

Roll.no:-106122056

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

Relational Database Design – Company Schema – Nested Queries

Q.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’).

```
mysql> use himanshu;
```

```
Database changed
```

```
mysql> show tables;
```

```
+-----+
| Tables_in_himanshu |
+-----+
| DEPENDENT           |
| DEPT_LOCATIONS      |
| Department          |
| WORKS_ON            |
| employee            |
| project             |
+-----+
```

```
mysql> SELECT Fname, Lname FROM employee WHERE Super_ssn IS not NULL;
```

```
+-----+-----+
| Fname | Lname |
+-----+-----+
| John  | Smith |
| Franklin | Wong  |
| Jennifer | Wallace |
| Alicia | Zelaya |
+-----+-----+
```

```
mysql> select Lname, Fname from employee where Salary>ALL (select Salary from employee where Dno=5);
```

```
+-----+-----+
| Lname | Fname |
+-----+-----+
| Wallace | Jennifer |
+-----+-----+
```

```
mysql> select Lname, Fname from employee where Salary>ALL (select Salary from employee where Dno=5);
```

```
+-----+-----+
| Lname | Fname |
+-----+-----+
| Wallace | Jennifer |
+-----+-----+
```

```
mysql> select salary from employee where Fname='Jennifer';
```

```
+-----+
```

```
| salary |
+-----+
| 43000 |
+-----+
```

mysql> describe Department;

```
+-----+-----+-----+-----+-----+
| Field      | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| Dname      | char(50) | YES  |     | NULL    |      |
| Dnumber    | int      | NO   | PRI | NULL    |      |
| Mgr_ssn    | int      | YES  |     | NULL    |      |
| Mgr_start_date | date    | YES  |     | NULL    |      |
+-----+-----+-----+-----+-----+
```

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 |
+-----+-----+-----+-----+
```

mysql> select Fname,Lname from employee where EXISTS (select* from employee where Ssn=Super_ssn);
Empty set (0.00 sec)

mysql> select Fname,Lname from employee where not EXISTS (select* from employee where Ssn=Super_ssn);

```
+-----+-----+
| Fname  | Lname |
+-----+-----+
| John   | Smith |
| Franklin | Wong  |
| Jennifer | Wallace |
| Alicia | Zelaya |
+-----+-----+
```

mysql> select distinct Essn from WORKS_ON where Pno in (1,2,3);
Empty set (0.01 sec)

mysql> select distinct Fname from employee where Dno in (3,4);

```
+-----+
| Fname |
+-----+
| Jennifer |
| Alicia |
+-----+
```

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

```
mysql> select Fname, Lname, Address from (employee join Department on Dno=Dnumber) where Dname='Research';
```

Fname	Lname	Address
John	Smith	731 Fondren
Franklin	Wong	638 Voss Houston TX

```
mysql> select Fname, Lname, Address from (employee join Department) where Dname='Research';
```

Fname	Lname	Address
John	Smith	731 Fondren
Franklin	Wong	638 Voss Houston TX
Jennifer	Wallace	291 Berry Bellaire TX
Alicia	Zelaya	3321 Castle Spring TX

```
mysql> select Fname, Lname, Address from (employee join Department on Dno=Dnumber) where Dname='Research';
```

Fname	Lname	Address
John	Smith	731 Fondren
Franklin	Wong	638 Voss Houston TX

```
mysql> select * from employee;
```

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss Houston TX	M	40000	888665555	5
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry Bellaire TX	F	43000	888665555	4
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle Spring TX	F	25000	987654321	4

```
mysql> select sum(Salary), max(Salary), min(Salary), avg(Salary) from employee;
```

sum(Salary)	max(Salary)	min(Salary)	avg(Salary)
168000	43000	25000	32000

138000	43000	25000	34500.0000
--------	-------	-------	------------

```
mysql> select count(distinct Salary) from employee;
```

count(distinct Salary)
4

```
mysql> select Dno,count(*), avg(Salary) from employee group by Dno;
```

Dno	count(*)	avg(Salary)
5	2	35000.0000
4	2	34000.0000

```
mysql> update employee set Salary = CASE WHEN Dno=4 THEN Salary+1000 WHEN Dno=3 THEN
Salary+2000 ELSE Salary+0;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your
MySQL server version for the right syntax to use near " at line 1
```

```
mysql> with recursive SUP_EMP(SupSsn, EmpSsn) AS (select Super_ssn, Ssn from employee union
select E.Ssn, S.SupSsn from employee as E, SUP_EMP as S where E.Super_ssn=S.EmpSsn) select* from
SUP_EMP;
```

SupSsn	EmpSsn
333445555	123456789
888665555	333445555
888665555	987654321
987654321	999887777
123456789	888665555
999887777	888665555
333445555	999887777
987654321	123456789

```
mysql> alter table employee add column job varchar(12);
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> describe employee;
```

Field	Type	Null	Key	Default	Extra
Fname	char(50)	YES		NULL	
Minit	char(50)	YES		NULL	

Lname	char(50)	YES		NULL		
Ssn	int	NO	PRI	NULL		
Bdate	date	YES		NULL		
Address	varchar(50)	YES		NULL		
Sex	char(10)	YES		NULL		
Salary	int	YES		NULL		
Super_ssn	int	YES		NULL		
Dno	int	YES		NULL		
job	varchar(12)	YES		NULL		

+-----+-----+-----+-----+-----+

Q.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.

