create database project;

use project;

* **Create table**

create table emp(id int,name varchar(20),dept varchar(20));

**select query**

select \* from emp;

**insert query**

insert into emp values(1,'ram','mechanic');

insert into emp values(2,'vasu','software');

insert into emp values(1,'hari','sales');

insert into emp values(1,'balaji','mechanic');

insert into emp values(1,'venkat','sales');

set sql\_safe\_updates=0;

**-- update**

update emp set id=5 where name ='venkat';

**-- delete**

delete from emp where id=5;

**-- alter**

alter table emp add salary int;

alter table emp add mobile int after name;

alter table emp drop column mobile;

alter table emp rename to employee;

**-- update**

update emp set salary=4000 where id=4;

select \* from employee;

show tables;

insert into employee values(5,'babu','sales',6000);

-**- where**

select \* from employee where id=3;

select name,dept from employee where id=3;

**-- And opertor**

select \* from employee where dept='sales' and salary>5000;

**-- or operator**

select \* from employee where dept='sales' or salary>5000;

**-- like operator**

select \* from employee where name like 'r%';

select \* from employee where name like '%r%';

select \* from employee where name like '\_r%';

**-- in condition**

select \* from employee where name in('hari','balaji','krishna');

**-- between condition**

select \* from employee where salary between 1000 and 4000;

**-- distinct**

select distinct dept from employee;

**-- order by**

select \* from employee order by salary desc;

select \* from employee order by salary asc;

select \* from employee order by salary;

**-- date yyyy-mm-dd**

select curdate() as today;

**Aggregate functions**

**count()**

**sum()**

**avg()**

**min()**

**max()**

-- count()

select count(\*) from employee;

-- sum()

select sum(salary) as sum from employee;

-- avg()

select avg(salary) as average from employee;

-- min()

select min(salary) as minimun from employee;

-- max()

select max(salary) as maximum from employee;

**Group By**

Group by statement groups rows that have the same values into summary rows, like

“ find the number of customers in each country”

The group by statement is often used with aggregate functions (count(),max(),min(),sum(),Avg()) to group the result-set by one or more columns.

-- group by

select count(\*) as count,dept from employee group by dept;

**Having clause**

Having clause was added to SQL because the where keyword cannot be used with aggregate functions.

Having clause is used with group by clause.

**-- having**

select count(\*) as number, dept from employee group by dept having count(\*)>1;

**Joins**

It is used to retrieve data from multiple tables. It is performed whenever you need to fetch records

From two or more tables.

There are three types of MySQL joins

Inner join

Left Outer Join

Right Outer join

create table officers(officerId int,officerName varchar(20),address varchar(20));

create table students(studentID int,StudentName varchar(20),courseName varchar(20));

select \* from officers;

insert into officers values(1,'ram','chennai'),

(2,'vasu','bangalore'),

(3,'janani','chennai'),

(5,'venkat','bangalore');

select \* from students;

insert into students values(1,'babu','java'),

(2,'hari','c++'),

(3,'ramu','c'),

(7,'narasim','python');

**-- inner join**

select officers.officerName,officers.address,students.courseName from officers inner join students on

officers.officerId=students.studentId;

**-- left outer join**

select officers.officerName,officers.address,students.courseName from officers left outer join students on

officers.officerId=students.studentId;

**-- right outer join**

select officers.officerName,officers.address,students.courseName from officers right outer join students on

officers.officerId=students.studentId;

**view**

Its contains rows and columns similar to the real table. The view is a virtual table created by a query by joining one or more tables.

-- view

create view OG as select name,salary from employee;

select \* from OG;

alter view OG as select name,salary,dept from employee;

drop view OG;

**Constraints**

Primary key

Unique key

Non null

Check

Foreign key

**-- primary key**

create table shop(id int primary key,name varchar(20),dept varchar(20));

select \* from shop;

desc shop;

insert into shop values(1,'balaji','manager');

insert into shop values(1,'ram','manager');

insert into shop values(null,'john','sales');

**-- unique key**

create table medical(id int primary key,name varchar(20)unique,dept varchar(20)unique);

insert into medical values(1,'balaji','manager');

insert into medical values(2,'balaji','sales');

insert into medical values(3,null,'salaes');

select \* from medical;

**-- not null**

create table hospital(id int primary key,name varchar(30)not null);

insert into hospital values(1,'ram');

insert into hospital values(2,null);

**-- check constraint**

create table person(id int primary key,name varchar(20),salary int,check(salary>5000));

select \* from person;

insert into person values(1,'ram',4000);

insert into person values(1,'ram',7000);

**Foreign key**

Foreign key is used to link one or more than one table together.

Another name for foreign key is referencing key. It creates a link between two tables.

This creates a parent-child type of relationship where the table with the foreign key in the child table refers to the primary key column in the parent table.

-- foreign key

create table customer(ID int not null auto\_increment,name varchar(20)not null,city varchar(20) not null,primary key(ID));

select \* from customer;

create table contact(id int,customer\_id int,customer\_info varchar(30)not null,type varchar(20)not null,

constraint fk\_customer foreign key(customer\_id) references customer(ID));

select \* from customer;

insert into customer(name,city) values('ram','chennai');

insert into customer(name,city) values('vasu','chennai'),('venkat','bangalore');

select \* from contact;

insert into contact values(1,1,'it','online');

insert into contact values(1,2,'sales','online');

insert into contact values(3,3,'market','online');

insert into contact values(1,4,'it','online');

delete from contact where customer\_id=3;

select \* from customer;

delete from customer where id=2;

drop table contact;

select \* from contact;

create table contact(id int,customer\_id int,customer\_info varchar(30)not null,type varchar(20)not null,

constraint fk\_customer foreign key(customer\_id) references customer(ID)

on delete cascade

on update cascade);

select \* from contact;

select \* from customer;

delete from customer where id=2;

update customer set id=4 where name='venkat';

set sql\_safe\_updates=0;