

BPA
Platform

Technical Overview

Database Query (OLEDB) Tool

Copyright

The copyright in this document is owned by Orbis Software Ltd T/A Codeless Platforms 2020. All rights reserved. This publication may not, in whole or part, be reproduced, transmitted, transcribed, stored in a retrieval system or translated into any language in any form or by any means without the prior written consent of Orbis Software Ltd T/A Codeless Platforms.

Head Office:

Codeless Platforms

Suite 1 & 2 Bourne Gate

25 Bourne Valley Road

Poole

BH12 1DY

United Kingdom

Tel: +44 (0) 330 99 88 700

Email: enquiries@codelessplatforms.com

Trademarks

Orbis Software Ltd T/A Codeless Platforms owns the registered trademark "TaskCentre®".

All other Trademarks used are acknowledged as the property of their respective owners.

The information provided in this publication may contain errors, omissions, or typographical errors or may be out of date. Orbis Software Ltd T/A Codeless Platforms may change, delete, or update any published information at any time and without prior notice. The information published in this document is provided for informational purposes only and is not binding on Orbis Software Ltd T/A Codeless Platforms in any way except to the extent that it is specifically indicated to be so.

Contents

Introduction	1
Features	1
Technical Summary	2
Working with other Tools	2
Consuming from Other Tools	2
Exposing to Other Tools	2
Objects Exposed	3
Prerequisites	3
Adding OLEDB Connections	4
Selecting an OLEDB Driver	5
Editing OLEDB Connection Properties	6
General Tab Properties	6
Connection Tab Properties	7
Cursors Tab Properties	8
SQL Tab Properties	9
Step Configuration	9
About the General Tab	10
About the Query Tab	11
Using the Design Interface	12
Using SQL Free Type Mode	16
Testing the SQL Statement	17
About the Results Tab	18
About the Options Tab	19

Introduction

The **Database Query (OLEDB)** tool allows the extraction of specific information sets from relational database systems such as ERP, Financial, Accounting, MRP, Production, CRM, HR, Project, and bespoke systems. Once obtained, the data is then available to other task steps for manipulation or delivery, or both.

The **Database Query (OLEDB)** tool has comprehensive OLEDB support for both modern and legacy platforms.

Features

- ▶ Support for SQL 89 & 92 and variations thereof
- ▶ System, user, and file DSNs
- ▶ Forward-only or scrollable cursors
- ▶ Catalogues and schemas
- ▶ Support for all primary SQL clauses
- ▶ System tables, user tables, views
- ▶ Inner joins, outer joins, formula joins, Cartesian product
- ▶ Formula columns
- ▶ Table aliasing
- ▶ Flexible pre-built date criteria
- ▶ Drag-and-drop visual criteria building with nested parentheses (brackets)
- ▶ Common and custom predicates
- ▶ SQL free-type mode

Technical Summary

Working with other Tools










The **Database Query (OLEDB)** tool can directly interact with the following tools:







Consuming from Other Tools

The **Database Query (OLEDB)** tool does not consume objects outputted by any other tool. Instead, it directly queries the database.

Exposing to Other Tools

The **Database Query (OLEDB)** tool exposes objects which can be directly consumed by the following tools:

Icon	Tool Name	Tool Category
	Convert Recordset to XML	Format
	Create Workflow Job	Format
	Format as Flat File	Format
	Format as HTML	Format
	Format as HTML Pro	Format
	Format as Text	Format
	Run Crystal Report	Format
	Run Microsoft Reporting Services	Format
	Run Microsoft Word (Merge)	Format

Icon	Tool Name	Tool Category
	Transform Data	Format
	Save File	Output
	Transfer File (FTP)	Output
	Call COM Object	Execute
	Run External Program	Execute
	Filter Data	General

Objects Exposed

The **Database Query (OLEDB)** tool exposes the following objects which can be directly consumed by the above tools:

- ▶ **Recordset** — Tabular data from any BPA Platform tool capable of exposing such data (see above)
- ▶ **Step Properties** — Standard step properties are available allowing you to use statistical data of the tool

Prerequisites

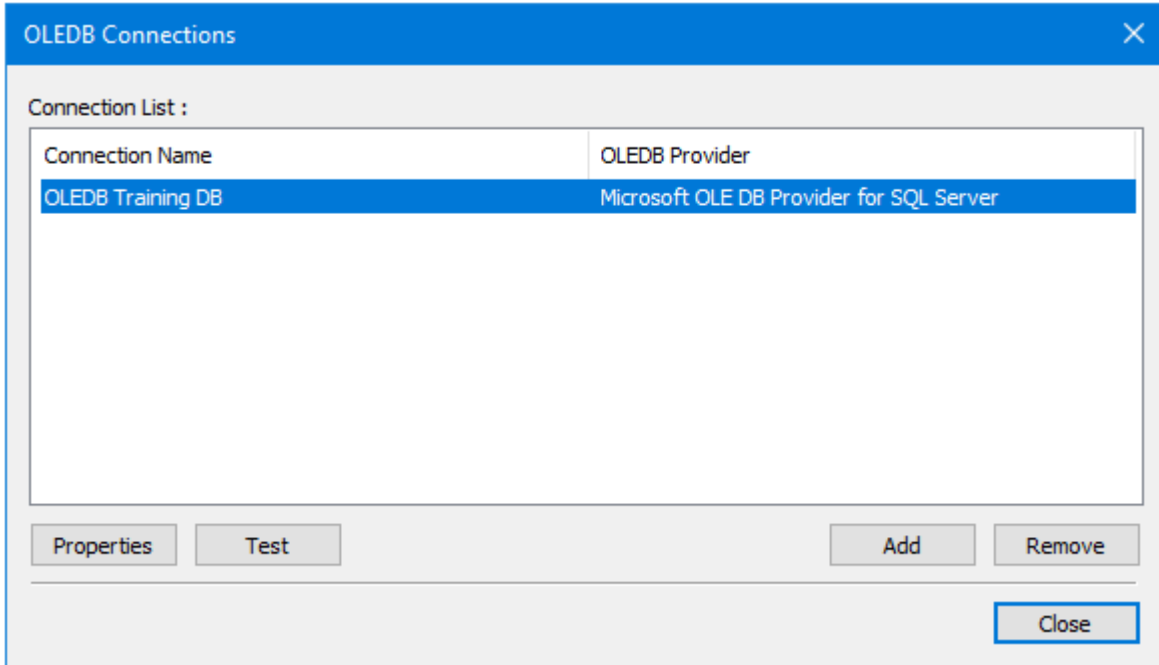
The **Database Query (OLEDB)** tool relies on the following third-party software in order to function correctly:

- ▶ OLEDB Compliant Data Source
- ▶ OLEDB driver

NOTE: At the time of writing, user-defined data types (UDT), available from SQL Server 2008, are not supported.

Adding OLEDB Connections

Before adding the **Database Query (OLEDB)** tool to a task, connections to OLEDB data sources must be configured. Once configured, all connections can be used by any **Database Query (OLEDB)** step.



