Name: Reethikka.S.S
106122100

#Session: 04 || Date: 08/08/2024

Moodle Due: 11/08/2024 at 06 PM

Relational Database Design – Company Schema – Nested Queries

Reference: Elmasri, R., 2018. Fundamentals of database systems. Pearson Education India.

1. With continuation to Session 03 exercise, execute all the example queries provided in Subsection 7.1.1 to 7.4.2 (excluding keywords ‘TRIGGER’, ‘VIEW’, ‘EXCEPT’ and ‘CONTAINS’).

IS, IS NOT:

```
mysql> SELECT Fname,Lname from EMPLOYEE WHERE Super_ssn IS NULL;
Empty set (0.00 sec)

mysql> SELECT Fname,Lname from EMPLOYEE WHERE Super_ssn IS NOT NULL;
+------------+--------+
| Fname      | Lname  |
+------------+--------+
| Richard    | Marini |
| Richardson | Mar    |
| Richson    | Mario  |
| Rich       | Mario  |
| Richton    | Mario  |
| Ramsay     | Marini |
+------------+--------+
6 rows in set (0.00 sec)
```

NESTED QUERIES,TUPLES,MULTISET COMPARISONS, CORRELATED NESTED QUERIES

The first nested query selects the project numbers of projects that have an employee with last name ‘Mar’ involved as manager, whereas the second nested query selects the project numbers of projects that have an employee with last name ‘Mar’ involved as worker.

```
mysql> SELECT DISTINCT Pnumber FROM PROJECT WHERE Pnumber IN ( SELECT Pnumber
FROM PROJECT,DEPARTMENT,EMPLOYEE WHERE Dnum=Dnumber AND Mgr_ssn=Ssn AND
Lname="Mar") OR Pnumber IN (SELECT Pno FROM WORKS_ON,EMPLOYEE WHERE Essn=Ssn
AND Lname="Mar");
+---------+
| Pnumber |
+---------+
|       2 |
+---------+
```

1 row in set (0.00 sec)

This query will select the Essns of all employees who work the same (project, hours) combination on some project that employee ‘Richard Marini’ (whose Ssn = ‘653298653’)

```
mysql> SELECT DISTINCT Essn FROM WORKS_ON WHERE (Pno,Hours) IN (SELECT Pno,Hours FROM WORKS_ON WHERE Essn="653298653");
+-----------+
| Essn      |
+-----------+
| 653298653 |
+-----------+
1 row in set (0.03 sec)
```

An example is the following query, which returns the names of employees whose salary is greater than the salary of all the employees in department 5:(Salary must be greater than all the selected tuples).

```
mysql> SELECT Fname,Lname FROM EMPLOYEE WHERE Salary> ALL( SELECT Salary FROM EMPLOYEE WHERE Dno=5);
+------------+--------+
| Fname      | Lname  |
+------------+--------+
| Richard    | Marini |
| Richardson | Mar    |
| Richson    | Mario  |
| Rich       | Mario  |
| Richton    | Mario  |
| Ramsay     | Marini |
+------------+--------+
6 rows in set (0.00 sec)

mysql> SELECT Fname,Lname FROM EMPLOYEE WHERE Salary> ALL( SELECT Salary FROM EMPLOYEE WHERE Dno=4);
Empty set (0.00 sec)
```

Retrieve the name of each employee who has a dependent with the same first name and is the same sex as the employee.

```
mysql> SELECT Fname,Lname FROM EMPLOYEE,DEPENDENT WHERE Fname=Dependent_name && EMPLOYEE.Sex=DEPENDENT.Sex;
Empty set, 1 warning (0.02 sec)

mysql> UPDATE DEPENDENT SET Dependent_name="Rich" WHERE Essn="653298656";
Query OK, 1 row affected (0.03 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT Fname,Lname FROM EMPLOYEE,DEPENDENT WHERE Fname=Dependent_name && EMPLOYEE.Sex=DEPENDENT.Sex;
+-------+-------+
```

| Fname | Lname |
+-------+-------+
| Rich | Mario |
+-------+-------+
1 row in set, 1 warning (0.00 sec)

mysql> SELECT E.Fname,E.Lname FROM EMPLOYEE AS E WHERE E.Ssn IN (SELECT D.Essn FROM DEPENDENT AS D WHERE E.fname=D.Dependent_name AND D.Sex=E.Sex);
+-------+-------+
| Fname | Lname |
+-------+-------+
| Rich | Mario |
+-------+-------+
1 row in set (0.00 sec)

EXISTS AND UNIQUE

mysql> SELECT E.Fname,E.Lname FROM EMPLOYEE AS E WHERE EXISTS (SELECT D.Essn FROM DEPEND
ENT AS D WHERE E.fname=D.Dependent_name AND D.Sex=E.Sex);
+-------+-------+
| Fname | Lname |
+-------+-------+
| Rich | Mario |
+-------+-------+
1 row in set (0.00 sec)

Retrieve the names of employees who have no dependents.
mysql> SELECT Fname,Lname FROM EMPLOYEE WHERE NOT EXISTS(SELECT * FROM DEPENDENT WHERE Ssn=Essn);
+--------+--------+
| Fname | Lname |
+--------+--------+
| Ramsay | Marini |
+--------+--------+
1 row in set (0.00 sec)

