

Table of Contents

SQL SERVER DATABASE OBJECT MANAGEMENT	4
SQL SERVER SCHEMAS	5
<i>SQL Server Schema Designer</i>	7
Editing SQL Server Schema General	8
<i>SQL Server Tables</i>	9
SQL Server Table Designer	14
SQL Server Table Fields	15
Setting SQL Server Table Field Properties	17
Setting Other SQL Server Table Field Properties	19
SQL Server Table Indexes	21
Setting SQL Server Table Index Properties	22
SQL Server Table Foreign Keys	24
Setting SQL Server Table Foreign Key Properties	25
SQL Server Table Uniques	27
Setting SQL Server Table Unique Properties	28
SQL Server Table Checks	29
Setting SQL Server Table Check Properties	30
SQL Server Table Triggers	31
Setting SQL Server Table Trigger Properties	32
SQL Server Table Storage	34
SQL Server Table Options	36
<i>SQL Server Views</i>	37
SQL Server View Designer	41
Working with SQL Server View Builder (Available only in Full Version)	42
Editing SQL Server View SQL Definition	43
Setting Advanced SQL Server View Properties	44
SQL Server View Preview	45
SQL Server View Explain	46
SQL Server View Viewer	48
<i>SQL Server Functions/Procedures</i>	50
Create SQL Server Functions/Procedures	54
SQL Server Function/Procedure Designer	55
Editing SQL Server Function/Procedure Definition	56
Viewing SQL Server Function/Procedure Result	57
<i>SQL Server Indexes</i>	58
SQL Server Index Designer	61
Editing SQL Server Nonclustered Index Properties	62

Editing SQL Server Nonclustered Index General	63
Editing SQL Server Nonclustered Index Filter	64
Editing Advanced SQL Server Nonclustered Index Properties	65
Editing SQL Server Nonclustered Index Storage	67
Editing SQL Server Clustered Index Properties	69
Editing SQL Server Clustered Index General	70
Editing Advanced SQL Server Clustered Index Properties	71
Editing SQL Server Clustered Index Storage	73
Editing SQL Server XML Index Properties	75
Editing SQL Server XML Index General	76
Editing Advanced XML Index Properties	77
Editing SQL Server Spatial Index Properties	78
Editing SQL Server Spatial Index General	79
Editing Advanced SQL Server Spatial Index Properties	81
<i>SQL Server Synonyms</i>	82
SQL Server Synonym Designer	84
Editing SQL Server Synonym General	85
<i>SQL Server Triggers</i>	86
SQL Server Trigger Designer	88
Editing SQL Server Trigger General	89
Setting Advanced SQL Server Trigger Properties	90
Editing SQL Server Trigger Definition	91
SQL SERVER LINKED SERVERS	92
<i>SQL Server Linked Server Designer</i>	94
Editing SQL Server Linked Server General	95
Editing SQL Server Linked Server Security	96
Setting Advanced SQL Server Linked Server Properties	97
SQL SERVER SERVER TRIGGERS	99
<i>SQL Server Server Trigger Designer</i>	101
Editing SQL Server Server Trigger General	102
Setting Advanced SQL Server Server Trigger Properties	103
Editing SQL Server Server Trigger Definition	104
SQL SERVER ASSEMBLIES	105
<i>SQL Server Assembly Designer</i>	107
Editing SQL Server Assembly General	108
SQL SERVER DATABASE TRIGGERS	109
<i>SQL Server Database Trigger Designer</i>	111
Editing SQL Server Database Trigger General	112

Setting Advanced SQL Server Database Trigger Properties	113
Editing SQL Server Database Trigger Definition	114
SQL SERVER PARTITION FUNCTIONS	115
<i>SQL Server Partition Function Designer</i>	117
Editing SQL Server Partition Function General	118
SQL SERVER PARTITION SCHEMES	119
<i>SQL Server Partition Scheme Designer</i>	121
Editing SQL Server Partition Scheme General	122

SQL Server Database Object Management

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

- [Schemas](#)
- [Tables](#)
- [Views](#)
- [Functions/Procedures](#)
- [Indexes](#)
- [Synonyms](#)
- [Triggers](#)
- [Linked Servers](#)
- [Server Triggers](#)
- [Assemblies](#)
- [Database Triggers](#)
- [Partition Functions](#)
- [Partition Schemes](#)

SQL Server Schemas

A schema contains named objects (tables, views, functions, etc) whose names may duplicate those of other objects existing in other schemas.

The schema name must be distinct from any existing schema name in the current database.

Create Schema

Note: Support from SQL Server 2005 or later and SQL Azure.

To create a new schema

- Control-click the database in the navigation pane and choose **New Schema....**
- Edit schema properties on the appropriate tabs of the Schema Designer.

Edit Schema

Note: Support from SQL Server 2005 or later and SQL Azure.

To edit the existing schema (manage its general etc)

- Control-click the schema in the navigation pane and choose **Schema Properties....**
- Edit schema properties on the appropriate tabs of the Schema Designer.

Open Schema

To open a schema which shows in the navigation pane

- Double-click the schema to open in the navigation pane.
or
- Control-click the schema and choose **Open Schema.**

Close Schema

To close a schema

- Control-click the schema in the navigation pane and choose **Close Schema.**

Delete Schema

Note: Support from SQL Server 2005 or later and SQL Azure.

To delete a schema

- Control-click the schema in the navigation pane and choose **Delete Schema...**
- Confirm removing in the dialog window.

SQL Server Schema Designer

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

Note: Support from SQL Server 2005 or later and SQL Azure.

- [Editing Schema General](#)
- Editing Schema Comment (SQL Azure does not support)
- Schema SQL Preview

Editing SQL Server Schema General

Schema Name


The name of a schema which is identified within the database.

Owner

The name of the database-level principal that will own the schema. This principal may own other schemas, and may not use the current schema as its default schema.


SQL Server Tables

Tables are database objects that contain all the data in a database. A table definition is a collection of columns. In tables, data is organized in a row-and-column format similar to a spreadsheet. Each row represents a unique record, and each column represents a field within the record.

Just simply click  to open an object pane for **Table**. By using the object pane toolbar, you can create new, edit, open and delete the selected table.

Create Table

To create a new table

- Select anywhere on the object pane.
- Click the  from the object pane toolbar.
or
- Control-click and select **New Table...** to choose the popup menu.
- Edit table properties and fields on the appropriate tabs of the Table Designer.

Hint: To create new table you can also control-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 database {same connection}

- Select the table(s) for copying in the navigation pane/object pane.
- Control-click and select the **Duplicate Table -> Structure + Data/Structure Only** 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 database {same connection}



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

Apply to: different database {same connection}

different database {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 only)
 - Copy here (Structure and Data)
 - 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.
- Control-click and select the **Design Table...** from the popup menu.
or
- Click the  from the object pane toolbar.
- Modify table properties and fields on the appropriate tabs of the Table Designer.
- Click  **Save As.**

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.
- Control-click and select the **Design Table...** from the popup menu.
or
- Click the  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.
- Control-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.
- Control-click and select the **Open Table** from the popup menu or simply double-click the table.
or
- Click the **Open** 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.
- Control-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.
- Control-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.
- Control-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.
- Control-click and select the **Delete Table...** from the popup menu.
or
- Click the  from the object pane toolbar.
- Confirm deleting in the dialog window.


