

DBMS LAB 4

NAME: Shweta Manjusha Santhosh
ROLL NO: 106122114

Q1.)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’).

```
mysql> SELECT *FROM DEPARTMENT;
+-----+-----+-----+-----+
| Dname   | Dnumber | Mgr_ssn | Mgr_start_date |
+-----+-----+-----+-----+
| Research |      5 | 333445555 | 1988-05-22   |
| Admin    |      4 | 987654321 | 1995-01-01   |
| Head     |      1 | 888665555 | 1981-06-19   |
+-----+-----+-----+-----+
3 rows in set (0.01 sec)

mysql> SELECT *FROM DEPT_LOCATIONS;
+-----+-----+
| Dnumber | Dlocation |
+-----+-----+
|      1 | Houston   |
|      4 | Stafford  |
|      5 | Bellarie  |
|      5 | Sugarland |
+-----+-----+ 4
rows in set (0.02 sec)

mysql> SELECT *FROM EMPLOYEE;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Fname   | Minit | Lname   | Ssn      | Bdate   | Address                | Sex | Salary | Super_ssn |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Franklin | T     | Wong    | 333445555 | 1955-12-08 | 638 Voss, Houston, TX | M   | 4000   | 888665555 |
| Alicia   | J     | Zelaya  | 999887777 | 1968-01-19 | 3321 Castle, spring, TX | F   | 2500   | 987654321 |
| Jennifer | S     | Wallace | 987654321 | 1941-06-20 | 291 Berry, Bellarie, TX | F   | 4300   | 888665555 |
+-----+-----+-----+-----+-----+-----+-----+-----+ 3
rows in set (0.02 sec)

mysql> SELECT *FROM PROJECT;
+-----+-----+-----+-----+
| Pname   | Pnumber | Plocation | Dnum |
+-----+-----+-----+-----+
| ProductX |      1 | Bellarie  |      5 |
| ProductY |      2 | Sugarland |      5 |
| ProductZ |      3 | Houston   |      5 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> SELECT *FROM WORKS_ON;
+-----+-----+-----+
| Essn      | Pno | Hours |
+-----+-----+-----+
| 123456789 |    1 | 32.5  |
| 123456789 |    2 | 7.5    |
| 666884444 |    3 | 40     |
| 453453453 |    2 | 20     |
+-----+-----+-----+
4 rows in set (0.02 sec)
```

The following questions were implemented on the above tables.

Query 18. Retrieve the names of all employees who do not have supervisors.

Sol) *Since the table doesnt have any record without a supervisor, it returned an empty set.*
mysql> SELECT Fname, Lname FROM EMPLOYEE WHERE Super_ssn IS NULL;
Empty set (0.00 sec)

Q4A, the first nested query selects the project numbers of projects that have an employee with last name ‘Wong’ involved as manager, whereas the second nested query selects the project numbers of projects that have an employee with last name ‘Wong’ 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='Wong') OR Pnumber IN(SELECT Pno FROM WORKS_ON, EMPLOYEE WHERE Essn=Ssn AND
Lname='Wong');
+-----+
| Pnumber |
+-----+
|      1 |
|      2 |
|      3 |
+-----+
```

3 rows in set (0.00 sec)
exp) here wong worked as a manager for project number 1, 2, 3 but didnt work as a worker in any of the projects so since we used an OR it returns project numbers 1, 2, 3.

```
mysql> SELECT DISTINCT Pnumber FROM PROJECT WHERE Pnumber IN(SELECT Pnumber FROM PROJECT, DEPARTMENT, EMPLOYEE WHERE
Dnum=Dnumber AND Mgr_ssn =Ssn AND Lname='Zelaya') OR Pnumber IN(SELECT Pno FROM WORKS_ON, EMPLOYEE WHERE Essn=Ssn AND
Lname='Zelaya');
Empty set (0.00 sec)
```

Since Zelaya did not work on any projects(as a worker as well as manager) it returned an empty set.

If a nested query returns a single attribute and a single tuple, the query result will be a single (scalar) value. In such cases, it is permissible to use = instead of IN for the comparison operator. In general, the nested query will return a table (relation), which is a set or multiset of tuples. (a query with this usage). sol)

