DATABASE

MODULES:-[5]

Submitted to :-

Mr. Jay Prajapati

Submitted by :-

Janvi Panchal

MODULE: 5 (DATABASE)

1. What do you understand By Database

Ans:-

DBMS (DataBase Mangment System)

Data is a collection of a distinct small unit of information. It can be used in a variety of forms like text, numbers, media, bytes, etc. it can be stored in pieces of paper or electronic memory, etc.

2.What is Normalization?

Ans:-

● Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
● Normalization divides the larger table into the smaller table and links them using relationship.
● The normal form is used to reduce redundancy from the database table.

• Types of Normal Forms

- **1NF**

A relation is in 1NF if it contains an atomic value.

- **2NF**

A relation will be in 2NF if it is in 1NF and all non-key attributes are fully functional dependent on the primary key.

- **3NF**

A relation will be in 3NF if it is in 2NF and no transition dependency exists..

3.What is Difference between DBMS and RDBMS?

Ans:-

Difference between DBMS and RDBMS

The following are the important differences between DBMS and RDBMS.

Key	DBMS	RDBMS
Definition	DBMS stands for Database Management System.	RDBMS stands for Relational Database Management System.
Data Storage	Data is stored as file.	Data is stored as tables.

Data Access	In DBMS, each data elements are to be accessed individually.	In RDBMS, multiple data elements can be accessed at same time.
Relationship	There is no relationship between data in DBMS.	Data is present in multiple tables which can be related to each other.
Normalization	Normalization cannot be achieved.	Normalization can be achieved.
Distributed database	DBMS has no support for distributed databases.	RDBMS supports distributed databases.
Data Quantity	DBMS deals with small quantity of data.	RDBMS deals with large quantity of data.
Data Redundancy	Data Redundancy is common in DBMS.	Data Redundancy can be reduced using key and indexes in RDBMS.
User	DBMS supports single user at a time.	RDBMS supports multiple users at a time.
Security	DBMS provides low security during data manipulation.	RDBMS has multilayer security during data manipulation.
Example	File systems, XML, etc.	Oracle, SQL Server.

4.What is EF Codd Rule of RDBMS Systems?

Ans:-

Dr Edgar F. Codd, after his extensive research on the Relational Model of database systems, came up with twelve rules of his own, which according to him, a database must obey in order to be regarded as a true relational database.

Rule 1: Information Rule

The data stored in a database, may it be user data or metadata, must be a value of some table cell. Everything in a database must be stored in a table format.

Rule 2: Guaranteed Access Rule

Every single data element (value) is guaranteed to be accessible logically with a combination of table-name, primary-key (row value), and attribute-name (column value). No other means, such as pointers, can be used to access data.

Rule 3: Systematic Treatment of NULL Values

The NULL values in a database must be given a systematic and uniform treatment. This is a very important rule because a NULL can be interpreted as one the following − data is missing, data is not known, or data is not applicable.

Rule 4: Active Online Catalog

The structure description of the entire database must be stored in an online catalog, known as data dictionary, which can be accessed by authorized users. Users can use the same query language to access the catalog which they use to access the database itself.

Rule 5: Comprehensive Data Sub-Language Rule

A database can only be accessed using a language having linear syntax that supports data definition, data manipulation, and transaction management operations. This language can be used directly or by means of some application. If the database allows access to data without any help of this language, then it is considered as a violation.

Rule 6: View Updating Rule

All the views of a database, which can theoretically be updated, must also be updatable by the system.

Rule 7: High-Level Insert, Update, and Delete Rule

A database must support high-level insertion, updation, and deletion. This must not be limited to a single row, that is, it must also support union, intersection and minus operations to yield sets of data records.

Rule 8: Physical Data Independence

The data stored in a database must be independent of the applications that access the database. Any change in the physical structure of a database must not have any impact on how the data is being accessed by external applications.

Rule 9: Logical Data Independence

The logical data in a database must be independent of its user's view (application). Any change in logical data must not affect the applications using it. For example, if two tables are merged or one is split into two different tables, there should be no impact or change on the user application. This is one of the most difficult rule to apply.

Rule 10: Integrity Independence

A database must be independent of the application that uses it. All its integrity constraints can be independently modified without the need of any change in the application. This rule makes a database independent of the front-end application and its interface.

Rule 11: Distribution Independence

The end-user must not be able to see that the data is distributed over various locations. Users should always get the impression that the data is located at one site only. This rule has been regarded as the foundation of distributed database systems.

Rule 12: Non-Subversion Rule

If a system has an interface that provides access to low-level records, then the interface must not be able to subvert the system and bypass security and integrity constraints.

5.What do you understand By Data Redundancy?

Ans:-

Data redundancy occurs when the same piece of data exists in multiple places, whereas data inconsistency is when the same data exists in different formats in multiple tables.

6.What is DDL Interpreter?

Ans:-

(Data Definition Language)

A data definition language (DDL) is a computer language used to create and modify the structure of database objects in a database.

For Example :-

Create

Alter

Rename

7.What is DML Compiler in SQL?

Ans:-

A DML (data manipulation language) refers to a computer programming language that allows you to add (insert), delete (delete), and alter (update) data in a database.

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

Ans:-

SQL Constraints

• terms that needs to be satisfied on data

• For ex all students must have unique roll number

• Can be defined as primary key, foreign key, unique key etc

• Primary key – column of table whose value can be used to uniquely identify records Foreign key – column inside table that is primary key of another table Web

• Unique key – like primary key can be used to uniquely identify a record • Difference between primary key and unique key is primary key will never allow null where as unique key will allow it for once.

