

## Table of Contents

<b>ORACLE DATABASE OBJECT MANAGEMENT</b>	<b>5</b>
ORACLE DEBUGGER (AVAILABLE ONLY IN FULL VERSION)	6
<i>Oracle Debugger Code Window</i>	7
<i>Oracle Debugger Views</i>	8
Repositioning Oracle Debugger Views	9
Oracle Debugger Breakpoints	10
Oracle Debugger Call Stack	11
Oracle Debugger DBMS Output	12
Oracle Debugger Log	13
Oracle Debugger Smart Data	14
Oracle Debugger Watch List	15
<i>Oracle Debugger Toolbar</i>	16
EDITING ORACLE PHYSICAL ATTRIBUTES/DEFAULT STORAGE CHARACTERISTICS	17
ORACLE TABLES	20
<i>Oracle Normal Tables</i>	25
Table Designer for Oracle Normal Tables (/Index Organized Tables)	26
Oracle Table Fields	27
Setting Oracle Table Field Properties	29
Setting Other Oracle Table Field Properties	32
Oracle Table Indexes	34
Setting Oracle Table Index Properties	36
Oracle Table Foreign Keys	38
Setting Oracle Table Foreign Key Properties	39
Oracle Table Uniques	40
Oracle Table Setting Unique Properties	41
Oracle Table Checks	42
Setting Oracle Table Check Properties	43
Oracle Table Triggers	44
Setting Oracle Table Trigger Properties	45
Oracle Table Options	47
<i>Oracle External Tables</i>	48
Table Designer for Oracle External Tables	49
Fields for Oracle External Tables	50
Setting Field Properties for Oracle External Tables	52
Setting Other Field Properties for Oracle External Tables	55
External Properties for Oracle External Tables	56
Access Parameters for Oracle External Tables	57

<i>Oracle Index Organized Tables</i>	58
Table Designer for Oracle Index Organized Tables	59
Options for Oracle Index Organized Tables	60
ORACLE VIEWS	62
<i>Oracle View Designer</i>	66
Working with Oracle View Builder (Available only in Full Version)	67
Editing Oracle View SQL Definition	68
Setting Advanced Oracle View Properties	69
Oracle View Preview	70
Oracle View Explain	71
<i>Oracle View Viewer</i>	72
ORACLE FUNCTIONS/PROCEDURES	74
<i>Oracle Function Wizard</i>	78
Setting Oracle Routine Type	79
Setting Parameters for Oracle Procedure/Function	80
Setting Return Type for Oracle Function	81
<i>Oracle Function/Procedure Designer</i>	82
Editing Oracle Function/Procedure Definition	83
Viewing Oracle Function/Procedure Result	84
Debugging Oracle Function/Procedure (Available only in Full Version)	85
ORACLE DATABASE LINKS	86
<i>Oracle Database Link Designer</i>	88
Edit Oracle Database Link General	89
ORACLE INDEXES	90
<i>Oracle Index Designer</i>	92
Editing Oracle Index General	93
Oracle Normal Index	94
Oracle Unique Index	95
Oracle Bitmap Index	96
Oracle Domain Index	97
Oracle Cluster Index	98
Editing Advanced Oracle Index Properties	99
ORACLE JAVA	101
<i>Oracle Java Designer</i>	103
Editing Oracle Java General	104
Oracle Java Source	105
Oracle Java Class	106
Oracle Java Resource	107

Setting Advanced Oracle Java Properties	108
ORACLE MATERIALIZED VIEWS	109
<i>Oracle Materialized View Designer</i>	112
Working with Oracle Materialized View Builder (Available only in Full Version)	113
Editing Oracle Materialized View SQL Definition	114
Setting Advanced Oracle Materialized View Properties	115
<i>Oracle Materialized View Preview</i>	119
<i>Oracle Materialized View Explain</i>	120
ORACLE MATERIALIZED VIEW LOGS	121
<i>Oracle Materialized View Log Designer</i>	124
Editing Oracle Materialized View Log General	125
ORACLE PACKAGES	127
<i>Oracle Package Designer</i>	130
Editing Oracle Package Definition	131
<i>Oracle Package Body Designer</i>	132
Editing Oracle Package Body Definition	133
<i>Viewing Oracle Package Result</i>	134
<i>Debugging Oracle Package (Available only in Full Version)</i>	135
ORACLE SEQUENCES	136
<i>Oracle Sequence Designer</i>	138
Editing Oracle Sequence General	139
ORACLE SYNONYMS	140
<i>Oracle Synonym Designer</i>	142
Editing Oracle Synonym General	143
ORACLE TRIGGERS	144
<i>Oracle Trigger Designer</i>	147
Editing Oracle Trigger General	148
Oracle Table Trigger	149
Oracle View Trigger	151
Oracle Schema Trigger	152
Oracle Database Trigger	153
Setting Advanced Oracle Trigger Properties	154
Editing Oracle Trigger Definition	155
ORACLE TYPES	156
<i>Oracle Object Type Designer</i>	160
Editing Oracle Object Type Definition	161
<i>Oracle Type Body Designer</i>	162
Editing Oracle Type Body Definition	163

<i>Oracle Collection Type Designer</i>	164
Editing Oracle Collection Type General	165
ORACLE XML SCHEMAS	166
<i>Oracle XML Schema Designer</i>	168
Editing Oracle XML Schema Doc	169
Setting Advanced Oracle XML Schema Properties	170
ORACLE RECYCLE BIN	172
ORACLE DIRECTORIES	174
<i>Oracle Directory Designer</i>	176
Editing Oracle Directory General	177
ORACLE TABLESPACES	178
<i>Oracle Tablespace Designer</i>	180
Editing Oracle Tablespace General	181
Editing Oracle Tablespace Storage	183
Setting Advanced Oracle Tablespace Properties	185
ORACLE PUBLIC DATABASE LINKS	186
ORACLE PUBLIC SYNONYMS	187

## Oracle Database Object Management

The following list contains the most common Oracle database objects supported by Navicat.

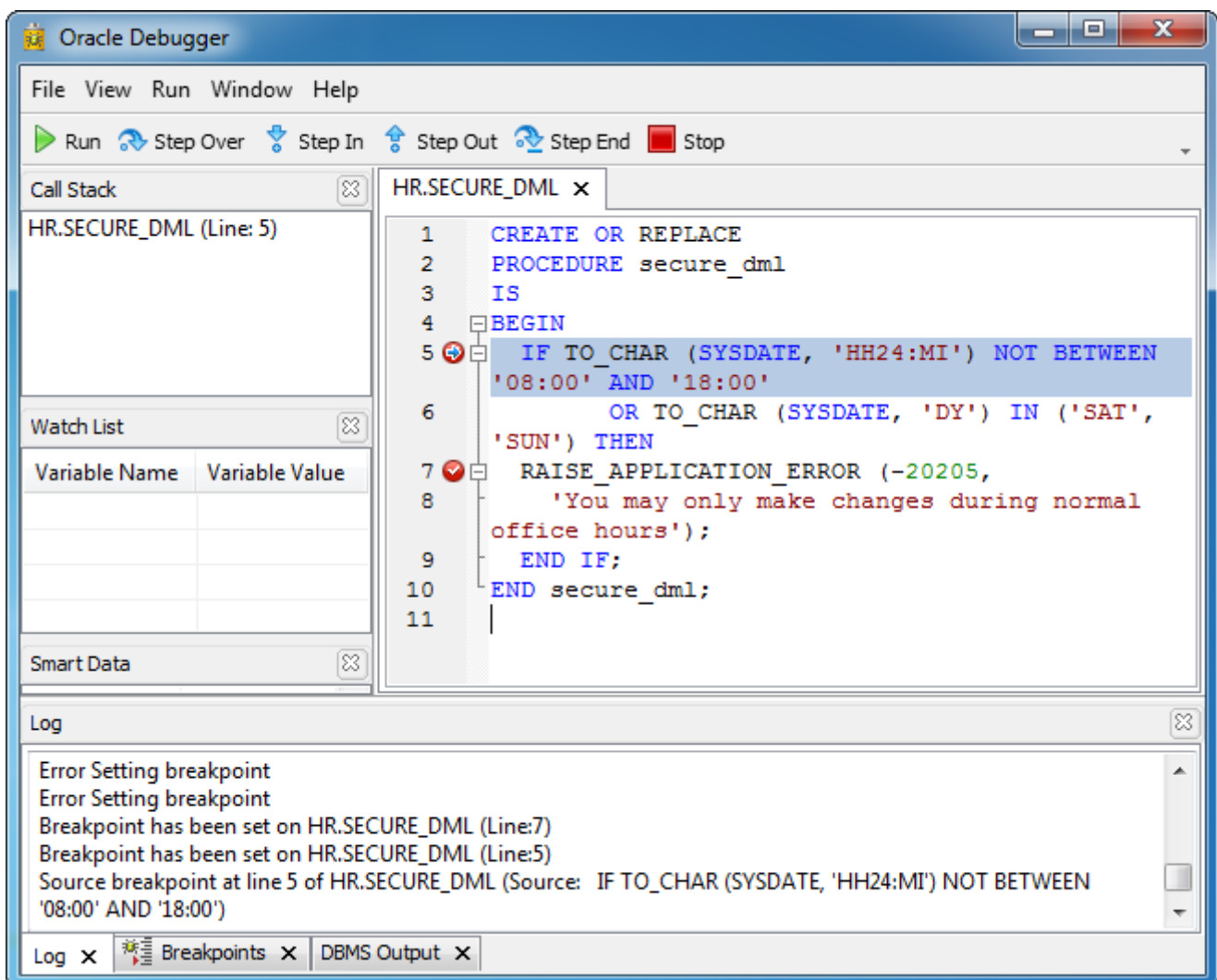
**Hint:** Oracle interprets non-quoted object identifiers as uppercase. In Navicat, all objects identifier will be quoted. That is, Navicat saves exactly what you have inputted.

- [Tables](#)
- [Views](#)
- [Functions/Procedures](#)
- [Database Links](#)
- [Indexes](#)
- [Java](#)
- [Materialized Views](#)
- [Materialized View Logs](#)
- [Packages](#)
- [Sequences](#)
- [Synonyms](#)
- [Triggers](#)
- [Types](#)
- [XML Schemas](#)
- [Recycle Bin](#)
- [Directories](#)
- [Tablespaces](#)
- [Public Database Links](#)
- [Public Synonyms](#)


## Oracle Debugger (Available only in Full version)

Navicat provides Oracle PL/SQL debugger for debugging Oracle functions, procedures, packages and queries.

- [Code Window](#)
- [Views](#)
- [Oracle Debugger Toolbar](#)



## Oracle Debugger Code Window

The **Code Window** shows the code of the procedure/function. You can add/remove breakpoints for debugging by clicking  in the grey area beside each statement.

To add a variable to the watch list, simply right-click the highlighted code and choose **Add to watch list**.

To show the debug tips, simply mouse-over the code.

```
1 CREATE OR REPLACE
2 PROCEDURE add_job_history
3   ( p_emp_id          job_history.employee_id%type
4     , p_start_date    job_history.start_date%type
5     , p_end_date      job_history.end_date%type
6     , p_job_id        job_history.job_id%type
7     , p_department_id job_history.department_id%type
8   )
9 IS
10 BEGIN
11   INSERT INTO job_history (employee_id, start_date,
12                          end_date,
13                          job_id, department_id)
14     VALUES (p_emp_id, p_start_date, p_end_date, p_job_id,
15             p_department_id); p_emp_id = 1
16 END add_job_history;
```

## Oracle Debugger Views

Under menu **View**, you can choose to show/hide the following view windows.

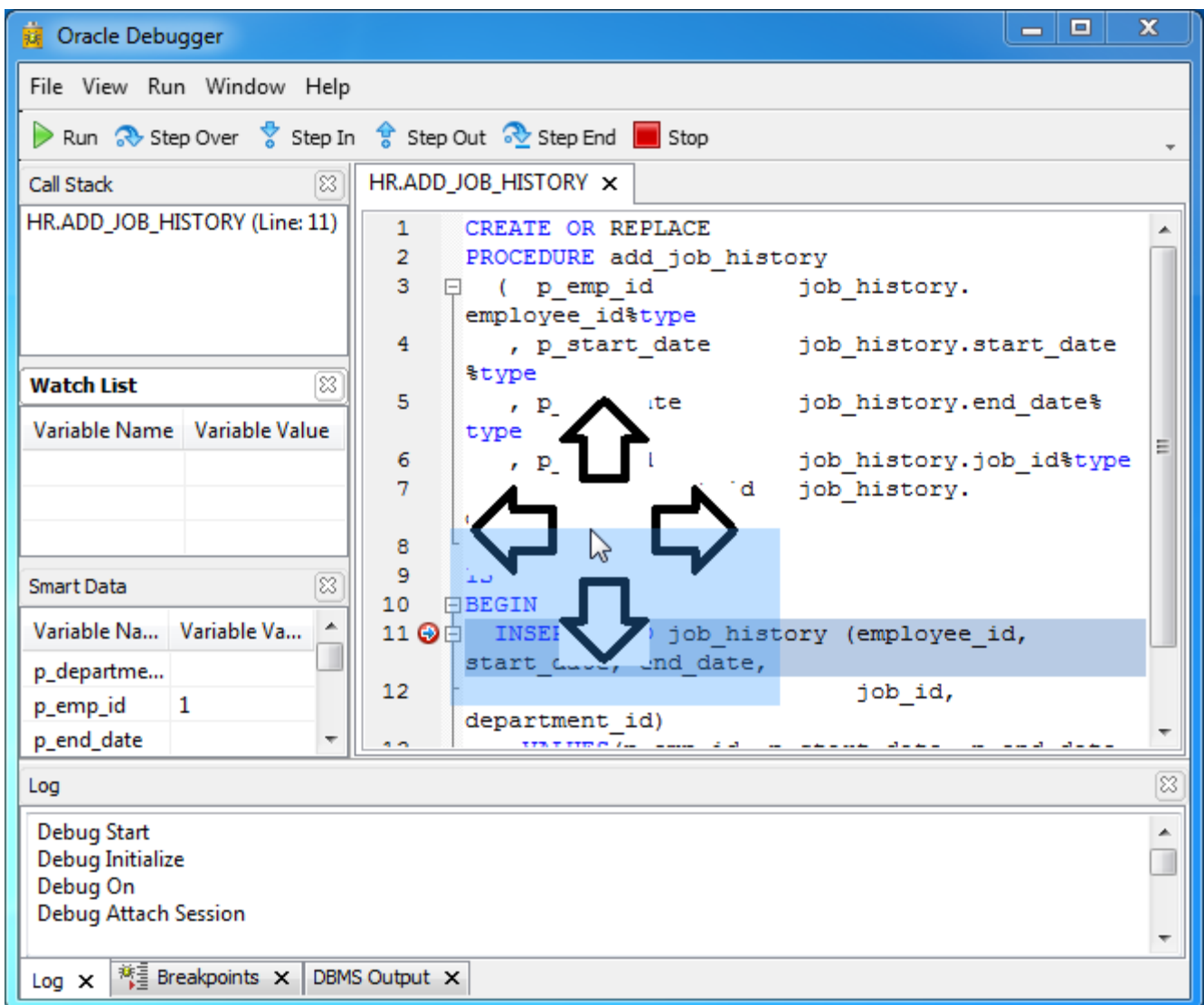
You can choose View -> **Default Layout** to restore the Layout to the default one.