You open the **OLEDB Connections** dialog box by either:

- ▶ Going to **Manage > Tools >**, then either **Data Connectors > Database Query (OLEDB)** or **Input > Database Query (OLEDB)**
- ▶ From the Task Manager, expanding **BPA Platform > System > Tools > Data Connectors** or **Input**, and double-clicking **Database Query (OLEDB)** in the Items List

NOTE: It doesn't matter which tool category you select the **Database Query (OLEDB)** tool from, the same dialog is opened.

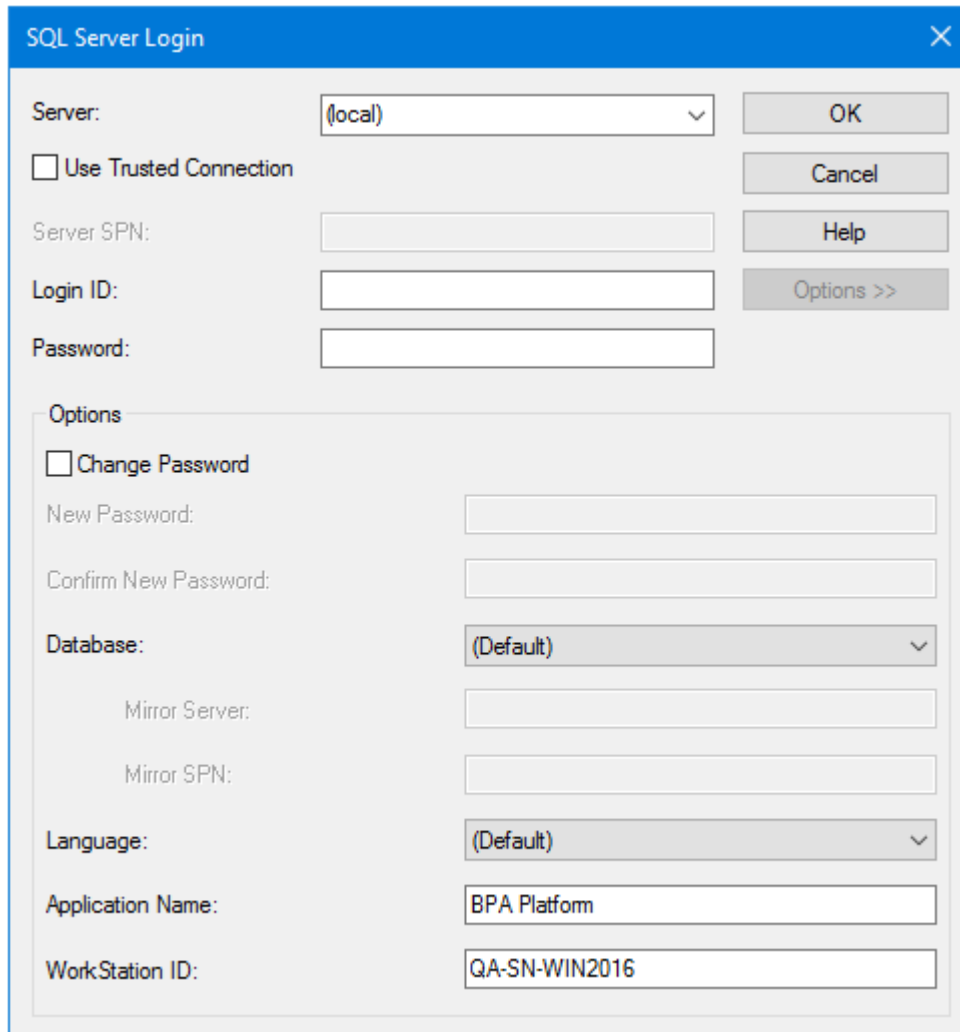
Use **Properties** to edit an existing connection, **Remove** to delete an existing connection, and **Test** to ensure the highlighted connection can be successfully made.

Click **Add** to create a new connection; select the required ODBC driver or data source.

Selecting an OLEDB Driver

NOTE: Any machine hosting a BPA Platform client that runs tasks using this connection, including the BPA Platform server, must have the selected driver installed on the machine. Without the driver, the task will fail.

Once a driver has been selected, add the connection details to the database. The configuration window is different for every driver available; the following example is shown when an SQL Server Native Client driver has been selected:



The image shows a screenshot of the "SQL Server Login" dialog box. The window has a blue title bar with the text "SQL Server Login" and a close button (X) in the top right corner. The main area is light gray and contains several input fields and buttons. At the top right, there are four buttons: "OK", "Cancel", "Help", and "Options >>".

The fields are as follows:

- Server:** A dropdown menu showing "(local)".
- Use Trusted Connection**
- Server SPN:** An empty text box.
- Login ID:** An empty text box.
- Password:** An empty text box.

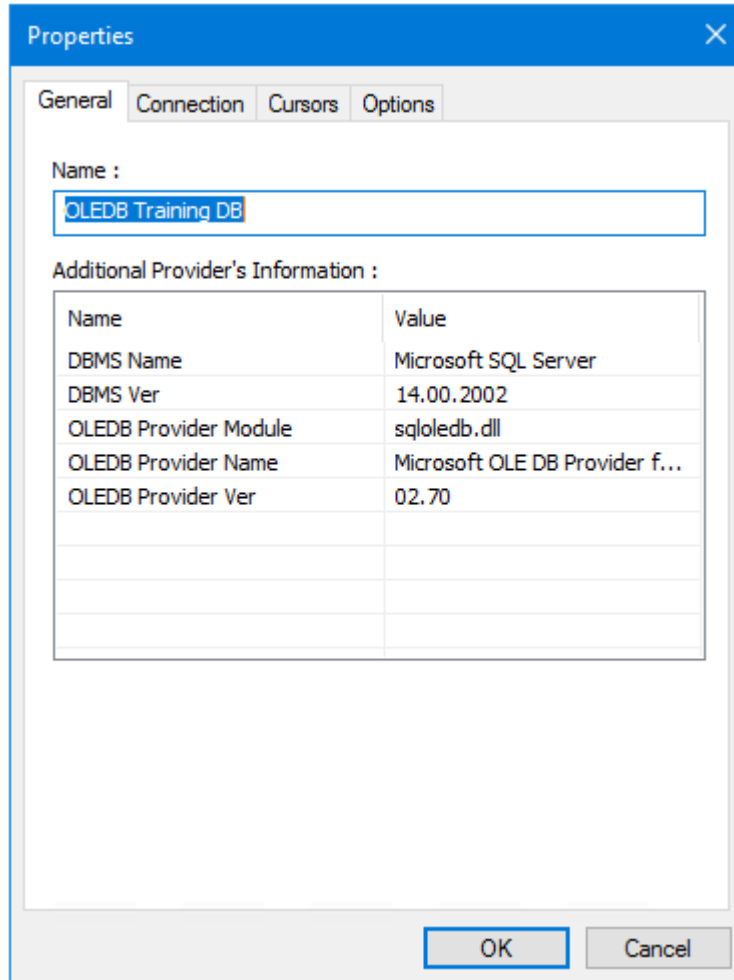
Below these fields is an "Options" section with a title bar and a close button. It contains:

- Change Password**
- New Password:** An empty text box.
- Confirm New Password:** An empty text box.
- Database:** A dropdown menu showing "(Default)".
- Mirror Server:** An empty text box.
- Mirror SPN:** An empty text box.
- Language:** A dropdown menu showing "(Default)".
- Application Name:** A text box containing "BPA Platform".
- WorkStation ID:** A text box containing "QA-SN-WIN2016".

