Date : 08/08/24

DBMS LAB - SESSION 4

Bunni Ranadeesh

106122026

Question 1: With continuation to Session 03 exercise, execute all the example queries provided in Subsection 7.1.1 to 7.4.2

- Retrieve the names of all employees who do not have supervisors.

```
mysql> select Fname from employee where Super_ssn is NULL;
+--------+
| Fname  |
+--------+
| James  |
+--------+
1 row in set (0.00 sec)
```

- Select the project numbers of projects that have an employee with last name 'Smith' involved as manager, whereas the second nested query selects the project numbers of projects that have an employee with last name 'Smith' involved as worker. In the outer query, we use the OR logical connective to retrieve a PROJECT tuple if the PNUMBER value of that tuple is in the result of either nested query

```
mysql> select distinct Pnumber from project where Pnumber in (select Pnumber from project,
 department, employee where Dnum=Dnumber and Mgr_ssn=Ssn and Lname='Smith') or Pnumber in
(select Pno from works_on, employee where Essn=Ssn and Lname='Smith');
+----------+
| Pnumber  |
+----------+
|        1 |
|        2 |
+----------+
2 rows in set (0.01 sec)
```

- Select the Essns of all employees who work the same (project, hours) combination on some project that employee 'John Smith' (whose Ssn = '123456789') works on

```
mysql> select distinct Essn from works_on where (Pno, Hours) in (select Pno, Hours from works_on where Essn='123456789');
+-----------+
| Essn      |
+-----------+
| 123456789 |
+-----------+
1 row in set (0.00 sec)
```

- Return the names of employees whose salary is greater than the salary of all the employees in department 5

```
mysql> select Fname from employee where Salary > All (select Salary from employee where Dno=5);
+----------+
| Fname    |
+----------+
| Reela    |
| James    |
| Jennifer |
+----------+
```

- Retrieve the name of each employee who has a dependent with the same sex as the employee.

```
mysql> select E.Fname from employee as E where E.Ssn in (select D.Essn from dependent as D where E.Sex=D.Sex);
+----------+
| Fname    |
+----------+
| John     |
| Franklin |
+----------+
2 rows in set (0.00 sec)
```

- Retrieve the names of employees who have no dependents.

```
mysql> select Fname from employee where not exists( select *from dependent where Essn=ssn);
+--------+
| Fname  |
+--------+
| Joyce  |
| Reela  |
| Ramesh |
| James  |
| Ahmad  |
| Alica  |
+--------+
6 rows in set (0.00 sec)
```

- 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      |
+-----------+
| 123456789 |
| 453453453 |
| 333445555 |
| 666884444 |
+-----------+
4 rows in set (0.00 sec)
```

- Retrieve the name and address of every employee who works for the ‘Research’ department.

```
mysql> select concat(Fname, ' ', Lname) as Name , Address from employee join department on Dno=Dnumber where Dname='Research';
+---------------+-------------------------+
| Name          | Address                 |
+---------------+-------------------------+
| John Smith    | 731 Fondren, Houston, TX |
| Franklin Wong | 638 Voss, Houston, TX   |
| Joyce English | 5631 Rice, Houston, TX  |
| Ramesh Narayan | 975 Fire Oak, Humble, TX |
+---------------+-------------------------+
4 rows in set (0.00 sec)
```

- Retrieve all employees with Supervisors and also employees with no supervisors (LEFT OUTER JOIN).

```
mysql> select concat(E.Fname, ' ', E.Lname) as Name , concat(S.Fname, ' ', S.Lname) as Supervisor from employee as E left outer join employee as S on E.Ssn=S.Super_ssn;
+-----------------+-----------------+
| Name            | Supervisor      |
+-----------------+-----------------+
| John Smith      | NULL            |
| Franklin Wong   | Ramesh Narayan  |
| Franklin Wong   | Joyce English   |
| Franklin Wong   | John Smith      |
| Joyce English   | NULL            |
| Reela Dhar      | NULL            |
| Ramesh Narayan  | NULL            |
| James Borg      | Jennifer Wallace |
| James Borg      | Franklin Wong   |
| Jennifer Wallace | Alica Zelaya    |
| Jennifer Wallace | Ahmad Jabbar    |
| Ahmad Jabbar    | NULL            |
| Alica Zelaya    | NULL            |
+-----------------+-----------------+
13 rows in set (0.00 sec)
```

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

```
mysql> select max(Salary) as MaxSalary , min(Salary) as MinSalary, avg(Salary) as AverageSalary, sum(Salary) as TotalSalary from employee;
+-----------+-----------+---------------+-------------+
| MaxSalary | MinSalary | AverageSalary | TotalSalary |
+-----------+-----------+---------------+-------------+
|  70000.00 |  25000.00 |  39000.000000 |   351000.00 |
+-----------+-----------+---------------+-------------+
1 row in set (0.01 sec)
```

- Retrieve the number of employees in the ‘Research’ department.

