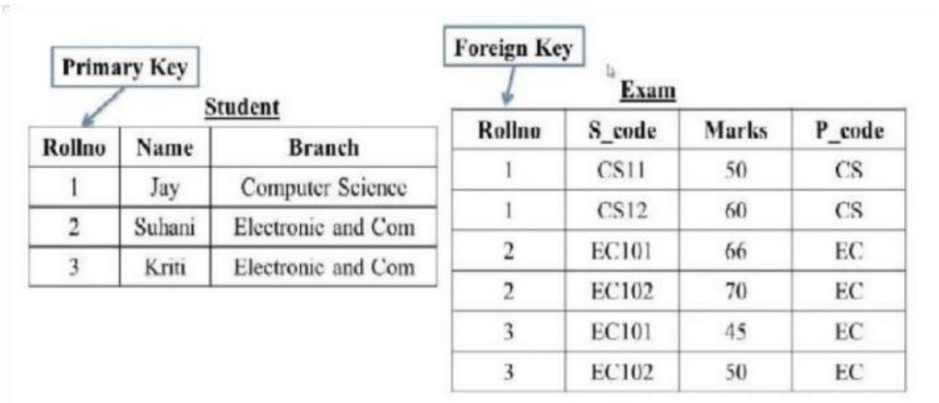
ASSIGNMENT_M5

Q.1 **Create Table Name : Student and Exam**


Primary Key
Foreign Key

Student

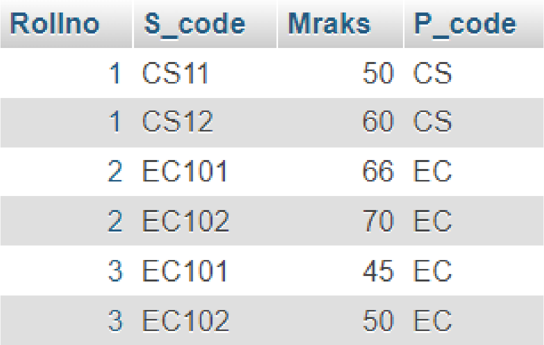
Rollno	Name	Branch
1	Jay	Computer Science
2	Suhani	Electronic and Com
3	Kriti	Electronic and Com

Exam

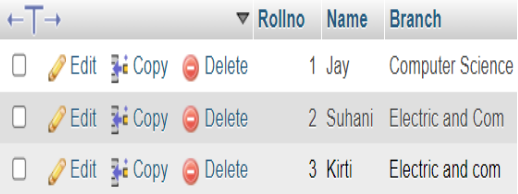
Rollno	S_code	Marks	P_code
1	CS11	50	CS
1	CS12	60	CS
2	EC101	66	EC
2	EC102	70	EC
3	EC101	45	EC
3	EC102	50	EC

```
CREATE TABLE student

(

    Rollno int,

    Name varchar(20),

    Branch varchar(20),

    PRIMARY KEY(Rollno)

    )

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

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

INSERT INTO student VALUES(3,'Kirti','Electric and com');

CREATE TABLE Exam

(

    Rollno int,

    S_code varchar(20),

    Mraks int,

    P_code varchar(20),

    foreign key(Rollno) REFERENCES student(Rollno)

);
```

```
INSERT INTO  exam VALUES (1,'CS11',50,'CS');

INSERT INTO  exam VALUES (1,'CS12',60,'CS');

INSERT INTO  exam VALUES (2,'EC101',66,'EC');

INSERT INTO  exam VALUES (2,'EC102',70,'EC');

INSERT INTO  exam VALUES (3,'EC101',45,'EC');

INSERT INTO  exam VALUES (3,'EC102',50,'EC');
```

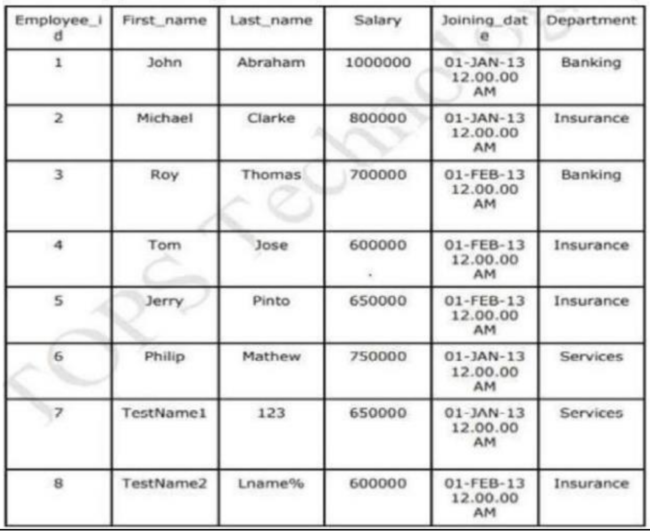


Rollno	S_code	Mraks	P_code
1	CS11	50	CS
1	CS12	60	CS
2	EC101	66	EC
2	EC102	70	EC
3	EC101	45	EC
3	EC102	50	EC

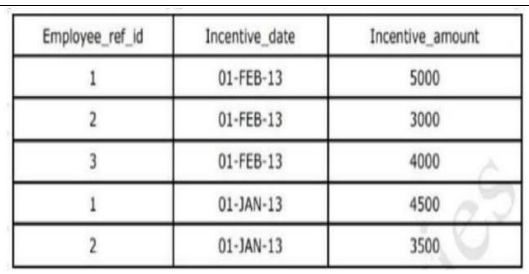


←T→	▼ Rollno	Name	Branch
□ Edit Copy Delete	1	Jay	Computer Science
□ Edit Copy Delete	2	Suhani	Electric and Com
□ Edit Copy Delete	3	Kirti	Electric and com

