CS 7330

Project – NoSQL Databases

The goal of this project is for you to get some exposure of NoSQL database by developing a small application using a NoSQL database.

Application – Bus network

The BW bus company operate a network of bus routes across cities in the United States. The company wants to build a database that store information about its bus network. It requires the following information be stored.

Routes

The company runs a set of routes. Each route has a unique alphanumeric ID (with at most 5 characters) associated with it. Some (but not necessarily all) routes have a name associated with it (at most 80 characters, any printable characters, including space, are allowed) -- notice that those names may not be unique between routes.

Each route is a *one-way* route from one city (departure) in the USA to another (destination). (We assume city names are unique within a state only). There is no guarantee that a return route for another route exists. (I.e. there is a route from A to B does not imply that there will be a route from B to A, and if there is, it will be under a different route number). There are three types of routes

- Those who runs once daily
- Those who runs once daily only on weekdays
- Those who runs once daily only on weekends

Each route has the same departure time on days that it operates.

We also want to record the travel time of that route.

Notice that there can be many different routes from one city to the other, and each of them may take a different amount of time. (You need to record the time in terms of hours and minutes, and the only other thing you can assume is that travel time is no longer than 72 hours).

Notice that for this project we are NOT interested about intermediate stops made by a route and you are not required to store them.

Bus drivers

The company have a set of bus drivers. For each driver, we records his/her last name and first name, and age. Also each driver has a unique ID associated with it.

We also want to record the hometown of each bus driver. The company only hire drivers from cities where there is a route that serves as its destination (destination ONLY, not departure city).

Driver Assignment

We need to keep track of the assignment of drivers to buses. Each driver is assigned to a set of routes on certain day (e.g. the Monday route of route 85, the Tuesday route of route A31B, the Thursday route of route 73 etc.). The assignment is the same every week.

There are a few constraints that need to be satisfied with respect to route assignment:

1. A driver cannot be driving more than one bus that travel at the same time
2. A driver must be given enough rest (defined to be half the duration of his previous route before taking on the next assignment)
3. If the departure city of a driver's next assignment is not the same as the destination of his current assignment. He/She can either take bus(es) from the company that will lead the driver to its next assignment (notice that connecting through multiple buses are allowed), or he/she has to be given 48 hours to reach the next assignment.
4. The driver should be assigned a route that reaches his/her hometown. Once the driver arrived at his hometown, he/she should be allowed at least 18 hours of rest before the next assignment. (If 18 hours is not enough rest based on condition 2, he/she need to be given more).

Notice that for all tables above, if the type of the field is not specified, you can make reasonable assumptions (but specify your assumptions).

Task 1 – Database selection and design (25% of the project score)

You are to choose a NoSQL database system to implement a database to store the above information. You will need to design the schema that is used to store the data.

You can choose amount MongoDB, Neo4j and Redis. If you want to use another NoSQL database system you need to let me know by 10/18 (Sun) at noon. I will get back to you by 10/19 (Mon) at noon to inform me my decision.

You will need to submit a report by Sunday (11/1) 11:59pm detailing your database design and schema. I will get back to each group via zoom (probably 5-10 minutes) to discuss that (hopefully within 3 days of submission).

You will be given a 10% bonus for this part if you hand in your report by 10/24 (Sun).

Task 2 – Implementation of system (75% of the project score)

You are to implement a program that allow user to enter data, as well as query information about the system. Your program can be implemented in either C, C++, Java, Python, Perl or Ruby. If you want to implement it in any other programming language, you need to get my approval (by 10/24) first.

Your system should do the following:

Data Entry:

The system will ask for the user to provide a set of files. Each of them is a .csv file that contains fields that are comma separated. You are to read those files and then populate the database with the file provided. The user will provide the following files:

- Routes file

- o The file contains information about the bus routes. Each line of the file denotes a single route. The fields are as follows (in order):
    - Route number
    - Route name (left empty if not present)
    - Departure city name
    - Departure city code (standard US state code, 2 characters)
    - Destination city name
    - Destination city code
    - Route type code (0 for daily, 1 for weekdays only, 2 for weekend only)
    - Departure time (hours)
    - Departure time (minutes)
    - Travel time (hours)
    - Travel time (minutes)
- Driver file
    - o The file contains information about drivers. Each line is a driver. The fields are as follows:
        - Driver ID
        - Last name
        - First name
        - Age (in years)
        - Hometown city name
        - Hometown state name
- Driver Assignment file
    - o The file contains assignment of drivers to buses. Each line is a single assignment. The fields are as follows:
        - Driver ID
        - Route number
        - Day of the week (M for Monday, T for Tuesday, W for Wednesday, U for Thursday, F for Friday, S for Saturday, s for Sunday)

You will decide on the order of the files to be read. You have to account for the fact that files can be huge.

You need to check for the correct data type and need to flag the system if there is an error. If a line has an error, the corresponding info for that line is not inserted, but the system should continue to examine the next line

You are to ensure that all integrity constraints are satisfied (especially for those rules about route assignments).

Notice that the format specified above are for input purposes, but how you actually store the data (e.g. using numbers instead of character for day of the week) is completely up to you to decide.

Query

- The program should get the name of a driver (first and last), and print out the driver's information, together with the route that the driver is assigned to. (If there is multiple drivers that satisfies the query, print them all (one after the other)).
- The program should get the name of a city, and print out all the routes that go through the city (separate departure and destination, order by time: assume Sunday 00:00 am is the starting time).
- The program should get the route of a bus and print out all information about the route, including the name and ID of the driver that is assigned to it.
- The program should get the name of two cities, and response if there is a bus route that go from the first city directly to the second. If so, print all info about that route (as the previous query). If there is more than one, print them all (one after the other)

You are NOT required to develop a GUI for the program. You can run the program from command line and keep asking user to provide the information via the keyboard.

Bonus

There are 2 possible bonuses, each award an extra 10% to the total grade. 7000 level students are required to implement at least one of the two

Bonus 1

Implement the following query:

- The program should get the name of two cities, and a day of the week (Sunday/Monday … etch.) and response if there is a way to get from the first city to the second, either by a single bus route, or by connecting buses, leaving on that day of the week. If connection is used, the total time from departure to arrival should be less than or equal to 72 hours. For each possible routes, list all the bus routes that is used. You should show departure and arrival time for the bus route. For example:

  Chicago, IL to Cleveland, OH

  Route 78 Chicago IL – Cleveland OH, leaving Monday 7:15am, arriving Monday 12:15pm

  Route 73 Chicago IL – Detroit MI, leaving Monday 8:15 am, arriving Monday 10:20 am
  Route A36 Detroit MI – Cincinnati OH, leaving Monday 11:35 am, arriving Monday 5:10 pm
  Route 88OH Cincinnati OH – Cleveland OH, leaving Tuesday 8:15 am, arriving Tuesday 12:05 pm

Bonus 2

- Provide a GUI interface to your program. If you want, you can leverage the web browser as the GUI. Functionality is the most important part, aesthetics counts only a minor part of the grade.

Due Date

Each group should make a 10-15 minute video that demo your system and submit to me before 11:59pm 12/1 (Tue). We will have them posted on Canvas and I will comment on those on the class at 12/2. You will than have a few days to make final changes. The project is due 12/6 (Sun) 11:59pm.

Notice that the majority of the grade will be based on what have been achieved by 12/2. You will need to provide resources needed to run your program (except materials/code/libraries that can be downloaded from other sites). You should also provide a 2-4 page user manual, and a 2-4 page developer manual (contain enough information for other people to continue your task).