DATA BASE ASSIGNMENT

* What do you understand By Database

A database is like a digital filing cabinet where data is stored and organized so it can be easily accessed, managed, and updated. It's used to keep track of information in a structured way.

* What is Normalization?

Normalization is a way to organize a database to reduce duplicate data and ensure accuracy. It involves splitting data into smaller, related tables.

* What is Difference between DBMS and RDBMS?

DBMS (Database Management System) stores data as files without relationships between them. It supports single users and has lower security.

RDBMS (Relational Database Management System) stores data in tables with relationships between them. It supports multiple users, offers higher security, and allows for data normalization.

* What is MF Cod Rule of RDBMS Systems?

Codd's Rules are 13 guidelines that define what makes a true RDBMS. They ensure data is stored in tables and managed relationally.

* What do you understand By Data Redundancy?

Data redundancy is when the same data is stored in multiple places. This can waste space and cause inconsistencies.

* What is DDL Interpreter?

A DDL (Data Definition Language) Interpreter processes commands that define or change the structure of a database, like creating or altering tables.

* What is DML Compiler in SQL?

A DML (Data Manipulation Language) Compiler converts commands like INSERT, UPDATE, and DELETE into instructions the database can execute.

* What is SQL Key Constraints writing an Example of SQL Key Constraints

SQL key constraints are rules applied to columns in a database table to enforce data integrity and uniqueness. Common types include PRIMARY KEY, FOREIGN KEY, UNIQUE, and CHECK constraints.

Example of SQL Key Constraints:

```
CREATE TABLE Employees (

    EmployeeID INT PRIMARY KEY,

    FirstName VARCHAR(50) ,

    LastName VARCHAR(50) ,

    Email VARCHAR(100) UNIQUE KEY,

    DepartmentID INT,

    FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)    );
```

* What is save Point? How to create a save Point write a Query?

A **SAVEPOINT** in SQL is a marker within a transaction that allows you to roll back to a specific point without affecting the entire transaction.

```
BEGIN; SAVEPOINT sp1; ROLLBACK TO sp1;    COMMIT;
```

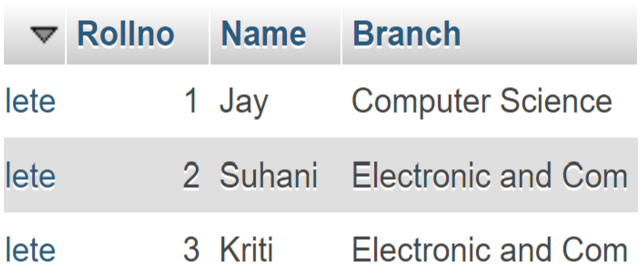
* What is trigger and how to create a Trigger in SQL?

A **trigger** in SQL is an automatic action that executes in response to events like inserting, updating, or deleting records in a table. It helps maintain data integrity and enforce business rules.

```
CREATE TABLE tiggar_01(

tid int,

tname varchar (30),

tsubject varchar(30),

tcity varchar (30),

tprice int ,

tim_date timestamp,

action_perform varchar (80),);
```

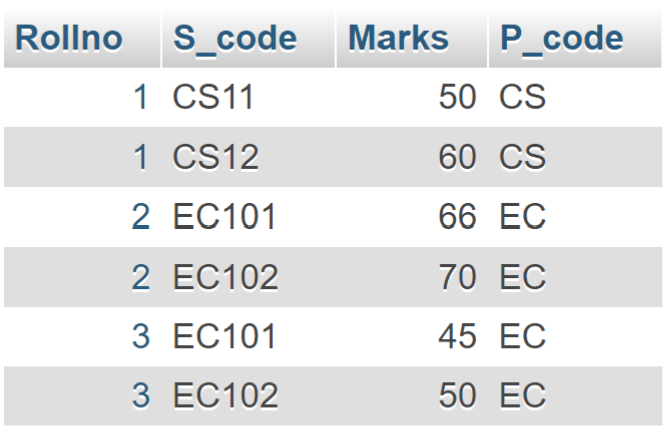
assignment 1

```
CREATE TABLE student(Rollno int PRIMARY key AUTO_INCREMENT NOT null, Name varchar (30), Branch varchar (30));

INSERT INTO student (Rollno,Name,Branch)
values (1,"Jay","Computer Science"),
        (2,"Suhani","Electronic and Com"),
        (3,"Kriti","Electronic and Com");
```