Q.2 **Create table given below: Employee and IncentiveTable**



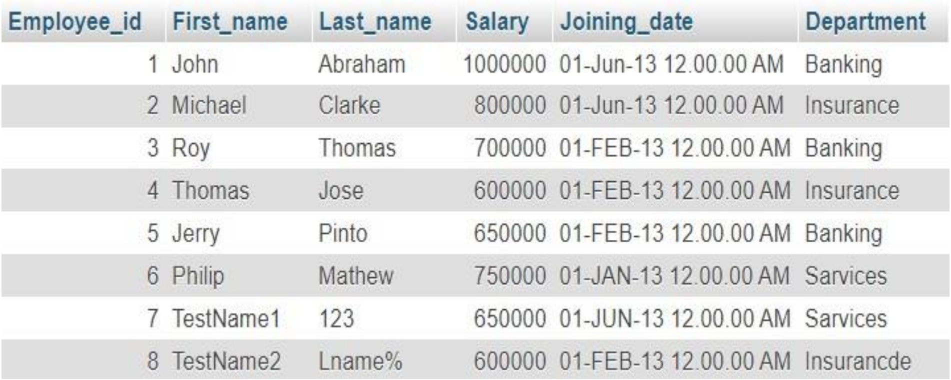
Employee_id	First_name	Last_name	Salary	Joining_date	Department
1	John	Abraham	1000000	01-JAN-13 12.00.00 AM	Banking
2	Michael	Clarke	800000	01-JAN-13 12.00.00 AM	Insurance
3	Roy	Thomas	700000	01-FEB-13 12.00.00 AM	Banking
4	Tom	Jose	600000	01-FEB-13 12.00.00 AM	Insurance
5	Jerry	Pinto	650000	01-FEB-13 12.00.00 AM	Insurance
6	Philip	Mathew	750000	01-JAN-13 12.00.00 AM	Services
7	TestName1	123	650000	01-JAN-13 12.00.00 AM	Services
8	TestName2	Lname%	600000	01-FEB-13 12.00.00 AM	Insurance



Employee_ref_id	Incentive_date	Incentive_amount
1	01-FEB-13	5000
2	01-FEB-13	3000
3	01-FEB-13	4000
1	01-JAN-13	4500
2	01-JAN-13	3500

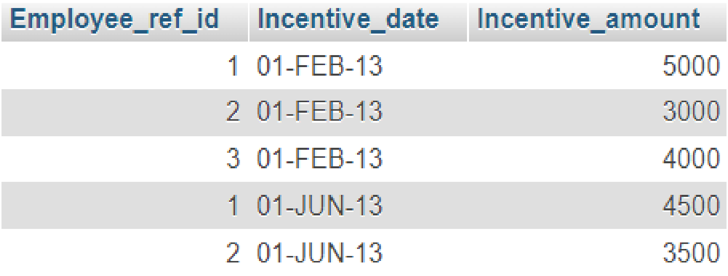
Employee **Incentive**

```
create table employee

(

Employee_id int,

First_name varchar(20),

Last_name varchar(20),

Salary int,

joining_date varchar(30),

department varchar(20)

)

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

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

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

INSERT INTO employee VALUES(4,'Thomas','Jose','600000','01-FEB-13 12.00.00 AM','Insurance');

INSERT INTO employee VALUES(5,'Jerry','Pinto','650000','01-FEB-13 12.00.00 AM','Banking');

INSERT INTO employee VALUES(6,'Philip','Mathew','750000','01-JAN-13 12.00.00 AM','Sarvices');
INSERT INTO employee VALUES(7,'TestName1','123','650000','01-JUN-13 12.00.00 AM','Sarvices');

INSERT INTO employee VALUES(8,'TestName2','Lname%','600000','01-FEB-13 12.00.00 AM','Insurancde');
```



Employee_id	First_name	Last_name	Salary	Joining_date	Department
1	John	Abraham	1000000	01-Jun-13 12.00.00 AM	Banking
2	Michael	Clarke	800000	01-Jun-13 12.00.00 AM	Insurance
3	Roy	Thomas	700000	01-FEB-13 12.00.00 AM	Banking
4	Thomas	Jose	600000	01-FEB-13 12.00.00 AM	Insurance
5	Jerry	Pinto	650000	01-FEB-13 12.00.00 AM	Banking
6	Philip	Mathew	750000	01-JAN-13 12.00.00 AM	Sarvices
7	TestName1	123	650000	01-JUN-13 12.00.00 AM	Sarvices
8	TestName2	Lname%	600000	01-FEB-13 12.00.00 AM	Insurancde

```
create table Incentive

(

    Employee_ref_id int,

    Incentive_date varchar(30),

    Incentive_amount int

    )

INSERT INTO incentive VALUES(1,'01-FEB-13','5000');

INSERT INTO incentive VALUES(2,'01-FEB-13','3000');

INSERT INTO incentive VALUES(3,'01-FEB-13','4000');

INSERT INTO incentive VALUES(1,'01-JUN-13','4500');

INSERT INTO incentive VALUES(2,'01-JUN-13','3500');
```

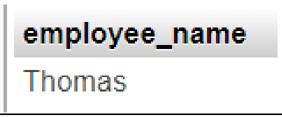


Employee_ref_id	Incentive_date	Incentive_amount
1	01-FEB-13	5000
2	01-FEB-13	3000
3	01-FEB-13	4000
1	01-JUN-13	4500
2	01-JUN-13	3500

