AIE303 Labsheet 5

Name: Aniketh Vijesh
Roll: AM.EN.U4AIE22009

Question 1

Consider the following relational schema.

Customer (cid, cname, ac_no, ac_type, branch)

Cust_Loan(cno, ln_no, amount)

Cust_FD(cno, fd_no, amt, int_rate)

1. Find the customers who have taken a loan amount greater than that of customer with cno:100
2. List the name of all customers who have taken a loan>50000
3. Find name of all customers who have same account type as customer 'James'
4. List the name of all customers who do not have a fixed deposit.
5. Find the name of customer having highest FD amount
6. Find the name of customers who have taken at least one loan

Creating the tables and entering sample data

```
CREATE TABLE Customer (
    cid INT PRIMARY KEY,
    cname VARCHAR(100),
    ac_no INT,
    ac_type VARCHAR(20),
    branch VARCHAR(50)
);
CREATE TABLE Cust_Loan (
    cno INT,
    ln_no INT PRIMARY KEY,
    amount DECIMAL(10, 2),
    FOREIGN KEY (cno) REFERENCES Customer(cid)
);
CREATE TABLE Cust_FD (
    cno INT,
    fd_no INT PRIMARY KEY,
    amt DECIMAL(10, 2),
```

```
    int_rate DECIMAL(5, 2),
    FOREIGN KEY (cno) REFERENCES Customer(cid)
);

INSERT INTO Customer (cid, cname, ac_no, ac_type, branch)
VALUES (100, 'John Doe', 12345, 'Saving', 'New York'),
       (101, 'James', 54321, 'Current', 'Los Angeles'),
       (102, 'Alice', 67890, 'Saving', 'Chicago'),
       (103, 'Bob', 98765, 'Current', 'Houston');

INSERT INTO Cust_Loan (cno, ln_no, amount)
VALUES (100, 2001, 50000),
       (101, 2002, 60000),
       (102, 2003, 30000);

INSERT INTO Cust_FD (cno, fd_no, amt, int_rate)
VALUES (100, 3001, 20000, 5.5),
       (101, 3002, 15000, 6.0);
```

Queries

```
postgres-# SELECT cname
FROM Customer C, Cust_Loan L
WHERE C.cid = L.cno
AND L.amount > (SELECT amount FROM Cust_Loan WHERE cno = 100);
ERROR:  syntax error at or near "clear"
LINE 1: clear
        ^
postgres=# SELECT cname
FROM Customer C, Cust_Loan L
WHERE C.cid = L.cno
AND L.amount > (SELECT amount FROM Cust_Loan WHERE cno = 100);
 cname
-------
 James
(1 row)

postgres=# SELECT cname
FROM Customer C, Cust_Loan L
WHERE C.cid = L.cno AND L.amount > 50000;
 cname
-------
 James
(1 row)

postgres=# SELECT cname
FROM Customer
WHERE ac_type = (SELECT ac_type FROM Customer WHERE cname = 'James');
 cname
-------
 James
 Bob
(2 rows)

postgres=# SELECT cname
FROM Customer C
WHERE NOT EXISTS (SELECT 1 FROM Cust_FD F WHERE F.cno = C.cid);
 cname
-------
 Alice
 Bob
(2 rows)

postgres=# SELECT cname
FROM Customer C, Cust_FD F
WHERE C.cid = F.cno
AND F.amt = (SELECT MAX(amt) FROM Cust_FD);
  cname
----------
 John Doe
(1 row)

postgres=# SELECT DISTINCT cname
FROM Customer C, Cust_Loan L
WHERE C.cid = L.cno;
  cname
----------
 Alice
 John Doe
 James
(3 rows)
```

Question 2

Consider the given schema

Employee(eno, ename,sal, job, dno)
Department(dept_no, dname,location)

1. Display the name of the employee who earns highest salary.
2. Display the employee number and name for employee working as clerk and earning highest salary among clerks.
3) Display the names of salesman who earns a salary more than the highest salary of any clerk.
4) Display the names of clerks who earn a salary more than the lowest salary of any salesman.
5)Display the names of employees who earn a salary more than that of Jones or that of salary greater than that of Scott.
6) Display the names of the employees who earn highest salary in their respective departments.
7) Display the names of the employees who earn highest salaries in their respective job groups.
8) Display the employee names who are working in accounting department.
9) Display the employee names who are working in Chicago.
10) Display the Job groups having total salary greater than the maximum salary for managers.
11) Display the names of employees from department number 10 with salary greater than that of any employee working in other department.
12) Display the names of the employees from department number 10 with salary greater than that of all employee working in other departments.

