

PERSONNIC
William

Course: Advanced Databases
Teacher: Roozbeh SOLTANI

LAB2: Advanced SQL Queries

Null values

1. **Find the employees whose commission is specified (i.e. including 0.0 commissions).**

```
SELECT * FROM EMP WHERE COMM IS NOT NULL;
```

2. **Find the number of employees whose commission is specified (2 methods).**

```
SELECT COUNT (*) FROM EMP WHERE COMM IS NOT NULL;
```

```
SELECT COUNT (*) FROM EMP WHERE COMM <> NULL;
```

3. **Find the number of employees whose commission is not specified (2 methods).**

```
SELECT COUNT (*) FROM EMP WHERE COMM IS NULL;
```

```
SELECT COUNT (*) FROM EMP WHERE COMM = NULL;
```

4. **Find the lowest, average and highest commission over all the employees (nulls ignored).**

```
SELECT MIN(COMM) FROM EMP WHERE COMM IS NOT NULL;
```

```
SELECT AVG(COMM) FROM EMP WHERE COMM IS NOT NULL;
```

```
SELECT MAX(COMM) FROM EMP WHERE COMM IS NOT NULL;
```

5. **Find the average commission over all the employees (nulls counted as 0.0).**

```
SELECT SUM(COMM)/COUNT (*) FROM EMP;
```

6. **Find the name and commission, expressed in Euro (1 € = \$ 1.2) of all the employees.**

```
SELECT ENAME, COMM/1.2 FROM EMP;
```

7. **Find the name and total salary (including commission) of all the employees.**

```
SELECT ENAME, SAL+COMM FROM EMP;
```

8. **Find the name of the company's top managers (i.e. who don't have a manager).**

```
SELECT ENAME FROM EMP WHERE MGR IS NULL;
```

9. **Find the employees whose commission is less than 25% (nulls excluded).**

```
SELECT * FROM EMP WHERE COMM IS NOT NULL AND COMM < SAL*25/100;
```

10. Find the employees whose commission is less than 25% (nulls counted as 0.0).

```
SELECT * FROM EMP WHERE COMM<SAL*25/100;
```

SQL92 Join Queries

- 1. Display (a) the product of tables EMP and DEPT, (b) the theta-join of EMP and DEPT on DID, and (c) the natural join of EMP and DEPT. Compare the schema and the population of the resulting tables.**

```
SELECT * FROM EMP CROSS JOIN DEPT ;
```

```
SELECT * FROM EMP E JOIN DEPT D ON E.DID < D.DID;
```

```
SELECT * FROM EMP NATURAL JOIN DEPT;
```

All the schema and population are different.

- 2. Find the name and the department of the employees who work in New-York.**

```
SELECT E.ENAME, D.DNAME FROM EMP E NATURAL JOIN DEPT D WHERE  
D.DLOC='New-York' ;
```

- 3. Find the name of the employees who did a mission in the city they work in.**

```
SELECT DISTINCT E.ENAME FROM EMP E NATURAL JOIN DEPT D NATURAL JOIN  
MISSION M WHERE D.DLOC = M.MLOC;
```

- 4. Find the name of the employees along with the name of their manager.**

```
SELECT E.ENAME,M.ENAME FROM EMP E JOIN EMP M WHERE E.MGR = M.EID;
```

- 5. Find the name of the employees who have the same manager as Allen.**

```
SELECT E1.ENAME FROM EMP E1 JOIN EMP E2 WHERE E1.ENAME <> 'Allen' AND  
E2.ENAME = 'Allen' AND E1.MGR = E2.MGR;
```

- 6. Find the name and hire date of the employees who were hired before their manager; also display the manager's hire date.**

```
SELECT E.ENAME,E.HIRED,M.HIRED FROM EMP E JOIN EMP M WHERE E.MGR =  
M.EID AND E.HIRED <M.HIRED;
```

- 7. Find the name of the employees in the Sales department who were hired the same day as an employee in the Research department.**

```
SELECT E1.ENAME FROM (EMP E1 NATURAL JOIN DEPT D1) JOIN (EMP E2  
NATURAL JOIN DEPT D2) WHERE D1.DNAME = 'SALES' AND D2.DNAME =  
'Research' AND E1.HIRED = E2.HIRED;
```