Table Shortcut

To create a table shortcut

- Select the table in the navigation pane/object pane.
- Drag the table out as a shortcut.

Achieve Table Information

To achieve a table information

- Select the table in the navigation pane/object pane.
- Choose View -> Object Information in the main menu.
or
- Click the  from the object pane toolbar.

SQL Server Table Designer

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.


- [Managing Table Fields](#)
- [Managing Table Indexes](#)
- [Managing Table Foreign Keys](#)
- [Managing Table Uniques](#)
- [Managing Table Checks](#)
- [Managing Table Triggers](#)
- [Managing Table Storage](#)
- [Managing Table Options](#)
- Managing Table Comment (SQL Azure does not support)
- Table SQL Preview

SQL Server Table Fields

Table fields are managed on the **Fields** tab of the Table Designer. Just simply click a field for editing. By using the field toolbar, you can 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.
- Click the  **Add Field** from the toolbar.
- Edit field properties.

Insert Field

To insert a field above an existing field when create a new table

- Create a new table.
- Open the **Fields** tab.
- Select field.
- Click the  **Insert Field** from the toolbar.
- Define field properties in the empty row.

Note: Hidden when edit existing tables.

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.


To change the order of the table fields when create a new table

- Create a new table.
- Open the **Fields** tab.
- Select the field to move and click the  **Move Up**/  **Move Down** from the toolbar.


Note: Hidden when edit existing tables.

Delete Field

To delete the table field

- Open the table in the Table Designer.
- Open the **Fields** tab.
- Click the  **Delete Field** from the toolbar.
- Confirm deleting in the dialog window.

Setting SQL Server Table Field Properties

Name	Type	Size	Scale	Allow Null	Key
EmployeeID	int	0	0	<input type="checkbox"/>	
NationalIDNumber	nvarchar	15	0	<input type="checkbox"/>	
ContactID	int	0	0	<input type="checkbox"/>	
LoginID	nvarchar	256	0	<input type="checkbox"/>	
ManagerID	int	0	0	<input checked="" type="checkbox"/>	

Name

The Name is a descriptive identifier for a field that can be up to 128 characters. 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 SQL Server 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. See [SQL Server Data Type](#) and [SQL Azure Support Data Type](#) for details.

Size

Define the size of the field.

Scale

Define the number of digits after the decimal point (the scale) for Floating Point data type.

Note: Be careful when shortening the field length as losing data might be caused.

Allow Null

Allow the NULL values for the field.

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.

Setting Other SQL Server Table Field Properties

For **bigint**, **decimal**, **int**, **numeric**, **smallint**, **tinyint** data types:

Identity

Indicate that the new column is an identity column.

For **uniqueidentifier** data types:

Row GUID Column

Indicate that the new column is a row GUID column. Only one uniqueidentifier column per table can be designated as the ROWGUIDCOL column.

Note: SQL Azure does not support.

For **char**, **nchar**, **ntext**, **nvarchar**, **text**, **varchar** data types:

Collate

Specify the collation for the column.

For **xml** data types:

Column Set

Combine all of the sparse columns of a table into a structured output.

Note: Support from SQL Server 2008 or later.

For **varbinary(max)** data types:

Filestream

Specify FILESTREAM storage for the varbinary(max) BLOB data.

Note: Support from SQL Server 2008 or later.

For **(USER-DEFINED)** data types:

Schema

Set the schema of the user defined type.

User-defined Type

Set the user defined type.

For **(COMPUTED)** data types:

Expression

Set an expression that defines the value of a computed column.

Persisted

Specify that the SQL Server Database Engine will physically store the computed values in the table, and update the values when any other columns on which the computed column depends are updated.

For most data types:

Default Value

Set the default value for the field.

Sparse

Indicate that the column is a sparse column.

Note: Support from SQL Server 2008 or later.

For all data types:

Comment

Set any optional text describing the current field.

Note: SQL Azure does not support.


SQL Server Table Indexes

Indexes are optional structures associated with tables. You can create indexes on one or more columns of a table to speed SQL statement execution on that table.

Table indexes are managed on the **Indexes** tab of the Table Designer. Just simply click on an index field for editing. By using the index toolbar, you can 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.
- 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 on the index to edit.

Delete Index


To delete a table index

- Open the table in the Table Designer.
- Open the **Indexes** tab.
- Click the  **Delete Index** from the toolbar.
- Confirm deleting in the dialog window.

Setting SQL Server Table Index Properties

Name	Fields	Index Type	Unique
AK_Employee_LoginID	[LoginID] ASC	Nonclu... ▾	<input checked="" type="checkbox"/>
AK_Employee_Nationall...	[NationalIDNumber] ASC	Nonclu... ▾	<input checked="" type="checkbox"/>
AK_Employee_rowguid	[rowguid] ASC	Nonclu... ▾	<input checked="" type="checkbox"/>

Use the **Name** edit box to set the index name. No schema name can be included here; the index is always created in the same schema as its parent table.

To include field(s) in the index, just simply double-click the **Fields** field or click  to open the editor for editing. The **Order** dropdown list is used to set sort direction.

Note: Some of data types do not allow indexing. For example: text



The **Index Type** dropdown list defines the type of the table index.

Clustered

Create an index in which the logical order of the key values determines the physical order of the corresponding rows in a table.

Nonclustered

Create an index that specifies the logical ordering of a table. With a nonclustered index, the physical order of the data rows is independent of their indexed order.

Spatial

Create a spatial index on a specified table and column. An index can be created before there is data in the table.

Note: Support from SQL Server 2008 or later and SQL Azure.

XML

Create an XML index on a specified table. An index can be created before there is data in the table.

Note: Support from SQL Server 2005 or later.

Unique

Create a unique index on a table.

Comment

Specify the comment of the index.

Note: SQL Azure does not support.


SQL Server Table Foreign Keys

A foreign key is a field in a relational table that matches the primary key column of another table. The foreign key can be used to cross-reference tables.

Foreign Keys are managed on the **Foreign Keys** tab of the Table Designer. Just simply click on a foreign key field for editing. By using the foreign key toolbar, you can 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.
- 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 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.
- Click the  **Delete Foreign Key** from the toolbar.
- Confirm deleting in the dialog window.

Setting SQL Server Table Foreign Key Properties

Name	Fields	Reference Schema	Reference Table	Reference Fields	On Delete	On Update	Not for Replication	Enable
FK_Employee_E...	ManagerID 	HumanResour 	Employee 	EmployeeID 	NO A... 	NO A... 	<input type="checkbox"/>	<input checked="" type="checkbox"/>
FK_Employee_C...	ContactID 	Person 	Contact 	ContactID 	NO A... 	NO A... 	<input type="checkbox"/>	<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** and **Reference Table** dropdown lists to select a foreign schema and table respectively.

