Assignment 7: ADVANCED Problems using Joins, Grouping and Subqueries

The problems in this assignment section get even more challenging. You'll acquire practice using Joins and working with subqueries. You will also be tested on your knowledge of correlated subqueries. Take as long as you need to work through this section.

The questions that follow will be related to the tables that you created in assignment one. Query those tables and try to figure out how the data is related. Those tables are: students, courses, student_enrollment, professors, and teach. The follow problems are related to these.

Questions for this assignment

1. Are the tables student_enrollment and professors directly related to each other? Why or why not?

2. Write a query that shows the student's name, the courses the student is taking and the professors that teach that course.

3. If you execute the query from the previous answer, you'll notice the student_name and the course_no is being repeated. Why is this happening?

4. In question 3 you discovered why there is repeating data. How can we eliminate this redundancy? Let's say we only care to see a single professor teaching a course and we don't care for all the other professors that teach the particular course. Write a query that will accomplish this so that every record is distinct.

HINT: Using the DISTINCT keyword will not help. :-)

5. Why are correlated subqueries slower that non-correlated subqueries and joins?

6. In the video lectures, we've been discussing the employees table and the departments table. Considering those tables, write a query that returns employees whose salary is above average for their given department.

7. Write a query that returns ALL of the students as well as any courses they may or may not be taking.

Do not scroll past here without trying out the assignment yourself

Instructor Solutions for this assignment

1. Are the tables student_enrollment and professors directly related to each other? Why or why not?

They are NOT related directly. The reason is, there is no common column shared amongst them. There cannot be a direct relationship formed between these 2 tables.

2. Write a query that shows the student's name, the courses the student is taking and the professors that teach that course.

```
SELECT student_name, se.course_no, p.last_name

FROM students s

INNER JOIN student_enrollment se

   ON s.student_no = se.student_no

INNER JOIN teach t

   ON se.course_no = t.course_no

INNER JOIN professors p

   ON t.last_name = p.last_name

ORDER BY student_name;
```

3. If you execute the query from the previous answer, you'll notice the student_name and the course_no is being repeated. Why is this happening?

The combination of student_name and course_no is being repeated for as many professors that are teaching that particular course. If you ORDER BY the student_name column, you'll clearly be able to see that multiple professors are teaching the same subject. For example, course CS110 is being taught by both Brown and Wilson. That is why you'll see the combination of the student Arnold with CS110 twice. Analyze the data and understand what's going on because in the next question you'll need to write a query to be eliminate this redundancy.

4. In question 3 you discovered why there is repeating data. How can we eliminate this redundancy? Let's say we only care to see a single professor teaching a course and we don't care for all the other professors that teach the particular course. Write a query that will accomplish this so that every record is distinct.

HINT: Using the DISTINCT keyword will not help. :-)

```
SELECT student_name, course_no, min(last_name)
FROM (
SELECT student_name, se.course_no, p.last_name
FROM students s
INNER JOIN student_enrollment se
   ON s.student_no = se.student_no
INNER JOIN teach t
   ON se.course_no = t.course_no
INNER JOIN professors p
   ON t.last_name = p.last_name
   ) a
   GROUP BY student_name, course_no
   ORDER BY student_name, course_no;
```

5. Why are correlated subqueries slower that non-correlated subqueries and joins?

A "correlated subquery" (i.e., one in which the where condition depends on values obtained from the rows of the containing/outer query) will execute once for each row. A non-correlated subquery (one in which the where condition is independent of the containing query) will execute once at the beginning. If a subquery needs to run for each row of the outer query, that's going be very slow!

6. In the video lectures, we've been discussing the employees table and the departments table. Considering those tables, write a query that returns employees whose salary is above average for their given department.

```
SELECT first_name
  FROM employees outer_emp
  WHERE salary > (
    SELECT AVG(salary)
      FROM employees
      WHERE department = outer_emp.department);
```

7. Write a query that returns ALL of the students as well as any courses they may or may not be taking.

```
SELECT s.student_no, student_name, course_no
FROM students s LEFT JOIN student_enrollment se
    ON s.student_no = se.student_no
```