List the names of managers who have at least one dependent.
mysql> SELECT Fname,Lname FROM EMPLOYEE WHERE EXISTS(SELECT * FROM DEPARTMEN
T WHERE Ssn=Mgr_ssn) AND EXISTS(SELECT * FROM DEPENDENT WHERE Ssn=Essn);
+------------+--------+
| Fname | Lname |
+------------+--------+
Richard	Marini
Richardson	Mar
Richson	Mario
Rich	Mario

| Richton | Mario |
+----------+--------+
5 rows in set (0.00 sec)

List the names of managers who have at least one dependent.
mysql> SELECT Fname,Lname FROM EMPLOYEE WHERE EXISTS(SELECT * FROM DEPENDENT WHERE Essn=Ssn) AND EXISTS(SELECT * FROM DEPARTMENT WHERE Ssn=Mgr_ssn);
+----------+--------+
| Fname | Lname |
+----------+--------+
Richard	Marini
Richardson	Mar
Richson	Mario
Rich	Mario
Richton	Mario
+----------+--------+
5 rows in set (0.00 sec)

Retrieve the name of each employee who works on all the projects con-
trolled by department number 5
mysql> SELECT Fname,Lname FROM EMPLOYEE WHERE NOT EXISTS(SELECT * FROM WORKS_ON B WHERE(B.Pno IN (SELECT Pnumber FROM PROJECT WHERE Dnum=5) AND NOT EXISTS(SELECT * FROM WORKS_ON C WHERE C.Essn=Ssn AND C.Pno=B.Pno)));
+---------+-------+
| Fname | Lname |
+---------+-------+
| Richton | Mario |
+---------+-------+
1 row in set (0.00 sec)

EXPLICIT SETS AND RENAMING

Retrieve the Social Security numbers of all employees who work on
project numbers 1, 2, or 3.
mysql> SELECT DISTINCT Essn FROM WORKS_ON WHERE Pno IN (1,2,3);
+----------+
| Essn |
+----------+
| 653298653 |
| 653298654 |
| 653298655 |
+----------+
3 rows in set (0.00 sec)

retrieve the last name of each
employee and his or her supervisor

```
mysql> SELECT E.FNAME AS "EMPLOYEE_NAME",S.LNAME AS "SUPERVISOR_NAME"
FROM E
MPLOYEE E,EMPLOYEE S WHERE E.SUPER_SSN=S.SSN;
+---------------+-----------------+
| EMPLOYEE_NAME | SUPERVISOR_NAME |
+---------------+-----------------+
| Richard       | Marini          |
| Richardson    | Marini          |
| Richson       | Marini          |
| Rich          | Marini          |
| Richton       | Marini          |
| Ramsay        | Marini          |
+---------------+-----------------+
6 rows in set (0.00 sec)
```

retrieves the name and address of every employee who works for the
'IT' department.

```
mysql> SELECT FNAME,LNAME,ADDRESS FROM (EMPLOYEE JOIN DEPARTMENT ON
DNO=DNUMBER) WHERE DNAME="IT";
+------------+--------+-----------------------+
| FNAME      | LNAME  | ADDRESS               |
+------------+--------+-----------------------+
| Richard    | Marini | 98 Oak Forest,Katy,TX  |
| Richardson | Mar    | 99 Oak Forest,Katy,TX  |
| Richson    | Mario  | 100 Oak Forest,Katy,TX |
| Rich       | Mario  | 101 Oak Forest,Katy,TX |
| Richton    | Mario  | 102 Oak Forest,Katy,TX |
| Ramsay     | Marini | 98 Oak Forest,Katy,TX  |
+------------+--------+-----------------------+
6 rows in set (0.00 sec)
```

Renaming columns

```
mysql> SELECT E.Lname AS Employee_name,S.Lname as Supervisor_name FROM EMPLOYEE AS
E,EMPLOYEE AS S WHERE E.Super_ssn=S.ssn;
+---------------+-----------------+
| Employee_name | Supervisor_name |
+---------------+-----------------+
| Marini        | Marini          |
| Mar           | Marini          |
| Mario         | Marini          |
| Mario         | Marini          |
| Mario         | Marini          |
| Marini        | Marini          |
+---------------+-----------------+
6 rows in set (0.00 sec)
```

JOINING TABLES

query Q1,
which retrieves the name and address of every employee who works for the

