**MYSQL**

**Database :** It is an application which store the collection of data.

* Each database has one or more distinct API’s for creating, managing , searching and replicating the data.

**File :** A file is a collection of data that is stored on a computer, smartphone, or server. Files can contain many types of data, including:

* Text
* Images
* Audio
* Video

**Excel :** By using this there is a security and storage issues.

**DBMS :** Database management system.

* Store the data in the form of tables.

**Challenges :**

* Relations is not possible for access data from outer side.

**RDBMS :** Relational database management system.

* We can store the data in the form of tables and can also map them from locations.
* It will retrieve data very quickly.
* Operations will be very effective.

**Databases are 2 types**

1. RDBMS
2. NON RDBMS

**Non RDBMS:** Store data in the form of key value pairs like json format.

**Database components :-** 2 types

1. Client
2. Server

**CLIENT** : A client initiates requests to a server and then waits for a response. It could be a browser requesting a website, an email program checking for new messages, or a software application querying a database.

**SERVER** : A server listens for requests from clients, processes those requests, and sends responses back to the clients.

**Types of Commands :**

1. **DDL**
2. **DML**

**DDL :** DDL commands are used to define or modify the structure of a database, table, or other database objects.

1. CREATE
2. ALTER
3. DROP
4. TRUNCATE
5. RENAME

**DML :** DML commands are used to manipulate data within tables.

1. INSERT
2. UPDATE
3. DELETE
4. RETRIVE

**Data types using in MYSQL :**

1. CHAR – Fixed length of string characters are allowed.
2. VARCHAR – A variable string length.
3. BINARY - A fixed-length binary string. The storage size is fixed and specified during creation.
4. TEXT – A variable-length string with a maximum length of 65,535 characters. It is used to store longer text data.
5. TINTEXT - A shorter version of TEXT, with a maximum length of 255 characters.

**For create Database :**

Create database mydatabase;

**For creating table query :**

create table department(branch\_name varchar(20), id int(20), sub varchar(30), subject\_faculty varchar(30));

**For insert data into the table :**

insert into department values('civil',1,'construction','govardhan'),('cse',2,'datascience','ramana'),

('mec',3,'thermodynamics','sunil'),('eee',4,'eletrical','veerappan'),('ece',5,'circutes','vivek'),

('automobile',6,'mobile','pawan'),('mathmetic',7,'statistics','mallikarjuna'),

('chemistry',8,'chemistry','rao'),('physics',9,'physics','narayanarao'),('alogorithams',10,'algorithems','chari');

**For update table :**

* UPDATE table\_name SET column1 = value1, column2 = value2, ... WHERE id = 2;

**For delete table :**

* DELETE FROM table\_name WHERE name = ‘chari’;

**For Empty the table :**

* TRUNCATE TABLE employees;

**For fetch records from the table :**

* Select \* from employees;

**FOR ADD COLUMN IN THE EXISTING DATABASE :**

* alter table EMPLOYE add column age int;

**clauses & operators :**

1. **where clauses :** This clauses is used for filter the records in the existing table.
2. **AND , OR, NOT :**

**AND :** If you Display the record need to satisfy all the conditions.

**CONDITION1 CONDITION-2 RESULT**

**T T T**

**T F F**

**F T F**

**F F F**

**OR :** IF any one condition true it will return result.

**CONDITION1 CONDITION-2 RESULT**

**T T T**

**T F T**

**F T T**

**F F F**

**NOT :** Display the records not in the particular data.

Sys : select \* from tablename where not condition;

1. **Orderby :** sort the records in ascending or descending order.

Sys: select \* from student

order by studen\_id desc;

1. **Insert into :** Used for inserting new record into the existing table.

Sys : insert into tem values(‘mani’,1);

1. **SELECT :** Fetch the records from the existing table.

**Sys:** select \* from tablename;

1. **Update :** Update data in the existing table.

