DYNISTICS

ACTIVE DASHBOARDS SOFTWARE
QUICK QUERY GUIDE v6.2

Active Dashboards Quick Query Guide

Date of issue	Brief Description
May 2017	Version 6.2

Comments and suggested improvements to this publication are welcome. Please address comments to:

Product Development
Dynistics Limited
Blythe Valley Innovation Centre
Blythe Valley Park
Solihull
B90 8AJ
United Kingdom

Or via e-mail to:

development@dynistics.com

About This Guide

Purpose

This guide describes how to configure and use the Quick Query feature of Active Dashboards.

Readership

The guide is intended for both end-users and designers of the Active Dashboards application.

This guide can be supplied in electronic format to enable organisations to customise its content to meet their specific operational requirements.

Recommended Use

Chapter 1 provides a brief overview of Quick Query - you may find it helpful to read this first.

Chapter 2 is for Dashboard users and administrators and describes how to setup access to Quick Query.

Chapter 3 is for Dashboard users who will use the Quick Query feature.

Chapter 4 is for Dashboard Designer users and describes how to setup Data Models that can be queried.

Other Recommended Reading

- Active Dashboards - Dashboard Designer Guide
- Active Dashboards - Dashboard User Guide

Assumptions

The guide assumes the user is familiar with operating a keyboard and mouse and with the standard features of standard operating systems and web browser software.

This guide assumes Dashboard Designer users are familiar with the concept of *Data Connectors* as described in the Dashboard Designer Guide.

Acknowledgements

All trademarks are the property of their respective owners.

Contents

Chapter 1

Introduction

Overview

Active Dashboards Quick Query provides Dashboard users with a self-service facility for querying data relevant to their role. It can be used as an alternative to, or to compliment, existing pre-defined Dashboards.

How it works

1. **Quick Query Data Models**

   Dashboard Designer users create Data Models, each describing a set of related data that is used to provide a user friendly view of the data to Quick Query users. See *Chapter 4* for details.

2. **Quick Query Access**

   Dashboard Web Application administrator users manage which groups of Dashboard users have access to Quick Query. This involves choosing whether Queries can be viewed, added, edited and deleted for each Data Model defined. See *Chapter 2* for details.

3. **Quick Query Page**

   Dashboard Web Application Users access their Dashboards in the usual way and then launch Quick Query which opens in a new browser page/tab. Each Data Model the user has access to is listed and the user can run/edit/copy/delete existing Queries or add new Queries - depending on the access granted to them. See *Chapter 3* for details.
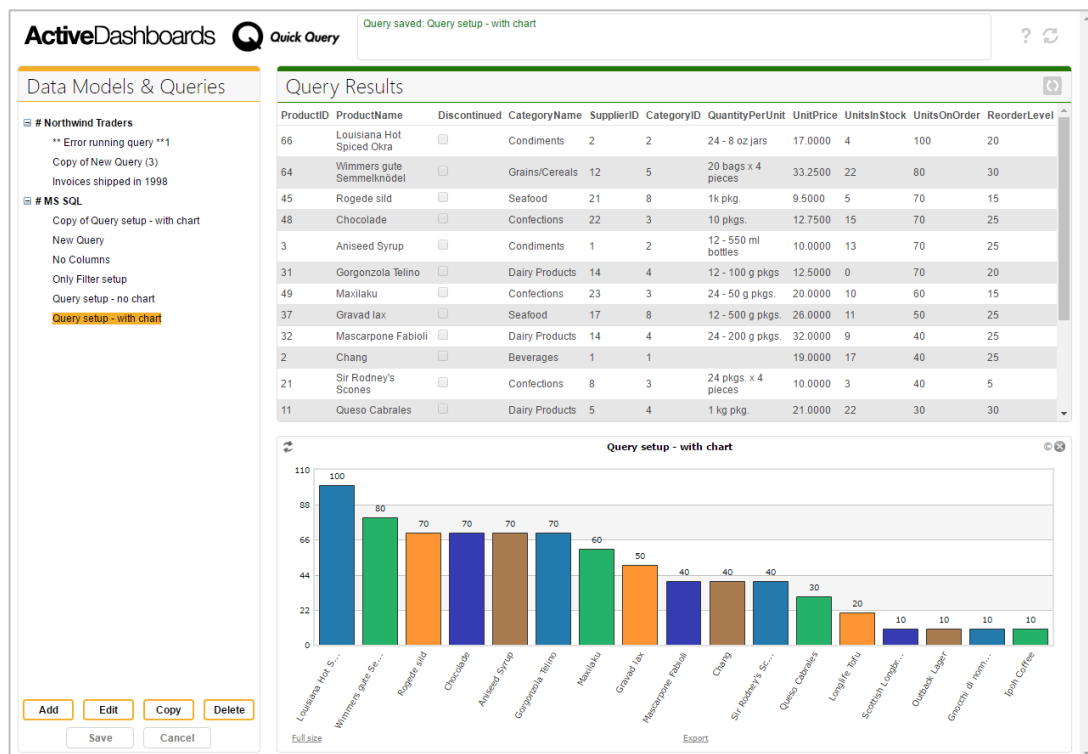

Figure 1: Quick Query Page - listing available Data Models & Queries (left) and the selected Query result (right)

Chapter 2

Controlling User Access

Overview

Access to the Quick Query page is configurable and different groups of Dashboard users can be assigned full, partial or no access to it.

For each Quick Query Data Model defined a user may have the following rights to query it:

- **No access**

  The Data Model and any associated Queries will not be displayed.

- **Read-only access**

  The Data Model and any associated Queries will be viewable only.

  Queries can be run and results viewed but Queries cannot be created, modified or deleted.

- **Add/Edit/Delete access**

  The Data Model and any associated Queries will be viewable, new Queries can be created and existing ones modified or deleted. Different combinations of these are possible, e.g. being able to add/edit Queries but not being allowed to delete them.

Configuring Access

Refer to the *Dashboard User Guide - Dashboard Administration* chapter for details of how to manage Quick Query access rights using the Dashboard Web Application Administration page.

Examples

With three Quick Query Data Models defined:

- *Customer data*
- *HR data*
- *Sales data*

Example 1

A Dashboard user with full Quick Query access will see all three Data Models and can view/add/edit/delete Queries in each one:
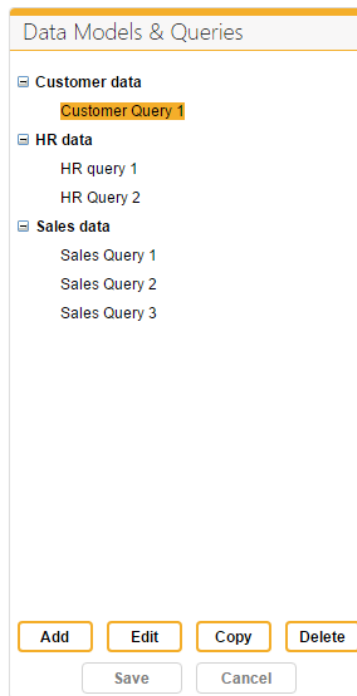
Figure 2: Quick Query Page – full access to all 3 Data Models and their Queries

Example 2

A Dashboard user with no access to *Customer data*, full access to the *HR data* and read-only access to *Sales data*:
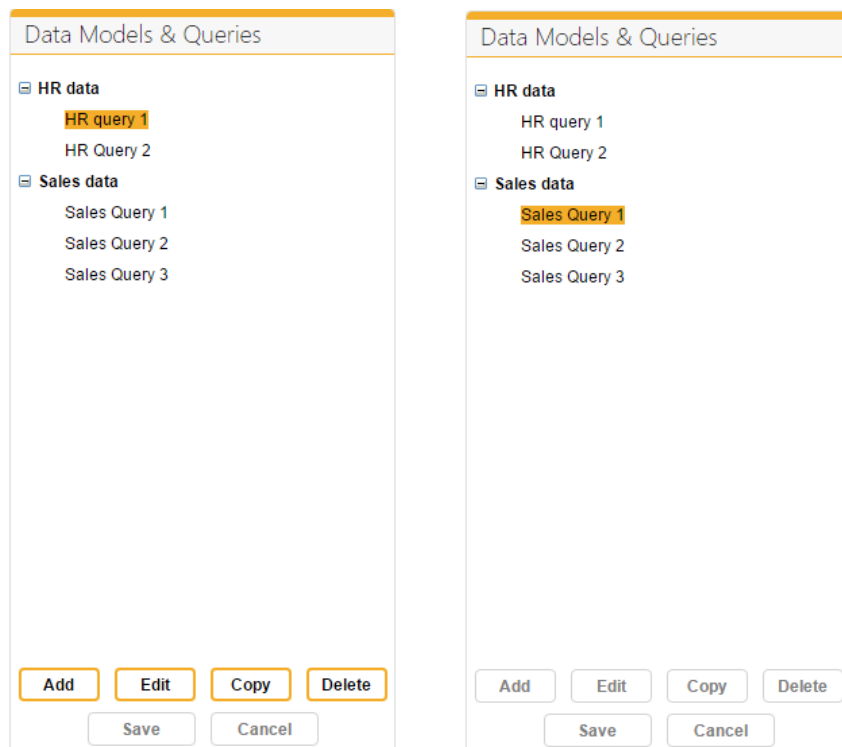

Figure 3: Quick Query Page – full access to *HR data* (left) and read-only access to *Sales data* (right)

Note: the *Add*, *Edit*, *Copy*, *Delete*, *Save* and *Cancel* buttons are only enabled when the user has access to these actions and if they are relevant to the current action being performed on the page.

Chapter 3

Quick Query Page

Overview

The Quick Query page is displayed in your browser alongside your Dashboards.

Different users may have different views of the page depending on the access granted to them.

Opening

1. Access the Dashboards in your browser in the usual way.

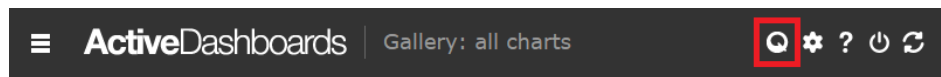2. Click the *Quick Query* link shown on the Dashboard header.

Figure 4: Dashboard page header with Quick Query icon highlighted

Depending on you Dashboard configuration this link may be an icon or text link.

3. The Quick Query page will open and display in a new browser tab or window depending on your browser's configuration. You can at any time switch back to the Dashboard page in your browser.

Note: This Quick Query link will only display if you have access rights to it. See *Chapter 2* for details of setting up user access.

