DATABASE MANAGEMENT FOR THE BIG DATA AGE

GROUP ASSIGNMENT – 1

Group Members:	Arun Ragavendhar Arunachalam Palaniyappan	104837257
	Christopher Opie	105380202
	Yung Lien Wu	105214039
	Nichakorn Srisupornwong	104835031

Task 1: SQL

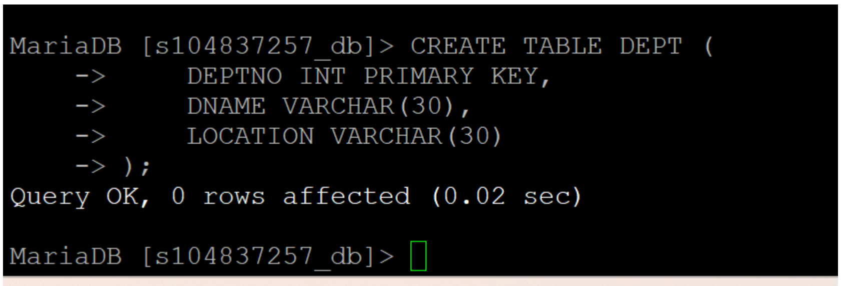
Question:2.1 Database Creation and Population

From the above given data, it is clear that EMP table references the DEPT table. Hence, DEPT table is to be created and populated first and then EMP table is to be created and populated.

SQL Queries:

- **Creating DEPT table:**

```
CREATE TABLE DEPT (
   DEPTNO INT PRIMARY KEY,
   DNAME VARCHAR(30),
   LOCATION VARCHAR(30)
);
```

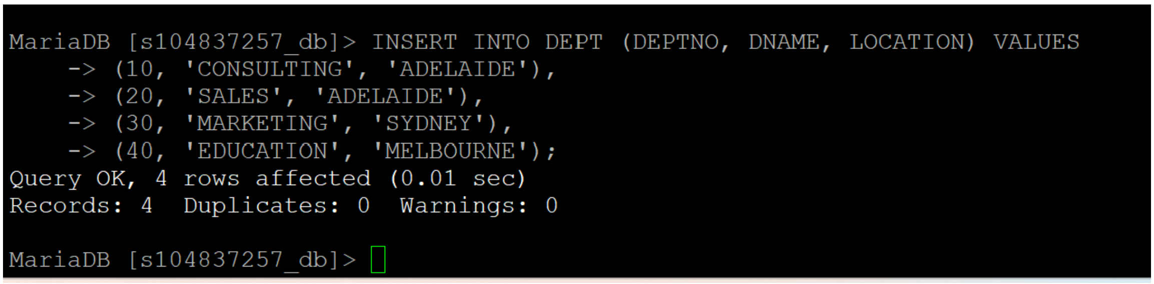


```
MariaDB [s104837257_db]> CREATE TABLE DEPT (
    ->     DEPTNO INT PRIMARY KEY,
    ->     DNAME VARCHAR(30),
    ->     LOCATION VARCHAR(30)
    -> );
Query OK, 0 rows affected (0.02 sec)

MariaDB [s104837257_db]> 
```

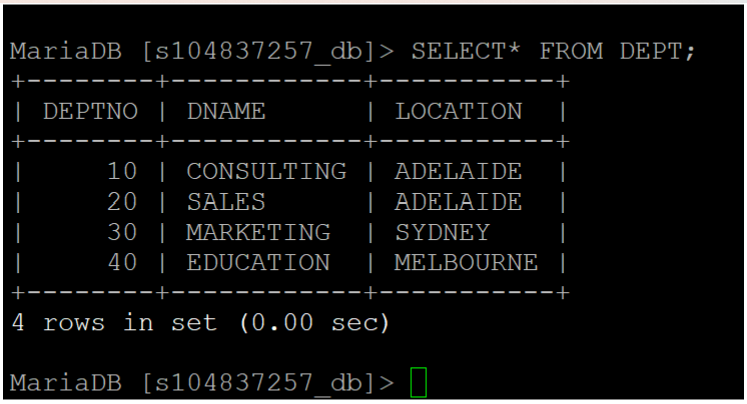
- **Populating DEPT table:**

```
INSERT INTO DEPT (DEPTNO, DNAME, LOCATION)
VALUES      (10, 'CONSULTING', 'ADELAIDE'),
            (20, 'SALES', 'ADELAIDE'),
            (30, 'MARKETING', 'SYDNEY'),
            (40, 'EDUCATION', 'MELBOURNE');
```



```
MariaDB [s104837257_db]> INSERT INTO DEPT (DEPTNO, DNAME, LOCATION) VALUES
    -> (10, 'CONSULTING', 'ADELAIDE'),
    -> (20, 'SALES', 'ADELAIDE'),
    -> (30, 'MARKETING', 'SYDNEY'),
    -> (40, 'EDUCATION', 'MELBOURNE');
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0

MariaDB [s104837257_db]> 
```

