**Que – 1). Create Table Name : Student and Exam**

Ans – 1) Create Database =

[CREATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-database.html) [DATABASE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-database.html) student\_db;

* create Student table =

[CREATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) [TABLE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) student( Rollno int PRIMARY KEY,

Name varchar(20) [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_not) NULL,

Branch varchar(20) [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_not) NULL );

* insert data in student table =

INSERT INTO student VALUES (1, 'Jay', 'Computer Science');

INSERT INTO student VALUES (2, 'Suhani', 'Electronic and Com');

INSERT INTO student VALUES (3, 'Kriti', 'Electronic and Com');

* create table Exam using foreign key =

[CREATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) [TABLE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) Exam ( Rollno int [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_not) NULL,

S\_code varchar(10),

Marks int, P\_code varchar(3),

FOREIGN KEY (Rollno) REFERENCES student(Rollno) );

* insert data into Exam table =

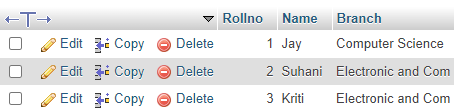
INSERT INTO Exam (Rollno, S\_code, Marks, P\_code) VALUES

(1, 'CS11', 50, 'CS'), (1, 'CS12', 60, 'CS'),

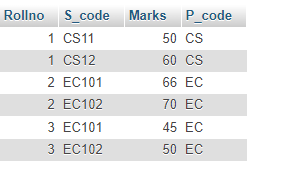
(2, 'EC101', 66, 'EC'), (2, 'EC102', 70, 'EC'),

(3, 'EC101', 45, 'EC'), (3, 'EC102', 50, 'EC');

Student Table =



Exam Table =



**Que – 2). Create table given below: Employee and IncentiveTable**

**Ans – 2).** create database name Employee\_db =

CREATE DATABASE Employee\_db;

* create table name Employee =

CREATE TABLE employee ( Employee\_id int, First\_name varchar(20),

Last\_name varchar(20), Salary int,

Joining\_date datetime,Department varchar(20));

* Insert value =

INSERT INTO employee VALUES (1, 'John', 'Abraham', 1000000, '2013-01-13 12.00.00', 'Banking');

INSERT INTO employee VALUES (2, 'Michael', 'Clarke', 800000, '2013-01-13 12.00.00', 'Insurance');

INSERT INTO employee VALUES (3, 'Roy', 'Thomas', 700000, '2013-01-13 12.00.00', 'Banking’);

INSERT INTO employee VALUES (4, 'Torm', 'Jose', 600000, '2013-02-13 12.00.00', 'Insurance');

INSERT INTO employee VALUES (5, 'Jerry', 'Pinto', 650000, '2013-02-01 12.00.00', 'Insurance');

INSERT INTO employee VALUES (6, 'Philip', 'Mathew', 750000, '2013-01-01 12.00.00', 'Services');

INSERT INTO employee VALUES (7, 'TestName1', '123', 650000, '2013-01-01 12.00.00', 'Services');

INSERT INTO employee VALUES (8, 'TestName2', 'Lname%', 650000, '2013-02-01 12.00.00', 'Insurance');

Step-4) Create the Incentive Table:

CREATE TABLE Incentive (

Employee\_ref\_id int,

Incentive\_date date,

Incentive\_amount int,

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

);

Step-5) Insert Data into Incentive table =

INSERT INTO incentive VALUES

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

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

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

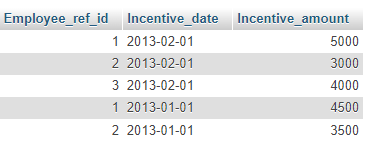
(1, '2013-01-01', 4500),

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

* Employee Table



* Incentive Table



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

Ans – 3) SELECT First\_name FROM employee WHERE First\_name LIKE '%Tom%';



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

Ans – 4) SELECT First\_name, Joining\_date, Salary FROM employee;



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

Ans – 5) SELECT \* FROM employee ORDER by First\_name ASC, Salary DESC;



**Que - 6). Get employee details from employee table whose first name contains ‘J’**

Ans – 6)

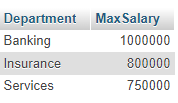
[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM Employee WHERE First\_name [LIKE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/string-comparison-functions.html%23operator_like) '%J%';



**Que – 7). Get department wise maximum salary from employee table order by**

Ans – 7)

SELECT Department , MAX(Salary) AS MaxSalary FROM employee GROUP by Department ORDER by Salary DESC;



**Que – 8). salaryascending?**

Ans – 8)

Query -->

SELECT \* FROM `employee` ORDER by Salary;

Output -->



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

Ans – 9) SELECT e.first\_name, i.Incentive\_amount

FROM employee

INNER JOIN incentive i ON e.employee\_id = i.Employee\_ref\_id

WHERE i.incentive\_amount > 3000;



