Module 5

[Assignment]

Que-1) What do you understand By Database

Ans: Database systems are basically developed for amount of data. When dealing with huge amount of data, there are two things that require optimization: Storage of data and retrieval of data

Que-2) What is Normalization?

Ans: Normalization is the process of minimizing redundancy (duplicity) from a relation or set of relations.

- Redundancy in relation may cause insertion, deletion and updation anomalies. So, it helps to minimize the redundancy in relations.

Que-3) What is Difference between DBMS and RDBMS?

Ans:

Aspects	DBMS	RDBMS
Data Organization	Stores data in files and directories.	Organizes data into tables with predefined relationships.
Data Relationships	Does not enforce relationships between data.	Enforces relationships between tables using primary and foreign keys.

Data Integrity	Offers limited support for data integrity.	Ensures data integrity through constraints and validations.
Data Model	Supports various data models, including NoSQL.	Follows a relational data model with tables and rows.
Schema Flexibility	Schema changes may require data migration.	Allows schema modifications without data migration.
ACID Properties	May not fully support ACID properties for transactions.	Fully supports ACID properties for reliable transactions.
Query Language	May have proprietary query languages.	Uses standard SQL (Structured Query Language) for data manipulation.
Performance Optimization	Performance optimization may be limited.	Utilizes advanced query optimization techniques for efficient queries.
Scalability	May have limitations in handling large datasets.	Offers scalability to manage large volumes of data effectively.
Examples of Systems	File systems, MongoDB, SQLite.	MySQL, PostgreSQL, Oracle, SQL Server, etc.

Que-4) What is MF Cod Rule of RDBMS Systems?

Ans: Codd's 12 rules are a set of principles that define a database as a Relational Database Management System (RDBMS). The rules were originally proposed in 1970 by

Edgar F. Codd and further developed in 1985. A database that follows these rules is called a true relational database management system.

Que-5) What do you understand By Data Redundancy?

Ans: s Data redundancy is the practice of storing the same data in multiple places, such as within a database or across multiple data systems.

- It can be useful for data security and backup, but it can also lead to issues like inconsistent data, increased storage costs, and complicated data management.

Que-6) What is DDL Interpreter?

Ans: DDL is used to create and modify the structure of objects in a database using predefined commands and a specific syntax.

- DDL uses commands such as create, alter, truncate, drop and rename. These database objects include tables, sequences, locations, aliases, schemas and indexes.

Que-7) What is DML Compiler in SQL?

Ans: DML Compiler converts these user interactions and queries into object code so that it can be understood by the DBMS and it then process the compiled code.

Que-8) What is SQL Key Constraints writing an Example of SQL Key Constraints

Ans: Key constraints in SQL are rules that enforce data integrity and uniqueness within a database table. They help maintain data consistency and prevent accidental data entry errors.

1. Not Null
2. Primary key
3. Foreign key
4. Auto Increment
5. Unique
6. Check
7. Default

Example:

⇨ Create table student (

SID integer Primary key Auto increment,

SName Varchar (10) not null,

City Varchar (10) Unique);

⇨ Create table university (

SID int,

SName varchar (10),

SCourse varchar (10),

City varchar (10),

Foreign Key (SID) references student(SID));

Que-9) What is save Point? How to create a save Point write a Query?

Ans: A Savepoint is a marker within a transaction that allows you to partially roll back to a specific point, rather than undoing the entire transaction. This is useful when you want to revert certain changes without affecting the rest of the transaction.

Example:

⇨ Savepoint new_savepoint;

Que-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.