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

Ans:-

A SAVEPOINT is a point in a transaction when you can roll the transaction back to a certain point without rolling back the entire transaction.

The Syntax for a SAVEPOINT command is as shown below.

```
SAVEPOINT SAVEPOINT_NAME;
```

This command serves only in the creation of a SAVEPOINT among all the transactional statements.

Query:-

```
 CREATE TABLE student(ID INT, Name VARCHAR(20),

 Percentage INT, Location VARCHAR(20), DateOfBirth DATE);



INSERT INTO student(ID, Name, Percentage, Location, DateOfBirth)
VALUES(1, "Manthan Koli", 79, "Delhi", "2003-08-20"),
         (2, "Dev Dixit", 75, "Pune", "1999-06-17"),
         (3, "Aakash Deshmukh", 87, "Mumbai", "1997-09-12"),
            (4, "Aaryan Jaiswal", 90, "Chennai", "2005-10-02"),
            (5, "Rahul Khanna", 92, "Ambala", "1996-03-04"),
            (6, "Pankaj Deshmukh", 67, "Kanpur", "2000-02-02"),
            (7, "Gaurav Kumar", 84, "Chandigarh", "1998-07-06"),
            (8, "Sanket Jain", 61, "Shimla", "1990-09-08"),
            (9, "Sahil Wagh", 90, "Kolkata", "1968-04-03");

SELECT *FROM student;

START TRANSACTION;

SAVEPOINT ini;

INSERT INTO student
VALUES (10, "Saurabh Singh", 54, "Kashmir", "1989-01-06");

UPDATE student SET Name = "Mahesh Kuwar" WHERE ID =1;

 SAVEPOINT upd;

 DELETE FROM student WHERE ID= 6;

  SELECT *FROM student;

  SAVEPOINT del;
```

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

Ans:-

A trigger is a Stored procedure in database which Automatically invokes whenever a special event in the database occurs.

For example,

A trigger can be invoked when a row is inserted into a specified table.

How to create a Trigger in SQL / Syntax:-

```
Create Trigger [Trigger_name] [Befor / After]

{INSERT| UPDATE| DELETE}

 On [Table_name] [For each rows]

       [Trigger_Body]
```

TASK

1. Create Table Name : Student and Exam

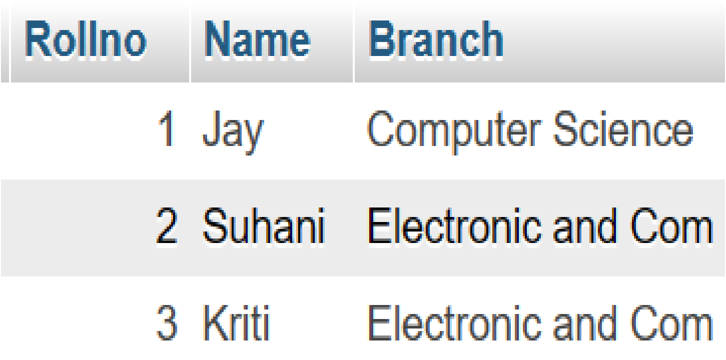
1) Student:-

Query:-

CREATE TABLE student(Rollno int(11) primary key, Name varchar(50),Branch varchar(50));

Primary key

STUDENT:-



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

2) Exam:-

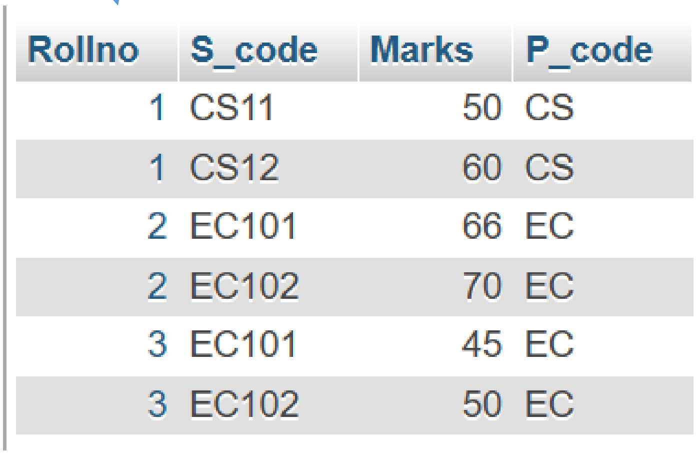
Query:-

```
CREATE TABLE exam(Rollno int(11), S_code varchar(255),
Marks int(11), P_code varchar(50),

FOREIGN KEY (Rollno) REFERENCES student(Rollno) ON
DELETE CASCADE ON UPDATE CASCADE);
```