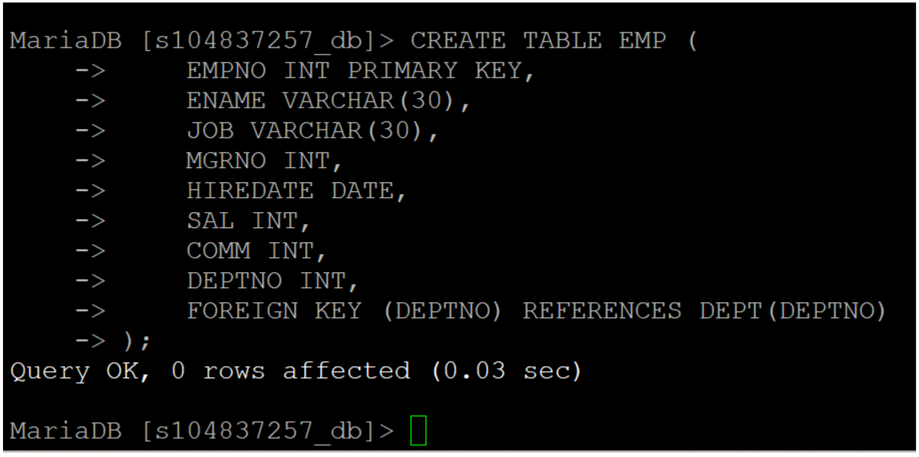


```
MariaDB [s104837257_db]> SELECT* FROM DEPT;
+--------+------------+-----------+
| DEPTNO | DNAME      | LOCATION  |
+--------+------------+-----------+
|     10 | CONSULTING | ADELAIDE  |
|     20 | SALES      | ADELAIDE  |
|     30 | MARKETING  | SYDNEY    |
|     40 | EDUCATION  | MELBOURNE |
+--------+------------+-----------+
4 rows in set (0.00 sec)

MariaDB [s104837257_db]> 
```

- **Creating EMP table:**

```
CREATE TABLE EMP (
        EMPNO INT PRIMARY KEY,
        ENAME VARCHAR(30),
        JOB VARCHAR(30),
        MGRNO INT,
       HIREDATE DATE,
        SAL INT,
       COMM INT,
       DEPTNO INT,
        FOREIGN KEY (DEPTNO) REFERENCES DEPT(DEPTNO)
       );
```

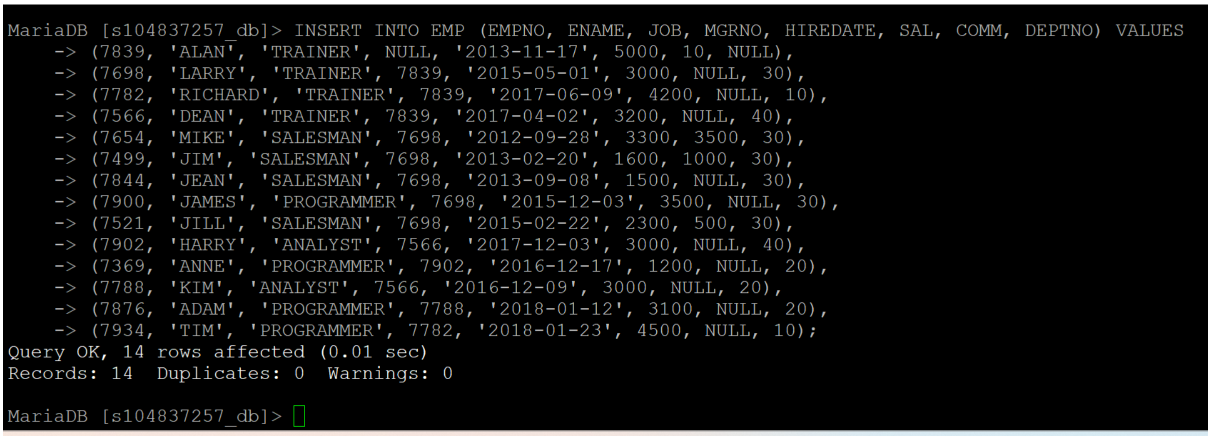


```
MariaDB [s104837257_db]> CREATE TABLE EMP (
    ->     EMPNO INT PRIMARY KEY,
    ->     ENAME VARCHAR(30),
    ->     JOB VARCHAR(30),
    ->     MGRNO INT,
    ->     HIREDATE DATE,
    ->     SAL INT,
    ->     COMM INT,
    ->     DEPTNO INT,
    ->     FOREIGN KEY (DEPTNO) REFERENCES DEPT(DEPTNO)
    -> );
Query OK, 0 rows affected (0.03 sec)

MariaDB [s104837257_db]>
```

