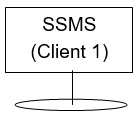
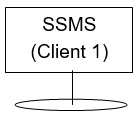
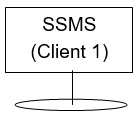
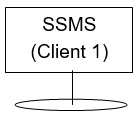
SQL Server

Connection to SSMS (SQL Server Management Studio)

SSMS (SQL Server Management Studio)

SSMS is client tool used to connect or communicate with SQL server (Database Server). It is not the server by itself. We connect SSMS to database server by specify server’s configuration in it.





Database Server

Object Explorer in SSMS

Object explorer shows all SQL server objects.

SQL server objects are:

1. Database
2. Trigger
3. Stored Procedure
4. Functions
5. View

Open Object Explorer: -

View -> Object Explorer

System Databases:

These databases installed automatically at the time of installation of SSMS. These are required for the functionality of the SSMS.

System Databases are:

1. master
2. model
3. msdb
4. tempdb

Database

Database is sql server object.

Create/Drop Database:

Create database <DatabaseName>

When we create a database, two types of files gets generated automatically (i.e. every database creation weather system/user define database, two files get automatically generated). These two files are:

1. <DatabaseName>.mdf

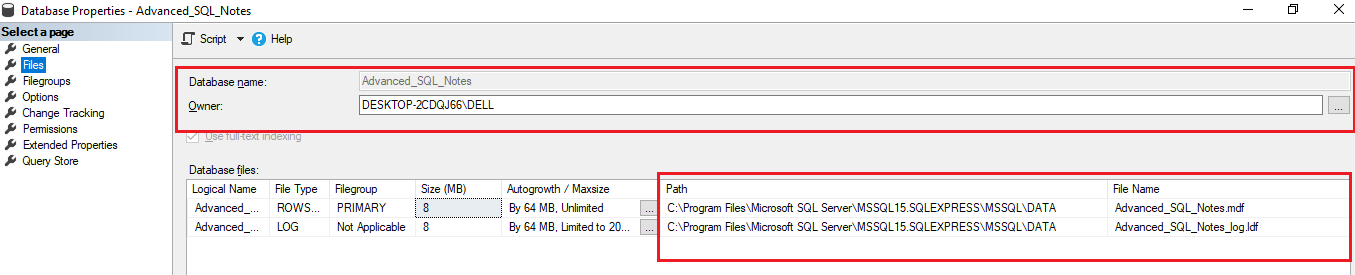
This file contains actual data.

1. <DatabaseName>.ldf

This file contains transaction log file. It is used to recover database.

Steps to view these files:

Right click Database > Properties > Files Tab > Path



Drop Database:

Drop database <DatabaseName>

Dropping a database, deletes the .ldf and .mdf files behind the scenes.

We can’t drop a database if it is currently in use.

So, If other users are connected, we should put databae in single user mode and then drop it.

Set Database at single user mode

Alter database <DatabaseName>

Set single\_user with Rollback immediate

Renaming Database:

1. Alter Database <OldDatabaseName>

Modify Name=<newDBName>

1. execute sp\_rename <oldDBName> <newDBName>

Table

Creating Table & enforce primary key, Unique Key ,Foreign Key constraints, check constraint, default constraint:

Create table <TableName>

(

[ColumnName1] <columnDataType><lengthOfCharacters> [<nullability constraint>],

[ColumnName2] <columnDataType><lengthOfCharacters> [<nullability constraint>]

Constraint <df\_ConstraintName> default (<defaultValue>),

Constraint <PK\_ConstraintName> **Primary Key** ([colName1][,colName2],…),

Constraint <UQ\_ConstraintName> **Unique** ([colName1][,colName2],…),

Constraint <FK\_ConstraintName> **Foreign Key** (FK\_ColumnName1)

References <TableNameWhichIsToRefer>(PrimaryKeyColumnName),

Constraint <ck\_constriantName> **Check** (<ConditionReturnsTrueOrFalse>),

~~Constraint <df\_constriantName>~~ **~~Default~~** ~~(DefaultValue) for <columnName>~~

)

Nullability Constraint:

Two nullability constraint:

1. null
2. not null

Key Points

1. If column name contains spaces then enclose column name inside [ ],

otherwise, no need to enclosed in [ ].

