**MYSQL**

**Database:** It is an application that stores the collection of data.

Each database has one or more distinct APIs.

**API**: It is used to create, manage, store, search, and replicate data.

**Files:** Unchanged, cannot update/cannot organize the data in the files.

**Excel:**

* It is less secure than files.
* By storing more data it works slower.
* For security and storage purposes we are not going for Excel.

**DBMS(Database Management System):** It can store data in the form of tables.

**Challenges of DBMS:**

* Relations are not possible for accessing the data.

**The database is of 2 types:**

1. **RDMS(Relational Database Management System):** It stores the data in the form of tables and map the data from one location to another location.

**Advantages:**

* It will retrieve data very quickly.
* Operations are also very effective.

**2.Non-RDMS(Non-Relational Database Management system):** It stores the data in the form of key values format.

MySQL Database(Mysql with SQL)

Mysql refers to the server platform SQL Language.

**To create a project we need**

1.frontend-- >To view the data

2. Back end---->interaction between the data and programming.

3. Database: To store the data and provide space to store applications.

Database Components

1. Client

2. Server

MySql uses 2 types of commands

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

DDL commands create, modify, and delete the database's structure like table, and schemas.

**Commands**

**1. Create:** Used to create databases, tables.

**2. Alter:** Add a row or column to the existing table.

**3. Drop:** delete records from the database.

**4. Truncate:**It will remove the records from the table.

**2.DML(Data Manipulation Language):**It deals with the manipulation of the data present in the database.

**Commands**

**Insert:** Used to insert data into the table.

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

**Delete** Deletes the records from the database.

**Call:** Call a PL/SQL or Java subprogram.

**Explain call:** Describe the access path to the data**.**

**Data Types:**

**Char(size):** Fixed lengths of characters are allowed.

**Varchar(size):** Variable length string is allowed.

**Binary(size):** Equal to char but stores binary values by default it is 1.

**Text(size):** Holds a string with a max length of 65 to 535 bytes.

**TINYTEXT**: Holds a string of a maximum of 255 characters.

Steps and syntax:

1. create a database as create **database name;**

2. Enter into the database as **use database name;**

3. To view the tables in the database use **show tables;**

4. create a table as

**Create table table name(col1 datatype(size),col2 datatype(size)……));**

5. Insert the data into the table as

**Insert into table name values(‘value1’,’value2’,’value3’…….);**

1. To delete a row from the table

**delete from tablename where condition;**

1. To delete the records from the database

**Drop table tablename;**

1. To delete all records in the table

**Truncate table tablename;**

1. To add a new column into the table

**Alter table tablename add column columnname datatype(size);**

10.To rename the columnname

**alter table tablename rename columnname to newcolumnname;**

11.To update a particular value in the table

**Update tablename set columnname=”new value” where condition;**

12.To drop a column from the table

**Alter table tablename drop cloumn columnname;**

**Clauses and operators**

1. **Where:**It is used to exact particular record in the table.Mainly used for filtering

Select colname from tablename where condition;

**2.AND,OR,NOT:** if all the conditions are satisfied then the result will be true.(AND).

T T T

T F F

F T F

F F F

If any one of the condition satisfy then the result will be true.(OR)

T T T

T F T

F T T

F F F

Display the records when condition fails.

T F

F T

1. **ORDERBY:**used for sorting the records in the table

Select col1,col2 from tablename orderby col1,col2.

1. **INSERT INTO:**used to insert new records into already existing records.

**Insert into tablename(col1,col2,col3……..)values(val1,val2.val3..);**

1. **SELECT :**To display /to obtain the data from particular table.

Select \* from tablename;

1. **UPDATE:**modify/change the existing value

**Update table\_name det col1=new value where condition;**

1. **DELETE:**deletes the existing record from the table

Delete from table where condition;

1. **MIN AND MAX:**returns minimum value in a record

Max is used to return maximum value in a record

**Select max(colname) from tablename where condition;**

**Select min(colname) from tablename where condition;**

1. **LIKE:**used in where clause if we want to obtain a specific pattern or search for a specific pattern in a col

Select col1,col2 from tablename where col1 like pattern

%a ->finds pattern ending with a

A%->find pattern starting with a

\_a%->find value a in 2nd position is a

1. **IN:**allow you to specify multiple values in the where clause

Select colname from tablename where colname in(val1,val2);

1. **BETWEEN:**used to select the middle value from range of values.