Editing OLEDB Connection Properties

After establishing the basic connection to the relevant OLEDB driver or data source, you can add more advanced configuration, if required. Highlight the relevant connection and click **Properties**.

General Tab Properties

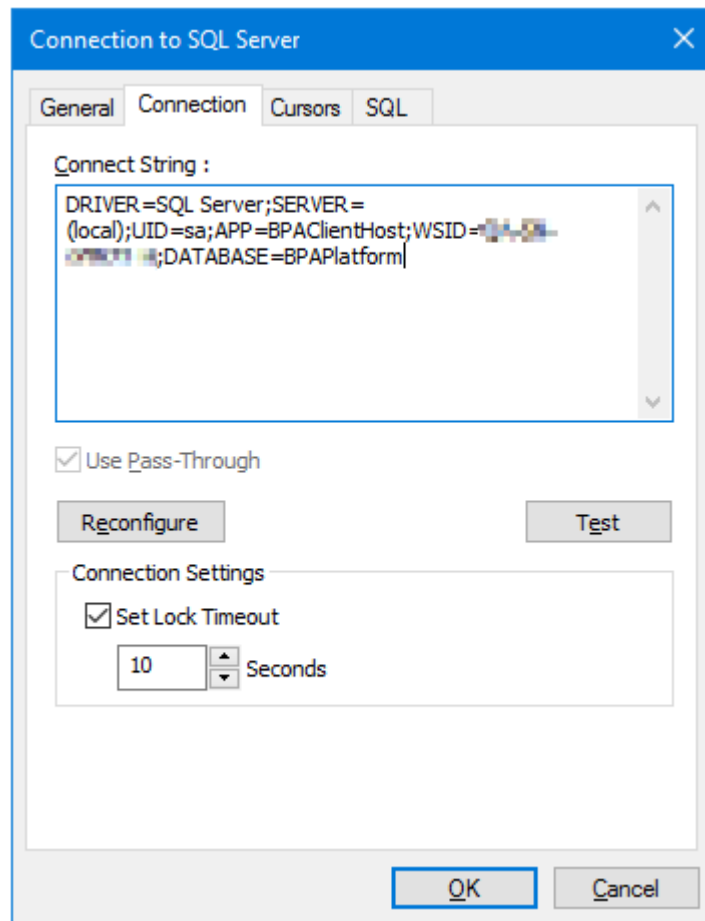


The screenshot shows a 'Properties' dialog box with a blue title bar and a close button. It has four tabs: 'General', 'Connection', 'Cursors', and 'Options'. The 'General' tab is selected. Below the tabs, there is a 'Name:' label followed by a text box containing 'OLEDB Training DB'. Below that is a section titled 'Additional Provider's Information:' containing a table with two columns: 'Name' and 'Value'. The table lists several properties related to the OLEDB provider and the Microsoft SQL Server database engine. At the bottom right of the dialog are 'OK' and 'Cancel' buttons.

Name	Value
DBMS Name	Microsoft SQL Server
DBMS Ver	14.00.2002
OLEDB Provider Module	sqloledb.dll
OLEDB Provider Name	Microsoft OLE DB Provider f...
OLEDB Provider Ver	02.70

- ▶ **Name** — The connection name
- ▶ **Additional Providers Information** — Where available, additional information for the connection is displayed

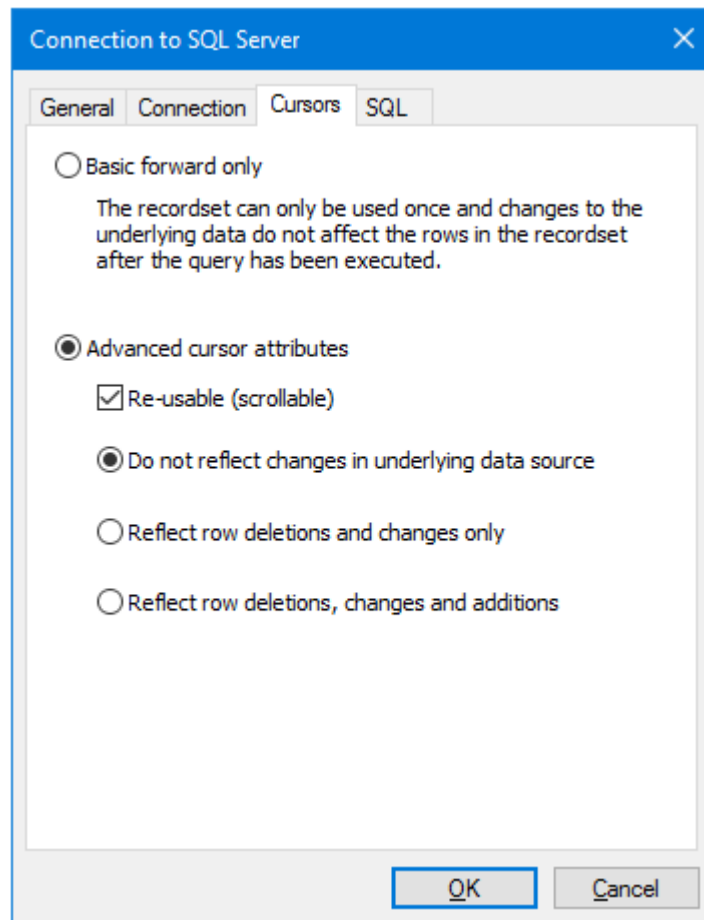
Connection Tab Properties



- ▶ **Connection String** — Contains the full connection string.
- ▶ **Use Pass-Through** — This property is only available with some drivers. You can choose to make the connection directly to the driver (enabled), or use the Microsoft Data Access Objects (DAO) API to make the connection to the driver.
- ▶ **Reconfigure** — If any changes are required, use **Reconfigure** to open the configuration window.
- ▶ **Test** — Tests the connection to the database.
- ▶ **Connection Settings** — If required, you can set a time-out period for connecting to the database in **Seconds**.

