Final: Databases — Spring 2020

Hand in as PDF files with your solution amply documented with screen shots of the results. You are not allowed to communicate about this final with anyone other than me until you and your communications partner have submitted. This does not include online documentation, manuals, stack-exchange answers and questions, but it would include for example a forum discussion that you initiate.

Problem 1:

10 points

(A) In this problem, use SQL statements only. Underlined attributes are primary keys. Create a database 'college' with the following tables:

```
students( sno, first_name, last_name, birthdate, entrance, graduation)
```

The last three should be dates. All attributes should be not-null, with the exception of the last that should have a default value of NULL. sno is an integer.

```
departments( abbr, name, chair)
```

The abbreviation is a four-letter abbreviation (like COSC or COEN), which should be reflected in the datatype. The chair's name should have default NULL, but the other ones should not be nullable.

```
courses(dep, number, semester)
```

The dep-attribute is the same as the abbreviation in departments, the number is an integer, and semester is a 5-letter character such as 'F2020' and 'S2020' to denote the fall and the spring semester. None of these should be allowed to be null. The dep attribute needs to exist in departments.

```
takes(student_id, dep, number, semester)
```

The student_id attribute is the same as sno and has to exist in the students table. The other three attributes need to exist in the courses table. None of the attributes are nullable.

10 points

(B) Insert the MATH-mathematics, COEN-computer engineering, and COSC-computer science departments into the department table.

Insert the values on the right into the courses table using MySQL.

Then use the linked exam.sql file in order to insert into students and takes. You can either execute the script or copy and paste. If you get errors, it should be because you made a mistake in the table definition.

dep	number	semester	
COEN	101	F2020	
COEN	101	S2020	
COEN	102	F2020	
COEN	102	S2020	
COEN	110	F2020	
COEN	110	S2020	
COSC	101	F2020	
COSC	101	S2020	
COSC	110	F2020	
COSC	110	S2020	
MATH	101	F2020	
MATH	101	S2020	
MATH	102	S2020	
▶ MATH	110	F2020	

10 points	(C) Create a view of all student first and last names as well as their birth dates that take COSC 110 in Fall 2020. Give a screenshot of the first 20 values in the view.
10 points	(D) Find all students who take the same class in Fall 2020 as in Spring 2020.
10 points	(E) Notice how the names of the students are repetitious. This is a consequence of me using a rather short list of first names and an only slightly larger list of last names (animals and chess openings). However, since the birth dates are randomly generated dates for 17 to 20 year olds, the combination of student first name, last name and birth date is unique. For each student first and last name, give the number of other students with the same first name and the same last name. Prevent counting the same pair of homonyms twice. E.g. if there are Michael Tartakovers with birth dates 1-1-2000, 1-2-2000, and 1-3-2000 and no other Michael Tartakover, the result of your query should have a row `Michael Tartakover 3.`
10 points	(F) If you solved the previous problem correctly, you can of course use it to find the maximum number of students with the same first and last name by scanning. Now write a single SQL query that finds this maximum number of homonym student names.

Select 3 of the following 4 problems

Problem 2:

10 points	Write a stored procedure that creates table with all students (student number, first name, last name, and birth date) who take more than 4 classes in a given semester.

Problem 3:

10 points	Add a table to the database graduated with a single attribute, the sno of a student. Then write a trigger that adds the sno of any students whose graduation date is updated to a non-null value.

Problem 4:

10 points	Create a Python (or whatever language you like to work with best that has a connector to sql). Create a program that adds 500 students with entrance data 1/1/2021 into the database. Names and birth dates should be generated semi-randomly.

Create a Python script that writes out the name of all students who are both enrolled in COSC 101 and COEN 101 in the same semester. Provide the scripts and screenshots of the results. |

Problem 5:

10 points	Create a MongoDB collection inventory (as in https://docs.mongodb.com/manual/tutorial/update-documents/). Create queries for:
• Update the quantity for sketchbook to 30
• Insert today's date for all entries with a quantity of 50 or more as a field with key "inventorized".
• Add a document "description" to sketchbook that describes a sketchbook in an English language sentence.