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

The **On Delete** and **On Update** dropdown list define the type of the actions to be taken.

No Action

The Database Engine raises an error and the delete or update action on the row in the parent table is rolled back.

Cascade

Corresponding rows are deleted from or updated in the referencing table if that row is deleted from or updated in the parent table.

Set Null

All the values that make up the foreign key are set to NULL when the corresponding row in the parent table is deleted or updated.

Set Default

All the values that make up the foreign key are set to their default values when the corresponding row in the parent table is deleted or updated.

Enable

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

Not for Replication

The constraint is not enforced when replication agents perform insert, update, or delete operations.

Note: SQL Azure does not support.

Comment

Specify the comment of the foreign key.

Note: SQL Azure does not support.


SQL Server 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 on an unique field for editing. By using the unique toolbar, you can 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.
- 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.
- Click the  **Delete Unique** from the toolbar.
- Confirm deleting in the dialog window.

Setting SQL Server Table Unique Properties

Name	Fields	Unique Type
AK_Product_Name	Name ASC	Nonclustered

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

Fields

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

Unique Type

Indicate that a clustered or nonclustered index is created for the unique constraint.

Comment

Specify the comment of the unique.

Note: SQL Azure does not support.


SQL Server Table Checks

A check is a constraint that enforces domain integrity by limiting the possible values that can be entered into a column or columns.

Checks are managed on the **Checks** tab of the Table Designer. Just simply click on a check field for editing. By using the check toolbar, you can 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.
- 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.
- Click the  **Delete Check** from the toolbar.
- Confirm deleting in the dialog window.

Setting SQL Server Table Check Properties

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

Expression

Set the logical expression that returns TRUE or FALSE, e.g. "field_name1 > 0 AND field_name2 > field_name1" in the **Expression** edit box.

Not for Replication

The constraint is not enforced when replication agents perform insert, update, or delete operations.

Note: SQL Azure does not support.

Enable

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

Comment

Specify the comment of the check.

Note: SQL Azure does not support.


SQL Server Table Triggers

A trigger is a special kind of stored procedure that automatically executes when an event occurs in the database server.

Triggers are managed on the **Triggers** tab of the Table Designer. Just simply click a trigger field for editing. By using the trigger toolbar, you can 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.
- 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.
- Click the  **Delete Trigger** from the toolbar.
- Confirm deleting in the dialog window.

Setting SQL Server Table Trigger Properties

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

Use the **Fires** dropdown list to define the trigger action time.

AFTER

Specify that the DML trigger is fired only when all operations specified in the triggering SQL statement have executed successfully.

INSTEAD OF

Specify that the DML trigger is executed instead of the triggering SQL statement, therefore, overriding the actions of the triggering statements.

Delete

The trigger is activated whenever a row is deleted from the table.

Insert

The trigger is activated whenever a new row is inserted into the table.

Update

The trigger is activated whenever a row is modified.

The **Body** tab defines the statement to execute when the trigger activates. To include your statement, just simply click to write. If you want to execute multiple statements, use the **BEGIN ... END** compound statement construct.

Advanced

Execute as

Specify the security context under which the trigger is executed.

Note: Support from SQL Server 2005 or later and SQL Azure.

User

Choose a user that the trigger executes in.

Note: Support from SQL Server 2005 or later and SQL Azure.

Definition Type

Choose the type of definition.

Note: Support from SQL Server 2005 or later.

Options

Encrypted

Obfuscate the text of the CREATE TRIGGER statement.

Note: Support from SQL Server 2005 or later.

Not for replication

Indicate that the trigger should not be executed when a replication agent modifies the table that is involved in the trigger.

Note: SQL Azure does not support.

Append

Specify that an additional trigger of an existing type should be added.

Note: SQL Azure does not support.

Comment

Specify the comment of the trigger.

Note: SQL Azure does not support.

SQL Server Table Storage

SQL Azure does not support this tab.

Store on Filegroup

Filegroup

Choose a filegroup that storing the table.

Text/Image Filegroup

Choose a filegroup for storing text, ntext, image, xml, varchar(max), nvarchar(max), varbinary(max), and CLR user-defined type columns.

Filestream Filegroup

Choose a filegroup for FILESTREAM data.

Note: Support from SQL Server 2008 or later.

Store on Partition Scheme

Note: Support from SQL Server 2005 or later.

Partition Scheme

Choose a partition scheme that storing the table.

Partition Column

Choose a partition column name.

Filestream Partition Scheme

Choose a partition scheme for FILESTREAM data.

Note: Support from SQL Server 2008 or later.

Data Compression

Note: Support from SQL Server 2008 or later.

Partition Number

The partition which the DATA_COMPRESSION setting applies.

Type

NONE

Table or specified partitions are not compressed.

ROW

Table or specified partitions are compressed by using row compression.

PAGE

Table or specified partitions are compressed by using page compression.

SQL Server Table Options

Identity

Seed

The value used for the very first row loaded into the table.

Increment

The incremental value added to the identity value of the previous row loaded.

Current

Set the current identity value.

Not for Replication

Values are not incremented in identity columns when replication agents perform inserts.

Note: SQL Azure does not support.

Change Tracking

Change Tracking

Specify change tracking is enabled for the table.

Note: Support from SQL Server 2008 or later.

Track Columns Updated

Specify the Database Engine tracks which change tracked columns were updated.

Note: Support from SQL Server 2008 or later.


Lock Escalation

Specify the allowed methods of lock escalation for a table.

Note: Support from SQL Server 2008 or later.


SQL Server Views

A view can be thought of as either a virtual table or a stored query. Unless a view is indexed, its data is not stored in the database as a distinct object. What is stored in the database is a SELECT statement. The result set of the SELECT statement forms the virtual table returned by the view. A user can use this virtual table by referencing the view name in Transact-SQL statements the same way a table is referenced.

Just simply click  to open an object pane for **View**. By using the object pane toolbar, you can create new, edit, open and delete the selected view.

Create View

To create a new view

- Select anywhere on the object pane.
- Click the  from the object pane toolbar.
or
- Control-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 control-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 popup menu)

Apply to: current database {same connection}

- Select the view(s) for copying in the navigation pane/object pane.
- Control-click and select the **Duplicate View** from the popup menu.
- The newly created view(s) will be named as "viewname_**copy**".

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

Apply to: current database {same connection}



- Select the view(s) for copying in the navigation pane/object pane.
- Click and drag the chosen view(s) to the target location.
- Select one of the following options:
 - Copy
 - Cancel
- The newly created view(s) will be named as "viewname_**copy**".

Apply to: different database {same connection}



different database {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
 - 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.
- Control-click and select the **Design View...** from the popup menu.
or
- Click the  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  from the object pane toolbar.
or
- Control-click and select **New View...** from the popup menu.
- Click  **Load.**

Edit View

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

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


Delete View

To delete a view

- Select the view for deleting in the navigation pane/object pane.
- Control-click and select the **Delete View...** from the popup menu.
or
- Click the  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.
- Choose View -> Object Information in the main menu.
or
- Click the  from the object pane toolbar.