Creating the table and entering sample data

```
CREATE TABLE Employee (
    eno INT PRIMARY KEY,
    ename VARCHAR(100),
    sal DECIMAL(10, 2),
    job VARCHAR(50),
    dno INT
);
CREATE TABLE Department (
    dept_no INT PRIMARY KEY,
    dname VARCHAR(100),
    location VARCHAR(50)
);
INSERT INTO Department (dept_no, dname, location)
VALUES (10, 'Accounting', 'Chicago'),
       (20, 'Sales', 'New York'),
       (30, 'HR', 'Los Angeles');
```

```sql
INSERT INTO Employee (eno, ename, sal, job, dno)
VALUES (1, 'Jones', 50000, 'Manager', 10),
       (2, 'Scott', 45000, 'Clerk', 20),
       (3, 'Smith', 55000, 'Salesman', 20),
       (4, 'Ford', 60000, 'Manager', 20),
       (5, 'Adams', 30000, 'Clerk', 30),
       (6, 'Allen', 40000, 'Clerk', 10);
```

Queries

```
postgres=# SELECT ename
FROM Employee
ORDER BY sal DESC
LIMIT 1;
 ename
-------
 Ford
(1 row)

postgres=# SELECT eno, ename
FROM Employee
WHERE job = 'Clerk'
ORDER BY sal DESC
LIMIT 1;
 eno | ename
-----+-------
   2 | Scott
(1 row)

postgres=# SELECT ename
FROM Employee
WHERE job = 'Salesman'
  AND sal > (SELECT MAX(sal) FROM Employee WHERE job = 'Clerk');
 ename
-------
 Smith
(1 row)

postgres=# SELECT ename
FROM Employee
WHERE job = 'Clerk'
  AND sal > (SELECT MIN(sal) FROM Employee WHERE job = 'Salesman');
 ename
-------
(0 rows)

postgres=# SELECT ename
FROM Employee
WHERE sal > (SELECT sal FROM Employee WHERE ename = 'Jones')
   OR sal > (SELECT sal FROM Employee WHERE ename = 'Scott');
 ename
-------
 Jones
 Smith
 Ford
(3 rows)

postgres=# SELECT ename
FROM Employee e
WHERE sal = (SELECT MAX(sal) FROM Employee WHERE dno = e.dno);
 ename
-------
 Jones
 Ford
 Adams
(3 rows)
```

```
postgres=# SELECT ename
FROM Employee e
WHERE sal = (SELECT MAX(sal) FROM Employee WHERE dno = e.dno);
 ename
-------
 Jones
 Ford
 Adams
(3 rows)

postgres=# SELECT ename
FROM Employee e
WHERE sal = (SELECT MAX(sal) FROM Employee WHERE job = e.job);
 ename
-------
 Scott
 Smith
 Ford
(3 rows)

postgres=# SELECT ename
FROM Employee
WHERE dno = (SELECT dept_no FROM Department WHERE dname = 'Accounting');
 ename
-------
 Jones
 Allen
(2 rows)

postgres=# SELECT ename
FROM Employee
WHERE dno = (SELECT dept_no FROM Department WHERE location = 'Chicago');
 ename
-------
 Jones
 Allen
(2 rows)

postgres=# SELECT job
FROM Employee
GROUP BY job
HAVING SUM(sal) > (SELECT MAX(sal) FROM Employee WHERE job = 'Manager');
   job
---------
 Manager
 Clerk
(2 rows)

postgres=# SELECT ename
FROM Employee
WHERE dno = 10
  AND sal > (SELECT MAX(sal) FROM Employee WHERE dno <> 10);
 ename
-------
(0 rows)

postgres=# SELECT ename
FROM Employee
WHERE dno = 10
  AND sal > ALL (SELECT sal FROM Employee WHERE dno <> 10);
 ename
-------
(0 rows)
```