```
mysql> INSERT INTO WORKS_ON(Essn, Pno, Hours) VALUES('0000000000','1','32.5'); Query
OK, 1 row affected (0.03 sec)
mysql> SELECT DISTINCT Essn FROM WORKS_ON WHERE (Pno, Hours) IN (SELECT Pno, Hours FROM WORKS_ON WHERE Essn='123456789');
+-----+
| Essn      | +-----+
| 123456789 |
|         0 |
+-----+
2 rows in set (0.00 sec)
mysql> INSERT INTO WORKS_ON(Essn, Pno, Hours) VALUES('1000','1','32.5'); Query
OK, 1 row affected (0.03 sec)
mysql> SELECT DISTINCT Essn FROM WORKS_ON WHERE (Pno, Hours) IN (SELECT Pno, Hours FROM WORKS_ON WHERE Essn='123456789');
+-----+
| Essn      | +-----+
| 123456789 |
|         0 |
|      1000 |
+-----+
3 rows in set (0.00 sec)
```

EXISTS and UNIQUE are Boolean functions that return TRUE or FALSE; hence, they can be used in a WHERE clause condition. The EXISTS function in SQL is used to check whether the result of a nested query is empty (contains no tuples) or not. The result of EXISTS is a Boolean value TRUE if the nested query result contains at least one tuple, or FALSE if the nested query result contains no tuples.

```
mysql> SELECT E.Fname, E.Lname FROM EMPLOYEE AS E WHERE EXISTS( SELECT * FROM WORKS_ON AS W WHERE W.Essn =E.Ssn);
Empty set (0.00 sec)
mysql> SELECT E.Fname, E.Lname FROM EMPLOYEE AS E WHERE NOT EXISTS( SELECT * FROM WORKS_ON AS W WHERE W.Essn =E.Ssn);
+-----+-----+
| Fname   | Lname   |
+-----+-----+
| Franklin| Wong    |
| Alicia  | Zelaya  |
```