Description

Each area of the Quick Query page is described below, with the relevant area being discussed highlighted.
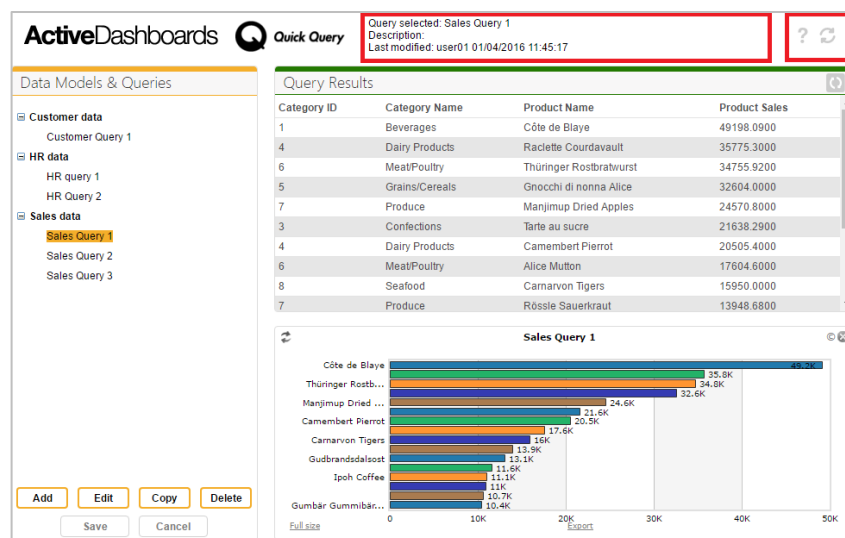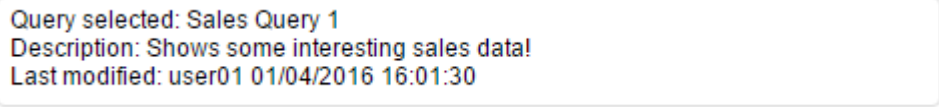
Message area / Page options

Figure 5: Quick Query Page - Message area (left) and Page options (right)

Message area

Information relevant to the last action you performed on the page are displayed in this area. For example:

Query selected: Sales Query 1
Description: Shows some interesting sales data!
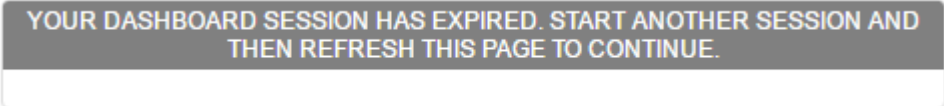Last modified: user01 01/04/2016 16:01:30

Figure 6: Message showing details of the Query selected including who last modified it

Query saved: Sales Query 1

Figure 7: Message showing a Query was successfully saved after editing
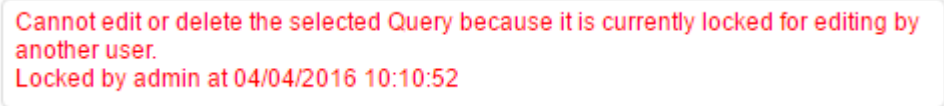
If your dashboard session expires then the following message is displayed. You can continue where you left of on the Quick Query page after starting another session, e.g. by returning to the Dashboard page and logging-in again.

YOUR DASHBOARD SESSION HAS EXPIRED. START ANOTHER SESSION AND
THEN REFRESH THIS PAGE TO CONTINUE.

Figure 8: Session expiry message

If an action cannot be performed then details are displayed in red text.

Cannot edit or delete the selected Query because it is currently locked for editing by
another user.
Locked by admin at 04/04/2016 10:10:52

Figure 9: Message detailing why an action cannot be performed

Page options

?   View help - click to open the Dashboard Quick Query Guide.

⟳   Refresh page - click to refresh the whole Quick Query page and rerun any loaded Query.

Data Model & Queries

Data Models you have access to along with any pre-defined Queries for each one are listed in this area.
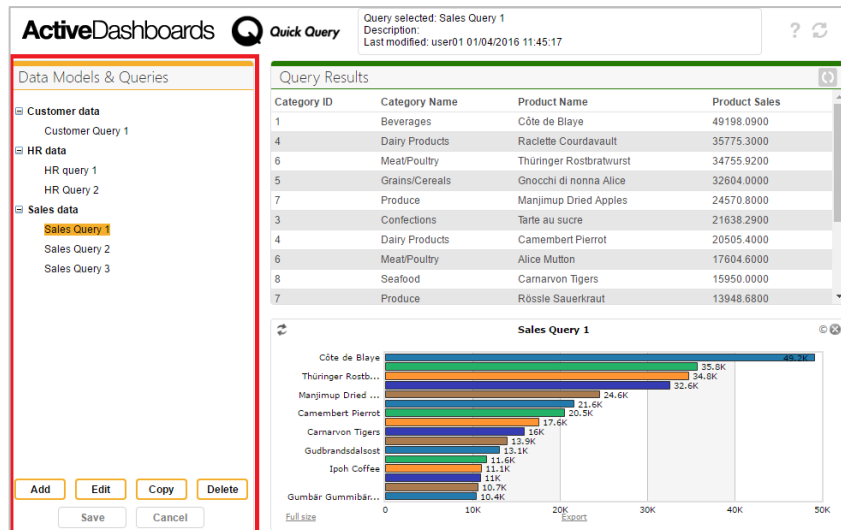
Figure 10: Quick Query Page – Data Models & Queries

You can select a Data Model or Query by clicking on it. The page will then update to show information relevant to the selected Model/Query.
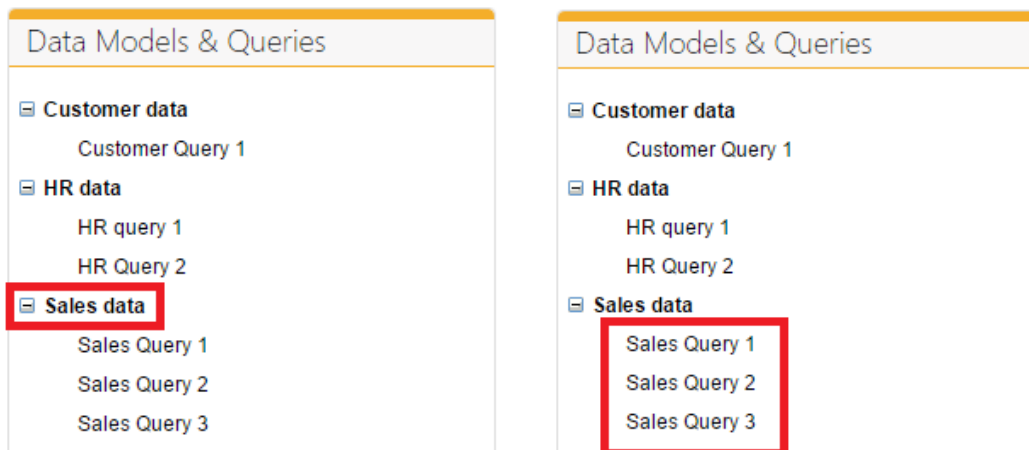

Figure 11: Data Model highlighted (left) and its Queries highlighted (right)

Action buttons relevant to the selected Data Model or Query are shown. When these buttons are enabled you can carry out the relevant action on the selected Data Model or Query.
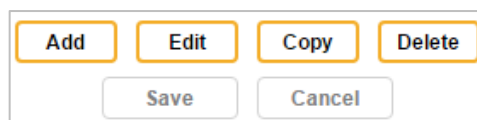

Figure 12: Action buttons – enabled (top row) and disabled (bottom row)

- **Add** – Creates a new Query for the selected Data Model. The new Query is only visible to other users after you save it.

- **Edit** – Reserves or "locks" the selected Query for you to edit. Any changes made are only visible to other users after you save it.

- **Copy** – Creates a new Query based on the selected Query. The new Query is only visible to other users after you save it.

- **Delete** – Permanently deletes the selected Query.

- **Save** – Saves changes made during Adding/Editing/Copying.

- **Cancel** – Cancel any Adding/Editing/Copying – any changes made whilst editing will be lost.

Locking of Queries

When you start editing an existing Query then it is "locked" so other Dashboard users cannot edit or delete the Query whilst you are working on it. Locked Queries are indicated by a padlock icon displayed next to them.
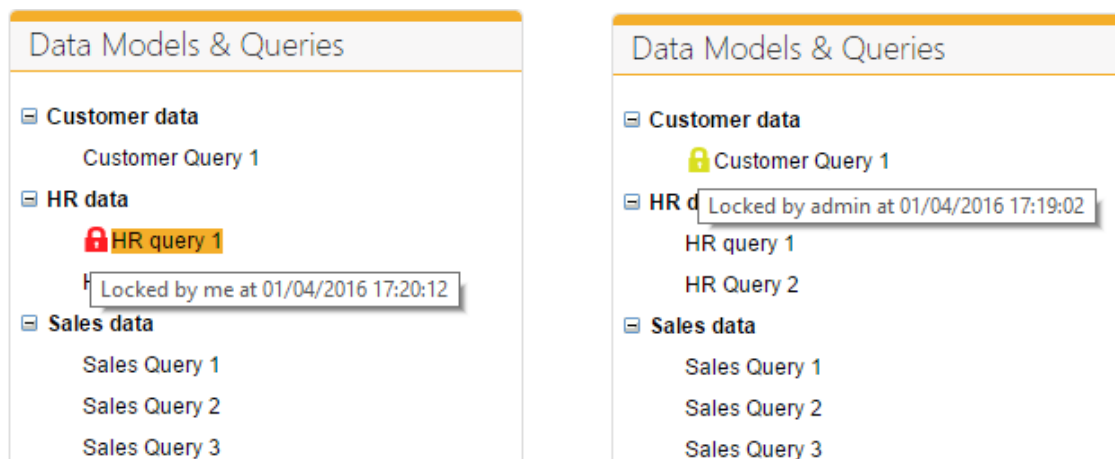

Figure 13: Locked Queries – locked by you (left) and locked by another user (right)

Unlocking of Queries

When you Save or Cancel a Query that you are editing then the lock is removed, freeing it up for other users to edit.

If you exit Quick Query whilst editing a Query then it will remain locked to you. You can (depending on how you logged onto the Dashboards) re-edit the Query and Save or Cancel to release the lock.

Note: Quick Query locks can also be released by Dashboard Designer users using the Designer *Lock Maintenance* facility.

Query Results

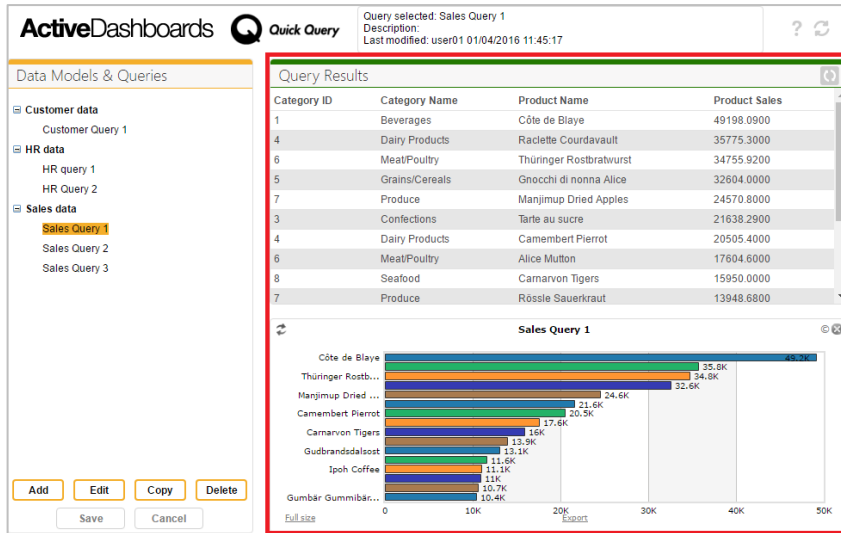Selecting a Query will run it and display the results in this area.

