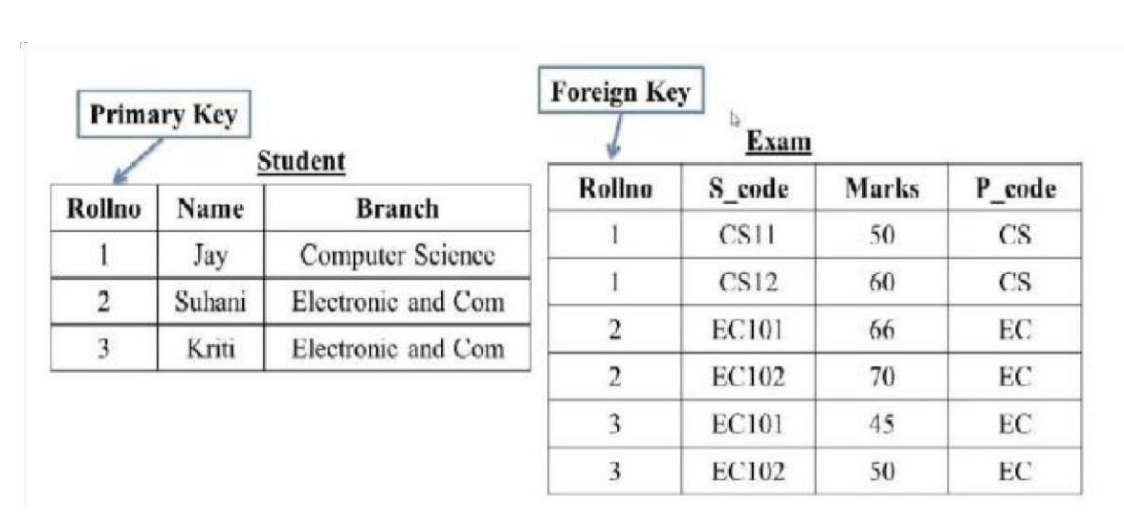
[1] Create Table Name: Student and Exam



CREATE TABLE student(

Rollno int PRIMARY KEY,

Name varchar(50),

Branch varchar(50)

);

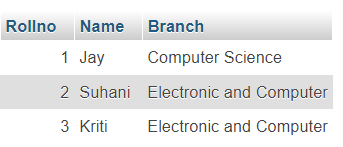
INSERT INTO student(

Rollno ,Name,Branch)VALUES

(1,"Jay","Computer Science"),

(2,"Suhani","Electronic and Computer"),

(3,"Kriti","Electronic and Computer");



CREATE TABLE Exam (

Rollno int,

S\_code varchar (50),

Marks int,

P\_code varchar(50),

FOREIGN KEY(Rollno) REFERENCES student(Rollno)

);

INSERT INTO exam(Rollno,S\_code,Marks,P\_code)VALUES

(1,"CS11",50,"CS"),

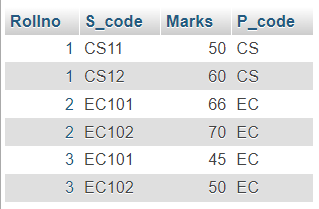
(1,"CS12",60,"CS"),

(2,"EC101",66,"EC"),

(2,"EC102",70,"EC"),

(3,"EC101",45,"EC"),

(3,"EC102",50,"EC");



[2] Create table given below:



Create table: Employee

CREATE TABLE Employee(

Employee\_id int PRIMARY KEY AUTO\_INCREMENT,

First\_name varchar(50),

Last\_name varchar(50),

Salary int,

Joining\_date DATETIME,

Department varchar(50)

);

Insert Data into table

INSERT INTO employee( Employee\_id,First\_name,Last\_name,Salary,Joining\_date,Department)VALUES

(1,"John","Abraham",1000000,” 2013-01-01” ,"Banking");



INSERT INTO employee (First\_name,Last\_name,Salary,Joining\_date,Department)VALUES

