 Due: 28^{th} March, by 3:00 pm

CMPSC 312
Database Systems
Spring 2022

Lab 3 Assignment:
Relational Data Modeling for Protein Data (following lab 2)
Submit deliverables through your assignment GitHib repository.
Place query document `writing/` directory

Objectives

To create an SQL build file for an extended database. To learn how to integrate more tables to a previously created database in SQLite3 to store downloaded data.

GitHub Starter Link

https://classroom.github.com/a/wU_hC_hJ

To use this link, please follow the steps below.

- Click on the link and accept the assignment
- Once the importing task has completed, click on the created assignment link which will take you to your newly created GitHub repository for this lab,
- Clone this repository (bearing your name) and work locally
- As you are working on your lab, you are to commit and push regularly. The commands are the following.
  - `git add -A`
  - `git commit -m ‘‘Your notes about commit here’’`
  - `git push`

Introduction

Often when you create a database, you will find it necessary to add extra tables to be able to enhance its use. In this lab, you will modify the SQL builder code from your previous lab (i.e., the file `proteinDB_build.txt`) to contain two additional tables, containing the downloaded protein data from `https://www.uniprot.org/` from searches for *m-protein* and *sars*. We note that these two additional tables will contain protein data which we believe to have connections to the original data from the original database.

Database Builder File

An empty database builder file is provided (see file `src/proteinDB2_build.txt` but will be empty and all code to create the four tables and to populate them will be left to you to complete.

Please note, you will not be using this command in the SQLite environment, this command is to be used at the TERMINAL (unix) prompt in your Docker container. The below command will be necessary to build the database from with a UNIX or Docker environment.

```
cat proteinDB2_build.txt | sqlite3 proteinDB2.sqlite3
```

Data Sets

Your database must be designed to hold the same protein data from the previous lab, in addition to two new tables which are listed below. All data for this lab will be taken from UniProt at `http://www.uniprot.org`. To obtain your additional data to be added to your original database, please open your browser and find the UniProt website and complete searches for *m-protein* and *sars*. Although you might decide to re-download new and updated data from *Uniprot* for your *s*-protein and *n*-protein tables, you may copy this data over the from your previous labs to populate their associated tables in your new database. If you have trouble using the search feature on the *Uniprot* website, you will find the links for the datasets that are to be contained in your SQLite database below.

Data References

Your database is to contain the data from the following sources. This implies that this database will have four tables in total.

- From last lab, *n-protein*: `https://www.uniprot.org/uniprot/?query=n-protein&sort=score`
- From last lab, *s-protein*: `https://www.uniprot.org/uniprot/?query=s-protein&sort=score`
- **New table** for *Sars*: `https://www.uniprot.org/uniprot/?query=sars&sort=score`
- **New table** for *m-protein*: `https://www.uniprot.org/uniprot/?query=m-protein&sort=score`

Please be sure to save these files into the `src/data/` directory and then to update the builder file's `.import` commands to import this data into the correct tables. Also note, it would be desirable to keep the same attributes i.e., header names) in each table to simplify your queries and to respond to the questions-in-blue below.

Downloading and Formatting

Please download the returned protein data from your searches at UniProt using the un-compressed, `tab-separated` options as shown in Figure 1. Please keep all your database building files in your GitHub classroom repository and not in the course *ClassDocs* repository.

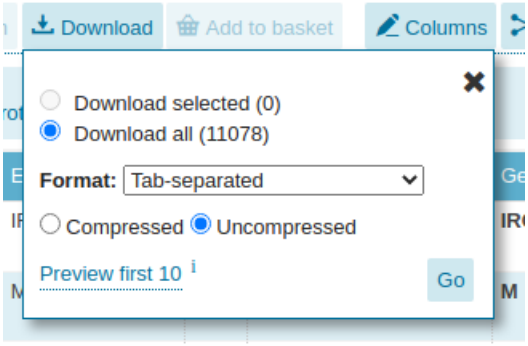

Figure 1: Download the data using the options for un-compressed and the tab-separated format. Please do not include these files in your submission repository as they are not necessary to evaluate your grade.

Trimming the File-Headers

As before, you will need to remove the first line of header information from each downloaded file. Your downloaded files have the column headers given on the first line which must be removed before you can use them to build your database. To do this, load each file in a text editor such as `Atom` and remove the top line of the file. This line will contain the terms; *Entry*, *Entry name*, *Protein names*, and etc., and were added in the file by Uniprot as column headers. Save your work as a text file.

Querying the Database Tables

Now that you have built a successful database, please answer the following questions-in-blue. When grading, the instructor will often be concentrating on the structure of your query, since the results may change depending on the age of the data. Note, to access your database with SQLite3, you will use the bellow command at your TERMINAL prompt.

```
sqlite3 proteinDB2.sqlite3
```

The following questions assume that the names of the tables are arranged in the following ways.

- *m*-protein: mprot
- *n*-protein: nprot
- *Sars* proteins: sars
- *s*-protein: sprot

Questions in Blue

Note, when asked to give your results below, you only need to give about five lines of the query result. To request a query where only five lines are given, modify your query similar to the following using the `LIMIT` n to display only n lines. Note: showing only n lines will ignore the rest of the results from your query.

```
SELECT <attributeName> FROM <tableName> WHERE <constraints> LIMIT 5;
```

1. In your own words, write at least two question(s) that the following queries answer;
   - select count(sa.protID) from nprot n, sars sa where sa.protID == n.protID;
   - select count(distinct(sa.protID)) from nprot n, sars sa, mprot m where sa.protID == n.protID AND sa.protID == m.protID;
2. Please give an obvious reason why the two queries below have the same output.
   - select count(distinct(s.protID)) from sars s;
   - select count(s.protID) from sars s;
3. Write a query to list all distinct *Organisms* which are common to both the `sars` and `nprot` tables.
4. Write the query to count all distinct *Organisms* which are common to all tables: `sars`, `nprot`, `mprot` and `sprot`.

5. Write a query to find out how many distinct protIDs are there in the `sars` where the organism == "Mus musculus (Mouse)"?
6. Write a query to find out the average length of the proteins in the `sars` table for the organism "Mus musculus (Mouse)"
7. Write a query to find out the largest length of the proteins in the `sprot` table for the organism "Mus musculus (Mouse)"
8. Write a query to list the protIDs and their associated organisms from the `sars` table where the length of the protein is equal to 240.
9. Write a query to list the protIDs and their associated organisms from the `nprot` table where the length of the protein is equal to 220.
10. Running queries can help you find patterns that you may not expect to find. What natural trend(s) did you notice from the result of your two previous queries above? If you saw nothing remarkable, then explain that you saw nothing unexpected.
11. Write a question of your own that uses all four tables to answer, then provide the query to respond to your question involving the four tables.

Summary of the Required Deliverables

Please submit your work by pushing it to your GitHub Classroom repository.

1. **Query and Results document**: You will modify the file `writing/queries.md` to respond to the questions-in-blue, above.
2. **Database-building file**: You will submit your edited build file (`src/proteinDB2_build.txt`) to be used to build your database from your data files.
3. **Data files:**: Please submit all your data files in `src/data/`

In adherence to the Honor Code, students should complete this assignment on an individual basis. While it is appropriate for students in this class to have high-level conversations about the assignment, it is necessary to distinguish carefully between the student who discusses the principles underlying a problem with others and the student who produces assignments that are identical to, or merely variations on, someone else's work. Deliverables that are nearly identical to the work of others will be taken as evidence of violating Allegheny College's Honor Code.