ASSIGNMENT_M5

Q.3 **Get First_Name from employee table using Tom name “Employee Name”.**

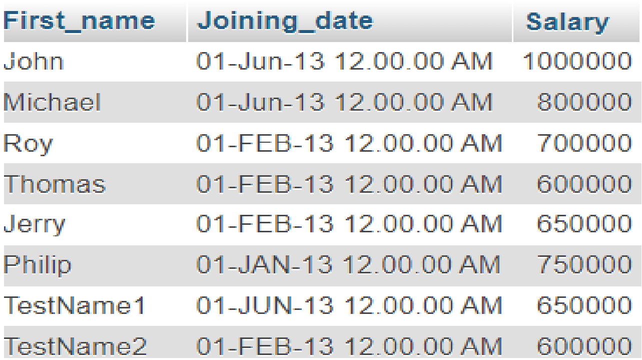
SELECT First_name AS employee_name FROM employee WHERE First_name='thomas';



employee_name
Thomas

Q.4 **Get FIRST_NAME, Joining Date, and Salary from employee table.**

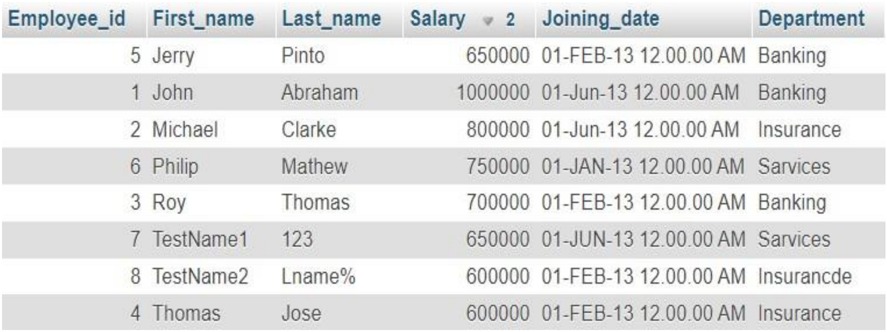
SELECT First_name,Joining_date,Salary FROM employee;



First_name	Joining_date	Salary
John	01-Jun-13 12.00.00 AM	1000000
Michael	01-Jun-13 12.00.00 AM	800000
Roy	01-FEB-13 12.00.00 AM	700000
Thomas	01-FEB-13 12.00.00 AM	600000
Jerry	01-FEB-13 12.00.00 AM	650000
Philip	01-JAN-13 12.00.00 AM	750000
TestName1	01-JUN-13 12.00.00 AM	650000
TestName2	01-FEB-13 12.00.00 AM	600000

Q.5 **Get all employee details from the employee table order by First_NameAscending and Salary descending?**

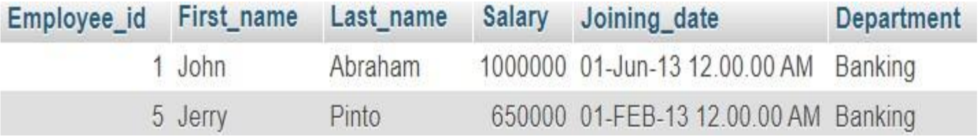
SELECT *FROM employee ORDER BY First_Name;



Employee_id	First_name	Last_name	Salary ▼ 2	Joining_date	Department
5	Jerry	Pinto	650000	01-FEB-13 12.00.00 AM	Banking
1	John	Abraham	1000000	01-Jun-13 12.00.00 AM	Banking
2	Michael	Clarke	800000	01-Jun-13 12.00.00 AM	Insurance
6	Philip	Mathew	750000	01-JAN-13 12.00.00 AM	Sarvices
3	Roy	Thomas	700000	01-FEB-13 12.00.00 AM	Banking
7	TestName1	123	650000	01-JUN-13 12.00.00 AM	Sarvices
8	TestName2	Lname%	600000	01-FEB-13 12.00.00 AM	Insurancde
4	Thomas	Jose	600000	01-FEB-13 12.00.00 AM	Insurance

Q.6 **Get employee details from employee table whose first name contains ‘J’.**

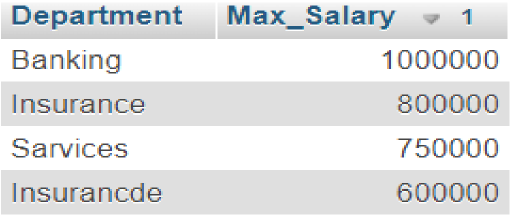
SELECT *FROM employee WHERE First_Name LIKE '%J%';



Employee_id	First_name	Last_name	Salary	Joining_date	Department
1	John	Abraham	1000000	01-Jun-13 12.00.00 AM	Banking
5	Jerry	Pinto	650000	01-FEB-13 12.00.00 AM	Banking

Q.7 **Get department wise maximum salary from employee table order by**

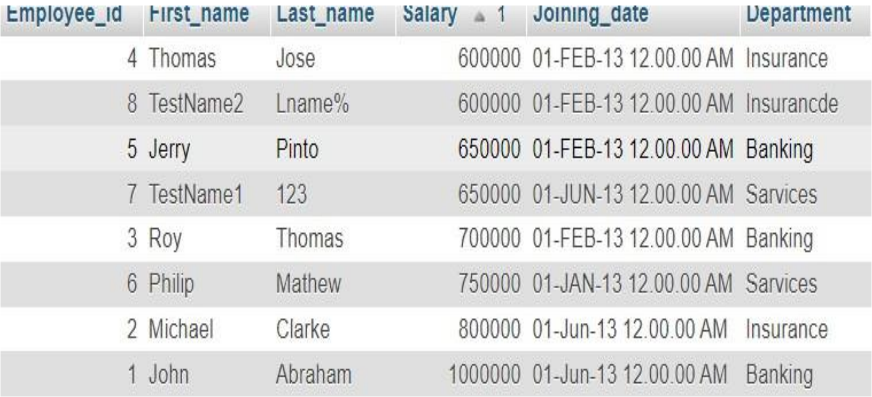
SELECT Department, MAX(Salary) AS Max_Salary FROM employee GROUP BY Department ORDER BY Max_Salary DESC;



