Solutions CS561 HW 6

All queries on this homework relate to the single company database in Elmasri & Navathe

1. In this exercise you are going to write an SQL statement with an embedded sub-query using the NOT EXISTS embedding operator. The query is "Find every person who doesn't work on any project."

   a. Write a schematic version of the SQL statement with the outer part in SQL and the inner part in English – as a strategy, the way we did it in the lecture

   b. Write the whole query in SQL

a. SELECT EMPLOYEE.ssn
   FROM EMPLOYEE
   WHERE NOT EXISTS (set of projects that this employee works on)

b. SELECT EMPLOYEE.ssn
   FROM EMPLOYEE
   WHERE NOT EXISTS (SELECT WORKS_ON.essn
                     FROM WORKS_ON
                     WHERE WORKS_ON.essn = EMPLOYEE.ssn)

2. In this exercise you are going to write an SQL statement with an embedded sub-query using the NOT IN embedding operator. The query is "Find every person who doesn't work on project 12."

   a. Write a schematic version of the SQL statement with the outer part in SQL and the inner part in English – as a strategy, the way we did it in the lecture

   b. Write the whole query in SQL

a. SELECT EMPLOYEE.ssn
   FROM EMPLOYEE
   WHERE EMPLOYEE.ssn NOT IN (set of employees who work on project 12)

b. SELECT EMPLOYEE.ssn
   FROM EMPLOYEE
   WHERE EMPLOYEE.ssn NOT IN (SELECT WORKS_ON.essn
                              FROM WORKS_ON
                              WHERE WORKS_ON.pno = 12)

3. In this exercise you are going to write an SQL statement with an embedded sub-query using the > ALL embedding operator. The query is "Find every department whose minimum salary is greater than the highest salary paid by the R&D department."

a. Write a schematic version of the SQL statement with the outer part in SQL and the inner part in English – as a strategy, the way we did it in the lecture

b. Write the whole query in SQL

a. SELECT DEPARTMENT.dnumber
FROM DEPARTMENT
WHERE ((minimum salary paid by department)
> ALL (set of salaries paid by R&D department))

b. SELECT DEPARTMENT.dnumber
FROM DEPARTMENT
WHERE ((SELECT MIN (EMPLOYEE.salary)
FROM EMPLOYEE
WHERE EMPLOYEE.dno = DEPARTMENT.dnumber)
> ALL (SELECT EMPLOYEE.salary
FROM EMPLOYEE, DEPARTMENT
WHERE (EMPLOYEE.dno = DEPARTMENT.dnumber)
AND (DEPARTMENT.name = "R&D")))

4. In this exercise you are going to write an SQL statement with an embedded sub-query using the > SOME embedding operator. The query is "Find all people who work for department 11 and don't make the lowest salary paid by department 11."

a. Write a schematic version of the SQL statement with the outer part in SQL and the inner part in English – as a strategy, the way we did it in the lecture

b. Write the whole query in SQL

a. SELECT EMPLOYEE.ssn
FROM EMPLOYEE
WHERE (employee works for department 11)
AND (employee's salary
> SOME (set of salaries paid by department 11))

b. SELECT EMPLOYEE.ssn
FROM EMPLOYEE
WHERE (EMPLOYEE.dno = 11)
AND (EMPLOYEE.salary
> SOME (SELECT EMPLOYEE.salary
FROM EMPLOYEE
WHERE (EMPLOYEE.dno = 11))))

5. Translate the following query into SQL: For each project, find the average salary of people who work on it. Output both the project number and the average salary.

```
SELECT WORKS_ON.pno, AVG(EMPLOYEE.salary)
FROM WORKS_ON, EMPLOYEE
WHERE WORKS_ON.essn = EMPLOYEE.ssn
GROUP BY WORKS_ON.pno
```

Note: It's possible to argue that a project on which nobody works has an average salary – of people who work on it – of zero; it's also possible to argue that a project on which nobody works has an undefined average salary. I've taken the second point of view in writing this query.

6. Translate the following query into SQL: For each city in which at least one department has a location, find the number of departments that have locations in the city. Output both the city and the number of departments that have locations in it.

```
SELECT DEPT_LOCATIONS.dlocation, COUNT (*)
FROM DEPT_LOCATIONS
GROUP BY DEPT_LOCATIONS.dlocation
```

7. Translate the following query into SQL using a GROUP BY clause and a HAVING clause: For each department with at least ten locations, find the average salary of the department's employees. Output the department number, the department name, and the average salary.

```
SELECT DEPARTMENT.dnumber, DEPARTMENT.dname, AVG (EMPLOYEE.salary)
FROM DEPT_LOCATIONS, EMPLOYEE, DEPARTMENT
WHERE (DEPT_LOCATIONS.dno = EMPLOYEE.dno)
        AND (EMPLOYEE.dno = DEPARTMENT.dnumber)
GROUP BY DEPARTMENT.dnumber, DEPARTMENT.dname
Must be
HAVING COUNT (DISTINCT DEPT_LOCATIONS.dlocation) >= 10
because each of the GROUP BY sub-tables has each location repeated as many times as there are
employees of the department.
```

8. Translate the following query into SQL using a GROUP BY clause **but no** HAVING clause: For each department with at least ten locations, find the average salary of the department's employees. Output the department number, the department name, and the average salary.

```
SELECT DEPARTMENT.dnumber, DEPARTMENT.dname, AVG (EMPLOYEE.salary)
FROM EMPLOYEE, DEPARTMENT
WHERE  (EMPLOYEE.dno = DEPARTMENT.dnumber)
        AND (SELECT COUNT(*)
                FROM DEPT_LOCATIONS
                WHERE DEPARTMENT.dnumber = DEPT_LOCATIONS.dno) >= 10
GROUP BY DEPARTMENT.dnumber, DEPARTMENT.dname
```