Exercises

Michail Kazimianec
Werner Nutt
Damiano Somenzi

SQL Queries (I)

-- Sample Solutions --

As the name says, these are sample solutions, that is, they are only some of the many different possible solutions.

QUERYING A COMPANY DATABASE

You are asked to write SQL queries that answer the questions listed below.

- 1. What are the names of managers that have a salary between 2000 and 3000?

```
SELECT e.ename
FROM   emp e
WHERE  e.job = 'Manager' AND
       e.sal >= 2000 AND e.sal <= 3000;
```

```
SELECT e.ename
FROM   emp e
WHERE  e.job = 'Manager' AND
       e.sal BETWEEN 2000 AND 3000;
```

- 2. What are the names of the salesmen who have an income (= salary plus commission) above 2000? (Note that salary is never, but the commission can be null).

```
SELECT e.ename
FROM   emp e
WHERE  e.job = 'Salesman' AND
       e.sal + coalesce(comm,0) > 2000;
```

```
*/The PostgreSQL function coalesce takes as argument a list of
expressions and evaluates them one after another until one of them is
not null.  Then the value of that expression is returned as result. If
all argument expressions evaluate to null, then null is the result./*
```

- 3. What are the names of the employees that work in New York?

```
SELECT e.ename
FROM   emp e natural join dept d
WHERE  d.loc = 'New York';
```

```
SELECT e.ename
FROM   emp e, dept d
WHERE  e.deptno = d.deptno AND d.loc = 'New York';
```

- 4. What is the average salary by town?

```
SELECT   d.loc, avg(e.sal)
FROM     emp e NATURAL JOIN dept d
GROUP BY d.loc;
```

```
SELECT   d.loc, avg(e.sal)
FROM     emp e, dept d
WHERE    e.deptno = d.deptno
GROUP BY d.loc;
```

- 5. What are the employees that earn more than their manager?

```
SELECT e.ename
FROM   emp e JOIN emp m ON e.mgr = m.empno
WHERE  e.sal + coalesce(e.comm,0)
                  > m.sal + coalesce(m.comm,0);
```

- 6. Which salesmen earn a salary of grade 2?

```
SELECT   e.ename
FROM     emp e JOIN salgrade s ON e.sal BETWEEN s.losal AND s.hisal
WHERE    s.grade = 2;


SELECT   e.ename
FROM     emp e, salgrade s
WHERE    e.sal BETWEEN s.losal AND s.hisal AND
         s.grade = 2;
```

- 7. What is the salary grade of each employee? (Sort the output according to the salary grades and within each grade, according to the alphabetical order of the names of the employees.)

```
SELECT   s.grade, e.ename
FROM     emp e JOIN salgrade s ON e.sal BETWEEN s.losal AND s.hisal
ORDER BY s.grade, e.ename;
```

- 8. What are the names of the employees who are managers of at least one salesman?

```
SELECT DISTINCT m.ename
FROM   emp e JOIN emp m ON e.mgr = m.empno
WHERE  e.job = 'Salesman';

SELECT m.ename
FROM   emp m
WHERE  m.empno IN (SELECT e.mgr
                   FROM   emp e
                   WHERE  e.job = 'Salesman');
```

/ How would we express that all employees should be deleted that are managers of at least one saleman?/

- 9. Which are the departments that employ someone who is not a salesman?

```
SELECT DISTINCT d.dname
FROM   emp e NATURAL JOIN dept d
WHERE  e.job <> 'Salesman';

SELECT d.dname
FROM   dept d
WHERE  d.deptno IN (SELECT e.deptno
                    FROM   emp e
                    WHERE  e.job <> 'Salesman');
```

- 10. What are the names of the employees who are managers of at least three people?

```
SELECT m.ename
FROM   emp m
WHERE  3 <= (SELECT count(*)
             FROM   emp e
             WHERE  e.mgr = m.empno);

SELECT   m.ename
FROM     emp m JOIN emp e ON m.empno = e.mgr
GROUP BY m.ename
HAVING   count(*) >= 3;
```

- 11. What are the names of the employees who are not managing anyone?

```
SELECT m.ename
FROM   emp m
WHERE  NOT EXISTS (SELECT *
                   FROM   emp e
                   WHERE  e.mgr = m.empno);
```

- 12. What are the names of the employees who either have the job title manager or who are managing other employees.

```
SELECT m.ename
FROM   emp m
WHERE  m.job = 'Manager' OR
       EXISTS (SELECT *
               FROM   emp e
               WHERE  e.mgr = m.empno);
```

- 13. For how many employees is it the case that the employee and

```
       his/her manager work in different towns?

SELECT count(*) AS emps_not_working_with_manager
FROM   emp e NATURAL JOIN dept ed,
       emp m NATURAL JOIN dept md
WHERE  e.mgr = m.empno AND
       ed.loc <> md.loc;


SELECT count(*) AS emps_not_working_with_manager
FROM   emp e, emp m, dept ed, dept md
WHERE  e.mgr = m.empno AND
       e.deptno = ed.deptno AND
       m.deptno = md.deptno AND
       ed.loc <> md.loc;


 - 14. Who is the lowest paid employee in Dallas?

SELECT e.ename
FROM   emp e NATURAL JOIN dept d
WHERE  d.loc = 'Dallas' AND
       NOT EXISTS (SELECT *
                   FROM   emp e_dallas NATURAL JOIN dept d_dallas
                   WHERE  d_dallas.loc = 'Dallas' AND
                          e_dallas.sal < e.sal);

SELECT e.ename
FROM   emp e NATURAL JOIN dept d
WHERE  d.loc = 'Dallas' AND
       e.sal = (SELECT min(e_dallas.sal)
                FROM   emp e_dallas NATURAL JOIN dept d_dallas
                WHERE  d_dallas.loc = 'Dallas');

/* We can create a view for the employees in Dallas and
   use it twice in the query */

CREATE VIEW dallas_emp(ename, sal) AS
SELECT e.ename AS ENAME, e.sal AS sal
FROM   emp e, dept d
WHERE  e.deptno = d.deptno AND
       d.loc = 'Dallas';

SELECT de.ename
FROM   dallas_emp de
WHERE  de.sal <= ALL (SELECT de.sal
                      FROM   dallas_emp de);


/* The same query with min */

SELECT de.ename
FROM   dallas_emp de
WHERE  de.sal = (SELECT MIN(de.sal)
                 FROM   dallas_emp de);

DROP VIEW dallas_emp;


 - 15. For each town, find the names of the employees with the highest
       salary in that town.

SELECT d.loc AS town, e.ename AS employee, e.sal AS salary
FROM   dept d NATURAL JOIN emp e
WHERE  e.sal >= ALL (SELECT e_in_loc.sal
                     FROM   emp e_in_loc NATURAL JOIN dept d_in_loc
                     WHERE  d_in_loc.loc = d.loc);
```