**SQL (Structured Query Language)**

**Data**: Data is a collection of raw bytes, unorganized facts that need to be processed.

* It is not sensible.
* It does not have any meaning.
* Data is not sufficient for decision making.
* Data does not depend on information.
* **Example:** 202001, 201301, (number, symbol etc.)

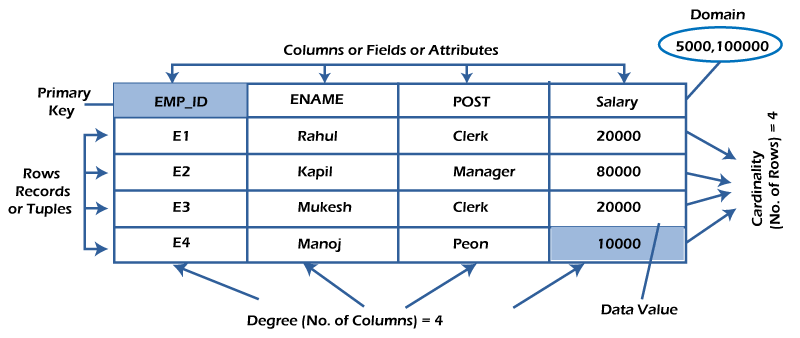
**Information**: Information is processed, organized and structured data.

* It is sensible
* It has meaning.
* Information is sufficient for decision making.
* Information depends on data.
* **Example:** 201009 is pin code of Ghaziabad

**Database:** Database is **a organized form of inter-related data** so that it can be easily accessed, managed and updated.

**Organized** means tabular form. **Related data** means information which stores in the form of Tables.

**Tables** contain Rows and Columns, Rows referred to as **Records or Tuples** and Columns referred to as **fields or attributes**.



**Entity:** An entity is a person, place, object which can be distinct identified. Each entity has its own attributes.

**Bit:** A bit is the smallest unit of data (value of a bit may be a 0 or 1). 8 bits make 1 byte which can represent a character.

**Field:** - A field represents an attribute of entity (object, person, place, or event).

**Record:** - A record represents a collection of attributes that describe a real-world entity.

**Purpose of Database: The** **Main purpose** of the database is to **operate a large amount of information** by storing, retrieving, and managing data.

**DBMS (Database Management System):** Database management system is **system software** which manages database. DBMS provides an interface to perform various operations on records. Like: creation, storing, updating, access.

* Stores data as files or hierarchical structures.
* Does not support relationships between data.
* **Example:** File Systems, XML, NoSQL

**RDBMS**

RDBMS stands for **Relational Database Management System**. RDBMS stores data in the form of **logical units called tables**. **Tables are inter-related** **to one another** and relations.

Table contains Rows and Columns, Rows referred to as **Records or Tuples** and Columns referred to as **fields or attributes**.

* Stores data in tables with rows and columns (relations).
* Supports relationships through foreign keys and primary keys
* **Example: Microsoft SQL Server, Oracle, MySQL,** PostgreSQL**.**

**Database Instance:** **Collection of information** stored at a particular moment.

**Data Modeling:** **Database creation** process is called data modeling.

**Schema:** The **overall description of a database** is called schema. It **represents the logical view** of the entire database. A schema contains objects like table, rows, columns, primary key, foreign key, views, data types, stored procedure, etc.

Example: **Student**

Student\_ID | Student\_Name | Student\_RollNo. | Student\_Age | Student\_Class | Student\_Address

**Employee**

Emp\_ ID | Emp\_Name | Emp\_Age | Department\_ID | Emp\_EMAIL | Emp\_Mob\_No. | Emp\_Address

**SQL**

**SQL stands for Structured Query Language** is a **query language** used to **perform operations on the records** stored in the database, such as inserting, deleting, updating records, and modifying database etc. It is pronounced as S-Q-L or sometimes as See-Quell.

**Microsoft SQL Server:**

SQL Server is a **Relational Database Management System Software** developed by Microsoft and used to perform queries. In SQL Server, Data Stores in the form of tables.

Table contains Rows and Columns, Rows referred to as **Records or Tuples** and Columns referred to as **fields or attributes**.

**DB Extension:**

The .DB [file extension](https://www.lifewire.com/what-is-a-file-extension-2625879) is often used by a program to indicate that the [file](https://www.lifewire.com/what-is-a-file-2625878) is storing information in some kind of structured database format.