- [Breakpoints](#)
- [Call Stack](#)
- [DBMS Output](#)
- [Log](#)
- [Smart Data](#)
- [Watch List](#)

## Repositioning Oracle Debugger Views

To re-arrange the workspace items, click on any pane in the workspace, then hold down your cursor and drag the pane to the desired area, and release the cursor. The selected pane will appear in its new position.

**Hint:** As you drag a pane to its new position, a **Pane Sticker** will appear and a bright blue indicator will mark the insertion point.



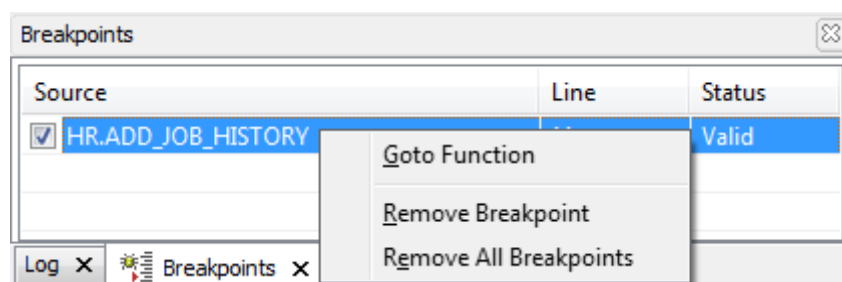
## Oracle Debugger Breakpoints

The **Breakpoints** view displays all the breakpoints which allowing you to delete, enable or disable breakpoints.

To enable/disable a breakpoint, simply check/uncheck the check box.

Also, you can delete a breakpoint or all breakpoints, simply right-click a breakpoint and choose **Remove Breakpoint** or **Remove All Breakpoints**.

To jump to the line of a breakpoint, right-click it and choose **Goto Function**.



## Oracle Debugger Call Stack

The **Call Stack** view displays the procedure or function calls of the current line.

To jump to a procedure or function, right-click it and choose **Goto Function**.

## Oracle Debugger DBMS Output

The **DBMS Output** view displays the results after the function or procedure has completed the execution.

## Oracle Debugger Log

The **Log** view shows the message log when debugging the code.

## Oracle Debugger Smart Data

The **Smart Data** view displays information about the variables associated with breakpoints.

To add a variable to the watch list, simply right-click a variable and choose **Add to watch list**.

You can adjust the value of a watch variable by simply right-click the variable and choose **Adjust Value**.

## Oracle Debugger Watch List

The **Watch List** view displays information about the variables being watched, allowing you to add, delete or edit watch variables.

To add a watch variable, simply right-click anywhere of Watch List view and choose **Add Variable**. Then, enter the **Variable Name**. Also, you can right-click the highlighted code in the Code Window or the variable in the Smart Data view and choose **Add to watch list**.


You can adjust the value of a watch variable by right-click the variable and choose **Adjust Value**.

To delete a watch variable or all watch variables, simply right-click a variable and choose **Remove Variable** or **Remove All Variable**.


## Oracle Debugger Toolbar

You can perform the most commonly used actions for debugging on the toolbar or menu:


### **Run**

Start running code in debug mode by clicking  **Run** or pressing **F9**. The debugger executes your code until the end of the code or the next breakpoint is reached.


### **Step Over**

While execution of your code is paused, you can resume it by clicking  **Step Over** or pressing **F8**. Then, the current line will be executed. If the line is a procedure or function call, it will bypass the procedure or function. The counter will then move to the next line of code.


### **Step In**

While execution of your code is paused, you can resume it by clicking  **Step In** or pressing **F7**. Then, it executes the current line. If the line is a procedure or function call, the counter goes to the first statement in the procedure or function. Otherwise, the counter will move to the next line of code.


### **Step Out**

While execution of your code is paused, you can resume it by clicking  **Step Out** or pressing **Shift+F7**. Then, the remaining part of the code within the current procedure or function will be executed. Subsequently, the counter will jump to the line which is just after the caller of the method.

### **Step End**

While execution of your code is paused, you can resume it by clicking  **Step End**. Then, the counter will jump to the last line of the procedure or function.

### **Stop**

While execution of your code is paused, you can stop stepping the code by clicking  **Stop**. Then, the execution will stop and cannot resume it.

## Editing Oracle Physical Attributes/Default Storage Characteristics

### **% Free**

Specify a whole number representing the percentage of space in each data block of the database object reserved for future updates to rows of the object. The value must be from 0 to 99. A value of 0 means that the entire block can be filled by inserts of new rows. The default value is 10.

### **% Used**

Specify a whole number representing the minimum percentage of used space that Oracle maintains for each data block of the database object. A block becomes a candidate for row insertion when its used space falls below this value. The value must be from 0 to 99 and defaults to 40.

### **Initial Transactions**

Specify the initial number of concurrent transaction entries allocated within each data block allocated to the database object. This value can range from 1 to 255 and defaults to 1.

**Note:** The default value for an index is 2.

### **Maximum Transactions**

Specify the maximum number of concurrent update transactions allowed for each data block in the segment.

### **Storage**

#### **Initial**

Specify the size of the first extent of the object. Use the dropdown list K, M, G, T, P or E to specify the size in kilobytes, megabytes, gigabytes, terabytes, petabytes, or exabytes. If you do not specify any of the abbreviations, then the size is interpreted as bytes.

#### **Next**

Specify the size of the next extent to be allocated to the object. Use the dropdown list K, M, G, T, P or E to specify the size in kilobytes, megabytes, gigabytes, terabytes, petabytes, or exabytes. If you do not specify any of the abbreviations, then the size is interpreted as bytes.

#### **Min Extents**

Specify the total number of extents to allocate when the object is created.

## **Max Extents**

Specify the total number of extents, including the first, that Oracle can allocate for the object. Check **Unlimited** if you want extents to be allocated automatically as needed.

## **Max Size**

Specify the maximum size of the storage element. Use the dropdown list K, M, G, T, P or E to specify the size in kilobytes, megabytes, gigabytes, terabytes, petabytes, or exabytes. If you do not specify any of the abbreviations, then the size is interpreted as bytes. Check **Unlimited** if you do not want to limit the disk space of the storage element.

## **% Increase**

Specify the percent by which the third and subsequent extents grow over the preceding extent. The default value is 50.

## **Freelists**

For objects other than tablespaces and rollback segments, specify the number of free lists for each of the free list groups for the table, partition, cluster, or index.

## **Freelist Group**

Specify the number of groups of free lists for the database object you are creating. The default and minimum value for this parameter is 1.

## **Optimal**

Specify an optimal size for a rollback segment. Use the dropdown list K, M, G, T, P or E to specify the size in kilobytes, megabytes, gigabytes, terabytes, petabytes, or exabytes. If you do not specify any of the abbreviations, then the size is interpreted as bytes. Check **Null** for no optimal size for the rollback segment.

## **Buffer Pool**

### **DEFAULT**

Choose this to indicate the default buffer pool. This is the default for objects not assigned to KEEP or RECYCLE.

### **KEEP**


Choose this to put blocks from the segment into the KEEP buffer pool. Maintaining an appropriately sized KEEP buffer pool lets Oracle retain the schema object in memory to avoid I/O operations. KEEP takes precedence over any NOCACHE clause you specify for a table, cluster, materialized view, or materialized view log.

## **RECYCLE**

Choose this to put blocks from the segment into the RECYCLE pool. An appropriately sized RECYCLE pool reduces the number of objects whose default pool is the RECYCLE pool from taking up unnecessary cache space.




## Oracle Tables


Relational databases use tables to store data. All operations on data are done on the tables themselves or produce another table as the result. A table is a set of rows and columns, and their intersections are fields. From a general perspective, columns within a table describe the name and type of data that will be found by row for that column's fields. Rows within a table represent records composed of fields that are described from left to right by their corresponding column's name and type. Each field in a row is implicitly correlated with each other field in that row.

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

### Create Table

To create a new table

- Select anywhere on the object pane.
- Click the  **New Table** from the object pane toolbar together with the  down arrow to choose the type **Normal / External / Index Organized**.
- or
- Right-click and select  **New Table** to choose the type **Normal / External / Index Organized** from the popup menu.
- Edit table properties and fields on the appropriate tabs of the Table Designer.

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

To create a new table with the same properties as one of the existing tables has (using popup menu)

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

- Select the table(s) for copying in the navigation pane/object pane.
- Right-click and select the **Duplicate Table** from the popup menu.
- The newly created table(s) will be named as "tablename\_**copy**".

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

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




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

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

different schema {different connection (same or cross server type)} (Data Transfer tool will be activated)

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

To create a new table with modification as one of the existing tables

- Select the table for modifying in the navigation pane/object pane.
- Right-click and select the  **Design Table** from the popup menu.  
or
- Click the  **Design Table** from the object pane toolbar.
- Modify table properties and fields on the appropriate tabs of the Table Designer.
- Click  **Save As**.

## Create Table Shortcut



To create a table shortcut

- Select the table for editing in the navigation pane/object pane.
- Right-click and select **Create Open Table Shortcut...** from the popup menu.
- Define the location you wish your shortcut to be saved.

**Note:** This option is used to provide a convenient way for you to open your table for entering data directly (Grid View/Form View) without activating the main Navicat.

## Edit Table

To edit the existing table (manage its fields, indexes, foreign keys and triggers etc)



- Select the table for editing in the navigation pane/object pane.
- Right-click and select the  **Design Table** from the popup menu.  
or
- Click the  **Design Table** from the object pane toolbar.
- Edit table properties and fields on the appropriate tabs of the Table Designer.


To change the name of the table

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


## Open Table (manage table data)

To open a table

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

**Note:** This option is only applied if you do wish Navicat loads all your images while opening the table. To open the graphical table with faster performance, use  **Open Table (Quick)** below.

To open a table with graphical fields

- Select the table for opening in the navigation pane/object pane.
- Right-click and select the  **Open Table (Quick)** from the popup menu.

**Note:** Faster performance for opening the graphical table, as BLOB fields (images) will not be loaded until you click on the cell.

## Empty Table

To empty a table

- Select the table in the navigation pane/object pane.
- Right-click the selected table and choose **Empty Table** from the popup menu.

**Note:** This option is only applied when you wish to clear all the existing records without resetting the auto-increment value. To reset the auto-increment value while emptying your table, use **Truncate Table** below.



## Truncate Table

To truncate a table

- Select the table in the navigation pane/object pane.
- Right-click the selected table and choose **Truncate Table** from the popup menu.

## Delete Table

To delete a table

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

## Achieve Table Information

To achieve a table information

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

## **Oracle Normal Tables**

Tables are the basic unit of data storage in an Oracle database. Data is stored in rows and columns. You define a table with a table name and set of columns.

In a normal (heap-organized) table, data is stored as an unordered collection (heap).

## Table Designer for Oracle Normal Tables (/Index Organized Tables)

**Table Designer** is the basic Navicat tool for working with tables. It allows you to create, edit and drop table's fields, indexes, foreign keys, and much more.

Note that **Table Designer** for **Index Organized Tables** differs from **Normal Tables** only on the [Options](#) tab.



- [Managing Table Fields](#)
- [Managing Table Indexes](#)
- [Managing Table Foreign Keys](#)
- [Managing Table Uniques](#)
- [Managing Table Checks](#)
- [Managing Table Triggers](#)
- [Managing Table Options](#)
- Managing Table Comment
- Table SQL Preview

## Oracle Table Fields

Table fields are managed on the **Fields** tab of the Table Designer. Just simply click a field for editing. A right-click displays the popup menu or by using the field toolbar, allowing you to create new and drop the selected field.

### Add Field

To add a field to the table

- Open the table in the Table Designer.
- Open the **Fields** tab.
- Right-click and select the  **Add Field** from the popup menu or click the  **Add Field** from the toolbar.
- Edit field properties.

To add a new field with modification as one of the existing fields

- Open the table in the Table Designer.
- Open the **Fields** tab.
- Select field.
- Right-click and select the **Duplicate Field** from the popup menu.
- Edit field properties.



### Edit Field

To edit the table field


- Open the table in the Table Designer.
- Open the **Fields** tab.
- Simply click on the field to edit.

## Delete Field

To delete the table field

- Open the table in the Table Designer.
- Open the **Fields** tab.
- Right-click and select the  **Delete Field** from the popup menu or click the  **Delete Field** from the toolbar.
- Confirm deleting in the dialog window.

## Setting Oracle Table Field Properties

Name	Type	Length	Scale	Allow Null	
▶ EMPLOYEE_ID	NUMBER	6	0	<input type="checkbox"/>	 1
FIRST_NAME	VARCHAR2	20	0	<input checked="" type="checkbox"/>	
LAST_NAME	VARCHAR2	25	0	<input type="checkbox"/>	
EMAIL	VARCHAR2	25	0	<input type="checkbox"/>	

### Name

The Name is a descriptive identifier for a field that can be up to 30 bytes by default (letters or numbers) including spaces. The names should be descriptive enough that anyone can easily identify them when viewing or editing records. For example, LastName, FirstName, StreetAddress, or HomePhone.

Use the **Name** edit box to set the field name. Note that the name of the field must be unique among all the field names in the table.

### Type

After you name a field, you choose a data type for the data to be contained in the field. When you choose a field's data type, you are deciding:

- What kind of values to allow in the field. You cannot store text in field with the **Numeric** data type.
- How much storage space Oracle is to set aside for the data in that field.
- What types of operations can be performed on the values in that field.

The **Type** dropdown list defines the type of the field data.

The following table shows all the built-in general-purpose data types.

Name	Description
CHAR	fixed-length character strings
NCHAR	fixed-length Unicode character data
VARCHAR2	variable-length character strings
VARCHAR	variable-length character strings
NVARCHAR2	variable-length Unicode character data
CLOB	database character set data
NCLOB	Unicode national character set data

LONG	variable-length character data containing up to 2 gigabytes of information
NUMBER	fixed and floating-point numbers
DATE	point-in-time values (dates and times)
INTERVAL DAY TO SECOND	period of time in terms of days, hours, minutes, and seconds
INTERVAL YEAR TO MONTH	stores a period of time using the YEAR and MONTH datetime fields
TIMESTAMP	point-in-time values (dates and times) (includes fractional seconds)
TIMESTAMP WITH TIME ZONE	TIMESTAMP with explicit time zone information
BLOB	unstructured binary data in the database
BFILE	unstructured binary data in operating-system files outside the database
RAW	can be indexed and is used for data that is not to be interpreted by Oracle Database
LONG RAW	cannot be indexed and is used for data that is not to be interpreted by Oracle Database
ROWID	the address (rowid) of every row in the database
CHARACTER	=CHAR <sup>1</sup>
CHARACTER VARYING	=VARCHAR2 <sup>1</sup>
CHAR VARYING	=VARCHAR2 <sup>1</sup>
NATIONAL CHARACTER	=NCHAR <sup>1</sup>
NATIONAL CHAR	=NCHAR <sup>1</sup>
NATIONAL CHARACTER VARYING	=NVARCHAR2 <sup>1</sup>
NATIONAL CHAR VARYING	=NVARCHAR2 <sup>1</sup>
NCHAR VARYING	=NVARCHAR2 <sup>1</sup>
NUMERIC	=NUMBER <sup>1</sup>
DECIMAL	=NUMBER <sup>1</sup>
INTEGER	=NUMBER(38) <sup>1</sup>
INT	=NUMBER(38) <sup>1</sup>
SMALLINT	=NUMBER(38) <sup>1</sup>
FLOAT	=FLOAT(126) <sup>1</sup>
DOUBLE PRECISION	=FLOAT(126) <sup>1</sup>
REAL	=FLOAT(63) <sup>1</sup>

**Note:** These are ANSI datatypes and datatypes from the IBM products SQL/DS and DB2. Oracle recognizes these datatypes and converts them to the equivalent Oracle datatype. [Click here](#) for detailed description on these datatypes.