```
| Jennifer | Wallace | +-
-----+-----+ 3
rows in set (0.00 sec)
```

here there is no employee common in works on and employee so exists becomes false and returns nothing but not exists returns true.

JOINED TABLES AND OUTER JOINS

The concept of a joined table (or joined relation) was incorporated into SQL to permit users to specify a table resulting from a join operation in the FROM clause of a query. This construct may be easier to comprehend than mixing together all the select and join conditions in the WHERE clause. For example, in the below query, which retrieves the name and address of every employee who works for the ‘Research’ department. It may be easier to specify the join of the EMPLOYEE and DEPARTMENT relations in the WHERE clause, and then to select the desired tuples and attributes.

Join:
mysql> SELECT Fname , Lname, Address FROM (EMPLOYEE JOIN DEPARTMENT ON Dno=Dnumber) WHERE Dname='Research';
+-----+-----+-----+
| Fname | Lname | Address |
+-----+-----+-----+
| Franklin | Wong | 638 Voss, Houston, TX |
+-----+-----+-----+
1 row in set (0.00 sec) left

outer join:
mysql> 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);
+-----+-----+
| Employee_name | Supervisor_name |
+-----+-----+ |
Wong | NULL |
| Zelaya | Wallace |
| Wallace | NULL |
+-----+-----+
3 rows in set (0.00 sec)

Right outer join mysql> SELECT E.Lname AS Employee_name, S.Lname AS Supervisor_name FROM (EMPLOYEE AS E RIGHT OUTER JOIN EMPLOYEE AS S ON E.Super_ssn=S.ssn);
+-----+-----+
| Employee_name | Supervisor_name |
+-----+-----+ |
NULL | Wong |
| NULL | Zelaya |
| Zelaya | Wallace |
+-----+-----+
3 rows 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), MAX(Salary), MIN(Salary), AVG(Salary) FROM EMPLOYEE;
+-----+-----+-----+-----+
| SUM(Salary) | MAX(Salary) | MIN(Salary) | AVG(Salary) |
+-----+-----+-----+-----+
| 10800 | 4300 | 2500 | 3600.0000 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

Find the sum of the salaries of all employees of the ‘Research’ depart- ment, 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 Dno=Dnumber) WHERE Dname='Research';
+-----+-----+-----+-----+
| SUM(Salary) | MAX(Salary) | MIN(Salary) | AVG(Salary) |
+-----+-----+-----+-----+
| 4000 | 4000 | 4000 | 4000.0000 |

+-----+-----+-----+-----+
1 row in set (0.00 sec)

USING COUNT

mysql> SELECT COUNT(*) FROM (EMPLOYEE AS E JOIN DEPARTMENT AS D ON E.Dno=D.Dnumber) WHERE D.Dname='Research';
+-----+
| COUNT(*) |
+-----+
| 1 |
+-----+
1 row in set (0.00 sec)

2. Execute the following Queries over the Company Schema you have already created. mysql>
SELECT *FROM EMPLOYEE;

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Fname | Minit | Lname | Ssn | Bdate | Address | Sex | Salary | Super_ssn | Dno |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000.00	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000.00	888665555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000.00	333445555	5
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000.00	333445555	5
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000.00	NULL	1
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellarie, TX	F	43000.00	888665555	4
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000.00	987654321	4
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000.00	987654321	4
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ 8
rows in set (0.00 sec)

mysql> SELECT *FROM DEPARTMENT;
+-----+-----+-----+-----+
| Dname | Dnumber | Mgr_ssn | Mgr_start_date |
+-----+-----+-----+-----+
Headquarters	1	888665555	1981-06-19
Administration	4	987654321	1995-01-01
Research	5	333445555	1988-05-22
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> SELECT *FROM WORKS_ON;
+-----+-----+-----+
| Essn | Pno | Hours |
+-----+-----+-----+ |
123456789 | 1 | 32.5 |
123456789	2	7.5
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
453453453	1	20.0
453453453	2	20.0
666884444	3	40.0
+-----+-----+-----+ 9
rows in set (0.00 sec)

mysql> SELECT *FROM DEPT_LOCATIONS;
+-----+-----+
| Dnumber | Dlocation |
+-----+-----+
+ | 1 | Houston |
| 4 | Stafford |
| 5 | Bellaire |
| 5 | Houston |
| 5 | Sugarland |
+-----+-----+ 5
rows in set (0.00 sec)

mysql> SELECT *FROM DEPENDENT;
+-----+-----+-----+-----+-----+
| Essn | Dependent_name | Sex | Bdate | Relationship |
+-----+-----+-----+-----+-----+

123456789	Alice	F	1988-12-30	Daughter	
123456789	Elizabeth	F	1967-05-05	Spouse	
123456789	Michael	M	1988-01-04	Son	
333445555	Alice	F	1986-04-05	Daughter	
333445555	Joy	F	1958-05-03	Spouse	
333445555	Theodore	M	1983-10-25	Son	
987654321	Abner	M	1942-02-28	Spouse	
+-----+-----+-----+-----+					

7 rows in set (0.00 sec)

```
mysql> SELECT *FROM PROJECT;
```

+-----+-----+-----+-----+					
Pname	Pnumber	Plocation	Dnum		
+-----+-----+-----+-----+					
ProductX	1	Bellaire	5		
ProductY	2	Sugarland	5		
ProductZ	3	Houston	5		
Computerization	10	Stafford	4		
Reorganization	20	Houston	1		
+-----+-----+-----+-----+					

5 rows in set (0.00 sec)

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.

```
Sol)mysql> SELECT d.Dname, COUNT(e.Ssn) AS Num_Employees FROM DEPARTMENT d JOIN EMPLOYEE e ON d.Dnumber=e.Dno GROUP BY d.Dname HAVING AVG(e.Salary)>30000;
```

+-----+-----+	
Dname	Num_Employees
+-----+-----+	
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.