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

Ans – 10) Create Database :

CREATE DATABASE TRIGGER\_db;

* Create table Employee :

CREATE TABLE Employee( id int, Ename varchar(20), Salary int);

* Create table Viewtable :

CREATE TABLE Viewtable( id int, name varchar(20),

Salary int, action\_perform text);

* Create Trigger :

DELIMITER $$

CREATE TRIGGER viewtable AFTER INSERT on employee for EACH ROW

BEGIN

INSERT INTO viewtable(id, name, Salary, action\_perform)

VALUES(new.id, new.Ename, new.Salary, 'Record Inserted');

END

* Insert Record in Employee Table :

INSERT INTO employee VALUES (1, 'Devarsh', 50000);

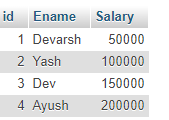
INSERT INTO employee VALUES (2, 'Yash', 100000);

INSERT INTO employee VALUES (3, 'Dev', 150000);

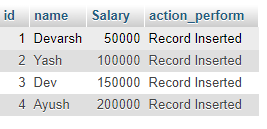
INSERT INTO employee VALUES (4, 'Ayush', 200000);

* Viewtable Data :

Employee Table :



Viewtable :



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

Ans – 11) Create Database :

CREATE DATABASE bussiness\_db;

🡪 Create Table Salesperson :

CREATE TABLE Salesperson (

SNo int, SNAME varchar(10), CITY varchar(10),

COMM int, PRIMARY KEY(SNo) );

🡪 Create Table Customer :

CREATE TABLE CUSTOMER (

CNM int, CNAME VARCHAR(15), CITY VARCHAR(15),

RATING INT, SNo int, PRIMARY KEY (CNM),

FOREIGN KEY (SNo) REFERENCES salesperson (SNo) );

🡪 Insert Data in Salesman table :

INSERT INTO salesperson VALUES (1001, 'Peel', 'London', 12);

INSERT INTO salesperson VALUES (1002, 'Serres', 'San Jose', 13);

INSERT INTO salesperson VALUES (1004, 'Motika', 'London', 11);

INSERT INTO salesperson VALUES (1007, 'Rafkin', 'Barcelona', 15);

INSERT INTO salesperson VALUES (1003, 'Axelord', 'New York', 1);

--> Insert Data into Customer Table :

INSERT INTO customer VALUES (201, 'Hoffman', 'London', 100, 1001);

INSERT INTO customer VALUES (202, 'Giovanne', 'Roe', 200, 1003);

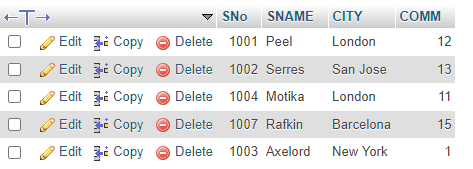
INSERT INTO customer VALUES (203, 'Liu', 'San Jose', 300, 1002);

INSERT INTO customer VALUES (204, 'Grass', 'Barcelona', 100, 1002);

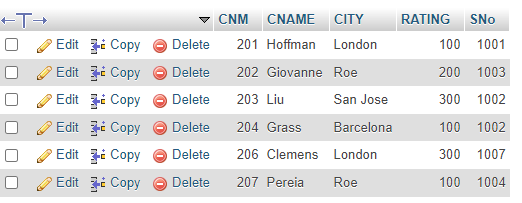
INSERT INTO customer VALUES (206, 'Clemens', 'London', 300, 1007);

INSERT INTO customer VALUES (207, 'Pereia', 'Roe', 100, 1004);

🡪 Salesperson Table :



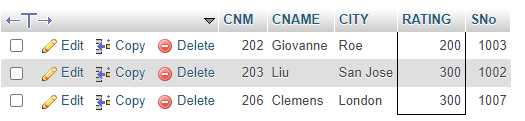
🡪 Customer Table :



**Que – 13). All orders for more than $100.**

Ans – 13)

SELECT \* FROM `customer` WHERE RATING > 100;



**Que – 14).Names and cities of all salespeople in London with commission above 0.12**

Ans – 14)

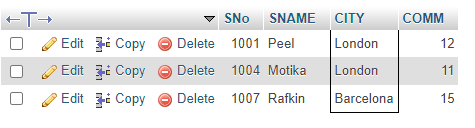
SELECT \* FROM `salesperson` WHERE CITY = 'London' AND COMM >= 12;



**Que – 15).All salespeople either in Barcelona or in London.**

Ans – 15)

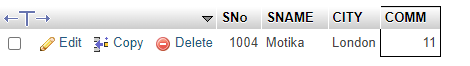
SELECT \* FROM `salesperson` WHERE CITY='Barcelona' OR CITY='London';