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

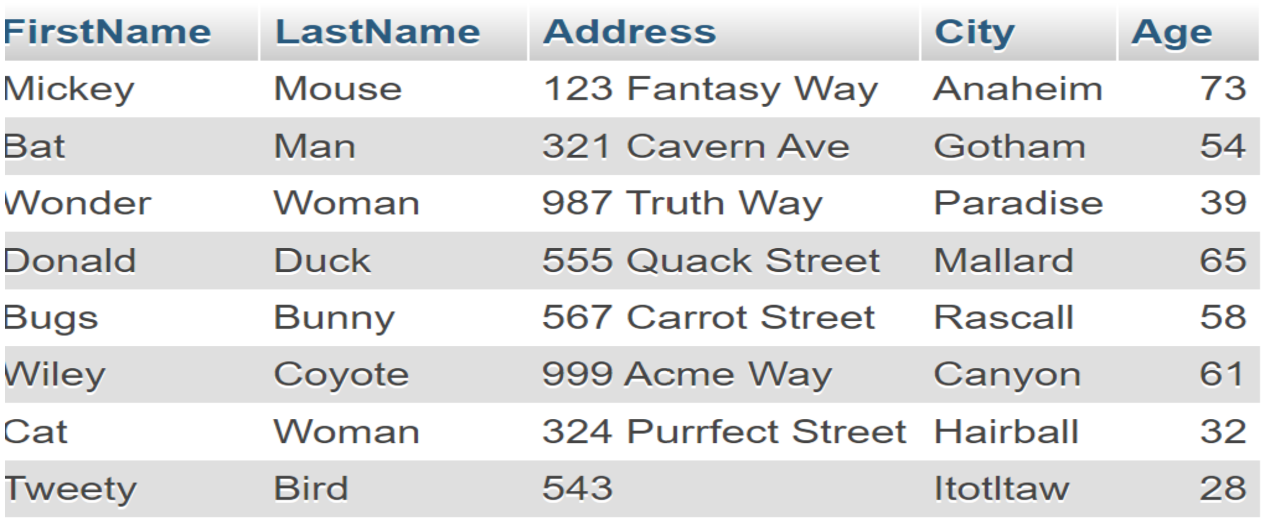
```
CREATE TABLE Exam
  (Rollno int ,
    S_code varchar (30),
    Marks int ,
    P_code varchar (30),
    FOREIGN KEY(Rollno) REFERENCES student (Rollno) );
INSERT INTO exam (Rollno,S_code,Marks,P_code)
values (1,"CS11",50,"CS"),
        (1,"CS12",60,"CS"),
        (2,"EC101",66,"EC"),
        (2,"EC102",70,"EC"),
        (3,"EC101",45,"EC"),
        (3,"EC102",50,"EC");
```



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

<u>assignment 2</u>

```
CREATE TABLE info
(FirstName varchar(30),
  LastName varchar(30),
  Address varchar (90),
  City varchar (50),
  Age int
);

insert into info (FirstName,LastName,Address,City,Age)
values ("Mickey","Mouse","123 Fantasy Way","Anaheim",73),
("Bat","Man","321 Cavern Ave","Gotham",54),
("Wonder","Woman","987 Truth Way","Paradise",39),
("Donald","Duck","555 Quack Street","Mallard",65),
("Bugs","Bunny","567 Carrot Street","Rascall",58),
("Wiley","Coyote","999 Acme Way","Canyon",61),
("Cat","Woman","324 Purrfect Street","Hairball",32),
("Tweety","Bird","543                    ","Itotltaw",28);
```

