SQL SERVER

Overview of SQL Server

- SQL server is the most popular database of Microsoft Company which was developed in 1989.
- It is RDBMS so in this case record is stored in tabular format.
- It is platform dependent because it runs only on windows operating system.
- It is not free to use because it is not open source technology.
- Its installation is easy because it is product of Microsoft.
- It is not case sensitive.
- It provides SQL(Structured Query Language)
- There are many edition of SQL server available in Market. Such as Enterprise, Developer, compact, web etc.

SQL Server Edition:

SQL Server is available in several edition:

- Enterprises Edition: It is a top level edition with full feature.

- Standard Edition: It has fewer features than enterprises edition. It needs when there is no requirement of advance feature.

- ***Web Edition:*** ***It is designed for web Application.***

- ***Developer Edition:*** ***It is similar to enterprises edition but it is licensed to only one user for development, testing and demo. It can be easily upgraded to enterprises without reinstallation.***

- ***Express Edition:*** ***It is free entry level database.***

- ***Compact Edition:*** ***It is free embedded database for mobile application development.***

- ***Workgroup Edition:*** ***It is suitable for remote office of a large company.***

Server Authentication:

There are two types of authentication in SQL Server

- ***Windows authentication:***

Windows authentication work on the user admin and when we work on the user admin and when we work on windows authentication then there is no required user name and password because operating system generates user id and password by default.

- ***SQL Server authentication:*** ***SQL server works on current user and when we work on SQL server authentication then user should enter user id and password. The user id and password will give at the time of SQL server installation.***

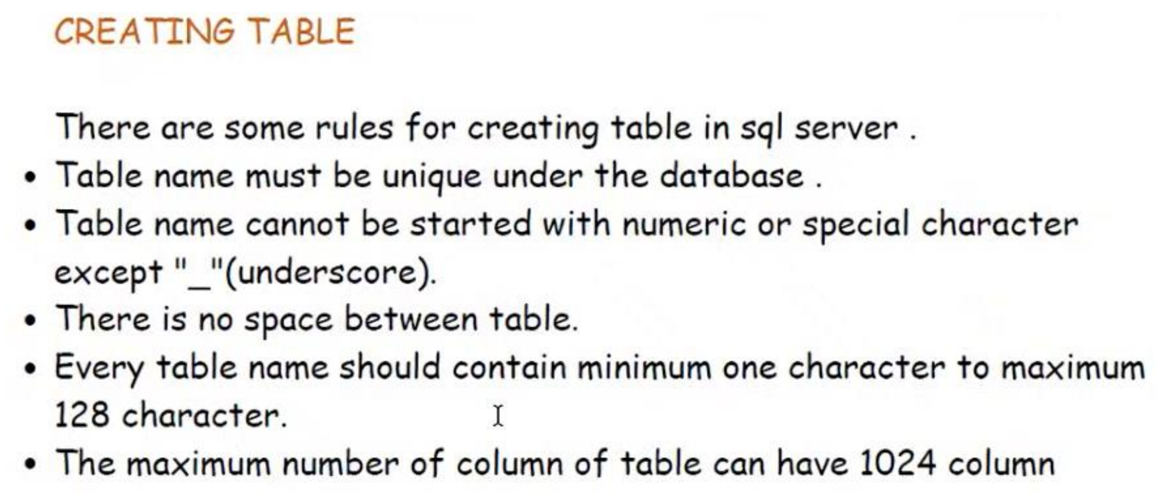
i) CREATE DATABASE statement is used to create a new SQL database.

Syntax: ***CREATE DATABASE dbname;***

Example: ***CREATE DATABASE cimage***

i) To remove your database from MS SQL Server, use drop database command.

Syntax: *Drop database <your database name>*
Ex: *DROP DATABASE cimage;*



CREATING TABLE

There are some rules for creating table in sql server .