1. If we don’t specify nullability constraint then it will be ‘null’ by default.
2. The foreign key should refer to the primary key column of another table.
3. We can define default constraint inline only.

If we define default constraint like strikethrough statement then Error.

Example:

Bank (Account Number, Bank Name, IFSC)

Employee (Employee Id, Employee Name, Father Name, Aadhar, BankAccount)



Constraint (Rule)

Constraint are the predefined rules and restrictions that are enforced in single or multiple columns, regarding the values allowed in columns, to maintain integrity, accuracy and reliability of that columns data.

If the inserted data meets the constraint rule, it will be inserted into table successfully. If inserted data violates any of the defined constraint, the data will not be inserted and insert operation will be aborted.

Constraint in SQL server can be defined:

1. **Column Level Constraint:**

The constraint applied to only one column of table are called column level constraint. Column level constraints refers to a single column in table and do not specify a column name (Except check constraint). These refers to the column that they follow.

1. **Table Level Constraint**:

The constraint applied on two or more columns in table are called Table level constraint. Table level constraints specify the names of the columns to which they apply.

E.g.: Check Constraint

SQL constraints are used to specify rules for the data.

The constraint can be created within Create Table command while creating table or added using Alter Table command after creating table.

**Note:**

Adding constraint after creating table, the existing data will be checked for the specified constraints rule before creating those constraints.

Constraints used in SQL are:

1. Nullability Constraint/ Not Null Constraint
2. Unique
3. Primary Key
4. Foreign Key
5. Check
6. Default
7. Create Index

Nullability constraint / Set NULL Constraint

If we don’t define any nullability constraint on any column of table then, by default, the columns are able to hold NULL values.

A NOT NULL constraint in SQL is used to prevent inserting NULL values into the specified column, considering it as a not accepted value for that column. This means that you should provide a valid SQL NOT NULL value to that column in the INSERT or UPDATE statements, as the column will always contain data.

If we try to insert null inside Not Null Column the, there will be Error.

Create Nullability Constraint on any column

1. At time of Creating Table

Create Table Person

(

[Id] nvarchar(10) not null,

[Name] nvarchar(100) not null,

[Aadhar] nvarchar(12) null,

[Email] nvarchar(50)

)

1. Alter table:
2. Alter table <tableName>
3. Constraint

Unique Constraint

Unique constraint enforces uniqueness of column i.e., the column should not allow any duplicate values.

Primary Key Constraint

The PRIMARY KEY constraint consists of one column or multiple columns with values that uniquely identify each row in the table.

The SQL PRIMARY KEY constraint combines between the UNIQUE and SQL NOT NULL constraints.

where the column or set of columns that are participating in the PRIMARY KEY cannot accept a NULL value.

If the PRIMARY KEY is defined in multiple columns, you can insert duplicate values on each column individually, but the combination values of all PRIMARY KEY columns must be unique.

Take into consideration that you can define only one PRIMARY KEY per each table, and it is recommended to use small or INT columns in the PRIMARY KEY.

In addition to providing fast access to the table data, the index that is automatically created, when defining the SQL PRIMARY KEY, will enforce the data uniqueness. The PRIMARY KEY is used mainly to enforce the entity integrity of the table. Entity integrity ensures that each row in the table is a uniquely identifiable entity.

PRIMARY KEY constraint differs from the UNIQUE constraint in that; you can create multiple UNIQUE constraints in a table, with the ability to define only one SQL PRIMARY KEY per each table. Another difference is that the UNIQUE constraint allows for one NULL value, but the PRIMARY KEY does not allow NULL values.