

# Viewing Table Data

Some of the features and screenshots in this section are for the DbVisualizer Pro edition.

A table's data can be viewed in various ways in the **Data** tab in its **Object View** tab.

- [Opening the Data tab](#)
- [Sorting](#)
- [Formatting](#)
- [Where Filter](#)
- [Column Filter](#)
- [Quick Filter](#)
- [Max Rows/Max Chars](#)
- [Max Rows at First Display](#)
- [Column Header Tooltips](#)
- [Highlight Primary Key Columns](#)
- [Auto Resize Columns](#)
- [Show Only Some Columns](#)
- [Right-Click Menu Operations](#)
  - [Script to SQL Commander](#)
- [Creating Monitors](#)
- [Aggregation Data for Selection](#)

Check [Editing Table Data](#) for information about the table data editor in DbVisualizer.

Check [Viewing BSON Document Data](#) for information about the document data viewer in DbVisualizer.

## Opening the Data tab

To open the Data tab for a table:

1. Locate the table in the **Databases** tab tree,
2. Double-click the table node to open its **Object View** tab,
3. Open the **Data** sub tab.

**Table: ACTOR** Actions...

Sakila H2 (dbvis)/Schemas/SAKILA/Tables/ACTOR

Info Columns **Data** Row Count Primary Key

Sort

* 1	ACTOR_ID	FIRST_NAME	LAST_NAME	LAST_UPDATE
1	71	ADAM	GRANT	2006-02-15 04:34:33
2	132	ADAM	HOPPER	2006-02-15 04:34:33
3	165	AL	GARLAND	2006-02-15 04:34:33
4	173	ALAN	DREYFUSS	2006-02-15 04:34:33
5	125	ALBERT	NOLTE	2006-02-15 04:34:33
6	146	ALBERT	JOHANSSON	2006-02-15 04:34:33
7	29	ALEC	WAYNE	2006-02-15 04:34:33
8	65	ANGELA	HUDSON	2006-02-15 04:34:33
9	144	ANGELA	WITHERSPOON	2006-02-15 04:34:33
10	76	ANGELINA	ASTAIRE	2006-02-15 04:34:33
11	49	ANNE	CRONYN	2006-02-15 04:34:33
12	34	AUDREY	OLIVIER	2006-02-15 04:34:33
13	190	AUDREY	BAILEY	2006-02-15 04:34:33
14	196	BELA	WALKEN	2006-02-15 04:34:33
15	83	BEN	WILLIS	2006-02-15 04:34:33
16	152	BEN	HARRIS	2006-02-15 04:34:33
17	6	BETTE	NICHOLSON	2006-02-15 04:34:33

Max Rows: 1000 Max Chars: -1 Timestamp: yyyy-MM-dd HH:mm:ss 0.002

Each column width is automatically resized to match the column width, including the column header, by default. You can disable this behavior in the the Tool Properties dialog, in the **Grid** category under the General tab.

If **Auto Resize Column Widths** is enabled, the **Max Column Width** setting can be used to limit the column width so that an extremely wide column does not take up all space.

In the same Tool Properties category, you can also disable **Show Grid Row Header**, i.e. the row number shown to the left of the data rows, for read-only grids such as the Data tab in the DbVisualizer Free edition and result sets from joined tables.

The column headers correspond to the column names by default, but you can specify in the Tool Properties dialog, in the Grid category under the General tab, that you like to use the column alias instead. This is mostly useful for grids representing SQL Commander result sets, but may also be useful in the Data tab grid for some databases.

The **Data** tab contains a number of features for locating and focusing on just the data of interest as described in the following sections.

## Sorting

You can sort the data grid based on the values in one or more columns:

1. Click on a column header to sort the grid in ascending order on the values in that column, indicated by an up-arrow in the column header.
2. Click the same column header again to sort in descending order, indicated by a down-arrow in the column header.
3. Click a third time to show the data in the order it was received from the database. This removes the sort indicator.

