


## Table of Contents

<b>QUERIES</b>	<b>2</b>
QUERY BUILDER (AVAILABLE ONLY IN FULL VERSION)	6
<i>Working with Diagram Area</i>	7
<i>Setting Field Association</i>	8
<i>Setting Output Fields</i>	9
<i>Setting Criteria</i>	10
<i>Setting Grouping Criteria</i>	11
<i>Setting Sorting Criteria</i>	12
<i>Setting Limit Criteria (Available only for MySQL, PostgreSQL and SQLite)</i>	13
QUERY EDITOR	14
<i>Editor View and More Features</i>	16
Code-Completion (Available only in Full Version)	17
SQL Formatting (Available only in Full Version)	19
Code Folding	20
Brace Highlight	21
Find and Replace	22
Copy With Quotes	24
Zoom In/Zoom Out	25
QUERY RESULTS	26
<i>Query Profile and Status (Available only for MySQL)</i>	28
QUERY EXPLAIN	29
<i>Explain Plan for MySQL</i>	30
<i>Explain Plan for Oracle</i>	31
<i>Explain Plan for PostgreSQL</i>	32
<i>Explain Plan for SQLite</i>	33
<i>Explain Plan for SQL Server</i>	34
QUERY PARAMETERS	36
DEBUGGING ORACLE QUERY (AVAILABLE ONLY IN FULL VERSION)	37



## Queries

A query is used to extract data from the database in a readable format according to the user's request. Navicat provides two powerful tools for working with the SQL queries: Query Editor for editing the query text directly and Query Builder for building queries visually. You can save your queries for setting schedule.



Just simply click  to open an object pane for **Query**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit, open and delete the selected query.

### Create Query




To create a new query in Query Editor


- Select anywhere on the object pane.
- Click the  **New Query** from the object pane toolbar.  
or
- Right-click and select  **New Query** from the popup menu.
- Edit query text on the Query Editor tab.

To create a new query in Query Builder

- Select anywhere on the object pane.
- Click the  **New Query** from the object pane toolbar.  
or
- Right-click and select  **New Query** from the popup menu.
- Edit query on the Query Builder tab.

To create a new query with loading from a SQL file

- Select anywhere on the object pane.
- Click the  **New Query** from the object pane toolbar.  
or
- Right-click and select  **New Query** from the popup menu.
- Click  **Load**.

**Hint:** To create new query you can also right-click the Queries node of the navigation pane and select the  **New Query** from the popup menu.

To create a new query with the same properties as one of the existing queries has (using drag and drop method)




**Apply to:** current database {same connection}

- Select the query(s) for copying in the navigation pane/object pane.
- Right-click and drag the chosen query(s) to the target location.
- Select one of the following options:
  - Copy here
  - Move here
  - Cancel
- The newly created query(s) will be named as "queryname\_**copy**"

**Apply to:** different database {same connection}  
different database {different connection}

- Select the query(s) for copying in the object pane.
- Drag and drop the chosen query(s) to the target database.
- Select one of the following options:
  - Copy here
  - Move here
  - Cancel



To create a new query with modification as one of the existing queries

- Select the query for modifying in the navigation pane/object pane.
- Right-click and select the  **Design Query** from the popup menu.  
or
- Click the  **Design Query** from the object pane toolbar.
- Modify query on the Query Editor/Query Builder tab.
- Click  **Save As**.

**Hint:** Queries(.sql) are saved under the Settings Save Path.

## Edit Query

To edit the existing query



- Select the query for editing in the navigation pane/object pane.
- Right-click and select the  **Design Query** from the popup menu.  
or
- Click the  **Design Query** from the object pane toolbar.
- Modify query on the Query Editor/Query Builder tab.

To change the name of the query

- Select the query for editing in the navigation pane/object pane.
- Right-click and select the **Rename** from the popup menu.

## Open Query



To open a query (manage query data)

- Select the query for opening in the navigation pane/object pane.
- Right-click and select the  **Open Query** from the popup menu or simply double-click the query.  
or
- Click the  **Open Query** from the object pane toolbar.

**Note:** Only *SELECT* queries will be run automatically with results being displayed on Result tab.

## Run Query

To run a query



- Create a new query/open the existing query.
- Click  **Run**. (Click  **Stop** to terminate the running)
- View/edit the returned data on the Result tab.

To run a saved query from the command line (set schedule)

- Create and save the query.
- In terminal, type the command (see Command for details)

## Delete Query

To delete a query

- Select the query for deleting in the navigation pane/object pane.
- Right-click and select the  **Delete Query** from the popup menu.  
or
- Click the  **Delete Query** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Query Information

