**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:**

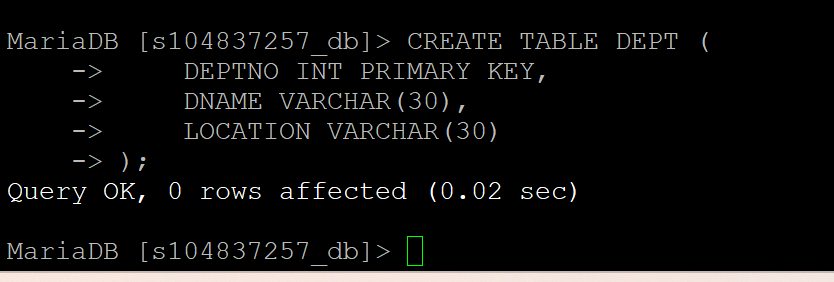
**CREATE** **TABLE** DEPT (

DEPTNO INT PRIMARY KEY,

DNAME VARCHAR(30),

LOCATION VARCHAR(30)

);



* **Populating DEPT table:**

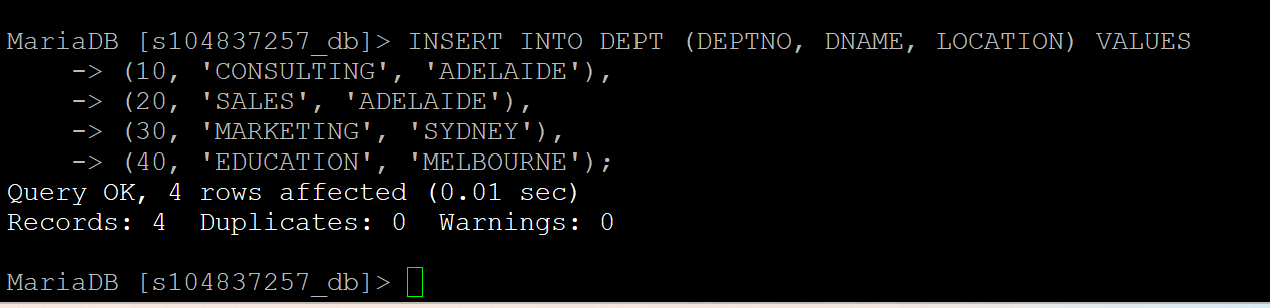
**INSERT INTO** DEPT (DEPTNO, DNAME, LOCATION)

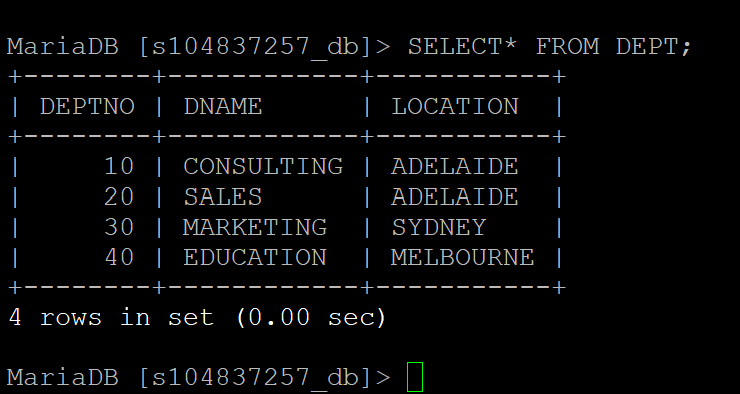
**VALUES** (10, 'CONSULTING', 'ADELAIDE'),

(20, 'SALES', 'ADELAIDE'),

(30, 'MARKETING', 'SYDNEY'),

(40, 'EDUCATION', 'MELBOURNE');