- Table name must be unique under the database .
- Table name cannot be started with numeric or special character except "_"(underscore).
- There is no space between table.
- Every table name should contain minimum one character to maximum 128 character.
- The maximum number of column of table can have 1024 column

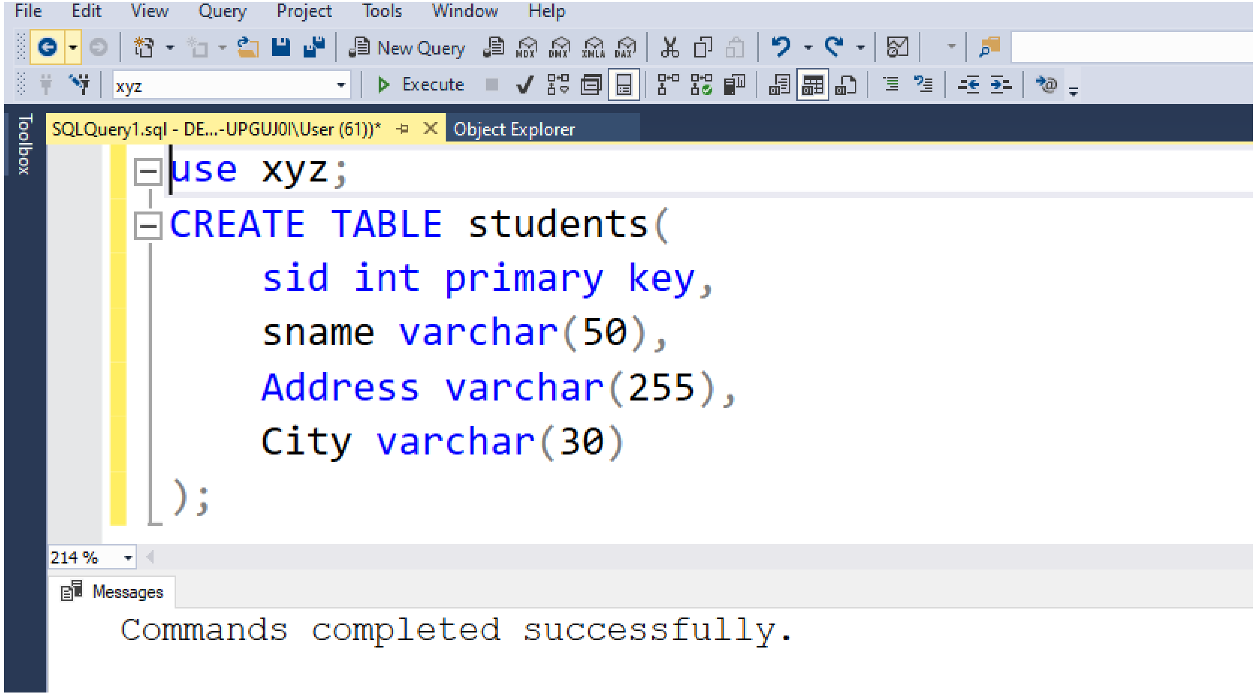
ii) The CREATE TABLE statement is used to create a new table in a database.

Syntax: CREATE TABLE table_name (

```
    column1 datatype constraint,
    column2 datatype constraint,
    column3 datatype constraint,
   ....
);
```

Example:

```
CREATE TABLE students(
    sid    int primary key,
    sname varchar(50),
    Address varchar(255),
    City varchar(30)
);
```

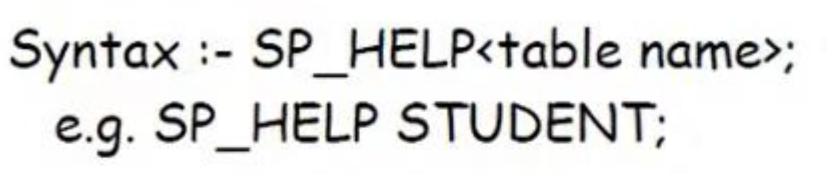


```
use xyz;
CREATE TABLE students(
    sid int primary key,
    sname varchar(50),
    Address varchar(255),
    City varchar(30)
);
```

```
Commands completed successfully.
```

Q) How to display structure of Table?