## Length and Scale

Use the **Length** edit box to define the **precision** (total number of digits) of the field and use **Scale** edit box to define the **scale** (number of digits to the right of the decimal point) for **numeric** column.

**Note:** Be careful when shortening the field length as it might result in data loss.

How scale factors affect numeric data storage:

Input Data	Specified As	Stored As
7,456,123.89	NUMBER	7456123.89
7,456,123.89	NUMBER(*,1)	7456123.9
7,456,123.89	NUMBER(9)	7456124
7,456,123.89	NUMBER(9,2)	7456123.89
7,456,123.89	NUMBER(9,1)	7456123.9
7,456,123.89	NUMBER(6)	not accepted, exceeds precision
7,456,123.89	NUMBER(7,-2)	7456100

### Allow Null

Allow the NULL values for the field.

### Primary Key

A **Primary Key** is a single field or combination of fields that uniquely defines a record. None of the fields that are part of the primary key can contain a null value.

### Primary Key Name

Right-click and select **Primary Key Name** from the popup menu to enter the primary key constraint name.

## Setting Other Oracle Table Field Properties

For **INTERVAL DAY TO SECOND** data type:

### Leading Field Precision

Set the number of digits in the leading field. Accepted values are 0 to 9. The default is 2.

### Fractional Seconds Precision

Set the number of digits in the fractional part of the SECOND datetime field. Accepted values are 1 to 9.

For **INTERVAL YEAR TO MONTH** data type:

### Year Precision

Set the number of digits in the year. The default is 2.

For **TIMESTAMP, TIMESTAMP WITH TIME ZONE, TIMESTAMP WITH LOCAL TIME ZONE** data types:

### Fractional Seconds Precision

Set the number of digits in the fractional part of the SECOND datetime field. Accepted values are 1 to 9.

For **CHAR, VARCHAR2** data types:

### Unit

Set the unit either in BYTE or CHAR.

For **COMPLEX** data types:

### Object Schema

Set the object schema for the field.

### Object Type

Set the object type for the field.

For most data types:

### Default

Set the default value for the field.

For all data types:

**Comment**

Set any optional text describing the current field.

## Oracle Table Indexes



Indexes are optional structures associated with tables and clusters. You can create indexes on one or more columns of a table to speed SQL statement execution on that table. An Oracle Database index provides a faster access path to table data. Indexes are the primary means of reducing disk I/O when properly used.

You can create many indexes for a table as long as the combination of columns differs for each index. You can create more than one index using the same columns if you specify distinctly different combinations of the columns.

Table indexes are managed on the **Indexes** tab of the Table Designer. Just simply click/double-click an index field for editing. A right-click displays the popup menu or using the index toolbar, allowing you to create new, edit and delete the selected index field.

### Add Index

To add a table index

- Open the table in the Table Designer.
- Open the **Indexes** tab.
- Right-click and select the  **Add Index** from the popup menu or click the  **Add Index** from the toolbar.
- Edit index properties.



### Edit Index

To edit a table index


- Open the table in the Table Designer.
- Open the **Indexes** tab.
- Just simply click/double-click on the index to edit.

## Delete Index


To delete a table index

- Open the table in the Table Designer.
- Open the **Indexes** tab.
- Right-click on the index to delete and select the  **Delete Index** from the popup menu or click the  **Delete Index** from the toolbar.
- Confirm deleting in the dialog window.

## Setting Oracle Table Index Properties

Name	Fields	Index Type
EMP_DEPARTMENT_IX	DEPARTMENT_ID ASC	Non-Unique
▶ EMP_JOB_IX	JOB_ID ASC	 Non-Unique

Use the **Name** edit box to set the index name.

To include field(s) in the index, just simply double-click the **Fields** field or click  to open the editor for editing.

The **Index Type** dropdown list defines the type of the table index. Oracle Database provides several indexing schemes.

### Non-unique

Non-unique indexes do not impose the restriction of unique indexes on the column values.

### Unique

Unique indexes guarantee that no two rows of a table have duplicate values in the key column (or columns).

### Bitmap

In a bitmap index, a bitmap for each key value is used instead of a list of rowids.

### Parallel With Degree

Parallel indexing can improve index performance when you have a large amount of data, and have multiple CPUs. Enter the degree that determines the number of separate indexing processes.

### Tablespace

The tablespace in which to create the index. An index can be created in the same or different tablespace as the table it indexes.

## Schema

The schema in which to create the index.

### Note:

To create an index in your own schema, at least one of the following conditions must be true:

- The table or cluster to be indexed is in your own schema.
- You have INDEX privilege on the table to be indexed.
- You have CREATE ANY INDEX system privilege.

To create an index in another schema, all of the following conditions must be true:

- You have CREATE ANY INDEX system privilege.
- The owner of the other schema has a quota for the tablespaces to contain the index or index partitions, or UNLIMITED TABLESPACE system privilege.



## Oracle Table Foreign Keys

A foreign key specifies that the values in a column (or a group of columns) must match the values appearing in some row of another table. We say this maintains the referential integrity between two related tables.

Foreign Keys are managed on the **Foreign Keys** tab of the Table Designer. Just simply click/double-click a foreign key field for editing. A right-click displays the popup menu or using the foreign key toolbar, allowing you to create new, edit and delete the selected foreign key field.

### Add Foreign Key

To add a foreign key

- Open the table in the Table Designer.
- Open the **Foreign Keys** tab.
- Right-click and select the  **Add Foreign Key** from the popup menu or click the  **Add Foreign Key** from the toolbar.
- Edit foreign key properties.



### Edit Foreign Key

To edit a foreign key


- Open the table in the Table Designer.
- Open the **Foreign Keys** tab.
- Just simply click/double-click on the foreign key to edit.

### Delete Foreign Key

To delete a foreign key


- Open the table in the Table Designer.
- Open the **Foreign Keys** tab.
- Right-click on the foreign key to delete and select the  **Delete Foreign Key** from the popup menu or click the  **Delete Foreign Key** from the toolbar.
- Confirm deleting in the dialog window.

## Setting Oracle Table Foreign Key Properties

Name	Reference Schema	Reference Table	Reference Constraint	Reference Fields	Fields	On Delete	Enable
EMP_DEPT_FK	HR	DEPARTMENTS	DEPT_ID_PK	DEPARTMENT_ID	DEPARTMENT_ID		<input checked="" type="checkbox"/>
▶ EMP_JOB_FK	HR	JOBS	JOB_ID_PK	JOB_ID	JOB_ID  		<input checked="" type="checkbox"/>

Use the **Name** edit box to enter a name for the new key and then select a table field to include in the key from the **Fields** group.

Use the **Reference Schema**, **Reference Table** and **Reference Constraint** dropdown lists to select a foreign schema, table and constraint respectively.

To include field(s) to the key, just simply double-click the **Fields** field or click  to open the editor(s) for editing.

The **On Delete** dropdown list defines the type of the actions to be taken.

### No Action (default)

Referenced key values will not be updated or deleted.

### Cascade

Delete any rows referencing the deleted row, or update the value of the referencing column to the new value of the referenced column, respectively.

### Set Null

Set the referencing column(s) to null.

### Enable

You can choose whether to enable / disable the foreign key constraint by checking / unchecking the box.



## Oracle Table Uniques

Unique constraints ensure that the data contained in a column or a group of columns is unique with respect to all the rows in the table.

Uniques are managed on the **Uniques** tab of the Table Designer. Just simply click/double-click an unique field for editing. Using the toolbar below, allowing you to create new, edit and delete the selected unique field.

### Add Unique

To add an unique

- Open the table in the Table Designer.
- Open the **Uniques** tab.
- Right-click and select the  **Add Unique** from the popup menu or click the  **Add Unique** from the toolbar.
- Edit unique properties.



### Edit Unique

To edit an unique

- Open the table in the Table Designer.
- Open the **Uniques** tab.
- Just simply click on the unique to edit.

### Delete Unique

To delete an unique


- Open the table in the Table Designer.
- Open the **Uniques** tab.
- Right-click on the unique to delete and select the  **Delete Unique** from the popup menu or click the  **Delete Unique** from the toolbar.
- Confirm deleting in the dialog window.

## Oracle Table Setting Unique Properties

Name	Fields	Enable
▶ EMP_EMAIL_UK	EMAIL	<input type="checkbox"/> <input checked="" type="checkbox"/>

Use the **Name** edit box to set the unique name.

### Fields

To set field(s) as unique, just simply double-click the **Fields** field or click  to open the editor(s) for editing.

### Enable

You can choose whether to enable / disable the unique constraint by checking / unchecking the box.



## Oracle Table Checks

A check constraint is the most generic constraint type. It allows you to specify that the value in a certain column must satisfy a Boolean (truth-value) expression.

Checks are managed on the **Checks** tab of the Table Designer. Just simply click/double-click a check field for editing. Using the check toolbar, allowing you to create new, edit and delete the selected check field.

### Add Check

To add a check

- Open the table in the Table Designer.
- Open the **Checks** tab.
- Right-click and select the  **Add Check** from the popup menu or click the  **Add Check** from the toolbar.
- Edit check properties.



### Edit Check

To edit a check

- Open the table in the Table Designer.
- Open the **Checks** tab.
- Just simply click on the check to edit.

### Delete Check

To delete a check

- Open the table in the Table Designer.
- Open the **Checks** tab.
- Right-click on the check to delete and select the  **Delete Check** from the popup menu or click the  **Delete Check** from the toolbar.
- Confirm deleting in the dialog window.

## Setting Oracle Table Check Properties

Use the **Name** edit box to set the check name.

### Check

Set the condition for checking, e.g. "field\_name1 > 0 AND field\_name2 > field\_name1" in the **Check** edit box. A check constraint specified as a column constraint should reference that column's value only, while an expression appearing in a table constraint may reference multiple columns.

### Definition

Type in the definition for the check constraint.

### **Enable**

You can choose whether to enable / disable the check constraint by checking / unchecking the box.



## Oracle Table Triggers

A trigger is a specification that the database should automatically execute a particular function whenever a certain type of operation is performed. Triggers can be defined to execute either before or after any INSERT, UPDATE, or DELETE operation, either once per modified row, or once per SQL statement.

Triggers are managed on the **Triggers** tab of the Table Designer. Just simply click a trigger field for editing. A right-click displays the popup menu or using the trigger toolbar, allowing you to create new, edit and delete the selected trigger field.

### Add Trigger

To add a trigger

- Open the table in the Table Designer.
- Open the **Triggers** tab.
- Right-click and select the  **Add Trigger** from the popup menu or click the  **Add Trigger** from the toolbar.
- Edit trigger properties.



### Edit Trigger

To edit a trigger

- Open the table in the Table Designer.
- Open the **Triggers** tab.
- Just simply click on the trigger to edit.

### Delete Trigger

To delete a trigger

- Open the table in the Table Designer.
- Open the **Triggers** tab.
- Right-click on the trigger to delete and select the  **Delete Trigger** from the popup menu or click the  **Delete Trigger** from the toolbar.
- Confirm deleting in the dialog window.

## Setting Oracle Table Trigger Properties

### Name

Set the trigger name.

### Compound

Check to set the trigger as a compound trigger.

**Note:** Support from Oracle 11g or later.

### Row trigger

Check to set the trigger as a row trigger.

### Fires

Specify the trigger timing whether the trigger action is to be run before or after the triggering statement.

*INSERT* - fires the trigger whenever an INSERT statement adds a row to a table or adds an element to a nested table.

*UPDATE* - fires the trigger whenever an UPDATE statement changes a value in one of the columns specified in **Update of Fields**. If no **Update of Fields** are present, the trigger will be fired whenever an UPDATE statement changes a value in any column of the table or nested table.

*DELETE* - fires the trigger whenever a DELETE statement removes a row from the table or removes an element from a nested table.

### Update Of Fields

Specify the fields for UPDATE statement trigger upon necessary.

### Enable

You can choose whether to enable / disable the trigger constraint by checking / unchecking the box.

## Definition

Type in the definition for the trigger.

Example:

```
BEGIN
    add_job_history(:old.employee_id, :old.hire_date, sysdate,
                  :old.job_id, :old.department_id);
END;
```

## Advanced

### When Clause

Specify the trigger condition, which is a SQL condition that must be satisfied for the database to fire the trigger. This condition must contain correlation names and cannot contain a query.

### Referencing Old

Specify correlation names. The default correlation names are OLD and NEW.

### Referencing New

Specify correlation names. The default correlation names are OLD and NEW.

### Follows

Specify the relative firing order of triggers of the same type.

**Note:** Support from Oracle 11g or later.

### Schema

Define the trigger on the current schema.

## Oracle Table Options

### Tablespace

Define a tablespace different from the default tablespace to create a table.

### Logging

Specify whether creation of a database object will be logged in the redo log file (LOGGING) or not (NOLOGGING).

### Compression

Specify whether to compress data segments to reduce disk use. It is valid only for heap-organized tables.

- *COMPRESS* - enables table compression.
- *COMPRESS FOR ALL OPERATIONS* - attempts to compress data during all DML operations on the table.
- *COMPRESS FOR DIRECT\_LOAD OPERATIONS* - attempts to compress data during direct-path INSERT operations when it is productive to do so.
- *NOCOMPRESS* - disables table compression.

### Cache

Indicate how blocks are stored in the buffer cache.

- *CACHE* - indicates that the blocks retrieved for this table are placed at the most recently used end of the least recently used (LRU) list in the buffer cache when a full table scan is performed.
- *NOCACHE* - indicates that the blocks retrieved for this table are placed at the least recently used end of the LRU list in the buffer cache when a full table scan is performed.

### Parallel With Degree

Specify the degree of parallelism, which is the number of parallel threads used in the parallel operation.

### Row Movement

With the option on, it allows the database to move a table row. It is possible for a row to move, for example, during table compression or an update operation on partitioned data.

### Physical Attributes

Refer to [Editing Physical Attributes/Default Storage Characteristics](#).

## **Oracle External Tables**

External tables access data in external sources as if it were in a table in the database. While creating external tables, you are actually creating metadata in the data dictionary that enables you to access external data.

Note that external tables are read only. No DML operations are possible and no index can be created.

## Table Designer for Oracle External Tables

**Table Designer** for External Tables allows you to create, edit table's fields, external properties and access parameters etc.