Cursors Tab Properties

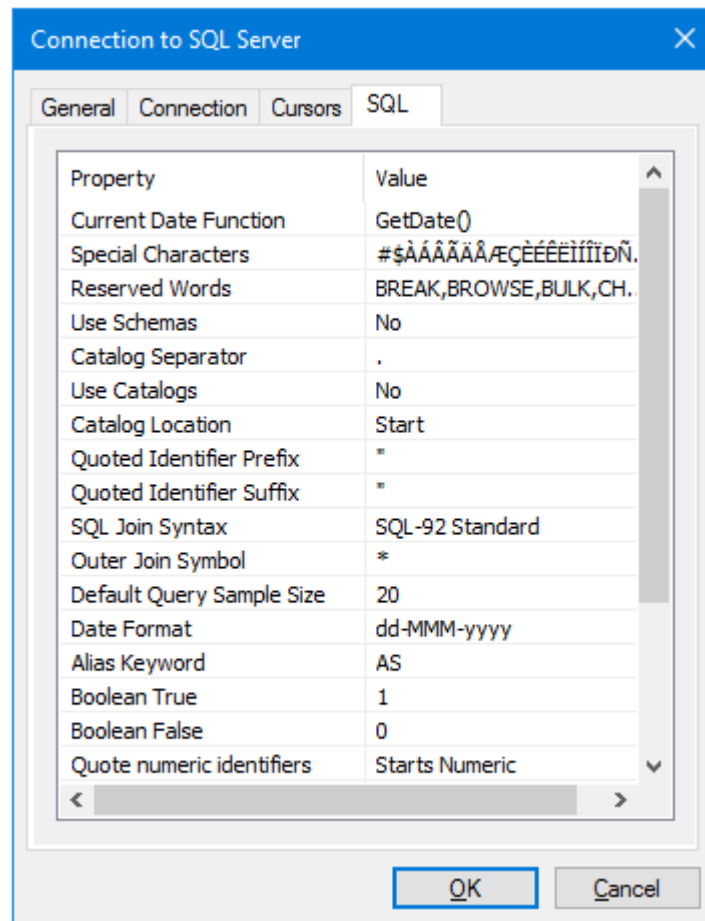
Use the properties in the **Cursors** tab to change the type of database cursor used when executing queries.



- ▶ **Basic forward only** — Use this option where you expect the task to only use the results from the **Database Query (OLEDB)** step once. The result set from the queried rows can only be read top-to-bottom. The cursor cannot be reset so the query cannot be re-used. All OLEDB drivers support this mode.
- ▶ **Advanced cursor attributes**
 - ❑ **Re-usable (scrollable)** — This property allows the cursor to move backwards and forwards through the result set, thus the **Database Query (OLEDB)** output can be consumed by multiple steps in the same task.
 - ❑ **Do not reflect changes in underlying data source** — The rows returned by a query are cached independently at the time the query is executed. When the result set is actually read, any changes made to the source data since the query was executed are not reflected in the result set.
 - ❑ **Reflect row deletions and changes only** — Any changes made to the source data since the query was first executed (updates and deletes to rows in the result set only) are reflected.
 - ❑ **Reflect row deletion, changes and additions** — Any changes made to the source data since the query was first executed (updates, inserts, and deletes) are reflected in the result set.

SQL Tab Properties

The properties available in the **SQL** tab set the supported rules for the selected OLEDB driver for the SQL statements.



For more information, refer to your OLEDB driver documentation.

Step Configuration

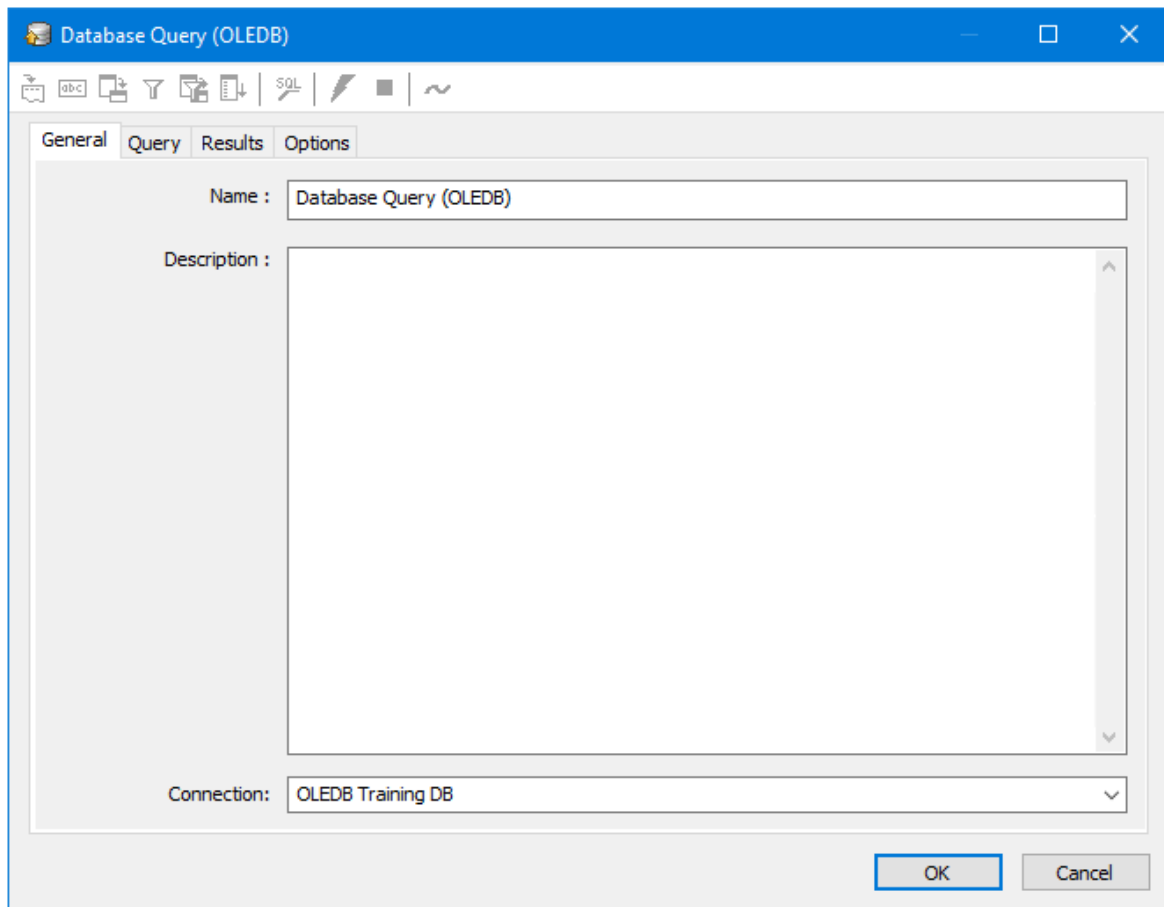
Even though the **Database Query (OLEDB)** tool is found under two different tool categories, they do not affect the functionality of the tool.

To add a new **Database Query (OLEDB)** step to an existing task, you either:

- ▶ Click and drag the **Database Query (OLEDB)** icon from the **Task Browser** to the task **Design** area.
- ▶ From the task's **Design** tab, right-click on empty space and select **Add > then either Data Connectors > Database Query (OLEDB) or Input > Database Query (OLEDB)**.

For a detailed description of how to create new tasks, refer to the product help.

