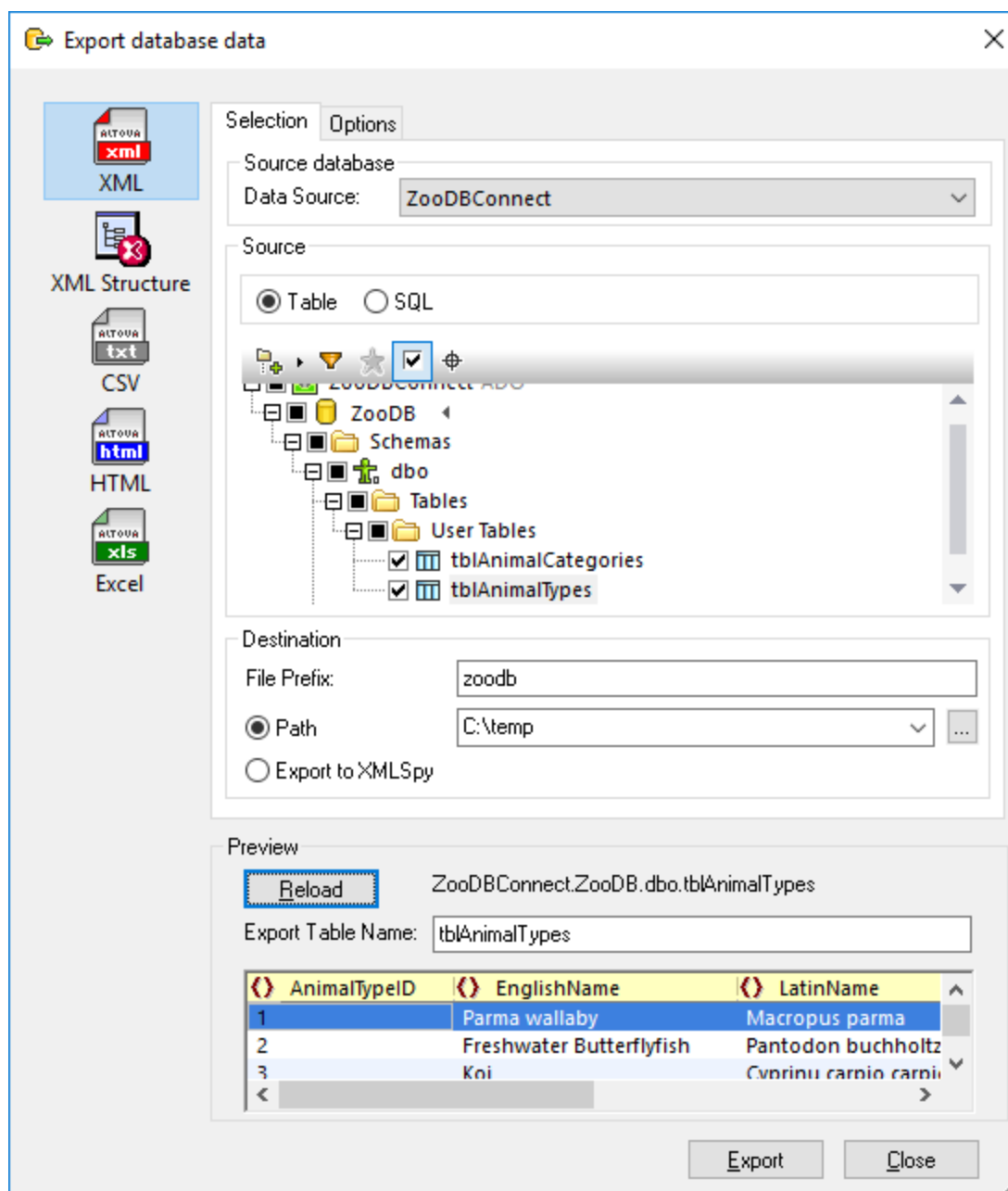
5. Under **File Prefix**, enter a prefix to be added to the exported files (in this example, the prefix is "zoodb").
6. Next to **Path**, enter the path to a destination directory.
7. Click **Export**. A separate XML file for each of the selected tables is created in the specified directory.

By default, the XML file includes comments that describe the structure of the source table. You can change this, as well as other options, from the **Options** tab of the dialog box illustrated above.

```
 1        <?xml version="1.0" encoding="UTF-8"?>
 2      <Import name="zoodbdbo.tblAnimalTypes" query="SELECT [AnimalTypeID], [EnglishName],
 3          <!--MS SQL Server-->
 4          <!--Column name:AnimalTypeID type:int maxlength:4-->
 5          <!--Column name:EnglishName type:varchar length:50 maxlength:8000-->
 6          <!--Column name:LatinName type:varchar length:50 maxlength:8000-->
 7          <!--Column name:AreaOfOrigin type:varchar length:50 maxlength:8000-->
 8          <!--Column name:Category type:int maxlength:4-->
 9          <Row>
10              <AnimalTypeID>1</AnimalTypeID>
11              <EnglishName>Parma wallaby</EnglishName>
12              <LatinName>Macropus parma</LatinName>
13              <AreaOfOrigin>Australia</AreaOfOrigin>
14              <Category>1</Category>
15          </Row>
```

For information about other options and target formats when exporting data from a database, see Exporting Database Data [406].

4   Connecting to a Data Source

In the most simple case, a database can be a local file such as a Microsoft Access or SQLite database file. In a more advanced scenario, a database may reside on a remote or network database server which does not necessarily use the same operating system as the application that connects to it and consumes data. For example, while DatabaseSpy runs on a Windows operating system, the database from which you want to access data (for example, MySQL) might run on a Linux machine.

To interact with various database types, both remote and local, DatabaseSpy relies on the data connection interfaces and database drivers that are already available on your operating system or released periodically by the major database vendors. In the constantly evolving landscape of database technologies, this approach caters for better cross-platform flexibility and interoperability.

The following diagram illustrates, in a simplified way, data connectivity options available between DatabaseSpy (illustrated as a generic client application) and a data store (which may be a database server or database file).
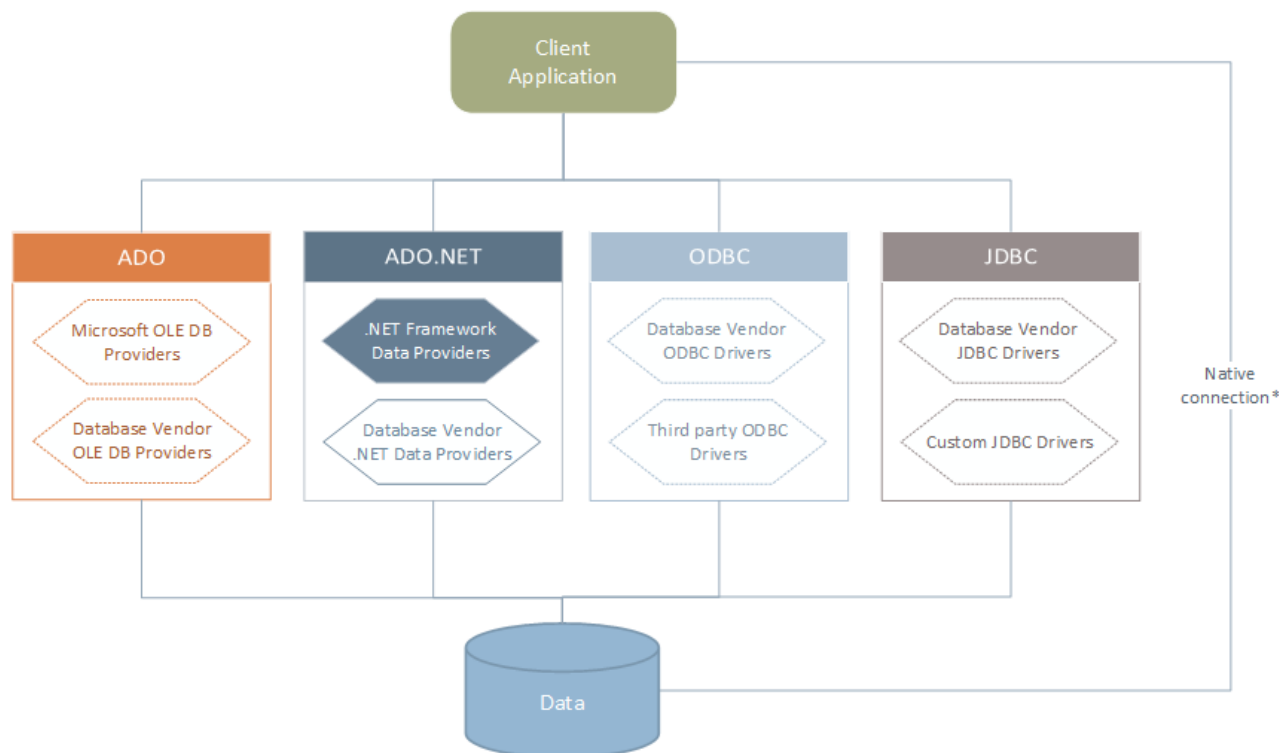
* Direct native connections are supported for SQLite, PostgreSQL, CouchDB and MongoDB databases. To connect to such databases, no additional drivers are required to be installed on your system.

As shown in the diagram above, DatabaseSpy can access any of the major database types through the following data access technologies:

- ADO (Microsoft® ActiveX® Data Objects), which, in its turn, uses an underlying OLE DB (Object Linking and Embedding, Database) provider
- ADO.NET (A set of libraries available in the Microsoft .NET Framework that enable interaction with data)

- JDBC (Java Database Connectivity)
- ODBC (Open Database Connectivity)

Note: Some ADO.NET providers are not supported or have limited support. See [ADO.NET Support Notes](#) [135].

About data access technologies

The data connection interface you should choose largely depends on your existing software infrastructure. You will typically choose the data access technology and the database driver which integrates tighter with the database system to which you want to connect. For example, to connect to a Microsoft Access 2013 database, you would build an ADO connection string that uses a native provider such as the **Microsoft Office Access Database Engine OLE DB Provider**. To connect to Oracle, on the other hand, you may want to download and install the latest JDBC, ODBC, or ADO.NET interfaces from the Oracle website.

While drivers for Windows products (such as Microsoft Access or SQL Server) may already be available on your Windows operating system, they may not be available for other database types. Major database vendors routinely release publicly available database client software and drivers which provide cross-platform access to the respective database through any combination of ADO, ADO.NET, ODBC, or JDBC. In addition to this, several third party drivers may be available for any of the above technologies. In most cases, there is more than one way to connect to the required database from your operating system, and, consequently, from DatabaseSpy. The available features, performance parameters, and the known issues will typically vary based on the data access technology or drivers used.

4.1   Start Database Connection Wizard

DatabaseSpy provides a Database Copnnection Wizard that guides you through the steps required to set up a connection to a data source. Before you go through the wizard steps, be aware that for some database types it is necessary to install and separately configure several database prerequisites, such as a database driver or database client software. These are normally provided by the respective database vendors, and include documentation tailored to your specific Windows version. For a list of database drivers grouped by database type, see [Database Drivers Overview](#) [119] .

To start the Database Connection Wizard (*see screenshot below*), do the following:

- On the **File** menu, click **Create a Database Connection**.

The Database Connection Wizard (*screenshot below*) is started. On the left hand side of the window, you can select the most suitable from the following ways to connect to your database:

- Connection Wizard, which prompts you to choose your database type and then guides you through the steps for connecting to a database of that type
- Select a data access technology: ADO, ADO.NET, ODBC, or JDBC
- Use an Altova global resource in which database connection is stored
- A native PostgreSQL connection

In the Connection Wizard pane (*see screenshot below*) databases can be sorted alphabetically by the name of the database type or by recent usage. Select the option you want in the *Sort By* combo box. After you have selected the database type to which you want to connect, click **Next**.
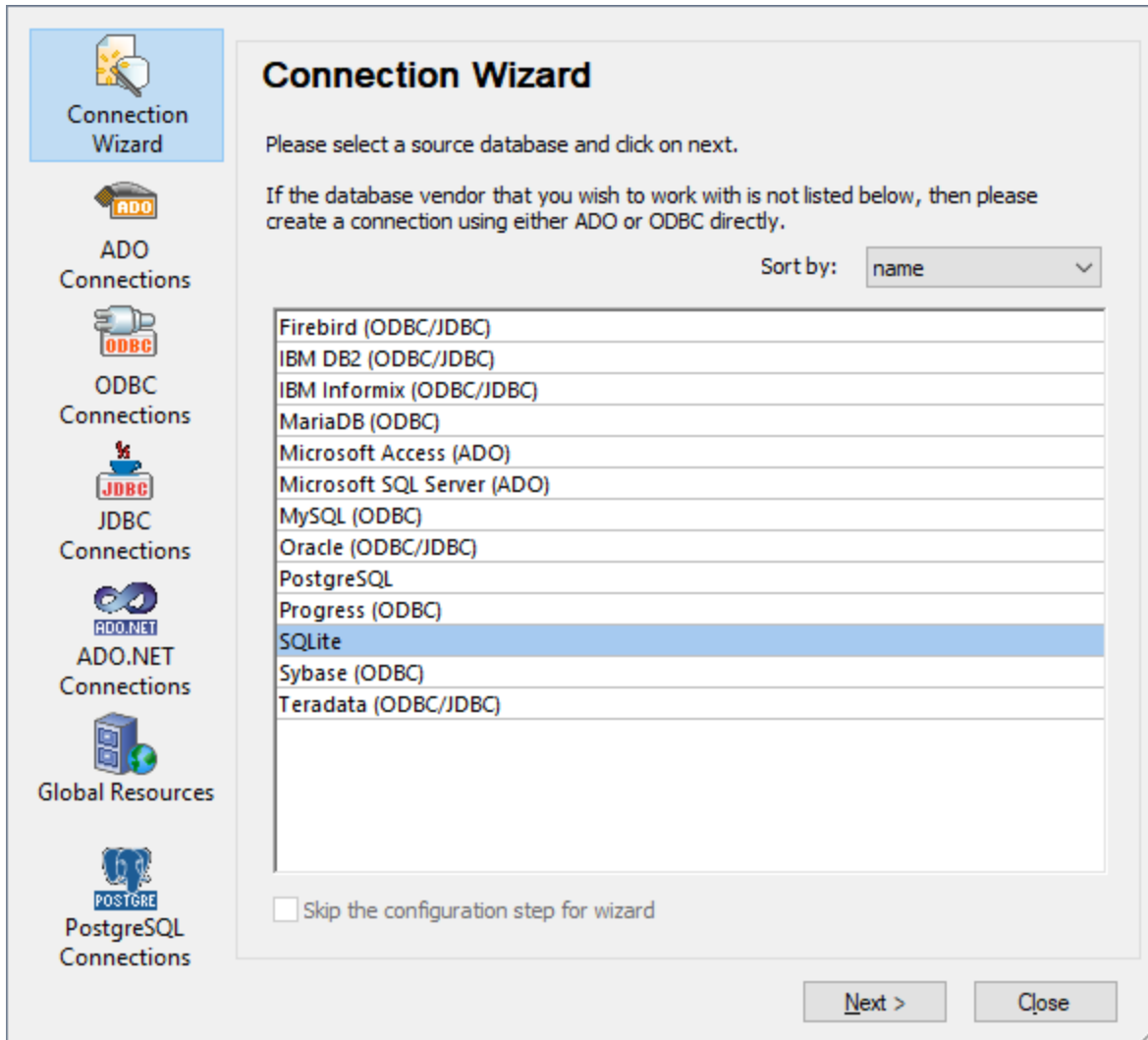
The wizard will take you through the next steps according to the database type, connection technology (ADO, ADO.NET, ODBC, JDBC), and driver that will be used. For examples applicable to each database type, see Database Connection Examples [152] .

Alternatively to using Connection Wizard, you can use one of the following database access technologies:

- Setting up an ADO Connection [122]
- Setting up an ADO.NET Connection [130]
- Setting up an ODBC Connection [137]
- Setting up a JDBC Connection [140]

4.2   Database Drivers Overview

The following table lists common database drivers you can use to connect to a particular database through a particular data access technology. Note that this list does not aim to be either exhaustive or prescriptive; you can use other native or third party alternatives in addition to the drivers shown below.

Even though a number of database drivers might be already available on your Windows operating system, you may still need to download an alternative driver. For some databases, the latest driver supplied by the database vendor is likely to perform better than the driver that shipped with the operating system.

Database vendors may provide drivers either as separate downloadable packages, or bundled with database client software. In the latter case, the database client software normally includes any required database drivers, or provides you with an option during installation to select the drivers and components you wish to install. Database client software typically consists of administration and configuration utilities used to simplify database administration and connectivity, as well as documentation on how to install and configure the database client and any of its components.

Configuring the database client correctly is crucial for establishing a successful connection to the database. Before installing and using the database client software, it is strongly recommended to read carefully the installation and configuration instructions of the database client; these may vary for each database version and for each Windows version.

To understand the capabilities and limitations of each data access technology with respect to each database type, refer to the documentation of that particular database product and also test the connection against your specific environment. To avoid common connectivity issues, note the following:

- Some ADO.NET providers are not supported or have limited support. See ADO.NET Support Notes [135].
- When installing a database driver, it is recommended that it has the same platform as the Altova application (32-bit or 64-bit). For example, if you are using a 32-bit Altova application on a 64-bit operating system, install the 32-bit driver, and set up your database connection using the 32-bit driver, see also Viewing the Available ODBC Drivers [139].
- When setting up an ODBC data source, it is recommended to create the data source name (DSN) as System DSN instead of User DSN. For more information, see Setting up an ODBC Connection [137].
- When setting up a JDBC data source, ensure that JRE (Java Runtime Environment) or Java Development Kit (JDK) is installed and that the CLASSPATH environment variable of the operating system is configured. For more information, see Setting up a JDBC Connection [140].
- For the installation instructions and support details of any drivers or database client software that you install from a database vendor, check the documentation provided with the installation package.

Database	Interface	Drivers
Firebird	ADO.NET	Firebird ADO.NET Data Provider (https://www.firebirdsql.org/en/additional-downloads/)
	JDBC	Firebird JDBC driver (https://www.firebirdsql.org/en/jdbc-driver/)
	ODBC	Firebird ODBC driver (https://www.firebirdsql.org/en/odbc-driver/)
IBM DB2	ADO	IBM OLE DB Provider for DB2
	ADO.NET	IBM Data Server Provider for .NET

Database	Interface	Drivers
	JDBC	IBM Data Server Driver for JDBC and SQLJ
	ODBC	IBM DB2 ODBC Driver
IBM DB2 for i	ADO	• IBM DB2 for i5/OS IBMDA400 OLE DB Provider • IBM DB2 for i5/OS IBMDARLA OLE DB Provider • IBM DB2 for i5/OS IBMDASQL OLE DB Provider
	ADO.NET	.NET Framework Data Provider for IBM i
	JDBC	IBM Toolbox for Java JDBC Driver
	ODBC	iSeries Access ODBC Driver
IBM Informix	ADO	IBM Informix OLE DB Provider
	JDBC	IBM Informix JDBC Driver
	ODBC	IBM Informix ODBC Driver
Microsoft Access	ADO	• Microsoft Jet OLE DB Provider • Microsoft Access Database Engine OLE DB Provider
	ADO.NET	.NET Framework Data Provider for OLE DB
	ODBC	• Microsoft Access Driver
MariaDB	ADO.NET	In the absence of a dedicated .NET connector for MariaDB, use **Connector/NET** for MySQL (https://dev.mysql.com/downloads/connector/net/).
	JDBC	MariaDB Connector/J (https://downloads.mariadb.org/)
	ODBC	MariaDB Connector/ODBC (https://downloads.mariadb.org/)
Microsoft SQL Server	ADO	• Microsoft OLE DB Driver for SQL Server (MSOLEDBSQL) • Microsoft OLE DB Provider for SQL Server (SQLOLEDB) • SQL Server Native Client (SQLNCLI)
	ADO.NET	• .NET Framework Data Provider for SQL Server • .NET Framework Data Provider for OLE DB
	JDBC	• Microsoft JDBC Driver for SQL Server (https://docs.microsoft.com/en-us/sql/connect/jdbc/microsoft-jdbc-driver-for-sql-server)
	ODBC	• ODBC Driver for Microsoft SQL Server (https://docs.microsoft.com/en-us/SQL/connect/odbc/download-odbc-driver-for-sql-server)
MySQL	ADO.NET	• Connector/NET (https://dev.mysql.com/downloads/connector/net/)
	JDBC	Connector/J (https://dev.mysql.com/downloads/connector/j/)
	ODBC	Connector/ODBC (https://dev.mysql.com/downloads/connector/odbc/)

Database	Interface	Drivers
Oracle	ADO	• Oracle Provider for OLE DB • Microsoft OLE DB Provider for Oracle
	ADO.NET	Oracle Data Provider for .NET (http://www.oracle.com/technetwork/topics/dotnet/index-085163.html)
	JDBC	• JDBC Thin Driver • JDBC Oracle Call Interface (OCI) Driver These drivers are typically installed during the installation of your Oracle database client. Connect through the OCI Driver (not the Thin Driver) if you are using the Oracle XML DB component.
	ODBC	• Microsoft ODBC for Oracle • Oracle ODBC Driver (typically installed during the installation of your Oracle database client)
PostgreSQL	JDBC	PostgreSQL JDBC Driver (https://jdbc.postgresql.org/download.html)
	ODBC	psqlODBC (https://odbc.postgresql.org/)
	Native Connection	Available. There is no need to install any drivers if using native connection.
Progress OpenEdge	JDBC	JDBC Connector (https://www.progress.com/jdbc/openedge)
	ODBC	ODBC Connector (https://www.progress.com/odbc/openedge)
SQLite	Native Connection	Available. There is no need to install any drivers if using native connection.
Sybase	ADO	Sybase ASE OLE DB Provider
	JDBC	jConnect™ for JDBC
	ODBC	Sybase ASE ODBC Driver
Teradata	ADO.NET	.NET Data Provider for Teradata (https://downloads.teradata.com/download/connectivity/net-data-provider-for-teradata)
	JDBC	Teradata JDBC Driver (https://downloads.teradata.com/download/connectivity/jdbc-driver)
	ODBC	Teradata ODBC Driver for Windows (https://downloads.teradata.com/download/connectivity/odbc-driver/windows)

4.3   ADO Connection

Microsoft ActiveX Data Objects (ADO) is a data access technology that enables you to connect to a variety of data sources through OLE DB. OLE DB is an alternative interface to ODBC or JDBC; it provides uniform access to data in a COM (Component Object Model) environment. ADO is a precursor of the newer ADO.NET[130] and is still one of the possible ways to connect to Microsoft native databases such as Microsoft Access or SQL Server, although you can also use it for other data sources.

Importantly, you can choose between multiple ADO providers, and some of them must be downloaded and installed on your workstation before you can use them. For example, for connecting to SQL Server, the following ADO providers are available:

- Microsoft OLE DB *Driver* for SQL Server (MSOLEDBSQL)
- Microsoft OLE DB *Provider* for SQL Server (SQLOLEDB)
- SQL Server Native Client (SQLNCLI)

From the providers listed above, the recommended one is MSOLEDBSQL; you can download it from https://docs.microsoft.com/en-us/sql/connect/oledb/download-oledb-driver-for-sql-server?view=sql-server-ver15. Note that it must match the platform of DatabaseSpy (32-bit or 64-bit). The SQLOLEDB and SQLNCLI providers are considered deprecated and thus are not recommended.

The **Microsoft OLE DB Provider for SQL Server (SQLOLEDB)** is known to have issues with parameter binding of complex queries like Common Table Expressions (CTE) and nested SELECT statements.

To set up an ADO connection:

1. Start the database connection wizard[117].
2. Click **ADO Connections**.
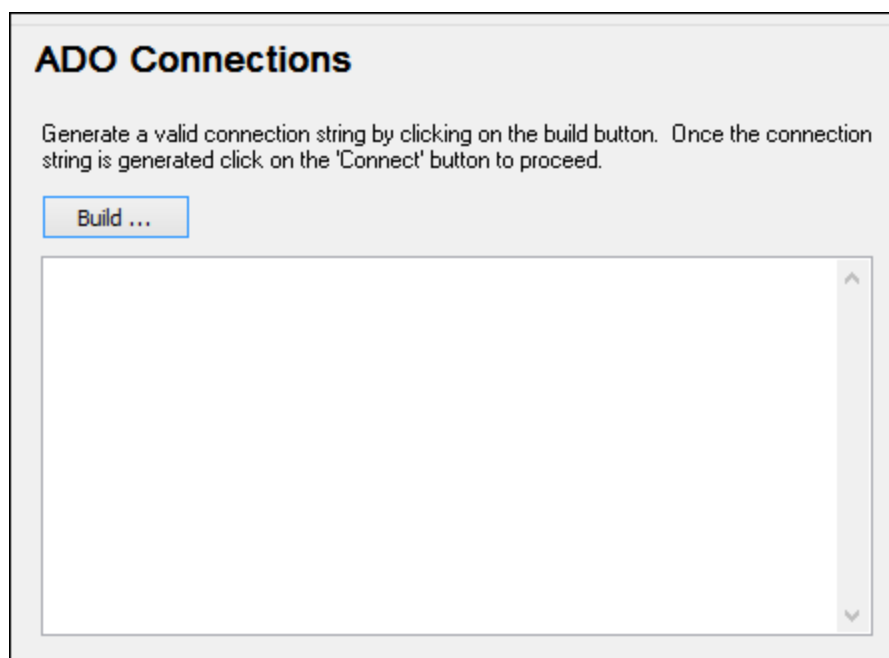
3. Click **Build**.
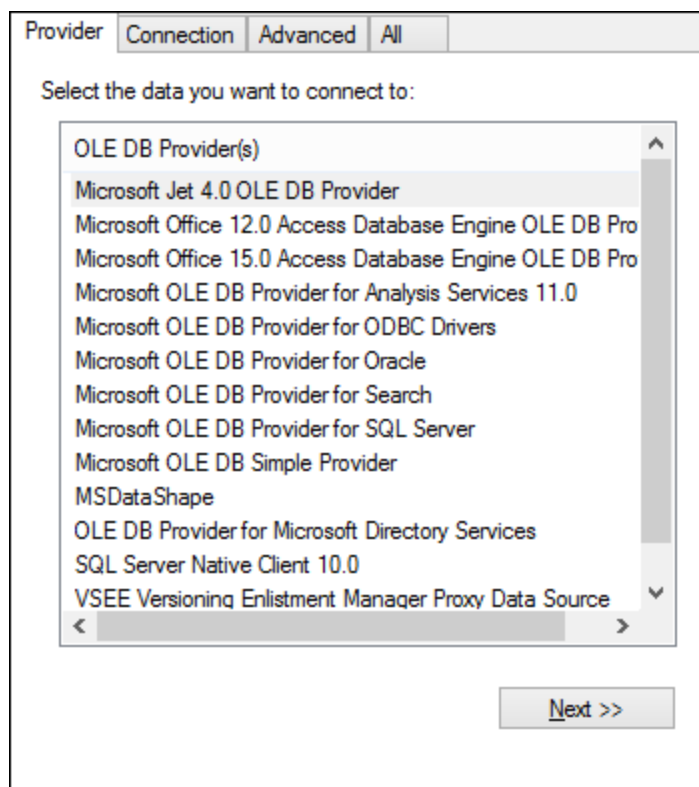
4. Select the data provider through which you want to connect. The table below lists a few common scenarios.

To connect to this database...	Use this provider...
Microsoft Access	• **Microsoft Office Access Database Engine OLE DB Provider** (recommended) • **Microsoft Jet OLE DB Provider** If the **Microsoft Office Access Database Engine OLE DB Provider** is not available in the list, make sure that you have installed either Microsoft Access or the Microsoft Access Database Engine Redistributable (https://www.microsoft.com/en-us/download/details.aspx?id=54920) on your computer.
SQL Server	• **Microsoft OLE DB Driver for SQL Server (MSOLEDBSQL)** - this is the recommended OLE DB provider. In order for this provider to appear in the list, it must be downloaded from https://docs.microsoft.com/en-us/sql/connect/oledb/download-oledb-driver-for-sql-server?view=sql-server-ver15 and installed. • **Microsoft OLE DB Provider for SQL Server (OLEDBSQL)** • **SQL Server Native Client (SQLNCLI)**
Other database	Select the provider applicable to your database. If an OLE DB provider to your database is not available, install the required driver from the database vendor (see Database Drivers Overview [119]). Alternatively, set up an ADO.NET, ODBC, or JDBC connection. If the operating system has an ODBC driver to the required database, you could also use the **Microsoft OLE DB Provider for ODBC Drivers**, or preferably opt for an ODBC connection [137].

5. Having selected the provider of choice, click **Next** and complete the wizard.

The subsequent wizard steps are specific to the provider you chose. For SQL Server, you will need to provide or select the host name of the database server, the authentication method, the database name, as well as the database username and password. For an example, see Connecting to Microsoft SQL Server (ADO) [173]. For Microsoft Access, you will be asked to browse for or provide the path to the database file. For an example, see Connecting to Microsoft Access (ADO) [171].

The complete list of initialization properties (connection parameters) is available in the **All** tab of the connection dialog box—these properties vary depending on the chosen provider and may need to be set explicitly in order for the connection to be possible. The following sections provide guidance on configuring the basic initialization properties for Microsoft Access and SQL Server databases:

• Setting up the SQL Server Data Link Properties [126]
• Setting up the Microsoft Access Data Link Properties [127]

4.3.1   Connecting to an Existing Microsoft Access Database

This approach is suitable when you want to connect to a Microsoft Access database which is not password-protected. If the database is password-protected, set up the database password as shown in Connecting to Microsoft Access (ADO) [171] .

To connect to an existing Microsoft Access database:

1. Run the database connection wizard (see Starting the Database Connection Wizard [117]).
2. Select **Microsoft Access (ADO)**, and then click **Next**.
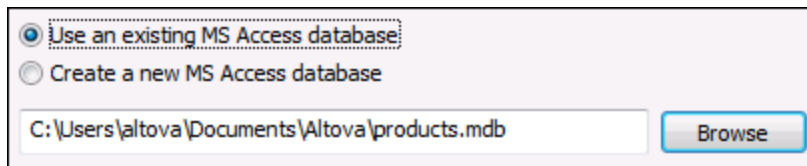
3. Select **Use an existing MS Access database**.
4. Browse for the database file, or enter the path to it (either relative or absolute).
5. Click **Connect**.

You can also open a Microsoft Access database directly from Windows Explorer, as follows:

1. Right-click an existing database file in Windows Explorer and select **Open With** from the context menu.
2. Choose **DatabaseSpy** from the list of suggestions. If this suggestion is not available, select **Choose another app**, and browse for the DatabaseSpy executable from the installation directory.
3. If the Database Connection Wizard appears, click **Close** to disregard it.

You can also drag and drop the database file into DatabaseSpy if the latter is already open.

Notes:

*   You can optionally configure DatabaseSpy from **Tools | Options** so as not to show the Database Connection Wizard whenever the application starts. To do this, clear the check box **Show create a database connection dialog** in the **General** group of settings.
*   If DatabaseSpy is open and a data source to the same database file already exists, it will be reused. In other words, a duplicate data source to the same database will not be created. This is applicable both when opening a database file with **Open With** and when using drag and drop.

4.3.2   Creating a New Microsoft Access Database

As an alternative to connecting to an existing database file, you can create a new Microsoft Access database file (.accdb, .mdb) and connect to it, even if Microsoft Access is not installed on the computer. The database file created by DatabaseSpy is empty. To create the required database structure, use Microsoft Access or the Design Editor of DatabaseSpy.

To create a new Microsoft Access database:

1. Run the database connection wizard (see Starting the Database Connection Wizard[117]).
2. Select **Microsoft Access (ADO)**, and then click **Next**.

○ Use an existing MS Access database
◉ Create a new MS Access database

C:\Users\altova\Documents\Altova\products.mdb   [Browse]

3. Select **Create a new MS Access database**, and then enter the path (either relative or absolute) of the database file to be created (for example, **c:\users\public\products.mdb**). Alternatively, click **Browse** to select a folder, type the name of the database file in the "File name" text box (for example, **products.mdb**), and click **Save**.

> **Notes**
> - Make sure that you have write permissions to the folder where you want to create the database file.
> - The database file name must have the **.mdb** or **.accdb** extension.

4. Click **Connect**.

4.3.3   Setting up the SQL Server Data Link Properties

When you connect to a Microsoft SQL Server database through ADO[122], you may need to set the following connection properties in the **All** tab of the Data Link Properties dialog box.
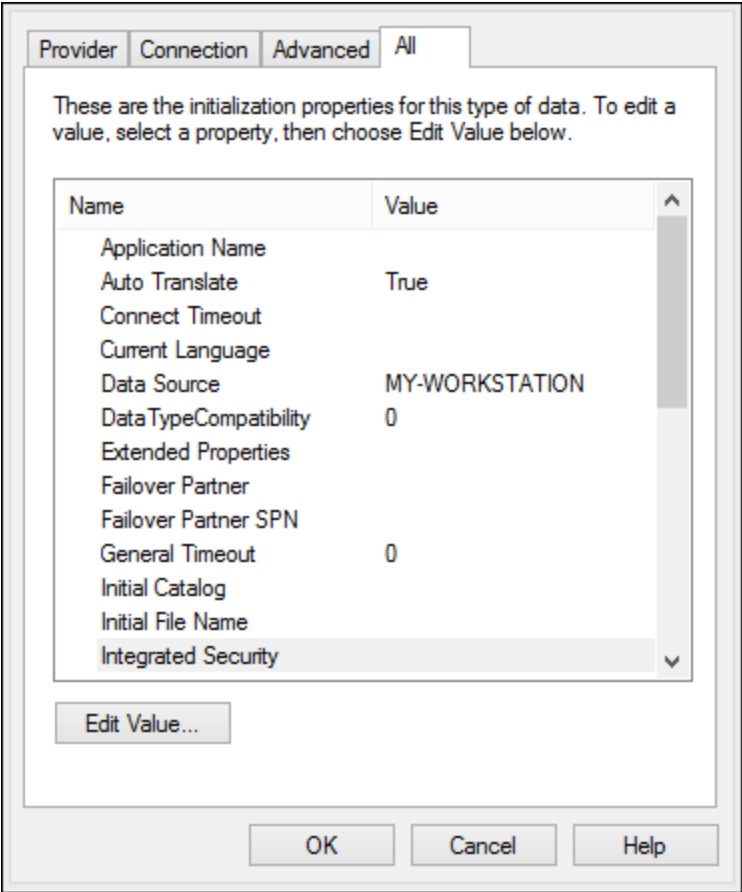
Data Link Properties dialog box

Property	Notes
Integrated Security	If you selected the **SQL Server Native Client** data provider on the **Provider** tab, set this property to a space character.
Persist Security Info	Set this property to **True**.

4.3.4   Setting up the Microsoft Access Data Link Properties

When you connect to a Microsoft Access database through ADO [122], you may need to set the following connection properties in the **All** tab of the Data Link Properties dialog box.

Data Link Properties dialog box

Property	Notes
Data Source	This property stores the path to the Microsoft Access database file. To avoid database connectivity issues, it is recommended to use the UNC (Universal Naming Convention) path format, for example: `\\anyserver\share$\filepath`
Jet OLEDB:System Database	This property stores the path to the workgroup information file. You may need to explicitly set the value of this property before you can connect to a Microsoft Access database. If you cannot connect due to a "workgroup information file" error, locate the workgroup information file (**System.MDW**) applicable to your user profile, and set the property value to the path of the **System.MDW** file.

Property	Notes
	Property Description Jet OLEDB:System database **Property Value** C:\Users\john.doe\AppData\Roaming\Microsoft\Access Reset Value OK Cancel
Jet OLEDB:Database Password	If the database is password-protected, set the value of this property to the database password. **Property Description** Jet OLEDB:Database Password **Property Value** •••••• Reset Value OK Cancel

4.4   ADO.NET Connection

ADO.NET is a set of Microsoft .NET Framework libraries designed to interact with data, including data from databases. To connect to a database from DatabaseSpy through ADO.NET, Microsoft .NET Framework 4 or later is required. As shown below, you connect to a database through ADO.NET by selecting a .NET provider and supplying a connection string.

A .NET data provider is a collection of classes that enables connecting to a particular type of data source (for example, a SQL Server, or an Oracle database), executing commands against it, and fetching data from it. In other words, with ADO.NET, an application such as DatabaseSpy interacts with a database through a data provider. Each data provider is optimized to work with the specific type of data source that it is designed for. There are two types of .NET providers:

1. Supplied by default with Microsoft .NET Framework.
2. Supplied by major database vendors, as an extension to the .NET Framework. Such ADO.NET providers must be installed separately and can typically be downloaded from the website of the respective database vendor.

Note: Certain ADO.NET providers are not supported or have limited support. See ADO.NET Support Notes [135].

To set up an ADO.NET connection:

1. Start the database connection wizard [117].
2. Click **ADO.NET Connections**.
3. Select a .NET data provider from the list.

> The list of providers available by default with the .NET Framework appears in the "Provider" list. Vendor-specific .NET data providers are available in the list only if they are already installed on your system. To become available, vendor-specific .NET providers must be installed into the GAC (Global Assembly Cache), by running the .msi or .exe file supplied by the database vendor.

4. Enter a database connection string. A connection string defines the database connection information, as semicolon-delimited key/value pairs of connection parameters. For example, a connection string such as `Data Source=DBSQLSERV;Initial Catalog=ProductsDB;User ID=dbuser;Password=dbpass` connects to the SQL Server database `ProductsDB` on server `DBSQLSERV`, with the user name `dbuser` and password `dbpass`. You can create a connection string by typing the key/value pairs directly into the "Connection String" dialog box. Another option is to create it with Visual Studio (see Creating a Connection String in Visual Studio [131]).

> The syntax of the connection string depends on the provider selected from the "Provider" list. For examples, see Sample ADO.NET Connection Strings [134].
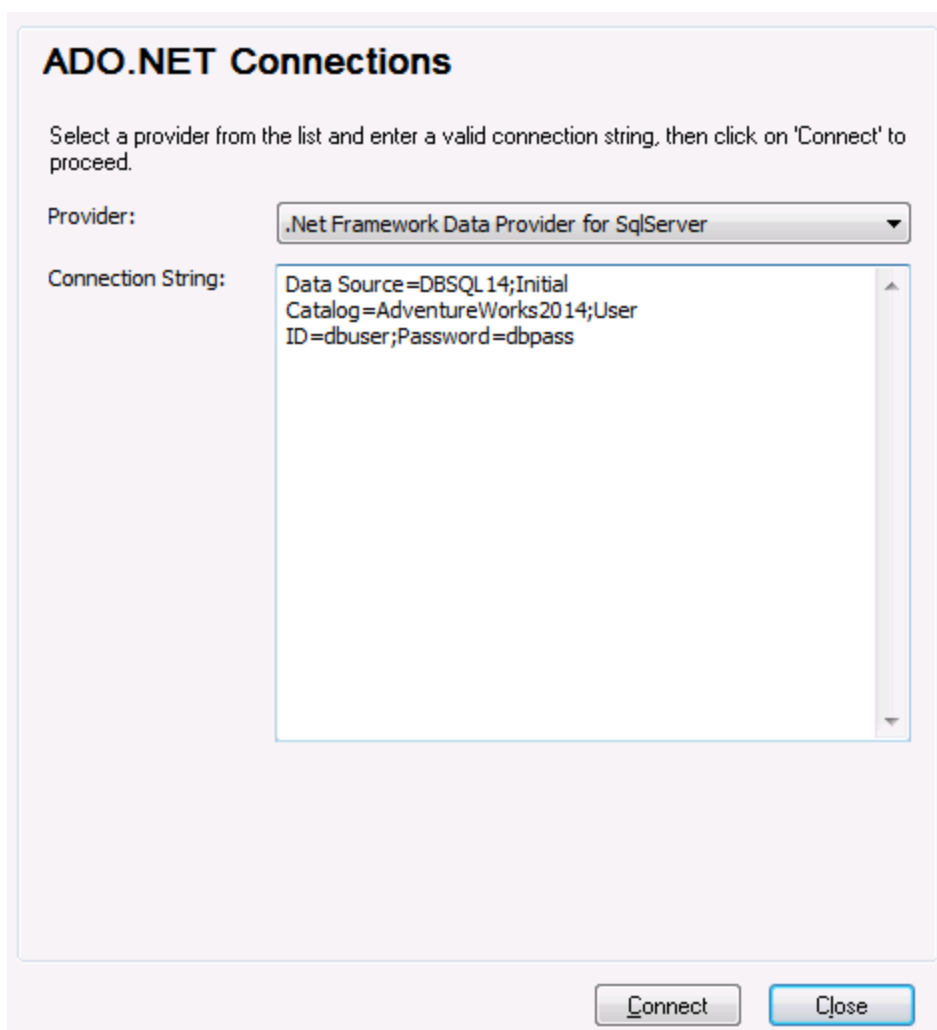
5.   Click **Connect**.

4.4.1   Creating a Connection String in Visual Studio

In order to connect to a data source using ADO.NET, a valid database connection string is required. The following instructions show you how to create a connection string from Visual Studio.

To create a connection string in Visual Studio:

1.   On the **Tools** menu, click **Connect to Database**.
2.   Select a data source from the list (in this example, Microsoft SQL Server). The Data Provider is filled automatically based on your choice.
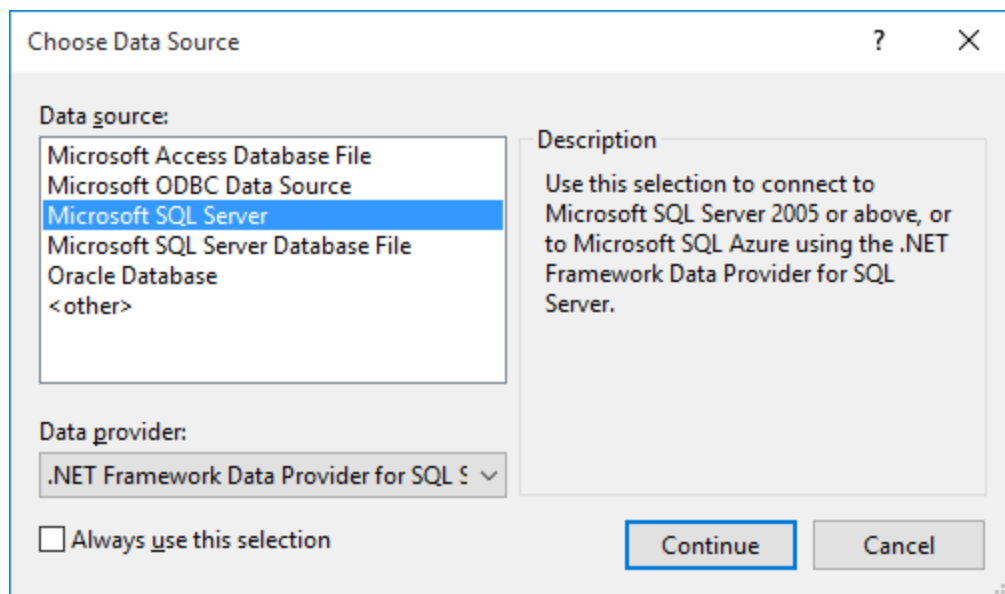
3.   Click **Continue**.

4. Enter the server host name and the user name and password to the database. In this example, we are connecting to the database `ProductsDB` on server `DBSQLSERV`, using SQL Server authentication.
5. Click **OK**.

If the database connection is successful, it appears in the Server Explorer window. You can display the Server Explorer window using the menu command **View | Server Explorer**. To obtain the database connection string, right-click the connection in the Server Explorer window, and select **Properties**. The connection string is now displayed in the Properties window of Visual Studio. Note that, before pasting the string into the "Connection String" box of DatabaseSpy, you will need to replace the asterisk (*) characters with the actual password.

4.4.2 Sample ADO.NET Connection Strings

To set up an ADO.NET connection, you need to select an ADO.NET provider from the database connection dialog box and enter a connection string (see also Setting up an ADO.NET Connection [130]). Sample ADO.NET connection strings for various databases are listed below under the .NET provider where they apply.

.NET Data Provider for Teradata

This provider can be downloaded from Teradata website (https://downloads.teradata.com/download/connectivity/net-data-provider-for-teradata). A sample connection string looks as follows:

```
Data Source=ServerAddress;User Id=user;Password=password;
```

.NET Framework Data Provider for IBM i

This provider is installed as part of *IBM i Access Client Solutions - Windows Application Package*. A sample connection string looks as follows:

```
DataSource=ServerAddress;UserID=user;Password=password;DataCompression=True;
```

For more information, see the ".NET Provider Technical Reference" help file included in the installation package above.

.NET Framework Data Provider for MySQL

This provider can be downloaded from MySQL website (https://dev.mysql.com/downloads/connector/net/). A sample connection string looks as follows:

```
Server=127.0.0.1;Uid=root;Pwd=12345;Database=test;
```

See also: https://dev.mysql.com/doc/connector-net/en/connector-net-programming-connecting-connection-string.html

.NET Framework Data Provider for SQL Server

A sample connection string looks as follows:

```
Data Source=DBSQLSERV;Initial Catalog=ProductsDB;User ID=dbuser;Password=dbpass
```

See also: https://msdn.microsoft.com/en-us/library/ms254500(v=vs.110).aspx

IBM DB2 Data Provider 10.1.2 for .NET Framework 4.0

```
Database=PRODUCTS;UID=user;Password=password;Server=localhost:50000;
```

Note: This provider is typically installed with the IBM DB2 Data Server Client package. If the provider is missing from the list of ADO.NET providers after installing IBM DB2 Data Server Client package, refer to the following technical note: https://www-01.ibm.com/support/docview.wss?uid=swg21429586.

See also:
https://www.ibm.com/support/knowledgecenter/en/SSEPGG_10.1.0/com.ibm.swg.im.dbclient.adonet.ref.doc/doc/DB2ConnectionClassConnectionStringProperty.html

Oracle Data Provider for .NET (ODP.NET)

The installation package which includes the ODP.NET provider can be downloaded from the Oracle website (see http://www.oracle.com/technetwork/topics/dotnet/downloads/index.html). A sample connection string looks as follows:

```
Data Source=DSORCL;User Id=user;Password=password;
```

Where DSORCL is the name of the data source which points to an Oracle service name defined in the **tnsnames.ora** file, as described in Connecting to Oracle (ODBC)[186].

To connect without configuring a service name in the **tnsnames.ora** file, use a string such as:

```
Data Source=(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=host)(PORT=port)))
(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=MyOracleSID)));User
Id=user;Password=password;
```

See also: https://docs.oracle.com/cd/B28359_01/win.111/b28375/featConnecting.htm

4.4.3   ADO.NET Support Notes

The following table lists known ADO.NET database drivers that are currently not supported or have limited support in DatabaseSpy.

Database	Driver	Support notes
All databases	**.Net Framework Data Provider for ODBC**	Limited support. Known issues exist with Microsoft Access connections. It is recommended to use ODBC direct connections instead.
	.Net Framework Data Provider for OleDb	Limited support. Known issues exist with Microsoft Access connections. It is recommended to use ADO direct connections instead.
Firebird	**Firebird ADO.NET Data Provider**	Limited support. It is recommended to use ODBC or JDBC instead.
Informix	**IBM Informix Data Provider for**	Not supported. Use **DB2 Data Server**

Database	Driver	Support notes
	.NET Framework 4.0	**Provider** instead.
IBM DB2 for i (iSeries)	**.Net Framework Data Provider for i5/OS**	Not supported. Use **.Net Framework Data Provider for IBM i** instead, installed as part of the *IBM i Access Client Solutions - Windows Application Package*.
Oracle	**.Net Framework Data Provider for Oracle**	Limited support. Although this driver is provided with the .NET Framework, its usage is discouraged by Microsoft, because it is deprecated.
PostgreSQL	-	No ADO.NET drivers for this vendor are supported. Use a native connection instead.
Sybase	-	No ADO.NET drivers for this vendor are supported.

4.5   ODBC Connection

ODBC (Open Database Connectivity) is a widely used data access technology that enables you to connect to a database from DatabaseSpy. It can be used either as primary means to connect to a database, or as an alternative to native, OLE DB, or JDBC-driven connections.

To connect to a database through ODBC, first you need to create an ODBC data source name (DSN) on the operating system. This step is not required if the DSN has already been created, perhaps by another user of the operating system. The DSN represents a uniform way to describe the database connection to any ODBC-aware client application on the operating system, including DatabaseSpy. DSNs can be of the following types:

- System DSN
- User DSN
- File DSN

A *System* data source is accessible by all users with privileges on the operating system. A *User* data source is available to the user who created it. Finally, if you create a *File DSN*, the data source will be created as a file with the .dsn extension which you can share with other users, provided that they have installed the drivers used by the data source.

Any DSNs already available on your machine are listed by the database connection dialog box when you click **ODBC connections** on the ODBC connections dialog box.
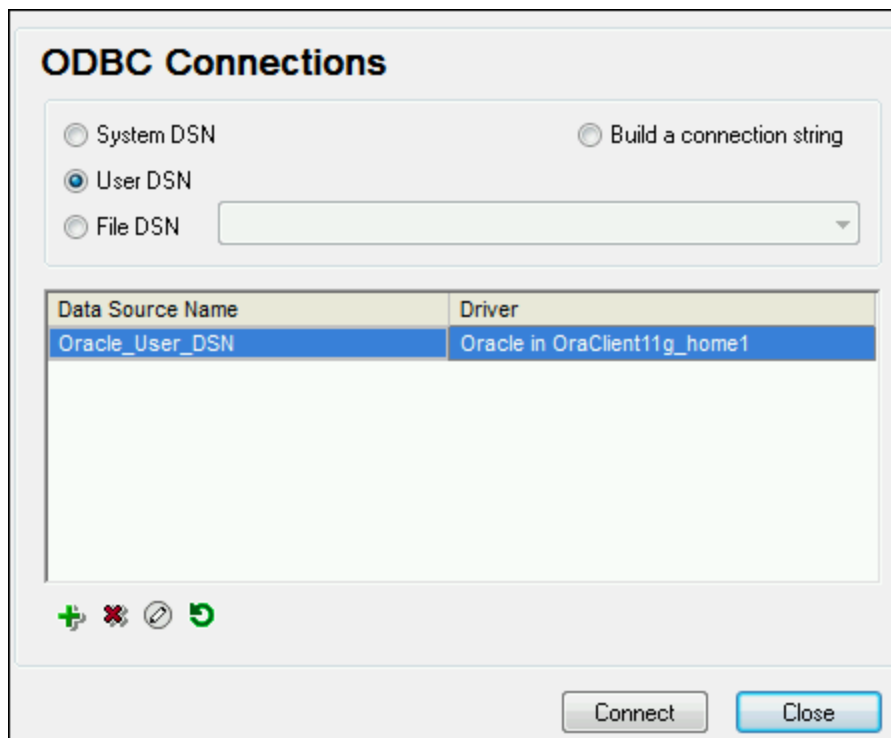
ODBC Connections dialog box

If a DSN to the required database is not available, the DatabaseSpy database connection wizard will assist you to create it; however, you can also create it directly on your Windows operating system. In either case, before

you proceed, ensure that the ODBC driver applicable for your database is in the list of ODBC drivers available to the operating system (see Viewing the Available ODBC Drivers [139]).

To connect by using a new DSN:

1. Start the database connection wizard [117].
2. On the database connection dialog box, click **ODBC Connections**.
3. Select a data source type (User DSN, System DSN, File DSN).

> To create a System DSN, you need administrative rights on the operating system, and DatabaseSpy must be run as administrator.

4. Click **Add** 💠 .
5. Select a driver, and then click **User DSN** or **System DSN** (depending on the type of the DSN you want to create). If the driver applicable to your database is not listed, download it from the database vendor and install it (see Database Drivers Overview [119]).
6. On the dialog box that pops up, fill in any driver specific connection information to complete the setup.

For the connection to be successful, you will need to provide the host name (or IP address) of the database server, as well as the database username and password. There may be other optional connection parameters—these parameters vary between database providers. For detailed information about the parameters specific to each connection method, consult the documentation of the driver provider. Once created, the DSN becomes available in the list of data source names. This enables you to reuse the database connection details any time you want to connect to the database. Note that User DSNs are added to the list of User DSNs whereas System DSNs are added to the list of System DSNs.

To connect by using an existing DSN:

1. Start the database connection wizard [117].
2. Click **ODBC Connections**.
3. Choose the type of the existing data source (User DSN, System DSN, File DSN).
4. Click the existing DSN record, and then click **Connect**.

To build a connection string based on an existing .dsn file:

1. Start the database connection wizard [117].
2. Click **ODBC Connections**.
3. Select **Build a connection string**, and then click **Build**.
4. If you want to build the connection string using a File DSN, click the **File Data Source** tab. Otherwise, click the **Machine Data Source** tab. (System DSNs and User DSNs are known as "Machine" data sources.)
5. Select the required .dsn file, and then click **OK**.

To connect by using a prepared connection string:

1. Start the database connection wizard [117].
2. Click **ODBC Connections**.

3. Select **Build a connection string**.
4. Paste the connection string into the provided box, and then click **Connect**.

4.5.1   Available ODBC Drivers

You can view the ODBC drivers available on your operating system in the ODBC Data Source Administrator. You can access the ODBC Data Source Administrator (**Odbcad32.exe**) from the Windows Control Panel, under **Administrative Tools**. On 64-bit operating systems, there are two versions of this executable:

- The 32-bit version of the **Odbcad32.exe** file is located in the **C:\Windows\SysWoW64** directory (assuming that **C:** is your system drive).
- The 64-bit version of the **Odbcad32.exe** file is located in the **C:\Windows\System32** directory.

Any installed 32-bit database drivers are visible in the 32-bit version of ODBC Data Source Administrator, while 64-bit drivers—in the 64-bit version. Therefore, ensure that you check the database drivers from the relevant version of ODBC Data Source Administrator.
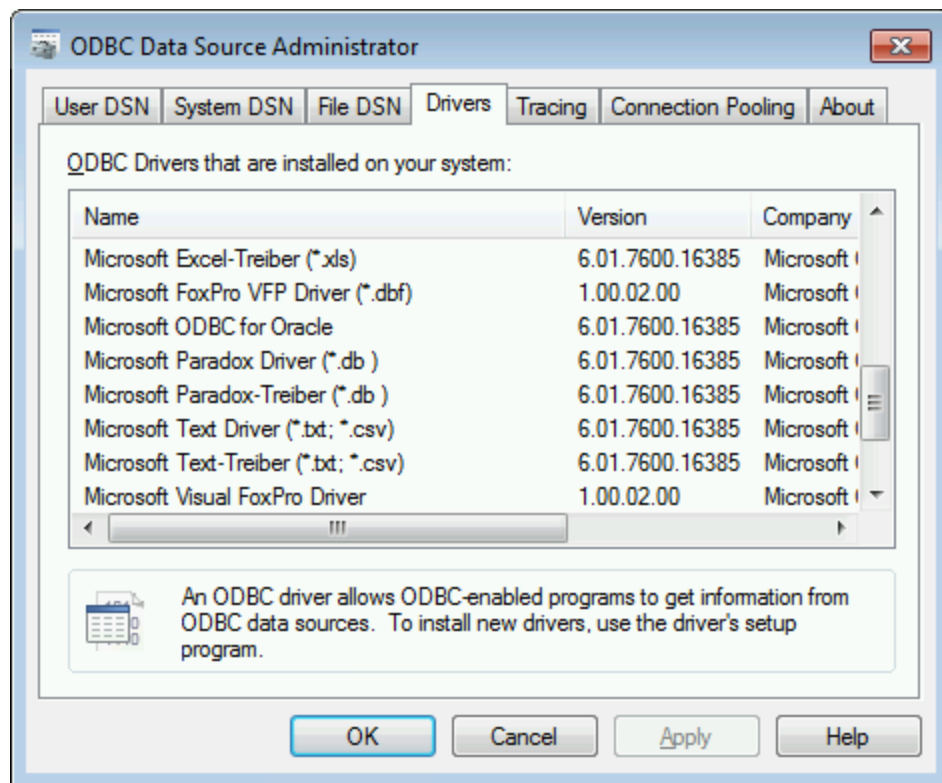
ODBC Data Source Administrator

If the driver to your target database does not exist in the list, or if you want to add an alternative driver, you will need to download it from the database vendor (see Database Drivers Overview[119]). Once the ODBC driver is available on your system, you are ready to create ODBC connections with it (see Setting up an ODBC Connection[137]).

4.6   JDBC Connection

JDBC (Java Database Connectivity) is a database access interface which is part of the Java software platform from Oracle. JDBC connections are generally more resource-intensive than ODBC connections but may provide features not available through ODBC.

Prerequisites

- JRE (Java Runtime Environment) or Java Development Kit (JDK) must be installed. This may be either Oracle JDK or an open source build such as Oracle OpenJDK. DatabaseSpy will determine the path to the Java Virtual Machine (JVM) from the following locations, in this order: a) The custom JVM path you may have set in application **Options**, see Java Settings [423]; b) The JVM path found in the Windows registry; c) The JAVA_HOME environment variable.
- Make sure that the platform of DatabaseSpy (32-bit, 64-bit) matches that of the JRE/JDK.
- The JDBC drivers from the database vendor must be installed. These may be JDBC drivers installed as part of a database client installation, or JDBC libraries (.jar files) downloaded separately, if available and supported by the database, see also Database Connection Examples [152].
- The CLASSPATH environment variable must include the path to the JDBC driver (one or several .jar files) on your Windows operating system. When you install some database clients, the installer may configure this variable automatically. See also Configuring the CLASSPATH [142].

Connecting to SQL Server via JDBC with Windows credentials

If you connect to SQL Server through JDBC with Windows credentials (integrated security), note the following:

- The **sqljdbc_auth.dll** file included in the JDBC driver package must be copied to a directory that is on the system PATH environment variable. There are two such files, one for the x86 and one for x64 platform. Make sure that you add to the PATH the one that corresponds to your JDK platform.
- The JDBC connection string must include the property `integratedSecurity=true`.

For further information, refer to *Microsoft JDBC driver for SQL Server* documentation, https://docs.microsoft.com/en-us/sql/connect/jdbc/building-the-connection-url.

Setting up a JDBC connection

1. Start the database connection wizard [117].
2. Click **JDBC Connections**.
3. Optionally, enter a semicolon-separated list of .jar file paths in the "Classpaths" text box. The .jar libraries entered here will be loaded into the environment in addition to those already defined in the CLASSPATH environment variable. When you finish editing the "Classpaths" text box, any JDBC drivers found in the source .jar libraries are automatically added to the "Driver" list (see the next step).

4. Next to "Driver", select a JDBC driver from the list, or enter a Java class name. Note that this list contains any JDBC drivers configured through the CLASSPATH environment variable (see Configuring the CLASSPATH [142]), as well as those found in the "Classpaths" text box.

> The JDBC driver paths defined in the CLASSPATH variable, as well as any .jar file paths entered directly in the database connection dialog box are all supplied to the Java Virtual Machine (JVM). The JVM then decides which drivers to use in order to establish a connection. It is recommended to keep track of Java classes loaded into the JVM so as not to create potential JDBC driver conflicts and avoid unexpected results when connecting to the database.

5. Enter the username and password to the database in the corresponding boxes.
6. In the Database URL text box, enter the JDBC connection URL (string) in the format specific to your database type. The following table describes the syntax of JDBC connection URLs (strings) for common database types.

Database	JDBC Connection URL
Firebird	`jdbc:firebirdsql://`**`<host>`**`[:`**`<port>`**`]/`**`<database path or alias>`**
IBM DB2	`jdbc:db2://`**`hostName`**`:`**`port`**`/`**`databaseName`**
IBM DB2 for i	`jdbc:as400://`**`[host]`**

Database	JDBC Connection URL
IBM Informix	`jdbc:informix-sqli://`**`hostName:port`**`/`**`databaseName`**`:INFORMIXSERVER=`**`myserver`**
MariaDB	`jdbc:mariadb://`**`hostName:port`**`/`**`databaseName`**
Microsoft SQL Server	`jdbc:sqlserver://`**`hostName:port`**`;`**`databaseName`**`=name`
MySQL	`jdbc:mysql://`**`hostName:port`**`/`**`databaseName`**
Oracle	`jdbc:oracle:thin:@`**`hostName:port:SID`** `jdbc:oracle:thin:@//`**`hostName:port`**`/`**`service`**
Oracle XML DB	`jdbc:oracle:oci:@//`**`hostName:port`**`:`**`service`**
PostgreSQL	`jdbc:postgresql://`**`hostName:port`**`/`**`databaseName`**
Progress OpenEdge	`jdbc:datadirect:openedge://`**`host:port`**`;databaseName=`**`db_name`**
Sybase	`jdbc:sybase:Tds:`**`hostName:port`**`/`**`databaseName`**
Teradata	`jdbc:teradata://`**`databaseServerName`**

Note: Syntax variations to the formats listed above are also possible (for example, the database URL may exclude the port or may include the username and password to the database). Check the documentation of the database vendor for further details.

7. Click **Connect**.

4.6.1   Configuring the CLASSPATH

The `CLASSPATH` environment variable is used by the Java Runtime Environment (JRE) or the Java Development Kit (JDK) to locate Java classes and other resource files on your operating system. When you connect to a database through JDBC, this variable must be configured to include the path to the JDBC driver on your operating system, and, in some cases, the path to additional library files specific to the database type you are using.

The following table lists sample file paths that must be typically included in the `CLASSPATH` variable. Importantly, you may need to adjust this information based on the location of the JDBC driver on your system, the JDBC driver name, as well as the JRE/JDK version present on your operating system. To avoid connectivity problems, check the installation instructions and any pre-installation or post-installation configuration steps applicable to the JDBC driver installed on your operating system.

Database	Sample CLASSPATH entries
Firebird	`C:\Program Files\Firebird\Jaybird-2.2.8-JDK_1.8\jaybird-full-2.2.8.jar`
IBM DB2	`C:\Program Files (x86)\IBM\SQLLIB\java\db2jcc.jar;C:\Program Files (x86)\IBM\SQLLIB\java\db2jcc_license_cu.jar;`

Database	Sample CLASSPATH entries
IBM DB2 for i	`C:\jt400\jt400.jar;`
IBM Informix	`C:\Informix_JDBC_Driver\lib\ifxjdbc.jar;`
Microsoft SQL Server	`C:\Program Files\Microsoft JDBC Driver 4.0 for SQL Server\sqljdbc_4.0\enu\sqljdbc.jar`
MariaDB	`<installation directory>\mariadb-java-client-2.2.0.jar`
MySQL	`<installation directory>\mysql-connector-java-`*version*`-bin.jar;`
Oracle	`ORACLE_HOME\jdbc\lib\ojdbc6.jar;`
Oracle (with XML DB)	`ORACLE_HOME\jdbc\lib\ojdbc6.jar;ORACLE_HOME\LIB\xmlparserv2.jar;` `ORACLE_HOME\RDBMS\jlib\xdb.jar;`
PostgreSQL	`<installation directory>\postgresql.jar`
Progress OpenEdge	`%DLC%\java\openedge.jar;%DLC%\java\pool.jar;` Note: Assuming the Progress OpenEdge SDK is installed on the machine, `%DLC%` is the directory where OpenEdge is installed.
Sybase	`C:\sybase\jConnect-7_0\classes\jconn4.jar`
Teradata	`<installation directory>\tdgssconfig.jar;<installation directory>\terajdbc4.jar`

- Changing the CLASSPATH variable may affect the behavior of Java applications on your machine. To understand possible implications before you proceed, refer to the Java documentation.
- Environment variables can be user or system. To change system environment variables, you need administrative rights on the operating system.
- After you change the environment variable, restart any running programs for settings to take effect. Alternatively, log off or restart your operating system.

To configure the CLASSPATH on Windows 7:

1. Open the **Start** menu and right-click **Computer**.
2. Click **Properties**.
3. Click **Advanced system settings**.
4. In the **Advanced** tab, click **Environment Variables**,
5. Locate the CLASSPATH variable under user or system environment variables, and then click Edit. If the CLASSPATH variable does not exist, click **New** to create it.
6. Edit the variable value to include the path on your operating system where the JDBC driver is located. To separate the JDBC driver path from other paths that may already be in the CLASSPATH variable, use the semi-colon separator (;).

To configure the CLASSPATH on Windows 10:

1. Press the Windows key and start typing "environment variables".
2. Click the suggestion **Edit the system environment variables**.
3. Click **Environment Variables**.
4. Locate the CLASSPATH variable under user or system environment variables, and then click **Edit**. If the CLASSPATH variable does not exist, click **New** to create it.
5. Edit the variable value to include the path on your operating system where the JDBC driver is located. To separate the JDBC driver path from other paths that may already be in the CLASSPATH variable, use the semi-colon separator (;).

4.7   PostgreSQL Connection

Connections to PostgreSQL databases can be set up either as native connections, or connections via ODBC, JDBC, and other drivers. The advantage of setting up a native connection is that no drivers are required to be installed on your system.
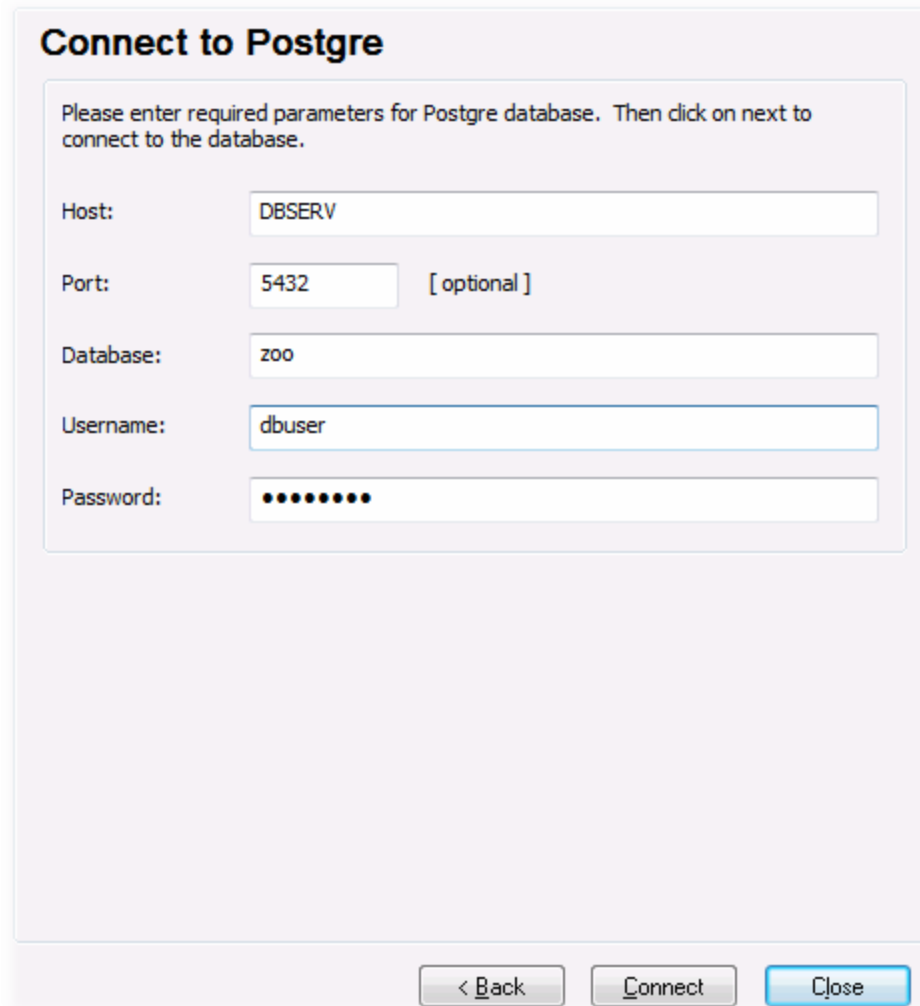
If you prefer to establish a connection by means of a non-native driver, see the following topics:

- Setting up a JDBC Connection [140]
- Connecting to PostgreSQL (ODBC) [190]

Otherwise, if you want to set up a native connection to PostgreSQL, follow the steps below. To proceed, you need the following prerequisites: host name, port, database name, username, and password.

To set up a native PostgreSQL connection:

1. Start the database connection wizard [117].
2. Click **PostgreSQL Connections**.
3. Enter the host (*localhost*, if PostgreSQL runs on the same machine), port (typically 5432, this is optional), the database name, username, and password in the corresponding boxes.

4.   Click **Connect**.

If the PostgreSQL database server is on a different machine, note the following:

- The PostgreSQL database server must be configured to accept connections from clients. Specifically, the **pg_hba.conf** file must be configured to allow non-local connections. Secondly, the **postgresql.conf** file must be configured to listen on specified IP address(es) and port. For more information, check the PostgreSQL documentation (https://www.postgresql.org/docs/9.5/static/client-authentication-problems.html).
- The server machine must be configured to accept connections on the designated port (typically, 5432) through the firewall. For example, on a database server running on Windows, a rule may need to be created to allow connections on port 5432 through the firewall, from **Control Panel > Windows Firewall > Advanced Settings > Inbound Rules**.

4.8   SQLite Connection

SQLite is a file-based, self-contained database type, which makes it ideal in scenarios where portability and ease of configuration is important. Since SQLite databases are natively supported by DatabaseSpy, you do not need to install any drivers to connect to them.

SQLite database support notes

- On Linux, statement execution timeout for SQLite databases is not supported.
- Full text search tables are not supported.
- SQLite allows values of different data types in each row of a given table. All processed values must be compatible with the declared column type; therefore, unexpected values can be retrieved and run-time errors may occur if your SQLite database has row values which are not the same as the declared column type.

Important
It is recommended to create tables with the STRICT keyword to ensure more predictable behavior of your data. Otherwise, the data may not be read or written correctly when values of different types are mixed in one column. To find out more about STRICT tables, see the SQLite documentation.

4.8.1   Connect to an Existing SQLite Database

You can connect to an existing SQLite database either using the connection wizard, or directly from Windows Explorer, by using the **Open With...** command.

Using the Connection Wizard

To connect to an existing SQLite database:

1. Run the database connection wizard (see Starting the Database Connection Wizard[117]).
2. Select **SQLite**, and then click **Next**.
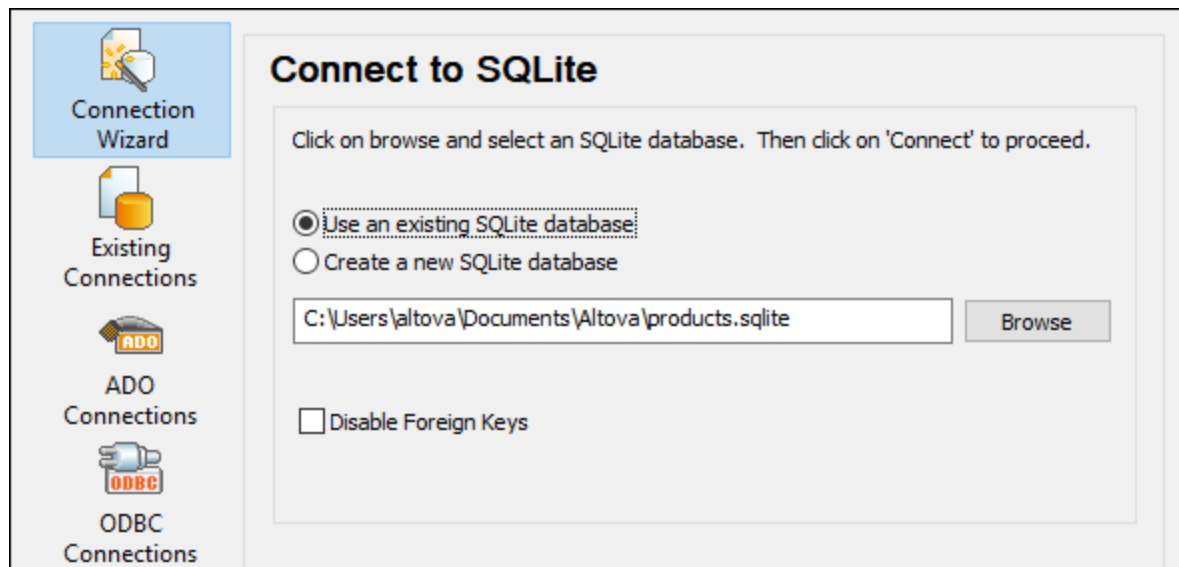
3. Select **Use an existing SQLite database**, and then browse for the SQLite database file, or enter the path (either relative or absolute) to the database. The **Connect** button becomes enabled once you enter the path to a SQLite database file.
4. Optionally, select the **Disable Foreign Keys** check box, see Foreign Key Constraints [149].
5. Click **Connect**.

From Windows Explorer

You can also open a SQLite database directly from Windows Explorer, as follows:

1. Right-click an existing database file in Windows Explorer and select **Open With** from the context menu.
2. Choose **DatabaseSpy** from the list of suggestions. If this suggestion is not available, select **Choose another app**, and browse for the DatabaseSpy executable from the installation directory.
3. If the Database Connection Wizard appears, click **Close** to disregard it.

You can also drag and drop the database file into DatabaseSpy if the latter is already open.

Notes:

- You can optionally configure DatabaseSpy from **Tools | Options** so as not to show the Database Connection Wizard whenever the application starts. To do this, clear the check box **Show create a database connection dialog** in the **General** group of settings.
- If DatabaseSpy is open and a data source to the same database file already exists, it will be reused. In other words, a duplicate data source to the same database will not be created. This is applicable both when opening a database file with **Open With** and when using drag and drop.

4.8.2   Create a New SQLite Database

You can create a new SQLite database file and connect to it, as an alternative to connecting to an existing database file. The database file created by DatabaseSpy is empty; use queries or scripts to create the required database structure and populate it with data.

To create a new SQLite database:

1. Run the database connection wizard (see Starting the Database Connection Wizard[117]).
2. Select **SQLite**, and then click **Next**.
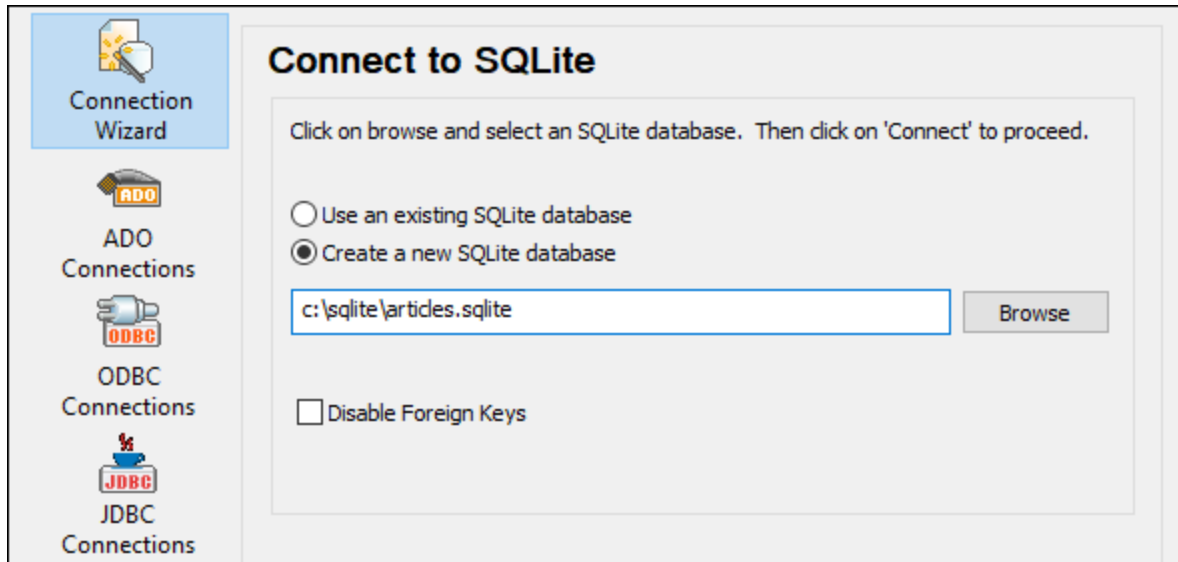
3. Select **Create a new SQLite database**, and then enter the path (either relative or absolute) of the database file to be created (for example, **c:\users\public\products.sqlite**). Alternatively, click **Browse** to select a folder, type the name of the database file in the "File name" text box (for example, **products.sqlite**), and click **Save**.

> Make sure that you have write permissions to the folder where you want to create the database file.

4. Optionally, select the **Disable Foreign Keys** check box, see Foreign Key Constraints[149].
5. Click **Connect**.

4.8.3 Foreign Key Constraints

When you connect to an existing SQLite database from DatabaseSpy, or when you create a new one, foreign key constraints are enabled by default. Foreign key constraints help preserve the integrity of data in your database. For example, when foreign keys are enabled, it is not possible to delete a record from a table if it has dependencies in another table.

In certain cases, you may want to temporarily override this behavior and disable foreign keys, perhaps, in order to update or insert multiple rows of data without getting data validation errors. To explicitly disable foreign keys before connecting to the SQLite database, select the **Disable Foreign Keys** option available on the database connection wizard.
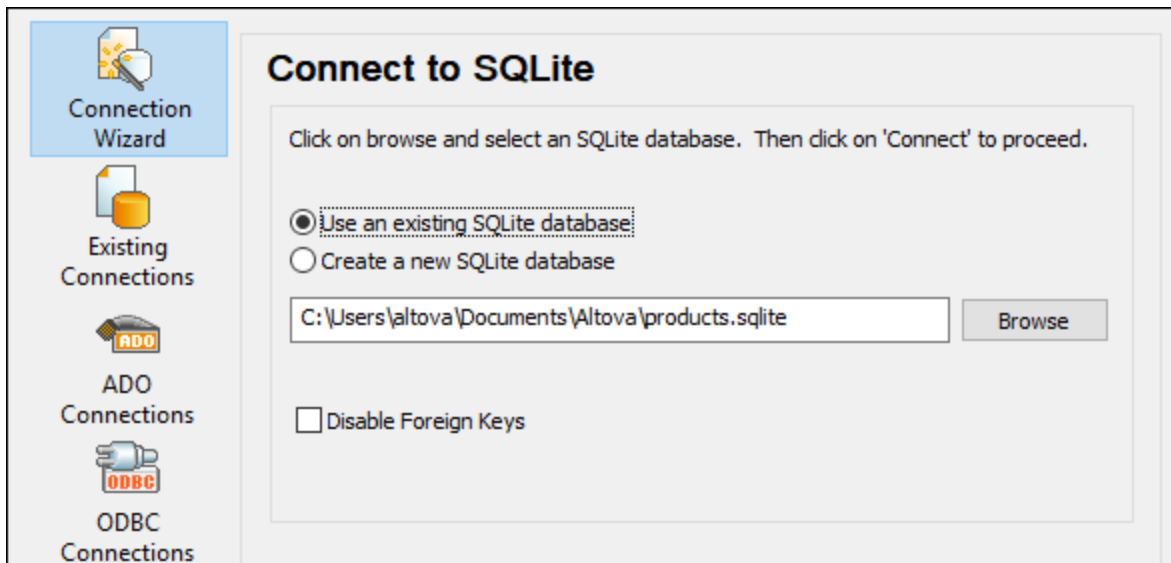
"Connect to SQLite" wizard page

When foreign keys are disabled, you will be able to perform operations against data that would otherwise not be possible due to validation checks. At the same time, however, there is also the risk of introducing incorrect data into the database, or creating "orphaned" rows. (An example of an "orphaned" row would be an address in the "addresses" table not linked to any person in the "person" table, because the person was deleted but its associated address was not.)

4.9   Global Resources

After you have created a database as a global resource, its connection details are stored and can be used across all Altova products installed on your machine.

Create a database as a global resource

To create a database as a global resource, do the following

1. On the **Tools** menu of DatabaseSpy, click **Global Resources**.
2. Click **Add**, and then click Database.
3. Type in a name for the global resource in the *Resource Alias* field.
4. Click **Choose Database**. The [Connection Wizard](117) appears.
5. Use the Connection Wizard to add a database connection as described above.

Use a global-resource database

To use a database that has been created as a global resource (*see above*), do the following:

1. Start the Connection Wizard as described above.
2. Select Global Resources. All the databases that have been created as global resources will be listed by their names in the Global Resources pane (*see screenshot below*).
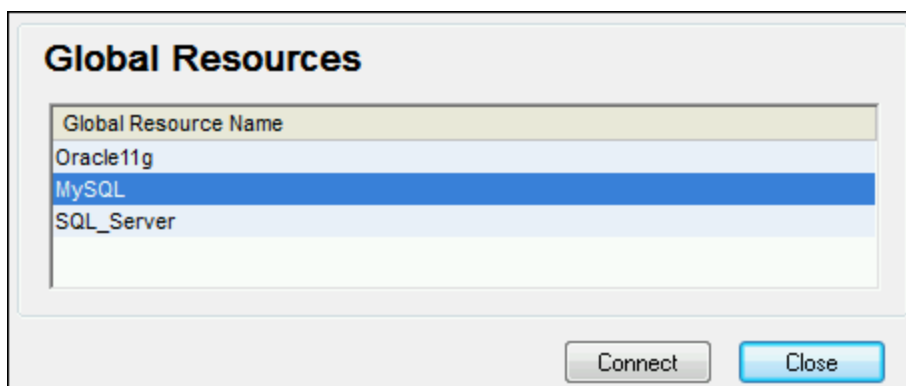
3. Select the global resource that you want. Tip: Move the mouse cursor over a global resource in the list to see information about the database.

4.10   Database Connection Examples

This section includes examples for connecting to a database from DatabaseSpy through ADO, ODBC, or JDBC. The ADO.NET connection examples are listed separately, see Sample ADO.NET Connection Strings [134]. For instructions about establishing a native connection to PostgreSQL and SQLite, see Setting up a PostgreSQL Connection [145] and Setting up a SQLite Connection [147], respectively.

Note the following:

- The instructions may differ if your Windows configuration, network environment and the database client or server software are not the same as the ones described in each example.
- For most database types, it is possible to connect using more than one data access technology (ADO, ADO.NET, ODBC, JDBC) or driver. The performance of the database connection, as well as its features and limitations will depend on the selected driver, database client software (if applicable), and any additional connectivity parameters that you may have configured outside DatabaseSpy.

4.10.1   Firebird (JDBC)

This example illustrates how to connect to a Firebird database server through JDBC.

Prerequisites:

- JRE (Java Runtime Environment) or Java Development Kit (JDK) must be installed. This may be either Oracle JDK or an open source build such as Oracle OpenJDK. DatabaseSpy will determine the path to the Java Virtual Machine (JVM) from the following locations, in this order: a) The custom JVM path you may have set in application **Options**, see Java Settings [423]; b) The JVM path found in the Windows registry; c) The JAVA_HOME environment variable.
- Make sure that the platform of DatabaseSpy (32-bit, 64-bit) matches that of the JRE/JDK.
- The Firebird JDBC driver must be available on your operating system (it takes the form of a .jar file which provides connectivity to the database). The driver can be downloaded from the Firebird website (https://www.firebirdsql.org/). This example uses *Jaybird 2.2.8*.
- You have the following database connection details: host, database path or alias, username, and password.

To connect to Firebird through JDBC:

1. Start the database connection wizard [117].
2. Click **JDBC Connections**.
3. Next to "Classpaths", enter the path to the .jar file which provides connectivity to the database. If necessary, you can also enter a semicolon-separated list of .jar file paths. In this example, the required .jar file is located at the following path: **C:\jdbc\firebird\jaybird-full-2.2.8.jar**. Note that you can leave the "Classpaths" text box empty if you have added the .jar file path(s) to the CLASSPATH environment variable of the operating system (see also Configuring the CLASSPATH [142]).
4. In the "Driver" box, select **org.firebirdsql.jdbc.FBDriver**. Note that this entry is available if a valid .jar file path is found either in the "Classpath" text box, or in the operating system's CLASSPATH environment variable (see the previous step).

Classpaths:	C:\jdbc\firebird\jaybird-full-2.2.8.jar
Driver:	org.firebirdsql.jdbc.FBDriver
Username:	prod_admin
Password:	•••••••
Database URL:	jdbc:firebirdsql://firebirdserv/COMPANY

5. Enter the username and password to the database in the corresponding text boxes.
6. Enter the connection string to the database server in the Database URL text box, by replacing the highlighted values with the ones applicable to your database server.

```
jdbc:firebirdsql://<host>[:<port>]/<database path or alias>
```

7. Click **Connect**.

4.10.2   Firebird (ODBC)

This example illustrates how to connect to a Firebird 2.5.4 database running on a Linux server.

Prerequisites:

- The Firebird database server is configured to accept TCP/IP connections from clients.
- The Firebird ODBC driver must be installed on your operating system. This example uses the Firebird ODBC driver version 2.0.3.154 downloaded from the Firebird website (https://www.firebirdsql.org/).
- The Firebird client must be installed on your operating system. Note that there is no standalone installer available for the Firebird 2.5.4 client; the client is part of the Firebird server installation package. You can download the Firebird server installation package from the Firebird website (https://www.firebirdsql.org/), look for "Windows executable installer for full Superclassic/Classic or Superserver". To install only the client files, choose "**Minimum client install - no server, no tools**" when going through the wizard steps.

> **Important:**
>
> - The platform of both the Firebird ODBC driver and client (32-bit or 64-bit) must correspond to that of DatabaseSpy.
> - The version of the Firebird client must correspond to the version of Firebird server to which you are connecting.

- You have the following database connection details: server host name or IP address, database path (or alias) on the server, user name, and password.

To connect to Firebird via ODBC:

1. Start the database connection wizard [117].
2. Click **ODBC Connections**.
3. Select **User DSN** (or **System DSN**, if you have administrative privileges), and then click **Add** .
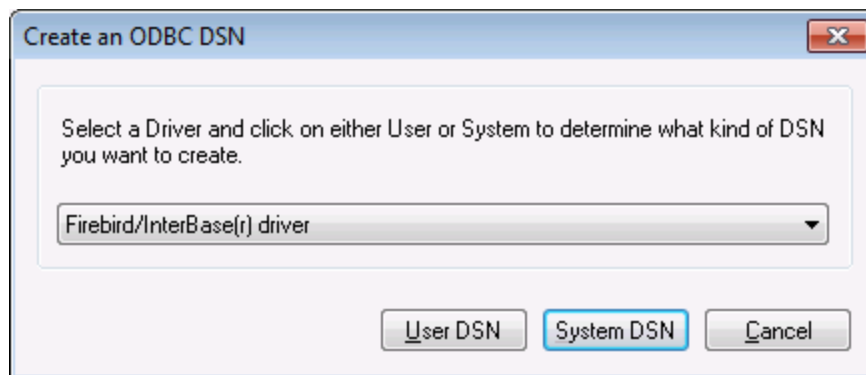
4. Select the Firebird driver, and then click **User DSN** (or **System DSN**, depending on what you selected in the previous step). If the Firebird driver is not available in the list, make sure that it is installed on your operating system (see also Viewing the Available ODBC Drivers [139]).
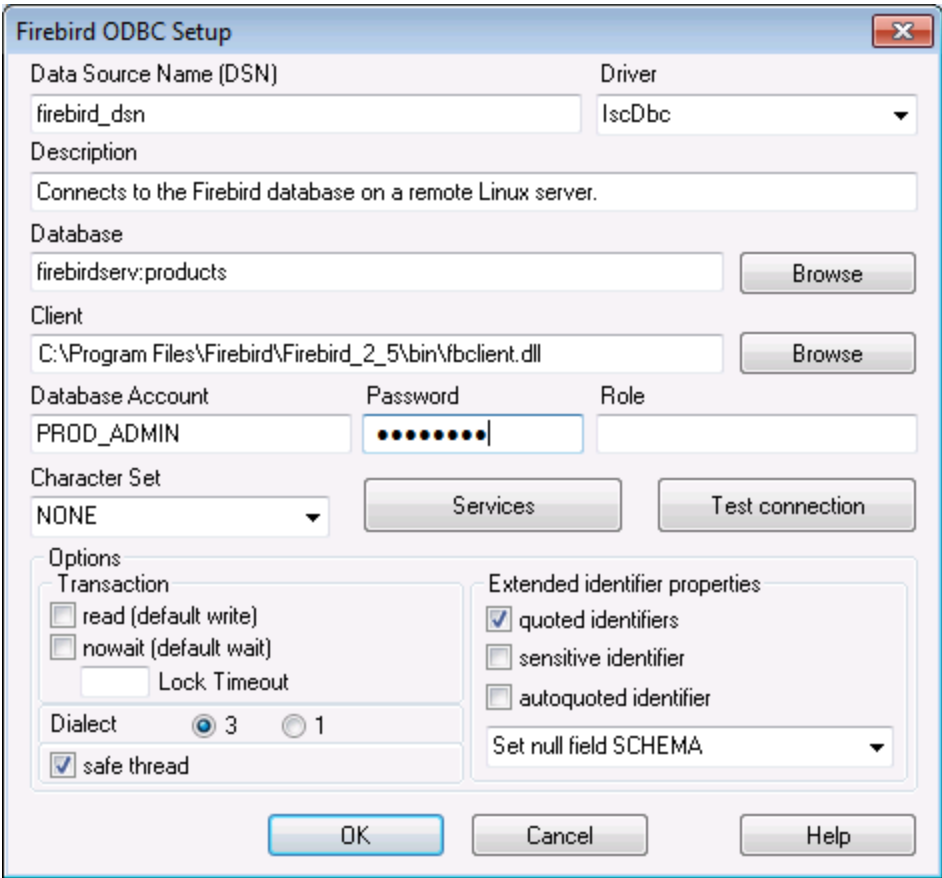
5. Enter the database connection details as follows:

Data Source Name (DSN)	Enter a descriptive name for the data source you are creating.

Database	Enter the server host name or IP address, followed by a colon, followed by the database alias (or path). In this example, the host name is `firebirdserv`, and the database alias is `products`, as follows:
	
	 `firebirdserv:products`
	
	 Using a database alias assumes that, on the server side, the database administrator has configured the alias *products* to point to the actual Firebird (.fdb) database file on the server (see the Firebird documentation for more details).
	
	 You can also use the server IP address instead of the host name, and a path instead of an alias; therefore, any of the following sample connection strings are valid:
	
	 `firebirdserver:/var/Firebird/databases/butterflies.fdb`
	 `127.0.0.1:D:\Misc\Lenders.fdb`

	If the database is on the local Windows machine, click **Browse** and select the Firebird (.fdb) database file directly.
Client	Enter the path to the **fbclient.dll** file. By default, this is the `bin` subdirectory of the Firebird installation directory.
Database Account	Enter the database user name supplied by the database administrator (in this example, `PROD_ADMIN`).
Password	Enter the database password supplied by the database administrator.

6. Click **OK**.

4.10.3   IBM DB2 (JDBC)

This example illustrates how to connect to an **IBM DB2** database server through JDBC.

Prerequisites:

- JRE (Java Runtime Environment) or Java Development Kit (JDK) must be installed. This may be either Oracle JDK or an open source build such as Oracle OpenJDK. DatabaseSpy will determine the path to the Java Virtual Machine (JVM) from the following locations, in this order: a) The custom JVM path you may have set in application **Options**, see Java Settings [423]; b) The JVM path found in the Windows registry; c) The `JAVA_HOME` environment variable.
- Make sure that the platform of DatabaseSpy (32-bit, 64-bit) matches that of the JRE/JDK. This example uses Oracle's OpenJDK 11.0 64-bit, and, consequently, the 64-bit version of DatabaseSpy.
- The JDBC driver (one or several .jar files that provide connectivity to the database) must be available on your operating system. This example uses the JDBC driver available after installing the **IBM Data Server Client** version 10.1 (64-bit). For the JDBC drivers to be installed, choose a **Typical** installation, or select this option explicitly on the installation wizard.

If you did not change the default installation path, the required .jar files will be in the **C:\Program Files\IBM\SQLLIB\java** directory after installation.

- You need the following database connection details: host, port, database name, username, and password.

To connect to IBM DB2 through JDBC:

1. Start the database connection wizard[117].
2. Click **JDBC Connections**.
3. Next to "Classpaths", enter the path to the .jar file which provides connectivity to the database. This examples refers to **C:\Program Files\IBM\SQLLIB\java\db2jcc.jar**. You may need to refer to the **db2jcc4.jar** driver, depending on the database server version. For driver compatibility, refer to IBM documentation (http://www-01.ibm.com/support/docview.wss?uid=swg21363866). Note that you can leave the "Classpaths" text box empty if you have added the .jar file path(s) to the CLASSPATH environment variable of the operating system (see also Configuring the CLASSPATH[142]).
4. In the "Driver" box, select **com.ibm.db2.jcc.DB2Driver**. This entry becomes available only if a valid .jar file path was found either in the "Classpaths" text box, or in the operating system's CLASSPATH environment variable (see the previous step).

Classpaths:	C:\Program Files\IBM\SQLLIB\java\db2jcc.jar
Driver:	com.ibm.db2.jcc.DB2Driver
Username:	username
Password:	●●●●●●●●●●
Database URL:	jdbc:db2://dbserver:50000/dbname

5. Enter the username and password of the database user in the corresponding text boxes.
6. Enter the JDBC connection string in the **Database URL** text box. Make sure to replace the connection details with the ones applicable to your database server.

```
jdbc:db2://hostName:port/databaseName
```

7. Click **Connect**.

4.10.4   IBM DB2 (ODBC)

This example illustrates how to connect to an IBM DB2 database through ODBC.

