CMPSC 380
Principles of Database Systems
Fall 2016

Laboratory Assignment Two (Part II): 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 and will continue to work with the database that we constructed during Lab 2 (part I).

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 Parkinson's disease, Apoptosis, and now, Alzheimer's disease. In this lab, we will be adding the new Alzheimer's table to the database of last week and will perform queries. It is recommended that the script text file from last week's database building be amended to reflect this new table and to be able to automatically rebuild the base from the script.

Update Your Table

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

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

After entering this site at UniProt (a giant public database concerning proteins), click on "Download" button to save the Alzheimer's data set as a "Tab-separated" compressed file. After the file is downloaded, you should click on it on your Ubuntu desktop to decompress the file. As before, use Vim (or Gedit) to remove the top column-header information. You previously used this header information to build the tables of your database.

Save this file into your working lab directory (i.e., "lab2/data") inside your CS380 course repository. We note that the data files from last week should be already found in this same directory (Apoptosis and Parkinson's).

Data file references:

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

Your table will take the following form and should be created from a database maker script containing the following code. This code was discussed in class. Note: once all code has been entered into the script, you can run it by typing:

- cat scriptFile.txt | sqlite3 database.sqlite3

```
drop table TableName;
create table TableName (
        Entry text NOT NULL,
        EntryName text NOT NULL,
        Status text NOT NULL,
        ProteinNames text NOT NULL,
        GeneNames text NOT NULL,
        Organism text NOT NULL,
Length text NOT NULL
);
```

Re-Building your Database from Last Lab

Add all script for tables (even the tables used last week) into your script to be able to recreate your database from scratch. If this base were used on a regular basis, then you would have to update it regularly with fresh data from UniProt. Updating the table might be easier by running your script. To insert the data from a text file here contained in the "data/" directory, add the following code to your database script

```
/* The delimiter ("|") may used to separate each field in some data.
This data uses the tab ("\t") delimiter */

.separator "\t"

/* find the file and load it into sqlite3 which will create the database.*/

.import data/FILENAME.tab TableName
```

Test that you are able to rebuild the databse containing three tables. Make sure that they are populated automatically from the tab-delimited files.

Please use this code to create your database before continuing on to the questions (red) below)

Creating Database Tables

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 of the Alz table .`
2. `Write a query that will return the distinct count elements in the Entry column of the Alz table.`

3. Discuss: From the above two queries, is this column a good primary key for the Alz 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 the organism ‘‘Zea mays (Maize)’’ in the Alz and Park tables.

5. Write a query that will report how many organisms were listed in each of the three tables.

6. Write a query that will return the number of organisms which are *common to both the Parkinsons and Alzheimer's tables* (i.e., This is the intersection of proteins of both tables).

7. Write a query that will return the number of proteins which are common to Apoptosis and Parkinson's, which are associated to the *Bothrops brazili* organism.

8. Create a query to determine how *many genes* are in common in the Alz and Park tables, for all organisms (i.e., the intersection of all information about genes across all organisms).

9. Create another query to determine the *names* of first ten of these genes which are at the intersection of Alz and Apop 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.