**SQL ASSIGNMENT**

**Que -1) Create Table Name : Student and Exam**

CREATE DATABASE assignment;

**STUDENT TABLE**

**QUERY** - CREATE TABLE Student(Rollnumber int primary key auto\_increment not null,

Name varchar(20),

Branch Varchar(30)

);

INSERT INTO student(Rollnumber,Name,Branch)VALUES(1,"JAY","Computer Science"),

(2,"suhani","electronic and comm"),

( 3,"prince","electronic and comm");

**EXAM TABLE**

**QUERY** - CREATE TABLE Exam(Rollnumber int NOT null,

S\_code varchar(10),

Marks int ,

p\_code varchar(10),

FOREIGN KEY(Rollnumber) REFERENCES student(Rollnumber)

);

INSERT INTO exam(Rollnumber,S\_code,Marks,p\_code) VALUES

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

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

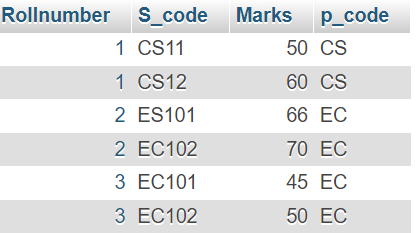
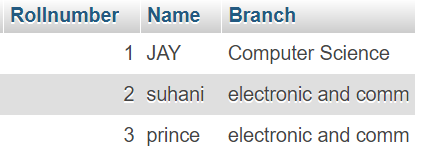
(2,"ES101",66,"EC"),

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

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

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

**Student Table Exam table**



**QUE -2 ) Create table given below: Employee and IncentiveTable**

**Employee Table**



CREATE TABLE EMPLOYEES(Employee\_id int PRIMARY KEY AUTO\_INCREMENT NOT null,

First\_name varchar(20),

Last\_name varchar(20),

Salary int,

Joining\_date date,

Department varchar(20)

) ;

**Incentive Table**

CREATE TABLE Incentive(Employee\_ref\_id int,

incentive\_date date,

incentive\_amount int);

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

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

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

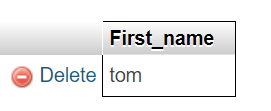
(3,2013-02-01,4000),

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

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

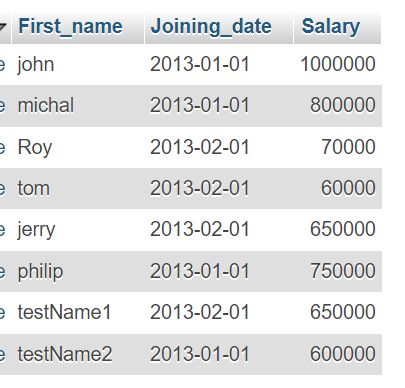
**QUE -3 ) Get First\_Name from employee table using Tom name “Employee Name”.**

**Query :-** SELECT First\_name FROM employees WHERE First\_name = "TOM";

****

**QUE -4 ) Get FIRST\_NAME, Joining Date, and Salary from employee table.**

**Query :-** SELECT First\_name,Joining\_date,Salary FROM employees;



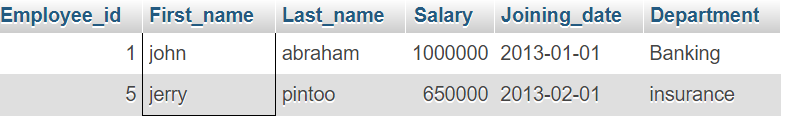
**QUE -5) Get all employee details from the employee table order by First\_Name Ascending and Salary descending?**

**Query :-** SELECT \* FROM employees ORDER BY First\_name ASC , Salary DESC;



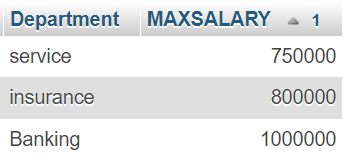
**QUE – 7) Get employee details from employee table whose first name contains ‘J’.**

**Query :-** SELECT \* FROM employee WHERE First\_name like "%J%";



**QUE -8) Get department wise maximum salary from employee table order by salaryascending?**

**Query :-** SELECT Department, max(SALARY) MAXSALARY FROM employee GROUP BY Department ORDER BY MAXSALARY ASC;



**QUE – 9) Select first\_name, incentive amount from employee and incentives table forthose employees who have incentives and incentive amount greater than 3000**

**Query :-** SELECT employees.First\_name , incentive.incentive\_amount FROM employees LEFT JOIN incentive ON employees.Employee\_id = incentive.Employee\_ref\_id WHERE incentive.incentive\_amount > 3000;