Prerequisites:

- IBM Data Server Client must be installed and configured on your operating system (this example uses IBM Data Server Client 9.7). For installation instructions, check the documentation supplied with your IBM DB2 software. After installing the IBM Data Server Client, check if the ODBC drivers are available on your machine (see [Viewing the Available ODBC Drivers](#) [139]).
- Create a database alias. There are several ways to do this:
  - ○ From IBM DB2 Configuration Assistant
  - ○ From IBM DB2 Command Line Processor
  - ○ From the ODBC data source wizard (for this case, the instructions are shown below)
- You have the following database connection details: host, database, port, username, and password.

To connect to IBM DB2:

1. [Start the database connection wizard](#) [117] and select **IBM DB2 (ODBC/JDBC)**.
2. Click **Next**.
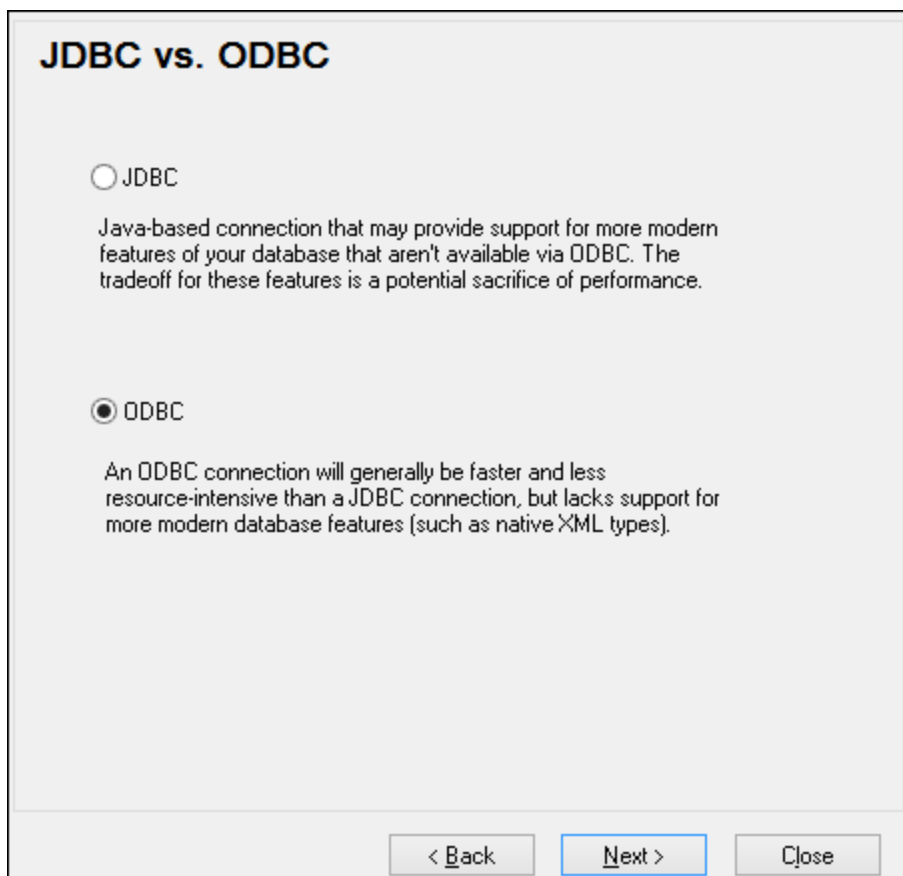
3.   Select **ODBC**, and click **Next**. If prompted to edit the list of known drivers for the database, select the database drivers applicable to IBM DB2 (see [Prerequisites](158)), and click **Next**.
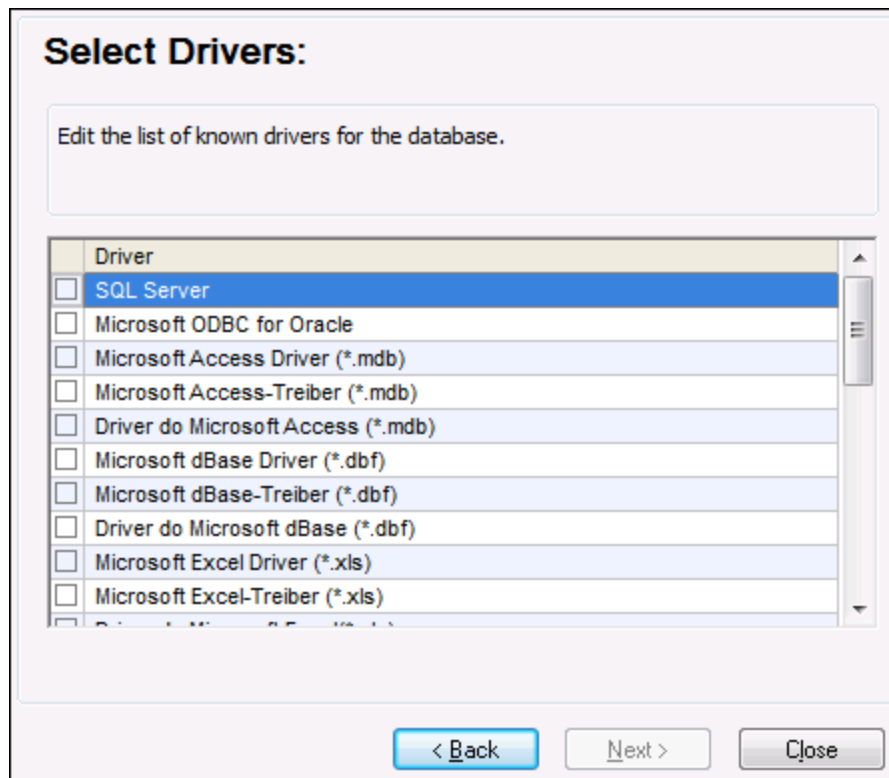
4. Select the IBM DB2 driver from the list, and then click **Connect**. (To edit the list of available drivers, click **Edit Drivers**, and then check or uncheck the IBM DB2 drivers you wish to add or remove, respectively.)
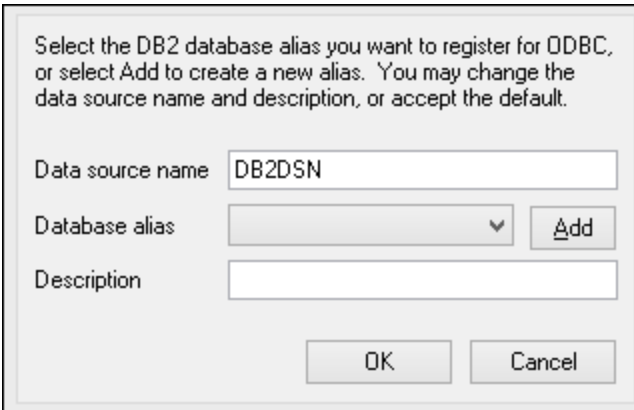
5. Enter a data source name (in this example, **DB2DSN**), and then click **Add**.

6. On the **Data Source** tab, enter the user name and password to the database.

7. On the **TCP/IP** tab, enter the database name, a name for the alias, the host name and the port number, and then click OK.

8.   Enter again the username and password, and then click **OK**.
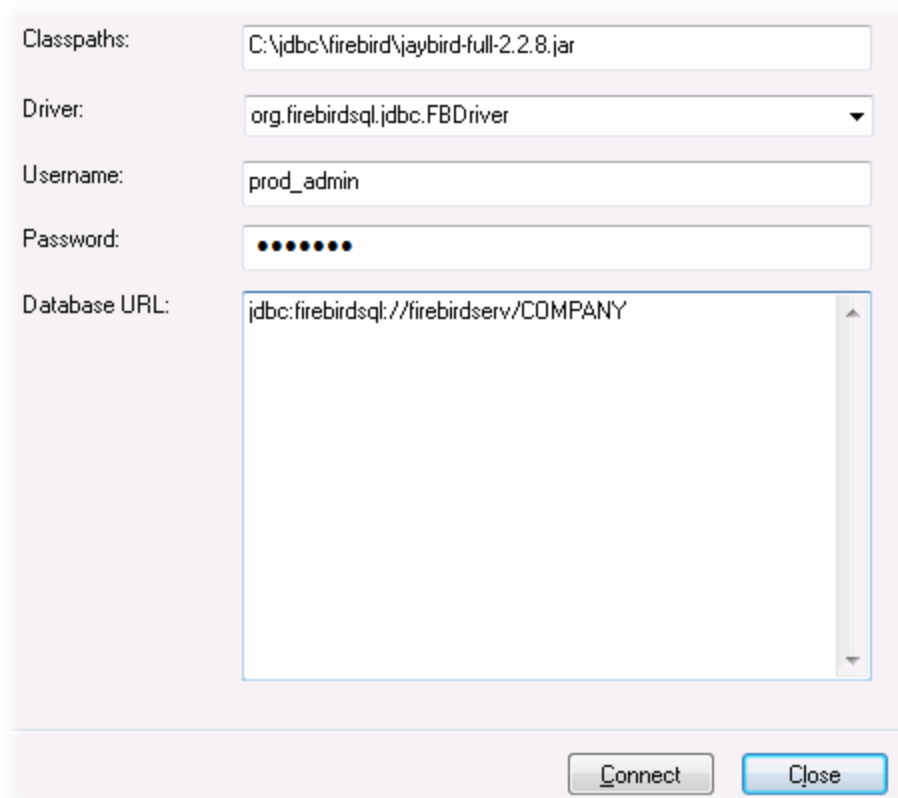
4.10.5 IBM DB2 for i (JDBC)

This example illustrates how to connect to an **IBM DB2 for i** database server through JDBC.

Prerequisites:

- JRE (Java Runtime Environment) or Java Development Kit (JDK) must be installed. This may be either Oracle JDK or an open source build such as Oracle OpenJDK. DatabaseSpy will determine the path to the Java Virtual Machine (JVM) from the following locations, in this order: a) The custom JVM path you may have set in application **Options**, see Java Settings⁽⁴²³⁾; b) The JVM path found in the Windows registry; c) The JAVA_HOME environment variable.
- Make sure that the platform of DatabaseSpy (32-bit, 64-bit) matches that of the JRE/JDK. This example uses Oracle's OpenJDK 11.0 64-bit, and, consequently, the 64-bit version of DatabaseSpy.
- The JDBC driver (one or several .jar files that provide connectivity to the database) must be available on your operating system. This example uses the open source **Toolbox for Java/JTOpen** version 9.8 (http://jt400.sourceforge.net/). After you download the package and unpack to a local directory, the required .jar files will be available in the **lib** subdirectory.
- You need the following database connection details: host, username, and password.

To connect to IBM DB2 for i through JDBC:

1. Start the database connection wizard⁽¹¹⁷⁾.
2. Click **JDBC Connections**.
3. Next to "Classpaths", enter the path to the .jar file which provides connectivity to the database. In this example, the required .jar file is at the following path: **C:\jdbc\jtopen_9_8\jt400.jar**. Note that you can leave the "Classpaths" text box empty if you have added the .jar file path(s) to the CLASSPATH environment variable of the operating system (see also Configuring the CLASSPATH⁽¹⁴²⁾).
4. In the "Driver" box, select **com.ibm.as400.access.AS400JDBCDriver**. This entry becomes available only if a valid .jar file path was found either in the "Classpaths" text box, or in the operating system's CLASSPATH environment variable (see the previous step).

Classpaths:	C:\jdbc\jtopen_9_8\lib\jt400.jar
Driver:	com.ibm.as400.access.AS400JDBCDriver
Username:	username
Password:	●●●●●●●●
Database URL:	jdbc:as400://192.0.2.0

5. Enter the username and password of the database user in the corresponding text boxes.
6. Enter the JDBC connection string in the **Database URL** text box. Make sure to replace **host** with the host name or IP address of your database server.

```
jdbc:as400://host
```

7. Click **Connect**.

4.10.6   IBM DB2 for i (ODBC)

This example illustrates how to connect to an *IBM DB2 for i* database through ODBC.

Prerequisites:

*   *IBM System i Access for Windows* must be installed on your operating system (this example uses *IBM System i Access for Windows V6R1M0*). For installation instructions, check the documentation supplied with your *IBM DB2 for i* software. After installation, check if the ODBC driver is available on your machine (see Viewing the Available ODBC Drivers [139]).
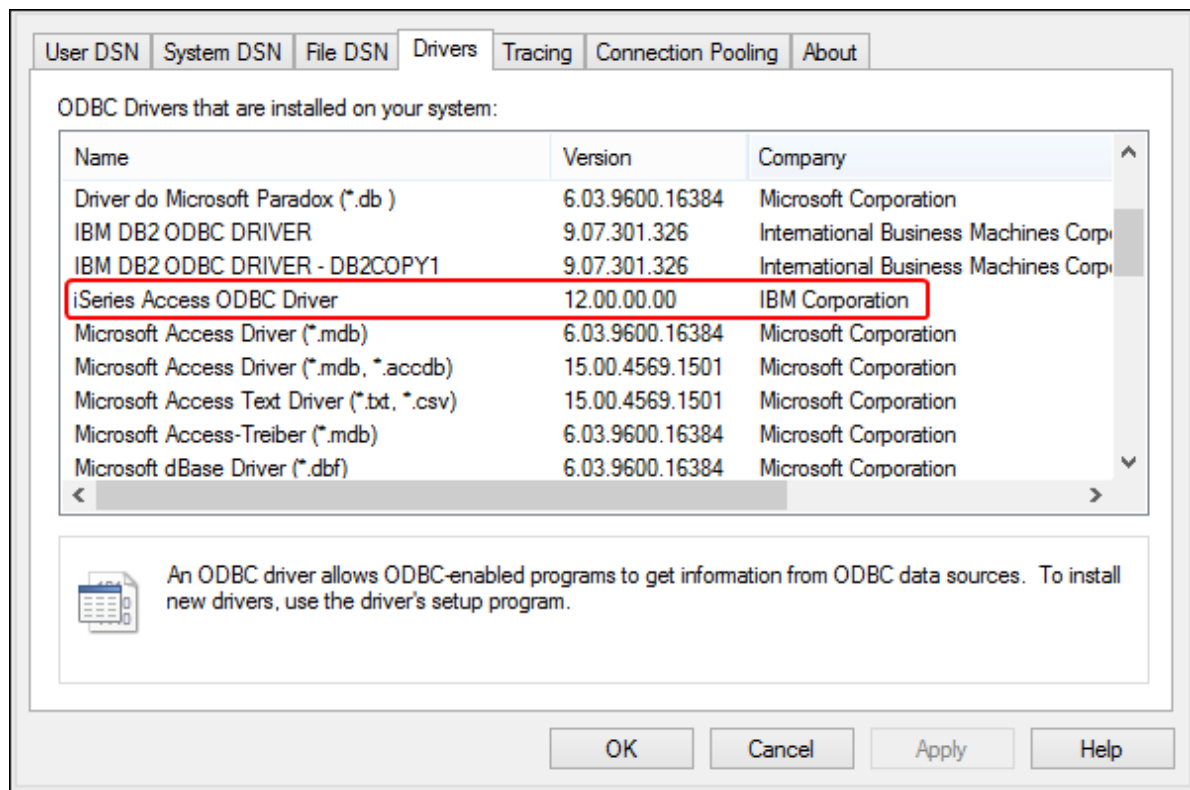
*   You have the following database connection details: the I.P. address of the database server, database user name, and password.
*   Run *System i Navigator* and follow the wizard to create a new connection. When prompted to specify a system, enter the I.P. address of the database server. After creating the connection, it is recommended to verify it (click on the connection, and select **File** > **Diagnostics** > **Verify Connection**). If you get connectivity errors, contact the database server administrator.

To connect to IBM DB2 for i:

1. Start the database connection wizard [117].
2. Click **ODBC connections**.

3. Click **User DSN** (alternatively, click **System DSN**, or **File DSN**, in which case the subsequent instructions will be similar).

4. Click **Add** .

5. Select the **iSeries Access ODBC Driver** from the list, and click **User DSN** (or **System DSN**, if applicable).

> Select a Driver and click on either User or System to determine what kind of DSN you want to create.
>
> iSeries Access ODBC Driver   ▼
>
>   User DSN   System DSN   Cancel
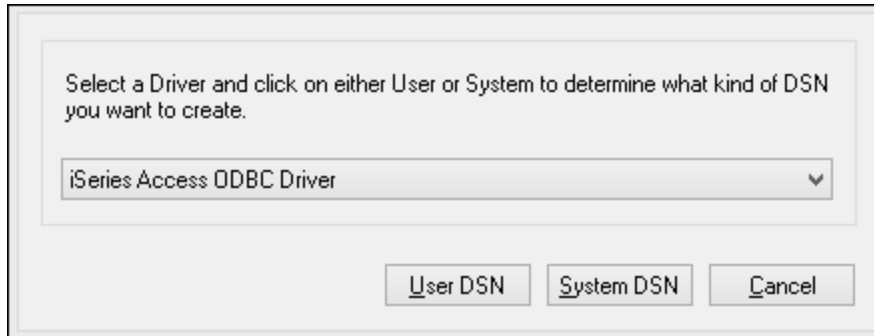
6. Enter a data source name and select the connection from the System combo box. In this example, the data source name is **iSeriesDSN** and the System is **192.0.2.0**.

> General | Server | Data Types | Packages | Performance | Language | Catalog | Conversions | Diagnostic
>
> Data source name:
>
> iSeriesDSN
>
> Description:
>
> System i Access for Windows ODBC data source
>
> System:
>
> 192.0.2.0   ▼   Connection Options...
>
>   OK   Cancel   Apply   Help
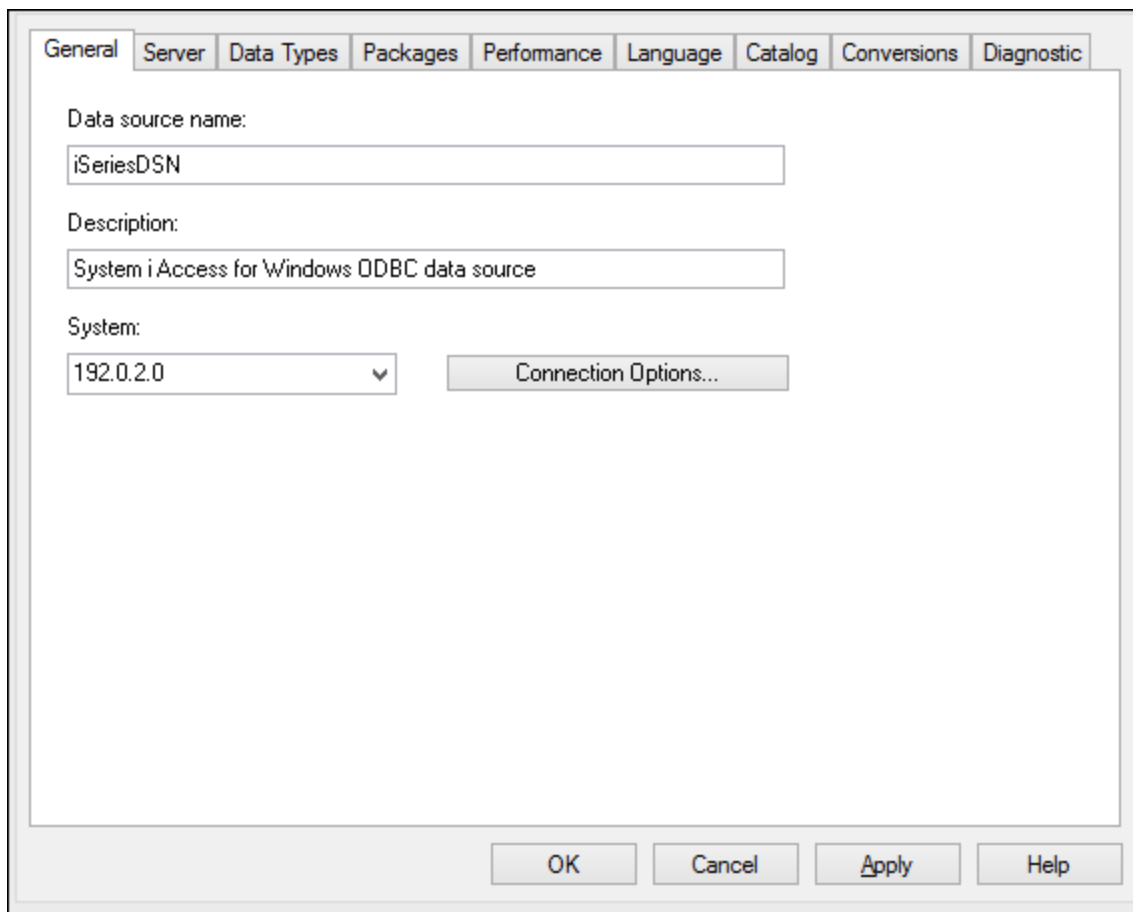
7. Click Connection Options, select **Use the User ID specified below** and enter the name of the database user (in this example, **DBUSER**).

8. Click **OK**. The new data source becomes available in the list of DSNs.
9. Click **Connect**.
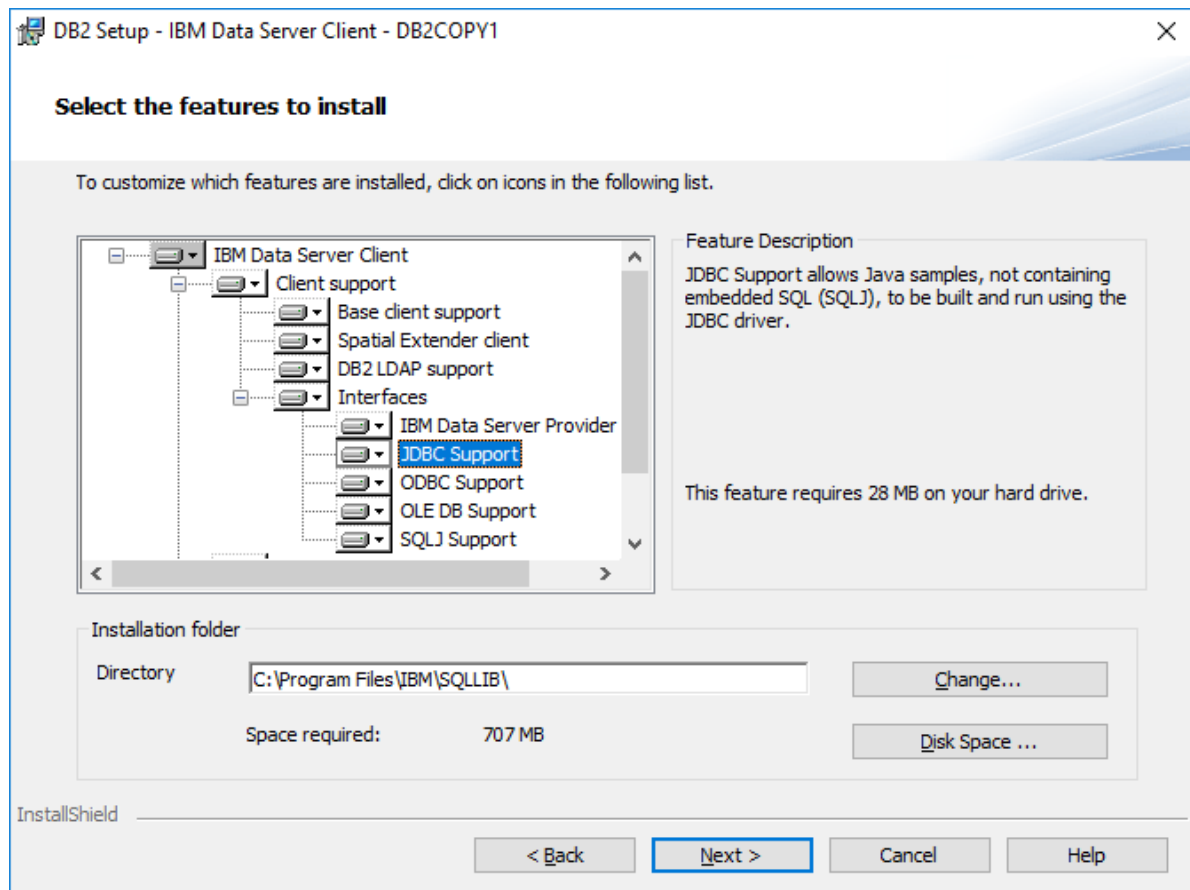10. Enter the user name and password to the database when prompted, and then click **OK**.

4.10.7   IBM Informix (JDBC)

This example illustrates how to connect to an IBM Informix database server through JDBC.

Prerequisites:

*   JRE (Java Runtime Environment) or Java Development Kit (JDK) must be installed. This may be either Oracle JDK or an open source build such as Oracle OpenJDK. DatabaseSpy will determine the path to the Java Virtual Machine (JVM) from the following locations, in this order: a) The custom JVM path you may have set in application **Options**, see Java Settings [423]; b) The JVM path found in the Windows registry; c) The JAVA_HOME environment variable.
*   Make sure that the platform of DatabaseSpy (32-bit, 64-bit) matches that of the JRE/JDK.
*   The JDBC driver (one or several .jar files that provide connectivity to the database) must be available on your operating system. In this example, IBM Informix JDBC driver version 3.70 is used. For the driver's installation instructions, see the documentation accompanying the driver or the "IBM Informix JDBC Driver Programmer's Guide").
*   You have the following database connection details: host, name of the Informix server, database, port, username, and password.

To connect to IBM Informix through JDBC:

1. Start the database connection wizard [117].
2. Click **JDBC Connections**.
3. Next to "Classpaths", enter the path to the .jar file which provides connectivity to the database. If necessary, you can also enter a semicolon-separated list of .jar file paths. In this example, the required .jar file is located at the following path: **C:\Informix_JDBC_Driver\lib\ifxjdbc.jar**. Note that you can leave the "Classpaths" text box empty if you have added the .jar file path(s) to the CLASSPATH environment variable of the operating system (see also Configuring the CLASSPATH [142]).
4. In the "Driver" box, select **com.informix.jdbc.IfxDriver**. Note that this entry is available if a valid .jar file path is found either in the "Classpaths" text box, or in the operating system's CLASSPATH environment variable (see the previous step).

Classpaths:	C:\jdbc\Informix_JDBC_Driver\lib\ifxjdbc.jar;
Driver:	com.informix.jdbc.IfxDriver
Username:	dbuser
Password:	•••••••
Database URL:	jdbc:informix-sqli://host:port/MyDatabase:INFORMIXSERVER=MyServerName

Connect   Close

5. Enter the username and password to the database in the corresponding text boxes.
6. Enter the connection string to the database server in the Database URL text box, by replacing the highlighted values with the ones applicable to your database server.

```
jdbc:informix-sqli://hostName:port/databaseName:INFORMIXSERVER=myserver;
```

7. Click **Connect**.

4.10.8   MariaDB (ODBC)

This example illustrates how to connect to a MariaDB database server through ODBC.

Prerequisites:

- The MariaDB Connector/ODBC (https://downloads.mariadb.org/connector-odbc/) must be installed.
- You have the following database connection details: host, database, port, username, and password.

To connect to MariaDB through ODBC:

1. Start the database connection wizard [117].
2. Select **MariaDB (ODBC)**, and then click **Next**.
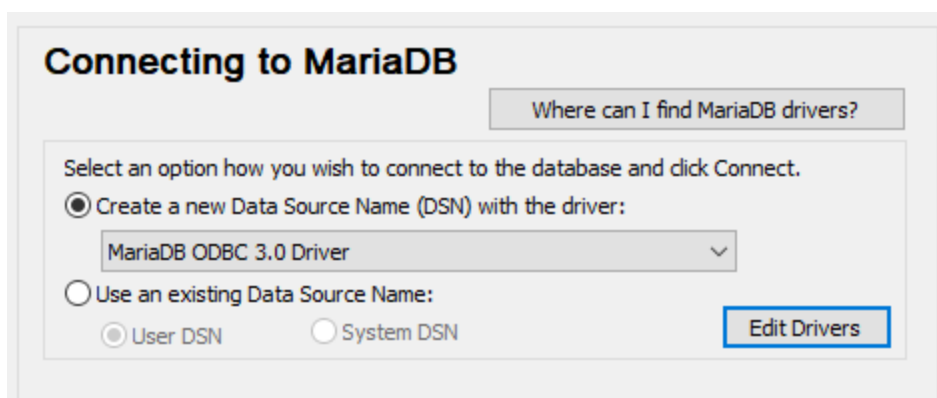
3. Select **Create a new Data Source Name (DSN) with the driver**, and choose **MariaDB ODBC 3.0 Driver**. If no such driver is available in the list, click **Edit Drivers**, and select any available MariaDB drivers (the list contains all ODBC drivers installed on your operating system).
4. Click **Connect**.

5. Enter name and, optionally, a description that will help you identify this ODBC data source in future.

6. Fill in the database connection credentials (TCP/IP Server, User, Password), select a database, and then click **Test DSN**. Upon successful connection, a message box appears:
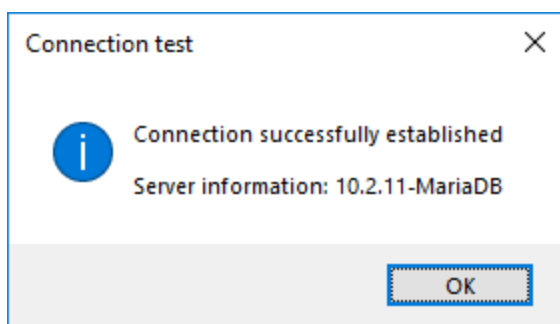
7. Click **Next** and complete the wizard. Other parameters may be required, depending on the case (for example, SSL certificates if you are connecting to MariaDB through a secure connection).

Note: If the database server is remote, it must be configured by the server administrator to accept remote connections from your machine's IP address.

4.10.9   Microsoft Access (ADO)

A simple way to connect to a Microsoft Access database is to follow the wizard and browse for the database file, as shown in Connecting to an Existing Microsoft Access Database [125]. An alternative approach is to set up an ADO connection explicitly, as shown in this topic. This approach is useful if your database is password-protected.

It is also possible to connect to Microsoft Access through an ODBC connection, but it has limitations, so it is best to avoid it.

To connect to a password-protected Microsoft Access database:

1. Start the database connection wizard [117].
2. Click **ADO Connections**.
3. Click **Build**.
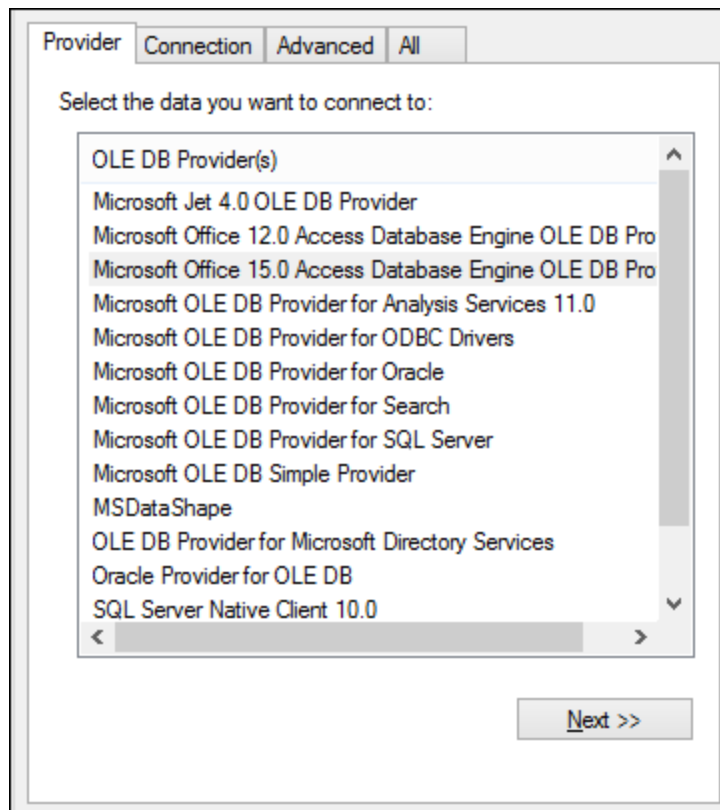
4. Select the **Microsoft Office 15.0 Access Database Engine OLE DB Provider**, and then click **Next**.

5. In the Data Source box, enter the path to the Microsoft Access file in UNC format, for example, **\ \myserver\\mynetworkshare\Reports\Revenue.accdb**, where **myserver** is the name of the server and **mynetworkshare** is the name of the network share.

6. On the **All** tab, double click the **Jet OLEDB:Database Password** property and enter the database password as property value.

Note: If you are still unable to connect, locate the workgroup information file (**System.MDW**) applicable to your user profile, and set the value of the **Jet OLEDB: System database** property to the path of the **System.MDW** file.

4.10.10   Microsoft SQL Server (ADO)

This example illustrates how to connect to a SQL Server database through ADO. These instructions are applicable when you use the recommended **Microsoft OLE DB Driver for SQL Server (MSOLEDBSQL)**,

---

which is available for download at https://docs.microsoft.com/en-us/sql/connect/oledb/download-oledb-driver-for-sql-server?view=sql-server-ver15.

Before following these instructions, make sure that you have downloaded and installed the provider above on your workstation. The ADO provider must match the platform of DatabaseSpy (32-bit or 64-bit).

If you would like to use other ADO providers such as **SQL Server Native Client (SQLNCLI)** or **Microsoft OLE DB Provider for SQL Server (SQLOLEDB)**, the instructions are similar, but these providers are deprecated and thus not recommended. Also, for the connection to be successful with a deprecated provider, you may need to set additional connection properties as described in Setting up the SQL Server Data Link Properties[126].

The **Microsoft OLE DB Provider for SQL Server (SQLOLEDB)** is known to have issues with parameter binding of complex queries like Common Table Expressions (CTE) and nested SELECT statements.

To connect to SQL Server:

1. Start the database connection wizard[117].
2. Select **Microsoft SQL Server (ADO)**, and then click **Next**. The list of available ADO providers is displayed. In this example, the **Microsoft OLE DB Driver for SQL Server** is used. If it's not in the list, make sure that it is installed on your computer, as mentioned above.

3. Click **Next**. The Data Link Properties dialog box appears.

4. Select or enter the name of the database server, for example, **SQLSERV01**. If you are connecting to a named SQL Server instance, the server name looks like **SQLSERV01\SOMEINSTANCE**.
5. If the database server was configured to allow connections from users authenticated on the Windows domain, select **Windows Authentication**. Otherwise, select **SQL Server Authentication**, clear the **Blank password** check box, and enter the database credentials in the relevant boxes.
6. Select the **Allow saving password** check box and the database to which you are connecting (in this example, "Nanonull").
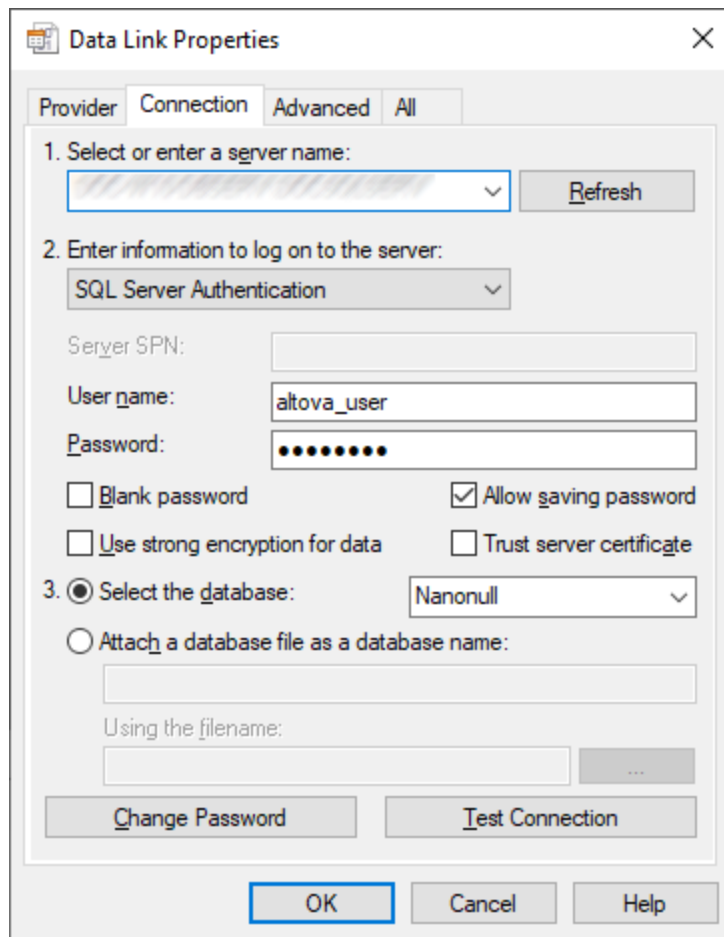
7.   To test the connection at this time, click **Test Connection**. This is an optional, recommended step.
8.   Click **OK**.

4.10.11   Microsoft SQL Server (ODBC)

This example illustrates how to connect to a SQL Server database through ODBC.

Prerequisites:

· Download and install the **Microsoft ODBC Driver for SQL Server** from the Microsoft website, see
  https://docs.microsoft.com/en-us/SQL/connect/odbc/download-odbc-driver-for-sql-server. This example
  uses **Microsoft ODBC Driver 17 for SQL Server** to connect to a **SQL Server 2016** database. You
  might want to download a different ODBC driver version, depending on the version of SQL Server where
  you want to connect. For information about ODBC driver versions supported by your SQL Server
  database, refer to the driver's system requirements.

To connect to SQL Server using ODBC:
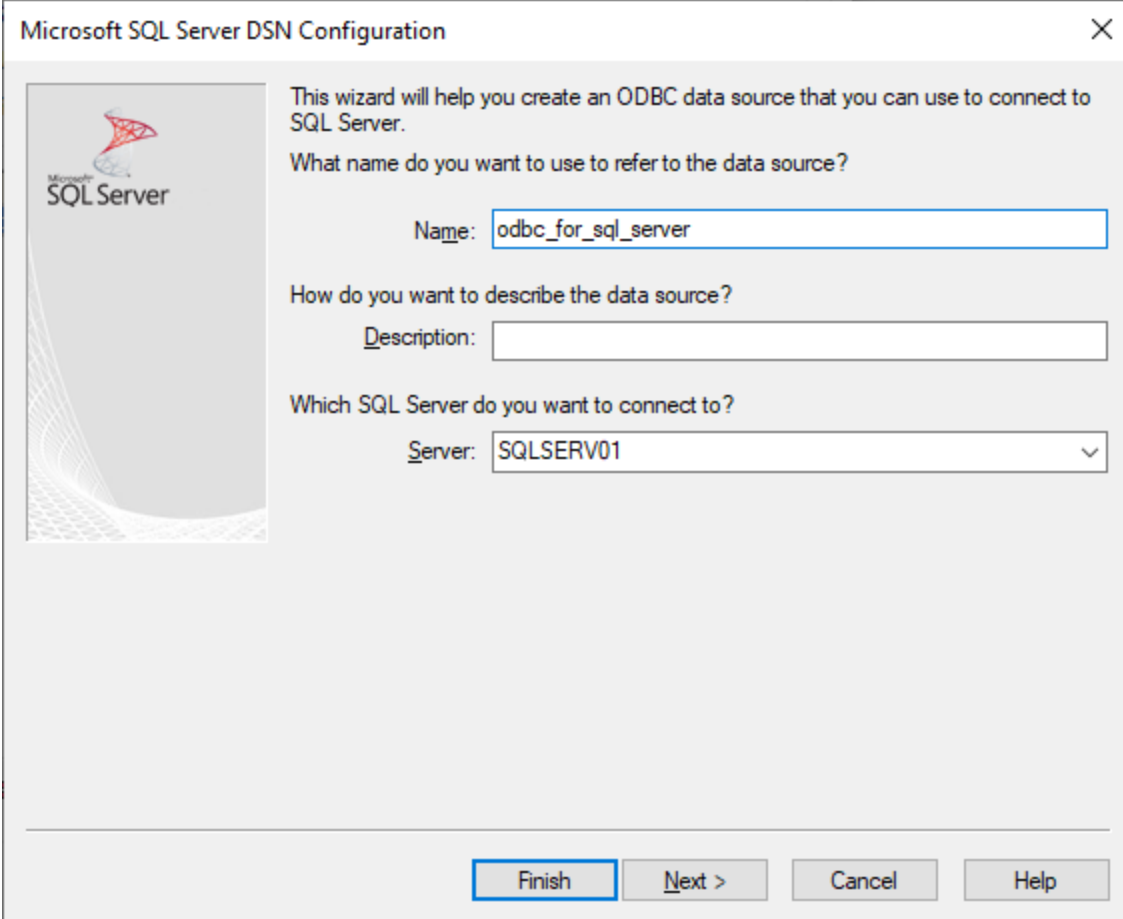
1. Start the database connection wizard [117].
2. Click **ODBC Connections**.
3. Select **User DSN** (or **System DSN**, if you have administrative privileges), and then click **Add** ➕ .
4. Select the driver from the list. Note that the driver appears in the list only after it has been installed.

```
Create an ODBC DSN                                                    ✕

   Select a Driver and click on either User or System to determine what kind of DSN
   you want to create.

   ODBC Driver 17 for SQL Server                                        ⌄

                                 User DSN      System DSN      Cancel
```
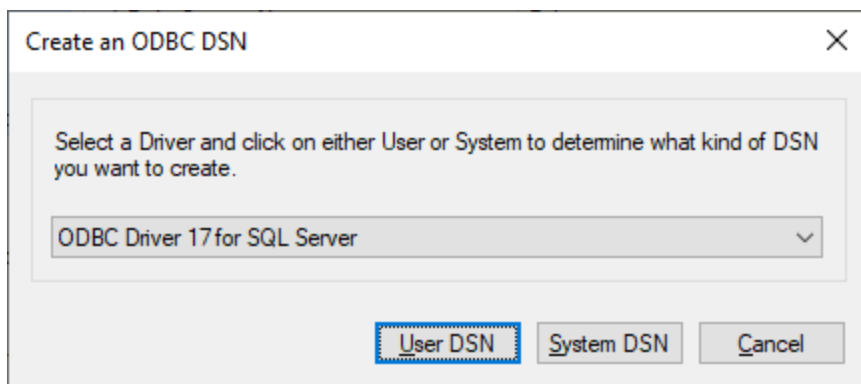
5. Click **User DSN** (or **System DSN** if you are creating a System DSN).

> Creating a **System DSN** requires that DatabaseSpy be run as an administrator. Therefore, in order
> to create a **System DSN**, cancel the wizard, make sure that you run DatabaseSpy as an
> administrator, and perform the steps above again.

6. Enter a name and, optionally, a description to identify this connection, and then select from the list the
   SQL Server to which you are connecting (**SQLSERV01** in this example).
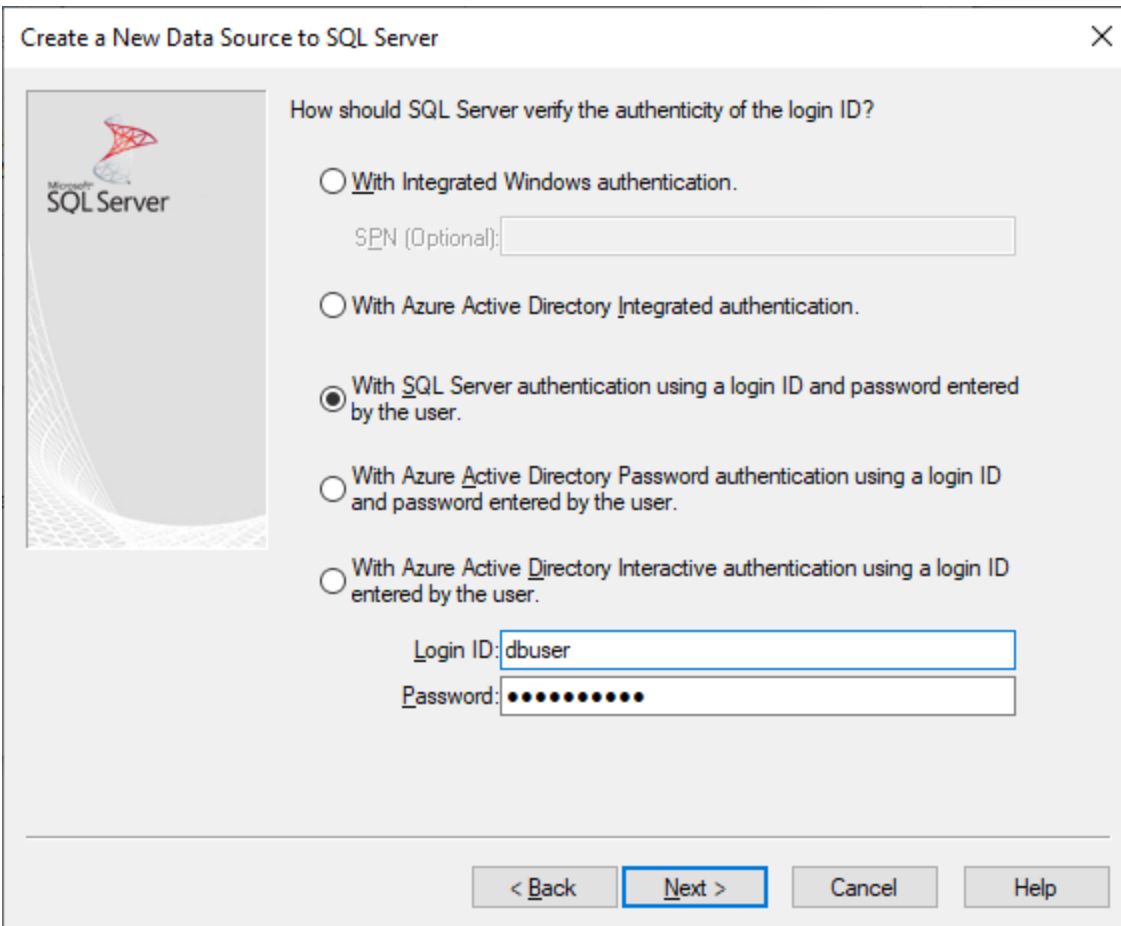
7. If the database server was configured to allow connections from users authenticated on the Windows domain, select **With Integrated Windows authentication**. Otherwise, select one of the other options, as applicable. This example uses **With SQL Server authentication...** , which requires that the user name and password be entered in the relevant boxes.
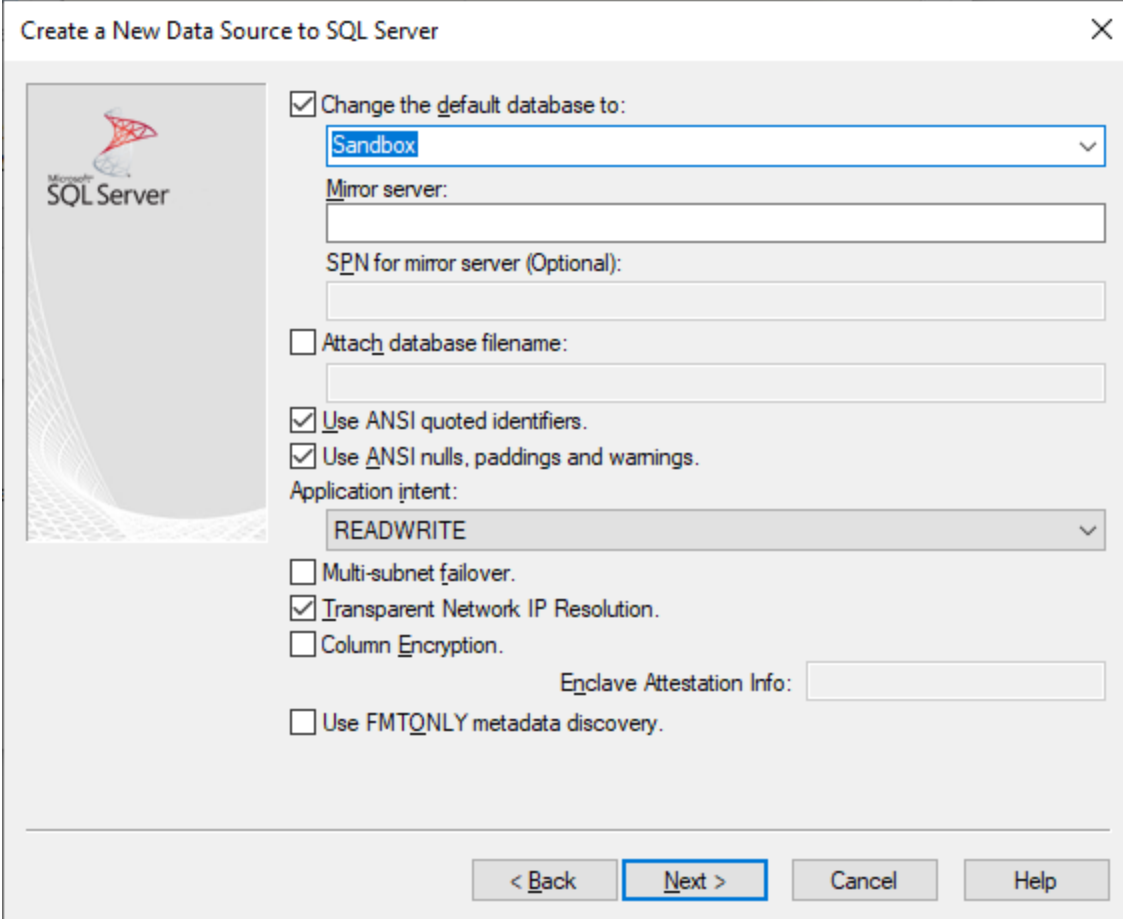
8. Optionally, select the **Change the default database to** check box and enter the name of the
    database to which you are connecting (in this example, **Sandbox**).

9. Click **Next** and, optionally, configure additional parameters for this connection.
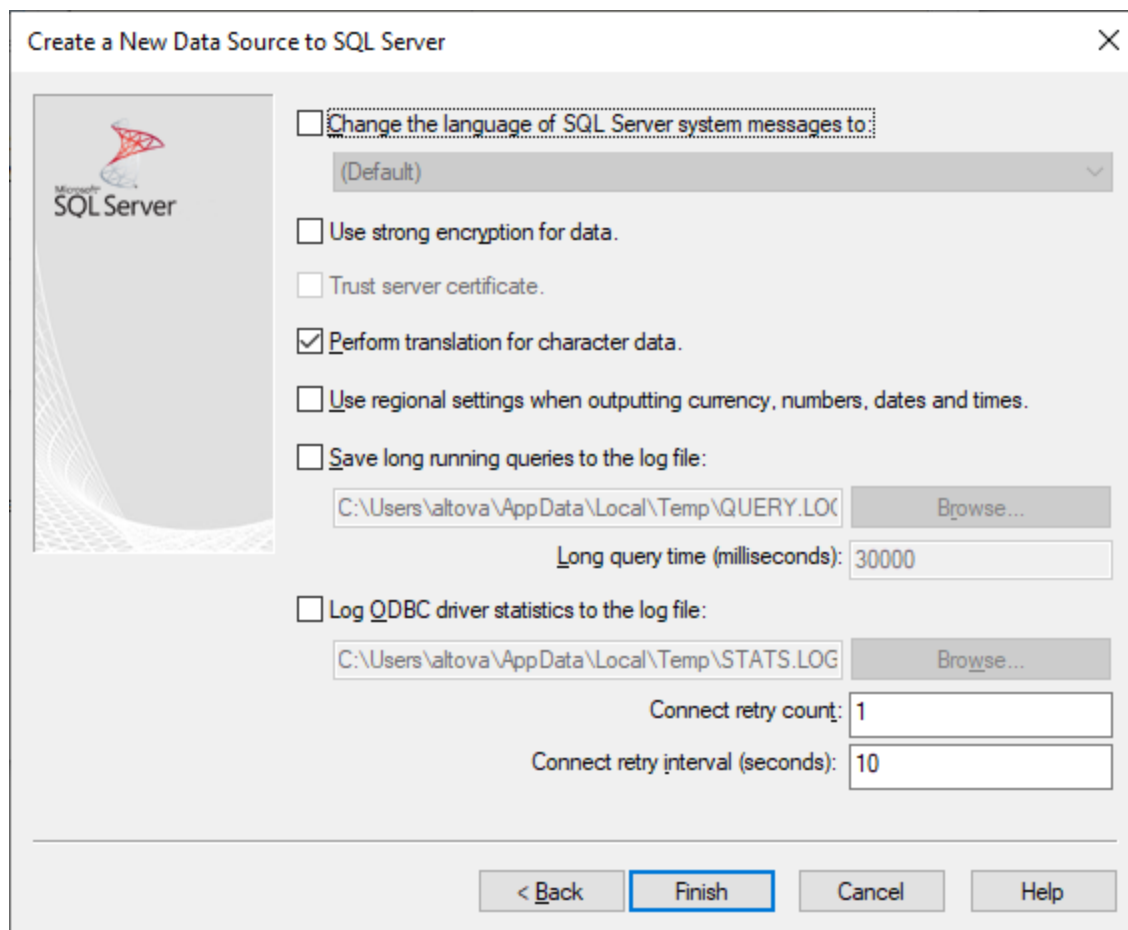
10. Click **Finish**. A confirmation dialog box listing the connection details opens.
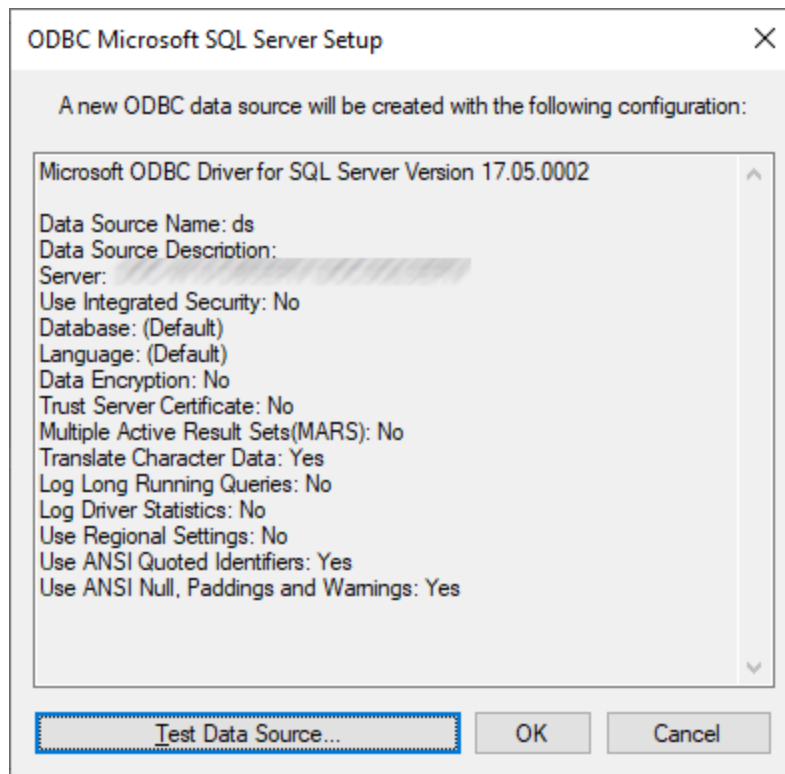
11. Click **OK**. The data source now appears in the list of **User** or **System** data sources, as configured, for example:
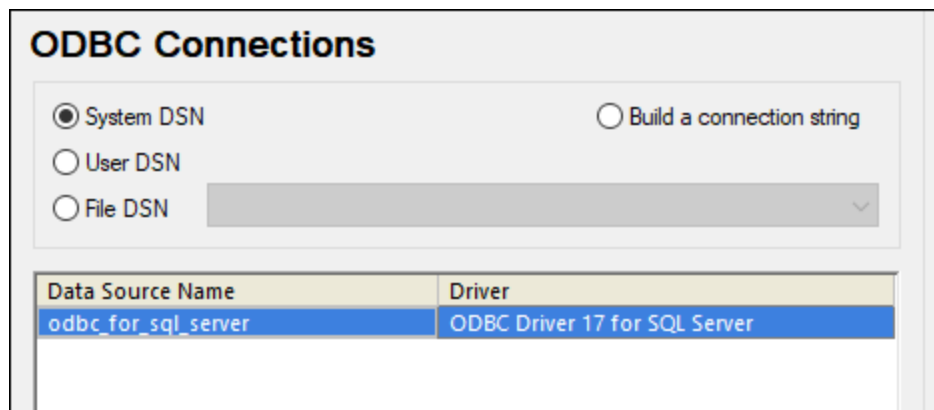
4.10.12   MySQL (ODBC)

This example illustrates how to connect to a MySQL database server from a Windows machine through the ODBC driver. The MySQL ODBC driver is not available on Windows, so it must be downloaded and installed separately. This example uses MySQL Connector/ODBC 8.0.

Prerequisites:

- MySQL ODBC driver must be installed on your operating system. Check the MySQL documentation for the driver version recommended for your database server version (see https://dev.mysql.com/downloads/connector/odbc/).
- You have the following database connection details: host, database, port, username, and password.

> If you installed MySQL Connector/ODBC for 64-bit platform, make sure to install DatabaseSpy for 64-bit platform as well.

To connect to MySQL via ODBC:

1. Start the database connection wizard [117].
2. Select **MySQL (ODBC)**, and then click **Next**.
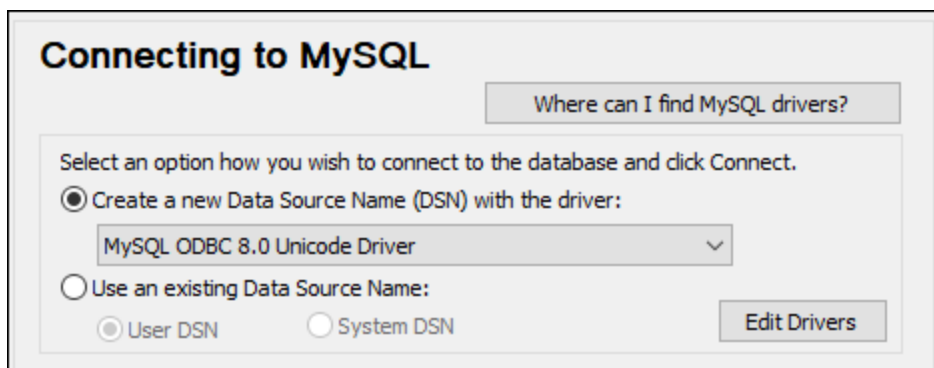
3. Select **Create a new Data Source Name (DSN) with the driver**, and select a MySQL driver. If no MySQL driver is available in the list, click **Edit Drivers**, and select any available MySQL drivers (the list contains all ODBC drivers installed on your operating system).

> If you installed DatabaseSpy 64-bit, then the 64-bit ODBC drivers are shown in the list. Otherwise, the 32-bit ODBC drivers are shown. See also Viewing the Available ODBC Drivers [139].

4. Click **Connect**.
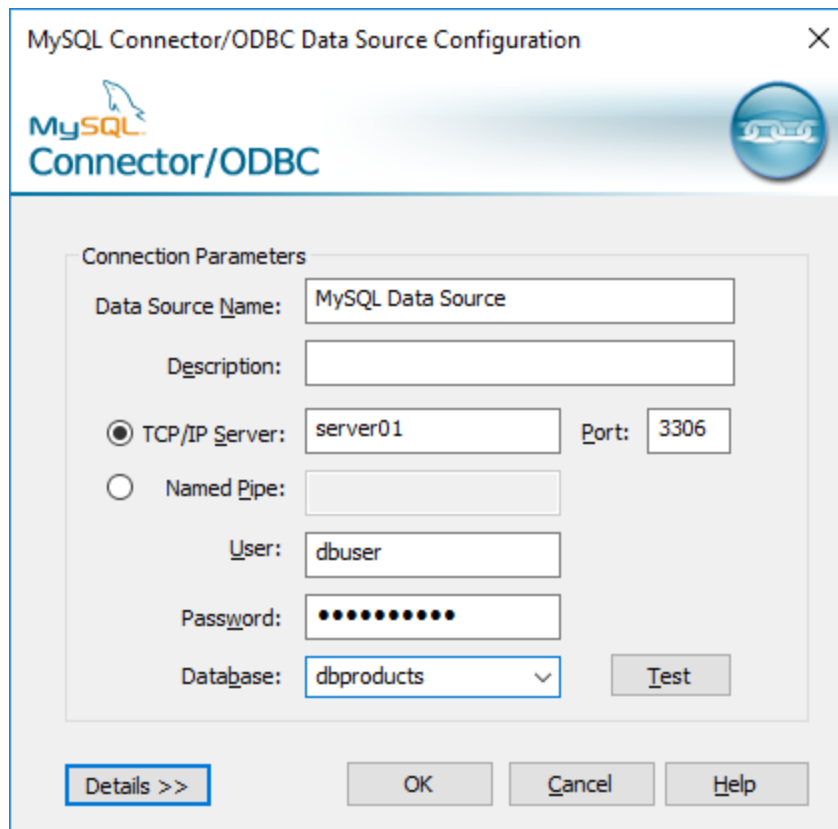
5. In the Data Source Name box, enter a descriptive name that will help you identify this ODBC data source in future.
6. Fill in the database connection credentials (TCP/IP Server, User, Password), select a database, and then click **OK**.

Note: If the database server is remote, it must be configured by the server administrator to accept remote connections from your machine's IP address. Also, if you click **Details>>**, there are several additional parameters available for configuration. Check the driver's documentation before changing their default values.

4.10.13   Oracle (JDBC)

This example shows you how to connect to an Oracle database server from a client machine, using the JDBC interface. The connection is created as a pure Java connection, using the **Oracle Instant Client Package (Basic)** available from the Oracle website. The advantage of this connection type is that it requires only the Java environment and the .jar libraries supplied by the Oracle Instant Client Package, saving you the effort to install and configure a more complex database client.

Prerequisites:

- JRE (Java Runtime Environment) or Java Development Kit (JDK) must be installed. This may be either Oracle JDK or an open source build such as Oracle OpenJDK. DatabaseSpy will determine the path to the Java Virtual Machine (JVM) from the following locations, in this order: a) The custom JVM path you

may have set in application **Options**, see Java Settings [423]; b) The JVM path found in the Windows registry; c) The JAVA_HOME environment variable.
· Make sure that the platform of DatabaseSpy (32-bit, 64-bit) matches that of the JRE/JDK.
· The **Oracle Instant Client Package (Basic)** must be available on your operating system. The package can be downloaded from the official Oracle website. This example uses Oracle Instant Client Package version 12.1.0.2.0, for Windows 32-bit and, consequently, Oracle JDK 32-bit.
· You have the following database connection details: host, port, service name, username, and password.

To connect to Oracle through the Instant Client Package:

1. Start the database connection wizard [117].
2. Click **JDBC Connections**.
3. Next to "Classpaths", enter the path to the .jar file which provides connectivity to the database. If necessary, you can also enter a semicolon-separated list of .jar file paths. In this example, the required .jar file is located at the following path: **C:\jdbc\instantclient_12_1\ojdbc7.jar**. Note that you can leave the "Classpaths" text box empty if you have added the .jar file path(s) to the CLASSPATH environment variable of the operating system (see also Configuring the CLASSPATH [142]).
4. In the "Driver" box, select either **oracle.jdbc.OracleDriver** or **oracle.jdbc.driver.OracleDriver**. Note that these entries are available if a valid .jar file path is found either in the "Classpaths" text box, or in the operating system's CLASSPATH environment variable (see the previous step).
5. Enter the username and password to the database in the corresponding text boxes.

Classpaths:	C:\jdbc\instantclient_12_1\ojdbc7.jar
Driver:	oracle.jdbc.driver.OracleDriver
Username:	johndoe
Password:	•••••••
Database URL:	jdbc:oracle:thin:@//ora12c:1521/orcl12c

                                                          [Connect]   [Close]

6. Enter the connection string to the database server in the Database URL text box, by replacing the highlighted values with the ones applicable to your database server.

```
jdbc:oracle:thin:@//host:port:service
```

7. Click **Connect**.

4.10.14   Oracle (ODBC)

This example illustrates a common scenario where you connect from DatabaseSpy to an Oracle database server on a network machine, through an Oracle database client installed on the local operating system.

The example includes instructions for setting up an ODBC data source (DSN) using the database connection wizard in DatabaseSpy. If you have already created a DSN, or if you prefer to create it directly from the **ODBC Data Source administrator** in Windows, you can do so, and then select it when prompted by the wizard. For more information about ODBC data sources, see [Setting up an ODBC Connection](#)[137].

Prerequisites:

- The Oracle database client (which includes the ODBC Oracle driver) must be installed and configured on your operating system. For instructions on how to install and configure an Oracle database client, refer to the documentation supplied with your Oracle software.
- The **tnsnames.ora** file located in Oracle home directory contains an entry that describes the database connection parameters, in a format similar to this:

```
ORCL =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = server01)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SID = orcl)
      (SERVER = DEDICATED)
    )
  )
```
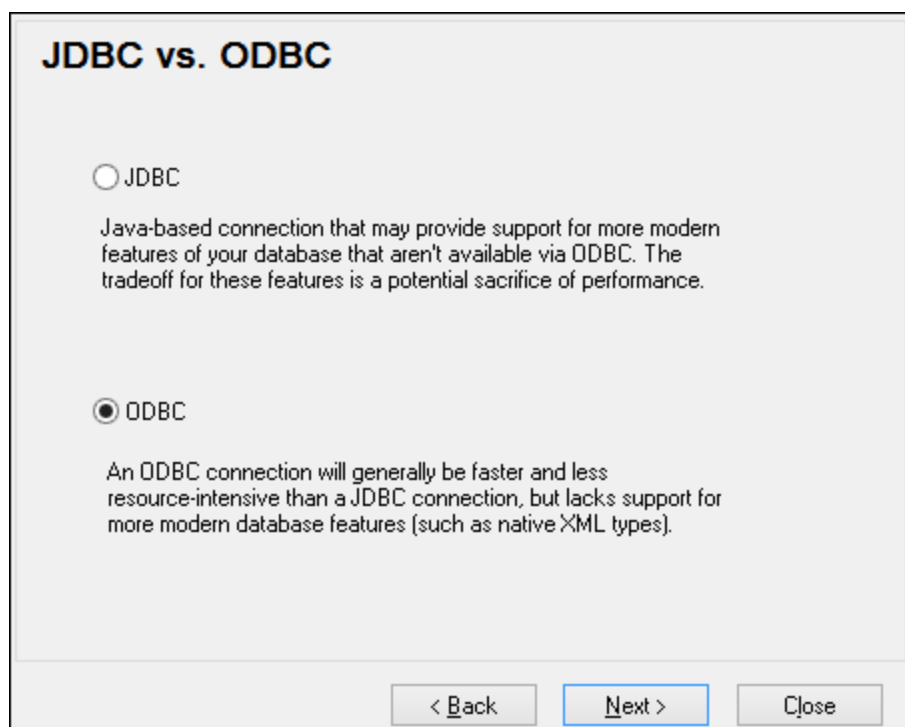
The path to the **tnsnames.ora** file depends on the location where Oracle home directory was installed. For Oracle database client 11.2.0, the default Oracle home directory path could be as follows:

```
C:\app\username\product\11.2.0\client_1\network\admin\tnsnames.ora
```

You can add new entries to the **tnsnames.ora** file either by pasting the connection details and saving the file, or by running the Oracle *Net Configuration Assistant* wizard (if available). If you want these values to appear in dropdown lists during the configuration process, then you may need to add the path to the admin folder as a **TNS_ADMIN** environment variable.

To connect to Oracle using ODBC:

1. [Start the database connection wizard](#)[117].
2. Select **Oracle (ODBC / JDBC)**, and then click **Next**.

3.   Select **ODBC**.

4.   Click **Edit Drivers**.

5. Select the Oracle drivers you wish to use (in this example, **Oracle in OraClient11g_home1**). The list displays the Oracle drivers available on your system after installation of Oracle client.
6. Click **Back**.
7. Select **Create a new data source name (DSN) with the driver**, and then select the Oracle driver chosen in step 4.

Avoid using the Microsoft-supplied driver called **Microsoft ODBC for Oracle** driver. Microsoft recommends using the ODBC driver provided by Oracle (see http://msdn.microsoft.com/en-us/library/ms714756%28v=vs.85%29.aspx)

8.   Click **Connect**.

9.   In the Data Source Name text box, enter a name to identify the data source (in this example, **Oracle DSN 1**).
10. In the TNS Service Name box, enter the connection name as it is defined in the **tnsnames.ora** file (see prerequisites [186]). In this example, the connection name is **ORCL**. *Note:* If you wish to have the dropdown list of the combo box populated with the values of the **tnsnames.ora** file, then you may need to add the path to the admin folder as a **TNS_ADMIN** environment variable.
11. Click **OK**.

12. Enter the username and password to the database, and then click OK.

4.10.15   PostgreSQL (ODBC)

This example illustrates how to connect to a PostgreSQL database server from a Windows machine through the ODBC driver. The PostgreSQL ODBC driver is not available on Windows, so it must be downloaded and

installed separately. This example uses the psqlODBC driver (version 11.0) downloaded from the official website (see also Database Drivers Overview [119]).

Note: You can also connect to a PostgreSQL database server directly (without the ODBC driver), see Setting up a PostgreSQL Connection [145] .

Prerequisites:
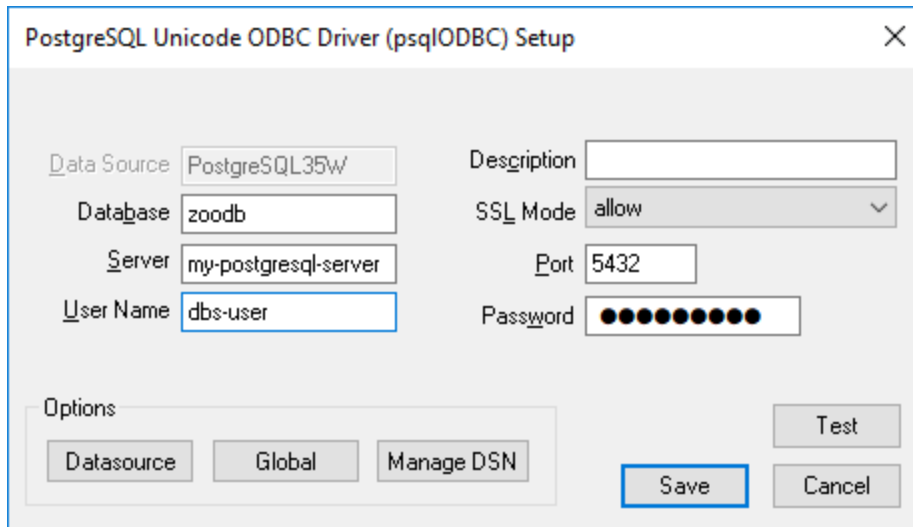
- *psqlODBC* driver must be installed on your operating system.
- You have the following database connection details: server, port, database, user name, and password.

To set up a connection to PostgreSQL using ODBC:

1. Start the database connection wizard [117] .
2. Click **ODBC Connections**.
3. Select the **User DSN** option.
4. Click **Create a new DSN** ✚ and select the driver from the drop-down list. If no PostgreSQL driver is available in the list, make sure that the PostgreSQL ODBC driver is installed on your operating system, as mentioned in the prerequisites above.
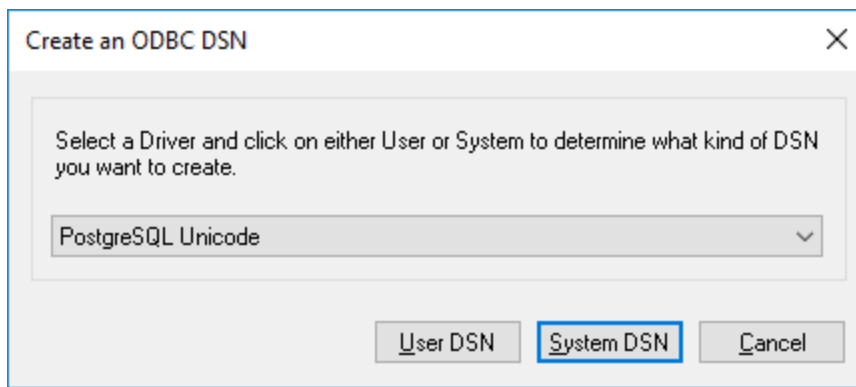
5. Click **User DSN**.
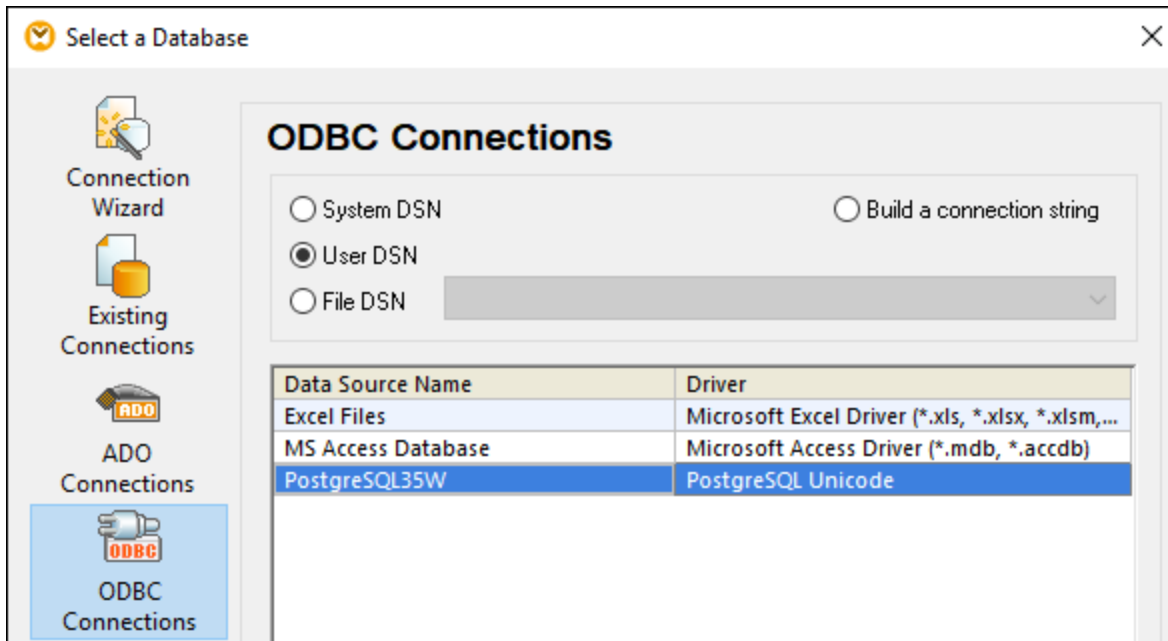
6. Fill in the database connection credentials (these must be supplied by the database owner), and then click **Save**.

The connection is now available in the list of ODBC connections. To connect to the database, you can either double-click the connection or select it, and then click **Connect**.

4.10.16   Progress OpenEdge (JDBC)

This example illustrates how to connect to a Progress OpenEdge 11.6 database server through JDBC.

Prerequisites

- JRE (Java Runtime Environment) or Java Development Kit (JDK) must be installed. This may be either Oracle JDK or an open source build such as Oracle OpenJDK. DatabaseSpy will determine the path to the Java Virtual Machine (JVM) from the following locations, in this order: a) The custom JVM path you may have set in application **Options**, see Java Settings ^{423}; b) The JVM path found in the Windows registry; c) The JAVA_HOME environment variable.
- Make sure that the platform of DatabaseSpy (32-bit, 64-bit) matches that of the JRE/JDK.
- The operating system's PATH environment variable must include the path to the bin directory of the JRE or JDK installation directory, for example C:\Program Files (x86)\Java\jre1.8.0_51\bin.
- The Progress OpenEdge JDBC driver must be available on your operating system. In this example, JDBC connectivity is provided by the **openedge.jar** and **pool.jar** driver component files available in **C:\Progress\OpenEdge\java** as part of the OpenEdge SDK installation.
- You have the following database connection details: host, port, database name, username, and password.

Connecting to OpenEdge through JDBC

1. Start the database connection wizard ^{117}.
2. Click **JDBC Connections**.
3. Next to "Classpaths", enter the path to the .jar file which provides connectivity to the database. If necessary, you can also enter a semicolon-separated list of .jar file paths. In this example, the required .jar file paths are: `C:\Progress\OpenEdge\java\openedge.jar;C:\Progress\OpenEdge\java\pool.jar;`. Note that you can leave the "Classpaths" text box empty if you have added the .jar file path(s) to the CLASSPATH environment variable of the operating system (see also Configuring the CLASSPATH ^{142}).
4. In the "Driver" box, select **com.ddtek.jdbc.openedge.OpenEdgeDriver**. Note that this entry is available if a valid .jar file path is found either in the "Classpaths" text box, or in the operating system's CLASSPATH environment variable (see the previous step).

5. Enter the username and password to the database in the corresponding text boxes.
6. Enter the connection string to the database server in the Database URL text box, by replacing the highlighted values with the ones applicable to your database server.

```
jdbc:datadirect:openedge://host:port;databaseName=db_name
```

7. Click **Connect**.

4.10.17   Progress OpenEdge (ODBC)

This example illustrates how to connect to a Progress OpenEdge database server through the Progress OpenEdge 11.6 ODBC driver.

Prerequisites:

- The *ODBC Connector for Progress OpenEdge* driver must be installed on your operating system. The Progress OpenEdge ODBC driver can be downloaded from the vendor's website (see also Database Drivers Overview [119]). Make sure to download the 32-bit driver when running the 32-bit version of DatabaseSpy, and the 64-bit driver when running the 64-bit version. After installation, check if the ODBC driver is available on your machine (see also Viewing the Available ODBC Drivers [139]).
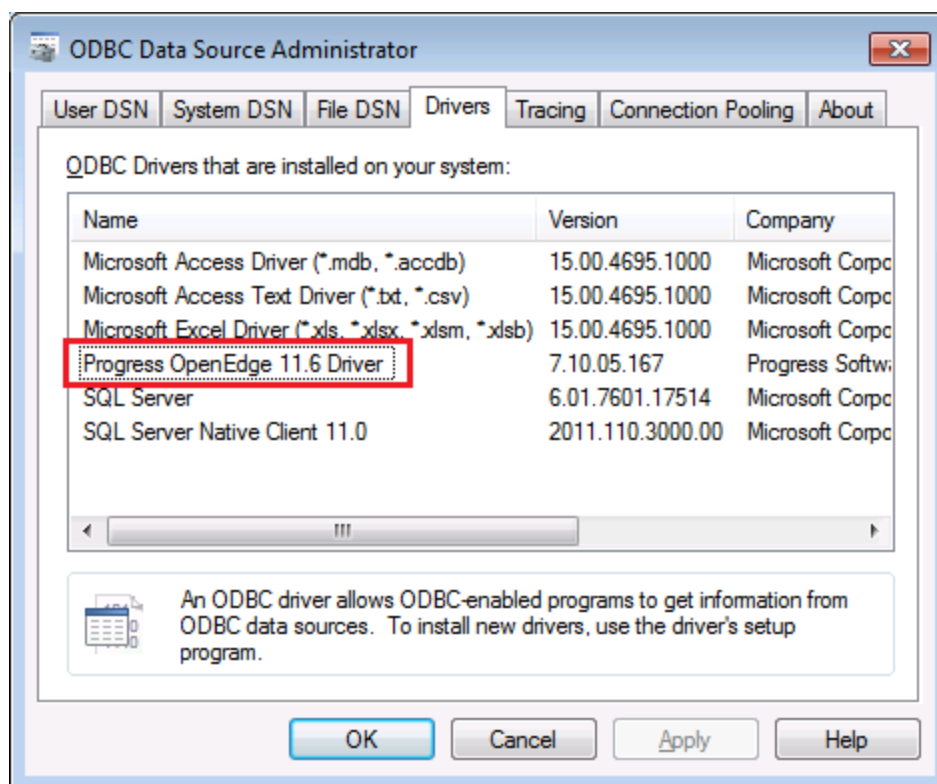
- You have the following database connection details: host name, port number, database name, user ID, and password.

Connecting to Progress OpenEdge through ODBC

1. Start the database connection wizard [117].
2. Click **ODBC Connections**.
3. Click **User DSN** (alternatively, click **System DSN**, or **File DSN**, in which case the subsequent instructions will be similar).
4. Click **Add** .
5. Select the **Progress OpenEdge Driver** from the list, and click **User DSN** (or **System DSN**, if applicable).
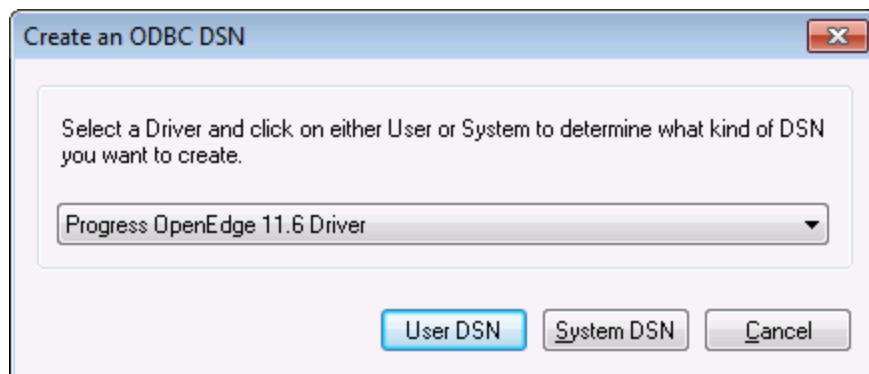
6. Fill in the database connection credentials (Database, Server, Port, User Name, Password), and then
    click **OK**. To verify connectivity before saving the entered data, click **Test Connect**.

7. Click OK. The new data source now appears in the list of ODBC data sources.
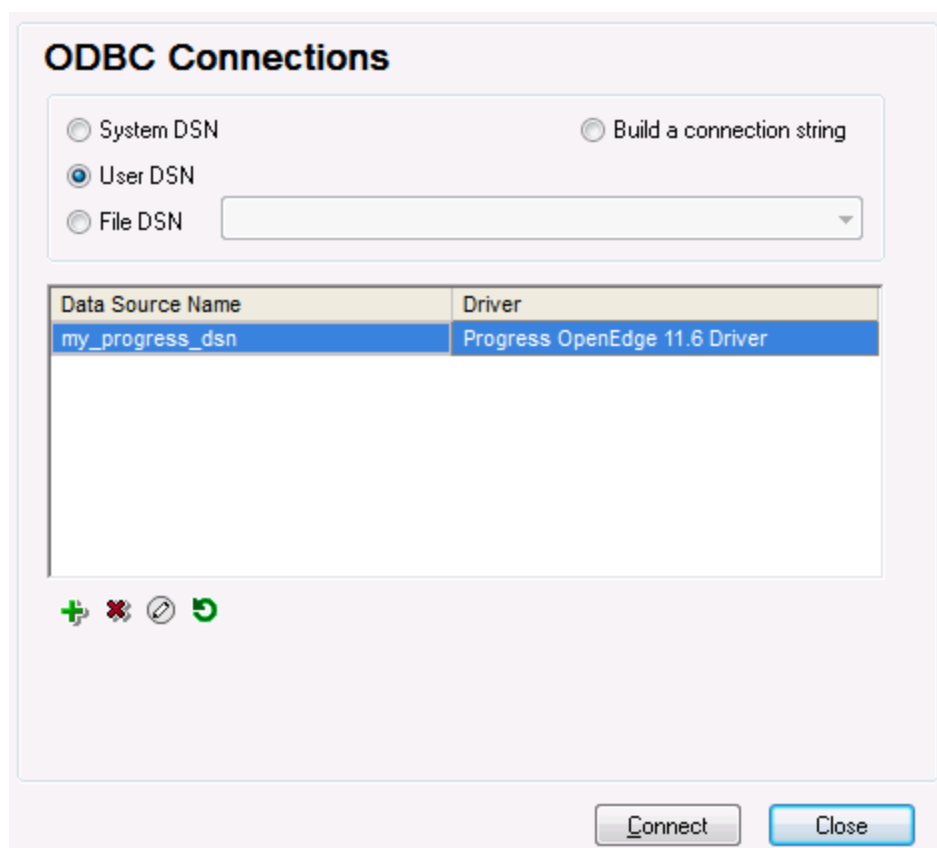
8. Click **Connect**.

4.10.18   Sybase (JDBC)

This example illustrates how to connect to a Sybase database server through JDBC.

Prerequisites:

- JRE (Java Runtime Environment) or Java Development Kit (JDK) must be installed. This may be either Oracle JDK or an open source build such as Oracle OpenJDK. DatabaseSpy will determine the path to the Java Virtual Machine (JVM) from the following locations, in this order: a) The custom JVM path you may have set in application **Options**, see Java Settings [423]; b) The JVM path found in the Windows registry; c) The JAVA_HOME environment variable.
- Make sure that the platform of DatabaseSpy (32-bit, 64-bit) matches that of the JRE/JDK.
- Sybase *jConnect* component must be installed on your operating system (in this example, *jConnect 7.0* is used, installed as part of the *Sybase Adaptive Server Enterprise PC Client* installation). For the installation instructions of the database client, refer to Sybase documentation.
- You have the following database connection details: host, port, database name, username, and password.

To connect to Sybase through JDBC:

1. Start the database connection wizard [117].
2. Click **JDBC Connections**.
3. Next to "Classpaths", enter the path to the .jar file which provides connectivity to the database. If necessary, you can also enter a semicolon-separated list of .jar file paths. In this example, the required .jar file path is: **C:\sybase\jConnect-7_0\classes\jconn4.jar**. Note that you can leave the "Classpaths" text box empty if you have added the .jar file path(s) to the CLASSPATH environment variable of the operating system (see also Configuring the CLASSPATH [142]).
4. In the "Driver" box, select **com.sybase.jdbc4.jdbc.SybDriver**. Note that this entry is available if a valid .jar file path is found either in the "Classpaths" text box, or in the operating system's CLASSPATH environment variable (see the previous step).

Classpaths:	C:\sybase\jConnect-7_0\classes\jconn4.jar;
Driver:	com.sybase.jdbc4.jdbc.SybDriver
Username:	dbuser
Password:	••••••
Database URL:	jdbc:sybase:Tds:SYBASE12:2048/PRODUCTSDB

[Connect] [Close]

5. Enter the username and password to the database in the corresponding text boxes.
6. Enter the connection string to the database server in the Database URL text box, by replacing the highlighted values with the ones applicable to your database server.

```
jdbc:sybase:Tds:hostName:port/databaseName
```

7. Click **Connect**.

4.10.19   Teradata (JDBC)

This example illustrates how to connect to a Teradata database server through JDBC.

Prerequisites:

- JRE (Java Runtime Environment) or Java Development Kit (JDK) must be installed. This may be either Oracle JDK or an open source build such as Oracle OpenJDK. DatabaseSpy will determine the path to the Java Virtual Machine (JVM) from the following locations, in this order: a) The custom JVM path you may have set in application **Options**, see Java Settings ⁴²³; b) The JVM path found in the Windows registry; c) The JAVA_HOME environment variable.
- Make sure that the platform of DatabaseSpy (32-bit, 64-bit) matches that of the JRE/JDK.
- The JDBC driver (one or more .jar files that provide connectivity to the database) must be available on your operating system. In this example, Teradata JDBC Driver 16.20.00.02 is used. For more information, see https://downloads.teradata.com/download/connectivity/jdbc-driver.
- You have the following database connection details: host, database, port, username, and password.

To connect to Teradata through JDBC:

1. Start the database connection wizard [117].
2. Click **JDBC Connections**.
3. Next to "Classpaths", enter the path to the .jar file which provides connectivity to the database. If necessary, you can also enter a semicolon-separated list of .jar file paths. In this example, the .jar files are located at the following path: **C:\jdbc\teradata**. Note that you can leave the "Classpaths" text box empty if you have added the .jar file path(s) to the CLASSPATH environment variable of the operating system (see also Configuring the CLASSPATH [142]).
4. In the "Driver" box, select **com.teradata.jdbc.TeraDriver**. Note that this entry is available if a valid .jar file path is found either in the "Classpath" text box, or in the operating system's CLASSPATH environment variable (see the previous step).

JDBC Connections

Enter a connection string and select (or enter manually) a valid JDBC driver. Click on 'Connect' to proceed.

Classpaths: `C:\jdbc\teradata\terajdbc4.jar;C:\jdbc\teradata\tdgssconfig.jar`

Driver: `com.teradata.jdbc.TeraDriver`

Username: `demouser`

Password: `●●●●●●●●`

Database URL: `jdbc:teradata://demodatabase`

[Connect] [Close]

5. Enter the username and password to the database in the corresponding text boxes.
6. Enter the connection string to the database server in the Database URL text box, by replacing the highlighted value with the one applicable to your database server.

```
jdbc:teradata://databaseServerName
```

7. Click **Connect**.

4.10.20   Teradata (ODBC)

This example illustrates how to connect to a Teradata database server through ODBC.

Prerequisites:

- The Teradata ODBC driver must be installed (see https://downloads.teradata.com/download/connectivity/odbc-driver/windows. This example uses Teradata ODBC Driver for Windows version 16.20.00.
- You have the following database connection details: host, username, and password.

To connect to Teradata through ODBC:

1. Press the **Windows** key, start typing "ODBC", and select **Set up ODBC data sources (32-bit)** from the list of suggestions. If you have a 64-bit ODBC driver, select **Set up ODBC data sources (64-bit)** and use 64-bit DatabaseSpy in the subsequent steps.
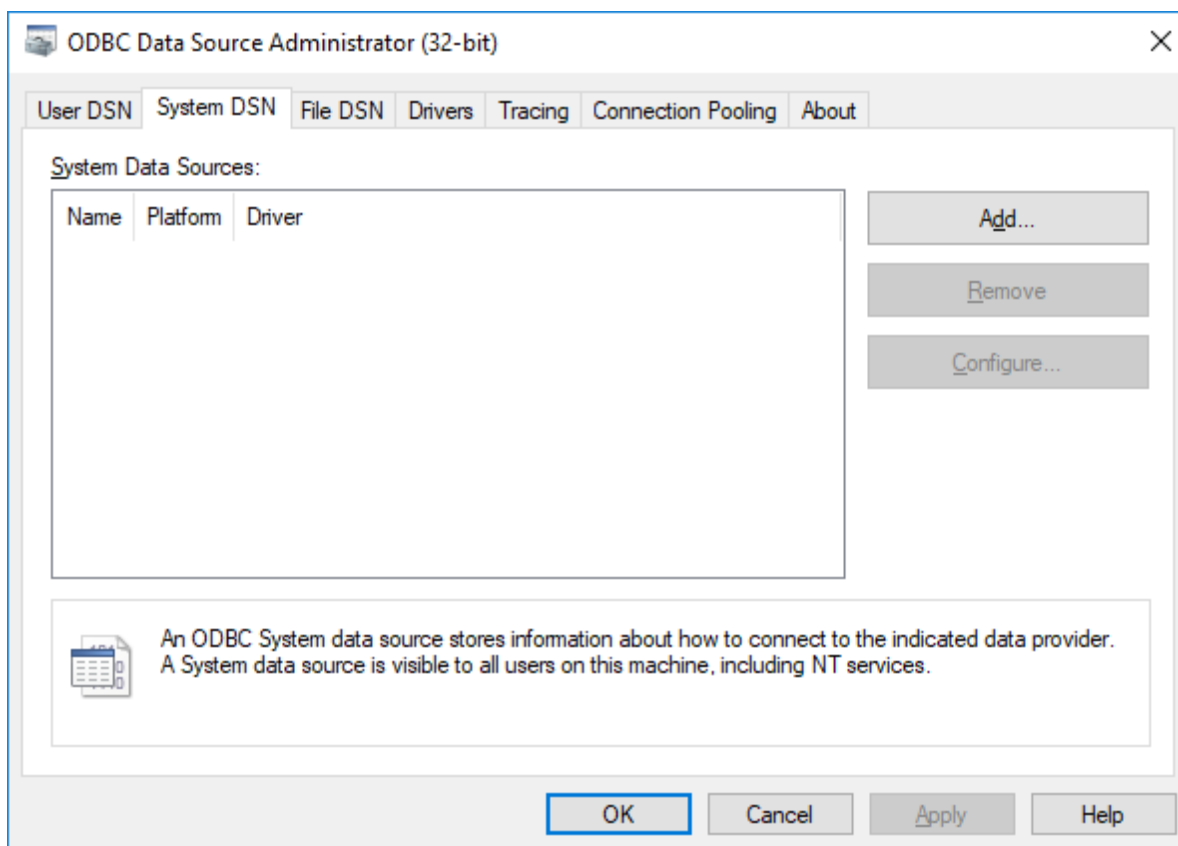
2. Click the **System DSN** tab, and then click **Add**.
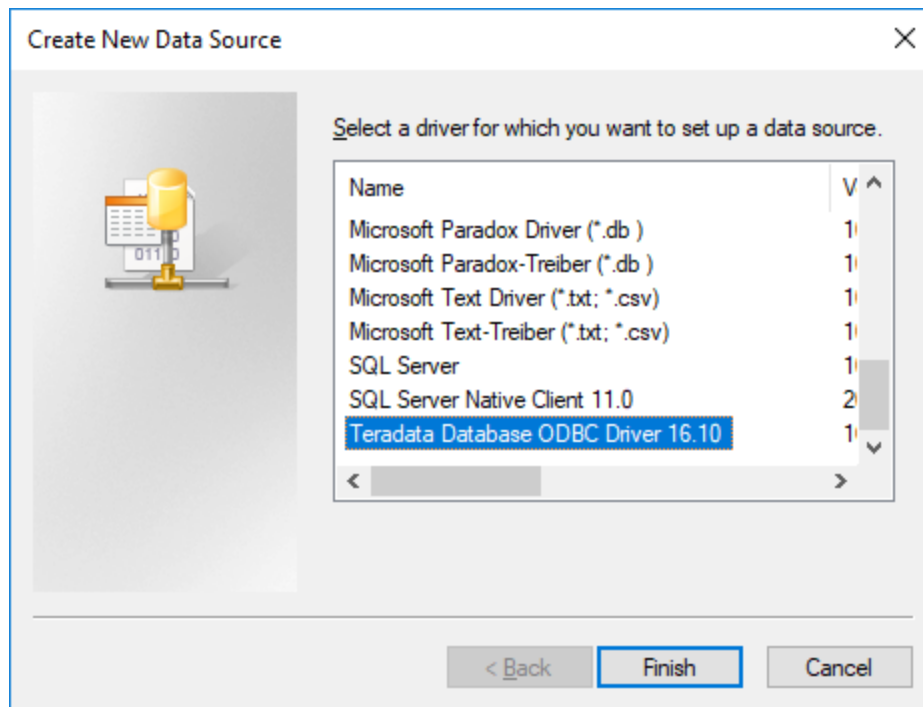
3. Select **Teradata Database ODBC Driver** and click **Finish**.