‘Research’ department.

```
mysql> SELECT Fname,Lname ,Address FROM(EMPLOYEE JOIN DEPARTMENT ON
Dno=Dnumber) WHERE Dname="Research";
Empty set (0.00 sec)

mysql> SELECT Fname,Lname ,Address FROM(EMPLOYEE JOIN DEPARTMENT ON
Dno=Dnumber) WHERE Dname="IT";
+------------+--------+----------------------+
| Fname      | Lname  | Address              |
+------------+--------+----------------------+
| Richard    | Marini | 98 Oak Forest,Katy,TX  |
| Richardson | Mar    | 99 Oak Forest,Katy,TX  |
| Richson    | Mario  | 100 Oak Forest,Katy,TX |
| Rich       | Mario  | 101 Oak Forest,Katy,TX |
| Richton    | Mario  | 102 Oak Forest,Katy,TX |
| Ramsay     | Marini | 98 Oak Forest,Katy,TX  |
+------------+--------+----------------------+
6 rows in set (0.00 sec)

mysql> SELECT Fname,Lname,Address FROM (EMPLOYEE NATURAL JOIN(DEPARTMENT AS
DEPT(Dname,Dno,Mssn,Msdate))) WHERE Dname="Research";

SELECT
E.Lname AS Employee_name,
S.Lname AS Supervisor_name
FROM
(EMPLOYEE AS E LEFT OUTER JOIN EMPLOYEE AS S
ON E.Super_ssn = S.Ssn);


mysql> SELECT FNAME,LNAME,ADDRESS FROM (EMPLOYEE JOIN DEPARTMENT ON
DNO=DNUMBER) WHERE DNAME="IT";
+------------+--------+----------------------+
| FNAME      | LNAME  | ADDRESS              |
+------------+--------+----------------------+
| Richard    | Marini | 98 Oak Forest,Katy,TX  |
| Richardson | Mar    | 99 Oak Forest,Katy,TX  |
| Richson    | Mario  | 100 Oak Forest,Katy,TX |
| Rich       | Mario  | 101 Oak Forest,Katy,TX |
| Richton    | Mario  | 102 Oak Forest,Katy,TX |
| Ramsay     | Marini | 98 Oak Forest,Katy,TX  |
+------------+--------+----------------------+
6 rows in set (0.00 sec)

mysql> SELECT E.FNAME AS "EMPLOYEE_NAME",S.FNAME AS "SUPERVISOR_NAME"
FROM (
EMPLOYEE AS E LEFT OUTER JOIN EMPLOYEE AS S ON E.SUPER_SSN=S.SSN);
+---------------+----------------+
| EMPLOYEE_NAME | SUPERVISOR_NAME |
```

```
+--------------+----------------+
| Richard      | Richard        |
| Richardson   | Richard        |
| Richson      | Richard        |
| Rich         | Richard        |
| Richton      | Richard        |
| Ramsay       | Richard        |
+--------------+----------------+
6 rows in set (0.00 sec)

mysql> SELECT E.FNAME AS "EMPLOYEE_NAME",S.FNAME AS "SUPERVISOR_NAME"
FROM (EMPLOYEE AS E RIGHT OUTER JOIN EMPLOYEE AS S ON E.SUPER_SSN=S.SSN);
+--------------+----------------+
| EMPLOYEE_NAME | SUPERVISOR_NAME |
+--------------+----------------+
| Ramsay       | Richard        |
| Richton      | Richard        |
| Rich         | Richard        |
| Richson      | Richard        |
| Richardson   | Richard        |
| Richard      | Richard        |
| NULL         | Richardson     |
| NULL         | Richson        |
| NULL         | Rich           |
| NULL         | Richton        |
| NULL         | Ramsay         |
+--------------+----------------+
11 rows in set (0.00 sec)

mysql> SELECT E.FNAME AS "EMPLOYEE_NAME",S.FNAME AS "SUPERVISOR_NAME"
FROM (EMPLOYEE AS E CROSS JOIN EMPLOYEE AS S);
+--------------+----------------+
| EMPLOYEE_NAME | SUPERVISOR_NAME |
+--------------+----------------+
| Ramsay       | Richard        |
| Richton      | Richard        |
| Rich         | Richard        |
| Richson      | Richard        |
| Richardson   | Richard        |
| Richard      | Richard        |
| Ramsay       | Richardson     |
| Richton      | Richardson     |
| Rich         | Richardson     |
| Richson      | Richardson     |
| Richardson   | Richardson     |
| Richard      | Richardson     |
| Ramsay       | Richson        |
| Richton      | Richson        |
| Rich         | Richson        |
```

```
| Richson       | Richson         |
| Richardson    | Richson         |
| Richard       | Richson         |
| Ramsay        | Rich            |
| Richton       | Rich            |
| Rich          | Rich            |
| Richson       | Rich            |
| Richardson    | Rich            |
| Richard       | Rich            |
| Ramsay        | Richton         |
| Richton       | Richton         |
| Rich          | Richton         |
| Richson       | Richton         |
| Richardson    | Richton         |
| Richard       | Richton         |
| Ramsay        | Ramsay          |
| Richton       | Ramsay          |
| Rich          | Ramsay          |
| Richson       | Ramsay          |
| Richardson    | Ramsay          |
| Richard       | Ramsay          |
+---------------+-----------------+
36 rows in set (0.00 sec)
```

Display details of manager of a project whose location is "Stafford"

```
mysql> SELECT PNUMBER,DNUM,FNAME,ADDRESS,BDATE FROM((PROJECT JOIN
DEPARTMENT
 ON DNUMBER=DNUM) JOIN EMPLOYEE ON MGR_SSN=SSN) WHERE
PLOCATION="Stafford";
+---------+------+-------+----------------------+------------+
| PNUMBER | DNUM | FNAME | ADDRESS              | BDATE      |
+---------+------+-------+----------------------+------------+
|       4 |    4 | Rich  | 101 Oak Forest,Katy,TX | 1962-12-30 |
+---------+------+-------+----------------------+------------+
1 row in set (0.00 sec)
```

Aggregate functions.

