CMPSC 312
Database Systems
Spring 2019

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

Objectives

To learn how to add more data to your database (from lab 2) in SQlite3. You will also learn some important skills for writing queries.

GitHub Starter Link

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

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

In this laboratory assignment, we will use relational data modeling to obtain hands-on experience with the SQL programming language. We will continue the work with the database that we constructed during Lab 2 but we will add more data to the database to have three tables to query.

You will be working with the protein data which is associated with Parkinson's disease, Apoptosis, and now, **Alzheimer's** disease. The link for downloading protein data which is associated to Alzheimer's disease is shown in Figure 1 and can be found at the following link.

`http://www.uniprot.org/uniprot/?query=alzheimer&sort=score`

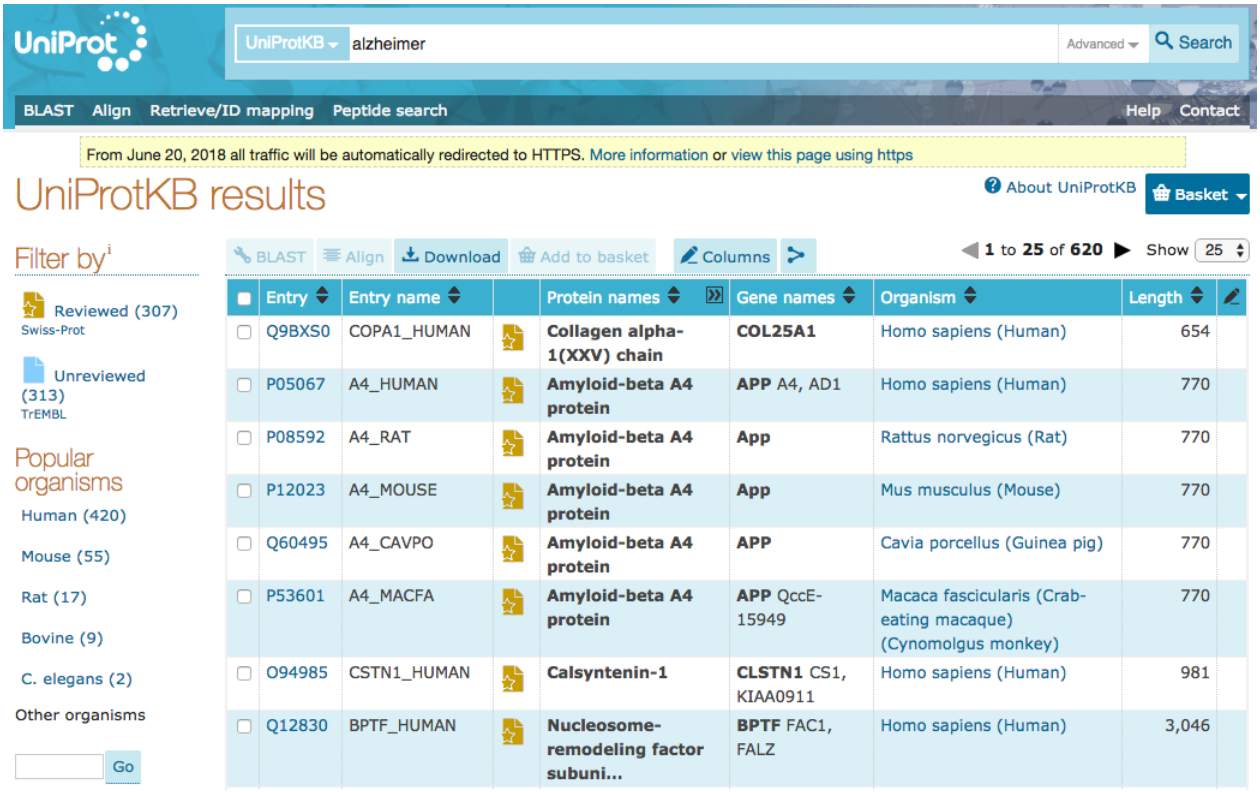

Figure 1: Results from searching *Alzheimer*. Reference `http://www.uniprot.org/uniprot/?query=alzheimer&sort=score`.

Your Alzheimer's data will be downloaded from UniProt using the compressed, `tab-separated` options as shown in Figure 2. **PLEASE DO NOT PLACE THESE FILES (OR YOUR COMPLETED DATABASE) IN YOUR SUBMISSION REPOSITORY; I do not want these files and they are not necessary to evaluate your grade.**

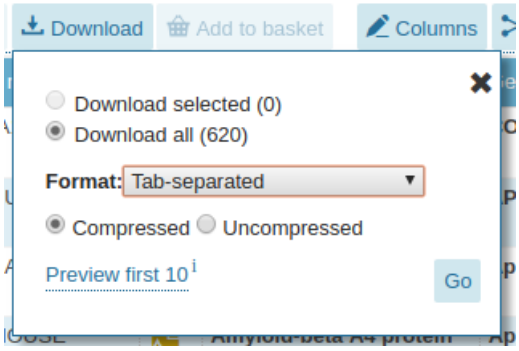

Figure 2: Download the data using the options for compressed and the tab-separated format. YOur data size is likely to vary. Please do not include these files in your submission repository which will come back to me. I do not need these files to evaluate your grade. Instead, place these files in a working directory outside of your submission repository. Use `mkdir` to create your directory for this work.

Update Your Table

To access the data files needed for this assignment, you must go to

After arriving at UniProt's data resource for proteins which have some connection to Alzheimer's disease using the above link, you should see a table similar to that of Figure 1. Note, your screen shot will likely be different in some way from this one due to the daily changes of the database itself.