SP_HELP command is used to show table

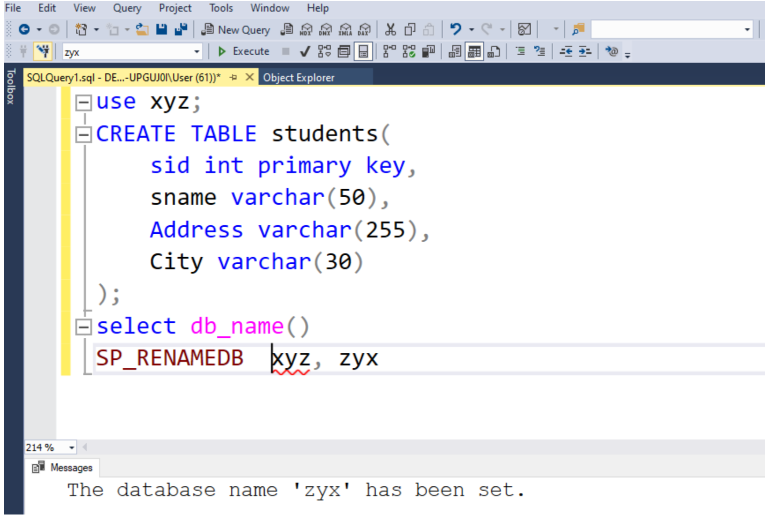

Syntax :- SP_HELP<table name>;
e.g. SP_HELP STUDENT;

Q) How do display current database?

Ex- Select DB_name()

Q) How to rename database?

Ex- SP_RENAMEDB xyz, zyx


File Edit View Query Project Tools Window Help
New Query
Execute
zyx
SQLQuery1.sql - DE...-UPGUJ0I\User (61))*
Object Explorer
Toolbox
214 %
Messages

```sql
use xyz;
CREATE TABLE students(
    sid int primary key,
    sname varchar(50),
    Address varchar(255),
    City varchar(30)
);
select db_name()
SP_RENAMEDB  xyz, zyx
```

```
The database name 'zyx' has been set.
```

Q) How to drop table?

DROP command is used to drop table

Syntax: DROP table <table name>

Eg drop table students

- The ALTER TABLE statement is used to add, delete, or modify columns in an existing table.

  Syntax: ALTER TABLE table_name
                   ADD column_name datatype;

```
Ex:-            ALTER TABLE students
                ADD Email varchar(55);
```

❖To delete a column in a table:

Syntax:

```
                ALTER TABLE table_name
                DROP COLUMN column_name;
```

```
Ex:-            ALTER TABLE students
                DROP COLUMN Email;
```

❖To rename a column in a table:

Syntax:

```
ALTER TABLE table_name
RENAME COLUMN old_name to new_name;
```

Ex:-

ALTER TABLE students

RENAME COLUMN ‘Email’ to ‘EmailID’;

❖ How to Rename a table?

SYNTAX:-

```
SP_RENAME old_tablename, new_tablename
```

EX-

```
sp_rename employee,  employees
```

```
ALTER TABLE stu
ADD dob date;
```

Syntax:

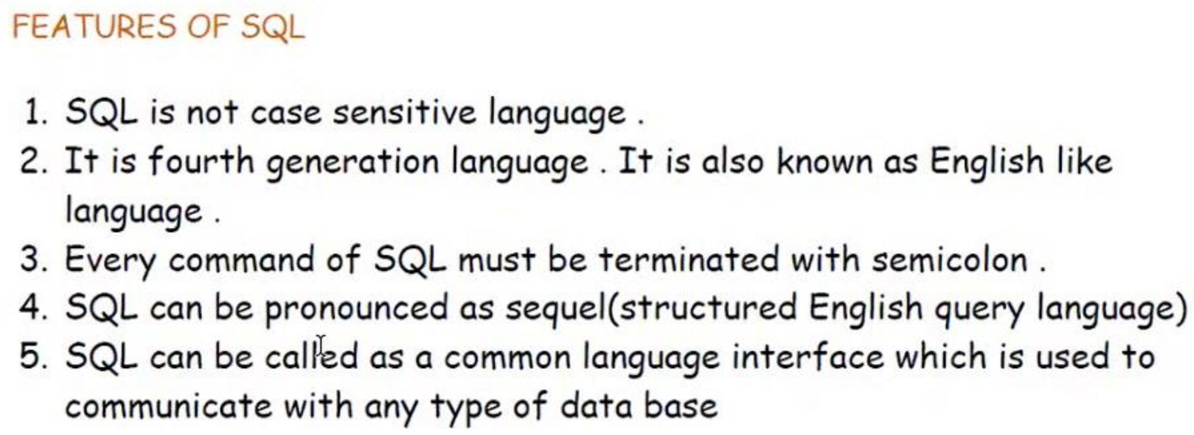
ALTER TABLE table_name

ALTER COLUMN column_name datatype;

