School of Computer Science, McGill University

COMP-421 Database Systems, Winter 2022

Project 2: Database Creation

Due Date Feb 24, 12:00 noon

In this assignment you are going to refine your schema, create your database using DB2 and write and run some SQL queries. I also recommend you go over the requirements that were listed in project 1, as we would be making some references back to those in this description.

Please note, that for this and the next project deliverable you might frequently need to work with the DB2 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, while you can use either the `db2` command line client or the IntelliJ IDEA IDE when you start development (some are shortly introduced in myCourses and/or demonstrated during lectures.), you must use the `db2` command line utility (through the template scripts given to you) to run your finalized project files and submit the output. A template, `project2.tar.gz` is included with this assignment to help with this.

WARNING !! Do not insert more than 50 records per table. Violators will be penalized. But make sure that you insert sufficient records into the tables such that your queries produce at least a couple of records for the questions below.

Your project 2 work should be based on your project 1 model (in addition to any fixes you have to do to it to make it work). Changing the model to make your queries easier is not allowed. I.e., any changes to the model should be because you cannot store the relevant data or introduces some issues (like inconsistent information). A solution outline for project 1 will be provided later but use it sparingly as a reference only if you are "stuck" in which case you can adapt from that outline to fix that specific part of your problem.

TAs will be manually grading this assignment work.

The Assignment

Some of the solutions to the below questions have to be provided in a document **project2.pdf** whereas others would be independent files of their own (described next to each question).

You are given a template, `project2.tar.gz` that you **must** use to execute and turn in these extra submissions.

Where required to produce a screenshot of a command execution, it must be executed from the DB2 command line from an ssh terminal logged into the `winter2022-comp421.cs.mcgill.ca` **server.** Make sure that the screenshots include both the command/SQL and the corresponding outputs, even when you are providing the SQL separately in the submission. We need to know that the SQL you provided us is what you actually executed to get the output you got. Where necessary, you may truncate part of the output to fit the screen (try to enlarge your ssh terminal when you can). It is important that ALL of the SQL is visible in the screenshot and at least a good part of the results (if all of the result does not fit in the screen).

1. **(0 Points)** Make sure you are using the DB2 database provided in the winter2022-comp421 server for your project (we have mechanisms to verify this). Not doing this and using another database (including PostgreSQL), using databases installed in your own personal computer, etc., will result in **5 points penalty**.

2. **(0 Points)** Include the relational model that you are using for this phase of the project - even if you have not made any changes from project 1. Include this in the document **project2.pdf** under the section "**Relational Model**". Not doing this will result in **1 point penalty**! Turning in your modified ER model is not required, but you are free to do so.

3. **(5 Points)** Write a SQL database schema for the relational schema you have designed using the `CREATE TABLE` SQL statements and enter them (create) in the database. Choose suitable data types for your attributes (You can read ahead in this project description to get a general idea as to what some of the queries would be like

and also review Project 1 description). Indicate primary keys, foreign keys or any other integrity constraints that you can express with the commands learnt. The Online Information contains detailed information about data types, and the `CREATE TABLE` statement.

Once you have figured out the DDLs, you can write them into the `createtbl.sql` file and have it executed using the `createtbl.sh`. Verify that all the tables got created correctly in the log file and turn in the `createtbl.log` produced by the script along with the `createtbl.sql` file.

Next, write the corresponding `DROP TABLE` statements into the `droptbl.sql` file and have it executed using the `droptbl.sh`. Verify that all the tables got dropped correctly in the log file and turn in the `droptbl.log` produced by the script along with the `droptbl.sql` file.

Once properly written, the above scripts will be very handy for you to "reset" the database quickly throughout your development time if you make mistakes or want to make changes, etc.

Under a section "**Pending constraints**" in the **project2.pdf**, indicate any constraints (use your ER and relational translation notes to check) that you cannot express in your database implementation.