```
mysql> select count(*) as Researchers_Count from employee join department on Dno=Dnumber and Dname='Research';
+-------------------+
| Researchers_Count |
+-------------------+
|                 4 |
+-------------------+
1 row in set (0.00 sec)
```

- Count the number of distinct salary values in the database

```
mysql> select count(distinct Salary) as Distinct_Salary from employee;
+-----------------+
| Distinct_Salary |
+-----------------+
|               7 |
+-----------------+
1 row in set (0.00 sec)
```

- Retrieve the names of all employees who have two or more dependents.

```
mysql> select concat(Fname, ' ', Lname) as Name from employee where (select count(*) from dependent where Ssn=Essn)>=2;
+---------------+
| Name          |
+---------------+
| John Smith    |
| Franklin Wong |
+---------------+
2 rows in set (0.00 sec)
```

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

```
mysql> select Dno as DeptNo , count(*) as NoofEmployees , avg(Salary) as AverageSalary from employee group  by Dno;
+--------+---------------+---------------+
| DeptNo | NoofEmployees | AverageSalary |
+--------+---------------+---------------+
|      5 |             4 |  33250.000000 |
|      2 |             1 |  70000.000000 |
|      1 |             1 |  55000.000000 |
|      4 |             3 |  31000.000000 |
+--------+---------------+---------------+
4 rows in set (0.00 sec)
```

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

```
mysql> select Pno as ProjectNo , Pname as ProjectName , count(*) as EmployeeCount from project, works_on where Pnumber=Pno group by Pno;
+-----------+-----------------+---------------+
| ProjectNo | ProjectName     | EmployeeCount |
+-----------+-----------------+---------------+
|        10 | Computerization |             2 |
|        30 | Newbenefits     |             1 |
|         1 | ProductX        |             2 |
|         2 | ProductY        |             3 |
|         3 | ProductZ        |             2 |
|        20 | Reorganization  |             1 |
+-----------+-----------------+---------------+
6 rows in set (0.00 sec)
```

- For each project on which more than two employees work, retrieve the project number, the project name, and the number of employees who work on the project.

```
mysql> select Pno as ProjectNo , Pname as ProjectName , count(*) as EmployeeCount from project, works_on where Pnumber=Pno group by Pno having count(*)>2;
+-----------+-------------+---------------+
| ProjectNo | ProjectName | EmployeeCount |
+-----------+-------------+---------------+
|         2 | ProductY    |             3 |
+-----------+-------------+---------------+
1 row in set (0.00 sec)
```

- For each project, retrieve the project number, the project name, and the number of employees from department 5 who work on the project.

```
mysql> select Pno as ProjectNo , Pname as ProjectName , count(*) as EmployeeCount from project, works_on , employee where Pnumber=Pno and Essn=Ssn and Dno=5 group by Pno,Pname;
+-----------+-----------------+---------------+
| ProjectNo | ProjectName     | EmployeeCount |
+-----------+-----------------+---------------+
|         1 | ProductX        |             2 |
|         2 | ProductY        |             3 |
|         3 | ProductZ        |             2 |
|        10 | Computerization |             1 |
|        20 | Reorganization  |             1 |
+-----------+-----------------+---------------+
5 rows in set (0.00 sec)
```

- Count the total number of employees whose salaries exceed $28,000 in each department, but only for departments where more than two employees work.

```
mysql> select Dno,count(*) from employee where Salary>28000 group by Dno having count(*)>2;
+-----+----------+
| Dno | count(*) |
+-----+----------+
|   5 |        3 |
+-----+----------+
1 row in set (0.00 sec)
```

Question 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> select D.Dname as Department, count(E.Ssn) as EmployeeCount from employee as E join
 department as D on E.Dno=D.Dnumber group by Dno having avg(E.Salary)>30000;
+----------------+---------------+
| Department     | EmployeeCount |
+----------------+---------------+
| Research       |             4 |
| Headquarters   |             1 |
| Administration |             3 |
+----------------+---------------+
3 rows in set (0.00 sec)
```