Foreign key

EXAM:-



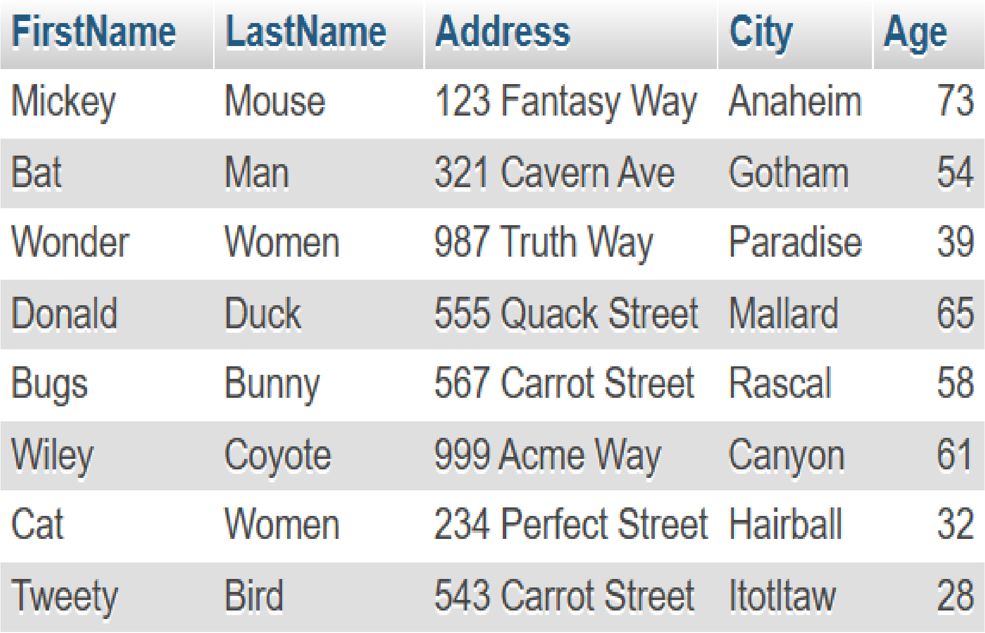
Rollno	S_code	Marks	P_code
1	CS11	50	CS
1	CS12	60	CS
2	EC101	66	EC
2	EC102	70	EC
3	EC101	45	EC
3	EC102	50	EC

2. Create table given below

Information:-

Query:-

```
INSERT Into info (FirstName,Lastname,Address,City,Age)
VALUES
('Mickey','Mouse','123 Fantasy Way','Anaheim',73),
('Bat','Man','321 Cavern Ave','Gotham',54),
('Wonder','Women','987 Truth Way','Paradise',39),
('Donald','Duck','555 Quack Street','Mallard',65),
('Bugs','Bunny','567 Carrot Street','Rascal',58),
('Wiley','Coyote','999 Acme Way','Canyon',61),
('Cat','Women','234 Perfect Street','Hairball',32),
('Tweety','Bird','543 Carrot Street','Itotltaw',28);
```



FirstName	LastName	Address	City	Age
Mickey	Mouse	123 Fantasy Way	Anaheim	73
Bat	Man	321 Cavern Ave	Gotham	54
Wonder	Women	987 Truth Way	Paradise	39
Donald	Duck	555 Quack Street	Mallard	65
Bugs	Bunny	567 Carrot Street	Rascal	58
Wiley	Coyote	999 Acme Way	Canyon	61
Cat	Women	234 Perfect Street	Hairball	32
Tweety	Bird	543 Carrot Street	Itotltaw	28