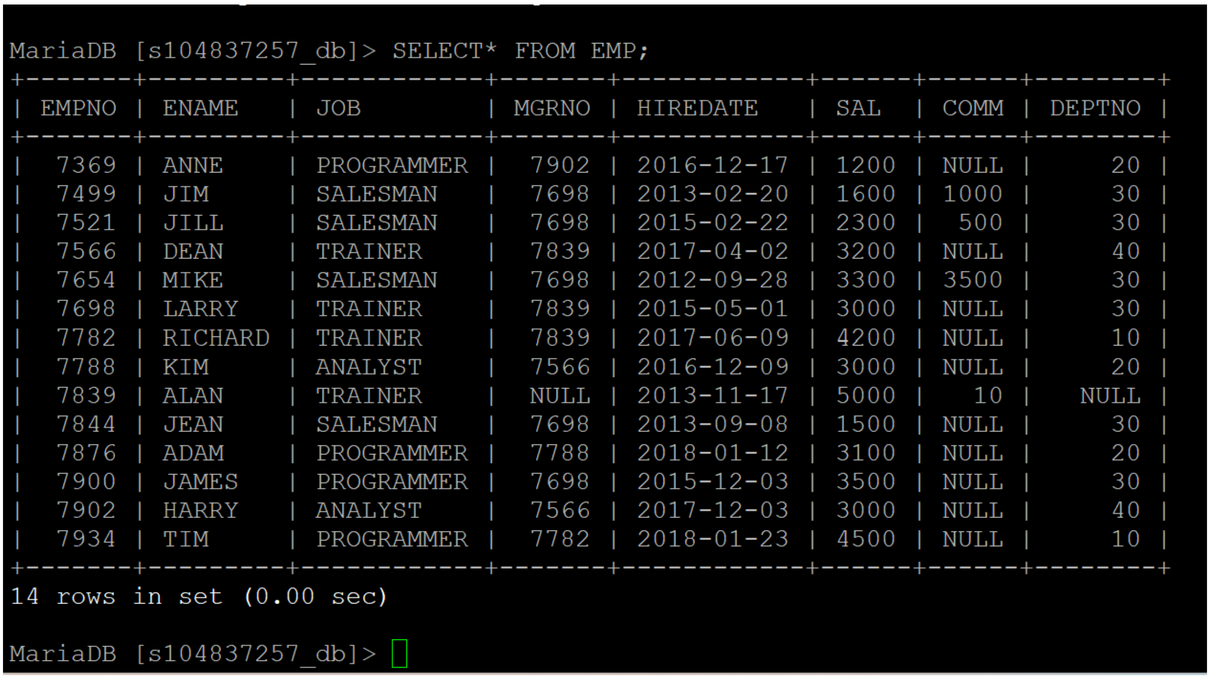
- **Populating EMP table:**

```
INSERT INTO EMP (EMPNO, ENAME, JOB, MGRNO, HIREDATE, SAL, COMM,
DEPTNO) VALUES
(7839, 'ALAN', 'TRAINER', NULL, '2013-11-17', 5000, 10),
(7698, 'LARRY', 'TRAINER', 7839, '2015-05-01', 3000, NULL, 30),
(7782, 'RICHARD', 'TRAINER', 7839, '2017-06-09', 4200, NULL, 10),
(7566, 'DEAN', 'TRAINER', 7839, '2017-04-02', 3200, NULL, 40),
(7654, 'MIKE', 'SALESMAN', 7698, '2012-09-28', 3300, 3500, 30),
(7499, 'JIM', 'SALESMAN', 7698, '2013-02-20', 1600, 1000, 30),
(7844, 'JEAN', 'SALESMAN', 7698, '2013-09-08', 1500, NULL, 30),
(7900, 'JAMES', 'PROGRAMMER', 7698, '2015-12-03', 3500, NULL, 30),
(7521, 'JILL', 'SALESMAN', 7698, '2015-02-22', 2300, 500, 30),
(7902, 'HARRY', 'ANALYST', 7566, '2017-12-03', 3000, NULL, 40),
(7369, 'ANNE', 'PROGRAMMER', 7902, '2016-12-17', 1200, NULL, 20),
(7788, 'KIM', 'ANALYST', 7566, '2016-12-09', 3000, NULL, 20),
(7876, 'ADAM', 'PROGRAMMER', 7788, '2018-01-12', 3100, NULL, 20),
(7934, 'TIM', 'PROGRAMMER', 7782, '2018-01-23', 4500, NULL, 10);
```



```
MariaDB [s104837257_db]> INSERT INTO EMP (EMPNO, ENAME, JOB, MGRNO, HIREDATE, SAL, COMM, DEPTNO) VALUES
    -> (7839, 'ALAN', 'TRAINER', NULL, '2013-11-17', 5000, 10, NULL),
    -> (7698, 'LARRY', 'TRAINER', 7839, '2015-05-01', 3000, NULL, 30),
    -> (7782, 'RICHARD', 'TRAINER', 7839, '2017-06-09', 4200, NULL, 10),
    -> (7566, 'DEAN', 'TRAINER', 7839, '2017-04-02', 3200, NULL, 40),
    -> (7654, 'MIKE', 'SALESMAN', 7698, '2012-09-28', 3300, 3500, 30),
    -> (7499, 'JIM', 'SALESMAN', 7698, '2013-02-20', 1600, 1000, 30),
    -> (7844, 'JEAN', 'SALESMAN', 7698, '2013-09-08', 1500, NULL, 30),
    -> (7900, 'JAMES', 'PROGRAMMER', 7698, '2015-12-03', 3500, NULL, 30),
    -> (7521, 'JILL', 'SALESMAN', 7698, '2015-02-22', 2300, 500, 30),
    -> (7902, 'HARRY', 'ANALYST', 7566, '2017-12-03', 3000, NULL, 40),
    -> (7369, 'ANNE', 'PROGRAMMER', 7902, '2016-12-17', 1200, NULL, 20),
    -> (7788, 'KIM', 'ANALYST', 7566, '2016-12-09', 3000, NULL, 20),
    -> (7876, 'ADAM', 'PROGRAMMER', 7788, '2018-01-12', 3100, NULL, 20),
    -> (7934, 'TIM', 'PROGRAMMER', 7782, '2018-01-23', 4500, NULL, 10);
Query OK, 14 rows affected (0.01 sec)
Records: 14  Duplicates: 0  Warnings: 0

MariaDB [s104837257_db]>
```



```
MariaDB [s104837257_db]> SELECT* FROM EMP;
+-------+---------+------------+-------+------------+------+------+--------+
| EMPNO | ENAME   | JOB        | MGRNO | HIREDATE   | SAL  | COMM | DEPTNO |
+-------+---------+------------+-------+------------+------+------+--------+
|  7369 | ANNE    | PROGRAMMER |  7902 | 2016-12-17 | 1200 | NULL |     20 |
|  7499 | JIM     | SALESMAN   |  7698 | 2013-02-20 | 1600 | 1000 |     30 |
|  7521 | JILL    | SALESMAN   |  7698 | 2015-02-22 | 2300 |  500 |     30 |
|  7566 | DEAN    | TRAINER    |  7839 | 2017-04-02 | 3200 | NULL |     40 |
|  7654 | MIKE    | SALESMAN   |  7698 | 2012-09-28 | 3300 | 3500 |     30 |
|  7698 | LARRY   | TRAINER    |  7839 | 2015-05-01 | 3000 | NULL |     30 |
|  7782 | RICHARD | TRAINER    |  7839 | 2017-06-09 | 4200 | NULL |     10 |
|  7788 | KIM     | ANALYST    |  7566 | 2016-12-09 | 3000 | NULL |     20 |
|  7839 | ALAN    | TRAINER    |  NULL | 2013-11-17 | 5000 |   10 |   NULL |
|  7844 | JEAN    | SALESMAN   |  7698 | 2013-09-08 | 1500 | NULL |     30 |
|  7876 | ADAM    | PROGRAMMER |  7788 | 2018-01-12 | 3100 | NULL |     20 |
|  7900 | JAMES   | PROGRAMMER |  7698 | 2015-12-03 | 3500 | NULL |     30 |
|  7902 | HARRY   | ANALYST    |  7566 | 2017-12-03 | 3000 | NULL |     40 |
|  7934 | TIM     | PROGRAMMER |  7782 | 2018-01-23 | 4500 | NULL |     10 |
+-------+---------+------------+-------+------------+------+------+--------+
14 rows in set (0.00 sec)

MariaDB [s104837257_db]>
```