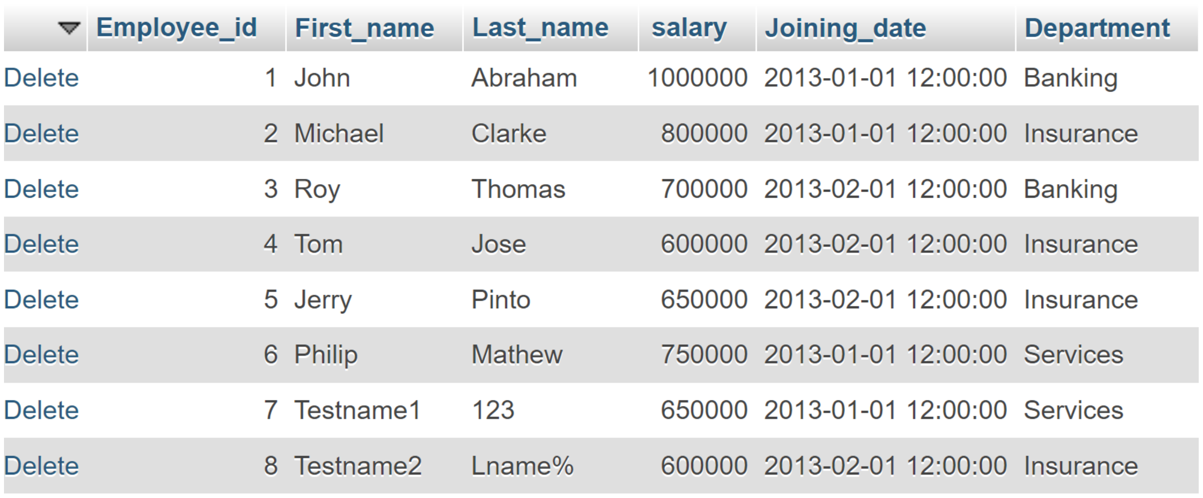


FirstName	LastName	Address	City	Age
Mickey	Mouse	123 Fantasy Way	Anaheim	73
Bat	Man	321 Cavern Ave	Gotham	54
Wonder	Woman	987 Truth Way	Paradise	39
Donald	Duck	555 Quack Street	Mallard	65
Bugs	Bunny	567 Carrot Street	Rascall	58
Wiley	Coyote	999 Acme Way	Canyon	61
Cat	Woman	324 Purrfect Street	Hairball	32
Tweety	Bird	543	Itotltaw	28

ASSIGNMENT 3

```
CREATE TABLE employee
(Employee_id int PRIMARY KEY AUTO_INCREMENT,
  First_name varchar (30),
  Last_name varchar (50),
  salary bigint,
  Joining_date datetime,
  Department varchar (50)
);

insert into employee (Employee_id,
                                    First_name,
                                    Last_name,
                                    salary,
                                    Joining_date,
                                    Department)
VALUES (1,"John","Abraham",1000000,"2013-01-01 12:00","Banking"),
(2,"Michael","Clarke",800000,"2013-01-01 12:00","Insurance");
```

```
(3,"Roy","Thomas",700000,"2013-02-01 12:00","Banking"),
(4,"Tom","Jose",600000,"2013-02-01 12:00","Insurance"),
(5,"Jerry","Pinto",650000,"2013-02-01 12:00","Insurance"),
(6,"Philip","Mathew",750000,"2013-01-01 12:00","Services"),
(7,"Testname1","123",650000,"2013-01-01 12:00","Services"),
(8,"Testname2","Lname%",600000,"2013-02-01 12:00","Insurance");
```



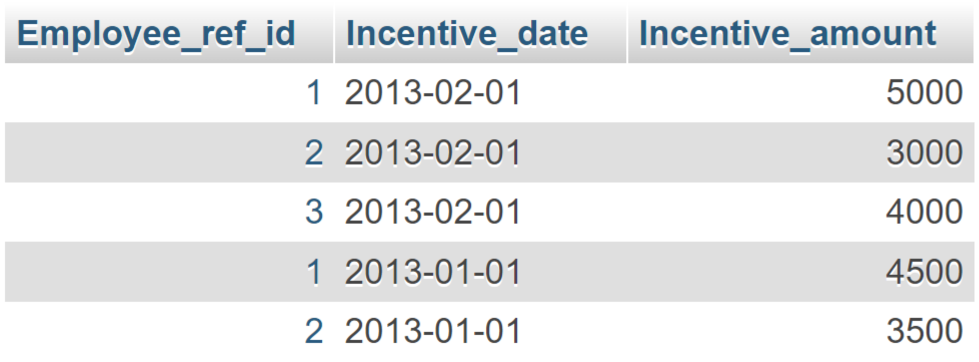
	Employee_id	First_name	Last_name	salary	Joining_date	Department
Delete	1	John	Abraham	1000000	2013-01-01 12:00:00	Banking
Delete	2	Michael	Clarke	800000	2013-01-01 12:00:00	Insurance
Delete	3	Roy	Thomas	700000	2013-02-01 12:00:00	Banking
Delete	4	Tom	Jose	600000	2013-02-01 12:00:00	Insurance
Delete	5	Jerry	Pinto	650000	2013-02-01 12:00:00	Insurance
Delete	6	Philip	Mathew	750000	2013-01-01 12:00:00	Services
Delete	7	Testname1	123	650000	2013-01-01 12:00:00	Services
Delete	8	Testname2	Lname%	600000	2013-02-01 12:00:00	Insurance

```
CREATE table incentive
(Employee_ref_id int,
  Incentive_date date,
  Incentive_amount bigint,
FOREIGN key(Employee_ref_id) REFERENCES employee (Employee_id)
);

INSERT INTO incentive (Employee_ref_id,
                       Incentive_date,
                       Incentive_amount)
VALUES(1,"2013-02-01",5000),
(2,"2013-02-01",3000),
```