**QUE – 10) Create After Insert trigger on Employee table which insert records in viewtable**

**QUE – 11)Create table given below: Salesperson and Customer**

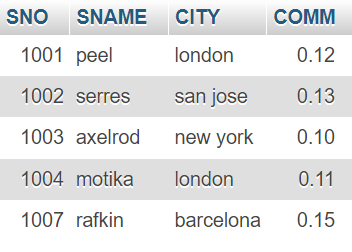
CREATE TABLE Salesperson(SNO INT PRIMARY KEY NOT null,

SNAME VARCHAR(20),

CITY VARCHAR(20),

COMM numeric(3,2);

**Salesperson Table**

￼**￼**

INSERT INTO salesperson(SNO,SNAME,CITY,COMM) VALUES (1001,"peel","london",0.12),

(1002,"serres","san jose",0.13),

(1004,"motika","london",0.11),

(1007,"rafkin","barcelona",0.15),

(1003,"axelrod","new york",0.1);

CREATE TABLE Customer(CNM INT PRIMARY KEY NOT NULL,

CNAME VARCHAR(20),

CITY VARCHAR(20),

RATING INT,

SNO INT,

FOREIGN KEY(SNO) REFERENCES salesperson(SNO));

INSERT INTO customer(CNM,CNAME,CITY,RATING,SNO) VALUES (201,"Hoffman","London",100,1001),

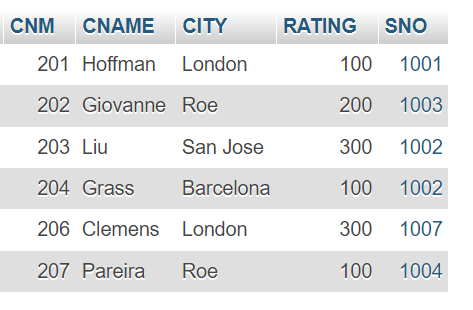
(202,"Giovanne","Roe",200,1003),

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

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

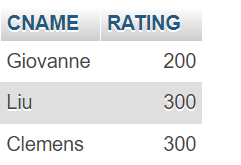
(206,"Clemens","London",300,1007),

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

****

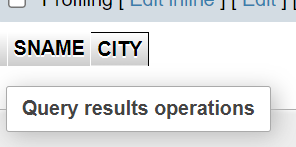
**QUE – 12 )All customer rating for more than 100.**

**Query :-** SELECT CNAME, RATING FROM `customer` WHERE Rating > 100;



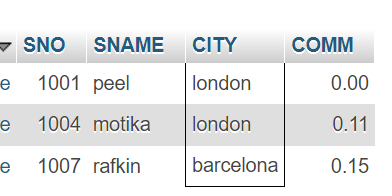
**QUE – 13) Names and cities of all salespeople in London with commission above 0.12**

**Query :-** SeLECT SNAME ,CITY FROM salesperson WHERE CITY = "london" AND COMM >0.12;



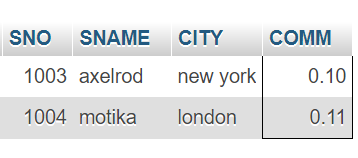
**QUE – 14) All salespeople either in Barcelona or in London**

**Query :-** SELECT \* FROM salesperson WHERE CITY = "Barcelona" Or CITY = "London";



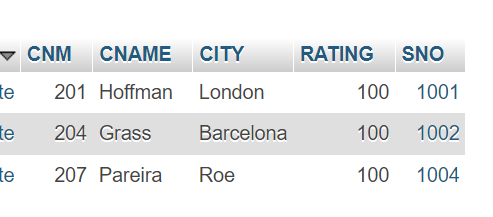
**QUE – 15) All salespeople with commission between 0.10 and 0.12.**

**Query :-** SELECT \* FROM salesperson WHERE COMM BETWEEN 0.10 AND 0.12;



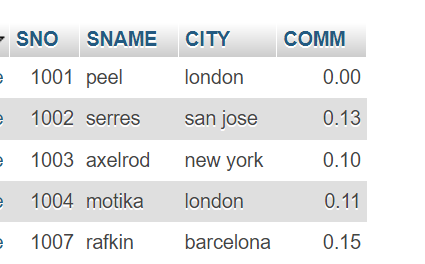
**QUE – 16) All customers excluding those with rating <= 100 unless they are located inRome**