.mdf = master database file

.ldf = log database file

**SQL Statements:** There are 4 types of SQL statements/query types:

**DDL (Data Definition Language):** Deals with the **structure** of database objects**.**

* CREATE → Creates a new table, database, index, or view.
* ALTER → Modify an existing table structure (add/drop/modify columns).
* DROP → Delete a table, database, or other objects permanently.
* TRUNCATE → Removes all records from a table but **keeps the structure**.
* RENAME → Rename an existing database object (e.g., table).

**DML (Data Manipulation Language)**: Used to **manipulate** data in tables.

* INSERT → Adds new records into a table.
* UPDATE → Modifies existing records in a table.
* DELETE → Removes specific records from a table.
* (MERGE → Combines INSERT, UPDATE, and DELETE based on conditions.)

**DQL(Data Query Language)**: Used to **retrieve** data from the database.

* SELECT → Fetches data from tables based on conditions.

**DCL (Data Control Language):** Used to **control access** and permissions in the database.

* GRANT → Assigns privileges (e.g., GRANT SELECT ON Employees TO user1).
* REVOKE → Removes granted privileges.

**SQL Server Data Types:**

A data type ensures which type of data can store in SQL Server.

**Why use data types?**

We can understand the importance of data type in [SQL Server](https://www.javatpoint.com/sql-server-tutorial) by taking a simple **Sign-up page** to create a new google account (only for reference purposes). This page shows the following input fields: **First Name, Last Name, User Name,** and **Password**.

These fields require the following data type characteristics:

* **First name and Last name:** These fields should require only alphabets.
* **Username:** This field can accept letters, numbers, and periods.
* **Password:** This field must contain alphabets, numbers, and special characters.
* **Contact Number:** This field must be numeric.
* **Pin Code:** It can only numeric fields.

**Numeric Data Type:** used to store numeric values.

bit (0,1)

tinyint (0 to 255)

smallint (-32768 to 32767)

int (-2,147,483,648 to 2,147,483,647 )

bigint (-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807)

Float (7 digits)

Real (15 digits)

Decimal (18, 2)

### String data type

char (0 to 8000 characters)

varchar (0 to 8000 characters)

varchar(max) (1,073,741,824 characters)

nchar (4000 Characters)

nvarchar(max) (4000 Characters)

text (2 GB)

ntext (2 GB)

binary (8000 bytes)

varbinary (8000 bytes)

image (2 GB)

**========================================================================================**

**++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

DataBase create database

Table 1 Table 2 Table 3

Creates tabel

Columns (Columns/fields: represent the attributes)

Rows (rows/records: Collection of attributes) insert record

Example: Database Name: Employee\_Database

Table Name: Employee

Attributes (Columns/fields):Emp\_id, Emp\_Name, Emp\_Salary, Emp\_Job, Emp\_Phone, Emp\_dept etc

Record (Row/Tuple): 1, Anurag, 40000, Java\_developer, 9056905684, IT

**SQL Constraints**: Constraints are the **rules** that decide what kind of data enter into the database tables. This ensures the accuracy and reliability of data.

* [NOT NULL](https://www.w3schools.com/sql/sql_notnull.asp)  - Ensures that a column cannot have a NULL value.
* [UNIQUE](https://www.w3schools.com/sql/sql_unique.asp)  - Ensures that all values in a column are different, values must be unique. Value can be null
* [PRIMARY KEY](https://www.w3schools.com/sql/sql_primarykey.asp)  - A Primary key is a combination of a NOT NULL and UNIQUE, used to uniquely identify value each row in a table.

Primary key is also called independent key, regular key.

* [FOREIGN KEY](https://www.w3schools.com/sql/sql_foreignkey.asp) - A Primary key is used to uniquely identify each row in another table.
* [CHECK](https://www.w3schools.com/sql/sql_check.asp)  - Ensures that the value in a column satisfies a specific condition.
* [DEFAULT](https://www.w3schools.com/sql/sql_default.asp)  - Set a default value for a column if no value is specified.

Column\_name date default getdate()

Exam: Tdate date default getdate().