To achieve a query information

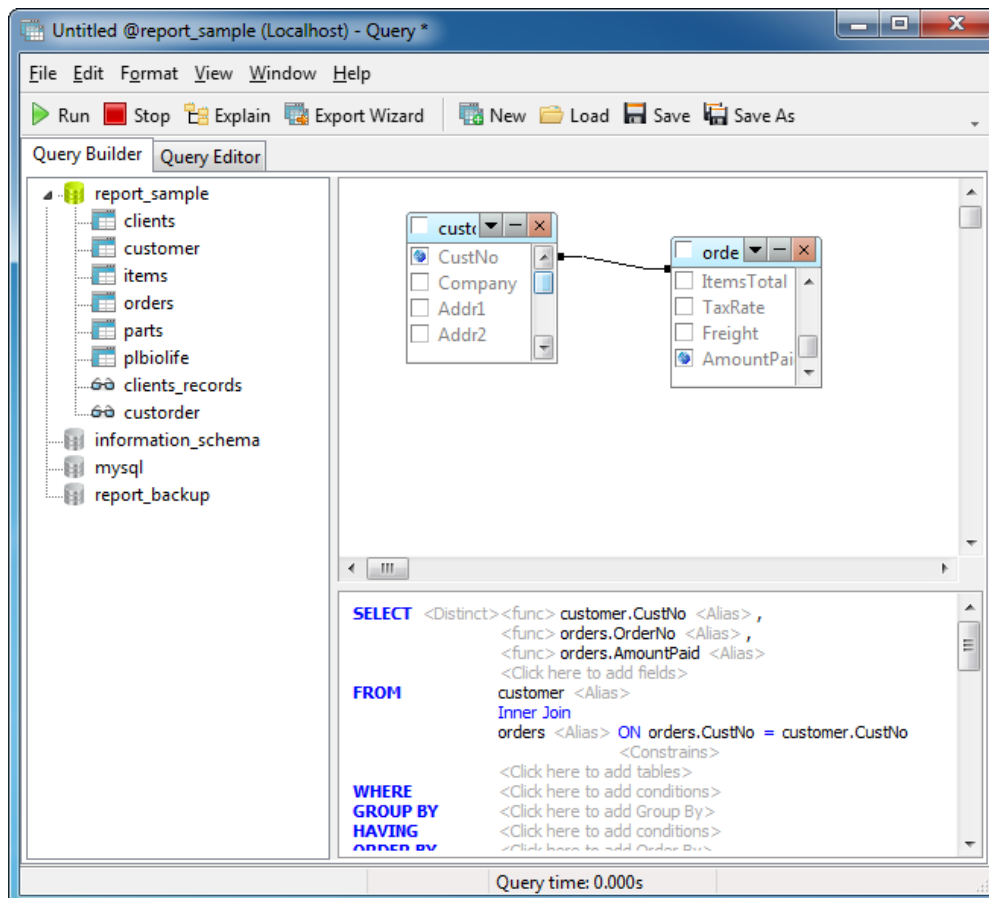
- Select the query in the navigation pane/object pane.
- Right-click the selected query and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## Query Builder (Available only in Full Version)

Navicat provides a useful tool called **Visual Builder** for building queries visually. It allows you to create and edit queries without knowledge of SQL. The database objects are displayed in left panel. Whereas in the right panel, it is divided into two portions: the upper **Graphical View**, and the lower **Syntax View**.

**Note:** Visual Builder supports *SELECT* statement only. Use Query Editor for creating complex queries.

- [Working with Diagram Area](#)
- [Setting Field Association](#)
- [Setting Output Fields](#)
- [Setting Criteria](#)
- [Setting Grouping Criteria](#)
- [Setting Sorting Criteria](#)
- [Setting Limit Criteria](#) (Available only for MySQL, PostgreSQL and SQLite)



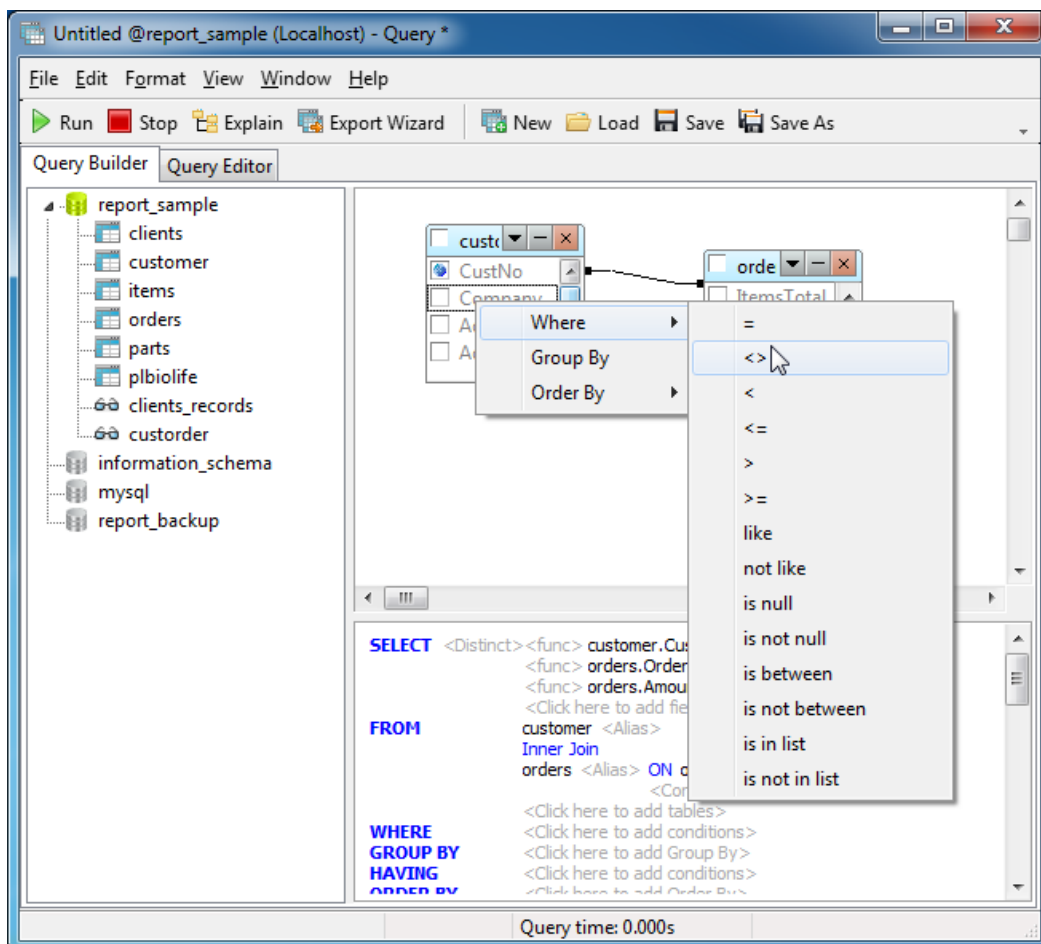
## Working with Diagram Area

To add a table to the query, simply drag it or double-click from the left panel to the Graphical View area. To include a table field in the query, check the left of the field name in the list. To include all the fields, click at the left of the table caption.