Find the sum of the salaries of all employees, the maximum salary,
the minimum salary, and the average salary

```
mysql> SELECT SUM(Salary) AS SUM,MAX(Salary) AS MAX,MIN(Salary) AS MIN,AVG(Salary)
AS AVG FROM EMPLOYEE;
+-----------+----------+----------+--------------+
| SUM       | MAX      | MIN      | AVG          |
+-----------+----------+----------+--------------+
| 228000.00 | 39000.00 | 37000.00 | 38000.000000 |
+-----------+----------+----------+--------------+
1 row in set (0.00 sec)
```

0. Find the sum of the salaries of all employees of the ‘IT’ department, as well as the maximum salary, the minimum salary, and the average
salary in this department.

```
mysql> SELECT SUM(SALARY),MAX(SALARY),MIN(SALARY),AVG(SALARY) FROM
(EMPLOYEE
 JOIN DEPARTMENT ON DNUMBER=DNO) WHERE DNAME="IT";
+-------------+-------------+-------------+--------------+
| SUM(SALARY) | MAX(SALARY) | MIN(SALARY) | AVG(SALARY)  |
+-------------+-------------+-------------+--------------+
|   228000.00 |    39000.00 |    37000.00 | 38000.000000 |
+-------------+-------------+-------------+--------------+
1 row in set (0.01 sec)
```

Retrieve the total number of employees in the company
(Q21) and the number of employees in the ‘IT’ department

```
mysql> SELECT COUNT(*) FROM EMPLOYEE;
+----------+
| COUNT(*) |
+----------+
|        6 |
+----------+
1 row in set (0.04 sec)

mysql> SELECT COUNT(*) FROM EMPLOYEE,DEPARTMENT WHERE DNO=DNUMBER AND
DNAME="IT";
+----------+
| COUNT(*) |
+----------+
|        6 |
+----------+
1 row in set (0.00 sec)
```

Count the number of distinct salary values in the database

```
mysql> SELECT COUNT(DISTINCT SALARY) FROM EMPLOYEE;
+------------------------+
| COUNT(DISTINCT SALARY) |
+------------------------+
|                      2 |
+------------------------+
1 row in set (0.01 sec)
```

retrieve the names of all employees who have one or more
dependents

```
mysql> SELECT FNAME,LNAME FROM EMPLOYEE WHERE(SELECT COUNT(*) FROM
DEPENDENT WHERE SSN=ESSN)>=1;
+------------+--------+
| FNAME      | LNAME  |
```

```
+------------+--------+
| Richard    | Marini |
| Richardson | Mar    |
| Richson    | Mario  |
| Rich       | Mario  |
| Richton    | Mario  |
+------------+--------+
5 rows in set (0.00 sec)
```

GROUPING

For each department, retrieve the department number, the number of employees in the department, and their average salary

```
mysql> SELECT DNO,COUNT(*),AVG(SALARY) FROM EMPLOYEE GROUP BY DNO;
+-----+----------+--------------+
| DNO | COUNT(*) | AVG(SALARY)  |
+-----+----------+--------------+
|   4 |        6 | 38000.000000 |
+-----+----------+--------------+
1 row in set (0.01 sec)
```

For each project, retrieve the project number, the project name, and the number of employees who work on that project

```
mysql> SELECT PNUMBER,PNAME,COUNT(*)  FROM (PROJECT JOIN WORKS_ON ON
PNO=PNU
MBER) GROUP BY PNUMBER;
+---------+----------+----------+
| PNUMBER | PNAME    | COUNT(*) |
+---------+----------+----------+
|       1 | ProjectX |        1 |
|       2 | ProjectY |        1 |
|       3 | ProjectZ |        1 |
|       4 | ProjectA |        1 |
|       5 | ProjectB |        1 |
+---------+----------+----------+
5 rows in set (0.00 sec)
```

For each project, retrieve the project number, the project name, and the number of employees from department 4 who work on the project.

```
mysql> SELECT PNUMBER,PNAME,COUNT(*)  FROM ((PROJECT JOIN WORKS_ON ON
PNO=PNUMBER) JOIN EMPLOYEE ON DNO=4 AND ESSN=SSN) GROUP BY PNUMBER;
+---------+----------+----------+
| PNUMBER | PNAME    | COUNT(*) |
+---------+----------+----------+
|       1 | ProjectX |        1 |
|       2 | ProjectY |        1 |
```

```
|      3 | ProjectZ |       1 |
|      4 | ProjectA |       1 |
|      5 | ProjectB |       1 |
+---------+----------+----------+
5 rows in set (0.00 sec)
```

suppose
that we want to count the total number of employees whose salaries exceed $38,000 in each department but only for departments where more than five employees work.

```
mysql> SELECT DNO,COUNT(*) FROM EMPLOYEE WHERE SALARY>38000 AND DNO IN (SELE
CT DNO FROM EMPLOYEE GROUP BY DNO HAVING COUNT(*)>5) GROUP BY DNO;
+-----+----------+
| DNO | COUNT(*) |
+-----+----------+
|   4 |        3 |
+-----+----------+
1 row in set (0.00 sec)
```