- [Managing External Table Fields](#)
- [Managing External Table External Properties](#)
- [Managing External Table Access Parameters](#)
- External Table SQL Preview

## Fields for Oracle External Tables

Table fields are managed on the **Fields** tab of the Table Designer. Just simply click a field for editing. A right-click displays the popup menu or by using the field toolbar, allowing you to create new and drop the selected field.

### Add Field

To add a field to the table

- Open the table in the Table Designer.
- Open the **Fields** tab.
- Right-click and select the  **Add Field** from the popup menu or click the  **Add Field** from the toolbar.
- Edit field properties.

To add a new field with modification as one of the existing fields

- Open the table in the Table Designer.
- Open the **Fields** tab.
- Select field.
- Right-click and select the **Duplicate Field** from the popup menu.
- Edit field properties.



### Edit Field

To edit the table field

- Open the table in the Table Designer.
- Open the **Fields** tab.
- Simply click on the field to edit.

## Delete Field

To delete the table field

- Open the table in the Table Designer.
- Open the **Fields** tab.
- Right-click and select the  **Delete Field** from the popup menu or click the  **Delete Field** from the toolbar.
- Confirm deleting in the dialog window.

## Setting Field Properties for Oracle External Tables

Name	Type	Length	Scale
▶ EMPLOYEE_ID	NUMBER	6	0
FIRST_NAME	VARCHAR2	20	0
LAST_NAME	VARCHAR2	25	0
EMAIL	VARCHAR2	25	0
PHONE_NUMBER	VARCHAR2	20	0

### Name

The Name is a descriptive identifier for a field that can be up to 30 characters by default (letters or numbers) including spaces. The names should be descriptive enough that anyone can easily identify them when viewing or editing records. For example, LastName, FirstName, StreetAddress, or HomePhone.

Use the **Name** edit box to set the field name. Note that the name of the field must be unique among all the field names in the table.

### Type

After you name a field, you choose a data type for the data to be contained in the field. When you choose a field's data type, you are deciding:

- What kind of values to allow in the field. You cannot store text in field with the **Numeric** data type.
- How much storage space Oracle is to set aside for the data in that field.
- What types of operations can be performed on the values in that field.

The **Type** dropdown list defines the type of the field data.

The following table shows all the built-in general-purpose data types.

Name	Description
CHAR	fixed-length character strings
NCHAR	fixed-length Unicode character data
VARCHAR2	variable-length character strings
VARCHAR	variable-length character strings
NVARCHAR2	variable-length Unicode character data
CLOB	database character set data
NCLOB	Unicode national character set data

LONG	variable-length character data containing up to 2 gigabytes of information
NUMBER	fixed and floating-point numbers
DATE	point-in-time values (dates and times)
INTERVAL DAY TO SECOND	period of time in terms of days, hours, minutes, and seconds
INTERVAL YEAR TO MONTH	stores a period of time using the YEAR and MONTH datetime fields
TIMESTAMP	point-in-time values (dates and times) (includes fractional seconds)
TIMESTAMP WITH TIME ZONE	TIMESTAMP with explicit time zone information
BLOB	unstructured binary data in the database
BFILE	unstructured binary data in operating-system files outside the database
RAW	can be indexed and is used for data that is not to be interpreted by Oracle Database
LONG RAW	cannot be indexed and is used for data that is not to be interpreted by Oracle Database
ROWID	the address (rowid) of every row in the database
CHARACTER	=CHAR <sup>1</sup>
CHARACTER VARYING	=VARCHAR2 <sup>1</sup>
CHAR VARYING	=VARCHAR2 <sup>1</sup>
NATIONAL CHARACTER	=NCHAR <sup>1</sup>
NATIONAL CHAR	=NCHAR <sup>1</sup>
NATIONAL CHARACTER VARYING	=NVARCHAR2 <sup>1</sup>
NATIONAL CHAR VARYING	=NVARCHAR2 <sup>1</sup>
NCHAR VARYING	=NVARCHAR2 <sup>1</sup>
NUMERIC	=NUMBER <sup>1</sup>
DECIMAL	=NUMBER <sup>1</sup>
INTEGER	=NUMBER(38) <sup>1</sup>
INT	=NUMBER(38) <sup>1</sup>
SMALLINT	=NUMBER(38) <sup>1</sup>
FLOAT	=FLOAT(126) <sup>1</sup>
DOUBLE PRECISION	=FLOAT(126) <sup>1</sup>
REAL	=FLOAT(63) <sup>1</sup>

**Note:** These are ANSI datatypes and datatypes from the IBM products SQL/DS and DB2. Oracle recognizes these datatypes and converts them to the equivalent Oracle datatype. [Click here](#) for detailed description on these datatypes.

## Length and Scale

Use the **Length** edit box to define the **precision** (total number of digits) of the field and use **Scale** edit box to define the **scale** (number of digits to the right of the decimal point) for **numeric** column.

**Note:** Be careful when shortening the field length as it might result in data loss.

How scale factors affect numeric data storage:

Input Data	Specified As	Stored As
7,456,123.89	NUMBER	7456123.89
7,456,123.89	NUMBER(*,1)	7456123.9
7,456,123.89	NUMBER(9)	7456124
7,456,123.89	NUMBER(9,2)	7456123.89
7,456,123.89	NUMBER(9,1)	7456123.9
7,456,123.89	NUMBER(6)	not accepted, exceeds precision
7,456,123.89	NUMBER(7,-2)	7456100

## Setting Other Field Properties for Oracle External Tables

For **INTERVAL DAY TO SECOND** data type:

### Leading Field Precision

Set the number of digits in the leading field. Accepted values are 0 to 9. The default is 2.

### Fractional Seconds Precision

Set the number of digits in the fractional part of the SECOND datetime field. Accepted values are 1 to 9.

For **INTERVAL YEAR TO MONTH** data type:

### Year Precision

Set the number of digits in the year. The default is 2.

For **TIMESTAMP, TIMESTAMP WITH TIME ZONE, TIMESTAMP WITH LOCAL TIME ZONE** data types:

### Fractional Seconds Precision

Set the number of digits in the fractional part of the SECOND datetime field. Accepted values are 1 to 9.

For **CHAR, VARCHAR2** data types:

### Unit

Set the unit either in BYTE or CHAR.

For **COMPLEX** data types:

### Object Schema

Set the object schema for the field.

### Object Type

Set the object type for the field.

## External Properties for Oracle External Tables

### Default Directory

Specify the default directory for the external table.

### Directory

Set the external directory.

### Location

Set the external source location.

### Access Driver

Specify the access driver for the external table. The default type for external tables is ORACLE\_LOADER.

### Reject Limit

Specify the limit on the number of errors that can occur during a query of the external data.

### Parallel With Degree

Check to enable parallel query on the data sources and specify the degree of parallel access.

## Access Parameters for Oracle External Tables

Describe the mapping of the external data to the Oracle Database data columns.

### **Using CLOB**

Check this option to get a CLOB data value of the returned query.

## **Oracle Index Organized Tables**

An index-organized table has a storage organization that is a variant of a primary B-tree. Data for an index-organized table is stored in a B-tree index structure in a primary key sorted manner. Each leaf block in the index structure stores both the key and nonkey columns.

Index-organized tables have full table functionality. They support features such as constraints, triggers etc with additional features such as key compression.

## Table Designer for Oracle Index Organized Tables

**Table Designer** for **Index Organized Tables** allows you to create, edit and drop table's fields, indexes, foreign keys, and much more.

Note that **Table Designer** for **Index Organized Tables** differs from **Normal Tables** only on the **Options** tab. Therefore, we will refer to **Table Designer for Normal Table(/Index Organized Table)** on the following similar chapters:

- [Managing Table Fields](#)
- [Managing Table Indexes](#)
- [Managing Table Foreign Keys](#)
- [Managing Table Uniques](#)
- [Managing Table Checks](#)
- [Managing Table Triggers](#)
- Managing Table Comment
- Table SQL Preview

### **Options tab for Index Organized Table:**

- [Managing Table Options](#)

## Options for Oracle Index Organized Tables

### Tablespace

Define a tablespace different from the default tablespace to create a table.

### Logging

Specify whether creation of a database object will be logged in the redo log file (LOGGING) or not (NOLOGGING).

#### Parallel With Degree

Specify the degree of parallelism, which is the number of parallel threads used in the parallel operation.

#### Row Movement

With the option on, it allows the database to move a table row. It is possible for a row to move, for example, during table compression or an update operation on partitioned data.

### Physical Attributes

Refer to [Editing Physical Attributes/Default Storage Characteristics](#).

### IOT Properties

#### Key Compress

Check this option to enable key compression. Upon necessary, you can also specify the prefix length (as the number of key columns), which identifies how the key columns are broken into a prefix and suffix entry.

#### Mapping Table

Specify if there is a mapping table for the index-organized table. Note that a mapping table is required for creating bitmap indexes on an index-organized table.

#### % Threshold

When an overflow segment is being used, it defines the maximum size of the portion of the row that is stored in the index block, as a percentage of block size.

#### Overflow Properties

Check to enable an overflow storage area.

**Note:** After saving the table, this option cannot be unchecked.

## **Overflow Column**

Specify the column to be put in a separate overflow data segment.

## **Overflow Tablespace**

Specify the tablespace in which the overflow segment to be stored.

## **Overflow Logging**


Specify whether creation of a database object will be logged in the redo log file (LOGGING) or not (NOLOGGING).

## **Overflow Physical Attributes**

Refer to [Editing Physical Attributes/Default Storage Characteristics](#).



## Oracle Views


Views are useful for allowing users to access a set of relations (tables) as if it were a single table, and limiting their access to just that. Views can also be used to restrict access to rows (a subset of a particular table).

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

### Create View

To create a new view

- Select anywhere on the object pane.
- Click the  **New View** from the object pane toolbar.  
or
- Right-click and select  **New View** from the popup menu.
- Edit view properties on the appropriate tabs of the View Designer.

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

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




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

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




**Apply to:** different schema {same connection}  
different schema {different connection} (Data Transfer tool will be activated)

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

To create a new view with modification as one of the existing views

- Select the view for modifying in the navigation pane/object pane.
- Right-click and select the  **Design View** from the popup menu.  
or
- Click the  **Design View** from the object pane toolbar.
- Modify view properties on the appropriate tabs of the View Designer.
- Click  **Save As**.

To create a new view with loading from a SQL file

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

## Create View Shortcut



To create a view shortcut

- Select the view for editing in the navigation pane/object pane.
- Right-click and select **Create Open View Shortcut...** from the popup menu.
- Define the location you wish your shortcut to be saved.

**Note:** This option is used to provide a convenient way for you to open your view for entering data directly (Grid View/Form View) without activating the main Navicat.

## Edit View

To edit the existing view (manage its SQL definition etc)



- Select the view for editing in the navigation pane/object pane.
- Right-click and select the  **Design View** from the popup menu.  
or
- Click the  **Design View** from the object pane toolbar.
- Edit view properties on the appropriate tabs of the View Designer.

To change the name of the view

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

## Open View

To open a view (manage view data)

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



## Maintain View

To maintain a view

- Select the view for maintaining in the object pane.
- Right-click and select the **Maintain** from the popup menu.
  - Compile

## Delete View

To delete a view

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

## Achieve View Information

To achieve a view information

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

## Oracle View Designer

**View Designer** is the basic Navicat tool for working with views. It allows you to create new view and edit the existing view definition (view name and the SELECT statement it implements).

- [Working with View Builder](#)
- [Editing View SQL Definition](#)
- [Setting Advanced View Properties](#)
- Editing View Comment
- View SQL Preview
- [View Preview](#)
- [View Explain](#)

## **Working with Oracle View Builder (Available only in Full Version)**

**View Builder** allows you to build views visually. It allows you to create and edit views without knowledge of SQL. See Query Builder for details.

## Editing Oracle View SQL Definition

The **Definition** tab allows you to edit the view definition as SQL statement (SELECT statement it implements).

Example:

```
SELECT
  HR.EMPLOYEES.EMPLOYEE_ID
FROM
  HR.EMPLOYEES
```

**Hint:** To customize the view of the editor and find out more features for sql editing, see [Editor View and More Features](#).

## Setting Advanced Oracle View Properties

The **Advanced** tab allows you to restrict the defining query of the view.

### **Restrict Query**

Specify the name. If you omit this identifier, then Oracle automatically assigns a name of the form SYS\_Cn, where n is an integer that makes the constraint name unique within the database.

### **Read Only**

Indicate that the table or view cannot be updated.


### **Check option**

Indicate that Oracle Database prohibits any changes to the table or view that would produce rows that are not included in the subquery.

### **Force On Create**


Check this option if you want to create the view regardless of whether the base tables of the view or the referenced object types exist or the owner of the schema containing the view has privileges on them.

## Oracle View Preview

To preview the result of the view, click  **Preview** on the toolbar. If the query statement is correct, the **Result** and **Message** tabs will be opened.

The **Result** tab displays the data of the view as a grid and the **Message** tab displays the message log.

## Oracle View Explain

To show the Explain Plan of the query, click  **Explain** on the toolbar. If the query statement is correct, the **Explain** tab opens with the columns in the PLAN\_TABLE.

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.

## Oracle View Viewer

**View Viewer** displays the view data as a grid. Data can be displayed in three modes:  **Grid View**,  **Form View** and **Text/Blob/BFile View**. See Data View for details.

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

- **Commit**

Make permanent all changes performed in the transaction.

**Hint:** The Commit button is visible only when **Auto Commit** is disabled under Option Settings.

- **Rollback**

Undo work done in the current transaction.

**Hint:** The Rollback button is visible only when **Auto Commit** is disabled under Option Settings.

- **Export Data**

Export data to MS Word, MS Excel, MS Access, TXT, DBF, HTML, SQL, RTF and more.

- **Filter Data**

Allow you to filter records by creating and applying filter criteria for the data grid.

- **Edit TEXT/BLOB/BFile**

Allow you to view and edit the content of TEXT, BLOB and BFile fields.

EMP\_DETAILS\_VIEW @HR (Basic Connection)

File Edit View Window Help

Commit Rollback Export Wizard Filter Wizard Grid View Form View


EMPLOYEE_ID	JOB_ID	MANAGER_ID	DEPARTMENT_ID
100	AD_PRES		(Null)
101	AD_VP		100
102	AD_VP		100
103	IT_PROG		102
104	IT_PROG		103
105	IT_PROG		103
106	IT_PROG		103
107	IT_PROG		103
108	FI_MGR		101
109	FI_ACCOUNT		108
110	FI_ACCOUNT		108
111	FI_ACCOUNT		108
112	FI_ACCOUNT		108
113	FI_ACCOUNT		108
114	PU_MAN		100
115	PU_CLERK		114
116	PU_CLERK		114
117	PU_CLERK		114
118	PU_CLERK		114
119	PU_CLERK		114
120	ST_MAN		100

SELECT \* FROM (SELECT "NAVICAT\_TABLE".\*, ROWNUM "NA Record 1 of 106 in Page 1

## Oracle Functions/Procedures



A procedure or function is a schema object that consists of a set of SQL statements and other PL/SQL constructs, grouped together, stored in the database, and run as a unit to solve a specific problem or perform a set of related tasks.

Procedures and functions are identical except that functions always return a single value to the caller, while procedures do not.