To remove the object from the Graphical View area, click the cross button at the object caption.

To add the table alias, simply double-click the table name and enter the alias in the the Graphical View area.

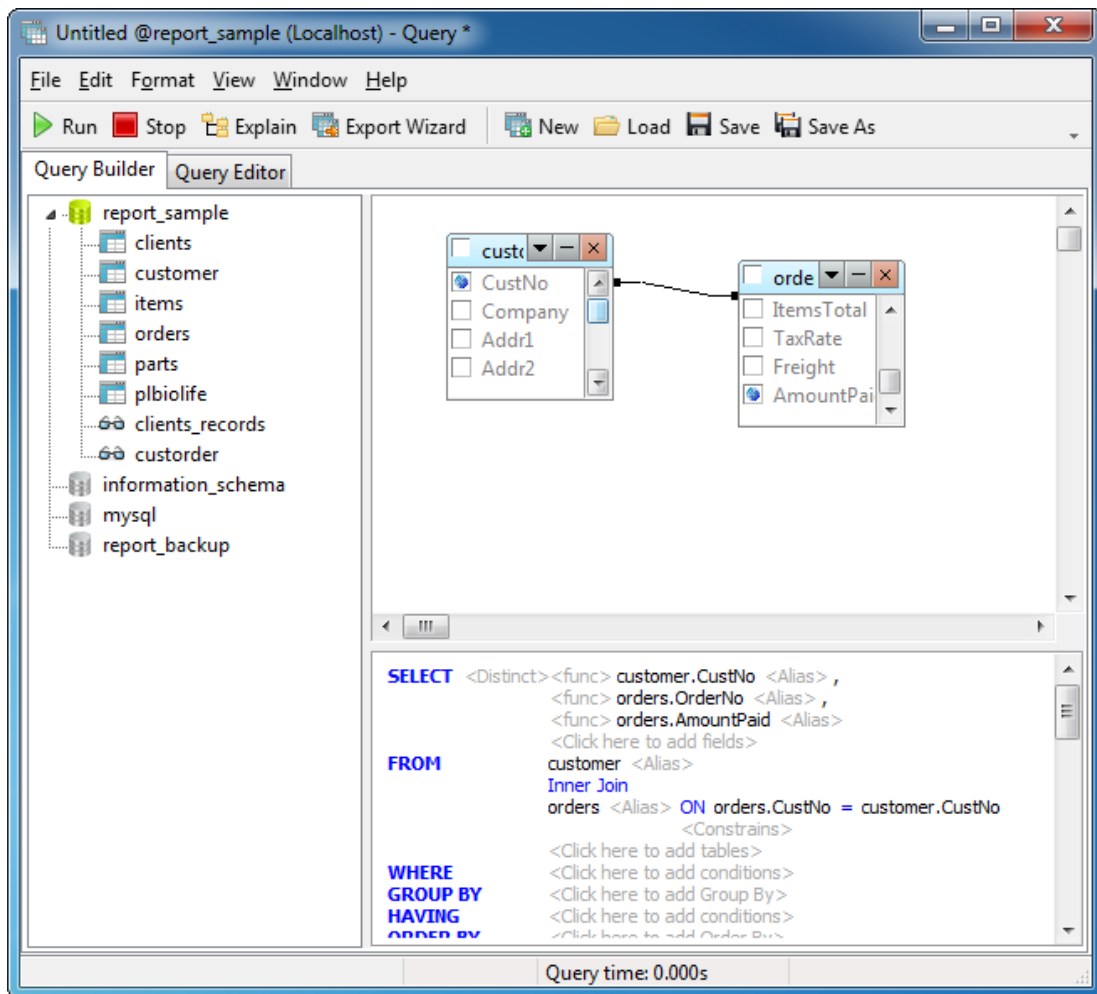
**Hint:** You are also allowed to set criteria by right-clicking any fields from the Graphical View area.



## Setting Field Association

To associate database objects by two fields, just drag one field from the object list to another and a line will appear between the linked fields.

**Hint:** To delete all the links of some object, click button '-' next to the object alias.



## Setting Output Fields

The fields you have selected in the graphical view will be displayed in the Syntax View which allows you to set their displaying order and modify the output fields of the query using **<Distinct>**, **<func>** and **<Alias>**.

### <Distinct>

Enable this option if you wish the repeated records are not included into the query result.

### <func>

Set the aggregate functions (SUM, MAN, MIX, AVG, COUNT) for each field.

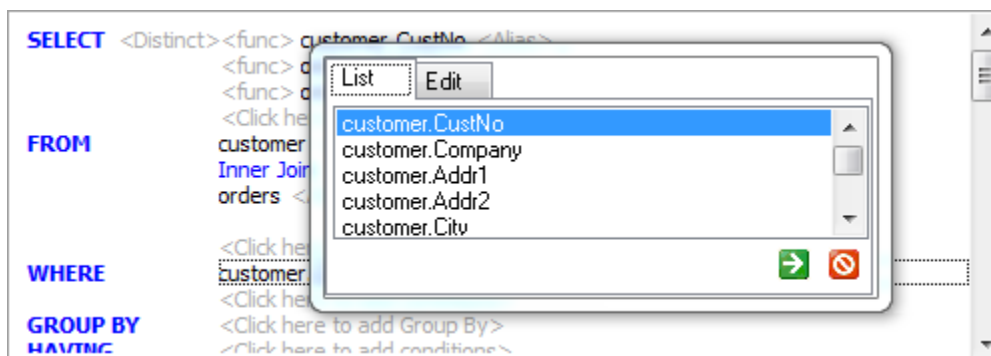
### <Alias>

Change the output query field name.

```
SELECT <Distinct><func> customer.CustNo <Alias> ,
      <func> orders.OrderNo <Alias> ,
      Sum(orders.AmountPaid) <Alias>
      <Click here to add fields>
FROM   customer <Alias>
      Inner Join
      orders <Alias> ON orders.CustNo = customer.CustNo
      <Constrains>
      <Click here to add tables>
WHERE  <Click here to add conditions>
GROUP BY customer.CustNo,
          orders.OrderNo
      <Click here to add Group By>
```