* **Creating EMP table:**

**CREATE TABLE** EMP (

EMPNO INT PRIMARY KEY,

ENAME VARCHAR(30),

JOB VARCHAR(30),

MGRNO INT,

HIREDATE DATE,

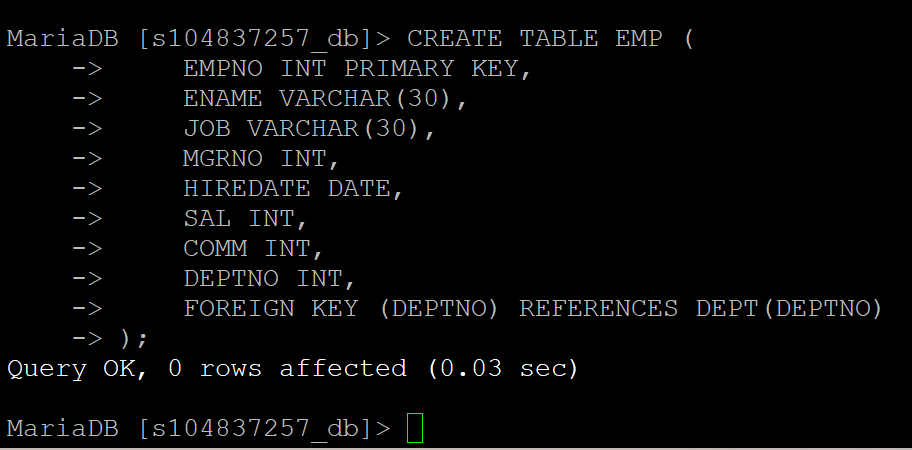
SAL INT,

COMM INT,

DEPTNO INT,

FOREIGN KEY (DEPTNO) REFERENCES DEPT(DEPTNO)

);



* **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),

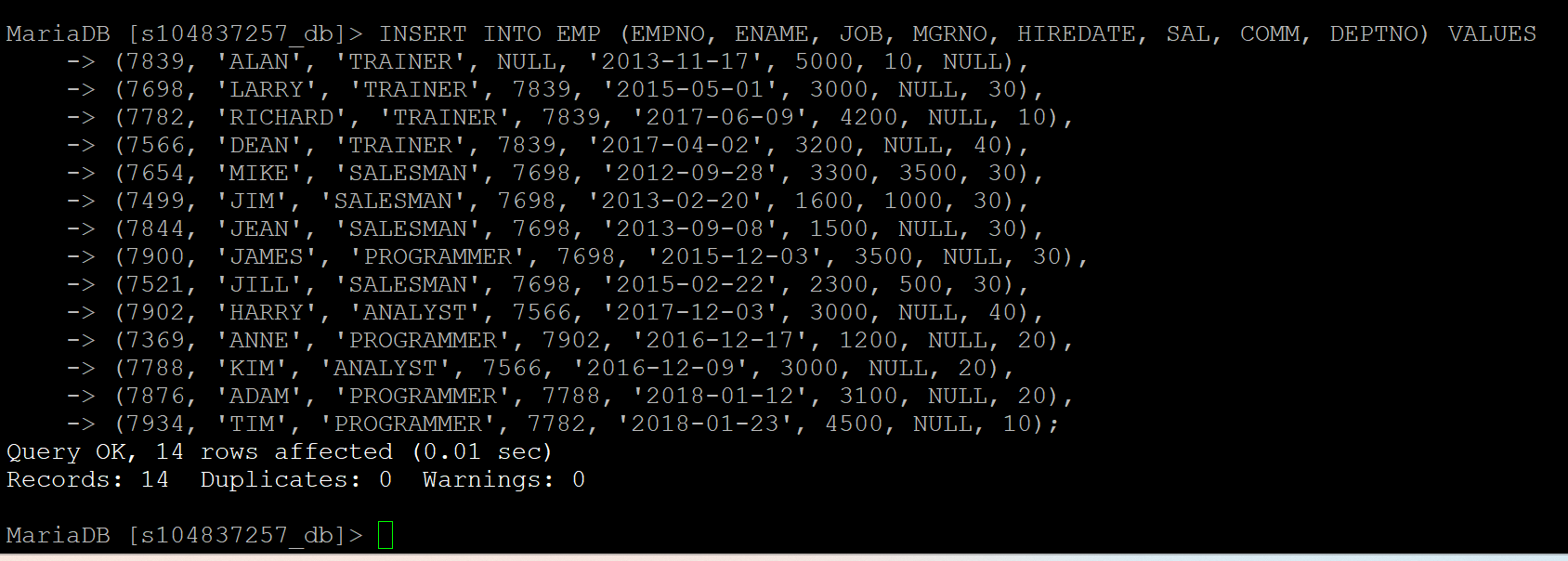
(7902, 'HARRY', 'ANALYST', 7566, '2017-12-03', 3000, NULL, 40),

