# 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.

SELECT EMPLOYEE.ssn

FROM EMPLOYEE

WHERE NOT EXISTS (set of employee on the work\_on)

b. Write the whole query in SQL

SELECT EMPLOYEE.ssn

FROM EMPLOYEE

WHERE NOT EXISTS (SELECT \*

FROM WORK\_ON

WHERE EMPLOYEE.ssn = WORK\_ON.essn)

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

SELECT EMPLOYEE.ssn

FROM EMPLOYEE

WHERE NOT IN (set of employees who work on project 12)

b. Write the whole query in SQL

SELECT EMPLOYEE.ssn

FROM EMPLOYEE

WHERE NOT IN (SELECT WORKS\_ON.essn

FROM WORKS\_ON

WHERE (WORKS\_ON.essn = EMPLOYEE.ssn

AND (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

SELECT DEPARETMENT.dname

FROM DEPARETMENT, EMPLOYEE

WHERE (EMPLOYEE.dno = DEPARTMENT.dnumber)

and MIN(EMPLOYEE.salary)

>ALL (set of highest salaries of employees

of R&D’s department)

b. Write the whole query in SQL

SELECT DEPARTMENT.dname

FROM DEPARETMENT, EMPLOYEE

WHERE (EMPLOYEE.dno = DEPARTMENT.dnumber)

and MIN(EMPLOYEE.salary)

>ALL (SELECT MAX(EMPLOYEE.salary)

FROM EMPLOYEE, DEPARTMENT as RD

WHERE ((EMPLOYEE.dno = RD.dnumber)

AND (RD.dname = “ R&D”)

AND (RD.dname <> DEPARTMENT.dname)))

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

SELECT EMPLOYEE.ssn

FROM EMPLOYEE

WHERE EMPLOYEE.dno = 11

AND EMPLOYEE.salary > SOME(set of employees salary of employees who work for department 11)

b. Write the whole query in SQL

SELECT EMPLOYEE.ssn

FROM EMPLOYEE

WHERE (EMPLOYEE.dno = 11)

AND (EMPLOYEE.salary > SOME(SELECT EMPLOYEE.saraly

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 PROJECT.pnumber, AVG(EMPLOYEE.salary) AS avg\_salary

FROM WORKS\_ON, EMPLOYEE, PROJECT

WHERE (WORKS\_ON.essn = EMPLOYEE.ssn)

AND (WORKS\_ON.pno = PROJECT.pnumber)

GROUP BY PROJECT.pnumber

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 d1.dept\_city, COUNT(\*) AS num\_departments

FROM (SELECT DISTINCT dept\_city, dept\_no

FROM department) d1

GROUP BY d1.dept\_city

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.dept\_no, department.dept\_name, AVG(employee.salary) AS avg\_salary

FROM department, dept\_locations, employee

WHERE department.dept\_no = dept\_locations.dept\_no

AND department.dept\_no = employee.dept\_no

GROUP BY department.dept\_no, department.dept\_name

HAVING COUNT(DISTINCT dept\_locations.loc) >= 10

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, AVE(EMPLOYEE.salary)

FROM DEPARTMENT, EMPLOYEE

WHERE (( SELECT COUNT(\*)

FROM DEPT\_LOCATION

WHERE DEPT\_LOCATION.dlocation = DEPARTMENT.dlocation)>= 10)

AND (EMPLOYEE.dno = DEPARTMENT.dnumber)

GROUP BY DEPARTMENT.dnumber