Parameterized SQL - Variables and Parameter Markers

A useful feature in the SQL Commander is to use variables or parameter markers in SQL scripts. Variables are used to express that certain parts of the SQL should be replaced with values when the SQL is executed. If you use a script to perform repetitive tasks, such as creating a user and granting permissions, just insert variables for the user name and permissions to grant in the script and DbVisualizer will prompt for the values at execution.

In addition to DbVisualizer's own variable syntax, two parameter marker syntaxes supported natively by some databases/drivers can also be used. This makes it easier to use SQL statements from other tools or code as-is in DbVisualizer, but you need to be aware of the limitations in how they are used compared to DbVisualizer variables.

The following gives an overview of the different formats and how they can be used.

Even if DbVisualizer supports several variable formats it doesn't mean you can always copy/paste the SQLs including the parameter markers to another application and successfully execute it. You need to check the compatibility for the actual connector/driver/framework and even that the database itself supports the used syntax.

The following variable syntaxes are supported by DbVisualizer:

DbVisualizer Variables

`${variable||value||type||options}$`

This is the most flexible syntax as it supports setting a name, default value, data type, and other options. Check the DbVisualizer Variables section for details.

A DbVisualizer variable can be used anywhere in the SQL as the specified value replaces the variable definition as a literal (unless a data type is specified; with a data type, its behavior is exactly the same as for **Named Parameter Markers**).

Example

```
select *
from   EMPLOYEE
where  FIRST_NAME like '${First Name||Phil}$'
and    AGE > ${Age||20}$
```

The variable identifiers, **${...}$** can be modified in **Tools->Tool Properties** and in the **General / Variables** category.

Named Parameter Markers

`&name`
`:name`
`:{name}`
`:'name'`

These syntaxes are supported natively by a few databases. This format allows only a name for the parameter and no other settings, such as type or default value. The parameter name is the name DbVisualizer shows in the prompt window.

Named parameter values are bound at runtime with the markers in the SQL. Some JDBC drivers/databases requires that the proper data type is set while some are more relaxed. For named (and unnamed) parameter markers, you may choose data type in the prompt window.

Named parameter markers should only be used in contexts supported by the actual database, usually for column values. For example, as opposed to a DbVisualizer variable, a parameter marker cannot be used for a table or column name.

The only difference between **&name**, **:name**, **:{name}** and **:'name'** is that the latter two, **:{name}** and **:'name'**, allow white spaces in the name.

Example

```
insert into EMPLOYEE (ID, FIRST_NAME, LAST_NAME, ADDRESS, AGE)
values (null, &FirstName, &LastName, &Address, &Age);

insert into EMPLOYEE (ID, FIRST_NAME, LAST_NAME, ADDRESS, AGE)
values (null, :FirstName, :LastName, :Address, :Age);

insert into EMPLOYEE (ID, FIRST_NAME, LAST_NAME, ADDRESS, AGE)
values (null, :{FirstName}, :{LastName}, :{Address}, :{Age});
```

Read more about named parameter markers.

Unnamed Parameter Markers

?	The question marker symbol is probably the most supported parameter marker among the supported databases. It is also the most unintuitive marker since the user has to remember the order of question marks and the corresponding values.
	
	Since there is no name associated with it, DbVisualizer shows these as **Parameter 1**, **Parameter 2** and so on in the prompt window.
	
	There is no technical difference between how unnamed and named parameter markers are handled internally in DbVisualizer or when processed by the database. All are bound with a prepared SQL statement.
	
	Example
	
	`insert into EMPLOYEE (ID, FIRST_NAME, LAST_NAME, ADDRESS, AGE)`
	`values (null, ?, ?, ?, ?)`
	
	Use named in favor of unnamed parameter markers if there is support in the target database based on the easier reading of named markers. Read more about unnamed parameter markers.

It is not possible to mix DbVisualizer variables and parameter markers, or named and unnamed parameter markers, in the same script. If you do, you will only be prompted for values for one type and the execution will fail.

For more information about the different syntaxes check Using DbVisualizer Variables and Using Parameter markers.