EPBI 414 (Fall 2016) - Assignment 4

Database Warehousing and Optimization

Overview

This homework consists of two parts. For the first part, you'll answer some questions about the material covered this week. Next, you'll review some data that is stored in the 3NF, and create a star schema out of it. And - you have a chance at a bit of extra credit!

Logistics

Submit your assignment in a .zip file labeled in the following matter:

<Case ID>_EPBI414_Fall2016_A4.zip

So, if your Case ID is tar9, you would submit the following zip file:

tar9_EPBI414_Fall2016_A4.zip

Part 1

1. You are given a very large table (tens of millions of records) which contains the following columns:

   ```
   social_security_number
   first_name
   last_name
   date_of_birth
   gender
   race
   ethnicity
   height
   weight
   city
   state
   ```

   For the purposes of this assignment, you can assume that the data is complete (all columns filled for all the rows), and that each Social Security Number (SSN) uniquely identifies a single person (that is, no one appears in the data twice, with two different SSNs). The table has as its primary key the Social Security Number.

   A multi-column index works in the same basic way as a single column index, except that the ordering is done based on multiple columns. This is the same as using Excel to sort by one column, and then by another column, and so on; the table is first ordered

by the outermost column, then within each band of that column, the rows are ordered using the next column, and so on.

a. Given this table, you are told that the analysts often run queries to find all the people in a few cities, or a few states, and that they often want to filter those results by the gender of the individual. What would you recommend as an index to be used on this table? Remember, the order of the columns in the index matters: an index on `column1, column2` is not the same as an index on `column2, column1`. [6 points] Why did you pick the order you used for the columns? [4 points]

b. Imagine that the table above has an index set on `race, gender`. An analyst writes a query and gets back all the people who have `race` = 1, which is the code for White, and `gender` = 1, which is the code for Female. Next, the analyst tries to write a query to return only the people who are male (`gender` = 2). They find that the request takes longer than the previous one, even though it is only filtering on one variable. Why does this take longer? [5 points] How would you suggest changing things to improve performance? [5 points]

c. Most database management systems automatically create an index on the primary key (this may vary among RDBMSs). In this case, that would be the Social Security Number. Give an example of a real-world use case where having an index on only the Social Security Number would be useful. [5 points]

2. Give an example of a real-world use case where having a database with ACID compliance is important. Explain why ACID is important in your hypothetical case. [5 points] Then, give a real-world use case where ACID compliance is less important, and explain why it is not as essential. [5 points]

3. Explain why transaction isolation might be very important for a multi-site clinical trial data capture system. [5 points]

4. Imagine that you have a database storing an identification number (`id_no`), along with two values, `val1` and `val2`. The table has as its primary key the identification number. You are often interested in the sum of `val1` and `val2`, and one of your analysts creates a column storing this, called `val_sum`. Why does this violate the 3NF? [5 points]

Part 2

The file "EPBI414_Fall2016_HW4_ERD.pdf" contains a figure called an *entity relationship diagram* (ERD). We may discuss ERDs in greater depth in the future, and there are different ways of drawing them. This is a very simple ERD, and it is designed to show a simple relational

system that might be used for tracking billing in a hospital system. (Obviously, real systems are far more complicated than this one!)

Each of those boxes represents a table in our data system. Primary key columns are marked by the letters PK, while foreign keys are marked with FK. Each foreign key is linked back to another table using a line. The lines end with a shape that looks something like this:

That means that the line is a *one-to-many* relationship, with the end that has this shape being the "many" and the other end being the "one".

There are six tables in this current system. However, some hospital analysts have been complaining that it is too hard to get useful information from this system. They have to use too many tables to get data they want, and that means that it takes a long time to train new analysts. They would like to use a star schema to aggregate data for reporting purposes, and they've asked for your opinion on how to design it!

Create a star schema to help meet the needs of the analysts. It should have a single fact table, called `fact_procedure_events`, with a primary key called `procedure_event_id`. This is the same column as the one from `procedure_events`. As the dimensions of the fact table, you should have procedure, visit, clinic, doctor, and patient. As the facts in the fact table, you should have the amount billed, the amount spent, and the total costs.

Your schema should support the following aggregations:

- Counting the total number of visits and patients from a specific city or state
- Breaking down the amounts billed and received by each clinic over the year, month, and quarter, and then aggregating this up to citywide and statewide totals
- Identifying which clinics had the lowest average cost for inpatient and outpatient procedures, broken down by procedure specialty
- Finding the procedure code with the largest average difference between amount billed and amount received
- Identifying the most expensive doctor assigned to each clinic, keeping in mind that while a doctor may do many procedures at many clinics, they are always assigned to only one

The analysts ask that, when you create the dimension tables, you include the names of doctors, patients, clinics, and procedures, so they can write their reports without having to go outside the system.

You can use whatever tool you want to draw out your schema. I used http://draw.io to create the examples for this homework. [38 points]

Now, based on your new star schema, answer the following questions.

5. Does the fact table you created meet the second normal form? [2 points] Why or why not? [5 points]

6. Does the fact table you created meet the third normal form? [2 points] Why or why not? [5 points]

7. Which of your dimension tables meet the second normal form? Why? [5 points] Which of them meet the third normal form? Why? [5 points]

8. If you added a sixth dimension to your fact table, called date_id, you could store the date of each visit in the fact table itself. Why would you use a date_id instead of just storing the date? [4 points] If you did this, the visit_id would be a strong candidate to become a ***degenerate dimension***. Why? [4 points]

Part 3 (extra credit)

During the lecture, I mentioned that searching for a unique value in an unordered list of unique values requires you to look at half the values, on average. Explain how this is outcome can be predicted using the negative hypergeometric distribution. [10 points]