SOEN 363: Data Systems for Software Engineers

Database Project - Phase I, Fall 2024

October 29, 2024

Date posted: Tuesday, October 28th, 2024.

Date due: Friday, November 22nd, 2024, by 23:59.

Weight: 10% of the overall grade.

Group Project. You must work strictly within your group.

Overview

This document outlines the phase I of the database project, in which you implement a relational database. You have already submitted your project proposal and outlined the scope and the topic of the database project.

Phase I addresses the relational database. Note that the output of phase I will be used to populate the no-SQL database in Phase II. While you submit the phase I of your project separately, during phase II, you may enhance and refine your database to address unforeseen design issues. The final presentation will cover both phases.

In this phase, you will design, implement, and populate the relational database from at least two various sources, by consuming public APIs [1]. See *Implementation Platform* and *Non-Functional Requirements* sections in the Project Outline document (P00).

Design Requirements

While you follow best design practices in the implementing your database (i.e. proper use of keys, indexes, integrity constraints, etc.) you must explicitly demonstrate:

- How you provide the link between the two data sources. Note that the data that you are collecting may not necessarily use same keys / identifiers.
- At least one IS-A relationship.
- At least one example of a weak entity.
- An example of a complex referential integrity (i.e. using assertions or triggers).
- Examples of a hard-coded views that filters some rows and columns, based on the user access rights (i.e. a full access user may see all columns while a low-key user may only see certain columns and for a subset of data.

In addition to the above, demonstrate use of domains and types[1]. A couple of examples per item would be sufficient.
Make sure that no real domain data is used as internal keys (primary / foreign)[2].

Query Implementation

You need to demonstrate the following query types:

- Basic select with simple where clause.
- Basic select with simple group by clause (with and without having clause).
- A simple join query as well as its equivalent implementation using cartesian product and where clause.
- A few queries to demonstrate various join types on the same tables: inner vs. outer (left and right) vs. full join. Use of null values in the database to show the differences is required.
- A few queries to demonstrate use of Null values for undefined / non-applicable.

[1]Note that MySQL and MariaDB do not support user-defined types; in which case you may demonstrate how you address it. There is a feature request for future support in MariaDB

[2]For instance, in case of IMDB movies, the imdbid should not be used as an internal key

- A couple of examples to demonstrate correlated queries.
- One example per set operations: intersect, union, and difference vs. their equivalences without using set operations.
- An example of a view that has a hard-coded criteria, by which the content of the view may change upon changing the hard-coded value (see L09 slide 24).
- Two queries that demonstrate the overlap and covering constraints.
- Two implementations of the *division* operator using a) a regular nested query using NOT IN and b) a correlated nested query using NOT EXISTS and EXCEPT (See [2])[3].

What to submit

Submit your code (data creation / population) as well as the DDL and DML queries along with the Data Model of your database. Note that some databases allow you with automatic generation of DDL and DML from the exisitng data. While this is accepted, you are required to remove unnecessary / conditional imports from your script. Make sure your script is clean!

Final Report and Presentation

Your final submission will include a report document that provides an overview of your system as well the data model and the approach / challenges you faces in populating the data. A presentation session will be arrange so that you demonstrate your project for peer review. Due to the class slze, the presentations will be offline. However, few teams may have a chance to present live. Details will be discussed in the class.

[3]If your database domain does not address this, create a simple data with a few entries (at least 4) and demonstrate an example of a SQL division

References

1. https://github.com/public-apis/public-apis
2. https://www.geeksforgeeks.org/sql-division/
3. https://www.postgresql.org/
4. https://www.mysql.com/
5. https://mariadb.com/
6. https://en.wikipedia.org/wiki/SQL_Server_Express