3. Create table given below: Employee and Incentive

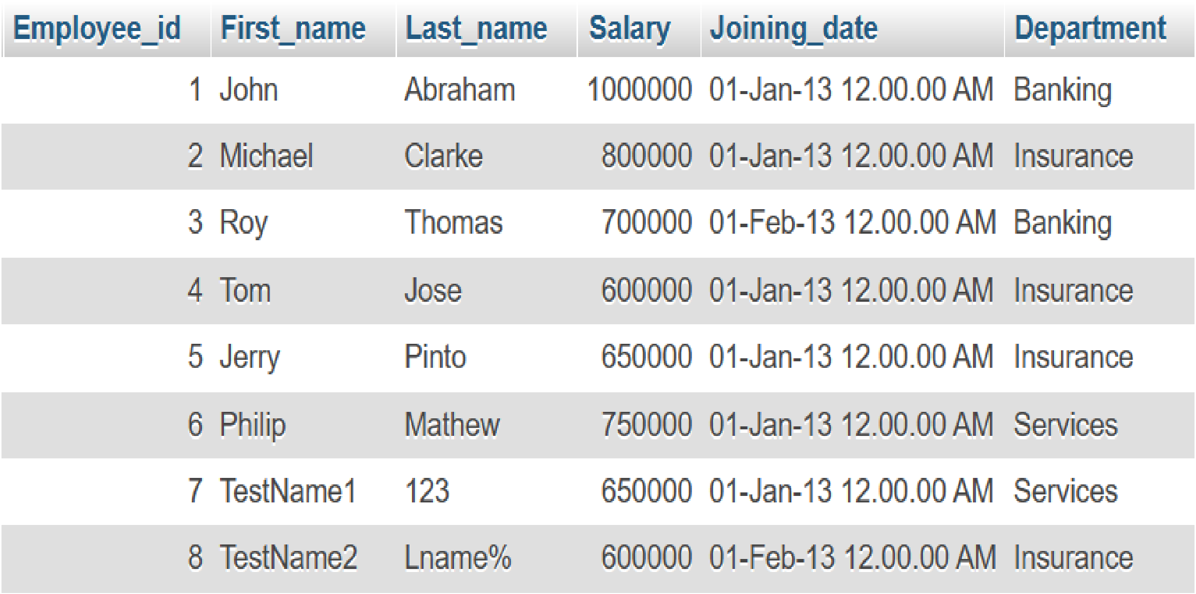
Table Name: Employee

Query:-

```
CREATE TABLE employee(
Employee_id int(11) PRIMARY KEY,
First_name varchar(50),
Last_name varchar(50),
Salary bigint(11),
Joining_date varchar(50),
Department varchar(50));


INSERT INTO

employee(Employee_id,First_name,Last_name,Salary,Joinin
g_date,Department) VALUES
(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-Jan-13 12.00.00
AM','Insurance'),
(5,'Jerry','Pinto',650000,'01-Jan-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_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-Jan-13 12.00.00 AM	Insurance
5	Jerry	Pinto	650000	01-Jan-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

Table Name: Incentive

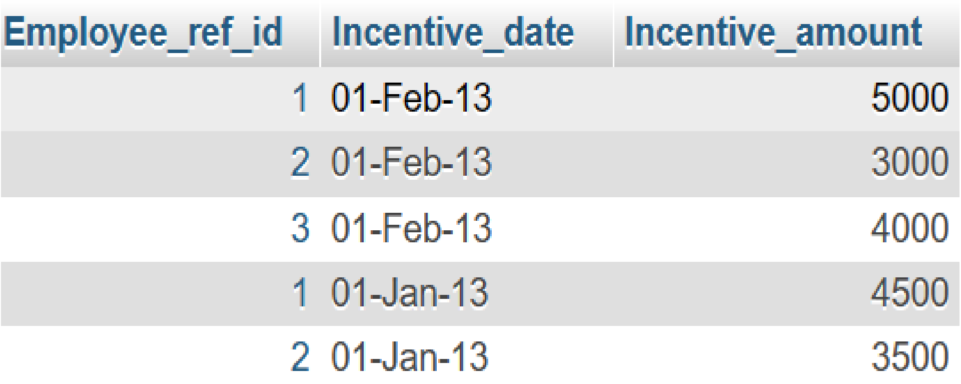
Query:-

CREATE TABLE incentive (Employee_ref_id int(11),Incentive_date varchar(50),Incentive_amount int(11),

FORIEGN KEY (Employee_ref_id) REFRENCES employee(Employee_id) ON DELETE CASCADE ON UPDATE CASCADE);

INSERT INTO incentive(Employee_ref_id,Incentive_date,Incentive_amount) VALUES
(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);
```



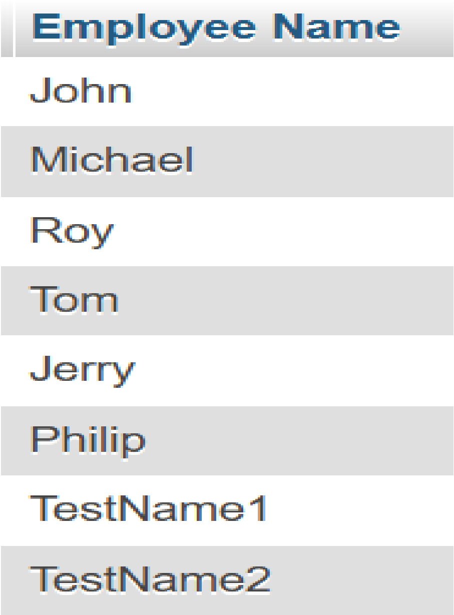
Employee_ref_id	Incentive_date	Incentive_amount
1	01-Feb-13	5000
2	01-Feb-13	3000
3	01-Feb-13	4000
1	01-Jan-13	4500
2	01-Jan-13	3500

Query:-

a) Get First_Name from employee table using Tom name “Employee Name”.

Ans:-

```
SELECT First_name "Employee Name" FROM
employee;
```

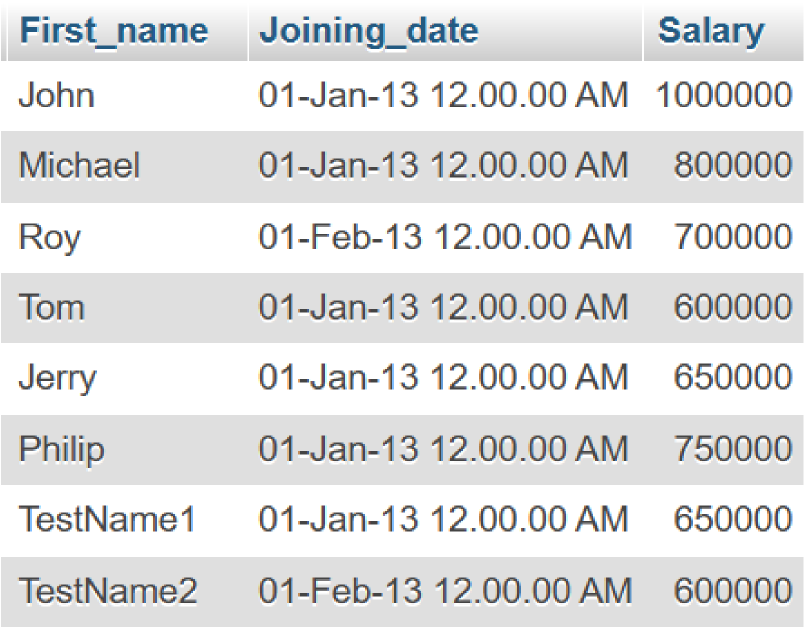


Employee Name
John
Michael
Roy
Tom
Jerry
Philip
TestName1
TestName2

b) Get FIRST_NAME, Joining Date, and Salary from employee table.

Ans:-

SELECT First_name,Joining_date,Salary FROM employee;