Department	Max_Salary ▼ 1
Banking	1000000
Insurance	800000
Sarvices	750000
Insurancde	600000

Q.8 **salaryascending?**

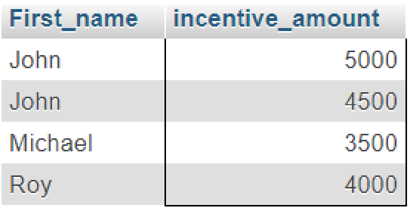
SELECT *FROM employee ORDER BY Salary ASC;



Employee_id	First_name	Last_name	Salary ▲ 1	Joining_date	Department
4	Thomas	Jose	600000	01-FEB-13 12.00.00 AM	Insurance
8	TestName2	Lname%	600000	01-FEB-13 12.00.00 AM	Insurancde
5	Jerry	Pinto	650000	01-FEB-13 12.00.00 AM	Banking
7	TestName1	123	650000	01-JUN-13 12.00.00 AM	Sarvices
3	Roy	Thomas	700000	01-FEB-13 12.00.00 AM	Banking
6	Philip	Mathew	750000	01-JAN-13 12.00.00 AM	Sarvices
2	Michael	Clarke	800000	01-Jun-13 12.00.00 AM	Insurance
1	John	Abraham	1000000	01-Jun-13 12.00.00 AM	Banking

Q.9 **Select first_name, incentive amount from employee and incentives table forthose employees who have incentives and incentive amount greater than 3000**

SELECT e.First_name,i.incentive_amount FROM employee e JOIN incentive i ON e.Employee_id=i.Employee_ref_id WHERE i.incentive_amount>3000;



First_name	incentive_amount
John	5000
John	4500
Michael	3500
Roy	4000

Q.10 **Create After Insert trigger on Employee table which insert records in view table**

```
create TRIGGER AfterinsertEmployee After INSERT ON Employee for each ROW
BEGIN
   INSERT INTO viewtable
(Employee_id,First_name,Last_name,Salary,Joining_date,Department)
   VALUES
(new.Employee_id,new.First_name,new.Last_name,new.salary,new.Joining_Date,new.Department)
END;
```

Q.11 **Create table given below: Salesperson and Customer**

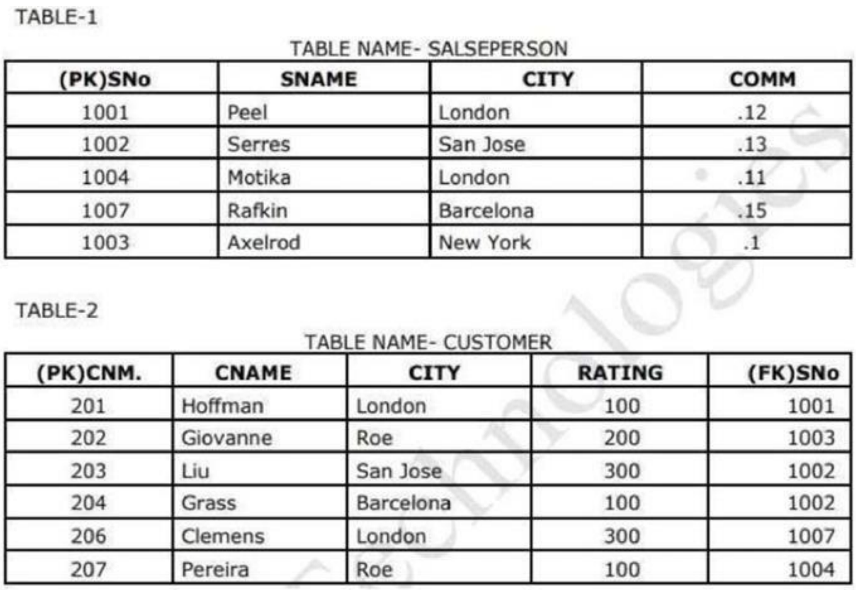


TABLE-1

TABLE NAME- SALSEPERSON

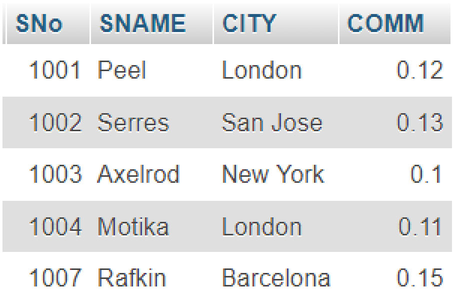
(PK)SNo	SNAME	CITY	COMM
1001	Peel	London	.12
1002	Serres	San Jose	.13
1004	Motika	London	.11
1007	Rafkin	Barcelona	.15
1003	Axelrod	New York	.1

TABLE-2

TABLE NAME- CUSTOMER

(PK)CNM.	CNAME	CITY	RATING	(FK)SNo
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

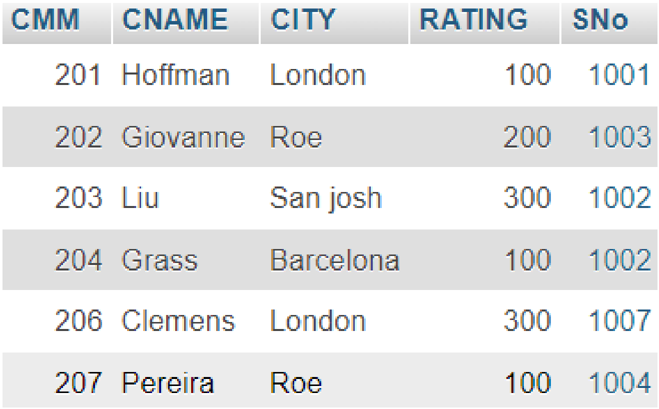
```
CREATE TABLE Salesperson

(

   SNo int,
    SNAME varchar(20),

    CITY Varchar(20),

    COMM float,

    PRIMARY KEY(SNo)

    )

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,'Axelrod','New York','.1');
```