4. **(5 Points)** Next we will load data (you should have the tables created in the database for this). Make sure that you have at least 5 (unless limited by the actual possible data values of its domain) records in each table that you have created. **Do not insert more than 50 records in any given table!**

When generating larger number of records, you can use clever techniques like using shell scripts, a spreadsheet (like Excel, google, etc.) and writing some formulae to convert a table of values in a spreadsheet into corresponding SQL insert statements. This can minimize your manual effort and chance of making mistakes. There are also some websites that help you generate random data (investigate and explore!).

Once you have figured out your insert statements, write them into the sql file `loaddata.sql` and execute them using the `loaddata.sh`. Verify the log file `loaddata.log`. Turn in both the SQL file and the log file.

If properly utilized, this script, along with the scripts from the previous question can make your development "birth pangs" mild, where you have to constantly keep changing structures and data till you "stabilize" your system.

I recommend that you read forward to the rest of the questions to help you get an understanding of the kind of data that you will need to write some of the SQL queries that are being asked and plan accordingly.

5. **(10 Points)** Write queries to retrieve the following information:

   (a) Write a SQL query that will list all the appointments for the midwife *Marion Girard* for the week *March 21 - March 25* of this year (2022). The output must be the date and time of the appointment as well as the health care card, name and phone number of the mother. You can assume that there is only one midwife with this particular name.

   (b) Write a SQL query that will list the results of all the *blood iron* tests that was performed on (Mother) *Victoria Gutierrez* for her *second* pregnancy. The output should have lab date and the result (the latter you can assume as a simple VARCHAR field with some text info - keep it a small field so your screen does not get cluttered).

   (c) Write a SQL query that will list the names of each birthing center/community clinic as well as the number of pregnancies that are due for the month of *July* this year (2022). For this purpose you will count a pregnancy towards a facility if the primary midwife is from that facility (and not based on the final birthing location). You should also use the "final agreed up on" due date if that is present, in order to figure out if the pregnancy will fall into a specific month. Otherwise (if that information is not yet available), use the expected time frame for birth (year-month) that was provided initially.

   (d) Write a SQL query that will list the health care card, name and phone number of the mothers who is currently pregnant and have not yet given birth and is under the care of a midwife employed by *Lac-Saint-Louis*. You can assume that there is only one facility by that name.

   (e) Write a SQL query that will list the health care card, name of mothers who have had more than one baby in a single pregnancy. List them only once even if they had multiple pregnancies where they had more than one baby. Count the babies whether they are born or not (yet).

Include each SQL as plain text under a section "**SQL Queries**" in your **project2.pdf**. Also include the screenshot of the query being executed in db2 using the command line utility that shows its output. Your screenshot may truncate off some part of the output if it is not possible to fit all of it into a single screen but

it should include ALL of the SQL. In the project document, include the question numbers for each query so that your TA knows which question the solution screen shots belong to.

6. **(3 Points)** Create a view `midwifeinfo` that shows the practitioner id, name, phone and email of all the midwives, the name of the clinic / birthing center that employs them, and the address of the facility. Include the responses to this question under the section "**Midwife Information**" in your **project2.pdf**.

   (a) Include the view definition SQL (as plain text).

   (b) Screenshot of the view creation being a success.

   (c) Screenshot of a SQL query that selects everything from the view, truncated to just 5 records.

   (d) Screenshot of a SQL query on the view that limits the previous output to only the midwives working at the facility whose name is *Lac-Saint-Louis*. You can assume there is only one facility with that name. Truncate the output to just 5 records.

   (e) Now try inserting a record into the view (practitioner id, name, phone, etc., the attributes that are mentioned in the view's output) that has valid domain values for these attributes (e.g. a new midwife, but an existing facility). Observe what happens. Take a screenshot and turn in that along with the explanation of why this happened.

   Note:- If the queries in this question produces less than 5 records it is fine. But it should produce at least a couple of records. Also make sure that the outputs are different between the select everything (6c) and select specific facility (6d).