Click on the "Download" button to save the Alzheimer's data set as a "Tab-separated", compressed file, as shown in Figure 2. After the file is downloaded, you should click on it on your Ubuntu desktop to decompress the file. As before, use Vim (or another text editor) to remove the top column-header information. Remember, here you are removing the names of the headers supplied by UniProt to the data you just downloaded and not the data itself! You previously used this header information to build the tables of your database.

Data References

Your database is to contain the data from the following sources.

- `http://www.uniprot.org/uniprot/?query=alzheimer&sort=score`
- `http://www.uniprot.org/uniprot/?query=parkinson&sort=score`

- `http://www.uniprot.org/uniprot/?query=apoptosis&sort=score`

Save Your Data and Build Tables From Modified Code

Place your data in your working lab directory (i.e., `myDB/data`, outside of your course directories). We note that the data files from last week should be already found in this same directory (Apoptosis and Parkinson's). You will rebuild your database in that directory from last week's code which you will modify to create a new table `Alz` to contain the protein data associated to Alzheimer's disease, according to UniProt.

Use the code given in your class repositories (the "sandbox" directories), as well as, the template code shown in Figure 3 to begin modifying your `CREATE TABLE` code to build your table for the Alzheimers protein. Note: once your tables have been built, you will use the `.separator` and `.import` keywords in SQL to import the data from the downloaded files to populate your database. The code to use is given below.

```
drop table Alz;
create table Alz(
        protID VARCHAR NOT NULL PRIMARY KEY,
        entryName ... ,
        Status ... ,
        ProteinNames ... ,
        GeneNames ... ,
        Organism ... ,
        Length ...
);
```

Figure 3: **This table code is not complete.** Complete the code to create your new table: `Alz`. Remember, you will need a line of code to create the attributes of your tables which contain each member of the row's tuple from the data. It is recommended that you use the following names for attributes to prepare your tables: `protID, entryName, status, proteinNames, geneNames, Organism`, and `Length`.

Re-Building Your Database From Our Last Lab

Modify your script code for creating tables from your previous database to be able to add the new data. For this, you will have to write the code for a new table `Alz`. Your code will look very similar to the code of Figure 3.

Use code as shown in class to create the tables and to load all your data from a text file. Place all database creation and data import commands neatly in a text file called `BD-Build_2.txt` which you will used to create and reproduce your database as needed. This is to be done exactly as we have build our databases in class. Remember, the command to build the base is shown below.

```
.separator "\t"

/* find the file and load it into sqlite3 which will create the database.*/
.import data/uniprot-apoptosis.tab Apop
.import data/uniprot-parkinson.tab Park
.import data/uniprot-alzheimer.tab Alz


/* cat <thisFile.txt> | sqlite3 ProtDB_2.sqlite3 */
```

Figure 4: Be sure to place your data files in the directory `data/`. If your files are found in another directory, the new directory must be reflected in the code above.

Questions Over Your DataBase

Please answer each of the following questions. Be sure to include your query programming with your answer.

1. Write a query to display the distinct `protID`'s that are common to all three tables.
2. Write a single single query to display the distinct `organism`'s that are common to all three tables.
3. Write a single query to display the `organisms` and the associated `geneNames` of all common `ProtID` to the three tables. For this task, use the `Alz` table to lead your query.
4. What information do the following queries provide? What does the `geneNames` attribute have to do with the queries?
   (a) `SELECT Apop.protID, Apop.geneNames FROM Apop WHERE Apop.geneNames != "";`
   (b) `SELECT count(Apop.geneNames) FROM apop WHERE Apop.geneNames = "";`
   (c) `SELECT Apop.protID, Apop.geneNames FROM Apop WHERE Apop.geneNames LIKE "A%";`
5. Explain what information is being return by the following query:
   ```
   SELECT DISTINCT(Organism) FROM apop
   INTERSECT
   SELECT DISTINCT(Organism) FROM park
   INTERSECT
   SELECT DISTINCT(Organism) FROM Alz ORDER BY Organism;
   ```
6. Explain what information is returned by the following query: `SELECT distinct(Apop.organism) from Apop LEFT JOIN Alz on Apop.organism != Alz.organism;`
7. Explain what information is returned by the following query and how the query works:
   `SELECT "Alz:  " || COUNT(DISTINCT(Alz.protID))`

```
FROM
Alz UNION SELECT "Park:  " || COUNT(DISTINCT(Park.protID))
FROM
Park UNION SELECT "Apop:  " || COUNT(DISTINCT(Apop.protID))
FROM Apop;
```

8. You may have noticed that there was a large difference in the number of proteins in the Alzheimer's table, when compared to the numbers of the same attribute in the other tables. In your opinion, why do you think this difference in counts exists?
9. How many *protID* can be found in the `Alz`, `Park` and `Alz` tables which are associated to the organism "Homo sapiens (Human)"?
10. In your opinion, why do you suppose that the numbers of proteins returned by each of the above queries were so varied?

Summary of the Required Deliverables

This assignment invites you to submit an electronic version of the following deliverables through Bitbucket:

1. **Query and Results document**: You will modify the file `writing/queries.md` to reflect your query code and results for each of the ten questions in blue, above.
2. **Database-building file**: You will edit the file `src/proteinBase-build.txt` to be used to build your database from data files.
3. Please consider **not** pushing your data files to GitHub as they will take up a lot of space. The instructor will not use them to grade your work. Please note, that this implies that you will have to make a build directory outside of your repository.
4. **Please do not forget to push the above two files to submit your work. The instructions for this are above.**

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.