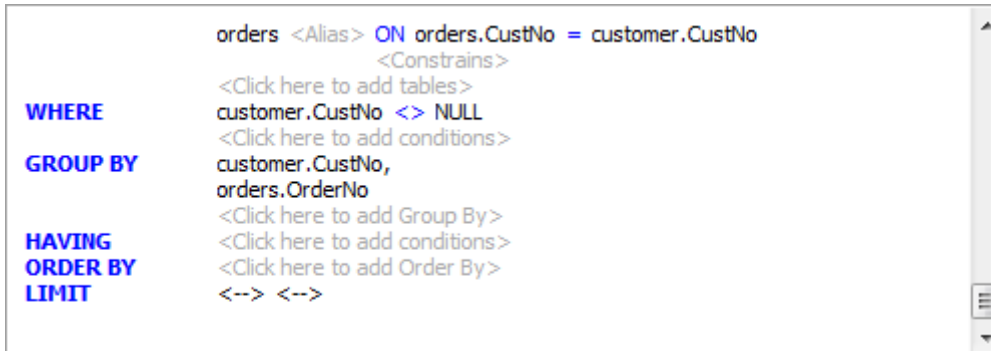
## Setting Criteria

To add a condition, click the  $\leftarrow\rightarrow = \leftarrow\rightarrow$  from the **Where** Clause in the Syntax View. Click  $\leftarrow\rightarrow$  to choose the field from the list of all the table fields, available in the query. To define your own criteria, type your values directly in the Edit Tab. Clicking = to set condition operator.



## Setting Grouping Criteria

You can set the conditions for grouping query records from the **Group By** Clause in the Syntax View. They are set in the same way as setting criteria. The conditions will be included into the **HAVING** statement of the current query.



```
orders <Alias> ON orders.CustNo = customer.CustNo
    <Constrains>
WHERE customer.CustNo <> NULL
    <Click here to add conditions>
GROUP BY customer.CustNo,
orders.OrderNo
    <Click here to add Group By>
HAVING <Click here to add conditions>
ORDER BY <Click here to add Order By>
LIMIT <--> <-->
```

## Setting Sorting Criteria

When you query your database/schema, you can sort the results by any field in an ascending or descending order by just adding *ORDER BY* at the end of your query.

In Visual Builder, you can set the way of sorting query records from the **Order By** Clause in the Syntax View. To change the sorting direction, click on either **ASC** or **DESC**.



```
orders <Alias> ON orders.CustNo = customer.CustNo
    <Constrains>
    <Click here to add tables>
WHERE customer.CustNo <> NULL
    <Click here to add conditions>
GROUP BY customer.CustNo,
    orders.OrderNo
    <Click here to add Group By>
HAVING <Click here to add conditions>
ORDER BY customer.CustNo ASC
    <Click here to add Order By>
LIMIT <--> <-->
```

## Setting Limit Criteria (Available only for MySQL, PostgreSQL and SQLite)

**Limit** Clause is used to limit your query results to those that fall within a specified range. You can use it to show the first X number of results, or to show a range from X - Y results. It is phrased as Limit X, Y and included at the end of your query. X is the starting point (remember the first record is 0) and Y is the duration (how many records to display).

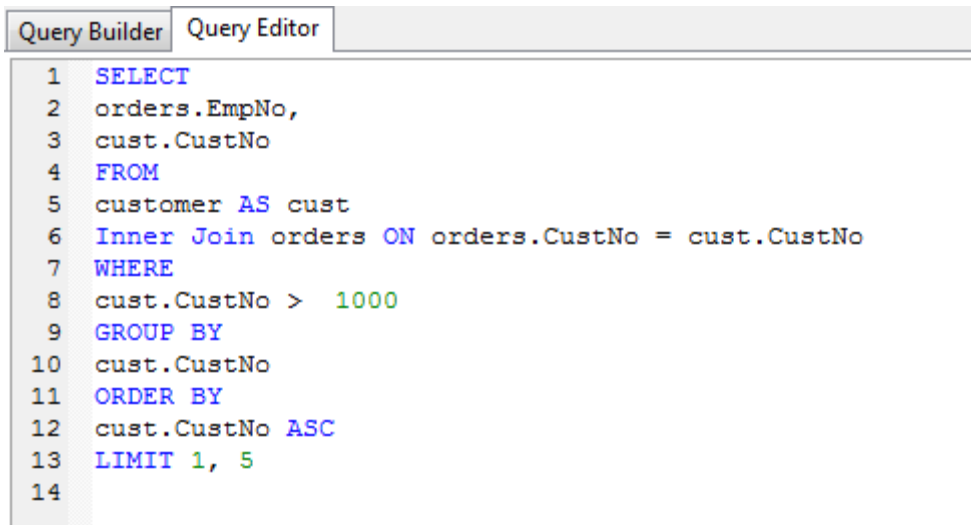
```
orders <Alias> ON orders.CustNo = customer.CustNo
    <Constrains>
    <Click here to add tables>
WHERE customer.CustNo <> NULL
    <Click here to add conditions>
GROUP BY customer.CustNo,
    orders.OrderNo
    <Click here to add Group By>
HAVING <Click here to add conditions>
ORDER BY customer.CustNo ASC
    <Click here to add Order By>
LIMIT 1 , 6
```

## Query Editor

Navicat provides a useful tool called **Query Editor** for creating and executing queries. It allows you to create and edit SQL text for a query, prepare and execute selected queries.