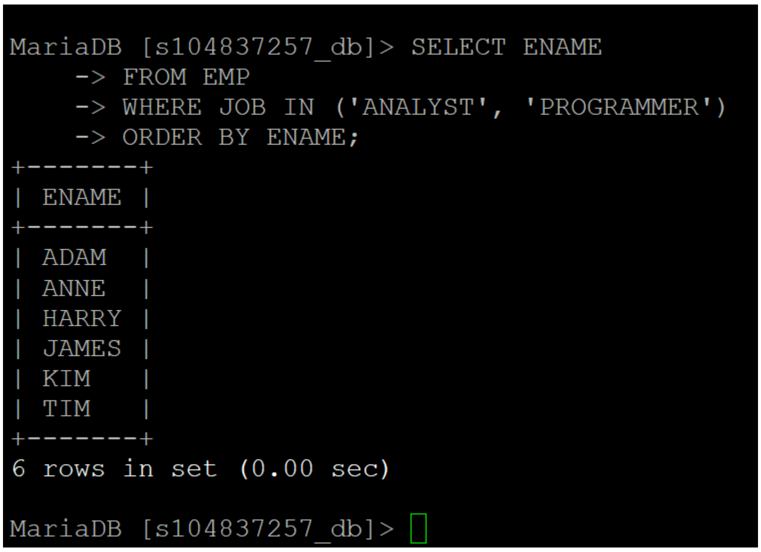
There are many ways /types of queries that can be written to get the same output.

But the approach here is to use the query that is most efficient and has least processing.

For Example – wherever there is a possibility to solve a question by using either a SUB QUERY or an INNER JOIN, the JOIN method has been used as it is more efficient has lesser processing required.

2.2) Find employees (ENAME) whose job (JOB) is either ANALYST or PROGRAMMER, ordered by employee name:

SELECT ENAME
FROM EMP
WHERE JOB **IN** ('ANALYST', 'PROGRAMMER')
ORDER BY ENAME;

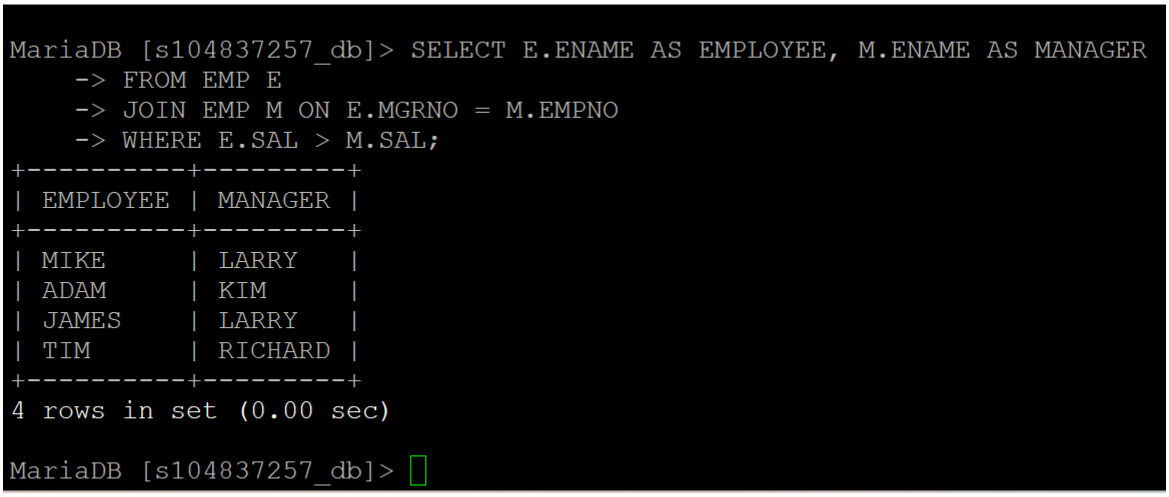


```
MariaDB [s104837257_db]> SELECT ENAME
    -> FROM EMP
    -> WHERE JOB IN ('ANALYST', 'PROGRAMMER')
    -> ORDER BY ENAME;
+--------+
| ENAME  |
+--------+
| ADAM   |
| ANNE   |
| HARRY  |
| JAMES  |
| KIM    |
| TIM    |
+--------+
6 rows in set (0.00 sec)

MariaDB [s104837257_db]>
```

2.3) Find employees whose salary (SAL) is higher than their manager's salary. List name of both employees (ENAME) and their managers (rename as MNAME):

SELECT E.ENAME AS EMPLOYEE, M.ENAME AS MANAGER
FROM EMP E
JOIN EMP M **ON** E.MGRNO = M.EMPNO
WHERE E.SAL > M.SAL;