Example:

```
CREATE TRIGGER trigger_name
   (AFTER | BEFORE) (INSERT | UPDATE | DELETE)
       ON table_name FOR EACH ROW
       BEGIN
      --variable declarations
      --trigger code
      END;
```

TASK 1) Create Table Name : Student and Exam

Ans:

Query-1]

```
create table student (rollno integer primary key
auto_increment,name varchar (20), branch varchar(30));

insert into student(name,branch) values
('jay','computer science'),
('suhani','electronic and com'),
('kriti','electronic and com');
```

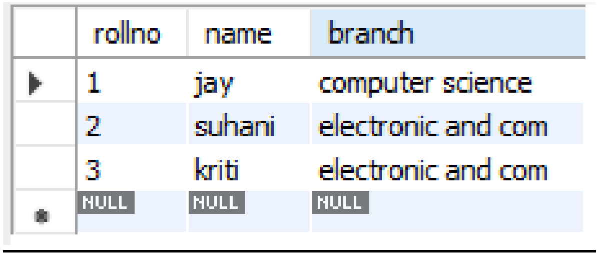
Query-2]

```
create table exam
(rollno integer, s_code varchar(10),marks int,p_code
varchar(10), FOREIGN KEY (rollno) REFERENCES
student(rollno));

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');
```

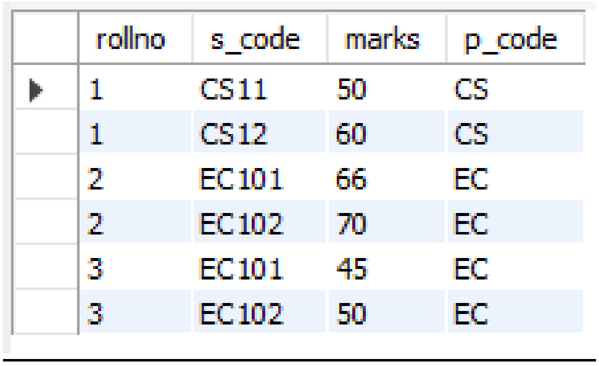
OUTPUT:

Student:



rollno	name	branch
1	jay	computer science
2	suhani	electronic and com
3	kriti	electronic and com
NULL	NULL	NULL

Exam:



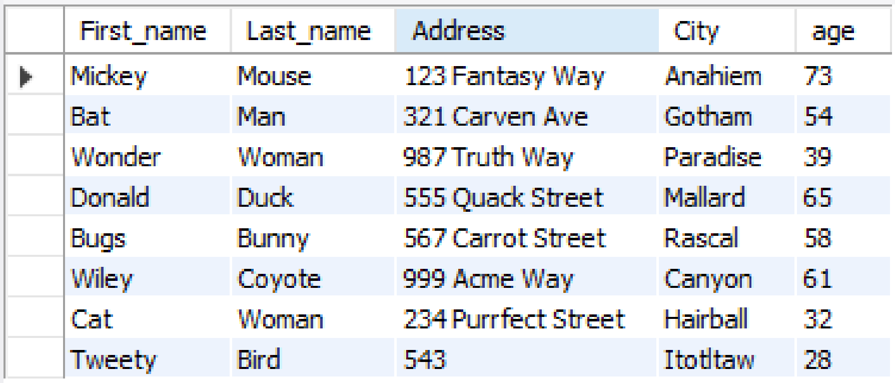
rollno	s_code	marks	p_code
1	CS11	50	CS
1	CS12	60	CS
2	EC101	66	EC
2	EC102	70	EC
3	EC101	45	EC
3	EC102	50	EC

Tast 2) Create table given below.

```
create table emp(
First_name varchar(10),
Last_name varchar(10),
Address varchar(20),
City varchar(10),
age int);
```

```
insert into emp(First_name,Last_name,Address,City,Age)
values
('Mickey','Mouse','123 Fantasy Way','Anahiem',73),
('Bat','Man','321 Carven Ave','Gotham',54),
('Wonder','Woman','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','Woman','234 Purrfect Street','Hairball',32),
('Tweety','Bird','543','Itotltaw',28);
```

Output:



First_name	Last_name	Address	City	age
Mickey	Mouse	123 Fantasy Way	Anahiem	73
Bat	Man	321 Carven Ave	Gotham	54
Wonder	Woman	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	Woman	234 Purrfect Street	Hairball	32
Tweety	Bird	543	Itotltaw	28

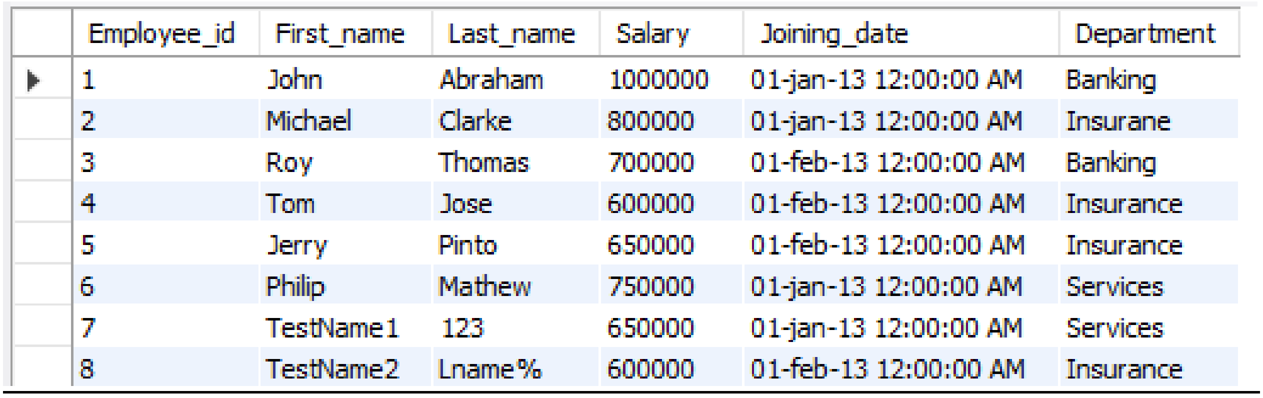
Task-3) Create table given below: Employee and Incentive

Create table Employee:

```
create table employee(
Employee_id integer primary key auto_increment,
First_name varchar(10),
Last_name varchar(10),
Salary integer,
Joining_date varchar(25),
Department varchar(15));
```

Insert into employee:

```
insert into
employee(First_name,Last_name,Salary,Joining_date,Department) values
('John','Abraham',1000000,'01-jan-13 12:00:00 AM','Banking'),
('Michael','Clarke',800000,'01-jan-13 12:00:00 AM','Insurane'),
('Roy','Thomas',700000,'01-feb-13 12:00:00 AM','Banking'),
('Tom','Jose',600000,'01-feb-13 12:00:00 AM','Insurance'),
('Jerry','Pinto',650000,'01-feb-13 12:00:00 AM','Insurance'),
('Philip','Mathew',750000,'01-jan-13 12:00:00 AM','Services'),
('TestName1','123',650000,'01-jan-13 12:00:00 AM','Services'),
('TestName2','Lname%',600000,'01-feb-13 12:00:00 AM','Insurance');
```



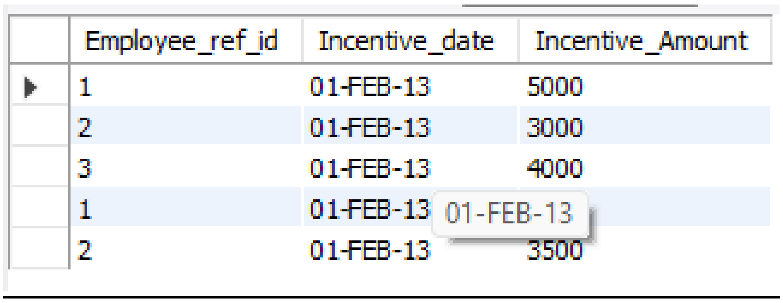
	Employee_id	First_name	Last_name	Salary	Joining_date	Department
▶	1	John	Abraham	1000000	01-jan-13 12:00:00 AM	Banking
	2	Michael	Clarke	800000	01-jan-13 12:00:00 AM	Insurane
	3	Roy	Thomas	700000	01-feb-13 12:00:00 AM	Banking
	4	Tom	Jose	600000	01-feb-13 12:00:00 AM	Insurance
	5	Jerry	Pinto	650000	01-feb-13 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