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

### Create Function/Procedure

To create a new function/procedure

- Select anywhere on the object pane.
- Click the  **New Function** from the object pane toolbar.  
or
- Right-click and select  **New Function** from the popup menu.
- Edit function/procedure properties on the appropriate tabs of the Function/Procedure Designer.

**Hint:** To create new function/procedure you can also right-click the Function node of the navigation pane and select the  **New Function** from the popup menu.

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

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

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



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

different schema {different connection} (Data Transfer tool will be activated)

- Select the function/procedure(s) for copying in the object pane.
- Drag and drop the chosen function/procedure(s) to the target database.
- Select one of the following options:
  - Copy here (Structure and Data)
  - Copy here (Structure only)
  - Cancel



## Edit Function/Procedure

To edit the existing function/procedure


- Select the function/procedure for editing in the navigation pane/object pane.
- Right-click and select the  **Design Function** from the popup menu or simply double-click the function/ procedure.  
or
- Click the  **Design Function** from the object pane toolbar.
- Edit function/procedure properties on the appropriate tabs of the Function/Procedure Designer.

## Run Function/Procedure

To run a function/procedure in the navigation pane/object pane

- Select the function/procedure for executing in the navigation pane/object pane.
- Click the  **Execute Function** from the object pane toolbar.  
or
- Right-click and select  **Execute Function** from the popup menu.
- View the returned data on the DBMS Output tab.

To run a function/procedure in the Function/Procedure Designer


- Create a new function/procedure or open the existing function/procedure.
- Click  **Run**.
- View the returned data on the DBMS Output tab.

## Debug Function/Procedure

To debug a function/procedure in the object pane

- Select the function/procedure for debugging in the object pane.
- Right-click and select the **Debug Function** from the popup menu.
- Debug the function/procedure in the Debugger.

To debug a function/procedure in the Function/Procedure Designer

- Create a new function/procedure or open the existing function/procedure.
- Click  **Debug**.
- Debug the function/procedure in the Debugger.



## Maintain Function/Procedure

To maintain a function/procedure

- Select the function/procedure for maintaining in the object pane.
- Right-click and select the **Maintain** from the popup menu.
  - Compile
  - Compile for Debug

## Delete Function/Procedure

To delete a function/procedure


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

## Achieve Function/Procedure Information

To achieve a function/procedure information

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

## Oracle Function Wizard

Click the  **New Function** from the object pane toolbar. The **Function Wizard** will pop up and it allows you to create a procedure/function easily.

- [Setting Routine Type](#)
- [Setting Parameters for Procedure/Function](#)
- [Setting Return Type for Function](#)

You are allowed not to show the **Function Wizard** when create new procedure/function.

**Hint:** Once uncheck the **Show wizard next time**, you can go to Options to enable it.

## Setting Oracle Routine Type

Specify the **Name** of the routine.

Select the type of the routine: **Procedure** or **Function**

## Setting Parameters for Oracle Procedure/Function

Define the parameter(s) of the procedure/function. Set the parameter **Name**, **Type**, **Mode** and **Default Value** under corresponding columns.

## Setting Return Type for Oracle Function

Select the **Return Type** from the list.

**Note:** Only function supports return type.

## Oracle Function/Procedure Designer

**Function/Procedure Designer** allows you to edit the existing function/procedure definition and more.







- [Editing Function/Procedure Definition](#)
- Function/Procedure SQL Preview
- [Viewing Function/Procedure Result](#)
- [Debugging Function/Procedure](#)

## Editing Oracle Function/Procedure Definition

Edit the function/procedure definition under the **Definition** tab.

The **Code Outline** window displays information about the function/procedure including parameter, code body, etc. To show the **Code Outline** window, simply choose View -> **Code Outline**.

**Note:** Available only in Full Version.


	Refresh the code outline.
	Show the detail view of the code outline.
	Turn mouse over highlight on or off.
	Expand the selected item.
	Collapse the selected item.
	Toggle sorting by position.

The SQL statements for creating procedures are CREATE PROCEDURE. In practice, it is best to use a CREATE OR REPLACE statement. The general form of these statements follows.

```
CREATE OR REPLACE
PROCEDURE "" AS
BEGIN
    -- routine body goes here, e.g.
    -- DBMS_OUTPUT.PUT_LINE('Navicat for Oracle');
END;
```

**Hint:** To customize the view of the editor and find out more features for sql editing, see Editor View and More Features.


## Viewing Oracle Function/Procedure Result

To run the function/procedure click  **Run** on the toolbar. If the SQL statement is correct, the statement will be executed and, if the statement is supposed to return data, the **Message** and **DBMS Output** tabs open with the message log and data returned by the function/procedure. If an error occurs while executing the function/procedure, execution stops, the appropriate error message is displayed.

If the function/procedure requires input parameter, the **Input Parameters** box will popup.

## Debugging Oracle Function/Procedure (Available only in Full Version)

To debug the function/procedure click  **Debug** on the toolbar to launch the [Oracle Debugger](#).


You can add/remove breakpoints for debugging by clicking  in the grey area beside each statement.

Enter the Input Parameters if necessary.

```
1 CREATE OR REPLACE
2 FUNCTION str2tbl( p_str IN VARCHAR2 ) RETURN
  charTableType
3 AS
4   l_str      LONG DEFAULT p_str || ',';
5   l_n        NUMBER;
6   l_data     charTableType := charTableType();
7 BEGIN
8   LOOP
9     l_n := instr( l_str, ',' );
10    exit when (nvl(l_n,0) = 0);
11    l_data.extend;
12    l_data( l_data.count ) := ltrim(rtrim(
  substr(l_str,1,l_n-1));
13    l_str := substr( l_str, l_n+1 );
14  END LOOP;
15  RETURN l_data;
16 END;
17
```



## Oracle Database Links

Database link is a named schema object that describes a path from one database to another and are implicitly used when a reference is made to a global object name in a distributed database. After you have created a database link, you can use it to refer to tables and views on the other database.

Just simply click -> **Database Link** to open an object pane for **Database Link**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new and delete the selected database link.



### Create Database Link

To create a new database link

- Select anywhere on the object pane.
- Click the  **New Database Link** from the object pane toolbar.  
or
- Right-click and select  **New Database Link** from the popup menu.
- Edit database link properties on the appropriate tabs of the Database Link Designer.

### Delete Database Link

To delete a database link

- Select the database link for deleting in the object pane.
- Right-click and select the  **Delete Database Link** from the popup menu.  
or
- Click the  **Delete Database Link** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Database Link Information

To achieve a database link information

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

## Oracle Database Link Designer

**Database Link Designer** is the basic Navicat tool for working with database links. It allows you to create new database link.

- [Editing Database Link General](#)
- Database Link SQL Preview

## Edit Oracle Database Link General

### Service Name

Specify the service name of a remote database.

### User Name

The user name used to connect to the remote database using a fixed user database link.

### Password

The password for connecting to the remote database.

### Current user


With this option checked, a current user database link is created. The current user must be a global user with a valid account on the remote database.

### Shared

Fill in **Authentication username** and **Authentication password** when Shared option is enabled.



## Oracle Indexes

Index provides a faster access path to table data. It is created using one or more columns of a table to speed SQL statement execution on that table.




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

### Create Index

To create a new index



- Select anywhere on the object pane.
- Click the  **New Index** from the object pane toolbar.  
or
- Right-click and select  **New Index** from the popup menu.
- Edit index properties on the appropriate tabs of the Index Designer.

To create a new index with modification as one of the existing index

- Select the index for modifying in the object pane.
- Right-click and select the  **Design Index** from the popup menu or simply double-click the index.  
or
- Click the  **Design Index** from the object pane toolbar.
- Modify index properties on the appropriate tabs of the Index Designer.
- Click  **Save As**.

### Edit Index

To edit the existing index (manage its properties etc)

- Select the index for editing in the object pane.
- Right-click and select the  **Design Index** from the popup menu or simply double-click the index.  
or
- Click the  **Design Index** from the object pane toolbar.
- Edit index properties on the appropriate tabs of the Index Designer.

To change the name of the index

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



## Maintain Index

To maintain an index

- Select the index for maintaining in the object pane.
- Right-click and select the **Maintain** from the popup menu.
  - Rebuild
  - Make Unusable
  - Coalesce
  - Compute Statistics
  - Monitoring Usage
  - No Monitoring Usage

## Delete Index

To delete an index

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

## Achieve Index Information

To achieve an index information

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

## Oracle Index Designer

**Index Designer** is the basic Navicat tool for working with indexes. It allows you to create new index and edit the existing index properties.

- [Editing Index General](#)
- [Editing Advanced Index Properties](#)
- Index SQL Preview

## Editing Oracle Index General

### Type

The types of the index.

#### Normal

Normal Index (A B-tree index)

#### Unique

Unique Index (No two rows of a table have duplicate values in the key columns)

#### Bitmap

Bitmap Index (A bitmap for each key value)

#### Domain

Domain Index (Instances of an application-specific index)

#### Cluster

Cluster Index

## Oracle Normal Index

A normal index does not impose restrictions on the column values.

### Type

Choose between **Normal**, Unique, Bitmap, Domain and Cluster.

### Table schema

The schema that contains the index.

### Table name

The table name.

### Fields

Use the **Name** dropdown list to select the field name and **Descend** dropdown list to define the order of the index (ASC or DESC).

## Oracle Unique Index

A unique index indicates that no two rows of a table have duplicate values in the key columns.

### Type

Choose between Normal, **Unique**, Bitmap, Domain and Cluster.

### Table schema

The schema that contains the index.

### Table name

The table name.

### Fields

Use the **Name** dropdown list to select the field name and **Descend** dropdown list to define the order of the index (ASC or DESC).

## Oracle Bitmap Index

A bitmap index created with a bitmap for each distinct key, rather than indexing each row separately. Bitmap indexes store the rowids associated with a key value as a bitmap. Each bit in the bitmap corresponds to a possible rowid.

### Type

Choose between Normal, Unique, **Bitmap**, Domain and Cluster.

### Table schema

The schema that contains the index.

### Table name

The table name.

### Fields

Use the **Name** dropdown list to select the field name and **Descend** dropdown list to define the order of the index (ASC or DESC).

### **Bitmap Join Index**

In addition to a bitmap index on a single table, you can create a bitmap join index, which is a bitmap index for the join of two or more tables. A bitmap join index is a space efficient way of reducing the volume of data that must be joined by performing restrictions in advance.

#### **Fields**

Use the **Schema**, **Table** and **Name** dropdown lists to select the schema, table and field name and **Descend** dropdown list to define the order of the index (ASC or DESC).

#### **Bitmap Join**

Use the **InnerSchema**, **InnerTable**, **InnerField**, **OuterSchema**, **OuterTable** and **OuterField** dropdown lists to select joined schemas, tables and fields respectively.

## Oracle Domain Index

A domain index is an index designed for a specialized domain, such as spatial or image processing. Users can build a domain index of a given type after the designer creates the indextype.

### Type

Choose between Normal, Unique, Bitmap, **Domain** and Cluster.

### Table schema

The schema that contains the index.

### Table name

The table name.

### Column

The column which the index is based.

### Index Type

#### Schema

The schema of the indextype.

#### Type

Select the created or built-in indextypes.

#### Parameters

Information about the path table and about the secondary indexes corresponding to the components of XMLIndex. The maximum length of the parameter string is 1000 characters.

## Oracle Cluster Index

A cluster index is an index designed for a cluster.

### Type

Choose between Normal, Unique, Bitmap, Domain and **Cluster**.

### Table schema

The schema that contains the index.

### Cluster name

The name of the cluster.

## Editing Advanced Oracle Index Properties

### Unusable

An unusable index must be rebuilt, or dropped and re-created, before it can be used.

### Tablespace

The name of the tablespace to hold the index.

### Compress

To enable key compression, which eliminates repeated occurrence of key column values and may substantially reduce storage.

**Note:** No compression for Bitmap Indexes.

### Parallel

The creation of the index will be parallelized.

### Reverse

To store the bytes of the index block in reverse order, excluding the rowid.

## Logging

### Logging

The creation of the index will be logged in the redo log file.

### No Logging

The creation of the index will be not logged in the redo log file.

## Visibility

### Visible

Specify the index is visible to the optimizer.

### Invisible

Specify the index is invisible to the optimizer.

## Create / Rebuild Option

### Online

To indicate that DML operations on the table will be allowed during creation of the index.

## **No Sort**


To indicate to the database that the rows are already stored in the database in ascending order, so that Oracle Database does not have to sort the rows when creating the index.

## **Physical Attributes**

Set the physical attributes of an index.



## Oracle Java

Java is an object-oriented programming language efficient for application-level programs. You can write and load applications within the database.




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

## Create Java

To create a new Java



- Select anywhere on the object pane.
- Click the  **New Java** from the object pane toolbar.  
or
- Right-click and select  **New Java** from the popup menu.
- Edit Java properties on the appropriate tabs of the Java Designer.

To create a new Java with modification as one of the existing Java

- Select the Java for modifying in the object pane.
- Right-click and select the  **Design Java** from the popup menu or simply double-click the Java.  
or
- Click the  **Design Java** from the object pane toolbar.
- Modify Java properties on the appropriate tabs of the Java Designer.
- Click  **Save As**.

## Edit Java

To edit the existing Java(manage its general, advanced etc)

- Select the Java for editing in the object pane.
- Right-click and select the  **Design Java** from the popup menu or simply double-click the Java.  
or
- Click the  **Design Java** from the object pane toolbar.
- Edit Java properties on the appropriate tabs of the Java Designer.



## Maintain Java

To maintain a Java

- Select the Java for maintaining in the object pane.
- Right-click and select the **Maintain** from the popup menu.
  - Compile or Resolve
  - Set AuthID Current User
  - Set AuthID Definer

## Delete Java

To delete a Java

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

## Achieve Java Information

To achieve a Java information

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

## Oracle Java Designer

**Java Designer** is the basic Navicat tool for working with Java. It allows you to create new Java and edit the existing Java properties.

- [Editing Java General](#)
- [Setting Advanced Java Properties](#)
- Java SQL Preview

## Editing Oracle Java General

You can create a Java source, class, or resource using the Java Designer.

- [Java Source](#)
- [Java Class](#)
- [Java Resource](#)

## Oracle Java Source

### Type

Choose between **Java Source**, Java Class and Java Resource.

### Create from

#### **BFile**

Select the **Directory** and type the **Server Filename**.

#### **Load from file**

Browse the **File Path** of Java source file.

#### **Plain source**

Type the source code in the **Source** box.

## Oracle Java Class

### Type

Choose between Java Source, **Java Class** and Java Resource.

### Create from

#### **BFile**

Select the **Directory** and type the **Server Filename**.

#### **Load from file**

Browse the **File Path** of the Java class file.

## Oracle Java Resource

### Type

Choose between Java Source, Java Class and **Java Resource**.

### Create from

#### **BFile**

Select the **Directory** and type the **Server Filename**.

#### **Load from file**

Browse the **File Path** of the Java resource file.

## Setting Advanced Oracle Java Properties

### Invoker Rights

Select **CURRENT\_USER** to indicate that the methods of the class execute with the privileges of CURRENT\_USER or **DEFINER** indicates that the methods of the class execute with the privileges of the owner of the schema in which the class resides, and that external names resolve in the schema where the class resides.