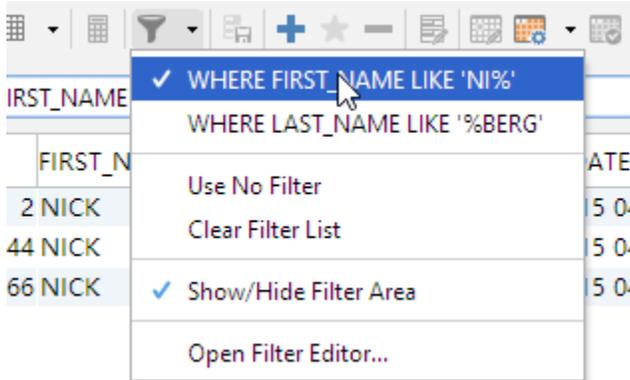
To sort on more than one column, Ctrl-click (keep the Ctrl key pressed when clicking) on additional columns. The grid is then sorted on the values in the first column you clicked on (indicated with a 1 next to the arrow), and then all rows with the same value in the first column are sorted on the values in the second sort column (indicated with a 2 next to the arrow), and so on.

## Formatting

Note that when a cell is selected, the format pattern used to display the cell value is visible at bottom of the grid. By clicking the blue link, the [Data Formats](#) section of **Tool Properties** will be opened.

## Where Filter

You can use the filter capability in the **Data** tab to limit the number of rows shown in the grid, using the same syntax as for an SQL WHERE clause. The Filter menu button in the grid toolbar contains all operations related to using a filter.

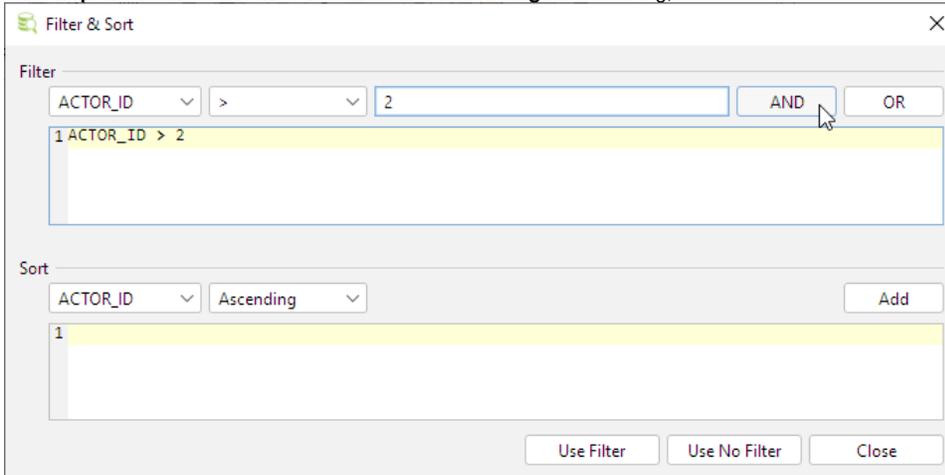


The top entries in the menu are previously used filters for the table, if any. The checkbox is selected for the filter that is currently in use. The filters are saved between DbVisualizer sessions, and you can toggle between them by selecting them from the menu. The maximum number of filters to save is specified in the Tool Properties dialog, in the **Table Data** category under the General tab.

You use the **Use No Filter** choice to disable all filters for the table, and the **Clear Filter List** to permanently remove all filters for the table.

To create a new filter:

1. Select **Open Filter Editor** to launch the **Filter & Sort Configuration** dialog,



2. Select a column name, an operation and the value for the condition using the controls in the **Filter** area,
3. Add the condition to the filter by clicking the **AND** or **OR** button, and create additional conditions in the same way if needed,
4. Click the **Use Filter** button to apply the filter, save it and close the dialog, or close the dialog without applying and saving the filter by clicking the **Close** button.

You can use **Ctrl-Enter** while editing the value field to reload the grid with that single condition applied, or in the editor to reload the grid based on all filter conditions created so far.

The **Sort** area is similar to the **Filter** area. You can select column names and sort order from the two lists, and click the **Add** button to add the sort criteria for the single column to the complete criteria.

If you often need to tweak the filter conditions and want a more compact user interface, you can use the inline filter view. Use the **Show/Hide Filter Area** choice in the **Filter** menu to toggle the visibility of the inline filter.

