Exam

SD202 – Databases

October 29, 2021

This is the final exam for the SD202 class, which will determine 100% of your grade for this class. The exam consists of two independent parts. **You must write your answer to each part on a *separate sheet* of paper.** You can choose to answer the questions in English or in French, as you prefer.

Write your name clearly on every sheet used for your exam answers, and number every page.

You are allowed **three A4 sheets** (i.e., six pages, one on each side) with personal notes of your choice. You may not use any other written material.

The exam is **strictly personal**: any communication or influence between students, or use of outside help, is prohibited. No electronic devices such as calculators, computers, or mobile phones, are permitted. Any violation of the rules may result in a grade of 0 and/or disciplinary action.

Part 1

Exercise 1: Functional dependencies

Question 1. Consider the following relation R having four attributes a, b, c, and d, and four tuples:

R

a	b	c	d
1	10	20	30
1	11	21	31
2	10	22	30
2	12	23	31

Which of the following FDs hold?

- $a \to b$
- $c \to ad$
- $d \to b$
- $b \to d$
- $ab \to c$
- $bd \to c$

Question 2. Propose a table of a relation S with three attributes a, b, and c, satisfying the following FDs:

- $ab \to c$
- $c \to b$

but not the following ones:

- $a \to c$
- $b \to c$
- $c \to a$

Exercise 2: Aggregation queries

We consider a table Student with the following attributes:

- id, a unique identifier, the primary key of the database,
- name, a string, the name of the student,
- grade, an integer, the student's grade for the year (averaged over all their courses),
- sgroup, an integer, the identifier of a student group to which the student belongs.

Question 1 Write a query in SQL returning one floating point value which is the average of the grade of all students.

Question 2. Write a query in SQL returningd the list of all group ids (without duplicates) where there is a student having a grade of 10 or less.

Question 3. Write a query in SQL to list the groups and the average of the grade of the students in each group, sorted by decreasing average.

Question 4. Write a query in SQL to compute the id, name, and grade of the student with the best grade (breaking ties at random).

Question 5. Explain in English (or French) what the following query computes.

```
SELECT S1.name, S2.name FROM Student AS S1, Student AS S2
    WHERE S1.grade = S2.grade AND S1.sgroup <> S2.sgroup;
```

Question 6. Explain in English (or French) what the following query computes.

```
SELECT S1.sgroup FROM Student AS S1
  WHERE NOT EXISTS (SELECT 1 FROM Student AS S2
    WHERE S2.id <> S1.id AND S1.sgroup = S2.sgroup);
```

Rewrite this query to a query that does not have a nested subquery and is equivalent (assuming that there are no NULLs in the database).

Part 2

*Reminder: please write your answers to the exercises in Part 2 on a **separate sheet of paper**!*

Exercise 3: Schema design

We wish to design a database system for a theater. Here is a specification of the business need:

> *The theater manages people (actors and stage directors), who have a name and a phone number. The theater welcomes several productions. A production has a title, exactly one stage director, and a list of characters: each character has a name and occurs in exactly one production. Each production is played on different dates: this is called a show. Each show plays exactly one production, and has exactly one date. Then, for each show, there is a cast, specifying which actor plays which character. On every show, every character of the play will be played by exactly one actor, but an actor can play several characters. Note that the characters and the stage director are always the same for a given production; by contrast, the information of which actor plays which character may vary depending on the show.*

Question 1. Write the SQL instructions to create a database schema satisfying the business need. Make sure to define primary keys and foreign keys.

Question 2. Write an SQL query over your schema to retrieve the name and phone number of actors who are playing today (October 29, 2021).

Question 3. Write an SQL query in your schema to retrieve the name and phone number of all stage directors who also play as an actor in some show of a production for which they are the stage director.

Exercise 4: Query evaluation

Consider a database where we have three tables:

- A table persons describing persons and containing two attributes:
  - name of type text, the name of the person,
  - id of type integer and also the primary key, a unique identifier for that person.
- A table movies describing movies containing four attributes:
  - id of type integer and also the primary key, a unique identifier for that movie,
  - title of type text, the title of the movie,
  - year of type integer, the year the movie was released,
  - director of type integer which is a foreign key referencing the id column in the persons table.
- A table actedIn describing which person played in which movie containing two attributes:
  - idPerson of type integer and a foreign key referencing the id column in the persons table,
  - idMovie of type integer and a foreign key referencing the id column in the persons movies.

If it helps, you can make reasonable assumptions about the content of the database (e.g. all movies in the database are released between 1850 and 2050, most movies have between 5 and 50 actors, etc.).

Question 1. When creating the tables, the `PostgreSQL` database engine automatically creates an index for the primary keys. Explain in one paragraph why `PostgreSQL` creates these indexes?

In addition to the indexes for the two primary keys, we also suppose that there are two other indexes: one on the title column of movies and one on the director column of movies. We also suppose that all the three tables are populated with many records.

Question 2. Consider the following query:

```
SELECT p.name, m.title
FROM movies m, persons p, actedIn a
WHERE
      a.idMovie = m.id
  AND a.idPerson = p.id ;
```

Explain in English (or French) what this query computes. Explain a reasonably efficient query plan that a database engine (such as `PostgreSQL`) might consider to run this query.

Note that for this question and the two following questions, there are multiple possible answers. Your answer will be accepted as long as the plan that you propose is reasonably efficient. You must describe the query plan in English (or French) and justify briefly why the proposed plan is interesting for this query; for instance, why it is better than some more naive plan, or how many operations it performs.

Question 3. Consider the following query:

```
SELECT m1.director, m1.title, m2.title
FROM movies m1, movies m2
WHERE   m1.director = m2.director
    AND m1.id != m2.id
    AND m1.year = m2.year
```

Explain what this query computes. Explain a reasonably efficient plan for this query. Could you create an index to optimize how `PostgreSQL` evaluates this query?

Question 4. Consider the following query:

```
SELECT m2.title
FROM movies m1, movies m2
WHERE
      m1.title = 'Snatch'
  AND m1.director = m2.director
  AND m1.year < m2.year  ;
```

Explain what this query computes. Explain a reasonably efficient plan for this query. Could you create an index to optimize how `PostgreSQL` evaluates this query?