About the General Tab



The **General** tab is used to enter the following details for the step:

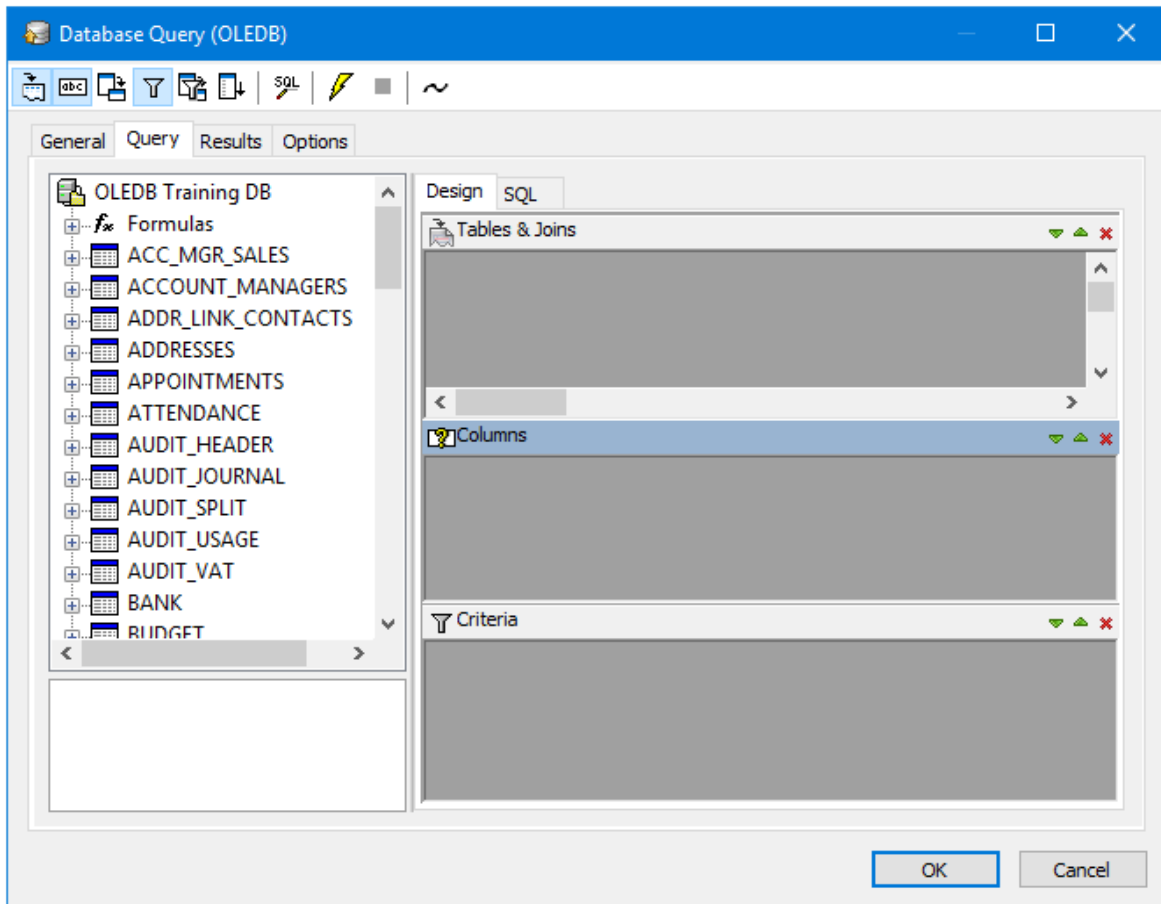
- ▶ **Name** — Enter a meaningful name for the step

TIP: If this task instance makes use of two or more **Database Query (OLEDB)** steps, ensure the **Name** used is unique for each individual step.

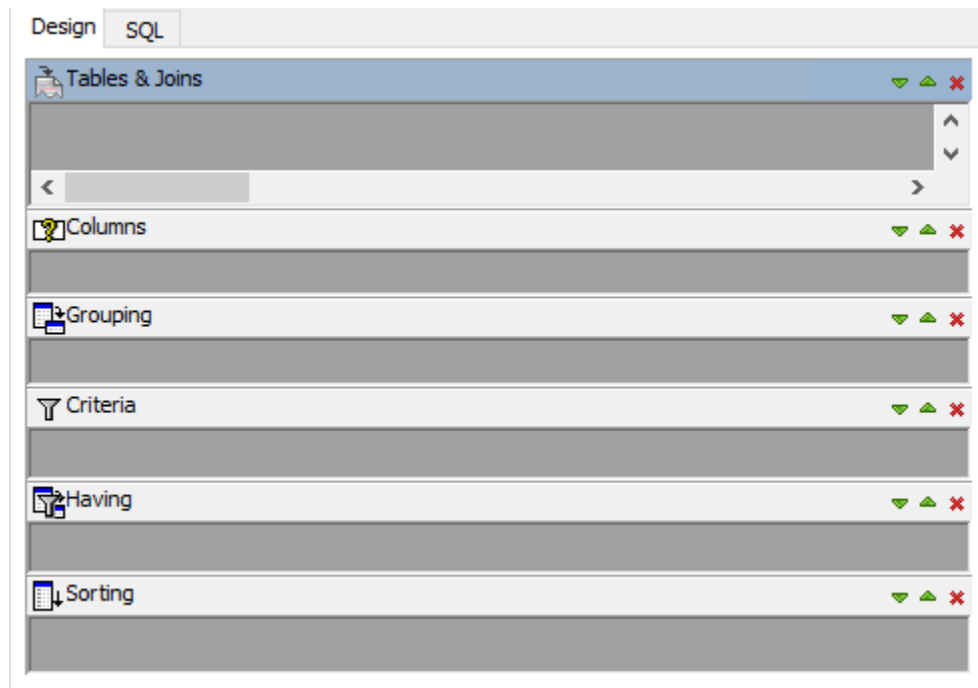
- ▶ **Description** — If required, enter a description of this step
- ▶ **Connection** — You must choose the [OLEDB connection](#) for this task step to use..

About the Query Tab

You use the **Query** tab to build the SQL statement to query the database. You can either use the **Design** interface to build the statement or create it manually in the **SQL** tab.




Using the Design Interface



The **Design** tab allows you to build the SQL statement by dragging-and-dropping database formulas, tables, and table columns from the schema tree on the left:


Tables & Joins

The **Tables & Joins** pane displays the tables within the query and the relationships (joins) between them.

If not shown, click  in the toolbar to display this section.


Columns

The **Columns** pane is used to specify the columns that are returned in the query. Available database columns are displayed in a tree structure to the left of the interface. Formulas can also be created "on-the-fly" to provide manipulation of data whether for formatting or calculations.

If not shown, click  in the toolbar to display this section.


Grouping

The **Grouping** pane is used to create a query that returns consolidated data and is commonly used in conjunction with aggregation functions such as **SUM** or **COUNT**. By selecting certain columns in the **Grouping** pane the data is consolidated so that no duplicates exist across the fields returned. In turn, the aggregate functions allow information such as totals or averages to be calculated across the records that are consolidated. The pane corresponds directly with the **GROUP BY** clause in the SQL syntax.

