**Q-3) Create table below : salesperson and customer:**

Query:-

1. *Creating table salesperson:*

create table salesperson (s\_no int primary key, s\_name varchar(50), city varchar(50), commision decimal(10,2));

1. *Inserting data into table salesperson*:

INSERT INTO `salesperson`(`s\_no`, `s\_name`, `city`, `commision`)

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', '0.10');



1. *Creating table customer*:

create table customer (c\_nm int primary key, c\_name varchar(50), city varchar(50),

rating int, s\_no int, foreign key(s\_no) REFERENCES salesperson(s\_no));

1. *Inserting data into table customer*:

INSERT INTO `customer`(`c\_nm`, `c\_name`, `city`, `rating`, `s\_no`)

VALUES

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

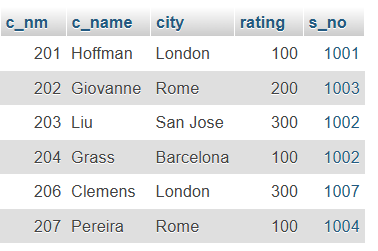
('202', 'Giovanne', 'Rome', '200', '1003'),

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

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

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

('207', 'Pereira', 'Rome', '100', '1004');



1. *Names and cities of all salespeople in London with commission above 0.12*:

Query:-

Select s\_name, city from salesperson where city = ‘London’ and commission > 0.12;

--no output because there is no data where city is London and commission is above 0.12

1. *All salespeople either in Barcelona or in London*:

Query:-

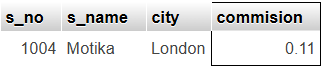
SELECT \* FROM `salesperson` WHERE city = 'Barcelona' or city = 'London';



1. *All salespeople with commission between 0.10 and 0.12*:

Query:-

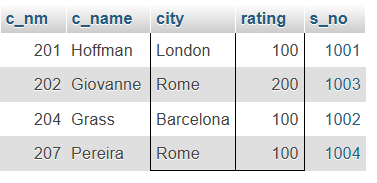
SELECT \* FROM `salesperson` WHERE commission > 0.10 and commision < 0.12;



1. *All customers whose rating <= 100 unless they are located in Rome*:

Query:-

SELECT \* FROM `customer` WHERE rating <= 100 or city = 'Rome';



**Q-4) Write a SQL statement that displays all the information about all salespeople:**

Query:-

1. *Creating the table salesman*:

create table salesman (salesman\_id int, name varchar(50), city varchar(50), commission decimal(4,2));

1. *Inserting values in table salesman*:

INSERT INTO salesman (salesman\_id, name, city, commission)

VALUES

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

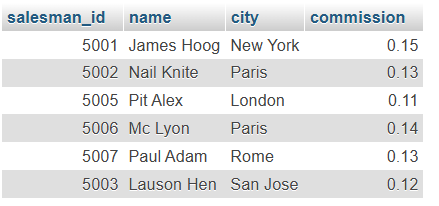
(5002, 'Nail Knite', 'Paris', 0.13),

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

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

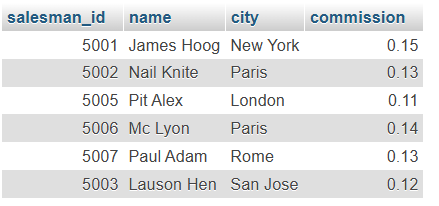
(5007, 'Paul Adam', 'Rome', 0.13),

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



1. *Displaying all the information about all sales people*:

Select \* from salesman;



**Q-5) 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:-

1. *Creating the sample table order*:

CREATE TABLE orders (ord\_no INT, purch\_amt DECIMAL(10, 2), ord\_date DATE, customer\_id INT, salesman\_id INT);

1. *Inserting values into table orders*:

INSERT INTO orders (ord\_no, purch\_amt, ord\_date, customer\_id, salesman\_id)