WITH AND CASE:

```
mysql> WITH BIGDEPTS(DNO) AS (SELECT DNO FROM EMPLOYEE GROUP BY DNO
HAVING COUNT(*)>5)
    -> SELECT DNO,COUNT(*)
    -> FROM EMPLOYEE
    -> WHERE SALARY>38000 AND DNO IN BIGDEPTS
    -> GROUP BY DNO;
```

we want to give employees different raise
amounts depending on which department they work for; for example, employees in department 5 get a $2,000 raise, those in department 4 get $1,500 and those in department 1 get $3,000

```
mysql> UPDATE EMPLOYEE SET SALARY=
    -> CASE WHEN DNO=5 THEN SALARY+2000
    -> WHEN DNO=4 THEN SALARY+1500
    -> WHEN DNO=1 THEN SALARY+3000
    -> ELSE SALARY+0;
```

RECURSION:

An example of a recursive operation is to retrieve all supervisees of a supervisory employee e at all levels

```
mysql> SELECT * FROM EMPLOYEE;
+------------+-------+--------+-----------+------------+----------------------+------+----------+-----------+-----+
| Fname      | Minit | Lname  | Ssn       | Bdate      | Address              | Sex  | Salary   | Super_ssn | Dno |
```

```
+------------+-------+--------+-----------+------------+----------------------+------+----------+-----------+-----+
| Richard    | K     | Marini | 653298653 | 1962-12-30 | 98 Oak Forest,Katy,TX  | M    | 37000.00 |
65329861  |   4 |
| Richardson | C     | Mar    | 653298654 | 1962-12-30 | 99 Oak Forest,Katy,TX  | M    | 37000.00 |
653298655 |   4 |
| Richson    | K     | Mario  | 653298655 | 1962-12-30 | 100 Oak Forest,Katy,TX | M    | 39000.00 |
653298653 |   4 |
| Rich       | G     | Mario  | 653298656 | 1962-12-30 | 101 Oak Forest,Katy,TX | M    | 39000.00 |
653298653 |   4 |
| Richton    | G     | Mario  | 653298657 | 1962-12-30 | 102 Oak Forest,Katy,TX | F    | 39000.00 |
653298653 |   4 |
| Ramsay     | K     | Marini | 65329869  | 1962-12-30 | 98 Oak Forest,Katy,TX  | M    | 37000.00 |
653298653 |   4 |
+------------+-------+--------+-----------+------------+----------------------+------+----------+-----------+-----+
6 rows in set (0.00 sec)

mysql> WITH RECURSIVE SUP_EMP(SUPERSSN,EMPSSN) AS (SELECT SUPER_SSN,SSN
FROM EMPLOYEE UNION SELECT E.SUPER_SSN,S.SUPERSSN FROM EMPLOYEE
E,SUP_EMP S WHERE E.SSN=S.SUPERSSN)
   -> SELECT * FROM SUP_EMP;
+-----------+-----------+
| SUPERSSN  | EMPSSN    |
+-----------+-----------+
| 65329861  | 653298653 |
| 653298655 | 653298654 |
| 653298653 | 653298655 |
| 653298653 | 653298656 |
| 653298653 | 653298657 |
| 653298653 | 65329869  |
+-----------+-----------+
6 rows in set (0.00 sec)
```

ASSERTIONS:

the salary of an employee must not be greater than
the salary of the manager of the department that the employee works for in SQL, we
can write the following assertion:

```
mysql> CREATE ASSERTION SALARY_CONSTRAINT CHECK(NOT EXISTS(SELECT * FROM
EMPLOYEE E,EMPLOYEE M,DEPARTMENT D WHERE E.SALARY>M.SALARY AND
E.DNO=D.DNUMBER AND D.MGR_SSN=M.SSN));
```

ALTER AND DROP COMMANDS:

DROP Command

```
mysql> USE DUMMY;
Database changed
mysql> CREATE TABLE TEST(NAME CHAR(5));
Query OK, 0 rows affected (0.04 sec)

mysql> INSERT INTO TEST VALUES("PARAM");
Query OK, 1 row affected (0.01 sec)
```

```
mysql> DROP TABLE TEST CASCADE;
Query OK, 0 rows affected (0.02 sec)
mysql> CREATE TABLE TEST(NAME CHAR(5));
Query OK, 0 rows affected (0.04 sec)

mysql> INSERT INTO TEST VALUES("PARAM");
Query OK, 1 row affected (0.01 sec)

mysql> DROP TABLE TEST RESTRICT;
Query OK, 0 rows affected (0.02 sec)

mysql> ALTER TABLE EMPLOYEE ADD COLUMN Job VARCHAR(12);
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE EMPLOYEE ADD COLUMN Experience DECIMAL(2,0) DEFAULT 3;
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE EMPLOYEE DROP COLUMN JOB CASCADE;
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE EMPLOYEE ALTER COLUMN EXPERIENCE DROP DEFAULT;
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE EMPLOYEE ALTER COLUMN EXPERIENCE SET DEFAULT 4;
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

2. Execute the following Queries over the Company Schema you have already created.
a. For each department whose average employee salary is more than 30,000, retrieve the department name and the number of employees working for that department.