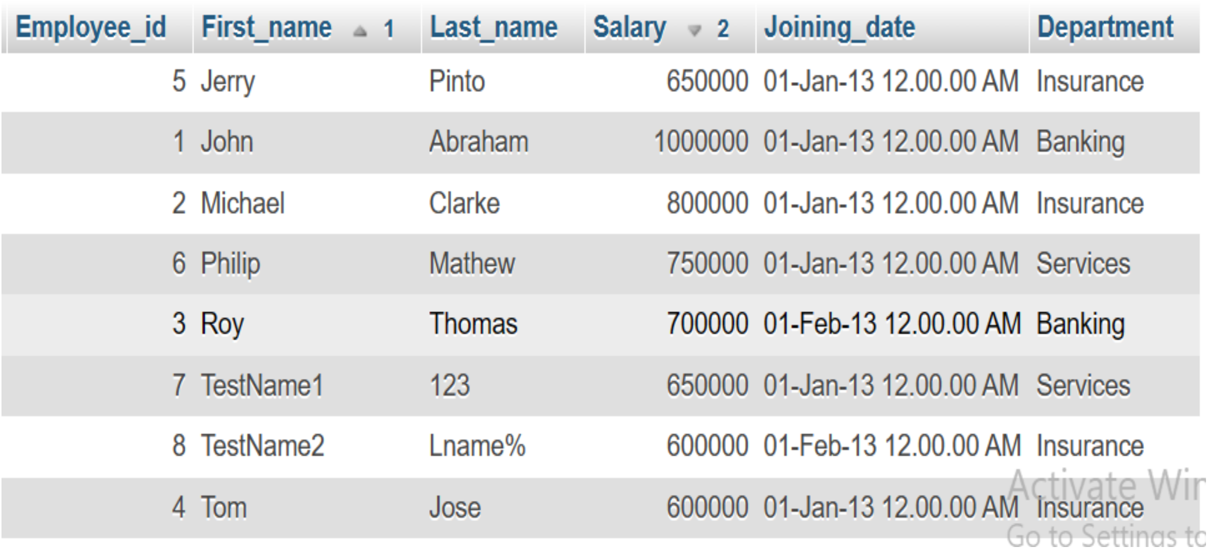


First_name	Joining_date	Salary
John	01-Jan-13 12.00.00 AM	1000000
Michael	01-Jan-13 12.00.00 AM	800000
Roy	01-Feb-13 12.00.00 AM	700000
Tom	01-Jan-13 12.00.00 AM	600000
Jerry	01-Jan-13 12.00.00 AM	650000
Philip	01-Jan-13 12.00.00 AM	750000
TestName1	01-Jan-13 12.00.00 AM	650000
TestName2	01-Feb-13 12.00.00 AM	600000

c) Get all employee details from the employee table order by First_Name Ascending and Salary descending?

Ans:-

SELECT * FROM employee ORDER BY First_name ASC,Salary DESC;

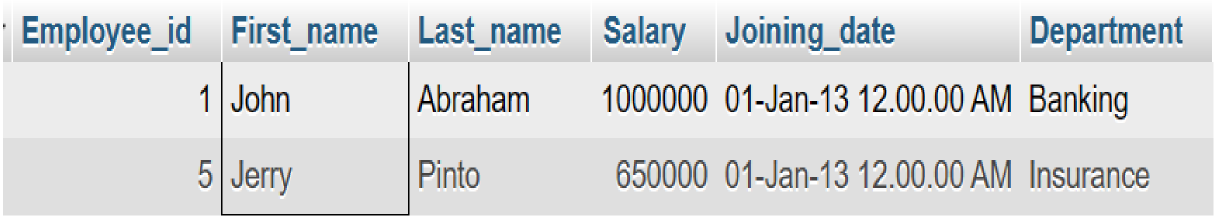


Employee_id	First_name ▲ 1	Last_name	Salary ▼ 2	Joining_date	Department
5	Jerry	Pinto	650000	01-Jan-13 12.00.00 AM	Insurance
1	John	Abraham	1000000	01-Jan-13 12.00.00 AM	Banking
2	Michael	Clarke	800000	01-Jan-13 12.00.00 AM	Insurance
6	Philip	Mathew	750000	01-Jan-13 12.00.00 AM	Services
3	Roy	Thomas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

d) Get employee details from employee table whose first name contains 'J'.

Ans:-

SELECT * FROM employee WHERE First_name LIKE '%j%';

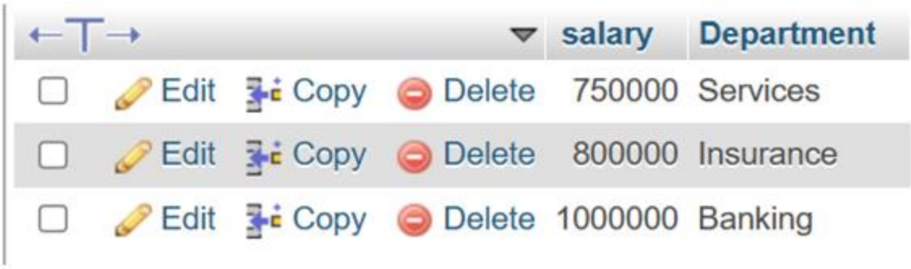


Employee_id	First_name	Last_name	Salary	Joining_date	Department
1	John	Abraham	1000000	01-Jan-13 12.00.00 AM	Banking
5	Jerry	Pinto	650000	01-Jan-13 12.00.00 AM	Insurance

e) Get department wise maximum salary from employee table order by salary ascending?

Ans:-

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



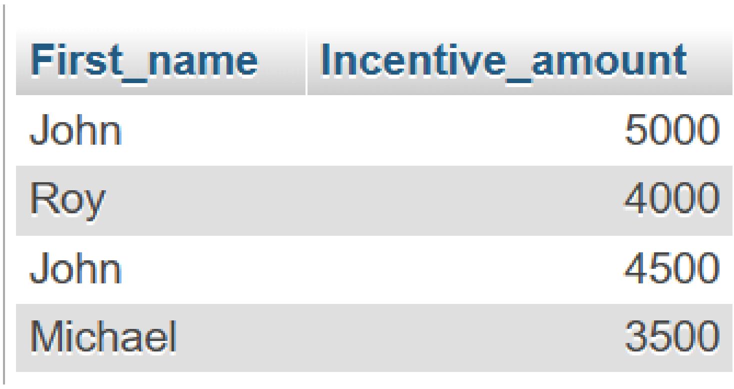
←T→			▼	salary	Department
☐	Edit	Copy	Delete	750000	Services
☐	Edit	Copy	Delete	800000	Insurance
☐	Edit	Copy	Delete	1000000	Banking

f) Select first_name, incentive amount from employee and incentives table for those employees who have incentives and incentive amount greater than 3000

