CS386D
4/4/20
V1.0

Homework 8

Assigned: 4/4/20
Milestone 1: Tuesday, April 7, 9:30AM[1]
Full Assignment Due: 4/10/19, (Friday), 11:59PM on Canvas

Objective: The problems in the homework are a little hodgepodge as we get the course material back under our feet and pointed in a good direction. FYI, we are proceeding straight into the use of fully managed database instances in the Cloud, starting with The PostgreSQL image offered by Amazon Relational Database Service (RDS), and Graph Databases.

Part 1: Back-fill Homework 6a

Raw I/O bandwidth vs. reality.

Per discussion in class, I aimed to convert the load time, in minutes, of your database load in homework 6a to a measure of I/O bandwidth in bytes per second. The first necessary calculation was to determine the total size of the database, 5 million rows, at 128 bytes each, or roughly 1.25Gbytes. Recall, my first effort at calculating this at the whiteboard was off by three orders of magnitude. In retrospect, I made a nearly equally large mistake in the calculation of bandwidth, and the discussion degenerated from there.

Given your measured load time, (sorted or unsorted, your choice. Simply be obvious in your answer that you have provided this figure)…. Given your measured load time, in seconds, for your database of roughly 1.25Gbytes, at what rate, in bytes per second, were you able to load the database?

Extra Credit: I don't know how, in the first step, I enunciated 1.25Mbytes instead of 1.25Gbytes, but, looking back, I know the mistake I made for the second calculation. What was it?

Part 2: Moving to the Cloud

Per Piazza post, optionally, (but highly recommended if you don't already have experience with running an RDBMS on AWS), the class is moving to PostgreSQL, (which is commonly simply called Postgres). See Box class folder for instructions on getting started on AWS

---

[1] Per Piazza, some students did not get an invite email to AWS. I have determined that there is no way to add a student to an AWS class account. (The accounts are set up in 1 shot). So I will have to apply for a new class account for any such student. According to AWS, approving an account may take 3 or 4 business days. So it is imperative that you determine if you have account privileges by Tuesday morning.

A) You should load Postgres and its GUI client pgadmin, onto to your personal computer, and verify from the GUI that you can create a table, insert a row, and run a query.

   You may also want to exploit the stack builder installer and at least review the optional parts available for creating a more expansive environment. Those of you using Python and not Java will especially want to do this.

B) Milestone 1: Similarly, you should spin-up a Postgres instance in Amazon RDS and verify that it functions, as you did in Part A.

Part 3: Yes, you can have a graph in a relational database. You can even use relational constraints to keep the contents of the database correct.

This is a programming exercise per the specification of referential integrity constraints (foreign-key constraints), including some of its finer details. It is also an introductory exercise intended to demonstrate that there is no problem storing data as a graph in a relational database. So we will see there are elements of the SQL 3 standard that were included just for this purpose**.**

Consider the data model in the Monge et.al book concerning employee and recursive associations.
http://web.csulb.edu/colleges/coe/cecs/dbdesign/dbdesign.php?page=recursive.php

Note, Ch. 4.7 and Ch. 4.8 of your text concern the use of UML class diagrams for data modeling. I use this on-line text http://web.csulb.edu/colleges/coe/cecs/dbdesign/dbdesign.php as the primary source in the undergraduate class, as it is much more expansive than your textbook, and at the undergraduate level this material is of primary importance.

For this homework problem, I bring the section of that book on recursive relations to your attention as it reveals in much greater detail that a 1-to-many association, (like the inventions/inventions model), can be implemented using foreign keys that reference primary keys, but in a recursive (self-referencing) association, a) the foreign key values are in a column in the same table as the primary key, b) this is tantamount to a method of representing trees in a relational database. From this starting point, we will soon discuss representing more general graphs.

For the following variations, create a table, load the database, per the following data, and attend to the details of the following scenarios.

The data is as follows:

Fenves, President (UT)
Goldbart, Dean (Natural Sciences)
Wood, Dean (Engineering)
Fussell, Chairman (Computer Science)

Beckner, Chairman (Mathematics)
Tewfik, Chairman (Electrical and Computer Engineering)
Miranker, Professor (Computer Science)
Mok, Professor (Computer Science)
Ghosh, Professor (Electrical and Computer Engineering)
Alcook, Professor (Mathematics)

For each variation start with the complete set of data above.

A. Write a SQL query that returns the names of all the professors, (but not the chairmen), in the college of Natural Sciences.

   Turn in the SQL query and the result.

Do not make the problem any more complicated than necessary. i.e. each employee has just one supervisor and the use of keys and referential integrity constraints is the basis of a straight forward solution.

B. Set up referential integrity constraints, such that if an employee's boss loses his job, the manager field is set to Null
   - Delete Chairman Fussell from the database.
   - Write and execute a query showing the final contents of the table.
   - Turnin your DDL (your "create tables" showing your referential integrity constraints and their extra qualifications.

C. Set up referential integrity constraints, such that if an employee's boss loses his job, he/she also loses their job.
   - Delete Dean Goldbart from the database.
   - Repeat as above

D. Set up the referential integrity constraints such that if a boss is changed as the result of an update, the table is updated per the new bosses name, and if an employee's boss loses his job, he/she also loses their job.
   - Update the Chairman of Computer Science to be "Smith".
   - Repeat as above.

   Challenge, think about, question (i.e. don't hand anything in, just spend a few minutes thinking about how you might do this):

   What if Fussell's appointment as Chairman reaches its end, and the CS department votes Miranker to be the new Chairman. Hence there are two updates to the table, yet no one should lose their job.

   If necessary, you may use a trigger.