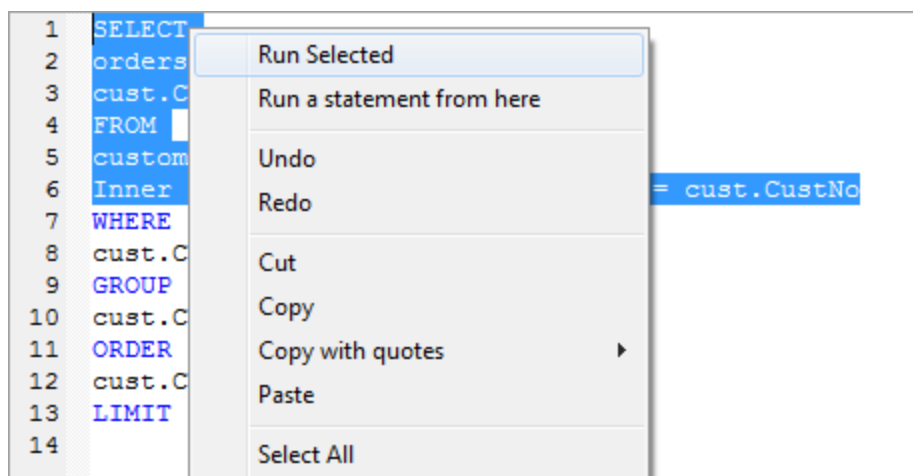
You can show the object tree, simply choose View -> **Show Object Tree**.

**Hint:** Query text will be automatically generated while you build in Query Builder.



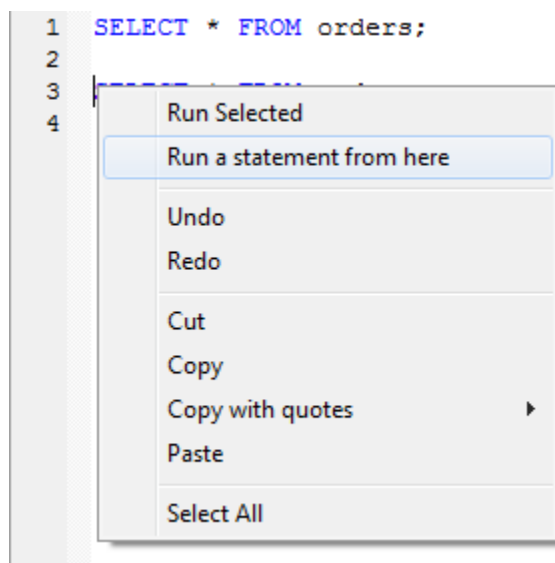
```
Query Builder Query Editor
1 SELECT
2 orders.EmpNo,
3 cust.CustNo
4 FROM
5 customer AS cust
6 Inner Join orders ON orders.CustNo = cust.CustNo
7 WHERE
8 cust.CustNo > 1000
9 GROUP BY
10 cust.CustNo
11 ORDER BY
12 cust.CustNo ASC
13 LIMIT 1, 5
14
```

You are allowed to run selected portion of query, just simply right-click the highlighted query and select **Run Selected**.



You can define multiple SQL statements in one Editor window, and the editor let you run the current statement your cursor is on (place your cursor in the front of the desired statement). Just simply select **Run a statement from here** or press **F7**.

**Note:** Select **Run a statement from here** or press **F7**, the next statement will be continue to run.



## Editor View and More Features

Navicat allows you to customize the view of the **SQL Editor** and provides a wide range of compelling code editing capabilities, smart code-completion, sql formatting, and more.

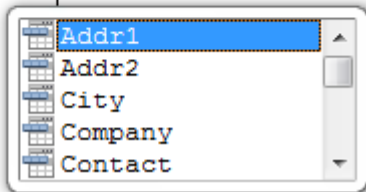
- [Code-Completion](#)
- [SQL Formatting](#)
- [Code Folding](#)
- [Brace Highlight](#)
- [Find and Replace](#)
- [Copy With Quotes](#)
- [Zoom In/Zoom Out](#)

## Code-Completion (Available only in Full Version)

Code-completion in Navicat displays information in drop-down lists as you type your SQL statement in the editor, it assists you with statement completion and the available properties of database objects, for example databases, tables, fields, views etc with their appropriate icons.

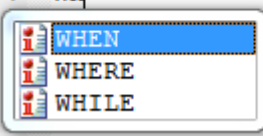
To activate the code-completion, just simply press '.' for the available properties of database object currently in the scope.

```
1 SELECT
2 orders.EmpNo,
3 cust.CustNo
4 FROM
5 customer AS cust
6 Inner Join orders ON orders.CustNo = cust.CustNo
7 WHERE
8 cust.CustNo > 1000
9 GROUP BY
10 cust.CustNo
11 ORDER BY
12 cust.
13
```



**Hint:** You may invoke code-completion by typing two characters or pressing Ctrl+SPACE on your keyboard for SQL keywords.

```
1 SELECT
2 orders.EmpNo,
3 cust.CustNo
4 FROM
5 customer AS cust
6 Inner Join orders ON orders.CustNo = cust.CustNo
7 WH
```



**Hint:** Smart code-completion will pop-up a list of variants for the word completion automatically.

**Note:** Code-completion can be also applied on View, Functions/Procedures, etc.

## SQL Formatting (Available only in Full Version)

To change the SQL statement format, simply choose from the **Edit** menu -

### **Indent**

Increase/decrease indent for the selected lines of codes.

### **Comment**

Comment/uncomment the selected lines of codes.

### **Convert Case**

Format the selected codes into upper/lower case.

### **Beautify SQL**

Format the selected codes with the Beautify SQL settings.