```
Sol)mysql> SELECT d.Dname, COUNT(e.Ssn) AS Num_Female_Employees
FROM DEPARTMENT d
JOIN EMPLOYEE e ON d.Dnumber = e.Dno
WHERE e.Salary > 30000 AND e.Sex = 'F'
GROUP BY d.Dname;
```

+-----+-----+	
Dname	Num_Female_Employees
+-----+-----+	
Administration	1
+-----+-----+	

1 row 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.

```
SELECT d.Dname, COUNT(e.Ssn) AS Num_Male_Employees
FROM DEPARTMENT d
JOIN EMPLOYEE e ON d.Dnumber = e.Dno
WHERE e.Sex = 'M'
AND d.Dnumber IN (
    SELECT Dno
    FROM EMPLOYEE
    GROUP BY Dno
    HAVING AVG(Salary) > 30000)
GROUP BY d.Dname;
```

+-----+-----+	
Dname	Num_Male_Employees
+-----+-----+	
Research	3
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. SELECT e.Fname, e.Lname FROM EMPLOYEE e

```
WHERE e.Salary = (
    SELECT MAX(e2.Salary)
    FROM EMPLOYEE e2
    WHERE e2.Dno = e.Dno
);
+-----+-----+
| Fname  | Lname  |
+-----+-----+
| Franklin | Wong   | |
James   | Borg   |
| Jennifer | Wallace | +-
-----+-----+ 3
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. SELECT
Fname, Lname
FROM EMPLOYEE
WHERE Salary >= (
    SELECT MIN(Salary) + 10000
    FROM EMPLOYEE
);
+-----+-----+
| Fname  | Lname  |
+-----+-----+
| Franklin | Wong   |
| Ramesh   | Narayan | |
James   | Borg   |
| Jennifer | Wallace | +-
-----+-----+ 4
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. SELECT
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 e.Dno = 5
AND p.Pname = 'ProductX'
AND w.Hours > 10;
+-----+-----+
| Fname | Lname  |
+-----+-----+ |
John   | Smith  |
| Joyce | English |
+-----+-----+ 2
rows in set (0.01 sec)
```

```
f. List the names of all employees who have a dependent with the same first name as themselves.
SELECT e.Fname, e.Lname
FROM EMPLOYEE e
JOIN DEPENDENT d ON e.Ssn = d.Essn
WHERE e.Fname = d.Dependent_name;
Empty set (0.00 sec)
```

```
g. Find the names of all employees who are directly supervised by 'Tejaswi Kumar'.
mysql> SELECT e.Fname, e.Lname
-> LOYEE s ON e.SupFROM EMPLOYEE e
-> JOIN EMPLOYEE s ON e.Super_ssn = s.Ssn
-> WHERE s.Fname = 'Tejaswi' AND s.Lname = 'Kumar'; Empty
set (0.00 sec)
```

```
h. Find the names of employees who work on all the projects controlled by department number 5.
SELECT 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 = 5
GROUP BY e.Ssn, e.Fname, e.Lname
HAVING COUNT(DISTINCT p.Pnumber) = (
```

```
SELECT COUNT(DISTINCT p2.Pnumber)
FROM PROJECT p2
WHERE p2.Dnum = 5
);
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 p.Pname, SUM(w.Hours) AS Total_Hours
-> FROM PROJECT p
-> JOIN WORKS_ON w ON p.Pnumber = w.Pno
-> GROUP BY p.Pname;
```

Pname	Total_Hours
Computerization	10.0
ProductX	52.5
ProductY	37.5
ProductZ	50.0
Reorganization	10.0

rows in set (0.00 sec)

j. Retrieve the names of all employees who work on every project. mysql> SELECT e.Fname, e.Lname