SNo	SNAME	CITY	COMM
1001	Peel	London	0.12
1002	Serres	San Jose	0.13
1003	Axelrod	New York	0.1
1004	Motika	London	0.11
1007	Rafkin	Barcelona	0.15

```
CREATE TABLE customer

(

    CMM INT,

    CNAME varchar(20),

    CITY varchar(20),

    RATING INT,

    SNo int,

    PRIMARY KEY(CMM),

    FOREIGN KEY(SNo) REFERENCES salesperson(SNo)

    );
```

```
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 josh',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,'Pereira','Roe',100,1004);
```



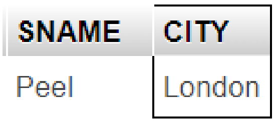
CMM	CNAME	CITY	RATING	SNo
201	Hoffman	London	100	1001
202	Giovanne	Roe	200	1003
203	Liu	San josh	300	1002
204	Grass	Barcelona	100	1002
206	Clemens	London	300	1007
207	Pereira	Roe	100	1004

Q.12 **Retrieve the below data from above table**

Q.13 **All orders for more than $1000.**

```
SELECT * FROM SALSEPERSON WHERE order=1000;
```

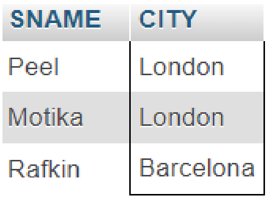
Q.14 **Names and cities of all salespeople in London with commission above 0.12**



SNAME	CITY
Peel	London

Q.15 **All salespeople either in Barcelona or in London**

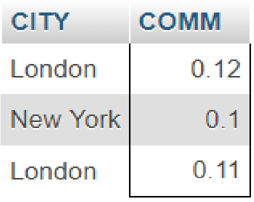
```
SELECT SNAME, CITY FROM salesperson WHERE CITY='Barcelona' OR CITY='London';
```



SNAME	CITY
Peel	London
Motika	London
Rafkin	Barcelona

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

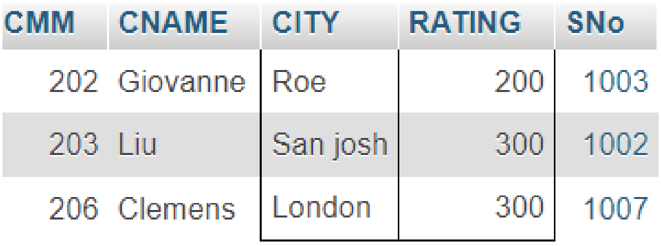
SELECT * FROM salesperson WHERE COMM >0.10 and COMM < 0.12;



CITY	COMM
London	0.12
New York	0.1
London	0.11

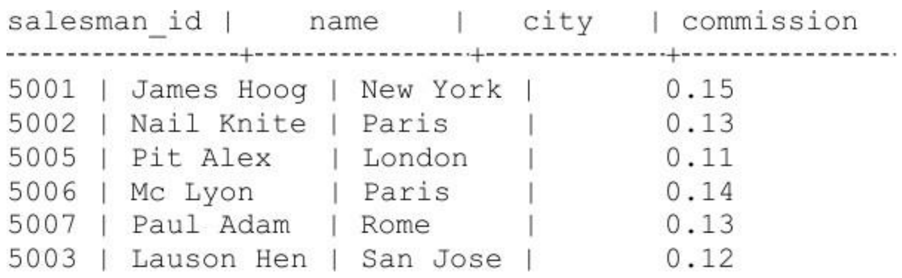
Q.17 **All customers excluding those with rating <= 100 unless they are located inRome**

SELECT * FROM customer WHERE RATING > 100 OR (RATING <= 100 AND CITY='ROME');



CMM	CNAME	CITY	RATING	SNo
202	Giovanne	Roe	200	1003
203	Liu	San josh	300	1002
206	Clemens	London	300	1007

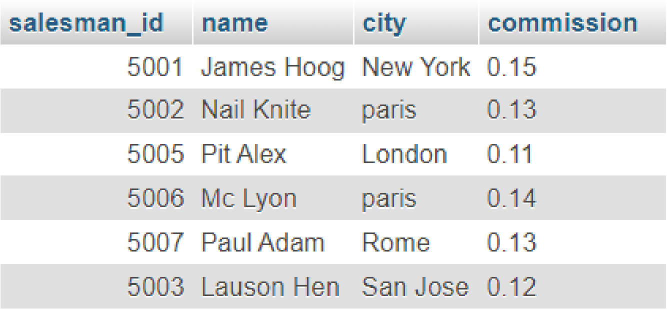
Q.18 **Write a SQL statement that displays all the information about all salespeople**



```
salesman_id |     name     |    city    | commission
------------+--------------+------------+------------
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
```