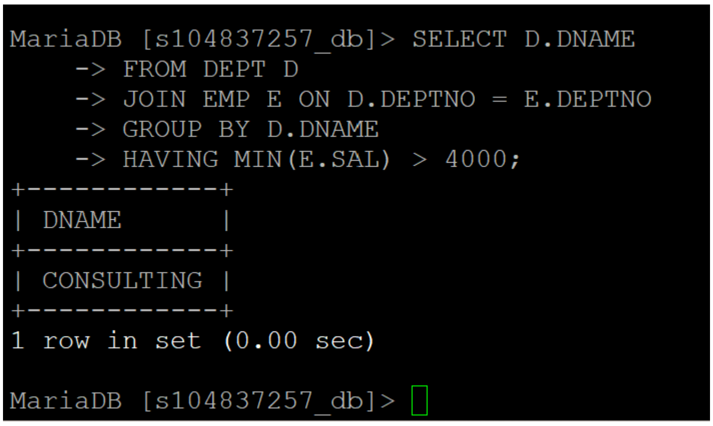


```
MariaDB [s104837257_db]> SELECT E.ENAME AS EMPLOYEE, M.ENAME AS MANAGER
    -> FROM EMP E
    -> JOIN EMP M ON E.MGRNO = M.EMPNO
    -> WHERE E.SAL > M.SAL;
+----------+---------+
| EMPLOYEE | MANAGER |
+----------+---------+
| MIKE     | LARRY   |
| ADAM     | KIM     |
| JAMES    | LARRY   |
| TIM      | RICHARD |
+----------+---------+
4 rows in set (0.00 sec)

MariaDB [s104837257_db]>
```

2.4) Find departments (DNAME) in which all employees earn more than 4000:

SELECT D.DNAME
FROM DEPT D
JOIN EMP E **ON** D.DEPTNO = E.DEPTNO
GROUP BY D.DNAME
HAVING MIN(E.SAL) > 4000;

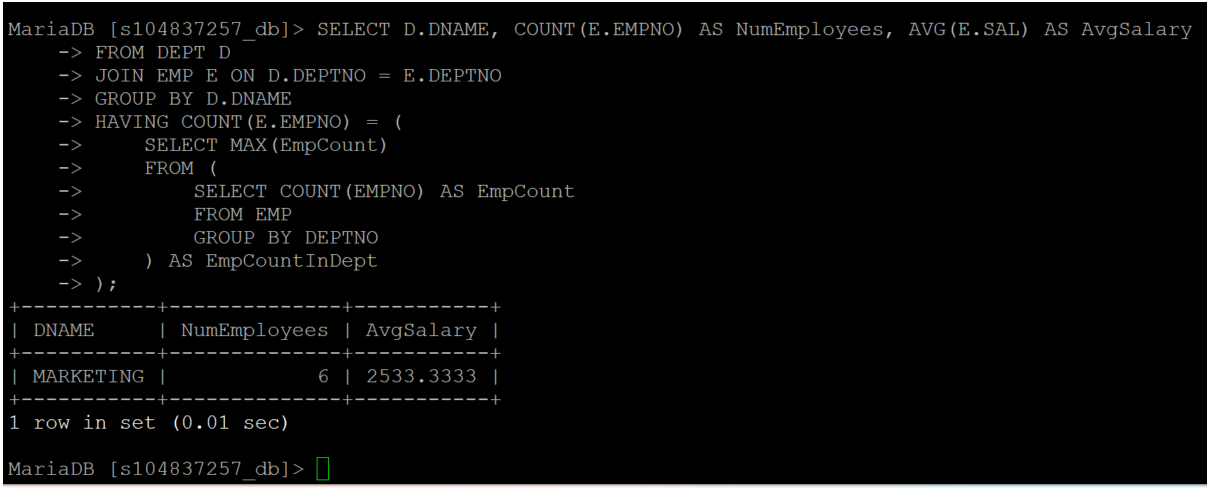


```
MariaDB [s104837257_db]> SELECT D.DNAME
    -> FROM DEPT D
    -> JOIN EMP E ON D.DEPTNO = E.DEPTNO
    -> GROUP BY D.DNAME
    -> HAVING MIN(E.SAL) > 4000;
+------------+
| DNAME      |
+------------+
| CONSULTING |
+------------+
1 row in set (0.00 sec)

MariaDB [s104837257_db]> 
```

2.5) Find the department with the largest number of employees. Show DNAME, the number of employees, and the average salary. If more than one such department exists, list all of them:

```
SELECT D.DNAME, COUNT(E.EMPNO) AS NumEmployees, AVG(E.SAL) AS AvgSalary
FROM DEPT D
JOIN EMP E ON D.DEPTNO = E.DEPTNO
GROUP BY D.DNAME
HAVING COUNT(E.EMPNO) = (
  SELECT MAX(EmpCount)
  FROM (
    SELECT COUNT(EMPNO) AS EmpCount
    FROM EMP
    GROUP BY DEPTNO
  ) AS EmpCountInDept
);
```