Ex-

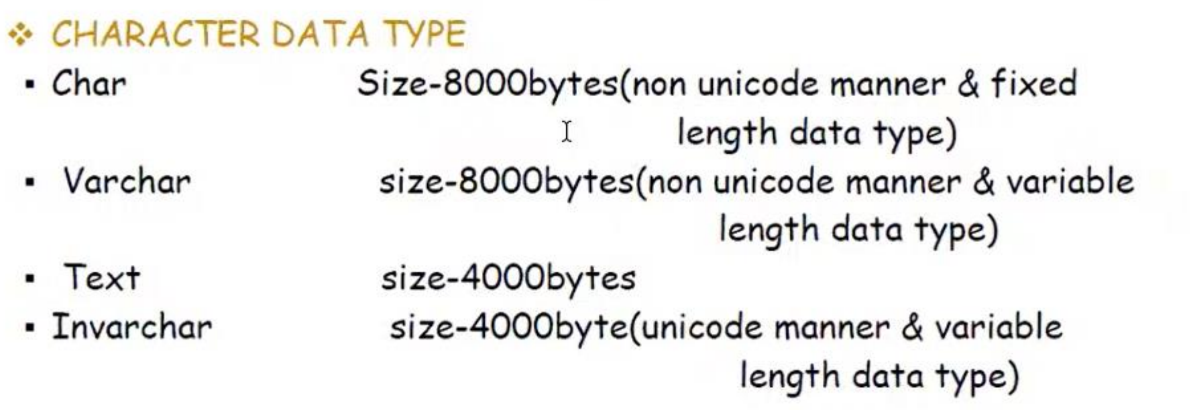
```
ALTER TABLE stu
ALTER COLUMN dob DATETIME;

ALTER TABLE dbo.stu ALTER COLUMN email DECIMAL (5, 2);
```

```
# change column name

EXEC sp_RENAME 'stu.email', 'emails', 'COLUMN'

EXEC sp_RENAME 'stu.emails', 'emaiD'
```



FEATURES OF SQL

1. SQL is not case sensitive language .
2. It is fourth generation language . It is also known as English like language .
3. Every command of SQL must be terminated with semicolon .
4. SQL can be pronounced as sequel(structured English query language)
5. SQL can be called as a common language interface which is used to communicate with any type of data base



❖ CHARACTER DATA TYPE

Char	Size-8000bytes(non unicode manner & fixed length data type)
Varchar	size-8000bytes(non unicode manner & variable length data type)
Text	size-4000bytes
Invarchar	size-4000byte(unicode manner & variable length data type)

Classification of SQL Command

SQL can be divided into 4 parts:

- ❖DDL (Data Definition Language)
- ❖DML(Data Manipulation Language)
- ❖DCL (Data Control Language)
- ❖TCL (Transaction Control Language)