b. i) Retrieve the number of female employees in each department making more than 30,000

```
mysql> select D.Dname as Department, count(E.Ssn) as FemaleCount from employee as E join d
epartment as D on E.Dno=D.Dnumber where E.Sex='F' and E.Salary>3000 group by E.Dno;
+----------------+-------------+
| Department     | FemaleCount |
+----------------+-------------+
| Research       |           1 |
| Administration |           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 D.Dname as Department , count(E.Ssn) as MaleCount from employee as E join de
partment as D on E.Dno=D.Dnumber where E.Sex='M' group by D.Dnumber having avg(Salary)>300
00;
+--------------+-----------+
| Department   | MaleCount |
+--------------+-----------+
| Research     |         3 |
| Headquarters |         1 |
+--------------+-----------+
2 rows in set (0.00 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 concat(E.Fname, ' ', E.Lname) as Name , E.Salary as Salary from employee as
E where E.Salary = (select max(E1.Salary) from employee as E1 where E1.Dno=E.Dno );
+------------------+----------+
| Name             | Salary   |
+------------------+----------+
| Franklin Wong    | 40000.00 |
| Reela Dhar       | 70000.00 |
| James Borg       | 55000.00 |
| Jennifer Wallace | 43000.00 |
+------------------+----------+
4 rows in set (0.00 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 concat(Fname, ' ', Lname) as Name, Salary from employee where Salary>=(select min(Sal
ary) from employee)+10000;
+------------------+----------+
| Name             | Salary   |
+------------------+----------+
| Franklin Wong    | 40000.00 |
| Reela Dhar       | 70000.00 |
| Ramesh Narayan   | 38000.00 |
| James Borg       | 55000.00 |
| Jennifer Wallace | 43000.00 |
+------------------+----------+
5 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> select concat(E.Fname, ' ', E.Lname) as Name , W.Hours, P.Pname from employee as E join work
s_on as W on E.Ssn=W.Essn join project as P on W.Pno = P.Pnumber where W.Hours>10 and E.Dno=5 and P.Pname='ProductX';
+---------------+-------+----------+
| Name          | Hours | Pname    |
+---------------+-------+----------+
| John Smith    |  32.5 | ProductX |
| Joyce English |  20.0 | ProductX |
+---------------+-------+----------+
2 rows 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 distinct Fname from employee join dependent where Dependent_name=Fname;
+--------+
| Fname  |
+--------+
| Alice  |
+--------+
1 row in set (0.00 sec)
```

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

```
mysql> select concat(E.Fname, ' ', E.Lname) as Name, S.Fname as Supervisor from  employee
E join employee as S on E.Super_ssn=S.ssn where S.Fname='Franklin';
+-----------------+------------+
| Name            | Supervisor |
+-----------------+------------+
| John Smith      | Franklin   |
| Joyce English   | Franklin   |
| Ramesh Narayan  | Franklin   |
+-----------------+------------+
3 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 concat(E.Fname, ' ', E.Lname) as Name from employee as E where not exists( select P.Pnumber from project
as P where P.Dnum=5 and not exists(select W.Pno from works_on as W where E.Ssn=W.Essn and P.Pnumber = W.Pno));
+------------+
| Name       |
+------------+
| John Smith |
+------------+
1 row in set (0.00 sec)

mysql>
```

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

```
mysql> select P.Pname as Project, Sum(W.Hours) as Hours from  project as P join works_on a
s W on P.Pnumber = W.Pno group by P.Pnumber;
+-----------------+-------+
| Project         | Hours |
+-----------------+-------+
| ProductX        |  52.5 |
| ProductY        |  37.5 |
| ProductZ        |  60.0 |
| Computerization |  30.0 |
| Reorganization  |  20.0 |
| Newbenefits     |  40.0 |
+-----------------+-------+
6 rows in set (0.00 sec)
```

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

```
mysql> select concat(E.Fname, ' ', E.Lname) as Name from employee as E where not exists( select P.Pnumber from
project as P where not exists(select W.Pno from works_on as W where E
.Ssn=W.Essn and P.Pnumber = W.Pno));
+------------+
| Name       |
+------------+
| John Smith |
+------------+
1 row in set (0.00 sec)
```

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

```
mysql> select concat(E.Fname, ' ', E.Lname) as Name from employee as E left outer join works_on as W on E.Ssn=W.Essn
 where W.Pno is NULL;
+------------------+
| Name             |
+------------------+
| Reela Dhar       |
| James Borg       |
| Hari Kumar       |
| Jennifer Wallace |
| Ahmad Jabbar     |
+------------------+
5 rows in set (0.00 sec)
```

l. Retrieve the average salary of all female employees.

```
mysql> select avg(Salary) as AvgFemalSalary from employee where Sex='F';
+-----------------+
| AvgFemalSalary |
+-----------------+
|   46000.000000 |
+-----------------+
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 Bellaire but whose department has no location in Madurai.