### Resolver

Specify a mapping of the fully qualified Java name to a Java schema object.

#### **Compile or Resolve**


Check this to specify that Oracle Database should attempt to resolve the Java schema object that is created if this statement succeeds.

#### **No Force**

Check this to roll back the results of the CREATE command of Java if you have enabled Compile or Resolve and the resolution or compilation fails. If you do not specify this option, then Oracle Database takes no action if the resolution or compilation fails, and the created schema object remains.



## Oracle Materialized Views

Materialized view is a schema object that can be used to summarize, compute, replicate, and distribute data.




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

### Create Materialized View




To create a new materialized view

- Select anywhere on the object pane.
- Click the  **New Materialized View** from the object pane toolbar.  
or
- Right-click and select  **New Materialized View** from the popup menu.
- Edit materialized view properties on the appropriate tabs of the Materialized View Designer.

To create a new materialized view with modification as one of the existing materialized view

- Select the materialized view for modifying in the object pane.
- Right-click and select the  **Design Materialized View** from the popup menu.  
or
- Click the  **Design Materialized View** from the object pane toolbar.
- Modify materialized view properties on the appropriate tabs of the Materialized View Designer.
- Click  **Save As**.

To create a new materialized view with loading from a SQL file

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

## Create Materialized View Shortcut

To create a materialized view shortcut

- Select the materialized view for editing in the object pane.
- Right-click and select **Create Open Materialized View Shortcut...** from the popup menu.
- Define the location you wish your shortcut to be saved.

**Note:** This option is used to provide a convenient way for you to open your materialized view for editing data directly (Grid View/Form View) without activating the main Navicat.



## Edit Materialized View

To edit the existing materialized view (manage its properties etc)

- Select the materialized view for editing in the object pane.
- Right-click and select the  **Design Materialized View** from the popup menu.  
or
- Click the  **Design Materialized View** from the object pane toolbar.
- Edit materialized view properties on the appropriate tabs of the Materialized View Designer.

## Open Materialized View

To open a materialized view (manage materialized view data)

- Select the materialized view for opening in the object pane.
- Right-click and select the  **Open Materialized View** from the popup menu or simply double-click the materialized view.  
or
- Click the  **Open Materialized View** from the object pane toolbar.



## Maintain Materialized View

To maintain a materialized view

- Select the materialized view for maintaining in the object pane.
- Right-click and select the **Maintain** from the popup menu.
  - Enable Row Movement
  - Shrink
  - Compile
  - Force Refresh

## Delete Materialized View

To delete a materialized view

- Select the materialized view for deleting in the object pane.
- Right-click and select the  **Delete Materialized View** from the popup menu.  
or
- Click the  **Delete Materialized View** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Materialized View Information

To achieve a materialized view information

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

## Oracle Materialized View Designer

**Materialized View Designer** is the basic Navicat tool for working with materialized views. It allows you to create new materialized view and edit the existing materialized view properties.

- [Working with Materialized View Builder](#)
- [Editing Materialized View SQL Definition](#)
- [Setting Advanced Materialized View Properties](#)
- Editing Materialized View Comment
- Materialized View SQL Preview

## **Working with Oracle Materialized View Builder (Available only in Full Version)**

**View Builder** allows you to build views visually. It allows you to create and edit materialized views without knowledge of SQL. See Query Builder for details.

## Editing Oracle Materialized View SQL Definition

The **Definition** tab allows you to edit the view definition as SQL statement (SELECT statement it implements).

Example:

```
SELECT
  HR.EMPLOYEES.EMPLOYEE_ID
FROM
  HR.EMPLOYEES
```

**Hint:** To customize the view of the editor and find out more features for sql editing, see [Editor View and More Features](#).

## Setting Advanced Oracle Materialized View Properties

### Refresh Options

#### When

##### **Demand**

The materialized view will be refreshed on demand by calling one of the three DBMS\_MVIEW refresh procedures.

##### **Commit**

A fast refresh is to occur whenever the database commits a transaction that operates on a master table of the materialized view.

##### **Automatic**

The database automatically refresh the materialized view with the automatic refresh time.

##### **Never**

The materialized view will not be refreshed with any Oracle Database refresh mechanism or packaged procedure.

#### Method

##### **FORCE**

When a refresh occurs, Oracle Database will perform a fast refresh if one is possible or a complete refresh if fast refresh is not possible.

##### **FAST**

A incremental refresh method, which performs the refresh according to the changes that have occurred to the master tables.

##### **COMPLETE**

A complete refresh method, which is implemented by executing the defining query of the materialized view.

#### Start On

A datetime expression for the first automatic refresh time.

#### Next

A datetime expression for calculating the interval between automatic refreshes.

**Note:** To edit the datetime, just simply click ... and choose / enter the desired data.

## Type

### **Primary Key**

A primary key materialized view. This is the default.

### **Row ID**

A rowid materialized view.

## Rollback Segment

### **Master**

The remote rollback segment is used at the remote master site for the individual materialized view.

### **Local**

The remote rollback segment is used for the local refresh group that contains the materialized view. This is the default.

## Constraints

### **Enforced**

Oracle Database use enforced constraints during the refresh operation.

### **Trusted**

Oracle Database use dimension and constraint information that has been declared trustworthy by the database administrator but that has not been validated by the database.

## Create Options

### **No Index**

Check this to suppress the creation of the default index.

## Build Type

### **Immediate**

The materialized view is to be populated immediately. This is the default.

## **Deferred**

The materialized view is to be populated by the next refresh operation.

## **Prebuilt**

To register an existing table as a preinitialized materialized view.

## **Materialized View Options**

### **Reduced Precision**

#### **With**

To authorize the loss of precision that will result if the precision of the table or materialized view columns do not exactly match the precision returned by subquery.

#### **Without**

To require that the precision of the table or materialized view columns match exactly the precision returned by subquery, or the create operation will fail. This is the default.

### **Compress**

Data segments are compressed to reduce disk and memory use.

### **Parallel**

Choose **NOPARALLEL** for serial execution or **PARALLEL** if you want Oracle to select a degree of parallelism equal to the number of CPUs available on all participating instances times the value of the `PARALLEL_THREADS_PER_CPU` initialization parameter.

### **With Degree**

Set the default degree of parallelism for queries and DML on the materialized view after creation.

### **Logging**

Choose **LOGGING** for logging the creation of Materialized view in the redo log file.  
Choose **NOLOGGING** for no logging.

### **Tablespace**

Choose the tablespace in which the materialized view is to be created.

### **Physical Attributes**

Set the physical attributes of the materialized view.

## **Enable Cache**

The blocks retrieved for the table are placed at the most recently used end of the least recently used (LRU) list in the buffer cache when a full table scan is performed.

## **For Update**

Check this to allow a subquery, primary key, object, or rowid materialized view to be updated. When used in conjunction with Advanced Replication, these updates will be propagated to the master.

## **Enable Query Rewrite**

The materialized view is used for query rewrite.

## **Using Index Clause**


### **Tablespace**

Choose the tablespace of the index.

### **Physical Attributes**


Set the physical attributes for the default index Oracle Database uses to maintain the materialized view data.

## Oracle Materialized View Preview

To preview the result of the view, click  **Preview** on the toolbar. If the query statement is correct, the **Result** and **Message** tabs will be opened.

The **Result** tab displays the data of the view as a grid and the **Message** tab displays the message log.


## Oracle Materialized View Explain

To show the Explain Plan of the query, click  **Explain** on the toolbar. If the query statement is correct, the **Explain** tab opens with the columns in the PLAN\_TABLE.

Column	Description
Operation	Name of the internal operation performed in this step. In the first row generated for a statement.
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.



## Oracle Materialized View Logs

Materialized view log is a schema object that records changes to a master table's data so that a [Materialized View](#) defined on the master table can be refreshed incrementally.




Just simply click -> **Materialized View Log** to open an object pane for **Materialized View Log**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected materialized view log.

### Create Materialized View Log

To create a new materialized view log



- Select anywhere on the object pane.
- Click the  **New Materialized View Log** from the object pane toolbar.  
or
- Right-click and select  **New Materialized View Log** from the popup menu.
- Edit materialized view log properties on the appropriate tabs of the Materialized View Log Designer.

To create a new materialized view log with modification as one of the existing materialized view log

- Select the materialized view log for modifying in the object pane.
- Right-click and select the  **Design Materialized View Log** from the popup menu or simply double-click the materialized view log.  
or
- Click the  **Design Materialized View Log** from the object pane toolbar.
- Modify materialized view log properties on the appropriate tabs of the Materialized View Log Designer.
- Click  **Save As**.



## Edit Materialized View Log

To edit the existing materialized view log (manage its general, advance, etc)

- Select the materialized view log for editing in the object pane.
- Right-click and select the  **Design Materialized View Log** from the popup menu or simply double-click the materialized view log.  
or
- Click the  **Design Materialized View Log** from the object pane toolbar.
- Edit materialized view log properties on the appropriate tabs of the Materialized View Log Designer.

## Open Materialized View Log Table

To open a materialized view log table (manage materialized view log data)

- Select the materialized view log table for opening in the object pane.
- Right-click and select the  **Open Materialized View Log Table** from the popup menu  
or
- Click the  **Open Materialized View Log Table** from the object pane toolbar.



## Maintain Materialized View Log

To maintain a materialized view log

- Select the materialized view log table for maintaining in the object pane.
- Right-click and select the **Maintain** from the popup menu.
  - Enable Row Movement
  - Disable Row Movement
  - Shrink Space

## Delete Materialized View Log

To delete a materialized view log

- Select the materialized view log for deleting in the object pane.
- Right-click and select the  **Delete Materialized View Log** from the popup menu.  
or
- Click the  **Delete Materialized View Log** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Materialized View Log Information

To achieve a materialized view log information

- Select the materialized view log in the object pane.
- Right-click the selected materialized view log and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## Oracle Materialized View Log Designer

**Materialized View Log Designer** is the basic Navicat tool for working with materialized view logs. It allows you to create new materialized view log and edit the existing materialized view log definition.

- [Editing Materialized View Log General](#)
- Materialized View Log SQL Preview

## Editing Oracle Materialized View Log General

Edit the materialized view log general properties under the **General** tab.

### Table

The table of the materialized view log.

### Tablespace

The tablespace of the materialized view log.

### Logging

To specify either **LOGGING** or **NOLOGGING** to establish the logging characteristics for the materialized view log.

### Cache

#### **CACHE**

The blocks retrieved for this log are placed at the most recently used end of the least recently used (LRU) list in the buffer cache when a full table scan is performed.

#### **NOCACHE**

The blocks are placed at the least recently used end of the LRU list. This is the default.

### New Values

#### **INCLUDING**

To save both new and old values in the log.

#### **EXCLUDING**

To disable the recording of new values in the log.

#### **Parallel With Degree**

To determine the number of parallel threads used in the parallel operation.

### [Physical Attributes](#)

Set the physical attributes of a materialized view log.

## With

### **Object ID**

The system-generated or user-defined object identifier of every modified row should be recorded in the materialized view log.

### **Primary Key**

The primary key of all rows changed should be recorded in the materialized view log.

### **Row ID**

The rowid of all rows changed should be recorded in the materialized view log.

### **Sequence**


A sequence value providing additional ordering information should be recorded in the materialized view log.

### **Fields**

Choose the fields whose values you want to be recorded in the materialized view log for all rows that are changed.



## Oracle Packages

Packages are encapsulated collections of related procedures, stored functions, and other program objects stored together in the database. Package bodies, specified subsequently, defines these objects. An package consists of two parts: a specification and a body.

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



### Create Package (Package Specification)

To create a new package

- Select anywhere on the object pane.
- Click the  **New Package** from the object pane toolbar.  
or
- Right-click and select  **New Package** from the popup menu.
- Edit package properties on the appropriate tabs of the Package Designer.



### Edit Package

To edit the existing package (manage its definition etc)

- Select the package for editing in the object pane.
- Right-click and select the  **Design Package** from the popup menu or simply double-click the package.  
or
- Click the  **Design Package** from the object pane toolbar.
- Edit package properties on the appropriate tabs of the Package Designer.


### Delete Package

To delete a package

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


## Create Package Body (Package Body)

To create a new package body

- Select the package for modifying in the object pane.
- Right-click and select  **New Package Body** from the popup menu.
- Edit package body properties on the appropriate tabs of the Package Body Designer.


## Edit Package Body

To edit the existing package body (manage its definition etc)

- Select the package for editing in the object pane.
- Right-click and select the  **Design Package Body** from the popup menu.
- Edit package body properties on the appropriate tabs of the Package Body Designer.



## Delete Package Body

To delete a package body


- Select the package for deleting in the object pane.
- Right-click and select the  **Delete Package Body** from the popup menu.
- Confirm deleting in the dialog window.

## Run Package

To run a package in the object pane

- Select the package for executing in the object pane.
- Click the  **Execute Package** from the object pane toolbar.  
or
- Right-click and select  **Execute Package** from the popup menu.
- View the returned data on the DBMS Output tab.

To run a package in the Package Designer


- Create a new package/open the existing package.
- Click  **Run**.
- View the returned data on the DBMS Output tab.

## Debug Package

To debug a package h

- Select the package for debugging in the object pane.
- Right-click and select the **Debug Package** from the popup menu.
- Debug the package in the Debugger.

To debug a package in the Package Designer

- Create a new package/open the existing package.
- Click  **Debug**.
- Debug the package in the Debugger.

## Maintain Package

To maintain a package

- Select the package for maintaining in the object pane.
- Right-click and select the **Maintain** from the popup menu.
  - Compile
  - Compile Debug

## Achieve Package Information


To achieve a package information

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

## Oracle Package Designer

**Package Designer** is the basic Navicat tool for working with packages. It allows you to create new package and edit the existing package definition.

- [Editing Package Definition](#)
- Package SQL Preview
- [Viewing Package Result](#)
- [Debugging Package](#)







After saving the package, you can edit the Package Body. Just click  **Design Package Body** to open the Package Body Designer.

## Editing Oracle Package Definition

Edit the package definition under the **Definition** tab.

The **Code Outline** window displays information about the package including function, procedure, parameter, code body, etc. To show the **Code Outline** window, simply choose View -> **Code Outline**.

**Note:** Available only in Full Version.

	Refresh the code outline.
	Show the detail view of the code outline.
	Turn mouse over highlight on or off.
	Expand the selected item.
	Collapse the selected item.
	Toggle sorting by position.

The SQL statements for creating packages are CREATE PACKAGE. In practice, it is best to use a CREATE OR REPLACE statement. The general form of these statements follows.

```
CREATE OR REPLACE
PACKAGE /*PACKAGE NAME*/ AS
    /* TODO enter package declarations (types, exceptions, methods etc) here */
end;
```

**Hint:** To customize the view of the editor and find out more features for sql editing, see Editor View and More Features.

## Oracle Package Body Designer

**Package Body Designer** is the basic Navicat tool for working with package bodies. It allows you to create new package body and edit the existing package body definition.

- [Editing Package Body Definition](#)
- Package Body SQL Preview
- [Viewing Package Result](#)
- [Debugging Package](#)







To edit the Package Specification, click  **Design Package Specification** to open the Package Designer.

## Editing Oracle Package Body Definition

Edit the package body definition under the **Definition** tab.