SQL

a) DDL

i) Create

ii) Alter

iii) Truncate

iv) Drop

v) SP_rename

b) DML

i) Select

ii) Insert

iii) Update

iv) Delete

c) TCL

i) Rollback

ii) Commit

iii) Savepoint

d) DCL

i) Grant

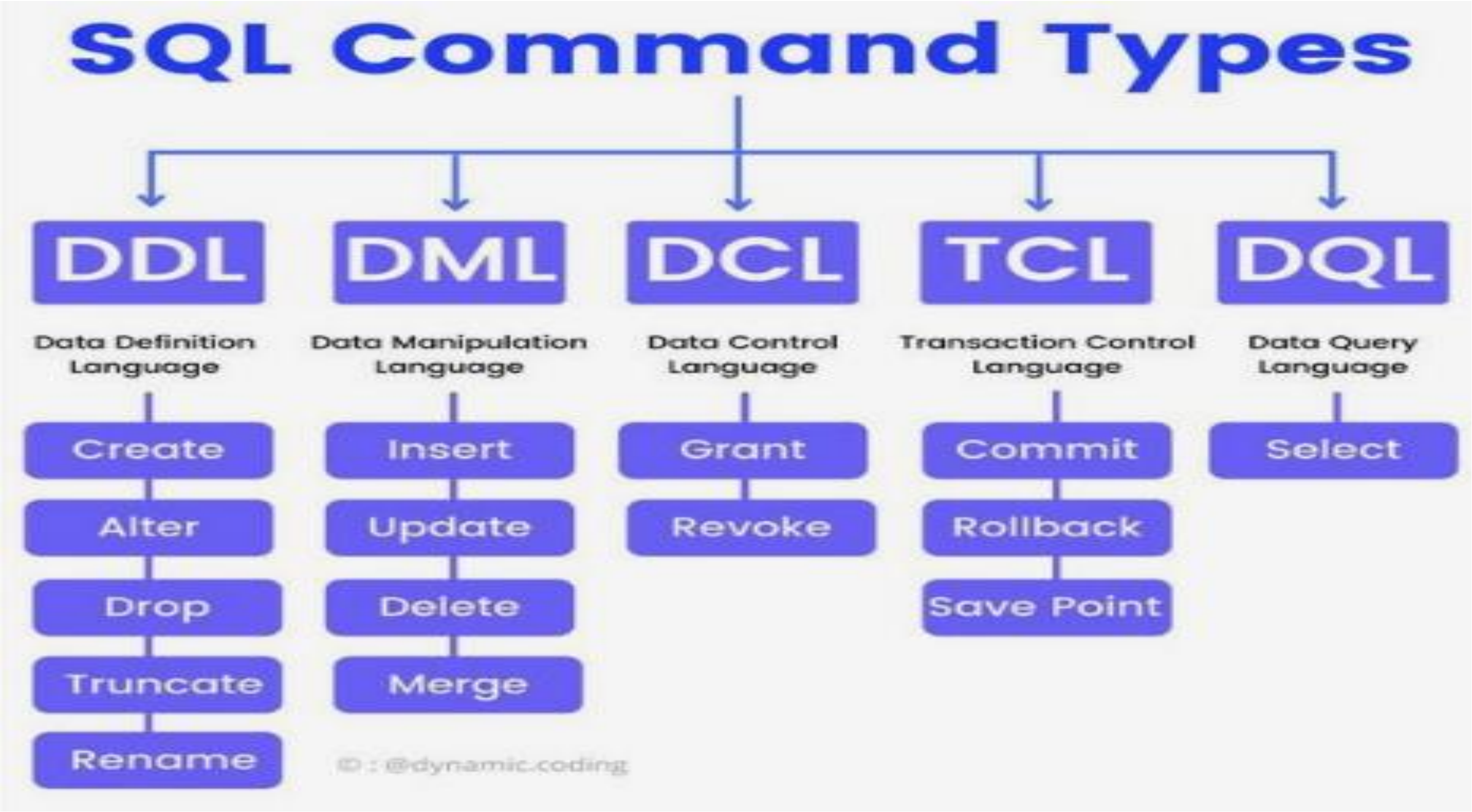
ii) Revoke

SQL COMMAND

- It is a non procedure language which is use to communicate with any data base such as ORACLE , SQL SERVER ,MY SQL , etc.
- This language was developed by the German scientist Mr. E.F. CODD .
- ANSI (American national standered institute)approved this concept and in 1972 SQL was released in to the market .

FEATURES OF SQL

1. SQL is not case sensitive language .
2. It is fourth generation language . It is also known as English like language .
3. Every command of SQL must be terminated with semicolon .
4. SQL can be pronounced as sequel(structured English query language)
5. SQL can be called as a common language interface which is used to communicate with any type of data base



