

# MySQL

→ Founded by Michael Widenius, David Axmark and Allan Larsson in Sweden.

## \* Advantages

- 1) It allows us to implement database operations on tables, rows, columns and indexes.
- 2) It defines the database relationship in the form of tables, also known as relations.
- 3) It provides the Referential Integrity between rows or columns of various tables.
- 4) It allows us to update the table indexes automatically.
- 5) It is a open-source database.
- 6) It supports many operating systems with many languages like PHP, PERL, C, C++, Java, etc.

## → Features of MySQL

- 1> RDBMS .
- 2> Easy to use .
- 3> It is secure .
- 4> Client / Server Architecture .
- 5> Free to download .
- 6> It is Scalable .
- 7> Speed .
- 8> High Flexibility .
- 9> High performance .
- 10> Platform Independent .
- 11> Memory efficiency .
- 12> Compatible with many operating systems .
- 13> GUI Support .

## → DDL Commands:

1> CREATE → Used to create a table or database.

i)> CREATE DATABASE DB\_NAME ;

ii)> CREATE TABLE TB-NAME ( column1 datatype1,  
column2 datatype2,  
column3 datatype3,  
);

2> ALTER

- to add a column to existing table.
- to rename any existing column
- to change datatype of any column or size
- to drop a column.

i)> ALTER TABLE TB-NAME ADD ( column-name datatype)  
You can add multiple columns too.

ii)> ALTER TABLE TB-NAME MODIFY (column1 datatype);

iii)> ALTER TABLE TB-NAME RENAME (old\_column\_name  
To new\_column\_name)

iv)> ALTER TABLE TB-NAME DROP (column\_name);

### 3> TRUNCATE

- Removes all the records from a table  
But the table structure is not destroyed.

TRUNCATE TABLE TB\_NAME ;

### 4> DROP

or database

- Completely removes a table from the database. It destroys the table structure too.

DROP TABLE TABLE-NAME ;

DROP DATABASE DB-NAME ;

### 5> RENAME

- Used to set a new name for an existing table.

RENAME TABLE OLD-NAME TO NEW-NAME ;

## → DML Commands

### 1> INSERT

to Insert data into a table

i> `INSERT INTO TB-NAME VALUES (data1, data2...);  
INSERT INTO STUDENT VALUES (101, 'Adam', 15);`

### 2> UPDATE

UPDATE Command is used to update any record of data in a table

`UPDATE table-name SET column-name = new-value WHERE  
condition;`

`UPDATE student SET age = 18 WHERE student_id = 102;`

`UPDATE student SET age = age + 1;`

### 3> DELETE

Used to delete data from a table

`DELETE FROM TABLE_NAME;`

`DELETE FROM Student;`

`DELETE FROM Student WHERE S-id = 103;`

## TCL Commands

### 1> COMMIT

Commit Command is used to permanently save any transaction into the database.

When we use any DML command like INSERT, UPDATE or DELETE, the changes made by these commands are not permanent, until the current session is closed, the changes made by these commands can be rolled back.

To avoid that, we use the COMMIT command to mark the changes as permanent.

COMMIT;

### 2> ROLLBACK

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

If we have used the UPDATE command to make some changes into the database, and realise that these changes were not required, then we can use the

ROLLBACK command to rollback those changes, if they were not committed using the COMMIT command.

ROLLBACK TO Savepoint-name ;

### 3) SAVEPOINT Command

SAVEPOINT command is used to temporarily save a transaction so that you can rollback to that point whenever required.

SAVEPOINT Savepoint-name ;

## → DCL Commands

### i) GRANT

Used to provide any user privileges or other privileges for the database

i) GRANT CREATE SESSION TO username;

ii) GRANT CREATE TABLE TO username;

iii) GRANT sysdba To username; (sysdba grants all permission)

2>

## REVOKE

Used to take back permissions from any user.

REVOKE CREATE TABLE from username;



## WHERE Clause :-

Where clause is used to specify/apply any condition while retrieving, updating or deleting data from a table.

DELETE FROM table-name WHERE [condition]



## LIKE Clause :-

Like clause is used in the condition in SQL query either with the WHERE clause.

## Wildcard operators

There are two wildcard operators that are used in LIKE clause.

1> Percent sign-% → represent zero, one or more than one character

2> Underscore sign-\_ → represents only a single character.

SELECT \* FROM STUDENT WHERE S-name LIKE 'A%';

→ ORDER BY Clause :-

Order by clause is used with select statement for arranging retrieved data in sorted order.

The Order by clause by default sorts the retrieved data in Ascending order.

DESC keyword is used to sort the data in descending order.

SELECT column list | \* FROM tablename ORDER BY ASC / DESC;

~~SAE~~ SELECT \* FROM Emp ORDER BY SALARY;

→ GROUP By clause :-

Group By clause is used to group the results of a SELECT query based on one or more columns.

SELECT column-name, function(column-name) FROM table-name WHERE condition GROUP BY column-name;

SELECT name, age FROM Emp Group By Salary;  
SELECT name, salary FROM Emp WHERE age > 25  
GROUP BY ~~salary~~ salary;

→ HAVING Clause :-

HAVING Clause is used with SQL queries to give more precise condition for a statement.

SELECT column-name , function (column-name)  
FROM table-name  
WHERE column-name condition  
GROUP BY column-name  
HAVING function (column-name) condition .