```
mysql> select concat(E.fname, ' ', E.Lname) as Name, E.Address from employee as E join wor
ks_on as W on E.Ssn=W.Essn join project P on W.Pno=P.Pnumber where P.Plocation='Bellaire'
and E.Dno not in (select D.Dnumber from department as D join locations as L on D.Dnumber=L
.Dnumber where L.Dlocation='Madurai');
+---------------+--------------------------+
| Name          | Address                  |
+---------------+--------------------------+
| John Smith    | 731 Fondren, Houston, TX |
| Joyce English | 5631 Rice, Houston, TX   |
+---------------+--------------------------+
2 rows in set (0.00 sec)
```

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

```
mysql> SELECT E.Lname
    -> FROM EMPLOYEE E
    -> JOIN DEPARTMENT D ON E.Ssn = D.Mgr_ssn
    -> LEFT JOIN DEPENDENT DEP ON E.Ssn = DEP.Essn
    -> WHERE DEP.Essn IS NULL;
+-------+
| Lname |
+-------+
| Borg  |
+-------+
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 "Borg".

```
mysql> SELECT E2.Fname, E2.Lname
    -> FROM EMPLOYEE E1
    -> JOIN EMPLOYEE E2 ON E1.Ssn = E2.Super_ssn
    -> JOIN EMPLOYEE E3 ON E1.Super_ssn = E3.Ssn
    -> WHERE E3.Lname = 'Borg';
+--------+---------+
| Fname  | Lname   |
+--------+---------+
| John   | Smith   |
| Joyce  | English |
| Ramesh | Narayan |
| Hari   | Kumar   |
| Ahmad  | Jabbar  |
| Alice  | Zelaya  |
+--------+---------+
6 rows in set (0.00 sec)
```

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

```
mysql> SELECT DISTINCT E.Fname, E.Lname
    -> FROM EMPLOYEE E
    -> JOIN WORKS_ON W ON E.Ssn = W.Essn
    -> JOIN PROJECT P ON W.Pno = P.Pnumber
    -> WHERE P.Dnum = 1;
+----------+-------+
| Fname    | Lname |
+----------+-------+
| John     | Smith |
| Franklin | Wong  |
+----------+-------+
2 rows 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 s.Ssn, s.Fname from employee s join employee e on e.Super_Ssn=s.Ssn join wor
ks_on w on e.Ssn=w.Essn join project p on p.Pnumber=w.Pno group by s.Ssn, s.Fname having c
ount(distinct p.Pnumber)>=2 order by count(distinct e.Ssn);
+-----------+----------+
| Ssn       | Fname    |
+-----------+----------+
| 888665555 | James    |
| 987654321 | Jennifer |
| 333445555 | Franklin |
+-----------+----------+
3 rows in set (0.00 sec)
```

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

```
mysql> select concat(e.Fname, ' ', e.Lname) as Name , d.Dependent_name from employee e join dependent d on e.Ssn=d.Essn where e.Sex='M';
+---------------+----------------+
| Name          | Dependent_name |
+---------------+----------------+
| John Smith    | Alice          |
| John Smith    | Elizabeth      |
| John Smith    | Michael        |
| Franklin Wong | Alice          |
| Franklin Wong | Joy            |
| Franklin Wong | Theodore       |
+---------------+----------------+
6 rows in set (0.00 sec)
```

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

```
mysql> select concat(e.Fname, ' ', e.Lname) as Name , e.Salary from employee e join depart
ment d on e.Dno=d.Dnumber join employee m on d.Mgr_ssn=m.Ssn where e.Salary>=m.Salary;
+-------------------+----------+
| Name              | Salary   |
+-------------------+----------+
| Franklin Wong     | 40000.00 |
| James Borg        | 55000.00 |
| Jennifer Wallace  | 43000.00 |
+-------------------+----------+
3 rows in set (0.00 sec)
```

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

```
mysql> select distinct e.Fname from employee e left join employee s on e.Ssn=s.Super_ssn join department d on e.Dno=d
.Dnumber where d.Dname='Research' or s.Dno=(select d2.Dnumber from department d2 where d2.Dname='Research');
+----------+
| Fname    |
+----------+
| John     |
| Franklin |
| Joyce    |
| Ramesh   |
| James    |
| Hari     |
+----------+
6 rows in set (0.00 sec)
```