The **Code Outline** window displays information about the package body including function, procedure, parameter, code body, etc. To show the **Code Outline** window, simply choose View -> **Code Outline**.

**Note:** Available only in Full Version.


	Refresh the code outline.
	Show the detail view of the code outline.
	Turn mouse over highlight on or off.
	Expand the selected item.
	Collapse the selected item.
	Toggle sorting by position.

The SQL statements for creating procedures are CREATE PACKAGE BODY. In practice, it is best to use a CREATE OR REPLACE statement. The general form of these statements follows.

```
CREATE OR REPLACE
PACKAGE BODY /*PACKAGE NAME*/ AS
....
END;
```


**Hint:** To customize the view of the editor and find out more features for sql editing, see Editor View and More Features.


## Viewing Oracle Package Result

To run the package click  **Run** on the toolbar. If the SQL statement is correct, the statement will be executed and, if the statement is supposed to return data, the **Message** and **DBMS Output** tabs open with the message log and data returned by the package. If an error occurs while executing the package, execution stops, the appropriate error message is displayed.

Select the function/procedure and enter the parameter(s) if the function/procedure has input parameter(s).

## Debugging Oracle Package (Available only in Full Version)

To debug the package click  **Debug** on the toolbar to launch the [Oracle Debugger](#).


You can add/remove breakpoints for debugging by clicking  in the grey area beside each statement.

Select the function/procedure and enter the parameter(s) if the function/procedure has input parameter(s).

```
1  CREATE OR REPLACE
2  PACKAGE emp_mgmt12 AS
3  FUNCTION hire (last_name VARCHAR2,
4  job_id VARCHAR2,
5  manager_id NUMBER, salary NUMBER,
6  commission_pct NUMBER, department_id
7  NUMBER)
8  RETURN NUMBER;
9  FUNCTION create_dept (department_id
10 NUMBER, location_id NUMBER)
11 RETURN NUMBER;
12 PROCEDURE remove_emp (employee_id NUMBER)
13 ;
14 PROCEDURE remove_dept (department_id
15 NUMBER);
16 PROCEDURE increase_sal (employee_id
17 NUMBER, salary_incr NUMBER);
18 PROCEDURE increase_comm (employee_id
19 NUMBER, comm incr NUMBER);
```



## Oracle Sequences

Sequence involves creating and initializing a new special single-row table. It is usually used to generate unique identifiers for rows of a table.




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

### Create Sequence

To create a new sequence



- Select anywhere on the object pane.
- Click the  **New Sequence** from the object pane toolbar.  
or
- Right-click and select  **New Sequence** from the popup menu.
- Edit sequence properties on the appropriate tabs of the Sequence Designer.

To create a new sequence with modification as one of the existing sequence

- Select the sequence for modifying in the object pane.
- Right-click and select the  **Design Sequence** from the popup menu or simply double-click the sequence.  
or
- Click the  **Design Sequence** from the object pane toolbar.
- Modify sequence properties on the appropriate tabs of the Sequence Designer.
- Click  **Save As**.



### Edit Sequence

To edit the existing sequence(manage its general etc)

- Select the sequence for editing in the object pane.
- Right-click and select the  **Design Sequence** from the popup menu or simply double-click the sequence.  
or
- Click the  **Design Sequence** from the object pane toolbar.
- Edit sequence properties on the appropriate tabs of the Sequence Designer.

## Delete Sequence

To delete a sequence

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

## Achieve Sequence Information

To achieve a sequence information

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

## Oracle Sequence Designer

**Sequence Designer** is the basic Navicat tool for working with sequence. It allows you to create new sequence and edit the existing sequence properties.

- [Editing Sequence General](#)
- Sequence SQL Preview

## Editing Oracle Sequence General

### Increment

To specify which value is added to the current sequence value to create a new value. A positive value will make an ascending sequence, a negative one a descending sequence. The default value is 1.

### Start with

To specify the first sequence number to be generated.

### Minimum

The minimum value a sequence can generate.

### Maximum

The maximum value for the sequence.

### Cache

To specify how many values of the sequence the database preallocates and keeps in memory for faster access. The minimum value for this parameter is 2. (The database caches 20 sequence numbers by default.)

#### No Cache

This option indicates that values of the sequence are not preallocated.

#### Cycled


This option allows the sequence continues to generate values after reaching either its maximum or minimum value. After an ascending sequence reaches its maximum value, it generates its minimum value. After a descending sequence reaches its minimum, it generates its maximum value.

#### Order

This option guarantees that sequence numbers are generated in order of request.



## Oracle Synonyms

Synonym is an alias for any table, view, materialized view, synonym, procedure, function, package, type, Java class schema object, user-defined object type, or another synonym. Because a synonym is simply an alias, it requires no storage other than its definition in the data dictionary.




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

### Create Synonym

To create a new synonym



- Select anywhere on the object pane.
- Click the  **New Synonym** from the object pane toolbar.  
or
- Right-click and select  **New Synonym** from the popup menu.
- Edit synonym properties on the appropriate tabs of the Synonym Designer.

To create a new synonym with modification as one of the existing synonym

- Select the synonym for modifying in the object pane.
- Right-click and select the  **Design Synonym** from the popup menu or simply double-click the synonym.  
or
- Click the  **Design Synonym** from the object pane toolbar.
- Modify synonym properties on the appropriate tabs of the Synonym Designer.
- Click  **Save As**.

## Edit Synonym

To edit the existing synonym (manage its general etc)



- Select the synonym for editing in the object pane.
- Right-click and select the  **Design Synonym** from the popup menu or simply double-click the synonym.  
or
- Click the  **Design Synonym** from the object pane toolbar.
- Edit synonym properties on the appropriate tabs of the Synonym Designer.

To change the name of the synonym

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

## Delete Synonym

To delete a synonym

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

## Achieve Synonym Information

To achieve a synonym information

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

## Oracle Synonym Designer

**Synonym Designer** is the basic Navicat tool for working with synonym. It allows you to create new synonym and edit the existing synonym properties.

- [Editing Synonym General](#)
- Synonym SQL Preview

## Editing Oracle Synonym General

### **Object DataBase Link**

A complete or partial database link to create a synonym for a schema object on a remote database where the object is located.

### **Object Schema**

The schema in which the object resides.

### **Object Type**

The object type.


### **Object**

The object for which the synonym is created.

## Oracle Triggers



Triggers are similar to procedures. A trigger stored in the database can include SQL and PL/SQL or Java statements to run as a unit and can invoke procedures.

See [Triggers](#) for details.




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

### Create Trigger

To create a new trigger



- Select anywhere on the object pane.
- Click the  **New Trigger** from the object pane toolbar.  
or
- Right-click and select  **New Trigger** from the popup menu.
- Edit trigger properties on the appropriate tabs of the Trigger Designer.

To create a new trigger with modification as one of the existing trigger

- Select the trigger for modifying in the object pane.
- Right-click and select the  **Design Trigger** from the popup menu or simply double-click the trigger.  
or
- Click the  **Design Trigger** from the object pane toolbar.
- Modify trigger properties on the appropriate tabs of the Trigger Designer.
- Click  **Save As**.

## Edit Trigger

To edit the existing trigger (manage its general, advance, etc)

- Select the trigger for editing in the object pane.
- Right-click and select the  **Design Trigger** from the popup menu or simply double-click the trigger.  
or
- Click the  **Design Trigger** from the object pane toolbar.
- Edit trigger properties on the appropriate tabs of the Trigger Designer.

To change the name of the trigger

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



## Maintain Trigger

To maintain a trigger

- Select the trigger for maintaining in the object pane.
- Right-click and select the **Maintain** from the popup menu.
  - Enable
  - Disable
  - Compile
  - Compile for Debug

## Delete Trigger

To delete a trigger

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

## Achieve Trigger Information

To achieve a trigger information

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

## Oracle Trigger Designer

**Trigger Designer** is the basic Navicat tool for working with triggers. It allows you to create new trigger and edit the existing trigger definition.

- [Editing Trigger General](#)
- [Setting Advanced Trigger Properties](#)
- [Editing Trigger Definition](#)
- Trigger SQL Preview

## Editing Oracle Trigger General

### **TABLE**

To define the trigger on the selected table.

### **VIEW**

To define the trigger on the selected view.

### **SCHEMA**

To define the trigger on the selected schema.

### **DATABASE**

To define the trigger on the entire database.

## Oracle Table Trigger

### Trigger Type

Choose the type of trigger: **TABLE**, VIEW, SCHEMA and DATABASE

#### Enable

An enabled trigger runs its trigger action if a triggering statement is issued and the trigger restriction (if any) evaluates to true.

### Table Schema

The table schema of the trigger.

### Table name

The table you wish to create the trigger.

#### Compound

A compound trigger is a single trigger on a table that allows you to specify actions for each of four timing points:

Timing Point	Section
Before the triggering statement executes	BEFORE STATEMENT
After the triggering statement executes	AFTER STATEMENT
Before each row that the triggering statement affects	BEFORE EACH ROW
After each row that the triggering statement affects	AFTER EACH ROW

**Note:** Support from Oracle 11g or later and you can edit the SQL in Trigger Definition.

### Fire

When defining a trigger, you can specify the trigger timing - whether the trigger action is to be run **BEFORE** or **AFTER** the triggering statement.

### For Each

Oracle Database fires a **ROW** trigger once for each row that is affected by the triggering statement and fires a **STATEMENT** trigger only once when the triggering statement is issued if the optional trigger constraint is met.

### When

To specify the trigger condition, which is a SQL condition that must be satisfied for the database to fire the trigger.

## On Event

It indicates the kind of statement that activates the trigger.

### **Insert**

The trigger is activated whenever adding a row to a table or adds an element to a nested table.

### **Delete**

The trigger is activated whenever removing a row from the table or removes an element from a nested table.

### **Update**

The trigger is activated whenever changing a value in one of the fields selected in **Update Of Fields**.

## Oracle View Trigger

### Trigger Type

Choose the type of trigger: TABLE, **VIEW**, SCHEMA and DATABASE

#### **Enable**

An enabled trigger runs its trigger action if a triggering statement is issued and the trigger restriction (if any) evaluates to true.

### Table Schema

The table schema of the trigger.

### Table name

The table you wish to create the trigger.

#### **Nested Table Field**

To select the nested table field.

#### **Compound**

To specify the Instead Of Trigger.

**Note:** Support from Oracle 11g or later and you can edit the SQL in Trigger Definition.

### On Event

It indicates the kind of statement that activates the trigger.

#### **Insert**

The trigger is activated whenever adding a row to a table or adds an element to a nested table.

#### **Delete**

The trigger is activated whenever removing a row from the table or removes an element from a nested table.

#### **Update**

The trigger is activated whenever changing a value in a row.

## Oracle Schema Trigger

### Trigger Type

Choose the type of trigger: TABLE, VIEW, **SCHEMA** and DATABASE

### Enable

An enabled trigger runs its trigger action if a triggering statement is issued and the trigger restriction (if any) evaluates to true.

### Table Schema

The table schema of the trigger.

### Fire

When defining a trigger, you can specify the trigger timing - whether the trigger action is to be run **BEFORE** or **AFTER** the triggering statement.

### When

To specify the trigger condition, which is a SQL condition that must be satisfied for the database to fire the trigger.

### Events

The kind of statement that activates the trigger.

## Oracle Database Trigger

### Trigger Type

Choose the type of trigger: TABLE, VIEW, SCHEMA and **DATABASE**

### Enable

An enabled trigger runs its trigger action if a triggering statement is issued and the trigger restriction (if any) evaluates to true.

### Fire

When defining a trigger, you can specify the trigger timing - whether the trigger action is to be run **BEFORE** or **AFTER** the triggering statement.

### When

To specify the trigger condition, which is a SQL condition that must be satisfied for the database to fire the trigger.

### Events

The kind of statement that activates the trigger.

## Setting Advanced Oracle Trigger Properties

The **Advanced** tab is available when the trigger type is TABLE or VIEW.

### Referencing Old

Correlation names of the old nested table.

### Referencing New

Correlation names of the new nested table.

### Referencing Parent

Correlation names of the parent table.

### Follows

To indicate that the trigger should fire after the specified triggers.

Use the **Schema** dropdown list to select the schema name and **Trigger** dropdown list to select the trigger.

To add triggers, just simply press plus button. To remove triggers, select a trigger and press the cross button.


**Note:** Support from Oracle 11g or later.

## Editing Oracle Trigger Definition

The **Definition** tab allows you to edit valid SQL or procedure statements in the trigger definition inside *BEGIN* and *END*.



## Oracle Types

Type is an user-defined datatype that model the structure and behavior of the data in an application. An object type consists of two parts: a specification and a body. The type body always depends on its type specification. A collection type is a named varying array (varray) or a nested table type.

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



### Create Object Type (Object Type Specification)

To create a new object type

- Select anywhere on the object pane.
- Click the  **New Object Type** from the object pane toolbar.  
or
- Right-click and select  **New Object Type** from the popup menu.
- Edit object type properties on the appropriate tabs of the Object Type Designer.



### Edit Object Type

To edit the existing object type (manage its definition etc)

- Select the object type for editing in the object pane.
- Right-click and select the  **Design Type** from the popup menu or simply double-click the object type.  
or
- Click the  **Design Type** from the object pane toolbar.
- Edit object type properties on the appropriate tabs of the Object Type Designer.


## Delete Object Type

To delete an object type

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


## Create Type Body (Object Type Body)

To create a new type body

- Select the object type for modifying in the object pane.
- Right-click and select  **New Type Body** from the popup menu.
- Edit type body properties on the appropriate tabs of the Type Body Designer.


## Edit Type Body

To edit the existing type body (manage its definition etc)

- Select the object type for editing in the object pane.
- Right-click and select the  **Design Type Body** from the popup menu.
- Edit type body properties on the appropriate tabs of the Type Body Designer.



## Delete Type Body

To delete a type body




- Select the object type for deleting in the object pane.
- Right-click and select the  **Delete Type Body** from the popup menu.
- Confirm deleting in the dialog window.

## Create Collection Type

To create a new collection type



- Select anywhere on the object pane.
- Click the  **New Collection Type** from the object pane toolbar.  
or
- Right-click and select  **New Collection Type** from the popup menu.
- Edit collection type properties on the appropriate tabs of the Collection Type Designer.

To create a new collection type with modification as one of the existing collection type

- Select the collection for modifying in the object pane.
- Right-click and select the  **Design Type** from the popup menu or simply double-click the collection type.  
or
- Click the  **Design Type** from the object pane toolbar.
- Modify collection type properties on the appropriate tabs of the Collection Type Designer.
- Click  **Save As**.



## Edit Collection Type

To edit the existing collection type(manage its general etc)

- Select the collection type for editing in the object pane.
- Right-click and select the  **Design Type** from the popup menu or simply double-click the collection type.  
or
- Click the  **Design Type** from the object pane toolbar.
- Edit collection type properties on the appropriate tabs of the Collection Type Designer.

## Delete Collection Type

To delete a collection type

- Select the collection type for deleting in the object pane.
- Right-click and select the  **Delete Type** from the popup menu.  
or
- Click the  **Delete Type** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Maintain Type

To maintain a type

- Select the type for maintaining in the object pane.
- Right-click and select the **Maintain** from the popup menu.
  - Compile
  - Compile for Debug

## Achieve Type Information


To achieve a type information

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

## Oracle Object Type Designer

**Object Type Designer** is the basic Navicat tool for working with object types. It allows you to create new object type and edit the existing object type definition.

