

**Advanced Database**

**Lab 2: Advanced SQL Queries**

Tutor : M.Soltani ROOZBEH

Student : Alex DONG

# **Preamble**

**We consider in this lab the database of some company. The schema of the database is as follows:**

**EMP(EID, ENAME, JOB, #MGR, HIRED, SAL, COMM, #DID) MGR references EMP(EID) DEPT(DID, DNAME, DLOC) MISSION(MID, #EID, CNAME, MLOC, ENDD)**

# **Null values**

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

SELECT ENAME FROM `emp` WHERE COMM >= 0;

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

SELECT COUNT(\*) FROM `emp` WHERE COMM >= 0;

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

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

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

SELECT SUM(CASE WHEN COMM IS NULL THEN 1 ELSE 0 END) AS emp\_null\_comm

FROM emp;

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

SELECT Min(COMM) as lowest,

Avg(COMM) as average,

Max(COMM) as highest

FROM `emp`;

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

SELECT AVG(COALESCE(COMM, 0)) as average FROM emp;

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

SELECT

ENAME,

COMM / 1.2 AS commission\_in\_euro

FROM

emp;

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

SELECT

ENAME,

(SAL + COMM) as total\_salary

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 ENAME

FROM emp

WHERE COMM IS NOT NULL AND COMM < (0.25 \* SAL);

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

SELECT ENAME

FROM emp

WHERE COALESCE(COMM, 0) < (0.25 \* SAL);

# **2 SQL92 Join Queries**

**Please answer the following questions using SQL92 joins only.**

**1.Display and Compare the schema and the population of the resulting tables :**

**The product of tables EMP and DEPT :**

It will contain all combinations possibles of rows from both tables.

SELECT \*

FROM EMP, DEPT;

**The theta-join of EMP and DEPT on DID :**

The theta-join will include only rows where the DID in EMP matches the DID in DEPT.

SELECT \*

FROM EMP, DEPT

WHERE EMP.DID = DEPT.DID;

**the natural join of EMP and DEPT :**

The natural join will use common column names to join like here with DID and include only matching rows.

SELECT \*

FROM EMP

NATURAL JOIN DEPT;

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

SELECT EMP.ENAME, DEPT.DNAME

FROM EMP, DEPT

WHERE EMP.DID = DEPT.DID AND DEPT.DLOC = 'NEW-YORK';

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

SELECT EMP.ENAME

FROM EMP, MISSION, DEPT

WHERE EMP.EID = MISSION.EID AND DEPT.DLOC = MISSION.MLOC

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

SELECT E.ENAME AS EmployeeName, M.ENAME AS ManagerName

FROM EMP E

LEFT JOIN EMP M ON E.MGR = M.EID;

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

SELECT E.ENAME

FROM EMP E

JOIN EMP A ON E.MGR = A.MGR

WHERE A.ENAME = 'Allen';

**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 as EmployeeHireDate, M.HIRED as ManagerHireDate

FROM EMP E

JOIN EMP M ON E.MGR = M.EID

WHERE 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, E2.ENAME

FROM EMP E1, EMP E2, DEPT D1, DEPT D2

WHERE E1.DID = D1.DID

AND E2.DID = D2.DID

AND D1.DNAME = 'Sales'

AND D2.DNAME = 'Research'

AND E1.HIRED = E2.HIRED;

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

SELECT D.DNAME

FROM DEPT D

LEFT JOIN EMP E ON D.DID = E.DID

WHERE E.EID IS NULL;

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

SELECT ENAME

FROM EMP

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

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

SELECT ENAME

FROM EMP

WHERE HIRED < ALL (SELECT HIRED

FROM EMP

WHERE DID = (SELECT DID FROM DEPT WHERE DNAME = 'ACCOUNTING'));

# **3 Subqueries**

**For each question below, please indicate whether you are using a standalone or a correlated subquery.**

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

Method 1 (Standalone Subquery):

SELECT E.ENAME

FROM EMP E

WHERE E.SAL = (SELECT MAX(SAL) FROM EMP);

Method 2 (Correlated Subquery):