SQL Command Types

DDL
Data Definition Language
- Create
- Alter
- Drop
- Truncate
- Rename

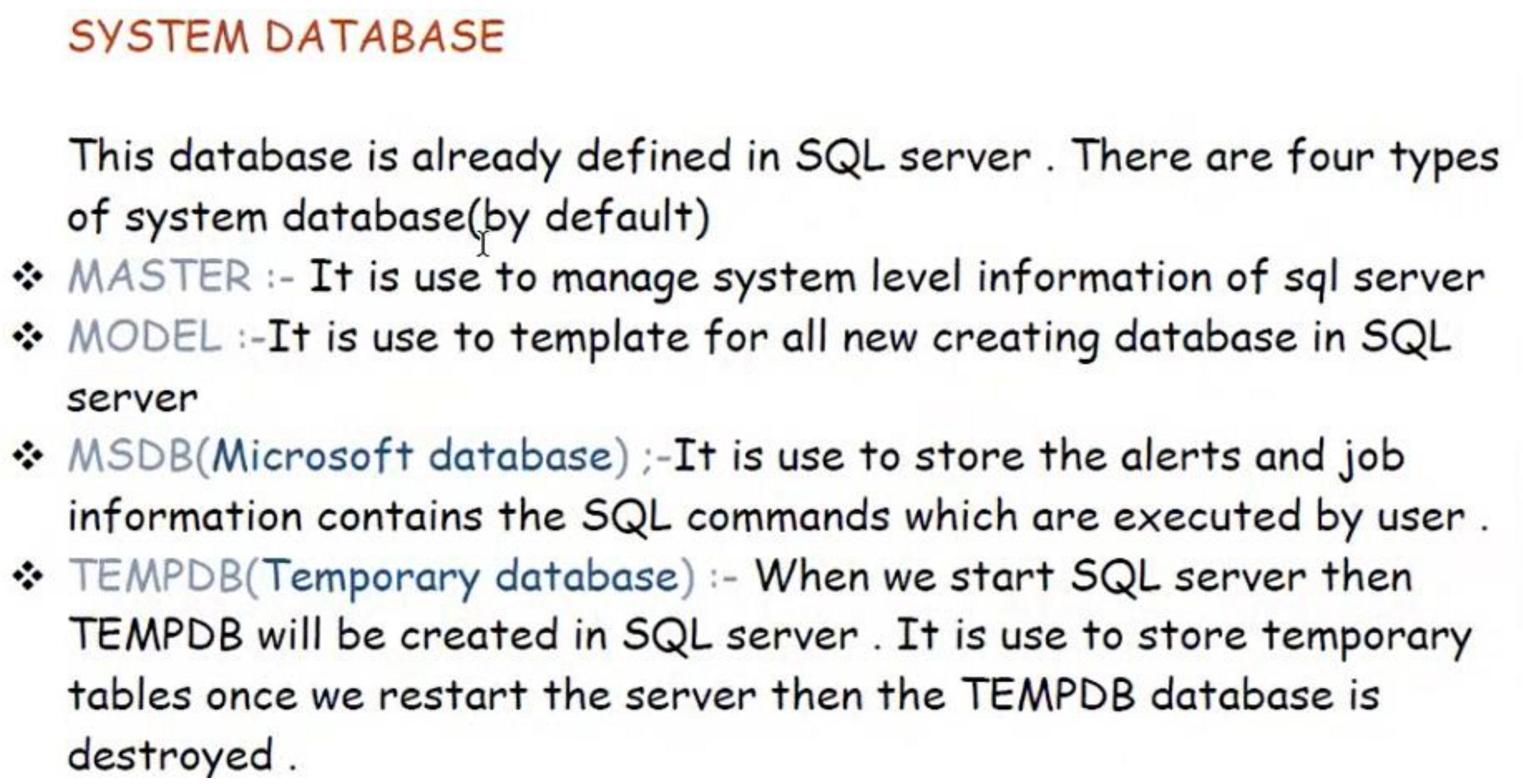
DML
Data Manipulation Language
- Insert
- Update
- Delete
- Merge

DCL
Data Control Language
- Grant
- Revoke

TCL
Transaction Control Language
- Commit
- Rollback
- Save Point

DQL
Data Query Language
- Select

© : @dynamic.coding



SYSTEM DATABASE

This database is already defined in SQL server . There are four types of system database(by default)

- MASTER :- It is use to manage system level information of sql server
- MODEL :-It is use to template for all new creating database in SQL server
- MSDB(Microsoft database) ;-It is use to store the alerts and job information contains the SQL commands which are executed by user .
- TEMPDB(Temporary database) :- When we start SQL server then TEMPDB will be created in SQL server . It is use to store temporary tables once we restart the server then the TEMPDB database is destroyed .

SQL Server provides the different types of data type

1) Integer
2) Decimal
3) Date and time
4) Character
5) Binary

SQL Constraints:

The available constraints in SQL are:

i) **NOT NULL:** This constraint tells that we cannot store a null value in a column.

