

W5. SQL Exercises

Part I. Little puzzles

1. What is this query computing?

```
SELECT *  
FROM Movies  
WHERE length <=120 OR length > 120;
```

2. What is the difference between

```
SELECT count(*)  
FROM Movie  
WHERE studio = 'Disney';
```

AND

```
SELECT count(length)  
FROM Movie  
WHERE studio = 'Disney';
```

3. What is the difference here?

```
SELECT COUNT(DISTINCT producer_cert)  
FROM Movie  
WHERE studio = 'Disney';
```

```
SELECT DISTINCT COUNT(producer_cert)  
FROM Movie  
WHERE studio = 'Disney';
```

4. What is the meaning of this query?

```
SELECT AVG(DISTINCT length)  
FROM Movie  
WHERE studio = 'Disney';
```

Part II. Understanding subqueries

1.

```
SELECT t.student_id, t.course_id, t.grade  
FROM takes t,  
      (SELECT * FROM teaches  
       WHERE professor_id=1) pcourses  
WHERE t.course_id = pcourses.course_id;
```

Can it be written without subquery?

```

2. SELECT p.lname, c.id
   FROM professor p, course c, teaches t,
        ( SELECT student_id, course_id
           FROM takes
           WHERE grade >= 90 ) High
   WHERE
   t.course_id = c.id AND t.professor_id = p.id AND
   c.id = High.course_id;

```

How can it be rewritten to make it more readable?

```

3. SELECT lname FROM student s
   WHERE s.id IN (SELECT student_id FROM takes);

```

Can we express this query without using IN?

```

4. SELECT pouter.lname
   FROM professor pouter
   WHERE EXISTS
        (SELECT '1'
         FROM student sinner
         WHERE sinner.lname = pouter.lname);

```

```

5. SELECT DISTINCT course_id
   FROM takes outer_t
   WHERE EXISTS
        ( SELECT *
          FROM takes t, course c, offeredby do
          WHERE
          t.course_id = c.id AND
          c.id = do.course_id AND
          t.course_id <> outer_t.course_id AND
          do.department_id = 1 AND
          outer_t.student_id = t.student_id );

```

Can this query be simplified?

```

6. SELECT professor_id
   FROM course c1, teaches t1
   WHERE c1.id = t1.course_id AND
   NOT EXISTS
        ( SELECT *
          FROM course, teaches
          WHERE
          course.id = teaches.course_id AND
          course.id <> c1.id AND
          professor_id = t1.professor_id);

```

Can the query be simplified?

Part III. More complex queries on Movies

1. Find the names of all stars which starred in at least 3 movies (according to our database).
2. Find the stars who have worked for only one studio.
3. Find the stars who have worked for Disney **but no other** studio.
4. Find the shortest movie of Disney.
5. For each star that has at least two movies with Paramount, find how many movies he/she has with Fox.

6. Find the stars who have worked for **every** studio.

7. Suppose StarsIn relation has an additional attribute "salary"

StarsIn (movie_title, movie_year, star_name, salary)

Now, find the stars who were paid for some movie more than the average salary for that movie.