Database

Project Description

Due Date: 2023-05-28

Contact person: Ilir Jusufi: ilir.jusufi@bth.se
Birgitta Hermanson: birgitta.hermanson@bth.se

This is a final project specification. You are responsible to come up with an interesting problem that can be solved with a database and programming. It is important that you find a reasonable scope for your project. You can use whatever interface you like, either multiple programs, and console UI, a web interface, or whatever else you like. All answers should be your own. You are allowed to work in groups of two. Make sure you include your names in the report when you submit.

Tasks

1.Idea

Come up with an idea for your project. Describe what problem it solves, who the main user(s) will be, and why your idea is a good fit for them and the problem. Describe the main features that your application must have to be complete including the source of the data. You can collect real data for this, or use any appropriate data generator to populate your database.

Optional deadline task: You can submit your idea, one page maximum, as an intermediate step to get feedback about the scope of your project. We will primarily give you hints about if your project is “large” or “complex” enough to get a passing grade.

Deadline: Monday 24 of April

2. Logical model

Design a data model for your project and present it as an E/R diagram. Make sure to include important attributes and relationships. Discuss and motivate your design.

3. Translate the Design to SQL schema

Translate your design to collections in SQL. Discuss and motivate how you translated entities and relationships.

4. SQL queries

Create at least five queries to your SQL design that are needed to implement the functionality of your application. You will probably need to create more than five queries to make your application functional, however we require some specific cases to be implemented and **described** in the assignment report. Focus on the more important queries and features of your application (i.e., there is no need to show how you insert documents in your various collections). Explain and motivate each query.

General guidelines for queries:

1. At least two of the five queries should deal with data from more than one table, i.e., you should use at least two multirelation queries
2. You should make use of SQL JOIN
3. You should make use of Aggregation and/or Grouping
4. You should make use of at least two of the following:
    a. Triggers
    b. Procedures
    c. Functions

5. Implementation

Write a program that implements your Idea in Task 1 with the design and queries from Task 2-4. You are of course allowed to introduce more queries. You should use Python for implementation. You are free to use web based frameworks such as Flask. However, you are not allowed to use Object Relational Mapping (ORM). You need to explicitly write your queries.

6. Supplemental video

Make a video (at most 10 minutes) demonstrating how your implementation works. It should show how it runs your queries and the results they produce (focus on guideline queries from Task 4). You should upload the video somewhere (ex. youtube or vimeo), where it is accessible to us and reference it in the project report.

Submission

Your submission should include solutions to all assignments above. Submit a report in **PDF format** on Canvas. Please submit the source code and database dump at an accessible Git or GitLab repository and include the link in your report. You can draw the E/R diagrams by hand and submit scanned versions or photos (as long as they are readable). The report should have at least 4 pages and should not be longer than 8 including Tables and Figures.

Finally, your report should have an appendix with the changelog to document who has done what on the project. Check the report template for more information.

NOTE: Follow the submission and implementation guidelines carefully. If they do not adhere to the requirements presented in this document you will **get a failing grade!**