Create table Incentive:

```
create table incentive(
Employee_ref_id int,
Incentive_date varchar(15),
Incentive_Amount int,
foreign key(Employee_ref_id) references employee(Employee_id));
```

insert into Incentives:

```
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-FEB-13',4500),
(2,'01-FEB-13',3500);
```

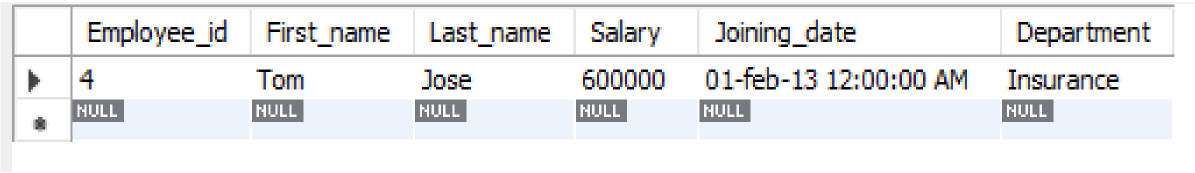


Employee_ref_id	Incentive_date	Incentive_Amount
1	01-FEB-13	5000
2	01-FEB-13	3000
3	01-FEB-13	4000
1	01-FEB-13	01-FEB-13
2	01-FEB-13	3500

Query-1) Get First_Name from employee table using Tom name “Employee Name”.

Ans: select * from employee where First_name='Tom';

Output:

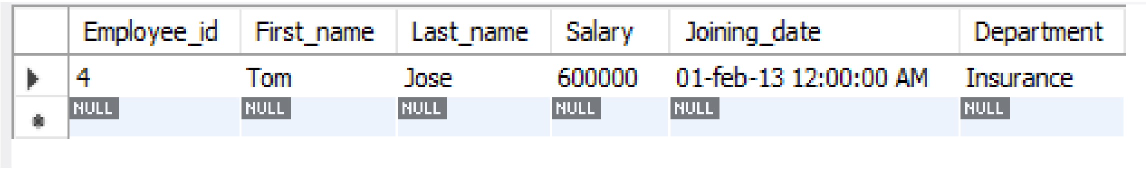


Employee_id	First_name	Last_name	Salary	Joining_date	Department
4	Tom	Jose	600000	01-feb-13 12:00:00 AM	Insurance
NULL	NULL	NULL	NULL	NULL	NULL

Query-2) Get FIRST_NAME, Joining Date, and Salary from employee table.

Ans: select * from employee where First_name='Tom';

Output:

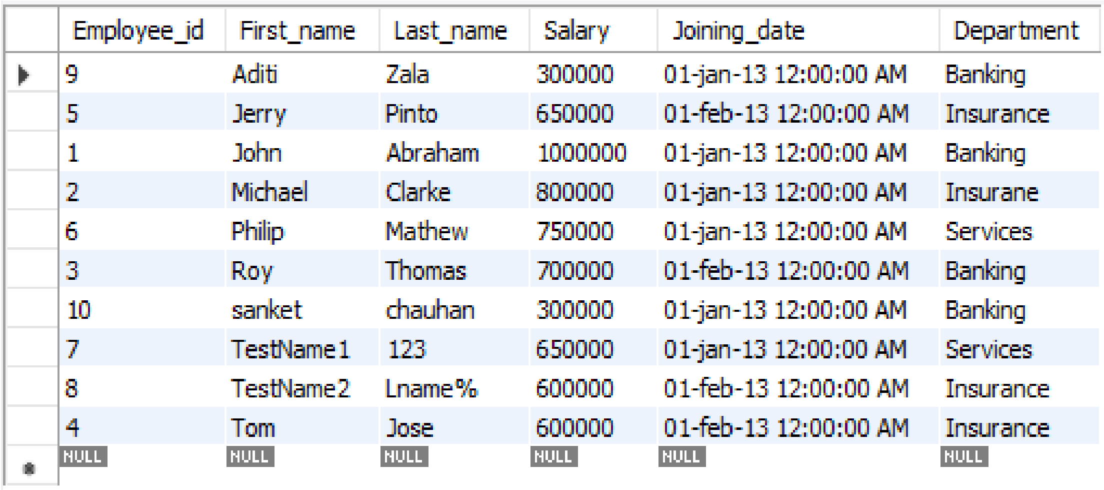


Employee_id	First_name	Last_name	Salary	Joining_date	Department
4	Tom	Jose	600000	01-feb-13 12:00:00 AM	Insurance
NULL	NULL	NULL	NULL	NULL	NULL

Query-3) 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;

Output:

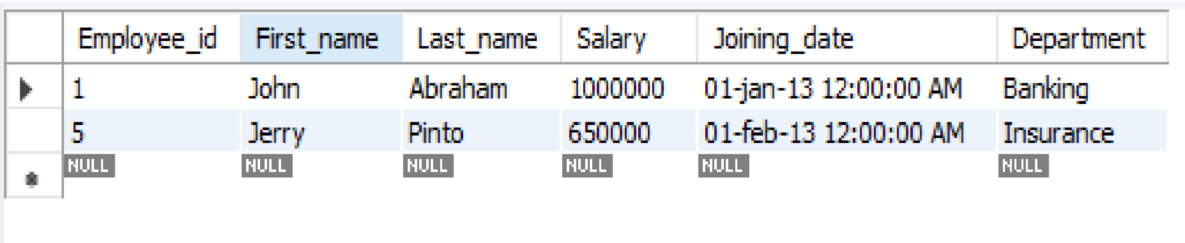


	Employee_id	First_name	Last_name	Salary	Joining_date	Department
▶	9	Aditi	Zala	300000	01-jan-13 12:00:00 AM	Banking
	5	Jerry	Pinto	650000	01-feb-13 12:00:00 AM	Insurance
	1	John	Abraham	1000000	01-jan-13 12:00:00 AM	Banking
	2	Michael	Clarke	800000	01-jan-13 12:00:00 AM	Insurane
	6	Philip	Mathew	750000	01-jan-13 12:00:00 AM	Services
	3	Roy	Thomas	700000	01-feb-13 12:00:00 AM	Banking
	10	sanket	chauhan	300000	01-jan-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-feb-13 12:00:00 AM	Insurance
*	NULL	NULL	NULL	NULL	NULL	NULL

Query-4) Get employee details from employee table whose first name contains ‘J’.

Ans: select * from employee where First_name like 'J%';

Output:

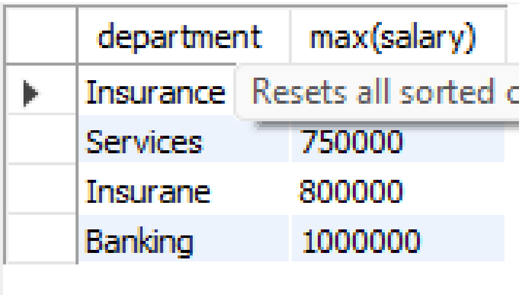


	Employee_id	First_name	Last_name	Salary	Joining_date	Department
▶	1	John	Abraham	1000000	01-jan-13 12:00:00 AM	Banking
	5	Jerry	Pinto	650000	01-feb-13 12:00:00 AM	Insurance
*	NULL	NULL	NULL	NULL	NULL	NULL

Query-5) Get department wise maximum salary from employee table order by salary ascending?