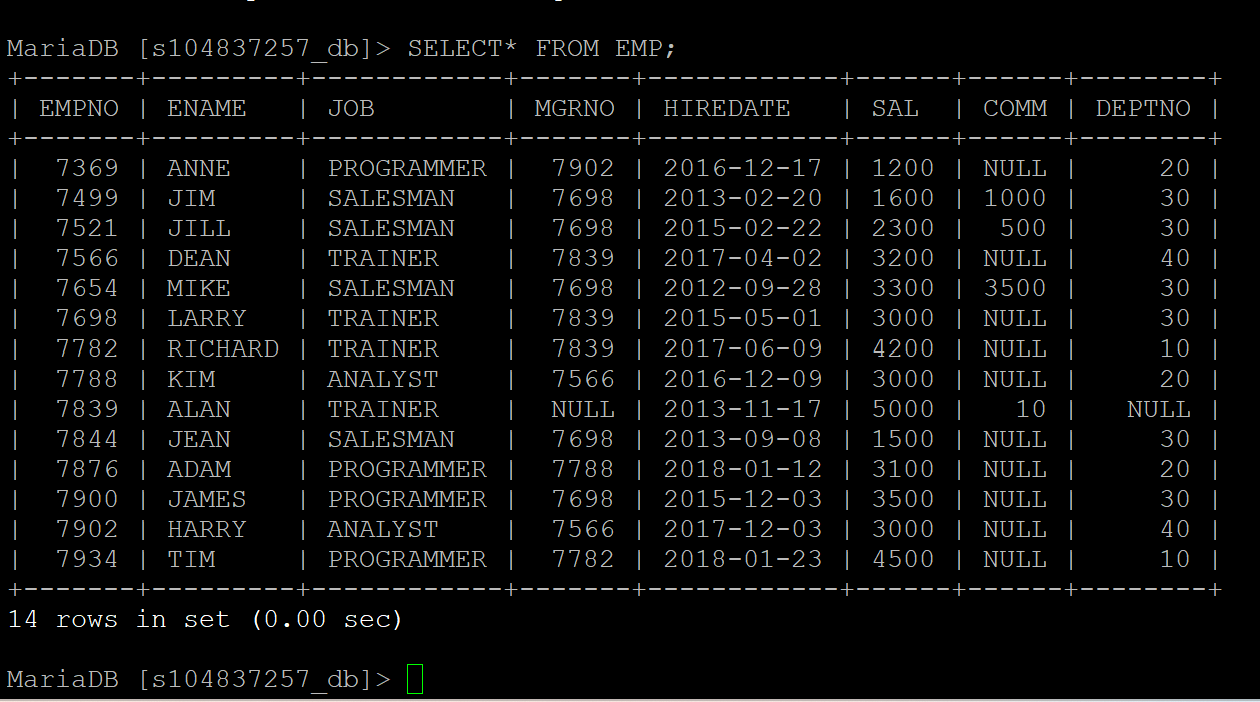
(7369, 'ANNE', 'PROGRAMMER', 7902, '2016-12-17', 1200, NULL, 20),

(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);