**Que – 16 ) . All salespeople with commission between 0.10 and 0.12. (Boundary valuesshould be excluded).**

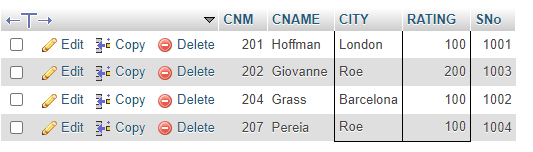
Ans – 16 )

SELECT \* FROM `salesperson` WHERE COMM > 10 and COMM < 12;



**Que – 17) . All customers excluding those with rating <= 100 unless they are located in Rome**

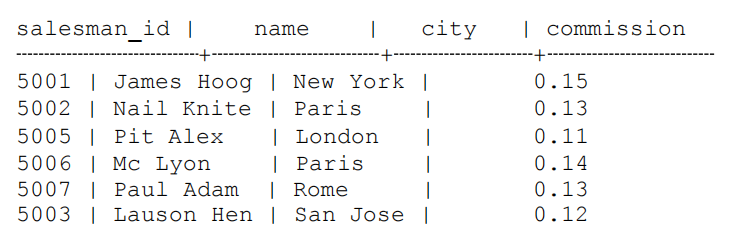
Ans – 17 ) SELECT \* FROM `customer` WHERE CITY = 'Roe' or RATING <=100;



**Que – 18 ).Write a SQL statement that displays all the information about all**

**Salespeople**

Ans – 18 )



CREATE TABLE Salespeople (

salesman\_id int,

name varchar(20),

city varchar(20),

commision int);

INSERT INTO salespeople VALUES

( 5001, 'James Hoog', 'New York', 0.15 ),

( 5002, 'Nail Knite', 'Paris', 0.13),

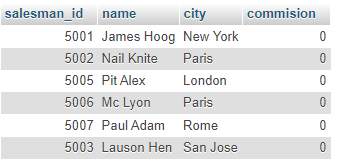
( 5005, 'Pit Alex', 'London', 0.11),

( 5006, 'Mc Lyon', 'Paris', 0.14),

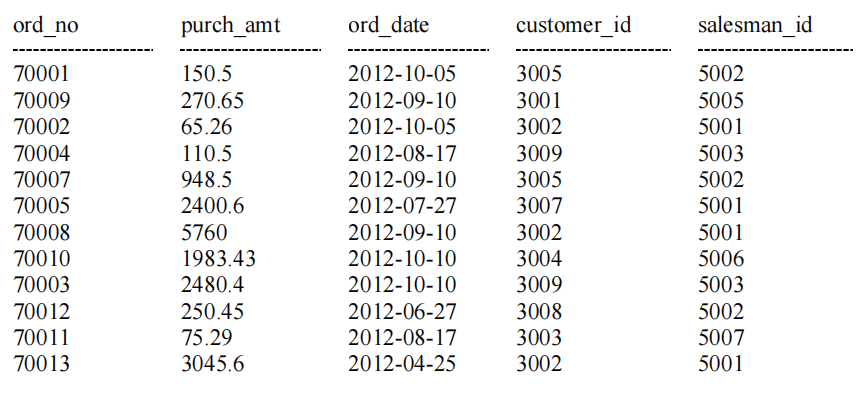
( 5007, 'Paul Adam', 'Rome', 0.13),

( 5003, 'Lauson Hen', 'San Jose', 0.12);

SELECT \* FROM `salespeople`;



**Que – 19).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.**



Ans – 19 )

CREATE TABLE orders (

ord\_no INT PRIMARY KEY,

purch\_amt DECIMAL(10,2),

ord\_date DATE,

customer\_id INT,

salesman\_id INT

);

INSERT INTO orders VALUES

(70001, 150.50, '2012-10-05', 3005, 5002),

(70009, 270.65, '2012-09-10', 3001, 5005),

(70002, 65.26, '2012-10-05', 3002, 5001),

(70004, 110.50, '2012-08-17', 3009, 5003),

(70007, 948.50, '2012-09-10', 3005, 5002),

(70005, 2400.60, '2012-07-27', 3007, 5001),

(70008, 5760.00, '2012-09-10', 3002, 5001),

(70010, 1983.43, '2012-10-10', 3004, 5006),

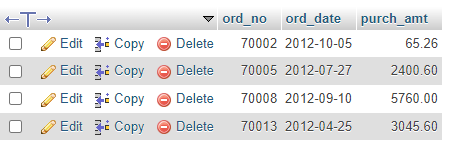
(70003, 2480.40, '2012-10-10', 3009, 5003),