4. Enter name and, optionally, a description that will help you identify this ODBC data source in future. Also, enter the database connection credentials (Database server, User, Password), and, optionally, select a database.
5. Click **OK**. The data source now appears in the list.
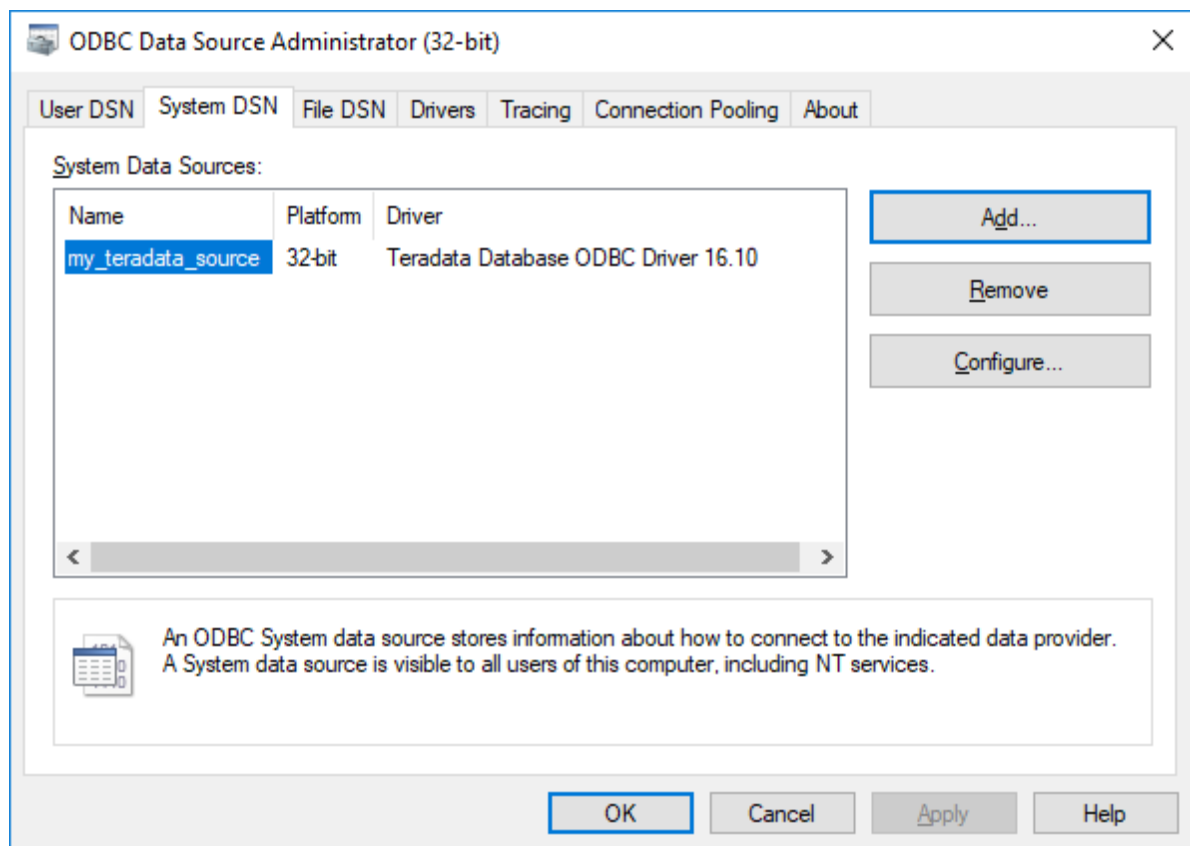
6. Run DatabaseSpy and start the database connection wizard [117].
7. Click **ODBC Connections**.
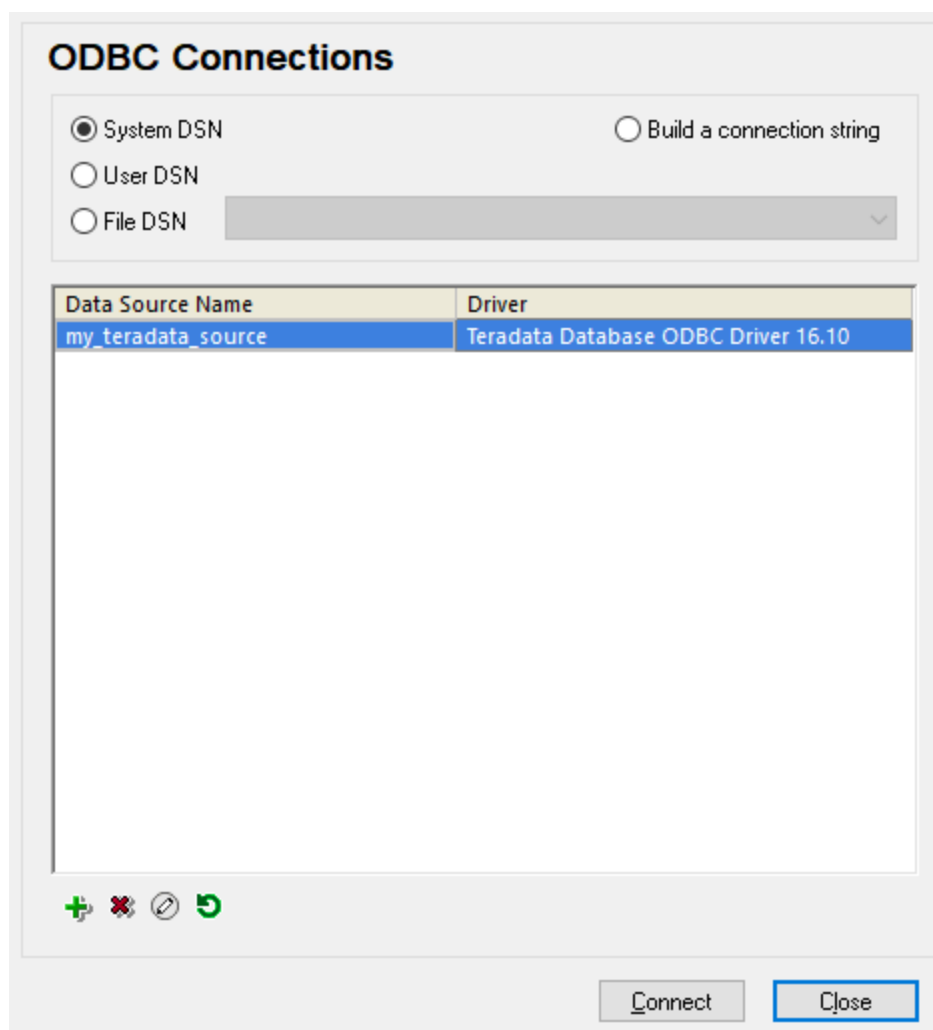
8. Click **System DSN**, select the data source created previously, and then click **Connect**.

Note: If you get the following error: "The driver returned invalid (or failed to return) SQL_DRIVER_ODBC_VER: 03.80", make sure that the path to the ODBC client (for example, **C:\Program Files\Teradata\Client\16.10\bin**, if you installed it to this location) exists in your system's PATH environment variable. If this path is missing, add it manually.

5   DatabaseSpy Projects

A DatabaseSpy project organizes all the files that you need when working with your databases in a single project file. It includes data source connections, SQL files, database design files, and shortcuts to your favorite database objects. A project can also include references to files external to DatabaseSpy—such files open in the application they were registered with.

The content of your project is displayed in the [Project][20] window.

Note: DatabaseSpy projects created in one version (for example Database 2023) can only be opened with the same or later versions. Also, DatabaseSpy Enterprise can open projects saved with DatabaseSpy Professional, but the opposite is not possible.
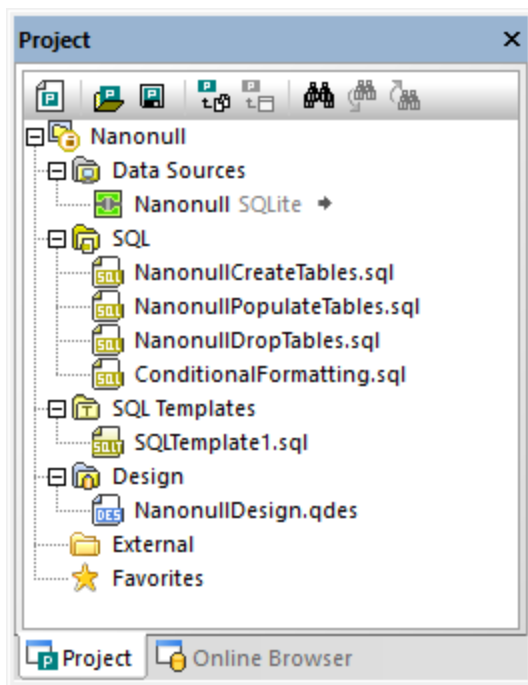
Project window

To create a new project:

- In the Project window, click the **Create a new project** button or select the menu option **File | New | Project**.

   If there is an unsaved project already open in the Project window, you are prompted to save this project.

To open an existing project:

- In the Project window, click the **Open a project** button or select the menu option **File | Open | Open Project** and choose a project file from the Open dialog box.
- Select the menu option **File | Open | Open File...**, select "Altova DatabaseSpy Project (*.qprj)" from the Files of type drop-down list, and choose a project file from the Open dialog box.

  If there is an unsaved project already open in the Project window, you are prompted to save this project.

To save a project:

- In the Project window, click the **Save Project** button or select the menu option **File | Save Project As**.

To save the project under a new name and change its title:

1. On the **File** menu, click **Save Project As**.
2. Select the folder where you want to save the project file and enter a name for the file.
3. Click **Save**.

To close a project:

- Open an existing project, or create a new one.

To delete a project:

- Delete the respective *.qprj file from your hard disk.

5.1   Adding Data Sources

Working with any kind of database in DatabaseSpy requires that you add a data source to the project. All data sources added to the project appear in the Data Sources folder in the Project window.

To add a data source to your project:

Do one of the following:

- On the **File** menu, click **Create a Database Connection**.
- Press the keyboard shortcut **Ctrl+Q**.
- Click the **Connect to a database** [icon] toolbar button.
- In the Project window, right-click the Data Sources folder and select **Add a New Data Source...** from the context menu.

From this point onwards, follow the connectivity instructions applicable to the database kind or to the interface you want to connect (ADO, ADO.NET, ODBC, JDBC, or native), see Connecting to a Database [115]. For database connection examples applicable to various database kinds, see Database Connection Examples [152].

Data source options

When you right-click a data source name in the Project window, you can choose from several options depending on the current state (connected or disconnected) of the data source and whether or not the data source is a global resource. The corresponding keyboard shortcuts are shown to the right of the options if they are available.

Connect		Connects to a data source if it is disconnected.
Disconnect		Disconnects from a data source if it is connected.
Remove	**Del**	Removes a data source from the project.
Rename	**F2**	Renames a data source. Note that only disconnected data sources can be renamed.
Convert to Global Resource		Converts the data source to an Altova Global Resource (see Altova Global Resources [436]) and adds a database alias with the data source name to the **GlobalResources.xml** definition file. If an alias with that name already exists, a warning message will be displayed and the operation will fail.
Copy Global Resource into Project		Generates a copy of the global resource and adds it to the project as stand-alone data source. You will have to specify a name for the new data source. The original global resource remains in the project.
Edit Global Resource...		Opens the Global Resource dialog box where you can edit the configuration of the database alias.

To copy a global resource into a project:

1. In the Project window, right-click a data source that has been added to the project as global resource and select **Copy Global Resource into Project** from the context menu.
2. Enter a name in the Set a Data Source Name dialog and click **OK**.
3. Optionally, remove the global resource from the project.

To remove a data source from a project:

Select the data source you want to remove and do one of the following:

- Right-click the data source and choose **Remove** from the context menu.
- Press **Delete.**

If the data source is connected, DatabaseSpy displays a warning that removing the data source will close the connection.

Renaming data sources

Data sources in projects can be renamed either in the Properties window or via the context menu directly in the Project window.

Note: You can only rename data sources that are currently disconnected. Both the **Rename** command in the context menu and the title bar in the Properties window are grayed out for connected data sources.

To rename a data source in a project:

1. Make sure that the data source is disconnected.
2. Select the data source you want to rename and do one of the following:

   - In the Properties window, double-click the title bar.
   - In the Project window, right-click the data source and select **Rename** from the context menu.
   - Press **F2**.

3. Enter the new name and press **Enter**.
4. Save the project [207].

5.2   Adding Files

You can add existing SQL scripts to your DatabaseSpy Project, or you can save designs as files and add them to the project. The Project window provides folders for the different file types and you can define properties (e.g., data source to connect to, execution options for SQL, etc.) on file or folder level. This way, you have all the files that you may need at hand, even if they are stored in different locations. If you need to find the exact path to a file that has been added to a project, DatabaseSpy provides the following context menu option:

· Right-click a file in the Project window and select **Locate File...** from the context menu. DatabaseSpy opens the folder that contains the file in Windows Explorer.

To add files to a project:

1. In the Project window, right-click a folder and select **Add Files to Project...** from the context menu. Alternatively, click the **Add files to the Project** button in the Project window toolbar. The **Open** dialog appears.
2. Browse for the file that you want to add to the project.
3. Click **Open**. The selected file is added to the corresponding folder of the project.
4. Save the project file [207].

In addition, you can also save the content of the SQL Editor or Design Editor to a file and add this file to the project at the same time. For every file that is saved for the first time or under a new name, DatabaseSpy pops up the Add to Project dialog box.
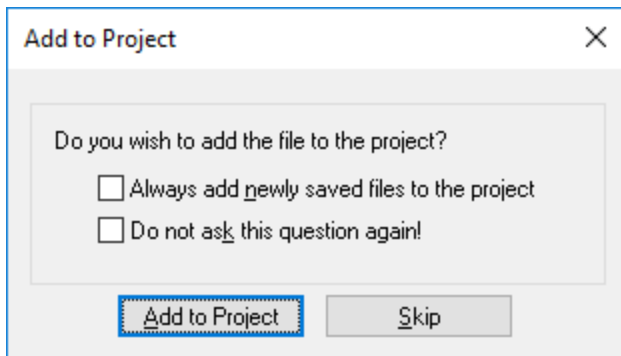
The Add to Project dialog box also serves as a shortcut to the General options [418], allowing you to define that newly saved files be always added to the project automatically. In addition, you can also suppress the display of this dialog box in the future.

To save a file:

1. Make the SQL Editor or Design Editor the active window and do one of the following:

· Click the **Save** button in the Standard toolbar.
· Select the menu option **File | Save**.
· Press **Ctrl+S**.

- Right click the naming tab at the bottom of the document window and select **Save...** from the context menu.

2. Enter a name for the new file, select a folder and click the **Save** button. The Add to Project dialog box opens (if the dialog box does not pop up, check whether the **Do not show dialog** option is checked in the Newly saved files group box of the General options ⁴¹⁸).

3. Do one or more of the following:

- Click the **Add to Project** button to add the file to the currently active project.
- Click **Skip** to close the dialog box without adding the file to the project.
- Select the **Always add newly saved files to the project** check box.
- Select the **Do not ask this question again!** check box to suppress the display of the dialog box in the future.

To add the active file to the project:

1. Make the window that contains the file you want to add to the project the active window.
2. Do one of the following:

- In the Project window, click the **Add the Active File** icon in the toolbar, or right-click the respective folder and select **Add Active File to Project** from the context menu.
- In the SQL Editor or Design Editor, right-click the tab at the bottom of the Editor window and select **Add to project** from the context menu. If the file has not been saved yet, the Windows **Save As** dialog box opens where you can enter a name and path for the file.

The file appears below the respective folder in the Project window.

3. Save the project file ²⁰⁷ .

To remove a file from a project:

Do one of the following:

- Right-click the file and select **Remove** from the context menu.
- Select the file and press **Delete**.

5.3 Favorites

Frequently used database objects that have been added to your favorites are displayed in the Favorites folder. All options for an individual object are available in the context menu and you can directly open the Online Browser by clicking the arrow to the right of the object name.

DatabaseSpy's Favorites feature allows you to mark frequently used objects in your project as favorites. These items then show up in the Favorites folder of the Project window and are shown bold in the Online Browser so that you always have quick access to them.
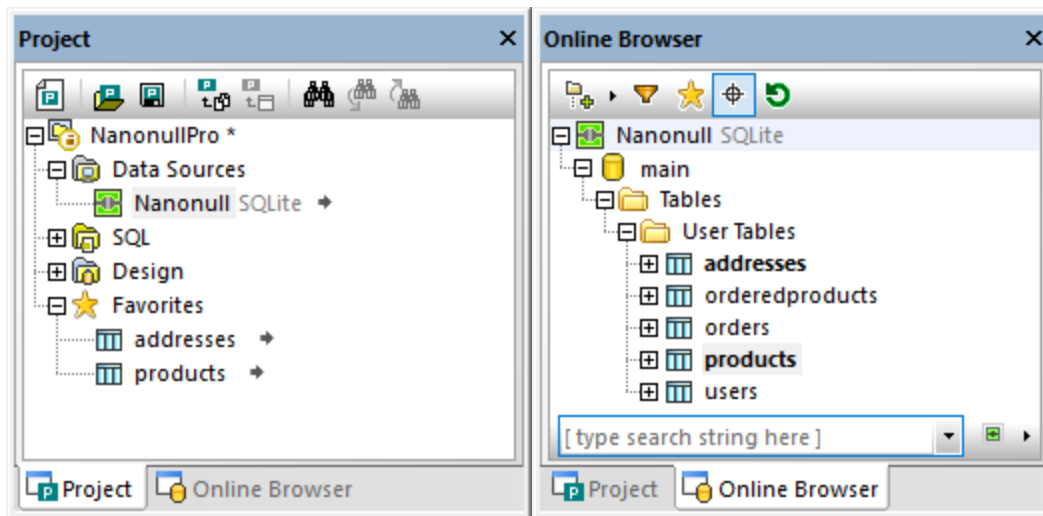
In the Project window, an arrow is shown in the Favorites folder next to the item name. When you click on this arrow, the Online Browser opens with the favorite item selected. Note that you can also add databases and schemas or individual table columns to the Favorites folder.

For every item in the Favorites folder, the relevant options are available in the context menu. For example, you can connect a data source directly from within the favorites folder.

To add items as favorites:

- In the Online Browser, right-click the respective tables or other database objects and select **Add to/Remove from Favorites (Ctrl+F2)** from the context menu.
- Alternatively, if you have activated the **Enable full row selection** check box in the Online Browser[420] options, click the **Favorites** icon that appears to the right of a database object if you move the mouse cursor over it in the Online Browser.

  The tables or objects are added to the Favorites folder and are shown in bold type in the Online Browser. In addition, the **Favorites** icon beside the object name appears yellow when full row selection is enabled.
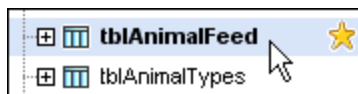
To view only the favorites in the Online Browser:

1. Click the **Favorites** [icon] icon to switch to the favorites view. While the **Favorites** icon is active, only database objects marked as favorites are displayed in the Online Browser.
2. Click the **Favorites** icon again to switch back to the standard Online Browser view.

To rename database objects in the Favorites folder:

1. Make sure that the corresponding data source of the favorites item is connected.
2. Select the database object you want to rename in the favorites folder and do one of the following:

    *   In the Properties window, double-click the title bar.
    *   In the Project window, right-click the database object in the favorites folder and select **Rename** from the context menu.
    *   Press **F2**.

3. Enter the new name and press **Enter**. A change script is generated in the Database Structure Change Script window.
4. Execute the change script.

To remove a database object from the Favorites folder:

Do one of the following:

*   In the Project tab, right-click the object in the Favorites folder and select **Remove from Favorites (Del)**.
*   In the Online Browser, right-click the object in its folder and select **Add to/Remove from Favorites (Ctrl+F2)**.
*   When full row selection [420] is enabled, in the Online Browser, move the mouse cursor over a favorite item and click the yellow favorites item to the right of its name.

To remove all database objects from the Favorites folder:

*   In the Project tab, right-click the Favorites folder and select **Remove all Favorites**.

5.4   Properties

In DatabaseSpy you can view properties by selecting objects in (i) the [Project window](#) [215], (ii) the Online Browser, or (iii) the Design Editor. In addition, you can define properties for [designs](#) [219] and [SQL files](#) [218] that have been added to a project. The properties are displayed in the Properties window.

Changing a property in the Property Window or in the Design Editor automatically generates an SQL change script which appears in the in the [Database Structure Change Script window](#) [37] either automatically or after you have clicked the **Edit Change Script** [icon] button, depending on how you have configured the [Change Script options](#) [420]. Properties that cannot be edited are grayed out.

When folders are displayed, clicking a folder shows an overview of the folder's content in the Properties window.

The properties that are displayed in the Properties window depend on the database object that is currently selected in the Online Browser or Design Editor window. If an object has child objects, the content of the Properties window changes in accordance with the level to which you expand the object, that is, only properties for expanded items are displayed in the Properties window.

For the data source connection, DatabaseSpy displays the same properties as it does in the [Project window](#) [215]. The properties that are displayed when you click on a database in the Online Browser are for information purposes only and cannot be changed.

Object count

For every database object that contains child objects, an overview section is displayed. This section provides hyperlinks for counting the number of the individual children of the database object as well as a hyperlink for counting all children of the object. Note that the count is retrieved automatically if you expand the object in the Online Browser or show a table in the Design Editor.

To retrieve the number of child objects in the Overview section of the Properties window, you have, therefore, the following options:

- Click the "Update all Counts" hyperlink to retrieve the count for all child objects.
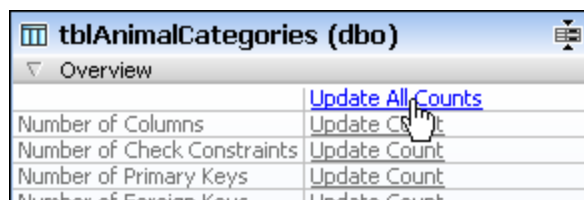
- Click the "Update Count" hyperlink for an individual database object to retrieve the count for this specific item.
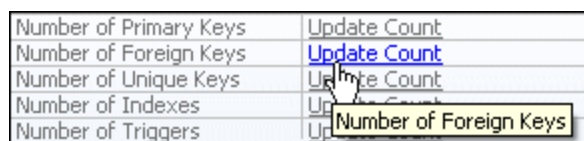
- Show a table in the Design Editor; this automatically retrieves the count for all child objects of the table.

- Right-click a table and select **Expand | Children** from the context menu. This expands all sub-folders of the table in the Online Browser and thus shows the number of child objects in the table properties.
- Expand one or more sub-folders in the Online Browser and then click the table again. The number of items contained in the folder is displayed in the table properties.

Changing properties

The support for changing the properties of database objects varies by database kind. If the database table already contains data, changing a property may not be possible if it violates the table or column constraints. For instructions specific to each database object, see Managing Database Objects [222].

5.4.1   Project Properties

The project properties are displayed in the Properties window [30] when you click on an object in the Project window [20]. Some of the properties are editable. Properties that cannot be edited are grayed out.

Inherited and overridden properties

If you create a folder or a file in the Project window, it will inherit the properties of the parent folder. You can always change (override) the property for each individual file or folder, in which case the property appears in bold. To restore the property value to that of its parent, right-click it, and select **Restore value to parent's option** from the context menu.

For example, let's assume you have created a sub-folder called "SQL Server" in your Project window, and that you have set the **Database Kind**, **Data Source**, and **Root Object** properties for that sub-folder as shown below. All these properties appear in bold type, because they are overridden—that is, no such properties have been defined for the parent "SQL" folder.
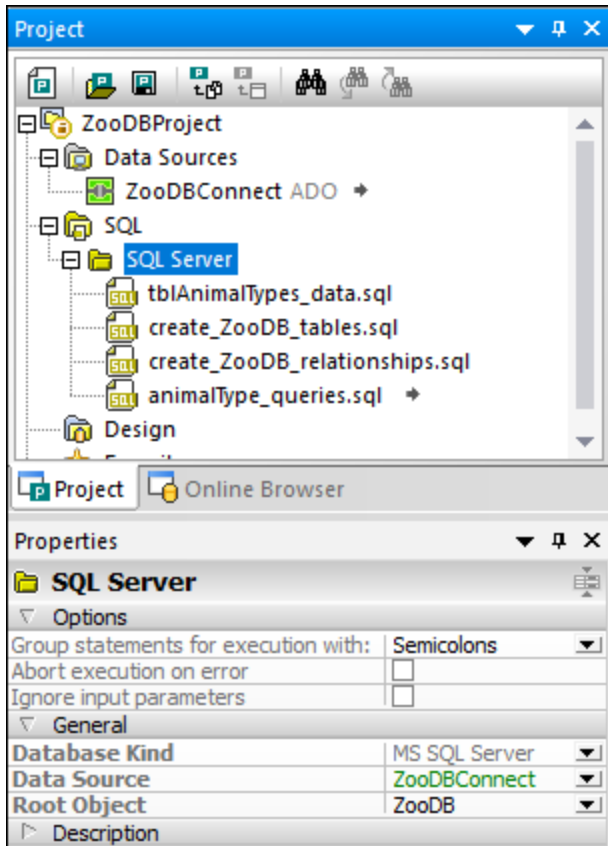
The file **animalType_queries.sql** in the "SQL Server" folder inherits the properties from the parent folder, so all properties appear in normal font type. However, if you select the **Abort execution on error** check box, this property is now displayed in bold type, because it is no longer inherited.
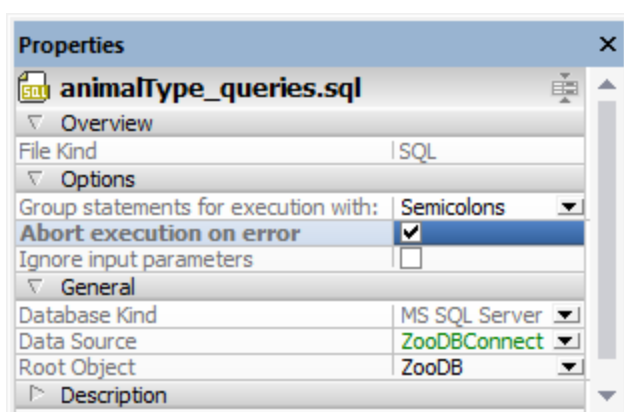
If you clear the **Abort execution on error** check box again, the property will still be displayed bold although the value is now identical to the parent folder's value. This indicates that the tie between the folder's properties and the file's properties has been broken and—although currently identical—an overridden value has been defined for this property.

To make the property inherit again the parent value, right-click it and select **Restore value to parent's option** from the context menu.

Data source properties

When you select a data source in the Project window, the Properties window display the properties applicable to it.

To automatically connect to the data source when the project is opened, select the **Connect on Open Project** check box in the Options section of the data source properties.

To view data sources in the Online Browser also when they are not connected, select the **Always show in Online Browser** check box.

The **Global Resource** check box is activated if the data source has been added to the projects as a global resource. Note that this property cannot be changed by clearing the check box. You must copy the global resource into the project [441] and remove the global resource from the project to convert a global resource into a local one.

When selected, the **Persistent Structure Storage** check box persists the structure information associated with the data source even after DatabaseSpy has been restarted. To discard persisted data, click the **Refresh** button at the top of the Online Browser window.

For ODBC connections, the **Use ODBC for retrieval** check box enables you to retrieve structural information about the database using the API methods provided from ODBC directly rather than DatabaseSpy's internal SQL queries. This might be desired when you are experiencing difficulties retrieving data, or feel the application is too slow when retrieving this information. The down side to this approach is that the amount of structural information that one can retrieve via the ODBC interface is limited.

Where applicable, the **Driver Information** section lists the name and version of the driver used for the selected data source connection.

You can also manually edit the connection details in the **Connection** section of the data source properties, but keep in mind that a project might not be able to establish a connection to the data source if the connection properties have been manipulated.

Detailed properties of the individual database objects [214] can be viewed by selecting the respective database object in the Online Browser. You must connect to a data source before you can view its database objects in the Online Browser.

SQL folder properties

When you select the **SQL** folder in the Project window, you can edit the properties that apply to the entire folder and are inherited by all files contained in this folder. These properties can be overridden by properties that you define on file or sub-folder level.

The General section provides three drop-down lists for database kind, data source, and root object as well as text fields containing the folder and project names. The Description section shows a text field for entering a description. Please note that the properties for database kind and root object are filled in automatically when you select a data source from the drop-down list. However, you can then change the root object, if applicable.

Design and Favorites properties

For the **Design** and **Favorites** folders, the Properties window provides drop-down lists for database kind, data source, and root object. In addition, a text field for entering a description is available.

5.4.2   SQL Properties

The following properties of SQL files as well as SQL Editor windows (with or without SQL statements) are displayed in the Properties window:

Overview

File Kind	Always reads "SQL" for SQL files.

Options

Group statements for execution with	Here you can choose how statements should be grouped when the script in the SQL Editor window is executed. See also [Executing SQL](#) [364].
Abort execution on error	SQL scripts may contain multiple statements in the same file. If any of the statements fails during execution, you may want to abort executing of the rest. Select this check box to enable this behavior.
Ignore input parameters	Applicable only for DatabaseSpy Enterprise edition. If the SQL file contains statements that contain parameters ([SQL Templates](#) [382]), a dialog box appears when you run the script. Select this check box to suppress the dialog box.

General

Database Kind	For unsaved SQL Editor windows, this drop-down list lets you define a database type. Note that this field will be adapted automatically if you choose a data source.
Data Source	Here you can choose from the data sources that have been added to the project. As several data sources may be active at one time, it is necessary to be able to assign data sources to SQL files as a group or individually.

Note: Assignments made in the **Project** tab apply to the SQL files when they are opened or executed. |
| Root Object | Displays the root object of the current connection. This is relevant only for those database kinds that support a single database as the root object. |

Description

Description	Here you can enter a description of the file.

Files

Creation Date Modification Date Access Date	Displays the respective creation and modification dates and shows when the file has last been accessed.
Size	Shows the file size.
Read Only Hidden	These properties are read-only and can only be changed outside DatabaseSpy.
Full Path	Displays the full path to the file. To jump directly to this location, right-click the file in the Project window and select **Locate File...** from the context menu.

Default properties for SQL Editor windows

The following scenarios can occur when you open a new SQL Editor window:

- **No project**: If no project has been opened in the Project window, any new SQL Editor window that you open will have the properties Database Kind = "Unknown" and Data Source = "Offline".
- **Data source connected**: If you open an SQL Editor window immediately after you have connected to a data source, the connection properties of this data source will be used for that SQL Editor window. Note that the data source still has to be selected in the Project window.
- **Several data sources defined in one project**: If you have opened a project with several different data sources, you can set the connection properties by clicking on a particular data source before opening an SQL Editor window. If you select a disconnected data source, the Execution Target Bar provides you with a **Connect** button to establish a connection to the data source. Clicking on any other item in the Project window will open an SQL Editor in the Offline mode.

Working with SQL files in the Offline mode

One of the options available to you when selecting from the **Data Source** combo box is "Offline". Designating an SQL file as offline means that it is blocked from being executed. The file cannot be executed when opened in the SQL Editor or when selecting **Execute SQL** from the context menu in the Project window.

All SQL Editor commands and functions are available when editing this file however, you just have the added security of not being able to execute it without having to change the settings in the Properties window.

5.4.3   Design Properties

In addition to the Properties window, you can show the properties of a column, an index, a key, or a check constraint directly in a Design Editor window. Design Editor's table display shows the data type and the **Nullable** check box for columns, the column and order for indexes, the column and referenced key for keys, and the expression for check constraints.

When you select an index, key, or check constraint in the table display in the Design Editor, the referenced column is automatically printed in bold.
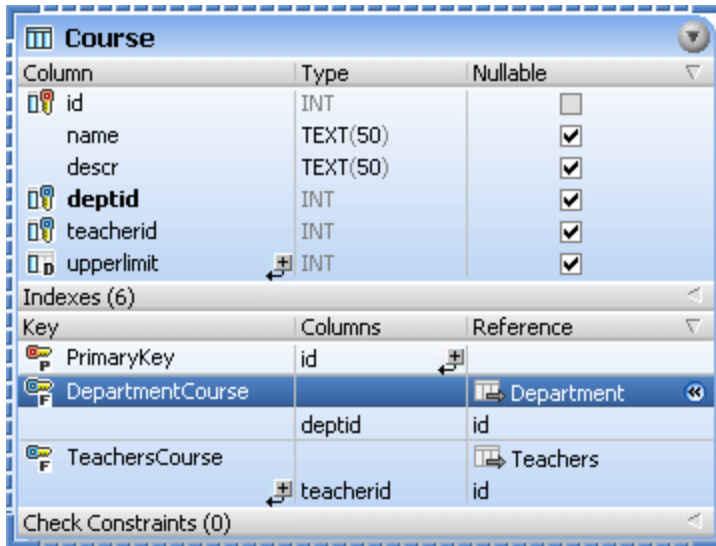
When you click into an empty area of a Design Editor window, the design properties are displayed in the Properties window. Here you can change the data source or schema and view the file properties and path of the design file.

Overview

File Kind	Always reads "Design" for design files (*.qdes).

General

Database Kind	For unsaved SQL Editor windows, this drop-down list lets you define a database type. Note that this field will be adapted automatically if you choose a data source.
Data Source	Here you can choose from the data sources that have been added to the project. As several data sources may be active at one time, it is necessary to be able to assign data sources to SQL files as a group or individually. **Note**: Assignments made in the **Project** tab apply to the SQL files when they are opened or executed.
Root Object	Displays the root object of the current connection. This is relevant only for those database kinds that support a single database as the root object.

Description

Description	Here you can enter a description of the file.

Files

Creation Date *Modification Date* *Access Date*	Displays the respective creation and modification dates and shows when the file has last been accessed.
Size	Shows the file size.
Read Only *Hidden*	These properties are read-only and can only be changed outside DatabaseSpy.
Full Path	Displays the full path to the file. To jump directly to this location, right-click the file in the Project window and select **Locate File...** from the context menu.

Note: The connection properties can be changed only for new designs. As soon as you show a table in a Design Editor window, the connection properties cannot be changed anymore.

To view the properties of a design file:

Do one of the following:

- Click into an empty area of a Design Editor window.
- If the design file has already been added to the project, click the .qdes file in the Design folder of the Project window.

The properties of the design are displayed in the Properties window.

6   Managing Database Objects

This chapter deals with viewing and modifying the *structure* of database objects. If you are looking to interact with *data* stored by database objects, see Retrieving and Editing Data [284].

In DatabaseSpy, the Online Browser [24] is the primary way to navigate the database structure and explore database objects in a hierarchical fashion. You can also add new database objects or modify the existing ones using the context menu actions.
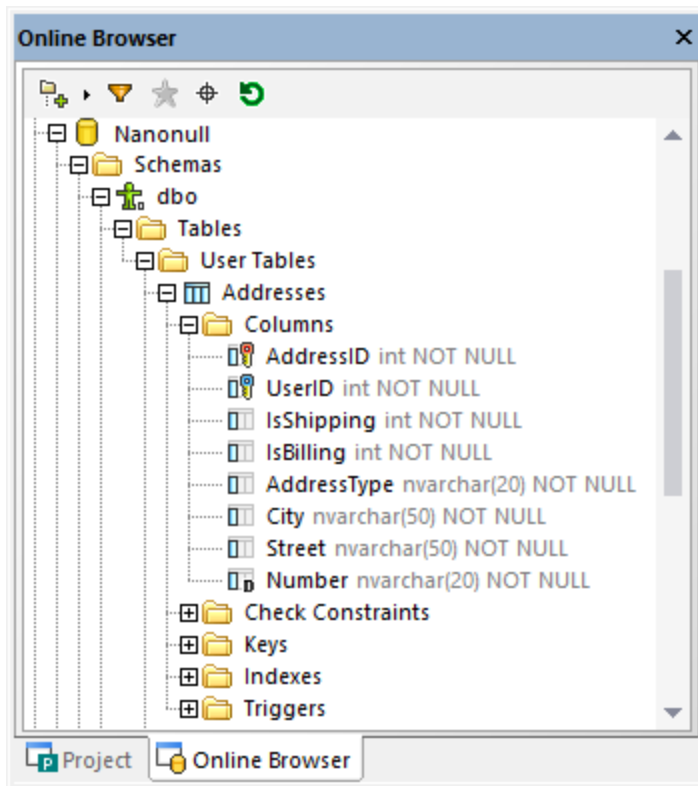
Online Browser

The Design Editor [46] provides the means to view multiple tables and relationships between them as a diagram that you can save to a file. In Design Editor, you can configure the level of displayed detail and you can change the database structure as well.
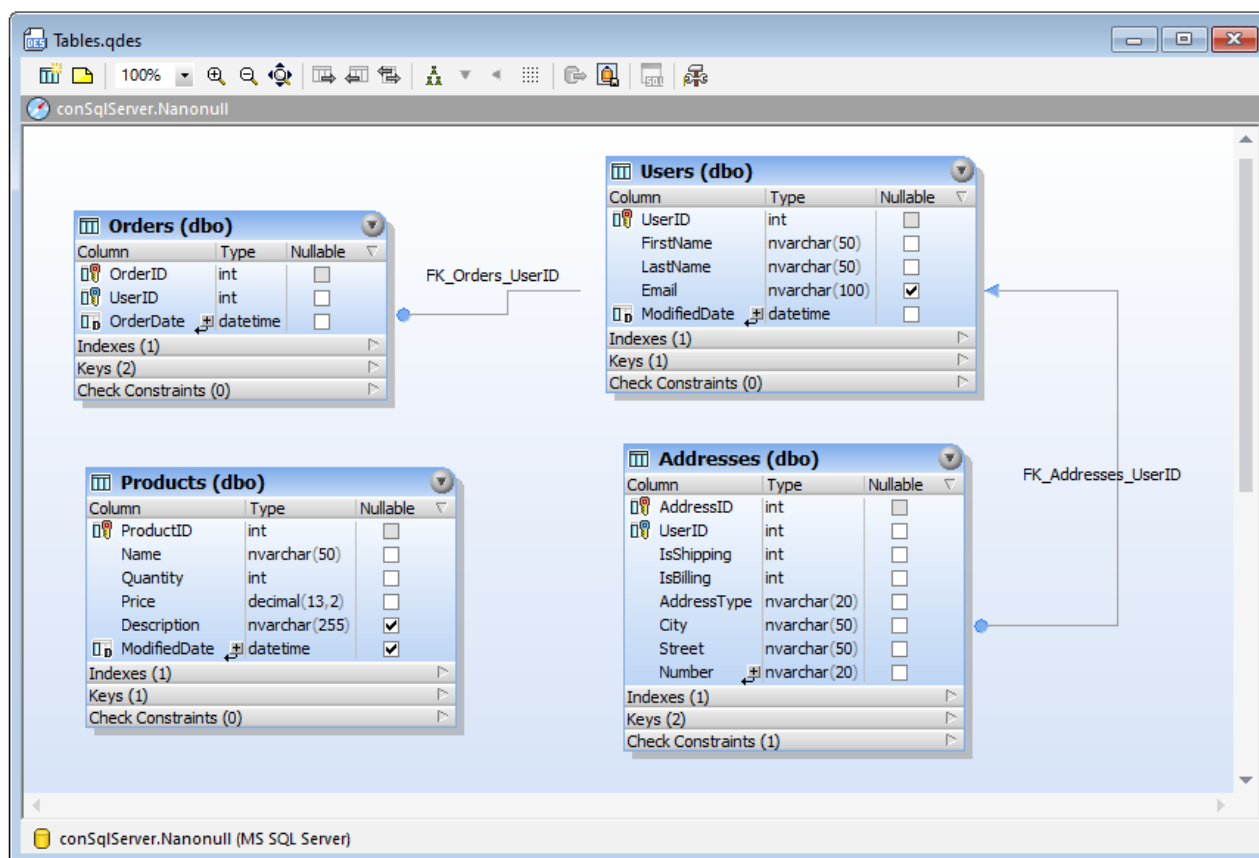
Design Editor

Designing Database Objects

You can create new database objects or change the structure of existing ones in any of the following ways, depending on your preferences:

- Visually, from the Online Browser [24]. This approach does not require writing SQL; it involves using context menu commands and editing object properties from the Properties window.
- Visually, from the Design Editor [46]. This approach also does not require writing SQL. The table structure displayed in the Design Editor window changes appearance as you make changes to it, and you can customize the level of displayed details.
- SQL-based approach, from the SQL Editor [39]. You can either generate Data Definition Language (DDL) statements from existing objects, write such statements from scratch, or perhaps run existing SQL scripts.

Note the following:

- Various database-specific limitations apply when editing existing database objects from either the Design Editor or Online Browser, as further mentioned in this documentation. When that is the case, the corresponding commands are disabled, or appropriate notification messages are displayed by DatabaseSpy.

- Any changes you make to the database structure (either from Online Browser or Design Editor) are not applied immediately but recorded in a change script which is displayed in the Database Structure Change Script [37] window. To actually deploy the changes to the database, you must execute the change script.
- In order to create or modify database objects successfully, your database user account must have the required server-level and database-level privileges.

Taking action on multiple database objects

From the Online Browser, you can take actions on multiple database objects, if permitted by the context. These actions include:

- Dropping multiple tables
- Dropping multiple columns
- Deleting data from multiple tables or views

For example, to drop multiple tables in the same operation, right-click the "Tables" folder under your database or schema and select **Drop Tables** from the context menu. Likewise, to drop several columns from the same table, first select several columns using the **Ctrl** key pressed, right-click them, and then select **Drop Column(s)** from the context menu.

Online Browser ✕

dbo
Tables
Addresses
Columns
AddressID int NOT NULL
UserID int NOT NULL
IsShippi...
IsBilling
Address
City nva...
Street n...
Number
Check Con...
Keys
Indexes
Triggers

	Show in new SQL Editor	▶
	Retrieve data	▶
⊘	Edit Data	Ctrl+Alt+E
	Show in new Design Editor	
	Delete	Delete
	Rename	F2
	Show referenced table	
	Assign XML schema...	
⭐	Add to/Remove from Favorites	Ctrl+F2
	Expand	▶
	Collapse	▶
🖘	Export database data...	
	Drop 2 Column(s)	

Project   Online Brows

The structural changes are not committed directly to the database. Instead, a change script is generated in the Database Structure Change Script Window [37], where you can review it before execution or revert it if necessary, see also Dropping Database Objects [282].

6.1   Tables

In DatabaseSpy, tables are shown with the ▥ icon. Depending on the database kind, tables may appear either under a database (catalog) 🟡 or under a schema 🟥.

In the [Online Browser](24), the hierarchical structure of the table is displayed when you expand the ▥ **[Table]** node. The "Columns", "Keys", "Check Constraints", "Indexes" (and "Triggers", if applicable) folders group together the respective objects of the same type, if they exist.
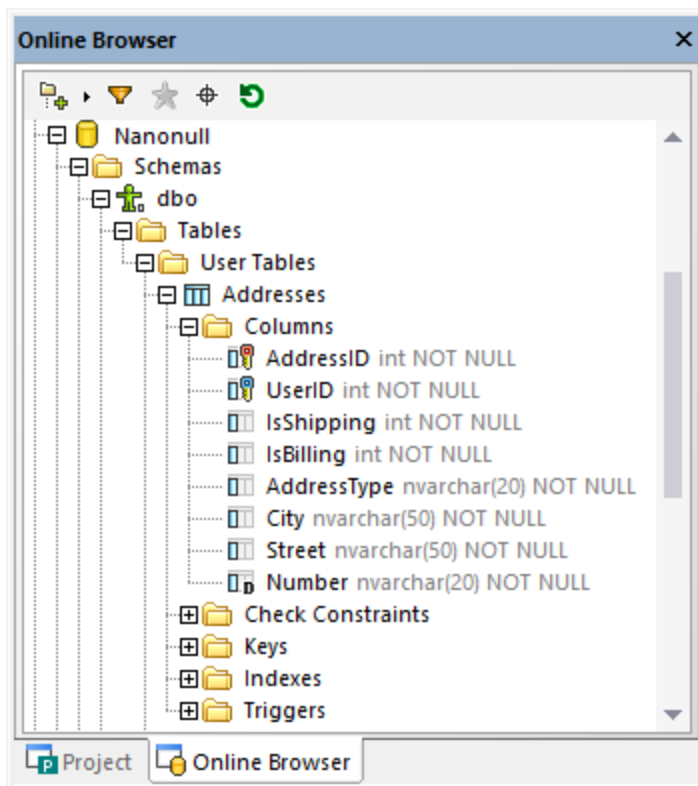
Where applicable, you can optionally group tables of a given catalog or schema into user and system tables. To toggle this option on or off, right-click the "Tables" folder and select **Sort into User and System tables** from the context menu.

As a rule of thumb, you can right-click any database object in the Online Browser in order to reveal and run commands applicable it. For example, by right-clicking a "Tables" folder which contains multiple tables, you can drop multiple tables in the same operation.

6.1.1   Opening the Design Editor

The [Design Editor](46) window enables you to view and edit tables and relationships between them as a diagram, see also [Viewing Tables as a Diagram](227).

To open a new Design Editor window:

1. Click a connected data source that is to be associated with the Design Editor window.
2. Click the **Design Editor** toolbar button, or select the menu command **File | New | Design Editor**, or press **Ctrl+D**.

> If you have more than one data source connected in your project, a new Design Editor window will be connected to the data source that was most recently selected in the Online Browser.

The currently connected data source is displayed in the Execution Target Bar [49] and in the lower-left corner of the Design Editor.

3. If necessary, change the data source of the Design Editor, as follows:

   a. Click into the Design Editor window whose properties you want to change.
   b. In the Properties window, select the data source and, if applicable, default schema from the respective drop-down lists.
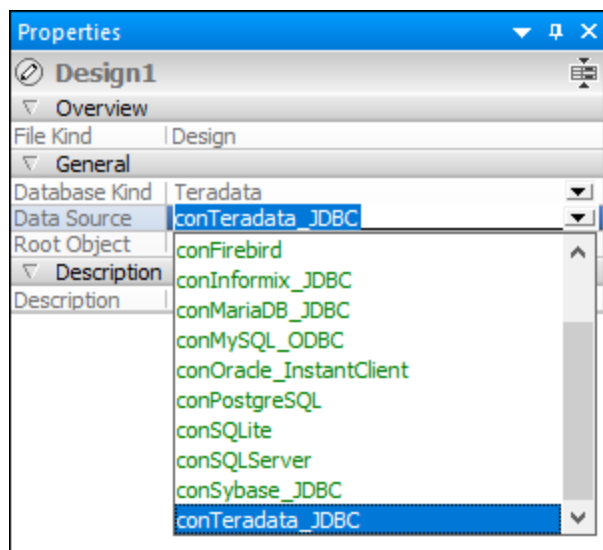
Alternatively, if the Execution Target Bar is visible in the Design Editor, click the data source or root object in the Execution Target Bar to jump directly to the corresponding field in the Properties window. Note that only connected data sources are available in the drop-down list.

6.1.2   Viewing Tables as a Diagram

To view the structure of any existing table inside a diagram in the Design Editor window, do one of the following:

- In the Online Browser [24], right-click the table you want to examine and select **Design Editor | Show in new Design Editor** from the context menu.

- If a Design Editor window is already open, you can also right-click the table and drag it into the Design Editor window or use the context menu command **Design Editor | Add to Design Editor** to add the table to the design.
- Select multiple tables in the Online Browser and drag them into Design Editor.
- Select an immediate parent folder containing database tables from the Online Browser and drag into Design Editor.

Note: Any number of tables can be selected in the Online Browser. Press **Shift+Click** to select a contiguous group, or **Ctrl+Click** to select individual tables.

To auto-size a table (after a table has been resized):

- In a Design Editor window, click the **Autosize** icon at the bottom of the table.

To remove a table from a Design Editor window, do one of the following:

- Right-click the table and select **Remove Table from Design** from the context menu.
- Select a table and press **Ctrl+Del**.
- Select a table and choose the menu option **Design Editor | Remove from Design**.

Note: Removing a table from a design does not delete it from the connected database.

Selecting tables in the Design Editor

Dragging on the Design Editor background creates a marquee which selects all the tables it encloses. The context menu of the Design Editor also contains several options in the **Select** sub-menu which allow you to select specific sets of tables.

To select only user tables, do one of the following:

- Right-click an empty area in the Design Editor window and choose **Select | Select User Tables** from the context menu.
- Click an empty area in the Design Editor window and press **Alt+T**.
- Select the menu command **Edit | Select User Tables**.

To select only system tables, do one of the following:

- Right-click an empty area in the Design Editor window and choose **Select | Select System Tables** from the context menu.
- Click an empty area in the Design Editor window and press **Alt+S**.
- Select the menu command **Edit | Select System Tables**.

To select all tables, do one of the following:

- Right-click an empty area in the Design Editor window and choose **Select | Select All** from the context menu.
- Click an empty area in the Design Editor window and press **Ctrl+A**.

- Select the menu command **Edit | Select All**.

To select several individual tables:

- Click a table and hold down the **Ctrl** or **Shift** key while selecting additional tables.

Customizing the table layout

You can define the default table layout in the [Design Editor options](#) ⁴³³; however, you can also change the layout in the Design Editor itself. Apart from the default view which shows the fields "Column Name", "Type", and "Nullable" for each column, you can choose two more options for displaying tables in the Design Editor.

The **compact view** shows only the column, index, and constraint names as well as the icons that indicate the constraints that have been defined for a column or the index or constraint type, respectively.
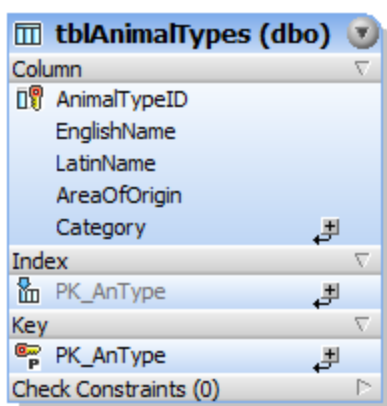
Compact view

The **collapsed view** shows only the table header.
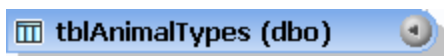
Collapsed view

This view is particularly useful if you want to view all the tables and relations of the entire database and do not need detailed information about the individual tables.

To view a table in the compact view:

- Right-click the table in a Design Editor window and activate **Compact View** in the context menu.

To switch off compact view for a table:

- Right-click the table and deactivate **Compact View** in the context menu.

To collapse or expand a table, do one of the following:

- Click the arrow in the upper right corner of a table to toggle between collapsed and expanded view.
- If you want to expand or collapse several tables at a time, select the tables using **Ctrl+Click** and click

    Expand selected Tables [▼] or **Collapse selected Tables** [◄], respectively, in the Design Editor
    toolbar.

6.1.3   Viewing Table Relationships

When the Design Editor contains two or more tables that are related with each other, the relationships between these tables are shown with lines connecting the related tables. DatabaseSpy uses different colors to distinguish the individual relationships:

- **Green**: Self-reference, i.e. the table uses a key that is assigned to one or more of the table's own columns.
- **Blue**: Incoming reference, i.e. the selected table is referenced by a key used in another table.
- **Orange**: Outgoing reference, i.e. the table has a key defined that references one or several column(s) in another table.

Note that the color of one and the same relation line may change depending on the table that is currently selected. For example, a foreign key relationship is displayed blue if you select the referenced table, and orange if you select the referencing table. The screenshot below shows an incoming relationship for `tblAnimalCategories`.
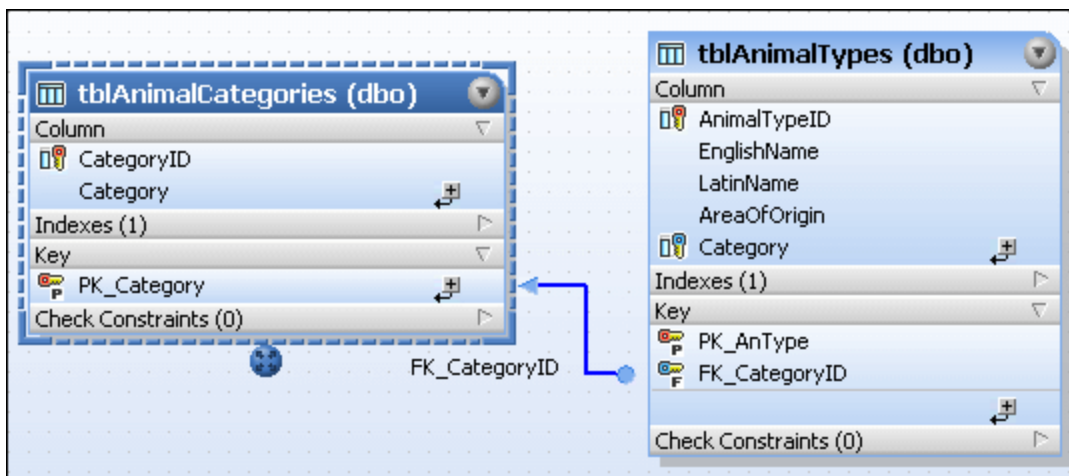
The same relationship, however, viewed from the referencing `tblAnimalTypes` table's point of view, is depicted with an orange relation line.
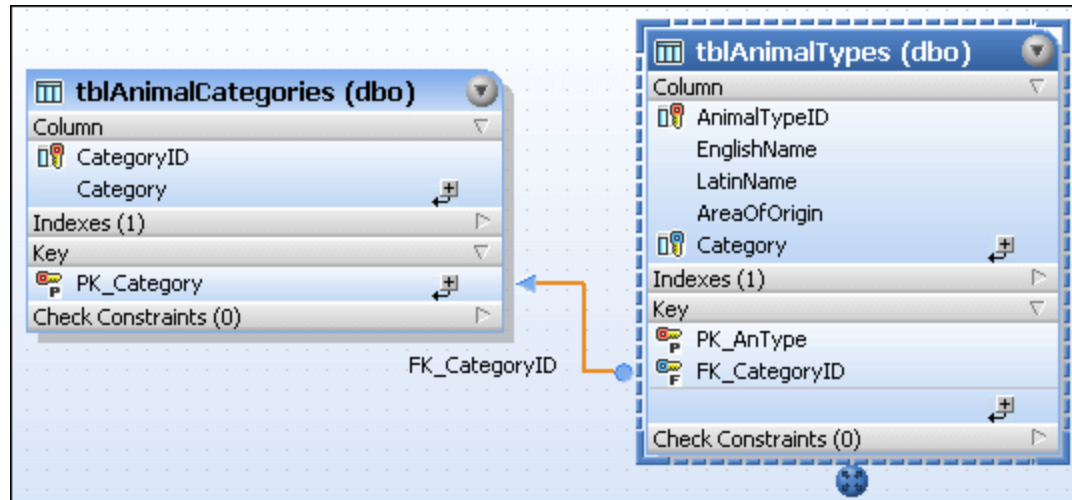
The screenshot below illustrates a self relation that makes sure that the manager number ("MgrNo") of an employee's manager is contained in the list of employee numbers ("EmpNo").
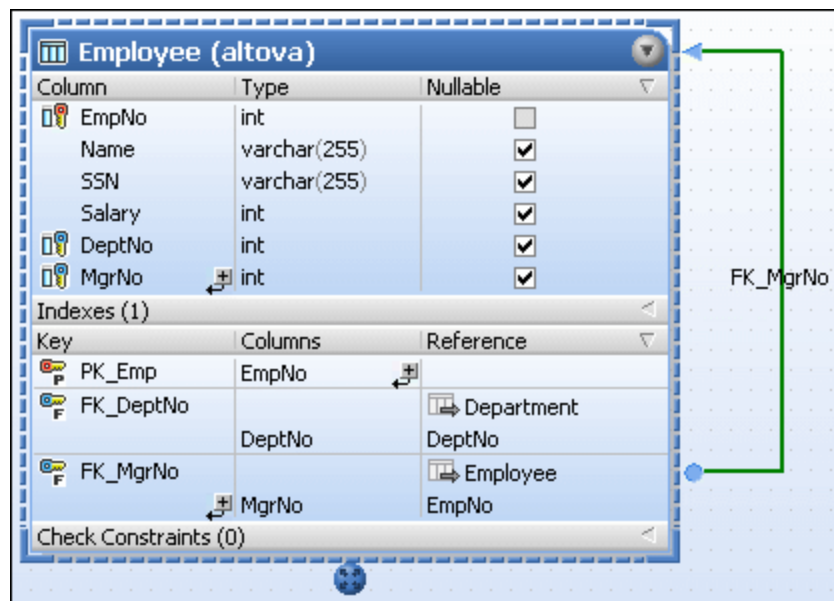
Finding related tables

DatabaseSpy provides an option that automatically adds referenced or referencing tables to the active design.
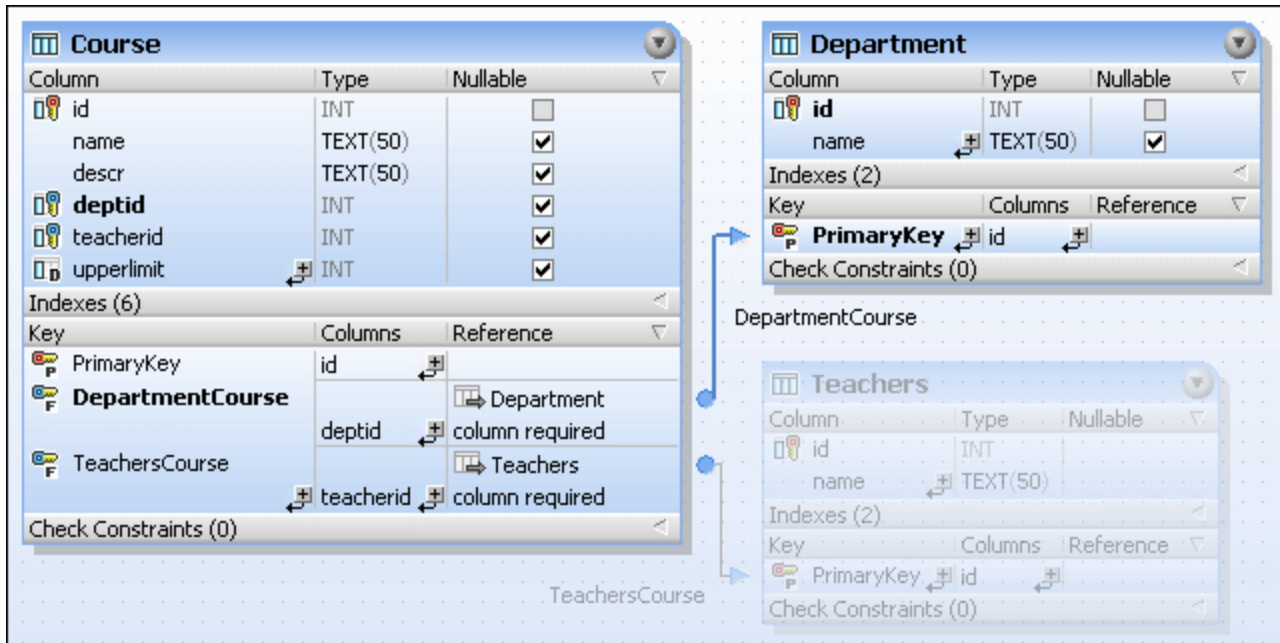
You can also define to highlight objects that form part of the currently selected relation in the Design Editor options [433]. In this case, as depicted in the screenshot above, tables that are not part of the selected relation are grayed out. Also note that the referenced columns are printed bold in both tables so that you can view at a glance which columns are related with each other.

To display related tables in the Design Editor:

1. Right-click a table in the Design Editor and select **Add Related tables** from the context menu.
2. Choose one of the following options from the sub-menu:

    - Referenced Tables
    - Referencing Tables
    - All Relations

To highlight objects of the currently selected relation:

1. Make sure that the **Highlight active relations objects check box** is activated in the Design Editor options [433].
2. Click on a connection line. The connection line is highlighted and all tables that are not part of this relation are grayed out in the Design Editor window.

See also Foreign Keys [250].

6.1.4   Opening, Saving and Printing Diagrams

The contents of the Design Editor can be saved for later use as a design file, exported as an image, or sent to a printer. DatabaseSpy design files have the .qdes extension.

Saving design files

DatabaseSpy provides several options for saving designs in the Design Editor:

- **Design file (*.qdes)**: These files can only be opened in DatabaseSpy
- **Image (*.png)**: Designs that have been saved as image can be opened with any picture viewer.

To save a design as .qdes design file:

1. Make the Design Editor window that you want to save the active window.
2. Do one of the following:

   - Right-click the tab of the Design Editor window and choose **Save** or **Save As...** if you want to save a design file under a new name or path.
   - Press **Ctrl+S**.
   - Click the **Save** button in the Standard toolbar.

To save a design as image:

1. If you have maximized the Design Editor window, click the **Restore** button in the upper right corner of the Design Editor.
2. Resize the window and arrange the table(s) as desired. The borders of the window define the white space that surrounds the image.
3. To save the image, do one of the following:

   - Click the **Save Diagram as Image** icon in the toolbar of the Design Editor window.
   - Choose the menu option **Design Editor | Save Diagram as Image**.

   The standard Windows **Save As** dialog box appears.

4. Enter name and path for the image.
5. In the **Save as type** drop-own list, select "PNG-files". Note: PNG is currently the only supported format.
6. Click **Save**.

Opening design files

To open an existing design file, do one of the following:

- Select the menu option **File | Open | Open File**..., or press **Ctrl+O**, and browse for the design file.
- Double-click the design file in the Project window.

Note: If the design file has a data source which is not currently connected in your project, you are prompted to connect to the data source first. Likewise, if you attempt to open a design whose data source does not exist in the current project, you are prompted to add a connection first.

Printing Designs

Designs in the Design Editor can also be printed. You can print out an entire design or select a number of objects to print only the selection. A print preview is available for you to check the layout of the printed page before sending it to the printer.

To print a design file:

1. Click in the Design Editor window to make it active.
2. Select the menu option **File | Print...**, or press **Ctrl+P**, or click the **Print** icon in the Standard toolbar. The **Print** dialog box appears.
3. In the What group box, define whether to print the whole diagram or only the selected objects.
4. Choose the appropriate zoom factor in the Zoom group box. The Use optimal option ensures that the design is printed on one page.
5. Optionally, define the printer setup by clicking the **Print Setup** button.
6. Optionally, click the **Preview** button to preview the print-out before actually sending it to the printer.
7. Click **Print**.

To preview a design file before printing:

1. Click in the Design Editor window to make it active.
2. Select the menu option **File | Print Preview**, or press **Ctrl+P**, or click the **Print** icon in the Standard toolbar. The **Print** dialog box appears.
3. In the **Print** dialog box, click **Preview**.

6.1.5   Creating Tables

After connecting to a data source, you can create a new table in DatabaseSpy in several ways.

From the Online Browser

The following instructions illustrate how to create a minimal table having two columns. The first column is auto-incremented and has a primary key. The second column is nullable. The instructions are for a SQLite database but are similar in other database kinds.

1. Right-click the "Tables" folder and select **Create Table** from the context menu.

> The "Tables" folder may appear either under a schema or a database, depending on the database kind.

The new table appears in the Online Browser with the default name "newtable", and a table creation script is generated in the Database Structure Change Script Window [37]. Also, a new column called "column" is created by default for the table.
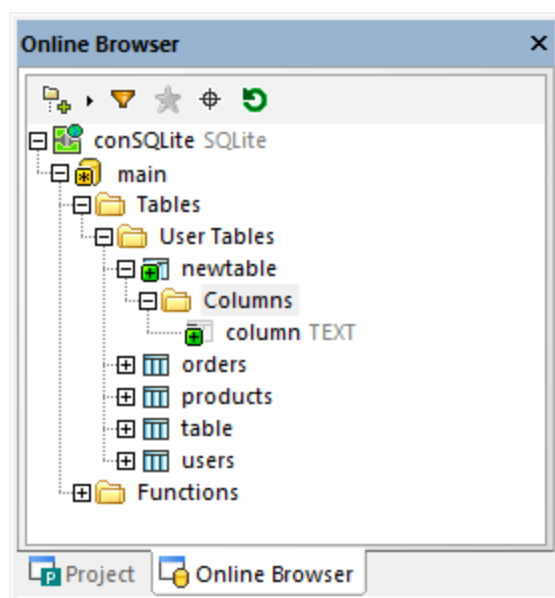
2. Select the new table and press F2 to rename it to "assets". (Alternatively, right-click the table and select **Rename** from the context menu).
3. Select the new column and press F2 to rename it to "id".
4. Select the column and make sure that the following properties are as follows in the Properties window:

   - Data Type: **INTEGER**
   - Nullable: **Off**

   > - The **Nullable** property must be switched off. This makes it possible to add a primary key to the table (see the next step).
   > - The list of properties varies depending on the database kind.

5. Right-click the column and select **Create Primary Key** from the context menu.
6. Right-click the table or the "Columns" folder and select **Create Column** from the context menu. Using the same approach as above, rename the new column "description". Make sure that the **Nullable** property in the Properties window is on.
7. Optionally, to change the order of columns while the changes are not committed to the database, drag a column up or down in the list of columns and drop it at the desired position.
8. Click the **Execute Change Script** ▷ button in the Database Structure Change Script window.

From the Design Editor

1. Open a Design Editor window (**Ctrl+D**).
2. Click the **New Table** ⊞ button in the Design Editor toolbar (or select the menu command **Design Editor | Create New Table**), and add the required table objects. For a step-by-step example, see Adding Tables Using Design Editor [82].
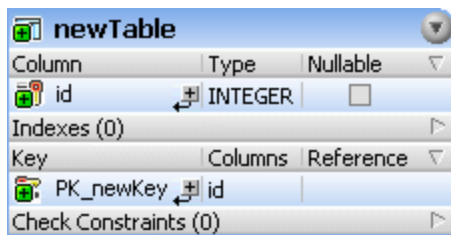
Note: By default, DatabaseSpy creates automatically the primary key for new tables in Design Editor. To disable this option, select the menu command **Tools | Options | Design Editor**, and then clear the option **Automatically create primary key for new tables**.

3. Using the same steps as above, add as many new columns as necessary. If you need to reorder columns while changes are not committed to the database, right-click a column and select **Move Column Up** or **Move Column Down** from the context menu. Alternatively, drag a column and drop it to a new position in the table.

4. Click the **Execute Change Script** button in the Database Structure Change Script window.

By generating SQL

This approach requires that the database already contains a table that you can use as template for the new table.

1. Right-click an existing table in the Online Browser and select **Show in new SQL Editor | Create** from the context menu. A CREATE statement appears in the SQL Editor.
2. Modify the generated SQL as required (at the minimum, you should rename the table to ensure the successful execution of the script).
3. Click the **Execute** button at the top of SQL Editor (or press **F5**). The outcome is reported in the Message window.

You can also generate the CREATE script from a different database kind. In this case, however, you will need to edit the SQL statement so that it is compatible with the target database. For example, let's assume that you need to create a new table in a SQLite database that is very similar to an existing table from an Access database. In this case, Access is the "source", and SQLite is the "target". You can duplicate the table as follows:

1. Connect to both databases.
2. In the Online Browser, right-click a table from the "source" database and select **Show in new SQL Editor | Create** from the context menu.
3. Click the Execution Target Bar [49] at the top of SQL Editor.
4. From the Properties window, change the **Data Source** property to point to the target database.
5. Edit the CREATE statement so that it's compatible with the target database.
4. Click the **Execute** button at the top of SQL Editor (or press **F5**). The outcome is reported in the Message window.

For a slightly different approach that performs automatic conversion to the new SQL grammar, see Migrate Table Structure [237] .

By writing SQL

Open a new SQL Editor window (**Ctrl+N**), enter a CREATE statement in the grammar of the currently connected database, and click **Execute** ▷. The code listing below is applicable for SQL Server databases.

```
CREATE TABLE newTable (
    id INT NOT NULL,
    CONSTRAINT PK_newKey PRIMARY KEY (id)
)
```

You must define at least one column for the new table.

6.1.6 Migrating Table Structure

You can convert the structure of a table so as to fit the syntax requirements of a database kind other than the table was initially created in. This could be useful when you need to migrate tables that contain a very large number of fields to a different database. Note that the following limitations apply:

- Properties that would require parsing (for example, check constraints or triggers) will not be converted
- Foreign keys will not be converted
- Indexes will not be converted

If database items cannot be converted into the target database, a warning message will be displayed in DatabaseSpy.

To convert a table:

1. Open a Design Editor window [226].
2. Optionally, display the source table in a Design Editor window of the "source" database.
3. Select a table in the Online Browser or Design Editor of the "source" database and drag it into the Design Editor window of the "target" database. A table creation script is generated in the Database Structure Change Script Window [37].
4. Click the **Execute Change Script** ▷ button in the Database Structure Change Script window.

6.1.7 Renaming Tables

See Renaming Database Objects [279].

6.1.8 Dropping Tables

You can drop either single tables or multiple tables in batch, as follows:

1. Do one of the following:

- To drop a single table, right-click it in the Online Browser and select **Drop Table** from the context menu.
- To drop multiple tables, right-click the "Tables" (or "User Tables", if applicable) folder, and select **Drop tables** from the context menu.

A change script is generated in the [Database Structure Change Script Window](#)⁽³⁷⁾, and the tables are marked for deletion, which is indicated by the ✖ icon. At this stage, you can still revert the changes by clicking the **Refresh** ↻ button in the Online Browser.

2.   Click **Execute Change Script** ▷ to commit the changes to the database.

See also [Dropping Database Objects](#) [282].

6.2   Columns

In DatabaseSpy, columns are shown with the ▯▯ icon.
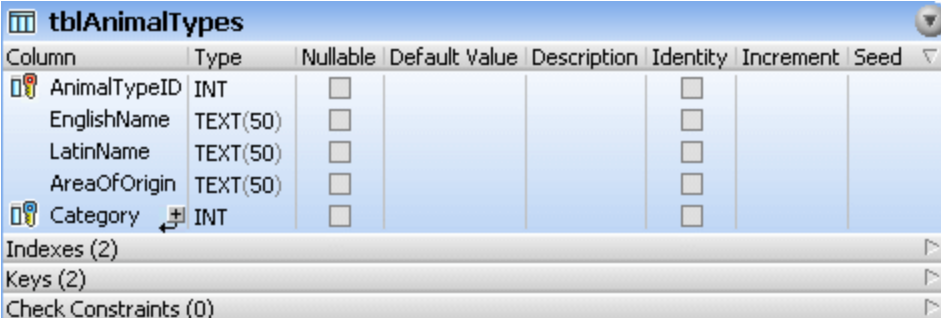
Columns that have a special quality, such as, for example, forming part of a relationship, may have an icon in front of their name. If a column has more than one constraint assigned, only the constraint with the highest priority is depicted in the column icon. The priority of the constraints is as follows:

▯🔑	**Primary key column**	This column is used as the table's primary key.
▯🔑	**Unique column**	This column has a unique constraint defined and therefore accepts only data that is unique in the context of the column.
▯🔑	**Referencing column**	This column has a foreign key constraint assigned and references the primary key of a different table.
▯▯ XML	**XML column**	The data contained in this column are of type XML.
▯ᴅ	**Default value**	A default constraint has been defined for this column.

You can view (or change, if applicable) a column's properties from the Properties window, after selecting the column in the Online Browser.

Displaying additional information in Design Editor

When you right-click the header of the table design's Column section, you can select a number of additional properties (Default Value, Description, Identity, Increment, and Seed) to be displayed in the design. Select the properties you want to display or select **Show all additional columns** from the context menu.

Column	Type	Nullable	Default Value	Description	Identity	Increment	Seed ▽
🔑 AnimalTypeID	INT	☐			☐		
EnglishName	TEXT(50)	☐			☐		
LatinName	TEXT(50)	☐			☐		
AreaOfOrigin	TEXT(50)	☐			☐		
🔑 Category	INT	☐			☐		
Indexes (2)							▷
Keys (2)							▷
Check Constraints (0)							▷

6.2.1   Creating Columns

You can add columns to an existing table using any of the approaches described below.

From the Online Browser

1. Right-click a table (or the "Columns" folder of a table) and select **Create Column** from the context menu. The new column appears in the Online Browser with the default name "column", and a structure change script is generated in the Database Structure Change Script Window [37].
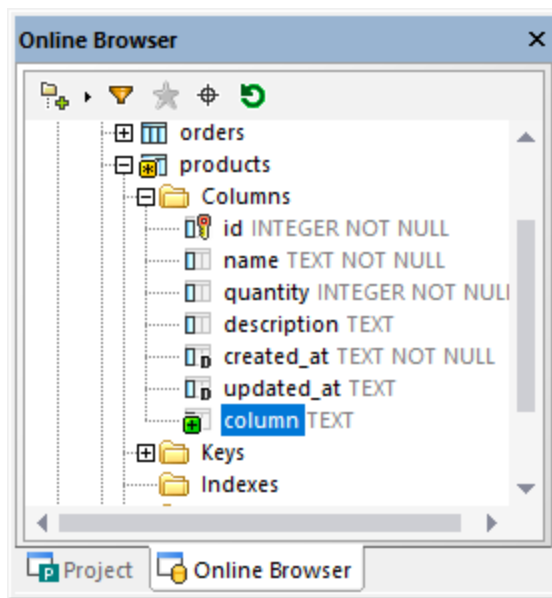
2. Select the new column and press **F2** to rename it. Alternatively, right-click the column and select **Rename** from the context menu.
3. Set the column's properties (such as **Type** or **Nullable**) from the Properties Window [30]. Note that the available column properties vary by database kind. If the table is empty or new (not yet committed to the database), the new column is NOT NULL by default.
4. Click the **Execute Change Script** ▷ button in the Database Structure Change Script window.

From the Design Editor

1. On the **File** menu, click **New | Design Editor**, or press **Ctrl+D**.
2. Do one of the following:

   - Select the table and run the menu command **Design Editor | Create New Column**.
   - Press **Alt+C**.
   - Right-click the table and select **Insert new | Column** from the context menu.
   - Click the plus ⊞ icon to the right of the last column name.

   The column is inserted into the table design and a New 🔲 icon appears to the left of the column name.

3. Enter a descriptive name for the column and press **Enter**.
4. Modify the column properties as required.

5. Click the **Execute Change Script** ▷ button in the Database Structure Change Script window.

To duplicate columns from other tables:

1. Select the column to be duplicated in the Online Browser or in the Design Editor and, keeping the mouse button pressed, drag it into the Columns section of the table design where you wish to add the column.
2. Edit the column properties and/or the column name.
3. Click the **Execute Change Script** ▷ button in the Database Structure Change Script window.

By generating SQL

1. In the Online Browser, right-click the column that serves as a model for the new column to be created.
2. Select the menu command **Show in new SQL Editor | Add** from the context menu.
3. In the SQL Editor, change the column name and edit the column definition, if required.
4. Click the **Execute** ▷ button or press **F5**.

6.2.2   Creating Identifier Columns

Some databases provide a special column kind which is known as "identifier", or "auto-increment" column. When a column is defined as an identity column, the database will automatically generate a unique, incremental value for this column whenever a new row is inserted into the table. Support for identifier columns varies by database.

Note: The Design Editor does not support defining identifier columns for the following database types: Firebird, Informix, Oracle, PostgreSQL, Progress OpenEdge, Teradata, and SQLite.

When supported by the database, a property called **Auto Increment** (or **Identity**, in some databases) becomes available in the Properties window when you click the column in the Design Editor (provided that the column is of numeric type, see "Prerequisites" below). Databases such as IBM DB2 and SQL Server also let you specify the **Seed** and **Increment** properties, in addition to the **Identity** property.
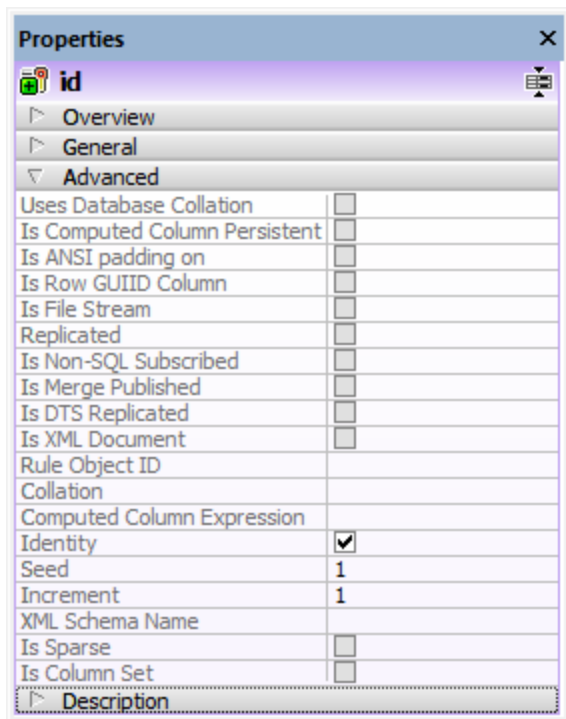
Properties window (SQL Server database)

Prerequisites:

- Typically, you can create identity columns at the table creation time, not later. Some databases (Microsoft Access, Microsoft SQL Server) let you specify a column as identity when you add a new column to an existing table.
- The column must not be nullable. In other words, the **Nullable** check box must not be selected in the Properties window (or in the table design).
- The column must have a numeric type (such as INT, INTEGER, SMALLINT, BIGINT, or even DECIMAL with a scale of 0). Other numeric types may qualify, depending on the database.

To create an identity column:

1. In the Design Editor or in the Online Browser, select the column you want to use an as identity column.
2. In the Properties Window, select the **Identity** check box (or the **Auto Increment** check box, depending on the database).

Note: In Microsoft Access, the column data type will be automatically converted to COUNTER when you select the **Auto Increment** check box. This data type is specific to Microsoft Access, it ensures that the value will be auto-incremented.

3. If applicable, change the default values in the **Seed** and **Increment** fields. **Seed** specifies the initial value that will be generated by the database while **Increment** specifies by how much the initial value should be incremented on each insert. Normally, these values are **1** and **1**, respectively.

As always when making changes to the database structure, a change script is generated which must be executed in order to actually create the identifier column in the database. DatabaseSpy performs a validation of the column definition that will be used for the identifier column. The image below illustrates a script where validation has failed because the column is nullable and not of numeric type:

```
Database Structure Change Script                                                    ✕

  ▷  ⬚SQL  ⬚  ⬚  ✕
  1      ⊟-- region creating tables
  2       -- Identity column 'id' must be of data type int, bigint,
          smallint, tinyint, or decimal or numeric with a scale of 0, and
          constrained to be nonnullable.
  3       -- CREATE TABLE [ZooDB].[dbo].[table] ( [id] varchar (255)
          IDENTITY (1, 1) NOT NULL )
  4      └-- endregion creating tables
  5      ⊟-- region adding primary keys
  6       ALTER TABLE [ZooDB].[dbo].[table] ADD
  7             PRIMARY KEY ([id]) ;
  8      └-- endregion adding primary keys
  9
  ZooDB

  ⇨ Output  ⬚SQL Database Structure Change Script
```

A valid script, on the other hand, could look like:

```
Database Structure Change Script                                                    ✕

  ▷  ⬚SQL  ⬚  ⬚  ✕
  1      ⊟-- region creating tables
  2       CREATE TABLE
  3            [ZooDB].[dbo].[table] (
  4                [id] int IDENTITY (1, 1) NOT NULL,
  5                [column] varchar (255) NOT NULL) ;
  6      └-- endregion creating tables
  7      ⊟-- region adding primary keys
  8       ALTER TABLE [ZooDB].[dbo].[table] ADD
  9             PRIMARY KEY ([id]) ;
 10      └-- endregion adding primary keys
 11
  ZooDB

  ⇨ Output  ⬚SQL Database Structure Change Script
```
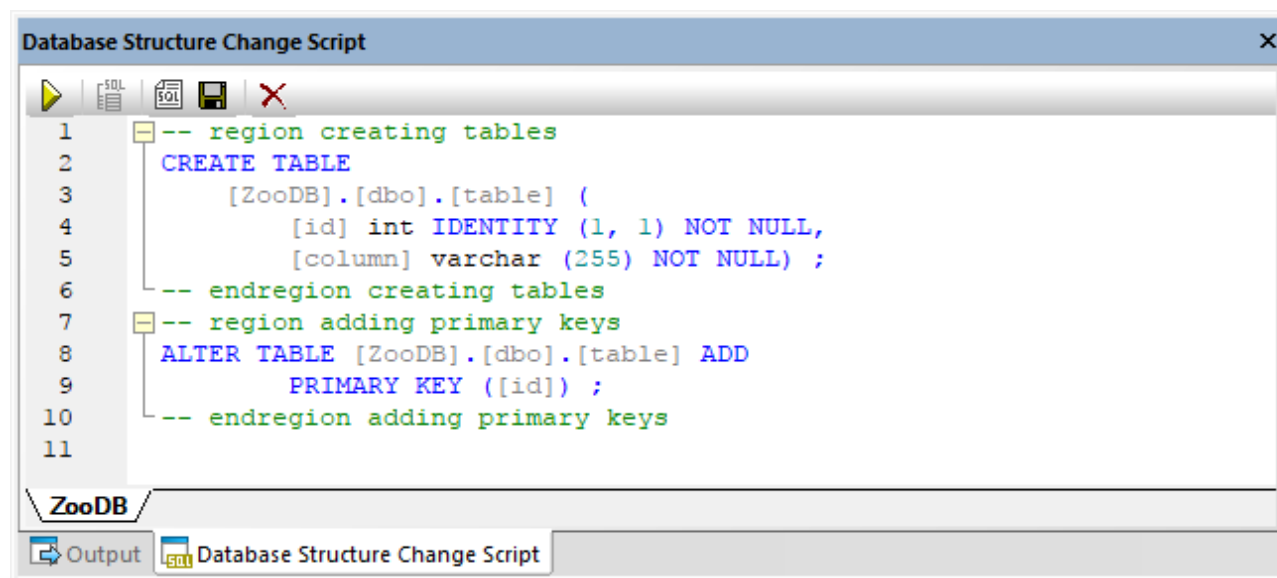
If the script is valid and you wish to commit the changes to the database, click **Execute** ▷ in the Database Structure Change Script window.

6.2.3   Renaming Columns

See Renaming Database Objects [279].

6.2.4   Dropping Columns

See Dropping Database Objects [282].

6.3   Primary Keys

In DatabaseSpy, columns that have a primary key assigned are shown with the primary key 🔑 icon. The primary key itself has the 🔑 icon.

In the Online Browser, primary keys appear in the "Keys" folder of a table. Expand the primary key to show the column (or multiple columns, if applicable) on which it is defined.

In the Design Editor, primary keys appear in the "Keys" section of a table design. The column to which the primary key is assigned is displayed in bold when you select the primary key in the "Keys" section.
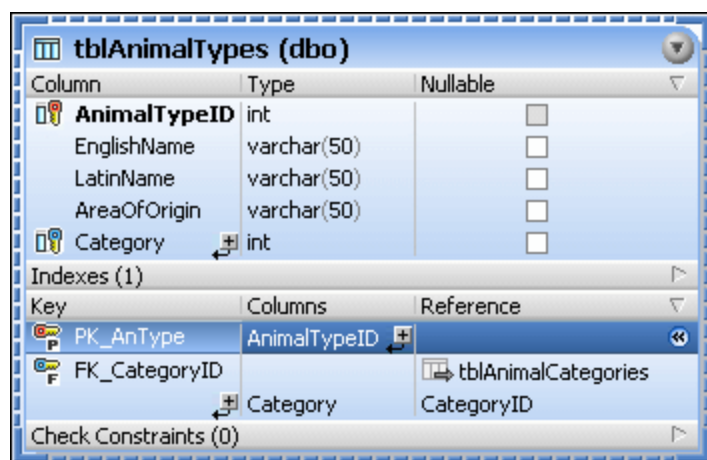
6.3.1   Creating Primary Keys

As with other database object types, you can create primary keys using several approaches. Before adding a primary key, note the following prerequisites:

1. You can add a primary key to a table only if the table does not already have one.
2. The column on which the primary key is defined must not be nullable (that is, the **Nullable** property must be turned off in the Properties window).

While a new primary key is created but not committed to the database yet, its name may appear with the label **<generated>** in Online Browser and Design Editor. This means that the name will be automatically generated by the database when you commit the change script. For details about support for generated names across various databases, see [Generated Names](#)[277].

Note the following:

•   In MySQL and MariaDB databases, the name of the primary key constraint is always PRIMARY.
•   In SQLite, the primary key displayed in DatabaseSpy appears with the name **<generated>** and you can rename it. However, after committing the changes, the primary key name will still be displayed with an arbitrary name, due to database limitations.

Creating primary keys from the Online Browser

To add a primary key on a single column:

1. Right-click a column and select **Create Primary Key** from the context menu.
2. Click the **Execute Change Script** ▷ button in the Database Structure Change Script window.

If you need to create a primary key on multiple columns, see [Composite Primary Keys](#) 247 .

Creating primary keys from the Design Editor

By default, when you create a new table in DatabaseSpy's Design Editor, a primary key is automatically defined on the first column. This option can, however, be disabled in the [Design Editor Options](#) 433 , and in this case you can define a primary key manually.

There are several ways to define a primary key in the Design Editor:

*   Right-a click a column in a table design and select **Make Primary key** from the context menu.
*   Drag a column from the "Column" section and drop it onto the "Key" section of a table design. To create a primary key on multiple columns, first select multiple columns while holding the **Shift** key pressed, and then drag them into the Keys section. This menu is also displayed if you click the **Create new Key Constraint** ⊞ icon in the Key section of the table design.
*   Right-click anywhere into the table design and select **Insert new | Key | Primary Key** from the context menu. Alternatively, expand the Keys section, click the **Create new Key Constraint** ⊞ icon at the bottom of the Key column and select **Create Primary Key** from the menu that pops up.

The primary key is added to the "Key" section of the table design.
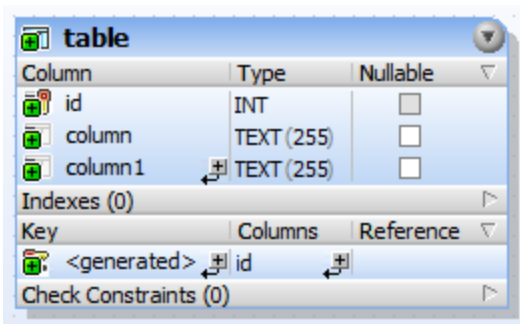
To add the primary key on multiple columns, click the **Add Column to Key** ⊞ button (at the bottom of the "Columns" column) and choose a column from the drop-down list that opens. To remove a column from a primary key, right-click the column (in the "Columns" column), and then select **Delete selected objects** from the context menu.

As always when making changes to the database structure, a change script is generated which must be executed in order to actually create the primary key in the database.

Creating primary keys by generating SQL

1. In the Online Browser, right-click the primary key that serves as template for the primary key that is to be created.
2. Select the menu command **Show in new SQL Editor | Add** from the context menu.
3. In the SQL Editor, change the generated statement as required.
4. Click the **Execute** ▷ button or press **F5**.

6.3.2   Composite Primary Keys

A primary key may include multiple columns, in which case it is known as a "composite" primary key.

To create a composite primary key:

1. In the Online Browser, right-click the first column and select **Create Primary Key** from the context menu.
2. Right-click another column and select **Add Column to Primary Key** from the context menu. If necessary, repeat this step for each column that needs to be added to the primary key.
3. Click the **Execute Change Script** ▷ button in the Database Structure Change Script window.

To remove a column from a composite primary key:

1. In the Online Browser, expand the primary key 🔑 in the "Keys" folder of a table.
2. Right-click the column that is to be removed from the primary key, and select **Remove Column from Key** from the context menu.
3. Click the **Execute Change Script** ▷ button in the Database Structure Change Script window.

6.3.3   Renaming Primary Keys

See Renaming Database Objects [279].

6.3.4   Dropping Primary Keys

See Dropping Database Objects [282].

6.4   Unique Keys

In DatabaseSpy, columns that have a unique key assigned are shown with the ▯🔑 icon.

In the Online Browser, the name of the unique key constraint is shown in the "Keys" folder of the respective table and will be marked with the unique key 🔑ᵤ icon. The only possible child of a unique key is the column ▯🔑 on which the key is defined.
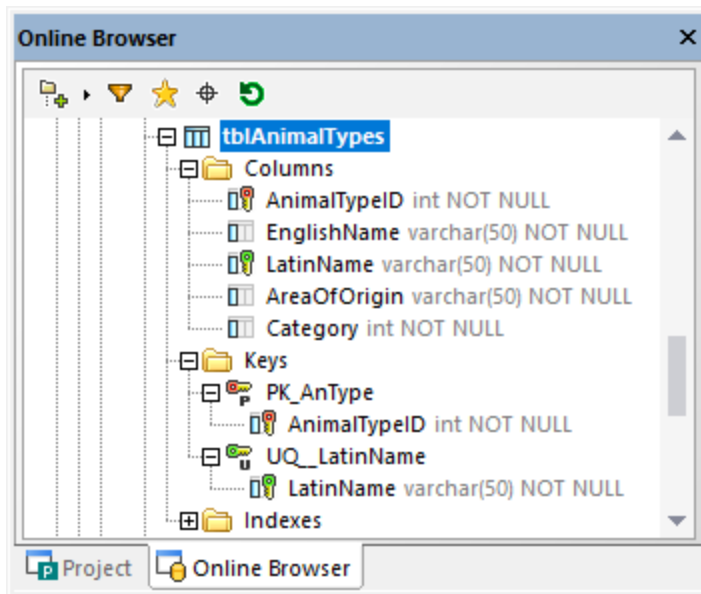
In the Design Editor, the column to which the unique key is assigned is displayed in bold when you select the unique key constraint in the "Keys" section.
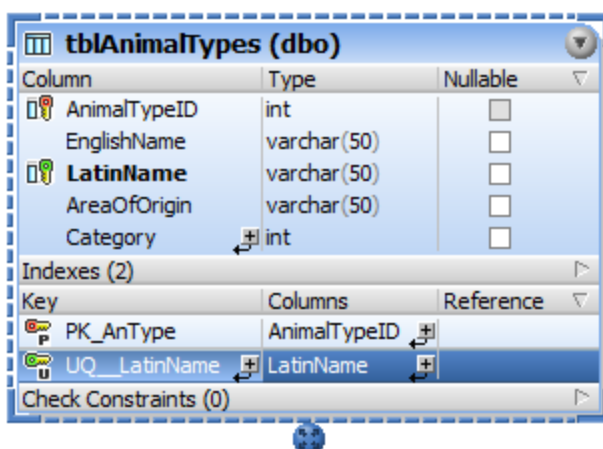
6.4.1   Creating Unique Keys

As with other database object types, you can create unique keys using several approaches. Before adding a unique key to a column, note the following:

- If the unique key constraint is applied to an existing column, this column must not contain duplicate values.
- The column may be nullable; however, the NULL value can occur only once.

While a new unique key is created but not committed to the database yet, its name may appear with the label **<generated>** in Online Browser and Design Editor. This means that the name will be automatically generated by the database when you commit the change script. For details about support for generated names across various databases, see Generated Names [277].

Adding a unique key from the Online Browser
1. Right-click a column and select **Create Unique Key** from the context menu.
2. Click the **Execute Change Script** ▷ button in the Database Structure Change Script window.