Ans:-

SELECT First_name ,Incentive_amount FROM employee A INNER JOIN incentive B ON

```
A.Employee_id=B.Employee_ref_id AND
Incentive_amount >3000;
```

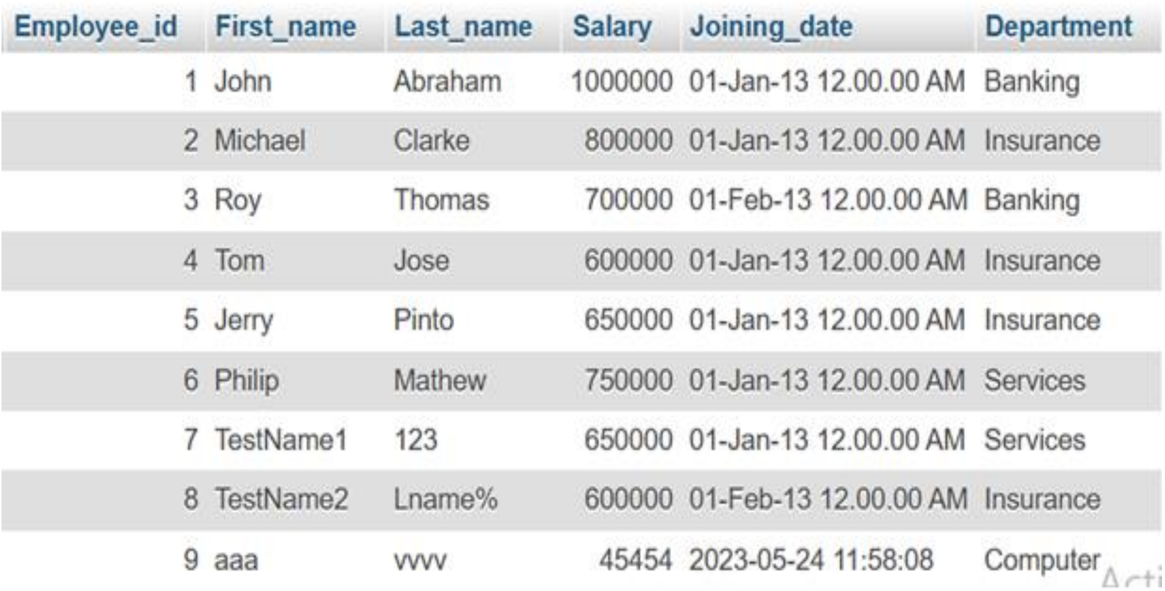


First_name	Incentive_amount
John	5000
Roy	4000
John	4500
Michael	3500

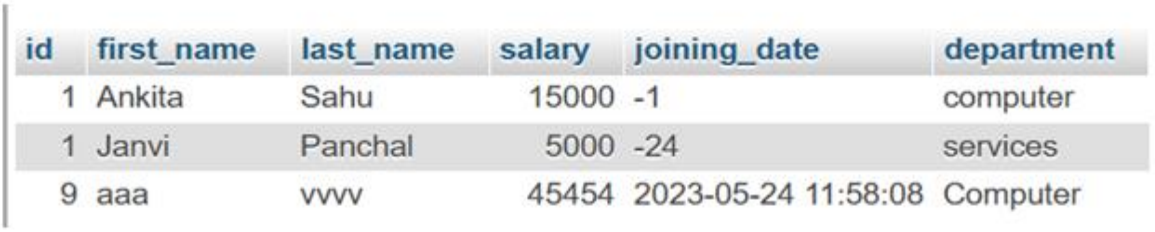
g) Create After Insert trigger on Employee table which insert records in view table

Ans:-

```
CREATE TRIGGER 'after insert' AFTER INSERT ON
'employee' FOR EACH ROW INSERT INTO Views
(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);
```

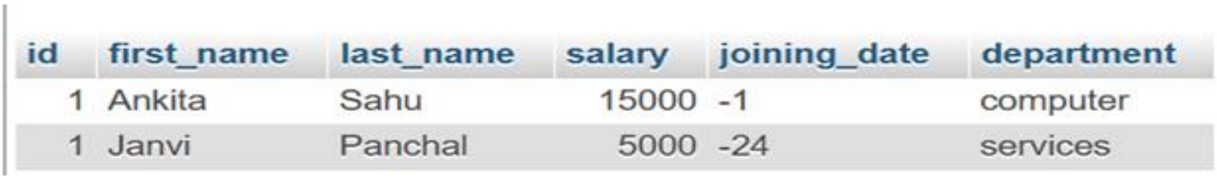


Employee_id	First_name	Last_name	Salary	Joining_date	Department
1	John	Abraham	1000000	01-Jan-13 12.00.00 AM	Banking
2	Michael	Clarke	800000	01-Jan-13 12.00.00 AM	Insurance
3	Roy	Thomas	700000	01-Feb-13 12.00.00 AM	Banking
4	Tom	Jose	600000	01-Jan-13 12.00.00 AM	Insurance
5	Jerry	Pinto	650000	01-Jan-13 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
9	aaa	vvvv	45454	2023-05-24 11:58:08	Computer



id	first_name	last_name	salary	joining_date	department
1	Ankita	Sahu	15000	-1	computer
1	Janvi	Panchal	5000	-24	services
9	aaa	vvvv	45454	2023-05-24 11:58:08	Computer