Ans: select department,max(salary) from employee group by department order by max(salary) asc;

Output:



	department	max(salary)
▶	Insurance	Resets all sorted c
	Services	750000
	Insurane	800000
	Banking	1000000

Query-6) Select first_name, incentive amount from employee and incentives table for those employees who have incentives and incentive amount greater than 3000

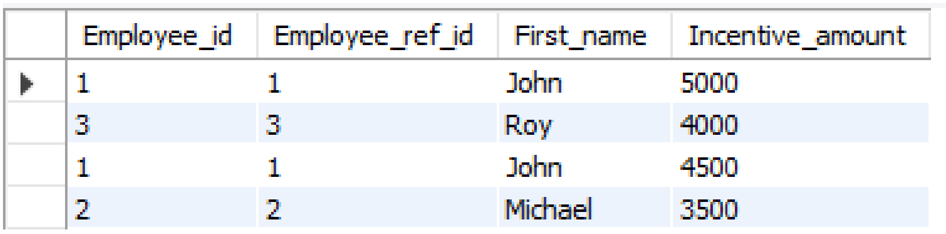
Ans: select

employee.Employee_id,Incentive.Employee_ref_id,Employee.First_name,Incentive.Incentive_amount

from employee left join incentive on employee.Employee_id = incentive.Employee_ref_id

where Incentive_amount > 3000;

Output:



	Employee_id	Employee_ref_id	First_name	Incentive_amount
▶	1	1	John	5000
	3	3	Roy	4000
	1	1	John	4500
	2	2	Michael	3500

Query-7) Create After Insert trigger on Employee table which insert records in view table

Ans:

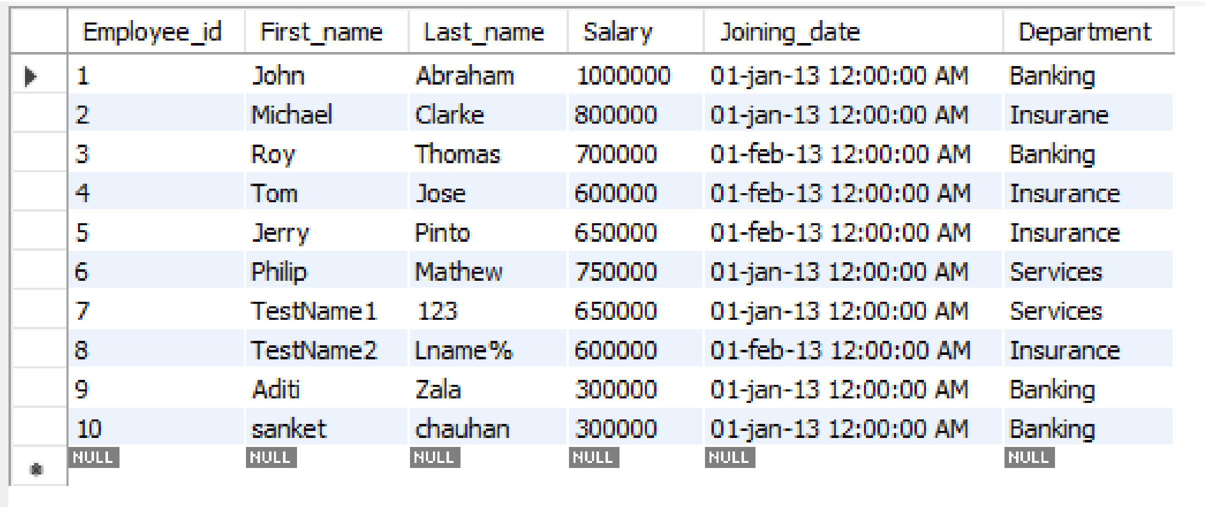
Trigger:

```
delimiter //
create trigger employee_backup
after insert
on employee
for each row
begin
insert into viewtable(First_name,Last_name,Salary,Joining_date,Department,joined_at)values
(new.First_name,new.Last_name,new.Salary,new.Joining_date,new.Department,now());
end; //
```

```
insert into employee(First_name,Last_name,Salary,Joining_date,Department)values
('sanket','chauhan',300000,'01-jan-13 12:00:00 AM','Banking');
```