Adding a unique key from the Design Editor
There are several ways to define a unique key in the Design Editor:

- Select the **Make Unique key** command from the context menu of a column in a table design
- Drag a column from the Columns section and drop it onto the Keys section of a table design
- Click the **Create new Key Constraint** 🔲 icon in the Keys section of a table design
- Right-click anywhere into the table design and select **Insert new | Key | Unique Key** from the context menu

As always when making changes to the database structure, a change script is generated which must be executed in order to actually create the unique key in the database. DatabaseSpy performs a check as to whether the content in the constrained column is unique. If the check fails, a dialog box opens displaying a "duplicate key was found" error. The exact contents of the error message depends on the database type.

Creating unique keys by generating SQL
1. In the Online Browser, right-click the unique key that serves as template for the unique key that is to be created.
2. Select the menu command **Show in new SQL Editor | Add** from the context menu.
3. In the SQL Editor, change the generated statement as required.
4. Click the **Execute** ▷ button or press **F5**.

6.4.2   Dropping Unique Keys

See Dropping Database Objects [282].

6.5   Foreign Keys

In DatabaseSpy, columns that have a foreign key assigned are shown with the 🔑 icon. The foreign keys themselves are shown with the 🔑 icon.

For any table that has references to other tables by means of foreign keys, you can identify and select the referenced tables in the Online Browser, as follows:

1.   In the Online Browser, right-click a foreign key in the "Keys" folder of a table.
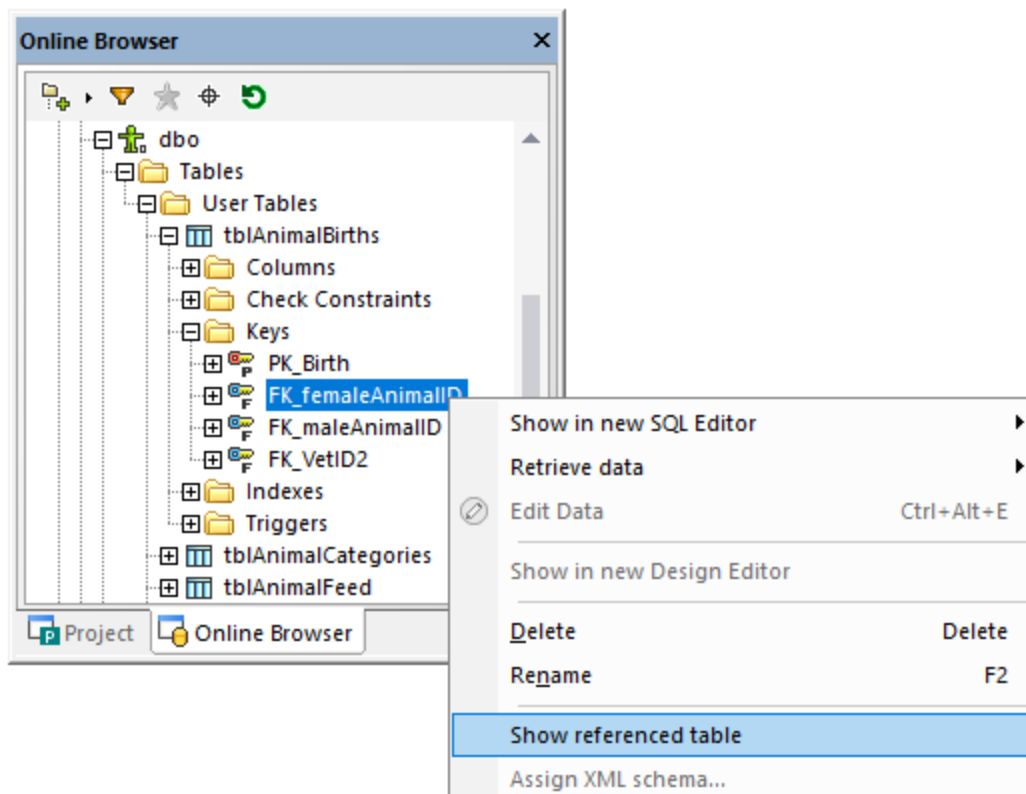
2.   Select **Show referenced table** from the context menu. The foreign key as well as the table it references are selected in the Online Browser.

Example

Foreign key constraints help preserve the integrity of the database. For example, let's assume that you are exploring a database that stores employee data. The employee details are stored in the `Employee` table and the department details are stored in the `Department` table.

```
+----+------------------+---------------------------+---------------+
| ID | Name             | Email                     | DepartmentID  +
+----+------------------+---------------------------+---------------+
| 1  | Toby Hughey      | t.hughey@nanonull.com     | 1             |
| 2  | Mia Dahill       | m.dahill@nanonull.com     | 2             |
| 3  | Fred Weinstein   | f.weinstein@nanonull.com  | 1             |
```

```
+----+-----------------+----------------------------+----------------+
```

The Employee table

```
+----+-------------+
| ID | Name        |
+----+-------------+
| 1  | Development |
| 2  | Marketing   |
+----+-------------+
```

The Department table

Note that the `DepartmentID` in the `Employee` table points to the `ID` of the department in the `Department` table. This relationship can be enforced through a foreign key constraint between these two columns, so as to ensure that:

- Each record in the `Employee` table can reference any of the existing departments in the `Department` table.
- If you ever attempt to add a new employee to the `Employee` table with a `DepartmentID` that does not exist in the `Department` table, the database would return a legitimate validation error (for example, "Foreign key constraint failed").
- If you ever attempt to delete a record from the `Department` table, and if there are `Employee` records pointing to that `Department`, the database would return a legitimate validation error.

6.5.1   Creating Foreign Keys

As with other database objects, there are several ways to create foreign key constraints:

- From the Online Browser
- From the Design Editor
- By generating SQL from an existing foreign key object

Prerequisites

When creating foreign keys, note the following prerequisites:

- The columns that take part in the foreign key relationship must be of the same data type and length.
- The referencing column (`Department.ID`, in this example) must be part of either a primary key or unique key.

Also, note the following database-specific behavior:

- In SQL Server, you can add a foreign key reference to a column not only if it's part of a primary or unique key but also if there is a unique index defined on it. This is also possible in newer versions of MySQL (provided that the storage engine is InnoDB).
- SQLite does not support adding foreign keys to *existing* tables. You can create foreign key constraints only at table creation time.

Preparing the database

The instructions below present various ways to create a foreign key relationship between two tables. We will use the tables discussed in the previous [example](#)[250], `Employee` and `Department`, and a SQL Server database. The instructions are similar in other databases. If you want to follow the instructions below literally, you can prepare your SQL Server database by running the following script:

```
CREATE TABLE
    Department (
        [ID] int IDENTITY (1, 1) NOT NULL,
        [Name] varchar (100) NOT NULL,
        CONSTRAINT [PK__Department] PRIMARY KEY ([ID]) ) ;

CREATE TABLE
    Employee (
        [ID] int IDENTITY (1, 1) NOT NULL,
        [Name] varchar (50) NOT NULL,
        [Email] varchar (100) NOT NULL,
        [DepartmentID] int NOT NULL,
        CONSTRAINT [PK__Employee] PRIMARY KEY ([ID]) ) ;
```

Table creation scripts (SQL Server)

In the SQL script above, notice that the `Department.ID` column is a primary key. Also, the columns `Employee.DepartmentID` and `Department.ID` are both of type integer, which meets the prerequisites for creating a foreign key relationship between these two columns.

Creating foreign keys in the Online Browser

1. Right-click the `Employee` table and select **Create Key | Create Foreign Key** from the context menu. The foreign key appears in the "Keys" folder, with the name `<generated>`. Also, a script is generated in the [Database Structure Change Script Window](#)[37]. At this stage, you can still revert the changes by clicking the **Refresh** 🔄 button.
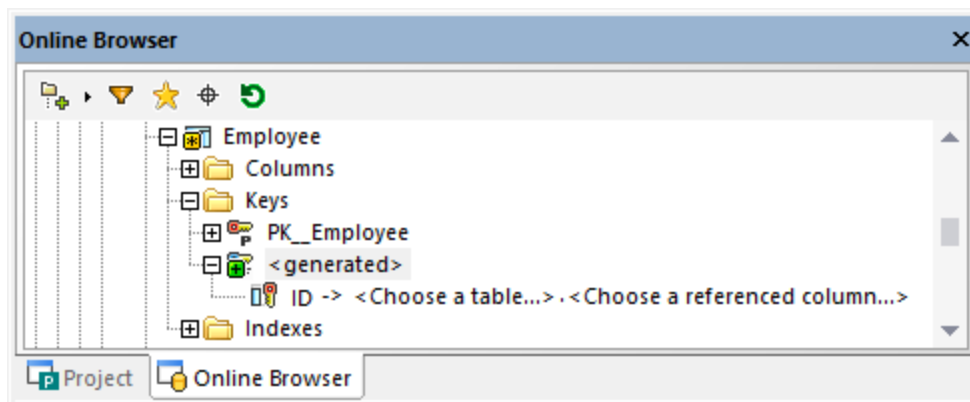
2. Optionally, select the **<generated>** name and press **F2** to rename it.
3. Double-click "Choose a table" and browse for the foreign table. In this example, the foreign table is `Department`. Note that the `ID` column is selected automatically, because it is the column on which the primary key is.
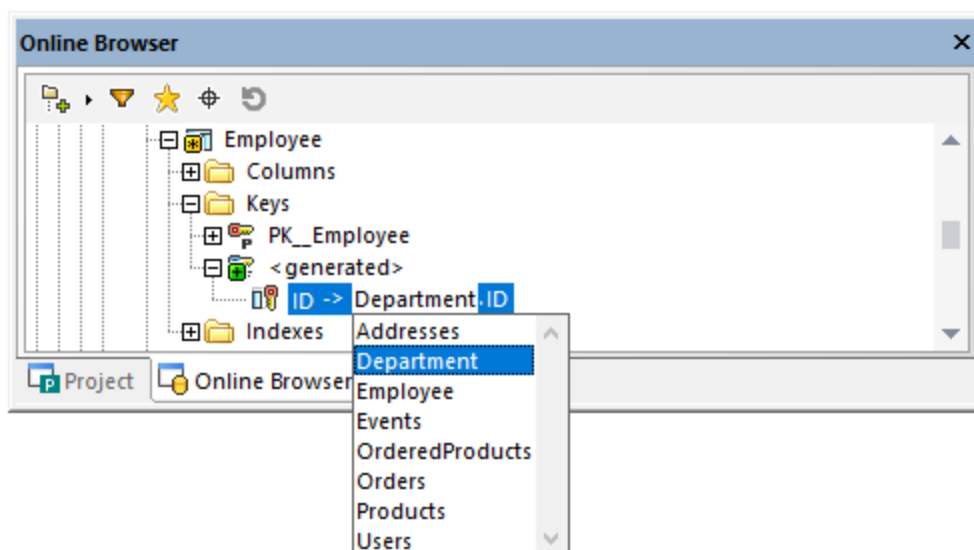
4. Click the **Execute Change Script** ▷ button in the Database Structure Change Script window.

Creating foreign keys in Design Editor (approach 1)

1. [Add](227) both tables to the Design Editor (`Employee` and `Department`).
2. Do one of the following:

    o   Drag the `DepartmentID` column from the "Columns" section of the `Employee` table (or from the Online Browser) and drop it onto the "Key" section. Select **Create Foreign Key** from the context menu when prompted.
    o   Drag the primary key from the "Key" section of the `Department` table (or from the Online Browser) and drop it onto the "Key" section of the `Employee` table.

    By default, a foreign key having the name `<generated>` is automatically created on the first column that has the same data type as the column that is referenced by the dragged key or index, respectively. This may or may not be the column on which you need the foreign key. Therefore, if applicable, double-click the column (in the "Key" section) and select the correct column from the list.
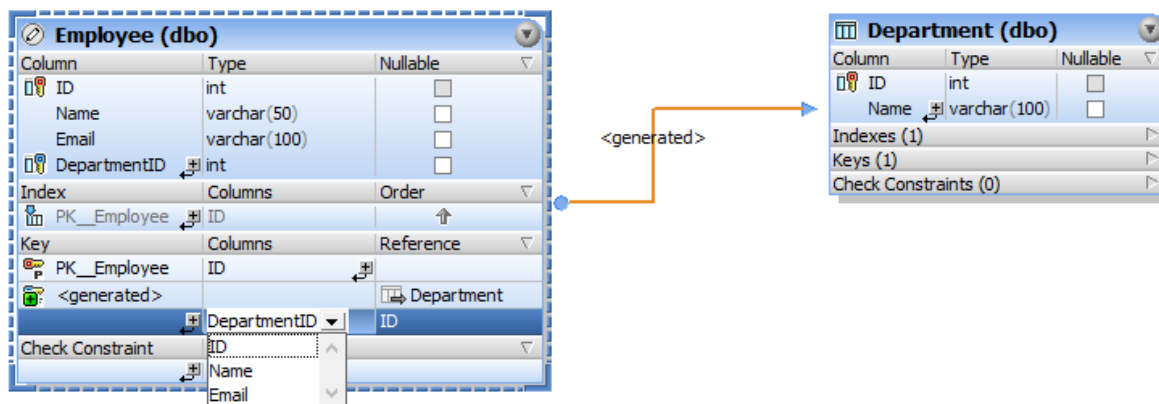
3.   Click the **Execute Change Script** ▷ button in the Database Structure Change Script window.

Creating foreign keys in Design Editor (approach 2)

1.   Do one of the following:

   o Click the **Create new Key Constraint** 🔲 icon in the "Key" section of a table design, and then
      select **Create Foreign Key** from the context menu.
   o Right-click anywhere into the table design and select **Insert new | Key | Foreign Key** from the
      context menu.

   A foreign key having the name `<generated>` is created automatically, without a reference to a
   particular table. To choose the referenced table, click **[select the referenced table]** and select the
   required table from the list.
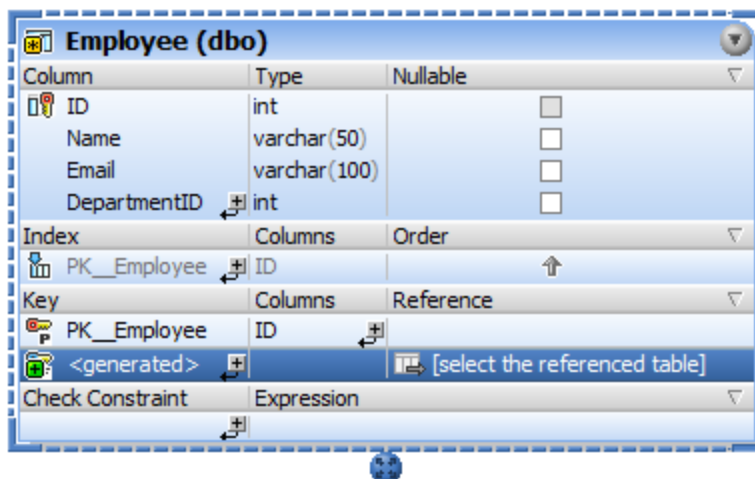
2.   If applicable, set the "from" and "to" columns for the foreign key, as follows:

   o   To change the column on which the foreign key is, double-click the column (in the "Key" section,
      under "Columns") and select the appropriate entry from the list.
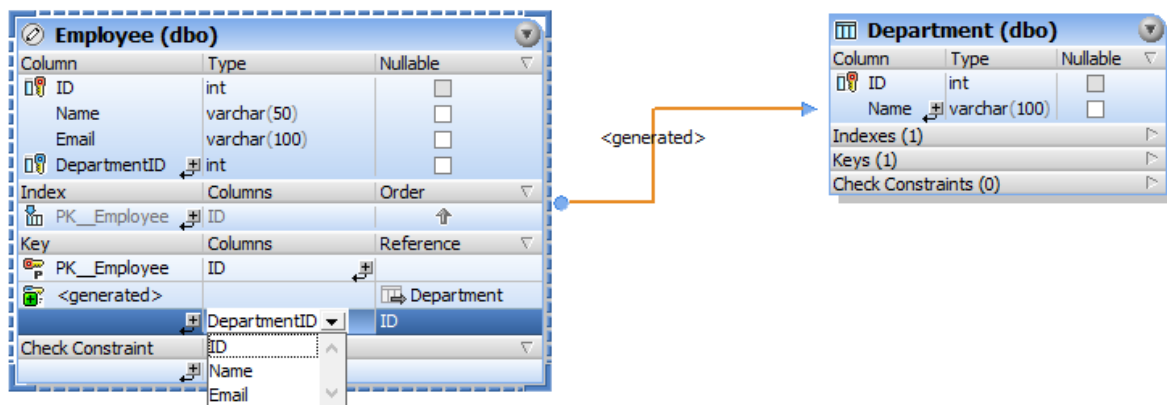
   o   To change the column to which the foreign key points, double-click the column (in the "Key"
      section, under "Reference"), and select the appropriate entry from the list.

3.   Click the **Execute Change Script** ▷ button in the Database Structure Change Script window.

Naming foreign keys

By default, the name of the foreign key is `<generated>`, which means its name will be automatically generated by the database. This ensures uniqueness of the name in the database. If necessary, you can override the generated name. To do this, enter a descriptive name in place of `<generated>`, and press **Enter**.

> If you override the generated name with a custom name, the new name must be unique; otherwise, errors will occur when the database change script is executed.

Note: In SQLite, custom key names are ignored by the database. The key name displayed in the Online Browser of DatabaseSpy is the internal key name as provided by the database.

Creating foreign keys by generating SQL

To create a foreign key, you can also generate an SQL statement using an existing foreign key as a model, as follows.

1.   In the Online Browser, expand the "Keys" folder of the table that contains the existing foreign key.
2.   Right-click the foreign key and select **Show in new SQL Editor | Add** from the context menu. Note that this command is disabled if the database does not support altering the structure of existing tables.
3.   In the SQL Editor, edit table name, column name, as well as the name of the foreign key as required.
4.   Click the **Execute** ▷ button to execute the SQL statement and create the new foreign key.

Cross-schema references

Foreign keys can also reference a column in a table that is located in a different schema. When defining this reference using the Design Editor, tables in the same schema are printed black, whereas tables located in a different schema are printed blue, for example:
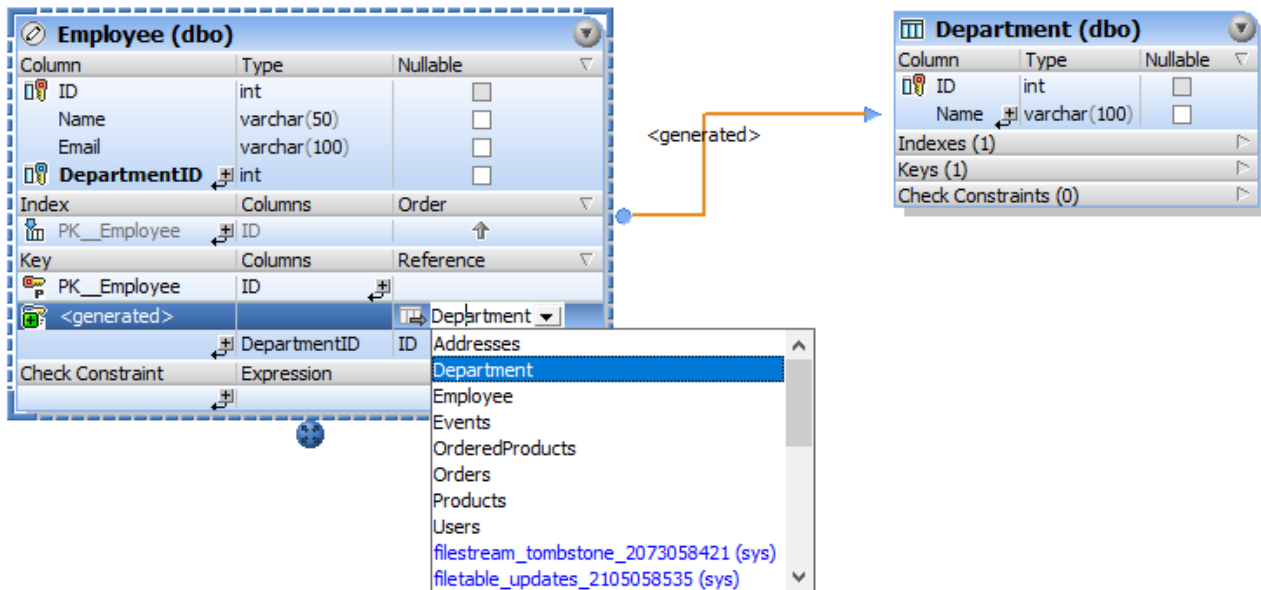
6.5.2   Renaming Foreign Keys

See <u>Renaming Database Objects</u> [279].

6.5.3   Dropping Foreign Keys

See <u>Dropping Database Objects</u> [282].

6.6   Default Constraints

Whenever you insert data into a column without supplying a value, a default constraint, if present, supplies a default value automatically. For example, a database could automatically generate the current date or time value for a data field whenever you insert a new record in a table. Any default constraint defines a value or expression that is to be used by default. This could be a string or numeric value, or an expression in the SQL grammar of the respective database (for example, a function call to get the current date or time).

In the Online Browser, columns that have a default constraint are shown with the ⬚ᴅ icon.
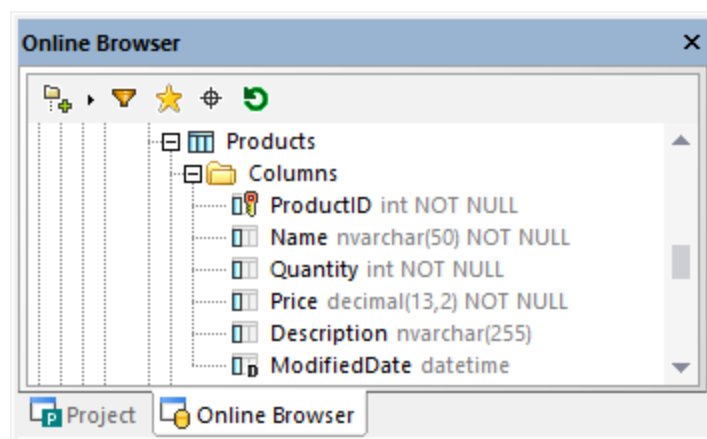
To view or edit the expression of the default constraint, first select the column, and observe the **Default** field in the Properties window.

In the Design Editor, default constraints appear in the "Constraints" section of a table design.

6.6.1   Adding Default Constraints

As with other database objects, there are several ways to create default constraints:

- From the Online Browser
- From the Design Editor

With most database types, you can add default constraints either when creating a new table, or later, by editing an existing table.

> SQLite does not support adding default constraints to existing tables. You can create default constraints only at table creation time.

Adding default constraints from the Online Browser
The following instructions use a SQL Server database as model. The instructions are similar in other database kinds.

---

1. Right-click an existing column and select **Create Default Constraint** from the context menu. The column and all its parent objects change appearance to indicate that an uncommitted change ⊞ is in progress. Also, a structure change script is generated in the [Database Structure Change Script Window][37].
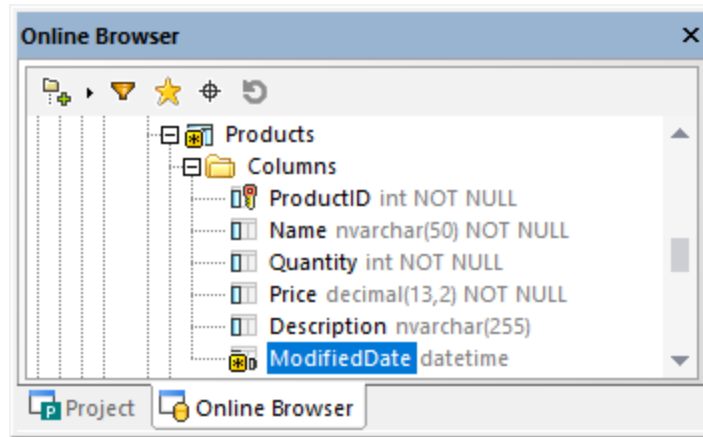
2. In the Properties window, find the **Default** property and enter the default value or expression to be used by this constraint.
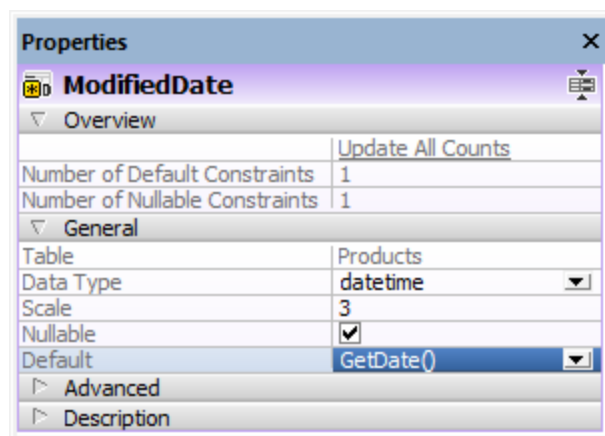
3. Click the **Execute Change Script** ▷ button in the Database Structure Change Script window.

Adding default constraints from the Design Editor

Use either of the two approaches to add a default constraint from the Design Editor.

Approach 1:

1. Right-click a table in Online Browser and select **Design Editor | Show in new Design Editor** from the context menu.
2. In the Design Editor, right-click the column for which you want to define a default value and select **Create Default Constraint** from the context menu.
3. In the Properties window, find the **Default** property and enter the default value or expression to be used by this constraint.
4. Click the **Execute Change Script** ▷ button in the Database Structure Change Script window.

Approach 2:

1. Right-click the title bar of the "Columns" section and select **Default value** from the context menu. A new "Default value" column appears on the table design.
2. Enter the default value or expression in the "Default Value" column.
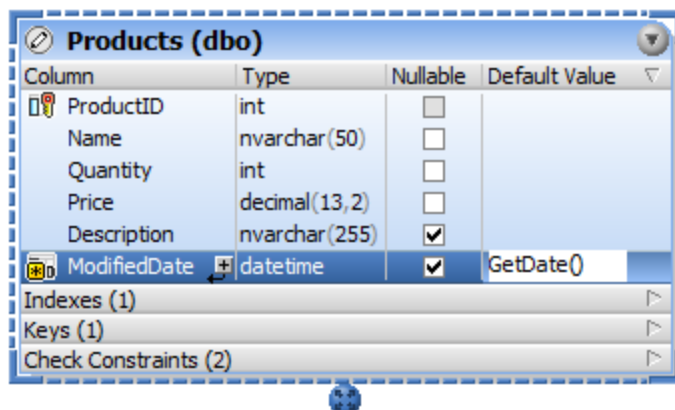
3. Click the **Execute Change Script** ▷ button in the Database Structure Change Script window.

6.6.2   Editing Default Constraints

When you change the value of an existing default constraint, the default constraint is always deleted first and then recreated with the new value.

To change the default value of a column:

1. Select the column in the Online Browser and make sure that its properties are being displayed in the Properties window.
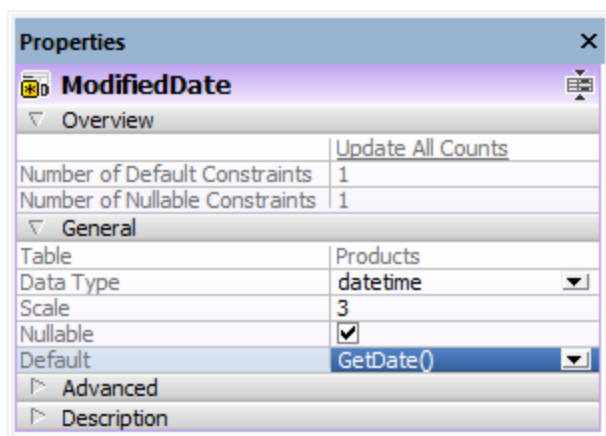
2. Edit the **Default** property as required. The respective SQL statement is generated automatically in the Database Structure Change Script window
3. Click the **Execute Change Script** ▷ button.

6.6.3 Dropping Default Constraints

To drop a default constraint:

1. In the Online Browser, right-click the column on which the default constraint is and select **Drop Default Constraint** from the context menu.
2. Click the **Execute Change Script** button in the Database Structure Change Script window.

6.7   Check Constraints

Check constraints are used to validate data accepted by a column. They provide an expression that must be satisfied before data can be inserted successfully. Check constraints can be defined on table level (that is, several columns of a table are checked in the expression) or on a single column. Multiple check constraints can be applied to a single column.

In the Online Browser, check constraints appear in the "Check Constraints" folder of a table, and are shown with the ✔ icon.

When you select a check constraint, the expression used by it is displayed in the **Default** property in the Properties window.

In the Design Editor, the expression is displayed in the "Expression" column of the "Check Constraint" section of a table design.
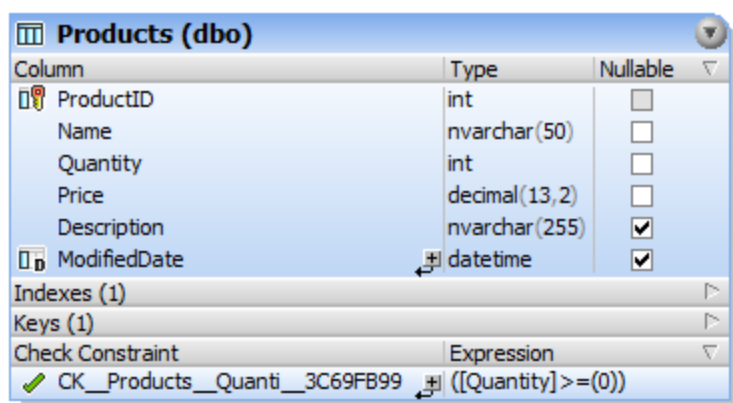
See also

6.7.1   Creating Check Constraints

As with other database objects, there are several ways to create check constraints:

- From the Online Browser
- From the Design Editor
- By generating SQL from an existing check constraint object

Adding check constraints is not supported for the following databases: MariaDB, MySQL, SQLite.

When adding a check constraint, make sure that the following prerequisites are met:

- The check constraint must have a valid expression in the SQL grammar of the currently connected database. For example, if column names are used in the expression, they must exist in the table.
- When adding the check constraint to a column that already has data, the column's data must satisfy the expression of the check constraint. Otherwise, the structure change script fails with an error (for example, *The ALTER TABLE statement conflicted with the CHECK constraint*).

While a check constraint is created but not committed to the database yet, its name may appear with the label **<generated>** in Online Browser and Design Editor. This means that the name will be automatically generated by the database when you commit the change script. For details about support for generated names across various databases, see Generated Names [277].

Adding check constraints from the Online Browser

The following instructions for adding a check constraint use a SQL Server database as model. The instructions are similar in other database kinds that support check constraints.

1. Right-click an existing column or table and select **Create Check Constraint** from the context menu. The new constraint appears in the "Check Constraints" folder of the table, and a structure change script is generated in the Database Structure Change Script Window [37].
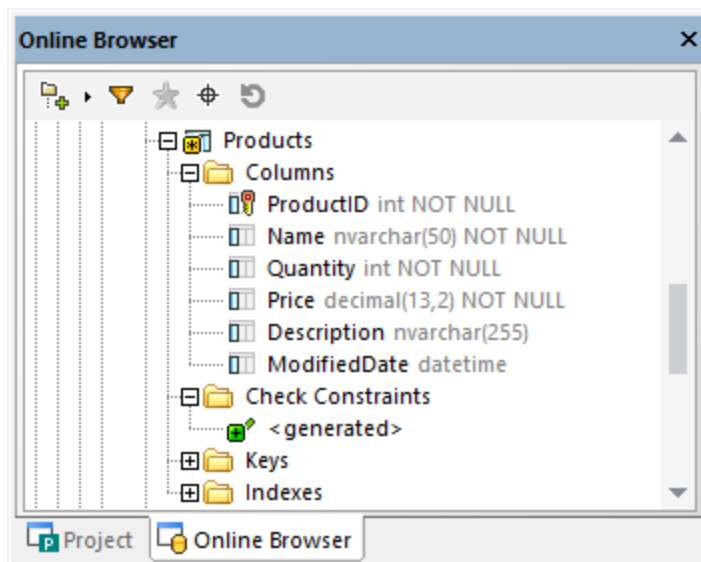
2. Optionally, select the `<generated>` name and press **F2** to rename it. For details about support for generated names across various databases, see Generated Names [277].

3. Making sure that the check constraint is selected in the Online Browser, edit the **Expression** property in the Properties window. Optionally, edit the **Description** property. In this example specific to a SQL Server database, the expression ensures that the column value is always greater than or equal to 0.

Properties	✕
⊞ **`<generated>`**	
▽ General	
Table	Products
Expression	[Quantity] >= 0
Description	
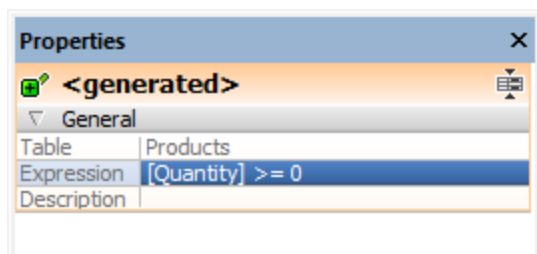
4. Click the **Execute Change Script** ▷ button in the Database Structure Change Script window.

You may need to **Refresh** ↻ the data source in order to view the latest changes in the Online Browser.

Adding check constraints from the Design Editor

To define a check constraint in the Design Editor:

1. Do one of the following:

   - Right-click into the table and select **Insert new | Check Constraint** from the context menu.
   - Expand the Check Constraints section by clicking the triangle at the right edge of the section header and click the **Create new Check Constraint** icon in the Check Constraint column.

   A new check constraint is added to the "Check Constraints" section and highlighted for editing.

2. In the "Expression" column, enter the expression that is to be checked when data is committed to the database and press **Enter**.
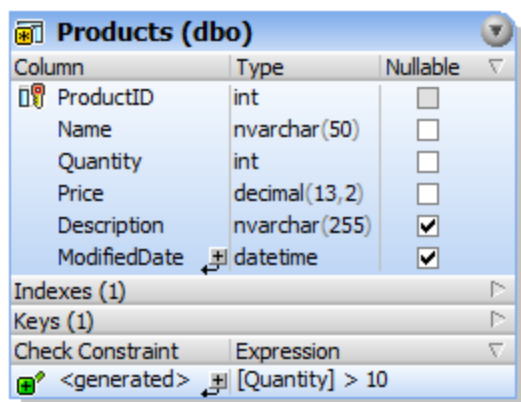
Products (dbo)		
Column	Type	Nullable ▽
🔑 ProductID	int	☐
Name	nvarchar(50)	☐
Quantity	int	☐
Price	decimal(13,2)	☐
Description	nvarchar(255)	☑
ModifiedDate	datetime	☑
Indexes (1)		▷
Keys (1)		▷
Check Constraint	Expression	▽
`<generated>`	[Quantity] > 10	

As always when making changes to the database structure, a change script is generated which must be executed in order to actually create the check constraint in the database.

3. Click the **Execute Change Script** ▷ button in the Database Structure Change Script window.

Adding check constraints using SQL

If check constraints already exist in a table, you can add new constraints by generating a statement that uses the existing check constraint as template, as follows:

1. Right-click an existing constraint in the "Constraints" folder.
2. Select **Show in new SQL Editor | Add** from the context menu or, holding down the right mouse button, drag the constraint into an open SQL Editor window and choose **Add** from the context menu. A Data Definition Language (DDL) statement is generated in the SQL Editor window.
3. Edit the generated statement, for example:

   a) The ALTER TABLE part of the statement must point to the correct table;
   b) The ADD CONSTRAINT part of the statement must specify a valid constraint name;
   c) The CHECK (Expression) section must contain a valid constraint expression in the SQL grammar of the currently connected database kind.

4. Click the **Execute** ▷ button to execute the statement.

6.7.2   Editing Check Constraints

You can change the expression of a check constraint using either of the approaches listed below. When you take this action, note that the check constraint is deleted first and then recreated with the new definition.

From the Online Browser

1. In the Online Browser, select a check constraint from the "Check Constraints" folder of a table.
2. Enter the new expression in the **Expression** property of the Properties window. A structure change script is generated automatically in the Database Structure Change Script window.
3. Click the **Execute Change Script** ▷ button in the Database Structure Change Script window.
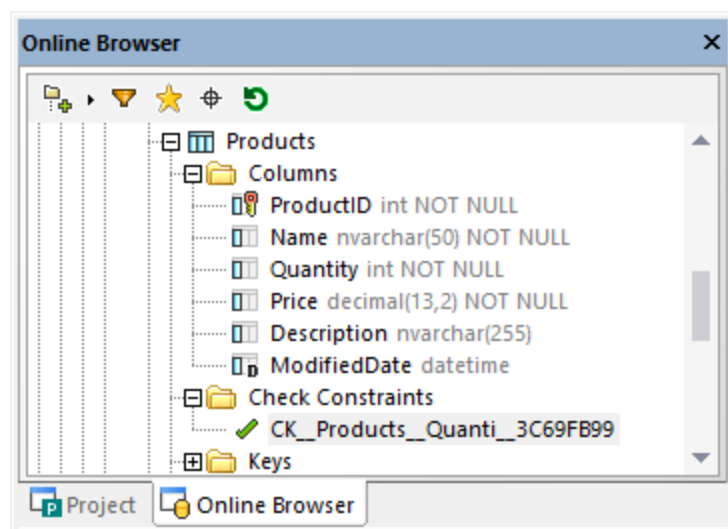
From the Design Editor

1. In the Design Editor, select a check constrains from the "Check Constraint" section of a table design.
2. Enter the new expression in the "Expression" column. A structure change script is generated automatically in the Database Structure Change Script window.
3. Click the **Execute Change Script** ▷ button in the Database Structure Change Script window.

6.7.3   Dropping Check Constraints

See Dropping Database Objects [282].

6.8   Indexes

Indexes are database objects that make queries of a database considerably faster; they are typically associated with one or more columns. In the Online Browser, indexes are displayed in the "Indexes" folder of a table. To view the index properties in detail, select an index and view the properties in the Properties window.
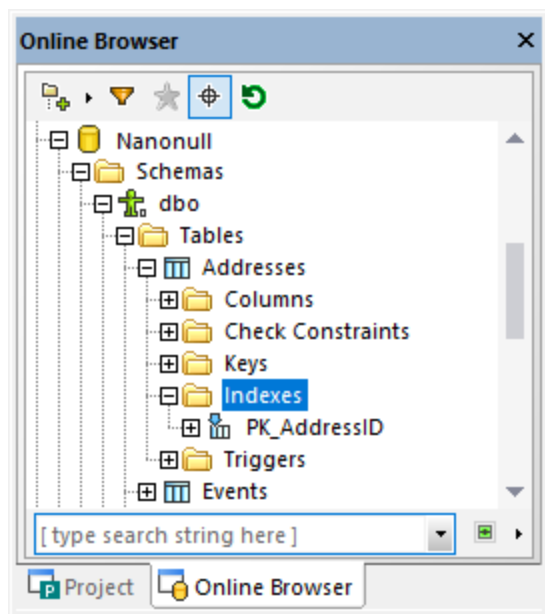
The type of an index is indicated by the following icons:

	Non-clustered index
	Clustered index created by a primary key constraint
	Unique non-clustered index
	XML index (available only for columns of XML type)

In the Design Editor, indexes are shown in the "Index" section of the table design. Here you see at a glance which columns are included in the index definition, and whether they will be queried in ascending or descending order. Indexes generated automatically are shown as grayed out and cannot be modified.
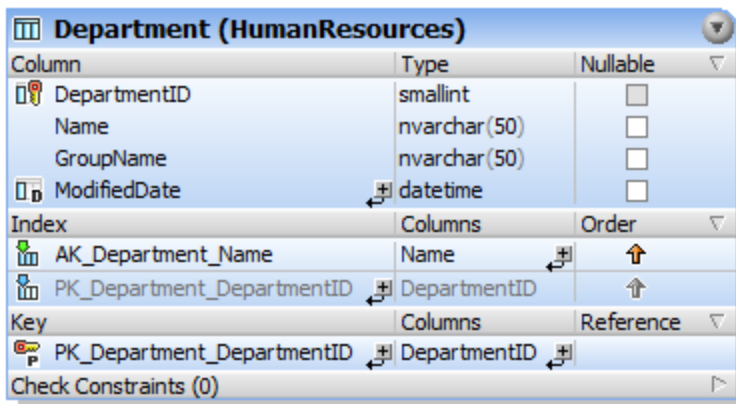
Table indexes in Design Editor

6.8.1   Creating Indexes

Some indexes in a database table are created automatically when you define a primary or unique key on a column. You can also add indexes explicitly, either at table design time, or later. Where supported by the database, DatabaseSpy allows for the definition of clustered, non-clustered, and XML indexes.

Note the following:

- In databases that support clustered indexes (DB2, DB2 for i, SQL Server, Sybase), only one clustered index can be defined per table.
- In MariaDB, MySQL, and SQLite databases, a unique key will be created automatically by the database when you add a unique index.
- For Teradata databases, you can manually create from Design Editor only secondary unique and non-unique indexes. To do this, right-click a table column in Design Editor, and select **Make index | Non Clustered Index** from the context menu. To specify that the index should be unique, select the **Unique** check box in the Properties window.
- With Teradata databases, a primary non-partitioned index is always created automatically for the first column in the table, even if you disabled the option **Automatically create primary key for new tables** from **Tools | Options | Design Editor**. Therefore, if you create a primary key manually from Design Editor, do not additionally create an index for that column; otherwise, the database change script will fail.

Creating indexes in the Online Browser

To create an index:

1. In the Online Browser, right-click a table, a column, or the "Indexes" folder of a table, and select **Create Index** from the context menu. The new index appears in the "Indexes" folder of the respective table, and a structure change script is generated in the Database Structure Change Script Window [37].

---
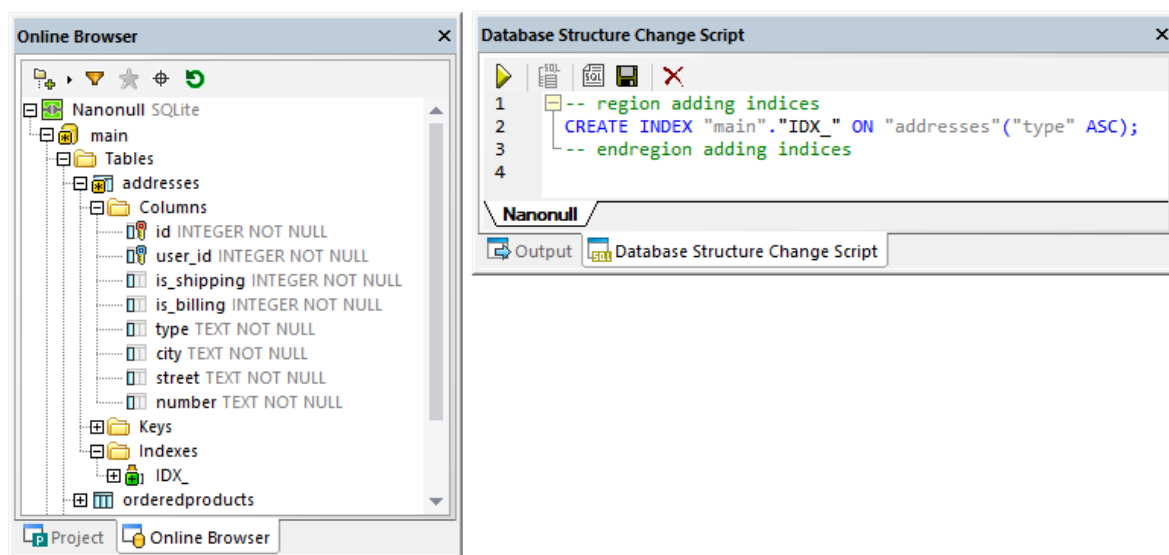
2. Select the new index and press **F2** to rename it. Alternatively, right-click the index and select **Rename** from the context menu.
3. Optionally, change the index properties in the Properties window. Available options vary by database kind. To see the available options in the Properties window, first click the index in the Online Browser.
4. Optionally, to add more columns to the index, right-click the index and select **Add Column | <name>** from the context menu. Note that only columns that are not already part of the index are shown in the context menu.
5. Click **Execute** ▷ to run the generated database change script.

Creating indexes in the Design Editor

To define an index in the Design Editor:

1. Do one of the following:

   - Right-click the column where you want to create the index definition, and select **Make Index** from the context menu.
   - Click the column and, keeping the mouse button pressed, drag it into the Index section.
   - Right-click into the table and select **Insert new | Index** from the context menu.
   - Expand the Index section (by default, it is collapsed) and click **Create new index** ⏎ .

2. Select the appropriate index type (non-clustered, clustered, or XML index) from the sub-menu that opens. The types of indexes that are available depend on the database kind you are using.
3. Optionally, change the index properties in the Properties window. Available options vary by database kind. To see the available options in the Properties window, first click the index in Design Editor.
4. In the Index column, enter a descriptive name and press **Enter**.
5. In the Columns column, optionally double-click the column and select a different column from the drop-down list.
6. Click into the Order column and select either ascending or descending sort order and press **Enter**.
7. Optionally, click the plus symbol ⏎ and select a column from the drop-down list and repeat this step to add one or more additional columns to the index definition.
8. Click **Execute** ▷ to run the generated database change script.

Creating indexes by generating SQL

If indexes already exist in a table, you can add new indexes to the table also by right-clicking an existing index and generating a statement that uses the properties of the existing item as default. After editing the properties as required and executing the SQL, a new index is added to the database.

To add an index to a table using SQL:

1. Connect to the database and show the respective data source in the Online Browser.
2. Expand the table to which you want to add an index.
3. Right-click an existing index in the Index folder, and select **Show in new SQL Editor | Create** from the context menu. Alternatively, holding down the right mouse button, drag the index into an open SQL Editor window and choose **Create** from the context menu. An SQL statement is generated in the SQL Editor window.
4. Edit the statement as required.
5. Click the **Execute** ▷ button to execute the statement and to add the new index to the database.
6. In the Online Browser, right-click the database and choose **Refresh** from the context menu.

6.8.2   Dropping Indexes

See [Dropping Database Objects](#) [282].

6.9   Views

If the currently connected database contains views, they are displayed in the "Views" folder of the Online Browser.

In the graphical user interface of DatabaseSpy, views appears with the 👓 icon.
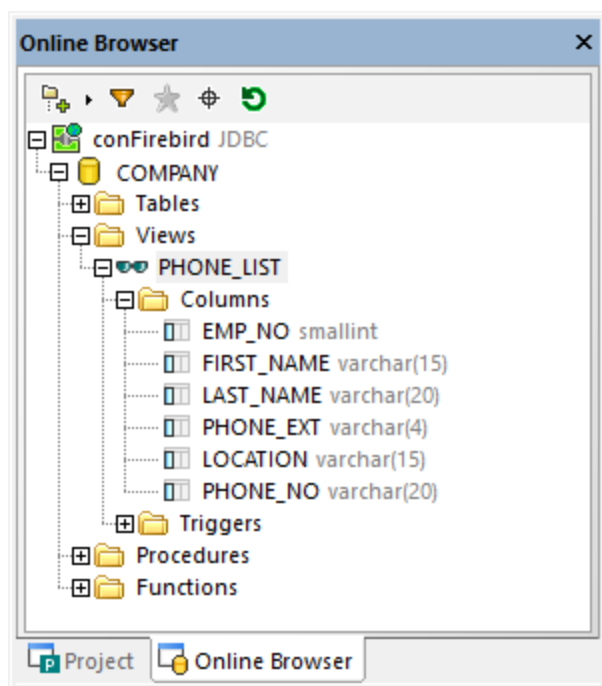
When you right-click a view in the Online Browser, the context menu provides several options to manage the views in your database, see also Creating Views [269], Modifying Views [270], and Deleting Views [271].

6.9.1   Creating Views

In DatabaseSpy you can create a view from a SELECT statement in the SQL Editor, or copy the definition of an existing view via the context menu in the Online Browser. This way, you have frequently used complex queries stored in database and can later execute them with a single mouse click or use them as a basis for more fine-tuned queries.

The SQL syntax in the instructions below may vary depending on the database kind you are working with.

Note: You may have to refresh the data source to view a newly created view in the Online Browser.

Creating Views in the SQL Editor
To create a view in SQL Editor:

1.   Write or generate a SELECT statement in the SQL Editor (for example, right-click a table in the Online Browser, and select **Show in new SQL Editor | SELECT** from the context menu).

---

2. Right-click in the SQL Editor window and select **Create View As** from the context menu. Alternatively, you can also select the menu option **SQL Refactoring | Create View As**. The statement CREATE VIEW [View1] AS is automatically inserted in front of the query and the view name View1 is automatically highlighted.
3. Change the view name to a more descriptive one. If a view with this name already exists, an error will occur.
4. Click the **Execute** ▷ button or press **F5** to create the view.
5. Refresh the data source to display the view in the Online Browser.

Be aware that, if your SELECT query retrieves data from more than one table, and if identical column names exist in selected tables, then ambiguous column names may occur in queries such as `SELECT * FROM table1, table2`. Creating views from SELECT queries which contain ambiguous columns is not supported and produces an error. The text of the error is different for each database kind and may not necessarily reflect the exact cause. Therefore, before creating views from a SELECT query, make sure that it does not contain ambiguous column names. For example, if the table INVOICE has a column "id" and the table INVOICEDATA has a column "id", an error would occur for a statement such as:

```
CREATE VIEW `View1` AS SELECT * FROM invoice, invoicedata WHERE invoice.id =
invoicedata.id;
```

However, the error would be eliminated with a statement such as:

```
CREATE VIEW `View1` AS SELECT invoice.id AS id1, invoicedata.id AS id2 FROM invoice,
invoicedata WHERE invoice.id = invoicedata.id;
```

Note the syntax in code listings above applies to MySQL.

To create a view based on an existing view:

1. In the Online Browser, right-click an existing view and select **Show in new SQL Editor | Create** from the context menu. A CREATE VIEW statement with the definition of the existing view is displayed in a new SQL Editor window.
2. Type a new name after the CREATE VIEW part of the statement and modify the SELECT part accordingly.
6. Click the **Execute** ▷ button or press **F5** to create the view.
3. Refresh the data source to display the view in the Online Browser.

6.9.2   Modifying Views

To modify a view:

1. In the Online Browser, right-click an existing view and select **Show in new SQL Editor | Alter** from the context menu. An ALTER VIEW statement with the definition of the existing view is displayed in a new SQL Editor window.
2. Modify the view as required.
3. Click the **Execute** ▷ button or press **F5** to create the view.

6.9.3   Dropping Views

See Dropping Database Objects [282].

6.10   Stored Procedures

If the currently connected database supports and contains stored procedures, these are displayed by default in the "Procedures" folder in the Online Browser.
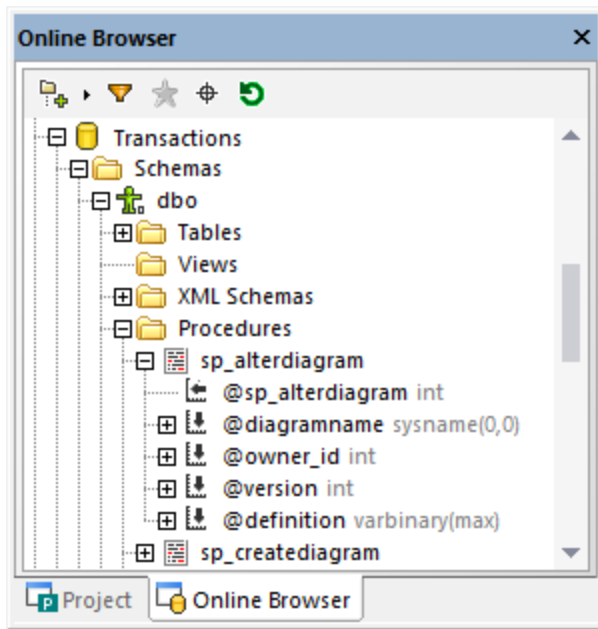
The DatabaseSpy icons associated with stored procedures are as follows:

⊞   Stored procedure

⬇   Input parameter

⬆   Input output parameter

⬆   Output parameter

⬅   Return parameter

To reveal various options for maintaining stored procedures, right-click the procedure in the Online Browser and select the required command from the context menu. The commands grouped under **Show in new SQL Editor** enable you to generate SQL statements [346] for creating, altering, dropping, or executing the stored procedure.

When renaming or dropping a stored procedure, a script is generated in the Database Structure Change Script Window [37]. The script must be executed in order for the changes to take effect.

6.11   Triggers

In databases such as IBM DB2, Oracle, and SQL Server, triggers are user-defined functions that are executed automatically when a database event occurs. The event can be a data definition event (for example, a table is created, altered, or dropped) or a data manipulation event (for example, a row is inserted, updated or deleted from table).

In DatabaseSpy, triggers are shown with the 🗄 icon.

In the Online Browser, triggers defined at table level are shown in the "Triggers" folder of the respective ⊞ table (or 👓 view, if applicable).
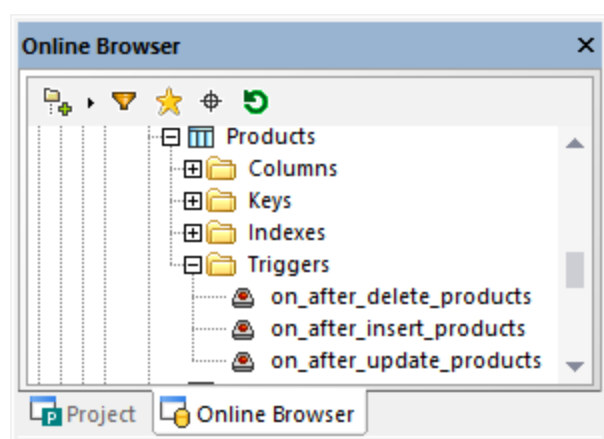
Triggers at 🗄 database or 🗄 schema level are shown in the "Triggers" folder of the respective database or schema.

Note: Server-level triggers (in SQL Server databases) are not supported in DatabaseSpy.

You can use an existing trigger as a basis for creating a new trigger definition or dropping a trigger from the database. The context menu in the Online Browser provides several options in this respect, see Generating SQL Statements[346].

6.12   User-defined Functions

If the database includes user-defined functions, such objects are displayed in a separate folder on the schema or database level in the Online Browser. The function's parameters and their data types are listed directly under the function.

Like other database objects, functions can be added to the favorites [212]. The maintenance options applicable to functions are available through the context menu, see Generating SQL Statements [346]. Please note that functions are not supported in Access databases.

6.13   PL/SQL Packages

In Oracle databases, packages are objects that group related PL/SQL types, items, and subprograms into modules and thus allow you to re-use code. Packages consist of a package specification and an optional package body. The specification is the interface and the body defines cursors and subprograms.

After you connect to an Oracle database with DatabaseSpy, the existing packages are displayed in the "Packages" folder of the Online Browser. If the package contains public functions or stored procedures, they are displayed in the "Procedures" and "Functions" sub-directories, respectively.
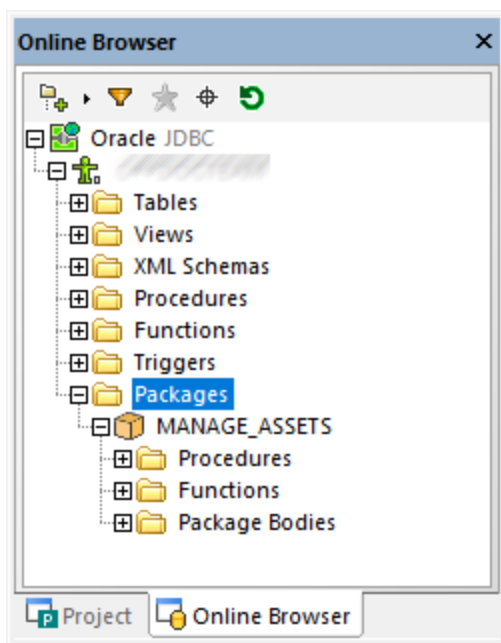
The DatabaseSpy icons associated with Oracle packages are as follows:

Package

Package body

Procedure

Function

You can create new packages by entering the respective commands in the SQL Editor window and then running them against the database, for example:

```
CREATE [OR REPLACE] PACKAGE package_name
...

[CREATE [OR REPLACE] PACKAGE BODY package_name
...
```

For the complete PL/SQL syntax, refer to Oracle documentation.

Alternatively, you can create new packages in DatabaseSpy by generating CREATE statements from existing packages, and then running them against the database like any other scripts. To do this, right-click a package or package body in the Online Browser, and select the menu command **Show in SQL Editor | CREATE**. Use the same approach to generate statements that alter an existing package or drop it from the database.

Note: The support for editing PL/SQL in DatabaseSpy is limited. PL/SQL statements must be sent to the database as an entire block, and not all PL/SQL statements may be recognized and parsed correctly by the editor. Also, there are known issues when using ADO connections.

6.14   Reference of Generated Names

When you create certain database objects (such as primary keys, check constraints, and others), they are given a default generated name which ensures that the object is unique in the database. Until you actually execute ▷ the database change script, the object name appears with the label **<generated>**, both in the Online Browser and Design Editor.

You can optionally change the **<generated>** name to a custom one before committing changes to the database. In this case, make sure that the custom name is unique; otherwise, the database change script will fail with an error.

Not all database objects support generated names. The following table illustrates support for generated object names across various databases.

Database kind	Primary key	Foreign key	Unique key	Check constraint	Index
Access	+	+	+	+	-
Firebird	+	+	+	+	-
IBM DB2	+	+	+	+	-
IBM DB2 for i	+	+	+	+	-
Informix	+	+	+	+	-
MariaDB	-	+	+	-	-
MySQL	-	+	+	-	-
Oracle	+	+	+	+	-
PostgreSQL	+	+	+	+	-
Progress OpenEdge	+	+	+	+	-
SQLite	+	+	+	-	-
SQL Server	+	+	+	+	-
Sybase	+	+	+	+	-
Teradata	-	-	-	-	-

Legend:

+	The object name will be generated automatically; you do not need to provide a name for it. Optionally, you can override the generated name.

| - | The object name cannot be generated automatically. Before committing changes to the database, you will need to define a unique name for the object. |

6.15   Renaming Database Objects

You can rename database objects from various contexts:

- From the Online Browser, by using the context menu
- From the Design Editor
- By generating an ALTER statement in the SQL Editor
- You can also rename an object by editing the "Title" property in the Properties window.

Note that support for renaming database objects is not uniform across all databases. The following table illustrates objects that can be renamed directly from the graphical user interface of DatabaseSpy, across various database kinds. Some databases may additionally support renaming of keys, check constraints, indexes, as well as stored procedures.

Database kind	Rename table?	Rename column?	Rename view?
Access	-	-	-
Firebird	-	Yes	-
IBM DB2	Yes	-	-
IBM DB2 for i	Yes	-	-
Informix	Yes	-	-
MariaDB	Yes	Yes	-
MySQL	Yes	Yes	-
Oracle	Yes	Yes	-
PostgreSQL	Yes	Yes	Yes
Progress OpenEdge	Yes	Yes	-
SQLite	Yes	Yes	Yes
SQL Server	Yes	Yes	Yes
Sybase	-	-	-
Teradata	Yes	Yes	Yes

Renaming objects from the Online Browser

To rename a database object:

1. Do one of the following:

- In the Online Browser, select a database object and press **F2** or double-click its title bar in the Properties window.
- In the Online Browser, right-click a database object and select **Rename** from the context menu.
- In a Design Editor window, double-click the title bar of a table or the database object, or right-click and select **Rename Table** or **Rename Column** or **Rename Key** or **Rename Index** or **Rename Check Constraint**, respectively, from the context menu.

2. Change the name as desired and press **Enter**. DatabaseSpy generates the required script and displays it in the Database Structure Change Script window.

Note: At this stage, the changes are not committed to the database yet. In the Online Browser, the icon appears to the left of the object name, which indicates that you are about to rename it. To undo the rename operation without committing it to the database, click **Refresh data source** in the Online Browser.

3. Optionally, to modify the generated script before running it, click the **Edit Change Script** button.
4. In the Database Structure Change Script window, click **Execute Change Script** .

Renaming objects from the Design Editor

To rename a database object:

1. Do one of the following:

   - In a Design Editor window, double-click the title bar of a table or the database object, or right-click and select **Rename Table** or **Rename Column** or **Rename Key** or **Rename Index** or **Rename Check Constraint**, respectively, from the context menu.

2. Change the name as desired and press **Enter**. DatabaseSpy generates the required script and displays it in the Database Structure Change Script window.

Note: At this stage, the changes are not committed to the database yet. To undo the rename operation without committing it to the database, click **Refresh data source** in the Online Browser.

3. Optionally, to modify the generated script before running it, click the **Edit Change Script** button.
4. In the Database Structure Change Script window, click **Execute Change Script** .

Renaming objects by generating SQL

To rename a database object by generating an ALTER statement:

1. In the Online Browser, right-click a database object and select **Show in new SQL Editor | Rename** from the context menu.
2. Click the **Execute** ▷ button at the top of SQL Editor (or press **F5**). The outcome is reported in the Message window.

6.16   Dropping Database Objects

Database objects can be dropped from various contexts:

- From the Online Browser
- From the Design Editor
- By generating a DROP statement in the SQL Editor

> Note that **SQLite** and **DB2 for i** databases do not support altering the structure of an existing table by dropping columns. For such databases, the commands for dropping a column are disabled in DatabaseSpy.

From the Online Browser

To drop a database object:

1. Do one of the following:

   - In the Online Browser, right-click a database object and select **Delete** from the context menu.
   - In the Online Browser, select one or more database objects and press **Delete**.

   DatabaseSpy generates the required script and displays it in the Database Structure Change Script window, for example:
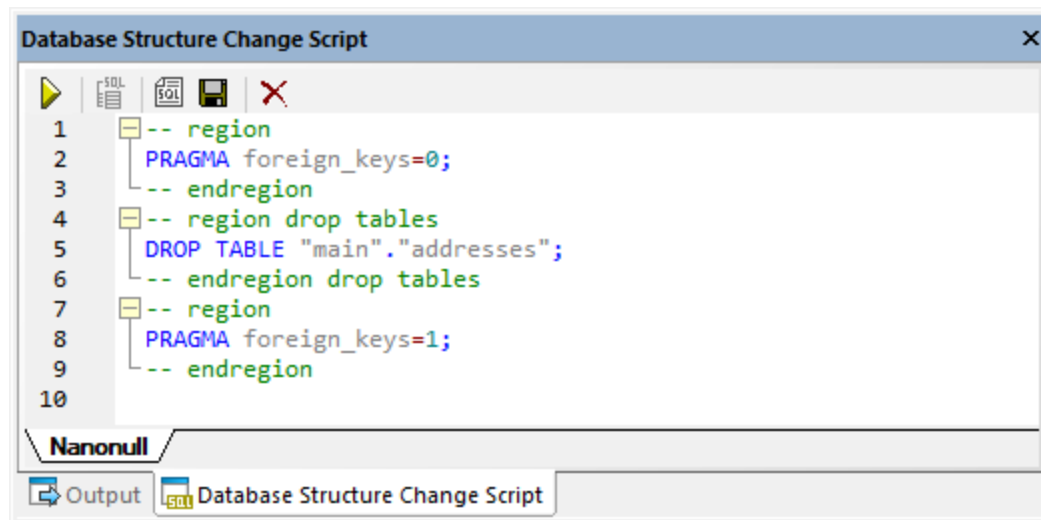
Note: At this stage, the changes are not committed to the database yet. In the Online Browser, the icon appears to the left of the object name, which indicates that you are about to delete it. To undo the drop operation without committing it to the database, click **Refresh data source** in the Online Browser.

2. Optionally, to modify the generated script before running it, click the **Edit Change Script** button.
3. In the Database Structure Change Script window, click **Execute Change Script** .

From the Design Editor

To drop a database object:

1. Do one of the following:

    -   In the Design Editor, right-click a database object and choose **Delete selected Objects** from the context menu.
    -   In the Design Editor, select one or more database objects and press **Delete**.

    DatabaseSpy generates the required script and displays it in the Database Structure Change Script window. Also, the icon ❎ appears to the left of the object name, which indicates that you are about to delete it (in the image below, a table is about to be deleted).
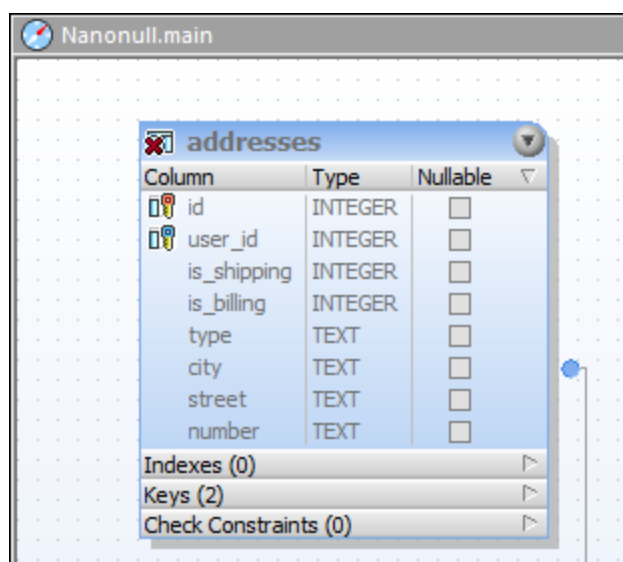
Note: At this stage, the changes are not committed to the database yet. To undo the delete operation without committing it to the database, click **Refresh data source** 🔄 in the Online Browser.

2. Optionally, to modify the generated script before running it, click the **Edit Change Script** 🔲 button.
3. In the Database Structure Change Script window, click **Execute Change Script** ▷.

By generating SQL

To drop a database object by generating a DROP statement:

1. In the Online Browser, right-click a database object and select **Show in new SQL Editor | Drop** from the context menu.
2. Click the **Execute** ▷ button at the top of SQL Editor (or press **F5**). The outcome is reported in the Message window.

---

7   Retrieving and Editing Data

Altova website: 🔗 Database tool

In DatabaseSpy, you can retrieve database data directly from the Online Browser or from a table design in the Design Editor using the **Retrieve data** option in the context menu. The required query is generated and executed and the data is displayed in a **Result** tab of the SQL Browser. You can choose to retrieve all rows at once or only a particular number of rows at a time.
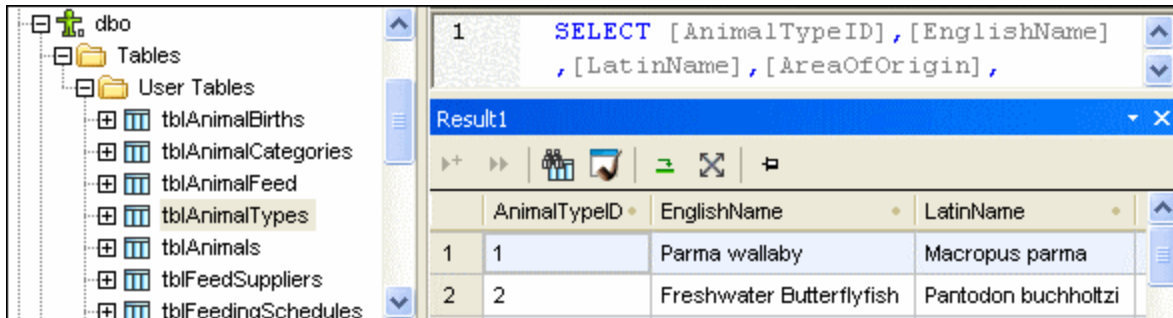
If you need to edit data after retrieving it, DatabaseSpy provides the **Edit Data** option in the context menu of the Online Browser for this purpose. The Result window is switched into the Editing mode in this case, indicating this condition in its status bar.
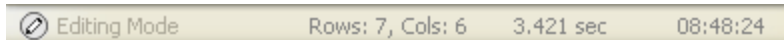
You can also select individual columns of a table when using the **Retrieve data** and **Edit data** commands. Only data from these columns will be retrieved in this case. Note, however, that all columns may be retrieved if you have deactivated the **Generate SELECT statements with full column list** check box in the SQL Generation⁴²⁶ options.

To retrieve data:

- In the Online Browser, right-click the object(s) you want to retrieve data from and select **Retrieve data | All rows** (**Ctrl+Alt+R**) or **Retrieve data | First n rows** (**Ctrl+Alt+T**) from the context menu.
- In the Design Editor, right-click either the title bar or one or several individual columns (using Ctrl+click) in a table design and select **SQL and Data | Retrieve data | All rows (Ctrl+Alt+R)** or **SQL and Data | Retrieve data | First n rows (Ctrl+Alt+T)** from the context menu.

  An SQL Editor window opens which displays the query for the data retrieval. If you have selected objects from different tables, separate SELECT statements for the individual tables are created. The results of the query are displayed in the Result window below.

To retrieve data for editing:

- Provided that direct editing of database data is supported for the respective data source connection, right-click a table or column in the Online Browser and select **Edit Data** from the context menu.

Partial retrieval

In the SQL Editor options [425], you can define the number n of rows to be retrieved if you select the **Retrieve data | First n rows** option from the context menu. You can then use the toolbar icons of the Result window [41] to retrieve the data in fractions.

Stopping the retrieval

While data is being retrieved, the status bar of the Result tab counts up the rows already retrieved and displays the execution time. In addition, the tab of the respective SQL Editor window shows an icon at the left side, indicating that data is still being retrieved.
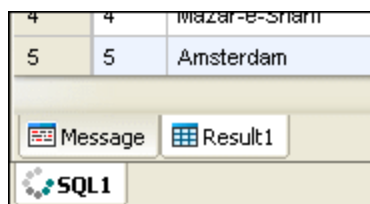
While data is being retrieved, the **Execute** button is replaced by the **Stop Retrieval** button in the SQL Editor toolbar. You can stop the retrieval process at any time by clicking the **Stop Retrieval** button. The status bar indicates that the retrieval has been stopped.
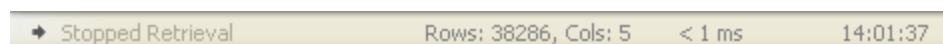
To stop the retrieval of database data:

· Click the **Stop Retrieval** button which is visible in the SQL Editor toolbar while data is being retrieved.

To resume the interrupted retrieval of database data:

Do one of the following:

· To retrieve the next n rows of data, click the **Retrieve next n rows** icon or right-click anywhere in the **Result** tab and select **Retrieve next n rows** from the context menu.

· To retrieve all outstanding data, click the **Retrieve outstanding rows** icon or right-click anywhere in the **Result** tab and select **Retrieve outstanding rows** from the context menu.

Editing database data in the Result window

To edit data contained in your database tables, you can either generate the corresponding SQL statements [346] and execute them in the SQL Editor or, more conveniently, retrieve data and edit it directly in the Result window.

The Result window has to be enabled for editing before you can make any changes to the data displayed in the result grid. For this purpose, the context menu of the Online Browser as well as the context menu that opens when you right-click into a table design in the Design Editor provides the **Edit Data** command which is available on table and column level. This command generates a SELECT statement which appears in a new SQL Editor window and is executed immediately, thus switching the Result window into editing mode.

In addition, the **Execute for Data Editing** ⬚ button is available in the SQL Editor toolbar, allowing you to execute a SELECT statement and, at the same time, switch the Result window into the Editing mode. This is particularly useful if you have already retrieved data and now want to update some of the records. The table below shows the editing options for the individual data source connections.

To select data for editing in the Result window:

Do one of the following:

- In the Online Browser or in a Design Editor window, right-click the table or column you want to edit, and choose **Edit Data** from the context menu. A SELECT statement is created and executed automatically in the SQL Editor.
- Generate a SELECT statement [348] in a new SQL Editor window and click the **Execute for Data Editing** ⬚ button, if available for the currently used data source connection.

Note: Not all editing options (update, insert, or delete row) may be available, depending on the data source connection you are currently using, and on whether the table you are editing has or has not a primary key defined.

The following restrictions apply when editing data in the Result window:

- Only SELECT statements may be present in the SQL Editor window if the **Execute for Data Editing** command is used. If you click the **Execute for Data Editing** button while any other statement is in the SQL Editor, the statement will be executed without switching the Result window into the Editing mode.
- Tables where row uniqueness cannot be guaranteed are only partly supported for editing. Namely, you can insert new rows but you cannot update or delete existing rows. This is typically the case for tables without primary or unique keys or indices (or SELECT queries where you omitted such columns). DatabaseSpy displays a warning dialog box if you select the **Edit Data** command for such a table or SELECT query.
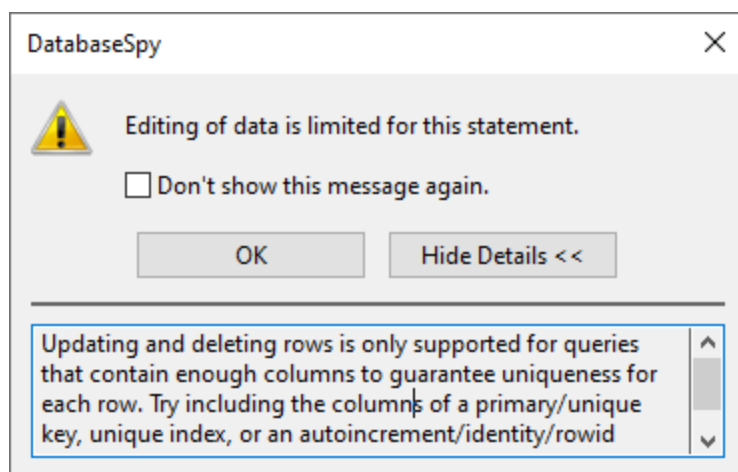
If you do not want to have this dialog box displayed, select the **Don't show this message again!** check box or disable this feature in the [SQL Editor options](#)[430].

- If the primary key is defined on multiple columns, all columns that form part of the table's primary key must be included in the statement. A warning dialog box similar to the one above ("Editing of data is limited for this statement") is displayed if you omit such a column in your SELECT statement.
- Columns that are the result of an equation or have certain properties such as auto-increment, timestamp, etc., are excluded from editing. If such columns are present, a warning dialog box similar to the one above appears before the results are displayed. You can, however, still edit the remaining columns of the table.

To make it possible to update or delete existing rows even in tables without primary or unique keys, edit the SELECT statement so as to include a column that guarantees uniqueness of the row. For example, if the table does not have a primary key, include an auto-increment column in the SELECT statement, or a column that has a unique key or unique index. In addition, SQLite databases provide by default an internal rowid column for this purpose. Even though this column is internal and thus not shown in the Online Browser, you can still use it in your SELECT statement, as shown below. You cannot update the value of rowid itself, though, so the warning dialog box is still displayed informing you that this column cannot be edited.

SQLite

```
SELECT "col1",
       "col2",
       "col3",
       rowid
FROM "main"."table";
```

Using a rowid is not possible if this behavior was explicitly disabled when creating the table (that is, if the table was created using the WITHOUT ROWID keyword).

7.1   Viewing Results

When an SQL Editor window includes more than one statement, the results of a query are by default displayed in separate result tabs with consecutive numbers: Result1, Result2, etc. To change this default display mode, turn on the **Show multiple results stacked** ![icon] option in the SQL Editor toolbar. When executing a script containing more than one query, only one result tab is displayed, and the results of the individual queries are displayed in separate panes of that tab. Note that you can change the display mode also after the query has been executed.

To display multiple queries in one result tab:

1. In the SQL Editor toolbar, switch the **Show multiple results stacked** ![icon] option on.
2. Click the **Execute** ![icon] button or press **F5**.
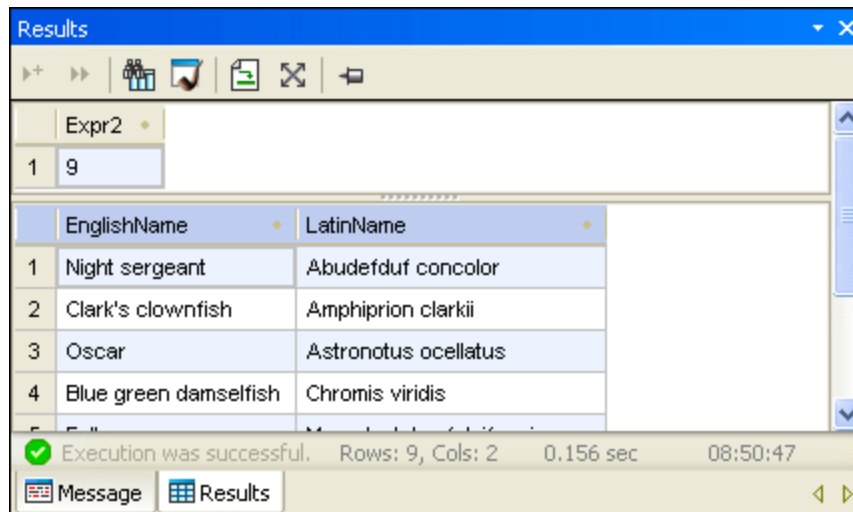
Only one result tab has been created. It consists of two panes however, each containing the result of the individual queries.

Viewing statistical data

The status bar of the Result window can display statistical information about any cells selected in the result grid, which includes:

- Number of selected cells (in the status bar, this statistic is shown as **Count**)
- Number of selected cells that have a numerical data type (**Numeric count**)
- The average value of all selected numeric cells (**Average**)
- The minimum and maximum value of all selected numeric cells (**Min** and **Max**)
- The sum of all selected numeric cells (**Sum**)
- The data type of a cell.

To view statistical information in the Result window, select one or more cells in the result grid. The values for the activated statistical fields are displayed in the status bar.

Not all statistics are enabled by default. To enable or disable a particular statistic, right-click the status bar and select the corresponding option from the context menu.

Note that the **Count** option displays the number of all selected cells, while the **Numerical Count** omits cells that do not have a numerical data type. Naturally, the **Average**, **Min**, **Max**, and **Sum** values are based on the numerical count.

When a single cell is selected, its data type is additionally displayed in the status bar.

Preserving results

If you want to keep a particular result, you can pin the corresponding tab in the Result window. DatabaseSpy

offers a toolbar button as well as several options in the context menu that opens when you right-click a tab in the Result window for this purpose. When the button is toggled on, the Result tab remains visible even if different queries are executed and new Result tabs are generated.
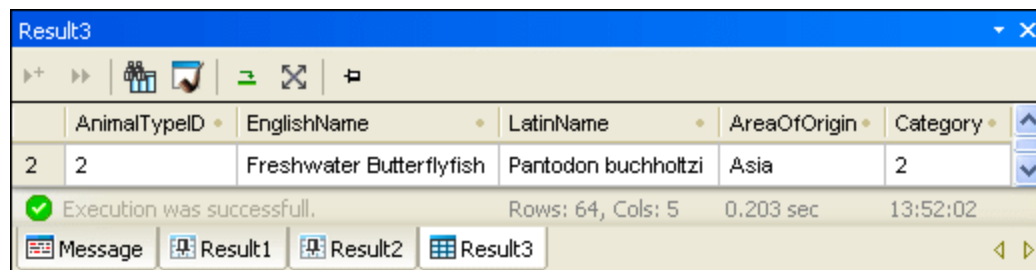
Note that pinned results show a different icon in the tab. When viewing pinned results, please bear in mind that the SQL statement displayed in the SQL Editor window above the Result tab might have been changed since the query has originally been executed and that re-executing the query may display a different result.

To pin a result tab:

1. Execute a query ²⁸⁴ and display its results in the Result window.
2. In the Result window, do one of the following:

- To pin an individual result, click the **Keep result visible** ⊡ button in the toolbar of the respective Result window or right-click the tab and select **Keep result visible** from the context menu.
- To pin all results that are currently displayed in the Result window, right-click any result tab and select **Keep all results visible** from the context menu.

Any pinned tab is kept in the Result window even if new queries are executed, and new Result tabs are generated.

To release a pinned result tab:

Do one of the following:
- To unpin a result, click the **Keep result visible** icon again, or right-click a pinned result tab and choose **Remove permanent visibility for result** from the context menu.
- To unpin all results in the Result window, right-click any result tab and select **Remove permanent visibility for all results** from the context menu.

7.2   Viewing Large Data Cells

In data cells containing a large amount of text, the Result tab displays only the first part of the text followed by an ellipsis (...). In this case, also the **Auto Size Columns** option in the context menu is limited to displaying 250 pixels. You can, however, always drag the column border to show more text.

Expanding cells temporarily

If cell content is larger than the currently visible cell width (e.g., in tables with a large number of columns), the width of a certain cell is temporarily extended if you move the mouse cursor over the data cell. This applies also to column headings.

	Ani...	AnimalN...	Ani...	A...	Ani...	Animal...	A...	Z...
45	5044	Penny	44	2	1	female	4	7
46	5045	Auntie	31	2	105	female	10	9
47	5046	Snuffelupigus		2	134	male	11	10
48	5047	Walter	1	3	12	male	6	8
49	5048	Maggie	1	3	9	female	3	8

To expand single data cells temporarily:

- Place the mouse cursor over the cell you want to expand.

Using the Data Inspector

Where the temporary expansion of long data cells is insufficient, for example if you want to check the content of XML columns, DatabaseSpy provides the Data Inspector[32] to display data cells containing a large amount of data.

Data Inspector ✕

WRAP 📄 Save As

Members of the 1980 USA men's ice hockey team, led by team captain Mike Eruzione
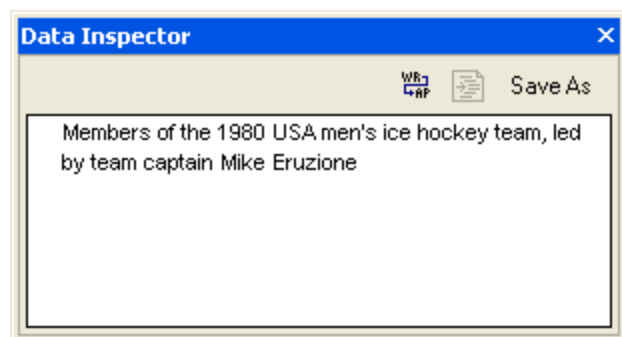
To view data using the Data Inspector:

1. Do one of the following:

a.   Click a data cell, and then click the **Show the Data Inspector window** toolbar button.

b.   Click a data cell, and then select the menu option **View | Data Inspector Window**.
c.   Right-click a data cell, and select **Show in Data Inspector Window** from the context menu.

The Data Inspector window opens and shows the content of the selected data cell.

2.   Optionally, click the **Word wrap** button to wrap the text in the Data Inspector window.

3.   Optionally, click the **Pretty Print** button to display the text in a hierarchical way.

4.   Optionally, click the **Save As** button to save the content as text or XML.

7.3   Counting Data Rows

You can display the number of rows contained in a table or view from the <u>Online Browser</u> [24], without having to retrieve the data first.

When you move the mouse cursor over tables or views in the Online Browser, a hyperlink appears to the right of the object name. This hyperlink reads "(count)" if no row count has been done yet, or "(n) update" if a number of n rows has already been retrieved. Click this hyperlink to update the row count information.

To disable the row count feature, clear the **Optional display of Table and View row counts** check box in the <u>Online Browser options</u> [420].

To show the row count in the Online Browser:

1. Make sure that the **Optional display of Table and View row counts** check box is selected in the Online Browser options.
2. Do one of the following:

   - Right-click a table, view, or folder containing tables or views in the Online Browser and select **Row Count | Show/Update** from the context menu.
   - Place the mouse cursor over a table or view and click the (count) hyperlink that appears.

To update the row count information:

- Click **Refresh** [icon].

To hide the row count:

- Right-click a table, view, or folder containing tables or views in the Online Browser and select **Row Count | Clear** from the context menu.

7.4   Searching and Sorting

Data that has been retrieved [284] from a database is displayed in DatabaseSpy's Result window [41]. If you are searching for a particular string in your data, you can either create an appropriate SELECT statement using a WHERE clause to retrieve only rows that include that string, or, you can make use of DatabaseSpy's search functionality directly in the Result window. This allows you to search for text in all the columns included in the result of a database query. You can also restrict the search to a particular column, row, or custom selection in the Result window.
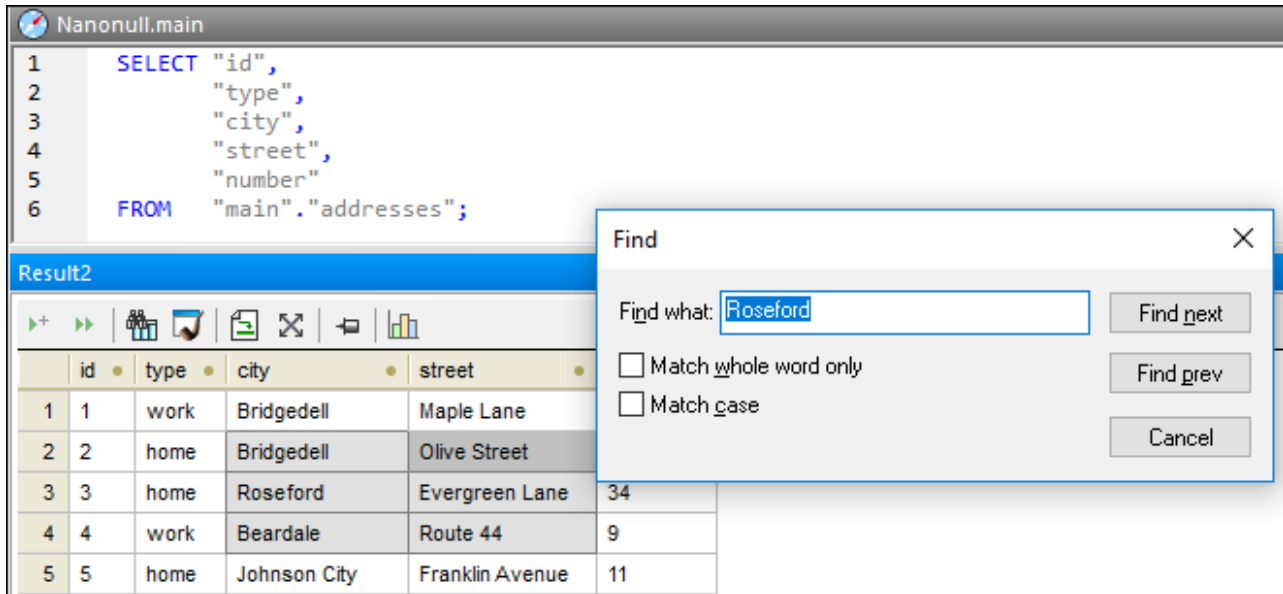
You can call the Find dialog box by clicking the **Find** 🔍 button in the Result window toolbar, or by pressing the standard shortcut **Ctrl+F** when the cursor is in the Result window.

The **Find** 🔍 button is also available when the Result window is in the Editing mode [298].

To search for text in the Result window:

1. Retrieve data [284] from the table you want to search.
2. Optionally, select the column or row where you would like to search for text. You can also select multiple cells by clicking while holding the **Ctrl** key pressed—in this case, the search will be restricted only to selected cells.
3. In the Result window, click the **Find** 🔍 button. Alternatively, click into the result grid and select the menu option **Edit | Find...** or press **Ctrl+F**.
4. Type the string you are looking for into the **Find what** field.
5. Optionally, select the **Match whole word** only check box if you want to restrict your search to entire words that match the search term.
6. Optionally, select the **Match case** check box if you want to consider the case when searching.
7. Optionally, select the **Search in selected cells only** check box if you want to restrict the search only to selected cells.

8. Click **Find next** to start the search. The first occurrence of the search term is highlighted in the result grid.
9. Optionally, click **Find next** again to jump to the next occurrence of the search term or click **Find prev** to jump to the previous occurrence of the search term, if available.
10. To quit the search, click **Cancel** or press **Esc**.

Sorting results

By default, the data is sorted according to its primary key column in ascending order. You can change the sort order in the result grid using the context menu, or by clicking the sorting arrows in the column headers, provided that this option has been activated in the SQL Editor options ⁴³⁰.

If sorting symbols are displayed in the column headers, you can click them to toggle the sorting order in the sequence:

1. ascending ▲
2. descending ▼
3. default ●

	id ●	name ▲	quantity ●
1	4	Accumax	0
2	3	Arepharm	65
3	9	Candysoft	23
4	1	Flexfan	200
5	8	Freshtex	10
6	11	n'drive	33
7	10	Ornamantis	2
8	7	Quadplus	550
9	6	Silver-Hold	54
10	2	Small Epsilon	4687
11	5	Sumtough	4

In the example above, data is sorted in ascending order according to the "name" column.

Note: Sort icons are not shown if only a partial result was retrieved (for example, if the execution was cancelled while in progress).

Sorting is case insensitive if the column by which you want to sort stores string values. For example, in the example above, ascending sorting was applied to the column "name", as indicated by the **ascending sort** ▲ icon. Thanks to case-insensitive sorting, the product "n'drive" occurs between "Freshtex" and "Ornamantis", even though it begins with a lowercase later.

To sort data in the result grid:

Do one of the following:

- Right-click anywhere in the column to be sorted and select either **Ascending** or **Descending** from the **Sorting** sub-menu of the context menu.
- Click the sorting symbol in any of the column headers to sort the data. The data is sorted according to the contents of that column in ascending order. Click the same column again to sort in descending order. A third click restores the default sort order.

To restore the default sort order:

- Right-click anywhere in the table and choose **Sorting | Restore default** from the context menu.

7.5   Printing Results

Data cells that you select in the result grid can be printed via the menu command **File | Print Preview**. This opens the selected cells in a separate pane, where you can zoom into or out of the grid and send the data to the printer. Alternatively, you can also use the **File | Print...** command, which opens the Print dialog box with the Selection option preselected in the Print range group box. Note that in the latter case no preview or zooming options are available.

	Employee ID	Supervisor ID	Last Name	First Name	Position	Birth Date
1	1	5	Davolio	Nancy	Sales Representative	1972-12-08
2	2	[NULL]	Fuller	Andrew	Vice President, Sales	1969-02-19
3	3	5	Leverling	Janet	Sales Representative	1971-08-30
4	4	5	Peacock	Margaret	Sales Representative	1973-09-19
5	5	2	Buchanan	Steven	Sales Manager	1975-03-04
6	6	5	Suyama	Michael	Sales Representative	1963-07-02
7	7	5	King	Robert	Sales Representative	1972-05-29
8	8	5	Callahan	Laura	Inside Sales Coordinator	1974-01-09
9	9	5	Dodsworth	Anne	Sales Representative	1976-01-27
10	10	2	Hellstern	Albert	Business Manager	1968-03-13
11	11	10	Smith	Tim	Mail Clerk	1973-06-06
12	12	10	Patterson	Caroline	Receptionist	1979-09-11
13	13	2	Brid	Justin	Marketing Director	1977-10-08
14	14	13	Martin	Xavier	Marketing Associate	1975-11-30

The example above shows the result of a retrieval of an employees table. You can select particular names in this result (e.g., all the managers) and print only their names and positions (see screenshot below). Only selected rows and columns will be printed.

7.6   Updating Data

To update data in the Result window, first [select data for editing](286). This switches the Result window into the Editing mode, which is indicated by the "Editing Mode" message in the status bar.

To edit records, double-click the cell you want to edit and proceed like in any spreadsheet application. You can use the context menu to copy, cut, paste, delete, or select all content or to undo your editing action. You can also paste into the Result grid cell ranges copied from another Result grid or from external files such as Microsoft Excel spreadsheets (see also [Adding and Copying Rows](304)).

Result1

	AnimalID	AnimalName	AnimalType	AnimalFeedSchedule	AnimalWeight	AnimalSex	AnimalAge	ZookResponsible
1	5000	Lobo	62	3	110	male	2	1
2	5001	Jumpy	62	3	1300	male	4	2
3	5002	Howler	62	3	100	male	3	1
4	5003	Crock	52					2
5	5004	Myra	64					3
6	5005	Bucky	41a	2	3	male	2	3
7	5006	Piggy	35	2	1	female	1	4
8	5007	Kermit	35	2	1	male	2	4
9	5008	Agatha	51	4	500	female	138	5
10	5009	Joe	51	4	623	male	147	5
11	5010	Ann	51	4	512	female	102	5

Data entered does not seem to be correct for this column's datatype

Editing Mode

Message   Result1

Result grid

The updated data is not immediately committed to the database but only displayed in the Result grid. To actually commit the update to the database, the Result window provides a **Commit** button in its toolbar.

Data cells that have been edited are indicated by a different background color. The color informs you if there are potential problems with the entered data. For example, if the entered text is not suitable for the data type of the column, the background color of the cell turns red.

If XML columns are present in the Result window, an additional button for each row appears inside the XML column: ▦. Clicking this button allows you to load an XML file into this XML cell or to save the XML content of the respective cell as XML file. Furthermore, you can [assign an XML schema](399) to the XML cell, provided that the current data source connection supports this feature. In databases where XML columns are currently not supported by DatabaseSpy, this icon changes to ▦ and you can only save the XML content of the respective cell to a file.