("Michael","Clarke",800000,"2013-01-01","Insurance"),

("Roy","Thomas",700000,"2013-02-01","Banking"),

("Tom","Jose",600000,"2013-02-01","Insurance"),

("Jerry","Pinto",650000,"2013-02-01","Insurance"),

("Philips","Mathew",750000,"2013-01-01","Services"),

("TestName1","123",650000,"2013-01-01","Services"),

("TestName2","Lname%",600000,"2013-02-01","Insurance");



Create table: Incentive

CREATE TABLE Incentive(

Employee\_ref\_id int,

Incentive\_date DATE,

Incentive\_amount int,

FOREIGN KEY(Employee\_ref\_id) REFERENCES employee(Employee\_id)

);

Insert data into table

INSERT INTO incentive(Employee\_ref\_id,Incentive\_date,Incentive\_amount)VALUES

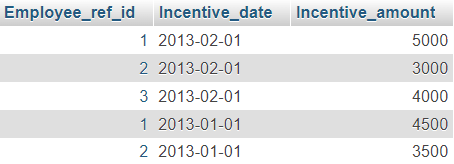
(1,"2013-02-01",5000),

(2,"2013-02-01",3000),

(3,"2013-02-01",4000),

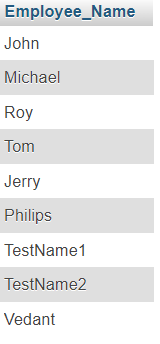
(1,"2013-01-01",4500),

(2,"2013-01-01",3500);



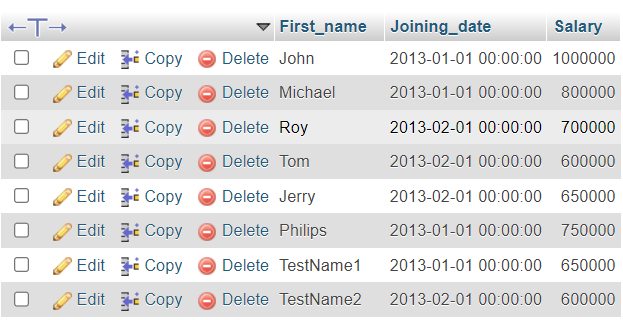
1. Get First\_Name from employee table using Tomname “Employee Name”.

SELECT First\_name AS Employee\_Name FROM employee;



1. Get FIRST\_NAME, Joining Date, and Salary from employee table.

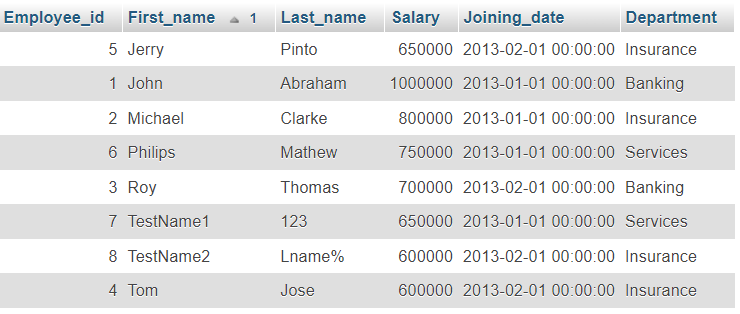
SELECT First\_name,Joining\_date,Salary FROM employee;



1. Get all employee details from the employee table order by First\_Name Ascending and salary descending?

First\_name Ascending:

SELECT \* FROM employee ORDER BY First\_name ASC;



Salary descending:

SELECT \* FROM employee ORDER BY Salary DESC;



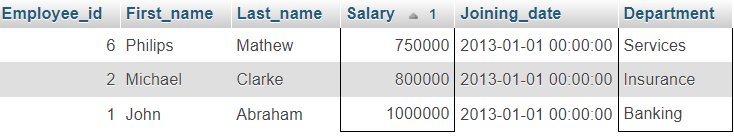
1. Get employee details fromemployee table whose first name contains ‘J’.

