**SQL**

**Database:-**

A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a [database management system (DBMS)](https://www.oracle.com/in/database/what-is-database/#WhatIsDBMS)

**SQL:-**

sql stands for structured query language programming language storing and processing a information in relational database . The relational database information is tabular form contain rows and columns. We use sql statement to insert, delete and update datas in database

**Types of SQL Commands:**



### **1. Data Definition Language (DDL)**

* DDL changes the structure of the table like creating a table, deleting a table, altering a table, etc.
* All the command of DDL are auto-committed that means it permanently save all the changes in the database.

**Create:**

**Create** It is used to create a new table in the database.

mysql> create table document(id int,name varchar(20),course varchar(20),fees int,age int);

Query OK, 0 rows affected (0.32 sec)

mysql> desc document;

+--------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+--------+-------------+------+-----+---------+-------+

| id | int | YES | | NULL | |

| name | varchar(20) | YES | | NULL | |

| course | varchar(20) | YES | | NULL | |

| fees | int | YES | | NULL | |

| age | int | YES | | NULL | |

+--------+-------------+------+-----+---------+-------+

5 rows in set (0.20 sec)

**Alter:**

It is used to alter the structure of the database. This change could be either to modify the characteristics of an existing attribute or probably to add a new attribute.

mysql> alter table document add gender char(1);

Query OK, 0 rows affected (0.38 sec)

Records: 0 Duplicates: 0 Warnings: 0

mysql> desc document;

+--------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+--------+-------------+------+-----+---------+-------+

| id | int | YES | | NULL | |

| name | varchar(20) | YES | | NULL | |

| course | varchar(20) | YES | | NULL | |

| fees | int | YES | | NULL | |

| age | int | YES | | NULL | |

| gender | char(1) | YES | | NULL | |

+--------+-------------+------+-----+---------+-------+

6 rows in set (0.00 sec)

**Rename:**

It is used to rename the table name and data object.

mysql> rename table document to records;

Query OK, 0 rows affected (0.13 sec)

**Truncate:**

It is used to remove all records from a table, including all spaces allocated for the records are removed.

mysql> truncate table records;

Query OK, 0 rows affected (0.37 sec)

**Drop:**

 It is used to delete both the structure and record stored in the table.

mysql> drop table records;

Query OK, 0 rows affected (0.06 sec)

mysql> desc records;

ERROR 1146 (42S02): Table 'dbfirst.records' doesn't exist.

### **2. Data Manipulation Language**

* DML commands are used to modify the database. It is responsible for all form of changes in the database.
* The command of DML is not auto-committed that means it can't permanently save all the changes in the database. They can be rollback.

**Insert:**

The INSERT statement is a SQL query. It is used to insert data into the row of a table.

mysql> insert into document values(101,'venkad','software testing',2000,21);

Query OK, 1 row affected (0.15 sec)

mysql> insert into document values(102,'murali','software testing',18000,23);

Query OK, 1 row affected (0.08 sec)

mysql> insert into document values(103,'madhav','java',15000,22);

Query OK, 1 row affected (0.06 sec)

mysql> insert into document values(104,'abi','sap',5000,22);

Query OK, 1 row affected (0.10 sec)

mysql> insert into document values(105,'agathiyan','sql',20000,21);

Query OK, 1 row affected (0.09 sec)

**UPDATE:**

It is used to update or modify the value of a column in the table.

update document set age=22 where id=105;

Query OK, 1 row affected (0.08 sec)

Rows matched: 1 Changed: 1 Warnings: 0

**Delete:**

 It is used to remove one or more row from a table.

delete from document where id=105;

Query OK, 1 row affected (0.06 sec)

### **Data Query Language**

DQL is used to fetch the data from the database.

It uses only one command.

**Select:**

This is the same as the projection operation of relational algebra. It is used to select the attribute based on the condition described by WHERE clause.

select \* from document;

+------+--------+------------------+-------+------+

| id | name | course | fees | age |

+------+--------+------------------+-------+------+

| 101 | venkad | software testing | 2000 | 21 |

| 102 | murali | software testing | 18000 | 23 |

| 103 | madhav | java | 15000 | 22 |

| 104 | abi | sap | 5000 | 22 |

+------+--------+------------------+-------+------+

4 rows in set (0.00 sec)

### **Data Control Language**

DCL commands are used to grant and take back authority from any database user.

Here are some commands that come under DCL.

**Grant:**

It is used to give user access privileges to a database.

Grant create table to hr;

**Revoke:**

It is used to take back permissions from the user.

Revoke create table from hr;

### **Transaction Control Language**

TCL commands can only use with DML commands like INSERT, DELETE and UPDATE only.

**Commit:**

Commit command is used to save all the transactions to the database.

SQL> insert into in5 values(101,'venkad');

1 row created.

SQL> commit;

Commit complete.

**Savepoint:**

 It is used to roll the transaction back to a certain point without rolling back the entire transaction.