****

****

**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.**

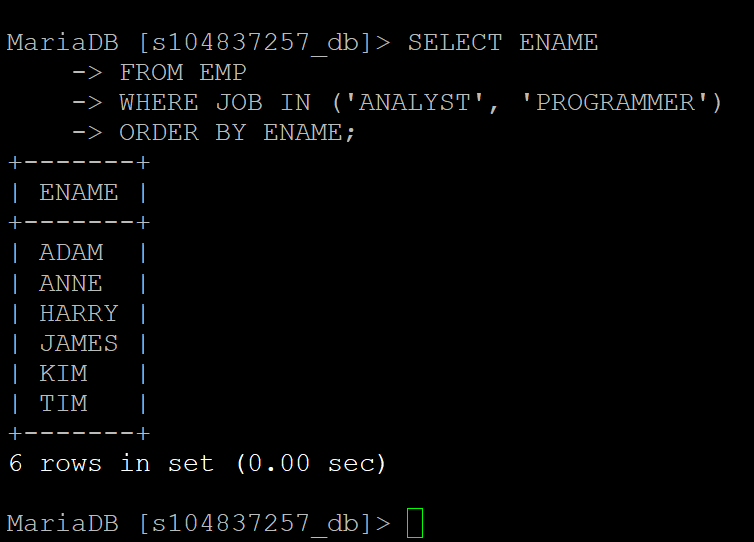
**2.2) Find employees (ENAME) whose job (JOB) is either ANALYST or PROGRAMMER, ordered by employee name:**

**SELECT** ENAME

**FROM** EMP

**WHERE** JOB **IN** ('ANALYST', 'PROGRAMMER')

**ORDER BY** ENAME;

****

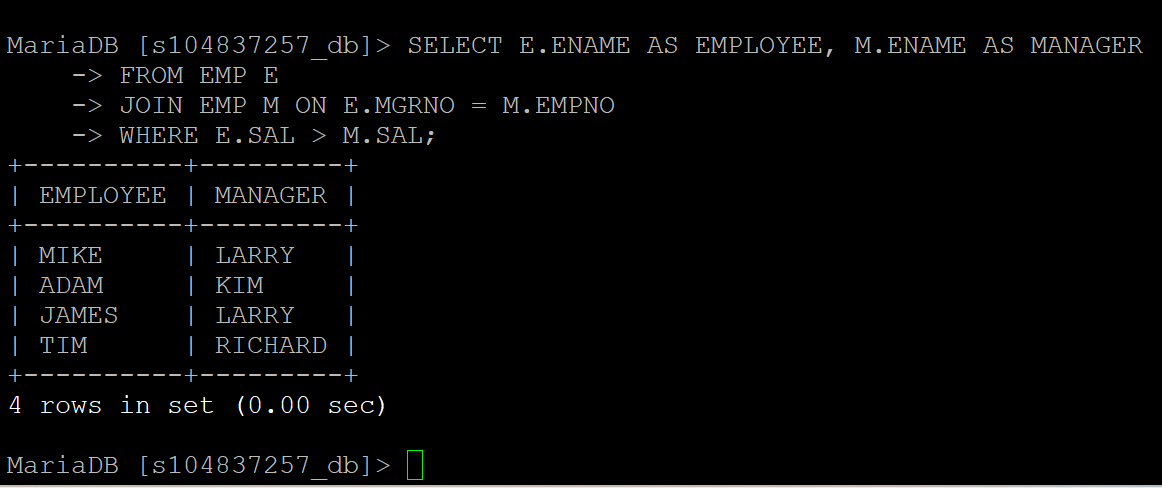
**2.3) Find employees whose salary (SAL) is higher than their manager’s salary. List name of both employees (ENAME) and their managers (rename as MNAME):**