If not shown, click  in the toolbar to display this section.

Criteria


The **Criteria** pane is used to restrict the rows that are returned by the query — the **WHERE** clause of the SQL statement. The defined criteria are applied before **Grouping**, if used — see above. Multiple criteria can be added and nested to ensure precedence when using a combination of **AND** and **OR** operators.

If not shown, click  in the toolbar to display this section.

Having

The **Having** pane has exactly the same features as the **Criteria** pane. Columns are added in the same way and criteria selected for filtering is identical. The difference between them is the order in which they are processed when the query is run. The **Having** filter is always applied after the **Grouping** filter. For example, a company has offices in different locations:


1. Columns are added in the **Columns** area to provide order and invoice information for all offices.
2. A **Criteria** filter is applied to only include orders received for the current month.
3. A **Grouping** filter is then applied to group all rows by office location.
4. Finally the **Having** filter is applied to only include orders for which an invoice has been raised.

If not shown, click  in the toolbar to display this section.

Sorting

Columns are added to the **Sorting** pane to set the order in which the query results are displayed — the **SORT BY** clause. The arrow to the left of the added column indicates the direction of the sort (ascending (default) or descending). Double-click the arrow to change the direction.

Columns are sorted in the order they are added but you can change this by dragging-and-dropping columns to new positions.

If not shown, click  in the toolbar to display this section.

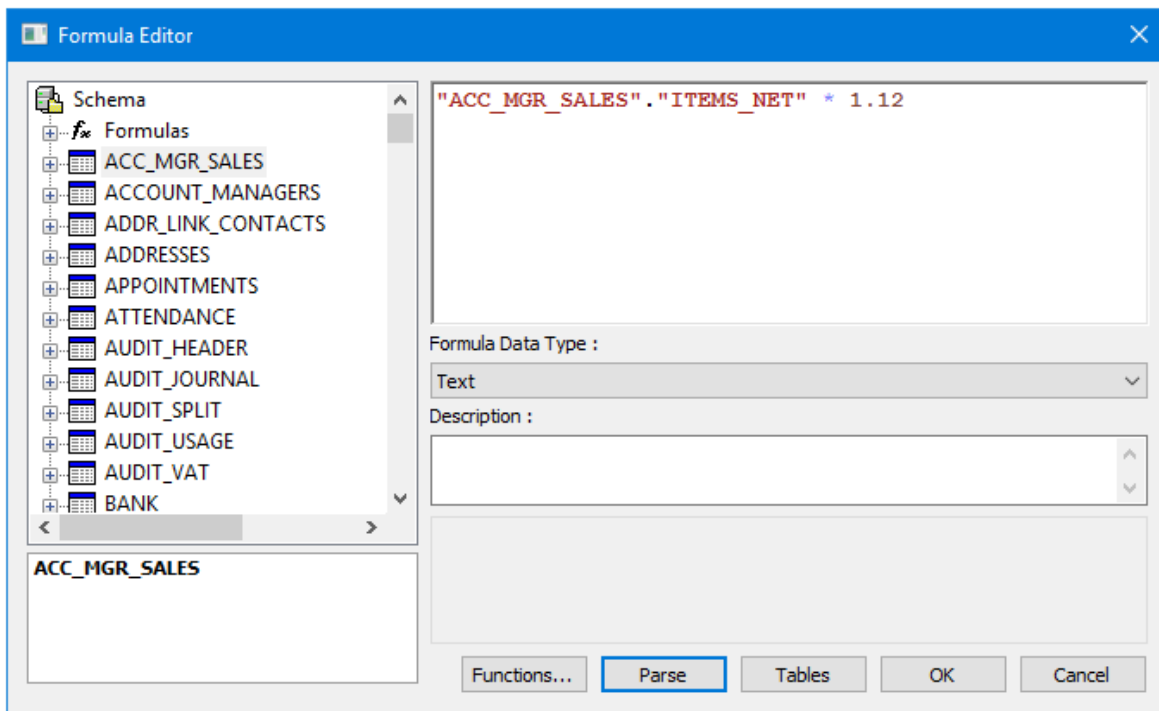
Adding Formulas

Formulas are used to create pseudo-columns only available for use by the task

For example, to display the monetary value of an item in a different currency, you would create a formula to multiple value by a fixed number. You'd then drag the new formula to **Columns** to be included in the query.

To create a formula, either:

- ▶ Right-click on **Formula** in the left-hand tree and select **New Formula**
- ▶ Right-click on a table join in the **Tables & Joins** section and select **Convert to Formula** (recommended when creating formulas based on the result set of the table join)



Drag-and-drop tables or columns from the **Schema** tree to the editing area — when converting a join to a formula, the columns are automatically added to the editing area joined with a =, new formulas have a blank editing area.

Choose the **Formula Data Type** from the drop-down; you must set this if adding columns to the formula which have different data types.


If required, add a short **Description** for the formula.

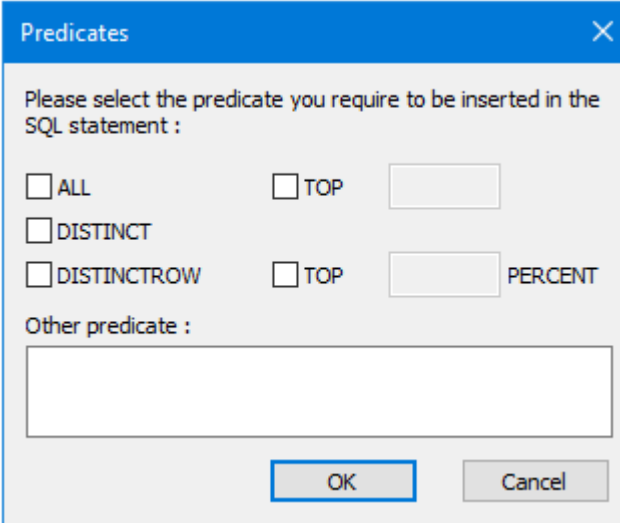
The available **Functions** are dependant on the underlying database engine and the driver in use.

Click **Tables** to ensure the correct tables are linked to the formula.

Click **Parse** to ensure your new formula is correct.