```
-> FROM EMPLOYEE e
-> JOIN WORKS_ON w ON e.Ssn = w.Essn
-> GROUP BY e.Ssn, e.Fname, e.Lname
-> HAVING COUNT(DISTINCT w.Pno) = (
-> SELECT COUNT(*)
-> FROM PROJECT
-> );
Empty set (0.01 sec)
```

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

```
mysql> SELECT e.Fname, e.Lname
-> FROM EMPLOYEE e
-> LEFT JOIN WORKS_ON w ON e.Ssn = w.Essn
-> WHERE w.Essn IS NULL;
```

Fname	Lname
James	Borg
Jennifer	Wallace
Ahmad	Jabbar
Alicia	Zelaya

rows in set (0.00 sec)

l. Retrieve the average salary of all female employees. mysql> SELECT AVG(Salary) AS

```
Average_Salary
-> FROM EMPLOYEE
-> WHERE Sex = 'F';
```

Average_Salary
31000.000000

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. SELECT e.Fname, e.Lname, e.Address

```
FROM EMPLOYEE e
JOIN WORKS_ON w ON e.Ssn = w.Essn
JOIN PROJECT p ON w.Pno = p.Pnumber
WHERE p.Plocation = 'Madurai'
AND e.Dno NOT IN (
```

```
SELECT d.Dnumber
FROM DEPT_LOCATIONS dl
JOIN DEPARTMENT d ON dl.Dnumber = d.Dnumber
WHERE dl.Dlocation = 'Madurai'
);
Empty set (0.01 sec)
```

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

```
JOIN EMPLOYEE eFROM DEPARTMENT d
-> JOIN EMPLOYEE e ON d.Mgr_ssn = e.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 “XYZ”.

```
SELECT e.Fname, e.Lname
FROM EMPLOYEE e
JOIN EMPLOYEE e_prime ON e.Super_ssn = e_prime.Ssn
JOIN EMPLOYEE e_super ON e_prime.Super_ssn = e_super.Ssn
WHERE e_super.Lname = 'XYZ';
Empty set (0.01 sec)
```

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

```
SELECT s.Ssn, s.Fname
FROM EMPLOYEE s
JOIN EMPLOYEE e ON s.Ssn = e.Super_ssn
JOIN WORKS_ON w ON e.Ssn = w.Essn
JOIN PROJECT p ON w.Pno = p.Pnumber
GROUP BY s.Ssn, s.Fname
HAVING COUNT(DISTINCT p.Pnumber) >= 2
ORDER BY COUNT(DISTINCT e.Ssn);
+-----+-----+
| Ssn    | Fname  | +-----+-----+
| 987654321| Jennifer |
| 333445555 | Franklin |
+-----+-----+ 2 rows in set (0.01 sec)
```

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

```
SELECT e.Fname AS Employee_Fname, e.Lname AS Employee_Lname, d.Dependent_name
FROM EMPLOYEE e
JOIN DEPENDENT d ON e.Ssn = d.Essn
WHERE e.Sex = 'M';
+-----+-----+-----+
| Employee_Fname | Employee_Lname | 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 the department managers salary that the employee(s) work for. SELECT e.Fname, e.Lname, e.Salary
FROM EMPLOYEE e


```
JOIN DEPARTMENT d ON e.Dno = d.Dnumber
```

```
JOIN EMPLOYEE m ON d.Mgr_ssn = m.Ssn
```

```
WHERE e.Salary > m.Salary;
```

```
Empty set (0.00 sec)
```

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

```
SELECT DISTINCT e.Fname, e.Lname
```

```
FROM EMPLOYEE e
```

```
LEFT JOIN EMPLOYEE s ON e.Ssn = s.Super_ssn
```

```
JOIN DEPARTMENT d ON e.Dno = d.Dnumber
```

```
WHERE d.Dname = 'CS'
```

```
OR s.Dno = (
```

```
    SELECT d2.Dnumber
```

```
    FROM DEPARTMENT d2
```

```
    WHERE d2.Dname = 'CS'
```

```
);
```

```
Empty set (0.01 sec)
```