School of Computer Science, McGill University

COMP-421B Database Systems, Winter 2012

Programming Project 2: Creating your Database

Due date: February 19, 23:30

In this assignment you are going to refine your schema, create your database using DB2 or PostgreSQL and write and run some SQL queries. We have created a UNIX and a DB2/PostgreSQL group account for each registered group. Information on group names and passwords can be found on myCourses

It is suggested that for all your deliverables you keep files with all statements. In particular, it would be good to generate a script that populates your database so that you can delete and recreate your database as needed. This will allow you to experiment with queries that potentially delete large parts of your database, and be able to get back to the original database.

Please note, that for this and the next two project assignments you might frequently need to work with the DB2/PostgreSQL online information (link from myCourses) in order to figure out how things work in the database system. It is an essential part of the project to learn how to find the needed information in the online help. To make life a bit easier at the beginning, some extra links/documents are given on myCourses. They describe the most essential things.

In this assignment, you can use any of the interactive user interfaces offered by DB2/PostgreSQL. CLP/psql are described on myCourses.

Note that you can use a different database system than the ones offered on the SOCS machines. You will have to provide printouts of the queries you write as well as the output. You have to figure out how to do so with the database system of your choice. In later deliverables you have to work with JDBC, indices, and stored procedures. Make sure that the database you choose supports these concepts. If the database system does not support the SQL standard for triggers and stored procedures, you will have to add extra explanation to your deliverable.

The tasks described in this deliverable provide 95 (of 100) points. The extra 5 points can be achieved by using SQL features that were not discussed in class, or by adding extra queries/updates/constraints/triggers that show something that your other queries/updates/constraints/triggers do not show.

1 Assignment (Please turn in one solution per team)

1. (0 Pts) Please attach a copy of your relational schema of Assignment #1. If you have modified your schema because of feedback (or any other reason), please hand in the modified schema instead; the new design will not be graded but will be used as a basis for the future assignments.
2. (25 Pts) Write an SQL database schema for the relational schema you have designed using the `CREATE TABLE` command and enter it in the DB2 database. Choose suitable data types for your attributes. Indicate primary keys, foreign keys or any other integrity constraints that you can express with the commands learnt. Indicate the constraints you cannot express. The Online Information contains detailed information about data types, and the *CREATE TABLE* statement.
   Turn in your `CREATE TABLE` statements. Furthemore, (i) for DB2 use command line processor command `DESCRIBE tablename` (ii) for PostgreSQL use
   `d table name` (prints the description of the relation on the screen) for each of your relations, print the result and turn it in.
3. (5 Pts) Execute five INSERT commands to insert tuples into one of your relations.

Turn in your `INSERT` statements. Furthermore, print and turn in the response of CPL/psql when you type the `INSERT` commands. Print and turn in the result when you issue a `SELECT * FROM relationname` command.

4. (5 Pts) Insert in all your tables enough meaningful information so that the queries that you create provide meaningful results. The results of the following queries that you develop should have a reasonable number of results so that we can be convinced that your queries are correct (maybe 5-10 tuples).

   For each table show the output, truncated to the first 5-10 tuples, that are returned when you issue a `SELECT * FROM relationname` command.

5. (25 Pts) Write five queries on your project database, using the select-from-where construct of SQL. The queries should be typical queries of the application domain. To receive full credit, all but perhaps one of your queries should exhibit some interesting feature of SQL: queries over more than one relation, subqueries, aggregations, grouping etc.

   Turn in a description of all the relations that you use in your queries (e.g., the original create statements or printouts from the SQL "describe table yourname" function), a description of what each of your queries is supposed to do, the SQL statement of each query, along with a script illustrating their execution (for example the screenshot when you execute the query). Your script should be sufficient to convince us that your commands run successfully. Please do not, however, turn in query results that are more than 50 lines long

6. (16 pts.) Write four data modification commands for your application. 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.

   Turn in a description of all the relations that you use in your modifications but are not described so far. Provide a short description of what each of your statements is supposed to do, the SQL statements themselvs and a script or screenshot that shows your modification commands running in a convincing fashion.

7. (10 pts.) Create two views on top of your database schema.

   Turn in an informal description what data each of the views represents, show 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).

8. (9 pts.) Add two CHECK constraints to relations of your database schema.

   Turn in the revised schema, its successful declaration, and the response of DB2 to modifications (insert/update) that violate the constraints.