- [Editing Object Type Definition](#)
- Object Type SQL Preview







After saving the object type, you can edit the Object Type Body. Just click  **Design Object Type Body** to open the Type Body Designer.

## Editing Oracle Object Type Definition

Edit the object type definition under the **Definition** tab.

The **Code Outline** window displays information about the object type including declaration, etc. To show the **Code Outline** window, simply choose View -> **Code Outline**.

**Note:** Available only in Full Version.

	Refresh the code outline.
	Show the detail view of the code outline.
	Turn mouse over highlight on or off.
	Expand the selected item.
	Collapse the selected item.
	Toggle sorting by position.

The SQL statements for creating object types are CREATE TYPE. In practice, it is best to use a CREATE OR REPLACE statement. The general form of these statements follows.

```
CREATE OR REPLACE
TYPE /*TYPE NAME*/ AS OBJECT (
    /* TODO enter type specification (methods, attributes etc) here */
)
```

**Hint:** To customize the view of the editor and find out more features for sql editing, see Editor View and More Features.

## Oracle Type Body Designer

**Type Body Designer** is the basic Navicat tool for working with object type bodies. It allows you to create new type body and edit the existing type body definition.

- [Editing Type Body Definition](#)
- Type Body SQL Preview







To edit the Object Type Specification, click  **Design Object Type Specification** to open the Object Type Designer.

## Editing Oracle Type Body Definition

Edit the type body definition under the **Definition** tab.

The **Code Outline** window displays information about the type body including declaration, etc. To show the **Code Outline** window, simply choose View -> **Code Outline**.

**Note:** Available only in Full Version.

	Refresh the code outline.
	Show the detail view of the code outline.
	Turn mouse over highlight on or off.
	Expand the selected item.
	Collapse the selected item.
	Toggle sorting by position.

The SQL statements for creating type bodies are CREATE TYPE BODY. In practice, it is best to use a CREATE OR REPLACE statement. The general form of these statements follows.

```
CREATE OR REPLACE
TYPE BODY /*TYPE NAME*/ AS
....
END;
```

**Hint:** To customize the view of the editor and find out more features for sql editing, see Editor View and More Features.

## Oracle Collection Type Designer

**Collection Type Designer** is the basic Navicat tool for working with collection types. It allows you to create new collection type and edit the existing collection type definition.

- [Editing Collection Type General](#)
- Collection Type SQL Preview

## **Editing Oracle Collection Type General**

### **Nested table**

Create a nested table type.

### **VArray**

Create a varray type.

### **Array Size**

Determine the array size of the varray type.

### **Data Type**


Select the Oracle Database built-in datatype or user-defined type of the attribute.

### **Data Type Parameter**

Determine the corresponding data type parameters.



## Oracle XML Schemas

XML Schema is a schema definition language written in XML. It can be used to describe the structure and various other semantics of conforming instance documents.

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

### Create XML Schema

To create a new XML Schema

- Select anywhere on the object pane.
- Click the  **New XML Schema** from the object pane toolbar.  
or
- Right-click and select  **New XML Schema** from the popup menu.
- Edit XML Schema properties on the appropriate tabs of the XML Schema Designer.



### Maintain XML Schema

To maintain an XML Schema

- Select the XML Schema for maintaining in the object pane.
- Right-click and select the **Maintain** from the popup menu.
  - Compile
  - Purge

### Delete XML Schema

To delete an XML Schema

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

## Achieve XML Schema Information

To achieve an XML Schema information

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

## Oracle XML Schema Designer

**XML Schema Designer** is the basic Navicat tool for working with XML Schemas. It allows you to create new XML Schema.

- [Editing XML Schema Doc](#)
- [Setting Advanced XML Schema Properties](#)
- XML Schema SQL Preview

## Editing Oracle XML Schema Doc

Enter a valid XML schema document under the **Schema Doc** tab. The general form follows.

```
<?xml version="1.0" encoding="UTF-8"?>  
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">  
</xsd:schema>
```

## Setting Advanced Oracle XML Schema Properties

### Local

Check this to register as local schemas.

### Force on schema registration

Check this to ignore errors generated during schema evolution.

## Generate

### Object Types

Check this to enable the schema compiler to generate object types.

### Java Beans

Check this to enable the schema compiler to generate Java beans.

### Default Tables

Check this to enable the schema compiler to generate default tables.

## Options

### REGISTER\_NODOCID

Check this to prevent the creation of this column if the user wishes to optimize on storage.

### REGISTER\_BINARYXML

Check this to register the schema for Binary XML.

### REGISTER\_NT\_AS\_IOT

Check this to store nested tables created during schema registration as index organized tables.

### REGISTER\_AUTO\_OOL

Check this to automatically move large types out of line.

## Enable Hierarchy

### **ENABLE\_HIERARCHY\_NONE**

Enable hierarchy will not be called on any tables created while registering that schema.

## **ENABLE\_HIERARCHY\_CONTENTS**


Enable hierarchy will be called for all tables created during schema registration with hierarchy\_type as DBMS\_XDBZ.ENABLE\_CONTENTS.

## **ENABLE\_HIERARCHY\_RESMETADATA**

Enable hierarchy will be called on all tables created during schema registration with hierarchy\_type as DBMS\_XDBZ.ENABLE\_RESMETADATA.



## Oracle Recycle Bin

Recycle bin is actually a data dictionary table containing information about dropped objects. Dropped tables and any associated objects such as indexes, constraints, nested tables, and the likes are not removed and still occupy space. They continue to count against user space quotas, until specifically purged from the recycle bin or the unlikely situation where they must be purged by the database because of tablespace space constraints.

Just simply click -> **Recycle Bin** to open an object pane for **Recycle Bin**. A right-click displays the popup menu or using the object pane toolbar, allowing you to flashback tables or purge the deleted objects.



### Restore tables

To restore table from recycle bin

- Choose a table in recycle bin.
- Click the  **Flashback Table** from the object pane toolbar.  
or
- Right-click and select  **Flashback Table** from the popup menu.

### Purge Objects

To remove an object in the recycle bin

- Select an object for purging in the object pane.
- Right-click and select the  **Purge Object** from the popup menu.  
or
- Click the  **Purge Object** from the object pane toolbar.
- Confirm deleting in the dialog window.

To remove all objects in the recycle bin

- Right-click and select the **Purge Recycle Bin** from the popup menu.
- Confirm deleting in the dialog window.

To remove all objects in the recycle bin of every user

- Log in a user has the **SYSDBA** privilege.
- Right-click and select the **Purge DBA Recycle Bin** from the popup menu.
- Confirm deleting in the dialog window.


## **Achieve Recycle Bin Object Information**

To achieve an object information in recycle bin

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



## Oracle Directories

A directory object specifies an alias for a directory on the server file system where external binary file LOBs (BFILEs) and external table data are located. All directories are created in a single namespace and are not owned by an individual schema.




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

### Create Directory

To create a new directory



- Select anywhere on the object pane.
- Click the  **New Directory** from the object pane toolbar.  
or
- Right-click and select  **New Directory** from the popup menu.
- Edit directory properties on the appropriate tabs of the Directory Designer.

To create a new directory with modification as one of the existing directory

- Select the directory for modifying in the object pane.
- Right-click and select the  **Design Directory** from the popup menu or simply double-click the directory.  
or
- Click the  **Design Directory** from the object pane toolbar.
- Modify directory properties on the appropriate tabs of the Directory Designer.
- Click  **Save As**.



### Edit Directory

To edit the existing directory(manage its general etc)

- Select the directory for editing in the object pane.
- Right-click and select the  **Design Directory** from the popup menu or simply double-click the directory.  
or
- Click the  **Design Directory** from the object pane toolbar.
- Edit directory properties on the appropriate tabs of the Directory Designer.

## Delete Directory

To delete a directory

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

## Achieve Directory Information

To achieve a directory information

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

## Oracle Directory Designer

**Directory Designer** is the basic Navicat tool for working with directories. It allows you to create new directory and edit the existing directory definition.

- [Editing Directory General](#)
- Directory SQL Preview

## **Editing Oracle Directory General**


Edit the directory general properties under the **General** tab.

### **Directory Path**

Specify the full path name of the operating system directory of the server where the files are located. The path name is case sensitive



## Oracle Tablespaces

Tablespaces are the allocation of space in the database that can contain schema objects.




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

### Create Tablespace

To create a new tablespace



- Select anywhere on the object pane.
- Click the  **New Tablespace** from the object pane toolbar.  
or
- Right-click and select  **New Tablespace** from the popup menu.
- Edit tablespace properties on the appropriate tabs of the Tablespace Designer.

To create a new tablespace with modification as one of the existing tablespace

- Select the tablespace for modifying in the object pane.
- Right-click and select the  **Design Tablespace** from the popup menu or simply double-click the tablespace.  
or
- Click the  **Design Tablespace** from the object pane toolbar.
- Modify tablespace properties on the appropriate tabs of the Tablespace Designer.
- Click  **Save As**.

### Edit Tablespace

To edit the existing tablespace (manage its properties etc)

- Select the tablespace for editing in the object pane.
- Right-click and select the  **Design Tablespace** from the popup menu or simply double-click the tablespace.  
or
- Click the  **Design Tablespace** from the object pane toolbar.
- Edit tablespace properties on the appropriate tabs of the Tablespace Designer.

To change the name of the tablespace

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



## Maintain Tablespace

To maintain a tablespace

- Select the tablespace for maintaining in the object pane.
- Right-click and select the **Maintain** from the popup menu.
  - [Read Only]
  - Read Write
  - Online
  - Offline
    - Normal
    - Temporary
    - Immediate
  - Coalesce
  - Shrink Space

## Delete Tablespace

To delete a tablespace

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

## Achieve Tablespace Information

To achieve a tablespace information

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

## Oracle Tablespace Designer

**Tablespace Designer** is the basic Navicat tool for working with tablespaces. It allows you to create new tablespace and edit the existing tablespace properties.

- [Editing Tablespace General](#)
- [Editing Tablespace Storage](#)
- [Setting Advanced Tablespace Properties](#)
- Tablespace SQL Preview

## Editing Oracle Tablespace General

### Tablespace Type

#### **PERMANENT**

A permanent tablespace contains persistent schema objects. Objects in permanent tablespaces are stored in datafiles.

#### **TEMPORARY**

A temporary tablespace contains schema objects only for the duration of a session. Objects in temporary tablespaces are stored in tempfiles.

#### **UNDO**

An undo tablespace is a type of permanent tablespace used by Oracle Database to manage undo data if you are running your database in automatic undo management mode.

### Data File

Use the **Name** and **Size** edit box to set the datafile / tempfile name and size.

The **Unit** drop-down list defines the unit of the size of the datafile / tempfile. Specify the maximum disk space allowed for automatic extension of the datafile. Use the dropdown list K, M, G, T, P or E to specify the size in kilobytes, megabytes, gigabytes, terabytes, petabytes, or exabytes. If you do not specify any of the abbreviations, then the size is interpreted as bytes.

#### **Reuse**

To allow Oracle to reuse an existing file.

### Path

Specify the path of the datafile / tempfile.

### Auto Extend

To **ON** (enable) or **OFF** (disable) the automatic extension of a new or existing datafile or tempfile.

## **Next Size**

Specify the size in bytes of the next increment of disk space to be allocated automatically when more extents are required. The default is the size of one data block. Use the dropdown list K, M, G, T, P or E to specify the size in kilobytes, megabytes, gigabytes, terabytes, petabytes, or exabytes. If you do not specify any of the abbreviations, then the size is interpreted as bytes.

## **Unlimited Max**

Unlimited disk space that Oracle can allocate to the datafile or tempfile.

## **Max Size**

Specify the maximum disk space allowed for automatic extension of the datafile. Use the dropdown list K, M, G, T, P or E to specify the size in kilobytes, megabytes, gigabytes, terabytes, petabytes, or exabytes. If you do not specify any of the abbreviations, then the size is interpreted as bytes.

## Editing Oracle Tablespace Storage

### File Type

#### **BIGFILE**

A bigfile tablespace contains only one datafile or tempfile, which can contain up to approximately 4 billion ( $2^{32}$ ) blocks. The maximum size of the single datafile or tempfile is 128 terabytes (TB) for a tablespace with 32K blocks and 32TB for a tablespace with 8K blocks.

#### **SMALLFILE**

A smallfile tablespace is a traditional Oracle tablespace, which can contain 1022 datafiles or tempfiles, each of which can contain up to approximately 4 million ( $2^{22}$ ) blocks.

### Min Extent Size

The minimum size of an extent in the tablespace. Use the dropdown list K, M, G, T, P or E to specify the size in kilobytes, megabytes, gigabytes, terabytes, petabytes, or exabytes. If you do not specify any of the abbreviations, then the size is interpreted as bytes.

### Block Size

The block size for the tablespace.

### [Default Storage Options](#)

Set the default storage characteristics for objects created in the tablespace.

### Table Compression

Use the dropdown list to select the type of compressing data segments to reduce disk use.

### Manual Segment Management

To manage the free space of segments in the tablespace using free lists.

### Extent Management

To specify how the extents of the tablespace will be managed.

#### **Extent Management**

##### **DICTIONARY**

Extent management by the data dictionary.

## **LOCAL**

Extent management by the bitmaps.

### **Local Extent**

#### **AUTOALLOCATE**

The tablespace is system managed.

#### **UNIFORM**

The tablespace is managed with uniform extents of size.

### **Uniform Size**

The size of uniform extent. The default size is 1 megabyte. Use the dropdown list K, M, G, T, P or E to specify the size in kilobytes, megabytes, gigabytes, terabytes, petabytes, or exabytes. If you do not specify any of the abbreviations, then the size is interpreted as bytes.

## Setting Advanced Oracle Tablespace Properties

### Logging

#### **LOGGING**

Log all objects within the tablespace in the redo log file.

#### **NOLOGGING**

No operations are logged.

#### **Force Logging**

Oracle Database will log all changes to all objects in the tablespace except changes to temporary segments, overriding any NOLOGGING setting for individual objects.

#### **Offline**

The tablespace is unavailable immediately (offline) after creation.

#### **Retention Guarantee**

Oracle Database should preserve unexpired undo data in all undo segments of tablespace even if doing so forces the failure of ongoing operations that need undo space in those segments.

### Tablespace Group

To determine whether tablespace is a member of a tablespace group.

### Flashback

#### **ON**

Oracle Database will save Flashback log data for this tablespace and the tablespace can participate in a FLASHBACK DATABASE operation.

#### **OFF**

Oracle Database will not save any Flashback log data for this tablespace.

### Encryption

#### **Use Encryption**

Enable the encryption properties of the tablespace.

#### **Algorithm**

To select the encryption algorithm.

## Oracle Public Database Links

Public database Link is a database link created by a *DBA* on a local database that is accessible to all users on that database.

See [Database Link](#) for details.

## Oracle Public Synonyms

Public synonym is a synonym owned by the special user group named *PUBLIC* and every user in a database can access it.

See [Synonyms](#) for details.