### **Beautify SQL With...**

Change the Beautify SQL settings.

#### **Use tab character**

Check this option to use tab character.

#### **Tab size**

Set the tab size.

#### **Short brace length**

Set the length of the short brace.

#### **Upper case keywords**

Format all the SQL keywords to upper case.

#### **Save settings**





Save the SQL beautify options settings after you click **Beautify** button.






### **Minify SQL**

Minify the format of the SQL in the SQL Editor.

## Code Folding

Code folding feature enables you to collapse blocks of code such that only the first line of the block appears in **Editor**.

A block of code that can be folded is indicated by an icon  to the left of the first line of the block. A vertical line extends from the icon to the bottom of the foldable code. In contrast, a folded block of code is indicated by an icon  to left of the code block. You can fold the block by clicking  or expand it by clicking  in **Editor**.

```
1 CREATE OR REPLACE
2  PROCEDURE SETEMVIEWUSERCONTEXT(...)
7 IS
8  BEGIN
9
10  IF...END IF;
13
14 MGMT_VIEW_PRIV.SET_VIEW_USER_CONTEXT(
    em_client_in, op_in, view_user, view_pwd);
15
16  IF time_coff IS NOT NULL THEN
17     MGMT_TIME_SYNC.setTimeCoffNoCheck(null);
18 END IF;
19
20 EXCEPTION
21     WHEN OTHERS THEN
22  IF time_coff IS NOT NULL THEN
23     MGMT_TIME_SYNC.setTimeCoffNoCheck(null
24 );
25     END IF;
26     RAISE;
27 END;
```

## Brace Highlight

Navicat supports to highlight the matching brace in the editor, i.e. (), Begin...End .

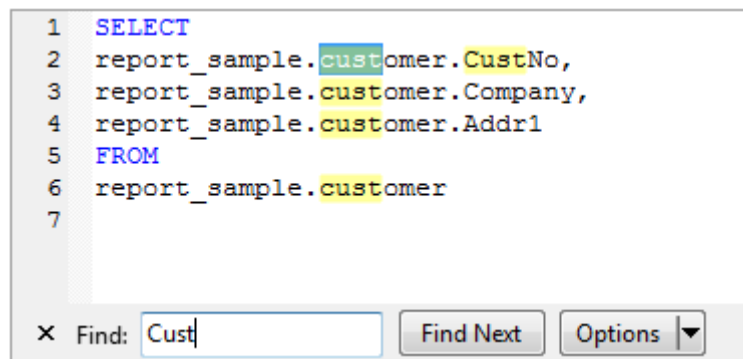
**Note:** The cursor must be on a brace to show the highlight.

```
1 CREATE OR REPLACE
2 PROCEDURE SETEMVIEWUSERCONTEXT(em_client_in
3 IN VARCHAR2,
4     op_in          IN INTEGER,
5     view_user      IN VARCHAR2,
6     view_pwd       IN VARCHAR2,
7     time_coff      IN VARCHAR2 DEFAULT NULL)
8 IS
9 BEGIN...END;
27
```

## Find and Replace

### Find

The **Find** Dialog is provided for quick searching for the text in the editor window. Just simply click **Edit** -> **Find** from the menu or press Ctrl+F and enter a search string.



The search starts at the cursor's current position to the end of the file. There will not have differentiates when performing a uppercase or lowercase search.

To find for the next text, just simply select **Edit** -> **Find Next** or press F3.

### Replace

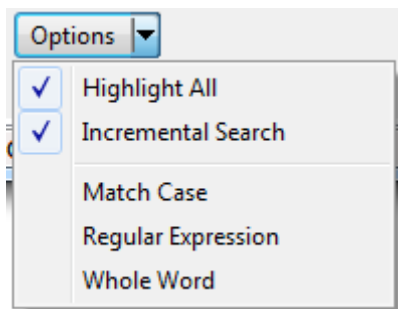
To open the **Replace** Dialog, simply click **Edit** -> **Replace** from the menu and enter the text you want to search and replace.

Click **Replace** button to replace the first occurrence.

Click **Replace All** button to replace all occurrences automatically .



There are some additional **Options** for Find and Replace:



## Copy With Quotes

To copy the SQL statement with quotes, just simply right-click the highlighted SQL. Then, select **Copy with quotes** and choose the format.

**Note:** Only available in Query, View and Materialized View.

## Zoom In/Zoom Out


Navicat has the ability to zoom in or zoom out the SQL in the editor. The zooming options are available from the **View** menu. The same effect can be achieved with keyboard shortcuts.

- Zoom In: [Ctrl + =]
- Zoom Out: [Ctrl + -]
- Reset: [Ctrl + 0]

**Hint:** Range from -10 to +20.

**Note:** Files are opened in different tabs will not be effected by the zoom.

## Query Results

To run the query click  **Run** on the toolbar. If the query statement is correct, the query executes and, if the query statement is supposed to return data, the **Result** tab opens with the data returned by the query. If an error occurs while executing the query, execution stops, the appropriate error message is displayed.

The **Result** tab displays the result data, returned by the query, as a grid. Data can be displayed in three modes: **Grid View**, **Form View** and **Text/Blob View**. See Data View for details.