**Table: ACTOR** Actions...

Sakila H2 (dbvis)/Schemas/SAKILA/Tables/ACTOR

Info Columns **Data** Row Count Primary Key

Filter: WHERE FIRST\_NAME LIKE 'NI%' OR LAST\_NAME LIKE '%BERG'

*  ACTOR_ID	FIRST_NAME	LAST_NAME	LAST_UPDATE
1	2 NICK	WAHLBERG	2006-02-15 04:34:33
2	44 NICK	STALLONE	2006-02-15 04:34:33
3	46 PARKER	GOLDBERG	2006-02-15 04:34:33
4	95 DARYL	WAHLBERG	2006-02-15 04:34:33
5	166 NICK	DEGENERES	2006-02-15 04:34:33

Max Rows: 1000 Max Chars: -1 Format: <Select a Cell> 0.002/0.000 sec 5/4

## Column Filter

Only in DbVisualizer Pro

 This feature is only available in the DbVisualizer Pro edition.

The **Column Filter** acts on the data that is already loaded in the grid, as opposed to a [Where Filter](#) which is used to limit the number of rows fetched from the database. With a **Column Filter**, you can easily list only those rows having a column value matching the **Column Filter**. By customizing the filter it is possible to add more complex filters such as only listing rows where the column value does not contain a certain value or substring.

A **Column Filter** is added by clicking the right part of the column header. A column having a Column Filter is indicated by the presence of a filter icon in the column header.

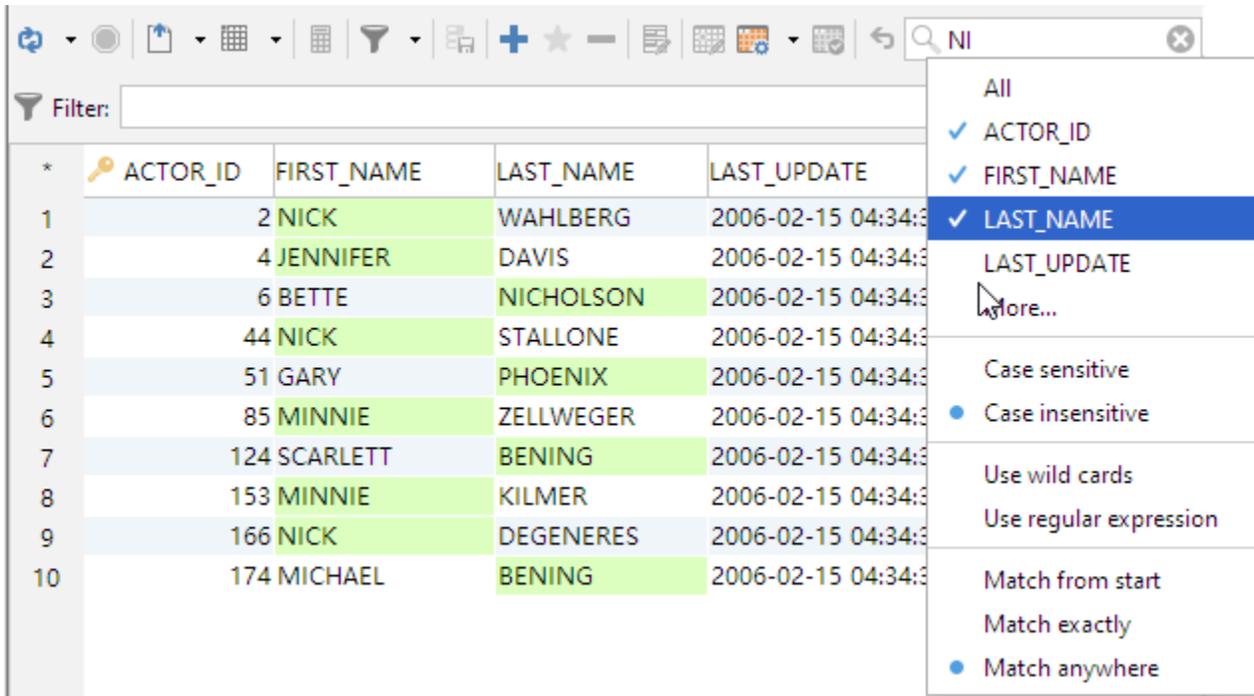
It is possible to add **Column Filters** to multiple columns. For cases where you don't care which column has a specific value we recommend the [Quick Filter](#)

## Quick Filter

Only in DbVisualizer Pro

 This feature is only available in the DbVisualizer Pro edition.