Adding Predicates

SQL predicates are logical conditions applied to clauses, functions, and expressions in the statement. Click  in the toolbar to add predicates:



The image shows a dialog box titled "Predicates" with a close button (X) in the top right corner. The main text inside the dialog says "Please select the predicate you require to be inserted in the SQL statement :". Below this text, there are five options, each with a checkbox and a text input field:

- ALL
- TOP
- DISTINCT
- DISTINCTROW TOP PERCENT
- Other predicate :

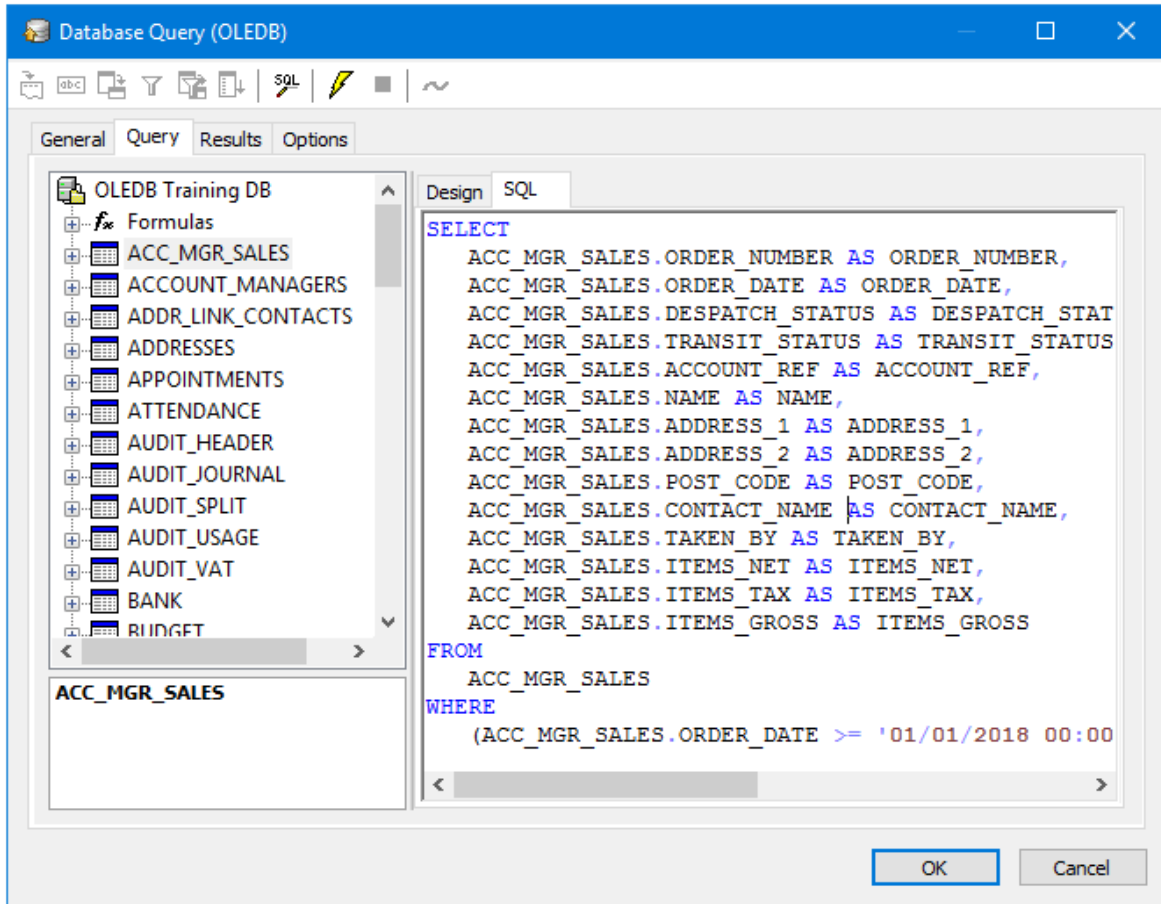
At the bottom of the dialog, there are two buttons: "OK" and "Cancel".


Standard SQL predicates are supported and provision is made to support proprietary predicates available with certain drivers:

- ▶ **ALL** — Return all query results as opposed to other predicates that restrict the query results. This predicate is included largely for completeness as the majority of drivers assume the **ALL** predicate if no other predicate is used.
- ▶ **DISTINCT** — Removes duplicate rows, taking into account only those columns that are requested in the query.
- ▶ **DISTINCTROW** — Removes duplicate rows, taking into account all columns in the source for the tables requested in the query, irrespective of the columns that are actually requested in the query.
- ▶ **TOP** — Return the number of rows specified, starting from the top of the result set. The sorting of the query affects which rows are returned.
- ▶ **TOP (PERCENT)** — Return the percentage of rows specified, starting from the top of the result set. The sorting of the query affects which rows are returned.
- ▶ **Other predicate** — Allows another predicate statement to be manually entered. Such statements are inserted after any standard predicates selected.

Using SQL Free Type Mode

The SQL tab displays the script version of the SQL statement being build in the **Design** tab:



If required, you can edit the existing script or create it from scratch using the "SQL free type mode" feature — click  in the toolbar.

NOTE: Once you change to using SQL free type mode, you cannot revert back to using the SQL designer for this task step.

The **Database Query (OLEDB)** tool can detect whether its driver can support a query language, no query language, or both. If a query language is supported, **Execute Command** is enabled which executes the whole query script entered. If the driver cannot support a query language, **Open Table / Object** is enabled which only opens the tables and objects stated in the script.

SQL

Select Query Type :



Execute Command

Open Table / Object

```
SELECT
ACC_MGR_SALES.ORDER_NUMBER AS ORDER_NUMBER,
ACC_MGR_SALES.ORDER_DATE AS ORDER_DATE,
ACC_MGR_SALES.DESPATCH_STATUS AS DESPATCH_STA
ACC_MGR_SALES.TRANSIT_STATUS AS TRANSIT_STATU
ACC_MGR_SALES.ACCOUNT_REF AS ACCOUNT_REF,
ACC_MGR_SALES.NAME AS NAME,
ACC_MGR_SALES.ADDRESS_1 AS ADDRESS_1,
ACC_MGR_SALES.ADDRESS_2 AS ADDRESS_2,
ACC_MGR_SALES.POST_CODE AS POST_CODE,
ACC_MGR_SALES.CONTACT_NAME AS CONTACT_NAME,
ACC_MGR_SALES.TAKEN_BY AS TAKEN_BY,
ACC_MGR_SALES.ITEMS_NET AS ITEMS_NET,
ACC_MGR_SALES.ITEMS_TAX AS ITEMS_TAX,
ACC_MGR_SALES.ITEMS_GROSS AS ITEMS_GROSS
FROM
```