SELECT \* FROM sale GROUP BY customer  
HAVING sum (previous-balance) > 3000 .

→ DISTINCT keyword :-

SELECT DISTINCT column-name FROM table-name;

SELECT DISTINCT Salary FROM Emp;

The distinct keyword is used with SELECT statement to retrieve unique values from the table .

Distinct removes all the duplicate records while retrieving records from any table

## → AND operator

It is used to set multiple conditions with the WHERE clause , alongside SELECT, UPDATE or DELETE queries .

```
SELECT * FROM Emp WHERE salary < 10000 AND  
age > 25;
```

## → OR operator

It is used to combine multiple conditions with WHERE Clause .

Atleast one condition from the conditions specified must be satisfied by any record .

```
SELECT * FROM Emp WHERE WHERE salary > 10000 OR  
age > 25;
```

## → SQL Data Constraints

- NOT NULL
- PRIMARY KEY
- FOREIGN KEY
- UNIQUE
- CHECK

## → Aggregate Functions :-

- 1> AVG()
- 2> COUNT()
- 3> FIRST()
- 4> LAST()
- 5> MAX()
- 6> MIN()
- 7> SUM()

## → SCALAR Functions :-

- 1> UCASE() - Uppercase
- 2> LCASE() - Lowercase
- 3> MID()
  - Used to extract substring from column values of string type in a table

SELECT MID(column-name, start, length) FROM  
TABLE-NAME;

SELECT MID(name, 2, 2) FROM emp;

4> ROUND() - Rounding up a numeric field.

→ SET operations

1> UNION

2> UNION ALL (Shows duplicate rows too)

3> INTERSECT.

4> MINUS.

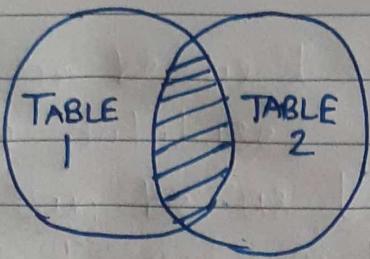
→ JOINS

Join in SQL are commands which are used to combine rows from tables (multiple) based on a related column between those tables.

They are predominantly used when a user is trying to extract data from tables which have one-to-many or many-to-many relationships between them.

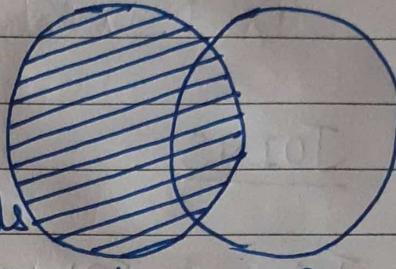
1> INNER JOIN

This type of join returns those records which have matching values in both tables. So, if you perform an INNER JOIN operation between two tables, all the tuples which have matching values in the both tables will be given as output.

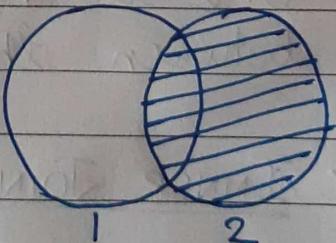


## 2) LEFT JOIN

The LEFT JOIN returns all the records from the left table and also those records which satisfy a condition from the right table. Also, for the records having no matching values in the right table, the output or the result-set will contain the NULL values.



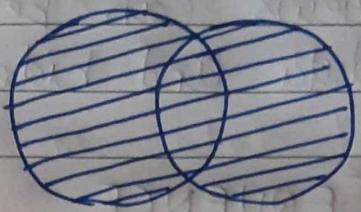
## 3) RIGHT JOIN



The Right join returns all the records from the right table and also those records which satisfy a condition from the left table. Also, for the records having no matching values in the left table ,the output or the result-set will contain the NULL values.

## 4) FULL JOIN

Full join returns all those records which either have a match in the left or the right table.



## → VIEWS

A view is nothing more than a saved SQL query. It can be considered as a virtual table.

i) Create View View-name

as

Select Id, Name Query.

Eg Create View View1 as

Select Id, Name, Salary, Gender, Dept.Name  
from tb1 join tb2 on tb1.DeptId = tb2.DeptId  
on tb1.Employee.

ii) To modify a view - ALTER VIEW statement.

ALTER VIEW View-name as Select Query

iii) To drop a View - Drop statement

DROP VIEW View-name

- Views can be used to reduce the complexity of the database Schema.
- Views can be used as a mechanism to implement row and column level security.
- Views can be used to present aggregated data and hide detailed data.

→ SEQUENCE

Sequence is a feature supported by some database systems to produce unique values on demand

```
CREATE SEQUENCE sequence-name
  START WITH initial-value
  INCREMENT BY increment-value
  MAXVALUE max-value
  CYCLE | NO CYCLE;
```

e.g. CREATE SEQUENCE seq1
 START WITH 1
 INCREMENT BY 1
 MAXVALUE 999
 CYCLE;

Using a Sequence :-

```
INSERT INTO class VALUE (seq1.nextval,<data>);
```

## MongoDB

- The keyword Cycle Specifies that if the maximum value exceeds the set limit, sequence will restart its cycle from the beginning.
- The keyword No Cycle Specifies that if sequence exceeds MAXVALUE value, an error will be thrown.