The **Quick Filter** acts on the data that is already loaded in the grid, as opposed to a [Where Filter](#) which is used to limit the number of rows fetched from the database. With a **Quick Filter**, you can easily list only those rows in the grid that match the entered search string.



Use the Quick Filter pull-down menu (click on the magnifying glass) to choose if the filter should match cells in all columns or just one selected column, case or case insensitive matching, and where in the cell the value must match.

For the **Use wild cards** option the following characters have special meaning:

- ? - The question mark indicates there is zero or one of the preceding element. For example, colour?r matches both "color" and "colour".
- \* - The asterisk indicates there are zero or more of the preceding element. For example, ab\*c matches "ac", "abc", "abbc", "abbbc", and so on.
- + - The plus sign indicates that there is one or more of the preceding element. For example, ab+c matches "abc", "abbc", "abbbc", and so on, but not "ac".

## Max Rows/Max Chars

DbVisualizer limits the number of rows shown in the Data tab to 1000 rows, by default. This is done to conserve memory. If this limit prevents you from seeing the data of interest, you should first consider:

1. Using a [Where Filter](#) to only retrieve the rows of interest instead of all rows in the table,
2. [Exporting the table](#) to a file

If you really need to look at more than 1000 rows, you can change the value in the **Max Rows** field in the grid status bar. Use a value of 0 or -1 to get all rows, or a specific number (e.g. 5000) to set a new limit.

Character data columns may contain very large values that use up lots of memory. If you are only interested in seeing a few characters, you can set the **Max Chars** field in the grid status bar to the number of characters you want to see.

You can define how to deal with columns that have more characters than the specified maximum in the Tool Properties dialog, in the Grid category under the General tab. You have two choices: **Truncate Values** or **Truncate Values Visually**.

- **Truncate Values** truncates the original value for the grid cell to be less than the setting of Max Chars.

This affects any subsequent edits and SQL operations that use the value since it's truncated. This setting is only useful to save memory when viewing very large text columns.

- **Truncate Values Visually** truncates the visible value only and leave the original value intact. This is the preferred setting since it will not harm the original value. The disadvantage is that more memory is needed when dealing with large text columns.

When the grid data is limited due to either the **Max Rows** or **Max Chars** value, you get an indication about this in the rows/columns field in the grid status bar and in the corresponding limit field. The color is also changed for the affected controls.

**Table: RENTAL** Actions...

Sakila H2 (dbvis)/Schemas/SAKILA/Tables/RENTAL

Info Columns **Data** Row Count Primary Key Indexes Grants Row Id DDL Reference: < >

Filter:

* RENTAL_ID	RENTAL_DATE	INVENTORY_ID	CUSTOMER_ID	RETURN_DATE	STAFF_ID	LAST_UPDATE
1	2005-05-24 22:53:30	367	130	2005-05-26 22:04:30	1	2006-02-15 21:30:53
2	2005-05-24 22:54:33	1525	459	2005-05-28 19:40:33	1	2006-02-15 21:30:53
3	2005-05-24 23:03:39	1711	408	2005-06-01 22:12:39	1	2006-02-15 21:30:53
4	2005-05-24 23:04:41	2452	333	2005-06-03 01:43:41	2	2006-02-15 21:30:53
5	2005-05-24 23:05:21	2079	222	2005-06-02 04:33:21	1	2006-02-15 21:30:53
6	2005-05-24 23:08:07	2792	549	2005-05-27 01:32:07	1	2006-02-15 21:30:53
7	2005-05-24 23:11:53	3995	269	2005-05-29 20:34:53	2	2006-02-15 21:30:53
8	2005-05-24 23:31:46	2346	239	2005-05-27 23:33:46	2	2006-02-15 21:30:53
9	2005-05-25 00:00:40	2580	126	2005-05-28 00:22:40	1	2006-02-15 21:30:53
10	2005-05-25 00:02:21	1824	399	2005-05-31 22:44:21	2	2006-02-15 21:30:53
11	2005-05-25 00:09:02	4443	142	2005-06-02 20:56:02	2	2006-02-15 21:30:53
12	2005-05-25 00:19:27	1584	261	2005-05-30 05:44:27	2	2006-02-15 21:30:53
13	2005-05-25 00:22:55	2294	334	2005-05-30 04:28:55	1	2006-02-15 21:30:53
14	2005-05-25 00:31:15	2701	446	2005-05-26 02:56:15	1	2006-02-15 21:30:53
15	2005-05-25 00:39:22	3049	319	2005-06-03 03:30:22	1	2006-02-15 21:30:53
16	2005-05-25 00:43:11	389	316	2005-05-26 04:42:11	2	2006-02-15 21:30:53
17	2005-05-25 01:06:36	830	575	2005-05-27 00:43:36	1	2006-02-15 21:30:53
18	2005-05-25 01:10:47	3376	19	2005-05-31 06:35:47	2	2006-02-15 21:30:53
19	2005-05-25 01:17:24	1941	456	2005-05-31 06:00:24	1	2006-02-15 21:30:53
20	2005-05-25 01:48:41	3517	185	2005-05-26 01:01:46	2	2006-02-15 21:30:53
21	2005-05-25 01:59:46	146	388	2005-05-26 01:01:46	2	2006-02-15 21:30:53