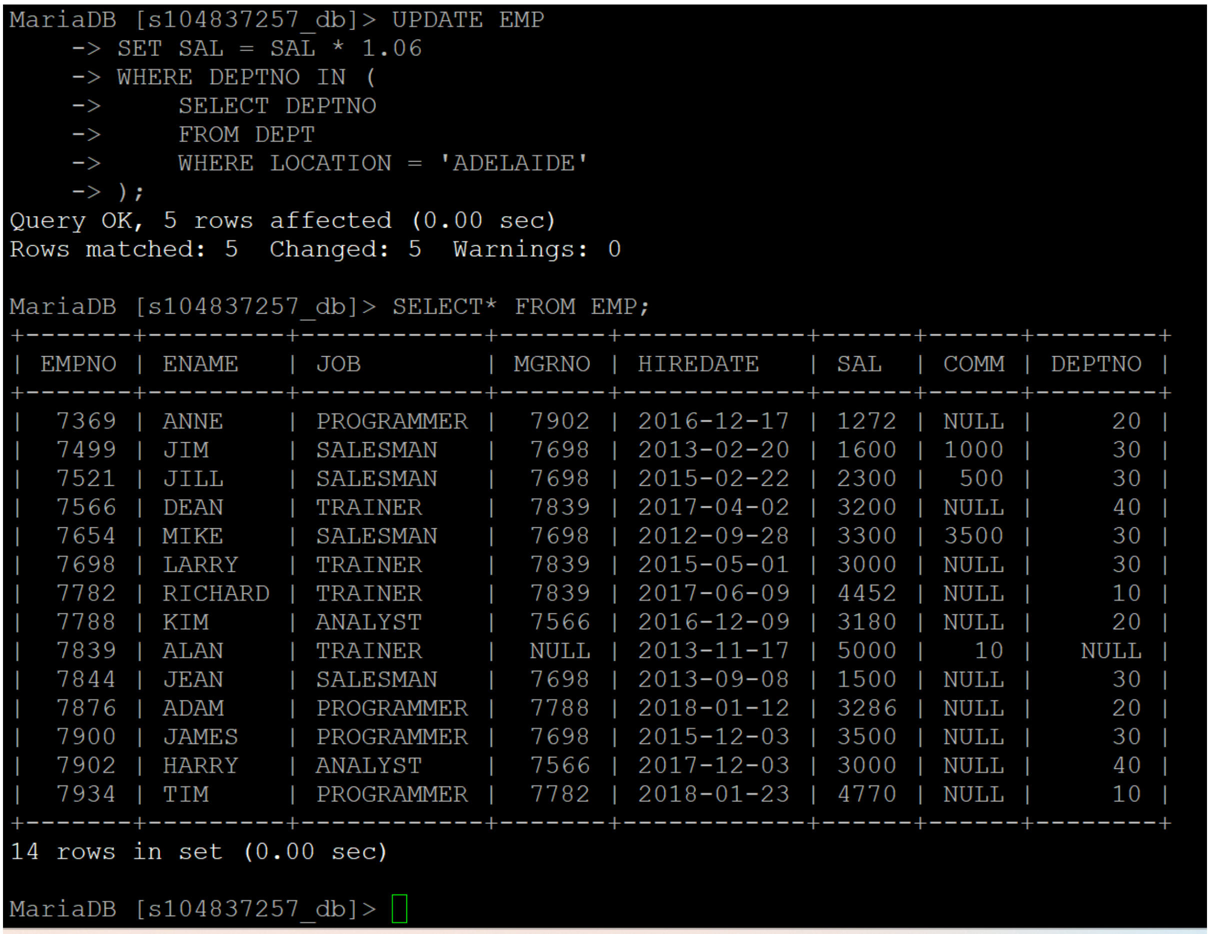
```
MariaDB [s104837257_db]> SELECT D.DNAME, COUNT(E.EMPNO) AS NumEmployees, AVG(E.SAL) AS AvgSalary
    -> FROM DEPT D
    -> JOIN EMP E ON D.DEPTNO = E.DEPTNO
    -> GROUP BY D.DNAME
    -> HAVING COUNT(E.EMPNO) = (
    ->     SELECT MAX(EmpCount)
    ->     FROM (
    ->         SELECT COUNT(EMPNO) AS EmpCount
    ->         FROM EMP
    ->         GROUP BY DEPTNO
    ->     ) AS EmpCountInDept
    -> );
+-----------+--------------+-----------+
| DNAME     | NumEmployees | AvgSalary |
+-----------+--------------+-----------+
| MARKETING |            6 | 2533.3333 |
+-----------+--------------+-----------+
1 row in set (0.01 sec)

MariaDB [s104837257_db]> 
```

2.6 Update for those employees who work for the departments located in “ADELAIDE” by increasing their salaries by 6%. Show all columns of the EMP table after the update:

UPDATE EMP
SET SAL = SAL * 1.06
WHERE DEPTNO **IN (**
            SELECT DEPTNO
            FROM DEPT
            WHERE LOCATION = ’ADELAIDE’
);

SELECT * FROM EMP;