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

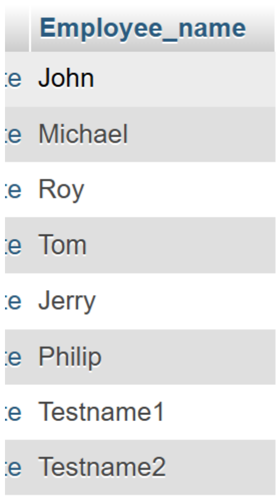
(1,"2013-01-01",4500),

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



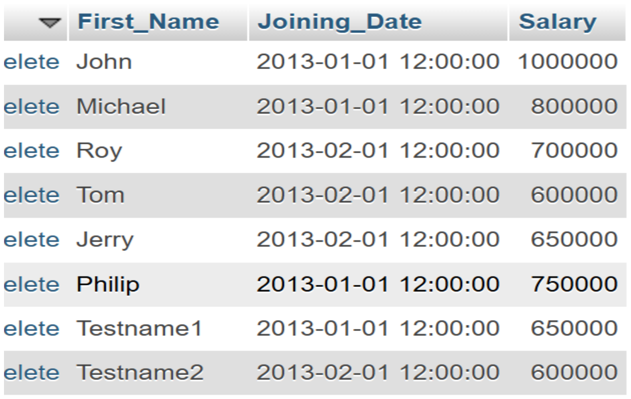
Employee_ref_id	Incentive_date	Incentive_amount
1	2013-02-01	5000
2	2013-02-01	3000
3	2013-02-01	4000
1	2013-01-01	4500
2	2013-01-01	3500

a. a) SELECT First_name as Employee_name FROM employee;



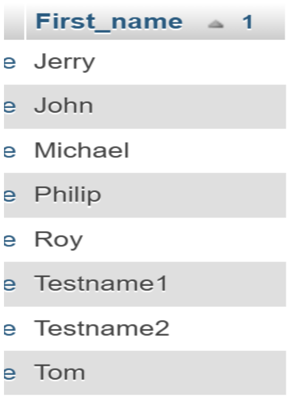
	Employee_name
:e	John
:e	Michael
:e	Roy
:e	Tom
:e	Jerry
:e	Philip
:e	Testname1
:e	Testname2

b. SELECT First_Name,Joining_Date,Salary from employee;



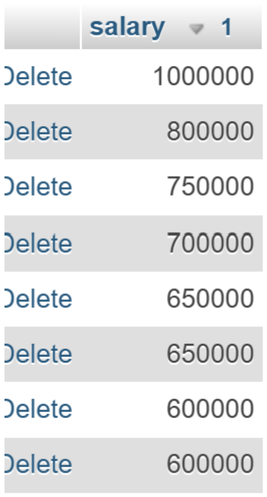
	First_Name	Joining_Date	Salary
elete	John	2013-01-01 12:00:00	1000000
elete	Michael	2013-01-01 12:00:00	800000
elete	Roy	2013-02-01 12:00:00	700000
elete	Tom	2013-02-01 12:00:00	600000
elete	Jerry	2013-02-01 12:00:00	650000
elete	Philip	2013-01-01 12:00:00	750000
elete	Testname1	2013-01-01 12:00:00	650000
elete	Testname2	2013-02-01 12:00:00	600000

c. SELECT First_name FROM employee ORDER BY First_name asc;



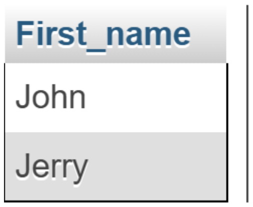
	First_name 1
e	Jerry
e	John
e	Michael
e	Philip
e	Roy
e	Testname1
e	Testname2
e	Tom

SELECT salary FROM employee ORDER BY salary DESC;