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),

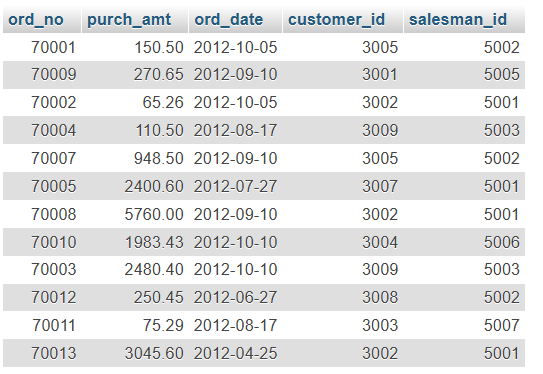
(70010, 1983.43, '2012-10-10', 3004, 5006),

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

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

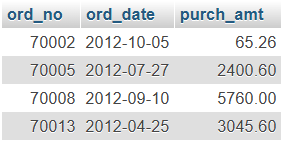
(70011, 75.29, '2012-08-17', 3003, 5007),

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



1. *Finding orders that are delivered by the salesman with id 5001, returning only ord\_date, ord\_no, purch\_amt*:

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



**Q-6) 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.**

Query:-

1. *Creating table item\_mast*;

CREATE TABLE item\_mast (pro\_id INT, pro\_name VARCHAR(50), pro\_price DECIMAL(10, 2), pro\_com INT);

1. *Inserting data into table item\_mast*:

INSERT INTO item\_mast (pro\_id, pro\_name, pro\_price, pro\_com)

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),

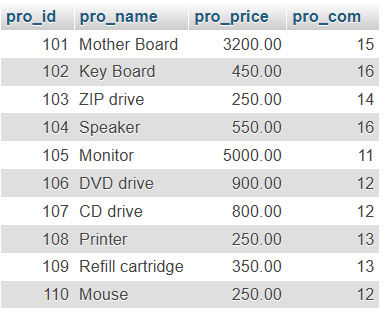
(106, 'DVD drive', 900.00, 12),

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

(108, 'Printer', 250.00, 13),

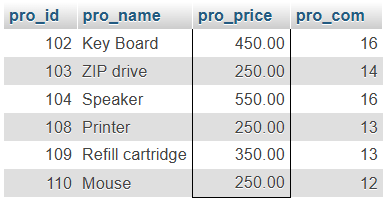
(109, 'Refill cartridge', 350.00, 13),

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



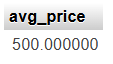
1. *select a range of products whose price is in the range Rs.200 to Rs.600. Begin and end values are included. Return pro\_id, pro\_name, pro\_price, and pro\_com*:

SELECT pro\_id, pro\_name, pro\_price, pro\_com FROM item\_mast WHERE pro\_price BETWEEN 200 AND 600;



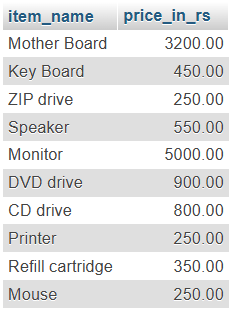
1. *From the following table, write a SQL query to calculate the average price for a manufacturer code of 16. Return avg*:

SELECT avg(pro\_price) as avg\_price FROM `item\_mast` WHERE pro\_com = 16;



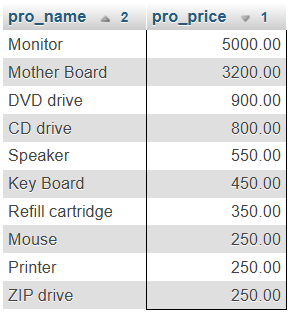
1. *From the following table, write a SQL query to display the pro\_name as 'Item Name' and pro\_priceas 'Price in Rs.'*:

SELECT pro\_name as item\_name, pro\_price as price\_in\_rs from item\_mast;



1. *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*:

SELECT pro\_name, pro\_price FROM `item\_mast` WHERE pro\_price >= 250 order by pro\_price desc, pro\_name asc;



1. *From the following table, write a SQL query to calculate average price of the items for each company. Return average price and company code*:

SELECT pro\_com, avg(pro\_price) as avg\_price FROM `item\_mast` group by pro\_com;