View Table:-



id	first_name	last_name	salary	joining_date	department
1	Ankita	Sahu	15000	-1	computer
1	Janvi	Panchal	5000	-24	services

4. Create table given below: Salesperson and Customer

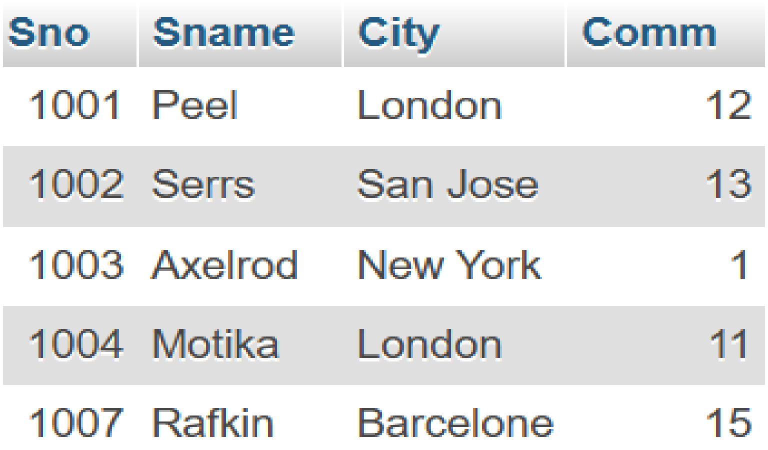
TABLE 1:

TABLE NAME SALESPERSON :-

Query:-

```
CREATE TABLE salesperson(Sno int(11) PRIMARY KEY,
Sname varchar(50), City varchar(50), Comm int(11));
```

```
NSERT INTO salesperson(Sno,Sname,City,Comm) VALUES
(1001,'Peel','London',12),
(1002,'Serrs','San Jose',13),
(1004,'Motika','London',11),
(1007,'Rafkin','Barcelone',15),
(1003,'Axelrod','New York',1);
```



Sno	Sname	City	Comm
1001	Peel	London	12
1002	Serrs	San Jose	13
1003	Axelrod	New York	1
1004	Motika	London	11
1007	Rafkin	Barcelone	15

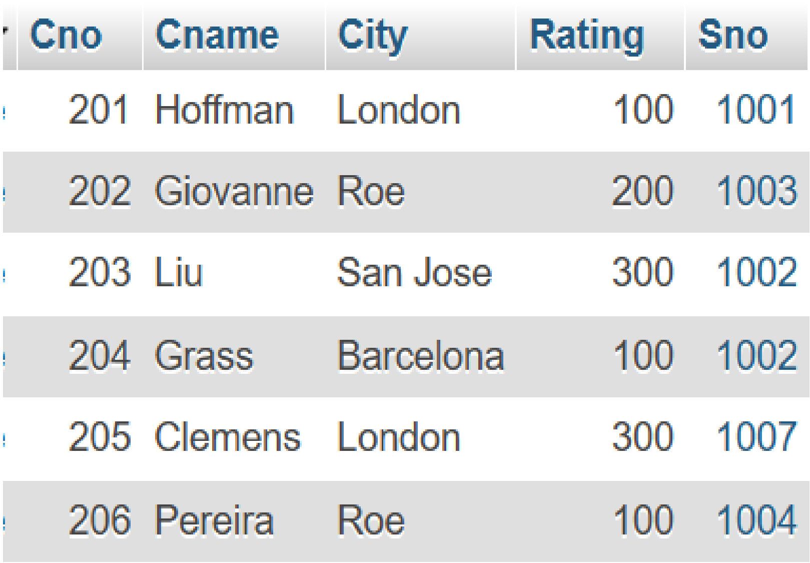
TABLE 2:

TABLE NAME CUSTOMER:-

Query:-

```
CREATE TABLE customer(Cno int(11) PRIMARY
KEY,Cname varchar(50),City varchar(50),Rating
int(11),Sno int(11),
 FOREIGN KEY (Sno) REFERENCES salesperon (Sno) ON
DELETE CASCADE ON UPDATE CASCADE);

INSERT INTO customer(Cno,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),
(205,'Clemens','London',300,1007),
(206,'Pereira','Roe',100,1004);
```



Cno	Cname	City	Rating	Sno
201	Hoffman	London	100	1001
202	Giovanne	Roe	200	1003
203	Liu	San Jose	300	1002
204	Grass	Barcelona	100	1002
205	Clemens	London	300	1007
206	Pereira	Roe	100	1004

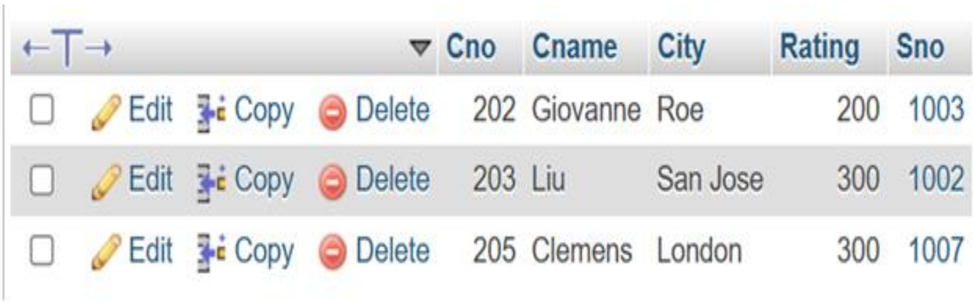
Retrieve the below data from above table

a) All orders for more than $1000.

