Assignment 1. Database design and implementation. Embedded SQL. Basic queries.

It is time to put your newly acquired skills into practice

In this assignment you are asked to design and implement a database app for managing and analyzing University course experiences. Let's call it the Course Experience App (CEA).

Specifications

The CEA database stores information about courses offered by the University, some basic information about instructors and students, and student course experiences.

Information about courses consists of the following facts: course number, the name and the code of the offering department, the general area of study (e.g. humanities, sciences, arts, business, engineering), a list of topics covered in the course, and a list of skills acquired upon successful completion of this course. We also store information about prerequisite and exclusion courses for any given course. Note that a course number is unique only within a specific department.

For each instructor, we store a name, gender, age, and areas of expertise (if applicable). An instructor can be permanent faculty or a freelancer. If an instructor is faculty, we store the year when they started their career as a faculty, and whether they are teaching or research faculty. If an instructor is research faculty, we record a list of their research interests.

For each student - identified by a unique user name - we store basic demographic data such as year and month of birth, gender, and country of birth. In addition, we store the year and month when this student begun his study at university. Additional information about students include their professional history outside academia. For each significant job that the student held in the past or is holding now, we record the job title, Company name (or self-employed), Company expertise (can be more than one), start and end date of the employment, and a list of skills used during or acquired on the job. For each skill, we record a level of expertise on a scale of beginner-intermediate-advanced-professional (2-5).

Each time the course is offered we call it an 'edition' of the course. For each edition we record the start and end date, the time of day the course was offered (morning, day, evening), an approximate overall number of students enrolled in this course, and the name of the course instructor. Each edition of a course can be co-taught by more than one instructor.

For the main part of this database app, we record all course editions taken by a particular student, together with the final grade (in numeric format - percentage). The student records their experiences after enrolling in an edition of the course. The experiences consist of the following facts:

- Overall satisfaction with the course on a scale from 1 to 5 (5 is the highest).
- An overall ranking of the course instructor(s) (5 is the highest).
- The list of skills that the student has acquired from this course. For each skill we record 2 self-assessment values: the rank of a skill before the course and the rank of the same skill upon the course completion – on a scale 1-zero knowledge, 2-beginner, 3-intermediate, 4-advanced, 5-professional.
- Overall interest in the course topics – also 2 values on a 5 point scale – one specifies the interest before taking the course, and the second - the interest after its completion.

Now the assignment.

Part 1. Database design and implementation

1.1. Create an E/R diagram of the CEA database

1.2. Convert this diagram into a relational schema. Specify all basic attribute types. Specify all primary keys.

1.3. Think of and list all required value constraints on course codes, department codes, start and end dates, course grades, year, ranking, semester, time of day, gender, and anything else you can think of to prevent inconsistent data from being added to the database.

1.4. Convert your logical CEA schema into a physical schema - set of tables with constraints - using the PostgreSQL DBMS. Learn how to create new data types such as domains and enums and apply this knowledge to restrict the domains of each data column.

1.5. Now, recall all the courses you have taken so far, and write and run SQL insert statements for inserting data about departments, courses and the instructors you encountered during your study at UofT.

1.6. Write all the required insert statements to enter complete information for two students (you and your project partner).

1.7. After this task is complete, convert your physical schema into an equivalent schema for SQLite. Record and report all the difficulties that you encountered during this process. Are there any constraints that you were not able to enforce in SQLite?

1.8. If you performed tasks 1.5 -1.6 accurately, the same insert statements should also work for SQLite. Insert your initial data into an SQLite database using your sql script for insertions.

Part 2. SQL embedded into a JDBC application

2.1. Create a Java application which provides an interface to enter user experiences into an SQLite version of the CEA database. A student who was previously added to the database, should be able to identify himself, to select a course from a list of available courses, and to insert all the experiences described in the specification.

2.2. Use the interface (either GUI or TUI) to truthfully and accurately record your own experiences about each course you have taken.

2.3. In order to perform meaningful data analytics in the second assignment, we will combine all the information you entered into a single database. Towards this goal, write an export utility which will perform basic select-project-join operations on all your data tables, and will output the data as a single report in a (comma-separated) csv format.

Each row in this table should contain complete information about a particular (course edition – user name pair). The column headers are as follows (be precise):

- Course number (with department code)
- Course start date
- Course end date
- Time of day
- Total number of students
- Name of the course instructor(s): one or many delimited by |
- Student user name
- Course grade
- Student age
- Student year and month of birth, space delimited
- Student gender
- Student country of birth
- List of skills acquired outside academia in form of skill-rank pair, delimited by |
- Student start date at the University
- Course satisfaction
- Instructor ranking
- List of skills learned in the course: in form of skill - rank before- rank after, delimited by |
- List of topics learned in the course: in form of topic-interest before-interest after, delimited by |

All the rankings are numeric on the scale 1-5 (5 is the highest)

Submit:

Part 1: E/R diagram, logical schema (in file **CEA_design.pdf**), table and constraint definitions for PostgreSQL (in file **CEA_db_postgre.sql**), the same for SQLite (**CEA_db_sqlite.sql**) plus the report about all the challenges in converting from Postgre to SQLite (in file **PG_vs_SQLite.pdf**), all the initial insert statements (in file **CEA_data.sql**)

Part 2: Your UI JDBC app deployed as a single **CEA.jar** file, which includes the source code (.java files), and also all the libraries that you used in your application. If you are unable to run this file from a command line: java -jar CEA.jar, then create and submit a special version of executable jar file named **CEA_x.jar**. The table with all the data in file **CEA_export.csv**.

Marking schema:

1.1. E/R diagram – 15%
1.2. Logical schema with referential integrity constraints – 10%
1.3. Value constraints – 5%
1.4. Physical implementation of tables and constraints in PostgreSQL – 15%
1.5. Initial insert statements about courses and instructors – 5%
1.6. Initial insert statements about students – 5%
1.7. Physical implementation of tables and constraints in SQLite – 5%

2.1. JDBC application with user interface – 15%
2.2. Course experience data -10%
2.3. Data export code and resulting report -15%

After submission, you are responsible for scheduling a face-to-face review with the instructor or one of the TAs to demonstrate your work in order to ensure fair grading and high-quality feedback.

This assignment is worth 15% of the course grade.