Max Rows: 1000 Max Chars: -1 Format: <Select a Cell> 0.001/0.002 sec 1000/7 1-22  
90M of 768M

Along with the highlighted field, a warning pops up close to the field. You can disable this behavior in the **Tool Properties** dialog, in the **General / Grid** category.

### Max Rows at First Display

By default, opening the Data tab for a table loads all rows, unless there is a Max Rows limit. If you have very large tables and don't want to risk memory issues if you accidentally open the Data tab and have no Max Rows limit, you can specify a **Max Rows at First Display** limit. You do this in the Tool Properties dialog, in the **Table Data** category under the General tab.

The default is -1, which means no limit. If you set it to a positive number, only the specified number of rows are loaded when the Data tab is first opened for a table. To load more rows, click the **Reload** button in the Data tab toolbar.

### Column Header Tooltips

The column header tooltip shows data type information about the column. To see the tooltip, let the mouse hover over the column header. The tooltip pops up in about a second.

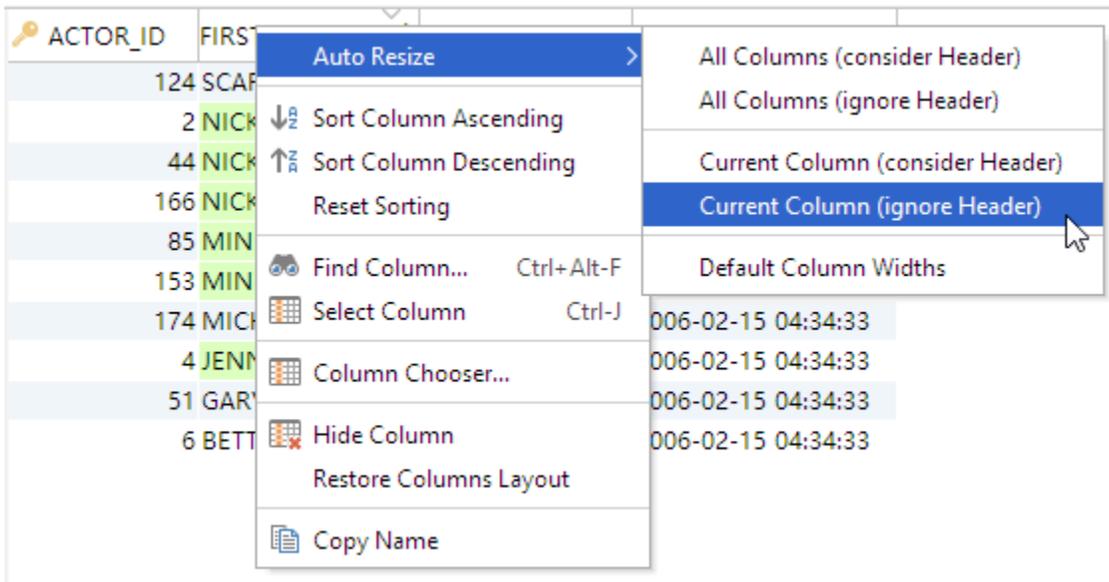
2: **FIRST\_NAME** VARCHAR (45)  
 Not NULL  
 JDBC: VARCHAR (type: 12), Java: String

### Highlight Primary Key Columns

By default, a Primary Key column is shown with an icon in the column header. You can disable this in the Tool Properties dialog, in the **Table Data** category under the General tab.

