1: Comments in SQL Server:

* single line comment in sql server --
* multiline comment in sql server : /\* your statements \*/

2: Create New Database or database creation in sql server

**Syntax:** Create database DatabaseName

ex:

create database demo;

create database demo;

create database organization;

3:Create a table in database /Table creation in sql server

**Syntax:**

create table TableName(column1 datatype, column2 datatype, column3 datatype,column4 datatype);

ex: if table name is studentmaster then their columns will be like

Studentid datatype, studentname datatype, course datatype,DOB or dateofbirth datatype

create table Studentmaster (studentid int, studentName nvarchar(80),Course nvarchar(15),DOB datetime);

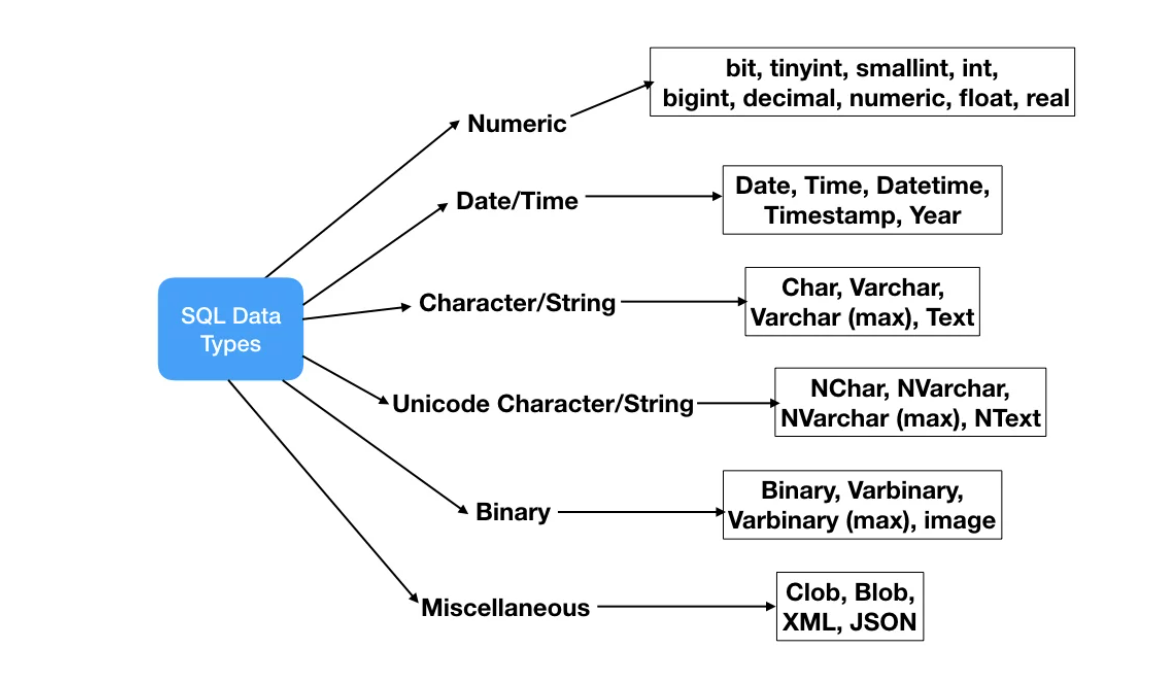
4: **Datatypes in SQL Server:**

SQL Data Types define the type of value that can be stored in a table column. For example, if we want a column to store only integer values, then we can define its data type as int

## SQL Data Types

SQL data types can be broadly divided into following categories.

1. Numeric data types such as int, tinyint, bigint, float, real, etc.
2. Date and Time data types such as Date, Time, Datetime, etc.
3. Character and String data types such as char, varchar, text, etc.
4. Unicode character string data types, for example nchar, nvarchar, ntext, etc.
5. Binary data types such as binary, varbinary, etc.
6. Miscellaneous data types – clob, blob, xml, cursor, table, etc.



