



ISM 6930 Healthcare Data Management

Group Assignment #3 – Indexing “Experiment of the Week”

Description

This group assignment provides an opportunity to gain some practical experience with indexing. The “experiment of the week” assignments are intended to be more open, with your group picking the specific topics to investigate. Basically, this gives you a chance to be a bit more creative and follow directions of specific interest to you (and more relevant to your career). Besides creativity, you should try to pick experiments that are interesting and carry them out with good technical skill. That does not mean the results have to be in line with your intuition, counter-intuitive yet interesting results are great. Besides the topic of the week, such as indexing, you are free to bring in any past topics to complement your experiments. For instance, you might develop some new queries that fit better with the current experiments.

The deliverable is an “experiment of the week” write-up that includes an explanation of each experiment, along with screenshots, figures, and/or tables that highlight key steps or results. Along with some ideas below, there are “hall of fame” examples that show fragments of past student projects. These fragments and hints are selected because of interesting features, but are not guaranteed to be completely correct. So, use them for inspiration and develop your own informed results (i.e., “trust but verify”). Together these hints and ideas should help you complete your assignments and learn along the way.

Project Ideas

Idea 1: Investigating Selectivity

Look up selectivity in your database textbook. Essentially, the optimizer makes a decision regarding index use based on the fraction of results returned. Indexes are most useful when selecting a small fraction of the available records. You can conduct a simple experiment to find out where the cutoff percentage lies by developing a simple single table query and gradually shrinking (or growing) the query range.

Idea 2: Start Simple and Show that Indexing Works

You can start with a simple experiment that builds on your query writing. Take a simple query and improve the performance by adding indexes (and/or using any subsequent techniques). The idea is to take a query scenario based approached to performance tuning. Then expand to more complex queries.

Idea 3: Primary Keys and Indexes

Why index a primary key? Primary key constraints can be expensive to enforce, since any new values have to be unique (and compared with all existing values). So, a fast lookup based on the primary key value is very useful (based on an index). You could explore this by creating a table with no primary key constraint, but with a unique constraint on the candidate key column. Do an INSERT and look at the execution plan and cost. Now put a primary key constraint in place, which automatically creates an index. How does the performance differ? I have not tried this, but it sounds interesting.

Idea 4: Indexing for Different Query Types

There are many types of queries. Some queries are highly focused and return a single row or small set (a “point queries”). Other queries return larger sets based on ranges of specific attributes (range queries). Report-like queries typically scan large amounts of data and often form aggregates for results (scan queries). You could explore the importance of index structures under these different scenarios.

Idea 5: Function-Based Indexes

Database systems often provide methods for improving performance for computed columns. It is often good practice to derive data from several existing columns. However, these multi-column computations can be expensive to calculate at query time. Several techniques such as materialized views or function-based indexes can improve performance in these situations. In particular, function-based index structures store the calculated values for efficient retrieval. As part of an experiment, you can create a computed attribute and try queries with and without function-based indexes.