- 8. Find the departments that do not have any employee.**

```
SELECT D.* FROM DEPT D NATURAL LEFT OUTER JOIN EMP E WHERE E.EID IS  
NULL;
```

9. Find the name of the employees with the highest salary.

```
SELECT ENAME FROM EMP HAVING MAX(SAL);
```

10. Find the name of the employees who were hired before all the employees of the Accounting department.

```
SELECT E1.ENAME FROM (EMP E1 NATURAL JOIN DEPT D1) WHERE D1.DNAME  
<> 'Accounting' AND E1.HIRED < (SELECT MIN(E2.HIRED) FROM (EMP E2  
NATURAL JOIN DEPT D2) WHERE D2.DNAME = 'Accounting')
```

3 Subqueries

1. Find the employees with the highest salary (2 methods).

```
SELECT * FROM EMP HAVING MAX(SAL);
```

```
SELECT * FROM EMP WHERE SAL = (SELECT MAX(SAL) FROM EMP);
```

standalone

2. Find the employees who earn less than all managers (2 methods).

```
SELECT * FROM EMP WHERE JOB <> 'manager' AND (SAL) < (SELECT MIN(SAL) FROM  
EMP WHERE JOB = 'manager');
```

```
SELECT * FROM EMP WHERE JOB <> 'manager' AND (SAL) < ALL( SELECT SAL FROM  
EMP WHERE JOB = 'manager');
```

standalone

3. Find the employees who earn more than some analyst (2 methods).

```
SELECT * FROM EMP WHERE JOB <> 'analyst' AND (SAL) > (SELECT MAX(SAL) FROM  
EMP WHERE JOB = 'analyst');
```

```
SELECT * FROM EMP WHERE JOB <> 'analyst' AND (SAL) > ANY( SELECT SAL FROM  
EMP WHERE JOB = 'analyst');
```

standalone

4. Find the employees who work in the Research or Sales departments.

```
(SELECT EMP.* FROM EMP NATURAL JOIN DEPT WHERE DNAME = 'Research') UNION  
(SELECT EMP.* FROM EMP NATURAL JOIN DEPT WHERE DNAME = 'Sales');
```

Standalone

5. Find the departments without any employee (3 methods).

```
SELECT D.* FROM DEPT D WHERE NOT EXISTS (SELECT * FROM EMP WHERE DID  
=D.DID);
```

```
SELECT D.* FROM DEPT D WHERE D.DID NOT IN ( SELECT E.DID FROM EMP E);
```

correlated subquery/standalone

6. Find the departments with at least 3 employees.

```
SELECT D.* FROM DEPT D WHERE (SELECT COUNT(*) FROM EMP E WHERE E.DID = D.DID) >= 3;
```

correlated subquery

7. Find the name of the employees who did a mission.

```
SELECT ENAME FROM EMP WHERE EID IN (SELECT EID FROM MISSION);
```

standalone

8. Find the employees who did a mission in the city they work in.

```
SELECT E.* FROM EMP E NATURAL JOIN DEPT D WHERE EXISTS ( SELECT * FROM MISSION M WHERE M.EID = E.EID AND M.MLOC = D.DLOC);
```

correlated subquery

9. Find the employees who did a mission in the same city Blake did a mission.

```
SELECT DISTINCT E1.* FROM (EMP E1 NATURAL JOIN MISSION M1) WHERE E1.ENAME <> 'Blake' AND EXISTS (SELECT * FROM EMP E2 NATURAL JOIN MISSION M2 WHERE E2.ENAME = 'Blake' AND M1.MLOC = M2.MLOC);
```

correlated subquery

10. Find the employees who did a mission in all the cities listed in MISSION (2 methods).

4 Grouping

1. For each employee who did at least one mission, display their ID and the number of missions they did.

```
SELECT M.EID,COUNT( M.EID) AS "NUMBER OF MISSIONS" FROM MISSION M GROUP BY M.EID;
```

2. For each employee who did at least one mission, display their name and the number of missions they did.

```
SELECT E.ENAME,COUNT( M.EID) AS "NUMBER OF MISSIONS" FROM EMP E NATURAL JOIN MISSION M GROUP BY E.EID;
```