Select name from tablename where colname between val1 and val2;

1. **GROUPBY:**groups the data present in the rows with same value

Select column name from tablename where condition groupby colname orderby colname;

1. **COUNT:**Returns the number of records which satisfy the condition

Select count(colname)from tablename where condition;

1. **AVG:**average value of a particular column.

Select avg(colname) from tablename where condition;

1. **SUM:**gives the total of the numbers present in a column

Select sum(coname)from tablename where condition;



**Joins**:

* Joins are used with select statement.
* It is used to retrieve data from multiple tables from same database.
* Two Tables should be in single database.
* It is used to fetch records from different tables easily.

**There are 3 types of MySql Joins**

1. **Inner join:**

It is also known as simple join.In order to return all the rows from multiple tables where join condition is satisfy.This is most commonly used join in mysql.

**Syntax**:**select columns from table1 inner join table2 on table1.column=table2.column;**

1. **Outer join: Left outer join**

**Right outer join**

**Outer Left Join:**Returns all rows from left hand side and all the rows from right hand side table by satisfying the join condition

**Syntax**:

Select columns from table1 Left Join table2 on table1.col=table2.col;

**Outer Right Join:(Right join)**Returns all the rows from right hand side and all the rows from left hand side by satisfying the join condition;

**Syntax**:

SELECT COLUMNS FROM TABLE1 RIGHT JOIN TABLE2 ON TABLE1.COL=TABLE2.COL;

**SELF JOIN:**Data/rows in the table or combined/joined with the same data/rows in the same table;

**Syntax**:SELECT COLUMNNAME FROM TABLE1,TABLE2 WHERE CONDITION;

**CROSS JOIN:**It will return all the rows /records from both tables (table1,table2).

**Syntax:**

SELECT COLUMNNAME FROM TABLE1 CROSS JOIN TABLE2 ON TABLE1.COL=TABLE2.COL

create database shailu;

use shailu;

create table student(student\_id int(10) primary key,name varchar(20),dept varchar(20),gendar varchar(10));

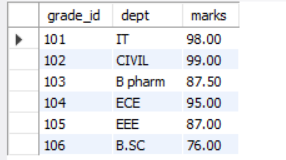
Insert into student values(1,'Shailaja','IT','Female'),(2,'Sreekar','CIVIL','Male'),(3,'kiran','B pharm','Male'),(4,'Mounika','ECE','Female'),(5,'Srikanth','EEE','Male'),(6,'pravalika','ECE','Female'),(7,'Priyanka','CSE','Female'),(8,'Gouthami','B.SC','Female');

select \* from student;

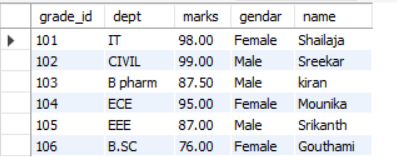
create table grade(grade\_id int(10) primary key,student\_id int(10),course varchar(20),marks decimal(10,2),foreign key (student\_id) references student(student\_id));

insert into grade values(101,1,'Maths',98.0),(102,2,'Science',99.0),(103,3,'Hindi',87.5),(104,4,'english',95),(105,5,'Maths',87),(106,8,'Social',76);

**select grade\_id,dept,marks from grade inner join student on grade.student\_id=student.student\_id;**

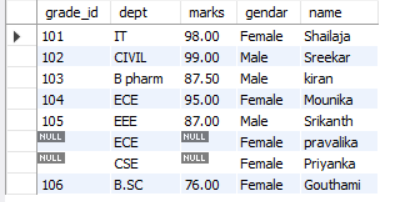


**select grade\_id,dept,marks,gendar,name from grade left join student on grade.student\_id=student.student\_id;**

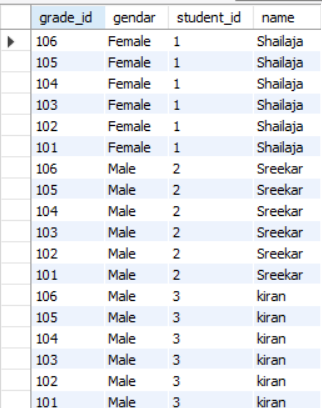


It returns all the rows from the grade table and values that satisfy the condition records will be return

**select grade\_id,dept,marks,gendar,name from grade right join student on grade.student\_id=student.student\_id;**



**select grade\_id,dept,marks,gendar,name from grade cross join student;**



**select \* from grade self join grade;**