### SQL Data Types important points

* Not all data types are supported by every relational database vendor. For example, Oracle database doesn’t support DATETIME and MySQL doesn’t support CLOB data type. So while designing database schema and writing SQL queries, make sure to check if the data types are supported or not.
* Data types listed here doesn’t include all the data types, these are the most popularly used data types. Some relational database vendors have their own data types that might be not listed here. For example, Microsoft SQL Server has money and smallmoney data types but since it’s not supported by other popular database vendors, it’s not listed here.
* Every relational database vendor has its own maximum size limit for different data types, you don’t need to remember the limit. Idea is to have the knowledge of what data type to be used in a specific scenario.

Let’s look into different categories of [SQL](https://www.journaldev.com/16767/sql) data types in detail.

### SQL Numeric Data Types

|  |  |  |
| --- | --- | --- |
| Datatype | From | To |
| bit | 0 | 1 |
| tinyint | 0 | 255 |
| smallint | -32,768 | 32,767 |
| int | -2,147,483,648 | 2,147,483,647 |
| bigint | -9,223,372,036, 854,775,808 | 9,223,372,036, 854,775,807 |
| decimal | -10^38 +1 | 10^38 -1 |
| numeric | -10^38 +1 | 10^38 -1 |
| float | -1.79E + 308 | 1.79E + 308 |
| real | -3.40E + 38 | 3.40E + 38 |

### SQL Date and Time Data Types

|  |  |
| --- | --- |
| Datatype | Description |
| DATE | Stores date in the format YYYY-MM-DD |
| TIME | Stores time in the format HH:MI:SS |
| DATETIME | Stores date and time information in the format YYYY-MM-DD HH:MI:SS |
| TIMESTAMP | Stores number of seconds passed since the Unix epoch (‘1970-01-01 00:00:00’ UTC) |
| YEAR | Stores year in 2 digits or 4 digit format. Range 1901 to 2155 in 4-digit format. Range 70 to 69, representing 1970 to 2069. |

### SQL Character and String Data Types

|  |  |
| --- | --- |
| Datatype | Description |
| CHAR | Fixed length with a maximum length of 8,000 characters |
| VARCHAR | Variable-length storage with a maximum length of 8,000 characters |
| VARCHAR(max) | Variable-length storage with provided max characters, not supported in MySQL |
| TEXT | Variable-length storage with maximum size of 2GB data |

Note that all the above data types are for character stream, they should not be used with Unicode data.

### SQL Unicode Character and String Data Types

|  |  |
| --- | --- |
| Datatype | Description |
| NCHAR | Fixed length with maximum length of 4,000 characters |
| NVARCHAR | Variable-length storage with a maximum length of 4,000 characters |
| NVARCHAR(max) | Variable-length storage with provided max characters |
| NTEXT | Variable-length storage with a maximum size of 1GB data |

Varchar vs nvarchar

The word ‘Neptune’ has ****7 characters****. If we use VARCHAR as the data type, SQL Server will use ****1 byte per character**** to store this value in memory. Therefore (if my math is correct), the word ‘Neptune’ will use ****7 bytes of memory****.

If we used NVARCHAR as  the data type instead, SQL Server will use ****2 bytes per character**** to store this value in memory. Therefore (using *advanced* mathematics), the word ‘Neptune’ would use ****14 bytes of memory****.

But why does NVARCHAR use 2 bytes per character? Because NVARCHAR is designed to store a character string value in *multiple languages*. It needs that extra space to store your value in several languages.

VARCHAR, on the other hand, only stores your character string in English and one other language.

[Heaps (Tables without Clustered Indexes) - SQL Server | Microsoft Docs](https://docs.microsoft.com/en-us/sql/relational-databases/indexes/heaps-tables-without-clustered-indexes?view=sql-server-ver16)

https://docs.microsoft.com/en-us/sql/relational-databases/indexes/heaps-tables-without-clustered-indexes?view=sql-server-ver16

Note that above data types are not supported in MySQL database.