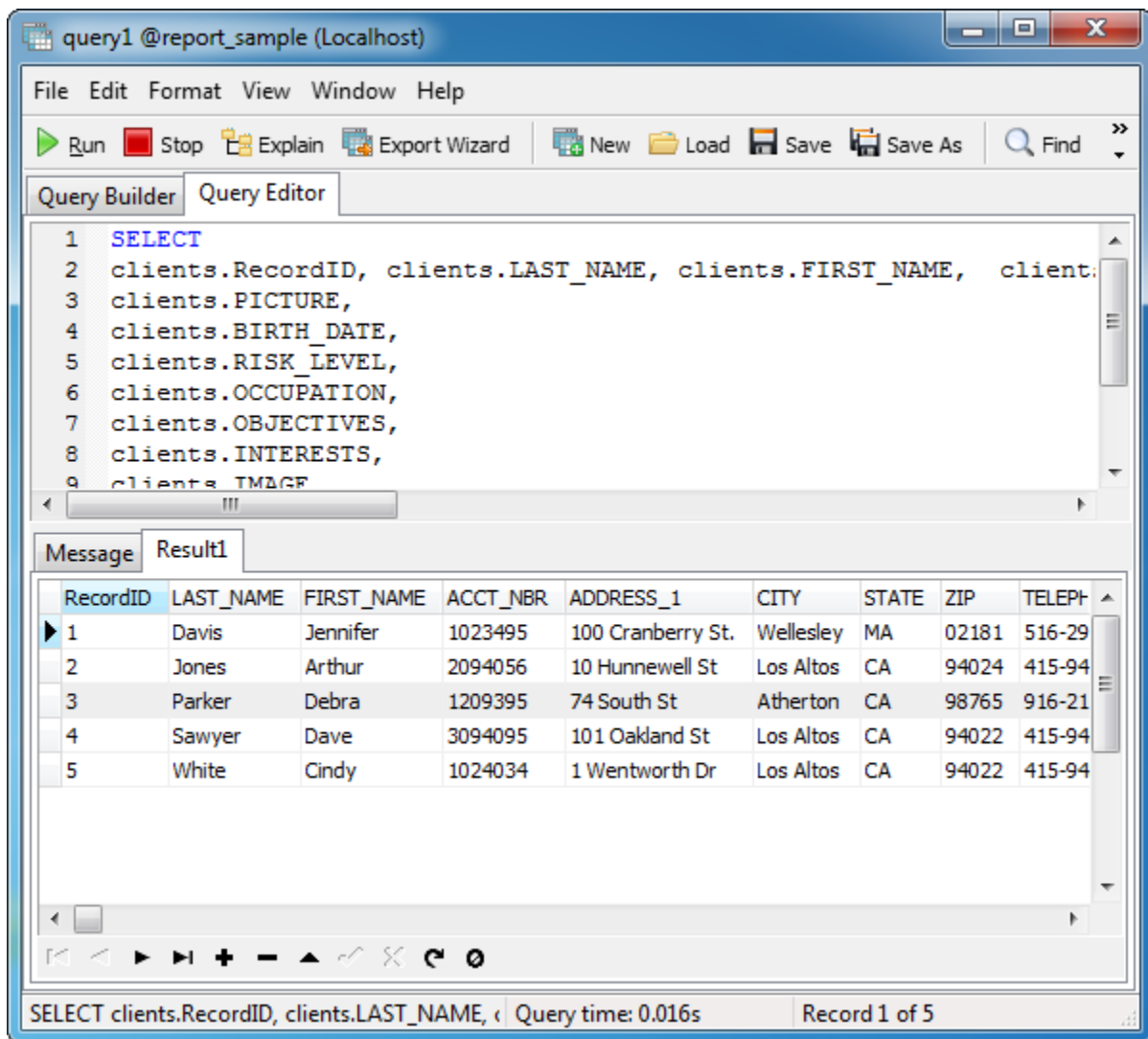
**Hint:** Navicat supports to return more than one resultset.

**Note:** You can choose to show the Result tab below the query editor or in a new tab by selecting View -> Show Result -> **Below Query Editor** or **In a New Tab**.

The toolbars of Query Result Viewer provides the following functions for managing data:

- Export Query Results  
Export data to MS Word, MS Excel, MS Access, TXT, DBF, HTML, SQL, RTF and more.
- Edit TEXT/BLOB/BFile  
Allow you to view and edit the content of TEXT/BLOB/BFile fields.

**Note:** Only Oracle supports BFile.



The screenshot shows the Navicat Query Editor window titled "query1 @report\_sample (Localhost)". The menu bar includes File, Edit, Format, View, Window, and Help. The toolbar contains icons for Run, Stop, Explain, Export Wizard, New, Load, Save, Save As, and Find. The Query Editor tab is active, displaying the following SQL query:


```
1 SELECT
2 clients.RecordID, clients.LAST_NAME, clients.FIRST_NAME, client:
3 clients.PICTURE,
4 clients.BIRTH_DATE,
5 clients.RISK_LEVEL,
6 clients.OCCUPATION,
7 clients.OBJECTIVES,
8 clients.INTERESTS,
9 clients.IMAGE
```

Below the query editor, the "Result1" tab is active, showing a table with 9 columns: RecordID, LAST\_NAME, FIRST\_NAME, ACCT\_NBR, ADDRESS\_1, CITY, STATE, ZIP, and TELEPH. The table contains 5 rows of data:

RecordID	LAST_NAME	FIRST_NAME	ACCT_NBR	ADDRESS_1	CITY	STATE	ZIP	TELEPH
1	Davis	Jennifer	1023495	100 Cranberry St.	Wellesley	MA	02181	516-29
2	Jones	Arthur	2094056	10 Hunnewell St	Los Altos	CA	94024	415-94
3	Parker	Debra	1209395	74 South St	Atherton	CA	98765	916-21
4	Sawyer	Dave	3094095	101 Oakland St	Los Altos	CA	94022	415-94
5	White	Cindy	1024034	1 Wentworth Dr	Los Altos	CA	94022	415-94

At the bottom of the window, the status bar shows the query text: "SELECT clients.RecordID, clients.LAST\_NAME, ..." and the execution details: "Query time: 0.016s" and "Record 1 of 5".

## Query Profile and Status (Available only for MySQL)

To show the profile and status when running the query, simply choose View -> **Show Profile and Status** and click  **Run** on the toolbar.

The **Profile** tab displays the query profile: Table lock, System lock, Statistic, etc.