```
mysql> INSERT INTO EMPLOYEE VALUES('Teju','G','Chouhan','653298660','1962-12-30','111 Oak
Forest,Katy,TX','M',39000,'653298653',3);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO EMPLOYEE VALUES('Tejas','G','Khana','653298661','1962-12-
30','112 Oak Forest,Katy,TX','M',41000,'653298660',2);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO EMPLOYEE VALUES('Kiran','P','Yadav','653298662','1962-12-
30','114 Oak Forest,Katy,TX','M',30000,'653298661',1);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO EMPLOYEE VALUES('Mukesh','H','Ragav','653298663','1962-12
```

-30','115 Oak Forest,Katy,TX','M',70000,'653298653',5);
Query OK, 1 row affected (0.01 sec)

mysql> SELECT DNAME,COUNT(*) FROM (DEPARTMENT JOIN EMPLOYEE ON DNUMBER=DNO)
GROUP BY DNAME HAVING AVG(SALARY)>30000;
+----------------+----------+
| DNAME | COUNT(*) |
+----------------+----------+
IT	6
Headquarters	1
Administration	1
Clinic	1
+----------------+----------+
4 rows in set (0.00 sec)

b. i. Retrieve the number of female employees in each department making more than 30,000.
mysql> UPDATE EMPLOYEE SET SEX="F" WHERE SSN LIKE "%3";
Query OK, 2 rows affected (0.01 sec)
Rows matched: 2 Changed: 2 Warnings: 0
SELECT DNO,COUNT(*) FROM EMPLOYEE WHERE SEX="F" and salary>30000 GROU
P BY DNO;
+-----+----------+
| DNO | COUNT(*) |
+-----+----------+
| 4 | 2 |
| 5 | 1 |
+-----+----------+
2 rows in set (0.00 sec)

ii. For each department whose average employee salary is more than 30,000, retrieve the department name and number of male employees working for that department.
mysql> SELECT DNAME,COUNT(*) FROM EMPLOYEE S,DEPARTMENT WHERE S.DNO=DNUMBER AND S.SEX="M" AND EXISTS(SELECT COUNT(*) FROM EMPLOYEE E WHERE S.DNO=E.DNO GROUP
BY E.DNO HAVING AVG(E.SALARY)>30000) GROUP BY DNAME;
+----------------+----------+
| DNAME | COUNT(*) |
+----------------+----------+
IT	4
Headquarters	1
Administration	1
+----------------+----------+
3 rows in set (0.01 sec)

c. Retrieve the names of all employees who work in the department that has the employee with the highest salary among all employees.
mysql> SELECT FNAME,LNAME FROM EMPLOYEE WHERE DNO IN (SELECT DNO FROM EMPLOYEE WHERE SALARY=(SELECT MAX(SALARY) FROM EMPLOYEE));
+--------+-------+

```
| FNAME  | LNAME |
+--------+-------+
| Mukesh | Ragav |
+--------+-------+
1 row in set (0.01 sec)
```

d. Retrieve the names of employees who make at least 10,000 more than the employee who is paid the least in the company.

```
mysql> SELECT FNAME,LNAME FROM EMPLOYEE WHERE SALARY>(SELECT
MIN(SALARY) FROM EMPLOYEE)+10000 ;
+--------+-------+
| FNAME  | LNAME |
+--------+-------+
| Tejas  | Khana |
| Mukesh | Ragav |
+--------+-------+
2 rows in set (0.00 sec)
```

e. Retrieve the names of all employees in department 5 who work more than 10 hours per week on the Product X's project.

```
mysql> INSERT INTO WORKS_ON VALUES("653298663",1,11);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO EMPLOYEE VALUES('Andrea','G','Khan','653298665','1962-12-30'
,'192 Oak Forest,Katy,TX','F',60000,'653298660',5);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO WORKS_ON VALUES("653298665",1,5);
Query OK, 1 row affected (0.01 sec)

mysql> SELECT FNAME,LNAME FROM EMPLOYEE,WORKS_ON,PROJECT WHERE
PNAME="ProjectX" AND DNO=5 AND PNUMBER=PNO AND ESSN=SSN AND HOURS>10;
+--------+-------+
| FNAME  | LNAME |
+--------+-------+
| Mukesh | Ragav |
+--------+-------+
1 row in set (0.00 sec)
```

f. List the names of all employees who have a dependent with the same first name as themselves.

```
mysql> SELECT FNAME,LNAME FROM EMPLOYEE,DEPENDENT WHERE
DEPENDENT_NAME=FNAME;
+-------+-------+
| FNAME | LNAME |
+-------+-------+
| Rich  | Mario |
+-------+-------+
1 row in set (0.01 sec)
```

g. Find the names of all employees who are directly supervised by ‘Tejaswi Kumar’.

```
mysql> UPDATE EMPLOYEE SET FNAME="Tejaswi Kumar" WHERE SSN="653298653";
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT FNAME,LNAME FROM EMPLOYEE S WHERE S.SUPER_SSN = (SELECT SSN
FROM
EMPLOYEE E WHERE FNAME="Tejaswi Kumar");
+---------+---------+
| FNAME   | LNAME   |
+---------+---------+
| Richson | Mario   |
| Rich    | Mario   |
| Richton | Mario   |
| Teju    | Chouhan |
| Mukesh  | Ragav   |
| Ramsay  | Marini  |
+---------+---------+
6 rows in set (0.00 sec)
```

h. Find the names of employees who work on all the projects controlled by department number 5.