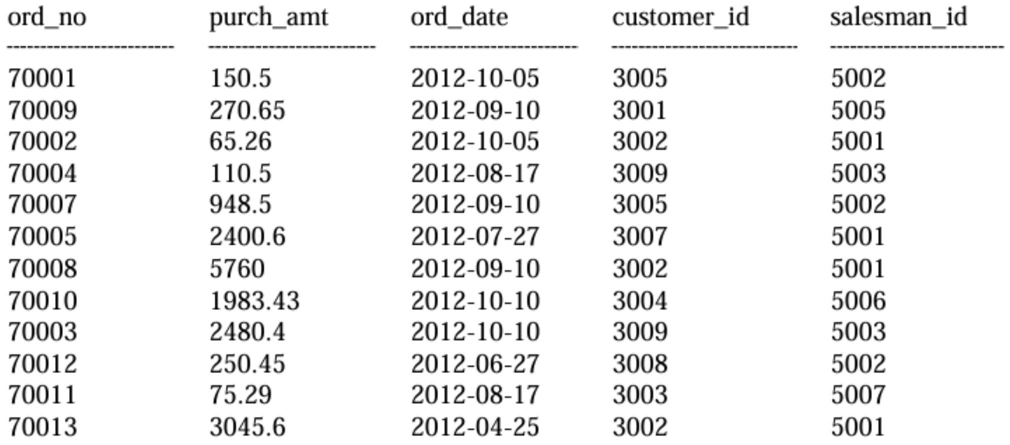
```
CREATE TABLE salespeople
(
 salesman_id int,
 name varchar(30),
 city text,
 commission VARCHAR(20)
 );
```

```
INSERT INTO salespeople VALUES(5001,'James Hoog','New York',0.15);
INSERT INTO salespeople VALUES(5002,'Nail Knite','paris',0.13);
INSERT INTO salespeople VALUES(5005,'Pit Alex','London',0.11);
INSERT INTO salespeople VALUES(5006,'Mc Lyon','paris',0.14);
INSERT INTO salespeople VALUES(5007,'Paul Adam','Rome',0.13);
 INSERT INTO salespeople VALUES(5003,'Lauson Hen','San Jose',0.12);
```



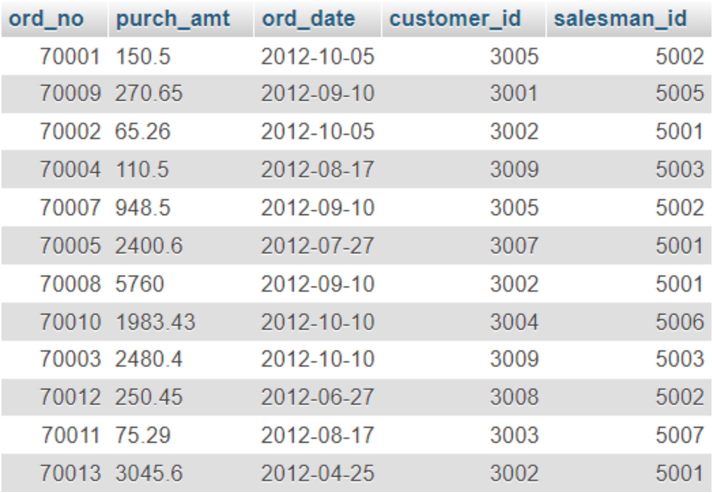
salesman_id	name	city	commission
5001	James Hoog	New York	0.15
5002	Nail Knite	paris	0.13
5005	Pit Alex	London	0.11
5006	Mc Lyon	paris	0.14
5007	Paul Adam	Rome	0.13
5003	Lauson Hen	San Jose	0.12

Q.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.**



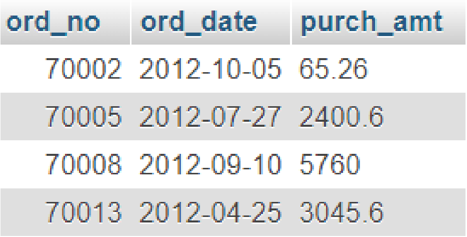
ord_no	purch_amt	ord_date	customer_id	salesman_id
70001	150.5	2012-10-05	3005	5002
70009	270.65	2012-09-10	3001	5005
70002	65.26	2012-10-05	3002	5001
70004	110.5	2012-08-17	3009	5003
70007	948.5	2012-09-10	3005	5002
70005	2400.6	2012-07-27	3007	5001
70008	5760	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.4	2012-10-10	3009	5003
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007
70013	3045.6	2012-04-25	3002	5001

```
REATE TABLE orders
(
  ord_no int,
purch_amt text,
ord_date date,
customer_id int,
salesman_id int
 );

INSERT INTO orders VALUES(70001,150.5,'2012-10-05',3005,5002);
INSERT INTO orders VALUES(70009,270.65,'2012-09-10',3001,5005);
INSERT INTO orders VALUES(70002,65.26,'2012-10-05',3002,5001);
INSERT INTO orders VALUES(70004,110.5,'2012-08-17',3009,5003);
INSERT INTO orders VALUES(70007,948.5,'2012-09-10',3005,5002);
INSERT INTO orders VALUES(70005,2400.6,'2012-07-27',3007,5001);
INSERT INTO orders VALUES(70008,5760,'2012-09-10',3002,5001);
INSERT INTO orders VALUES(70010,1983.43,'2012-10-10',3004,5006);
INSERT INTO orders VALUES(70003,2480.4,'2012-10-10',3009,5003);
INSERT INTO orders VALUES(70012,250.45,'2012-06-27',3008,5002);
INSERT INTO orders VALUES(70011,75.29,'2012-08-17',3003,5007);
INSERT INTO orders VALUES(70013,3045.6,'2012-04-25',3002,5001);
```