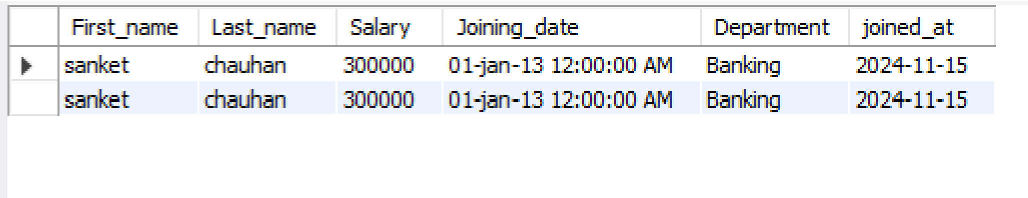
```
create table viewtable(
First_name varchar(10),
Last_name varchar(10),
Salary integer,
Joining_date varchar(25),
Department varchar(15),
joined_at date);
```

Output:



	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	Insurane
	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
	9	Aditi	Zala	300000	01-jan-13 12:00:00 AM	Banking
	10	sanket	chauhan	300000	01-jan-13 12:00:00 AM	Banking
*	NULL	NULL	NULL	NULL	NULL	NULL

<u>View table:</u>



	First_name	Last_name	Salary	Joining_date	Department	joined_at
▶	sanket	chauhan	300000	01-jan-13 12:00:00 AM	Banking	2024-11-15
	sanket	chauhan	300000	01-jan-13 12:00:00 AM	Banking	2024-11-15

Task-4) Create table given below: Salesperson and Customer

Salesperson:

```
create table salesperson (
SNo int primary key ,
SNAME varchar(10),
CITY varchar(10),
COMM float);

insert into salesperson
(SNo,SNAME,CITY,COMM) values
(1001,'Peel','London',.12),
(1002,'Serres','San Jose',.13),
(1004,'Motika','London',.11),
(1007,'Rafkin','Barcelona',.15),
(1003,'Axelrod','New York',.1);
```

Customer:

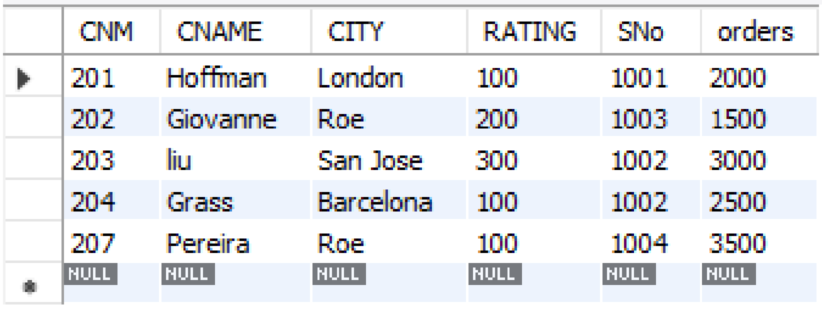
```
create table customer (
CNM int primary key,
CNAME varchar(10),
CITY varchar(10),
RATING int ,
SNo int,
foreign key(SNo) references salesperson(SNo));
```

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

Query-1) All orders for more than $1000.