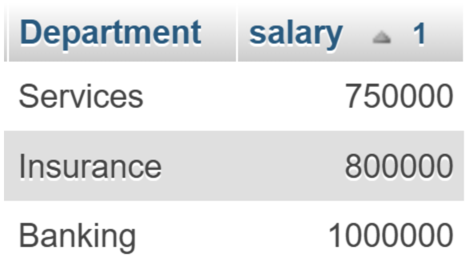
	salary 1
)elete	1000000
)elete	800000
)elete	750000
)elete	700000
)elete	650000
)elete	650000
)elete	600000
)elete	600000

d. select First_name from employee WHERE First_name like 'J%';



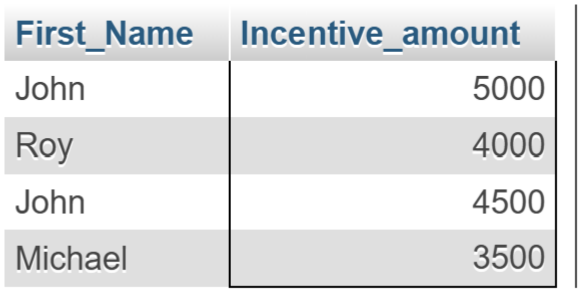
First_name
John
Jerry

e. SELECT Department, MAX(salary) AS salary FROM employee GROUP BY Department ORDER BY salary ASC;



Department	salary ▲ 1
Services	750000
Insurance	800000
Banking	1000000

f. SELECT First_name, Incentive_amount FROM employee JOIN incentive ON Employee_id = Employee_ref_id WHERE Incentive_amount > 3000;



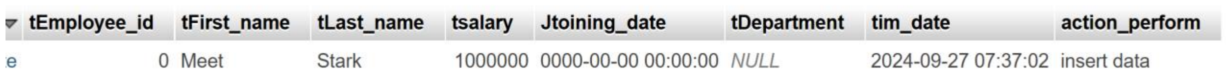
First_Name	Incentive_amount
John	5000
Roy	4000
John	4500
Michael	3500

g. CREATE TABLE trigger_01

```
(tEmployee_id int PRIMARY KEY ,
  tFirst_name varchar (30),
  tLast_name varchar (50),
  tsalary bigint,
  tJoining_date datetime,
  tDepartment varchar (50),
tim_date timestamp,

action_perform (50)

);

    CREATE trigger meet1 BEFORE insert on employee for each row

insert into trigger_01
(tEmployee_id,tFirst_name,tLast_name,tsalary,tDepartment,Jtoining_date,action_perform)

VALUES
(new.Employee_id,new.First_name,new.Last_name,new.Salary,new.Joining_date,new.depa
rtment,"insert data");
```

<u>Inserted data→</u> insert into employee(First_name,Last_name,salary,Department)

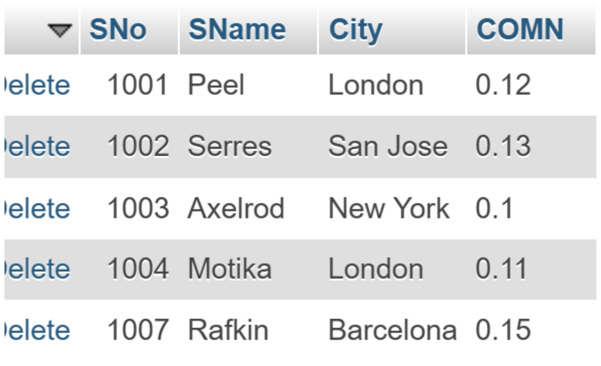
values ("Meet","Stark",1000000,"ITC");



tEmployee_id	tFirst_name	tLast_name	tsalary	Jtoining_date	tDepartment	tim_date	action_perform
0	Meet	Stark	1000000	0000-00-00 00:00:00	*NULL*	2024-09-27 07:37:02	insert data

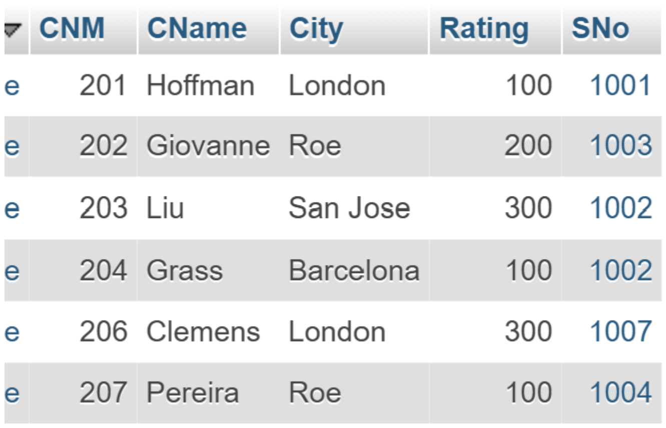
<u>Assignment 4</u>

```
CREATE TABLE salsperson1

(SNo bigint PRIMARY KEY,

SName varchar (30),

City varchar (30),

COMN varchar(30) );


INSERT INTO salsperson1 (SNo,SName,City,COMN)

values (1001,"Peel","London",.12),

(1002,"Serres","San Jose",.13),

(1004,"Motika","London",.11),

(1007,"Rafkin","Barcelona",.15),

(1003,"Axelrod","New York",.1);
```