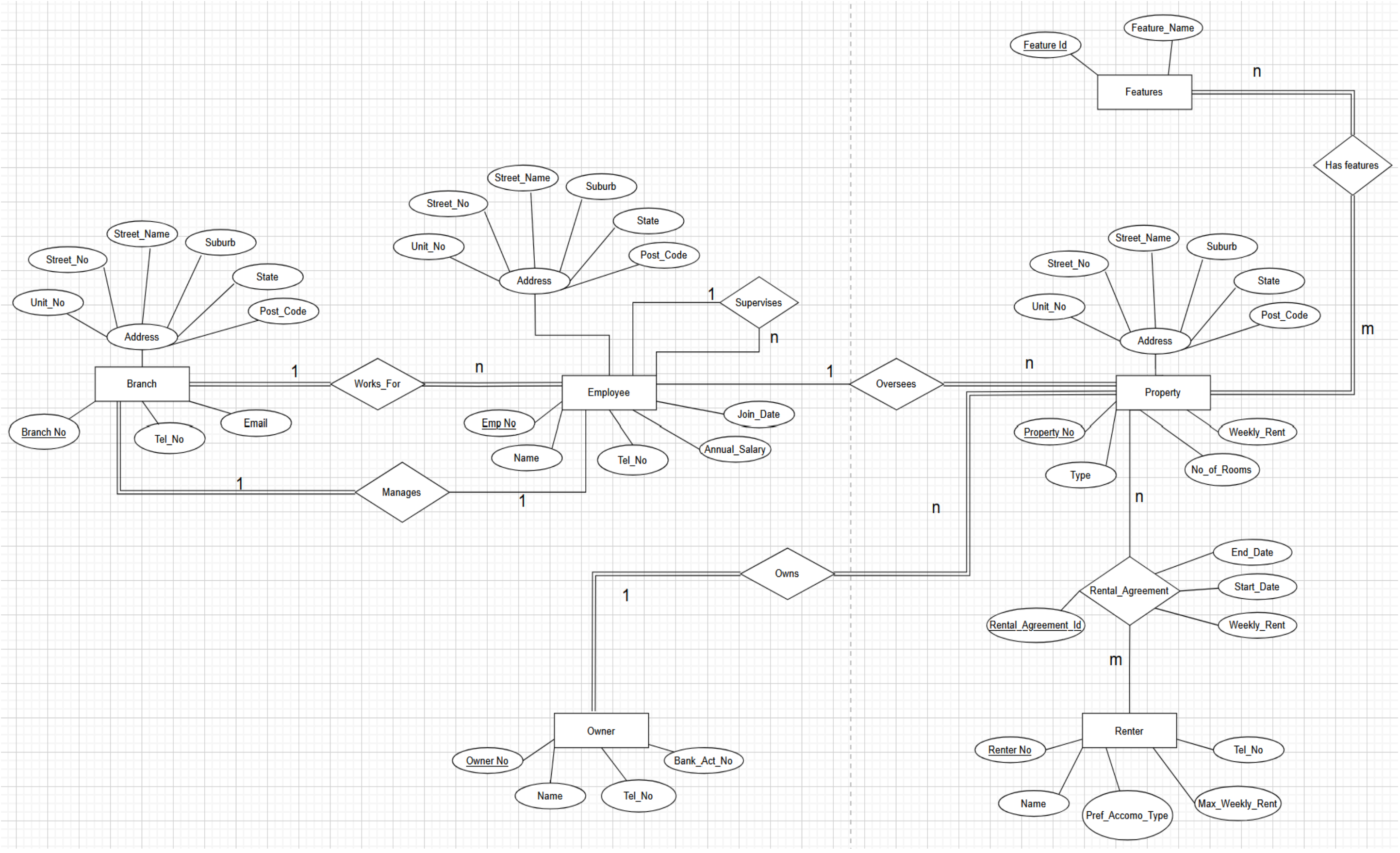
```
MariaDB [s104837257_db]> UPDATE EMP
    -> SET SAL = SAL * 1.06
    -> WHERE DEPTNO IN (
    ->     SELECT DEPTNO
    ->     FROM DEPT
    ->     WHERE LOCATION = 'ADELAIDE'
    -> );
Query OK, 5 rows affected (0.00 sec)
Rows matched: 5  Changed: 5  Warnings: 0

MariaDB [s104837257_db]> SELECT* FROM EMP;
+-------+---------+------------+-------+------------+------+------+--------+
| EMPNO | ENAME   | JOB        | MGRNO | HIREDATE   | SAL  | COMM | DEPTNO |
+-------+---------+------------+-------+------------+------+------+--------+
|  7369 | ANNE    | PROGRAMMER |  7902 | 2016-12-17 | 1272 | NULL |     20 |
|  7499 | JIM     | SALESMAN   |  7698 | 2013-02-20 | 1600 | 1000 |     30 |
|  7521 | JILL    | SALESMAN   |  7698 | 2015-02-22 | 2300 |  500 |     30 |
|  7566 | DEAN    | TRAINER    |  7839 | 2017-04-02 | 3200 | NULL |     40 |
|  7654 | MIKE    | SALESMAN   |  7698 | 2012-09-28 | 3300 | 3500 |     30 |
|  7698 | LARRY   | TRAINER    |  7839 | 2015-05-01 | 3000 | NULL |     30 |
|  7782 | RICHARD | TRAINER    |  7839 | 2017-06-09 | 4452 | NULL |     10 |
|  7788 | KIM     | ANALYST    |  7566 | 2016-12-09 | 3180 | NULL |     20 |
|  7839 | ALAN    | TRAINER    |  NULL | 2013-11-17 | 5000 |   10 |   NULL |
|  7844 | JEAN    | SALESMAN   |  7698 | 2013-09-08 | 1500 | NULL |     30 |
|  7876 | ADAM    | PROGRAMMER |  7788 | 2018-01-12 | 3286 | NULL |     20 |
|  7900 | JAMES   | PROGRAMMER |  7698 | 2015-12-03 | 3500 | NULL |     30 |
|  7902 | HARRY   | ANALYST    |  7566 | 2017-12-03 | 3000 | NULL |     40 |
|  7934 | TIM     | PROGRAMMER |  7782 | 2018-01-23 | 4770 | NULL |     10 |
+-------+---------+------------+-------+------------+------+------+--------+
14 rows in set (0.00 sec)

MariaDB [s104837257_db]>
```

Task 2:

a) ER Diagram Drawing for BestChoice Mortgage RDBMS system



b) Now, from the ER diagram a relational schema is derived and the primary keys and foreign keys are marked.

Relation schema for BestChoice:

1. Branch (**Branch_No**, Tel_No, Email, Unit_No, Street_No, Street_Name, Suburb, State, Post_Code, **Mgr_Emp_No,**)
   - Primary Key: Branch_No
   - Foreign Key: Mgr_Emp_No – references – Employee (Emp_No)

2. Employee (**Emp_No**, Name, Tel_No, Unit_No, Street_No, Street_Name, Suburb, State, Post_Code, Annual_salary, Supervisor_Emp_No, Join_Date, **Branch_No**)
   - Primary Key: Emp_No
   - Foreign Key: Branch_No - references – Branch (Branch_No)

3. Property (**Property_No**, Type, Unit_No, Street_No, Street_Name, Suburb, State, Post_Code, No_of_Rooms, Weekly_Rent, **Emp_No, Owner_No**)
   - Primary Key: Property_No
   - Foreign Key: Emp_No – references – Employee (Emp_No)
   - Foreign Key: Owner_No – reference – Owner (Owner_No)