SELECT E1.ENAME

FROM EMP E1

WHERE NOT EXISTS (SELECT 1 FROM EMP E2 WHERE E2.SAL > E1.SAL);

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

Method 1 (Correlated Subquery):

SELECT E.ENAME

FROM EMP E

WHERE E.SAL < ALL (SELECT M.SAL FROM EMP M WHERE E.EID = M.MGR);

Method 2 (Correlated Subquery):

SELECT E.ENAME

FROM EMP E

WHERE NOT EXISTS (SELECT 1 FROM EMP M WHERE E.SAL >= M.SAL AND E.EID = M.MGR);

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

Method 1 (Correlated Subquery):

SELECT E.ENAME

FROM EMP E

WHERE E.SAL > ANY (SELECT A.SAL FROM EMP A WHERE A.JOB = 'Analyst');

Method 2 (Correlated Subquery):

SELECT E.ENAME

FROM EMP E

WHERE EXISTS (SELECT 1 FROM EMP A WHERE A.JOB = 'Analyst' AND E.SAL > A.SAL);

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

Standalone Subquery:

SELECT ENAME

FROM EMP

WHERE DID IN (SELECT DID FROM DEPT WHERE DNAME IN ('Research', 'Sales'));

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

Method 1 (Standalone Subquery):

SELECT D.DNAME

FROM DEPT D

WHERE DID = ALL (SELECT DID FROM DEPT WHERE DNAME = 'OPERATIONS');

Method 2 (Correlated Subquery):

SELECT D.DNAME

FROM DEPT D

WHERE D.DID NOT IN (SELECT DISTINCT DID FROM EMP);

Method 3 (Standalone Subquery):

SELECT D.DNAME

FROM DEPT D

WHERE NOT EXISTS (SELECT 1 FROM EMP E WHERE E.DID = D.DID);

**6. Find the departments with at least 3 employees.**

Correlated Subquery:

SELECT D.DNAME

FROM DEPT D

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

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

Standalone Subquery:

SELECT DISTINCT E.ENAME

FROM EMP E

WHERE E.EID IN (SELECT DISTINCT EID FROM MISSION);

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

Correlated Subquery:

SELECT EMP.ENAME

FROM EMP, MISSION, DEPT

WHERE EMP.EID = MISSION.EID AND DEPT.DLOC = MISSION.MLOC

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

Correlated Subquery:

SELECT DISTINCT e.ENAME, m.MLOC

FROM emp e

JOIN mission m ON e.EID = m.EID

WHERE m.MLOC IN (SELECT MLOC

FROM mission

WHERE EID = (SELECT EID

FROM emp

WHERE ENAME = 'BLAKE'));

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

Correlated Subquery:

SELECT e.ENAME, m.MLOC

FROM emp e, mission m

WHERE e.EID = m.EID

Correlated Subquery:

SELECT DISTINCT EID, ENAME

FROM emp e

WHERE NOT EXISTS (

SELECT DISTINCT MLOC

FROM mission m

WHERE NOT EXISTS (

SELECT \*

FROM mission m2

WHERE m2.EID = e.EID

AND m2.MLOC = m.MLOC

)

);

# 4 Grouping

**By default, MySQL allows you to write group-by queries that are illegal. Please make sure you answer the following questions with the ONLY\_FULL\_GROUP\_BY mode activated:**

set session sql\_mode = 'ONLY\_FULL\_GROUP\_BY' ;

**This will force MySQL to adhere to standard SQL regarding goup-by queries.**

**test.sql**

*-- Activate ONLY\_FULL\_GROUP\_BY mode*

SET SESSION sql\_mode = 'ONLY\_FULL\_GROUP\_BY';

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

SELECT M.EID, COUNT(\*) AS NumMissions

FROM MISSION M

GROUP BY M.EID

HAVING COUNT(\*) >= 1;

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

SELECT E.ENAME, COUNT(\*) AS NumMissions

FROM EMP E

INNER JOIN MISSION M ON E.EID = M.EID

GROUP BY E.ENAME

HAVING COUNT(\*) >= 1;

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

SELECT E.ENAME, COUNT(M.EID) AS NumMissions

FROM EMP E

LEFT JOIN MISSION M ON E.EID = M.EID

GROUP BY E.ENAME;