	SNo	SName	City	COMN
elete	1001	Peel	London	0.12
elete	1002	Serres	San Jose	0.13
elete	1003	Axelrod	New York	0.1
elete	1004	Motika	London	0.11
elete	1007	Rafkin	Barcelona	0.15

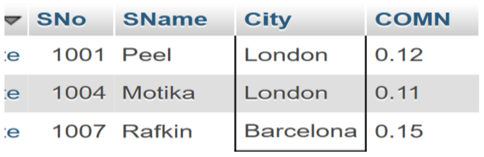
```
CREATE TABLE customer
(CNM bigint PRIMARY KEY,
  CName varchar (50),
  City varchar (50),
  Rating bigint ,
  SNo bigint,
  FOREIGN KEY(SNo)REFERENCES salsperson2 (SNo));

INSERT INTO customer6(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);
```



	CNM	CName	City	Rating	SNo
e	201	Hoffman	London	100	1001
e	202	Giovanne	Roe	200	1003
e	203	Liu	San Jose	300	1002
e	204	Grass	Barcelona	100	1002
e	206	Clemens	London	300	1007
e	207	Pereira	Roe	100	1004

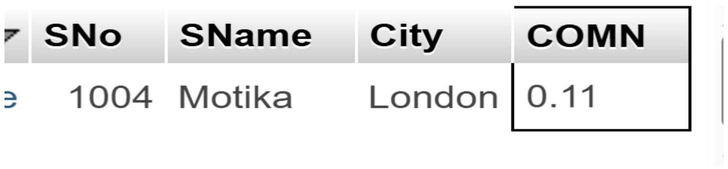
b. SELECT SName,City from salsperson2 where City = "London" and COMN > 0.12;

c. SELECT * from salsperson2 where City = "Barcelona" or City = "London";



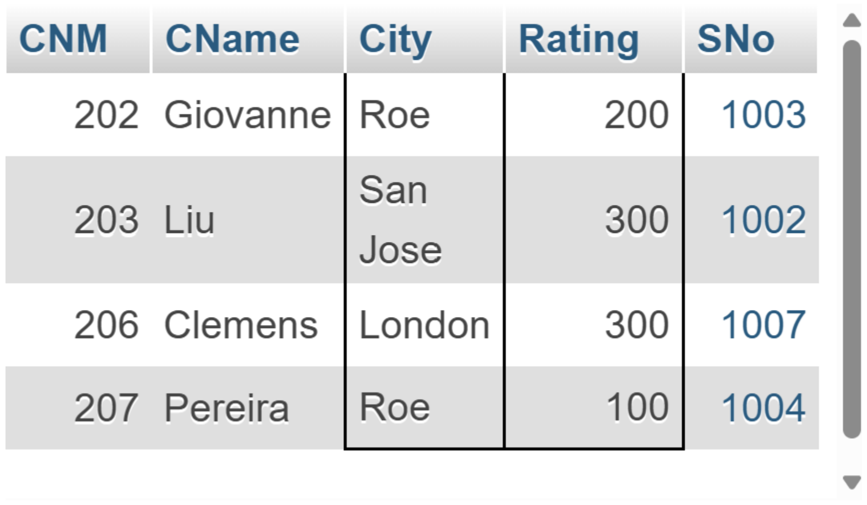
	SNo	SName	City	COMN
:e	1001	Peel	London	0.12
:e	1004	Motika	London	0.11
:e	1007	Rafkin	Barcelona	0.15

d. SELECT * FROM salsperson2 WHERE COMN > 0.10 AND COMN < 0.12;



	SNo	SName	City	COMN
e	1004	Motika	London	0.11

e. SELECT * FROM customer6 WHERE (Rating > 100 AND City != "Roe") OR City = "Roe";



CNM	CName	City	Rating	SNo
202	Giovanne	Roe	200	1003
203	Liu	San Jose	300	1002
206	Clemens	London	300	1007
207	Pereira	Roe	100	1004