In the Editing mode, the toolbar shows the following additional commands:

+	Appends a new row. This command is also available in the context menu, when your right-click a cell. See also [Adding and Copying Rows](304).

x	Deletes an existing row. This command is also available in the context menu, when your right-click a cell.
🗙	Undoes all changes made to the Result grid (provided that they have not been committed to the database yet). To undo the editing of a particular cell, right-click it, and select **Undo Changes for this Cell** from the context menu.
Def	Sets the currently selected cell(s) to the default database value. This command is also available in the context menu, when your right-click a cell. Note that the command (and its corresponding button) is enabled only if the database column provides a default value.
0	Set the currently selected cell(s) to a NULL value. This command is also available in the context menu, when your right-click a cell. Note that the command (and its corresponding button) is enabled only if the corresponding database column allows nulls.

When you finished editing data, click **Commit** to send the changes to the database. DatabaseSpy automatically creates the required SQL statements and executes them. If the execution fails, you can view the statements in the Output window and check why they have not been executed successfully.

```
Message
 ▽  ▼  ▲ | 🗐 🗐 🗐 | 🔍 🔍 🔍 | ✕
⊟ ⊘ Edit Data
      UPDATE "USER"."CLIENTS" SET "ID"=917a, "STATUS"='Gold'  WHERE (  "USER"."CLIENTS"."ID"=9177)
      Update Failed Reason: Database Generic Error
  ❌ Reason:
      [IBM][CLI Driver][DB2/NT] SQL0103N  The numeric literal "917a" is not valid.  SQLSTATE=42604
```
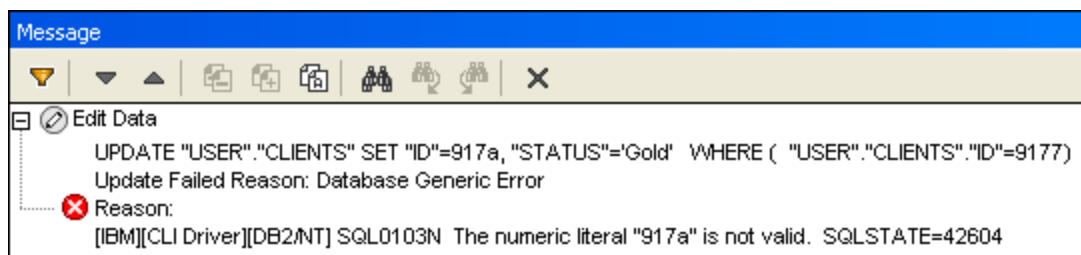
Using SQL

If you prefer to update data by means of SQL statements, DatabaseSpy can generate the required SQL statement based on the detected table structure. You can choose to update either all columns of a table or only specific ones.

To update a table by means of SQL statements:

1. Connect to the database and show the respective data source in the Online Browser.
2. Optionally, right-click the table you want to update and choose **Retrieve data | All rows** to see the data that is contained in the table.
3. Do one of the following:
   • Select the respective table you want to update.
   • Expand the respective table and select the column(s) you want to update.
4. Right-click and select **Show in new SQL Editor | Update** from the context menu or, holding down the right mouse button, drag the selected objects into an open SQL Editor window and choose **Update** from the context menu. An UPDATE statement for all the selected objects is generated in the SQL Editor.

Note: Auto-increment and computed columns do not appear in the generated UPDATE statement. See also "The UPDATE statement" below.

5. Edit the SET part of the generated UPDATE statement with the values that you want to be inserted.
6. Edit the WHERE clause to select the rows that are to be updated.

Note: If you remove the WHERE clause, all rows of the table will be updated with the same values.

7. Click the **Execute** [>] button to execute the statement and to update the data in the database.

The UPDATE statement

By default, the UPDATE statement is intentionally generated so as to prevent accidental update of multiple rows. For this reason, you will likely need to edit it in order to achieve the desired result.

The WHERE clause of the UPDATE statement is generated so as to include the table's primary key, if the table has one. For example, the statement below is generated for a SQLite table which has a primary key on the **id** column.

```
UPDATE "products" SET "id"=0, "name"='', "quantity"=0, "description"=NULL WHERE "id"=0;
```

If the table does not have a primary key, the WHERE clause will include the unified list of all columns of all unique keys. If the table has neither primary nor unique keys, the WHERE clause will include all the columns, except for computed columns.

The generated values that appear in the WHERE clause depend on the column's default value, on whether it is nullable, and on the data type of the column. For example, the following is an example of an UPDATE statement generated from a SQL Server table where (i) no primary or unique keys exist, (ii) the first column is of **integer** type, (iii) the second column is of **nvarchar** type, and (iv) **id** is an auto-increment column:

```
UPDATE [dbo].[Events]
SET    [event]=N''
WHERE  [id] = 0
       AND [event] = N'';
```

7.7   Editing Binary Columns

Binary content can be stored in databases using the following data types:

Oracle	BLOB, LONG RAW, RAW, BFILE
MS SQL Server	binary, varbinary, image
MS Access	BINARY, IMAGE
MySQL	binary, tinyblob, varbinary, blob, mediumblob, longblob
IBM DB2	BLOB
IBM DB2 for i	BLOB, BINARY LARGE OBJECT, BINARY, VARBINARY
SQLite	BLOB
Sybase	binary, varbinary, image
PostgreSQL	byte array

In DatabaseSpy you can choose to (i) save binary content from the result grid to a file, or (ii) load binary content from a file and store it in the database. The respective menu entries are available when you click the 🖳 button inside a data cell of the result grid.

	id	name	photo	status
1	3227	Ella Kimpton	🖳 0xFFD8FFE00010...	Gold
2	8877	Chris Bontempo	🖳 [NULL]	Gold
3	9077	Lisa Hansen	🖳 0xFFD8FFE00010...	Silver
4	9177	Rita Gomez	Load Binary from File...	
5	5681	Paula Lipneski	Save Binary to File...	
6	4309	Tina Wang	🖳 [NULL]	Silver

 DatabaseSpy presents the standard Windows **Open**, respectively the Save As, dialog box for this purpose. Binary content can be saved to the following formats BIN, BMP, JPG, PNG, MP3, AVI, and PDF.

To change the binary content of a database data cell:

1.   Click the Browse 🖳 button inside the data cell you want to edit.
2.   From the menu that appears, select **Load Binary from File...** or **Save Binary to File...**.
3.   Do one of the following:

   •   In the Open dialog box, browse to the desired binary file and click **Open**.
   •   In the **Save As** dialog box, select the folder where you want to save the binary file, enter a name for the file, and click **Save**.

4.   To commit your changes to the database, click the [Commit] button in the Result window toolbar.

Note: MS Access does not support the loading of a binary file in a binary column.

Note: When connecting to PostgreSQL through the ODBC driver, select the **bytea as LO** check box on the
second page of the advanced data source options of the PostgreSQL driver settings.

7.8   Inserting Data

In DatabaseSpy, you have several possibilities to populate your tables with data:

- Import database data from CSV or XML files, see <u>Importing Database Data</u> ⁴⁰⁰
- Manually add new rows directly into the Result grid. You can also paste into the Result grid cell ranges copied from another Result grid or from external files such as Microsoft Excel spreadsheets, see <u>Adding and Copying Rows</u> ³⁰⁴.
- Generate multiple INSERT statements by means of SQL templates and run them against the database, see <u>SQL Templates</u> ³⁸²

When you want to add only a few rows of data, you can also generate INSERT statements and enter the values manually in the SQL Editor. You can enter data for a particular column or for the entire row of a table.

To insert data using INSERT statements:

1. Connect to the database and show the respective data source in the Online Browser.
2. Right-click the table and select **Show in new SQL Editor | Insert** from the context menu. Alternatively, hold down the right mouse button, drag the table into an open SQL Editor window and select **Insert** from the context menu. An INSERT INTO... statement is generated in the SQL Editor.

Note: Auto-increment and computed columns do not appear in the generated INSERT statement.

3. Enter the desired values in the VALUES part of the statement (the number of commas depends on the number of columns in the respective table). If required, check the data type of the individual columns in the Properties window of the Online Browser.
4. Click the **Execute** button to execute the statement and to insert the data into the database.
5. Optionally, save the SQL statement if you need to insert the same kind of data frequently, see <u>Opening, Saving, and Closing SQL Files</u> ³⁵¹.

To insert data only into specific columns of a table:
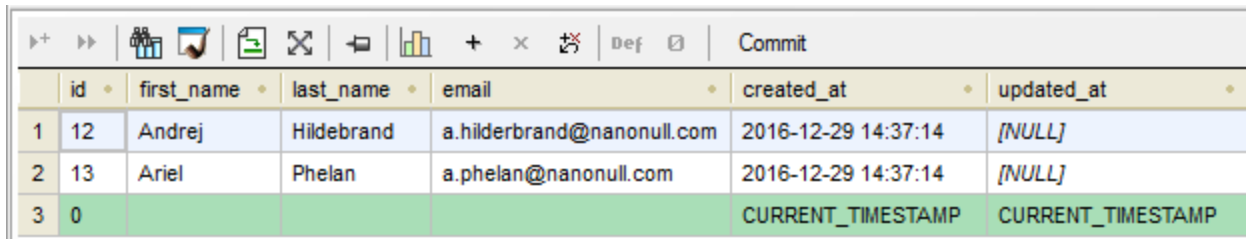
1. Connect to the database and show the respective data source in the Online Browser.
2. In the Online Browser, expand the table you want to edit to and select the column(s) into which you want to insert data. If your table contains columns that are not nullable, make sure to include these columns in the selection.
3. Right-click and select **Show in new SQL Editor | Insert** from the context menu. Alternatively, hold down the right mouse button, drag the column(s) into an open SQL Editor window and select **Insert** from the context menu. An INSERT INTO... statement is generated in the SQL Editor.

Note: Auto-increment and computed columns do not appear in the generated INSERT statement.

4. Enter the desired value(s) in the VALUES part of the statement. You may also have to enter the primary key for the new table row. Add the name and the value of the primary key to the statement in this case.
5. Click the **Execute** button to execute the statement and to insert the data into the database.
6. Optionally, save the SQL statement if you need to insert the same kind of data frequently, see <u>Opening, Saving, and Closing SQL Files</u> ³⁵¹.

7.9   Adding and Copying Rows

New rows can be added in the result grid of the SQL Editor, provided that data was retrieved in Editing mode, see Retrieving and Editing Data [286]. The toolbar of the Result window provides the **Append a new row** command for this purpose. This command inserts an empty row where you can enter data and save the record using the **Commit** button.

	id	first_name	last_name	email	created_at	updated_at
1	12	Andrej	Hildebrand	a.hilderbrand@nanonull.com	2016-12-29 14:37:14	[NULL]
2	13	Ariel	Phelan	a.phelan@nanonull.com	2016-12-29 14:37:14	[NULL]
3	0				CURRENT_TIMESTAMP	CURRENT_TIMESTAMP

You can define in the SQL Editor options [430] that the respective default values of the individual columns be inserted into new rows. In the example above, only the columns created_at and updated_at have a default value defined; DatabaseSpy hence populates them automatically. If the default property has not been set accordingly, a blank row will be inserted into the result grid.

In addition to editing rows manually, you can also paste into the result grid entire cell ranges (rows, columns, or both) from external files such as Microsoft Excel spreadsheets. The external cell ranges can be pasted over existing cells of the Result grid.

To paste clipboard data from external files into new rows, first click **Append a new row** (+) to create the required number of rows, and then paste the clipboard cells at the desired location. The **Paste** command is available both as a keyboard shortcut (**Ctrl+V**) or from the **Edit | Paste** menu.

If you are pasting data from a different result grid of DatabaseSpy, it is possible to create the required number of new rows automatically as you paste data. To do this, right-click the Result window and select the **Paste as new rows** (**Ctrl+Shift+V**) command from the context menu.

Adding new records to incomplete selections of data

When not the entire table but a collection of individual columns is selected for data editing, the following restrictions apply when a new row is to be inserted into the result grid:

- The column(s) containing the table's primary key must be included to make data editing possible.
- All columns with the "Nullable" property not set must be included unless they have a default value defined.

Using the table below as example, let's assume that both the NAME and the STATUS columns are not nullable, but only the STATUS column has a default value ("Standard") defined.

	ID	NAME	STATUS	CONTACTINFO
1	3227	Ella Kimpton	Gold	<?xml version="1.0" encoding="UTF-8" ?><C...
2	8877	Chris Bontempo	Gold	<?xml version="1.0" encoding="UTF-8" ?><C...
3	9077	Lisa Hansen	Silver	<?xml version="1.0" encoding="UTF-8" ?><C...
4	9177	Rita Gomez	Gold	<?xml version="1.0" encoding="UTF-8" ?><C...
5	5681	Paula Lipenski	Standard	<?xml version="1.0" encoding="UTF-8" ?><C...
6	4309	Tina Wang	Silver	<?xml version="1.0" encoding="UTF-8" ?><C...
7	*[NULL]*	*[NULL]*	*[NULL]*	*[NULL]*

We could further assume that each new customer is automatically assigned status "Standard" and thus omit the STATUS column when creating new customers. Adding new rows is possible in this case since the default value is automatically assigned when the record is committed to the database.

	ID	NAME	CONTACTINFO
1	3227	Ella Kimpton	<?xml version="1.0" encoding="UTF-8" ?><Clie...
2	8877	Chris Bontempo	<?xml version="1.0" encoding="UTF-8" ?><Clie...
3	9077	Lisa Hansen	<?xml version="1.0" encoding="UTF-8" ?><Clie...
4	9177	Rita Gomez	<?xml version="1.0" encoding="UTF-8" ?><Clie...
5	5681	Paula Lipenski	<?xml version="1.0" encoding="UTF-8" ?><Clie...
6	4309	Tina Wang	<?xml version="1.0" encoding="UTF-8" ?><Clie...
7	8015	John Doe	<?xml version="1.0" encoding="UTF-8"?>□□<...

When checking the results of the entire table, you will see that the default value "Standard" has been automatically inserted into the Status column of the new record.

	ID	NAME	STATUS	CONTACTINFO
1	3227	Ella Kimpton	Gold	<?xml version="1.0" encoding="UTF-8" ?><Clie...
2	8877	Chris Bontempo	Gold	<?xml version="1.0" encoding="UTF-8" ?><Clie...
3	9077	Lisa Hansen	Silver	<?xml version="1.0" encoding="UTF-8" ?><Clie...
4	9177	Rita Gomez	Silver	<?xml version="1.0" encoding="UTF-8" ?><Clie...
5	5681	Paula Lipenski	Standard	<?xml version="1.0" encoding="UTF-8" ?><Clie...
6	4309	Tina Wang	Silver	<?xml version="1.0" encoding="UTF-8" ?><Clie...
7	8015	John Doe	Standard	<?xml version="1.0" encoding="UTF-8" ?><Clie...

However, if you omit, for instance, the `Name` column, a warning message ("Editing of data is limited for this statement") is displayed when you select the **Edit Data** command. If you do not want to have this dialog box displayed, select the **Don't show this dialog again!** check box or disable this feature in the SQL Editor

options [430]. After clicking **OK**, you can still edit data in the Result window but the **Append a new row** button and context menu option are disabled.

	ID •	NAME •	STATUS •	CONTACTINFO •
1	3227	Ella Kimpton	Gold	▦ <?xml version="1.0" encoding="UTF-8" ?><C...
2	8877	Chris Bontempo	Gold	▦ <?xml version="1.0" encoding="UTF-8" ?><C...

Copying rows

If you need to duplicate some of the data stored in your database tables, DatabaseSpy provides a feature allowing you to copy data from the result grid and append that content to the grid in a new row. You can either copy an entire row or individual cells. If you copy data from more than one row and append it as new rows, DatabaseSpy will create as many new rows as there were to host the original data.

To copy a row and append it to the result grid:

1. Make sure that the Result window is in the Editing mode [286].
2. In the Result window, right-click into the row you want to copy and choose **Selection | Row** from the context menu. Alternatively, click several individual data cells using **Ctrl+Click**.
3. Right-click anywhere into the Result window and choose **Copy selected cells** from the context menu or click **Ctrl+C**.
4. Right-click anywhere into the Result window and choose **Paste as new rows** from the context menu or click **Ctrl+Shift+V**.

Copying records and using data outside DatabaseSpy

You can use the clipboard to export selected records to any other application that supports the clipboard. It might be helpful to include also the headers of the selected data, even if only a fraction of a record set is copied. The context menu in the Result tab provides the **Copy selected cells with header** command for this purpose. The corresponding header will be copied to the clipboard together with each selected cell.

7.10   Deleting Data

When you need to delete data from database tables or columns, you have the following options:

- Delete all or specific records (rows) from a table. This means that the respective record(s) will be removed completely.
- Reset one or multiple database fields to null or to the default value, in one operation. This means that no database records will be deleted; only the selected fields will be changed. For more information about resetting database fields to their null or default value, see Updating Data [298].

The options to delete database records are available, depending on the case, in the Online Browser, in the Result grid, or in both, as described below.

Deleting records with a DELETE statement

To delete records by generating a DELETE statement:

1. Connect to the database and show the respective data source in the Online Browser.
2. Optionally, right-click the table in the Online Browser and choose **Retrieve data | All rows** to see the data that is contained in the table.
3. Do one of the following:

    - In the Online Browser, right-click the table and select **Show in new SQL Editor | Delete data** from the context menu, or
    - Holding down the right mouse button, drag the table into an open SQL Editor window and choose **Delete data** from the context menu.

    > The **Delete data** command is available for tables only; it is disabled if you right-clicked a column.

    A DELETE statement for the selected table is generated in the SQL Editor and a default WHERE clause is added, for example:
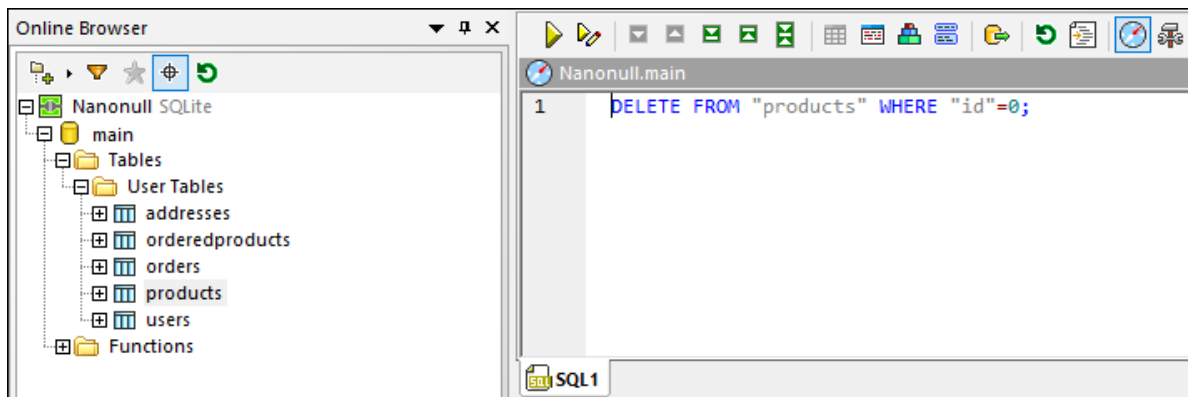
> As further described below, the DELETE statement generated by default is constructed so as to prevent accidental deletion of data, which is intentional. This means that you will need to edit the WHERE clause so that it filters exactly the rows you want to delete.

4. If applicable, edit the WHERE condition so as to select only the rows that you want to delete. To delete all rows, remove the WHERE clause.
5. Click the **Execute** ▷ button to execute the statement and to delete the data from the database.

The DELETE statement generated by default is constructed so as to prevent accidental deletion of data. You will almost certainly need to edit the WHERE clause in order to be able to delete rows. If you would like to delete all rows of the table, just remove the WHERE clause from the DELETE statement.

To help you easily filter the rows of interest, the generated WHERE clause includes the name of the primary key with some pre-filled value, assuming that the table has a primary key. For example, the following is a DELETE statement generated from a SQLite table where a primary key **id** exists:

```
DELETE FROM "addresses" WHERE "id"=0;
```

If the table does not have a primary key, the WHERE condition will include the unified list of all columns of all unique keys. If the table has neither primary nor unique keys, the generated WHERE clause will include all the columns, except for computed columns.
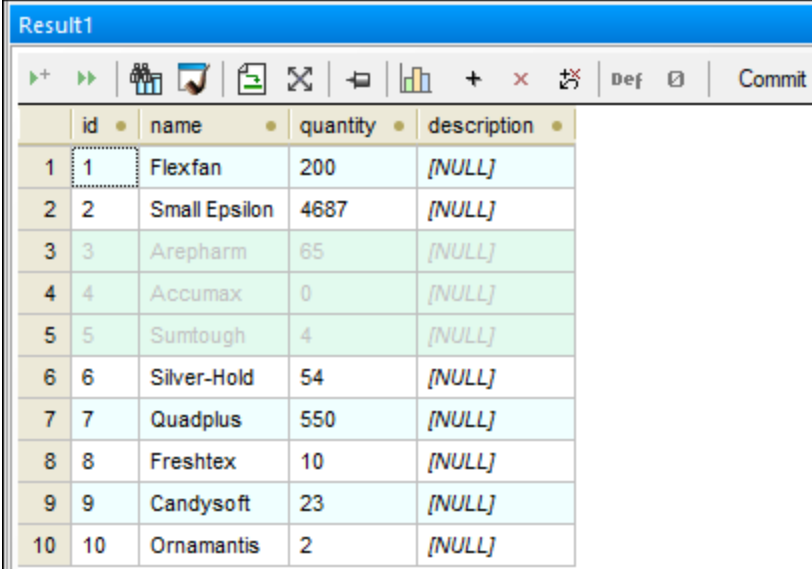
The generated values that appear in the WHERE clause depend on the column's default value, on whether it is nullable, and on the data type of the column. For example, the following is an example of a DELETE statement generated from a SQL Server table where (i) no primary or unique keys exist, (ii) the first column is of **integer** type, and (iii) the second column is of **nvarchar** type:

```
DELETE FROM [dbo].[Events] WHERE [id]=0 AND [event]=N'';
```

Deleting records visually

To delete records visually, from the Result grid:

1. Right-click the table in the Online Browser and choose **Edit Data** from the context menu.
2. Right-click one of the fields in the row you want to delete and then select **Delete Row** from the context menu. Alternatively, click anywhere into the row and then click the **Delete row** ⊠ button. To select multiple rows for deletion, click the numeric identifier to the left of the row while holding the **Ctrl** key pressed, and then click the **Delete row** ⊠ button.

At this stage, the selected rows are marked for deletion, and you can still revert the changes. To revert changes, either right-click the row and select **Undo Changes for this Cell** from the context menu, or

click the **Undo all changes made to the result grid** button.

3. Click **Commit**.

8   Displaying Charts

After retrieving data from a database, you can generate a chart that depicts the selected data. Several types of charts, such as pie chart, bar chart, line chart, area chart, or gauge chart, are available and the appearance of the chart can be customized. Charts are displayed in the Charts Window [35]. This is where you can also change the chart type and appearance, define what data is to be displayed and how, or export the chart to file, clipboard, or printer.

The easiest way to display a chart window in DatabaseSpy is to retrieve data and then click the **Show Data in Chart Window** ⊞ button in the Result window toolbar. This opens a dialog box which lists all the columns that have been retrieved. You can deactivate unwanted columns and change the sequence of the selected columns by dragging them to the desired position in the dialog box.
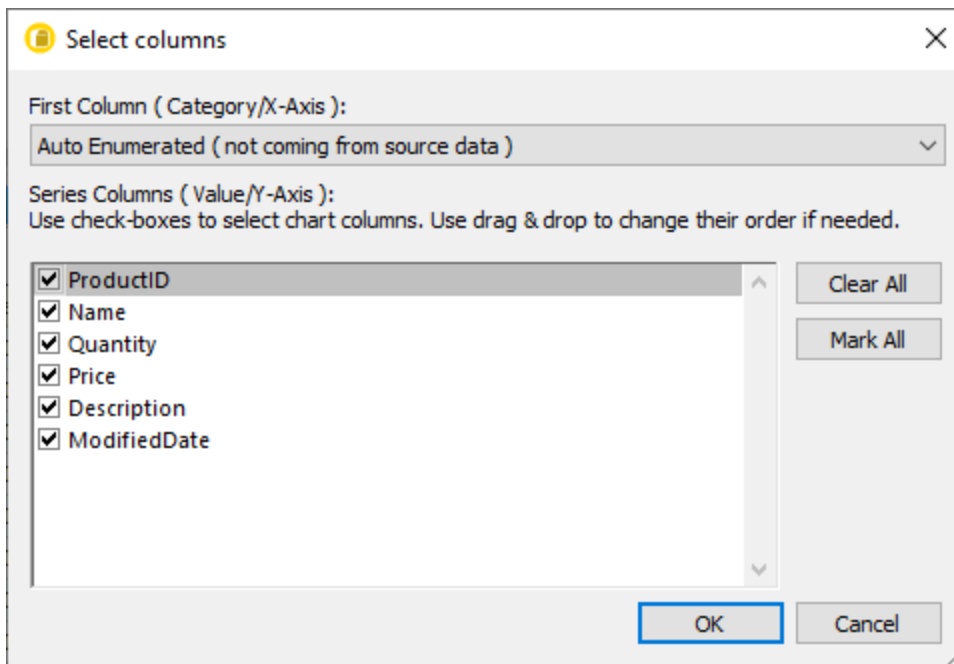
Unchecked columns can later be activated again when you choose the chart data in the Chart Data dialog box.

If you have retrieved all columns of a table but want to display only specific columns in a chart, you can select one or several columns in the Result window before you click the **Show Data in Chart Window** ⊞ button. Then only the selected columns will appear in the Select columns dialog box and only these columns can be used for the chart.

You can choose the X-axis for the chart from the First Column (Category/X-Axis) drop-down list or leave the default setting "Auto Enumerated (not coming from source data)". In the latter case, all available columns are depicted on the Y-axis and an auto-generated value is used for the X-axis.

8.1   About Charts

Charts provide a graphical representation of data in the source document. A chart is set up by defining XPath expressions to specify a sequence of items for each axis of the chart. DatabaseSpy then automatically generates the chart. The table below shows the types of charts that can be created, and the kind of data items that are required for each of the chart's axes.

Chart type	X-Axis (Category)	Y-Axis (Value)	Number of Series (on Z-Axis)
Pie Charts (2D, 3D)	Text	Numeric	1
Bar Charts Ungrouped (2D, 3D)	Text	Numeric	1
Bar Charts Grouped (2D, 3D)	Text	Numeric	> 1
Category Line Graphs	Text	Numeric	1 line = 1 series
Value Line Graphs	Numeric	Numeric	1 line = 1 series
Area and Stacked Area Charts	Text	Numeric	1 area = 1 series
Candlestick Charts	Text	Numeric	3 or 4
Gauge	—	Numeric	1
Overlay Charts	Text	Numeric	= 1 or > 1 per chart

8.2   Chart Data Selection

The Select Data dialog box allows you to add and delete series as well as to define the column or data range that should be illustrated by a particular series. The available columns are displayed in a list box at the bottom of the dialog box. You can add and remove columns by clicking the **Select Columns** button and activate the required columns in the Select columns dialog box that appears (note that columns can only be added to, or removed from, this dialog box via the Result window).
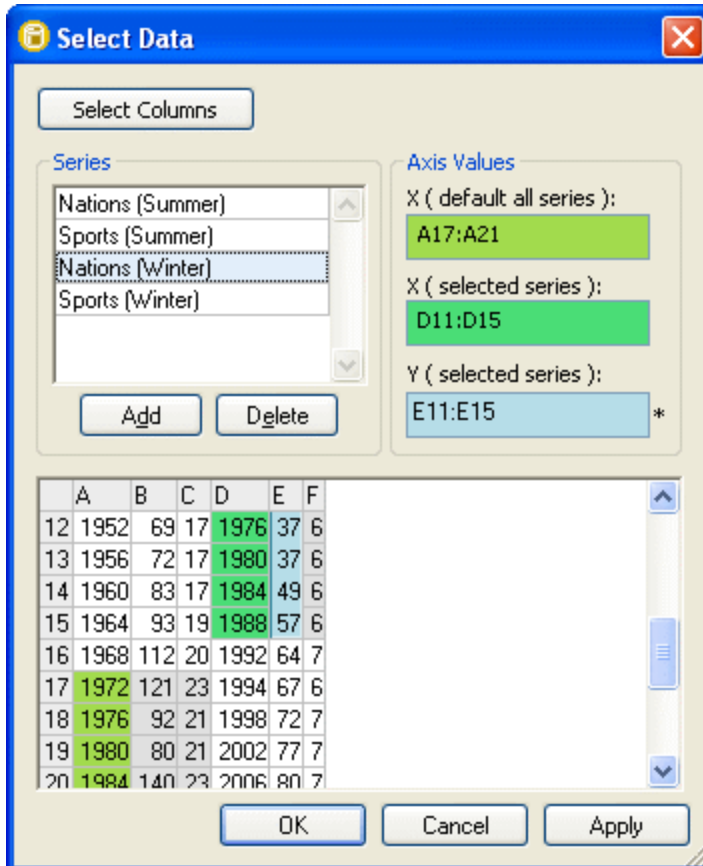
In the Axis Values group box, you can define a column, row, or data range that is used as an X-axis for all series. To specify the column, row, or data range for the Y-axis, select a Series in the Series group box and then click into the Y (selected series) field in the Axis Values group box. An asterisk appears next to the field to mark it as the field that can be edited.

In Value Line Charts, you can add an additional X-axis for the selected series if, for example, the X-axes of two series do overlap but do not match entirely. Then the additional axis can be laid over the default axis, which allows to display the data in the correct place.
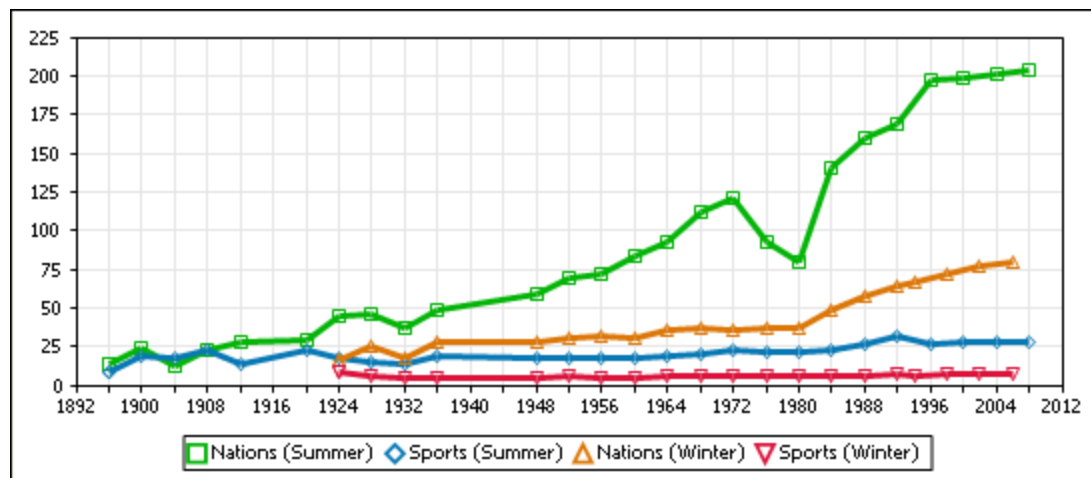
For Candlestick Charts, please note that the sequence of the series is important in order to display the chart correctly. A corresponding note is displayed in the Select Data dialog box.

Adding, deleting, and renaming series

The Series group box shows all the data series that are currently included in the chart. You can rename a series by clicking its name in the group box and enter the new name. A series can be removed from the chart by clicking the **Delete** button. To add a series to the chart, click the **Add** button, enter a descriptive name, and assign a column, row, or data range to the Y-axis.

To add a series to a chart:

1. Make sure that the column you want to display in the chart is listed in the list box at the bottom of the dialog box.
2. Click the **Add** button in the Series group box and give the new series a descriptive name.
3. Click into the Y (selected series) field to select it for editing.
4. In the list box below, click into the column, row, or data range you want to assign to the series (or type the corresponding letter into the Y (selected series) field. You can also select a single value or a value range from the list box.

5. In Value Line Charts, optionally assign an individual X-axis in the same way.
6. Optionally, click the **Apply** button to see a preview of the new series.
7. Click **OK**.

Reloading source data

If the result grid that has been used to create a particular chart has changed since the chart was originally generated, the source data has to be reloaded. This can be done either automatically or by clicking the **Reload** button in the Chart window toolbar. The content of a result grid may change because it is in the Editing mode and the data has been edited—such changes will be depicted in the chart immediately if the **Auto** toggle is activated—or because the result grid has been closed or the underlying SQL script has been changed and/or re-executed.

If the name of a result grid changes, the connection to the chart is broken and DatabaseSpy displays the Select Result grid dialog box when you click the **Reload** button or when automatic reloading occurs.
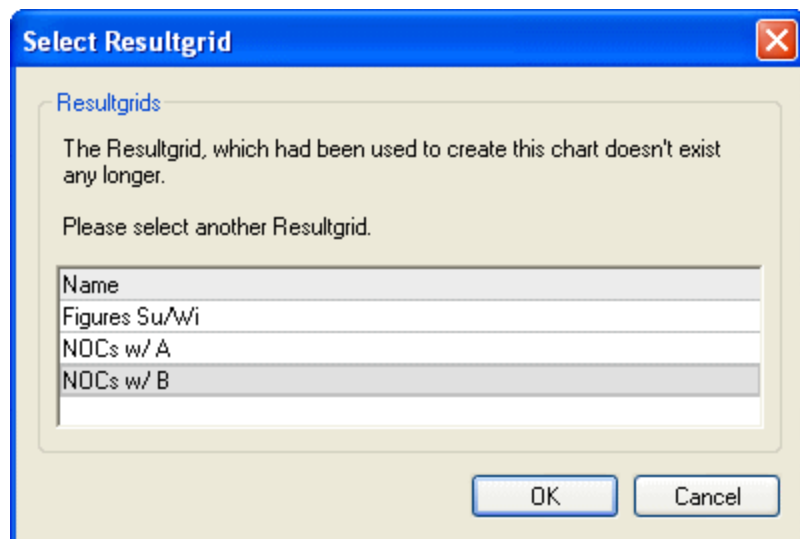
Choose one of the available result grids and click **OK**.

8.3   Exporting Charts

The toolbar of the Charts window provides the **Export** button which, when clicked, opens a sub-menu with the following options:

- **Save chart in file...**: Opens the Image Size dialog box where you can enter the height and width of the image in pixels. Upon clicking **OK**, the Save As dialog box is displayed where you can specify a name and path for the image file and choose from among several image types (such as PNG, BMP, GIF, and JPG).
- **Copy current image to clipboard**: Copies an image of the chart to the clipboard; the content of the clipboard can then be pasted into an external application such as Paint. Note that the chart will be copied exactly as it is displayed in the Charts window, so you would have to adjust the size before you execute the copy or use the option **Copy resized image to clipboard**.
- **Copy resized image to clipboard...**: Opens the Image Size dialog box where you can enter the height and width of the image in pixels. Upon clicking **OK**, the image is resized before it is copied to the clipboard.
- **Print this chart...**: Sends the content of the Charts window to a printer.

Printing a chart

When printing charts in DatabaseSpy, you can specify how much of the page size the chart should occupy. After you have selected the **Print this chart...** command a Print dialog box is displayed where you can enter the percentage of page height and width.
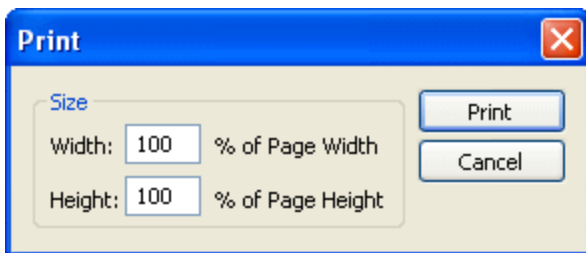
After you have clicked **Print**, the Print dialog box for your printer is displayed where you can change the properties (orientation, paper format, etc.) if needed. The chart will be printed with the name of the result grid where the data originates from, DatabaseSpy logo, date and time in the header.

Some chart settings require transparency for drawing the chart which may cause problems on printing. You can automatically set the following options to printing compatible settings before the print is started:

- Transparency for pie charts and area charts: transparency will be set to 0 for printing
- Dropshadow on nonsolid background in 3d pie charts: dropshadow will be deactivated for printing

A corresponding dialog box pops up where you can choose to change the settings or print the chart with the existing settings.

8.4   Chart Overlays

For some chart types (Bar Chart, Stacked Bar Chart, Line Chart, Area Chart, Stacked Area Chart, and Candlestick Chart), you can display several charts as overlays in one Charts window. This way you can combine charts of different type containing data even from different databases in one Charts window.
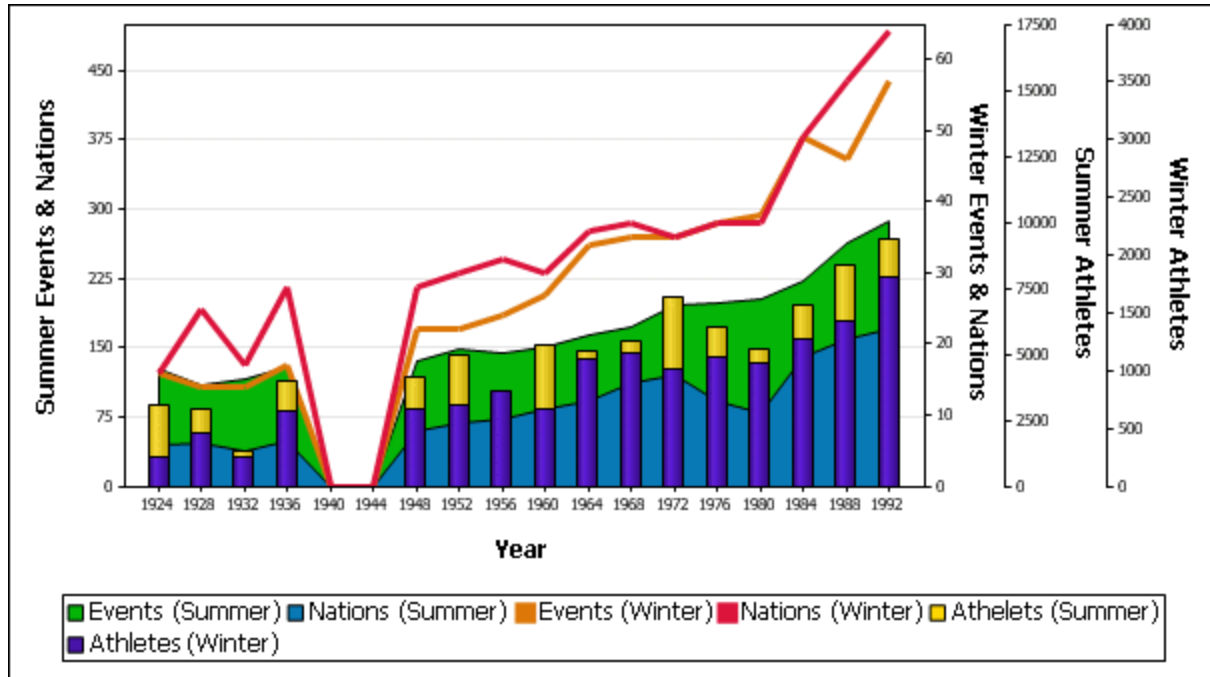
A separate Y axis will be displayed for each overlay chart and can be labeled. Note that an overlay will always be displayed in front of the existing chart(s). You should, therefore, be careful if you combine different chart types in one Charts window as, for example, area charts displayed as overlay may hide bar or line charts. By default, subsequent colors from the color schema of the Main chart will be used for the overlay. You can, however, also deactivate the **Use subsequent colors from previous chart layer** radio button in any layer and define a different color schema for this particular layer. Note that this color schema will be used for subsequent layers if these have the **Use subsequent colors from previous chart layer** radio button activated.

Chart overlays are managed by clicking the **Overlays** button in the menu bar of the Charts window; this will open the **Overlays** menu where you can add and remove overlays and select the active chart. Note that any changes you make to chart type and/or appearance, and in the Select data dialog box apply only to the active chart.

Tip: You can use [targets](#)[373] for the SQL statements that retrieve the source data of the charts; this way it is easier to identify the various layers in the **Overlays** menu.

```
   Main ( Area Chart from Summer Games )
✓  Layer 2 ( Line Chart from Winter Games )
   Layer 3 ( Bar Chart from Result1 )
   Layer 4 ( Bar Chart from Result1 )

   Append
   Delete Current
```

To create a chart overlay:

1. Create a chart (the Main Chart).
2. In the Charts window, click the **Overlays** button and choose **Append** from the drop-down menu.
3. Do one of the following:

   - Create a new chart from the existing Result window.
   - Create a new query in an SQL Editor window or from the Online Browser and click the **Show data in Chart window** button.

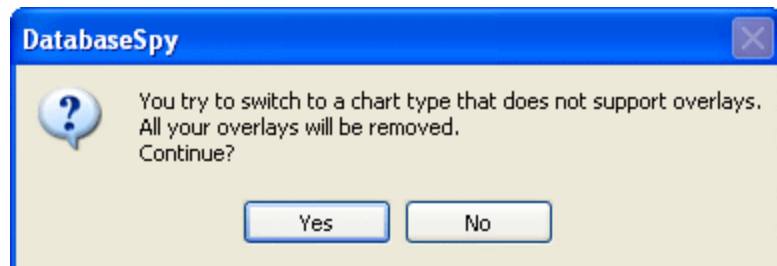   The new chart is displayed as overlay of the existing chart(s) in the same Charts window.

4. In the Change Type dialog box, which appears automatically for the first overlay of each tab in the Charts window, define the type of the overlay chart.
5. Optionally, change the appearance of the overlay chart.

To remove a chart overlay:

1. In the Charts window, click the **Overlays** button and, in the drop-down menu, select the chart that you want to remove.
2. Select **Delete Current** from the **Overlays** menu.

Changing the chart type of the Main chart

When you have selected the Main chart as the active chart, you can change the chart type also to types that do not support overlays (i.e., pie charts and gauge charts as well all types of 3d charts). In such a situation all overlays will be removed from the Charts window and DatabaseSpy displays a warning message.

```
DatabaseSpy
┌──────────────────────────────────────────────────────────────┐
│ ?   You try to switch to a chart type that does not support overlays. │
│     All your overlays will be removed.                         │
│     Continue?                                                  │
│                                                                │
│              [  Yes  ]      [  No  ]                           │
└──────────────────────────────────────────────────────────────┘
```

8.5   Chart Settings and Appearance

Chart settings are organized as follows:

- [Basic Chart Settings](#) [318]: The most basic setting is the chart type. To select the chart type, click **Change Type** in the toolbar of the chart window. The [Change Type dialog](#) [318] is displayed.
- [Advanced Chart Settings](#) [323], which enable you to change the appearance of a chart (its title, legend, colors, fonts, etc). Advanced settings are defined in the [Change Appearance dialog](#) [323]. To access this dialog, click **Change Appearance** in the toolbar of the chart window.

8.5.1   Basic Chart Settings

This section:

- [Setting the chart type](#) [318]
- [List of chart types](#) [318]
- [Other basic settings](#) [323]

Setting the chart type

The most basic chart setting is the chart type. To select the chart type, click **Change Type** in the toolbar of the chart window.

Chart types

The various types of charts that are available are listed below. In the [Change Type dialog](#) [318] (*screenshot above*), select the chart type you want and click **OK**.

▼ *Pie charts*

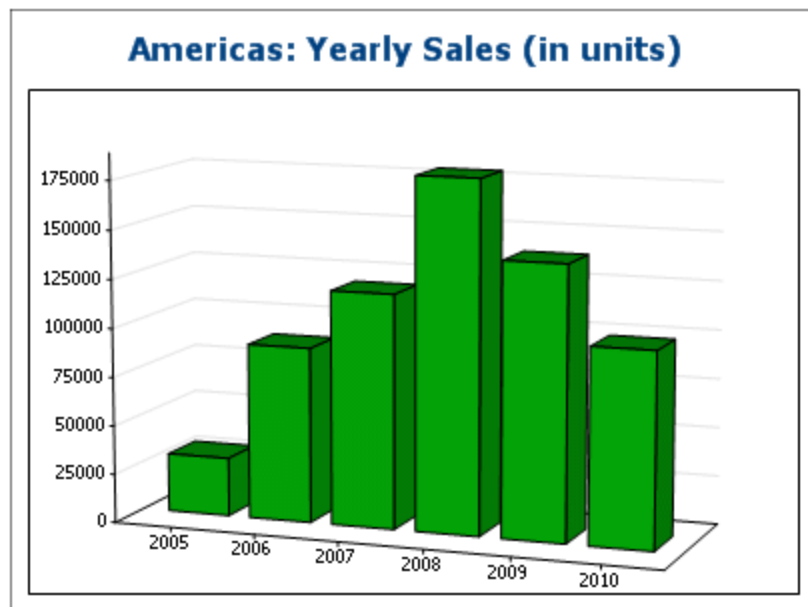   In pie charts, one column/axis provides the values, another column/axis provides labels for these values. The labeling column/axis can take non-numeric values.
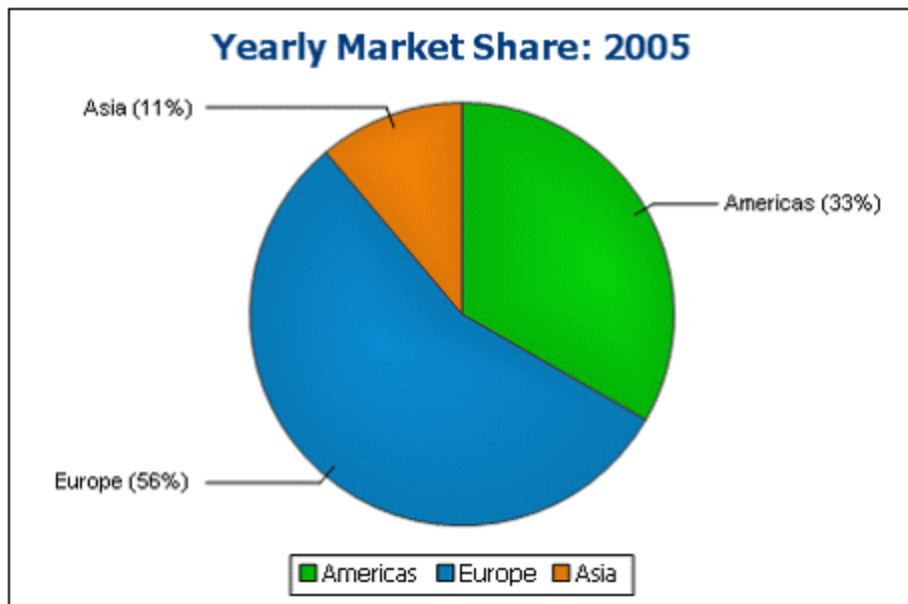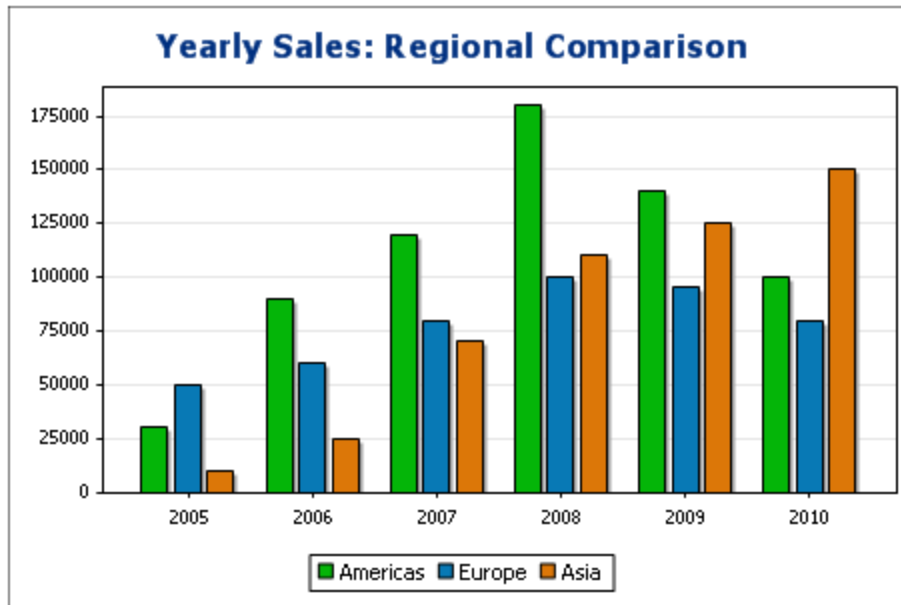
▼ *Bar charts*

Bar charts can have two sets of values used along two axes (*below*).

They can also use three sets of values, as in the example below: (i) continent, (ii) year, (iii) sales volume. Bar charts can be displayed in 2D (*below*) or 3D (*above*).

A three-axis bar chart can also be stacked if you need to show totals. Compare the stacked chart below with the chart above. The stacked chart shows the total of sales on all continents.
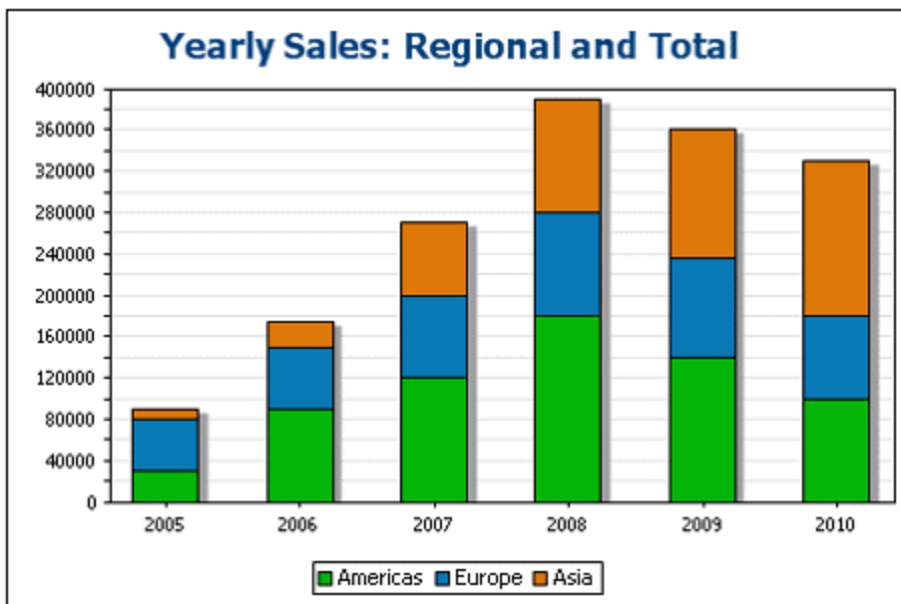
▼ *Line charts*

The difference between a line chart (*below left*) and a value line chart (*below right*) is that value line charts only take numerical values for the X-axis. If you need to display line charts with text values on the X-axis, use line charts.
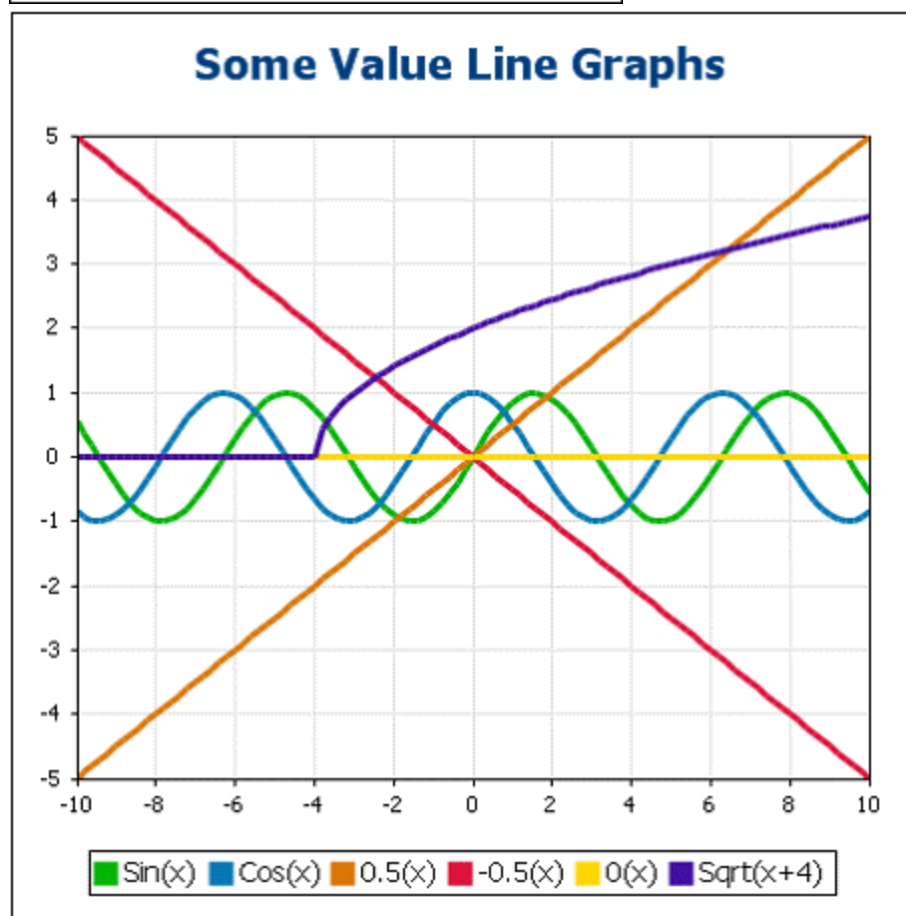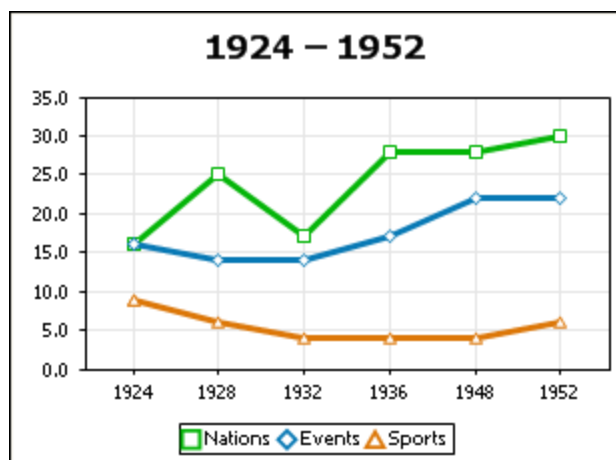
▼ *Area charts*

Area charts are a variation of line charts, in which the areas below the lines are also colored. Note that area charts can also be stacked (*see bar graphs above*).
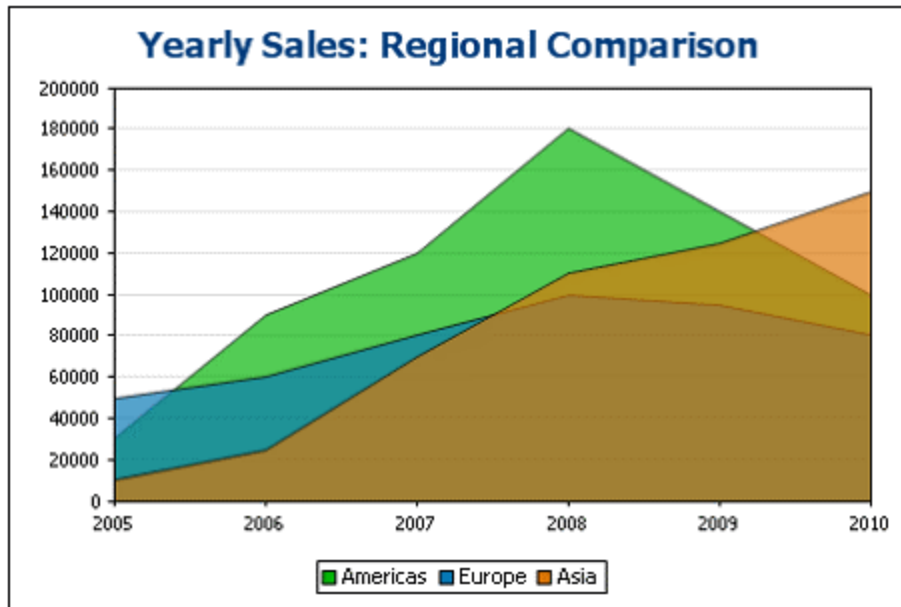
▼ *Candlestick charts*

A candlestick chart can be used to depict price movements of securities, commodities, currencies, etc over a period of time. The chart indicates not only how prices developed over time, but also the daily close, high, low, and (optionally) open. The Y-axis takes three or four series (close, high, low, and (optionally) open). The screenshot below shows a four-series candlestick chart.
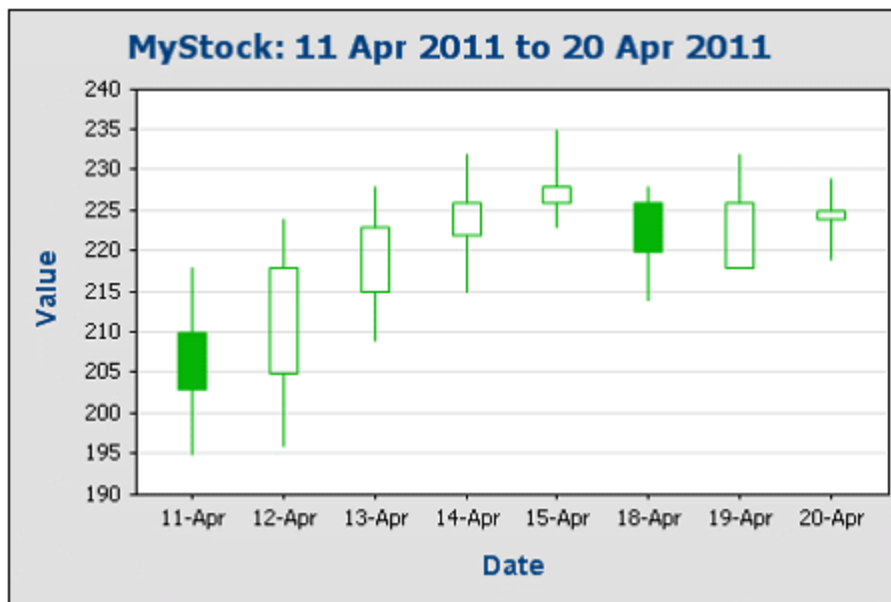
▼ *Gauge charts*

Gauge charts are used to illustrate a single value and show its relation to a minimum and a maximum value.

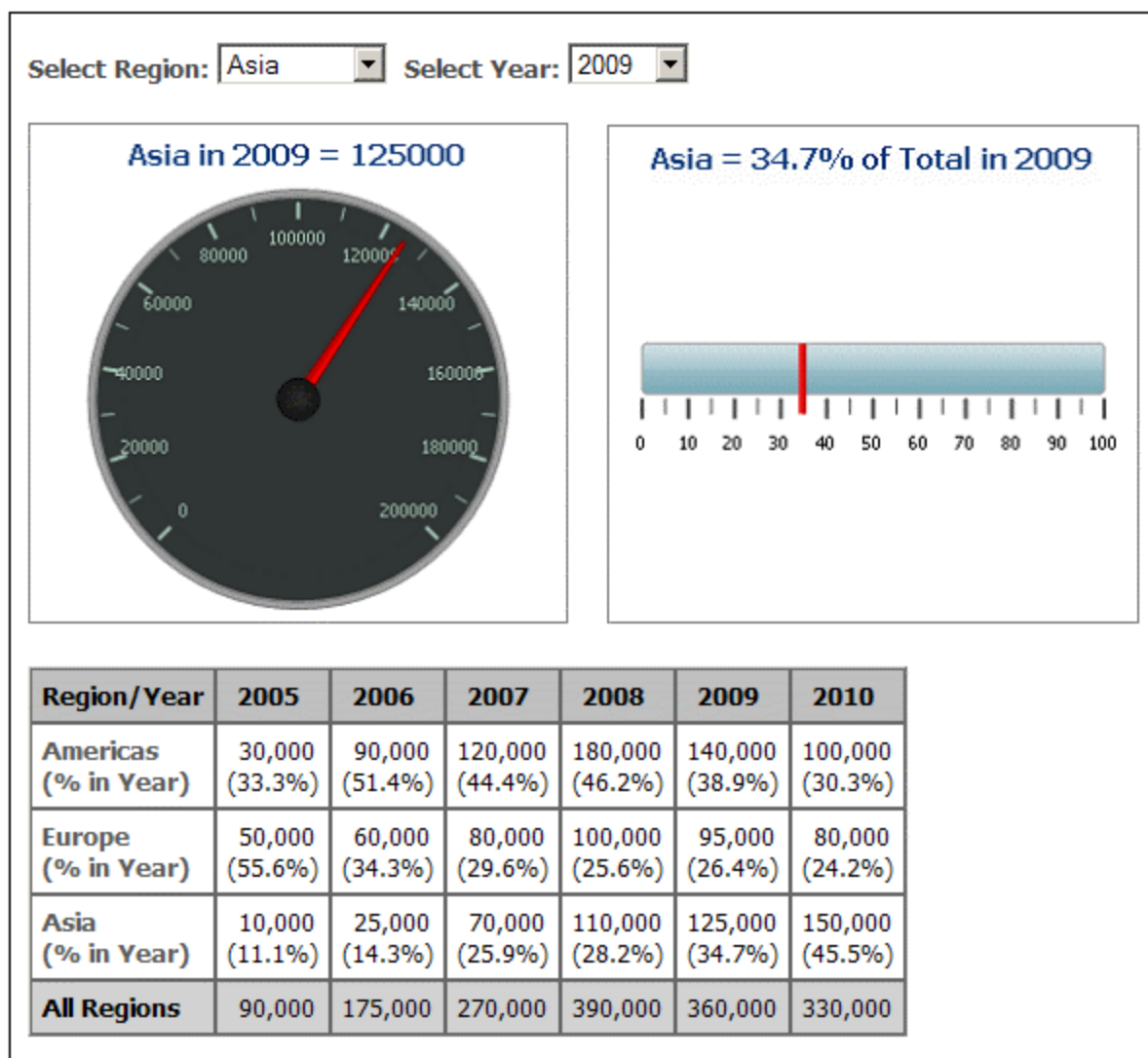| Region/Year	2005	2006	2007	2008	2009	2010
Americas (% in Year)	30,000 (33.3%)	90,000 (51.4%)	120,000 (44.4%)	180,000 (46.2%)	140,000 (38.9%)	100,000 (30.3%)
Europe (% in Year)	50,000 (55.6%)	60,000 (34.3%)	80,000 (29.6%)	100,000 (25.6%)	95,000 (26.4%)	80,000 (24.2%)
Asia (% in Year)	10,000 (11.1%)	25,000 (14.3%)	70,000 (25.9%)	110,000 (28.2%)	125,000 (34.7%)	150,000 (45.5%)
All Regions	90,000	175,000	270,000	390,000	360,000	330,000

Other basic settings
In the Chart Settings pane, you can also set the title of the chart (*see screenshot below*).
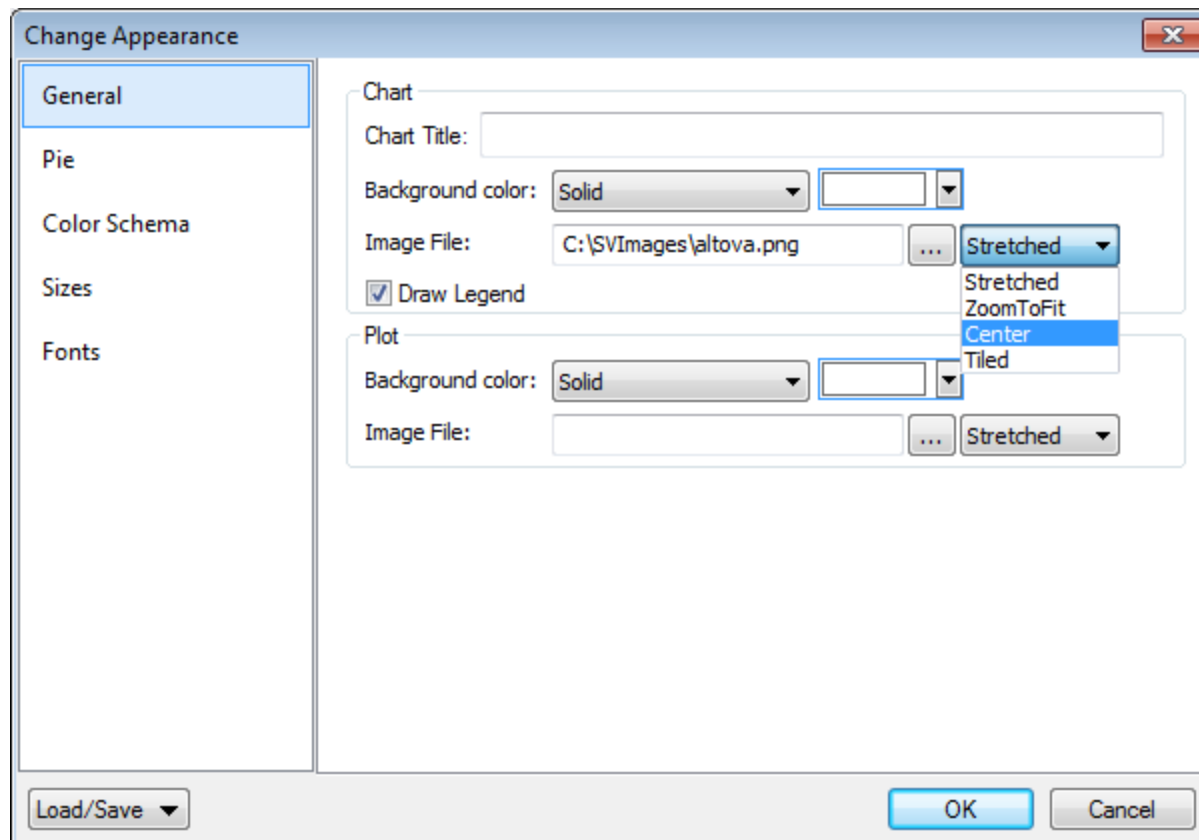
8.5.2   Advanced Chart Settings

This section:

- Accessing the advanced settings [324]

-
-

Accessing the advanced settings

To access a chart's advanced settings do the following: Click **Change Appearance** in the toolbar of the chart window. This displays the Change Appearance dialog for that particular chart type (*the screenshot below shows the Change Appearance dialog of a pie chart*).


Overview of advanced settings

The advanced settings are organized into tabs that are common to all chart types and those that are specific to a single chart type.

Common chart settings

▼ *General*

The chart title (*see screenshot below*) is the same as the basic setting (*see above*) and can be edited as an advanced setting also. Other settings in this dialog are the background color of the chart and the plot. In the screenshot below, the plot has been given a pale green background color. An image file can also be set as the background image of the chart and/or the plot. This image can be stretched to cover the entire area of the chart or plot; zoomed to fit so that the zoom matches one of the two dimensions (of chart/plot); centered; or tiled. The legend is the key to the color codes in the chart, and it can be turned on or off.
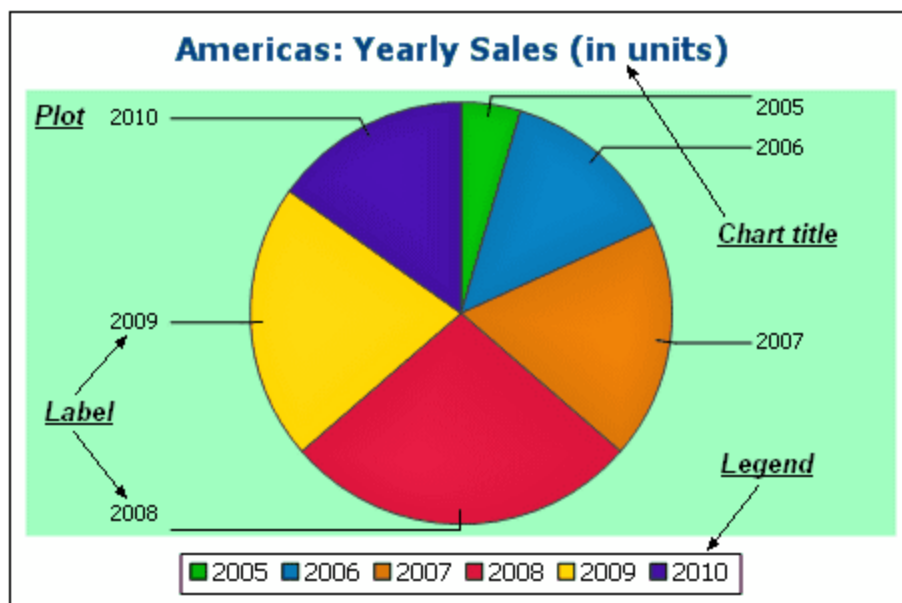
▼ *Color scheme*

Four predefined color schemes are available plus a user-defined color scheme. You can modify any of the color schemes by adding colors to and/or deleting colors from a scheme. The color scheme selected in this tab will be used in the chart.

▼ *Sizes*

Sizes of various aspects of the chart can be set, either as pixels or as a percentage ratio.

▼ *Font*

The font properties of the chart title and of legends and labels can be specified in this tab. Sizes can be set as a percentage of the chart size or absolutely as points.

▼ *Load/Save button*

Settings can be saved to an XML file and can be loaded from an XML file having the correct structure. To see the structure, save the settings of a chart and then open the XML file. Clicking this button also gives you the option of resetting chart settings to the default.

Type-specific chart settings
▼ *Pie charts*

Settings for: (i) the angle from which the first slice should be drawn; (ii) the direction in which slices should be drawn; (iii) the outline color; (iv) whether the colors receive highlights (in 3D pie charts: whether

dropshadows and transparency are used); (v) whether labels should be drawn; and (vi) whether values and percentages should be added to labels and how many decimal places should be added to the percentages.

▼ *Bar charts*

Settings for: *(General)* Drawing the X and Y axes exchanged generates a horizontal bar chart; *(Bar)* Bar outlines and dropshadows (dropshadows in 2D bar charts only); *(X-Axis)* Label and color of the x-axis, and vertical gridlines; *(Y-Axis)* Label and color of the y-axis, horizontal gridlines, the range of values to be displayed, and the tick marks on the y-axis; *(Z-Axis, 3D only)* Label and color of the z-axis; *(3D)* the vertical tilt, horizontal rotation, and the width of the view.

▼ *Line graphs*

Settings for: *(General)* Drawing the X and Y axes exchanged; *(Line)* including the plot points or not; *(X-Axis)* Label and color of the x-axis, and vertical gridlines; *(Y-Axis)* Label and color of the y-axis, horizontal gridlines, the range of values to be displayed, and the tick marks on the y-axis.

▼ *Gauge charts*

Settings for: (i) the angle at which the gauge starts and the angular sweep of the scale; (ii) the range of the values displayed; (iii) the interval and color of major and minor ticks; (iv) colors of the dial, the needle, and the border.

▼ *Area charts*

The transparency of areas can be set as a value from 0 (no transparency) to 255 (maximum transparency). In the case of non-stacked area charts transparency makes parts of areas that lie under other areas visible to the viewer. Outlines for the areas can also be specified.

▼ *Candlestick charts*

The fill color can be specified for the two situations: (i) when the closing value is greater than the opening value, and (ii) when the opening value is greater than the closing value. In the latter case, the Series color is also available as an option. The Series color is specified in the Color Schema tab of the Change Appearance dialog.

Loading, saving, resetting chart settings

Chart settings that are different from the default settings can be saved in an XML file. These settings can subsequently loaded as the settings of a chart, which can help you save time and effort. The **Load/Save** button (see first screenshot in this section[323]) provides the following options when clicked:

- **Set to default**: Rejects changes made to the settings, and restores the default settings to **all** settings sections.

- **Load from file**: Enables settings to be imported that have been previously saved in an XML file (*see next command*). The command displays the Open dialog, in which you enter the location of the required file.
- **Save to file**: Opens the Save As dialog box. You can specify an XML file in which to save the settings. This file lists those settings that are different from the default settings.

8.5.2.1 General

The General section of the Change Appearance dialog box lets you define the title of the chart, add or remove a legend, and define background pictures and colors and—for bar, line, area, and candlestick charts—orientation of the chart.


Chart

Enter a descriptive title for your chart into the `Chart Title` field and select a background color for the entire chart from the drop-down list. You can choose a solid background, vertical gradient, or horizontal gradient and define start and end colors for the gradient, if applicable. In addition, or instead of a colored background, you can also define a background image and choose one of the available display options from the drop-down list:

- Stretched: the image will be stretched to the height and width of the chart
- Zoom to Fit: the image will be fit into the frame of the chart and the aspect ratio of the image will be maintained
- Center: the image will be displayed in its original size in the center of the chart
- Tiled: if the image is smaller than the chart, duplicates of the image will be displayed to fill the background area

The `Draw Legend` check box is activated by default, clear the check box if you do not want to display a legend in your chart.

Plot

The Plot is the area where the actual data of the chart is displayed. You can draw a border around the plot and specify a different background color and/or image for the plot area. In the screenshot below, the background color of the chart has been changed to gray (vertical gradient) whereas the plot is still white, a red border has been drawn around it, and a background image has been added.
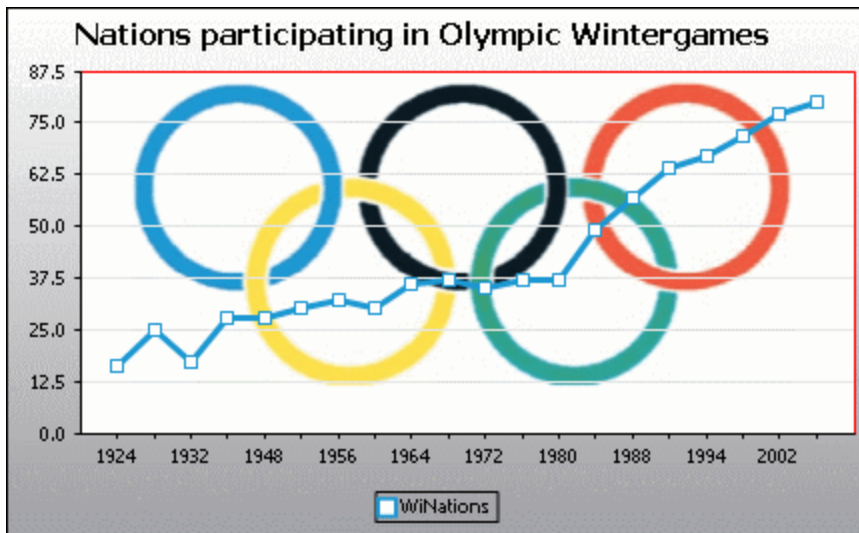
Orientation

If you have a small series of large values it may be convenient to swap the X and Y axis for a better illustration. Note that in the screenshot below also the background color of the plot has been set to "Transparent" and the background image has been applied to the chart.
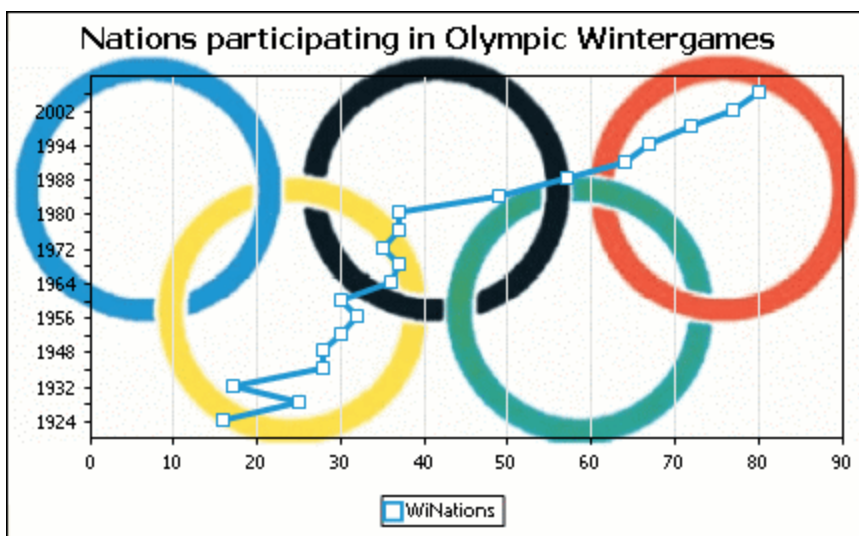
Note that this option is not available for pie and gauge charts.

8.5.2.2 Type-Related Features

For each of the chart types, and even for the various sub-types, the Change Appearance dialog box provides a section where you can define the type-related features of the chart.

Pie chart

Most settings are the same for the 2d and 3d versions. In 2d pie charts, you can additionally draw highlights.

The `Start Angle` value defines where the first row of the selected column will be displayed in the chart. An angle of 0 degrees corresponds to 12 o'clock on a watch.

You can show labels in addition to, or instead of, the legend, add values and/or percentage to the labels, and define for the percentage values the number of decimal digits to be displayed.

The color that you can select next to the `Draw Outline` check box is used for the optional border drawn around the chart and the individual pie segments. The `Clockwise` check box allows you to specify whether the rows should be listed clockwise or counter-clockwise.

In 3d pie charts, you can draw a drop shadow and define its color, add transparency to the chart, and define the 3d tilt. In 2d charts, the `Draw Highlights` option adds additional structure to the chart.

In 3d pie charts, you can display drop shadows, add transparency and define the 3d tilt.

Bar chart


For bar charts, you can make the following setting:

- Add an outline to the bars and define its color.
- In 2d bar charts, you can also draw a drop shadow and define its color (this option is not available for 3d bar charts).
- By default, the shape of the bars resembles a cylinder, however you can also choose "Vertical Gradient" or "Solid" from the `Fill style` drop-down list (this option is available for 2d bar charts only).
- The values of a bar (corresponding to the height of the bar on the Y-axis) can be drawn on the bar. The font of the values can be specified in the *Fonts* settings (this option is available only for 2d bar charts, not stacked).
- The distance between the series of a bar-group and between bar-groups can be specified as a decimal fraction of the width of a single bar. For example, in the screenshot below, which shows bar-groups that each consist of a blue series and a green series, the distance between the series has been set to a 25% (=`0.25`) of the width of a bar; the distance between bar-groups has been set to 100% (=`1.0`) of the width of a bar. This option is available only for 2d bar charts.
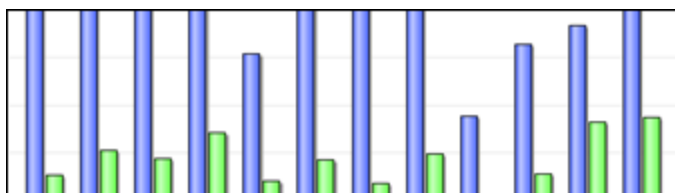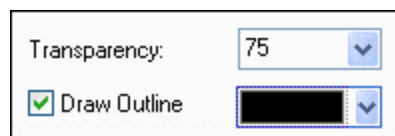
Line chart
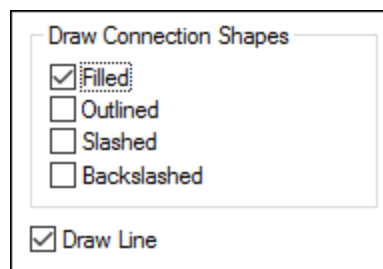
To draw connection shapes that mark the values in line charts, you need to activate at least one check box in the Draw Connection Shapes group box. You can use five different shapes to mark a series: square, rhomb, triangle, inverted triangle, and circle. If there are more than five series in your chart you can combine the connection shapes by selecting more than one option in the Draw Connection Shapes group box. In the screenshot below, both `Filled` and `Slashed` have been selected and the `Slashed` type is used for the sixth series and beyond.

The *Draw Line* option enables the graph to be drawn with (i) only connection shapes, or (ii) with connection shapes joined by a line.
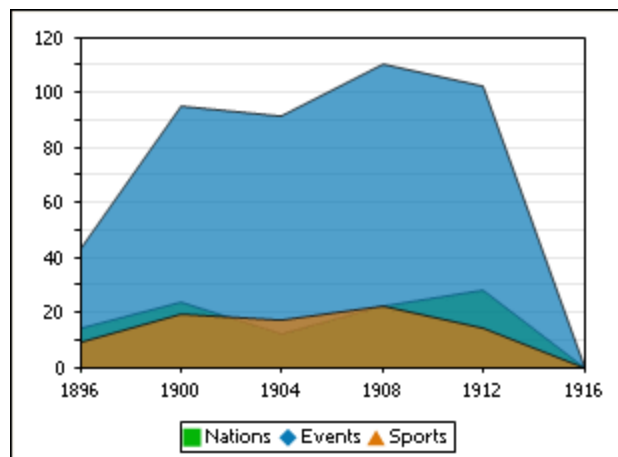

Connection shapes are available for both line charts and value line charts.

Area chart

Among the properties that you can change for area charts is transparency; this way you can prevent that one series is hidden by another series in the chart. In addition, you can add an outline to the individual data areas and define its color (*see screenshot below*).

Candlestick chart

If both opening and closing value are defined as series, you can choose the colors and whether or not the candle should be filled if the closing value is greater than the opening value.

Gauge chart

The `Start` value in the Angles group box defines the position of the 0 mark and the `Sweep` value is the angle that is used for display. In the Value Range group box you can define the minimum and maximum values to be displayed. Tick marks are displayed with (major ticks) or without (minor ticks) the corresponding value; you can define separate colors for them. In the Colors group box you can define colors for the dial fill, needle, needle base (hides the first part of the needle in the center of the chart), and the border that surrounds the chart. The current value and an extra label can be shown at any angle you like.

8.5.2.3 Colors

Depending on the chart type you have selected, DatabaseSpy provides two different sections for the definition of colors to be used in charts:

- Color Schema for pie, bar, line, area, and candlestick charts
- Color Range for gauge charts

Color Schema

The Color Schema section of the Change Appearance dialog box provides four predefined color schemas (i.e., default, grayscale, colorful, and pastel) that can be customized; in addition you can also define your own color schema from scratch.
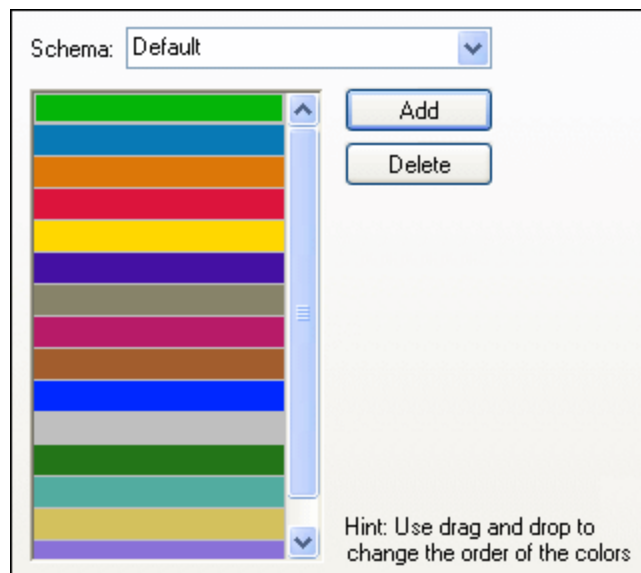
The top color will be used for the first series, then the second color and so on. You can change the order of the colors by selecting a color and dragging it to its new position with the mouse. To add a new or delete an unwanted color, click the corresponding button. In candlestick charts, only the first color will be used.

If you have appended one or several layers of overlay charts to a Charts window, the Color Schema section of the Change Appearance dialog box contains the additional radio button `Use subsequent colors from previous chart layer` which is activated by default.
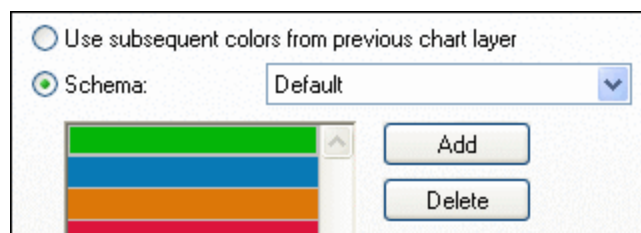
When the radio button is activated, the color schema from the previous layer will be used and you cannot choose a separate color schema for the overlay. The series of the active layer will be displayed using subsequent colors from the color schema of the previous layer. This way, all series of the Charts window have different colors and can therefore be distinguished more easily.
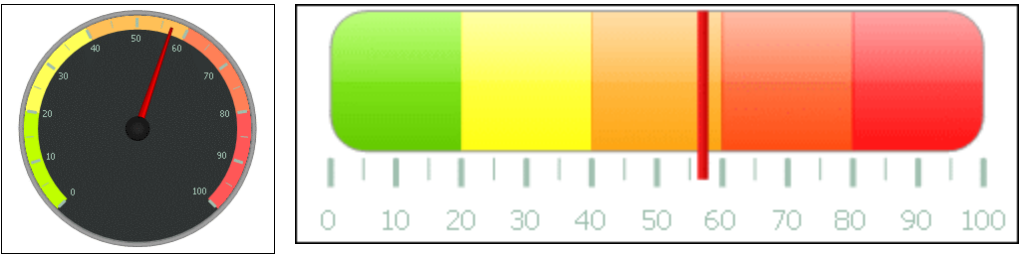
You can break this link on any additional layer that you add and choose a different color schema that then can also be re-used in subsequent layers.

Color Range

In gauge charts, you can customize the appearance of the gauge by applying colors to certain value ranges.

The definition shown in the screenshot above will appear in the gauge charts as follows:

8.5.2.4 X-Axis

In the X-axis section of the Change Appearance dialog box, you can enter a label for the axis, and define colors for the axis and the grid lines (if displayed). You can also define whether or not you want to display tick marks and axis values. This section is the same for all bar, line, area, and candlestick charts. The *Show Categories* options enables you to specify that only a subset of all categories (X-Axis values) are displayed, that is, only the ticks, grid lines, and values of the selected categories will be displayed. Create the subset of displayed categories by entering (i) the index of the first value to display, and (ii) the number of indices to step. For example, if there are 101 categories, from `1900, 1901, 1902 ... 1999, 2000`, then you can show every tenth year from `1900` to `2000` by setting *First index* to `1` and *Step* to `10`.

Label

Line

Show Categories
This allows you to define for which categories the ticks, gridlines and values should be shown.
Can be used if you have more data points than you want to see in the legend.

First index: 1   Step: 1

Gridlines
☐ Show Gridlines

Tick Drawing
☑ Show Ticks
☑ Show Values

In Value Line Charts however, you can also define the value range, and define at what interval tick marks should be displayed.

Label

The text entered into the Label field will be printed below the axis as a description of the X-axis.

Range

By default, the Auto radio button is selected in the Range group box. If you want to display a fragment of the chart in greater detail, activate the Manual radio button and enter minimum and maximum values into the respective fields. If the column that is used for the X-axis does not include zero, you can deactivate the Include Zero check box and the X-axis will start with the minimum value that is available in the series. The Invert Axis option enables you to invert the values of the X-Axis. For example, if the values run from the 0 to 360, selecting this option will generate the X-Axis so that 360 is at the origin and the values progress down to 0 as the X-Axis goes upwards.

Line

The axis is displayed in the color that you choose from the Line drop-down-list. You can use one of the preselected colors, or click the **Other color...** button to choose a standard color or define a custom color. Click the **Select...** button on the Custom tab and use the pipette to pick a color that is displayed somewhere on your screen.

Grid lines

If the Show Grid lines check box is activated, you can choose a color from the corresponding drop-down list box.

Tick Interval
If you are not satisfied with the default tick marks, you can activate the `Manual` radio button in the Tick Interval group box and enter the difference between the individual tick marks into the corresponding field.

Tick Drawing
You can switch the display of tick marks on the axis and/or axis values on or off.

Axis Position
From the drop-down list, you can choose the position where the axis is to be displayed. When selecting "At Value / On Category Number", you can also position the axis anywhere within the plot.

8.5.2.5 Y-Axis

In the Y-axis section of the Change Appearance dialog box, you can enter a label for the axis, define colors for the axis and the grid lines (if displayed), define the value range, and decide if and where tick marks should be displayed and whether or not you want to show the axis values. This section is the same for all bar and line charts.

Label
The text entered into the `Label` field will be printed to the left of the axis as a description of the Y-axis.

Range
By default, the `Auto` radio button is selected in the Range group box. If you want to display a fragment of the chart in greater detail, activate the `Manual` radio button and enter minimum and maximum values into the respective fields. If the column that is used for the Y-axis does not include zero, you can deactivate the `Include Zero` check box and the Y-axis will start with the minimum value that is available in the series. The `Invert Axis` option enables you to invert the values of the Y-Axis. For example, if the values run from the `0` to `360`, selecting this option will generate the Y-Axis so that `360` is at the origin and the values progress down to `0` as the Y-Axis goes upwards.

Line
The axis is displayed in the color that you choose from the `Line` drop-down-list. You can use one of the preselected colors, or click the **Other color...** button to choose a standard color or define a custom color. Click the **Select...** button on the Custom tab and use the pipette to pick a color that is displayed somewhere on your screen.

Grid lines
If the `Show Grid lines` check box is activated, you can choose a color from the corresponding drop-down list box.

Tick Interval
If you are not satisfied with the default tick marks, you can activate the `Manual` radio button in the Tick Interval group box and enter the difference between the individual tick marks into the corresponding field.
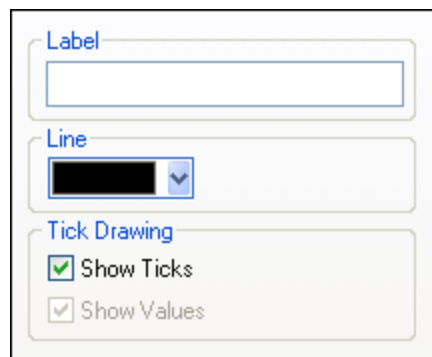
Tick Drawing
You can switch the display of tick marks on the axis and/or axis values on or off.

Axis Position
From the drop-down list, you can choose the position where the axis is to be displayed. When selecting "At Value / On Category Number", you can also position the axis anywhere within the plot.

8.5.2.6 Z-Axis

In the Z-axis section of the Change Appearance dialog box, you can enter a label for the axis, define colors for the axis, and decide whether or not you want to show tick marks on the axis. This section is the same for all 3d bar charts (Bar Chart 3d and Bar Chart 3d Grouped).

Label
The text entered into the Label field will be printed to the right of the axis as a description of the Z-axis.

Line
The axis is displayed in the color that you choose from the Line drop-down-list. You can use one of the preselected colors, or click the **Other color...** button to choose a standard color or define a custom color. Click the **Select...** button on the Custom tab and use the pipette to pick a color that is displayed somewhere on your screen.

Tick Drawing
You can switch the display of tick marks on the axis on or off.

8.5.2.7 3D Angles

In 3d bar charts you can customize the 3d appearance of the chart in the 3d Angles section of the Change Appearance dialog box.
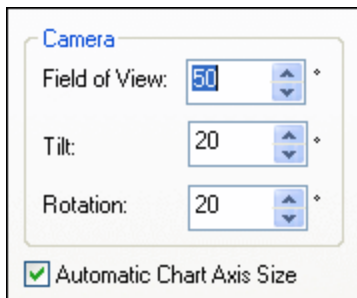
The **Field of view** option causes the diagram to appear as if observed from a small or great distance. Values ranging from 1 through 120 are valid. Higher values cause the diagram to appear as if observed from a greater distance.

The **Tilt** value determines the rotation around the X-axis, whereas the **Rotation** value defines the rotation around the Y-axis. You can automatically adapt the size of the chart axis to the Chart window width by selecting the corresponding check box.

If the **Automatic Chart Axis Size** check box is selected, DatabaseSpy will automatically calculate the optimum size of the X-axis as well as the Y-axis for the current Chart window size. The width and height of the chart will change dynamically when you resize the Chart window.

8.5.2.8 Sizes

In the Sizes section of the Change Appearance dialog box, you can define different margins as well as the size of axis and gauge ticks. Note that not all the properties listed below are available for all chart types.

General
  Outside margin   Space between the plot and the edge of the Chart window.

Title to Plot	Space between the chart title and the upper edge of the plot.
Legend to Plot	Space between the lower edge of the plot and the legend.

Pie

Plot to Label	In pie charts, the space between the most left and right edge of the pie and its labels.
Pie Height	In 3d pie charts, the height of the pie.
Pie Drop Shadow	In 3d pie charts, the length of the shadow (if it is activated in the Pie section).

X-Axis

X-Axis to Axis Label	In bar and line charts, the space between the X-axis and its label.
X-Axis to Plot	In 2d bar charts and line charts, the space between the X-axis and the plot.
X-Axis Tick Size	In bar and line charts, the length of the ticks on the X-axis.

Y-Axis

Y-Axis to Axis Label	In bar and line charts, the space between the Y-axis and its label.
Y-Axis to Plot	In 2d bar and line charts, the space between the Y-axis and the plot.
Y-Axis Tick Size	In bar and line charts, the length of the ticks on the Y-axis.

Z-Axis

Z-Axis to Axis Label	In 3d bar charts, the space between the Z-axis and its label.
Z-Axis Tick Size	In 3d bar charts, the length of the ticks on the Z-axis.

Line Drawing

Connection Shape Size	In line charts, the size of the squares that mark the values in the chart.
Line width	In line charts, the width of the line.