SQL Server 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 (SQL Azure does not support)
- View SQL Preview
- [View Preview](#)
- [View Explain](#)

Working with SQL Server 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 SQL Server View SQL Definition

View Editor allows you to edit the view definition as SQL statement (SELECT statement it implements).

Example:

```
SELECT
    report_sample.clients.RecordID
FROM
    report_sample.clients
```

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

Setting Advanced SQL Server View Properties

Encryption

Encrypt the entries in sys.syscomments that contain the text of the CREATE VIEW statement.

Schema binding

Bind the view to the schema of the underlying table or tables.

View metadata

Specify that the instance of SQL Server will return to the DB-Library, ODBC, and OLE DB APIs the metadata information about the view, instead of the base table or tables, when browse-mode metadata is being requested for a query that references the view.

Check option


Force all data modification statements executed against the view to follow the criteria set within select_statement.

SQL Server View Preview

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

The **Preview** tab displays the data of the view as a grid.

SQL Server View Explain

To return information about how the SQL statement would have operated, press  **Preview** and choose **Explain** or **Preview and Explain**. If the statement is correct, the **Explain** tab opens.

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

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

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

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

SQL Server 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 Preferences.

- **Rollback**

Undo work done in the current transaction.

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

- **Export Data**

Export data to TXT, CSV, DBF, XML and MS Excel.

- **Filter Data**

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

- **Edit TEXT/BLOB**

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

Commit Rollback Grid View Form View Image Text Hex Filter Wizard Export Wizard Sort Ascending

EmployeeID	Title	FirstName	MiddleName	LastName	Suffix
202	[Null]	Elizabeth	I	Keyser	[Null]
203	[Null]	Terry	J	Eminhizer	[Null]
204	[Null]	John	N	Frum	[Null]
205	[Null]	Merav	A	Netz	[Null]
206	[Null]	Brian	P	LaMee	[Null]
207	[Null]	Kitti	H	Lertpiriyasawat	[Null]
208	[Null]	Jay	G	Adams	[Null]
209	[Null]	Jan	S	Miksovsky	[Null]
210	[Null]	Brenda	M	Diaz	[Null]
211	[Null]	Andrew	M	Cencini	[Null]
212	[Null]	Chris	K	Norred	[Null]
213	[Null]	Chris	O	Okelberry	[Null]
214	[Null]	Shelley	N	Dyck	[Null]
215	[Null]	Gabe	B	Mares	[Null]
216	[Null]	Mike	K	Seamans	[Null]
217	[Null]	Michael	[Null]	Raheem	[Null]
218	[Null]	Gary	E.	Altman	III
219	[Null]	Charles	B	Fitzgerald	[Null]
220	[Null]	Ebru	N	Ersan	[Null]
221	[Null]	Subroto	A	Valdez	[Null]

select * from [HumanResources].[vEmployeeDepartmentHistory] 1


296 records in page 1

SQL Server Functions/Procedures

A user-defined function, which is a Transact-SQL or common language runtime (CLR) routine that accepts parameters, performs an action, such as a complex calculation, and returns the result of that action as a value. The return value can either be a scalar (single) value or a table.


Stored procedures are similar to procedures in other programming languages in that they can:

- Accept input parameters and return multiple values in the form of output parameters to the calling procedure or batch.
- Contain programming statements that perform operations in the database, including calling other procedures.
- Return a status value to a calling procedure or batch to indicate success or failure (and the reason for failure).

Just simply click  to open an object pane for **Function**. By using the object pane toolbar, you can create new, edit and delete the selected function/procedure.

Create Function/Procedures

To create a new function/procedure

- Select anywhere on the object pane.
- Click the  from the object pane toolbar.
or
- Control-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 control-click the Function node of the navigation pane and select **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.
- Click and drag the chosen function/procedure(s) to the target location.
- Select one of the following options:
 - Copy
 - 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
 - Cancel

To create a new function/procedure with the same properties as one of the existing function/procedure has

- Control-click the function/procedure in the navigation pane or object pane and choose **Duplicate Function**.
- The newly created function/procedure will be named as "function/procedurename_copy".

Edit Function/Procedure

To edit the existing function/procedure


- Select the function/procedure for editing in the navigation pane/object pane.
- Control-click and select the **Design Function...** from the popup menu or simply double-click the function/procedure.
or
- Click the  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** from the object pane toolbar.
or
- Control-click and select **Execute Function...** from the popup menu.
- View the returned data on the Result tab.

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

- Create a function/procedure or open the existing function/procedure.
- Click  **Execute**.
- View the returned data on the Result tab.


Delete Function/Procedure

To delete a function/procedure

- Select the function/procedure for deleting in the navigation pane/object pane.
- Control-click and select the **Delete Function...** from the popup menu.
or
- Click the  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.
- Choose View -> Object Information in the main menu.
or
- Click the  from the object pane toolbar.

Create SQL Server Functions/Procedures

Click the **+** from the object pane toolbar. A Window will pop up and it allows you to create a procedure/function easily.

Type

Define whether it is a procedure/function.

Name

Specify the name for the procedure/function.

Parameters

Define the parameter(s) for the procedure/function. Set the parameter **Name**, **Schema**, **Type**, **Type**, **Default Value**, etc under corresponding columns.


SQL Server Function/Procedure Designer

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







- [Editing Function/Procedure Definition](#)
- Editing Function/Procedure Comment (SQL Azure does not support)
- Function/Procedure SQL Preview
- [Viewing Function/Procedure Result](#)

Editing SQL Server 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 click  button.

Note: Available only in Full Version.


	Refresh the code outline.
	Turn mouse over highlight on or off.
	Show the code outline in a detail way.
	Sort by type and name.
	Expand the selected item.
	Collapse the selected item.

Example:

```
CREATE PROCEDURE []
AS
NULL
;
```

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



Viewing SQL Server Function/Procedure Result

To run the function/procedure click  **Execute** on the toolbar. If the SQL statement is correct, the statement will be executed and, if the statement is supposed to return data, the **Result** tab opens with the 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.


SQL Server Indexes

An index in a database lets you quickly find specific information in a table or indexed view. An index contains keys built from one or more columns in the table, or view, and pointers that map to the storage location of the specified data. You can significantly improve the performance of database queries and applications by creating well-designed indexes to support your queries. Indexes can reduce the amount of data that must be read to return the query result set. Indexes can also enforce uniqueness on the rows in a table, ensuring the data integrity of the table data.



Just simply click  ->  **Index** to open an object pane for **Index**. A control-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  from the object pane toolbar.
or
- Control-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.
- Control-click and select the **Design Index...** from the popup menu or simply double-click the index.
or
- Click the  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.
- Control-click and select the **Design Index...** from the popup menu or simply double-click the index.
or
- Click the  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.
- Control-click and select the **Rename** from the popup menu.