**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;



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

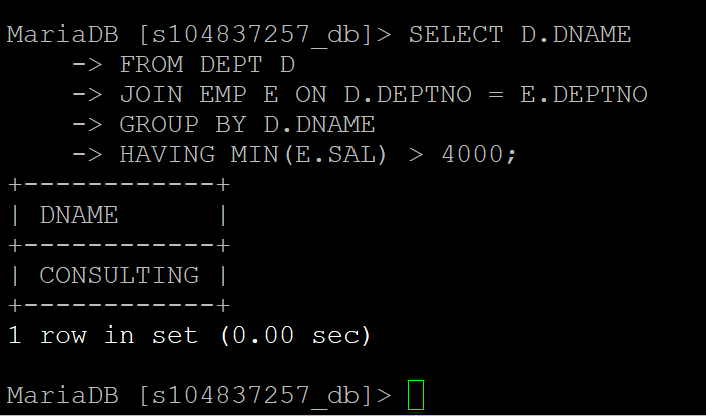
**SELECT** D.DNAME

**FROM** DEPT D

**JOIN** EMP E **ON** D.DEPTNO = E.DEPTNO

**GROUP** BY D.DNAME

**HAVING** MIN(E.SAL) > 4000;



**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** (

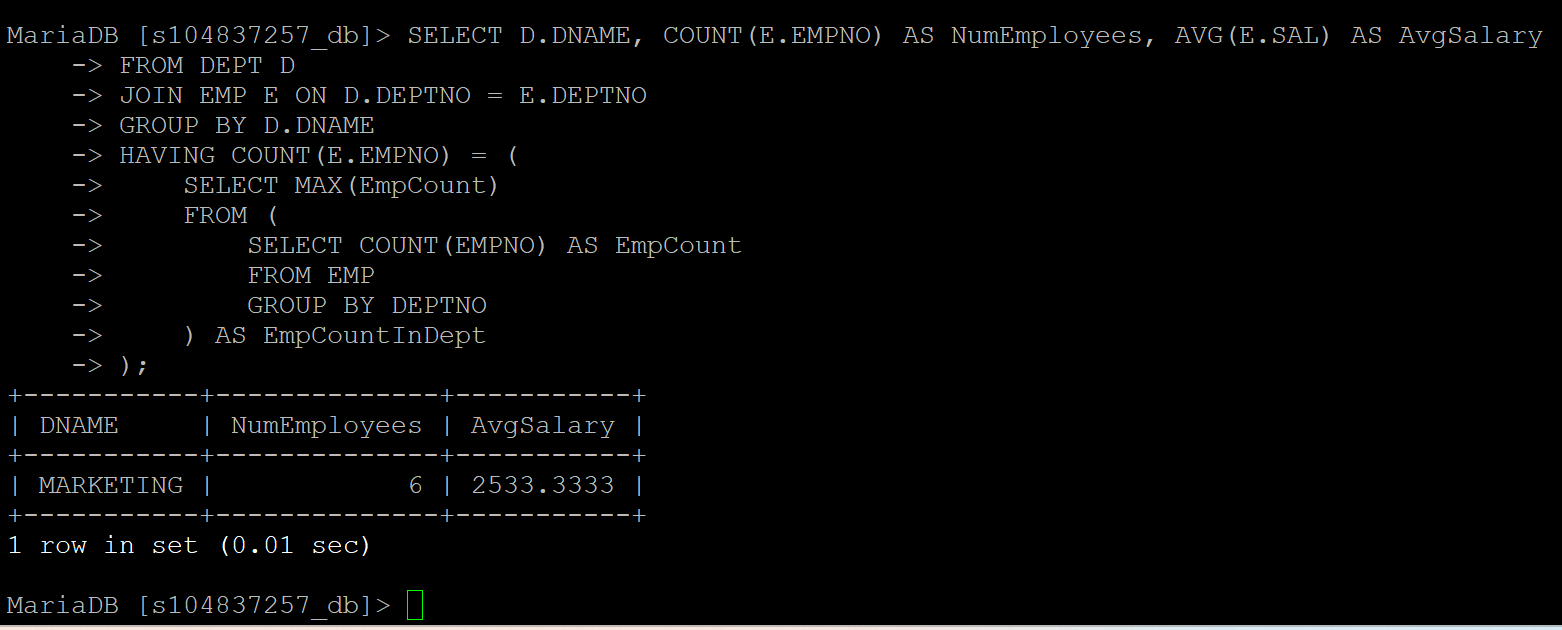
**SELECT** COUNT(EMPNO) AS EmpCount

**FROM** EMP

**GROUP BY** DEPTNO

) **AS** EmpCountInDept

);