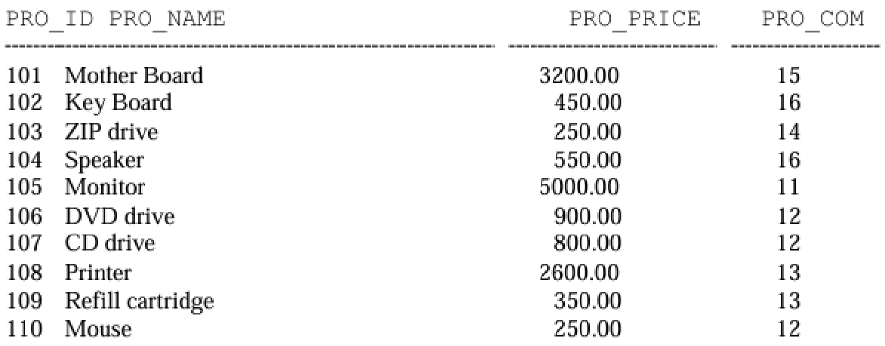
ord_no	purch_amt	ord_date	customer_id	salesman_id
70001	150.5	2012-10-05	3005	5002
70009	270.65	2012-09-10	3001	5005
70002	65.26	2012-10-05	3002	5001
70004	110.5	2012-08-17	3009	5003
70007	948.5	2012-09-10	3005	5002
70005	2400.6	2012-07-27	3007	5001
70008	5760	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.4	2012-10-10	3009	5003
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007
70013	3045.6	2012-04-25	3002	5001

```
SELECT ord_no,ord_date,purch_amt FROM orders WHERE salesman_id=5001;
```



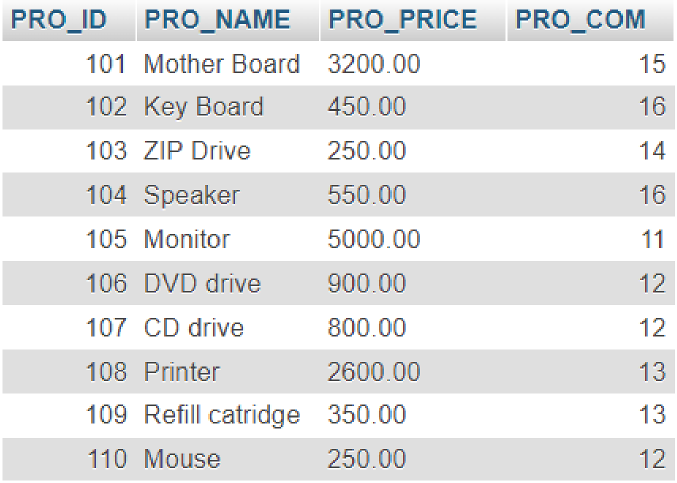
ord_no	ord_date	purch_amt
70002	2012-10-05	65.26
70005	2012-07-27	2400.6
70008	2012-09-10	5760
70013	2012-04-25	3045.6

Q.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.**



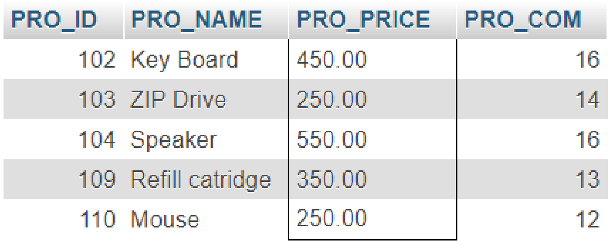
```
PRO_ID PRO_NAME                         PRO_PRICE    PRO_COM
-------------------------------------  -----------  ---------
101    Mother Board                      3200.00       15
102    Key Board                          450.00       16
103    ZIP drive                          250.00       14
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
```

INSERT INTO item_mast VALUES(101,'Mother Board',3200.00,15);
INSERT INTO item_mast VALUES(102,'Key Board',450.00,16);
INSERT INTO item_mast VALUES(103,'ZIP Drive',250.00,14);
INSERT INTO item_mast VALUES(104,'Speaker',550.00,16);
INSERT INTO item_mast VALUES(105,'Monitor',5000.00,11);
INSERT INTO item_mast VALUES(106,'DVD drive',900.00,12);
INSERT INTO item_mast VALUES(107,'CD drive',800.00,12);
INSERT INTO item_mast VALUES(108,'Printer',2600.00,13);
INSERT INTO item_mast VALUES(109,'Refill catridge',350.00,13);
INSERT INTO item_mast VALUES(110,'Mouse',250.00,12);



PRO_ID	PRO_NAME	PRO_PRICE	PRO_COM
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 catridge	350.00	13
110	Mouse	250.00	12

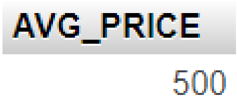
SELECT PRO_ID , PRO_NAME, PRO_PRICE ,PRO_COM from item_mast WHERE PRO_PRICE BETWEEN 200 and 600;



PRO_ID	PRO_NAME	PRO_PRICE	PRO_COM
102	Key Board	450.00	16
103	ZIP Drive	250.00	14
104	Speaker	550.00	16
109	Refill catridge	350.00	13
110	Mouse	250.00	12

Q.21 **From the following table, write a SQL query to calculate the average price for a manufacturer code of 16. Return avg.**

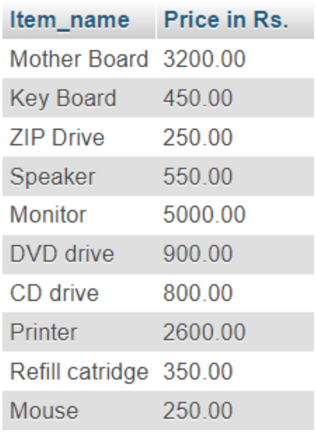
```
SELECT AVG(PRO_PRICE) AS AVG_PRICE FROM item_mast WHERE PRO_COM = 16;
```



AVG_PRICE
500

Q.22 **From the following table, write a SQL query to display the pro_name as 'Item Name' and pro_priceas 'Price in Rs.'**