SQL> insert into in5 values(102,'murali');

1 row created.

SQL> savepoint a;

Savepoint created.

SQL> insert into in5 values(103,'madhav');

1 row created.

SQL> savepoint b;

Savepoint created.

SQL> insert into in5 values(104,'abi');

1 row created.

SQL> savepoint c;

Savepoint created.

select \* from in5;

ID NAME

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

101 venkad

102 murali

103 madhav

104 abi

**Rollback:**

rollbacks a transaction in case of any error occurs.

This command restores the database to last commited state. It is also used with SAVEPOINT command to jump to a savepoint in an ongoing transaction.

SQL> rollback to savepoint a;

Rollback complete.

SQL> select \* from in5;

ID NAME

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

101 venkad

102 murali

mysql> use dbfirst;

Database changed

mysql> create table in3(id int,name varchar(20),age int,salary int);

Query OK, 0 rows affected (0.39 sec)

mysql> insert into in3 values(100,'venki',21,20000);

Query OK, 1 row affected (0.14 sec)

mysql> insert into in3 values(101,'murali',23,18000);

Query OK, 1 row affected (0.05 sec)

mysql> insert into in3 values(102,'abi',22,23000);

Query OK, 1 row affected (0.05 sec)

mysql> insert into in3 values(103,'madhav',23,20000);

Query OK, 1 row affected (0.07 sec)

mysql> insert into in3 values(104,'agathiyan',21,21000);

Query OK, 1 row affected (0.05 sec)

mysql> select \* from in3;

+------+-----------+------+--------+

| id | name | age | salary |

+------+-----------+------+--------+

| 100 | venki | 21 | 20000 |

| 101 | murali | 23 | 18000 |

| 102 | abi | 22 | 23000 |

| 103 | madhav | 23 | 20000 |

| 104 | agathiyan | 21 | 21000 |

+------+-----------+------+--------+

5 rows in set (0.07 sec)

## Count:

The **COUNT**() function returns the number of rows that matches a specified criterion.

mysql> select count(age) from in3;

+------------+

| count(age) |

+------------+

| 5 |

+------------+

1 row in set (0.02 sec)

## Sum:

The **SUM**() function returns the total **sum** of a numeric column.

mysql> select sum(salary) from in3;

+-------------+

| sum(salary) |

+-------------+

| 102000 |

+-------------+

1 row in set (0.00 sec)

## Min:

The **MIN**() function returns the **smallest** value of the selected column.

mysql> select min(salary) from in3;

+-------------+

| min(salary) |

+-------------+

| 18000 |

+-------------+

1 row in set (0.04 sec)

## Avg:

The **AVG() function** returns the average value of a numeric column.

mysql> select avg(salary)from in3;

+-------------+

| avg(salary) |

+-------------+

| 20400.0000 |

+-------------+

1 row in set (0.00 sec)

## Max:

The MAX() function returns the largest value of the selected column.

mysql> select max(salary)from in3;

+-------------+

| max(salary) |

+-------------+

| 23000 |

+-------------+

1 row in set (0.00 sec)

## And:

The AND operator displays a record if all the conditions separated by AND are TRUE.

mysql> select \* from in3 where name='venki' and age=21;

+------+-------+------+--------+

| id | name | age | salary |

+------+-------+------+--------+

| 100 | venki | 21 | 20000 |

+------+-------+------+--------+

1 row in set (0.03 sec)

## Or:

The OR operator displays a record if any of the conditions separated by OR is TRUE.

mysql> select \* from in3 where name='murali' or salary=23000;

+------+--------+------+--------+

| id | name | age | salary |

+------+--------+------+--------+

| 101 | murali | 23 | 18000 |

| 102 | abi | 22 | 23000 |

+------+--------+------+--------+

2 rows in set (0.00 sec)

## Not:

The NOT operator displays a record if the condition(s) is NOT TRUE.

mysql> select \* from in3 where not name='abi';

+------+-----------+------+--------+

| id | name | age | salary |

+------+-----------+------+--------+

| 100 | venki | 21 | 20000 |

| 101 | murali | 23 | 18000 |

| 103 | madhav | 23 | 20000 |

| 104 | agathiyan | 21 | 21000 |

+------+-----------+------+--------+

4 rows in set (0.01 sec)

## Like:

The LIKE command is used in a WHERE clause to search for a specified pattern in a column.