**Sys :** update tablename set columnName = ‘hi’ where id = 2;

1. **DELETE :** Delete the records in the existing table.

**sys :**Delete from tablename where condition;

1. **Limit :** used to specify the number of records to return.

**Sys:** select column1 from tablename where condition limit number;

1. **Min & MAX :**

**MIN() :** Used to return the minimum value in the column;

**Sys :** select MIN(COLUMN) from tablename;

**MAX() :** Used to return the max value in the column.

**Sys:** select MAX(COLUMN) from tablename;

1. **LIKE OPERATOR :**

Used for search for specific pattern in a column;

**-> SELECT \***

**FROM employees**

**WHERE name LIKE '%John%';**

* This query returns all records from the employees table where the name column contains the substring John anywhere in the string.

**->SELECT \***

**FROM employees**

**WHERE name LIKE 'John%';**

* This will return all employees whose name starts with John

**->SELECT \***

**FROM employees**

**WHERE name LIKE '%John';**

* This will return all employees whose name ends with John.

**IN** : It will allow us to specify multiple values in where clause.

**Sys:** select \* from emp where state in (‘ap’,’bihar’);

1. BETWEEN : select the middle value from the range of value.

**SELECT \***

**FROM products**

**WHERE price BETWEEN 100 AND 500;**

* This query will return all products where the price is between 100 and 500 (inclusive).

1. **AVG(): I**t will return the average of all column values.

**Sys : select avg(column) from tablename;**

1. **SUM() :** It will give total sum of columns values.

**Sys :** select SUM() FROM TABLENAME;

1. **COUNT** : Count the all values in column.

**Sys**: select count(col) from tablename where condition.

**Joins**

**JOIN** is used to combine data from two or more tables based on a related column between them. Joins are used with select statement. And used to retrieve data from multiple tables from from same database.

* **INNER JOIN**
* **OUTER JOIN**
* **CROSS JOIN**
* **SELF JOIN**

**Inner join :** Returns only the rows where there is a match in both tables.

Sys : SELECT employees.name, departments.name

FROM employees

INNER JOIN departments

ON employees.dept\_id = departments.dept\_id;

**OUTER JOIN :** An Outer Join in MySQL is used to combine rows from two or more tables, even when there is no matching data in one of the tables. There are three types of outer joins: LEFT OUTER JOIN, RIGHT OUTER JOIN, and FULL OUTER JOIN**.**  These joins are subdivided into 3 types

* Left outer join
* RIGHT OUTER JOIN
* **FULL OUTER JOIN**

**LEFT OUTER JOIN** : Returns all the rows from the left table and the matching rows from the right table If there is no match, the result will still include all rows from the left table, with NULL values for columns from the right table.

**SELECT employees.name, departments.name**

**FROM employees**

**LEFT JOIN departments**

**ON employees.dept\_id = departments.dept\_id;**

**RIGHT OUTER JOIN** : Returns all the rows from the right table and the matching rows from the left table If there is no match, the result will still include all rows from the right table, with NULL values for columns from the left table**.**

**SELECT employees.name, departments.name**

**FROM employees**

**RIGHT JOIN departments**

**ON employees.dept\_id = departments.dept\_id;**

**FULL OUTER JOIN :** This join combines the results of both a LEFT JOIN and a RIGHT JOIN. It returns all rows from both tables, filling in NULL where there is no match. MySQL does not directly support FULL OUTER JOIN

**CROSS JOIN :** Returns all possible combinations of rows from both tables.

**SELECT employees.name, departments.name**

**FROM employees**

**CROSS JOIN departments;**

**SELF JOIN :** A self join is when a table is joined with itself. It's used to compare rows within the same table.

**SELECT e1.name AS Employee1, e2.name AS Employee2**

**FROM employees e1, employees e2**

**WHERE e1.dept\_id = e2.dept\_id AND e1.emp\_id != e2.emp\_id;**

**or**

**Select e1.name from Employe e1 join Employe e2**

**On e1.id = e2.id;**