SOEN 363: Data Systems for Software Engineers

Database Project - Phase I, Winter 2024

March 4, 2024

Date posted: Monday, March 4th, 2024.

Date due: Wednesday, April 3rd, 2024, by 23:59.

Weight: 8(+2)% of the overall grade.

Group Project. You must work strictly within your group.

Project Outline

This document outlines the phase I of the database project, in which you implement a relational database. Using a topic of your choice, you want to implement a relational database whose data is populated from at least two different public sources. In this project, you need to design, implement, and populate your database using the data that are available via public APIs.

Implementation Platform

You may use MySQL or PostgreSQL as the database implementation platform and use any programming language of your choice to populate the data into the database.

Database Size

You database must contain a large data set. While this may be subjective, a database of size 300MB and above may be a good example of a large data set.

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 collection may not necessarily have the same key.
- at least one IS-A relationship.
- at least one example of a weak entity.
- an example of a an complex referential integrity using triggers.

In addition to the above, demonstrate use of views, and domains and types. A couple of examples per item would be sufficient.
Make sure that no real domain data is used as internal keys (primary / foreign).
Example: imdbId for movies.

NOTE: Use a topic other than movies (IMDB or any other similar data sources).

Query Implementation

You need to provide 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 select query using cartesian product and where clause vs. a join query using on.
- 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 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 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 [4]).
- Provide queries that demonstrates the overlap and covering constraints.

What to submit

Submit your code (data create / population) as well as the DDL and DML queries along with the Data Model of your database.

In your submission 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.

Demo

A presentation session will be arrange so that you demonstrate your project for peer review.

References

1. `https://github.com/public-apis/public-apis`
2. `https://www.mysql.com/`
3. `https://www.postgresql.org/`
4. `https://www.geeksforgeeks.org/sql-division/`