SELECT \* FROM employee WHERE First\_name LIKE '%J%';

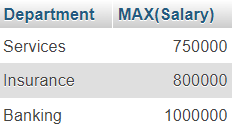


1. Get department wise maximum salary from employee table order by salary ascending?

SELECT \* FROM employee WHERE (Department,Salary) IN (SELECT Department,MAX(Salary) FROM employee GROUP BY Department) ORDER BY Salary ASC;

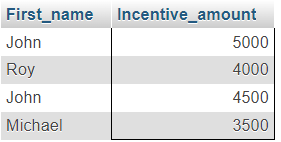


SELECT Department,MAX(Salary) FROM employee GROUP BY Department ORDER BY Salary ASC;



(F) Select first\_name, incentive amount from employee and incentives table for those employees who have incentives and incentive amount greater than 3000.

SELECT employee.First\_name,incentive.Incentive\_amount FROM employee INNER JOIN incentive ON employee.Employee\_id = incentive.Employee\_ref\_id WHERE incentive.Incentive\_amount > 3000;



(G) Create After Insert trigger on Employee table which insert records in viewtable.

CREATE TABLE Employee\_trigger(

Id int,

F\_name varchar(50),

L\_name varchar(50),

T\_salary int,

Join\_date datetime,

Dept varchar(50),

FETCH\_Time TIMESTAMP,

ACTION varchar(50)

);



Create Trigger:

CREATE TRIGGER trigger\_ins AFTER INSERT ON employee FOR EACH ROW INSERT INTO employee\_trigger(Id,F\_name,L\_name,T\_salary,Join\_date,Dept,ACTION)

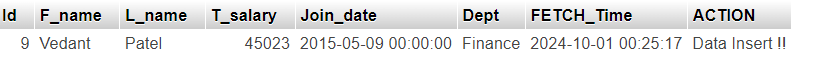
VALUES(NEW.Employee\_id, NEW.First\_Name, NEW.Last\_Name, NEW.Salary, NEW.Joining\_date, NEW.Department,"Data Insert !!");

Perform and check:

INSERT INTO employee(First\_name,Last\_name,Salary,Joining\_date,Department)

VALUES

("Vedant","Patel",45023,"2015-05-09","Finance");



[3] Create table given below: Salesperson and Customer.

Table – 1: Salesperson table

CREATE TABLE Sales\_Person(

SNo int PRIMARY KEY NOT null,

SNAME varchar(50),

CITY varchar(50),

COMM float(2)

);

INSERT INTO sales\_person(SNo,SNAME,CITY,COMM)VALUES

(1001,"Peel","London",.12),

(1002,"Serres","San Jose",.13),

(1004,"Motika","London",.11),

(1007,"Rafkin","Barcelona",.15),

(1003,"Axelrod","New York",.1);



Tabel – 2: Customer Table

CREATE TABLE Customer(

CNM int,

CNAME varchar(50),

CITY varchar(50),

RATING int,

SNo int,

FOREIGN KEY (SNo) REFERENCES sales\_person (SNo)

);

INSERT INTO customer(CNM,CNAME,CITY,RATING,SNo)VALUES

(201,"Hoffman","London",100,1001),

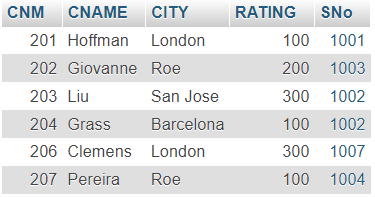
(202,"Giovanne","Roe",200,1003),

(203,"Liu","San Jose",300,1002),

(204,"Grass","Barcelona",100,1002),

(206,"Clemens","London",300,1007),

(207,"Pereira","Roe",100,1004);



Retrieve the below data from above table

(A) All Customer name whose rating is more than 100.

SELECT CNAME FROM customer WHERE RATING > 100;

(B) Names and cities of all salespeople in London with commission above 0.12.

SELECT SNAME,CITY FROM sales\_person WHERE CITY = "London" AND COMM > 0.12;



(C) All salespeople either in Barcelona or in London.

SELECT \* FROM sales\_person WHERE CITY = "London" OR CITY = "Barcelona";



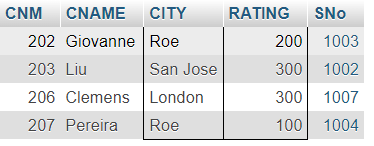
(D) All salespeople with commission between 0.10 and 0.12. (Boundary values should be excluded).