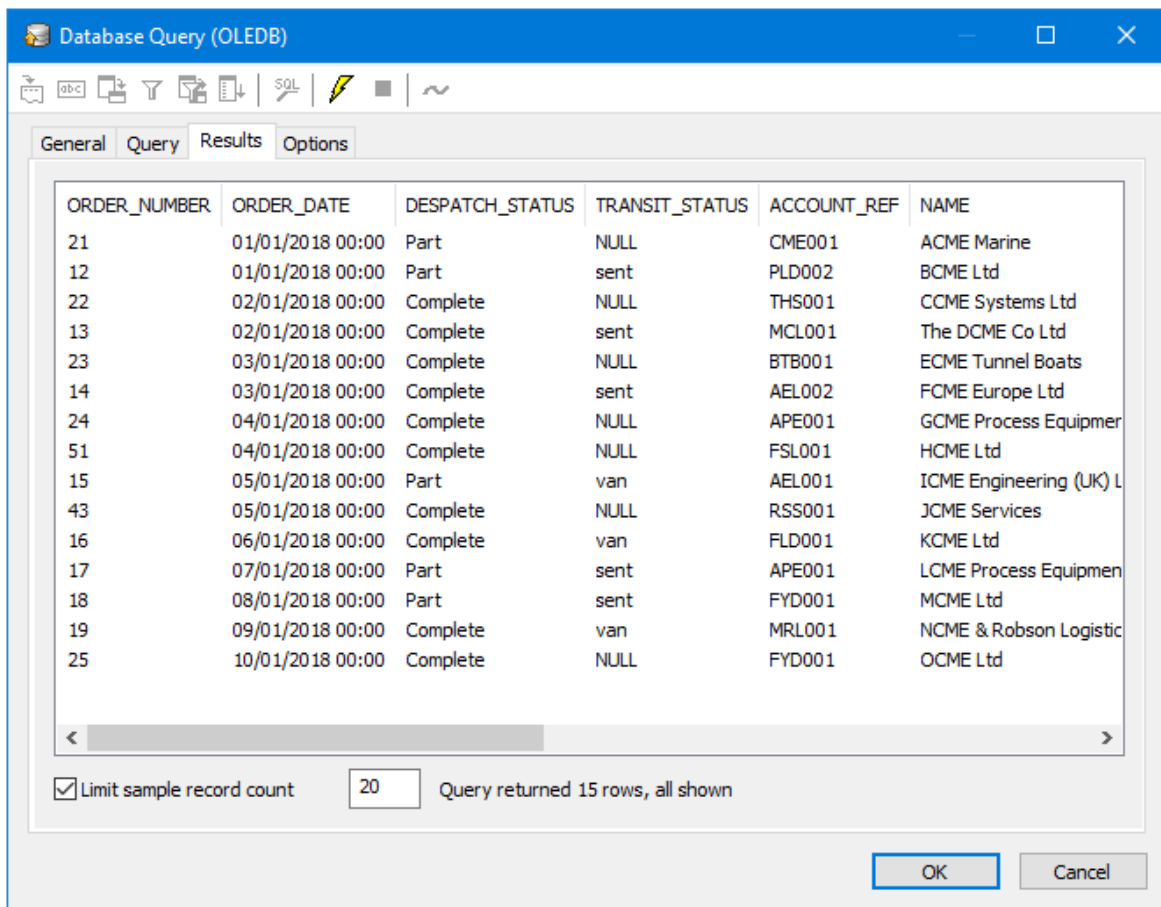
TIP: Syntax highlighting in free type mode is governed by the OLEDB driver in use, namely the **Quoted Identifier Prefix** and **Quote Identifier Suffix** properties in the global connection's **Options** tab. The **Database Query (OLEDB)** tool does not add identifiers for string values therefore these must be added in manually — refer to the two properties for the correct identifiers to use.

Testing the SQL Statement

From the toolbar, click  to test the script. Results are shown in the [Results tab](#). If required, click  to cancel the current test being run.

About the Results Tab

The **Results** tab displays the results of the SQL statement built in the **Query** tab — click  first.



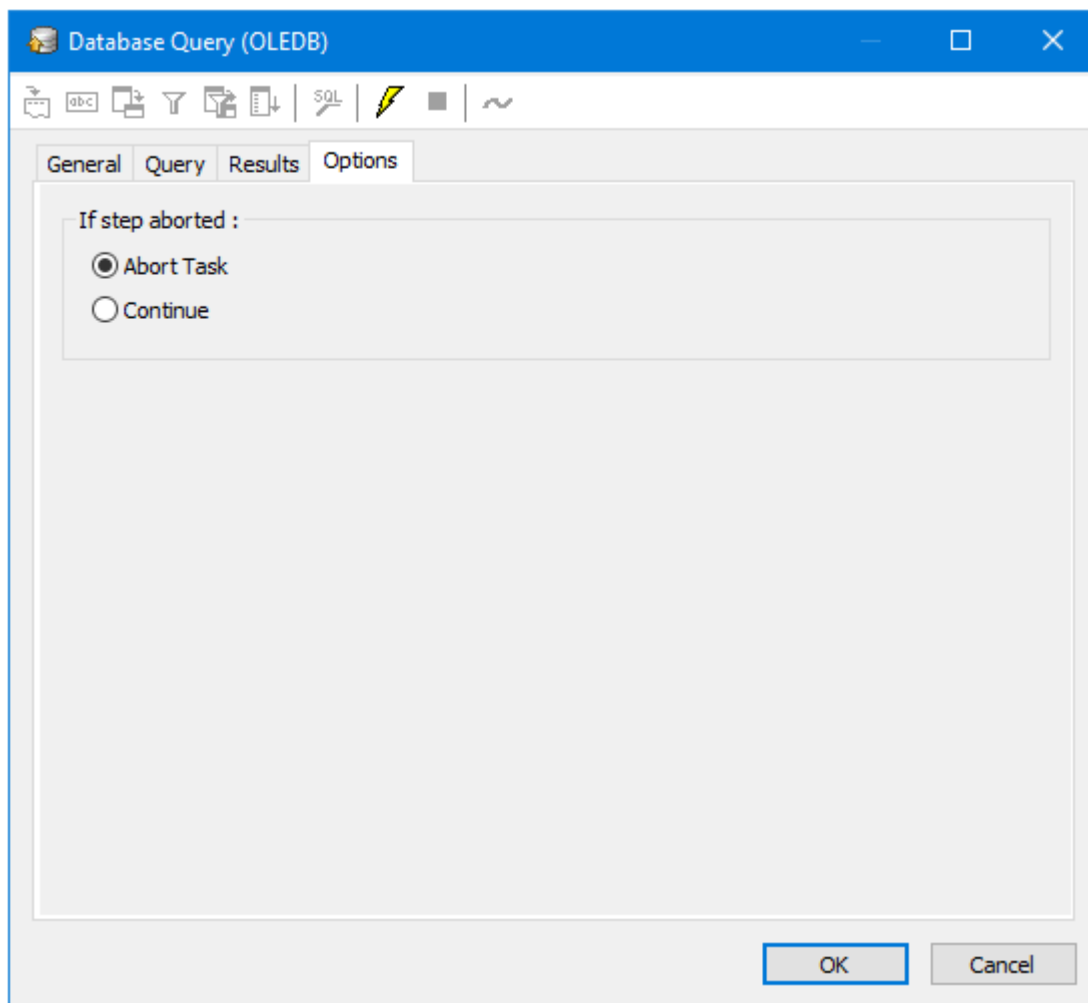
If required, to speed up the test you can **Limit [the] sample record count** to return a reduced number of rows.

NOTE: Binary information cannot be displayed effectively in this view. Such values are displayed as **Binary**.

NOTE: Not all drivers can return the total number of rows returned by the query so this statement may be missing.

About the Options Tab

The **Options** tab allows you to define how errors in this step are handled at task runtime.



If the step is aborted, you can choose to **Continue** processing onto the next step in the task, or terminate the whole task immediately (**Abort Task**).

Want to learn more?

Discover how Codeless Platforms can help your business by improving performance, boosting efficiency and cutting costs.



+44 (0) 330 99 88 700



enquiries@codelessplatforms.com



www.codelessplatforms.com