7. **(2 Points)** Add a `CHECK` constraint to your respective table to ensure that the lab date associated with a test cannot be a date before the date the test was prescribed. Attach a screenshot of running this command and the response from the database as well as an instance where you try to insert a record that violates this constraint and the database throwing an error. (Use the DB2 command line to demonstrate this) - Include this screen shot in **project2.pdf** under the section "**Check Constraints**".

Additional Restrictions

- For your SQL queries, you are only allowed to hard code the data explicitly given in the question. For example, in 5a, you can use *Marion Girard* (and of course the date) in your query. But you may not hard code the values of her practitioner id, or email, etc., to make it easy to write your query as this is not a data given to you in the question.

- You may reformat the data given in the question to fit your model. For example, if you stored the first name and last name of the midwives as different attributes, you can use *Girard* and *Marion* as two different attributes in your SQL. Similarly, if you used a number *2* instead of the text *second* in your model, this is also an acceptable accommodation for your SQL query.

- Other than where explicitly asked to do so, you should not be creating any additional tables, views, etc. (Questions 5 6). You may use the "derived tables" concept or the `WITH` clause, similar to assignment 1.

Files to Submit

Please use this as a check list to ensure that you do not miss uploading any required files!

Your submission will contain the following files:

1. **createtbl.sql**, **createtbl.log**, **droptbl.sql**, **droptbl.log** : These will contain your `CREATE TABLE` and `DROP TABLE` DDLs, the log files showing successful execution, etc.

2. **loaddata.sql** that contains the insert statements and the log file **loaddata.log** showing that they were inserted successfully.

3. **project2.pdf** this will contain your current relational model, any screen shots, etc., as indicated under various questions. Make sure you put them under the correct section headings so that TAs can match your solutions to the questions. Haphazard clutter in the document can result in point deductions.

These are the only file formats that you are allowed to use for submission!

You may submit these files separately or tar them. Resubmissions are allowed (please do not email me saying you do not know how to resubmit files). Please try to submit an entire set, so that TAs do not have to find your files across multiple submissions.

Please turn in your submission in mycourses under project 2. Submissions to wrong folders may not get graded. **You are responsible to download and verify that your submission is correct (not corrupted or incorrect file, etc).** There will be no accommodations if you decided to ignore doing due diligence (you will get a 0). Project grades also influences the pass/fail criteria of the course. Please review the course outline for any of these details.

Important Database Etiquette

Please remember that the server `winter2022-comp421.cs.mcgill.ca` and the databases installed them are meant to be used for the course work of COMP 421. Using these systems for other work (including other course work) is not allowed. You are sharing these resources with other classmates, so be mindful of its proper use.

Please go through the `README` file in MyCourses under "Database" → "Connecting to DB Servers" for comprehensive list of restrictions. Not following them may result in your id being disabled as stated in the course outline.

Questions ?

Please use **Ed** for any clarifications you need (`projects → p2`). You may also want to review the project 1 requirements to refresh some of the concepts and terminologies. Do not email the instructor or TAs as this leads to a lot of duplicate questions and responses (not an efficient system). Such emails will not receive any replies.

The only emails you are allowed to write is to your grading TA if you have questions on the feedback you received (as they are personal in nature) - do not post them in Ed, your grading TA may not see it.

Please check the pinned post "P2 general clarifications" in Ed before you post a new question. It might have been already addressed there, in which case we will not address it again.

Questions about general clarifications must be marked public (as other students will also benefit from this and may even have a valid response). TAs and Instructors upon their discretion may toggle any private posts into public mode for the benefit of the student population at large.

There will be specific office hours for the project that will be announced closer to the due date.

Extensions and Late submissions

- Remember, your submission is **due on Feb 24th, 12:00 noon**. There is no place for excuses.

- A maximum of 2 days of late submission is allowed with a penalty of 20% of the achieved grade per day (rounded up, even for a minute).

- Penalty waivers are granted only for medically documented emergencies and under any circumstances **will not be granted unless requested 24 hours before the due**. I expect you to be better organized and get the work done ahead and leave the last 24 hours only for one last final check.

- There is no "partial late penalty" concept, it is applied to your entire submission, and not just specific questions that you submitted late.