Maintain Index

To maintain an index

- Select the index for maintaining in the object pane.
- Control-click and select the **Maintain** from the popup menu.
 - Rebuild Index
 - Disable Index


Delete Index

To delete an index

- Select the index for deleting in the object pane.
- Control-click and select the **Delete Index...** from the popup menu.
or
- Click the  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.
- Choose View -> Object Information in the main menu to view the Object Information.
or
- Click the  from the object pane toolbar.

SQL Server 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 Nonclustered Index Properties](#)
- [Editing Clustered Index Properties](#)
- [Editing XML Index Properties](#) (Support from SQL Server 2005 or later)
- [Editing Spatial Index Properties](#) (Support from SQL Server 2008 or later and SQL Azure)
- Editing Index Comment (SQL Azure does not support)
- Index SQL Preview

Editing SQL Server Nonclustered Index Properties

To create a nonclustered index:

- [Editing Nonclustered Index General](#)
- [Editing Nonclustered Index Filter](#) (Support from SQL Server 2008 or later and SQL Azure)
- [Editing Advanced Nonclustered Index Properties](#)
- [Editing Nonclustered Index Storage](#)

Editing SQL Server Nonclustered Index General

Type

Choose the index type: **Nonclustered**, Clustered, XML or Spatial

Unique

A unique index is one in which no two rows are permitted to have the same index key value.

Table/View

Choose to create a table index or a view index.

Table Name or View Name

Select a table or a view.

Fields

Select the column or columns on which the index is based and the sorting order.

Included Columns

Select the non-key columns to be added to the leaf level of the nonclustered index.

Note: Support from SQL Server 2005 or later and SQL Azure.

Editing SQL Server Nonclustered Index Filter

To create a filtered index, specify which rows to include in the index.

Note: Support from SQL Server 2008 or later and SQL Azure.

Example:

```
StartDate > '20040101' AND EndDate <= '20040630'
```

Editing Advanced SQL Server Nonclustered Index Properties

Ignore duplicate key values

A warning message will occur when duplicate key values are inserted into a unique index. Only the rows violating the uniqueness constraint will fail.

Recompute statistics

Enable automatic statistics updating.

Allow row locks

Row locks are allowed when accessing the index. The Database Engine determines when row locks are used.

Note: Support from SQL Server 2005 or later.

Allow page locks

Page locks are allowed when accessing the index. The Database Engine determines when page locks are used.

Note: Support from SQL Server 2005 or later.

Create/Rebuild Option

Fill Factor (%)

Specify a percentage that indicates how full the Database Engine should make the leaf level of each index page during index creation or rebuild. Fill Factor must be an integer value from 1 to 100.

Note: SQL Azure does not support.

Pad index

The percentage of free space that is specified by fillfactor is applied to the intermediate-level pages of the index.

Note: Support from SQL Server 2005 or later.

Sort in tempdb

Specify to store temporary sort results in tempdb.

Note: SQL Azure does not support.

Online

Long-term table locks are not held for the duration of the index operation.

Note: Support from SQL Server 2005 or later and SQL Azure.

Max. degree of parallelism

Override the max degree of parallelism configuration option for the duration of the index operation.

Note: Support from SQL Server 2005 or later.

Editing SQL Server Nonclustered Index Storage

SQL Azure does not support this tab.

Store on filegroup

Filegroup

Choose a filegroup.

Filestream Filegroup

Choose a filestream filegroup.

Note: Support from SQL Server 2008 or later.

Store on partition scheme

Note: Support from SQL Server 2005 or later.

Partition Scheme

Choose a partition scheme.

Partition Column

Choose a partition column name.

Filestream Partition Scheme

Choose a filestream partition scheme.

Note: Support from SQL Server 2008 or later.

Data Compression

Note: Support from SQL Server 2008 or later.

Partition Number

The partition which the DATA_COMPRESSION setting applies.

Type

NONE

Index or specified partitions are not compressed.

ROW

Index or specified partitions are compressed by using row compression.

PAGE

Index or specified partitions are compressed by using page compression.

Editing SQL Server Clustered Index Properties

To create a clustered index:

- [Editing Clustered Index General](#)
- [Editing Advanced Clustered Index Properties](#)
- [Editing Clustered Index Storage](#)

Editing SQL Server Clustered Index General

Type

Choose the index type: Nonclustered, **Clustered**, XML or Spatial

Unique

A unique index is one in which no two rows are permitted to have the same index key value.

Table/View

Choose to create a table index or a view index.

Table Name or View Name

Select a table or a view.

Fields

Select the column or columns on which the index is based and the sorting order.

Editing Advanced SQL Server Clustered Index Properties

Ignore duplicate key values

A warning message will occur when duplicate key values are inserted into a unique index. Only the rows violating the uniqueness constraint will fail.

Recompute statistics

Enable automatic statistics updating.

Allow row locks

Row locks are allowed when accessing the index. The Database Engine determines when row locks are used.

Note: Support from SQL Server 2005 or later.

Allow page locks

Page locks are allowed when accessing the index. The Database Engine determines when page locks are used.

Note: Support from SQL Server 2005 or later.

Create/Rebuild Option

Fill Factor (%)

Specify a percentage that indicates how full the Database Engine should make the leaf level of each index page during index creation or rebuild. Fill Factor must be an integer value from 1 to 100.

Note: SQL Azure does not support.

Pad index

The percentage of free space that is specified by fillfactor is applied to the intermediate-level pages of the index.

Note: Support from SQL Server 2005 or later.

Sort in tempdb

Specify to store temporary sort results in tempdb.

Note: SQL Azure does not support.

Online

Long-term table locks are not held for the duration of the index operation.

Note: Support from SQL Server 2005 or later and SQL Azure.

Max. degree of parallelism

Override the max degree of parallelism configuration option for the duration of the index operation.

Note: Support from SQL Server 2005 or later.

Editing SQL Server Clustered Index Storage

SQL Azure does not support this tab.

Store on filegroup

Filegroup

Choose a filegroup.

Filestream Filegroup

Choose a filestream filegroup.

Note: Support from SQL Server 2008 or later.

Store on partition scheme

Note: Support from SQL Server 2005 or later.

Partition Scheme

Choose a partition scheme.

Partition Column

Choose a partition column name.

Filestream Partition Scheme

Choose a filestream partition scheme.

Note: Support from SQL Server 2008 or later.

Data Compression

Note: Support from SQL Server 2008 or later.

Partition Number

The partition which the DATA_COMPRESSION setting applies.

Type

NONE

Index or specified partitions are not compressed.

ROW

Index or specified partitions are compressed by using row compression.

PAGE

Index or specified partitions are compressed by using page compression.

Editing SQL Server XML Index Properties

To create a XML index:

Note: Support from SQL Server 2005 or later.

- [Editing XML Index General](#)
- [Editing Advanced XML Index Properties](#)

Editing SQL Server XML Index General

Type

Choose the index type: Nonclustered, Clustered, **XML** or Spatial

Table/View

Must be Table.

Table Name

Select a table.