Figure 14: Quick Query Page – Query Results

Depending on how the Query has be defined, the results are displayed in a table with an optional chart displaying the same results underneath.

To re-run the Query and refresh the results either use the Refresh page button, reselect the Query in the Data Models & Queries area or use the following buttons:

     Refresh the tabular results

     Refresh the chart results

If no Query is selected then the Query Results area will be blank.

Figure 15: Results when no Query selected

If running the Query results in no data being returned then this is displayed as follows:

Figure 16: Results when Query returns no data

If the Query cannot be run for any reason then details why are displayed in the results table. This usually indicates that your system administrator needs to rectify an error connecting to the data or a Dashboard Designer user needs to correct the relevant Quick Query Data Model.
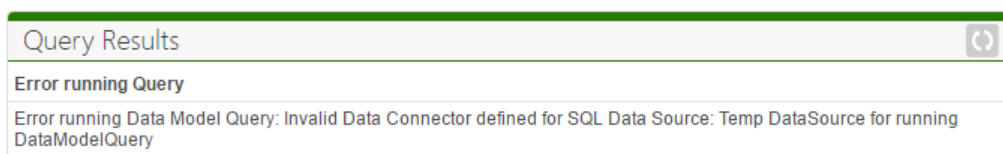

Figure 17: Example of results when error running Query

12

Query Builder

When you add, edit or copy a Query then the Query Builder is displayed. This allows you to setup the Query based on the chosen Data Model.
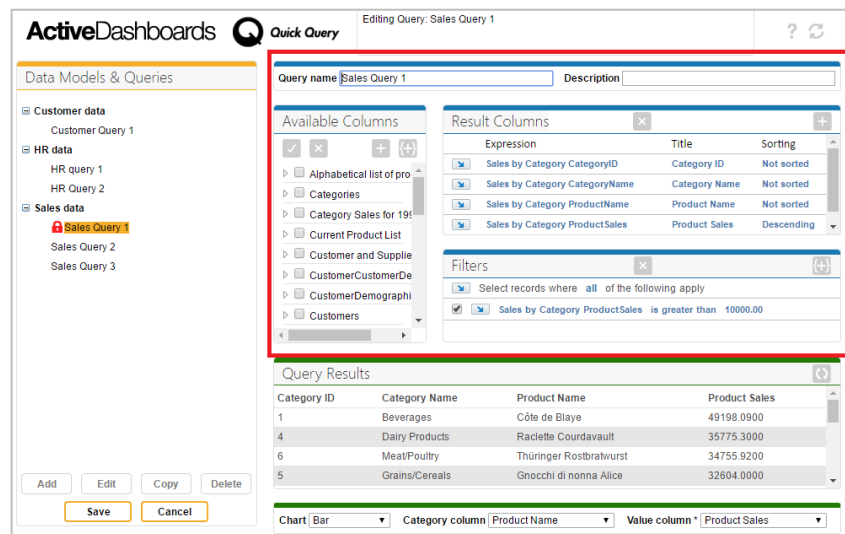
Figure 18: Quick Query Page – Query Builder

The Query Builder is divided into 4 areas:

- **Query Name and Description** – enter the name of the Query and an optional Description.
- **Available Columns** – Choose multiple columns to query or filter by.
- **Result Columns** – Lists the columns that will be shown in the Query result.
- **Filters** – Optional filters to restrict the data returned in the Query result.
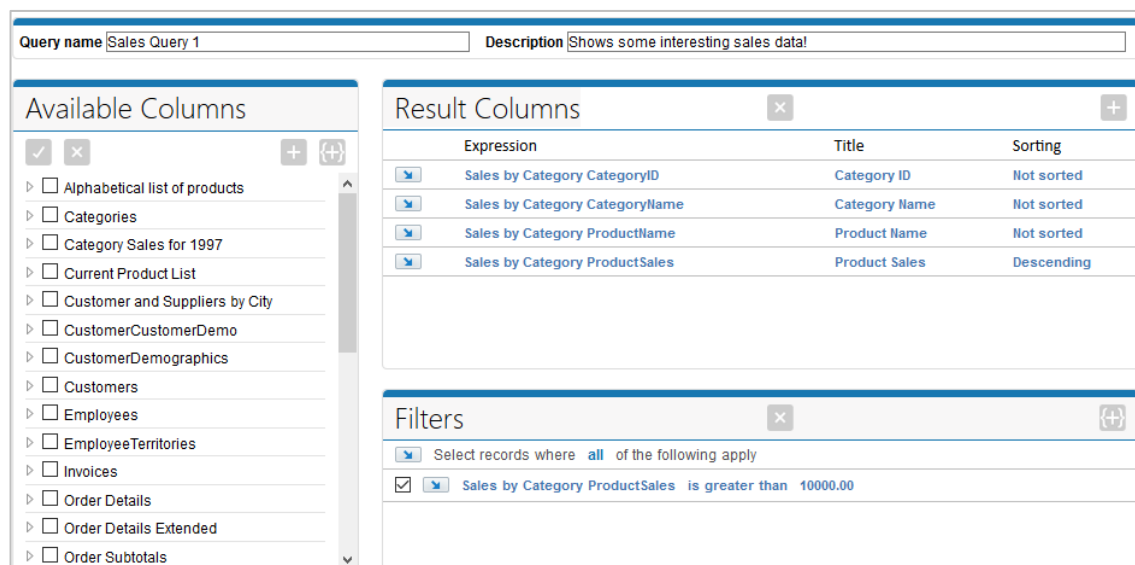

Figure 19: Query Builder

Query Builder - Name and Description

Enter a name and optional description to describe the Query. Names don't have to be unique but it worth carefully naming Queries to avoid potential confusion.

Query name	Sales Query 1	Description	Shows some interesting sales data!

Figure 20: Query Builder - Name and Description

Query Builder - Available Columns

This lists all the Data Model's tables and columns available for querying. It is a convenient way of adding multiple columns to the Query in one go.
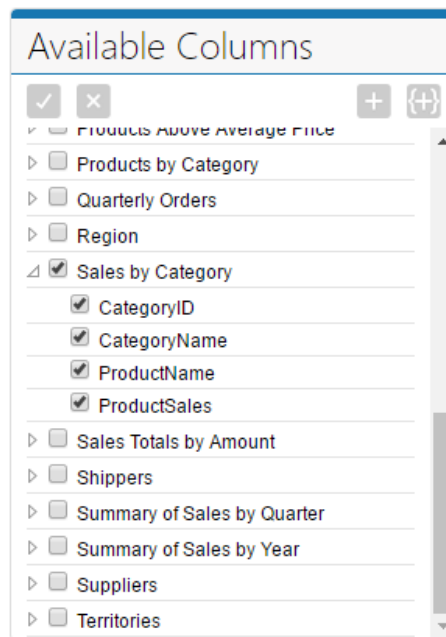
Figure 21: Query Builder Available Columns – with one Table expanded showing its Columns

To include columns in the Query, select the required columns by ticking the boxes next to them and use the following buttons:

Selects all listed tables and columns.

Deselects all listed tables and columns.

Add the selected data columns to the Query Results list – if not already present.

Add the selected data columns to the Filters list – if not already present.

Expand/collapse a data table to show/hide its columns.

Note: The *Available Columns* area is only used to select data columns, the ticked columns don't necessarily reflect the Columns currently listed in the Result Columns or Filters area.

Note: The Data Model may restrict which columns can be used in the *Result Columns* or *Filters* area, in which case trying to add columns these will have no effect.

Query Builder - Result Columns

This lists all the Data Model columns to include in the Query Results.

Figure 22: Query Builder Result Columns – listing 4 columns

Use the following buttons to manage the list:

☒	Clears the columns currently listed.
➕	Add a single column to the list.
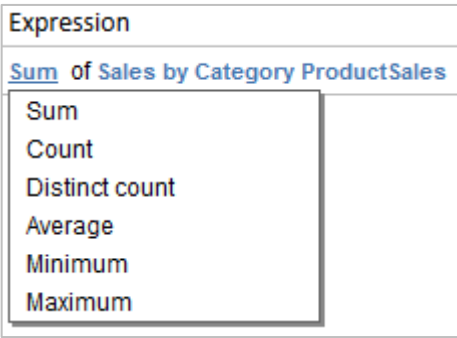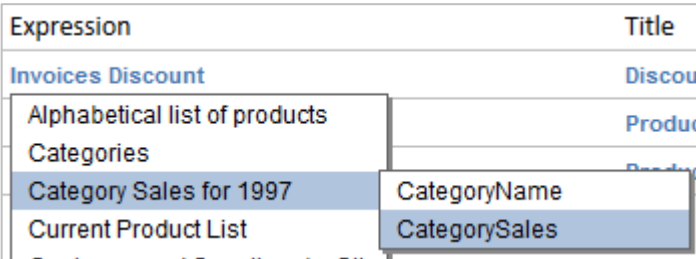
Each line in the list represents one column:
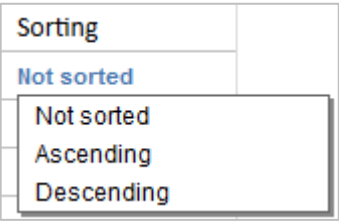
Figure 23: Query Builder Result Columns – a single column

Each element is described below

⬇	Select to display a list of options relevant to the column: Sorting Move to top Move up Move down Move to bottom Delete column Simple column Aggregate function **Sorting** An alternative way of changing the sorting order of this column – see *Sorting* below.

	Move to top, **Move up**, **Move down**, **Move to bottom**
	These change the display order of the column in the Query Results. Columns listed top to bottom are outputted left to right in the Query Results.
	Delete column
	Removes the column from the Result Columns list.
	Simple column
	Output the data values from this column of data in the Query Results (this is the default behaviour).
	Aggregate column
	Output a calculated value based on the data values held in this column of data in the Query Results.
	When this is selected an extra option appear in the column's *Expression* showing the aggregate calculation type, in this case "Sum". Select this to see other aggregation options available:
	Choose the aggregation required from the list.
	Note: The list of available aggregations varies depending on the type of data held in the column – as defined in the Data Model.
Expression	Displays the table/column chosen.
	Select the *Expression* to choose an alternative column:
Title	The display name of the column to use in the Query Results.
	Select the *Title* to edit it:
Sorting	How the data returned in this columns in the Query Results affects the ordering of the results.

Select to change the sorting:

Note: The Data Model may restrict which columns can be sorted, in which case Sorting options will not be displayed for those columns.

Query Builder - Filters

This lists all the Data Model columns conditions to filter the data by – so the Query Results will include only the data that satisfy those conditions.
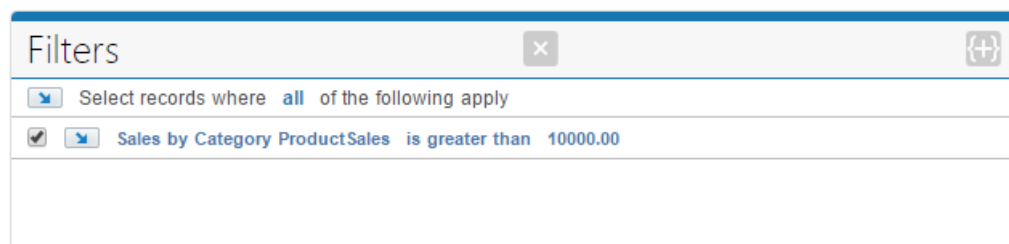
Figure 24: Query Builder Filters – listing 1 filter condition

Use the following buttons to manage the filter list:

☒   Clears all the current filters conditions.