**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 (**

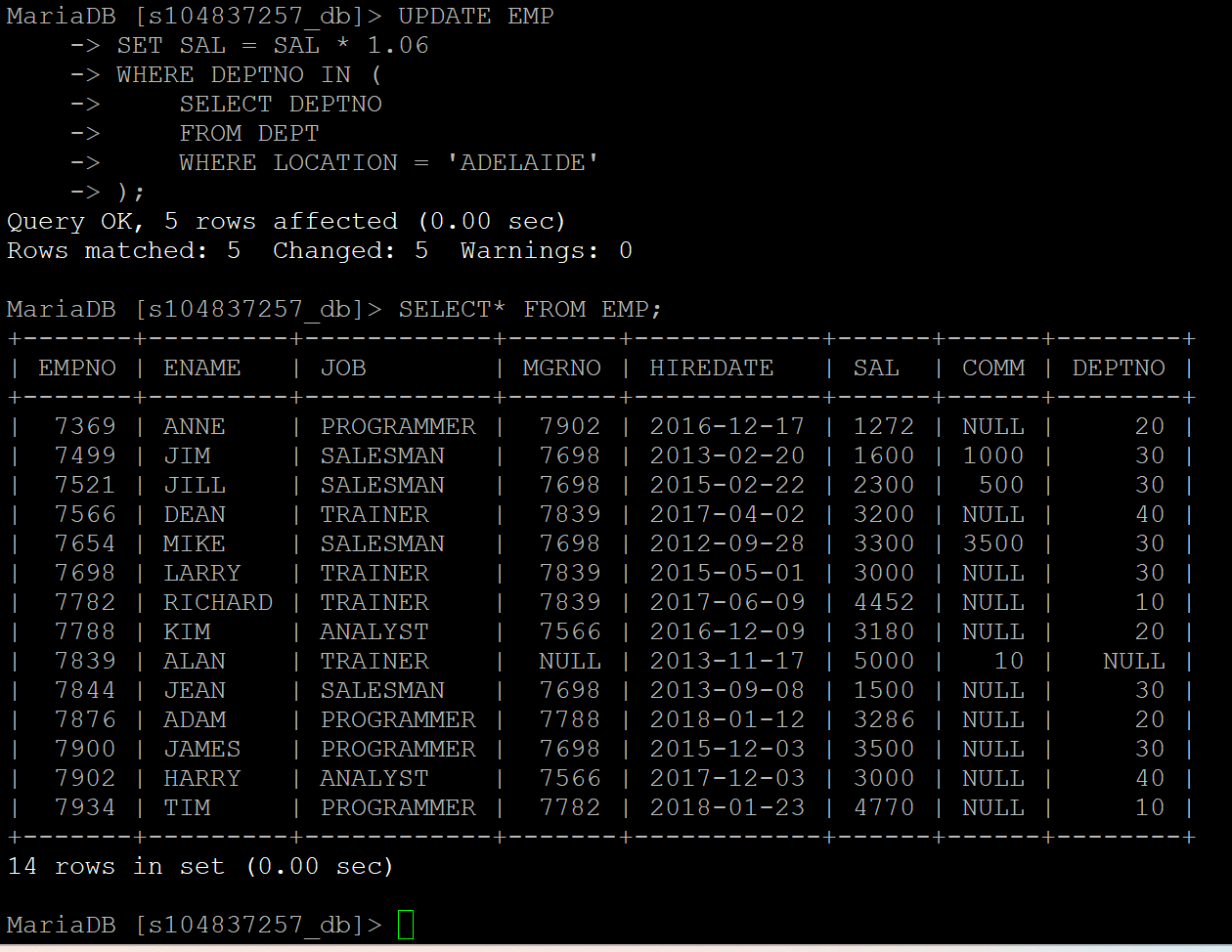
**SELECT** DEPTNO

**FROM** DEPT