XML Column

Select the xml column on which the index is based.

XML Index Type

PRIMARY

A clustered index is created with the clustered key formed from the clustering key of the user table and an XML node identifier.

PATH secondary

Create a secondary XML index on columns built on path values and node values in the primary XML index. In the PATH secondary index, the path and node values are key columns that allow efficient seeks when searching for paths.

VALUE secondary

Create a secondary XML index on columns where key columns are (node value and path) of the primary XML index.

PROPERTY secondary

Create a secondary XML index on columns (PK, path and node value) of the primary XML index where PK is the primary key of the base table.

Primary XML index

Specify the primary XML index to use in creating a secondary XML index.

Editing Advanced XML Index Properties

Allow row locks

Row locks are allowed when accessing the index. The Database Engine determines when row locks are used.

Allow page locks

Page locks are allowed when accessing the index. The Database Engine determines when page locks are used.

Create/Rebuild Option

Fill Factor (%)

Specify a percentage that indicates how full the Database Engine should make the leaf level of each index page during index creation or rebuild. Fill Factor must be an integer value from 1 to 100.

Pad index

The percentage of free space that is specified by fillfactor is applied to the intermediate-level pages of the index.

Sort in tempdb

Specify to store temporary sort results in tempdb.

Max. degree of parallelism

Override the max degree of parallelism configuration option for the duration of the index operation.

Editing SQL Server Spatial Index Properties

To create a spatial index:

Note: Support from SQL Server 2008 or later and SQL Azure.

- [Editing Spatial Index General](#)
- [Editing Advanced Spatial Index Properties](#)

Editing SQL Server Spatial Index General

Type

Choose the index type: Nonclustered, Clustered, XML or **Spatial**

Table/View

Must be Table.

Table Name

Select a table.

Spatial Column

Select a spatial column which the index is based.

Tessellation Scheme

The tessellation scheme for the spatial index.

Bounding Box

Specify a numeric four-tuple that defines the four coordinates of the bounding box: the x-min and y-min coordinates of the lower, left corner, and the x-max and y-max coordinates of the upper right corner.

Min. Coordinates

Specify the x-coordinate (X) and y-coordinate (Y) of the lower-left corner of the bounding box.

Max. Coordinates

Specify the x-coordinate (X) and y-coordinate (Y) of the upper-right corner of the bounding box.

Grid Density

Define the density of the grid at each level of a tessellation scheme.

Level 1

Specify the first (top) level grid.

Level 2

Specify the second-level grid.

Level 3

Specify the third-level grid.

Level 4

Specify the fourth-level grid.

Cells per Object

Specify the number of tessellation cells per object that can be used for a single spatial object in the index by the tessellation process.

Editing Advanced SQL Server Spatial Index Properties

Recompute statistics

Enable automatic statistics updating.

Allow row locks

Row locks are allowed when accessing the index. The Database Engine determines when row locks are used.

Note: SQL Azure does not support.

Allow page locks

Page locks are allowed when accessing the index. The Database Engine determines when page locks are used.

Note: SQL Azure does not support.

Create / Rebuild Option

Note: SQL Azure does not support.

Fill Factor (%)

Specify a percentage that indicates how full the Database Engine should make the leaf level of each index page during index creation or rebuild. Fill Factor must be an integer value from 1 to 100.

Pad index

The percentage of free space that is specified by fillfactor is applied to the intermediate-level pages of the index.

Sort in tempdb



Specify to store temporary sort results in tempdb.

Max. degree of parallelism

Override the max degree of parallelism configuration option for the duration of the index operation.


SQL Server Synonyms

A synonym is an alternative name for a schema-scoped object. Client applications can use a single-part name to reference a base object by using a synonym instead of using a two-part, three-part, or four-part name to reference the base object.



Just simply click ->  **Synonym** to open an object pane for **Synonym**. A control-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  from the object pane toolbar.
or
- Control-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.
- Control-click and select the **Design Synonym...** from the popup menu or simply double-click the synonym.
or
- Click the  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.
- Control-click and select the **Design Synonym...** from the popup menu or simply double-click the synonym.
or
- Click the  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.
- Control-click and select the **Rename** from the popup menu.


Delete Synonym

To delete a synonym

- Select the synonym for deleting in the object pane.
- Control-click and select the **Delete Synonym...** from the popup menu.
or
- Click the  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.
- Choose View -> Object Information in the main menu to view the Object Information.
or
- Click the  from the object pane toolbar.

SQL Server 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](#)
- Editing Synonym Comment (SQL Azure does not support)
- Synonym SQL Preview

Editing SQL Server Synonym General

Server Name

The name of the server on which base object is located.

Database Name

The name of the database in which the base object is located.

Schema Name

The name of the schema of the base object.

Object Type



The object type.

Object Name

The name of the base object that the synonym references.


SQL Server Triggers

A trigger is a special kind of stored procedure that automatically executes when an event occurs in the database server.



Just simply click ->  **Trigger** to open an object pane for **Trigger**. A control-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  from the object pane toolbar.
or
- Control-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.
- Control-click and select the **Design Trigger...** from the popup menu or simply double-click the trigger.
or
- Click the  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, advanced, etc)

- Select the trigger for editing in the object pane.
- Control-click and select the **Design Trigger...** from the popup menu or simply double-click the trigger.
or
- Click the  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.
- Control-click and select the **Rename** from the popup menu.


Maintain Trigger

To maintain a trigger

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


Delete Trigger

To delete a trigger

- Select the trigger for deleting in the object pane.
- Control-click and select the **Delete Trigger...** from the popup menu.
or
- Click the  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.
- Choose View -> Object Information in the main menu to view the Object Information.
or
- Click the  from the object pane toolbar.

SQL Server 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](#)
- Editing Trigger Comment (SQL Azure does not support)
- Trigger SQL Preview

Editing SQL Server Trigger General

Trigger Type

Choose Table or View on which the DML trigger is executed.

Enable

Check this option to enable the trigger.

Table Name or View Name

Choose a table or a view.

Fire

After

Specify that the DML trigger is fired only when all operations specified in the triggering SQL statement have executed successfully.

Instead of

Specify that the DML trigger is executed instead of the triggering SQL statement, therefore, overriding the actions of the triggering statements.

On event

Insert

The trigger is activated whenever a new row is inserted into the table.

Update

The trigger is activated whenever a row is modified.

Delete

The trigger is activated whenever a row is deleted from the table.

Definition Type

Choose the definition type.

Note: Support from SQL Server 2005 or later.

Setting Advanced SQL Server Trigger Properties

Execute as

Specify the security context under which the trigger is executed.

Note: Support from SQL Server 2005 or later and SQL Azure.

Options

Note: SQL Azure does not support.

Encrypted

Obfuscate the text of the CREATE TRIGGER statement.

Note: Support from SQL Server 2005 or later.

Not for replication

Indicate that the trigger should not be executed when a replication agent modifies the table that is involved in the trigger.


Append

Specify that an additional trigger of an existing type should be added.