**Note:** For MySQL 5.0, support from 5.0.37 or above.

For MySQL 5.1, support from 5.1.24 or above.

For MySQL 6.0, support from 6.0.5 or above.

The **Status** tab displays the query status: Bytes received, Bytes sent, etc.

## Query Explain

To show the Explain Plan of the query, just simply click  **Explain**.

- [Explain Plan for MySQL](#)
- [Explain Plan for Oracle](#)
- [Explain Plan for PostgreSQL](#)
- [Explain Plan for SQLite](#)
- [Explain Plan for SQL Server](#)

## Explain Plan for MySQL

The **Explain** tab displays the information as a grid:

Column	Description
id	The SELECT identifier. This is the sequential number of the SELECT within the query.
select_type	The type of SELECT, which can be SIMPLE, PRIMARY, UNION, DEPENDENT UNION, UNION RESULT, SUBQUERY, DEPENDENT SUBQUERY, DERIVED, UNCACHEABLE SUBQUERY or UNCACHEABLE UNION.
table	The table to which the row of output refers.
type	The join type.
possible_keys	The possible_keys column indicates which indexes MySQL can choose from use to find the rows in this table.
key	The key column indicates the key (index) that MySQL actually decided to use.
key_len	The key_len column indicates the length of the key that MySQL decided to use.
ref	The ref column shows which columns or constants are compared to the index named in the key column to select rows from the table.
rows	The rows column indicates the number of rows MySQL believes it must examine to execute the query.
filtered	The filtered column indicates an estimated percentage of table rows that will be filtered by the table condition.
Extra	This column contains additional information about how MySQL resolves the query.

## Explain Plan for Oracle

The **Explain** tab displays the data in the Oracle PLAN\_TABLE as a grid:

Column	Description
Operation	Name of the internal operation performed in this step.
Object	Name of the table or index.
Optimizer	Current mode of the optimizer.
Cost	Cost of the operation as estimated by the optimizer's query approach. Cost is not determined for table access operations. The value of this column does not have any particular unit of measurement; it is merely a weighted value used to compare costs of execution plans. The value of this column is a function of the CPU_COST and IO_COST columns.
Cardinality	Estimate by the query optimization approach of the number of rows accessed by the operation.
Bytes	Estimate by the query optimization approach of the number of bytes accessed by the operation.
Partition Start	Start partition of a range of accessed partitions.
Partition ID	Step that has computed the pair of values of the PARTITION_START and PARTITION_STOP columns.
Access Predicates	Predicates used to locate rows in an access structure. For example, start or stop predicates for an index range scan.
Filter Predicates	Predicates used to filter rows before producing them.

## Explain Plan for PostgreSQL

The **Message** tab displays the query plan.

## Explain Plan for SQLite

The **Explain** tab displays the query plan as a grid.

## Explain Plan for SQL Server

The **Explain** tab displays detailed information about how the statements are executed and provides estimates of the resource requirements for the statements.

Column	Description
StmtText	For rows that are not of type PLAN_ROW, this column contains the text of the Transact-SQL statement. For rows of type PLAN_ROW, this column contains a description of the operation. This column contains the physical operator and may optionally also contain the logical operator. This column may also be followed by a description that is determined by the physical operator.
StmtId	Number of the statement in the current batch.
NodeId	ID of the node in the current query.
Parent	Node ID of the parent step.
PhysicalOp	Physical implementation algorithm for the node. For rows of type PLAN_ROWS only.
LogicalOp	Relational algebraic operator this node represents. For rows of type PLAN_ROWS only.
Argument	Provides supplemental information about the operation being performed. The contents of this column depend on the physical operator.
DefinedValues	Contains a comma-separated list of values introduced by this operator. These values may be computed expressions which were present in the current query (for example, in the SELECT list or WHERE clause), or internal values introduced by the query processor in order to process this query. These defined values may then be referenced elsewhere within this query. For rows of type PLAN_ROWS only.
EstimateRows	Estimated number of rows of output produced by this operator. For rows of type PLAN_ROWS only.
EstimateIO	Estimated I/O cost* for this operator. For rows of type PLAN_ROWS only.
EstimateCPU	Estimated CPU cost* for this operator. For rows of type PLAN_ROWS only.

AvgRowSize	Estimated average row size (in bytes) of the row being passed through this operator.
TotalSubtreeCost	Estimated (cumulative) cost* of this operation and all child operations.
OutputList	Contains a comma-separated list of columns being projected by the current operation.
Warnings	Contains a comma-separated list of warning messages relating to the current operation. Warning messages may include the string "NO STATS:()" with a list of columns. This warning message means that the query optimizer attempted to make a decision based on the statistics for this column, but none were available. Consequently, the query optimizer had to make a guess, which may have resulted in the selection of an inefficient query plan.
Type	Node type. For the parent node of each query, this is the Transact-SQL statement type (for example, SELECT, INSERT, EXECUTE, and so on). For subnodes representing execution plans, the type is PLAN_ROW.
Parallel	0 = Operator is not running in parallel. 1 = Operator is running in parallel.
EstimateExecutions	Estimated number of times this operator will be executed while running the current query.

\* Cost units are based on an internal measurement of time, not wall-clock time. They are used for determining the relative cost of a plan in comparison to other plans.

## Query Parameters

Query Builder and Query Editor both support using of parameters inside the query text. You can set query parameters to add variable values to a query each time you run it. The parameter should appear as an identifier with **\$** at its beginning, quote with **[ ]**, e.g. `[$any_name]`.

Execute the query and the **Input Parameter** Dialog is provided for you to enter the desired data you wish to search.

## **Debugging Oracle Query (Available only in Full Version)**

To debug the Oracle query click  **Debug** on the toolbar to launch the Oracle Debugger.

Enter the parameter(s) if the query has input parameter(s).