3d Axis Sizes

Manual X-Axis Size of Base	In 3d bar charts, defines the relation between the length of the X-axis and the Chart window size. Please note that the `Automatic Chart Axis Size` check box in the 3d Angles section must be deactivated, otherwise the size will still be calculated automatically.
Manual Y-Axis Size of Base	In 3d bar charts, defines the relation between the length of the Y-axis and the Chart window size. Please note that the `Automatic Chart Axis Size` check box in the 3d section must be deactivated, otherwise the size will still be calculated automatically.
Z-Axis Series Margin	In 3d bar charts, the distance on the Z-axis between the individual series.

Gauge

Border Width	In round gauge charts, the width of the border around the gauge.

Gauge Ticks

Border to Tick Distance	In round gauge charts, the space between the inner edge of the border and the ticks that mark the values.
Major Tick Length	In round gauge charts, the length of the major ticks (i.e., ticks that show a label).
Major Tick Width	In round gauge charts, the width of the major ticks (i.e., ticks that show a label).
Minor Tick Length	In round gauge charts, the length of ticks that do not have a value displayed.
Minor Tick Width	In round gauge charts, the width of ticks that do not have a value displayed.

Gauge Needle

Needle Length	In round gauge charts, the length of the needle. (Note that the percentage is calculated from the diameter of the gauge, so if you choose a value greater than 50%, the needle will point to somewhere outside the gauge!)
Needle Width at Base	In round gauge charts, the width of the needle at the center of the gauge.
Needle Base Radius	In round gauge charts, the radius of the base that covers the center of the gauge.

Gauge Color Range

Border to Color Range Distance	In round gauge charts, the space between the inner edge of the border and the outer edge of the color range [335].
Color Range Width	In round gauge charts, the width of the customizable color range. (Note that the percentage is calculated from the diameter of the gauge!)

Gauge Value

Offset to Center	Distance from the center at which the gauge value is displayed.

Extra Value

Offset to Center	Distance from the center at which the extra label (defined in the Gauge Chart settings [329]) is displayed.

8.5.2.9 Fonts

The Fonts section of the Change Appearance dialog box lets you configure fonts for objects in the Chart window.
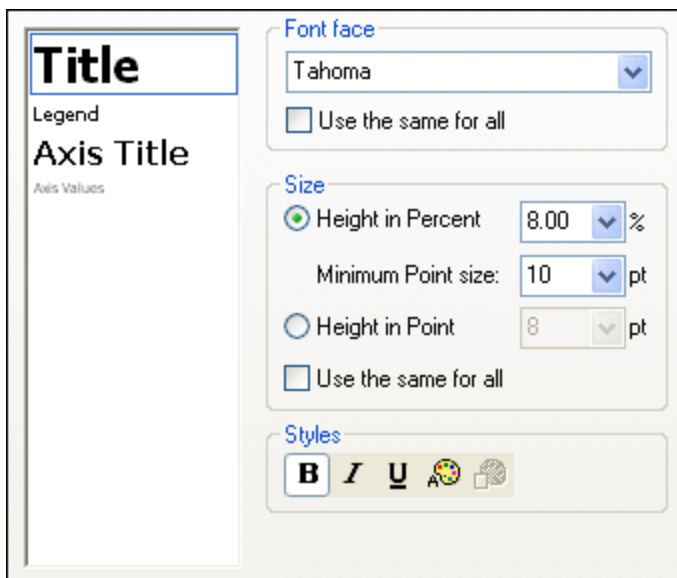
Font settings

You can choose the font face, size, and style for the individual elements displayed in the Chart window. You can define the size as a percentage of the chart size and define a minimum size in points, or specify an

absolute value (in points). To apply the same font and/or size to all text elements, activate the respective `Use the same for all` check box.

The element names in the list box are defined as follows:

- **Title**: The name of a chart
- **Legend**: The key to the colors used in the chart
- **Labels**: The designation of the pieces of a pie chart
- **Axis Title**: The name of the X, Y, and Z axis in a bar or line chart
- **Axis Values**: The units displayed on an axis in a bar or line chart
- **Tick Values**: The units displayed on a gauge chart
- **Values**: The values displayed on the bars of a bar chart

9   SQL Editor

Altova website: 🔗 SQL editor

The **SQL Editor** is the area where you can create, display, edit, and execute the SQL statements that are required for your database work. You can save SQL scripts that you have created in the SQL Editor and add them to your project. In addition, you can also open existing SQL files and display them in the SQL Editor.

The SQL Editor is started automatically when you open an SQL file or generate a statement from an existing database object. Apart from that, you can always open a new SQL Editor window as follows:

- Click the **SQL Editor** 🔲 icon in the Standard toolbar, or select the menu option **File | New | SQL Editor**, or press **Ctrl+N**.

As soon as an SQL Editor window is active, the **SQL Editor** menu and toolbars are available. In addition, each SQL Editor window has a toolbar of its own which provides buttons for executing SQL as well as various text selection and display options discussed further in this chapter.

Note: A new SQL Editor window is automatically associated with the data source that is active in the Online Browser [24] or Project Window [20]. The active data source is displayed at all times in the Execution Target Bar [49]. That is, all commands issued from SQL Editor will affect this particular data source only. If there are multiple data sources, the SQL Editor belongs to the one which was last clicked before you opened the SQL Editor. Any subsequent SQL Editor windows that you open will inherit the currently active data source.

If there is no data source connected, the SQL Editor window is in the "Offline" state. You can easily change the data source to which the SQL Editor window is connected, as shown below.


To assign a data source to an SQL Editor window:

1. Click into the SQL Editor window whose properties you want to change.
2. In the Properties window, select the data source, and, if applicable, the root object.
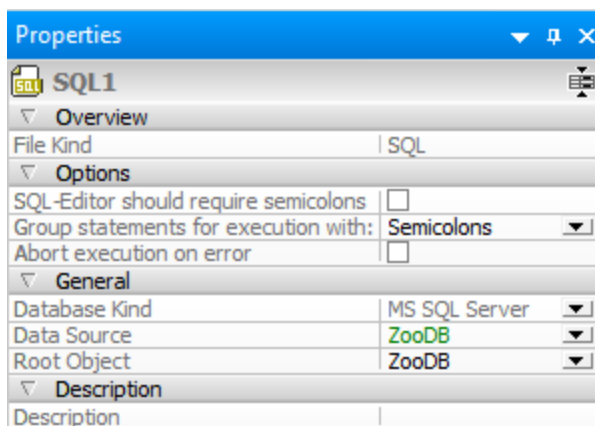
To jump to the Properties window, click the **Unknown** or **Offline** hyperlinks in the Execution Target Bar

(provided that the **Show/Hide Execution Target Bar** button 🔵 is toggled on). In the image below, no data

source is assigned yet to the SQL Editor window, which is why the Execution Target bar displays "Unknown - Offline" as execution target.
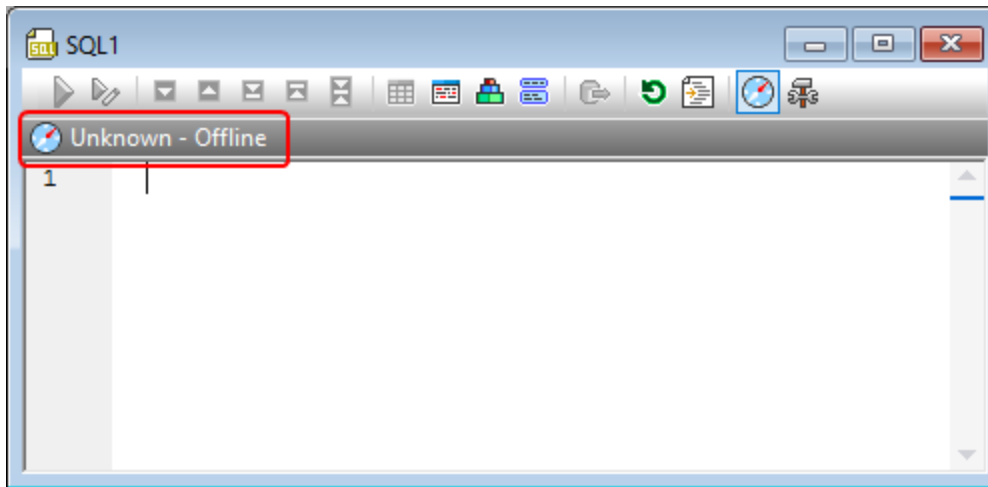
You can edit SQL statements in the SQL Editor like in any other text editor using the standard text manipulation commands such as **Copy** (**Ctrl + C**), **Paste** (**Ctrl + V**), and **Cut** (**Ctrl + X**). In addition, the SQL Editor provides features such as syntax coloring, comments, bookmarks, regions, and text highlighting (see SQL Editor Features [359]). Also, the SQL Editor provides advanced Find and Replace functionality, including the ability to search and replace using regular expressions (see Finding and Replacing Text [375]).

The auto-completion feature can assist you with entering statements into SQL Editor (see Auto-completion [369]). You can also drag existing database objects into the SQL Editor window to create SQL statements automatically (see Generating SQL Statements [346]).

Support for syntax coloring, text auto-completion, and formatting in SQL Editor may vary depending on the database version. More recent database versions typically provide better support for these features.

When SQL Editor contains multiple statements, you can choose to execute them in bulk, if supported by the database type or driver, or sequentially, see Executing SQL [364]. It is also possible to give custom descriptive names to the resulting tabs, see Naming Result Tabs [373].

The appearance of SQL statements in the SQL Editor can be configured in the SQL Formatting Configurations dialog box. For example, you can configure to generated SELECT statements so that a line break should always occur before each FROM keyword. The formatting syntax can be fine-tuned not only for each database kind, but also at statement level, see Formatting SQL [354].

When you need to export only a subset of data from the database, you can write or generate the required SELECT statement in the SQL Editor, and then proceed to exporting results (see Selecting Data for Export [379]). SQL Editor can also be used to create views directly from SQL statements, see Creating Views [269].

The text displayed in the SQL Editor can be printed and also previewed before printing (see Printing SQL [381]).

9.1   Generating SQL Statements

In DatabaseSpy, you can automatically generate SQL statements based on existing objects displayed in the Online Browser. You can either drag a database object from the Online Browser into an SQL Editor window, or right-click a database object in the Online Browser and use the context menu. The objects from which you can generate SQL statements (or extract information such as name and path) are as follows:

- Tables 🔲
- Columns 🔲
- Keys 🔑
- Indexes 📊
- Triggers ⚙
- Views 👓
- Procedures 📋
- Functions ƒ⒳

The following table shows which SQL statements can be generated (or information be extracted into the SQL Editor) for each database object type, using SQL Server 2016 as an example.

Select	🔲	🔲	-	-	-	👓	-	-
Name	🔲	🔲	🔑	📊	⚙	👓	📋	ƒ⒳
Path	🔲	🔲	🔑	📊	⚙	👓	📋	ƒ⒳
Create	🔲	-	-	📊	⚙	👓	📋	ƒ⒳
Drop	🔲	🔲	🔑	📊	⚙	👓	📋	ƒ⒳
Add	-	🔲	🔑	-	-	-	-	-
Alter	-	🔲	-	📊	⚙	👓	📋	ƒ⒳
Rename	🔲	🔲	🔑	📊	-	👓	📋	ƒ⒳
Insert	🔲	🔲	-	-	-	👓	-	-
Update	🔲	🔲	-	-	-	👓	-	-
Delete data	🔲	🔲	-	-	-	👓	-	-
Execute	-	-	-	-	-	-	📋	-

SQL Server 2016

Note: The list of supported statements may vary depending on the database kind.

Select	Creates a SELECT statement that retrieves data from (i) all columns of the source table, (ii) the selected column(s) of the parent table, or (iii) all columns that are included in a view. Basically, this is the same procedure as when selecting Retrieve data [284] from the context menu, however the query is not automatically executed in this case.
Name	Returns the name of the selected database object. You can also select several objects. The names are printed in individual lines, separated by commas.
Path	Returns the full path of the selected database object (for example, `Database.Schema.Table.Column`). You can also select several objects. The paths are printed in individual lines, separated by commas.

Create	Generates a CREATE statement based on the selected database object. Depending on the selected database object, you have to edit this statement as follows: • Edit the **index** name as well as the properties, and enter the appropriate column to create a new index for the parent column of the selected index. • Edit the **trigger** name as well as the table that invokes the trigger, and enter the action that is to be triggered. • Edit procedure name and parameters, and adapt the body accordingly to create a new **stored procedure**. • Edit the **function** name as well as the body of the function. Define additional or remove parameters, if required.
Drop	Creates a DROP or ALTER statement, respectively, that deletes the selected database object from the database.
Add	Generates an ALTER statement that (i) adds a new column to the parent table of the selected column and uses the name and definition of the selected column as default values, or (ii) adds a constraint and uses the properties of the selected constraint as default. The default values have to be edited.
Alter	Generates an ALTER statement that allows you to edit the properties of the selected database object.
Rename	Allows you to rename the selected table, column, or view. Change this parameter to the desired new table name.
Insert	Allows you to insert data into the selected table, column, or view. By default, no default values specified. You must edit the statement and specify the values that are to be inserted into the table. When inserting data into individual columns, or if the primary key is not included in the view, you must also specify a value for the primary key unless an IDENTITY statement has been used for auto-generation of a primary key when the table was created.
Update	Creates an UPDATE statement that updates (i) all columns of the selected table, (ii) the selected column, or (iii) the columns that are included in the selected view. No default is specified for the new values. Note that this command will update **all** rows with the same values if no WHERE statement is added to select a specific row.
Delete data	Generates a DELETE statement that deletes data (i) from the selected table and adds a WHERE clause using the table's primary key to specify the row to be deleted, or (ii) from the parent table of the selected view. Note that this command will delete **all** rows of the table if no WHERE statement is added.
Execute	Declares the parameters needed for execution and creates an EXEC statement that executes the stored procedure with the required parameters.

Generating SQL statements with drag and drop

When you drag a table or column from the Online Browser and drop it into an SQL Editor window, a SELECT statement is generated by default. The kind of statement that should be generated for any database object is configurable in the Online Browser options 419. For example, you can configure that dropping a table into the SQL editor should generate an UPDATE statement, not a SELECT.

To generate SQL statements using the default configured statement type:

- Drag and drop one or several database objects from the Online Browser into an existing SQL Editor window. An SQL statement appears in the SQL Editor. Note that one statement per table will be generated if you select objects from different tables.

You can also right-click a database object and drag it into an open SQL Editor window. In this case, when you release the mouse button, a context menu opens from which you can choose the desired statement type to be generated (for example, SELECT, INSERT, UPDATE, etc.).

To generate SQL statements of specific type:

1. Right-click one or several database objects in the Online Browser and use drag right to drop the object(s) into an existing SQL window.
2. Select from the context menu the type of SQL statement you want to generate.

You can define several additional options that determine how the generated SQL statement appears in the SQL Editor. In the example below, the **Append semi-colons to statement end** has been activated in the SQL Generation [426] options. This causes the semi-colon character to appear at the end of the statement. Also, the **Enable SQL Formatting for the SQL Editor** has been deactivated in the SQL Formatting [427] options. This causes the FROM keyword to appear on the same line as the SELECT keyword (if SQL formatting had been enabled, a line break would have been automatically inserted before the FROM keyword).
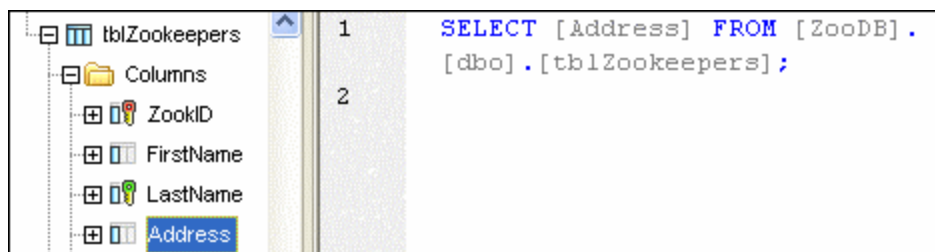
Generated SELECT statement

Generating statements in a new SQL Editor window

If you want your SQL statements to be generated in a new SQL Editor window, do the following:

1. Right-click a database object in the Online Browser and select **Show in new SQL Editor** from the context menu.
2. From the sub-menu, select the appropriate statement you want to use (for example, SELECT, INSERT, etc).

9.2   Generating Complete DDL Scripts

DatabaseSpy enables you to generate DDL (Data Definition Language) statements for all objects of a database or schema, as one SQL script. You can also generate DDL scripts for multiple schemas or databases in the same operation.

Generating the complete DDL scripts of a database or schema may be useful, for example, if you want to re-create the structure of that schema or database on another server or environment, or if you want to back up the current database structure, or if you want to create a text snapshot of the current database structure for comparing it later to another snapshot.

To generate complete DDL scripts from a schema or database:

1. In Online Browser, select the database 🗄 or schema 🕴 from which you would like to generate DDL scripts. In the example below, the "dbo" schema of the ["ZooDB" tutorial database](72) is selected.
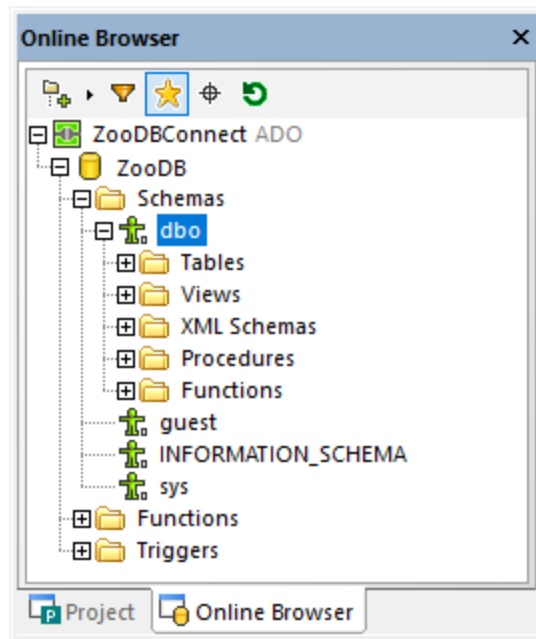
2. Right-click the schema or database ("dbo", in this example), and select the command **Show in new SQL Editor | Complete DDL Script** from the context menu.

If the current database kind supports multiple schemas (like the **ZooDB** database above), it is possible to generate DDL scripts for multiple schemas in one operation (each schema in a separate file). To select multiple schemas or databases, click the required schemas while holding the **Shift** key pressed.

Note that, in this example, the immediate parent of all tables, views, procedures, and so on, is a schema. This may be a database in other database kinds. The structure displayed in the Online Browser depends on the kind of the currently connected database.

When you run the **Complete DDL Script** command for a single schema or database, DatabaseSpy analyzes all objects (such as tables, views, stored procedures, and so on), and then generates their corresponding DDL

statements as a single script, in a new SQL Editor window. If multiple schemas or multiple databases were selected, the DDL script of each schema or database will be in a new SQL Editor window.

In SQL Server, if you selected both a schema and a database, then scripts for the database will be generated. The generated database script will include all objects from all schemas in that database.

While DDL statements are being generated, you are informed of the progress in the Output window as well as through a dialog box from where you can also cancel the operation.

Once DDL statement generation is complete, you can save the script to a file, as described in Opening, Saving, and Closing SQL Files [351].

9.3   Opening, Saving, and Closing SQL Files

Any file that is stored in the file system can be opened via the **File | Open** menu (**Ctrl+O**) and will be displayed under its file name in a new SQL Editor window.

```
 1        -- target: AnimalTypeCountQuery
 2        SELECT COUNT (ZooDB.dbo.tblAnimalTypes.EnglishName)
 3        FROM ZooDB.dbo.tblAnimalTypes, ZooDB.dbo.
          tblAnimalCategories
 4        WHERE ZooDB.dbo.tblAnimalTypes.Category=ZooDB.dbo.
          tblAnimalCategories.CategoryID AND
 5        ZooDB.dbo.tblAnimalCategories.Category='perch-like';
 6
 7        -- target: AnimalNameQuery
 8        SELECT EnglishName, LatinName
 9        FROM ZooDB.dbo.tblAnimalTypes
10        INNER JOIN ZooDB.dbo.tblAnimalCategories
11        ON ZooDB.dbo.tblAnimalTypes.Category=ZooDB.dbo.
```

If an SQL file has already been added to the project, it can be opened from the Project window and will be displayed under its file name in a new SQL Editor window. A file that has been added to the Global Resources [436] can be opened via the **File | Open | Open Global Resource...** menu and will be displayed under its Alias name in a new SQL Editor window.

To open an SQL file:

Do one of the following:

- Select the menu option **File | Open | Open File...** or press **Ctrl+O**. The standard Windows **Open** dialog box appears. Select an SQL (*.sql) file and click **Open**.
- In the Project window, in the SQL folder do one of the following:

  - Select an SQL file and double-click the file name.
  - Right-click the file and select Open from the context menu.

The SQL file is displayed in a new SQL Editor window.

To open a global resource:

- Select the menu option **File | Open | Open Global Resource...** to display the Choose Global Resource dialog box and do one of the following:

  - Select a file-type global resource under the Files directory and click **OK**. The file is displayed under its Alias name in a new SQL Editor window.
  - Select a folder-type global resource under the Folders folder and click **OK**. The standard Windows Open dialog box appears and displays the content of the directory that is defined for the active configuration. Select a file and click **Open**. The file is displayed under its file name in a new SQL Editor window.

Alternatively, do the following:

1. Select the menu option **Tools | Global Resources** or, in the Global Resources toolbar, click the **Manage Global Resources** 🖳 button.
2. Under the Files folder, click the Alias name of the SQL file you want to open.

3. Click the **View** button. The file is opened with the configuration that is selected in the drop-down list of the Global Resources toolbar. If the selected configuration is not applicable, the SQL file opens with the Default configuration.

Reparsing

It may occur that a script contains parsing errors when it is first displayed in the SQL Editor. If, for example, the execution blocks are not set correctly, such a script would produce an error when being executed. (To check the execution blocks, toggle on the **Show groupings for execution** 🔲 icon.) Also, the syntax highlighting may not be displayed correctly when the script is initially parsed or edited. To correct such errors, the SQL Editor toolbar provides the **Reparse the whole document** 🔄 button.

To correct a SQL script with parsing errors:

- In the SQL Editor toolbar, click the **Reparse the whole document** 🔄 button.

Saving and closing

DatabaseSpy offers several options for saving SQL statements in, or closing one or several windows of, the SQL Editor:

- Save: saves the contents of the active window to the file from which it has been opened.
- Save As: pops up the familiar Windows **Save As** dialog box, in which you enter the name and location of the file you wish to save the active file as.
- Save All: saves all modifications that have been made to any open documents.
- Close: closes the active window.
- Close All: closes all open windows.
- Close All But Active: closes all open windows but leaves the active file open in the SQL Editor.

For all Close commands, if the file was modified (indicated by an asterisk * after the file name in the title bar), you will be asked if you wish to save the file first.

All commands except **Close All But Active** are available in the **File** menu, and all commands that affect only one particular window are included in the context menu that opens when you right-click the tab of the respective SQL Editor window.

```
12        ZooDB.dbo.tblAnimalCategories.Cate
13        ORDER BY LatinName ASC;
```

animalType_queries.sql

💾	Save...	Strg+S
	Save As...	
	Close	
	Close All But Active	
	Add to Project	

Output

🔻 ▼ ▲ | 📑 📑 📑

⊟ ◉ Connecting 🔷 ZooDBCor
     ➕ Loading Schemas in
     ➕ Loading Tables in 🌿 vie
     ➕ Done
     ➕ Loading Databases in 🔷 ZooDBConnect
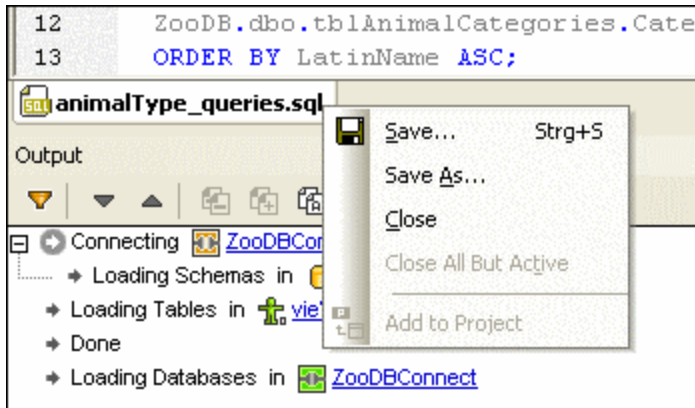
In addition to the menu commands listed above, you can also use the keyboard shortcut **Ctrl+S** to save the active SQL file.

Saving SQL scripts

You can save any SQL that appears in an SQL Editor window and re-use the script later on. If the content of an SQL Editor window has already been saved, the file name appears in the title bar of the SQL Editor window. Unsaved files can be identified by the asterisk to the right of the generic title in the title bar.

A saved SQL script can be made available for all Altova applications by defining it as a global resource [443]. Global resources can be used in all Altova applications that share the same Global Resources XML File [440].

To save the content of an SQL Editor window to a file:

1. Do one of the following:

   - Right-click the tab of the respective SQL Editor window and choose **Save...** from the context menu.
   - Make the SQL Editor window you want to save the active window and press **Ctrl+S**.

2. If you save the file for the first time, choose a name and path for the file in the **Save As...** dialog box and click **Save**.

To save an SQL file under a new name:

1. Right-click the tab of the respective SQL Editor window and choose **Save As...** from the context menu.
2. In the **Save As...** dialog box, choose a new name and path for the file and click **Save**.

9.4   **Formatting SQL**

Most database versions support SQL code formatting in SQL Editor. For all such databases, the SQL code formatting preferences can be customized. You can also disable formatting for the SQL Editor and the Database Structure Change Script window. By default, SQL formatting is enabled in these two windows.

Any changes you make to the default formatting are recorded in a configuration file, at the following path: **..\Documents\Altova\DatabaseSpy2023\SQLFormattingConfigurations.qfmt**. Note that the configuration file is not automatically transferred during upgrades if the installation folder changes. After upgrading to a new major version, you have to copy the configuration file to the new installation folder (**..\Documents\Altova\DatabaseSpy<Version>**) yourself.

Note: Configurations that have been saved in DatabaseSpy 2009 or later cannot be opened in earlier releases.

The formatting is applied automatically when you generate SQL [346]. To apply formatting on demand, first click inside the SQL Editor, and then do one of the following:

- Click the **Apply SQL Formatting to the active SQL Editor** button.
- On the **SQL Refactoring** menu, select **Format SQL**.

To customize the formatting preferences for a specific database kind:

1. On the **Tools** menu, click **Options**. (Alternatively, press **Ctrl+Alt+O**.)
2. Go to **SQL Editor | Formatting settings** page.
3. Click **SQL Formatting Configurations**. The SQL Formatting Configurations dialog box opens.
4. Select the database kind from the **Configurations** list and then change your preferences as required.

The SQL Formatting Configurations dialog box displays a tree structure where you can define preferences on root level or specify more specific options for individual statement types. The Preview area is updated immediately to reflect any changes you make.
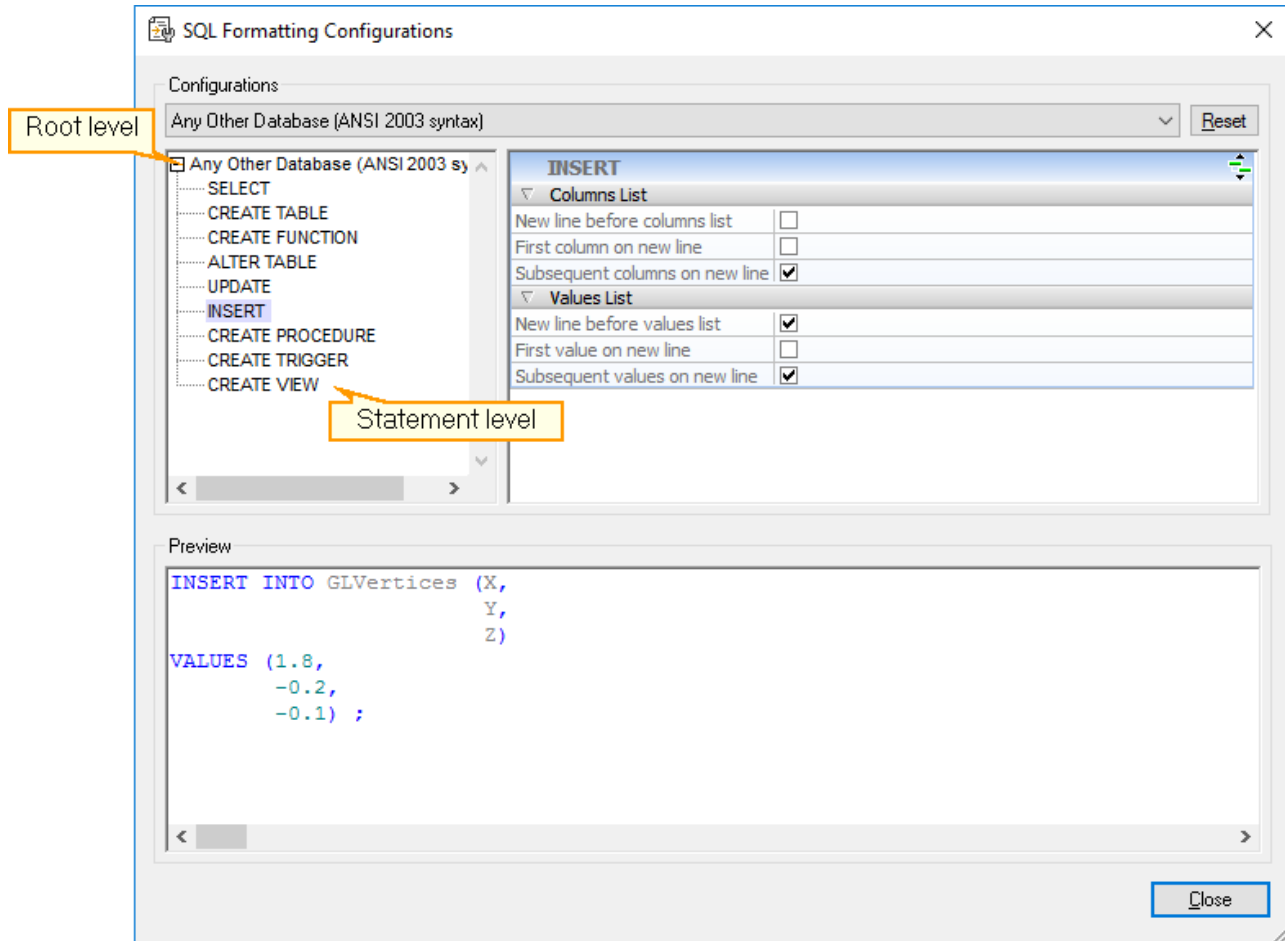
Options on root level

At the root level, you can either deactivate the formatting options for a specific database type (without deactivating the SQL formatting feature as a whole), or define general formatting options which may affect all statements.

To deactivate SQL formatting for a certain database kind:

1. Select the menu command **Tools | Options** or press **Ctrl+Alt+O** to open the Options dialog box.
2. Click **SQL Formatting Configurations**.
3. In the SQL Formatting Configurations dialog box, select the respective database kind from the **Configurations** list.
4. In the Options section, select the **Ignore All** check box.

Options for the individual statement types

When you select one of the statement types in the tree structure, the corresponding formatting options are displayed in the window on the right side of the dialog box. Different options are available for the individual statement types.

The header of the options list provides an icon on the right edge that displays the properties inherited from the

root level. If you click the **Show inherited properties** ⬍ icon, the options from the root level are displayed
and you can change the settings for the selected statement type. Note that these changes apply only to the
selected statement type; for all other parts of the statements, the settings defined on the root level are still
valid! Options that differ from those defined on root level appear in bold type. These options remain also visible if

you click the **Hide inherited properties** ⬍ button.

You can customize the display of the following statement types:

- **Select**

Statement
New line before ORDER BY
Select Items List
First item on new line
Subsequent select list items on new lines
FROM Clause
New line before FROM
First table source on new line
Subsequent table sources on new lines
WHERE Clause
New line before WHERE
First search condition on new line
Subsequent search conditions on new lines

- **Create Table**

Statement
First column/key/index/etc. on new line
Subsequent columns/keys/indices/etc. on new lines
New line before the table definition

- **Create Function**

Statement
Indent contents

- **Alter Table**

Statement
New line before (ADD) CONSTRAINT
New line before FOREIGN KEY
New line before REFERENCES
New line before PRIMARY KEY
New line before UNIQUE

- **Alter Function**

Statement
Place BEGIN and END on new lines
Indent contents

- **Update**

Columns List
New line before columns list
First column on new line
Subsequent columns on new line
WHERE Clause

New line before WHERE
First search condition on new line
Subsequent search conditions on new lines

- **Insert**

Columns List
New line before columns list
First column on new line
Subsequent columns on new line
Values List
New line before values list
First value on new line
Subsequent values on new line

- **Create Procedure**

General Formatting
Disable all formatting
New line before definition

- **Create Trigger**

General Formatting
Disable all formatting
New line before definition

- **Alter Procedure**

General Formatting
Disable all formatting
New line before definition

- **Create View**

General Formatting
Disable all formatting
New line before definition

- **Alter View**

General Formatting
Disable all formatting
New line before definition

Note: Some statement types listed above are not available for configuration if they are not applicable to the selected database kind.

Restoring the default settings

You can reset your custom formatting preferences for any database kind to the default settings as follows:

1.   In the SQL Formatting Configurations dialog box, select the respective database kind from the **Configurations** list.
2.   Click **Reset**.

Removing formatting and comments

You can remove all additional whitespace and line feeds that have been added by the **Format SQL** command. The menu command **SQL Refactoring | Remove Comments and Formatting** not only deletes all these characters but also removes any text that is located between block comments [371] or after line comments [371].

The left pane shows a SELECT statement when SQL formatting is enabled in the SQL Formatting options [427] and the Whitespace markers and End-of-Line markers options are activated in the Text View Settings [363]. The right pane shows the same statement with the same options after the **Remove Comments and Formatting** command has been selected.

Removing comments

If you want to get rid of all that text in your statement that has been commented out, select the **Remove comments** command in the **SQL Refactoring** menu. Any text that is enclosed by block comment [371] characters or appears after a line comment [371] will be deleted from the active SQL Editor window.

9.5   SQL Editor Features

The SQL Editor has multiple visual aids to make the display, editing and navigation of text easier. These include:

- [Line numbers](360) [360]
- [Syntax coloring](360) [360]
- [SQL statement selection and navigation](361) [361]
- [Bookmarks](361) [361]
- [Regions](362) [362]
- [Source folding](362) [362]
- [End-of-Line and whitespace markers](363) [363]
- [Zooming](363) [363]
- [Word wrapping](363) [363]
- [Text highlighting](363) [363]

> Support for syntax coloring, text auto-completion, and formatting in SQL Editor may vary depending on the database version. More recent database versions typically provide better support for these features.

Where applicable, you can toggle or customize the features above from the **Text View Settings** dialog box. Settings in the **Text View Settings** dialog box apply to the entire application—not only to the active document.
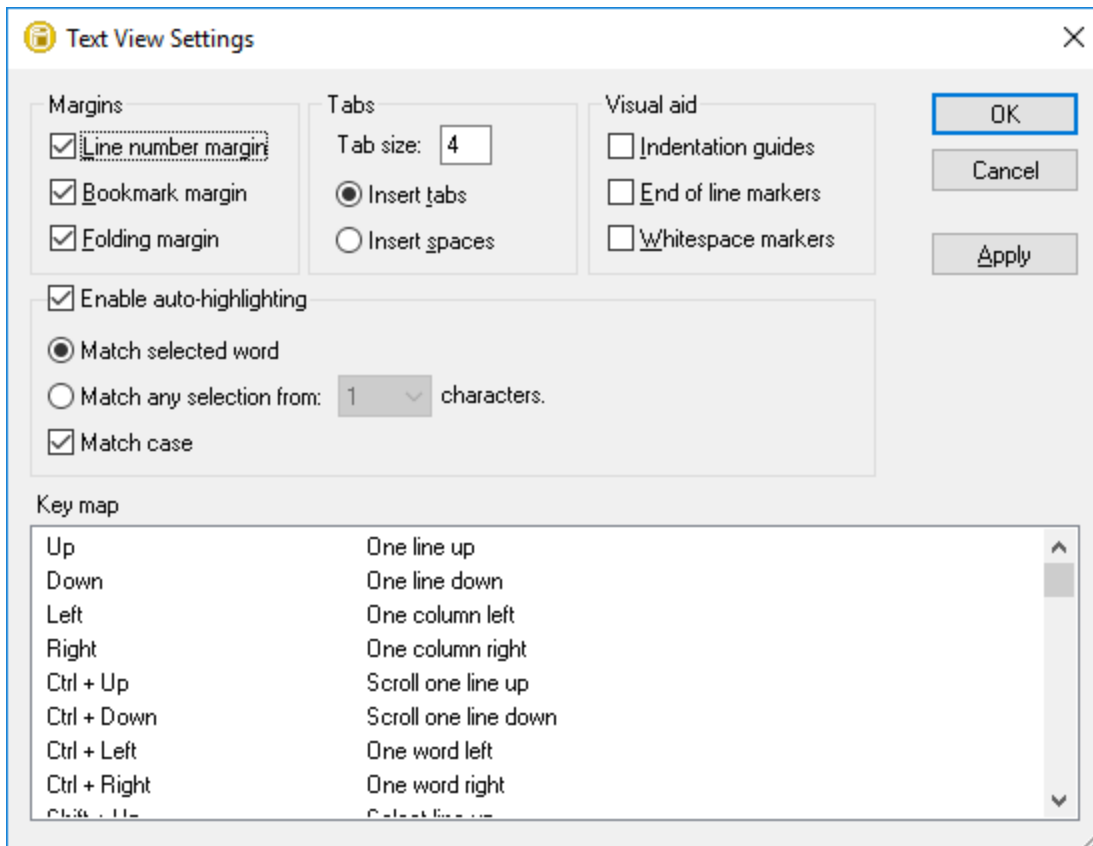
Text View Settings dialog box

To open the **Text View settings** dialog box, do one of the following:

- On the **SQL Editor** menu, select **Text View Settings**.
- Right-click the SQL Editor, and select **Text View Settings** from the context menu.

Some of the navigation or viewing aids can also be toggled from the application menu, toolbars, or keyboard shortcuts. For reference to all applicable shortcuts, see the "Key Map" section of the **Text View Settings** dialog box illustrated above.

Line numbers

Line numbers are displayed in the line numbers margin, which can be toggled on and off in the **Text View Settings** dialog box. When a section of text is collapsed, the line numbers of the collapsed text are also hidden.

Syntax coloring

Syntax coloring is applied according to the semantic value of the text. You can enable or disable syntax coloring from the **Tools | Options** menu, **SQL Editor** group.

Statement selection and navigation

The following buttons in the SQL Editor toolbar allow you to select text, depending on where the cursor is positioned.

Next statement (Alt+Down): Selects the statement after the statement where the cursor is currently located.

Previous statement (Alt+Up): Selects the statement before the statement where the cursor is currently located.

Last statement (Alt+End): Selects the last statement in the SQL Editor.

First statement (Alt+Home): Selects the first statement in the SQL Editor.

Current statement (Shift+Alt+Enter): Selects the statement where the cursor is currently located. Alternatively, you can also triple-click into a statement or click into any of the margins, if visible.

Bookmarks

Lines in the document can be bookmarked for quick reference and access. If the bookmarks margin is toggled on, bookmarks are displayed in the bookmarks margin.

```
1      SELECT [id], [name], [descr], [deptid], [teacherid],
       [upperlimit] FROM [Course];
2  ●   SELECT [id], [name] FROM [Department];
3      SELECT [id], [name] FROM [Teachers];
```

Otherwise, bookmarked lines are highlighted in cyan.

```
1      SELECT [id], [name], [descr], [deptid], [teacherid],
       [upperlimit] FROM [Course];
2      SELECT [id], [name] FROM [Department];
3      SELECT [id], [name] FROM [Teachers];
```

The bookmarks margin can be toggled on or off in the **Text View Settings** dialog box.

You can edit and navigate bookmarks using the following commands:

Insert/Remove Bookmark

Go to Next Bookmark

Go to Previous Bookmark

Delete All Bookmarks

The commands above are available in the **SQL Editor | Bookmarks** menu and the SQL Editor toolbar. Bookmark commands are also available through the context menu, when you right-click the SQL Editor.

Regions

In the SQL Editor, regions are sections of text that you mark and declare as a unit to structure your SQL scripts. The benefit of regions is that their enclosed text can be easily collapsed or expanded, and thus simplify navigation in long scripts (see also "Source folding" below). It is also possible to nest regions within other regions.

To create a region:

1. In the SQL Editor, select the statements you want to make into a region.
2. Click the **Add Region** button in the SQL Script toolbar or select the menu option **SQL Editor | Insert | Insert Region**.

```
1    -- region Department
2    INSERT INTO [Department] ( [id], [name] ) VALUES ( 0, 'Science' );
3    INSERT INTO [Department] ( [id], [name] ) VALUES ( 1, 'Math' );
4    INSERT INTO [Department] ( [id], [name] ) VALUES ( 2, 'Art' );
5    -- endregion
```

To remove a region:

*   Delete the `-- region` and `-- endregion` comments.

Source folding

Source folding refers to the ability to expand and collapse regions and is displayed in the source folding margin. The margin can be toggled on and off in the Text View Settings dialog box. To expand or collapse portions of text, click the "+" and "-" nodes at the left side of the window. Any portions of collapsed code are displayed with an ellipsis symbol. To preview the collapsed code without expanding it, move the mouse cursor over the ellipsis. This opens a tooltip that displays the code being previewed, as shown in the image below. Note that, if the previewed text is too big to fit in the tooltip, an additional ellipsis appears at the end of the tooltip.

```
1    -- region Department
2    INSERT INTO [Department] ( [id], [name] ) VALUES ( 0, 'Science' );
3    INSERT INTO [Department] ( [id], [name] ) VALUES ( 1, 'Math' );
4    INSERT INTO [Department] ( [id], [name] ) VALUES ( 2, 'Art' );
5    -- endregion
6
7    -- region Teachers ... -- endregion
14
15   -- region Course       INSERT INTO [Teachers] ( [id], [name] ) VALUES ( 0, 'Mr Smith' );
16   INSERT INTO Cours       INSERT INTO [Teachers] ( [id], [name] ) VALUES ( 1, 'Mrs Jones' );
17   INSERT INTO Cours       INSERT INTO [Teachers] ( [id], [name] ) VALUES ( 2, 'Mr Patel' );
18   INSERT INTO Cours       INSERT INTO [Teachers] ( [id], [name] ) VALUES ( 3, 'Mr Harris' );
19   INSERT INTO Cours       INSERT INTO [Teachers] ( [id], [name] ) VALUES ( 4, 'Ms Wong' );
20   INSERT INTO Course ( id, name, descr, deptid, teacherid, upperlimit ) VALUES ( '4',
21   INSERT INTO Course ( id, name, descr, deptid, teacherid, upperlimit ) VALUES ( '5',
22   INSERT INTO Course ( id, name, descr, deptid, teacherid, upperlimit ) VALUES ( '6',
23   INSERT INTO Course ( id, name, descr, deptid, teacherid, upperlimit ) VALUES ( '7',
24   -- endregion
```

The **Toggle All Folds** ![icon] command toggles all regions to their expanded or collapsed forms.

End-of-line markers, whitespace markers

End-of-line (EOL) markers and whitespace markers can be toggled on in the **Text View Settings** dialog box. The image below shows a document where both end-of-line and whitespace markers are visible. An arrow represents a tab character, a "CR" is a carriage return, and a dot represents a space character.

Zooming in and out

You can zoom in and out by scrolling (with the scroll-wheel of the mouse) while holding the **Ctrl** key pressed. Alternatively, press the "-" or "+" keys while holding the **Ctrl** key pressed.

Word wrapping

To toggle word wrapping in the currently active document, select the **SQL Editor | Word Wrap** menu command, or click the **Word Wrap** ![icon] toolbar button.

Text highlighting

When you select text, all matches in the document of the text selection that you make are highlighted automatically. The selection is highlighted in pale blue, and matches are highlighted in pale orange. The selection and its matches are indicated in the scroll bar by gray marker-squares. The current cursor position is given by the blue cursor-marker in the scroll bar.

To switch text highlighting on, select **Enable auto-highlighting** in the Text View Settings dialog box. A selection can be defined to be an entire word or a fixed number of characters. You can also specify whether casing should be taken into account or not.

For a character selection, you can specify the minimum number of characters that must match, starting from the first character in the selection. For example, you can choose to match two or more characters. In this case, one-character selections will not be matched, but a selection consisting of two or more characters will be matched. So, in this case, if you select t, then no matches will be shown; selecting ty will show all ty matches; selecting typ will show all typ matches; and so on.

9.6   Executing SQL

From the SQL Editor, you can execute both SQL statements written from scratch and those generated from database objects, as described in Generating SQL Statements [346]. You can also execute SQL scripts opened from files. You can call the **Execute** command from the **SQL Editor** menu, from a keyboard shortcut or by clicking the **Execute** ▷ button.

To execute SQL statements:

1. If there are multiple SQL Editor windows open, click inside the SQL Editor window where the statement to be executed is.
2. Make sure that the data source associated with the SQL Editor is connected 🔵 .
3. Optionally, select the statement or statements you want to execute.
4. Do one of the following:

    *   Click **Execute** ▷.
    *   Select the menu command **SQL Editor | Execute**.
    *   Press **F5**.

If the data source is not connected, a dialog box appears asking you to connect to the data source. Click **Connect** if applicable.

After running the **Execute** command, all SQL statements that are in the SQL Editor, or the selected statements, respectively, are executed. If the statement retrieves multiple recordsets, the results appear in separate result windows [41], one for each statement.

You can also execute SQL scripts in the Project window if the SQL files are included in the project, as follows:

1. In the Project window, select an SQL file that you want to execute.
2. Right-click the file and choose **Execute SQL** from the context menu. If the data source is not connected, a dialog box appears asking you to connect to the data source. Click **Connect** if applicable.

The SQL script opens in a new SQL Editor window and is executed immediately.

Execute for Data Editing

When supported by the data source, you can execute a SELECT statement for data editing. This makes it possible to edit the retrieved records in the Result window after the SELECT query is executed.

**To execute SELECT statements for data editing:**

1. Generate or enter a SELECT statement in the SQL Editor window.
2. Make sure that the data source is connected 🔵. If it supports data editing from within the SQL Editor, the **Execute for Data Editing** 📝 button is enabled, and the menu command **SQL Editor | Execute for Data Editing** command is available.
3. Do one of the following:

a. Click **Execute for Data Editing** 🖊 .
b. Select the menu command **SQL Editor | Execute for Data Editing**.

The Result window is switched into the Editing mode and you can edit the database data [298] directly in the result grid.

Stopping execution

DatabaseSpy uses asynchronous execution and retrieval when a query is started in the SQL Editor or from within a Design Editor window. While the execution is in progress, DatabaseSpy displays a message in the Message window.
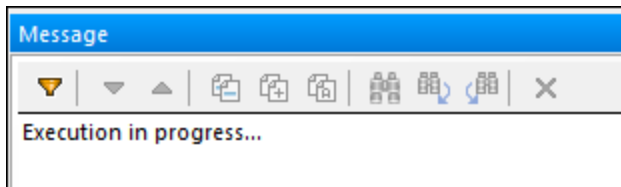
In the execution takes a long time, you may be able to cancel it by clicking the **Stop Execution/Retrieval** 🟥 button in the SQL Editor toolbar. This button appears only while the execution is in progress. On cancelling the execution or retrieval of records, an "Execution was cancelled" message is displayed in the status bar of the Result window.

You can define a timeout for the execution of queries in the SQL Editor options [425]. DatabaseSpy will attempt to stop retrieving data after the specified amount of time has elapsed. Note that the database kind and database driver must support setting a command timeout; otherwise, the query will continue running beyond the timeout period. In this case, you can stop execution using the **Stop Execution/Retrieval** 🟥 button; however, be aware that some drivers might not support cancelling the execution.

Defining the execution mode

You can choose whether to send the SQL statements to the database engine as a whole, or as separate batches. The Properties window provides the settings **SQL Editor should require semicolons** and **Group statements for execution with** for this purpose.

The setting **Group statements for execution with** provides the following options:

- **Semicolons**: The parser uses semicolons as statement separator. Any SQL code that is terminated by a semicolon is considered a statement and sent to the server separately. If semicolons are missing from your SQL script, a dialog box may prompt you to add them automatically.
- **No Grouping**: The script is sent to the server as a whole, without any modification. If the script contains multiple statements, and if the driver is known not to support executing multiple statements in bulk, this option is not meaningful. In this case, a dialog box may prompt you about this limitation.
- **SQL Grammar**: The respective flavor of SQL grammar is used to separate the statements when sending them to the server.
- **GO keyword**: A GO keyword must be present in the SQL script to separate execution blocks.

Showing execution groups

The toolbar in the SQL Editor provides the **Show groupings for execution** [icon] button, which allows for a graphical representation of the statement blocks that will be sent to the database engine for execution.

```
-- region products
INSERT INTO "main"."products" ("name","quantity","description") VALUES
    ('Flexfan',200,NULL),
    ('Small Epsilon',4687,NULL),
    ('Arepharm',65,NULL),
    ('Accumax',0,NULL),
    ('Sumtough',4,NULL),
    ('Silver-Hold',54,NULL),
    ('Quadplus',550,NULL),
    ('Freshtex',10,NULL),
    ('Candysoft',23,NULL),
    ('Ornamantis',2,NULL) ;
-- endregion

-- region addresses
INSERT INTO "main"."addresses" ("user_id","is_shipping","is_billing","type","city","street","number") VALUES
    (1,1,0,'work','Bridgedell','Maple Lane','1'),
    (1,0,1,'home','Bridgedell','Olive Street','6'),
    (3,1,1,'home','Roseford','Evergreen Lane','34'),
    (4,1,1,'work','Beardale','Route 44','9'),
    (6,1,1,'home','Johnson City','Franklin Avenue','11'),
    (7,1,1,'home','North Kingstown','Beach Alley','5'),
    (8,1,1,'home','Merrowmeadow','Freybeach Street','85'),
    (10,1,1,'work','Barrowedge','Penn Street','8'),
    (12,1,1,'home','Elfville','Creek Road','3'),
    (13,1,1,'home','Roseford','Bowman Ave.','853'),
    (14,1,1,'work','Beardale','Iroquois Street','98'),
    (17,1,1,'home','Bridgedell','Smith Road','7'),
    (18,1,0,'home','Roseford','Wood Street','7'),
    (18,0,1,'work','Johnson City','Thorne Lane','9677'),
    (20,1,1,'home','Mechanicsville','Vine Street','9065');
-- endregion
```

In the screenshot above, the groupings for execution are displayed when the "Semicolons" option is selected in the **Group statements for execution with** drop-down list.

Stop execution on error

When the SQL Editor window is open, the Properties window includes a property called **Abort execution on error**. This option may be useful when you execute SQL scripts that contain multiple statements, and, if any statement fails, you would like the execution to stop at the corresponding statement. To enable this behavior, select the **Abort execution on error** check box.
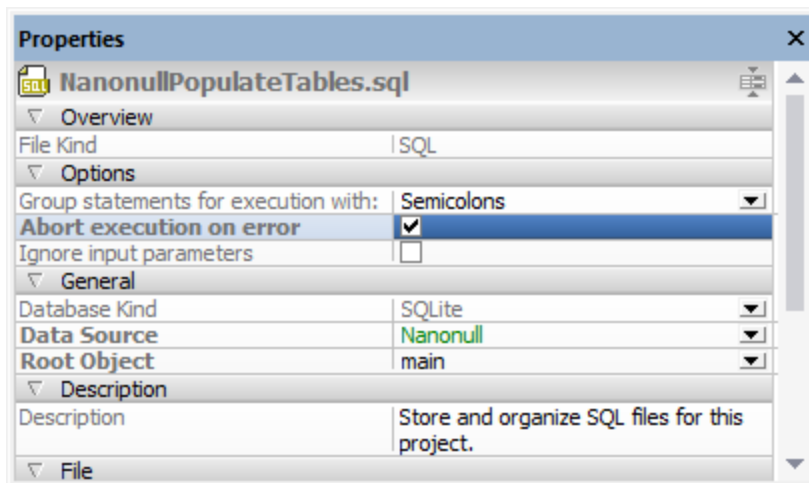
Properties window

Consider the following example: you would like to run against the database a script which contains multiple INSERT statements. If the **Abort execution on error** check box is enabled, and an INSERT statement fails, the script will stop running at the INSERT statement which failed. In this case, any subsequent INSERT statements will not be executed. If the **Abort execution on error** check box is disabled, and an INSERT statement fails, the script will proceed to executing the rest of the INSERT statements, until the end of the script.

The execution status of the script (including the number of affected rows) can be viewed in the Message window, along with additional information about the error which occurred (if any). Click the underlined text in the Messages window to jump to the corresponding line in the SQL Editor.

Stop execution through a script comment

You can also stop execution at the first encountered error by adding the following comment to your SQL script:

```
-- abort_execution_on_error: true
```

The `abort_execution_on_error` comment must be on one line and it can take any of the following values:

```
-- abort_execution_on_error: true|false|yes|no|y|n|1|0
```

To add the comment above to a script, do one of the following:

- Type the comment in the SQL Editor window.
- Place the cursor on an empty line in SQL Editor and then click the **Insert Abort on Execution** 🔗 toolbar button
- Right-click in the SQL Editor window and select **Insert Abort on Execution** from the context menu.

Make sure that the execution mode [365] is set to "Semicolons", "GO keyword", or "SQL Grammar". Otherwise, the comment will be ignored.

You can add the comment anywhere in the script. Note, however, that it will affect only those SQL statements that follow it. For example, in the script illustrated below, only the second INSERT will be affected by the comment. In other words, any SQL statement before the comment will stop execution on error only if the **Abort execution on error** property is set.

```
-- region products
INSERT INTO "main"."products" ("name","quantity","description") VALUES
    ('Flexfan',200,NULL),
    ('Small Epsilon',4687,NULL),
    ('Arepharm',65,NULL),
    ('Accumax',0,NULL),
    ('Sumtough',4,NULL),
    ('Silver-Hold',54,NULL),
    ('Quadplus',550,NULL),
    ('Freshtex',10,NULL),
    ('Candysoft',23,NULL),
    ('Ornamantis',2,NULL) ;
-- endregion

-- abort_execution_on_error: yes

-- region addresses
INSERT INTO "main"."addresses" ("user_id","is_shipping","is_billing","type","city","street","number") VALUES
    (1,1,0,'work','Bridgedell','Maple Lane','1'),
    (1,0,1,'home','Bridgedell','Olive Street','6'),
    (3,1,1,'home','Roseford','Evergreen Lane','34'),
```

If there is text selected, the comment must be in that text for it to be applied.

Setting the comment is independent of the **Abort execution on error** property described previously—it neither modifies nor obeys the property.

9.7   Auto-Completion

When entering an SQL statement in the SQL Editor, auto-completion helps you by offering lists of appropriate keywords, data types, identifiers, separators, and operators depending on the type of statement you are entering. By default, auto-completion is invoked automatically in certain contexts (for example, when you start typing a statement).
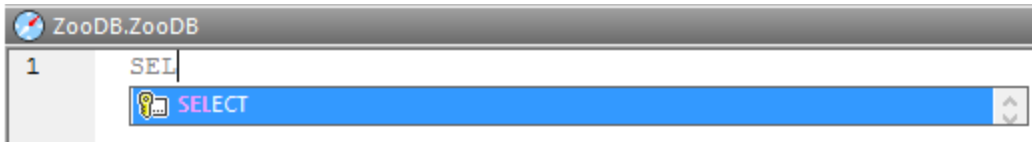
Auto-completion

Use the **Up** and **Down** keyboard keys to navigate through the list of suggestions. To pick a suggested entry, click it or press **Enter**.

To disable auto-completion suggestions:

1. On the **Tools** menu, click **Options** (or press **Ctrl+Alt+O**).
2. Click **SQL Editor**.
3. Under **Entry Helpers**, clear the **Automatically open** check box.

To invoke auto-completion suggestions manually as and when required:

-   In the SQL Editor, press **Ctrl+Space**.

For example, in the image below, auto-completion has been invoked manually by pressing **Ctrl+Space**. The goal here is to quickly select a table available in the `[dbo]` schema of a SQL Server database.
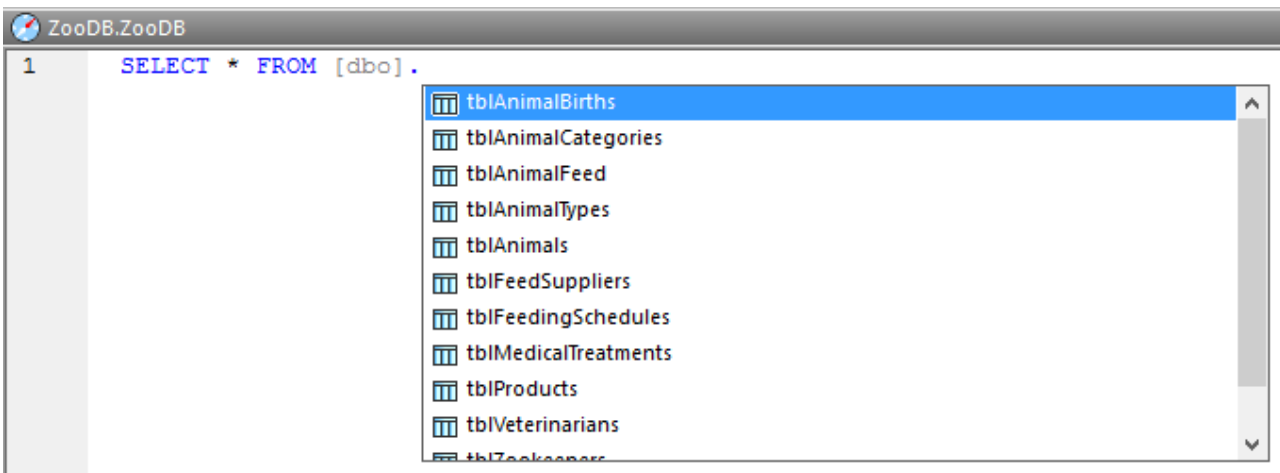
Auto-completion invoked manually

The object types that may appear in the auto-completion window can be identified by icon type, as follows:

Schemas

Tables

Views

Columns

Functions

Data types

Procedures

Keywords

Separators

9.8   Commenting Out Text

In the SQL Editor it is possible to comment out statements, parts of statements, or groups of statements. These statements, or the respective parts of them, will be skipped when the SQL script is being executed.

To comment out a section of text:

1. Select a statement or part of a statement.

```
-- target: AnimalNameQuery
SELECT EnglishName, LatinName
FROM ZooDB.qa_usr.tblAnimalTypes
INNER JOIN ZooDB.dbo.tblAnimalCategories
ON ZooDB.qa_usr.tblAnimalTypes.Category=ZooDB.dbo.tblAnimal
AND
ZooDB.dbo.tblAnimalCategories.Category='perch-like'
ORDER BY LatinName ASC
```

2. Click the **Comment/Uncomment selected block** /* button in the SQL Script toolbar or select
    Insert | Insert/Remove Block Comment from the **SQL Editor** menu. The statement is commented
    out.

To comment out text line by line:

1. Put the cursor in a statement at the point from which you want to comment it out.

```
-- target: AnimalNameQuery
SELECT EnglishName, LatinName
FROM ZooDB.qa_usr.tblAnimalTypes
INNER JOIN ZooDB.dbo.tblAnimalCategories
ON ZooDB.qa_usr.tblAnimalTypes.Category=ZooDB.dbo.tblAnimal
 AND
ZooDB.dbo.tblAnimalCategories.Category='perch-like'
ORDER BY LatinName ASC
```

2. Click the **Comment/Uncomment selected lines** -- button in the SQL Script toolbar or select **Insert
    | Insert/Remove Line Comment** from the **SQL Editor** menu. The statement is commented out from
    the current position of the cursor to the end of the line.

To remove a block comment or a line comment:

1. Select the part of the statement that is commented out. If you want to remove a line comment, it is
    sufficient to select only the comment marks -- before the comment.
2. Do one of the following:

- Click the **Comment/Uncomment selected block** button in the SQL Script toolbar or select **Insert | Insert/Remove Block Comment** from the **SQL Editor** menu.

- Click the **Comment/Uncomment selected lines** button in the SQL Script toolbar or select **Insert | Insert/Remove Line Comment** from the **SQL Editor** menu.

9.9   Naming Result Tabs

When you execute a script that contains more than one SELECT query, the results of each query are displayed in separate result windows. By default, these windows are called "Result1", "Result2", etc. Using *targets*, you can specify more meaningful names for query result windows.

Prerequisites
- The **Show multiple results stacked** option must be switched off.
- Select either "Semicolons" or "SQL Grammar" in the **Group statements for execution with** drop-down list in the Properties window. Otherwise, the query is still executed, but the results are displayed in unnamed result tabs.

To name a target result window:

1. Place the cursor anywhere in the SELECT statement. Click the **Current statement** button (**Shift+Alt+Enter**) in the SQL Editor window. This selects the entire statement.

2. Click the **Define Target Name** button in the SQL Script toolbar. The text -- target: Result appears above the statement.

```
-- target: Result
SELECT [ZookID], [FirstName], [LastName], [Address],
[City], [State], [Telephone], [DOB] FROM
[ZooDB].[dbo].[tblZookeepers];
```

3. Change the text "Result" as desired.

```
-- target: zookepers
SELECT [ZookID], [FirstName], [LastName], [Address],
[City], [State], [Telephone], [DOB] FROM
[ZooDB].[dbo].[tblZookeepers];
```

4. In the Properties window, make sure that either "Semicolons" or "SQL Grammar" is selected in the **Group statements for execution with** drop-down list.

5. Click the **Execute** button or press **F5**.
   The results appear in a Result window that has a tab with the text that you specified as target, e.g., zookeepers.

```
1      -- target: zookeepers
2      SELECT [ZookID],[FirstName],[LastName],
       [Address],[City],[State],[Telephone],[DOB]
       FROM [ZooDB].[dbo].[tblZookeepers];
```

zookeepers

	ZookID	FirstName	LastName	Address	City	State
1	1	Annette	Bearhug	451 Capybara Lane	Newark	NJ
2	2	Bob	Barker	68 Cage Boulevard	Philadelphia	PA
3	3	Oliver	Birdsong	987 Flamingo Avenue	Harrisburg	PA
4	4	Jenny	Lionheart	276 Cheetah Street	Sayre	PA

Execution was successful.   Rows: 15, Cols: 8   0.250 sec   10:47:39

Message   zookeepers

9.10   Finding and Replacing Text

You can search for a term in the entire document or within a text selection. To start a search in SQL Editor, press **Ctrl+F** or select the menu command **Edit | Find**. You can enter a string or use the combo box to select a string from one of the last 10 strings. When you enter or select a string, all matches are highlighted, and the positions of the matches are indicated by orange markers in the scroll bar (*see screenshot below*). The position of the currently selected match (highlighted in gray) depends on where the cursor was last located.

You can see the total number of matches and the index position of the currently selected match. Use the ◀ (**Previous**) and ▶ (**Next**) buttons to switch between the matches. To switch between the **Find** and **Replace** modes, click the ▼ button.
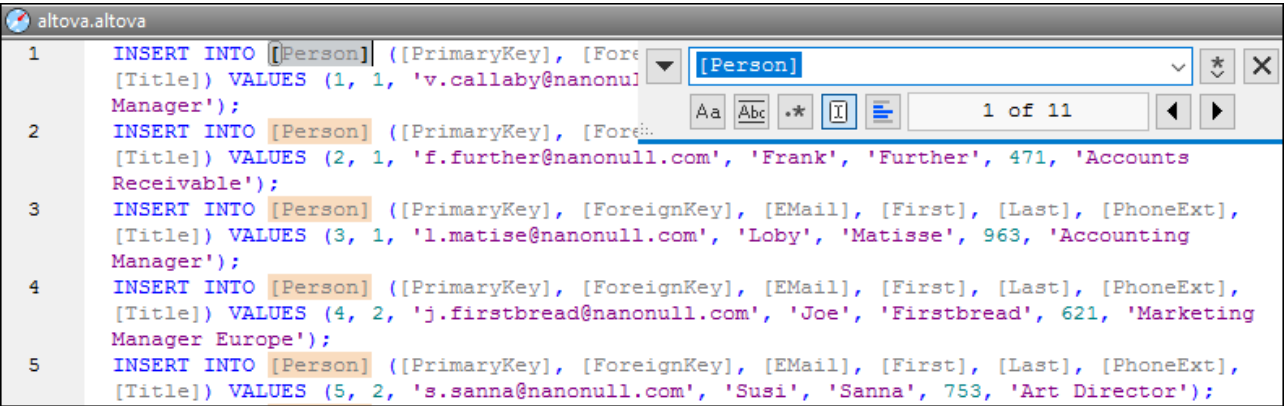

Find options
You can specify find criteria with the help of the buttons located under the search field. The list of the available options is given in the table below.

Option	Icon	Description
Match case	Aa	Does a case-sensitive search: e.g., *Address* is not the same as *address*.
Match whole word	Abc	Only identical words will match.
Use regular expression	.*	If this option is toggled on, the search term will be read as a regular expression. See *Regular expressions* below.
Find anchor	I	The position of an anchor depends on the place where the cursor was last located. Clicking the **Previous** and **Next** buttons does not change the position of the anchor.
Find in selection	≡	A selection is a marked piece of text. To find a term within a selection, mark a piece of text, press **Ctrl+F**, make sure the **Find in selection** button is pressed, and type the term in the search field.

Regular expressions

You can use regular expressions to find a text string. To do this, switch on the **Use regular expressions** option (*see table above*). Then enter a regular expression in the search field. Clicking (**Regular Expression Builder**) gives you a list of sample regular expressions (*see below*). The screenshot below shows a regular expression that helps find email addresses.
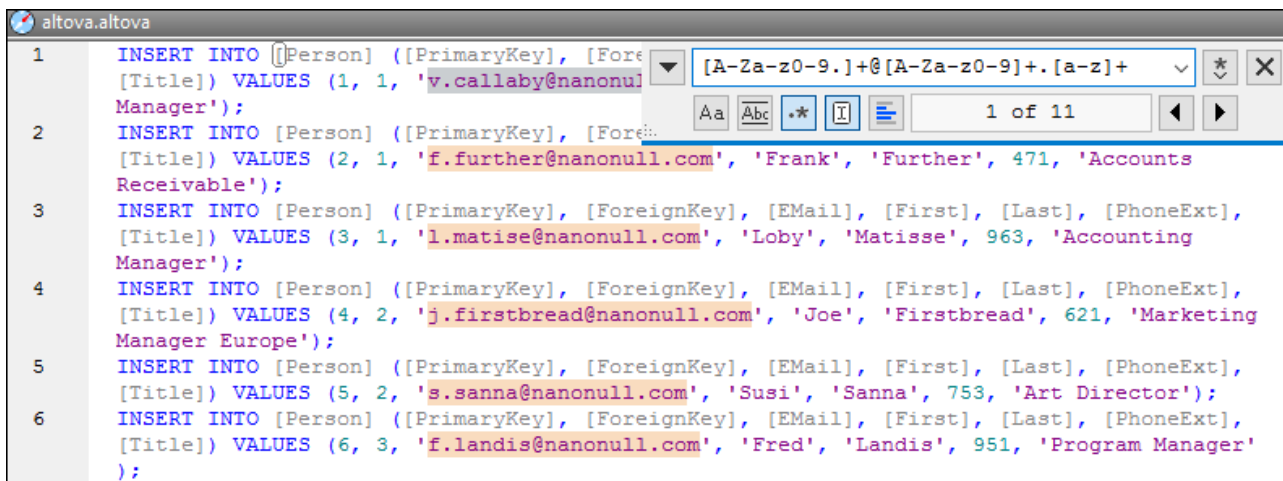
Regular expression metacharacters
The table below shows metacharacters that you can use to find and replace text. All the metacharacters except for the last two correspond to the menu items in **Regular Expression Builder** (*see above*).

Menu item	Metacharacters	Description
Any Character	.	Matches any character. This is a placeholder for a single character.
Character in Range	[...]	Matches any characters in this set. For example, **[abc]** matches any of the characters a, b or c. You can also use ranges: e.g., **[a-z]** for any lower case character.
Character Not in Range	[^...]	Matches any characters not in this set. For example, **[^A-Za-z]** matches any character except an alphabetic character.
Beginning of Word	\\<	Matches the beginning of a word.
End of Word	\\>	Matches the end of a word.
Beginning of Line	^	Matches the beginning of a line unless it is used inside a set (*see above*).
End of Line	$	Matches the end of a line. For example, **A+$** matches one or more A's at the end of a line.
Tagged Expression	(abc)	The parentheses mark the start and end of a tagged expression. Tagged expressions may be useful when you need to tag

Menu item	Metacharacters	Description
		("remember") a matched region to refer to it later. Up to nine sub-expressions can be tagged and then back-referenced later. For example, **(the) \1** matches the string the the. This expression can be explained as follows: Match the string the and remember it as a tagged region; the expression must be followed by a space character and a back-reference to the tagged region matched previously.
0 or More Matches	*	Matches zero or more matches of the preceding expression. For example, **Sa*m** matches Sm, Sam, Saam, Saaam and so on.
1 or More Matches	+	Matches one or more occurrences of the preceding expression. For example, **Sa+m** matches Sam, Saam, Saaam and so on.
	\n	Where n is 1 through 9, n refers to the first through ninth tagged region (*see above*). For example, if the search string is **Fred\([1-9]\)XXX** and the replace string is **Sam\1YYY**, when applied to **Fred2XXX**, this will generate **Sam2YYY**.
	\x	Allows using a character x, which would otherwise have a special meaning. For example, \[would be interpreted as [and not as the start of a character set.

Replace text

To replace text, follow the instructions below:

1. Open the **Replace** dialog: Press **Ctrl+H** or select the menu command **Edit | Replace**. Alternatively, you can switch to **Replace** mode of the **Find/Replace** dialog by clicking the ▼ button in the top left corner of the dialog.
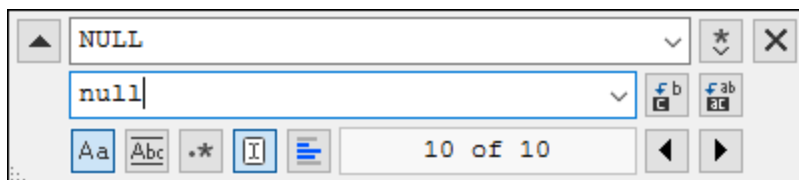
2. Enter the string you would like to replace in the **Find** field and enter the new string in the **Replace** field.
3. The **Replace Next** and **Replace All** buttons are located to the right of the **Replace** field. If you click **Replace Next**, one of the following happens: (i) If the cursor is located near a match or inside a match, the match is replaced; (ii) if the cursor is located outside a match, the cursor jumps to the next match. Click **Replace All** to replace all matches.

Replace text within a selection
To replace text within a selection, take the steps below:

1. Mark a piece of text.
2. Toggle on the **Find in selection** option.
3. Enter the **Find** and **Replace** text strings.
4. Click **Replace Next** or **Replace All** as required.

To undo the replace action, press **Ctrl+Z** or select **Edit | Undo**.

Find and replace special characters

If the **Use regular expressions** option is enabled, you can search for any of the following special characters within the text:

- \t (Tab)
- \r (Carriage Return)
- \n (New line)
- \\ (Backslash)

For example, to find a tab character, press **Ctrl + F**, select the **Use regular expressions** option, and enter \t in the **Find** dialog box.

The special characters above can also be used as replacement text. For example, to replace all occurrences of semi-colons by carriage returns within the text A;B;C;D, use the following find and replace options:


The outcome is shown below:

A
B
C
D

9.11   **Selecting Data for Export**

In addition to DatabaseSpy's powerful export function [406], you can use SELECT statements to retrieve data from a table and export only this particular data to a range of file formats such as XML or HTML. The SQL Editor window provides an additional **Export** button to open the **Export database data** dialog box to facilitate this. When you click this button, the dialog box opens with the **SQL** radio button activated and the content of the SQL Editor window inserted into the Source group box.
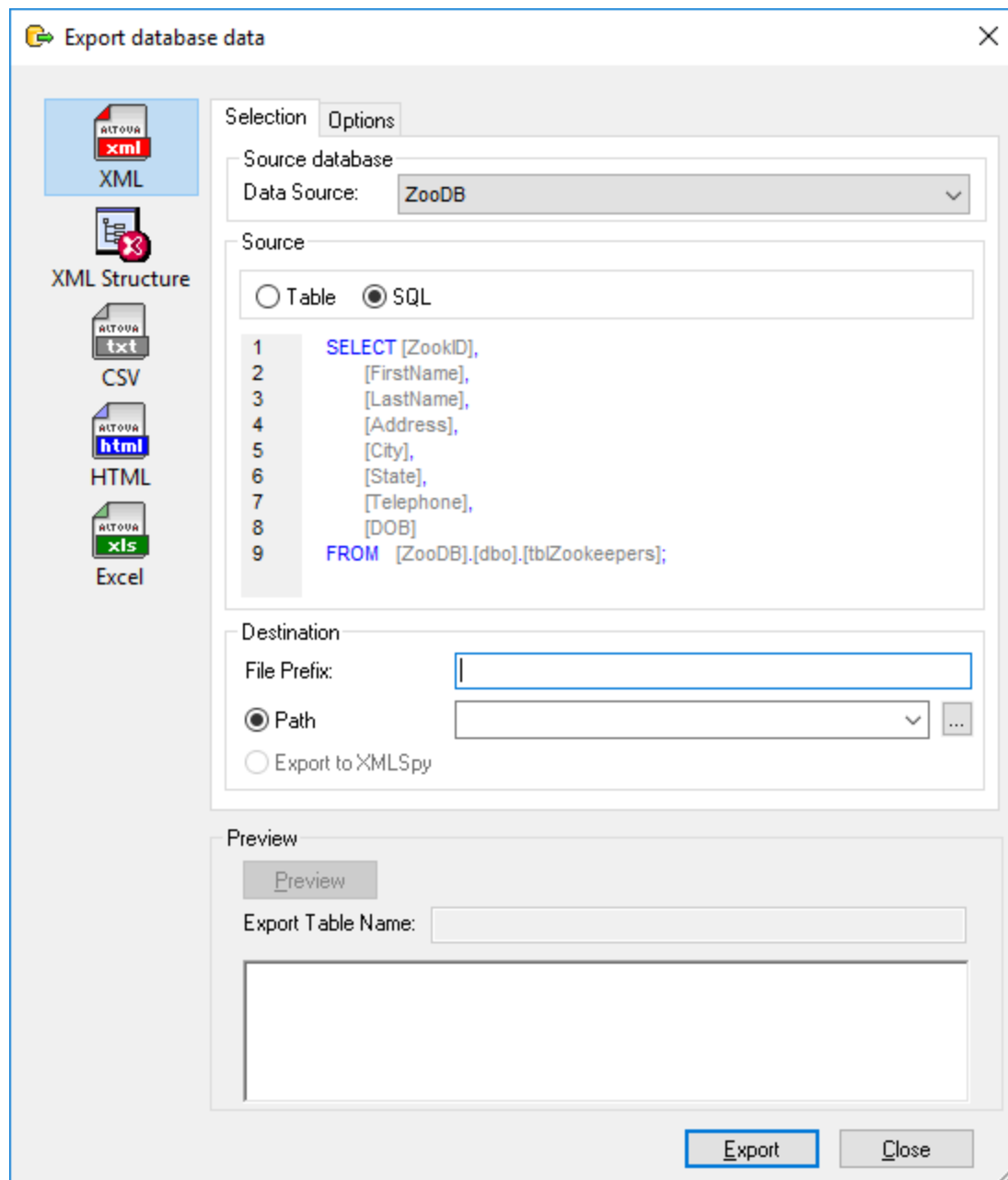
Export database data dialog box

Note: Clicking the **Export** ![icon] button in the Tools toolbar will also open the Export database data dialog box. However, in this case the Table radio button will be preselected and you would have to type in or paste the SQL code manually.

**To export data using SQL statements in an SQL Editor window:**

1.   Make the SQL Editor window whose SQL statements you want to use for export the active window.
2.   Press the **Export** ![icon] button in the SQL Editor window. The Export database data dialog box opens. The SQL statements that are currently in the SQL Editor appear in the Source group box.
3.   Optionally, enter a file prefix that will be added to the file name.
4.   Do one of the following:

     - Select the **Path** option and enter the Path you want the destination files to appear in.
     - Select the **Export to XMLSpy** option if you want your exported data to be displayed in XMLSpy. Note that this option is not available for export to Excel files, or if XMLSpy is not installed on your computer.

5.   Optionally, click the **Options** tab to change the options for XML, XML Structure [412], CSV [414], HTML [415], or Excel [416].
6.   Click **Export**.

9.12   Printing SQL

You can print the content of the SQL Editor like in any other text editor. When turning on the line numbers margin, you can also print out line numbers.

Note: The source folding margin has no effect on the print-out (that is, the expand/collapse icons will not show up in the print-out). If any bookmarks are defined in the bookmark margin, the line where the bookmark occurs will appear highlighted on the printed page.

A print preview is available for you to check the layout of the printed page before sending it to the printer.
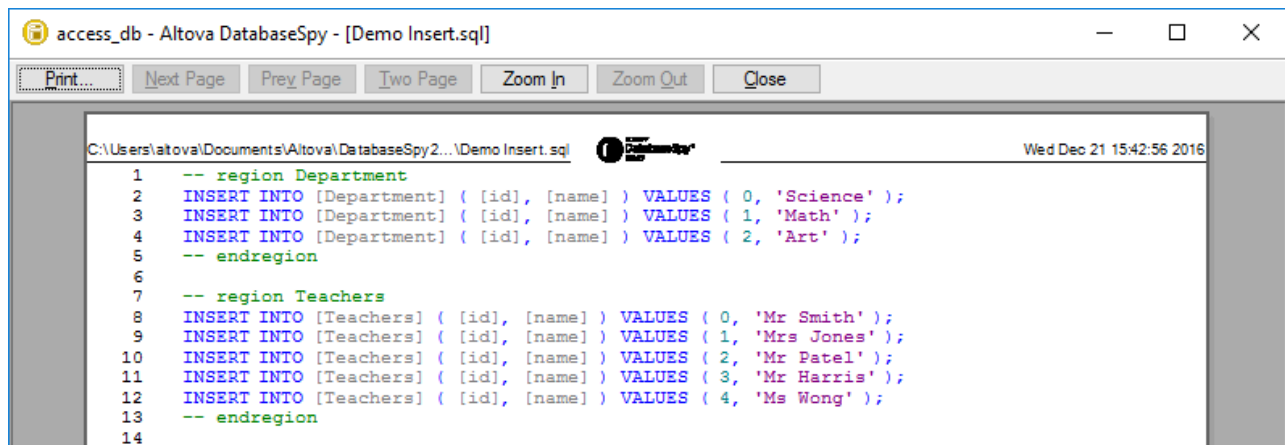
To print the content of an SQL Editor window:

1. Click in the SQL Editor window to make it active.
2. Select the menu option **File | Print...**, or press **Ctrl+P**, or click the **Print** icon in the Standard toolbar.
3. In the **Print range** group box, define whether you want to print all or specific pages or only the selected part of the SQL.
4. Optionally, define the printer setup by clicking the **Properties** button.
5. Click **OK**.

To preview SQL before printing:

1. Click in the SQL Editor window to make it active.
2. Select the menu option **File | Print Preview**.
3. In the print preview, do one of the following:
    o Click the **Print** button to print the file.
    o Click **Close** to return to the SQL Editor.

9.13   SQL Templates

With DatabaseSpy Enterprise Edition, you can create and run SQL statements which contain parameters (subsequently referred to as "SQL Templates"). The only difference between standard SQL queries and SQL templates is that the latter contain parameter placeholders. For example, the SELECT statement below is a template which defines two parameters, **:color** and **:size**.

```
SELECT id FROM products WHERE color = :color AND size > :size;
```

You can define parameters not only in SELECT statements but also in other statement types (such as INSERT or UPDATE). For example, an INSERT statement with parameters could look like:

```
INSERT INTO users (first_name,last_name,email,created_at,updated_at) VALUES
(:fname,:lname,:email,CURRENT_TIMESTAMP,NULL);
```

INSERT templates such as the one above can greatly simplify the transfer of large amounts of data from one table or view to another (even across various databases). Specifically, when you need to populate a table, multiple INSERT statements can be generated in a few simple steps from the given template, as further described below.

Note the following:

- In order to be recognized by DatabaseSpy as such, parameters within SQL statements must be preceded by a colon (:) character.
- Parameters may substitute values inside statements; however, in order to prevent SQL injections, they may not be used to substitute SQL reserved keywords or identifiers (such as column or table names).
- Parameters are by default treated as text, and parameterized queries will fail with certain database connections unless you explicitly change the parameter type (see the next section below, "Changing the parameter type").

Changing the parameter type

By default, parameters are treated as type `text` or `varchar`, which makes parameterized queries fail with certain database drivers (for example, with ADO.NET or JDBC connections to SQL Server). In such cases, you can explicitly specify the type, and, if applicable, the precision, scale, or the size of the parameter by adding a comment immediately after the parameter.

For string types, consider the following example:

```
SELECT [col1],
       [col2]
FROM   [mytable] WHERE col1 = :p1 /*# varchar, size:max */
```

In the code listing above, the parameter type is explicitly set as `varchar` and its **size** is set to `max`. The modifier **length** can also be used instead of **size**.

For numeric types, consider the following example:

```
SELECT [col1],
       [col2]
FROM   [mytable] WHERE col1 = :p1 /*# decimal, precision:23, scale:4 */
```

In the code listing above, the type of parameter `p1` is explicitly set as `decimal`, the **precision** (total number of digits) is set to 23, and the **scale** (number of digits right of the decimal point) is set to 4. You can also use the short name **prec** for precision. Both **precision** and **scale** require integer values.

Below are a few other examples of attributes that may follow a parameter (make sure to adjust the details as required by your database):

```
/*# float */

/*# decimal, precision:23, scale:4 */

/*# decimal, prec:23, scale:4 */

/*# varchar, size:max */

/*# nvarchar, size:32 */

/*# varchar, length:255 */

/*# number, precision:38, scale:0 */
```

Note the following:

- The character # explicitly instructs DatabaseSpy to handle the corresponding parameter with the declared type and size.
- The data type and the attributes following it must be separated by a comma. Each attribute must also be separated by a comma.
- Attributes other than **prec** or **precision**, **scale**, **size** or **length** will not be recognized. All, some, or none of these attributes may be required in order to run the parameterized query successfully, depending on the database driver and how much information it is able to determine automatically when the statement is prepared.
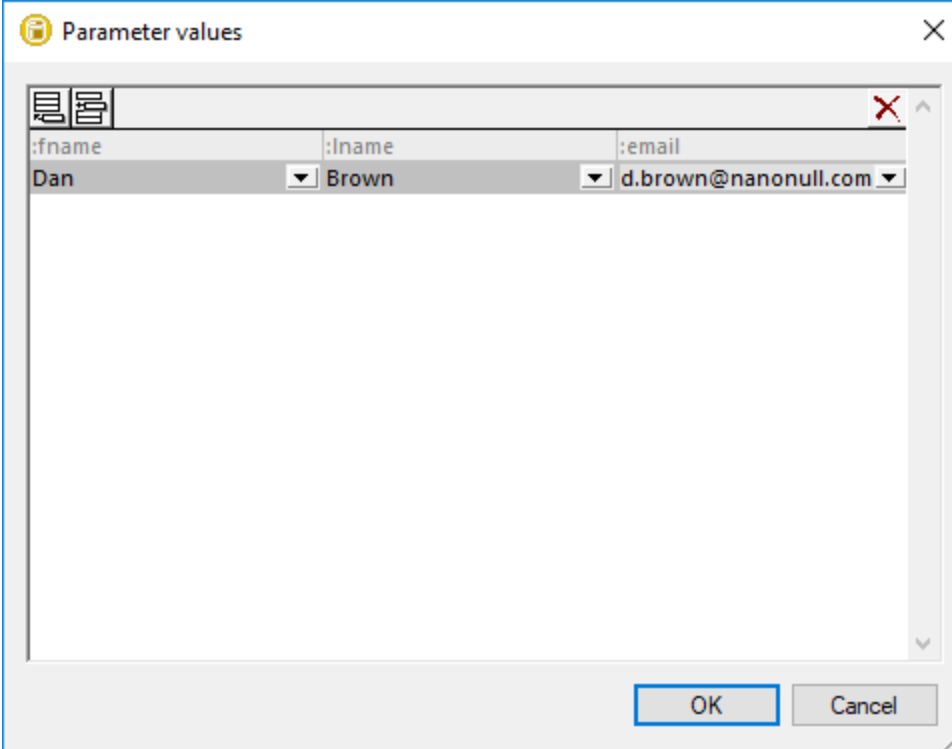
Running a query with parameters

1. Enter or generate a SQL statement in SQL Editor, and make sure that it contains at least one parameter. For example, the INSERT statement below contains three parameters:
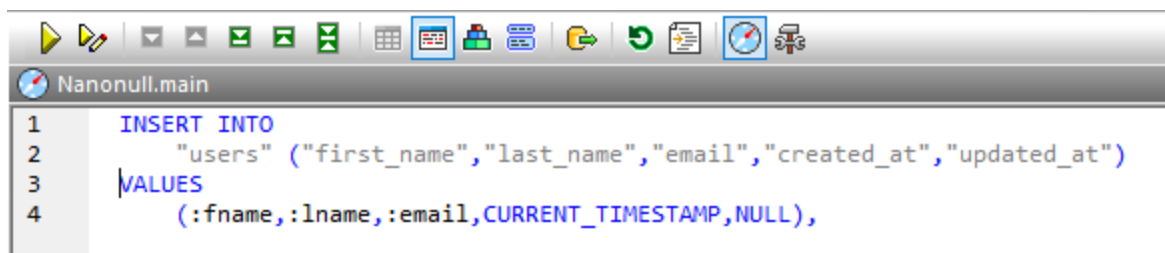

2. Click the **Execute** or **Execute for Data Editing** button in the SQL Editor's toolbar.

3.   Fill in the required parameter values, for example:

You can also supply multiple parameter rows to the query. To do this, click either the **Append New** ⊞ or **Insert New** ⊟ button. The **Append New** button adds a parameter row after all existing rows, while **Insert New** inserts a parameter at the current position (for example, between two existing rows).
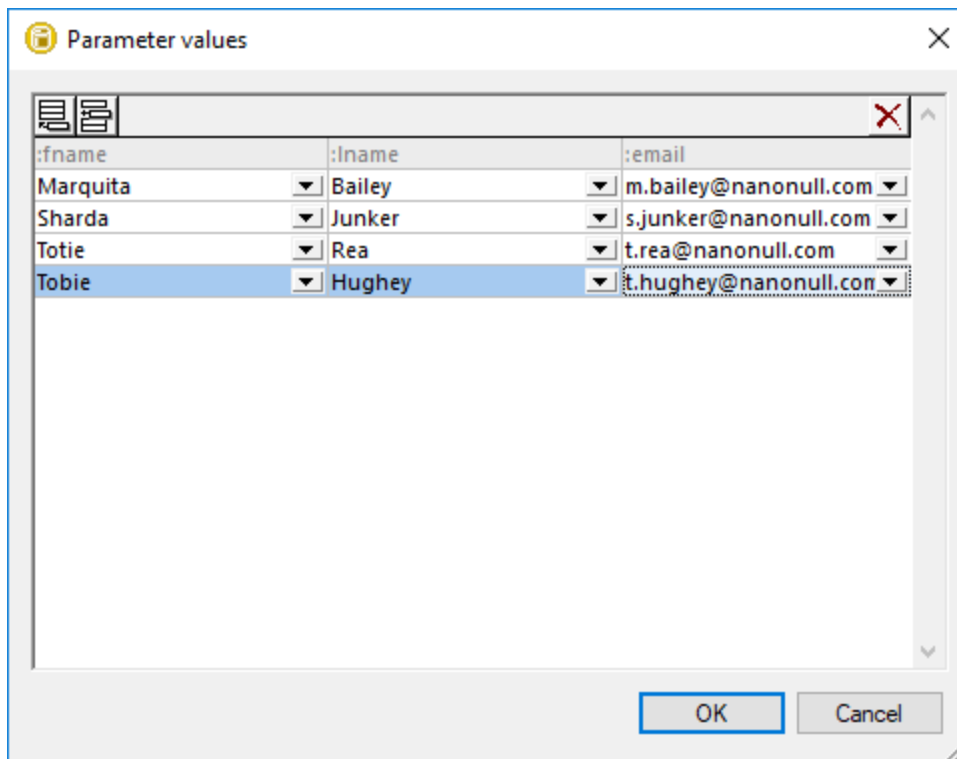
4. Click **OK**. DatabaseSpy executes the query with supplied parameters and displays the execution
    result in the **Result** tab.

Note the following:

-   If the SQL Editor contains multiple SQL statements, the "Parameter values" dialog box contains a tab
    for each SQL statement where parameters exist. Also, multiple **Result** tabs are displayed after you run
    the query (one **Result** tab for each statement containing parameters).
-   DatabaseSpy remembers the last 25 parameter values entered in the "Parameter Values" dialog box.
    To use a previous value, select it from the drop-down list available for each parameter.

Saving parameterized queries as SQL Template

A query which contains parameters can be saved for later use, similar to how this works for standard SQL files.
To save the contents of the SQL Editor as SQL template, do one of the following:

-   In the **File** menu, click **Save** (or **Save As**).
-   Click the **Save** toolbar button.
-   Press **Ctrl + S**.

SQL template files are saved with .sql extension, like other SQL scripts. To help you distinguish between SQL
templates and other SQL scripts, the DatabaseSpy project contains a specially designated folder called "SQL
Templates".
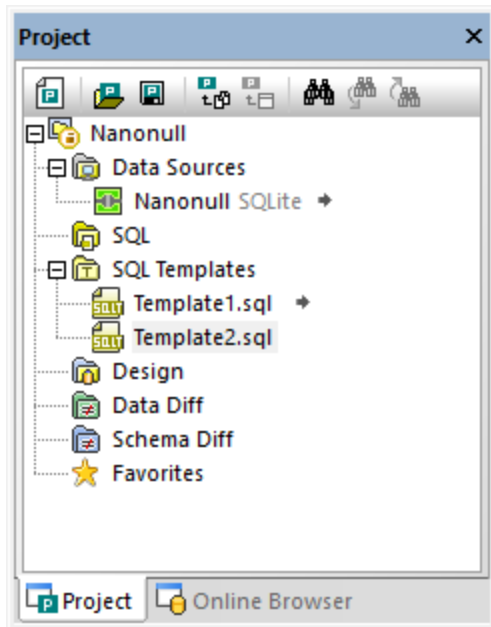
All SQL templates created for the project must be in the "SQL Templates" folder. Only when a SQL template is inside the "SQL Template" folder can DatabaseSpy execute it with values, as further described below.

Executing SQL templates

You can execute SQL templates against sets (rows) of dynamically supplied values. Executing a SQL template means that DatabaseSpy replaces all parameter placeholders found in the template with the values you supply, creates a new SQL statement for each value (or row of values, depending on the case), and runs it against the database. As a result, it is possible to generate and run a very large number of SQL statements with a click of a button.

Prerequisites

1. Create a SQL Template by typing it in the SQL Editor window. For example, the SQL template below takes three parameters: `:fname`, `:lname`, and `:email`.

```
INSERT INTO users (first_name,last_name,email,created_at,updated_at) VALUES
(:fname,:lname,:email,CURRENT_TIMESTAMP,NULL);
```

As an alternative to typing the statement, you can also right-click a table in the Online Browser, generate the INSERT statement, and modify it to add the required parameters, see Generating SQL Statements[346].

2. Save the SQL Template and add it to your DatabaseSpy project. The SQL template must be inside the "SQL Templates" folder of the project.

At this stage, you can also test whether the new SQL Template executes successfully, by running it with some manually-entered parameters. To test the SQL template, click **Execute** ▷, and enter the required parameters when prompted, see Running a query with parameters ⁽³⁸³⁾.

To execute a template with values:

1. Connect to a data source and retrieve data from the table which contains the values you want to supply as input to the SQL template (for example, right-click a table in the Online Browser, and select **Edit Data** from the context menu).
2. Hold the **Ctrl** key pressed and select one or more cells from the **Results** grid. To select multiple columns, click on their header while holding the **Ctrl** key pressed. The SQL template will be executed as many times as there are rows in the selection. Note that the number of selected columns must correspond to the number of parameters that the SQL template accepts, therefore:

   o If the template accepts only one parameter, all the selected cells (one or more rows) must be in the same column.
   o If the template accepts multiple parameters, there must be as many selected columns as there are parameters.