➕   Add a single new filter condition.

Select the [⬎] buttons and blue text to display various options that allow the filters to be modified.

Note: The list of available options varies depending on the type of column being filtered – as defined in the Data Model.

Tips:

- To disable a filter condition without deleting it, untick the box to the left of it.

- You can create a condition with field-to-field comparison (e.g. "Order Date is less then Ship Date"). To do this add a condition as usual and then click on the value element (at the right side of condition) while pressing CTRL button. Then select "Attribute" from the menu that appears.

Query Builder – Updating Query Results

Whilst editing the Query, select the Query Results refresh button to see the latest results based on the currently defined Query.

Figure 25: Query Results – Refresh button

Note: After you Save or Cancel the editing of a Query the Results will automatically update.

Chart Builder

When you add, edit or copy a Query then the Chart Builder area is displayed. This allows you to define an optional chart to display the Query results in a graphical format.
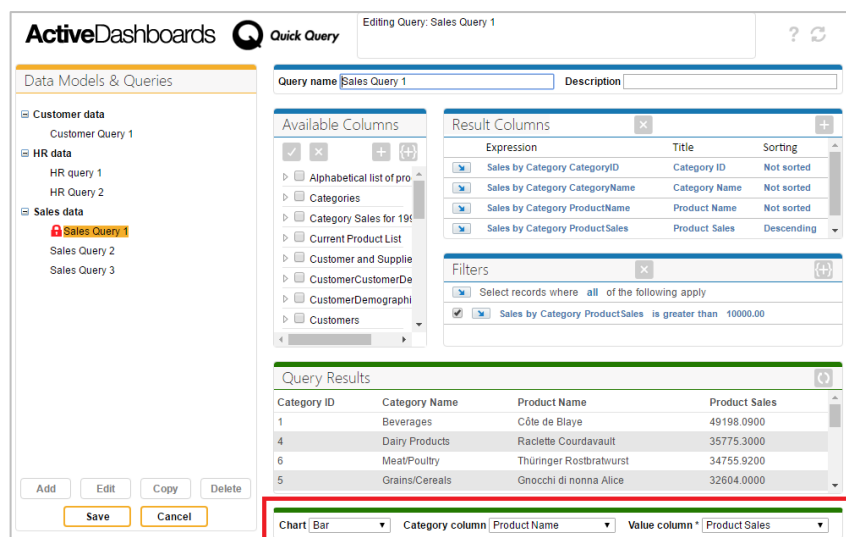
Figure 26: Quick Query Page – Chart Builder

Select the chart type required from the dropdown and then select the relevant Query result columns to display on the chart using the *field* dropdowns.

Figure 27: Chart Builder for a Multi-series chart

Tip: Depending on the display device, you can also select the chart columns by dragging column headings from the *Query Results* onto the chart *field* dropdowns.

Note: The Chart Results are only updated after saving any changes to the Query.

Chapter 4

Data Models

Overview

This chapter is intended for Dashboard Designer users. It describes how to build Data Models for use in the Quick Query page.

Data Model editing requires some understanding of the relational databases handling. Designer users totally unfamiliar with database management should better leave this dialog for a database Operator or another experienced user.

Understanding the Data Model

A Data Model is a user-friendly representation of a database for use by Quick Query. It defines the following main parameters:

- what tables can be used in queries and the links between these tables;

- which fields are shown to the Quick Query users and how they are presented;

- what filters can be applied and how they are presented;

Tables and Links

Each Table in a Data Model represents one table in a database but one table in the database can be represented several times in a Data Model through the use of table aliases.

Tables are not shown to the end-user, they are necessary only for the internal working of Quick Query as it needs to know which tables can be used and how they should be linked to each other.

Usually you start building a Data Model definition by adding of tables and specifying the links between them.

Entities and their Attributes

Each Entity in a Data Model represents for the end-user some object from the real word (e.g. Customer, Order, Employee, etc.).

Each Entity can contains several Attributes, for example "Customer name", "Number of orders", "Company address", "Tax fee", etc.

In most cases an Attribute corresponds to some field in database table. Alternatively, you can define "virtual" Attributes which will be calculated by an expression, e.g. concatenation of two data fields: `FirstName + ' ' + LastName`.

Each Attribute can be configured with Operators and a Value Editor which control how the end-user can apply Filter conditions to the Query results.

Operators

Operator represents and some logical comparison or other predicate (e.g. LIKE, IN, BETWEEN, etc) used to define Query conditions. For example: "is equal to", "is less than", "starts with", etc.

Data Models Editor

To define and manage Data Models select the *Quick Query Data Models* option from the Dashboard Designer *File* menu.

The editor works in the standard Dashboard Designer way - allowing you to add, edit, copy and delete Data Models and group them logically.
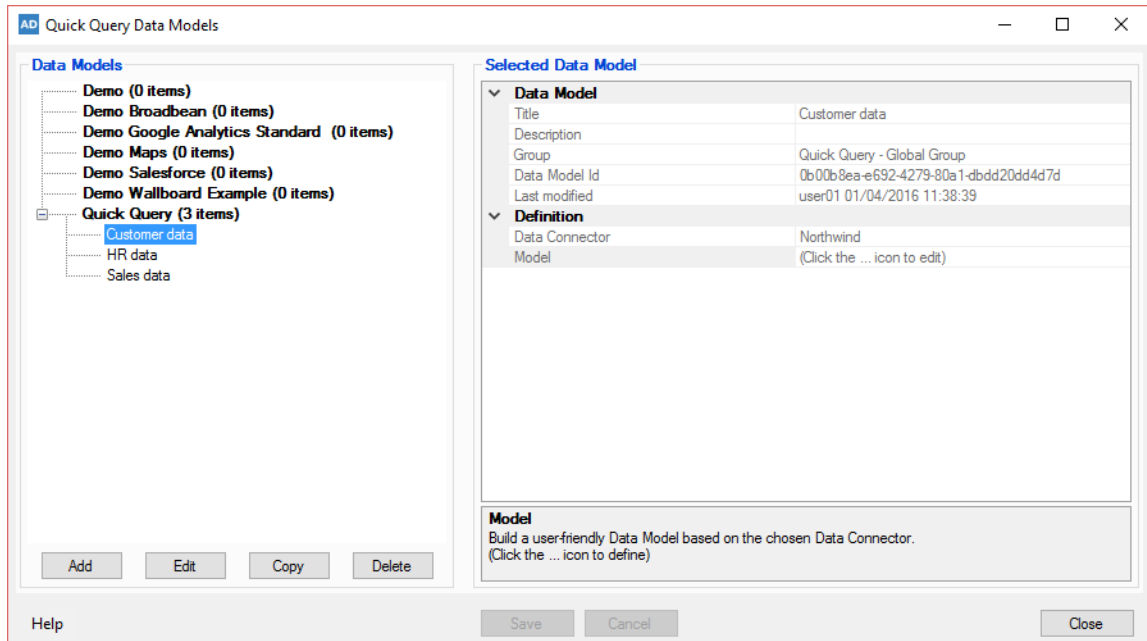
Figure 28: Quick Query Data Models editor

Title / Description

Enter a unique name for the Data Model and an optional description.

Group

The Management Group to which this Data Model belongs. Models in the same group are listed together to aid identification. Note that this is only used for grouping Data Models in the Dashboard Designer, not the Dashboard Web Application.

Data Connector

Choose a Data Connector that connects to the data you wish to model.

SQL format

Only applicable if an ODBC or OLEDB based Data Connector is chosen.

Choose the most applicable format of the SQL Select statement that should be generated when Quick Queries are run using this the Data Model.

Model

Click the '...' button to launch the *Data Model Definition* editor, described in the next section.

Note: Each Data Model defined is exposed in the Dashboard Web Application as follows:

- In the Dashboard Administration page – allowing user access rights for each Data Model to be defined. See *Chapter 2* for details.

- In the Quick Query page – where users with access to the Data Model will be able to query it. See *Chapter 3* for details.

Tip: Right-clicking on one of the Data Models listed displays a context menu from where the *View in browser* option can be selected. This opens a web browser showing the selected Data Model on the Quick Query page. You will require a valid Dashboard session in the same browser (e.g. login to the Dashboards) and access rights to view the selected Data Model first for this to work.

Save / Cancel

Press *Save* to keep any changes or *Cancel* to lose any changes made.

Tip: After altering an existing Data Model it is recommended you review any existing Queries that are using the model to ensure they still function as intended.

Data Model Definition Editor

This editor allows a Data Model to be defined against the chosen Data Connector.
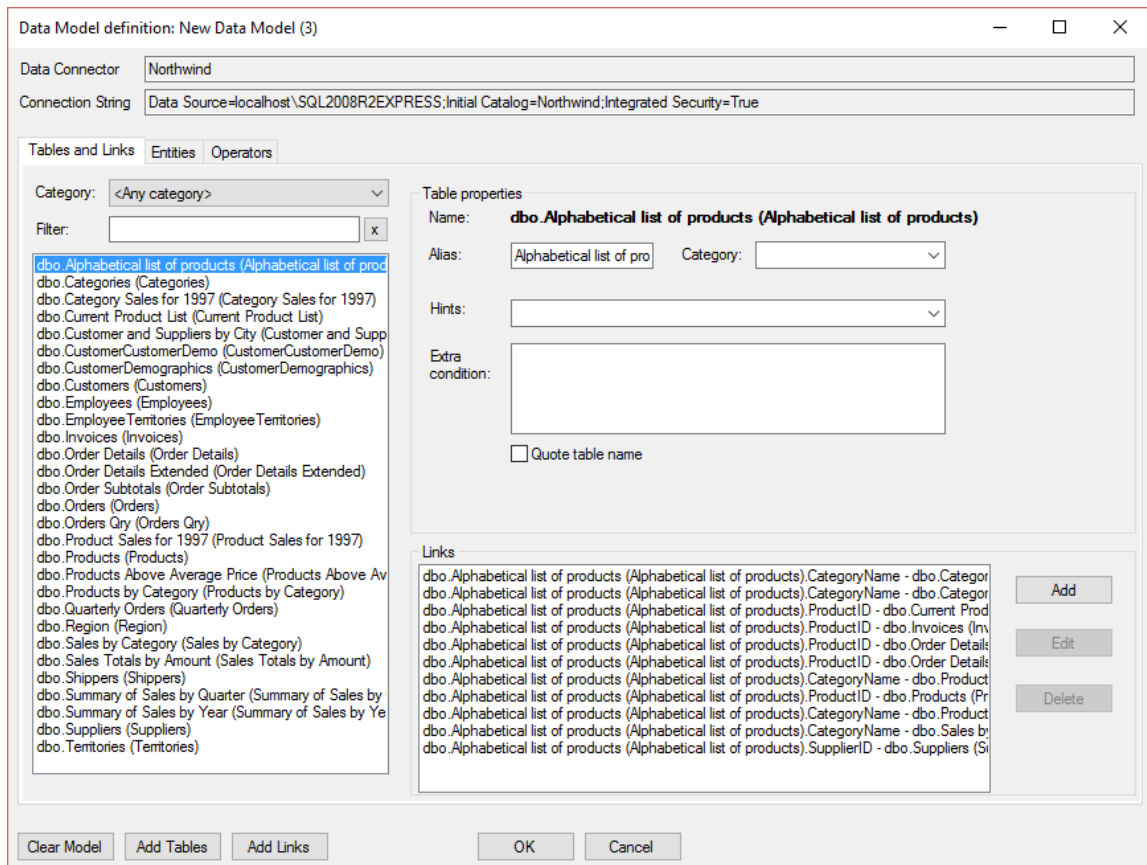

Figure 29: Data Models Definition editor

The editor is split into 3 main tabs:

- **Tables and Links** - describing data tables taking part in the Data Model and how they relate to each another.