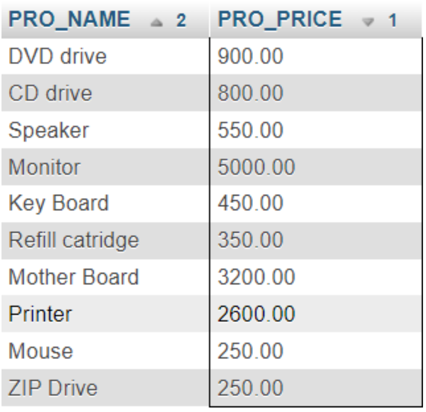
```
SELECT PRO_NAME AS "Item_name",PRO_PRICE AS "Price in Rs." FROM item_mast;
```



Item_name	Price in Rs.
Mother Board	3200.00
Key Board	450.00
ZIP Drive	250.00
Speaker	550.00
Monitor	5000.00
DVD drive	900.00
CD drive	800.00
Printer	2600.00
Refill catridge	350.00
Mouse	250.00

Q.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.**

```
SELECT PRO_NAME , PRO_PRICE FROM item_mast WHERE PRO_PRICE >= 250
ORDER BY PRO_PRICE  DESC,PRO_NAME ASC;
```

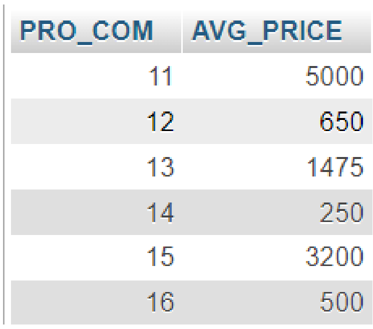


PRO_NAME ▲ 2	PRO_PRICE ▼ 1
DVD drive	900.00
CD drive	800.00
Speaker	550.00
Monitor	5000.00
Key Board	450.00
Refill catridge	350.00
Mother Board	3200.00
Printer	2600.00
Mouse	250.00
ZIP Drive	250.00

Q.24 **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
```



PRO_COM	AVG_PRICE
11	5000
12	650
13	1475
14	250
15	3200
16	500

Basic of database

1. What do you understand By Database

→ A database is a structured collection of data, stored and accessed electronically. Databases are used to store and manage large amounts of structured and unstructured data, and they can be used to support a wide range of activities, including data storage, data analysis, and data management.

2. What is Normalization?

→ Normalization is a process used in database design to organize tables and reduce data redundancy. Its primary goal is to structure data to minimize redundancy and dependency by dividing large tables into smaller, related tables and defining relationships between them. This process helps improve data integrity and efficiency in querying and updating the database.

3. What is Difference between DBMS and RDBMS?

DBMS	RDBMS
● database management system	● Relational database management system
● Allow one user at a time	● Allow more then one user at a time
● Data stored is a small quantity.	● Data is Stored in large amount.
● DBMS stores data as file	● RDBMS stores in table format
● Individual access of data element.	● Multiple data element is accessible together.

● No connection between data.	● Data in the form of a table are linked together

4. What is MF Cod Rule of RDBMS Systems?

→ The MF Cod Rule of RDBMS Systems states that for a system to qualify as an RDBMS, it must be able to manage database entirely through the relational capabilities Rule 0 of the MF Cod Rules states that the system must qualify as relational, as a database, and as a management system. For a system to qualify as an RDBMS, that system must use its relational facilities exclusively to manage the database.

5. What do you understand By Data Redundancy?

→ Data redundancy refers to the unnecessary duplication of data within a database or between databases. It occurs when the same piece of data is stored in multiple places unnecessarily. Redundancy can be useful for data recovery in case of corruption or loss. In computer memory and storage, data redundancy allows for error correction.

6. What is DDL Interpreter?

→ DML Compiler: It processes the DML statements into low level instruction , so that they can be executed. DDL Interpreter: It processes the DDL statements into a set of tables containing meta data.

7. What is DML Compiler in SQL?

→ the data Manipulation Language Represents a collection of programming languages explicitly used to make changes to the database. his generally entails inserting, editing, or deleting rows in SQL tables.

8. What is SQL Key Constraints writing an Example of SQL Key Constraints

→ Constraints are the rules that we can apply on the type of data in a table. That is, we can specify the limit on the type of data that can be stored in a particular column in a table using constraints.

- **Primary key:**
  A primary key uniquely identifies each record in a table. It must contain unique values and cannot have NULL values. Each table can have only one primary key constraint.
- **Foreign key:**
  A foreign key constraint establishes a relationship between two tables. It ensures referential integrity by enforcing a link between a column in one table and a column in another table.
- **Unique:**
  A unique constraint ensures that all values in a column (or a group of columns) are unique, but unlike the primary key, it can contain NULL values. It can be applied to one or more columns in a table.

9. What is save Point? How to create a save Point writes a Query?

⟶ A save point in SQL is a logical rollback point within a transaction. It allows you to specify a point in a transaction that you can roll back to without affecting the entire transaction.

⟶ To create a, save point, use the following syntax: 'SAVEPOINT savepoint_name'. You can then perform various SQL operations within the transaction.

⟶ To roll back to a specific save point, use 'ROLLBACK TO save_point_name'.

10. What is trigger and how to create a Trigger in SQL?

⟶ A trigger is a stored procedure in a database that automatikali invokes whenever a special event in the database occurs.

1. AFTER INSERT: after insert data is inserted into the table.
2. AFTER UPDATE: after update data in the table is modified.
3. AFTER DELETE: after delete data is deleted/removed from the table.
4. BEFORE INSERT: before insert data is inserted into the table.
5. BEFORE UPDATE: before update data in the table is modified.
6. BEFORE DELETE: before delete is deleted/removed from the table.