Ex:-
```
fname varchar(10) NOT NULL,
```

ii) **UNIQUE:** This constraint when specified with a column, tells that all the values in the column must be unique.

iii) **PRIMARY KEY:** A primary key is a field which can uniquely identify each row in a table.

```
CREATE TABLE Student
(
ID int NOT NULL UNIQUE,
NAME varchar(10),
ADDRESS varchar(20),
PRIMARY KEY(ID)
);
```

iv) **FOREIGN KEY:** A Foreign key is a field which can uniquely identify each row in a another table.

```
CREATE TABLE Orders
(
O_ID int NOT NULL,
ORDER_NO int NOT NULL,
C_ID int,
PRIMARY KEY (O_ID),
FOREIGN KEY (C_ID) REFERENCES Customers(C_ID)
)
```

v) **CHECK:** This constraint helps to validate the values of a column to meet a particular condition.

```
CREATE TABLE Student
```

```
(
ID int NOT NULL,
NAME varchar(10) NOT NULL,
AGE int NOT NULL CHECK (AGE >= 18)
);
```

vi) **DEFAULT:** This constraint specifies a default value for the column when no value is specified by the user.

```
CREATE TABLE Student
(
ID int(6) NOT NULL,
NAME varchar(10) NOT NULL,
AGE int DEFAULT 18
);
```

vii) The INDEX is used to create and retrieve data from the database very quickly. An Index can be created by using a single or group of columns in a table.

```
CREATE INDEX cindx
ON orders (oid, cid);
```

The constraints used in SQL are:

Constraint Description

NOT NULL	values cannot be null
UNIQUE	values cannot match any older value
PRIMARY KEY	used to uniquely identify a row
FOREIGN KEY	references a row in another table
CHECK	validates condition for new value

DEFAULT	set default value if not passed
CREATE INDEX	used to speedup the read process

Data Types

Text Datatypes

The text has the capacity to store from 1 byte to 4 Gigabytes of data. We need to specify length in char and varchar but in the case of text, we do not need to specify length. Yet text works slower than char and varchar. There are several categories in Text Datatypes:

TINY TEXT

TEXT

MEDIUM TEXT

LONG TEXT

NTEXT

a) TINY TEXT

The non-Unicode is the character string datatype that stores data up to 255 characters. For example, the following statement creates a column called stu_name with a length of 255 characters and we also use TEXT for the large number of characters.

```
    CREATE TABLE students (
      id INT NOT NULL AUTO_INCREMENT,
      stu_name TINYTEXT NOT NULL
);
```

b) TEXT

It is a non-Unicode, character string datatype storing around 64KB of data. For example, the following statement creates a column called stu_name with a length of 65,535 characters:

```
     CREATE TABLE students (
       id INT NOT NULL AUTO_INCREMENT,
       stu_name TEXT NOT NULL
);
```

c) MEDIUM TEXT

It stores up to 16MB of data. We can write description-length data with medium text.

d) LONG TEXT

We can store up to 4GB of data using this non-unicode datatype. Using this data type, we can type data to the length of an article.

e) NTEXT

A Unicode data type that stores the data without having to specify length. Storage size is double the size that is specified in the column. There are no subcategories for next.

Data Type	Descriptions	Lower Range	Upper Range	Storage
bit	It is an integer type that allows us to store 0, 1, and NULL values.	0	1	1 byte

tinyint	It allows us to store whole numbers from 0 to 255.	0	255	1 byte
smallint	It allows us to store whole numbers between -32768 to 32767.	-2^15 (-32,768)	2^15-1 (32,767)	2 byte
int	It allows to store whole numbers between -2,147,483,648 and 2,147,483,647	-2^31 (-2,147,483,648)	2^31-1 (2,147,483,647)	4 byte
bigint	It allows to store whole numbers between -9,223,372,036,854,775,808 and 9,223,372,036,854,775,807	-2^63 (-9,223,372,036,854,775,808)	to 2^63-1 (9,223,372,036,854,775,807)	8 byte
Decimal(p, s)	It is used for scale and fixed precision numbers. The p parameter indicates the maximum total number of digits that can be stored (both to the left and the right of the decimal point). By default, it is 18 but can be in a range of 1 to 38. The s parameter indicates the maximum number of digits stored to the right of the decimal point. By default, its value is 0 but can be from 0 to p.	?10^38+1	10^381?1	5 to 17 bytes
Numeric(p,	It is similar to the decimal	?10^38+1	10^38?1	5 to 17

s)	data type because both are synonyms.			bytes.
smallmoney	It allows storing monetary or currency values.	-214,748.3648	214,748.3647	4 byte
money	It allows to store monetary or currency values.	-922,337,203,685,477.5808	922,337,203,685,477.5807	8 byte