Editing SQL Server Trigger Definition

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

Note: This tab will appear when the **Definition Type** is set to **SQL statements** in General tab or when connecting to SQL Azure.

The **Code Outline** window displays information about the trigger. To show the **Code Outline** window, simply click  button.

Note: Available only in Full Version.

	Refresh the code outline.
	Turn mouse over highlight on or off.
	Show the code outline in a detail way.
	Sort by type and name.
	Expand the selected item.
	Collapse the selected item.



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

SQL Server Linked Servers

A linked server configuration enables SQL Server to execute commands against OLE DB data sources on remote servers. Linked servers offer the following advantages:


- Remote server access.
- The ability to issue distributed queries, updates, commands, and transactions on heterogeneous data sources across the enterprise.
- The ability to address diverse data sources similarly.

Note: SQL Azure does not support.



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

Create Linked Server

To create a new linked server


- Select anywhere on the object pane.
- Click the  from the object pane toolbar.
or
- Control-click and select **New Linked Server...** from the popup menu.
- Edit linked server properties on the appropriate tabs of the Linked Server Designer.

To create a new linked server with modification as one of the existing linked server

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

Edit Linked Server

To edit the existing linked server (manage its general etc)

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


Test Linked Server

To test the existing linked server

- Select the linked server for editing in the object pane.
- Control-click and select the **Test Linked Server** from the popup menu.


Delete Linked Server

To delete a linked server

- Select the linked server for deleting in the object pane.
- Control-click and select the **Delete Linked Server...** from the popup menu.
or
- Click the  from the object pane toolbar.
- Confirm deleting in the dialog window.

Achieve Linked Server Information

To achieve a linked server information

- Select the linked server in the object pane.
- Choose View -> Object Information in the main menu to view the Object Information.
or
- Click the  from the object pane toolbar.

SQL Server Linked Server Designer

Linked Server Designer is the basic Navicat tool for working with linked servers. It allows you to create new linked server and edit the existing linked server properties.

- [Editing Linked Server General](#)
- [Editing Linked Server Security](#)
- [Setting Advanced Linked Server Properties](#)
- Linked Server SQL Preview

Editing SQL Server Linked Server General

Server Type

Choose the server type: **SQL Server** or **Other data source**

If you choose **Other data source**, define the required information.

Provider

Choose the unique programmatic identifier (PROGID) of the OLE DB provider corresponding to the data source.

Product Name

Define the product name of the OLE DB data source to add as a linked server.

Data Source

Define the name of the data source as interpreted by the OLE DB provider.

Provider String

Define the OLE DB provider-specific connection string that identifies a unique data source.

Location

Define the location of the database as interpreted by the OLE DB provider.

Catalog

Define the catalog to be used when making a connection to the OLE DB provider.

Editing SQL Server Linked Server Security

In this tab, add or delete a mapping between logins on the local instance of SQL Server and remote logins on the linked server.

Local Login

Choose a login on the local server.

Impersonate

Check this option to specify that logins use their own credentials to connect to the linked server.

Remote User

Enter the username used to connect the linked server.

Remote Password

Enter the user password.

Set the action when a login not defined in the list:

- not be made
- be made without using a security context
- be made using the login's current security context
- be made using the following security context:
Set the Remote login and Password

Setting Advanced SQL Server Linked Server Properties

Connect timeout

Define the time-out value for connecting to a linked server. If 0, use the sp_configure default.

Query timeout

Define the time-out value for queries against a linked server. If 0, use the sp_configure default.

Data Access

Check this option to enable a linked server for distributed query access.

Collation Compatible

If this option is checked, SQL Server assumes that all characters in the linked server are compatible with the local server, with regard to character set and collation sequence (or sort order). This enables SQL Server to send comparisons on character columns to the provider.

Use Remote Collation

If this option is checked, the collation of remote columns is used for SQL Server data sources, and the collation specified in collation name is used for non-SQL Server data sources.

Collation

Specify the name of the collation used by the remote data source if Use Remote Collation is checked and the data source is not a SQL Server data source. The name must be one of the collations supported by SQL Server.

Lazy Schema Validation

If this option is checked, skip schema checking of remote tables at the beginning of the query.

Publisher

Check this option to enable publisher.

Subscriber

Check this option to enable subscriber.

Distributor

Check this option to enable distributor.

RPC

Check this option to enable RPC from the given server.

RPC Out

Check this option to enable RPC to the given server.

Promotion of Distributed Transactions for RPC



Use this option to protect the actions of a server-to-server procedure through a Microsoft Distributed Transaction Coordinator (MS DTC) transaction.

Note: Support from SQL Server 2005 or later.

SQL Server Server Triggers


A server trigger can be a DDL or logon trigger for current server. DDL triggers execute in response to a variety of data definition language (DDL) events. These events primarily correspond to Transact-SQL CREATE, ALTER, and DROP statements, and certain system stored procedures that perform DDL-like operations. Logon triggers fire in response to the LOGON event that is raised when a user sessions is being established.

Note: Support from SQL Server 2005 or later.



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

Create Server Trigger

To create a new server trigger


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

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

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

Edit Server Trigger

To edit the existing server trigger(manage its general etc)

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


Maintain Server Trigger

To maintain a server trigger

- Select the server trigger for maintaining in the object pane.
- Control-click and select the **Maintain** from the popup menu.
 - Enable Trigger
 - Disable Trigger


Delete Server Trigger

To delete a server trigger

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

Achieve Server Trigger Information

To achieve a server trigger information

- Select the server trigger in the object pane.
- Choose View -> Object Information in the main menu to view the Object Information.
or
- Click the  from the object pane toolbar.

SQL Server Server Trigger Designer

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

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

Editing SQL Server Server Trigger General

Trigger Type

Choose the trigger type.

Enable

Check this option to enable the trigger.

Definition Type

Choose the definition type.

Event

Check the DDL event form the list.

Setting Advanced SQL Server Server Trigger Properties

Execute as

Specify the security context under which the trigger is executed.

Options


Encrypted

Obfuscate the text of the CREATE TRIGGER statement.







Editing SQL Server Server Trigger Definition

The **Definition** tab allows you to edit valid SQL or procedure statements in the server trigger definition.

Note: This tab will appear when the **Definition Type** is set to **SQL statements** in General tab.

The **Code Outline** window displays information about the trigger. To show the **Code Outline** window, simply click  button.

Note: Available only in Full Version.



	Refresh the code outline.
	Turn mouse over highlight on or off.
	Show the code outline in a detail way.
	Sort by type and name.
	Expand the selected item.
	Collapse the selected item.

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

SQL Server Assemblies


An assembly is a managed application module that contains class metadata and managed code as an object in an instance of SQL Server. By referencing this module, common language runtime (CLR) functions, stored procedures, triggers, user-defined aggregates, and user-defined types can be created in the database.

Note: Support from SQL Server 2005 or later.