(70012, 250.45, '2012-06-27', 3008, 5002),

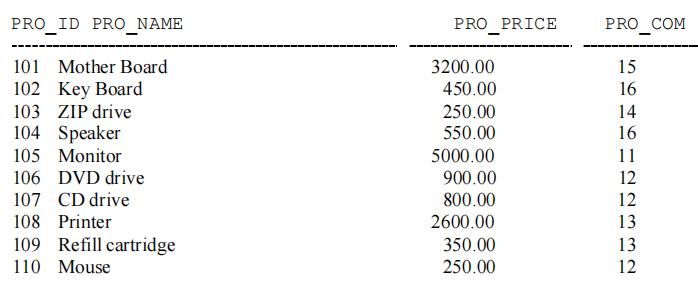
(70011, 75.29, '2012-08-17', 3009, 5007),

(70013, 3045.60, '2012-04-25', 3002, 5001);

SELECT ord\_no, ord\_date, purch\_amt FROM orders WHERE salesman\_id = 5001;



**Que – 20).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.**



Ans – 20) CREATE TABLE Products (

PRO\_ID INT PRIMARY KEY,

PRO\_NAME VARCHAR(50),

PRO\_PRICE DECIMAL(10, 2),

PRO\_COM INT

);

INSERT INTO Products VALUES

(101, 'Mother Board', 3200.00, 15),

(102, 'Key Board', 450.00, 16),

(103, 'ZIP drive', 250.00, 14),

(104, 'Speaker', 550.00, 16),

(105, 'Monitor', 5000.00, 11),

(106, 'DVD drive', 900.00, 12),

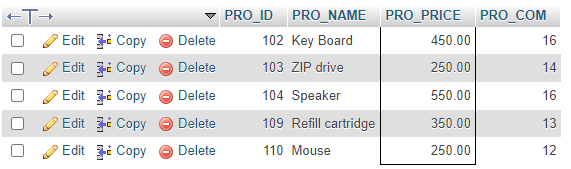
(107, 'CD drive', 800.00, 12),

(108, 'Printer', 2600.00, 13),

(109, 'Refill cartridge', 350.00, 13),

(110, 'Mouse', 250.00, 12);

SELECT \* FROM `products` WHERE PRO\_PRICE BETWEEN 200 AND 600;



**Que – 21) .From the following table, write a SQL query to calculate the averageprice for a manufacturer code of 16. Return avg.**

Ans – 21 )

CREATE TABLE Products (

PRO\_ID INT PRIMARY KEY,

PRO\_NAME VARCHAR(50),

PRO\_PRICE DECIMAL(10, 2),

PRO\_COM INT

);

INSERT INTO Products VALUES

(101, 'Mother Board', 3200.00, 15),

(102, 'Key Board', 450.00, 16),

(103, 'ZIP drive', 250.00, 14),

(104, 'Speaker', 550.00, 16),

(105, 'Monitor', 5000.00, 11),

(106, 'DVD drive', 900.00, 12),

(107, 'CD drive', 800.00, 12),

(108, 'Printer', 2600.00, 13),

(109, 'Refill cartridge', 350.00, 13),

(110, 'Mouse', 250.00, 12);

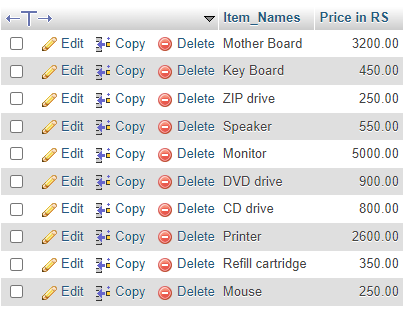
SELECT AVG(PRO\_PRICE) AS avg\_price FROM product WHERE PRO\_COM = 16;



**Que – 22) .From the following table, write a SQL query to display the pro\_nameas 'Item Name' and pro\_priceas 'Price in Rs.'**

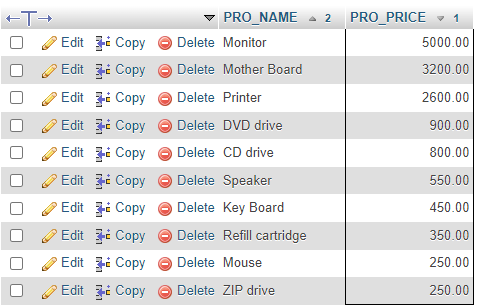
***Sample table*: item\_mast**

Ans – 22 )SELECT PRO\_NAME AS 'Item\_Names', PRO\_PRICE AS 'Price in RS' FROM product;



**Que - 23.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.**

Ans – 23)SELECT PRO\_NAME , PRO\_PRICE FROM product WHERE PRO\_PRICE >= 250 ORDER by PRO\_PRICE DESC, PRO\_NAME ASC;



**Que – 24).From the following table, write a SQL query to calculate average price ofthe items for each company. Return average price and companycode.**

Ans – 24 ) SELECT AVG(PRO\_PRICE) AS average\_price, PRO\_COM AS companycode FROM product GROUP by PRO\_COM;