### SQL Binary Data Types

|  |  |
| --- | --- |
| Datatype | Description |
| BINARY | Fixed length with a maximum length of 8,000 bytes |
| VARBINARY | Variable-length storage with a maximum length of 8,000 bytes |
| VARBINARY(max) | Variable-length storage with provided max bytes |
| IMAGE | Variable-length storage with maximum size of 2GB binary data |

### SQL Miscellaneous Data Types

|  |  |
| --- | --- |
| Datatype | Description |
| CLOB | Character large objects that can hold up to 2GB |
| BLOB | For binary large objects |
| XML | for storing XML data |
| JSON | for storing JSON data |

That’s all for a quick roundup on SQL data types.

5: Identity Key /identity key creation in sql (in mysql we know as Auto\_Increment )

CREATE TABLE Persons (  
    Personid int IDENTITY(1,1) PRIMARY KEY,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int  
);

Identity

Identity(1,1)=1,2,3,4,5

Identity(1,2)=1,3,5,7,9

Identity(1,3)=1,4,7,10,13

Identity(1,4)=1,5,9,13,17

Identity(2,2)=2,4,6,8,10,12

Identity(3,3)=3,6,9,12,15,18

Identity(4,4)=4,8,12,16,20,24

Identity(10,1)=10,11,12,13,14,15

Identity(10,2)=10,12,14,16,18,20

Identity(10,3)=10,13,16,19,22,25

Identity(10,4)=10,14,18,22,26,30

Identity(10,5)=10,15,20,25,30,35

Definition :

The MS SQL Server uses the IDENTITY keyword to perform an auto-increment feature.

In the example above, the starting value for IDENTITY is 1, and it will increment by 1 for each new record.

****Tip:**** To specify thTo insert a new record into the "Persons" table, we will NOT have to specify a value for the "Personid" column (a unique value will be added automatically):

at the "Personid" column should start at value 10 and increment by 5, change it to IDENTITY(10,5).

## 6:

## ALTER TABLE Statement

The ALTER TABLE statement is used to add, delete, or modify columns in an existing table. The ALTER TABLE statement is also used to add and drop various constraints on an existing table.

ALTER TABLE table\_name  
ADD column\_name datatype;

**The following SQL adds an "EmpName" column to the "Company" table:**

**Example**

ALTER TABLE Company  
ADD EmpName nvarchar(50);

#making column as not null in existing table

alter table login

alter column loginid int not null

alter table login add constraint PK\_login\_loginId PRIMARY KEY CLUSTERED (loginid)

We are not right person to perform the below statement related works:

alter,truncate,delete,drop,backup

To perform the execute the above commands , we need to do discussion about the new changes and new requirement with DBA person and before performing any of the commands need to take a backup of the column if possible.

Keys / SQL Constraints/Primary Key /Foreign Key/Unique Key

7: Insert data into table

insert into demo\_mst(Name,Email,state,countrycode)

values

('kuhu','kuhu@1325gmail.com','Bhoppal',462022);

----------------------------------------------------------------------------------------------------------

create table concern\_mst1(id int, name char(10),Address nvarchar(80),email nvarchar(20),

phone varchar(10),countrycode nvarchar(8));

select \* from concern\_mst1

insert into concern\_mst1(name,Address,email,phone,countrycode)

values

('booboo','wardno13','kumar05@gmail.com',9035449980,110045);

/\*

alter table concern\_mst

add countrycode nvarchar(8)

\*/

8: order by column name ASC

order by column name DESC

9: where clause

10: datetime & between

11: having

12: group by

13: Alias

14:joins

15:view

16:store procedure

17:function

Download database from online and create our own database for PowerBI & Tableau

Query under query called subquery

create table demo\_mst(id int,Name varchar(50),Email nvarchar(40),

state varchar(30),countrycode nvarchar(45));

select \* from demo\_mst