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

Create Assembly

To create a new assembly


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

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

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

Edit Assembly

To edit the existing assembly (manage its general etc)

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


Maintain Assembly

To maintain an assembly

- Select the assembly for maintaining in the object pane.
- Control-click and select the **Maintain** from the popup menu.
 - Set Visible
 - Set Invisible


Delete Assembly

To delete an assembly

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

Achieve Assembly Information

To achieve a assembly information

- Select the assembly in the object pane.
- Choose View -> Object Information in the main menu to view the Object Information.
or
- Click the  from the object pane toolbar.

SQL Server Assembly Designer

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

- [Editing Assembly General](#)
- Editing Assembly Comment
- Assembly SQL Preview

Editing SQL Server Assembly General

Assembly Owner

Specify the name of a user or role as owner of the assembly.

Unchecked data

This option allows postponing the checks until a later time by using DBCC CHECKTABLE.

Permission Set

Specify a set of code access permissions that are granted to the assembly when it is accessed by SQL Server. If not specified, SAFE is applied as the default.

Visible

Indicate whether the assembly is visible for creating common language runtime (CLR) functions, stored procedures, triggers, user-defined types, and user-defined aggregate functions against it.

Assembly

Specify the local path or network location where the assembly that is being uploaded is located, and also the manifest file name that corresponds to the assembly.



Dependent Assemblies

Uploads a file to be associated with the assembly, such as source code, debug files or other related information, into the server and made visible in the sys.assembly_files catalog view.

SQL Server Database Triggers


A database trigger is a DDL trigger to the current database. DDL triggers execute in response to a variety of data definition language (DDL) events. These events primarily correspond to Transact-SQL CREATE, ALTER, and DROP statements, and certain system stored procedures that perform DDL-like operations.

Note: Support from SQL Server 2005 or later and SQL Azure.



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

Create Database Trigger

To create a new database trigger


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

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

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

Edit Database Trigger

To edit the existing database trigger(manage its general etc)

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


Maintain Database Trigger

To maintain a database trigger

- Select the database trigger for maintaining in the object pane.
- Control-click and select the **Maintain** from the popup menu.
 - Enable Trigger
 - Disable Trigger


Delete Database Trigger

To delete a database trigger

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

Achieve Database Trigger Information

To achieve a database trigger information

- Select the database trigger in the object pane.
- Choose View -> Object Information in the main menu to view the Object Information.
or
- Click the  from the object pane toolbar.

SQL Server Database Trigger Designer

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

- [Editing Database Trigger General](#)
- [Setting Advanced Database Trigger Properties](#)
- [Editing Database Trigger Definition](#)
- Editing Database Trigger Comment (SQL Azure does not support)
- Database Trigger SQL Preview

Editing SQL Server Database Trigger General

Trigger Type

Trigger type must be Database Trigger.

Enable

Check this option to enable the trigger.

Definition Type

Choose the definition type.

Note: SQL Azure does not support.

Event

Check the DDL event form the list.

Setting Advanced SQL Server Database Trigger Properties

Execute as

Specify the security context under which the trigger is executed.

Options

Note: SQL Azure does not support.


Encrypted

Obfuscate the text of the CREATE TRIGGER statement.







Editing SQL Server Database Trigger Definition

The **Definition** tab allows you to edit valid SQL or procedure statements in the database trigger definition.

Note: This tab will appear when the **Definition Type** is set to **SQL statements** in General tab or when connecting to SQL Azure.

The **Code Outline** window displays information about the trigger. To show the **Code Outline** window, simply click  button.

Note: Available only in Full Version.



	Refresh the code outline.
	Turn mouse over highlight on or off.
	Show the code outline in a detail way.
	Sort by type and name.
	Expand the selected item.
	Collapse the selected item.

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

SQL Server Partition Functions


A partition function is a function in the current database that maps the rows of a table or index into partitions based on the values of a specified column.

Note: Support from SQL Server 2005 or later.



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

Create Partition Function

To create a new partition function


- Select anywhere on the object pane.
- Click the  from the object pane toolbar.
or
- Control-click and select **New Partition Function...** from the popup menu.
- Edit partition function properties on the appropriate tabs of the Partition Function Designer.

To create a new partition function with modification as one of the existing partition function

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

Edit Partition Function

To edit the existing partition function (manage its general etc)


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

To change the name of the partition function

- Select the partition function for editing in the object pane.
- Control-click and select the **Rename** from the popup menu.


Delete Partition Function

To delete a partition function

- Select the partition function for deleting in the object pane.
- Control-click and select the **Delete Partition Function...** from the popup menu.
or
- Click the  from the object pane toolbar.
- Confirm deleting in the dialog window.

Achieve Partition Function Information

To achieve a partition function information

- Select the partition function in the object pane.
- Choose View -> Object Information in the main menu to view the Object Information.
or
- Click the  from the object pane toolbar.

SQL Server Partition Function Designer

Partition Function Designer is the basic Navicat tool for working with partition functions. It allows you to create new partition function.

- [Editing Partition Function General](#)
- Editing Partition Function Comment
- Partition Function SQL Preview

Editing SQL Server Partition Function General

Parameter

Type

Choose the data type of the column used for partitioning.

Length

Specify the length of the data type if necessary.

Decimals

Specify the decimals of the data type if necessary.

Collation

Specify the collation of the data type if necessary.

Boundary values belongs to right interval

Specify to the right of each boundary value interval.



Boundary values

Specify the boundary values for each partition of a partitioned table or index that uses `partition_function_name`.

SQL Server Partition Schemes


A partition scheme is a scheme in the current database that maps the rows of a table or index into partitions based on the values of a specified column. The number and domain of the partitions of a partitioned table or index are determined in a partition scheme.

Note: Support from SQL Server 2005 or later.



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

Create Partition Scheme

To create a new partition scheme


- Select anywhere on the object pane.
- Click the  from the object pane toolbar.
or
- Control-click and select **New Partition Scheme...** from the popup menu.
- Edit partition scheme properties on the appropriate tabs of the Partition Scheme Designer.

To create a new partition scheme with modification as one of the existing partition scheme

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

Edit Partition Scheme

To edit the existing partition scheme (manage its general etc)


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

To change the name of the partition scheme

- Select the partition scheme for editing in the object pane.
- Control-click and select the **Rename** from the popup menu.


Delete Partition Scheme

To delete a partition scheme

- Select the partition scheme for deleting in the object pane.
- Control-click and select the **Delete Partition Scheme...** from the popup menu.
or
- Click the  from the object pane toolbar.
- Confirm deleting in the dialog window.

Achieve Partition Scheme Information

To achieve a partition scheme information

- Select the partition scheme in the object pane.
- Choose View -> Object Information in the main menu to view the Object Information.
or
- Click the  from the object pane toolbar.

SQL Server Partition Scheme Designer

Partition Scheme Designer is the basic Navicat tool for working with partition functions. It allows you to create new partition scheme.

- [Editing Partition Scheme General](#)
- Editing Partition Scheme Comment
- Partition Scheme SQL Preview

Editing SQL Server Partition Scheme General

Partition Function

Choose the partition function.

Filegroups

Specify the filegroups to hold the partitions specified by partition_function_name.