**WHERE** LOCATION = ’ADELAIDE’

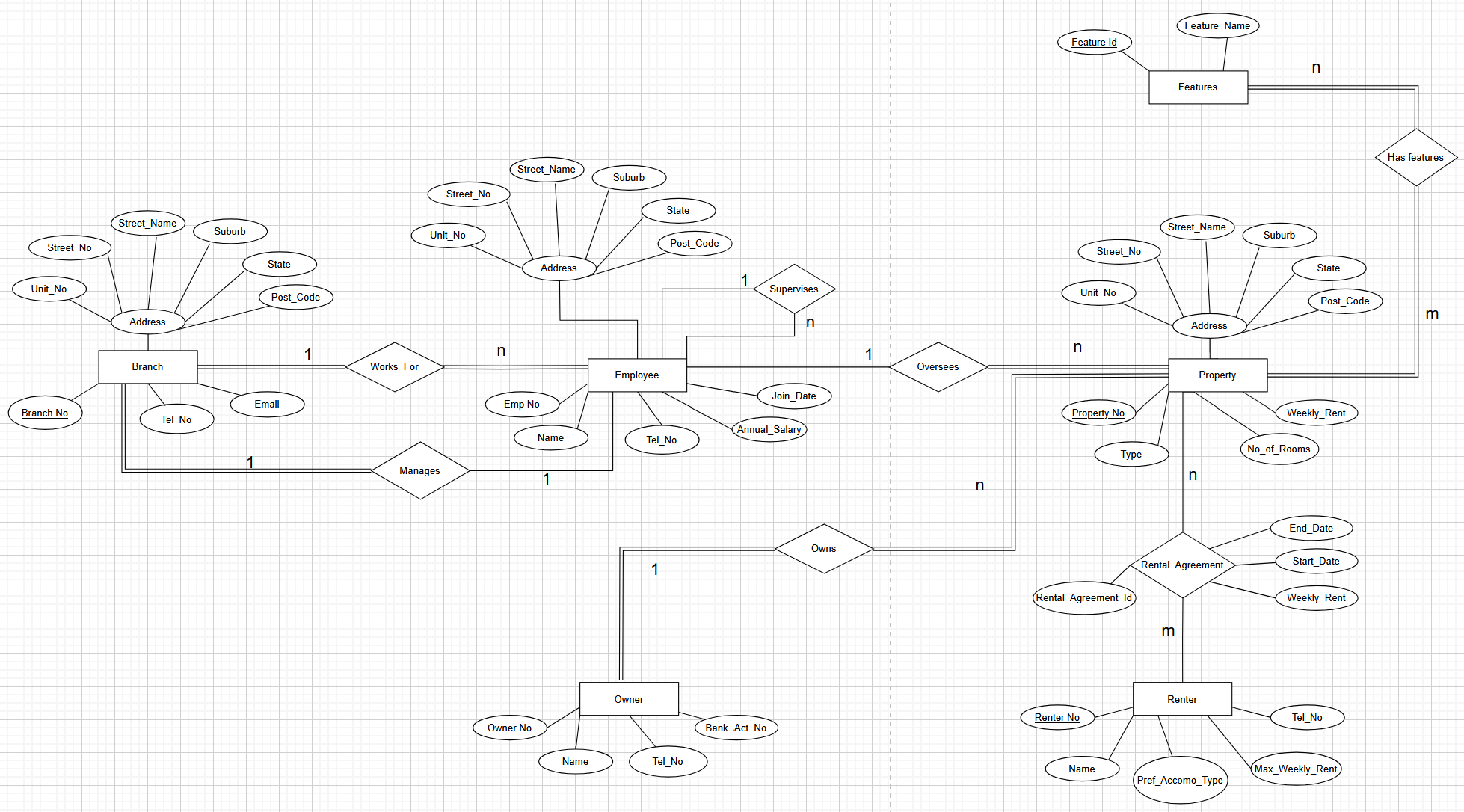
**);**

**SELECT \*** FROM EMP;



**Task 2:**

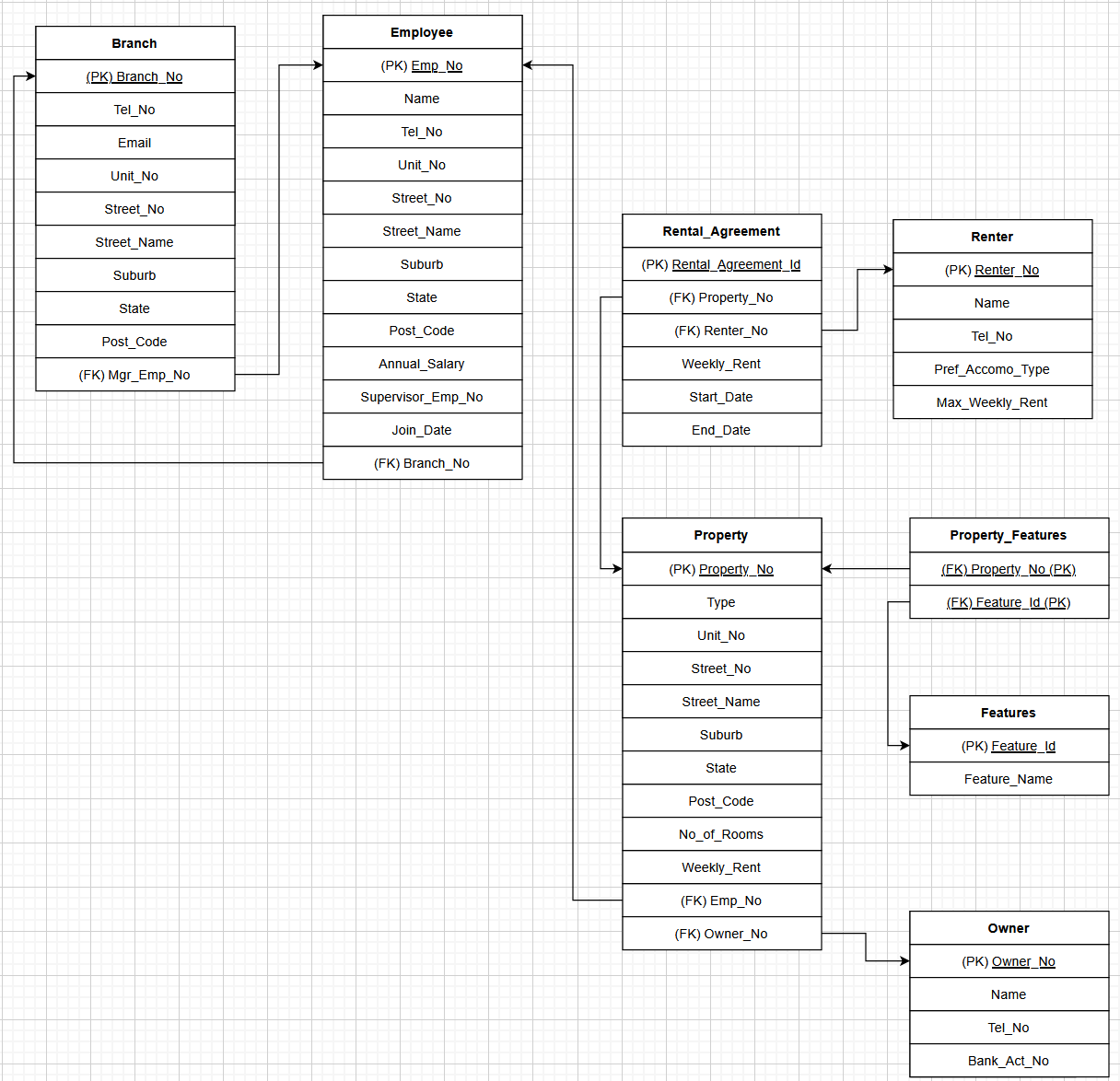
**a) ER Diagram Drawing for BestChoice Mortgage RDBMS system**

****

**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)

****