

COSC 3P32 – Introduction to Database Systems

Winter 2020

Group Project, Part 2

Due Date for Report: 9th April at 3pm or the start of your demonstration, whichever comes earlier. There is NO LATE DATE.

This work accounts for 12% of your final grade and is worth a total of 120 marks.

In this part of your project, you will use PostgreSQL to build the database that you designed in part 1, and demonstrate different ways of interacting with the database.

Important information about your group account:

- If you have not yet submitted a group account form, you should do so as soon as possible. To do so, fill out the form at the following location: <http://www.cosc.brocku.ca/forms/grpacct> and then email it directly to the instructor, who will forward it to our system administrator for creation of the account.
- After the demonstration, your group account password will be reset so that the TA can access it for marking purposes. Therefore you should make sure that everything required to run your project is *in your group account prior to the start of your demonstration*. An electronic copy of your report should also be in the group account.

A. Implementation: 100 marks

Create tables and insert data

Use the DB schema obtained from part 1 to create the tables required for your application. If you received a poor mark for your design from part 1, or are unhappy with your design, then you should discuss it with the instructor and may wish to use as a starting point the ER diagram and list of relations provided by the instructor.

Input relevant data into your tables. You should input enough data that it is possible to verify any query you might have, and also any constraints that must hold.

Note: if you are using the `DATE` type and having difficulties then you can use an integer to represent the year, or a different format (e.g. break down into day, month, year). Similarly, you should investigate `VARCHAR` vs. `CHAR`.

As with any implementation of the SQL standard, some features will not be supported or will have some slight variation. For the list of unsupported features in PostgreSQL, see:

<https://www.postgresql.org/docs/10/unsupported-features-sql-standard.html>

Enforce constraints

You must ensure that all of the following constraints are enforced. Some of these constraints can only be enforced via check constraints and/or triggers (note that assertions are not available).

- All domain constraints, primary key constraints and foreign key constraints.
- Every droid of the same type has the same main skill.
- All vehicles of the same type have the same hyperspace capability.
- The level of a Jedi/Sith character must be “padawan”, “knight” or “master” (or null)
- Only Jedi/Sith can use lightsabers.

- Only Jedi/Sith can hold a position in the Jedi Council (political unit).
- For the positions held by characters in political units, the final year must be no earlier than the starting year.
- Similarly, in any mentorship, the final year must be no earlier than the starting year.
- Every political unit must have at least one character holding a position within it. This participation constraint is the most difficult of the set of constraints to be enforced. It will require a set of triggers for each type of change (insert / delete / update) that affects one of the tables involved in the constraint.

Implement functionality

Your project must be able to support insertions, deletions and updates for all data stored in the database while also ensuring that constraints are satisfied.

The project must also support all of the following **queries**:

- Given a character name, output all the basic information about that character, including home planet and (if applicable) Jedi/Sith-specific information or Droid-specific information.
- Given a weapon, output all information on that weapon, including the names of the characters that use that weapon (if any).
- Given a vehicle, output the names of the characters that use that vehicle (if any) and the battles in which the vehicle is used (if any).
- Given a political unit, output all information on mentorships approved by that political unit, i.e. character names and levels of each participant in the mentorship, along with the start and end dates of that mentorship.
- Given a planet name, output the coordinates of that planet, as well as the names of the characters who have that home planet.

Important note concerning queries: as this is a scaled-down version of the project normally assigned at this time of year, it is not possible to use a front end and so everything will be implemented exclusively in PostgreSQL. As a result, the following should be done for each query:

- Create a view that implements the query for all possible data, then in postgres write a simple query on the view for the given input.
- For example, for the last query listed above the view is defined for all possible planets, and the simple query to be written in postgres queries that view for a given planet.

B. Documentation: 10 marks

Prior to the demonstration, you must submit a report including all of the following:

- A cover page that includes the names and ID numbers of all members in your group,
- The database schema,
- A list of all data in your tables, and
- Report on division of labour within your group. Reminder: unless there are exceptional circumstances, all members of the same group will receive the same mark. If the instructor has been informed that there are indeed exceptional circumstances, then this report may be used to adjust relative marks within a group.

Be sure to spend adequate time on preparing the report as this is one of the main things the TA will look at to mark your project.

C. Demonstration: 10 marks

Your project must be demonstrated to a TA through Microsoft Teams, during a prescheduled time between noon and 3pm on 9th April. The TA who marked the first part of your project will contact the team leaders regarding how to arrange demonstration times. Presentation times will be allocated on a *first-come, first-served* basis.

At least one group member, and preferably all group members, must participate in the demonstration. The group member(s) present must be able to walk through the functionality of your application. If *no* group member is able to be present at *any* of these time slots then it is the responsibility of the group to make other arrangements with the TA.

To ensure that no changes are made after the due date regardless of which day your group has its demonstration, the passwords to all group accounts will be changed at 3pm on 9th April. In the case that your demonstration is on a different day, you will be granted access again for the duration of the demonstration only.

Your documentation is to be submitted electronically prior to the start of your demonstration, or by 3pm on 9th April, whichever comes first.

Documentation is to be submitted as follows:

1. Documentation must be submitted **using your group account**. It is to be submitted electronically as a PDF document. Under your group account, create a directory on Sandcastle containing the PDF document, and run the script submit3p32 from this directory. When running the script, you will be queried as to assignment number: the documentation is to be submitted as assignment 4. Note that your group account has no email attached to it, so you will not receive email confirmation. Further details on electronic submission can be found at the following location: <https://www.cosc.brocku.ca/help/esubmit>
2. Note that all department policies concerning academic misconduct still apply in the case of electronic submission.

At the demonstration, the TA will participate in the walkthrough of your application by viewing the data, providing appropriate input and asking questions. The TA may write additional queries, etc. at this time to test your application. Due to time constraints it will not be possible to fully test the entire application during the demonstration, but the TA/instructor will look for specific functionality. The entire application will be marked after the demonstration, by access to the group account as specified above.