Create database Employee

use Employee

create table Emp\_table

(

Emp\_Id int primary key identity(1,m1),

Firstname nvarchar(30) not null,

Lastname nvarchar(30) not null,

age tinyint check(age>=18 and age<=99),

Email nvarchar(30) not null unique,

Password nvarchar(30) not null,

Mobile bigint not null unique,

Salary decimal(18,2),

Job\_Title nvarchar(30) not null,

city nvarchar(30)

)

insert into Emp\_table

values('Rajesh','Sharma',20,'rajesh123@gmail.com','rajesh987',9870867564,25000,'dotnet developer', 'noida')

select \*from Emp\_table

Example:

**Condition 1: If database is exists.**

sp\_helpdb --used to return all the list of the existing database.

use <database\_name> --used to work on particular database

sp\_helpdb <database\_name> --used to check information about the specified database

Select \* from sys.tables --used to show all the tables in the particular database.

select \* from <table\_name> --used to view the structure/retrieve the data of database.

sp\_help <table\_name> --used to check information about table specified database.

**Condition 2: If database does not exists. Create a new database.**

sp\_helpdb --used to returns all the list of the existing database.

Create Database <database\_name> --used to create new database. (128 char max)

use <database\_name> --used to work on database

sp\_helpdb <database\_name> --used to check information about the specified database

select \*from sys.tables --used to show all the tables in the particular database.

create table <table\_name> --used to create new table in database

**1: DDL(Data Definition Language):** Deals with the **structure** of database objects**.**

(Create, Rename, Alter (add, drop), Drop, Truncate)

Example:

Create database Employee

use Employee

create table Emp\_table

(

Emp\_Id int primary key identity(1,1),

Firstname nvarchar(30) not null,

Lastname nvarchar(30) not null,

age tinyint check(age>=18 and age<=99),

Email nvarchar(30) not null unique,

Password nvarchar(30) not null,

Mobile bigint not null unique,

Salary decimal(18,2),

Job\_Title nvarchar(30) not null,

city nvarchar(30)

)

insert into Emp\_table

values('Sudarshan','Sharma',20,'rajesh123@gmail.com','rajesh987',9870867564,25000,'Dontet Developer', 'Noida')

select \*from Emp\_table

**Rename the Database name**

sp\_renamedb '<old db name>','<new db name>' --rename the database name

**Rename the table name**

sp\_rename ‘<old table name>’,’<New table name>’ --rename table name

sp\_rename 'Product', 'Products' --rename table name

**Add column into table**

alter table <table\_name> add <colum\_name> datatype constrains --add column table

alter table Products add mobile bigint not null --add column table

**Drop column from table**

alter table <table\_name> drop column <column\_name> --drop the column

alter table Products drop column mobile --drop the column

**Rename the column name**

sp\_rename <table\_name.old column\_name>,<new column\_name> --change column name

sp\_rename 'Products.Product\_Name' , 'item\_Name' --change column name

**Change the data type of attribute**

alter table <table\_name> alter column <column\_name> datatype

alter table Products alter column Product\_Name varchar(150) --change the data type of attribute

**2: DML (Data Manipulation Language):** Used to **manipulate** data in tables.

(insert, update, delete)

**Insert**

insert into <table\_name> values (value1,value2,value3.....)

or

insert into <table\_name>(column1,column2,column3....)

values (value1,value2,value3.....)

insert into Emp\_table

values('Rajesh','Sharma',20,'rajesh123@gmail.com','rajesh987',9870867564,25000,'PHP Developer','Noida')

****

**Identity:** It is used to auto-generated incremental value.

create table Employee

(

Emp\_id int, primary key identity(1,1)

name varchar(100)

)

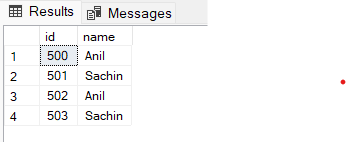
or

create table Employee

(

Emp\_id int, primary key, identity(500,1),-- by default 1, start from 500

name varchar(100)



**Nvarchar (National Variable Character):** It is a **Unicode string data type** that supports multiple language and special characters.

create table Employee

(

Emp\_id int, primary key identity(1,1)

Fname nvarchar(100),

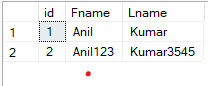
Lname nvarchar(100),

)

insert into Employee values ('Anil', 'Kumar')

insert into Employee values ('Anil123', 'Kumar3545')

select \*from Employee



Update: It is used to update the table.

