ECE PARIS
ÉCOLE D'INGÉNIEURS

Prénom : Jeremy
NOM : ....
Promotion :
Groupe : ....

ING 4
Bases de données
Devoir surveillé

Int.

16 décembre 2016
17:00 - 19:00
Durée : 02:00

Sujet proposé par :	BUSCA Jean-Michel
Calculatrice autorisée :	NON
Documents autorisés :	NON
Ordinateur autorisé :	NON

RAPPEL :

- NOM et Prénom de l'élève doivent être portés sur toutes les copies rendues.
- Les copies doivent être numérotées.
- Tous les appareils électroniques (téléphones portables, PDA, ordinateurs, montre connectée, etc.) doivent être éteints et rangés.
- **Toute erreur constatée sur le sujet doit être signalée sur la copie. Le correcteur en tiendra compte lors de la correction du devoir.**
- Il est interdit de communiquer.
- Toute fraude, ou tentative de fraude, qu'elle soit passive ou active, fera l'objet d'un rapport de la part du surveillant et sera sanctionnée par la note zéro, assortie d'une convocation devant le Conseil de discipline. Aucune contestation ne sera possible. Tous les documents et supports utilisés frauduleusement, devront être remis au surveillant.
- Les élèves ne sont pas autorisés à quitter la salle où se déroule l'épreuve moins de 45 minutes après le début de l'épreuve. Au-delà de ces 45 premières minutes, toute sortie est définitive (sauf dans le cas d'une épreuve durant plus de deux heures).

Advanced Databases SI INI – Exam

Duration: 2 hours. No document, no electronic device allowed. The marking-scheme is tentative. Please clearly indicate the number of the question and sub-question you are answering.

Throughout the exam, we consider the database of a school, which includes information about its courses, its students and their enrollments. The relational schema of the database is as follows:

`students(sid, sname, address, syear)`

> Student ID, name, address and year, e.g. syear = 4 for a 4th year student. Note: Two students may have the same name.

`courses(cid, cname, duration, cyear)`

> Course ID, name, duration, e.g. 30 hours, and year, e.g. cyear = 4 for a 4th year course. Note: Two courses may have the same name.

`enrolled(#sid, #cid, mark)`

> Student ID, course ID, and the final mark the student got in the course; `mark` is null if the exam was not taken yet. Note: new courses and new students might not have any enrollment yet.

Unless otherwise stated, all answers must be given in **standard** SQL.

1. SQL Queries (6 marks)

Write the SQL query that returns each piece of information below. **Beware**: any query that is not syntactically correct will be marked 0.

1. The name of the students who don't have any mark below 10
2. The pair of students (names) who have the same mark in the same course
3. The name of the students who are not enrolled in all the courses corresponding to their year
4. For each course, its name and the name of the students who have the highest mark in the course
5. For each course, its name, the number of enrolled students and the highest mark
6. For each year, the name of the courses with the longest duration

2. Creating the Database (3 marks)

Write the SQL statements that create the tables of the database.

3. Constraints (2 marks)

Modify and/or complete the above statements to include the following constraints:

- the duration of 1st and 2nd year courses must not exceed 24 hours
- the per-year average duration of courses must not exceed 20 hours

For each of the above constraint, name the type of SQL constraint you need and explain why.

4. Triggers (2 marks)

You may use MySQL's syntax to answer that question.

4.1 Write one or more triggers that ensure that all course names are stored in upper case in the database.

4.2 The database contains the table `studenthistory(sid, udate, address, year)`, whose purpose is to keep track of student information updates from the time the student enrolls the school until the time he/she leaves it after graduating. Write the triggers that automatically populate `studenthistory`.

5. Conceptual Modeling (3 marks)

Draw the Entity/Relationship diagram of the database, using **three entity sets**: Students, Courses and Enrollments. You must use the formalism described in the lecture notes.

6. Java DataBase Connectivity (2 marks)

Consider the following class:

```
public class DataAccess {

    private Connection connection;

    public DataAccess(String url, String login, String password)
       throws SQLException {
          connection = DriverManager.getConnection(url, login, password);
    }

    public List<Float> getMarks(int year) throws SQLException {
          // returns the marks of the students of the specified year
    }

}
```

Develop the `getMarks()` method using prepared statements. You may modify the `DataAcces` class if needed.

7. Views (2 marks)

We assume that every student can connect to the database using their `sid` as login, i.e. when the student with `sid` 1234 connects to the database, the `current_user()` function returns 1234. We also assume that the tables of the database were created by the database administrator. Define a view that allows any student to see the courses he/she is enrolled in and the marks he/she has, **and only this information**.