- **Entities** - describing the Entities and Attributes (i.e. tables and columns) which Quick Query users can operate with to build queries.

- **Operators** - describing the available filter operations (like comparisons) available for each Attribute.

Other elements of the editor are:

- **Data Connector** (read-only) – The name of the Data Connector you are building the Data Model against.

- **Connection String** (read-only) – The connection string of the Data Connector.

- **Clear Model** – This button resets the Data Model back to its original state.

- **Add Tables** – This button launches the *Add table(s)* dialog, described later, which allows data tables and columns to be added to the Data Model automatically tables by querying the source data's schema.

- **Add Links** – This button attempts to automatically add links between the Data Model's tables by querying the source data's schema.

Data Model Definition - Tables and Links tab

This tab is used to describe data tables taking part in the Data Model.

It consists of a list of Tables (displayed on the left) and details of selected table (displayed on the right), including a list of the table's Links.
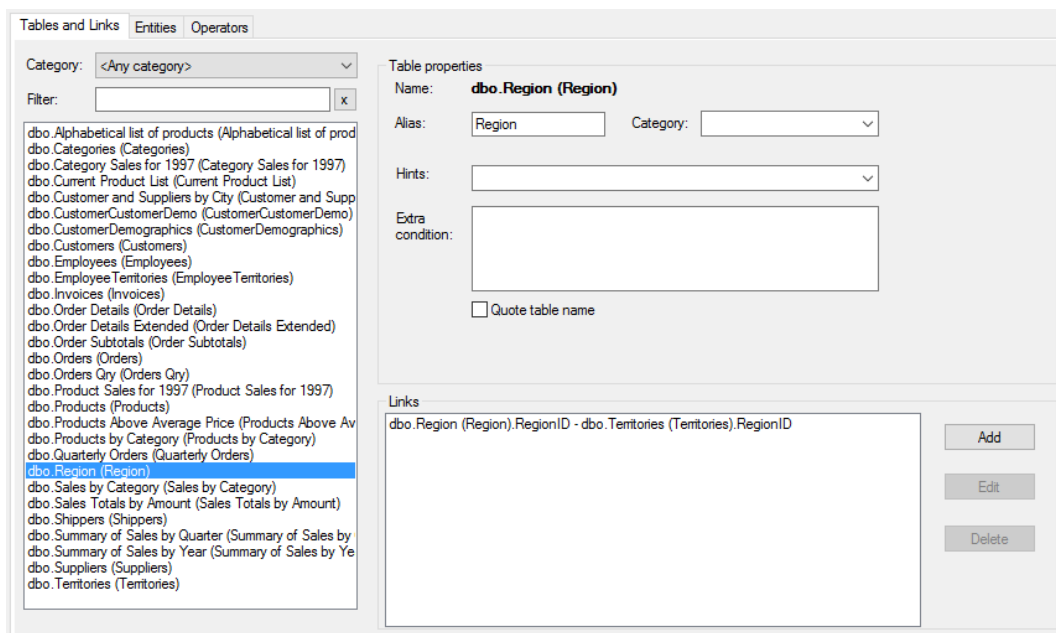
Figure 30: Data Models Definition – Tables and Links tab

The *Category* dropdown and *Filter* text box can be used to filter the list with matching entries. This does not alter the Data Model but can be useful when working with large lists of Tables.

Important: When defining a Data Model it is important to define all required Links between the Tables, as these are used when Quick Query generates SQL statements to query the underlying database when the end-user runs a Query.

Adding Tables

Add Tables to the Data Model by clicking the *Add Tables* button or right-clicking the list and selecting the *Add table(s)* option. This opens the *Add table(s)* dialog.

Figure 31: Add table(s) dialog

Select the required Tables from the list and any options, described below, then click *OK* to add them.

- **Automatically add entities** – Select to also add the selected Tables (and their columns) as corresponding Entities and Attributes in the Data Model.
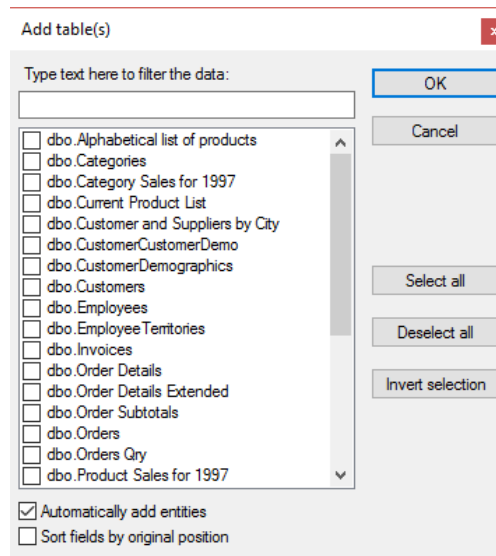
- **Sort fields by ordinal position** – If automatically adding Entities to the Data Model then selecting this adds their Attributes in the order the columns are defined in the tables added.

Adding Virtual Tables

Add Virtual Tables to the Data Model by right-clicking the list and selecting the *Add virtual table* option.

Virtual Tables do not correspond to a table in the database but are defined through a SQL SELECT statement instead.

Removing Tables

Remove Tables from the Data Model by right-clicking the relevant table and selecting the *Delete selected* option.

Table properties

Properties of the selected Table can be configured here.

Figure 32: Table properties

- **Name** – Read-only name of Table.

- **Alias** – Alias for the Table in generated SQL statements. If a Table is added to the Data Model multiple times then they required different aliases.

  Don't confuse this with the name of the Entity displayed to the end-user – this can be defined in the on the *Entities* tab – see the *Data Model Definition – Entities* section below.

- **Category** – Optional way of managing Tables in the editor. Enter a new category name or select an existing one to group the tables. Then use the *Category* dropdown to filter the Tables listed.

- **Hints** – Optional comma separated Table hint(s) to specify locking method used in Microsoft SQL Server syntax (like NOLOCK, ROWLOCK or READCOMMITED).

- **Extra condition** – Optional SQL expression for extra condition. This condition will be added automatically into the generated SQL when this Table is used in Queries.

- **Quote table name** – Tick if the Table name should be put in double quotes in SQL statements - it is useful for Table names including spaces and national characters.

Virtual Table properties

Properties of the selected virtual table can be configured here.

Figure 33: Virtual Table properties

- **Name** - Read-only name of the Virtual Table.

- **Alias** - See definition in *Table properties* above.

- **Category** - See definition in *Table properties* above.

- **Expression** - SQL SELECT statement defining the Virtual Table.

Table Links

Links between the selected Table and any other Tables in the Data Model are listed here.

Figure 34: Table Links list showing a link between 2 Tables

Use the *Add*/*Edit*/*Delete* buttons to manually define links between the selected Table and other Tables in the Data Model.
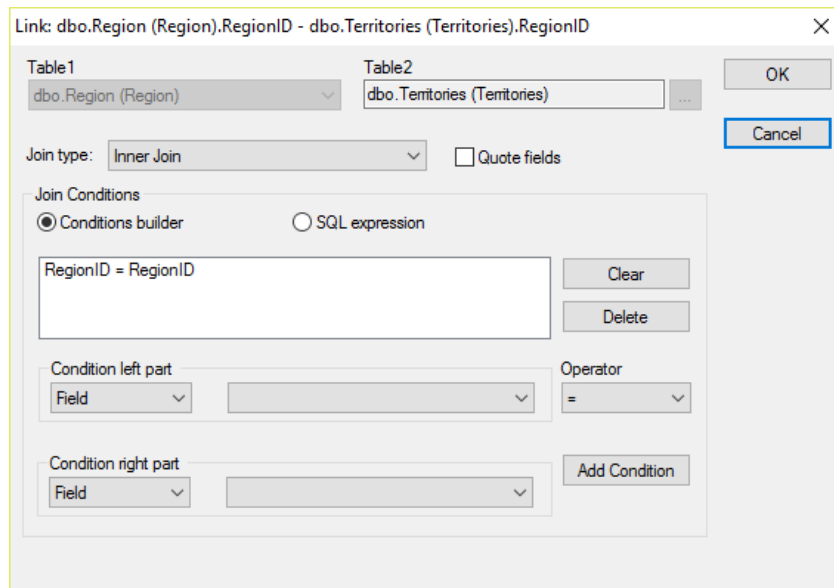
Figure 35: Table Link editor

Data Model Definition – Entities tab

This tab is used to describe Entities and their Attributes participating in the Data Model.

These are what the end-user will see and can choose from when building queries using this Data Model.

It consists of a list of Attributes grouped by Entity (displayed on the left) and details of selected Entity/Attribute (displayed on the right).
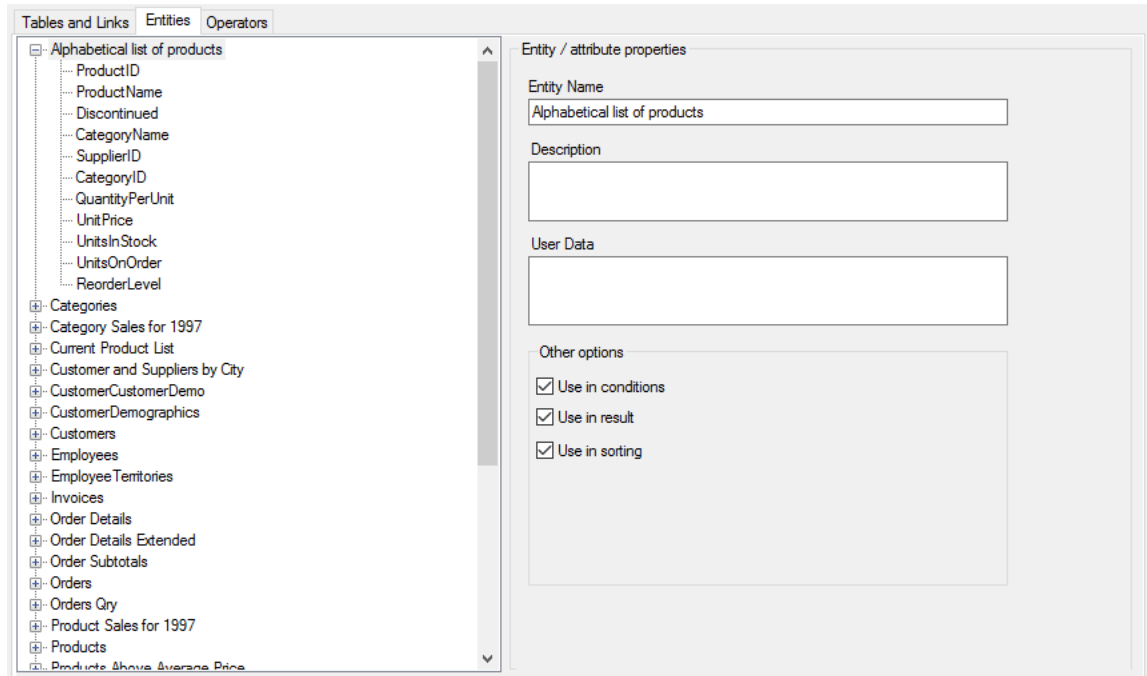