   For example, for an INSERT template with three parameters like the one illustrated above, the following selection is valid:

	id	first_name	last_name	email	created_at	updated_at
1	1	Marquita	Bailey	m.bailey@nanonull.com	2017-08-14 08:36:39	[NULL]
2	2	Sharda	Junker	s.junker@nanonull.com	2017-08-14 08:36:39	[NULL]
3	3	Totie	Rea	t.rea@nanonull.com	2017-08-14 08:36:39	[NULL]
4	4	Tobie	Hughey	t.hughey@nanonull.com	2017-08-14 08:36:39	[NULL]
5	5	Eadith	Lafreniere	e.lafreniere@nanonull.com	2017-08-14 08:36:39	[NULL]

   A selection of multiple rows which are not necessarily consecutive is also valid, for example:

	id	first_name	last_name	email	created_at	updated_at
1	1	Marquita	Bailey	m.bailey@nanonull.com	2017-08-14 08:36:39	[NULL]
2	2	Sharda	Junker	s.junker@nanonull.com	2017-08-14 08:36:39	[NULL]
3	3	Totie	Rea	t.rea@nanonull.com	2017-08-14 08:36:39	[NULL]
4	4	Tobie	Hughey	t.hughey@nanonull.com	2017-08-14 08:36:39	[NULL]
5	5	Eadith	Lafreniere	e.lafreniere@nanonull.com	2017-08-14 08:36:39	[NULL]

3. Right-click the selection and select **Execute template with values | <name of the template>** from the context menu. All generated statements are displayed in the SQL Editor and executed.

9.14   Conditional Formatting

When you run SELECT statements in the SQL Editor, the retrieved data is displayed in the Result window [41] (or multiple Result windows, if multiple queries were executed). You can change the appearance of retrieved data (such as color, background, font weight) based on custom SQL conditions. For example, you can have certain values in the Result grid formatted with yellow background if they exceed some threshold.

The image below illustrates an example of conditional formatting in a SQLite database. This example project is available at the following path: **C:\Users\<username>\Documents\Altova\DatabaseSpy2023\DatabaseSpyExamples\Tutorial\NanonullEnterprise.qprj**. In this example, the goal is to apply conditional formatting as follows:

- Records where quantity exceeds 200 must have coral background and bold font
- Records where quantity is less than 10 must have turquoise background and italic font.

Conditional formatting in SQLite syntax

As illustrated above, in order to support conditional formatting, the SQL statement must include an *alias* column that must be conventionally named **DbSpyRowSettings**. This instructs DatabaseSpy to process this

column as conditional formatting instructions. Note that the alias column consists of a "CASE" construct. The WHEN branch is the SQL expression where you supply the condition to check for, in the corresponding database grammar. The THEN branch is a string that specifies formatting styles such as font weight or background color. This string must be quoted, in the grammar specific to the database, so that it looks like a string to the database.

> The SQL syntax that defines an alias column depends on the database type. See below for examples specific to various database kinds. For details, refer to the documentation of the database vendor.

The styles you can use are as follows:

Style	Remarks	Example

Text style	Text can formatted as bold, italic or underline.	`font:bold`
		`font:italic`
		`font:underline`

Font size	The font size can be expressed in fixed units (the same units as those in the Options [430] dialog box) or as a percentage. A percentage value is relative to the font size defined in Options dialog box.	`font-size:10` `font-size:150%`
Font family	Unlike other styles, font names are case-sensitive. Font names may be quoted as long as the SQL statement remains valid.	`font-family:Arial`
Text color	Text color can be expressed as a hexadecimal value (case insensitive) or as a named color. For reference to all the named colors you can use, see https://www.w3.org/TR/2003/REC-SVG11-20030114/types.html, Section 4.2.	`color:yellow` `color:#FFFF00`
Background color	Background colors can have the same values as text colors.	`background-color:gold` `background-color:#FFD700`

Remarks:

- You can use multiple styles separated by semi-colons, for example: `background-color:gold; font: bold; color: blue;`. Whitespace between styles is not relevant.
- If the styles you entered are not recognized, DatabaseSpy will ignore them without displaying any errors or warnings.
- The **DbSpyRowSettings** column may appear at any location in the column list.
- The case of the **DbSpyRowSettings** column or the case of styles is not relevant (except for font names, see above).
- When a conditionally formatted cell, column or row is in focus (selected), the background color has no effect. Only text color and font are applied.
- For better display results, you might want to clear the **Show grid with alternating colors** check box in the Result View [429] options.

Conditional formatting in various database kinds

The following are some database-specific examples of SQL statements that use conditional formatting. Note that the examples below illustrate just some of the ways to apply conditional formatting. If supported by the database, it is also possible to achieve the same result with more complex techniques, such as sub-queries, stored procedures, and so on.

Database Kind	Example

Access	Use the SWITCH function, see https://support.office.com/en-us/article/Switch-Function-D750C10D-0C8E-444C-9E63-F47504F9E379
	
	``
	SELECT [id],
	       [name],
	       [quantity],
	       SWITCH ([quantity] > 200, 'color:blue') AS [DbSpyRowSettings]
	FROM [products];
	``
	
	Alternatively, use the IIF function, see https://support.office.com/en-us/article/Iif-Function-32436ECF-C629-48A3-9900-647539C764E3
	
	``
	SELECT [id],
	       [name],
	       [quantity],
	       IIF ([quantity] > 200, 'color:blue', '') AS [DbSpyRowSettings]
	FROM [products];
	``

IBM DB2
IBM DB2 for i
Oracle
PostgreSQL
Progress OpenEdge
SQLite
Sybase
Teradata
	``
	SELECT "id",
	       "name",
	       "quantity",
	       CASE WHEN "quantity" > 200 THEN 'color:blue;' END AS
	"DbSpyRowSettings"
	FROM "products";
	``

IBM Informix	```
	SELECT id,
	       name,
	       quantity,
	       CASE WHEN quantity > 200 THEN 'color:blue;' END AS
	DbSpyRowSettings
	FROM products;
	``

MariaDB
MySQL
	``
	SELECT `id`,
	       `name`,
	       `quantity`,
	       CASE WHEN `quantity` > 200 THEN 'color:blue;' END AS
	`DbSpyRowSettings`
	FROM `products`;
	``

SQL Server	```
	SELECT [ProductID],
	       [Name],
	       [Quantity],
	``

Database Kind	Example
	`[DbSpyRowSettings] = ` **`CASE WHEN`** ` [Quantity] > 200 ` **`THEN`** ` 'color:blue;'` **`END`** **`FROM`** `    [Products];`

10   XML Management

Some of the databases supported by DatabaseSpy provide XML data types and other XML-related features. In DatabaseSpy, support for the XML-related features across various database kinds can be summarized as follows:
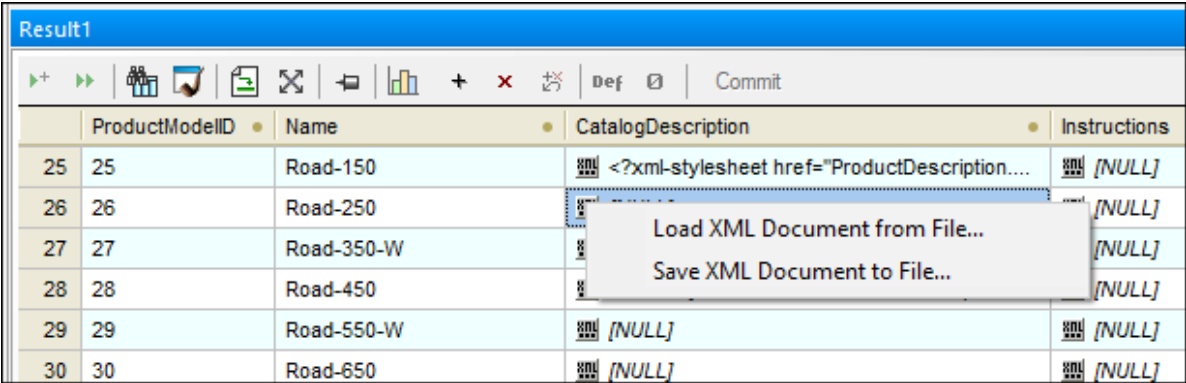
Feature	Description	Database support
Editing of XML columns	If a database column is of type XML, you can load XML instance files into the database fields, or save the content of XML database fields to local files, see Editing XML Columns [393]	• DB2 9.1 and later • Oracle • PostgreSQL 8.3 and later • SQL Server
XML Schema management	You can register or drop database XML schemas through a dedicated dialog box, see Managing XML Schemas [397]	• DB2 9.1 or later • Oracle • SQL Server
Assign XML schema to a column	See Assigning XML Schemas [399]	• SQL Server
Assign XML schema to a data field	See Assigning XML Schemas [399]	• DB2 9.1 and later

10.1   Editing XML Columns

Columns of type XML cannot be edited directly like other column types. Instead, you can load XML instance files to a cell, or save the existing XML instances from a cell to a file.

To load an XML instance into a cell:

1. In Online Browser, right-click a database table that contains XML columns, and select **Edit Data**. Alternatively, generate a SELECT statement [348] in a new SQL Editor window and click the **Execute for Data Editing** [icon] button.

2. Click the **Browse** [icon] button inside the data cell and select **Load XML Document from File** from the context menu. Note that the [icon] button is displayed only for columns of XML type.

3. In the Open dialog box, browse for the desired XML file.
4. Click **Commit**.

To save an XML instance from a cell to a file:

1. Click the Browse [icon] button inside the data cell and select **Save XML Document to File** from the context menu.
2. In the Save As dialog box, select a destination path or name for the XML file, and click **Save**.

Note: DB2 databases include support for assigning an XML schema to a cell. In such databases, the **Assign XML Schema** command is additionally present in the context menu, see Assigning XML Schemas [399] .

10.2   Viewing XML Schemas

In the Online Browser, all XML schemas belonging to particular database schema are displayed in the "XML Schemas" folder.
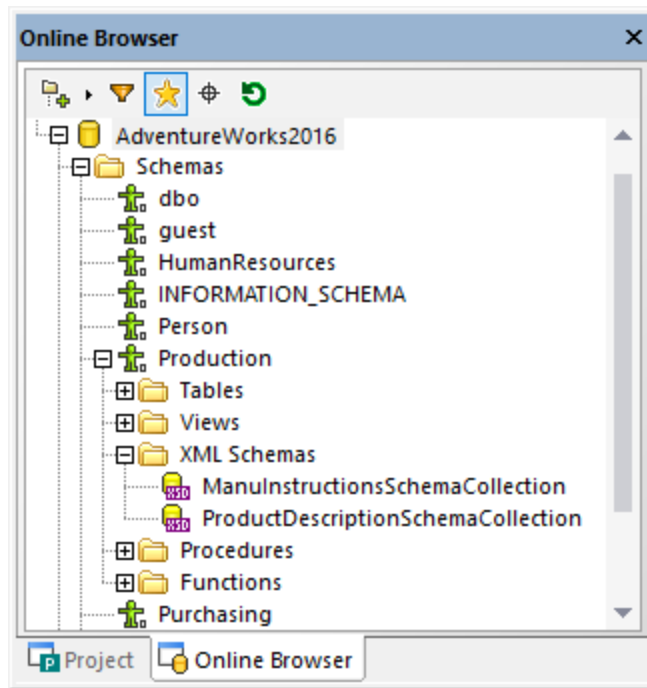
If an XML schema has dependencies, the dependent schemas are displayed in a new "XML Schemas" folder nested under the parent schema.

All the XML schemas assigned to an XML column are displayed in the Online Browser in an "XML Schemas" sub-folder below the respective column. This way, you can see at a glance which XML schemas are used to validate the content of a particular XML column in the database.

The properties of XML schemas can be viewed and—where possible—edited in the Properties window.

Editing a property (for example "Description") generates a script in the Database Structure Change Script Window [37]. To update the database, click the **Execute the Generated SQL Change Script** ▷ button.

Viewing XML schemas in XMLSpy

If Altova XMLSpy Editor (https://www.altova.com/xmlspy-xml-editor) is installed on your PC, you can open an XML schema in XMLSpy, as follows:

- In the Online Browser, right-click a schema from the "XML Schemas" folder and select **View in XMLSpy** from the context menu.

- In the Online Browser, right-click a schema and select **Manage XML schemas** from the context menu. In the dialog box that opens, select a schema and click the **View Schema** button.

10.3   Managing XML Schemas

To view, add, or drop XML schemas in a database, DatabaseSpy provides the **XML Schema management for databases** dialog box. To open this dialog box, do one of the following:

- Click the **XML Schema Management** 🗓 toolbar button.
- Select the menu command **Tools | XML Schema Management for Databases**.
- Right-click an XML schema in the Online Browser and choose **Manage XML Schemas...** from the context menu.

If you are not connected to a data source, the dialog box appears in offline state. To connect to a database, click the **Connect to a database** 🖳 button and follow the wizard steps. On successful connection to a database that supports XML, the list of available XML schemas appears in the dialog box, for example:
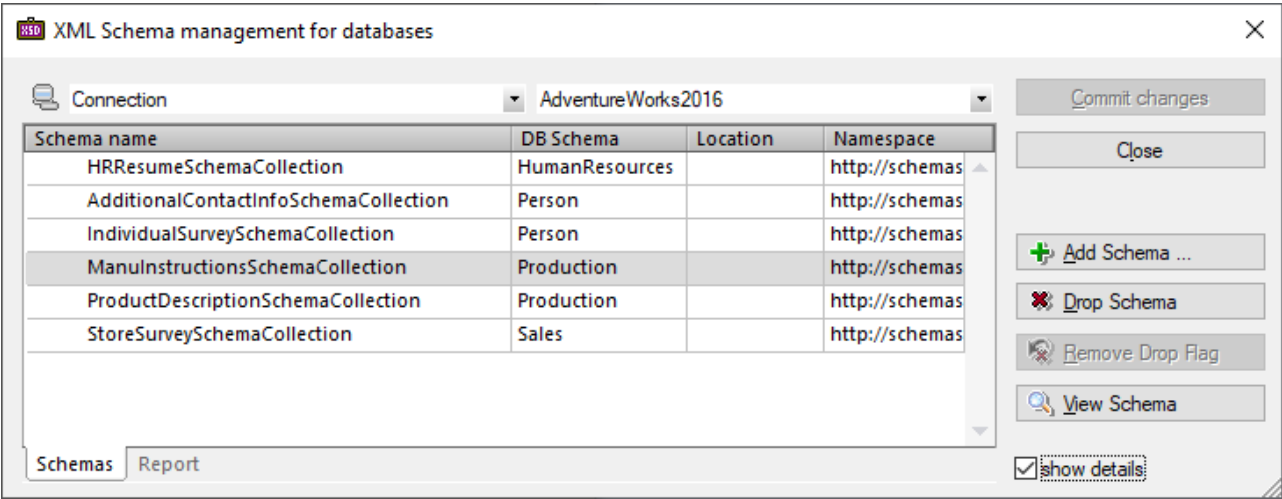
You can now add schemas, drop schemas, or view a schema in XMLSpy by clicking the corresponding buttons. You can also edit a schema's description (and "Decomposition" property if applicable) directly from the grid.

Note that any changes you make on the dialog box are not immediately committed to the database. Edited objects are marked as new (➕), changed (🔲), or deleted (✖) in the grid. You must click **Commit changes** in order for the changes to become effective. To undo the deletion of a schema while the change is not committed, first select the schema and then click 🗙 **Remove Drop Flag** .

Commit changes	Sends an SQL transaction to the database to make the changes on the dialog box permanent. This button is disabled if no changes have been made yet.
Close	Closes the dialog box. If there are uncommitted changes, a message box asks for your confirmation to commit or discard them.
➕ Add Schema	Opens a dialog box where you can select a new XML schema to be registered with the database. If the schema has references to other schemas, these will be registered as well.

✖ **Drop Schema**	Marks the schema for deletion. The actual deletion occurs only after you click **Commit changes**.
🔖 **Remove Drop Flag**	Undoes the deletion of the schema. This button is enabled only for schemas that have been marked for deletion.
🔍 **View Schema**	Opens the XML schema (for viewing) in Altova XMLSpy. Note that any schema changes made from XMLSpy are saved locally and do not affect the database.

To display additional information about the schema, select the **Show details** check box. To resize the dialog box, click and drag its lower-right corner outwards.

After you click **Commit changes**, the outcome of the SQL transaction is reported in the **Report** tab.

> Your database user account must have the relevant database rights in order to add or drop XML schemas; otherwise, the SQL transaction fails with an error.

All the functions in this dialog box are also available as a context menu when you right-click anywhere in the data grid.

10.4   Assigning XML Schemas

You can assign an XML schema to a database column (or, in IBM DB2 databases, to a data field) of type XML. This schema is used for data validation when an INSERT or UPDATE action is performed.

To assign an XML schema to a database column or field of type XML:

1. Depending on the database type, do one of the following:

    *   In DB2 databases, and provided that the Result window is in the Editing mode, click the Browse 🗟 button in the XML data cell you want to edit, and choose **Assign XML Schema** from the menu that appears.
    *   In SQL Server databases, right-click the column in the Online Browser and choose **Assign XML Schema...** from the context menu.
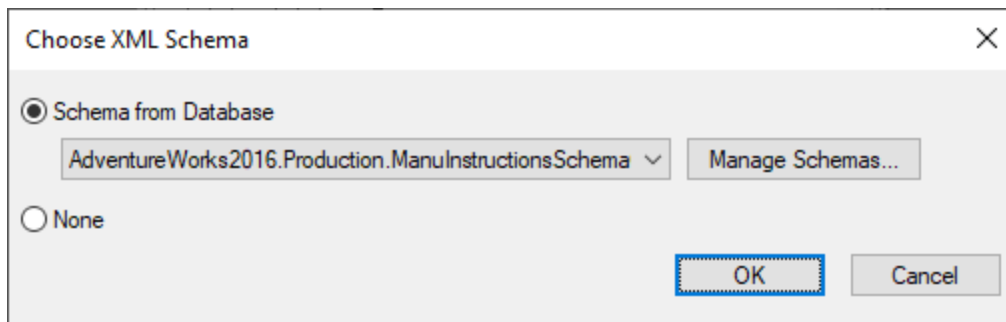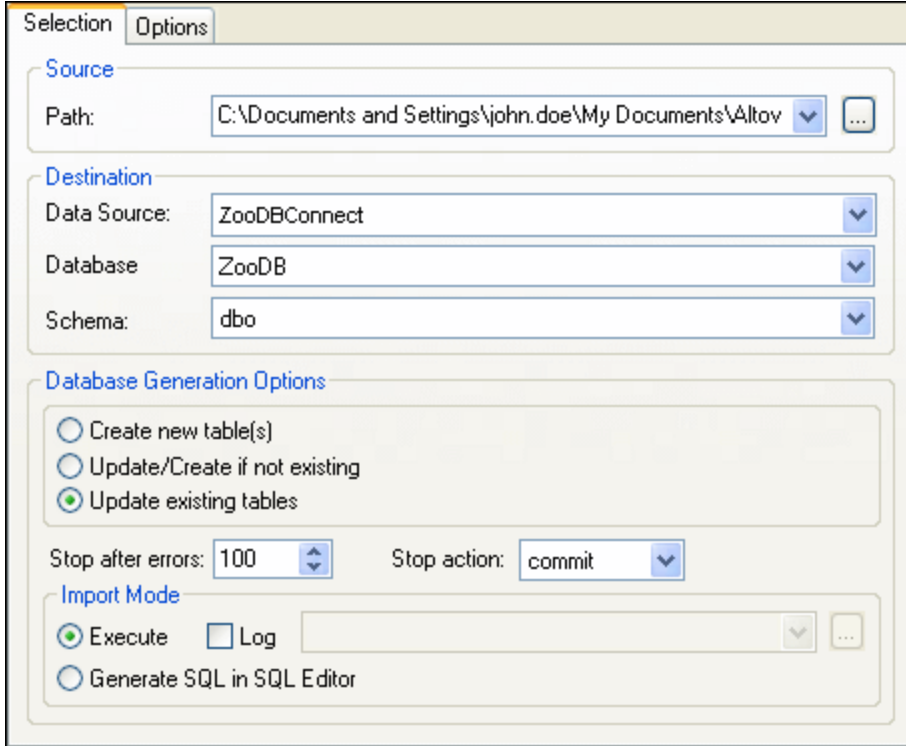
2. Do one of the following:

    *   Choose one of the XML schemas from the drop-down list (the list contains XML schemas registered with the database). Optionally, click **Manage Schemas** to add a schema to the database.
    *   To remove any XML schema assignment from the column (or data cell if applicable), select **None**.

3. Click **OK**.
4. Depending on the database type, do one of the following:

    *   In DB2 databases, click the **Commit** button in the Result Window [41].
    *   In SQL Server databases, execute the change script in the Database Structure Change Script Window [37].

At this stage, DatabaseSpy validates the XML column or data cell against the assigned XML schema and reports any errors in the Message Window [41]. If the validation fails, the XML schema assignment cannot be committed to the database.

11   Importing Database Data

With DatabaseSpy, you can import data from XML or CSV files to your databases. When you click the **Import** button in the Tools toolbar or select the menu option T**ools | Import data to the database... (Ctrl+I)**, the "Import data to the database" dialog box opens where you can select the XML or CSV file to be imported and define the import options. The "Import data to the database" dialog box contains two tabs, **Selection** and **Options**.
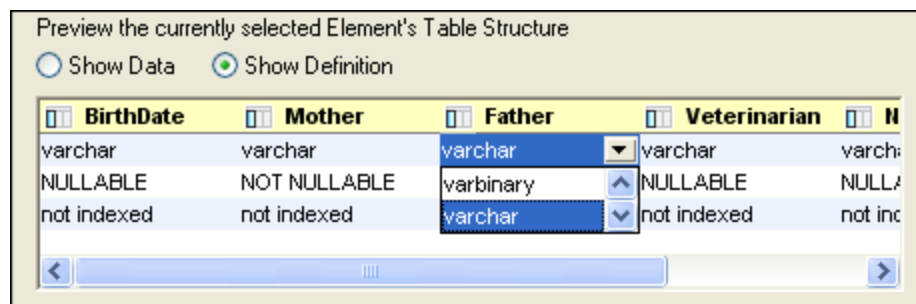
The lower part of the dialog box always shows a preview of the selected data, which can be updated to reflect the selected options.
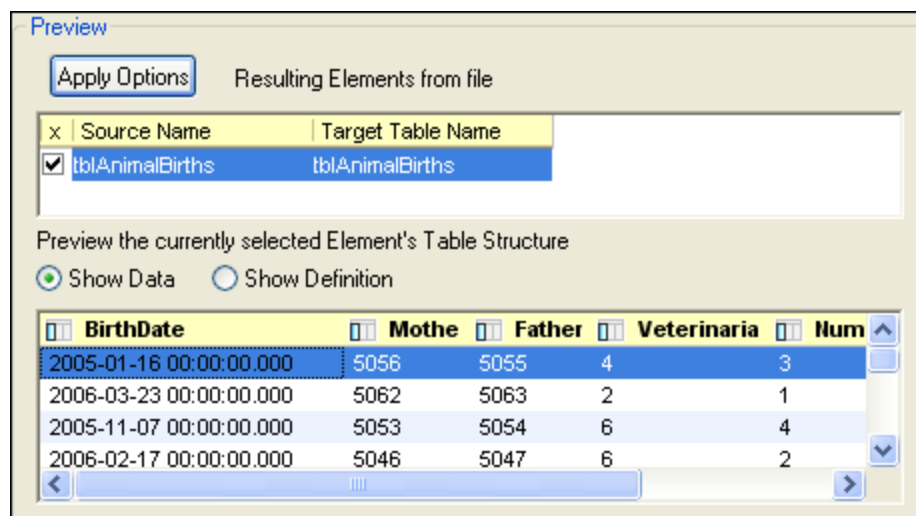
You can choose to preview the data (*screenshot above*) or the definition (*screenshot below*) of the selected element's table structure.

All fields will be created as Text fields with the length specified in the Options tab. You can change the data type separately for every column in the Preview group box if you select the **Show Definition** radio button (see screenshot above).

To import data from a CSV or XML file into a database:

1. Make sure that a data source is active in the project folder.

2. Select the menu option **Tools | Import data to the database**, or press **Ctrl+I**, or click the **Import** button in the Tools toolbar. The "Import data to the database" dialog opens.

3. Click either the **XML** or **CSV** button at the top-left of the dialog box, as required.

4. In the **Path** text box, enter the path to the text file you want to import. You can type it in, select it from the drop-down list, or click the **Browse** button.

5. Under "Destination", fill in the destination data source, database, and schema, if applicable. If you previously selected a data source, the three fields are automatically filled in. Otherwise, do the following:

   a) Select the data source, database, and schema from the drop-down lists.

b) Select **Create new table(s)**, **Update/Create if not existing**, or **Update existing tables**, depending on where you want to import the data.

c) In the **Stop after errors** drop-down list, specify the number of errors after which the import process should be stopped.

d) In the **Stop action** drop-down list, select the action to perform when import is stopped: **Commit** commits the tables that did not contain errors; **rollback** does not commit any imported data.

6. In the Import Mode group box, do one or more of the following:

   - Select the **Execute** option if the necessary SQL should be generated and executed in one step.
   - Select the **Generate SQL in SQL Editor** option if the generated SQL file should be placed in the SQL Editor ready for execution.
   - Select the **Log** check box to enable the drop-down list where you can enter the name and path of the SQL log. You can select a previously used log file or an existing log file by clicking the **Browse** [...] button.

7. Click the **Reload** button to preview the data and further define the settings. You may need to click this button if you changed any of the previous options.

8. Optionally, click inside the "Target Table Name" column to change the target table name.

9. Click the **Show Data** or **Show Definition** radio button to edit the different aspects of the table structure.

10. Double-click a field in the preview window to edit the data type, nullability, or indexing. Define how/whether you want to import each column by clicking the column header to change it.

   Import

   Import as unique key

   Import as primary key

   Do not import

11. Do one of the following:

   - If the **Execute** option was selected previously, click **Import**. The text file is imported into the database and a message box appears asking if you want to view the log.sql file.
   - If the **Generate SQL in SQL Editor** option was selected previously, click **Generate SQL**. In this case, you have to execute the generated SQL script in order to import the text file into the database, see Executing SQL 364.

11.1   Defining XML Import Options

The Options tab of the Import data to the database dialog box allows you to define the options when importing data from files to the database. When the XML icon on the left side of the dialog box is selected, you may change the import options for files of type XML.

Start point of import

You can choose to import the entire XML document or restrict your import to the data hierarchy starting from the currently selected element. The number of sub-levels below the start point that will be imported is specified in the Import Depth option.

Import depth

Specifies the number of sub-levels below the start point that will be imported.

Import fields

Depending on your XML data, you may want to import only elements, attributes, or the textual content of your elements. Note that you can deselect the import of individual elements in the Preview window.

Automatic fields

DatabaseSpy will produce one output file or table for each element type selected. You can choose to automatically create primary/foreign key pairs to link your data in the relational model, or define a primary key for each element.
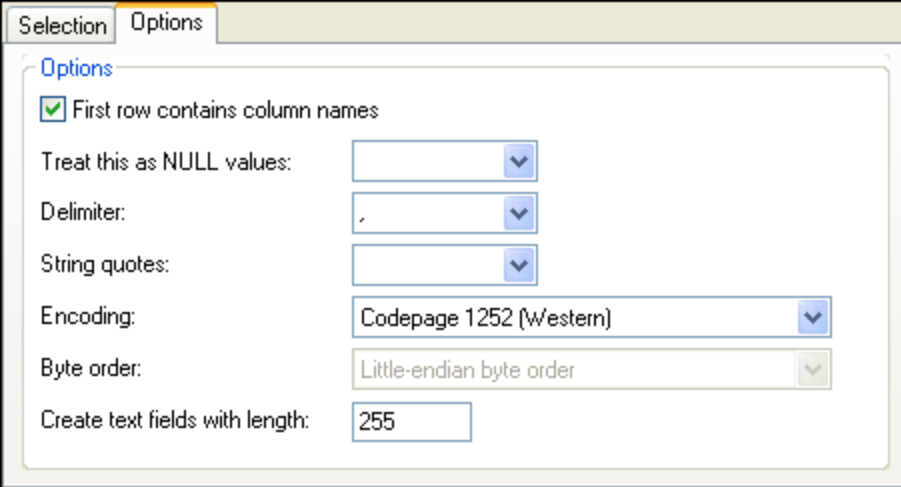

Other options

The Exclude namespace name radio button, together with the Replace Colon With Underscore radio button, is an either/or choice. Specifies whether namespace prefixes of elements and attributes should be excluded or whether the colon in the namespace prefix should be replaced with an underscore.

Text strings in the XML document that should be treated as NULL values as well as the length of text fields in the database can be specified.

11.2   Defining CSV Import Options

When the CSV icon on the left side of the dialog box is selected, you may change the import options for files of
type CSV on the Options tab of the **Import data to the database** dialog box.
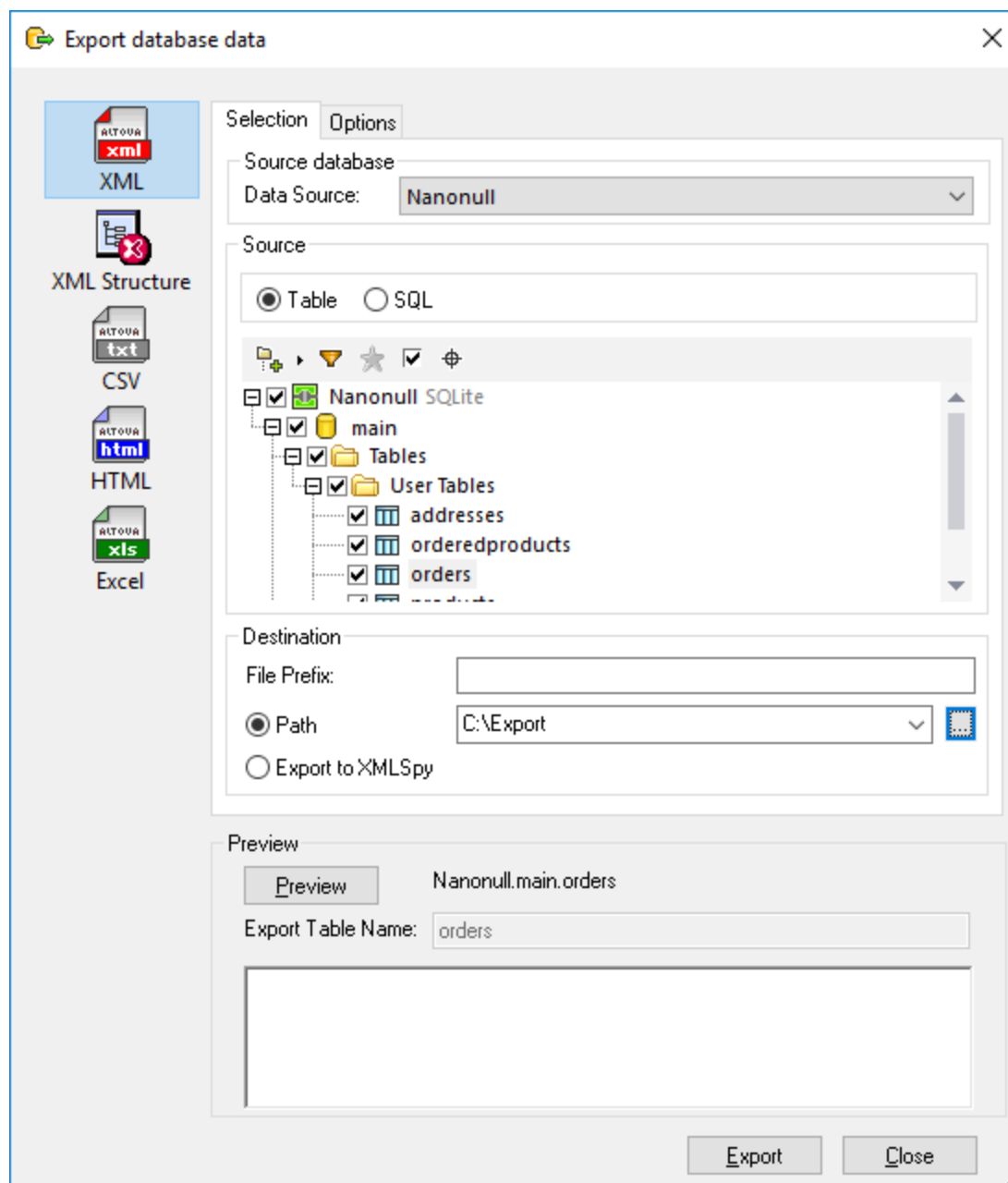
Options

If the files you are importing contain column names, select the **First row contains column names** check box
to exclude them from the import. In the **Treat this as NULL values** drop-down list, you can choose the
character that should be recognized as a NULL value in the file being imported. The delimiter character and the
quotes to be used to identify strings can also be selected from the respective drop-down lists. In the **Encoding**
drop-down list, select the encoding of the files you are importing from the drop-down list. If a two- or four-byte
encoding is selected as the default encoding (i.e., UTF-16, UCS-2, or UCS-4), you can choose between little-
endian and big-endian byte ordering. A fixed length of the text fields being imported can be defined in the
Create text fields with length field.

12 Exporting Database Data

DatabaseSpy can export database data to XML, CSV, HTML, and Excel 2007 or later (.xlsx) format. You can export data either directly from specific tables or views, or by generating or typing SQL SELECT statements. At table level, you can choose, for each column, whether it should be exported or omitted. If the export format is XML, you can choose whether to export each column as element or attribute, or not export it at all.

To export data, click the **Export** ![Export icon] toolbar button or select the menu option **Tools | Export database data...** (**Ctrl+E**). This opens the "Export database data" dialog box, where you can select the data to be exported and define the export options.

The "Export database data" dialog box contains two tabs, **Selection** and **Options**. The **Selection** tab is the same for all export types while the **Options** tab depends on each export file type. You can change the export file type by clicking the respective icon in the left area of the dialog box.

In case of XML, you can choose whether the exported XML should reflect the child-parent (foreign key) relationships that exist between database tables—to do this, click **XML Structure** on the export dialog box. Below is an example of an XML file exported with the **XML Structure** option. Notice that the "addresses" table is a child table of "users" in the database. Namely, the "user_id" column in the "addresses" table references the "id" column in the "users" table.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Import name="users">
   <users>
      <id>3</id>
      <first_name>Totie</first_name>
      <last_name>Rea</last_name>
      <email>t.rea@nanonull.com</email>
      <created_at>2019-04-15 10:11:13</created_at>
      <addresses>
         <id>3</id>
         <user_id>3</user_id>
         <is_shipping>1</is_shipping>
         <is_billing>1</is_billing>
         <type>home</type>
         <city>Roseford</city>
         <street>Evergreen Lane</street>
         <number>34</number>
      </addresses>
   </users>
</Import>
```

The **Source** tree view is very similar to the Online Browser window [24], with all its features and options. Here you can select the individual tables for export, as further described in Selecting Database Data for Export [409].

12.1   Selecting Database Data for Export

You can select database data for export in the same way for all export file types.

To export database data:

1. Make sure that at least one connection to a data source is active.
2. Select the menu option **Tools | Export database data**, or press **Ctrl+E**, or click the **Export** toolbar button. The "Export database data" dialog box opens.
3. Select the desired export format on the left side of the dialog box.
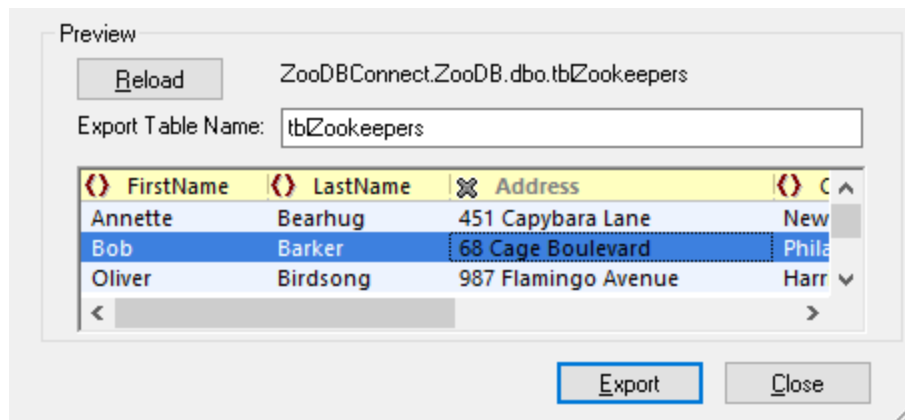4. Select the desired data source from the **Data Source** list.
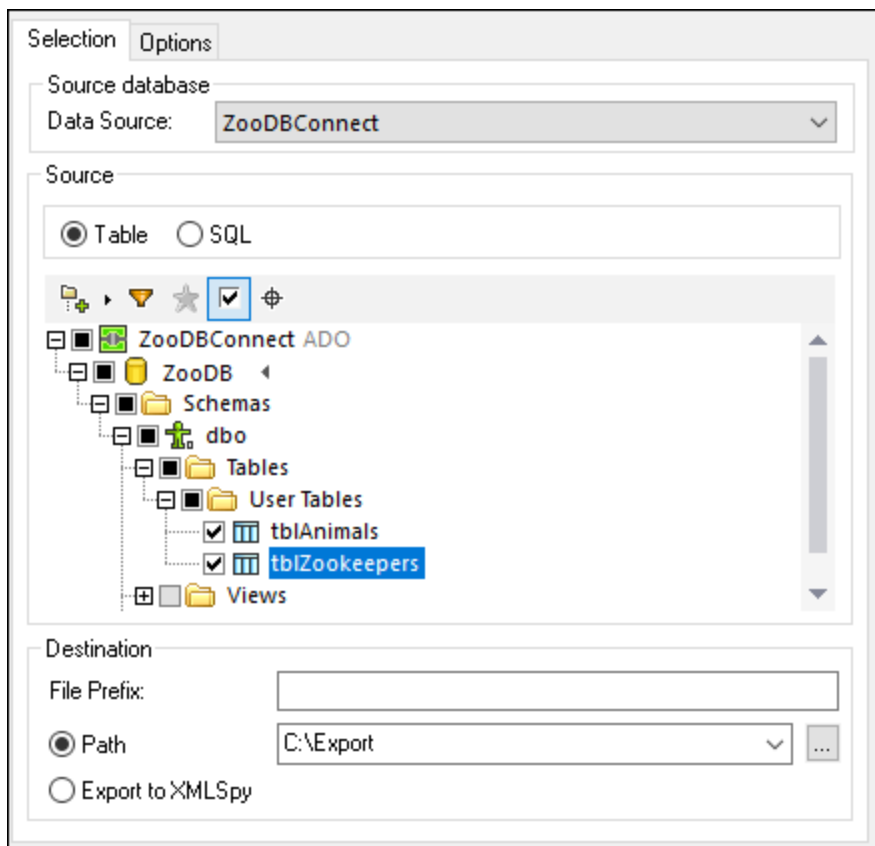
5. In the **Source** tree view, select the data you want to export, as follows:

   - To change the root database object, click the ◄ arrow icon and switch to the required schema or database. This is applicable only for those databases where multiple root objects (schema or databases) exist, such as SQL Server.
   - To select data from tables or views, leave the default **Table** option selected, and then click the check boxes next to the required tables or views. You can optionally refine the selection with the help of views, filters, favorites, and Object Locator, as well as the **Show checked objects only** option, like in the Online Browser [24]. Only the selected tables or views will be exported.

Note: The default layout shown in the tree view depends on the export file type. For **XML**, **CSV**, **HTML**, and **Excel**, the default layout is [Folders](#) 25. For **XML Structure**, the default layout is [Table Dependencies](#) 25. The latter helps you decide which of the dependent tables should be included in the export file. You can change the layout by clicking the **Layouts** button at the top of the tree view.

- To select data by entering SQL commands, select the **SQL** option, and enter the desired SELECT statement. You may enter multiple SELECT statements delimiting them with a separator applicable to the database kind, typically ";".

   When you initiate the data export from the [SQL Editor](#) 344, the **SQL** option is selected by default and the SELECT statement is pre-filled. If you select tables while the **Table** option is selected and then click the **SQL** option, the SELECT statements for those tables will be generated automatically.

6. Specify the **Destination** to which you want to export the data, as follows:

   - Optionally, enter a prefix that will be added to the table name when the table is exported.
   - To export data to a directory, select **Path** and enter the directory path in the text box or browse for it.
   - To open data in XMLSpy, select **Export to XMLSpy**.

7. Optionally, click the table you want to preview and then click **Preview**. Note that the **Preview** button is enabled only after you click a table. Also, the **Preview** button changes to **Reload** after the first preview.

Preview
Reload   ZooDBConnect.ZooDB.dbo.tblZookeepers

   Export Table Name: | tblZookeepers

() FirstName	() LastName	✖ Address	() C
Annette	Bearhug	451 Capybara Lane	New
Bob	Barker	68 Cage Boulevard	Phila
Oliver	Birdsong	987 Flamingo Avenue	Harr

   [Export] [Close]

8. Optionally, choose, for each column, whether it should be included or excluded from the export. To do this, click the icon next to the column name and switch to the desired state: **include** ☑ or **exclude** ✖ . In case of XML, you can choose whether each column should be exported as element, as attribute, or not be exported at all. The possible states for XML export format are, therefore, as follows: **export as element** (), **export as attribute** =, or **exclude** ✖. To restore the original settings, click the **Reload** button.

9. Optionally, change the file name of the selected table in the **Export Table Name** text box.

10. Click **Export**.

Exporting data from designs

If you have a specific design open in a [Design Editor](#) 46 window, you can easily export the data that is contained in the tables of this particular design. The Design Editor window provides an additional **Export** ▣

button which is active if at least one table in the Design Editor window is selected. When you click this button, the "Export database data" dialog box opens with the selected tables or views already checked. You can still add tables to the export definition.

Note: Clicking the **Export** toolbar button also opens the "Export database data" dialog box. However, in this case no tables are pre-selected for export.

12.2   XML and XML Structure Export Options

In the Options tab for XML and XML Structure you can define the appearance of the exported files.
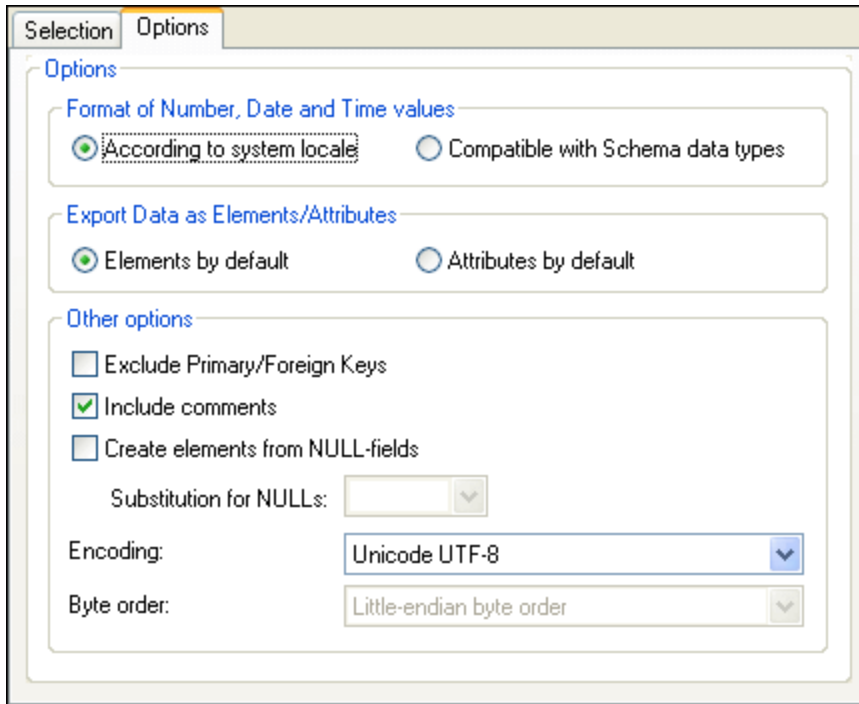
To define the extended options for XML and XML Structure export files:

1. Choose one of the following options for the format of number, date, and time values:

   - According to system locale
   - Compatible with Schema data types

2. Choose one of the following options for the default data type:

   - Select **Elements by default** to export the data as elements
   - Select **Attributes by default** to export the data as attributes

Note: You can change individual columns in tables to be exported as elements or attributes in the Preview group box of the dialog.

3. If you want to exclude primary and foreign keys from the export file, activate the **Exclude Primary/Foreign Keys** check box.
4. Activate the **Include comments** check box to include an attribute in the exported XML file that shows the SQL query used to select the data, as well as a list of comments containing one item for each column header in the database table.
5. Activate the **Create elements from NULL-fields** check box to include an empty element for each null value in the data being exported.

6. Optionally, select from the **Substitution for NULLs** drop-down list or type in the string that should appear in the XML file when there is a NULL value in the data being exported. You can also type in a value that is not contained in the drop-down list. This option is only available if the **Create elements from NULL-fields** check box is activated.

7. Select the desired encoding for the files that are generated during export from the **Encoding** drop-down list.

8. If a two- or four-byte encoding is selected as the default encoding (i.e., UTF-16, UCS-2, or UCS-4), you can choose between little-endian and big-endian byte ordering.

12.3   CSV Export Options

In the Options tab for CSV files, you can choose from among several options concerning the appearance of the
CSV file that is created as a result of the export.

To define the extended options for CSV export files:

1. Select the **Include comments** check box to include an attribute in the exported CSV file that shows
    the SQL query used to select the data, as well as a list of comments containing one item for each
    column header in the database table.
2. Select the **Remove delimiters** check box to remove delimiters that are contained in text values in the
    exported data. Set the delimiter you want to remove by using the Delimiter drop-down list, the second
    drop-down list in this tab. For example, if this option is activated and you export a table containing the
    string "Ba'ker", the string will be "Baker" in the exported text.
3. If the exported tables are to include the database column names, select the **Include column headers**
    check box.
4. Select the **Remove newlines** check box if you want to remove newlines from exported data.
5. Select from the **Substitution for NULL-values** drop-down list or type in the string that should appear
    in the CSV file when there is a NULL value in the data being exported. You can also type in a value
    that is not contained in the drop-down list.
6. Select from the **Delimiter** drop-down list the character you would like to use to delimit columns in the
    text file or enter the desired character string.
7. If you want to put each string into quotes, select the desired character from the **String quotes** drop-
    down list. You can also type in a character that is not contained in the drop-down list.
8. Select the desired encoding for the files that are generated during export from the **Encoding** drop-down
    list.
9. If a two- or four-byte encoding is selected as the default encoding (i.e., UTF-16, UCS-2, or UCS-4), you
    can choose between little-endian and big-endian byte ordering.

12.4   HTML Export Options

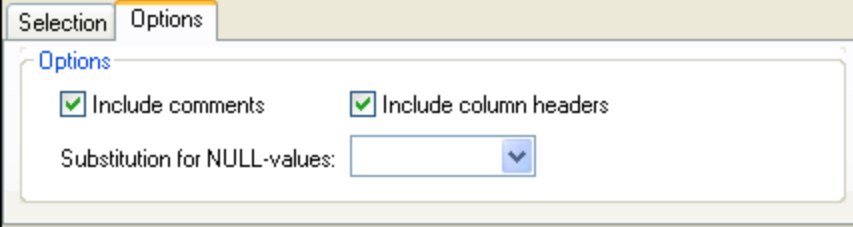The Options tab for HTML files provides some settings that apply to exported HTML files.
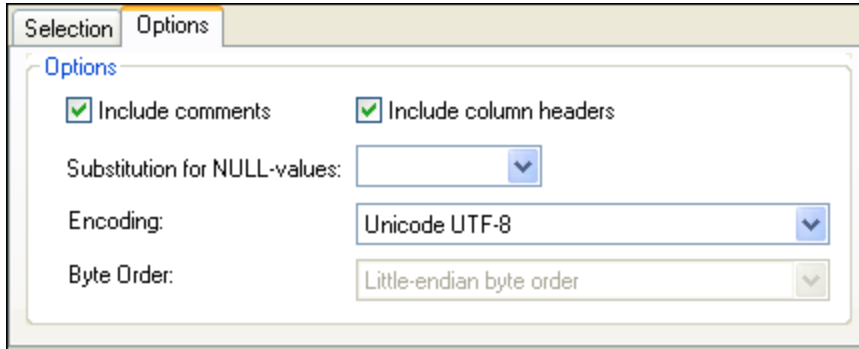
To define the extended options for HTML export files:

1. Activate the **Include comments** check box to include a comment in the HTML code of the exported HTML file that shows the SQL query used to select the data, as well as a list containing one item for each column header in the database table.
2. If the HTML table generated is to include the database column names, activate the **Include column headers** check box.
3. Select from the **Substitution for NULL-values** drop-down list or type in the string that should appear in the exported HTML file when there is a NULL value in the data being exported. You can also type in a value that is not contained in the drop-down list.
4. Select the desired encoding for the files that are generated during export from the **Encoding** drop-down list.
5. If a two- or four-byte encoding is selected as the default encoding (i.e., UTF-16, UCS-2, or UCS-4), you can choose between little-endian and big-endian byte ordering.

12.5 Excel Export Options

The export options applicable for Excel export files are as follows:

Include comments

If you select this check box, the exported Excel file will include metadata about each column from the source database table (name, data type, size).

Include column headers

Select this check box if the exported Excel table is to include the database column names.

Substitution for NULL-values

Select from the list or type the string that should appear in the exported Excel file when there is a NULL value in the data being exported. You can also type in a value that is not contained in the drop-down list.

13   DatabaseSpy Settings

The **Tools | Options** command enables you to define global application settings. These settings are specified in a dialog box with a navigation pane and saved in the registry. They apply to all current and future document windows. The **Apply** button in the Options dialog displays the changes in the currently open documents and fixes the current settings. The changes are seen immediately in the background windows. Each dialog page provides also a **Reset to Page Defaults** button which sets all changes you made on this page back to the default settings.

Each page of the **Options** dialog is described in detail in this section.

To change DatabaseSpy settings:

1. Select **Tools | Options...** or press **Ctrl+Alt+O**. The Options dialog box appears.
2. In the navigation pane, select the dialog pages for the following options:

    -   General [418]
    -   SQL Editor [425]
    -   Design Editor [433]

3. Make the desired changes and click **OK**.

13.1   General Options

The main page of the General options defines the general appearance of DatabaseSpy as well as general options when working with the application. Related settings are defined in the Online Browser[419], Change Script[420], File Types[422], and Encoding[423] options.

Application Startup

You can choose to automatically open the project file that was open when you last closed DatabaseSpy or to have the Add a Data Source dialog box displayed when DatabaseSpy is started. If you do not choose any of the options in this group box, you must open files or projects in DatabaseSpy via the **File** menu.

Program Logo

Choose whether you want the DatabaseSpy logo displayed when the application is started and whether it should appear on top of a print-out you make.

Line Breaks

When you open a file, the character coding for line breaks in it is preserved if **Preserve old** is selected. Alternatively, you can choose to code line breaks in any of three codings: CR&LF (for PC), CR (for MacOS), or LF (for Unix).

Frame Title

The frame title in DatabaseSpy shows the project name and the name of the object that is currently active in the SQL Editor or the Design Editor, respectively. You can choose whether you want to see just the file name or the entire path of the object in the frame title of the application.

Automatic reload of files

If you are working in a multi-user environment, or if you are working on files that are dynamically generated on a server, you can watch for changes to files that are currently open in the interface. Each time DatabaseSpy detects a change in an open document, it will prompt you about whether you want to reload the changed file.

Connection Wizard

If you activate the **Skip Configuration Step in Connection Wizard** check box, the Configure page in the Add a Data Source dialog box is skipped and clicking **Next** in the first dialog page of the Add a Data Source dialog box will bring you directly to the Data Link Properties dialog box.

Newly saved files

You can automatically add files to your project when you save them to the file system or suppress the display of the respective dialog box.

13.1.1 Online Browser

The Online Browser section of the Options dialog box lets you configure layout and behaviour aspects of the Online Browser [24].
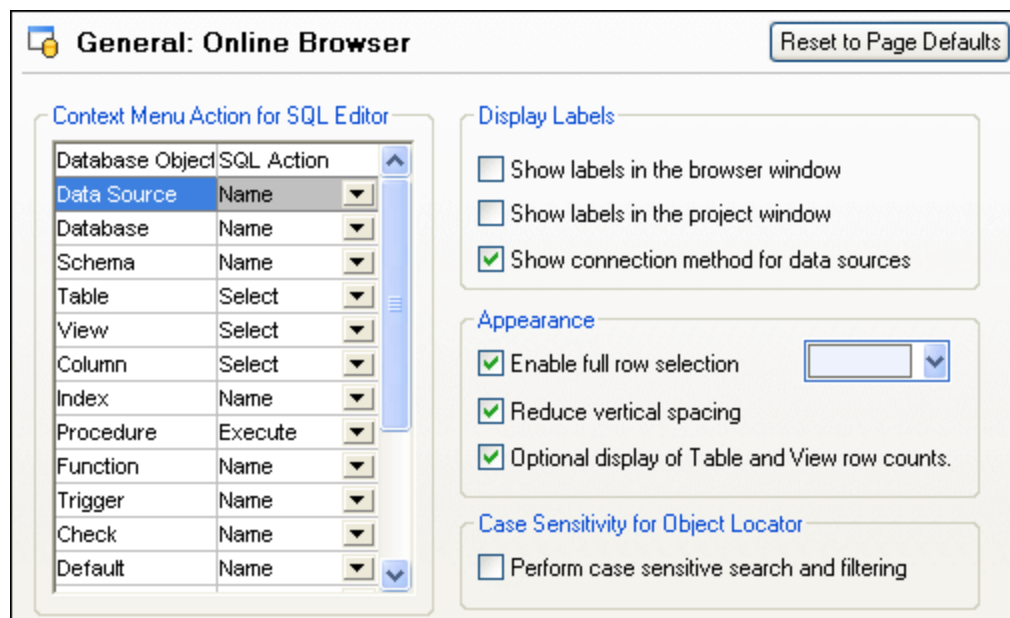
Context Menu Action for SQL Editor

Specify the default action taken when an object is dragged and dropped from the Online Browser to the SQL Editor. You can specify a different default action for each database object type.

Display Labels

When selected, the **Show labels in the browser window** check box adds a prefix to all database objects in the Online Browser. For example, tables are prefixed with a "Table: " prefix, columns are prefixed with a "Column: " prefix, and so on.

The **Show labels in the project window** check box adds a "Data Source: " prefix to all data sources in the Project window.

When enabled, the option **Show connection method for data sources** displays the connection kind (for example, "ADO", "ADO.NET", "ODBC", "JDBC") next to each data source, both in the Project Window and in the Online Browser.

Appearance

When the **Enable full row selection** check box is activated, in the Online Browser the row in which the cursor is currently positioned is highlighted and you can click to select the row without having to move the cursor exactly over the text of the row. From the drop-down list, you can choose a color to be used for selecting.

If you prefer a wider display when browsing databases, clear the **Reduce vertical spacing** check box. This option applies to both Project window and Online Browser.
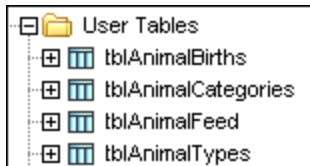
Reduced vertical spacing   *Increased vertical spacing*

The **Optional display of Table and View row counts** check box is selected by default, allowing you to show the number of data rows of a table or view without having to retrieve data first.

Case Sensitivity for Object Locator

When searching and filtering database objects, you can decide whether or not this should be case sensitive.

13.1.2   Change Script

The Change Script section of the Options dialog box lets you configure the script generation and execution of the Database Structure Change Script.
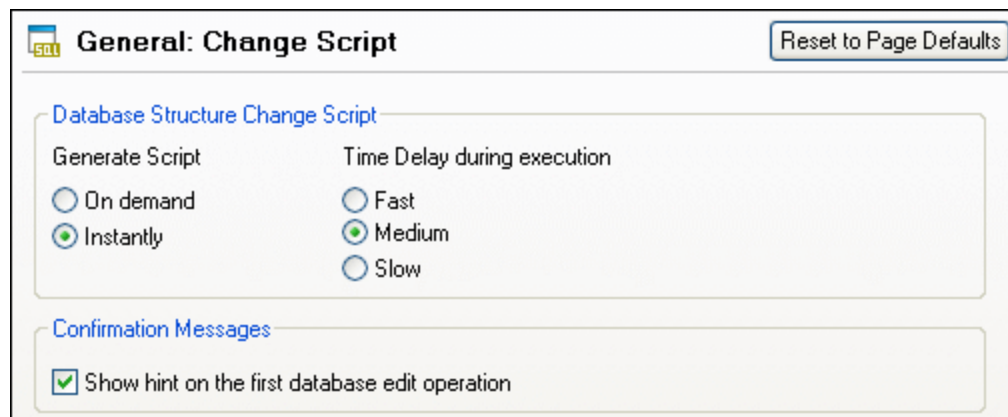
Database Structure Change Script

The Database Structure Change Script can be generated either instantly when a change occurs, or on demand. In the latter case, you must generate the change script manually by clicking on the **Update Change Script**

icon. In this group box, you can also define the time delay during execution of the change script. Choosing **Slow**, for example, increases the time delay and gives you the opportunity to cancel the execution of the change script at a certain stage.

Confirmation Messages

DatabaseSpy can display a popup alarming you that a change script has been generated when a database object is edited for the first time. You can turn off this message by deactivating the **Show hint on the first database edit operation** or by activating the **Don't show this dialog again!** check box in the dialog box.
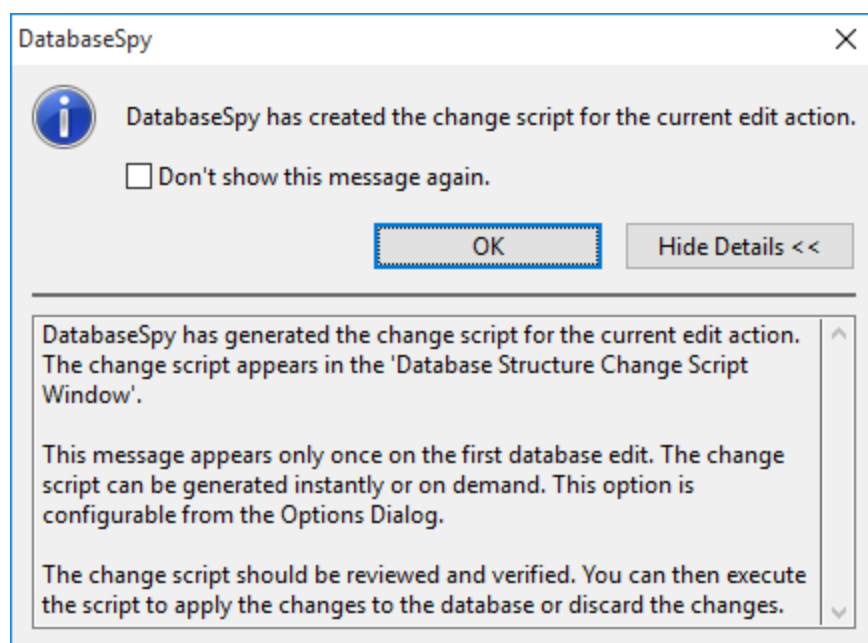
If you want DatabaseSpy to inform you when a change script is created, but do not want to have the details displayed, click the **Hide Details** button and the dialog will appear in a minimized form next time.

13.1.3   File Types

The File Types section of the Options dialog box allows you to customize the behavior of DatabaseSpy on a per-file-type basis.
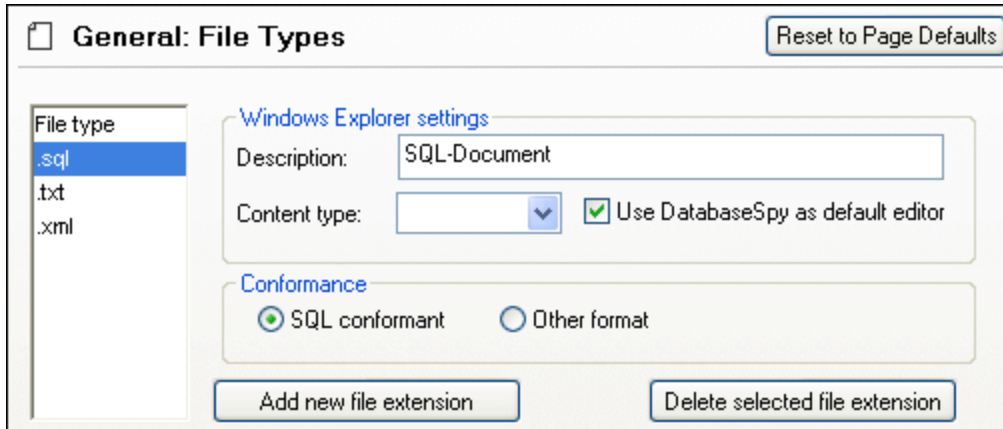
Windows Explorer settings

You can define the file type description and MIME-compliant content type used by Windows Explorer and whether DatabaseSpy is to be the default editor for documents of this file type.

Conformance

DatabaseSpy provides specific editing and other features for various file types. The features for a file type are set by specifying the conformance in this option. DatabaseSpy lets you set file type to conform with SQL and other (text) grammars. A large number of file types is defined with a default conformance that is appropriate for the file type. We recommend that you do not modify these settings unless you are adding a new file type or deliberately wish to set a file type to another kind of conformance.

Add new file extension

Adds a new file type to the File types list. You must then define the settings for this new file type using the other options in this section.

Delete selected file extension

Deletes the currently selected file type and all its associated settings.

13.1.4   Encoding

In the Encoding section of the Options dialog box, you can specify several options for file encodings.
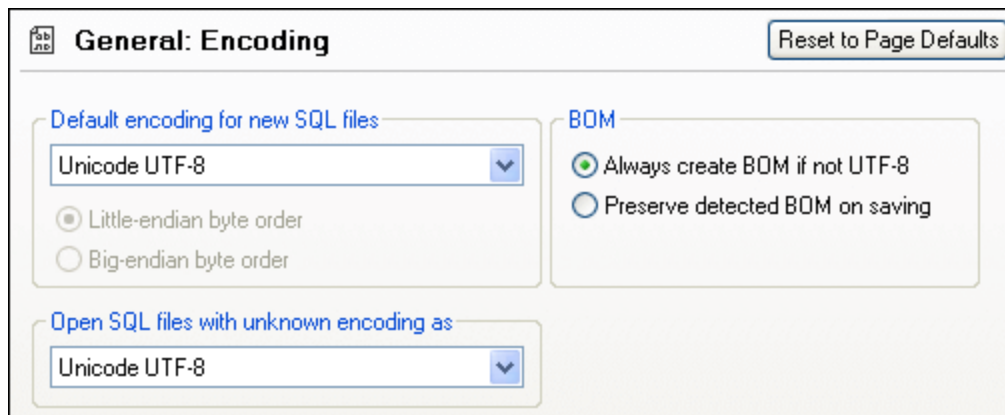
Default encoding for new SQL files

Define the default encoding for new files so that each new document includes the encoding-specification that you specify here. If a two- or four-byte encoding is selected as the default encoding (i.e., UTF-16, UCS-2, or UCS-4), you can also choose between little-endian and big-endian byte ordering for the SQL files. The encoding for existing files will, of course, always be retained.

Open SQL files with unknown encoding as

You can select the encoding with which to open an SQL file with no encoding specification or where the encoding cannot be detected.

Note: SQL files which have no encoding specification are correctly saved with a UTF-8 encoding.

BOM (Byte Order Mark)

When a document with two-byte or four-byte character encoding is saved, the document can be saved either with (i) little-endian byte-ordering and a little-endian BOM (**Always create BOM if not UTF-8**); or (ii) the detected byte-ordering and the detected BOM (**Preserve detected BOM on saving**).

13.1.5   Java

In the *Java* section (*see screenshot below*), you can optionally enter the path to a Java VM (Virtual Machine) on your file system. Note that adding a custom Java VM path is not always necessary. By default, DatabaseSpy attempts to detect the Java VM path automatically by reading (in this order) the Windows registry and the JAVA_HOME environment variable. The custom path added in this dialog box will take priority over any other Java VM path detected automatically.

You may need to add a custom Java VM path, for example, if you are using a Java virtual machine which does not have an installer and does not create registry entries (e.g., Oracle's OpenJDK). You might also want to set this path if you need to override, for whatever reason, any Java VM path detected automatically by

DatabaseSpy.

Note the following:

- The Java VM path is shared between Altova desktop (not server) applications. Consequently, if you change it in one application, it will automatically apply to all other Altova applications.
- The path must point to the `jvm.dll` file from the `\bin\server` or `\bin\client` directory, relative to the directory where the JDK was installed.
- The DatabaseSpy platform (32-bit, 64-bit) must be the same as that of the JDK.
- After changing the Java VM path, you may need to restart DatabaseSpy for the new settings to take effect.

Changing the Java VM path affects database connectivity via JDBC.

13.2   SQL Editor Options

The main page of the SQL Editor options defines the visual appearance of the editor. Additional SQL Editor-related settings are defined in the SQL Generation [426], SQL Formatting [427], Autoinsertion [428], Result View [429], and Fonts [430], options.

General

Syntax coloring emphasizes different elements of SQL syntax using different colors. By activating the **Connect data source on execute** check box, you can have DatabaseSpy connect to the corresponding data source automatically whenever an SQL file is executed and its data source is not connected.

Retrieval

You can have DatabaseSpy display a dialog box asking you whether you want to change the time out settings when the permissible execution period is exceeded. Specify the maximum amount of time that is permissible for the execution of retrievals. The number n you enter into the **Buffered Amount (Rows)** field will appear in the context menu of the Online Browser if you select the **Retrieve data | First n rows** option.

Entry Helpers

To enable auto-completion suggestions as you start typing in SQL Editor, select the **Automatically open** check box.

The entry helper buffer for auto-completion and auto-insertion can be filled either when you connect to a data source or when it is used for the first time. Note that filling the buffer may take some time. Use the **Clear Buffer** button to reset the buffer.

If you are running DatabaseSpy as part of Altova MissionKit or alongside other Altova products, note that this setting is preserved across multiple Altova products that have an SQL editor.

Text View Settings

Text view settings are defined in a separate dialog box, which can be accessed via the **Text View Settings** button. See SQL Editor Features [359].

13.2.1   Generation

The Generation section of the Options dialog box lets you configure how the statements are generated.
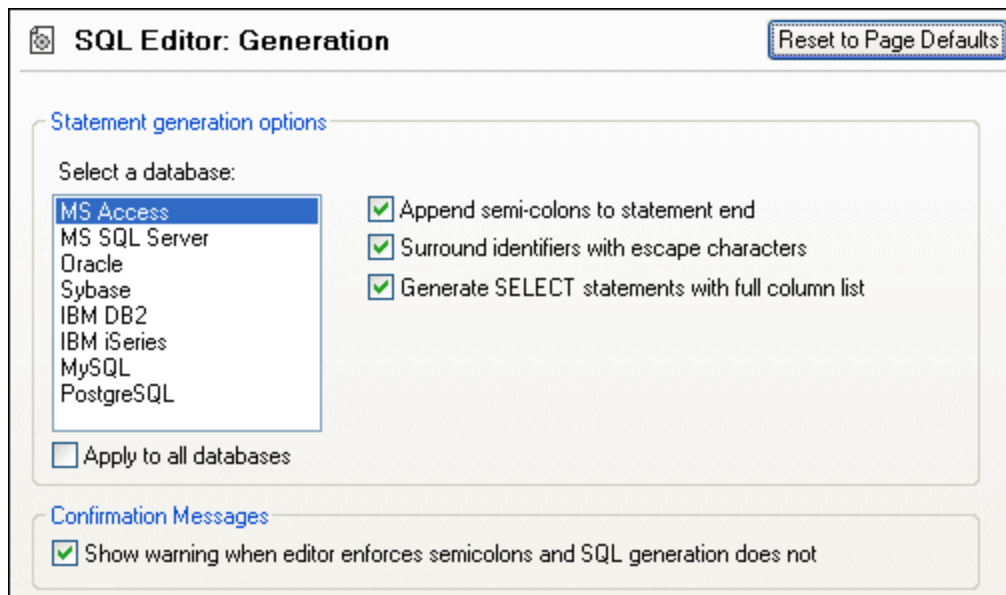
Statement generation options

Use the **Select a database** field list to select a database kind and set the statement generation options individually for the different database kinds you are working with. Activating the **Apply to all databases** check box sets the options that are currently selected for all databases.

When the **Append semicolons to statement end** check box is activated, a semicolon is appended when you generate an SQL statement in the SQL Editor. Note that editing of data in Oracle databases and IBM iSeries and DB2 databases via a JDBC connection is possible only if this check box is unchecked.

By default, identifiers in SQL statements are enclosed in database-specific escape characters. You can deactivate the **Surround identifiers with escape characters** check box if you do not want to display these characters.

The **Generate SELECT statements with full column list** check box is also activated by default. If you prefer `SELECT * FROM ...` when retrieving all columns of a table, deactivate it.

Confirmation Messages

When you have deactivated the **Append semicolons to statement end** check box and activated the **Show warning when editor enforces semicolons and SQL generation does not** check box, DatabaseSpy will display a message box when a statement is generated (see Generating SQL Statements [346]).
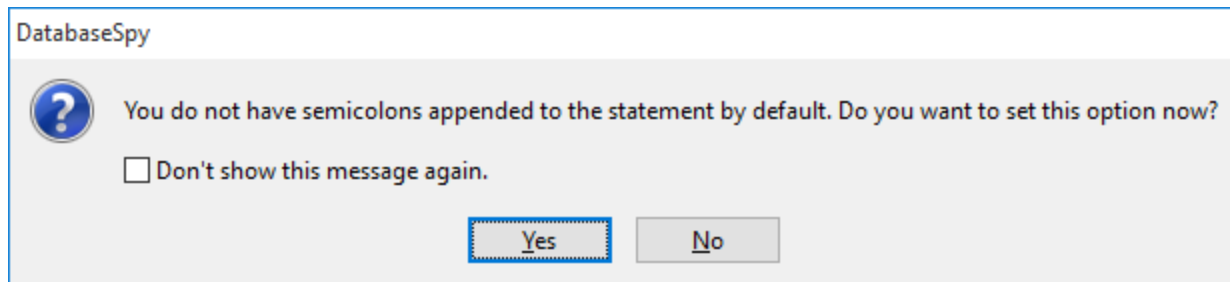
13.2.2   Formatting

The Formatting section of the Options dialog box lets you switch off SQL formatting for the SQL Editor and Change Script windows and configure the behavior of open SQL Editor windows.
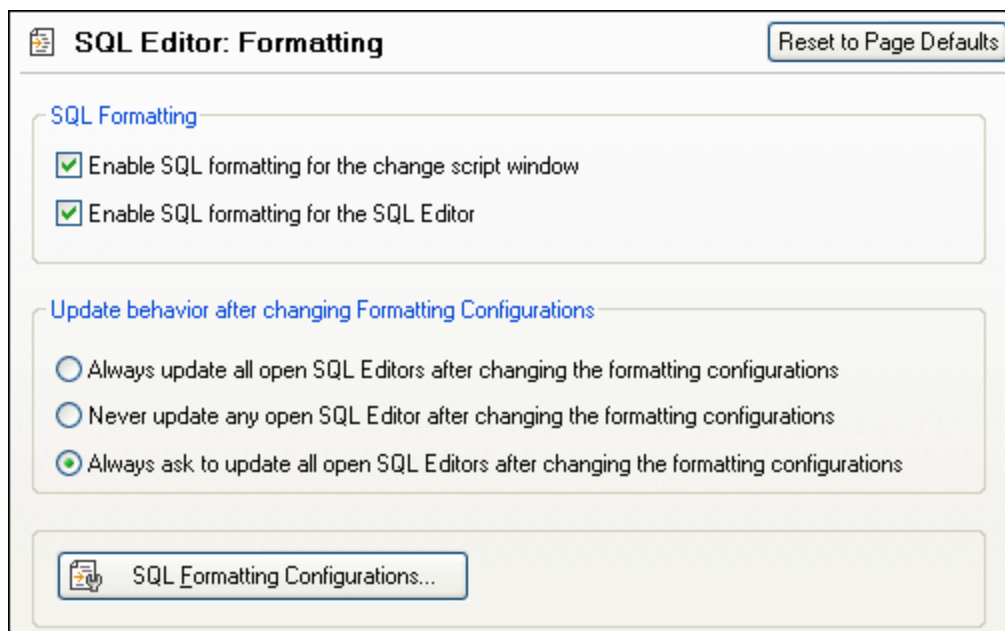
SQL Formatting

You can disable the SQL formatting options for the Change Script window and/or the SQL Editor by deactivating the respective check boxes. Note that in the Change Script window, SQL formatting is applied automatically when the next change is recorded, and that you cannot apply formatting without changing the database structure.

Update behavior after changing Formatting Configurations

This group box allows you to specify whether or not open SQL Editor windows should be automatically updated after formatting configurations have changed or if a prompt should be displayed, where you can choose to update the windows or to keep the existing formatting.

SQL Formatting Configurations

The **SQL Formatting Configurations...** button opens the **SQL Formatting Configuration** dialog box, where you can define the configurations [354] for the different database types or disable formatting on database or even statement type level.

13.2.3   Autoinsertion

The Autoinsertion section of the Options dialog box lets you configure the options that apply for text that is inserted automatically into the SQL Editor. Autoinsertion occurs, for example, when you type an opening bracket "(" and DatabaseSpy automatically inserts the closing bracket ")".

Closing Characters

You can disable the autoinsertion of closing characters completely by deactivating the **Autoinsert closing characters** check box. If you only want to exclude some of the characters from being inserted automatically, activate the **Autoinsert closing characters check box** and deactivate the corresponding check box for the respective character. If you insert an opening character for which autoinsertion is enabled, the corresponding closing character is automatically inserted and the cursor is placed between the two characters.

When inserting comment marks, please note that you have to highlight a text section before entering a "/" or "-" in order to comment out this text. DatabaseSpy will then automatically insert a "*" after the "/" in front of the text as well as "*/" after the text if you enter a "/", or add a second "-" if you enter a "-" while the text to be commented out is selected.

Autoindent

The **Match previous line** option helps you preserve text indentation after you press the **Enter** key in SQL Editor. For example, if line 1 is indented by a tab character, you may want the next line to have the same indentation as line 1 after you press **Enter**. To enable this behaviour, select the **Match previous line** check box.

Behaviour

When the corresponding option is activated, the closing character is deleted automatically when the opening character is deleted in the SQL Editor.

To place selected text inside a pair of autoinserted characters, activate the **Wrap selected text when autoinserting** check box and then type the opening character. The opening and closing characters are inserted automatically before and after the text.

Star Expansion

DatabaseSpy can automatically replace the "*" in a SELECT * FROM . . . statement with all of the columns of the respective table if the **Expand columns with tab** option is checked. Please note that the statement has to be valid and that you must place the cursor behind the "*" in the statement and then press the **Tab** key to achieve this.

13.2.4   Result View

The Result View section of the Options dialog box lets you configure aspects of the appearance of the Result window in the SQL Editor.

Colors

You can have rows in Result tabs displayed as simple grid or with alternating white and colored rows. The color can be changed using the color chooser beside the check box.

Display Options

By default, horizontal and vertical grid lines as well as line numbers and the Result toolbar are displayed. You can switch any of these options off by deactivating the respective check box.

DatabaseSpy allows you to sort data contained in the result grid either ascending or descending by repeatedly clicking the column headers. To enable this option, activate the **Show sorting arrows** check box.

Data Editing

When the default setting is applied, DatabaseSpy uses transactions to save changes in the result grid and rolls back these changes if a failure occurs during committing the changes to the database. Any editing is then written into a transaction which is executed as a whole. If the **Rollback transactions on failure** check box is selected, all changes contained are undone upon failure of a single update statement in the transaction.

Note: Transactions are not supported by MS Access databases.

If default values are defined for the individual database fields, DatabaseSpy can automatically insert these values when creating a new line in the result grid. Clear the **Initialize cells with default values** check box if you want to create empty lines.

When data editing is not possible to the full extent, DatabaseSpy shows a warning message before the data is displayed in the Result window. You can suppress this hint by clearing the **Show hint when data editing is limited** check box.

13.2.5   Fonts

The Fonts section of the Options dialog box lets you configure color and font settings of different parts of SQL statements. Use the drop-down list below the heading of the dialog page to switch between Text Editor and Result View settings.

Text Editor Font

The types listed in the left hand pane are elements of SQL statements. You can choose the common font face, style, and size of all text that appears in SQL Editor. Note that the same font and size is used for all text types. Only the style can be changed for individual text types. This enables the syntax coloring feature.

Click the **Reset to Page Defaults** button to restore the original settings.

Result View Font
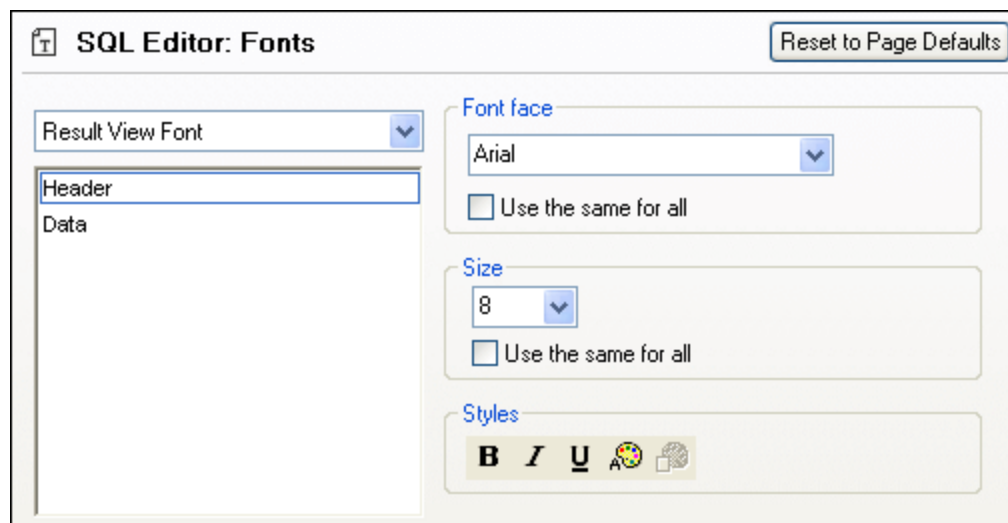
In the left hand pane, you can choose either the result grid header or the data rows. You can choose the common font face, style, and size of all text that appears in the result grid. Note that you can apply the same font and/or size to all text types by activating the respective **Use the same for all** check box.

Click the **Reset to Page Defaults** button to restore the original settings.

13.3   Design Editor Options

The main page of the Design Editor options defines the visual appearance and general behaviour of the editor. Additional Design Editor-related settings are defined in the Element Colors [434] and Fonts [435] options.

General Behavior

If you have several data sources defined in your project but not all are active, you can automatically connect to the related data source when you open a design file in Design Editor. A prompt asking you if you want to connect will appear if this option is not activated.

DatabaseSpy can display tooltips when database objects are dragged from the Online Browser or from different table designs into the Columns or Keys section of a table design.
When creating new database objects, you can automatically display the properties box in the Design Editor so as to fine-tune the definition of the objects without having to change to the properties window. For this purpose, select the **Automatically expand newly created objects** check box.

Tables

When new tables are created in the Design Editor, DatabaseSpy can automatically create primary keys for them. The first column of a new table is then renamed to "id" and a primary key constraint is added to the Keys section.

You can have the schema name displayed in parentheses in the title bar of a table.

Relations

By default, all connection lines between related tables are labeled in Design Editor. You can, however, hide the labels of unselected lines by selecting the respective check box. Highlighting active relations objects will gray out tables that do not form part of a relation when you click on the line that connects related tables.

Default options on insertion

The default table layout can be set to compact and/or expanded, see Viewing Tables as a Diagram [227]. In the compact layout only the column names of a table are visible (select the **Insert with compact layout (single column)** check box to switch to compact layout). If the Expand table object when inserted check box is not checked, only the title bar of a table will be displayed.

Grid Settings

Adjust the grid size according to your needs and specify whether or not database objects in the Design Editor should snap to grid.

13.3.1   Element Colors

The Colors section of the Options dialog box lets you configure colors for objects in the Design Editor.
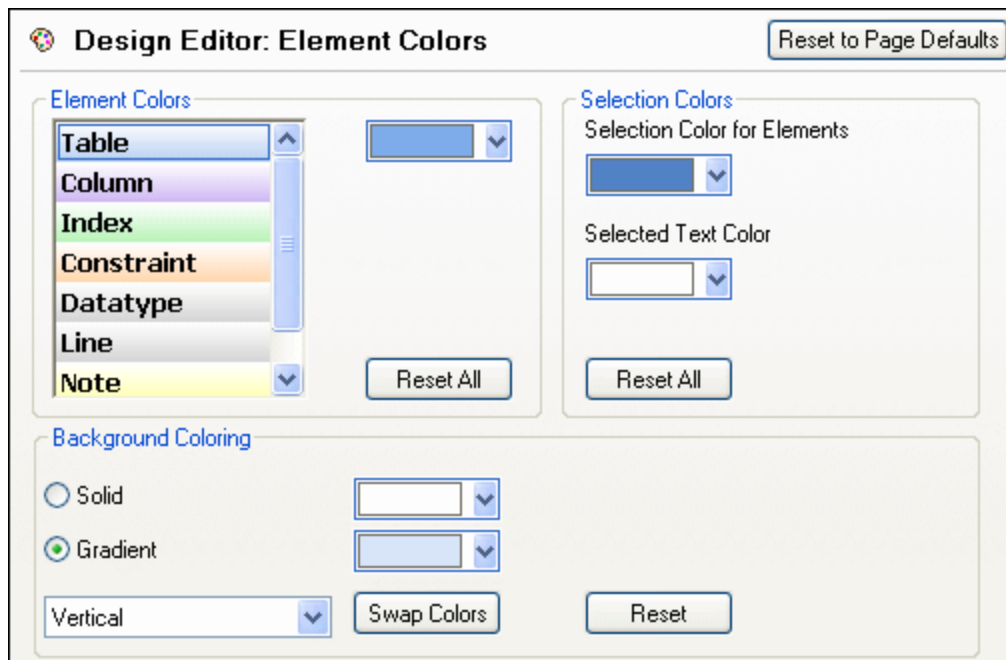
Element Colors

Specify the color that is used for displaying each element in the Properties windows. Select an element from the list and choose a color for it from the drop-down list. Click the **Reset All** button to restore the original settings.

Selection Colors

Define the colors for selected elements and text. Click the **Reset All** button to restore the original settings.

Background Coloring

Defines the appearance of Design Editor's background. Choose between solid and gradient background and specify the colors and the way they are used in the gradient.

13.3.2   Fonts

The Fonts section of the Options dialog box lets you configure fonts for objects in the Design Editor.
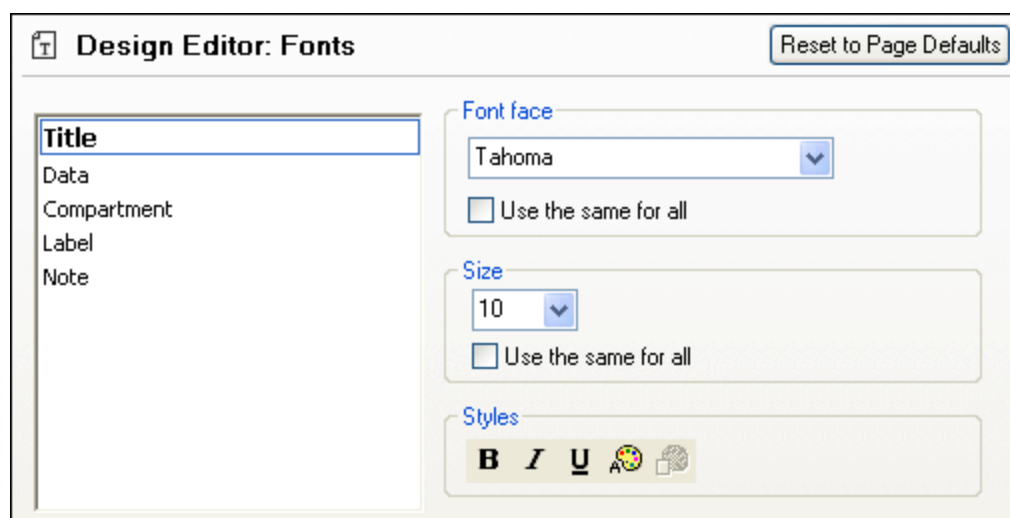
Font settings

You can choose the font face, size, and style for the individual elements displayed in the Design Editor. Note that, unlike in the Text Font section, it is possible to specify different font faces for header and data rows. To apply the same font and/or size to all text types, activate the respective **Use the same for all** check box.

The element names in the list box are defined as follows:

- **Title**: The name of a table in its title bar
- **Data**: The content of the table, i.e., the column names
- **Compartment**: The sub-titles in a table, i.e., Column Name, Indexes, Keys, Check Constraints
- **Label**: The designation of the connection line which links related tables in the Design Editor
- **Note**: The content of sticky notes that you add to the design view

14   Altova Global Resources

Altova Global Resources are aliases for file, folder, and database resources. Each alias can have multiple configurations, and each configuration maps to a single resource. Therefore, when you use a global resource, you can switch between its configurations. For example, you could create a database resource with two configurations: development and production. Depending on your goals, you can switch between these configurations. In DatabaseSpy, you can switch between the development and production databases by choosing the desired configuration from the drop-down list.

Global resources can be used across different Altova applications (*see subsection below*).

Global resources in other Altova products

When stored as global resources, files, folders, and database connection details become reusable across multiple Altova applications. For example, if you often need to open the same file in multiple Altova desktop applications, you can define this file as a global resource. If you need to change the file path, you will need to change it only in one place. Currently, global resources can be defined and used in the following Altova products:

- Altova Authentic
- DatabaseSpy
- MobileTogether Designer
- MapForce
- StyleVision
- XMLSpy
- FlowForce Server
- MapForce Server
- RaptorXML Server/RaptorXML+XBRL Server

14.1   Creating Global Resources

Altova Global Resources are defined in the **Manage Global Resources** dialog, which can be accessed in two ways:

- Click the menu command **Tools | Global Resources**.
- Click the **Manage Global Resources** icon in the Global Resources toolbar (*screenshot below*).

Global Resources Definitions file

Information about global resources is stored in an XML file called the Global Resources Definitions file. This file is created when the first global resource is defined in the **Manage Global Resources** dialog box (*screenshot below*) and saved.
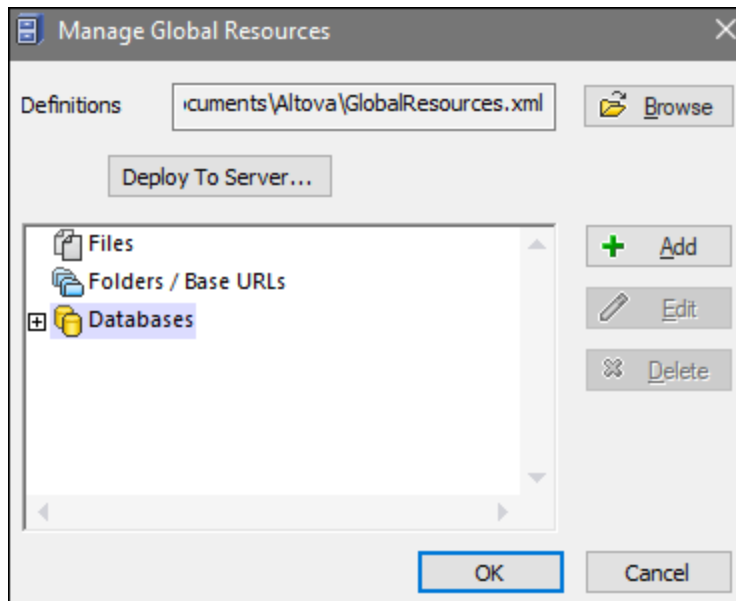
When you open the **Manage Global Resources** dialog box for the first time, the default location and name of the Global Resources Definitions file is specified in the *Definitions* text box (*see screenshot above*):

        C:\Users\<username>\Documents\Altova\GlobalResources.xml

This file is set as the default Global Resources Definitions file for all Altova applications. A global resource can be saved from any Altova application to this file and will immediately be available to all other Altova applications as a global resource. To define and save a global resource to the Global Resources Definitions file, add the global resource in the **Manage Global Resources** dialog and click **OK** to save.

To select an already existing Global Resources Definitions file to be the active definitions file of a particular Altova application, browse for it via the **Browse** button of the *Definitions* text box (*see screenshot above*).

The **Manage Global Resources** dialog box also allows you to edit and delete existing global resources.

> **Notes:**
>
> - You can give any name to the Global Resources Definitions file and save it to any location accessible to your Altova applications. All you need to do in each application, is specify this file as the Global Resources Definitions file for that application (in the *Definitions* text box). The resources become global across Altova products when you use a single definitions file across all Altova products.
>
> - You can also create multiple Global Resources Definitions files. However, only one of these can be active at any time in a given Altova application, and only the definitions contained in this file will be available to the application. The availability of resources can therefore be restricted or made to overlap across products as required.

Converting existing database connections to Global Resources

You can easily convert any data source connection that appears in a DatabaseSpy project to a Global Resource. To do this, right-click the connection in the Project window, and select **Convert to Global Resource** from the context menu.
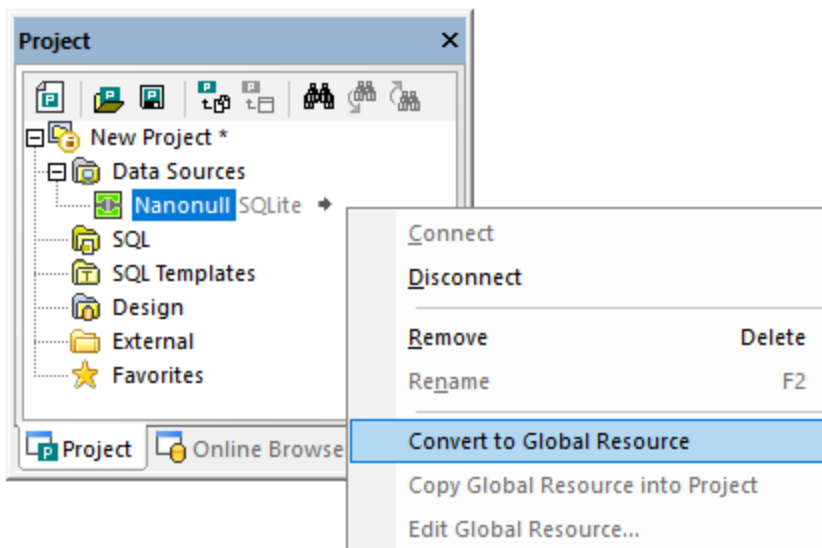
The data source is added to the <u>The Global Resources XML File</u>[440] and hence available in all Altova applications. Note that the icon in the Project window changes to indicate that the data source connection is now a global resource.
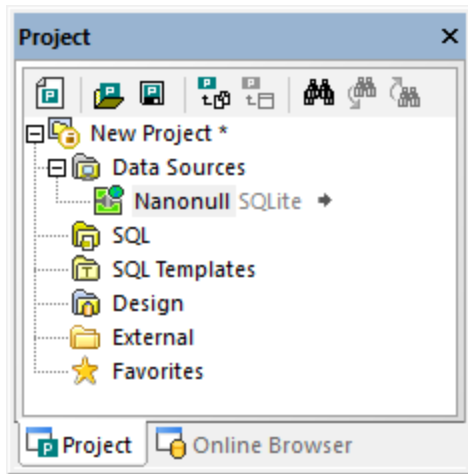
When you right-click the Data Sources folder in the Project window, you also have the option to convert all data sources in the project to global resources.

14.2   The Global Resources XML File

By default, all Global Resources, regardless of the Altova application where they were created, are stored at the following path: **C:\Users\<username>\Documents\Altova\GlobalResources.xml**. This makes them transparent, easy to backup, as well as portable to other workstations where Altova products are installed. It is also possible to rename or duplicate the **GlobalResources.xml** file and thus create multiple Global Resource files. However, only one Global Resource file can be active at a time in an Altova application.

> Unlike other Altova applications, FlowForce Server does not work with global resource files. Instead, resources are managed like other FlowForce configuration data (they are reusable objects with access permissions).

To set up the active Global Resource file:

1. On the **Tools** menu, click **Global Resources**. (Alternatively, click the **Global Resource** ⊞ toolbar button.)
2. Click **Browse** and select the required Global Resource XML file.

If you are using multiple Global Resource files, make sure that the currently active file contains all Global Resources required by the current context.

14.3   Using Database Global Resources in Projects

From the database connection wizard, you can reuse any connection that was previously defined as a Global Resource and add it to your project. To do this, click the **Global Resources** tab, select the respective resource, and then click **Connect**.
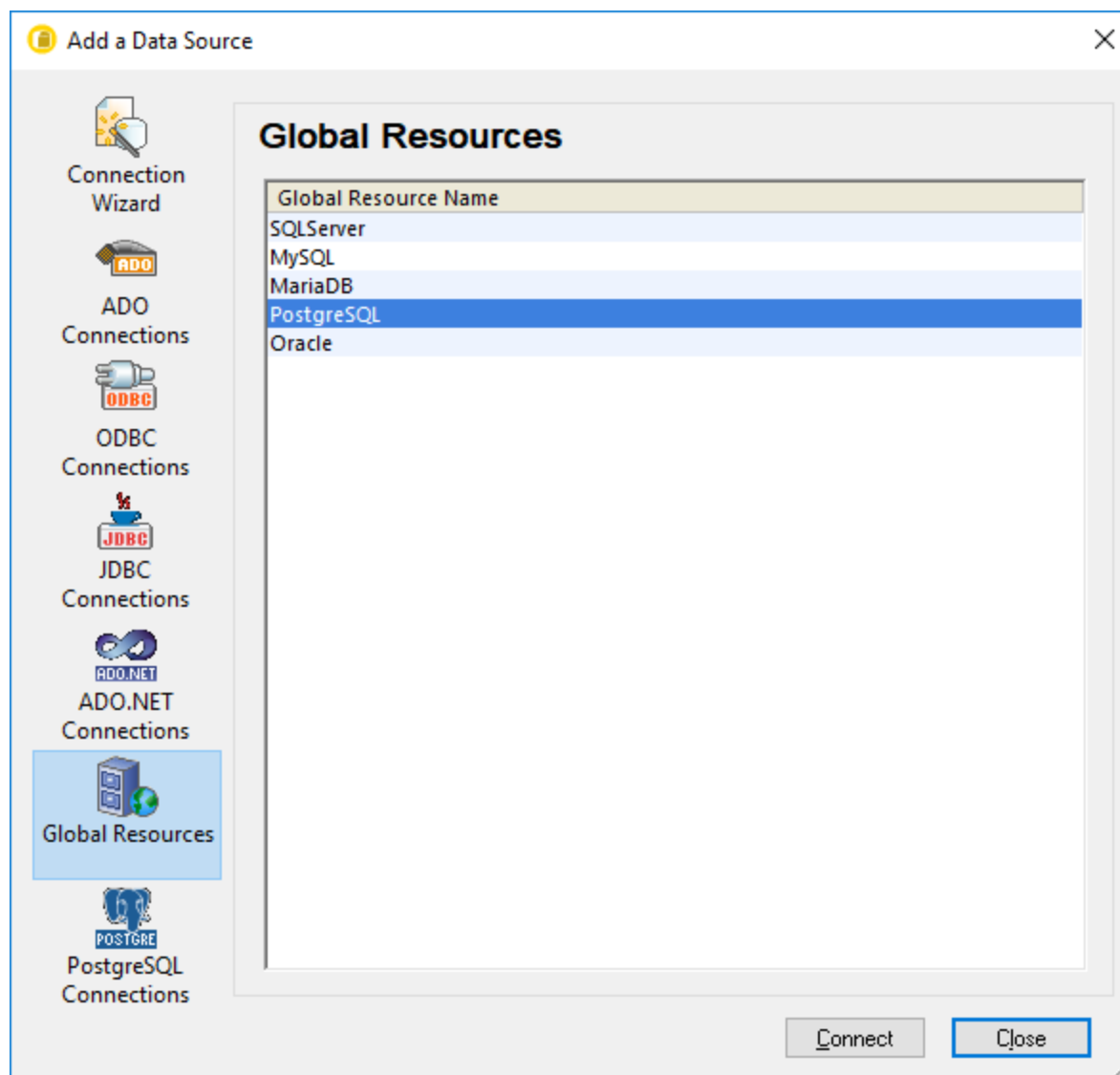
The Global Resource is added to the currently opened project. If the Global Resource has multiple configurations, you can always change the active configuration, and this will be reflected accordingly in the project. For example, if a database resource has configurations for Default, ADO.NET, and JDBC, you can switch to the desired configuration using either the **Tools | Active Configurations** menu command above or the Global Resource toolbar:
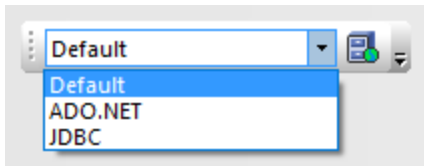
You can also make the Global Resource local to the project and turn it to a standard connection having any of the desired configurations, as follows:

1. Select the desired configuration using either the **Tools | Active Configurations** menu command or the Global Resource toolbar.
2. Right-click the Global Resource in the Project window, and select **Copy Global Resource into Project**.

The resource now becomes a standard connection applicable only to the current project. Taking the example above, the connection will become an ADO.NET connection if you've chosen that configuration.

14.4   Example: Switch Environments

With Global Resources, you can change not only the database connection, but also reload any SQL scripts that you've defined as Global Resources. This example shows you how Global Resources can help you switch the working environment from a "development" to a "production" one.

First, you will create a Global Resource of type "database" that contains two configurations: a "development" and a "production" database. Next, you will create two SQL scripts that query the "development" and the "production" database, respectively. Once this is done, you will be able to quickly switch to the desired environment from the graphical user interface of DatabaseSpy.

Step 1: Create the database resources

1. On the **Tools** menu, click **Global Resources**. (Alternatively, click the **Global Resource** 🔳 toolbar button.)
2. Click **Add | Database**.
3. Enter a descriptive name in the **Resource alias** text box (for example, "MyDB").
4. Click **Choose Database**, select the desired connection method (for example, **SQLite**), and browse for (or connect to) the *development* database. Notice that the database connection details are attached to the **Default** configuration.
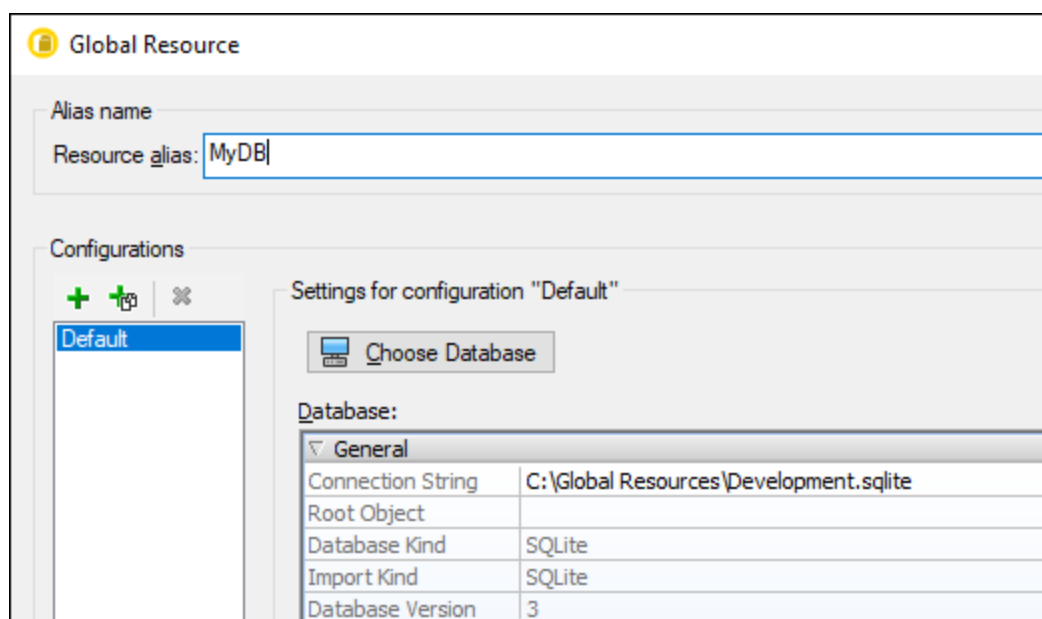
5. Click **Add Configuration** ➕ and name it "Release".
6. Click **Choose Database**, select the desired connection method (for example, **SQLite**), and this time browse for (or connect to) the *production* database.
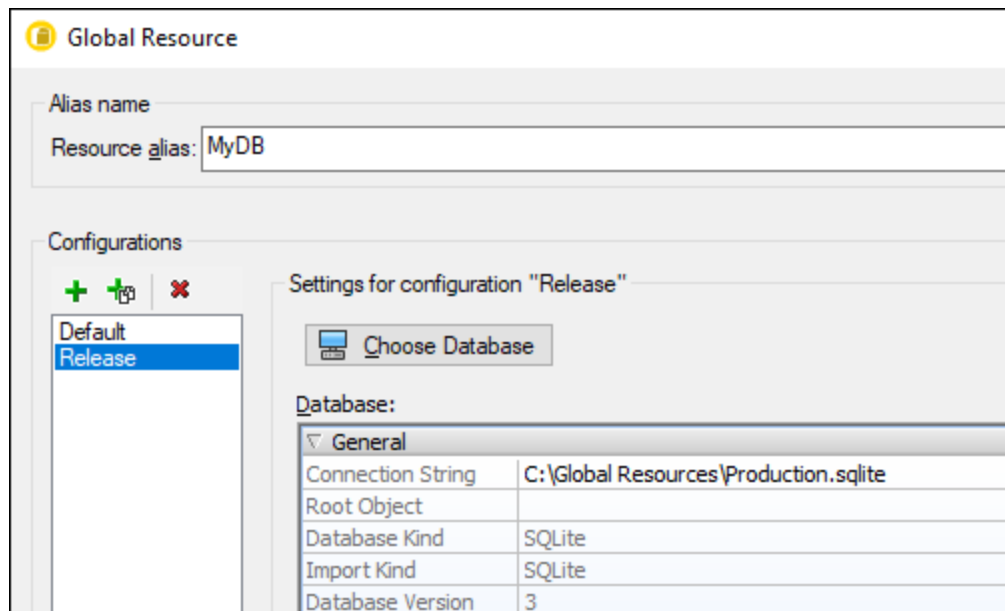
At this stage, you have created two configurations for the "MyDB" resource:

1. **Default** - points to the development database
2. **Release** - points to the production database.

Step 2: Create the SQL scripts as resources

1. On the **Tools** menu, click **Global Resources**. (Alternatively, click the **Global Resource** 🗔 toolbar button.)
2. Click **Add | File**.
3. Enter a descriptive name in the **Resource alias** text box (for example, "MyInsertScript").
4. Under "The Resource will point to this file", click **Browse** and select a SQL script that contains queries meant for the *development* database.
5. Click **Add Configuration** 🞡 and name it "Release".
6. Under "The Resource will point to this file", click **Browse** and select a SQL script that contains queries meant for the *production* database.

At this stage, you have created two configurations for the "MyInsertScript" resource:

1. **Default** - points to a development SQL script
2. **Release** - points to a production SQL script.

Step 3: Switch Environments

1. On the **File** menu, click **New Project**.
2. Click **Connect to a database** 🗔, or press **Ctrl+Q**.
3. On the database connection wizard, click **Global Resources** and select the "MyDB" Global Resource created previously.
4. On the **File** menu, point to **Open | Open Global Resource**.
5. Select the "MyInsertScript" resource created previously, and click **Open**.
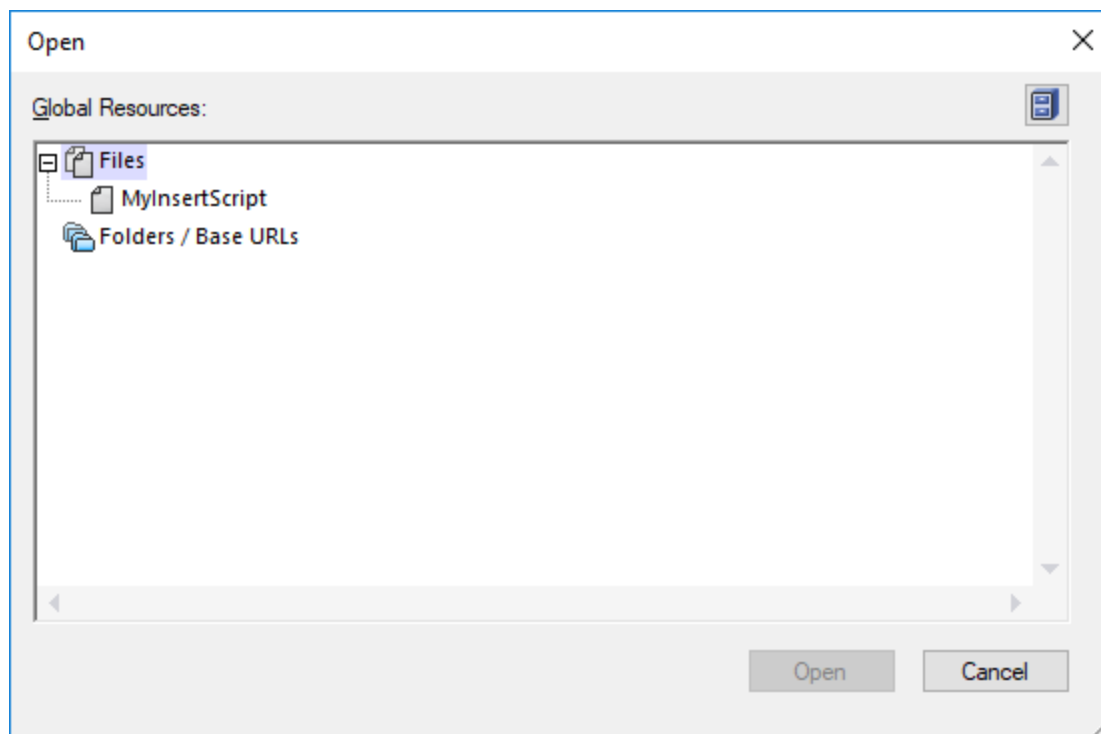
You can now easily switch between development and production environments, by selecting the desired configuration from the list of active configurations.

Each time the environment is reloaded, a message box opens, asking for confirmation, and also indicating the exact resources that are being reloaded, for example:
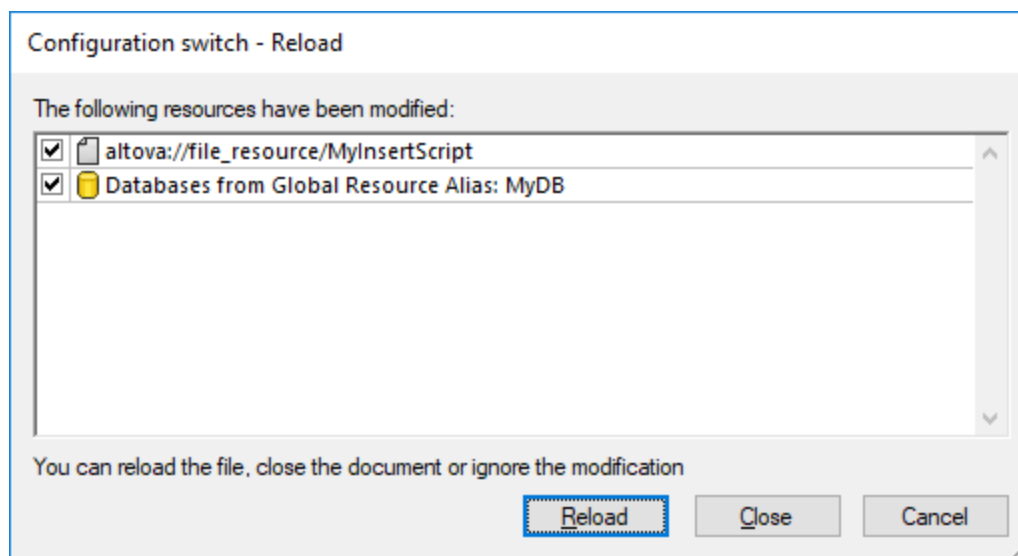
6.   Click **Reload** to confirm the configuration switch.

15   Menu Commands

The **Menus** section contains a complete description of all DatabaseSpy menu commands. We've tried to make this user manual as comprehensive as possible. If, however, you have questions which are not covered in the User Reference or other parts of this documentation, please look up the FAQs and Discussion Forums on the Altova website. If you are still not able to have your problem satisfactorily addressed, please do not hesitate to contact us through the Support Center on the Altova website.

Note that in the File [448] and Edit [453] menus, all standard Windows commands are supported, as well as additional database-related commands.

15.1   File Menu

The **File** menu contains all commands relevant to manipulating files, in the order common to most Windows software products.
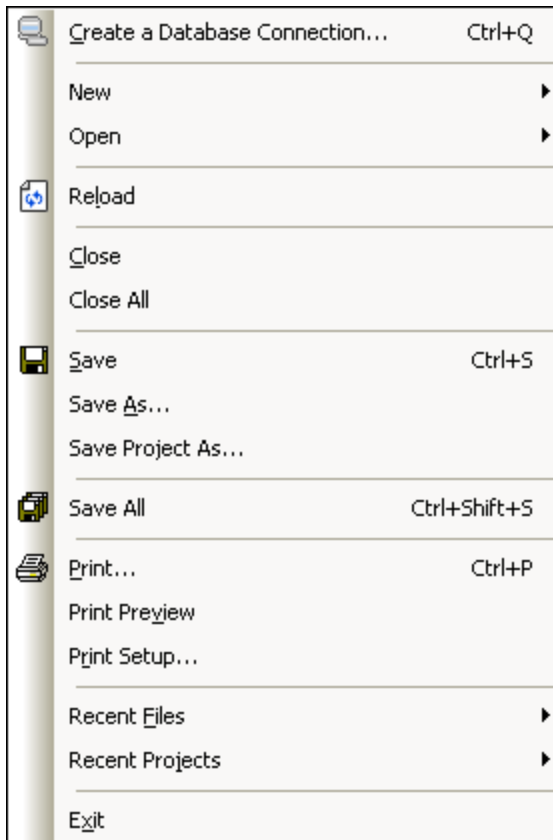
In addition to the standard Open [450], Save [451], Print [451], Print Setup [452], and Exit [452] commands, DatabaseSpy offers a range of application-specific commands.

15.1.1   Create a Database Connection...

    Ctrl+Q

The **Create Database Connection...** command opens a database connection wizard which assists you to connect to various database kinds supported by DatabaseSpy. For more information, see Starting the Database Connection Wizard [117].

In the General Options [418], you can specify whether or not the wizard should be displayed each time when DatabaseSpy starts.

15.1.2   New

The **New** sub-menu contains options for opening new [projects](#)[449], [SQL Editor](#)[449] and [Design Editor](#)[449] windows.

15.1.2.1 Project

[icon]   **Ctrl+Shift+N**

The **Project** command creates a new [project](#)[206] in DatabaseSpy. If you are currently working with another project, a prompt appears, asking if you want to close all documents belonging to the current project.

15.1.2.2 SQL Editor

[icon]   **Ctrl+N**

The **SQL Editor** command opens a new [SQL Editor window](#)[39].

15.1.2.3 Design Editor

[icon]   **Ctrl+D**

The **Design Editor** command opens a new [Design Editor window](#)[46].

15.1.3   Open

The **Open** sub-menu provides options for opening [projects](#)[449], [files](#)[450], and [Altova Global Resources](#)[450].

[icon] Open Project	Ctrl+Shift+O
[icon] Open File...	Ctrl+O
[icon] Open Global Resource...	
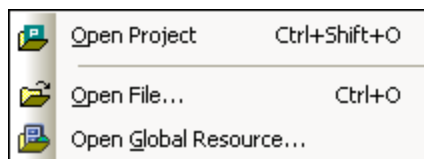
15.1.3.1 Open Project...

[icon]   **Ctrl+Shift+O**

The **Open Project...** command opens an existing [project](#)[206] in DatabaseSpy. The standard Windows Open dialog box appears where you can choose a project file (*.qprj). If you are currently working with another project, the previous project is closed first.

15.1.3.2 Open File...

🗁   **Ctrl+O**

The **Open...** command pops up the familiar Windows Open dialog box and allows you to open any project file (*.qprj), SQL file (*.sql), or design file (*.qdes). In the Open dialog box, you can select more than one file to open. Use the Files of Type combo box to restrict the kind of files displayed in the dialog box.

15.1.3.3 Open Global Resource...

The **Open Global Resource...** command opens the Choose Global Resource dialog box where you can select a file or folder and display it in the SQL Editor. The **Open Global Resources dialog...** button displays a dialog box where you can manage Global Resources. For more information, see [Example: Switch Environments](443).

If you select a folder that has been defined as a global resource, then the standard Windows Open dialog box appears and displays the content of the directory that is defined for the active configuration.

15.1.4   Reload

The **Reload** command replaces a file currently open in DatabaseSpy (for example, a SQL script in SQL Editor) with the file version last saved to the disk. Any unsaved changes that you made in DatabaseSpy will be lost. You can still keep the updated version by clicking **No** in the message box that pops up.

15.1.5   Close

The **Close** command closes the active window. If the file was modified (indicated by an asterisk * after the file name in the title bar), you will be asked if you wish to save the file first.

15.1.6   Close All

The **Close All** command closes all open windows. If any document has been modified (indicated by an asterisk * after the file name in the title bar), you will be asked if you wish to save the file first.

Please note that the project will not be closed by the **Close All** command. A project can only be closed by opening a new project or by exiting DatabaseSpy.

15.1.7   Save

🖫   **Ctrl+S**

The **Save** command saves the contents of the active window to the file from which it has been opened. If the active window contains a new file, the Save As... [451] dialog box is opened automatically.

15.1.8   Save As...

The **Save As...** command pops up the familiar Windows **Save As** dialog box, in which you enter the name and location of the file you wish to save the active file as.

15.1.9   Save Project As...

The **Save Project As...** command saves the project that is currently open under a new name or at a different path. The standard Windows **Save As** dialog box appears where you can enter the desired file name and path.

15.1.10   Save All

Ctrl+Shift+S

The **Save All** command saves all modifications that have been made to any open documents. The command is useful if you edit multiple documents simultaneously. If a document has not been saved before (for example, after being newly created), the **Save As...** dialog box is presented for that document.

15.1.11   Print...

🖨   **Ctrl+P**

The **Print...** command opens the Print dialog box, in which you can select printer options and print the active file.

Please note that the **Print...** command behaves differently depending on whether it is called from an SQL Editor window or Design Editor window:

- Choosing the **Print...** command from an SQL Editor window brings up the standard Windows Print dialog box.
- If you select the **Print...** command for a design file in the Design Editor, DatabaseSpy presents a Print dialog box where you can choose which part of the design file is to be printed and adapt the zoom. For more information, see Printing Designs [234].

15.1.12   Print Preview

The **Print Preview** command opens a preview window where you can check the layout of your active window before printing. Note that the **Print Preview** command behaves differently depending on whether it is called from an SQL Editor window or Design Editor window:

- If you select **File | Print Preview** from an active SQL Editor window, a preview window opens immediately where you can browse the pages and zoom in or zoom out using the appropriate buttons. For more information, see Printing SQL [381].
- When called from an active Design Editor window, the **Print Preview** command will first open the Print dialog box for design files. For more information, see Printing Designs [233].

15.1.13   Print Setup...

The **Print Setup...** command displays the printer-specific Print Setup dialog box, in which you specify such printer settings as paper format and page orientation. These settings are applied to all subsequent print jobs.

15.1.14   Recent Files

The **Recent Files** command in the **File** menu opens a sub-menu displaying a list of the most recently used files, with the most recently opened file shown at the top of the list. You can open any of these files by clicking its name. To open a file in the list using the keyboard, press **Alt+F** to open the **File** menu, and then press the number of the file you want to open.

The list of recent files does not include any files that are part of the currently open DatabaseSpy project.

15.1.15   Recent Projects

The **Recent Projects** command opens a submenu displaying the file name for the four most recently used projects, allowing quick access to these files.

Also note, that DatabaseSpy can automatically open the last project that you used, whenever you start DatabaseSpy. (**Tools | Options | General | Open last project on startup** [418]).

15.1.16   Exit

The **Exit** command is used to quit DatabaseSpy. If you have any open files with unsaved changes, you are prompted to save these changes. DatabaseSpy also saves modifications to program settings and information about the most recently used files.

15.2   Edit Menu

The **Edit** menu contains commands for editing text in the SQL Editor window.

↰	Undo	Alt+Backspace
↱	Redo	Ctrl+Y
✂	Cut	Shift+Delete
📄	Copy	Ctrl+C
📋	Paste	Ctrl+V
	Select All	Ctrl+A
	Find...	Ctrl+F
	Find next	F3
	Replace...	Ctrl+H
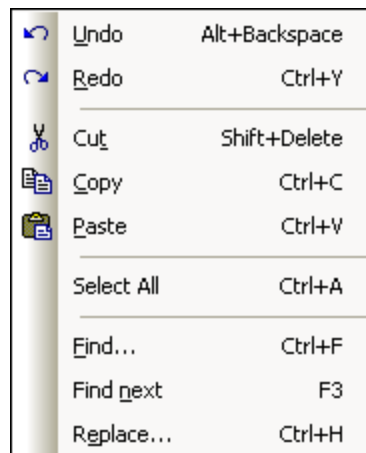
Different options are provided if the active window is a Design Editor.

↰	Undo	Alt+Backspace
↱	Redo	Ctrl+Y
✂	Cut	Shift+Delete
📄	Copy	Ctrl+C
📋	Paste	Ctrl+V
	Select User Tables	Alt+T
	Select System Tables	Alt+S
	Select All	Ctrl+A
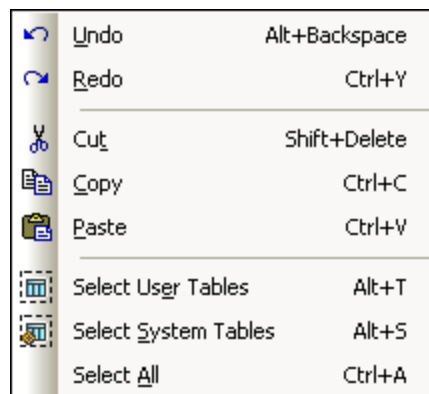
15.2.1   Undo

↰   Alt+Backspace

The **Undo** command contains support for unlimited levels of Undo. Every action can be undone and it is possible to undo one command after another. The Undo history is retained after using the **Save** command, enabling you go back to the state the document was in before you saved your changes.

15.2.2   Redo

↱   Ctrl+Y

The **Redo** command allows you to redo previously undone commands, thereby giving you a complete history of work completed. You can step back and forward through this history using the **Undo** and **Redo** commands.

15.2.3   Cut

Shift+Delete

The **Cut** command copies the selected text to the clipboard and deletes it from its present location.

15.2.4   Copy

Ctrl+C

The **Copy** command copies the selected text to the clipboard. This can be used to duplicate data within DatabaseSpy or to move data to another application.

15.2.5   Paste

Ctrl+V

The **Paste** command inserts the contents of the clipboard at the current cursor position.

15.2.6   Select User Tables

Alt+T

The **Select User Tables** command selects all user tables that are contained in the active Design Editor window.

15.2.7   Select System Tables

Alt+S

The **Select System Tables** command selects all system tables that are contained in the active Design Editor window.

15.2.8   Select All

Ctrl+A

The **Select All** command selects (i) any text in the currently active SQL Editor window or (ii) all tables in the currently active Design Editor window.

15.2.9   Find...

Ctrl+F

The **Find** command pops up the Find [375] dialog, in which you can specify the string you want to find and other options for the search.

15.2.10   Find Next

F3

The **Find next** command repeats the last Find... [455] command to search for the next occurrence of the requested text.

15.2.11   Replace...

Ctrl+H

The **Replace** command enables you to find and replace one text string with another text string. It features the same options as the Find... [455] command. You can replace each item individually or you can use the **Replace All** button to perform a global search-and-replace operation.

15.3   View Menu

The **View** menu contains commands for switching the various windows on or off.

Online Browser	
Project Window	
Property Window	Alt+Enter
Overview Window	
Data Inspector Window	
Charts Window	
Output Window	
Database Structure Change Script	
✓ Status	
Toggle All Side Windows	
Toggle Output and Change Script Windows	
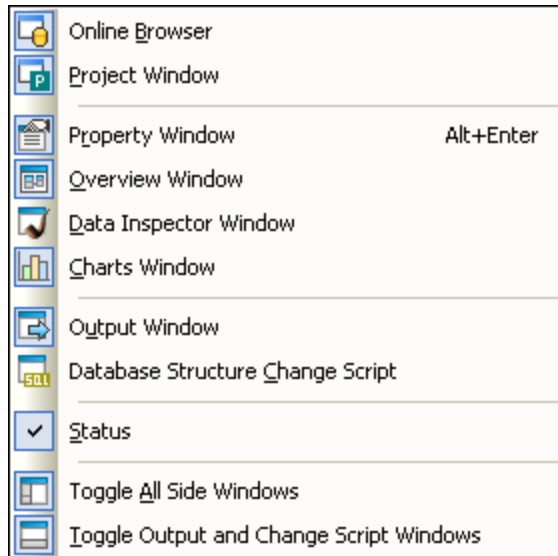
15.3.1   Online Browser

This command lets you switch the Online Browser [24] on or off.

This is a dockable window. Dragging on its title bar detaches it from its current position and makes it a floating window. Click right on the title bar to allow docking or hide the window.

15.3.2   Project Window

This command lets you switch the Project window [20] on or off.

This is a dockable window. Dragging on its title bar detaches it from its current position and makes it a floating window. Click right on the title bar to allow docking or hide the window.

15.3.3   Property Window

This command lets you switch the [Property window](#) [30] on or off.

This is a dockable window. Dragging on its title bar detaches it from its current position and makes it a floating window. Click right on the title bar to allow docking or hide the window.

15.3.4   Overview Window

This command lets you switch the [Overview window](#) [31] on or off.

This is a dockable window. Dragging on its title bar detaches it from its current position and makes it a floating window. Click right on the title bar to allow docking or hide the window.

15.3.5   Data Inspector Window

This command lets you switch the [Data Inspector window](#) [32] on or off.

This is a dockable window. Dragging on its title bar detaches it from its current position and makes it a floating window. Click right on the title bar to allow docking or hide the window.

15.3.6   Charts Window

This command lets you switch the [Chart window](#) [35] on or off.

This is a dockable window. Dragging on its title bar detaches it from its current position and makes it a floating window. Click right on the title bar to allow docking or hide the window.

15.3.7   Output Window

This command lets you switch the [Output window](#) [33] on or off.

This is a dockable window. Dragging on its title bar detaches it from its current position and makes it a floating window. Click right on the title bar to allow docking or hide the window.

15.3.8   Database Structure Change Script

This command lets you switch the [Database Structure Change Script window](#)[37] on or off.

This is a dockable window. Dragging on its title bar detaches it from its current position and makes it a floating window. Click right on the title bar to allow docking or hide the window.

Note: The Database Structure Change Script window becomes visible whenever you take an action that is about to change the structure of the database, even if you have previously closed this window.

15.3.9   Status

The **Status** command toggles on and off the Status bar.

15.3.10   Toggle All Side Windows

This command toggles on and off the display of the Online Browser and the Project, Properties, and Overview windows as well as any other window that has been docked to the side of the application. Note that this command applies also to windows that are docked on the right side of DatabaseSpy.

15.3.11   Toggle Output and Change Script Windows

This command toggles the display of the Output and Change Script windows on and off.

15.4   SQL Editor Menu

The **SQL Editor** menu contains commands for selecting, editing, and executing SQL statements.

▷	Execute	F5
▷	Execute for Data Editing	
	Navigation	▶
	Insert	▶
	Bookmarks	▶
⊞	Result view	
▦	Message view	
♨	Stack Result Windows	
WR RAP	Word Wrap	
	Text View Settings	
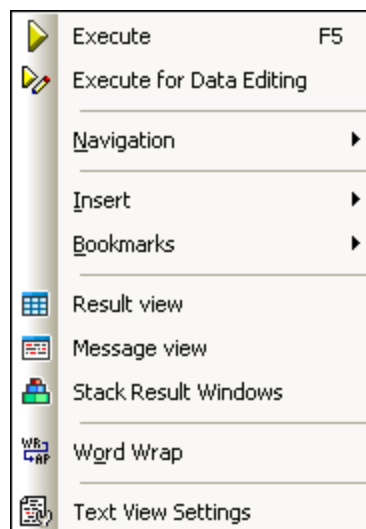
The **SQL Editor** menu is only visible if the active window contains an SQL file.

15.4.1   Execute

▷   **F5**

The **Execute command** executes all the SQL statements[364] that are currently in the active window of the SQL Editor. You can customize the behaviour[365] of the **Execute** command in Properties window of the active SQL Editor options.

◼

The **Stop Retrieval** command stops the retrieval of database data. The data that has been retrieved so far is displayed in the Result tab. Use the **Retrieve next n rows** button or the **Retrieve outstanding rows** button, respectively, or the context menu in the Result view to resume retrieval. This button replaces the **Execute** button while data is being retrieved.

15.4.2   Execute for Data Editing

▷

The **Execute for Data Editing** command executes a SELECT statement in the Online Browser and switches the Result window into the Editing Mode so that you are able to update or even add new data directly in the result grid. This command is only available if data editing from within the SQL Editor is supported for the data source connection you are currently working with.

15.4.3   Navigation

The **Navigation** sub-menu contains options for selecting statements in the active SQL Editor window.

▼	Next Statement	Alt+Down
▲	Previous Statement	Alt+Up
▣	Last Statement	Alt+End
▣	First Statement	Alt+Home
▣	Select entire Statement	Shift+Alt+Enter

15.4.3.1 Next Statement

Alt+Down

The **Next Statement** command selects the statement following the statement in which the cursor is currently located.

15.4.3.2 Previous Statement

Alt+Up

The **Previous Statement** command selects the statement preceding the statement in which the cursor is currently located.

15.4.3.3 Last Statement

Alt+End

The **Last Statement** command selects the last statement in active window of the SQL Editor.

15.4.3.4 First Statement

Alt+Home

The **First Statement** command selects the first statement in the active window of the SQL Editor.

15.4.3.5 Select Entire Statement

Shift+Alt+Enter

The **Select Entire Statement** command selects the SQL statement in which the cursor is currently located.

15.4.4   Insert

The Insert sub-menu contains options for inserting block [461] and line [461] comments, targets [461], and regions [461] into the active SQL Editor window.

The **Insert/Remove Block Comment** and **Insert/Remove Line Comment** options are toggle commands that can also be used to remove a previously inserted comment.

15.4.4.1 Insert/Remove Block Comment

The **Insert/Remove Block comment** inserts or removes a block comment [371] around the currently selected text block in the active window of the SQL Editor.

15.4.4.2 Insert/Remove Line Comment

The **Insert/Remove line comment** command inserts line comment [371] from the current position of the cursor to the end of the line or removes a selected line comment.

15.4.4.3 Insert Target

The **Insert Target** command inserts `--Target: Result` before the currently marked SELECT statement. Executing the statement places the result in a tab named "Result". The text "Result" can be edited, allowing you to create your own named tabs, see Naming Result Tabs [373].

15.4.4.4 Insert Region

The **Insert Region** command creates a region [362] for the currently selected block of text. Regions can be folded or expanded by clicking the + icon.

15.4.4.5 Insert Abort on Execution

Ā

This command adds a comment that stops the execution of the script if an error is encountered in any SQL statements following the comment. For details, see [Stop execution through a script comment](#) [367].

15.4.5 Bookmarks

The **Bookmarks** sub-menu provides commands for inserting and removing bookmarks, and allows you to navigate between the bookmarks in an SQL Editor window.

▣	Insert/Remove Bookmark
▣	Go to Next Bookmark
▣	Go to Previous Bookmark
▣	Remove All Bookmarks
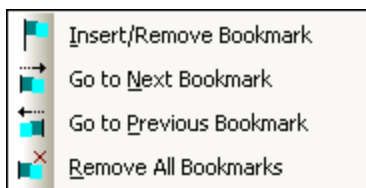
15.4.5.1 Insert/Remove Bookmark

▣

The **Insert/Remove Bookmark** command inserts a bookmark at the line of the current cursor position, or removes the bookmark if the cursor is in a line that has been bookmarked previously.

Bookmarked lines are displayed in one of the following ways:

- If the [bookmarks margin](#) [361] has been enabled, then a solid cyan ellipse appears to the left of the text in the bookmark margin.
- If the bookmarks margin has not been enabled, then the complete line containing the cursor is highlighted.

15.4.5.2 Go to Next Bookmark

▣

The **Go to Next Bookmark** command places the text cursor at the beginning of the next bookmarked line.

15.4.5.3 Go to Previous Bookmark

▣

The **Go to Previous Bookmark** command places the text cursor at the beginning of the previous bookmarked line.

15.4.5.4 Remove All Bookmarks

The **Remove All Bookmarks** command removes all the defined bookmarks from the SQL Editor window that is currently active.

Note: The Undo command does not undo the effects of this command.

15.4.6   Result View

The **Result view** command toggles all **Result** tabs in SQL Editor windows on and off.

15.4.7   Message View

The **Message view** command toggles the **Message** tab in SQL Editor windows on and off.

15.4.8   Stack Result Windows

If the **Stack Result Windows** option is deactivated, results of queries are displayed in separate, sequentially numbered tabs of the Result window (Result1, Result2, etc.). If the option is activated, all results are displayed in a single result tab with multiple panes for the individual queries.

15.4.9   Show Groupings For Execution

The **Show groupings for execution** command toggles the display of execution groups for SQL statements [366] on and off.

15.4.10   Word Wrap

The **Word Wrap** command enables or disables word wrapping in the SQL Editor.

15.4.11   Text View Settings

The **Text View Settings** command opens the Text View Settings dialog box where you can [define settings](359)
for the display of line number, bookmark, and source folding margins, specify the tab size, and define which
visual aids are available in the SQL Editor.

15.5   SQL Refactoring Menu

The **SQL Refactoring** menu contains commands for changing the layout of SQL code in the SQL Editor.

Format SQL
Add Semicolons
Remove Semicolons
Add quotation to identifiers
Remove quotation from identifiers
Create View As
Uppercase Keywords
Expand columns for star expressions
Remove Comments
Remove Comments and Formatting
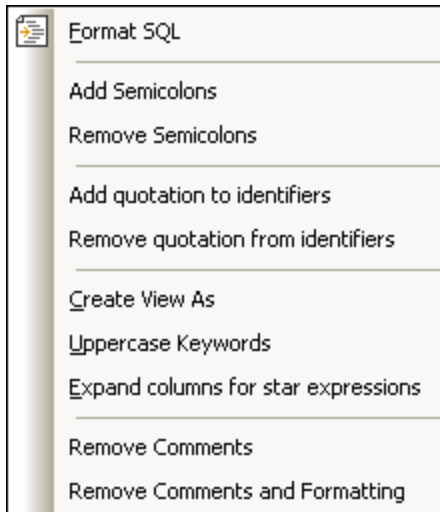
The **SQL Refactoring** menu is only visible if the active window contains an SQL file.

15.5.1   Format SQL

The **Format SQL** command applies SQL formatting as currently defined for the respective database kind to the active SQL Editor Window.

A message is displayed in the **Message** tab, after the formatting has been completed.

15.5.2   Add Semicolons

The **Add Semicolons** command appends a semicolon to the end of each statement. No semicolon will be added however if it already exists in a statement.

15.5.3   Remove Semicolons

The **Remove Semicolons** command removes all semicolons from the end of statements. It makes no difference whether the semicolons have been entered manually or using the Add Semicolons[465] command.

15.5.4   Add Quotations to Identifiers

This command adds database-specific quotations (e.g., ", ' [], etc.) to the table and column identifiers in the SQL Editor.

15.5.5   Remove Quotations from Identifiers

This command removes database-specific quotations (e.g., ", ' [], etc.) from the table and column identifiers in the SQL Editor. It makes no difference whether the identifiers have been entered manually or using the Add Quotations to Identifiers [466] command.

15.5.6   Create View As

The **Create View As** command creates a view from a SELECT statement in the active SQL Editor window, see Creating Views [269].

15.5.7   Uppercase Keywords

The **Uppercase Keywords** converts all keywords in the active SQL Editor into uppercase.

15.5.8   Expand Columns for Star Expressions

The **Expand columns for star expressions** command inserts the full column list of a table if the cursor is placed after the asterisk in a SELECT statement and the command is executed.

Note that the SQL statement must be complete and valid. If, for example, the SQL Editor should require semicolons option is checked, the terminating semicolon must indeed be present in the SQL Editor.

15.5.9   Remove Comments

This command removes all text that is commented out. This includes text enclosed within block comment marks (`/*` and `*/`) and text preceded by the line comment mark `--`.

15.5.10   Remove Comments and Formatting

The **Remove Comments and Formatting** command reverts all formatting that has been added by the Format SQL [465] command, that is, any additional line feed or whitespace is removed and the SQL is displayed in one line. In addition, this command removes all text that has been commented out.

15.6   Design Editor Menu

The **Design Editor** menu contains commands for maintaining database objects in a graphical manner.

🔳	Create New Table	Ctrl+T
🔲	Create New Column	Alt+C
🔲	Add Sticky Note	Alt+N
	Add related tables	▶
	SQL and Data	▶
🔍	Zoom In	Ctrl+Num +
🔍	Zoom Out	Ctrl+Num -
◀◦▶	Zoom to fit	
🔺	Auto Layout whole Diagram	
🔺	Auto Layout selected Tables	
	Remove from Design	Ctrl+Delete
	Delete selected Objects	Delete
🔜	Export database data	
🔲	Save Diagram as Image	
🔲	Generate Database Structure Change Script	
🔳	Show Design Options	

The Design Editor menu is only visible if the active window contains a design file.

15.6.1   Create New Table

🔳   **Ctrl+T**

The **Create New Table** command creates a new table in the currently active Design Editor window and automatically assigns it to the active data source. A Database Structure Change Script is generated and has to be executed to add the newly created table to the database.

15.6.2   Create New Column

🔲   **Alt+C**

The **Create New Column** command adds a new column to the selected table. If you have selected more than one table, the column will be added to the table that was selected last.

15.6.3   Add Sticky Note

Alt+N

The **Add Sticky Note** command adds a Sticky Note to the Design Editor window. Double-click the note to enter text and drag it to the desired position.

15.6.4   Add Related Tables

The **Add Related Tables** command opens a sub-menu which adds tables that are either referenced by or are referencing the selected table.

Referenced Tables
Referencing Tables
All Relations
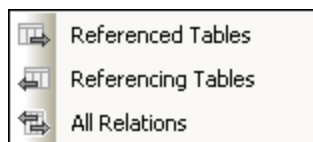
15.6.4.1 Referenced Tables

The **Referenced Tables** command adds tables that are referenced by the selected table to the active Design Editor window.

15.6.4.2 Referencing Tables

The **Referencing Tables** command adds tables which include a reference to the selected table to the active Design Editor window.

15.6.4.3 All Relations

The **All Relations** command adds both referenced and referencing tables to the active Design Editor window.

15.6.5   SQL and Data

The **SQL and Data** command opens a sub-menu which provides options for generating SQL scripts [469], retrieving [469], and editing [470] data from within the design view.

Show in new SQL Editor   ▶
Retrieve data   ▶
🖉 Edit Data   Alt+E
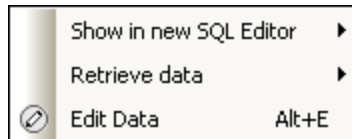
15.6.5.1 Show in new SQL Editor

The **Show in new SQL Editor** command opens a sub-menu which enables you to create SQL statements from the selected tables in the Design Editor.

Select
────────
Create
Alter
Drop
────────
Insert
Update
Delete
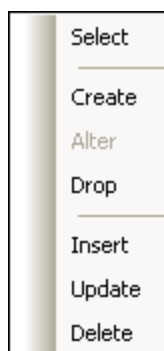
If more than one table is selected, a statement is created for each of the selected tables. Please note that different statements are created depending on what you have selected:

- If you have selected the entire table, one statement is created for all of the columns in the table.