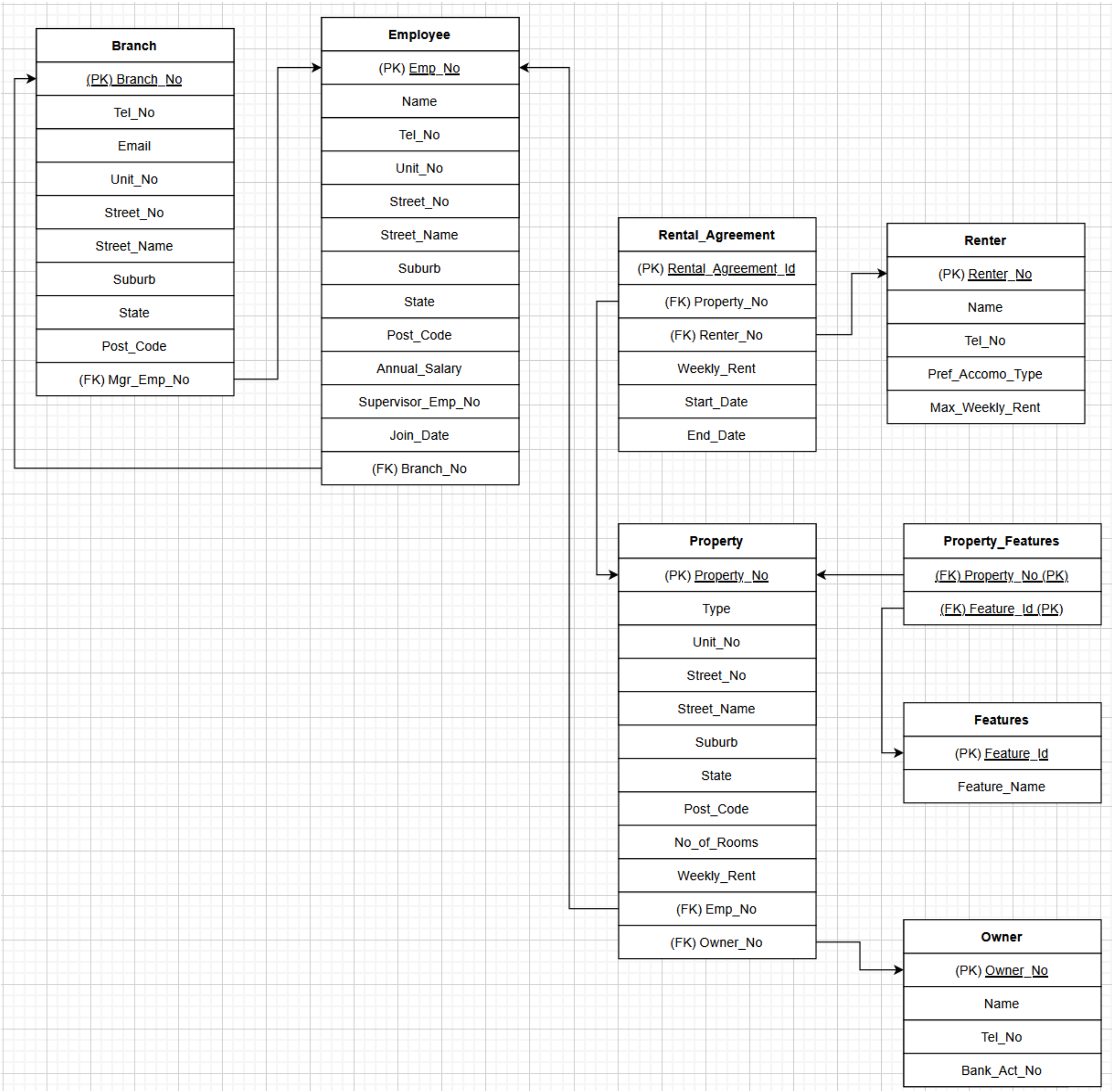
4. Features (**Feature_Id,** Feature_Name)
   - Primary Key: Feature_Id

5. Property_Features (**Property_No, Feature_Id**)
   - Primary Key: (Property_No, Feature_Id)
   - Foreign Key: Property_No – references - Property (Property_No)
   - Foreign Key: Feature_Id – references - Features (Feature_Id)

6. Owner (**Owner_No**, Name, Tel_No, Bank_Act_No)
   - Primary Key: Owner_No

7. Renter (**Renter_No**, Name, Tel_No, Pref_Accomo_Type, Max_Weekly_Rent)
   - Primary Key: Renter_No

8. Rental_Agreement (**Rental_Agreement_Id, Property_No, Renter_No,** Weekly_rent, Start_Date, End_Date)
   - Primary Key: **Rental_Agreement_Id**
   - Foreign Key: Property_No – references - Property (Property_No)
   - Foreign Key: Renter_No – references – Renters (Renter_No)


Branch
(PK) Branch_No
Tel_No
Email
Unit_No
Street_No
Street_Name
Suburb
State
Post_Code
(FK) Mgr_Emp_No
Employee
(PK) Emp_No
Name
Tel_No
Unit_No
Street_No
Street_Name
Suburb
State
Post_Code
Annual_Salary
Supervisor_Emp_No
Join_Date
(FK) Branch_No
Rental_Agreement
(PK) Rental_Agreement_Id
(FK) Property_No
(FK) Renter_No
Weekly_Rent
Start_Date
End_Date
Renter
(PK) Renter_No
Name
Tel_No
Pref_Accomo_Type
Max_Weekly_Rent
Property
(PK) Property_No
Type
Unit_No
Street_No
Street_Name
Suburb
State
Post_Code
No_of_Rooms
Weekly_Rent
(FK) Emp_No
(FK) Owner_No
Property_Features
(FK) Property_No (PK)
(FK) Feature_Id (PK)
Features
(PK) Feature_Id
Feature_Name
Owner
(PK) Owner_No
Name
Tel_No
Bank_Act_No