Ans:-

Query:-

SELECT * FROM customer where RATING>100;



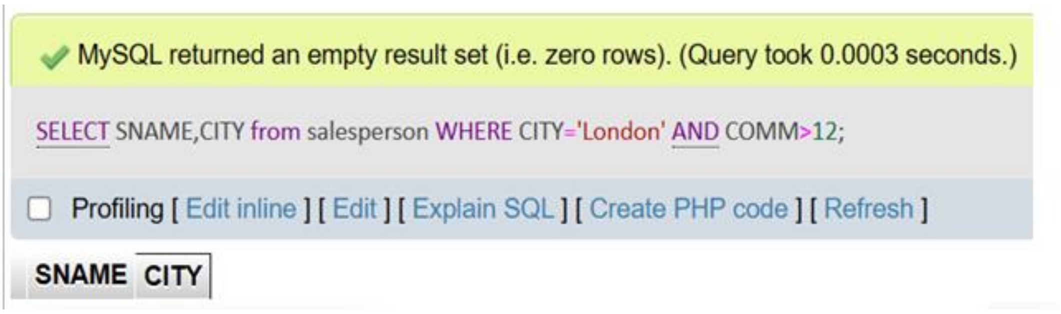
←T→			▾	Cno	Cname	City	Rating	Sno
□	Edit	Copy	Delete	202	Giovanne	Roe	200	1003
□	Edit	Copy	Delete	203	Liu	San Jose	300	1002
□	Edit	Copy	Delete	205	Clemens	London	300	1007

b) Names and cities of all salespeople in London with commission above 0.12

Ans:-

Query:-

SELECT SNAME,CITY from salesperson WHERE CITY='London' AND COMM>12;



✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0003 seconds.)

```
SELECT SNAME,CITY from salesperson WHERE CITY='London' AND COMM>12;
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

SNAME	CITY

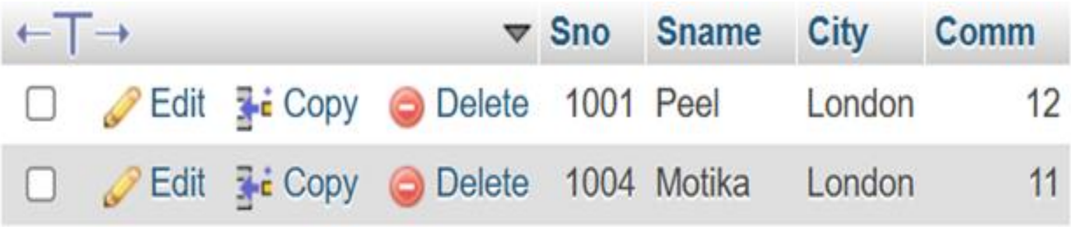
c) All salespeople either in Barcelona or in

London

Ans:-

Query:-

```
SELECT * from salesperson where CITY='barcelona'
OR CITY='london';
```



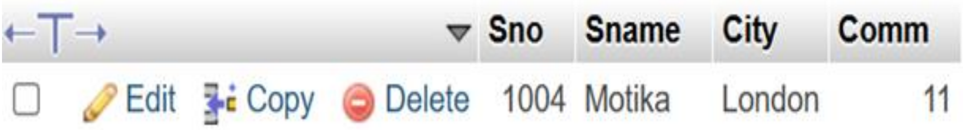
←T→			▼	Sno	Sname	City	Comm
□	Edit	Copy	Delete	1001	Peel	London	12
□	Edit	Copy	Delete	1004	Motika	London	11

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

Ans:-

Query:-

```
SELECT * from salesperson where COMM>10 and
COMM<12;
```



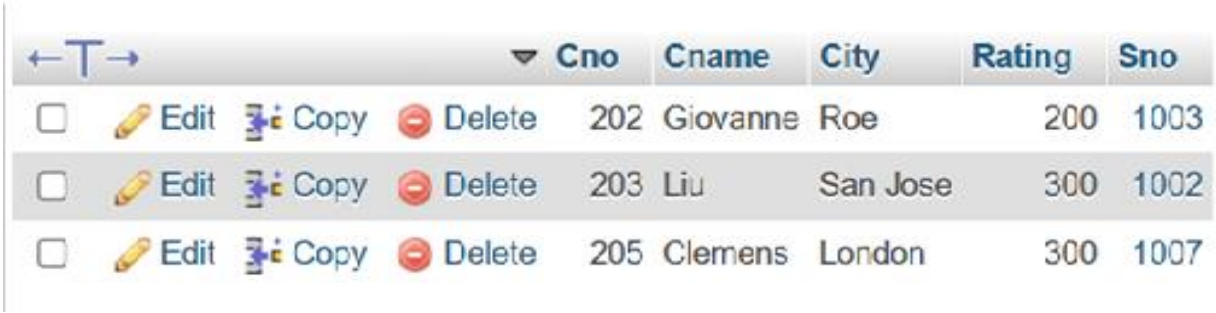
←T→			▼	Sno	Sname	City	Comm
□	Edit	Copy	Delete	1004	Motika	London	11

e) All customers excluding those with rating <= 100 unless they are located in Rome

Ans:-

Query:-

```
SELECT * FROM CUSTOMER WHERE RATING>100
AND CITY !='ROM';
```



←T→				▼	Cno	Cname	City	Rating	Sno
□	Edit	Copy	Delete		202	Giovanne	Roe	200	1003
□	Edit	Copy	Delete		203	Liu	San Jose	300	1002
□	Edit	Copy	Delete		205	Clemens	London	300	1007

XOXOXOXOXOXOXOXO