Ans: select * from customer where orders>=1000;

Output:

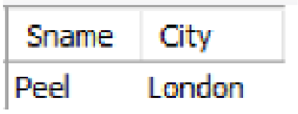


CNM	CNAME	CITY	RATING	SNo	orders
201	Hoffman	London	100	1001	2000
202	Giovanne	Roe	200	1003	1500
203	liu	San Jose	300	1002	3000
204	Grass	Barcelona	100	1002	2500
207	Pereira	Roe	100	1004	3500
NULL	NULL	NULL	NULL	NULL	NULL

Query-2) Names and cities of all salespeople in London with commission above 0.12

Ans: select SNAME,CITY from salesperson where COMM >= .12 and CITY='London';

Output:

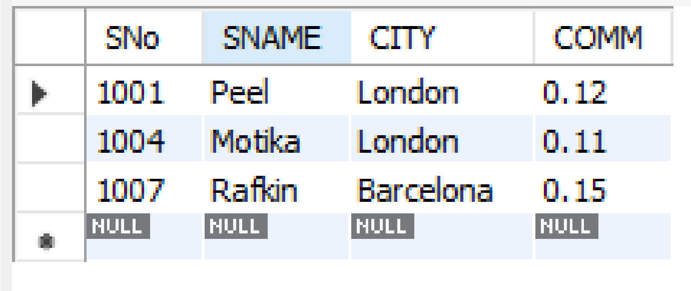


Sname	City
Peel	London

Query-3) All salespeople either in Barcelona or in London

Ans: select * from salesperson where city='Barcelona' or city='London';

Output:

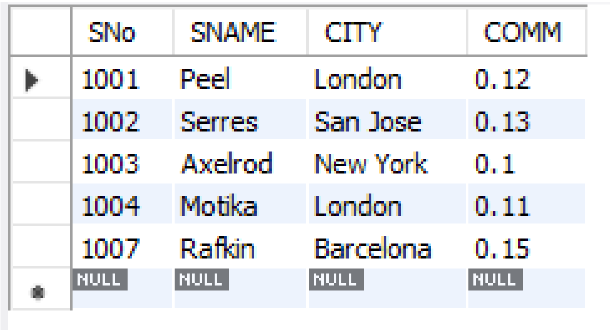


SNo	SNAME	CITY	COMM
1001	Peel	London	0.12
1004	Motika	London	0.11
1007	Rafkin	Barcelona	0.15
NULL	NULL	NULL	NULL

Query-4) All salespeople with commission between 0.10 and 0.12. (Boundary values should be excluded).

Ans: select * from salesperson where COMM > .10 or COMM <.12;

Output:

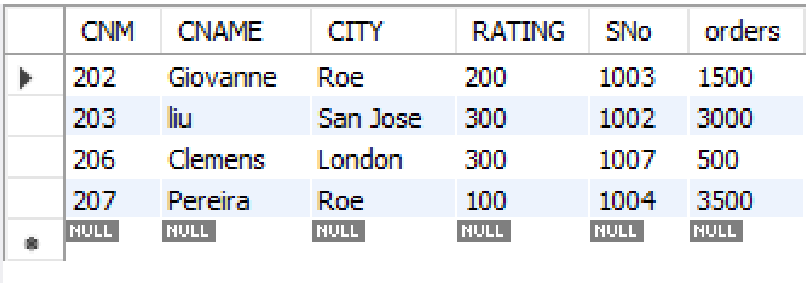


SNo	SNAME	CITY	COMM
1001	Peel	London	0.12
1002	Serres	San Jose	0.13
1003	Axelrod	New York	0.1
1004	Motika	London	0.11
1007	Rafkin	Barcelona	0.15
NULL	NULL	NULL	NULL

Query-5) All customers excluding those with rating <= 100 unless they are located in Rome

Ans: select * from customer where RATING > 100 or CITY='Roe';

Output:



CNM	CNAME	CITY	RATING	SNo	orders
202	Giovanne	Roe	200	1003	1500
203	liu	San Jose	300	1002	3000
206	Clemens	London	300	1007	500
207	Pereira	Roe	100	1004	3500
NULL	NULL	NULL	NULL	NULL	NULL