

Executing SQL Statements

In the SQL Commander, you can execute one or multiple statements. You can also control if the execution should stop or continue when the execution of a statement results in a warning or error.

- [Execute a Script with Multiple Statements](#)
- [Execute Only the Current Statement](#)
- [Execute Buffer](#)
- [Control Execution after a Warning or an Error](#)

Execute a Script with Multiple Statements

Use the **SQL Commander->Execute** main menu operation to execute the SQL in the SQL Commander editor. The SQL Commander executes the statements one by one and indicates the progress in the log area. The currently selected Database Connection is used for all statements. The SQL Commander does not support executing SQLs for multiple database connections in one batch.

DbVisualizer uses the delimiters specified in the Tool Properties dialog, in the **SQL Commander/Statement Delimiters** category under the General tab, to separate one statement from the next. Usually semicolon ";" following the actual statement and "go" which should be the only command on a new line directly after the statement that should be executed. The "go" command supports setting the number of times the statement should be executed. For example "go 5" will then execute the statement 5 times.

The result of the execution is displayed in the results area based on the type of results result(s) that are returned. If there are several results and an error occurred in one of them, the Log tab is automatically displayed to indicate the error.

If you select a statement in the SQL editor and choose **SQL Commander->Execute** main menu option, only the selected statement is executed. This is a useful feature when you have several SQL statements in the SQL editor and you just want to execute one or a few of the statements.

Comments in the SQL editor are sent to the database when you use **SQL Commander->Execute**, unless you have enabled **Strip Comments when Executing** in the **SQL Commander->SQL Commander Options** menu.

Execute Only the Current Statement

The **SQL Commander->Execute Current** operation is useful when you have a script with several SQL statements. It lets you execute the statement at the cursor position without first having to select the SQL statement. The default key binding for execute current is **Ctrl-PERIOD** (Ctrl-.).

Execute Current determines the actual statement by parsing the editor buffer using the standard statement delimiters. The current statement is the statement containing the caret or that ends on the line with the caret. This means that the caret may be after the statement delimiter as long as there is no other statement on the same line.

If you are unsure what the boundaries are for the current statement then use **Edit->Select Current Statement**. This will highlight the current statement without executing it.

Execute Buffer

The **SQL Commander->Execute Buffer** sends all of the content in the SQL editor to the database in a single run. This is useful when executing complex SQL statements such as CREATE STORE PROCEDURE (or similar) where statement splitting on semicolon must not be done as with SQL Commander->Execute and SQL Commander->Execute Current.

Control Execution after a Warning or an Error

You can control whether subsequent statements should be executed when a statement results in an error, a warning or returns or affects no rows.

Open **Tools->Tool Properties** and select the **SQL Commander** category under the **General** tab. There you find **Stop on Error**, **Stop on SQL Warning** and **Stop on No Rows** check boxes for enabling these features in all **SQL Commander** tabs.

Alternatively, you can use DbVisualizer client side commands to enable or disable these features in a script.

```
@stop on error;
@stop on sqlwarning;
@stop on norows;

@continue on error;
@continue on sqlwarning;
@continue on norows;
```