Figure 36: Data Models Definition – Entities tab

The Entities and its Attributes defined here are displayed on the Quick Query page in the same order. You can drag Entities or Attributes to re-order them.
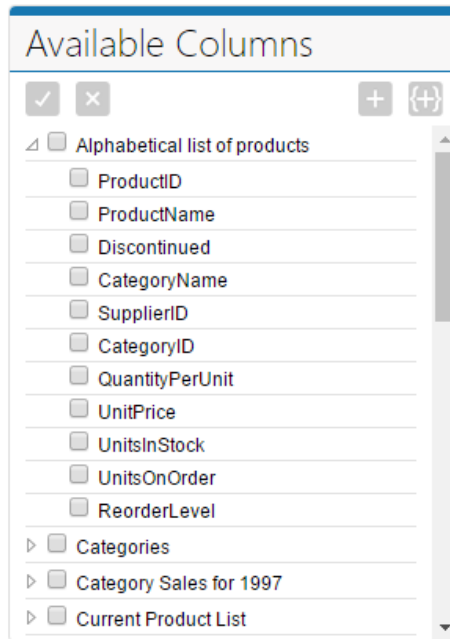
Figure 37: The same Entities and Attributes displayed in Quick Query page

Entities are intended to hide data storage details from the end-user. Instead of operating with tables, views and fields (such as Orders.CustNo, Customers.Addr1) users see some Entities from the real world (Order, Customer, Vendor, etc.) and their Attributes (Customer Name, Order Ship Date, Vendor Country, etc.).

For large Data Models, Entities can be organized in a hierarchy for easier manipulation.

There are two main types of Attributes:

- Data Attribute - corresponds to particular field in a database table (e.g. Attribute Order Ship Date corresponds to database field Orders.ShipDate);

- Virtual Attribute - a calculated Attribute which is defined by an expression containing several fields, operators (+, -, ||, etc.), constants and even functions or storage procedure calls;

Main operations

- To add a data Attribute – right-click the Entity node you would like to add it to and then choose *Add data attribute…* from the context menu. Select the table and its field in the dialog that appear and click OK button.

- To add a Virtual Attribute (i.e., calculated value), right-click the Entity node you would like to add it to and then choose *Add virtual attribute…* from the context menu.

- To delete an Attribute or Entity - right-click it and choose *Delete selected* from the context menu

- To edit an Attribute, select it in the tree and change its properties in the editor appearing in the dialog's right part.

- To move the Attribute from one Entity to another (or to change its appearance order within the Entity) - just drag it to the appropriate place.

Entity properties

The select Entity can be configured here.

Figure 38: Entity properties

- **Entity Name** – The name displayed to the end-user.

- **Description** - Optional description of the Entity. Anything entered is only viewable in the above form and is not exposed to the end-user.

- **User Data** - This is reserved for future use and should not be used.

- **Use in conditions** - Specify if the end-user can filter a Query using Attributes of this Entity.

- **Use in result** - Specify if the end-user can use the Attributes of this Entity in Query results.

- **Use in sorting** - Specify if the end-user can sort the Query results using the Attributes of this Entity.

Attribute Properties

The Attribute's property editor has *General*, *Operators* and *Value Editors* sub-tabs, each described below.

General sub-tab

Figure 39: Data Attribute – General properties

Figure 40: Virtual Attribute – General properties

- **Caption** - The name of the Column that will be displayed to end-user.

- **Expression** - Contains table and field names for data Attributes and full SQL expression for virtual ones. For data Attributes this field is read-only.

- **Used tables** panel - Contains the list of all tables used in the selected Attribute. For data Attributes this field is read-only. For virtual Attributes select the table(s) to use.

- **Data type** and **Size** - Contain type of Attribute and its size. These values are taken from corresponding field definition for data Attributes. For Virtual attributes these should be set manually.

- **Custom function** – Optional SQL function to be applied to the Attribute in all SQL statements that it is used. For example: LOWER({expr}) will result in the Attribute being wrapped in the LOWER function when the SQL statement for a Query is run.

- **Lookup attribute** – Optionally link two Attributes in the Data Model together so they are exposed to the end-user as a single Attribute but behind the scenes applied to the query in different ways

  For example:

  1. You have a foreign key field, CityID, in a table called Customers. This field references a Cities table containing a list of all cities in CityID and CityName fields.

  2. In the Data Model, you expose the CityID field in an Attribute called City, set so the end-user can only use it to filter results and not display it in the results. The Attribute is setup to use a *SQL LIST* value editor so the end-user can pick from a list of city names when filtering.

  e.g. The end-user will see a filter that looks like "Customer City is equal to London" but in the query SQL statement generated it will be something like: Customers.CityID = 234

  3. Now you also need to show the city name in a column in the query result.

  To achieve this you add another City attribute to the Data Model that corresponds to the CityName field of the Cities table. You set it so the end-user can only use it in the query results and not filter with it.

4. You now set the *Lookup attribute* property to link these fields and the end-user will only see one `City` Attribute. The end-user can then use this Attribute in the query results and to filter with - behind the scenes the `CityID` field will be used when filtering and the `CityName` field will be used when displaying it in the query results.

- **Use in conditions** - Specify if the end-user can filter a Query using this Attribute.

- **Use in result** - Specify if the end-user can use this Attribute in Query results.

- **Use in sorting** - Specify if the end-user can sort the Query results using this Attribute.

- **Quote field name in SQL** - Specify if the field name should be put in quotes in SQL statements. Useful for field names which includes some reserved words. Field names with spaces will be quoted automatically.

- **Use alias** - Specifies if the Query builder will generate an alias for the result column where this Attribute is used.

- **Aggregate** – This option is used for Virtual Attributes to indicate that the Attribute is aggregate - so it contains some of aggregate function (SUM, COUNT, etc.).

- **Description** - Optional textual description to be associated with the Attribute. Anything entered is currently exposed in the Data Model editor only – and does not affect the Quick Query page.

- **Custom Data** - This is reserved for future use and should not be used.

Operators sub-tab

This tab lists all filter operations applicable to the Attribute, the first one listed is the default one presented to the Quick Query user, you can drag Operators to re-order them.
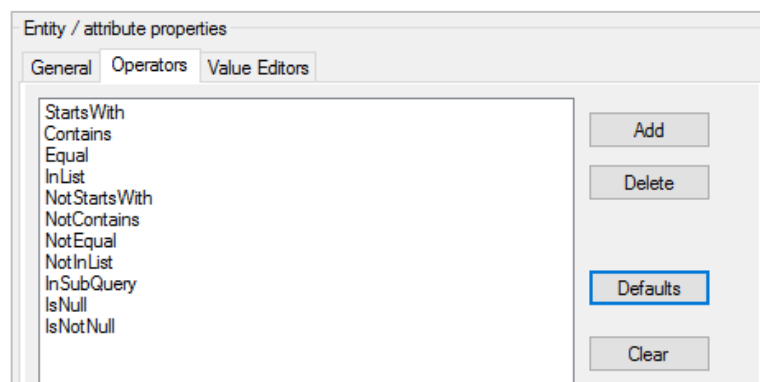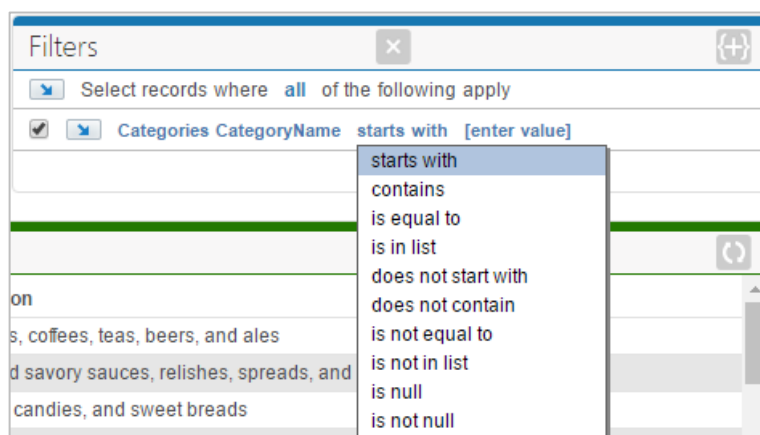

Figure 41: An Attribute's Operators

Figure 42: The same Operators displayed on the Quick Query page

- **Add** button – add a new Operator to the Attribute.

- **Delete** button – remove the selected Operator

- **Defaults** button - resets the list of Operators for the Attribute to the default state. The default Operator list contains the most appropriate Operators for Attribute's data type.

- **Clear** button - remove all Operators from the list.

See the *Data Model Definition – Operators tab* section for more details on Operators.

Value Editors sub-tab

This tab allows you to define how end-users will edit the parameters which the selected Attribute is compared to in Query filter conditions. To specify editor parameters click on *Settings* button.
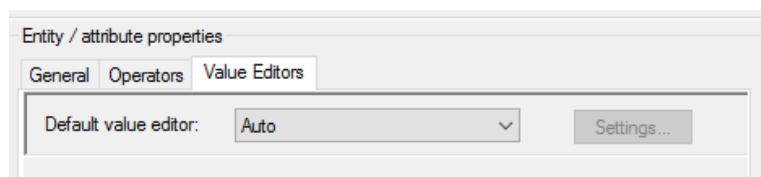

Figure 43: Attribute - Value Editors

You can select one of the following editors:

- **Auto -** The most appropriate value editor will be used depending on the Attribute's data type and Operator which used in condition. For example for date Attribute – a Data/Time picker control will be shown, for Boolean - the user will get an ability to select the value from the list of two items: False and True.

- **Text editor** - Values will be edited in a text box field. You can specify a default value and type of edited data for this editor.

- **Date/time editor** – A date/time picker value editor will be used.

- **List of constants** - Prompt the user to select one (or more in case of using "is in list" Operator) value from a list of available values when filtering a Query.

