WESTERN UNIVERSITY
FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

SE3309a – Database Management Systems

Fall 2021

Assignment 3

October 27, 2021

Deadline:
November 26, 2021

Please read the whole assignment first.

1. Install the MySQL server and MySQL Workbench following the Installing MySQL instructions. Login to the MySQL server, remove the anonymous user, and change the root password. Use the Getting Started with MySQL to see how to execute these steps. Try some simple commands, such as creating table (relation) and inserting/ retrieving data. No credit will be given for this task, but it is important that you try logging in as soon as possible.
2. Write an SQL database schema for your application, using the CREATE TABLE commands described in Getting Started with MySQL. Use referential constraints and consider using other constraints, indexes, and triggers. Hand in a file with the commands you used to create your database schema. Show the response of the SQL Editor to requests that describe each of your relation schemas. For example, to see the schema for relation Client type DESCRIBE Client;
3. Execute **three** different types of INSERT commands to insert tuples into one of your relations. Show the response of the SQL Editor and the relation that results when you issue a SELECT *command. Again, the information on how to do this step is in Getting Started with MySQL. The INSERT commands should be interesting so that we do not get the same INSERT statement with different values (they will be considered as the same). For example, make use of the INSERT ... SELECT to create these interesting INSERTs.
4. Develop a substantial amount of data for your database and load them into your relations. If you are using real data for your application, your program will need to transform the data into files of records conforming to your application schema. Alternatively, you can write a program to fabricate data: your program will generate either random or non-random (e.g., sequential) records conforming to your schema. Note that it is both fine and expected for your data values--strings especially--to be meaningless. The point of generating large amounts of data is so that you can experiment with a database of realistic size, rather than a "toy" database. The data you generate and load should be on the order of:
    - At least two relations with a few thousand tuples each.
    - At least one additional relation with hundreds of tuples.

   When writing a program to fabricate data, there are two important points to keep in mind:

   a) Make sure not to generate duplicated values for the keys in your relations.

   b) Your application almost certainly includes relations that are expected to join with others. For example, you may have a Rent relation with attribute clientNo that is expected to join with attribute clientNo in the Client relation. In generating data, be sure to generate values that actually do join--otherwise all of your interesting queries will have empty results! One way to guarantee joinability is to generate the values in one relation, then use the generated values in this relation to select joining values for the other relation. For example, you could generate client numbers first (either sequentially or randomly) in the Client relation, then use these numbers to fill in the clientNo values in the Rent relation.

Turn in a small sample of the records generated for each relation (5 or so records per relation).

5. Write **seven** queries on your application database, using the SELECT-FROM-WHERE construct of SQL. To receive full credit, all but one of your queries should exhibit some interesting feature of SQL: queries over more than one relation, subqueries, exists or group by for example. The queries should be diverse (use different SQL features); the queries that differ only in table and attribute names will be considered the same. If you discover that most or all of your "interesting" queries return an empty answer on your large database, check whether you followed the instructions for generating data values that join properly. You may need to modify your data generator accordingly. Turn in a copy of all of your SQL queries and the response of the system to convince us that your commands run successfully. Please do not turn in query results that are thousands (or hundreds of thousands) of lines long!

6. Write **three** data modification commands on your application database. Most of these commands should be "interesting", in the sense that they involve some complex feature, such as inserting the result of a query, updating several tuples at once, or deleting a set of tuples that is more than one but less than all the tuples in a relation. Hand in your modification commands and the response of the system.

7. Create **three** views on top of your database schema. Hand in your CREATE VIEW statements and the response of the system. Also, show a query involving each view and the system response (but truncate the response if there are more than a few tuples produced). Finally, show a script of what happens when you try to modify your view, say by inserting a new tuple into it. Are your views updatable? Tell why or why not? (Essentially, a view is updatable if it is a selection on one base table.)
8. Was there an SQL clause that you learned in class (from ISO standard) that you were not able to use in MySQL because the dialect has not implemented it?

HOW TO HAND THE ASSIGNMENT IN

- Upload a **single document on OWL**, **AND**
- Upload all your code to the GitHub repository:
  - Create a private repository and add your Lab section TA to your repository. TAs GitHub IDs:
    - Lab section 002: Santiago (sgomezro, sgomezro@uwo.ca)
    - Lab section 003: Dagimawi (Dagimawi-Eneyew, deneyew@uwo.ca)
    - Lab section 004: Ruiqi (LiquidMethane, rtian25@uwo.ca)
  - In the repository, create a folder `assignment3`;
  - Inside the `assignment3` folder, create folders `sql` and `src`;
  - In the `sql` folder, you must have the following files:
    - `ex2.sql`: create table statements from item 2;
    - `ex3.sql`: insert statements from item 3;
    - `ex5.sql`: select statements from item 5;
    - `ex6.sql`: data modification statements from item 6;
    - `ex7.sql`: create view statements from item 7;
  - In the `src` folder, you must upload the source code that you created to load data on your database and / or the code you used to fabricate data.