**CMPSC 380**
**Principles of Database Systems**
**Fall 2016**

**Laboratory Assignment Two: Relational Data Modeling**

## Introduction

In this laboratory assignment, we will use a relational data modeling and obtain a hands-on experience with the SQL programming language. SQL is a declarative language in which you specify the data you want in terms of its properties. This assignment focuses on the SELECT subset of SQL, which is all about querying data rather than modifying it.

## Downloading and Preparing the Data Sets

During the completion of this laboratory assignment, we will rely upon some data sets provided by UniProt protein database. In particular, you will be working with the protein data, which is associated with the Parkinson's disease, and an Apoptosis protein data set, which concerns cell death.

To access the data files needed for this assignment, you must go to `http://www.uniprot.org/uniprot/?query=parkinson&sort=score`, click on "Download" link and save this data set as a "Tab-separated" compressed file. After the file is downloaded, you should untar it and save it into "lab2/data" directory inside your course repository. Repeat the previous steps for the second data set, found on `http://www.uniprot.org/uniprot/?query=apoptosis&sort=score`.

Now, open each data set and note the column attributes (column headers). You will need these column headers in the next step of the assignment. Copy the column attributes to a separate temporary text file (to hold on to them for now) and then remove their line from your original data sets. This will be the first line of the file of data.

## Discussion Questions

Please answer these questions before you create your database.

- What is the relation schema for each data set that describes the association between their entities?
- How are the column headers helpful for designing your schemas?
- What is an appropriate primary key for each data set?
- What kinds of memory (`VARCHAR memory allocation`) do you think you need to create your base? Why?

## Creating Database Tables

To create the required tables, start SQL by typing `sqlite3` in your terminal. Next, following the example from the class exercise corresponding to a text file that you should have received

in your Bitbucket directory after class on the 8th Sept 2016. If necessary, Please use the slides concerning other SQL, tables and schema information from your course to help with this project. Locate the correct commands to create an SQL table for Parkinson's and Apoptosis protein data. Call your tables, Park and Apop, respectively. Using the schemas you designed in the previous step, write SQL commands necessary for creating tables for Parkinson's and Apoptosis data sets.

Now, consult the SQL documentation at http://sqlite.org/ to learn how to import the Parkinson's and Apoptosis data files (that you downloaded from the above links) using .import command. Use this command to populate each database table from the data files.

At this point, you should use the .save command to save your database in your repository's ''lab2'' directory. Remember, please do not submit your database or the data files to the instructor since these files will be very big.

## Querying the Database Tables

During this part of the lab you will design and run several queries to answer the following questions:

1. Write a query that will return the count of elements in the Entry columns in both tables.
2. Write a query that will return the distinct count elements in the Entry column.
3. Discuss: From the above two queries, is this column a good primary key for each table? why or why not? (if not, then what column would you recommend, instead?)
4. Write a query that will return the number of records associated with ''Zea mays (Maize)'' in both tables. You might want to first determine where entity is found in the table to create your query.
5. Write a query that will report how many organisms were listed in each table.
6. Write a query that will return the number of organisms which are *common to both tables* (as in, the intersection of the tables for this attribute).
7. Write a query that will return the number of proteins which are common to Apoptosis and Parkinson's, which are associated to the Zea mays (Maize) organism.
8. Modify this query to print the first 15 proteins (Entry entites), according to Zea mays (Maize) are related.
9. Create a query to determine how *many genes* are in common in both tables, for all organisms (i.e., the intersection of all information about genes across all organisms).
10. Create another query to determine the *names* of first ten of these genes which are at the intersection of both tables, across all organisms.

Note: To link the two tables, named Apop and Park, where the entries are the same, use the following command:

```
select distinct A.Entry, P.Entry from Apop A, Park P where A.Entry == P.Entry;
```

## Summary of the Required Deliverables

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

1. A reflection text document containing the answers to the questions in red
2. Queries (in a reflection document)
3. Output from the query runs (in a reflection document)
4. SQL source code

Before you turn in this assignment, you also must ensure that the downloaded data sets are not pushed into the repository (I don't need that much repeated data!). You must place all of the required deliverable into a directory named lab2. Please see the instructor if you have any questions about this assignment.

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.