

# COMP-421/764 Database Systems, Winter 2021

## Project 2: Database Creation

Due Date Feb 25, 05:00pm

In this assignment you are going to refine your schema, create your database using DB2 and write and run some SQL queries.

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 any of the interactive user interfaces offered by DB2 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 100 records per table. Violators can 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). 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 can 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 winter2021-comp421.cs.mcgill.ca server.** Make sure that the screenshots include both the command/SQL and the corresponding outputs. 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) 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”.
2. (5 Points) Write a SQL database schema for the relational schema you have designed using the `CREATE TABLE` commands 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). 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.

3. (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) records in each table that you have created. Do not insert more than 100 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.

4. (10 Points) Write queries to retrieve the following information:
  - (a) Write a SQL query that will list all the available (unassigned to any individual) slots in the vaccination location *Jewish General* on the day of March 20th. (The output depends on what your model captured as a slot - at a minimum it should cover the information of the day and time - this may involve getting information from other tables as well.)
  - (b) Write a SQL query to find the expiry date of the vaccine dose that was administered to *Jane Doe* on Feb 6th, 2021. (You can assume that there is no other person with that name and that this was the only dose of vaccine that person ever received).
  - (c) Write a SQL query to find the total number of people who were vaccinated in *Montreal* (defined as received vaccination from a location based on *Montreal*) on Feb 6th, 2021. You can assume that nobody receives two doses of vaccines on the same day.
  - (d) Write a SQL query that will list the name, phone number and insurance number of those individuals who got one dose of *Pfizer-BioNTech* vaccine before Feb 1, 2021 and have not yet received another dose of the same vaccine. (Assume people do not get vaccinated by two different brands).
  - (e) Write a SQL query that will list the number of people who have been given at least one dose of vaccine for each category (*Elderly*, *Teachers*, etc. - see the project 1 description for the complete list.) Your output need not show categories without any individuals, however you should have at least 1 category per priority in your output data (so make sure to include appropriate data in your tables). Remember to include category in the output.

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

5. (3 Points) Create a view `mtlnurses` that shows the license number and name of all the nurses, the name of the hospital that (currently) employs them, postal code and street address for nurses employed by the hospitals in the city of *Montreal* (Their actual employer and not the location where they are assigned for vaccination duty). Include the responses to this question under the section “Montreal Nurses” in your **project2.pdf**.
  - Include the view definition SQL (as plain text).
  - Screenshot of the view creation being a success.
  - Screenshot of a SQL query that selects everything from the view, truncated to just 5 records.

- Screenshot of a SQL query on the view that limits the previous output to only the nurses working in the hospital *Jewish General*. Truncate the output to just 5 records.
  - Now try inserting a record into the view (license number, name and a hospital name, etc.,) that has valid domain values for these attributes (e.g. a new nurse, but an existing hospital). Observe what happens. Take a screenshot and turn in that along with the explanation of why this happened.
6. (2 Points) Add a **CHECK** constraint to your respective table to ensure that the expiry date associated with a vaccine batch is past its manufacturing date. 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”.

## Files to Submit

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.

You may submit these files separately or tar them.

Please turn in your submission in mycourses under project 2. Only a maximum of 3 late days allowed (15% of the maximum allocated points deducted per day, rounded up). Project grades also influences the pass/fail criteria of the course. Please review the course outline for any of these details.

## Questions ?

Please use Piazza for any clarifications you need. Do not email the instructor or TAs as this leads to a lot of duplicate questions and responses (not an efficient system). Please check the pinned post “P2 general clarifications” before you post a new question. It might have been already addressed there, in which case we will not address it again.

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