You can use two wildcards with LIKE:

* % - Represents zero, one, or multiple characters
* \_ - Represents a single character (MS Access uses a question mark (?) instead)

mysql> select \* from in3 where name like 'v%';

+------+-------+------+--------+

| id | name | age | salary |

+------+-------+------+--------+

| 100 | venki | 21 | 20000 |

+------+-------+------+--------+

1 row in set (0.01 sec)

mysql> select salary from in3 where name like 'v\_nk\_';

+--------+

| salary |

+--------+

| 20000 |

+--------+

1 row in set (0.00 sec)

## Reverse:

The **REVERSE**() function reverses a string and returns the result.

mysql> select reverse(name) from in3;

+---------------+

| reverse(name) |

+---------------+

| iknev |

| ilarum |

| iba |

| vahdam |

| nayihtaga |

+---------------+

5 rows in set (0.03 sec)

## Upper():

The UPPER() function converts a string to **upper-case**.

mysql> select upper(name) from in3;

+-------------+

| upper(name) |

+-------------+

| VENKI |

| MURALI |

| ABI |

| MADHAV |

| AGATHIYAN |

+-------------+

5 rows in set (0.03 sec)

## Lower():

## The LOWER() function converts a string to lower-case.

mysql> select lower(name)from in3;

+-------------+

| lower(name) |

+-------------+

| venki |

| murali |

| abi |

| madhav |

| agathiyan |

+-------------+

5 rows in set (0.02 sec)

## Initcap():

## INITCAP returns char , with the first letter of each word in uppercase, all other letters in lowercase.

mysql> select initcap(name)from in3;

+-------------+

| initcap(name) |

+-------------+

| Venki |

| Murali |

| Abi |

| Madhav |

| Agathiyan |

+-------------+

## Length:

## The LEN() function returns the length of a string.

mysql> select length(name) from in3;

+--------------+

| length(name) |

+--------------+

| 5 |

| 6 |

| 3 |

| 6 |

| 9 |

+--------------+

5 rows in set (0.01 sec)

mysql> select name,char\_length(name) as lon from in3;

+-----------+------+

| name | lon |

+-----------+------+

| venki | 5 |

| murali | 6 |

| abi | 3 |

| madhav | 6 |

| agathiyan | 9 |

+-----------+------+

5 rows in set (0.03 sec)

## Concat:

## The CONCAT() function adds two or more strings together.

mysql> select concat(name,salary) from in3;

+---------------------+

| concat(name,salary) |

+---------------------+

| venki20000 |

| murali18000 |

| abi23000 |

| madhav20000 |

| agathiyan21000 |

+---------------------+

5 rows in set (0.00 sec)

mysql> select concat(name," ",age," ",salary)as con from in3;

+--------------------+

| con |

+--------------------+

| venki 21 20000 |

| murali 23 18000 |

| abi 22 23000 |

| madhav 23 20000 |

| agathiyan 21 21000 |

+--------------------+

5 rows in set (0.00 sec)

## JOINS

mysql> select\*from A;

+------+----------+--------+------+-------------+

| id   | name     | amount | age  | phonenumber |

+------+----------+--------+------+-------------+

|  112 | ruthra   |  20000 |   23 |         987 |

|  113 | santhya  |  15000 |   21 |        7654 |

|  116 | akash    |  34000 |   25 |        8765 |

|  117 | sathish  |  20000 |   27 |        8978 |

|  119 | suren    |  30000 |   24 |        9867 |

|  210 | narmatha |  23000 |   22 |        9768 |

+------+----------+--------+------+-------------+

6 rows in set (0.00 sec)

mysql> SELECT \* FROM B;

+------+---------+--------------+

| id   | course  | address      |

+------+---------+--------------+

|  117 | phython | saligramam   |

|  210 | ccna    | nungambakkam |

|  113 |  sql    | nandhanam    |

|  220 | java    | annanagar    |

+------+---------+--------------+

4 rows in set (0.00 sec)

## INNER JOIN

sql> select A.name,A.amount,B.course from A inner join B on A.id = B.id;

+----------+--------+---------+

| name     | amount | course  |

+----------+--------+---------+

| santhya  |  15000 |  sql    |

| sathish  |  20000 | phython |

| narmatha |  23000 | ccna    |

+----------+--------+---------+

3 rows in set (0.00 sec)

## LEFT OUTER JOIN

mysql> select A.name,A.amount,B.course from A left outer join B on A.id = B.id;

+----------+--------+---------+

| name     | amount | course  |

+----------+--------+---------+

| ruthra   |  20000 | NULL    |

| santhya  |  15000 |  sql    |

| akash    |  34000 | NULL    |

| sathish  |  20000 | phython |

| suren    |  30000 | NULL    |

| narmatha |  23000 | ccna    |

+----------+--------+---------+

6 rows in set (0.00 sec)

## RIGHT OUTER JOIN

mysql> select A.name,A.amount,B.course from A right outer join B on A.id = B.id;

+----------+--------+---------+

| name     | amount | course  |

+----------+--------+---------+

| sathish  |  20000 | phython |

| narmatha |  23000 | ccna    |

| santhya  |  15000 |  sql    |

| NULL     |   NULL | java    |

+----------+--------+---------+

4 rows in set (0.00 sec)

mysql> select A.name,A.amount,B.course from A full outer join B on A.id = B.id;

ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'outer join B on A.id = B.id' at line 1