```
mysql> SELECT FNAME,LNAME FROM EMPLOYEE WHERE NOT EXISTS((SELECT
PNUMBER FROM PROJECT WHERE DNUM=5) EXCEPT(SELECT PNO FROM WORKS_ON
WHERE SSN=ESSN));
+---------+-------+
| FNAME   | LNAME |
+---------+-------+
| Richton | Mario |
+---------+-------+
1 row in set (0.01 sec)

mysql> INSERT INTO PROJECT VALUES("ProjectG",6,"India",5);
Query OK, 1 row affected (0.01 sec)

mysql> SELECT FNAME,LNAME FROM EMPLOYEE WHERE NOT EXISTS((SELECT
PNUMBER FROM PROJECT WHERE DNUM=5) EXCEPT(SELECT PNO FROM WORKS_ON
WHERE SSN=ESSN));
Empty set (0.00 sec)
```

i. For each project, list the project name and the total hours per week (by all employees) spent on that project.

```
mysql> SELECT PNAME,SUM(HOURS) FROM PROJECT,WORKS_ON WHERE
PNO=PNUMBER GROUP BY
 PNAME;
+----------+------------+
| PNAME    | SUM(HOURS) |
+----------+------------+
```

ProjectA	4.0
ProjectB	5.0
ProjectX	17.0
ProjectY	2.0
ProjectZ	3.0
+----------+------------+
5 rows in set (0.00 sec)

j. Retrieve the names of all employees who work on every project.

```
mysql> SELECT FNAME,LNAME FROM EMPLOYEE WHERE NOT EXISTS((SELECT
PNUMBER FROM PROJECT) EXCEPT(SELECT PNO FROM WORKS_ON WHERE
SSN=ESSN));
Empty set (0.00 sec)

mysql> INSERT INTO WORKS_ON VALUES("653298653",2,2);
Query OK, 1 row affected (0.02 sec)

mysql> INSERT INTO WORKS_ON VALUES("653298653",3,3);
Query OK, 1 row affected (0.02 sec)

mysql> INSERT INTO WORKS_ON VALUES("653298653",4,4);
Query OK, 1 row affected (0.02 sec)

mysql> INSERT INTO WORKS_ON VALUES("653298653",5,5);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO WORKS_ON VALUES("653298653",6,6);
Query OK, 1 row affected (0.01 sec)

mysql> SELECT FNAME,LNAME FROM EMPLOYEE WHERE NOT EXISTS((SELECT
PNUMBER FROM PROJECT) EXCEPT(SELECT PNO FROM WORKS_ON WHERE
SSN=ESSN));
+---------------+--------+
| FNAME         | LNAME  |
+---------------+--------+
| Tejaswi Kumar | Marini |
+---------------+--------+
1 row in set (0.00 sec)
```

k. Retrieve the names of all employees who do not work on any project.

```
mysql> SELECT FNAME,LNAME FROM EMPLOYEE WHERE NOT EXISTS(SELECT PNO
FROM WORKS_
ON WHERE SSN=ESSN);
+--------+---------+
| FNAME  | LNAME   |
+--------+---------+
| Teju   | Chouhan |
| Tejas  | Khana   |
| Kiran  | Yadav   |
```

```
| Ramsay | Marini  |
+--------+---------+
4 rows in set (0.00 sec)
```

l. Retrieve the average salary of all female employees.

```
mysql> SELECT AVG(SALARY) FROM EMPLOYEE WHERE SEX="F";
+--------------+
| AVG(SALARY)  |
+--------------+
| 51500.000000 |
+--------------+
1 row in set (0.00 sec)
```

m. Find the names and addresses of all employees who work on at least one project located in Madurai but whose department has no location in Madurai.

```
mysql> INSERT INTO PROJECT VALUES("ProjectP",7,"Madurai",1);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO PROJECT VALUES("ProjectQ",8,"Madurai",2);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO WORKS_ON VALUES("653298654",6,4);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO WORKS_ON VALUES("653298655",7,4);
Query OK, 1 row affected (0.01 sec)

mysql> SELECT FNAME,LNAME FROM EMPLOYEE WHERE EXISTS(SELECT * FROM
WORKS_ON WHERE ESSN=SSN AND PNO IN(SELECT PNUMBER FROM PROJECT WHERE
PLOCATION="Madurai" AND DNUM IN ( SELECT DNUMBER FROM DEPT_LOCATIONS
WHERE DLOCATION!="Madurai")));
+---------+-------+
| FNAME   | LNAME |
+---------+-------+
| Richson | Mario |
+---------+-------+
1 row in set (0.00 sec)
```

n. List the last names of all department managers who have no dependents.