*-- 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 M.ENAME AS ManagerName, COUNT(E.EID) AS NumberOfEmployees

FROM EMP M

INNER JOIN EMP E ON M.EID = E.MGR

GROUP BY M.ENAME

ORDER BY M.ENAME;

*-- 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.EID) AS NumEmployees, MAX(E.SAL) AS HighestSalary

FROM DEPT D

LEFT JOIN EMP E ON D.DID = E.DID

GROUP BY D.DNAME;

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

SELECT D.DNAME, E.JOB, AVG(E.SAL) AS AvgSalary

FROM DEPT D

LEFT JOIN EMP E ON D.DID = E.DID

GROUP BY D.DNAME, E.JOB;

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

*-- Method 1: Subquery*

SELECT MAX(AvgSalary) AS HighestAvgSalary

FROM (

    SELECT D.DNAME, AVG(E.SAL) AS AvgSalary

    FROM DEPT D

    LEFT JOIN EMP E ON D.DID = E.DID

    GROUP BY D.DNAME

) AS AvgSalaries;

*-- Method 2: HAVING clause*

SELECT D.DNAME, AVG(E.SAL) AS AvgSalary

FROM DEPT D

LEFT JOIN EMP E ON D.DID = E.DID

GROUP BY D.DNAME

HAVING AVG(E.SAL) = (

    SELECT MAX(AvgSalary)

    FROM (

        SELECT D.DNAME, AVG(E.SAL) AS AvgSalary

        FROM DEPT D

        LEFT JOIN EMP E ON D.DID = E.DID

        GROUP BY D.DNAME

    ) AS MaxAvgSalaries

);

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

SELECT D.DNAME AS DepartmentName, AVG(E.SAL) AS AverageSalary

FROM DEPT D

LEFT JOIN EMP E ON D.DID = E.DID

GROUP BY D.DNAME;

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

SELECT D.DNAME

FROM DEPT D

WHERE (

        SELECT COUNT(\*)

        FROM EMP E

        WHERE E.DID = D.DID

    ) >= 5

AND NOT EXISTS (

    SELECT 1

    FROM EMP E

    WHERE E.DID = D.DID AND E.SAL < 900

);

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

SELECT D.DNAME

FROM DEPT D

JOIN (

    SELECT DID, COUNT(\*) AS NumEmployees

    FROM EMP

    GROUP BY DID

    HAVING NumEmployees >= 5

) AS DeptWithFiveOrMoreEmployees ON D.DID = DeptWithFiveOrMoreEmployees.DID

WHERE D.DLOC = 'Chicago';

# 5 Miscellaneous

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

SELECT D.DNAME

FROM DEPT D

WHERE NOT EXISTS (

SELECT 1

FROM EMP E

WHERE E.DID = D.DID AND E.SAL < 1000

);

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

SELECT DISTINCT D.DNAME

FROM DEPT D

INNER JOIN EMP E ON D.DID = E.DID

WHERE E.SAL < 1000;

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

SELECT D.DNAME AS DepartmentName

FROM DEPT D

WHERE NOT EXISTS (

SELECT 1

FROM EMP E

WHERE E.DID = D.DID AND E.SAL >= 1000

)

AND EXISTS (

SELECT 1

FROM EMP E

WHERE E.DID = D.DID

);

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

SELECT D.DNAME

FROM DEPT D

WHERE NOT EXISTS (

SELECT 1

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.

-- (a) Cities listed in DEPT or MISSION

SELECT DLOC AS City FROM DEPT

UNION

SELECT MLOC AS City FROM MISSION;

-- (b) Cities listed in both DEPT and MISSION

SELECT DLOC AS City FROM DEPT

INTERSECT

SELECT MLOC AS City FROM MISSION;

-- (c) Cities listed in DEPT but not in MISSION

SELECT DLOC AS City FROM DEPT

EXCEPT

SELECT MLOC AS City 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).

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.

SELECT D.DNAME AS DepartmentName, E.JOB, COUNT(\*) AS NumberOfEmployees

FROM DEPT D

JOIN EMP E ON D.DID = E.DID

GROUP BY D.DNAME, E.JOB;