```
SELECT [AnimalTypeID], [EnglishName], [LatinName],
[AreaOfOrigin], [Category] FROM [ZooDB].[dbo].[tblAnimalTypes];
```

- If you have selected a single column, the statement is created for this particular column only.

```
SELECT [EnglishName] FROM [ZooDB].[dbo].[tblAnimalTypes];
```

- If you have selected some but not all of the columns, one statement is created for the selected columns.

```
SELECT [EnglishName], [AreaOfOrigin] FROM
[ZooDB].[dbo].[tblAnimalTypes];
```

15.6.5.2 Retrieve Data

The **Retrieve Data** command opens a sub-menu which enables you to perform a query on the selected table in the Design Editor.

All rows
First 200 Rows
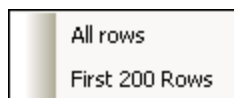
15.6.5.2.1   All rows

The **All rows** command generates a SELECT statement for all rows in a new SQL Editor window and executes the query immediately. The result of the query is displayed in the **Result** tab below the SQL Editor window.

15.6.5.2.2   First n rows

The **First n rows** command generates a SELECT statement for the first n rows in a new SQL Editor window and executes the query immediately. The result of the query is displayed in the **Result** tab below the SQL Editor window.

You can customize the number of rows to be retrieved in the SQL Editor options[425].

15.6.5.3 Edit Data

⌀   **Ctrl+Alt+E**

The **Edit Data** command retrieves data from the selected table or columns and switches the result grid into the editing mode (if supported for the respective data source).

15.6.6   Zoom In

⊕   **Ctrl+Num +**

The **Zoom In** command increases the zoom factor of the Design Editor window by 10 percent. You can also zoom in by scrolling (with the scroll-wheel of the mouse) while keeping the **Ctrl** key pressed.

15.6.7   Zoom Out

⊖   **Ctrl+Num -**

The **Zoom Out** command decreases the zoom factor of the Design Editor window by 10 percent. You can also zoom out by scrolling (with the scroll-wheel of the mouse) while keeping the **Ctrl** key pressed.

15.6.8   Zoom to Fit

⟠

The **Zoom to Fit** command sizes the working area to include all tables in the current Design Editor window.

15.6.9   Auto Layout Whole Diagram

The **Auto Layout Whole Diagram** command arranges all tables in the active Design Editor window in a way that considers relations between the individual tables as well as proper layouting.

15.6.10   Auto Layout Selected Tables

The **Auto Layout Selected Tables** command arranges the selected tables in the active Design Editor window in a way that considers relations between the individual tables as well as a proper layouting.

15.6.11   Remove from Design

Ctrl+Del

The **Remove from Design** command removes the selected table from the Design Editor window. Please note that the table is not deleted from the database in this case. To drop a table from the database, use the Delete Selected Objects [471] command instead.

15.6.12   Delete Selected Objects

Del

The **Delete Selected Objects** command generates a Database Structure Change Script and marks the selected objects for deletion. You must execute the change script in order to actually delete the objects from the database.

15.6.13   Export Database Data

The **Export Database Data** command opens the Export database data [406] dialog box where you can define the export settings. If you have tables selected in the Design Editor window, the check boxes of these tables will already be activated in the **Source** group box of the **Export database data** dialog box and thus selected for export.

15.6.14   Save Diagram As Image

The **Save Diagram As Image** command opens the standard Windows Save As dialog box where you can specify a name and location for the design image. The designs can be saved as Portable Network Graphics (*.png).

15.6.15   Generate Database Structure Change Script

The **Generate Database Structure Change Script** command starts the generation of a change script or updates it after changes occurred to the database structure. Please note that this menu option is not available if in the General options [418] you have activated the **Instantly** radio button in the **Database Structure Change Script** group box.

15.6.16   Show Design Options

The **Show Design Options** command opens the Design Editor page of the Options [433] dialog box where you can customize the settings for the Design Editor.

15.7   Tools Menu

The **Tools** menu allows you to:

- Export [473] and import [473] database data
- Generate SQL change scripts [474]
- Manage XML schemas [474] for databases
- Access user-defined tools [474]
- Manage Global Resources [474] and configurations [474]
- Customize [474] menus, toolbars, etc.
- Restore toolbars and windows [486] to their default state
- Define the global program settings

⮕	Export database data...	Ctrl+E
⬅	Import data to the database...	Ctrl+I
🗄	Generate Change Script	
🗐	XML Schema Management for Databases ...	
	User-defined tools	▶
🗂	Global Resources...	
	Active configurations	▶
	Customize ...	
	Restore Toolbars and Windows...	
⚙	Options ...	Ctrl+Alt+O
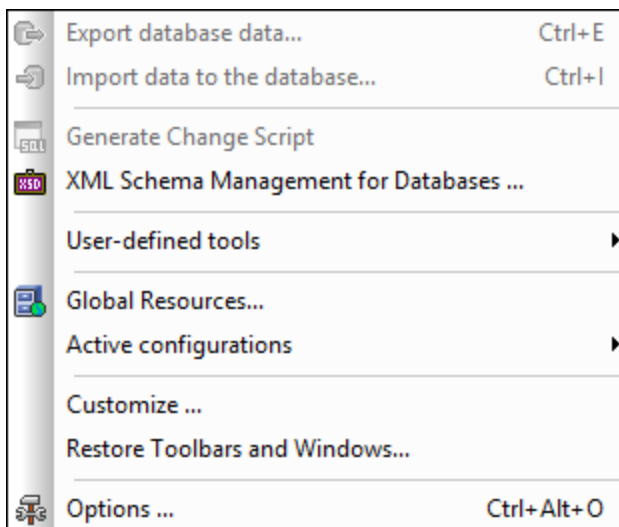
15.7.1   Export Database Data...

⮕   **Ctrl+E**

The **Export Database Data...** command opens the Export Database Data [406] dialog box where you can define your export options.

15.7.2   Import Data to the Database...

⬅   **Ctrl+I**

The **Import Data to Database...** command opens the Import data to the database [400] dialog box where you can define your import options.

15.7.3   Generate Change Script

The **Generate Change Script** command starts the generation of a change script or updates it after changes occurred to the database structure. Please note that this menu option is not available if in the General options [418] you have activated the **Instantly** radio button in the **Database Structure Change Script** group box.

15.7.4   XML Schema Management for Databases...

The **XML Schema Management for Databases** command opens the **XML Schema management for databases** dialog box which allows you to manage XML schemas for the active data source, see Managing XML Schemas [397].

15.7.5   User-defined Tools

The **User-defined Tools** command opens a sub-menu where you can add your personal menu items, e.g., links to other applications.

To add a user-defined tool, select the menu option **Tools | Customize** or right-click into the toolbar and select **Customize** from the context menu. Change to the Tools tab [477] and add name of and path to the EXE file of the application you want to start from within DatabaseSpy. The name you entered in the **Menu contents** box appears in the **Tools | User-defined Tools** sub-menu.

15.7.6   Global Resources...

The **Global Resources...** command opens the **Manage Global Resources** dialog box, where you can add, edit, or delete global resources [436].

15.7.7   Active Configurations

The **Active configurations** command opens a sub-menu, where all available configurations are listed and the active configuration is indicated. Selecting a configuration in the sub-menu will make it the active configuration.

15.7.8   Customize...

The **Customize...** command lets you customize DatabaseSpy to suit your personal needs.

15.7.8.1 Commands

The **Commands** tab allows you customize your menus or toolbars.
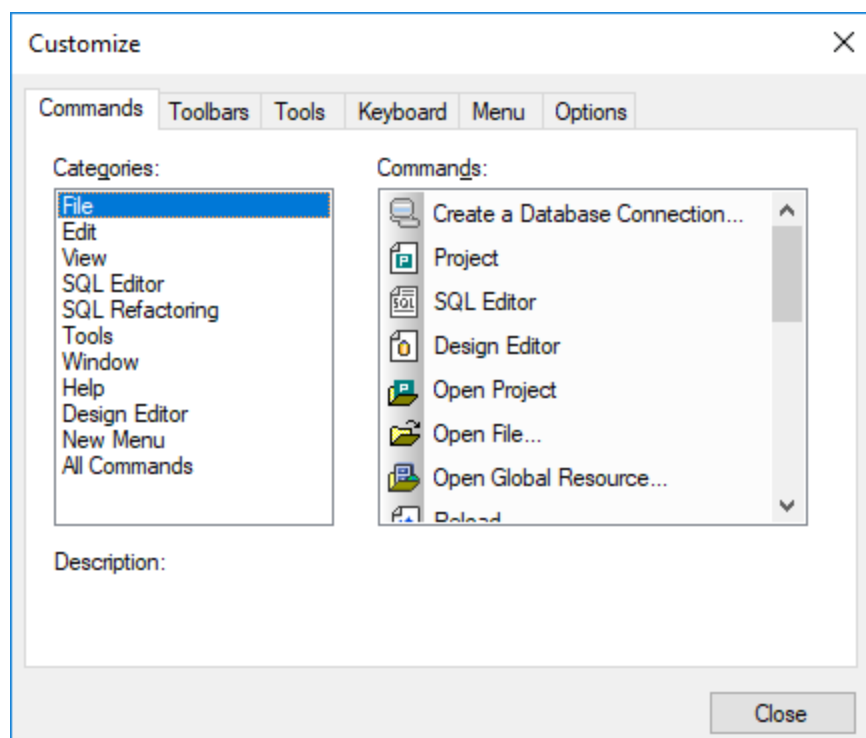
To add a command to a toolbar or menu:

1. Select the menu option **Tools | Customize**. The Customize dialog box appears.
2. Select the **All Commands** category in the Categories list box. The available commands appear in the Commands list box.
3. Click on a command in the Commands list box and drag it to an existing menu or toolbar. An I-beam appears when you place the cursor over a valid position to drop the command.
4. Release the mouse button at the position you want to insert the command.

   - A small button appears at the tip of mouse pointer when you drag a command. The "x" below the pointer means that the command cannot be dropped at the current cursor position.
   - The "x" disappears whenever you can drop the command (over a tool bar or menu).
   - Placing the cursor over a menu when dragging opens it, allowing you to insert the command anywhere in the menu.
   - Commands can be placed in menus or toolbars. If you created you own toolbar you can populate it with your own commands/icons.

Note: You can also edit the commands in the context menus [484] (right-click anywhere to open the context menu), using the same method. Click the **Menu** tab and then select the specific context menu available in the **Context Menus** combo box.

15.7.8.2 Toolbars

The **Toolbars** tab allows you to activate or deactivate specific toolbars, as well as create your own specialized ones.

DatabaseSpy toolbars contain symbols for the most frequently used menu commands. For each symbol you get a brief "tool tip" explanation when the mouse cursor is directly over the item and the status bar shows a more detailed description of the command.

You can drag the toolbars from their standard position to any location on the screen, where they appear as a floating window. Alternatively, you can also dock them to the left or right edge of the main window.
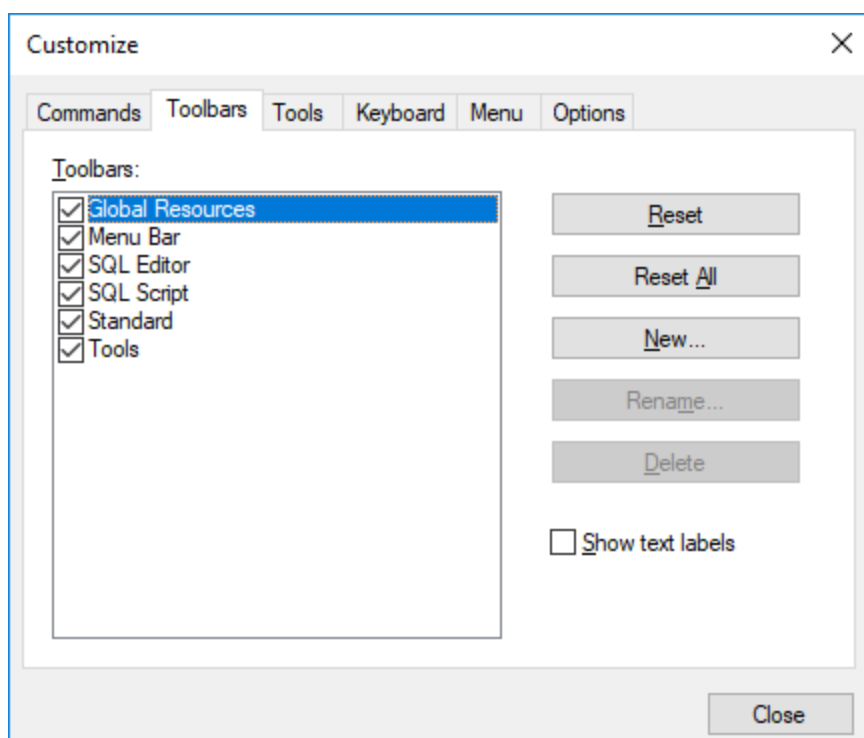
The **Show text labels** option displays explanatory text below toolbar icons when activated. You can activate or deactivate this option for each toolbar individually.

To activate or deactivate a toolbar:

- Click the check box to activate (or deactivate) the specific toolbar.

To create a new toolbar:

1. Click the New... button, and give the toolbar a name in the Toolbar Name dialog box that appears.
2. Drag commands to the toolbar in the <u>Commands</u> [475] tab of the Customize dialog box.

To reset the Menu Bar:

1. Click the Menu Bar entry.
2. Click the **Reset** button, to reset the menu commands to the state they were in when DatabaseSpy was installed.

To reset all toolbar and menu commands:

1. Click the **Reset All** button to reset all the toolbar commands to the state they were when the program was installed. A prompt appears stating that all toolbars and menus will be reset.
2. Click **Yes** to confirm the reset.

To change a toolbar name:

- Click the **Rename...** button to edit the name of the toolbar.

This option is available only for user-defined toolbars.

To delete a toolbar:

1. Select the toolbar you want to delete in the **Toolbars** list box.
2. Click the **Delete** button. A prompt appears, asking if you really want to delete the toolbar.
3. Click **Yes** to confirm the deletion.

This option is available only for user-defined toolbars.

15.7.8.3 Tools

The **Tools** tab allows you to add your personal menu items, such as shortcuts to other applications, to the **Tools** menu.
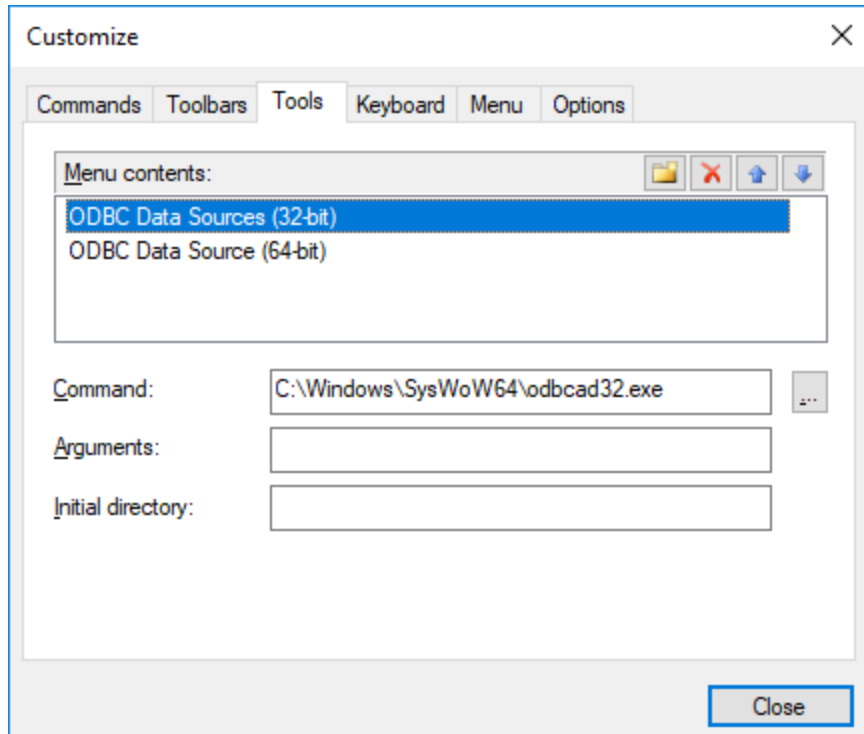
Any user-defined entries will appear in the **User-defined tools** sub-menu.
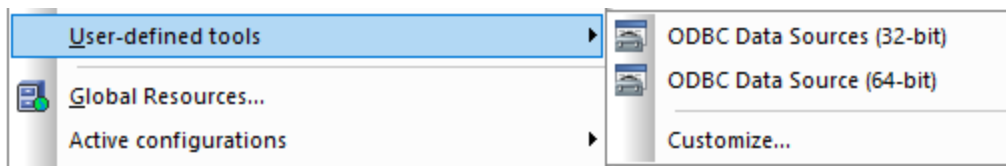
To add a user-defined tool:

1. Click the **New** 📂 icon in the **Menu contents** title bar or press the Insert key.
2. In the text field that appears, enter the string that is to appear as a menu entry in the **Tools** menu.
3. In the **Command** field, enter the path to the executable (.exe) file of the application or click **Browse** 📄 and select the path.
4. If applicable, enter the arguments to be passed to the application at start-up in the **Arguments** field.
5. Enter the path to the initial directory, if required.

To change the sequence of user-defined tools:

- Select an entry and use the **Move Item Up** 🔼 and **Move Item Down** 🔽 buttons to move the menu item. Alternatively, you can use the shortcut keys **Alt+Up** and **Alt+Down** to move the entry.

To rename a user-defined menu item:

- Double-click an item in the **Menu contents** field and edit the entry.

To remove a user-defined tool:

- Select the entry you want to remove from the **Tools** menu and click the **Delete** ☒ icon in the **Menu contents** title bar or press the **Delete** key.

15.7.8.4 Keyboard

The **Keyboard** tab allows you to define (or change) keyboard shortcuts for any DatabaseSpy command.
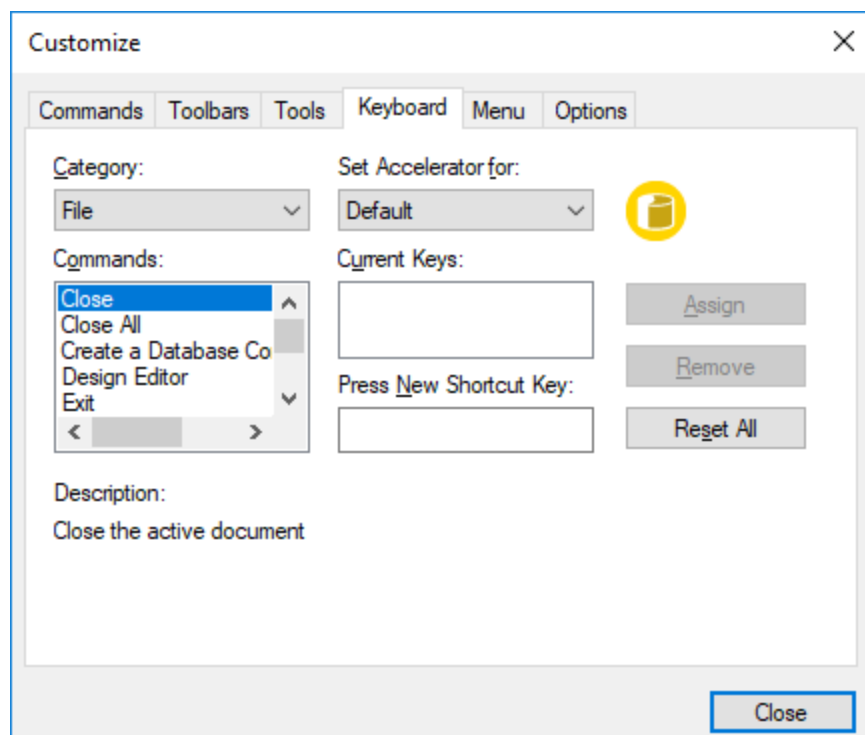
In the **Set accelerator for** drop-down list, you can differentiate between shortcuts that are valid in the Default menu or in the DatabaseSpy Design menu.

To assign a new Shortcut to a command:

1. Select the **All Commands** category using the **Category** combo box.
2. Select the command you want to assign a new shortcut to, in the **Commands** list box.

3.   Click in the **Press New Shortcut Key** text box, and press the shortcut keys that are to activate the command. The shortcuts appear immediately in the text box. If the shortcut was assigned previously, then that function is displayed below the text box. (To clear this text box, press any of the control keys, **Ctrl**, **Alt** or **Shift**).
4.   Click the **Assign** button to assign the shortcut.

The shortcut now appears in the **Current Keys** list box.

To de-assign or delete a shortcut:

1.   Click the shortcut you want to delete in the **Current Keys** list box.
2.   Click the **Remove** button.
3.   Click the **Close** button to confirm.

To reset all shortcut keys:

1.   Click the **Reset All** button to reset all the shortcut keys to the state they were when the program was installed. A prompt appears stating that all toolbars and menus will be reset.
2.   Click **Yes** to confirm the reset.

The following tables list the currently assigned keyboard shortcuts by key and by function, respectively.

Hotkeys by key

Shortcut	Command name
F1	Table of Contents
F3	Find next
F5	Execute
Alt	Show Paths
Alt+Enter	Property Window
Shift+Alt+Enter	Select entire Statement
Shift+Delete	Cut
Alt+Backspace	Undo
Ctrl+Insert	Copy
Shift+Insert	Paste

Shortcut	Command name
Alt+Down	Next Statement
Alt+Up	Previous Statement
Alt+Home	First Statement
Alt+End	Last Statement
Alt+Insert	Append new row
Ctrl+C	Copy/Copy selected cells
Ctrl+D	New Design Editor
Ctrl+E	Export database data...
Ctrl+F	Find...
Ctrl+H	Replace...
Ctrl+I	Import data to the database...
Ctrl+N	New SQL Editor
Ctrl+O	Open...
Ctrl+P	Print...
Ctrl+Q	Create a database connection...
Ctrl+S	Save
Ctrl+S	Save Project
Ctrl+V	Paste
Ctrl+X	Cut
Ctrl+Y	Redo
Ctrl+Z	Undo
Ctrl+Alt+E	Edit data
Ctrl+Alt+O	Options...
Ctrl+Alt+R	Retrieve all rows
Ctrl+Alt+T	Retrieve first n rows

Shortcut	Command name
Ctrl+Shift+N	New Project
Ctrl+Shift+O	Open Project
Ctrl+Shift+S	Save All
Ctrl+Shift+V	Paste as new row

Hotkeys by function

Command name	Shortcut	Alternative shortcut
Append new row	**Alt+Insert**	
Copy	**Ctrl+C**	**Ctrl+Insert**
Copy selected cells	**Ctrl+C**	
Cut	**Ctrl+X**	**Shift+Delete**
Edit data	**Ctrl+Alt+E**	
Execute	**F5**	
Export database data...	**Ctrl+E**	
Find next	**F3**	
Find...	**Ctrl+F**	
First Statement	**Alt+Home**	
Import data to the database	**Ctrl+I**	
Last Statement	**Alt+End**	
New Design Editor	**Ctrl+D**	
New Project	**Ctrl+Shift+N**	
New SQL Editor	**Ctrl+N**	
Next Statement	**Alt+Down**	
Open...	**Ctrl+O**	
Open Project	**Ctrl+Shift+O**	
Options...	**Ctrl+Alt+O**	
Paste	**Ctrl+V**	**Shift+Insert**
Paste as new row	**Ctrl+Shift+V**	

Command name	Shortcut	Alternative shortcut
Previous Statement	**Alt+Up**	
Print...	**Ctrl+P**	
Property Window	**Alt+Enter**	
Create a database connection	**Ctrl+Q**	
Redo	**Ctrl+Y**	
Replace...	**Ctrl+H**	
Retrieve all rows	**Ctrl+Alt+R**	
Retrieve first n rows	**Ctrl+Alt+T**	
Save	**Ctrl+S**	
Save All	**Ctrl+Shift+S**	
Save Project	**Ctrl+S**	
Select entire Statement	**Shift+Alt+Enter**	
Show Paths	**Alt**	
Table of Contents	**F1**	
Undo	**Ctrl+Z**	**Alt+Backspace**

15.7.8.5 Menu

The **Menu** tab allows you to customize menu bars as well as context menus.
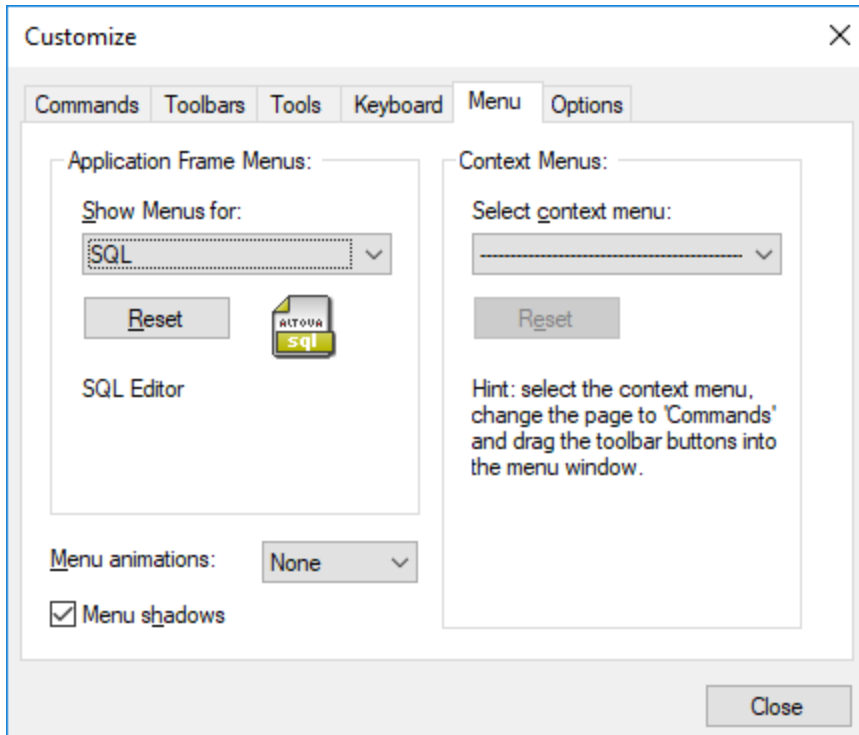
Customizing menus

The **Default Menu** bar is the menu bar that is displayed when no document is open in the main window. The **SQL** menu bar is the menu bar that is displayed when a file is open in SQL Editor. The **Design** menu bar is the menu bar that is displayed when a file is open in Design Editor. Each menu bar can be customized separately, and customization changes made to one do not affect the other.

To customize a menu bar, select it from the **Show Menus For** drop-down list. Then switch to the **Commands** tab and drag commands from the **Commands** list box to the menu bar or into any of the menus.

Deleting commands from menus and resetting the menu bars

To delete an entire menu or a command inside a menu, do the following:

1. Select one of the following from the **Show Menus for** drop-down list:

   - **Default Menu** (this shows available menus when no document is open)
   - **SQL** (this shows available menus when a file is open in SQL Editor)
   - **Design** (this shows available menus when a file is open in Design Editor)

2. With the Customize dialog open, select (i) the menu you want to delete from the application's menu bar, or (ii) the command you want to delete from one of these menus.

3. Either (i) drag the menu from the menu bar or the menu command from the menu, or (ii) right-click the menu or menu command and select **Delete**.

You can reset any menu bar to its original installation state by selecting it from the **Show Menus For** drop-down list and then clicking the **Reset** button.

Customizing the application's context menus

Context menus are the menus that appear when you right-click certain objects in the application's interface. Each of these context menus can be customized by doing the following:

1. Select the context menu from the **Select context menu** drop-down list. This pops up the context menu.
2. Click the **Commands** tab.
3. Drag a command from the **Commands** list box into the context menu.
4. To delete a command from the context menu, right-click that command in the context menu, and select **Delete**. Alternatively, drag the command out of the context menu.

You can reset any context menu to its original installation state by selecting it in the **Select context menu** drop-down list and then clicking the **Reset** button.

Menu shadows

Select the **Menu shadows** check box to give all menus shadows.

You can choose from among several menu animations if you prefer animated menus. The **Menu animations** drop-down list provides the following options:

- None (default)
- Unfold
- Slide
- Fade

15.7.8.6 Options

The **Options** tab allows you to set general environment settings.
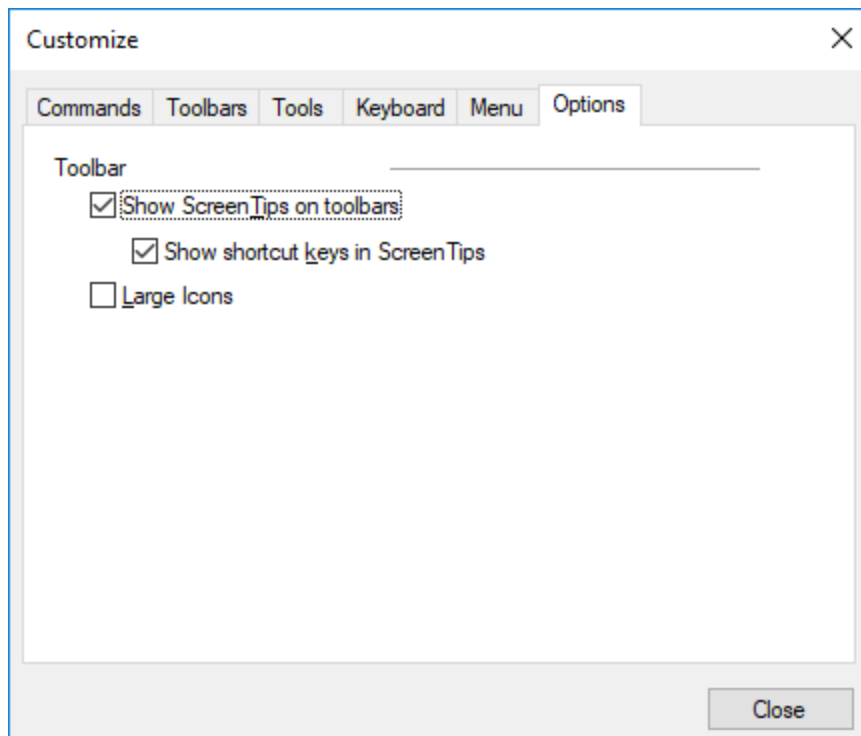
When active, the **Show ScreenTips on toolbars** check box displays a tooltip when the mouse pointer is placed over any toolbar button. The tooltip contains a short description of the button, as well as the associated keyboard shortcut, if one has been assigned.

The **Show shortcut keys in ScreenTips** check box allows you to decide whether or not you want to have the shortcut displayed in the tooltip.

When active, the **Large icons** check box switches between the standard size icons, and larger versions of the icons.

15.7.9   Restore Toolbars and Windows

This menu command resets the toolbars, entry helper windows, docked windows etc. to their defaults. DatabaseSpy needs to be restarted for the changes to take effect.

15.7.10   Options...

Ctrl+Alt+O

The **Options** command opens the Options dialog box where you can set the General options [418] as well as specific settings for SQL [425] or Design Editor [433].

15.8   Window Menu

To organize the individual windows in an DatabaseSpy session, the **Window** menu contains standard commands common to most Windows applications. You can also switch to an open document window directly from the menu.

🗗	Cascade
🗖	Tile horizontally
🗖	Tile vertically
✓	1 SelectHostByYear.sql
	Windows...
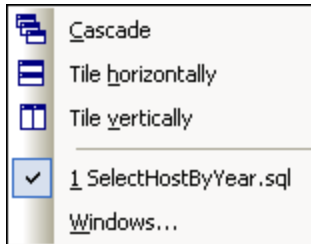
You can cascade the open windows, tile them, or arrange document icons once you have minimized them.

15.8.1   Cascade

🗗

The **Cascade** command rearranges all open document windows so that they are all cascaded (i.e., staggered) on top of each other.

15.8.2   Tile horizontally

🗖

The **Tile horizontally** command rearranges all open document windows as horizontal tiles, making them all visible at the same time.

15.8.3   Tile vertically

🗖

The **Tile vertically** command rearranges all open document windows as vertical tiles, making them all visible at the same time.

15.8.4   Currently open window list

This list shows all currently open windows and lets you quickly switch between them.

You can also use the **Ctrl+Tab** or **Ctrl+F6** keyboard shortcuts to cycle through the open windows.

15.8.5   Windows...

At the bottom of the list of open windows is the **Windows...** command. Clicking this entry opens the Windows dialog box, which displays a list of all open windows and provides commands that can be applied to the selected window(s). A window is selected by clicking on its name.

Note: To exit the Windows dialog box, click **OK**; do not click the **Close Window(s)** button in the dialog box. The **Close Window(s)** button closes the window(s) currently selected in the Windows dialog box.

15.9   Help Menu

This section describes all the menu commands available in the **Help** menu.

⊟ Table of Contents

Opens the onscreen help manual of DatabaseSpy with the Table of Contents displayed in the left-hand-side pane of the Help window. The Table of Contents provides an overview of the entire Help document. Clicking an entry in the Table of Contents takes you to that topic.

⊟ Index

Opens the onscreen help manual of DatabaseSpy with the Keyword Index displayed in the left-hand-side pane of the Help window. The index lists keywords and lets you navigate to a topic by double-clicking the keyword. If a keyword is linked to more than one topic, a list of these topics is displayed.

⊟ Search

Opens the onscreen help manual of DatabaseSpy with the Search dialog displayed in the left-hand-side pane of the Help window. To search for a term, enter the term in the input field and press Enter or List Topics. The Help system performs a full-text search on the entire Help documentation and returns a list of hits. Double-click any item to display that item.

⊟ Software Activation

License your product
After you download your Altova product software, you can license—or activate—it using either a free evaluation key or a purchased permanent license key.

- *Free evaluation license.* When you first start the software after downloading and installing it, the **Software Activation** dialog will pop up. In it is a button to request a free evaluation license. Enter your name, company, and e-mail address in the dialog and click **Request**. A license file is sent to the e-mail address you entered and should reach you in a few minutes. Save the license file to a suitable location.

   When you clicked **Request**, an entry field appeared at the bottom of the Request dialog. This field takes the path to the license file. Browse for or enter the path to the license file and click **OK**. (In the **Software Activation** dialog, you can also click **Upload a New License** to access a dialog in which the path to the license file is entered.) The software will be unlocked for a period of 30 days.

- *Permanent license key.* The **Software Activation** dialog allows you to purchase a permanent license key. Clicking this button takes you to Altova's online shop, where you can purchase a permanent license key for your product. Your license will be sent to you by e-mail in the form of a license file, which contains your license-data.

   There are three types of permanent license: *installed*, *concurrent user*, and *named user*. An installed license unlocks the software on a single computer. If you buy an installed license for *N* computers, then the license allows use of the software on up to *N* computers. A concurrent-user license for *N* concurrent users allows *N* users to run the software concurrently. (The software may be installed on 10N computers.) A named-user license authorizes a specific user to use the software on up to 5 different computers. To activate your software, click **Upload a New License**, and, in the dialog that appears, enter the path to the license file, and click **OK**.

Note: For multi-user licenses, each user will be prompted to enter his or her own name.

> *Your license email and the different ways to license (activate) your Altova product*
> The license email that you receive from Altova will contain your license file as an attachment.
> The license file has a `.altova_licenses` file extension.
>
> To activate your Altova product, you can do one of the following:
>
> - Save the license file (`.altova_licenses`) to a suitable location, double-click the
>   license file, enter any requested details in the dialog that appears, and finish by
>   clicking **Apply Keys**.
> - Save the license file (`.altova_licenses`) to a suitable location. In your Altova
>   product, select the menu command **Help | Software Activation**, and then **Upload a
>   New License**. Browse for or enter the path to the license file, and click **OK**.
> - Save the license file (`.altova_licenses`) to any suitable location, and upload it from
>   this location to the license pool of your Altova LicenseServer. You can then either: (i)
>   acquire the license from your Altova product via the product's Software Activation
>   dialog *(see below)* or (ii) assign the license to the product from Altova LicenseServer.
>   *For more information about licensing via LicenseServer, read the rest of this topic.*

You can access the **Software Activation** dialog (*screenshot below*) at any time by clicking the **Help |
Software Activation** command.

Activate your software
You can activate the software by registering the license in the Software Activation dialog or by licensing via
Altova LicenseServer (*see details below*).

- *Registering the license in the Software Activation dialog*. In the dialog, click **Upload a New
  License** and browse for the license file. Click **OK** to confirm the path to the license file and to
  confirm any data you entered (your name in the case of multi-user licenses). Finish by clicking
  Save.

- *Licensing via Altova LicenseServer on your network:* To acquire a license via an Altova
  LicenseServer on your network, click **Use Altova LicenseServer**, located at the bottom of the
  Software Activation dialog. Select the machine on which the LicenseServer you want to use has
  been installed. Note that the auto-discovery of License Servers works by means of a broadcast
  sent out on the LAN. As these broadcasts are limited to a subnet, License Server must be on the
  same subnet as the client machine for auto-discovery to work. If auto-discovery does not work,
  then type in the name of the server. The Altova LicenseServer must have a license for your Altova
  product in its license pool. If a license is available in the LicenseServer pool, this is indicated in
  the **Software Activation** dialog (*see screenshot below showing the dialog in Altova XMLSpy*).
  Click **Save** to acquire the license.
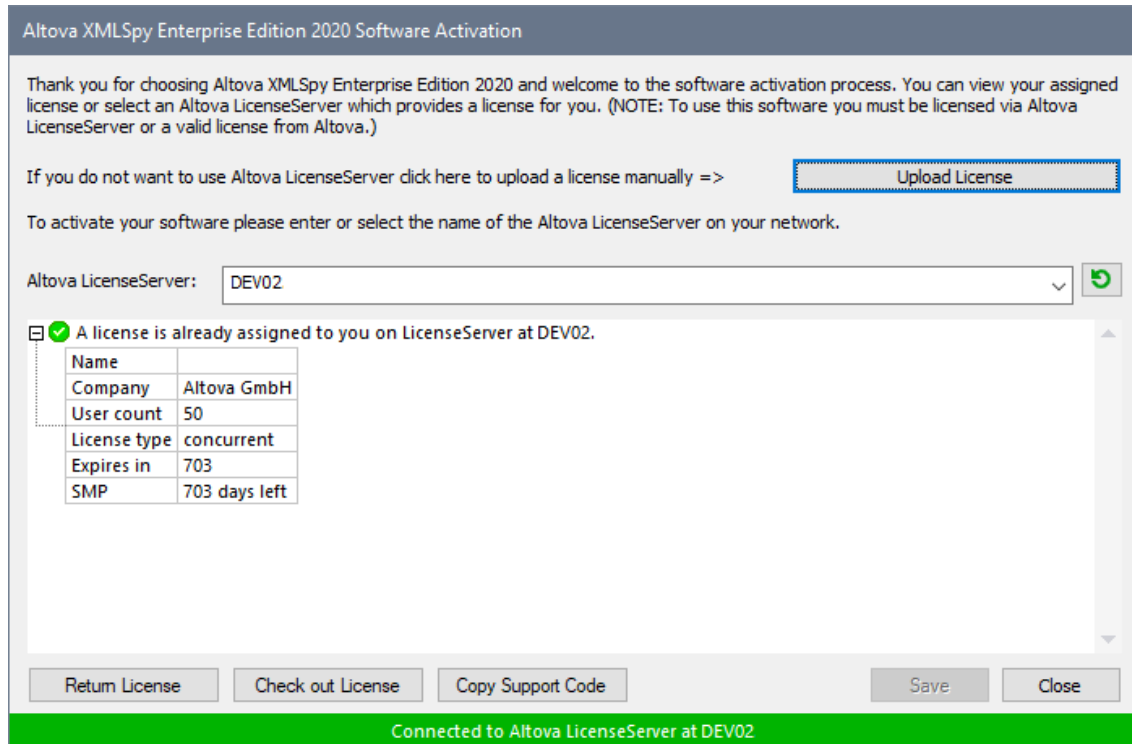
After a machine-specific (aka installed) license has been acquired from LicenseServer, it cannot be returned to LicenseServer for a period of seven days. After that time, you can return the machine license to LicenseServer (click **Return License**) so that this license can be acquired from LicenseServer by another client. (A LicenseServer administrator, however, can unassign an acquired license at any time via the administrator's Web UI of LicenseServer.) Note that the returning of licenses applies only to machine-specific licenses, not to concurrent licenses.

Check out license
You can check out a license from the license pool for a period of up to 30 days so that the license is stored on the product machine. This enables you to work offline, which is useful, for example, if you wish to work in an environment where there is no access to your Altova LicenseServer (such as when your Altova product is installed on a laptop and you are traveling). While the license is checked out, LicenseServer displays the license as being in use, and the license cannot be used by any other machine. The license automatically reverts to the checked-in state when the check-out period ends. Alternatively, a checked-out license can be checked in at any time via the **Check in** button of the **Software Activation** dialog.

To check out a license, do the following: (i) In the **Software Activation** dialog, click **Check out License** (*see screenshot above*); (ii) In the **License Check-out** dialog that appears, select the check-out period you want and click **Check out**. The license will be checked out. After checking out a license, two things happen: (i) The **Software Activation** dialog will display the check-out information, including the time when the check-out period ends; (ii) The **Check out License** button in the dialog changes to a **Check In** button. You can check the license in again at any time by clicking **Check In**. Because the license automatically reverts to the checked-in status after the check-out period elapses, make sure that the check-out period you select adequately covers the period during which you will be working offline.

License check-ins must be to the same major version of the Altova product from which the license was checked out. So make sure to check in a license before you upgrade your Altova product to the next major version.

Note: For license check-outs to be possible, the check-out functionality must be enabled on LicenseServer. If this functionality has not been enabled, you will get an error message to this effect when you try to check out. In this event, contact your LicenseServer administrator.

Copy Support Code
Click **Copy Support Code** to copy license details to the clipboard. This is the data that you will need to provide when requesting support via the online support form.

Altova LicenseServer provides IT administrators with a real-time overview of all Altova licenses on a network, together with the details of each license as well as client assignments and client usage of licenses. The advantage of using LicenseServer therefore lies in administrative features it offers for large-volume Altova license management. Altova LicenseServer is available free of cost from the Altova website. For more information about Altova LicenseServer and licensing via Altova LicenseServer, see the Altova LicenseServer documentation.

☐ Order Form

When you are ready to order a licensed version of the software product, you can use either the **Purchase a Permanent License Key** button in the **Software Activation** dialog (*see previous section*) or the **Order Form** command to proceed to the secure Altova Online Shop.

☐ Registration

Opens the Altova Product Registration page in a tab of your browser. Registering your Altova software will help ensure that you are always kept up to date with the latest product information.

☐ Check for Updates

Checks with the Altova server whether a newer version than yours is currently available and displays a message accordingly.

☐ Support Center

A link to the Altova Support Center on the Internet. The Support Center provides FAQs, discussion forums where problems are discussed, and access to Altova's technical support staff.

☐ FAQ on the Web

A link to Altova's FAQ database on the Internet. The FAQ database is constantly updated as Altova support staff encounter new issues raised by customers.

☐ Download Components and Free Tools

A link to Altova's Component Download Center on the Internet. From here you can download a variety of companion software to use with Altova products. Such software ranges from XSLT and XSL-FO processors to Application Server Platforms. The software available at the Component Download Center is typically free of charge.

☐ DatabaseSpy on the Internet

A link to the [Altova website](#) on the Internet. You can learn more about DatabaseSpy, related technologies and products on the [Altova website](#).

⊟ About DatabaseSpy

Displays the splash window and version number of your product. If you are using the 64-bit version of DatabaseSpy, this is indicated with the suffix (x64) after the application name. There is no suffix for the 32-bit version.

16   License Information

This section contains information about:

- the distribution of this software product
- software activation and license metering
- the license agreement governing the use of this product

Please read this information carefully. It is binding upon you since you agreed to these terms when you installed this software product.

To view the terms of any Altova license, go to the Altova Legal Information page at the Altova website.

16.1   Electronic Software Distribution

This product is available through electronic software distribution, a distribution method that provides the following unique benefits:

- You can evaluate the software free-of-charge for 30 days before making a purchasing decision. *(Note: Altova MobileTogether Designer is licensed free of charge.)*
- Once you decide to buy the software, you can place your order online at the Altova website and get a fully licensed product within minutes.
- When you place an online order, you always get the latest version of our software.
- The product package includes an onscreen help system that can be accessed from within the application interface. The latest version of the user manual is available at www.altova.com in (i) HTML format for online browsing, and (ii) PDF format for download (and to print if you prefer to have the documentation on paper).

30-day evaluation period

After downloading this product, you can evaluate it for a period of up to 30 days free of charge. About 20 days into the evaluation period, the software will start to remind you that it has not yet been licensed. The reminder message will be displayed once each time you start the application. If you would like to continue using the program after the 30-day evaluation period, you must purchase a product license, which is delivered in the form of a license file containing a key code. Unlock the product by uploading the license file in the Software Activation dialog of your product.

You can purchase product licenses at https://shop.altova.com/.

Helping Others within Your Organization to Evaluate the Software

If you wish to distribute the evaluation version within your company network, or if you plan to use it on a PC that is not connected to the Internet, you may distribute only the installer file, provided that this file is not modified in any way. Any person who accesses the software installer that you have provided must request their own 30-day evaluation license key code and after expiration of their evaluation period, must also purchase a license in order to be able to continue using the product.

16.2   Software Activation and License Metering

As part of Altova's Software Activation, the software may use your internal network and Internet connection for the purpose of transmitting license-related data at the time of installation, registration, use, or update to an Altova-operated license server and validating the authenticity of the license-related data in order to protect Altova against unlicensed or illegal use of the software and to improve customer service. Activation is based on the exchange of license related data such as operating system, IP address, date/time, software version, and computer name, along with other information between your computer and an Altova license server.

Your Altova product has a built-in license metering module that further helps you avoid any unintentional violation of the End User License Agreement. Your product is licensed either as a single-user or multi-user installation, and the license-metering module makes sure that no more than the licensed number of users use the application concurrently.

This license-metering technology uses your local area network (LAN) to communicate between instances of the application running on different computers.

Single license

When the application starts up, as part of the license metering process, the software sends a short broadcast datagram to find any other instance of the product running on another computer in the same network segment. If it doesn't get any response, it will open a port for listening to other instances of the application.

Multi-user license

If more than one instance of the application is used within the same LAN, these instances will briefly communicate with each other on startup. These instances exchange key-codes in order to help you to better determine that the number of concurrent licenses purchased is not accidentally violated. This is the same kind of license metering technology that is common in the Unix world and with a number of database development tools. It allows Altova customers to purchase reasonably-priced concurrent-use multi-user licenses.

We have also designed the applications so that they send few and small network packets so as to not put a burden on your network. The TCP/IP ports (2799) used by your Altova product are officially registered with the IANA (*see the IANA Service Name Registry for details*) and our license-metering module is tested and proven technology.

If you are using a firewall, you may notice communications on port 2799 between the computers that are running Altova products. You are, of course, free to block such traffic between different groups in your organization, as long as you can ensure by other means, that your license agreement is not violated.

Note about certificates

Your Altova application contacts the Altova licensing server (`link.altova.com`) via HTTPS. For this communication, Altova uses a registered SSL certificate. If this certificate is replaced (for example, by your IT department or an external agency), then your Altova application will warn you about the connection being insecure. You could use the replacement certificate to start your Altova application, but you would be doing this at your own risk. If you see a *Non-secure connection* warning message, check the origin of the certificate and consult your IT team (who would be able to decide whether the interception and replacement of the Altova certificate should continue or not).

If your organization needs to use its own certificate (for example, to monitor communication to and from client machines), then we recommend that you install Altova's free license management software, Altova LicenseServer, on your network. Under this setup, client machines can continue to use your organization's certificates, while Altova LicenseServer can be allowed to use the Altova certificate for communication with Altova.

16.3   Altova End-User License Agreement

- The Altova End-User License Agreement is available here: https://www.altova.com/legal/eula
- Altova's Privacy Policy is available here: https://www.altova.com/privacy

Index

A

Activating,

    row count, 419

    toolbars, 476

Adding,

    all related tables, 468

    commands, 475

    data in result grid, 304

    escape characters to identifiers, 426

    file extensions, 422

    quotations to identifiers, 466

    records in result grid, 304

    referenced tables, 468

    referencing tables, 468

    related tables, 468

    semicolons, 465

    sticky note, 468

    user-defined tools, 477

ADO,

    as data connection interface, 115

    setting up a connection, 122

ADO.NET,

    setting up a connection, 130

Appending,

    rows with copied data, 304

    semi-colons to statement end, 426

Application startup, 418

Applying SQL formatting, 465

area chart features, 329

Auto layout,

    selected tables, 471

    whole diagram, 471

Autocompletion,

    context-sensivite suggestion, 369

    invoking manually, 369

    single mode, 369

    triggering automatically, 369

Autoinsertion,

    choosing characters, 428

    deleting closing characters, 428

    disabling, 428

    enabling, 428

    wrapping selected text, 428

B

bar chart features, 329

Block comment, 371

    inserting, 461

    removing, 461

Bookmark margin, 359

Bookmarks,

    bookmark margin, 359

    go to next, 462

    go to previous, 462

    in SQL View, 462

    inserting, 359, 462

    navigating, 359

    removing, 359, 462

    removing all, 463

C

candlestick chart features, 329

Cascade, 488

Case sensitivity, 419

Change script,

    delay during execution, 420

    disabling SQL formatting, 427

    generating, 472, 474

    generation mode, 420

Change Script window,

    toggling on and off, 458

Changing,

    binary content, 301

    content of binary columns, 301

    grid colors, 429

    toolbar name, 476

Charts,

    3d settings, 340

    adding legend, 327

    appearance, 318

    area chart features, 329

    background color, 327

    bar chart features, 329

T