```
mysql> SELECT LNAME FROM EMPLOYEE,DEPARTMENT WHERE SSN=MGR_SSN AND
NOT EXIST
S(SELECT * FROM DEPENDENT WHERE SSN=ESSN);
Empty set (0.02 sec)

mysql> INSERT INTO DEPARTMENT VALUES("Accounts",6,"65329869","1988-06-22");
Query OK, 1 row affected (0.01 sec)

mysql> SELECT FNAME,LNAME FROM EMPLOYEE,DEPARTMENT WHERE SSN=MGR_SSN
AND NOT
```

```
 EXISTS(SELECT * FROM DEPENDENT WHERE SSN=ESSN);
+--------+--------+
| FNAME  | LNAME  |
+--------+--------+
| Ramsay | Marini |
+--------+--------+
1 row in set (0.00 sec)
```

o. Display employee names (e") who are supervised by an e' who is immediately supervised by an employee with lname "XYZ".

```
mysql> UPDATE EMPLOYEE SET FNAME="Alam XYZ" WHERE SSN="653298653";
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT E.FNAME,S.FNAME FROM EMPLOYEE E,EMPLOYEE S WHERE
E.SUPER_SSN=S.SSN AND EXISTS(SELECT N.SSN FROM EMPLOYEE N WHERE FNAME
LIKE "%XYZ%" AND  N.SSN=S.SUPER_SSN);
+------------+---------+
| FNAME      | FNAME   |
+------------+---------+
| Richardson | Richson |
| Tejas      | Teju    |
| Andrea     | Teju    |
+------------+---------+
3 rows in set (0.02 sec)
```

p. Display names of all employees who work on some project controlled by department number 10.

```
mysql> INSERT INTO PROJECT VALUES("Project10",9,"Hydrabad",10);
Query OK, 1 row affected (0.01 sec)

mysql>  INSERT INTO WORKS_ON VALUES("653298655",9,4);
Query OK, 1 row affected (0.01 sec)

mysql> SELECT FNAME,LNAME FROM EMPLOYEE,WORKS_ON WHERE SSN=ESSN AND
PNO IN (SELECT PNUMBER FROM PROJECT WHERE DNUM=10);
+---------+-------+
| FNAME   | LNAME |
+---------+-------+
| Richson | Mario |
+---------+-------+
1 row in set (0.00 sec)
```

q. Print all the ssn and the first name of supervisors who supervise at least 2 projects in ascending order of the number of employee he/she supervise under him/her.

```
mysql> SELECT DISTINCT S.SSN,S.FNAME FROM EMPLOYEE E,EMPLOYEE S WHERE
E.SUPER_SSN=S.SSN AND (SELECT COUNT(*) FROM (SELECT A.SUPER_SSN FROM
EMPLOYEE A,WORKS_ON WHERE A.SUPER_SSN=S.SSN AND A.SSN=ESSN) AS P)>=2
ORDER BY (SELECT COUNT(*) FROM EMPLOYEE B WHERE B.SUPER_SSN=S.SSN);
```

```
+-----------+----------+
| SSN       | FNAME    |
+-----------+----------+
| 653298663 | Mukesh   |
| 653298655 | Richson  |
| 653298653 | Alam XYZ |
+-----------+----------+
3 rows in set (0.00 sec)
```

r. Display all male employee names who also have dependents along with their dependent names.

```
mysql> SELECT FNAME,LNAME,DEPENDENT_NAME FROM EMPLOYEE,DEPENDENT
WHERE SSN=ESSN AND EMPLOYEE.SEX="M";
+------------+-------+---------------+
| FNAME      | LNAME | DEPENDENT_NAME |
+------------+-------+---------------+
| Richardson | Mar   | Joy           |
| Richson    | Mario | Rosy          |
| Rich       | Mario | Rich          |
+------------+-------+---------------+
3 rows in set (0.00 sec)
```

s. Display those employees whose salary exceeds the department managers salary that the employee(s) work for.

```
mysql> SELECT E.FNAME,E.LNAME FROM EMPLOYEE E WHERE EXISTS(SELECT
MGR_SSN FROM DEPARTMENT WHERE DNUMBER=E.DNO AND EXISTS(SELECT
F.FNAME FROM EMPLOYEE F WHERE F.SALARY<E.SALARY AND F.SSN=MGR_SSN));
+--------+-------+
| FNAME  | LNAME |
+--------+-------+
| Tejas  | Khana |
| Mukesh | Ragav |
| Andrea | Khan  |
+--------+-------+
3 rows in set (0.00 sec)
```

t. Display employee names who either work in CS department or supervise an employee working for CS department

```
mysql>  INSERT INTO DEPARTMENT VALUES("CS",9,"653298662","1988-06-22");
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO EMPLOYEE VALUES('Rocky','H','Stone','653298698','1962-12-
30','201 Oak Forest,Katy,TX','M',51000,'653298654',9);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO EMPLOYEE VALUES('Hiran','L','Farook','653298100','1962-12
-30','21 Oak Forest,Katy,TX','M',90000,'65329869',9);
Query OK, 1 row affected (0.01 sec)
```

```
mysql> SELECT FNAME,LNAME FROM EMPLOYEE WHERE DNO=9 UNION SELECT
S.FNAME,S.L
NAME FROM EMPLOYEE E,EMPLOYEE S WHERE E.SUPER_SSN=S.SSN AND E.DNO=9;
+------------+--------+
| FNAME      | LNAME  |
+------------+--------+
| Hiran      | Farook |
| Rocky      | Stone  |
| Ramsay     | Marini |
| Richardson | Mar    |
+------------+--------+
4 rows in set (0.00 sec)
```