```
SELECT D.dept_name, COUNT(E.emp_id) AS num_employees
FROM Department D
JOIN Employee E ON D.dept_id = E.dept_id
GROUP BY D.dept_name
HAVING AVG(E.salary) > 30000;
```

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

```
SELECT D.dept_name, COUNT(E.emp_id) AS num_female_employees
FROM Department D
JOIN Employee E ON D.dept_id = E.dept_id
WHERE E.gender = 'F' AND E.salary > 30000
GROUP BY D.dept_name;
```

- 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.dept_name, COUNT(E.emp_id) AS num_male_employees
FROM Department D
JOIN Employee E ON D.dept_id = E.dept_id
WHERE E.gender = 'M'
GROUP BY D.dept_name
HAVING AVG(E.salary) > 30000;
```

- 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.name
FROM Employee E
WHERE E.dept_id = (SELECT dept_id FROM Employee ORDER BY salary DESC LIMIT 1);
```

- 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 E.name
FROM Employee E
```

WHERE E.salary > (SELECT MIN(salary) + 10000 FROM Employee);

- 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.name
FROM Employee E
JOIN Project P ON E.emp_id = P.emp_id
WHERE E.dept_id = 5 AND P.project_name = 'Product X' AND E.hours_per_week > 10;
```

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

```
SELECT E.name
FROM Employee E
JOIN Dependent D ON E.emp_id = D.emp_id
WHERE E.first_name = D.first_name;
```

- g. Find the names of all employees who are directly supervised by 'Tejaswi Kumar'.

```
SELECT E.name
FROM Employee E
WHERE E.supervisor_id = (SELECT emp_id FROM Employee WHERE name = 'Tejaswi Kumar');
```

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

```
SELECT E.name
FROM Employee E
WHERE NOT EXISTS (
  SELECT P.project_id
  FROM Project P
  WHERE P.dept_id = 5 AND NOT EXISTS (
    SELECT EP.emp_id
    FROM Employee_Project EP
    WHERE EP.emp_id = E.emp_id AND EP.project_id = P.project_id));
```

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

```
SELECT P.project_name, SUM(E.hours_per_week) AS total_hours
FROM Project P
JOIN Employee_Project EP ON P.project_id = EP.project_id
JOIN Employee E ON EP.emp_id = E.emp_id
GROUP BY P.project_name;
```

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

```
SELECT E.name
FROM Employee E
WHERE NOT EXISTS (
  SELECT P.project_id
  FROM Project P
  WHERE NOT EXISTS (
    SELECT EP.emp_id
    FROM Employee_Project EP
```

```
WHERE EP.emp_id = E.emp_id AND EP.project_id = P.project_id));
```

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

```
SELECT E.name  
FROM Employee E  
WHERE NOT EXISTS (  
SELECT EP.emp_id  
FROM Employee_Project EP  
WHERE EP.emp_id = E.emp_id);
```

- l. Retrieve the average salary of all female employees.

```
SELECT AVG(E.salary) AS average_female_salary  
FROM Employee E  
WHERE E.gender = 'F';
```

- 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.name, E.address  
FROM Employee E  
JOIN Dependent D ON E.emp_id = D.emp_id  
WHERE D.location = 'Madurai' AND E.dept_id NOT IN (  
SELECT dept_id FROM Department WHERE location = 'Madurai');
```

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

```
SELECT E.last_name  
FROM Employee E  
WHERE E.role = 'Manager' AND NOT EXISTS (  
SELECT D.emp_id  
FROM Dependent D  
WHERE D.emp_id = E.emp_id  
);
```

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

```
SELECT E1.name  
FROM Employee E1  
WHERE E1.supervisor_id IN (  
SELECT E2.emp_id  
FROM Employee E2  
WHERE E2.supervisor_id = (SELECT emp_id FROM Employee WHERE last_name =  
'XYZ')  
);
```

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

```
SELECT E.name  
FROM Employee E  
JOIN Employee_Project EP ON E.emp_id = EP.emp_id
```

```
JOIN Project P ON EP.project_id = P.project_id WHERE  
P.dept_id = 10;
```

- 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 E.ssn, E.first_name  
FROM Employee E  
WHERE (SELECT COUNT(P.project_id)  
FROM Project P  
WHERE P.supervisor_id = E.emp_id) >= 2  
ORDER BY (SELECT COUNT(E2.emp_id) FROM Employee E2 WHERE  
E2.supervisor_id = E.emp_id) ASC;
```

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

```
SELECT E.name, D.dependent_name  
FROM Employee E  
JOIN Dependent D ON E.emp_id = D.emp_id  
WHERE E.gender = 'M';
```

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

```
SELECT E.name  
FROM Employee E  
WHERE E.salary > (SELECT M.salary FROM Employee M WHERE M.emp_id =  
E.manager_id);
```

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

```
SELECT E.name  
FROM Employee E  
WHERE E.dept_id = (SELECT dept_id FROM Department WHERE dept_name = 'CS')  
OR E.emp_id IN (SELECT supervisor_id FROM Employee WHERE dept_id = (SELECT dept_id  
FROM Department WHERE dept_name = 'CS'));
```