Update <table\_name>

set <column\_name>= value...>

where condition --use of update statement

update Employee

set Mobile\_no=9056784589

where Emp\_id=4 --use of update statement

**Delete:** It is a DML command used to delete one or more records. Memory will remain.

delete from <table\_name> where condition

delete from Employee where Emp\_id=2

**Truncate:** It is a DDL command used to delete all the records. Memory will delete but structure will remain.

truncate table <table\_name> --it will delete all the rows of table.

**Drop:** It is a DDL command used to delete whole table and database.

drop database <database\_name> --it will delete whole database.

drop table <table\_name> --it will delete table.

**3: DQL (Data Query Language):** Used to **retrieve data** in tables.

**Select:** Select command is a DQL command used to retrieve the data of tables.

select \* from <table\_name> --used to view the structure/retrieve the data of the table of database.

select \* from Employee

To select fields from the table

Select <column\_name> from <table\_name>

To select distinct data from the table

select distinct <column\_name> from <table\_name>

**Operator**

**1: Where Clause:** Where clause is a keyword used to filter records.

update <table\_name> set <column\_name = value...> where condition

**2: Comparison/Conditional Operators (<, >, <=, >=, =, !=)**

select\*from product where Addresss='Benglore' --used to show record

select\*from product where id>4 --greater than

select\*from product where id<=4 --less than equal to

select\*from product where id>=4 --greater than equal to

select\*from product where id!=2 --not equal to

**3: Logical Operators:**

**3.1: AND, OR, NOT**

Select \* from <table\_name> where condition1 AND condition2

Select \* from <table\_name> where condition1 OR condition2

Select \* from <table\_name> where condition1 NOT condition2

select\*from product where id=2 and Addresss='Mumbai' --and operator used where both condition is true

select\*from product where id=2 or Addresss='Ghaziabad' --or operator used where one condition is true

**3.2: IN**

Select \* from <table\_name> where <column\_name> IN (value1, value2…) --in (specify multiple value)

select\*from product where id in (34,6,7,90,34,5,6,5,)

**3.3: Between & Not Between**

Select \*from <table\_name> where <column\_name> BETWEEN value1 AND Value2

Select \*from <table\_name> where <column\_name> NOT BETWEEN value1 AND Value2

select\*from product where id between 1 and 5 --between

select\*from product where id not between 1 and 5 --not between

**4: Order by clause (ascending/descending):** It is used to sort the result-set in ascending or descending order.

select\*from <table\_name> order by <column\_name> asc --ascending

select\*from <table\_name> order by <column\_name> desc --descending

select <column\_name> from <table\_name> order by <column\_name> asc/desc

**NULL, NOT NULL**

select \* from <table\_name> where <column\_name> IS NULL

select \* from <table\_name> where <column\_name> IS NOT NULL

**5: Aggregate function (max,min,count,sum,avg)🡪** works on Integer value.

select\*from product

select max(<column\_name>)from <table\_name>

select min(<column\_name>)from <table\_name>

select count(\*) from <table\_name>

select sum(<column\_name>) from <table\_name>

select avg(<column\_name>) from <table\_name>

example:

select max(salary) as Maximum\_Salary from product

select min(salary) as Mimimum\_Salary from product

select count(salary)from product

select sum(salary) as Total\_Salary from product

select avg(salary) as Average\_Salary from product

**6: Wildcard(order by)**

select\*from product

select\*from product where Fullname='Anil Kumar Sharma'

select\*from product where Fullname like 'Anil kapoor'

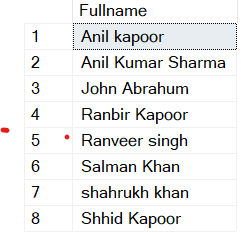
select\*from product where Fullname like 'R%' -- first word should be start form R letter.

select\*from product where Fullname like '%R%' -- word should include R letter. All data will display

select\*from product where Fullname like '[SP]%' -- word should start either S or P.

**7: Distinct:** It is used to retrieve unique data.

select Distinct(<column\_name>) from <table\_name> --show unique value



<https://www.youtube.com/watch?v=rfWYbMd3ApA&list=PLdOKnrf8EcP17p05q13WXbHO5Z_JfXNpw&index=7>

<https://www.youtube.com/watch?v=55_UN5gVARs&list=PLdOKnrf8EcP17p05q13WXbHO5Z_JfXNpw&index=8>

<https://www.youtube.com/watch?v=hlGoQC332VM> Shraddha