  The *Value* column contains values to filter by and the *Text* column contains the corresponding text label to display to the user.
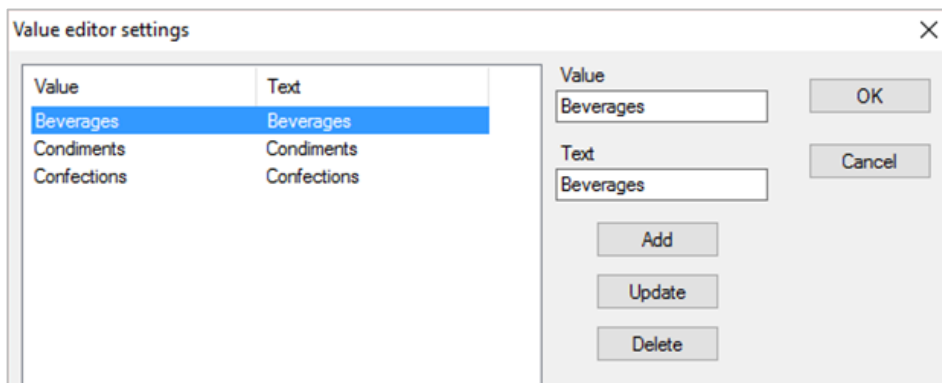

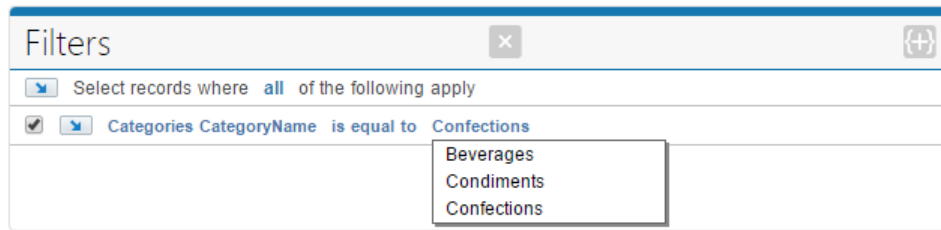**Figure 44: Defining a List of constants for a dropdown filter**

Figure 45: The same filter options displayed on the Quick Query page

- **Custom list** - This editor is reserved for future use and should not be used.

- **SQL list -** Prompt the user to select one (or more in case of using "is in list" Operator) value from a list of available values when filtering a Query.

  Define a SQL Select statement to populate the values. This SQL is run against the same Data Connector used by the Data Model.

Figure 46: Defining a SQL statement to populate a dropdown filter

The 1st column returned by the SQL should contain the values to filter by and 2nd column should contain the corresponding text label to display to the user. If only 1 column of data is returned by the SQL (as in the above example) then this is used for both value and text.
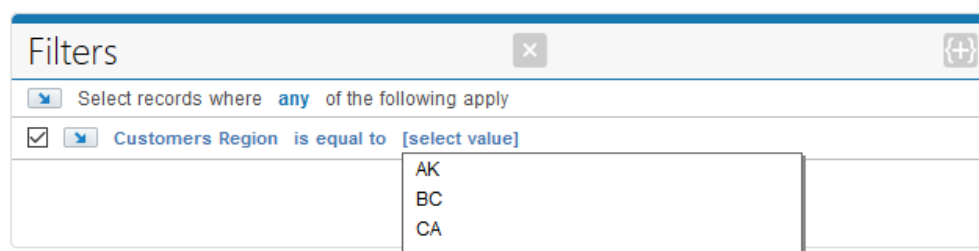

Figure 47: The same filter options displayed on the Quick Query page

- **Custom (user defined)** - This editor is reserved for future use and should not be used.

Data Model Definition – Operators tab

This tab defines default Operators which can be used in Attribute conditions (Filters), such as 'is equal to', 'less than', etc.

Note: In most cases no changes are required to the values on this tab – it is for advanced use only.

It consists of a list of defined operations (displayed on the left) and details of selected Operator (displayed on the right).
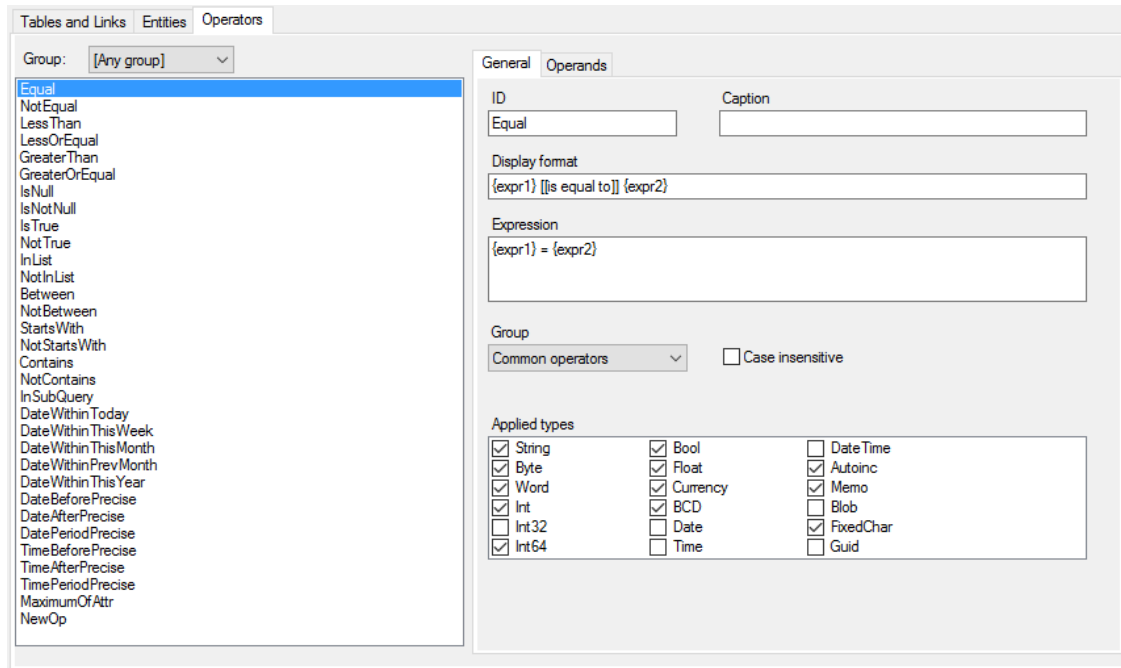

Figure 48: Data Models Definition – Operators tab

Add or delete an Operator by right-clicking the list and choosing the relevant option from the context menu. To edit an Operation choose it from the list and modify its properties displayed on the right.

Operator - General Properties:

- **ID** - The internal identifier for the Operator.

- **Caption** - Allows to specify how this Operator will be presented to the end-user when they select it.

- **Display format** - A template which describes how the Operator will be shown on the Quick Query page.

  The most usual value of this field is:

  ```
  {expr1} [[operator text]] {expr2}
  ```

  Here `{expr1}` and `{expr2}` will be substituted by corresponding expression in condition (an Entity Attribute or constant).

  The text in `[[ ]]` brackets will be shown as link for Operator selection.

  For "between" Operators the *Display format* property will also contain 'and' word between second and third expression:

  ```
  {expr1} [[is between]] {expr2} and {expr3}
  ```

- **Expression** - A template for condition expression in generated SQL query. It may contain any correct SQL expressions (operators such as =, >,<, functions or even names of stored procedures) and the following special variables:

  `{expr1}` - is substituted with the Entity Attribute selected by user;

  `{expr2}`, `{expr3}`, ... - are substituted by constant values typed or selected by user or Entity Attributes selected at the right part of condition;

Additionally you can use the following constructions:

`{exprN.table}` - will be replaced by table name of N expression if this expression represents an Entity Attribute or by empty string in other case.

`{exprN.field}` - will be replaced by field name of N expression if this expression represents an Entity Attribute or by empty string in other case.

Examples:

For the simple "is equal to" Operator the format string is: `{expr1} = {expr2}`

The "starts with" Operator has the following format: `{expr1} LIKE {expr2}`

You can define more complex Operators even with sub-queries. For example here is an expression for "more than average" Operator:

`{expr1} > (SELECT AVG({expr2.field}) FROM {expr2.table})`

- **Group** - Allows to select the group which Operators belongs to.
- **Applied types** – A list of check boxes defining data types to which the operation is applicable.

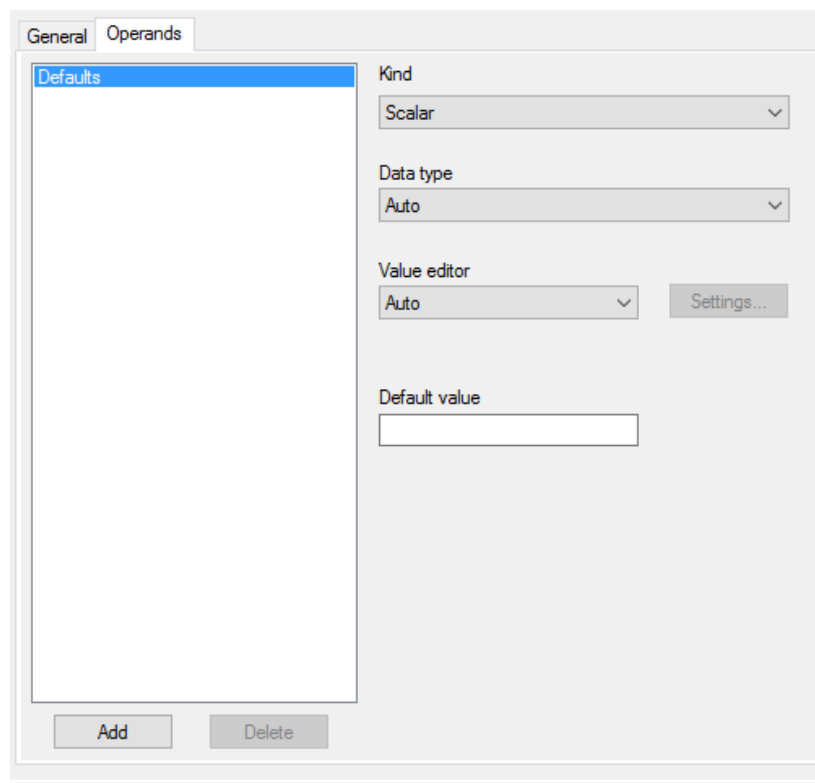**Operator – Operands properties:**

Figure 49: Selected Operator – Operands sub-tab

- **Kind** - Defines the kind of data which has the parameters of this Operator.

"Scalar" - means simple single value is needed: one string, one number, etc. This kind allows also to specify Entity Attribute.

"Const" - the same as previous one but allows to specify only constant value (Entity Attributes are not allowed);

"Attribute" - the same as the first one but allows to specify only Entity Attribute (constant values are not allowed);

"List" type requires list of scalar values separated by comma. E.g., having this option checked, when the user enters a, b, c as parameter value, it's treated as 'a', 'b', 'c' instead of 'a, b, c' in the generated SQL text.

"Query" type means that the Operator requires a SQL SELECT statement as a value in the right part of condition. To build this statement a query panel opens separate dialog.

- **Data type** - Represents expected type of Operator parameters used in condition expression. "Auto" value means the same type as the type of Entity Attribute used at the left part of condition, other values are correspond to particular data types.

- **Value editor** - The default value editor for Operator's parameters.

  The value editor chosen here has higher priority in a condition than the editor defined for Entity Attribute used at the left part of this condition.

  The available value editors are described in the *Attribute Properties* section above.

- **Default value** – Optional value that is compatible with operand's data type. This value will be used automatically when the end-user selects this Operator in a Query Filter.