### Auto Resize Columns

The column header right-click menu contains a number of options for automatic resizing of column widths.



## Show Only Some Columns

The **Grid Column Chooser** dialog controls which columns you want to appear in a grid. Open the dialog by the right-click menu on the column header or the button above the vertical scrollbar in the grid. See the screenshot below.

Grid Column Chooser with the **default layout**.  
The icon in the top right corner of the grid indicates the standard layout.

Original Index	Visible	Width Changed	Key Column	Column Name
1	<input checked="" type="checkbox"/>	<input type="checkbox"/>		ACTOR_ID
2	<input checked="" type="checkbox"/>	<input type="checkbox"/>		FIRST_NAME
3	<input checked="" type="checkbox"/>	<input type="checkbox"/>		LAST_NAME
4	<input checked="" type="checkbox"/>	<input type="checkbox"/>		LAST_UPDATE

Buttons: Default Layout, OK

Grid Column Chooser with a **custom layout**.  
The icon in the top right corner of the grid indicates the custom layout.

Original Index	Visible	Width Changed	Column Name
1	<input type="checkbox"/>		ACTOR_ID
2	<input checked="" type="checkbox"/>		FIRST_NAME
3	<input checked="" type="checkbox"/>		LAST_NAME
4	<input type="checkbox"/>		LAST_UPDATE

Buttons: Default Layout, OK

The **Grid Column Chooser** dialog shows all columns that are available in the grid. The checkmark in front of a column name indicates that the column is visible in the grid, while an unchecked box indicates that it is excluded from the grid. Click the checkmark to change the visibility of a column. You can change the visibility for all columns at once using the two visibility buttons in the dialog.

The order of the columns can also be adjusted in this dialog. Just select one or several rows and use the **Up** and **Down** buttons to move the rows up (left in the grid) or down (right in the grid). You can also use **Drag&Drop** to move the rows.

If you want to revert your changes, you can click on the **Default Layout** button to reset the grid, i.e., making all columns visible and putting them in their default locations.

Modifications of column visibility, size, and order are saved between invocations of DbVisualizer for the **Log** tab and all grids in the various **Object View** tabs except for the **Data** tab.

If you modify the column visibility in the **Data** tab, the changes persist throughout the session. For instance, if you remove the **ACTOR\_ID** column in the **Data** tab for the table Actor, the **ACTOR\_ID** column remains excluded when you reload the table or come back to the **Data** tab for that table later in the same session. You must manually make it visible again to bring it back. The changes are, however, reset when you restart the application.

If you modify the column visibility in the **Log** tab in the SQL Commander, the changes will affect the Log tab immediately and other SQL Commander Log tabs the next time they are opened or refreshed by running a query. Any custom column setups are saved individually for the Log tab in SQL Commander, Actions, Export (all but export grid), Import, and Procedure Editor.

## Right-Click Menu Operations

The right-click menu for the grid contains a lot of operations for working with the data without changing it. In addition to the common select, copy, and print operations, some operations that may require a bit of an explanation are described below. Examples are based on the following simple grid that contains 9 cells on 3 rows with 3 columns, where the first two rows are selected:

*  COL1	COL2	COL3
1	A	c3
2	B	c3
3	C	c3

Operation	Description
<b>Copy Selection</b>	Copy all selected cells onto the system clipboard.  <u>Example:</u> 1 A c3 2 B c3
<b>Copy Selection with Column Header</b>	Copy all selected cells including column header onto the system clipboard.  <u>Example:</u> col1 col2 col3 1 A c3 2 B c3
<b>Copy Selection as Formatted Text</b>	Copy all selected cells including column header in fixed width columns onto the system clipboard.  <u>Example:</u> col1 col2 col3 ---- 1 A c3 2 B c3
<b>Copy Selection as Comma List</b>	Copy all selected cells onto the system clipboard, formatted as Comma Separated Values (CSV), one row for each column.  <u>Example:</u> 1, 2 A, B c3, c3
<b>Copy Selection as IN Clause</b>	Copy all selected cells onto the system clipboard, formatted as an IN clause.  <u>Example:</u> (col1 IN (1, 2)) and (col2 IN ('A', 'B')) and (col3 IN ('c3'))
<b>Copy Selection as IN Clause with AND (ALL)</b>	Copy all selected cells onto the system clipboard, formatted as a clause to select rows where ALL selected values match .  <u>Example:</u> (col1 = 1 and col2 = 'A' and col3 = 'c3') or (col1 = 2 and col2 = 'B' and col3 = 'c3')
	This is different from <b>Copy Selection as IN Clause with OR(ANY)</b> since the clause matches all column values on each row; it will <i>not</i> select a row with column values 1, B, c3.

<b>Copy Selection as IN Clause with OR (ANY)</b>	<p>Copy all selected cells onto the system clipboard, formatted as a clause to select rows where ANY of the selected values match.</p> <p><u>Example:</u>  <code>(col1 IN (1, 2))</code>  <code>and (col2 IN ('A', 'B'))</code>  <code>and (col3 = 'c3')</code></p> <p>This is similar to <b>Copy Selection as IN Clause</b>, but is slightly more efficient in some situations; in this example, col3 is matched using equals (=) rather than IN since the selected rows have the same value in this column.</p> <p>This is different from <b>Copy Selection as IN Clause with AND (ALL)</b> since the clause matches any column value on each row; it <i>will</i> select a row with column values 1, B, c3.</p>									
<b>Copy Selection as HTML Table</b>	<p>Copy all selected cells onto the system clipboard, formatted as <b>HTML</b> code. Paste this into an HTML-capable editor to get a nicely formatted table.</p> <p><u>Example:</u></p> <pre>col1 col2 col3 1    A    c3 2    B    c3</pre>									
<b>Copy Selection as JIRA Table</b>	<p>Copy all selected cells onto the system clipboard, formatted as markup code suitable for <b>JIRA</b> and similar tools. Paste this into a suitable editor to get a nicely formatted table.</p> <p><u>Example:</u></p> <table border="1" data-bbox="399 894 586 1045"> <thead> <tr> <th>col1</th> <th>col2</th> <th>col3</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>A</td> <td>c3</td> </tr> <tr> <td>2</td> <td>B</td> <td>c3</td> </tr> </tbody> </table>	col1	col2	col3	1	A	c3	2	B	c3
col1	col2	col3								
1	A	c3								
2	B	c3								
<b>Save Selected Cell</b>	<p>Save the value of the selected cell to a file, selected with a file chooser dialog</p>									
<b>Compare</b>	<p><a href="#">Compare</a> the data in this grid to the data in other open grids.</p>									
<b>Compare Selected Cells</b>	<p>Compare the data in the two cells as text</p>									
<b>Reset Grid</b>	<p>Reset the visual state of the grid (does not affect the column settings or the actual data in the grid).</p>									
<b>Browse Row in Window</b>	<p>Display all data for the selected row in a separate window.  <b>Note:</b> for a read/write grid, this entry is named <b>Edit Row in Window</b>.</p>									
<b>Browse Cell in Window</b>	<p>Display the cell value in a separate window. This is especially useful for BLOB/CLOB data.  <b>Note:</b> for a read/write grid, this entry is named <b>Edit Cell in Window</b>.</p>									
<b>Show in Navigator</b>	<p>Open the <a href="#">Navigator</a> tab with the current selections and sorting.</p>									
<b>Describe Data</b>	<p>Show detailed information about the columns in the grid.</p>									
<b>Aggregation Data for Selection</b>	<p>Displays aggregation data for the current selection. Read more in <a href="#">Aggregation Data for Selection</a> below.</p>									
<b>Generate Filter &amp; Sort</b>	<p>The operations in this submenu help you create common <a href="#">Where Filters</a>.</p>									
<b>Create Row Count Data Monitor</b>	<p><a href="#">Creates a monitor</a> for tracking the row count in the table over time.</p>									
<b>Create Row Count Diff Data Monitor</b>	<p><a href="#">Creates a monitor</a> for tracking the number of added or removed rows in the table over time.</p>									

## Script to SQL Commander

There are also a set of operations for generating SQL statements based on the current selection. Choosing any of these creates the appropriate SQL and then switches the view to a new **SQL Commander** tab. You must use these operations to edit table data in the DbVisualizer Free edition. With the DbVisualizer Pro edition, you can instead use [inline and form based editing](#).

