COMP 3380 Assignment 3

Updated Nov. 17, 2021

Due Tuesday, December 7, 2021, 11:59 PM (Winnipeg Time)

HANDIN

Crowdmark does not allow for the submission of code files, so you will hand in your solution to Question 3 on UMLearn.

Question 0 (0 Marks) – Hackerman

Optional (but fun): Go to https://hackertyper.net/# (read the help first) and spend five minutes feeling like a movie hacker before getting back to your assignment.

Question 1 (15 Marks) – SQL Injection

You must perform a SQL injection attack on some provided (and vulnerable) data. There are two versions of this question available, one in Java and one in Python. ONLY DO ONE VERSION, whichever you prefer. All instructions are the same across both versions, apart from how the code is accessed.

Java Version

- Download A3Q1Java.zip and extract
- Run the code using the following command to access the database interface:
  - `java -cp .:hsqldb.jar A3P2 (Linux/Mac)`
  - `java -cp .;hsqldb.jar A3P2 (Windows)`
- See the Injection instructions below

Python Version

- Download A3Q1Python.zip and extract
  - You will need to make sure the `sqlite` package is installed in your Python distribution. If you have Anaconda, it should be there by default.
  - https://docs.python.org/3/library/sqlite3.html
- Run A3Q1.py
- See the Injection instructions below

Injection Instructions

- Type h for help to see the available commands.
- Using injection, perform the following tasks:
  - Delete records you should not be able to delete given the interface
  - Update records you should not be able to update given the interface
  - View records not intended from the given command
    - It is okay if the records would be otherwise viewable; the point is that you're making a command do something it wasn't supposed to do.
- DO NOT edit the script files provided; all your injection should be done via commands in the program

Submission

- Document your malicious commands and the results using screenshots
- Submit your screenshots and a 1-3 sentence writeup on what you accomplished and how.

Question 2 (15 Marks) – Elf Infection

It seems that even the North Pole is not immune to infection, so Santa wants a contact tracing solution. Download the provided ElfContacts Cypher file and run it in Neo4J to create an Elf Contacts database.

Find the answers to following questions:

1. Which elf had the most contacts?
2. Which elf had the fewest contacts?
3. The most popular elf (found in question 2.1 above) was infected at time 1500000000. Which elves did that elf meet while they were contagious?
4. Show three levels of infected elves from the most popular elf.
    a. That is, show the most popular elf, all the people they infected, all the people those people infected, and all the people ***those*** people infected.
5. Repeat question 2.4, but for the least popular elf (also infected at time 1500000000).

Hints and Notes:

- The virus has a 100% infection rate; if an infected elf meets an uninfected elf while they are contagious, the uninfected elf will become infected.
- Elves are contagious for EXACTLY two weeks after infection (accurate to the second).
- Most popular means the same thing has having the most contacts
- "Most contacts" means having most encounters, ***not*** the elf who met the most other distinct elves.
- There are 604,800 seconds in a week
- All times in the database are in UNIX Epoch Time: Seconds since January 1, 1970
- This article may be useful for doing question 2.4 and 2.5
    - https://neo4j.com/developer/kb/comparing-relationship-properties-within-a-path/

Handin

- For each query, submit two images to Crowdmark:
    - A screenshot of the query by itself so it is clearly readable.
    - A screenshot of the query with the results as either a table or a node graph

Question 3 (20 Marks) – Programming

Download the provided files in A3Q3.zip. Each text file is in CSV format and contains the data for a single table. You must perform JOIN operations in code on these tables. You may use any programming language you wish.

Write code to run a NATURAL JOIN, a LEFT JOIN, and FULL OUTER JOIN on the two tables. Output your results to a CSV file, one per result.

Notes:

- Do not edit the provided CSV files in any way.
- Your code must be generic; the files we test your code on will not be the same files as the ones you are using here.
- Assume that the top row of any file will always contain headers and you should join on ALL matching column names.
- If there are no matching column names, your code should return a Cartesian product
- Do not import any libraries that do merges/joins for you; you must do them manually in your code. You can import libraries for data structures (such as ArrayList in Java), but nothing that does the joining for you.

Handin (on umlearn)

- Submit your code file, which the markers must be able to run without edits.
- Include a simple README with any instructions necessary. Ideally this should just say "run the file".
- Also submit your three CSV output files