3. For each employee listed in EMP, display their name and the number of missions they did.

```
SELECT E.ENAME,COUNT( M.EID) AS "NUMBER OF MISSIONS" FROM EMP E NATURAL LEFT OUTER JOIN MISSION M GROUP BY E.EID;
```

4. Find the number of employees each manager (i.e. an employee listed in the MGR column) manages, along with the manager's name.

```
SELECT E1.*, COUNT(E2.EID) FROM EMP E1 JOIN EMP E2 ON E1.EID = E2.MGR GROUP BY E2.MGR;
```

5. For each department, display the name of the department, the number of employees and the highest salary in the department

```
SELECT D.DNAME, COUNT(E.DID), MAX(SAL) FROM DEPT D NATURAL LEFT OUTER JOIN EMP E GROUP BY D.DID;
```

6. Find the average salary per department and per job, along with department and job names.

```
(SELECT D.DNAME, AVG(E.SAL) FROM DEPT D NATURAL LEFT OUTER JOIN EMP E GROUP BY D.DID) UNION (SELECT E.JOB, AVG(E.SAL) FROM EMP E GROUP BY E.JOB);
```

7. Find the highest of the per-department average salary (2 methods).

```
SELECT MAX(DEPTSALARY) FROM (SELECT AVG(SAL) AS DEPTSALARY FROM DEPT NATURAL LEFT OUTER JOIN EMP GROUP BY DID) AS HIGHESTSalary;
```

8. Find the departments with the highest of the per-department average salary.

9. Find the name of the departments with at least 5 employees and no salary less than 900.

```
SELECT DNAME FROM DEPT NATURAL JOIN EMP GROUP BY DID HAVING COUNT(EID) >= 5 AND MIN(SAL) >= 900;
```

10. Find the name of the departments with at least 5 employees and located in Chicago.

```
SELECT DNAME FROM DEPT NATURAL JOIN EMP WHERE DLOC = 'chicago' GROUP BY DID HAVING COUNT(EID) >= 5;
```

5 Miscellaneous

1. Find the departments with no employee earning less than 1,000.

```
SELECT D.* FROM DEPT D WHERE NOT EXISTS (SELECT MIN(E.SAL) FROM EMP E WHERE E.DID = D.DID AND E.SAL < 1000);
```

2. Find the departments with some employees earning less than 1,000.

```
D.* FROM DEPT D NATURAL JOIN EMP E WHERE E.SAL = ANY (SELECT E.SAL FROM EMP E WHERE E.DID = D.DID AND E.SAL < 1000);
```

3. Find the departments with only employees earning less than 1,000.

```
SELECT D.* FROM DEPT D NATURAL JOIN EMP E WHERE E.SAL = ALL (SELECT E.SAL FROM EMP E WHERE E.DID = D.DID AND E.SAL < 1000);
```

4. Find the departments with all of the employees earning less than 1,000

```
SELECT D.* FROM DEPT D NATURAL JOIN EMP E WHERE E.SAL = ALL ( SELECT E.SAL  
FROM EMP E WHERE E.DID = D.DID AND E.SAL< 1000);
```

5. Find (a) the cities listed in tables DEPT or MISSION, (b) the cities listed in both DEPT and MISSION and (c) the cities listed in DEPT but not in MISSION.

```
(SELECT DLOC FROM DEPT) UNION (SELECT MLOC FROM MISSION) ;
```

```
(SELECT DLOC FROM DEPT) INTERSECT (SELECT MLOC FROM MISSION);
```

```
(SELECT DLOC FROM DEPT) EXCEPT (SELECT MLOC FROM MISSION);
```

6. For each city listed in DEPT or MISSION, display the city, the number of employees working in the city (DLOC), the number of employees who did a mission in the city (MLOC).

```
(SELECT D.DLOC,COUNT(E.DID) AS "NUMBER OF EMPLOYEES" FROM DEPT D  
NATURAL JOIN EMP E GROUP BY D.DID) UNION (SELECT M.MLOC,COUNT(M.MLOC)  
AS "NUMBER OF EMPLOYEES" FROM MISSION M GROUP BY M.MLOC) ;
```

7. For each department and for each job listed in EMP, display the department's name, the job, and the number of employees in that department with that job.