Example

In this example we show how to build a Data Model from a relational database in the Dashboard Designer and view it in the Quick Query web page

Dashboard Designer

1. Login to the Dashboard Designer.

2. On the Data Connector tab setup and test a new Data Connector to a relational database.

   In this example the Microsoft SQL Server sample database *AdventureWorks* is used:

 | Selected Data Connector | |
   | --- | --- |
   | ∨ **Data Connector** | |
   | Title | **AdventureWorks** |
   | Description | |
   | Group | **Demo - Global Group** |
   | Data provider | **SQL Server** |
   | Connection string | Data Source=localhost\SQL2008R2EXPRESS;Initial Catalog=AdventureWorks |
   | Encrypt connection string | **True** |

3. Launch the *Quick Query Data Models* editor from the Designer *File* menu.

4. Click the *Add* button to create a new Data Model titled "Adventure Works" and chose the above Data Connector.

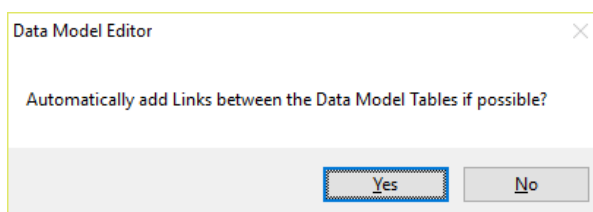 | Selected Data Model | |
   | --- | --- |
   | ∨ **Data Model** | |
   | Title | **Adventure Works** |
   | Description | |
   | Group | **Demo - Global Group** |
   | Data Model Id | 233f1582-8983-4a67-ab49-47b56db8753e |
   | Last modified | user01 05/04/2016 15:12:39 |
   | ∨ **Definition** | |
   | Data Connector | **AdventureWorks** |
   | Model | **(Click the … icon to define)** |

5. On the *Model* property click the '...' button to launch the Data Model Definition editor.

6. This should prompt you to add the database Tables and Links into the Data Model:
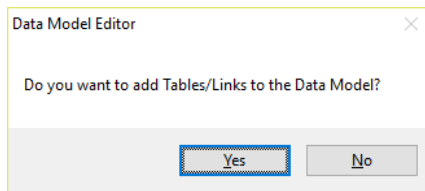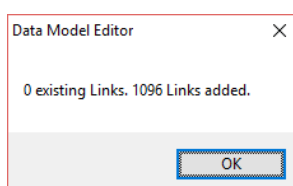

7. Click *Yes* and the *Add table(s)* dialog appears:
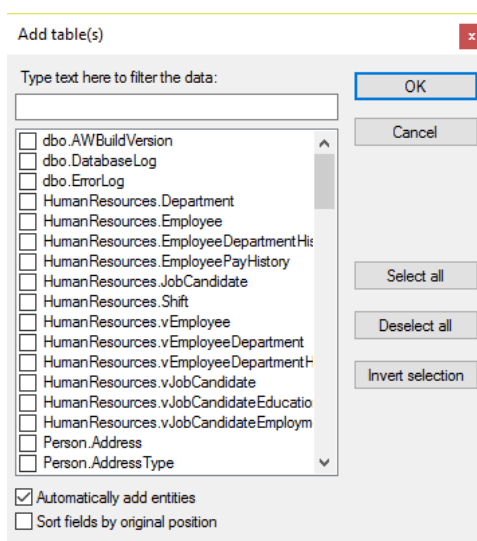
8. Select all the tables from the list and ensure the *Automatically add entities* option is selected.

9. Click *OK* after the tables have been added to the Data Model. You will then be prompted to automatically add any links between the tables to the Data Model:
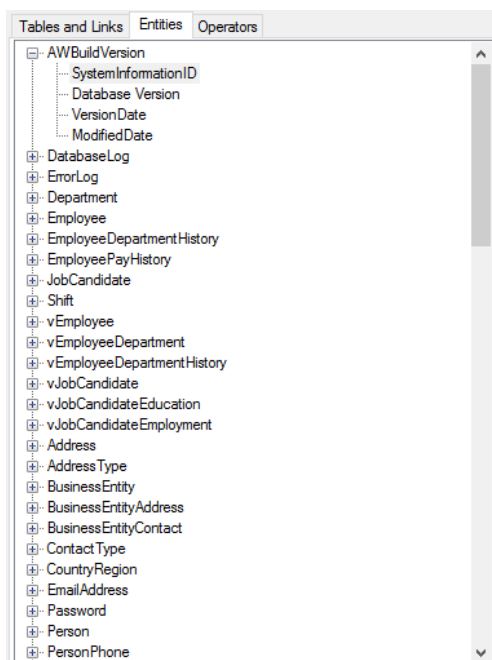

10. Click *Yes* and wait for the links to be added.

11. Click *OK* and the Tables and Links added to the Data Model are displayed:
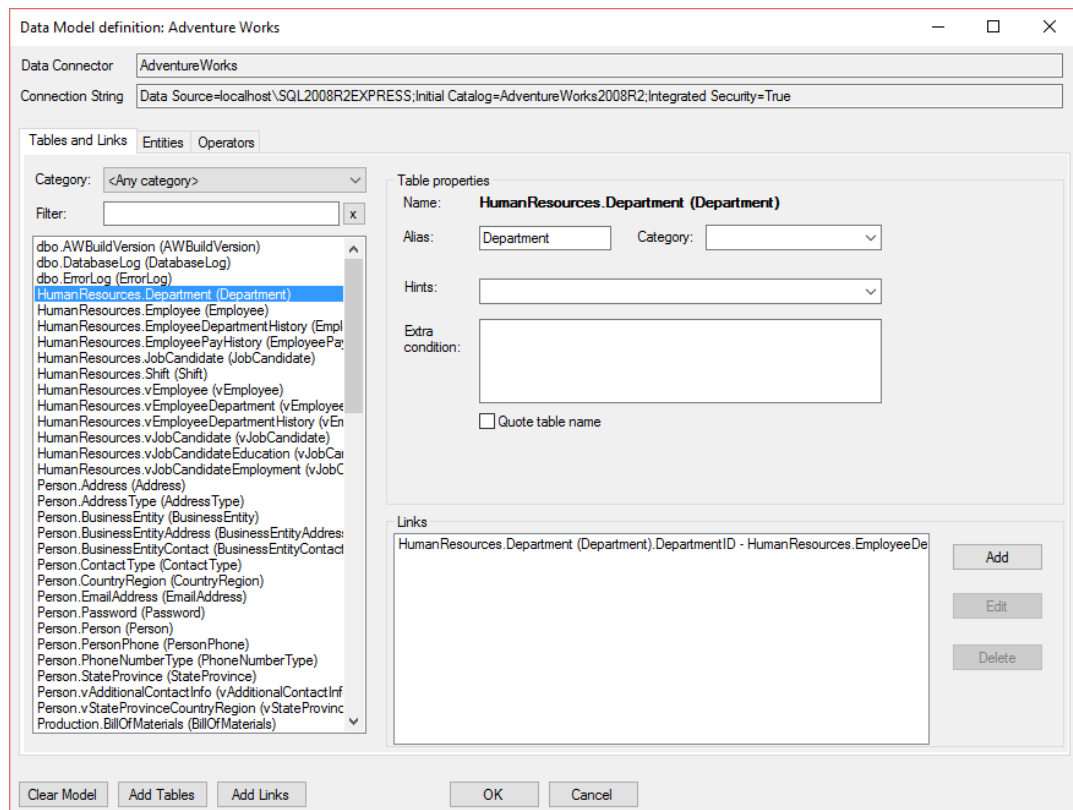
12. Select the *Entities* tab to see the corresponding Entities and Attributes added to the Data Model that the end-user will see when building Queries based on this Data Model:
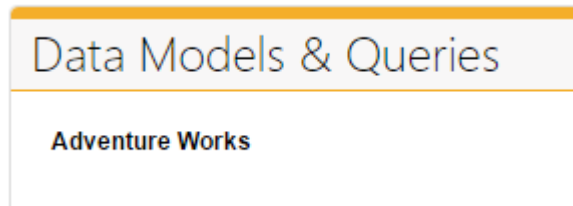

13. Click *OK* to keep your changes and exit the Data Model Definition editor.

14. Click *Save* on the Quick Query Data Models editor.

Dashboard Web Application – Quick Query page

1. Open the Dashboards in your browser and launch Quick Query.

2. You should see the "Adventure Works" Data Model listed, select it and click the *Add* button to create a new Query of it.

If you don't see this Data Model listed or cannot create a new Query for it then check you Quick Query access to resolve this. See *Chapter 2* for details.

3. The Query Builder displays, allowing you to query the data exposed through this Data Model.

For example:

Click *Save* to store the new Query.

Appendix A

Troubleshooting

Quick Query Page

1. **Why don't I see any Data Models or Queries listed?**

    This may be because you have been granted full access to Quick Query but no Data Models have been defined by Dashboard Designer users yet.

2. **I know a Data Model exists so why can't I see it listed?**

    Probably because a Dashboard Administrator user has not granted you access to the Data Model. You can also try refreshing the page to update the list.

3. **Why do I see "Error running query" in the Query Results?**

    This could be for a variety of reasons such as:

    -   The Data Model used by the Query is no longer correct or has been altered.

    -   There is an issue connecting to the underlying data being queried

    Report the issue to your system administrator or Dashboard Designer user.

4. **Why do I get an "Error saving…" or "Error deleting…" or similar message?**

    This indicates an issue with the Dashboard installation – probably related to file permissions. Contact your system administrator with details.

5. **Why do I see messages such as "You no longer have permission to view this page." or "You no longer have permission to perform this action"?**

    Probably because your Quick Query access rights have recently changed by a Dashboard Administrator user.

6. **Why can't I edit or delete a particular Query – I have been granted access to do this?**

    Probably because another user has started editing the Query since your page last updated and they have a lock on the Query meaning no one else can edit it.

7. **I'm sure I saw a particular Data Model or Query listed a moment ago but it has disappeared - why?**

    Probably because another user has recently renamed or deleted the Query or Data Model. It could also be because your Quick Query access rights have recently changed by a Dashboard Administrator user.

8. **Why is the Query Results not displaying a particular Column any more – the Query hasn't changed recently?**

    This is probably because the Query's Data Model has been altered since the Query was last edited, with Tables or Columns removed from the Data Model. These changes to the Data Model are reflected in the results.

Data Model Editor (Dashboard Designer Users)

1. **Why do users report they are seeing "Error running Data Model Query: Error building SQL statement from Query: Cannot find a path between tables "x" and "y" messages or similar?**

Check the Data Model they are using is fully defined, i.e. all Tables exposed to Quick Query users should have the relevant Links defined so Quick Query knows how to construct the SQL Statement to perform the query.

2. **Why do users report they are seeing the following message when trying to add a new Query "The selected Data Model is not fully defined - it cannot be queried yet."**

The Data Model they are using is either not fully setup yet or a dependency such as its Data Connector has been deleted.

3. **Users are complaining that Queries are locked by other users – so they cannot be edited.**

Quick Query implements a locking mechanism to prevent multiple users editing the same Query simultaneously. Locks should be released automatically but if users forget to save or cancel the editing of a Query then locks will persist. "Web App - Quick Query" locks can be managed using the *Lock Maintenance* facility available from the *Maintenance* menu of the Dashboard Designer.