SELECT \* FROM sales\_person WHERE COMM > 0.10 AND COMM < 0.12;



(E) All customers excluding those with rating <= 100 unless they are located in Roe.

SELECT \* FROM customer WHERE (RATING > 100 AND CITY != "Roe") OR CITY = "Roe";



(F) Write a SQL statement that displays all the information about all salespeople.

(G) From the following table, write a SQL query to find orders that are delivered by a salesperson with ID. 5001. Return ord\_no, ord\_date, purch\_amt.

Sample table: orders

[4] From the following table, write a SQL query to select a range of products whose price is in the range Rs.200 to Rs.600. Begin and end values are included. Return pro\_id, pro\_name, pro\_price, and pro\_com.

🡪CREATE TABLE item\_mast(

PRO\_ID int AUTO\_INCREMENT PRIMARY KEY,

PRO\_NAME varchar(100),

PRO\_PRICE FLOAT(2),

PRO\_COM int

);

🡪INSERT INTO item\_mast(PRO\_ID,PRO\_NAME,PRO\_PRICE,PRO\_COM)VALUES

(101,"Mother Board",3200.,15);

🡪INSERT INTO item\_mast(PRO\_NAME,PRO\_PRICE,PRO\_COM)VALUES

("Key Board",450.00,16),

("ZIP drive",250.00,14),

("Speaker",550.00,16),

("Monitor",5000.00,11),

("DVD drive",900.00,12),

("CD drive",800.00,12),

("Printer",2600.00,13),

("Refill cartridge",350.00,13),

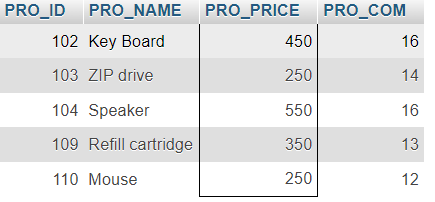
("Mouse",250.00,12);

🡪Write a SQL query to select a range of products whose price is in the range Rs.200 to Rs.600. Begin and end values are included. Return pro\_id, pro\_name, pro\_price, and pro\_com.

SELECT PRO\_ID, PRO\_NAME, PRO\_PRICE, PRO\_COM

FROM item\_mast

WHERE PRO\_PRICE BETWEEN 200 AND 600;



🡪Write a SQL query to calculate the average price for a manufacturer code of 16. Return avg.

SELECT AVG(PRO\_PRICE) AS avg\_price

FROM item\_mast

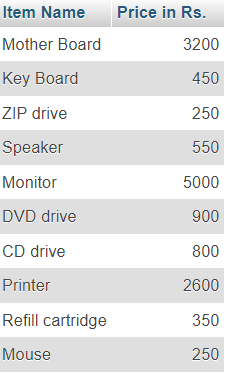
WHERE PRO\_COM = 16;



🡪Write a SQL query to display the pro\_name as 'Item Name' and pro\_priceas 'Price in Rs.'

SELECT PRO\_NAME AS "Item Name", PRO\_PRICE AS "Price in Rs."

FROM item\_mast;

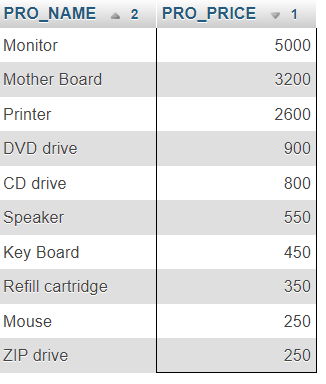


🡪Write a SQL query to find the items whose prices are higher than or equal to $250. Order the result by product price in descending, then product name in ascending. Return pro\_name and pro\_price.

SELECT PRO\_NAME, PRO\_PRICE

FROM item\_mast WHERE PRO\_PRICE >= 250

ORDER BY PRO\_PRICE DESC, PRO\_NAME ASC;



🡪Write a SQL query to calculate average price of the items for each company. Return average price and companycode.

SELECT PRO\_COM AS company\_code, AVG(PRO\_PRICE) AS average\_price

FROM item\_mast GROUP BY PRO\_COM;