**Query :-** SELECT \* FROM customer WHERE Rating <= 100;



**QUE – 17) Write a SQL statement that displays all the information about all salespeople**

**Query :-** Select \* from salesperson



**QUE – 18)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.**

**Query :-** CREATE TABLE Orders(order\_no INT PRIMARY KEY NOT null,

putch\_amt numeric(5,2),

ord\_date date,

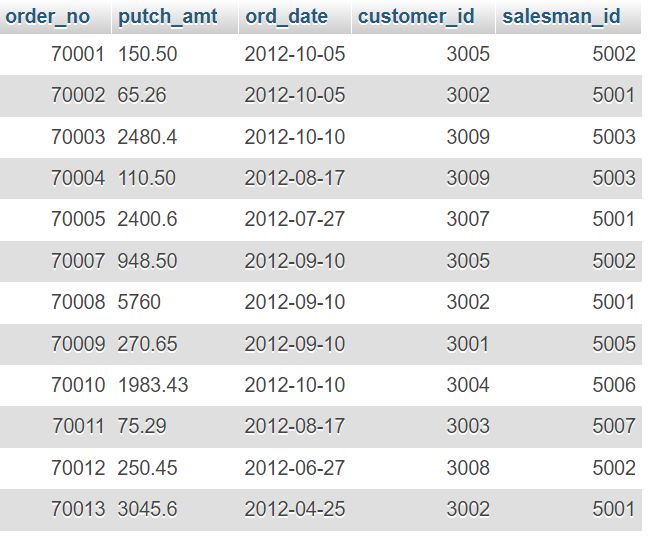
customer\_id int,

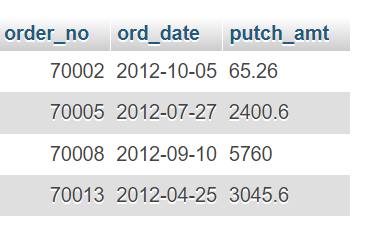
salesman\_id int );

SELECT order\_no, ord\_date, putch\_amt

FROM orders

WHERE salesman\_id = 5001;

****

****

**QUE – 19) 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.**

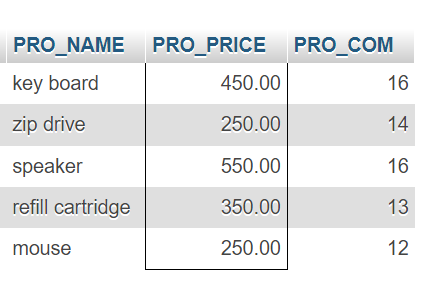
CREATE TABLE item\_mast(PRO\_ID INT PRIMARY KEY NOT null,

PRO\_NAME VARCHAR(20),

PRO\_PRICE NUMERIC(7,2),

PRO\_COM int );

**Query :-** SELECT PRO\_ID , PRO\_NAME,PRO\_PRICE , PRO\_COM FROM item\_mast WHERE PRO\_PRICE BETWEEN 200 AND 600 ;



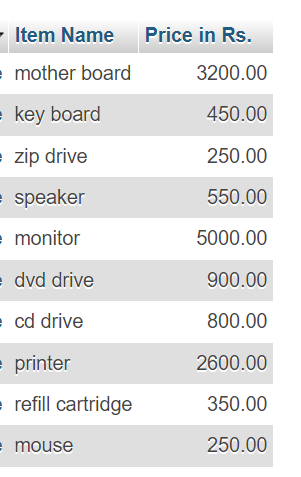
**QUE – 20) From the following table, write a SQL query to calculate the average price for a manufacturer code of 16. Return avg.**

**Query :-** SELECT AVG(PRO\_PRICE) FROM item\_mast WHERE PRO\_COM = 16;



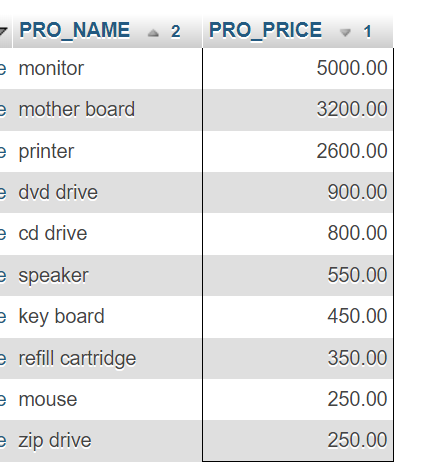
**QUE – 21) From the following table, write a SQL query to display the pro\_name as 'Item Name' and pro\_priceas 'Price in Rs.'**

**Query :-** SELECT PRO\_NAME as "Item Name", PRO\_PRICE AS "Price in Rs." FROM item\_mast;



**QUE – 22) From the following table, 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.**

**Query : -** SELECT PRO\_NAME,PRO\_PRICE FROM item\_mast WHERE PRO\_PRICE >= 250 ORDER by PRO\_PRICE DESC ,PRO\_NAME ASC;



**QUE – 23)From the following table, write a SQL query to calculate average price of the items for each company. Return average price and companycode**

**Query :-** SELECT AVG(PRO\_PRICE) FROM item\_mast;