Operation	SQL Example
<b>Script: SELECT ALL</b>	SELECT * FROM COUNTRY;
<b>Script: SELECT ALL WHERE</b>	select * FROM SAKILA.COUNTRY where COUNTRY = 'Brazil'
<b>Script: SELECT ALL WITH FILTER</b>	select * from SAKILA.COUNTRY where COUNTRY_ID = 15 // If this is the filter, see above
<b>Script: INSERT INTO TABLE</b>	insert into SAKILA.COUNTRY (COUNTRY_ID, COUNTRY) values (15, 'Brazil')
<b>Script: INSERT COPY INTO TABLE</b>	insert into SAKILA.COUNTRY (COUNTRY_ID, COUNTRY) values (15, 'Brazil')
<b>Script: UPDATE WHERE</b>	update SAKILA.COUNTRY set COUNTRY_ID = 15, COUNTRY = 'Brazil' where COUNTRY = 'Brazil'
<b>Script: DELETE WHERE</b>	delete from SAKILA.COUNTRY where COUNTRY = 'Brazil'

You can generate SQL with either static values as they appear in the grid, or with [DbVisualizer variables](#). A variable is essentially a placeholder for a value in an SQL statement. When the statement is executed, DbVisualizer locates all variables and presents them in a dialog where you can enter or modify values for the variables. DbVisualizer replaces the variable placeholders with the new values before executing the statement.

Variables are used in the generated SQL statements by default. You can disable the **Include Variables in SQL** setting in the Tool Properties dialog, in the **Table Data** category under the General tab, to use literal values are instead.

Here is an example of the SQL generated for **Script: SELECT ALL WHERE** with the **Include Variables in SQL** setting enabled, assuming the table is named SAKILA.COUNTRY and has a column named COUNTRY with the value 'Brazil' on the selected row:

```
SELECT * FROM COUNTRY WHERE COUNTRY = ${COUNTRY (where)}|Brazil|String|where ds=50 dt=VARCHAR}§;
```

And here is the same example with the **Include Variables in SQL** setting disabled:

```
SELECT * FROM COUNTRY WHERE COUNTRY = 'Brazil';
```

## Creating Monitors

A monitor in DbVisualizer is an SQL query executed at a specified frequency so you can track changes in data over time. The result can be viewed either as a grid or a graph. The right-click menu for the grid in the Data tab contains operations for creating two common types of monitors for the table: a **Row Count Data** monitor or a **Row Count Diff Data** monitor. The first tracks the number of rows in the table over time and the second tracks the number of added or removed rows over time. Please read more about monitors in [Monitoring Data Changes](#).

## Aggregation Data for Selection

Only in DbVisualizer Pro



This feature is only available in the DbVisualizer Pro edition.

The **Aggregation Data for Selection** feature presents aggregation data organized per data type on the current selection in a grid. It provides information about cells holding numbers, text, date/time information and more. The following is an example of what it shows:

Aggregation Data for Selection

<b>Cells Count</b>	184
Rows	46
Columns	4
<b>Text Count</b>	46
<a href="#">Shortest</a>	10
<a href="#">Avg Length</a>	14
<a href="#">Longest</a>	27
$\Sigma$ Total Length	678
<b>Number Count</b>	138
<a href="#">Min</a>	0.99
<a href="#">Avg</a>	2.93
<a href="#">Median</a>	2.99
<a href="#">Max</a>	7
$\Sigma$ Sum	404.54
$\sqrt{x}$ Std Deviation	2.00

Handle Numbers in Text Types as Numbers      Max Scale

Auto Update

With **Auto Update** checked, the data is updated automatically when you change the selection in the grid. For very large selections, you may prefer to disable this feature and instead click **Update** when you want to refresh the data. Click a link (blue underlined text) in the aggregation table to locate and highlight the actual value in the source data grid. The **Handle Number Values in Text Types as Numbers** setting simply treats all valid numbers in text data types as numbers and include them in the **Number Count** summary.