Character string data type

Data Type	Descriptions	Lower Range	Upper Range	Storage
char(n)	It is used to store fixed-length non-Unicode character data.	0 characters	8000 characters	n bytes
varchar(n)	It is used to store variable-length non-Unicode character data.	0 characters	8000 characters	n bytes + 2 bytes

varchar(max)	It is a data type that stores variable-length data. It is recommended to avoid this data type unless required because of its huge memory storage.	0 characters	2^31 characters	n bytes + 2 bytes
text	It is a variable-length character string. It is also recommended to avoid this data type because it would be deprecated in future releases.	0 characters	2,147,483,647 chars	n bytes + 4 bytes

Date and Time data types

Data Type	Descriptions	Lower Range	Upper Range	Storage
date	It is used to store dates in SQL Server. By default, its format is YYYY-MM-DD, and its value is 1900-01-01.	0001-01-01	9999-12-31	3 bytes
datetime	It is used to store date and time with fractional seconds. Its default value is 1900-01-01 00:00:00. Its accuracy is in the increment of .000, .003, or .007 seconds. It is recommended to avoid this data type and use datetime2 instead.	1753-01-01	9999-12-31	8 bytes

datetime2	It specifies date and time with fractional seconds and an accuracy of 100 nanoseconds. It provides precision from 0 to 7 digits. By default, its precision is 7, and the format is YYYY-MM-DD hh:mm: ss[.fractional seconds].	0001-01-01 00:00:00	9999-12-31 23:59:59.9999999	6 to 8 bytes
datetimeoffset	It is the same as datetime2 with the addition of a time zone offset. Its timezone offset value is between -14:00 through +14:00.	0001-01-01 00:00:00	9999-12-31 23:59:59.9999999	10 bytes
smalldatetime	It specifies a date along with the time of day and an accuracy of 1 minute. Here time is calculated on a 24-hour clock, with seconds starting at zero (:00) and no fractional seconds.	1900-01-01 00:00:00	2079-06-06 23:59:59	4 bytes
time	It specifies time data only with an accuracy of 100 nanoseconds. It is based on a 24-hour clock without time zone awareness. By default, its format is hh:mm:ss[.nnnnnnn].	00:00:00.0000000	23:59:59.9999999	3 to 5 bytes

Primary Key:

i) A primary key is a unique identifier for each record in a table.

ii) Does not allow NULL values in the primary key field.

iii) Only one primary key can exist per table.

iv) No duplicate primary key values are allowed within the table.

Foreign Key:

i) A foreign key establishes a relationship between tables by referencing the primary key of another table.

ii) Multiple foreign keys can exist within a table, depending on the relationships with other tables.

iii) Allows NULL values in the foreign key field

iv) Can contain duplicate foreign key values, reflecting multiple records associated with the same reference.

Unique Key:

i) The unique key can store a null value, but only one NULL value is allowed.

ii) It enforces unique data.

iii) A table can have multiple unique keys.

iv) It prevents duplicate entries in a column, except for a NULL value.

---- This is single line comment

/* This is multiline comment */

-- sql constraints

-- in a database table, we can add rules to a column known as constraints

/*

These rules control the data that can be stored in a column

Ex- If a column has 'not null’ constraints. It means the column cannot store 'Null ' values.

The constraints used in SQL are:

```
*/
create table customer(
id int not null primary key,
cname varchar(55),
ccode varchar(20) unique
);
sp_help customer

create table orders(
id int primary key,
cid int references customer(id),
amount int check(amount>100),
country varchar(30) default 'India');
sp_help orders

create index employee_index on employee(ecode)
```

Data integrity is the accuracy, consistency, and completeness of data throughout its lifecycle. It ensures that data is unaltered and free from corruption.