CS434 – Data Base Theory and Design

Project #4

Author: Mark McKenney
Organization: CS434, Department of Computer Science, SIUE
Points: This assignment is worth 35 points.

Team Database Application (TDA): Part 4 - Loading Large Data Setsg

Description

In this homework assignment, you will develop and load your database with a substantial amount of data so that we can test the efficiency of your schema and perform simple tuning procedures.

You MUST use your real data set. If, for some reason, you cannot get your real data set to work, you may generate fake data, **but you must inform me of this at least 2 days BEFORE the project due date**. If you fabricate data, it can consist of random values, but you must be sure not to violate any database constraints; for example, you must ensure primary keys are unique, foreign key constraints hold, etc. Note that it is both fine and expected for your data values—strings especially—to be meaningless gibberish. The point of using large amounts of data is so that you can experiment with a database of realistic size, rather than the small "toy" database we have created so far.

Regardless of your data source, you must create data corresponding to the schemas of your relations. You may use either a bulk loading program provided by your database, or you may create a program that generates the appropriate INSERT statements to insert all the data into the database.

The data you generate (either real or fake) and load should be on the order of:

- At least one relation with **hundreds of thousands** of tuples.
- At least one relation with **tens of thousands** of tuples.
- At least two relations with **thousands** of tuples.
- All other relations must have at least 5 tuples.

If your real data set has trouble meeting the above criteria, **inform me at least 2 days BEFORE the project due date.**

If the semantics of your application includes relations that are expected to be relatively small (e.g., schools within a university), it is fine to use some small relations, but be sure that you have relations of the sizes noted above as well. When using fabricated data, there are two important points to keep in mind:

1. Be sure not to generate duplicate values for primary key or unique attributes.

2. Your TDA almost certainly includes relations that are expected to join with each other. For example, you may have a `Student` relation with attribute `courseNo` that's expected to join with attribute number in relation `Course`. In generating data, be sure to generate values that actually do join—otherwise all of your interesting queries will have empty results! One way to guarantee joinability is to generate the values in one relation, then use the generated values in one relation to select joining values for the other relation. For example, you could generate course numbers first (either sequentially or randomly), then use these numbers to fill in the `courseNo` values in the `Student` relation.

With a real data set, joinability is normally not a problem. However, it is sometimes the case that you will have to split data from a single data source into multiple files. In this case, it is useful to generate ALL INSERT STATEMENTS for that data at once. In this way, you can make sure to use the correct values to have joinability.

What to Turn In

Turn in **your program code** for generating or transforming data, a **VERY SMALL sample of the records** generated for each relation (5 or so records per relation), **and a script** showing the loading of your data into the database by a bulk loader, or the source code and a script showing the execution of the program that you write to insert the data into the database. You must provide enough documentation to convince us that the required information is in your database! You have some freedom in the documentation you choose to submit, but it should verify that the required number of tuples reside in the database.

Please remember to indicate your names on the submission and **attach a copy of your relational schema**.

Deliverables

A **SINGLE PDF** containing the artifacts listed above. To generate a single PDF, it is probably easiest to create separate PDFs for your samples of records, code, script/screenshots, then merge them into a single large PDF.

On Mac, you can use Preview to merge PDFs. To do so, open 2 PDFs in separate preview windows. Under the "View" menu, choose "Thumbnails", which will cause a sidebar to appear showing thumbnails of the pages of the PDF. Do this for both PDFs. Then, highlight ALL thumbnails in one PDF, and drag them to the sidebar of the other PDF. One problem is that if you drop the thumbnails too far from the existing thumbnails in a sidebar, it will just use the one window to show 2 separate PDFs, so make sure this does not happen. It will be a single PDF if only 1 document title is present in the sidebar.

On Windows, you will have to ask a Windows expert.

On Linux, you can figure it out!

Maintaining Your Databases

You will be using both your small (previous TDA) and large (this TDA) databases for the rest of the course. The idea is to use the small database to experiment on meaningful-looking data, and the large one to experiment on data of more realistic size. We suggest that you keep the load/data files for the two databases for the duration of the project. Specifically, we suggest that you establish some kind of routine that includes reloading your database from the files created in this or the previous project part each time you want to get a "fresh" start with your database. Remember to delete the contents of each relation (or destroy and recreate the relations) before reloading. Otherwise, your database will happily append the new data to your old relation, causing your relation size to double, triple, quadruple, etc (unless key constraints or some other constraints prevent this, in which case you will simply see a lot of error messages).

If you simple skipped the small database and loaded everything for the previous project, it is perfectly OK to have only a LARGE database!.