CS 564, Fall 2019
Assignment #4: Query Optimization

Due Date: TBA. ***No late days.***

Project Grade Weight: 15% of the total grade

Introduction

In this assignment, you will still use SQLite3 to perform various queries on TPC-H database (introduced in assignment 2). However, for this assignment, the TPC-H database you'll be working with in this assignment is about 1.2 GB (scale factor = 1) instead of 100MB in assignment 2. We have provided you the TPC-H database file, available here.

The goal of this assignment is to help you understand query optimization from hands-on experience. Specifically, you will be asked to rewrite several given queries to make these queries as fast as possible.

Queries

For each query below, you are going to optimize it through various strategies (e.g., to run with a different join order, to push grouping operations below a join, to add indexes to the database, etc.) and put your optimized query in a separate file. For example, given ***query2.sql***, your need to save your rewritten query in ***query2_opt.sql***.

query2.sql

query3.sql

query4.sql

At least three different strategies should be used for this assignment!

Measuring Performance

Note that each file above starts with `.timer on` to print the execution time as below after query results.

```
Run Time: real 0.119 user 0.039955 sys 0.041150
```

As the execution time varies each time, you can run the same query five times to measure the **average execution time**. You only need to collect the **user** time (instead of real/sys time) for this assignment.

Modifications on Database

For this assignment, please download the provided database and only modify your local copy of the database. For any modifications you need to make on the database (e.g. indexes), you need to provide a file named ***preprocessing.sql*** and run this file before running your rewritten query. You also need to provide a file named ***clean.sql*** to clean up any changes you made. (Remember to clean any changes you made before measuring your baseline, i.e. provided queries)

Requirement on creating indexes: the final, optimized queries must all run against the same database, i.e., **the same set of indexes** and **no more than three indexes should be created per table**.

Report

Fill up a report using the template file, report.txt, describing how you optimize each query and the percentage of improvements after optimization.

- Measure the percentage of improvements (using query2 as an example)
    - Run query2.sql multiple times and take down the **average execution time** (user time) as *t_base*
    - Run query2_opt.sql multiple times and take down the **average execution time** (user time) as *t_opt*
    - percentage of improvements = *(t_base – t_opt) / t_base* (keep two decimal points)
- Note that we require your *t_opt* to be exactly smaller than *t_base* for each query.

Grading

- An optimized query that do not return the correct results will result in **ZERO** points for the query. So, **please make sure your rewritten query always returns the correct results** (exactly the same as the results returned by given query).
- Detailed rubrics will be released later on piazza.

Submitting Your Assignment

Only the following files are required for this assignment:

- query2_opt.sql,
- query3_opt.sql,
- query4_opt.sql,
- report.txt
- preprocessing.sql (if made any changes like creating indexes on the database)
- clean.sql (if made any changes like creating indexes on the database)

When you're finished, please follow these instructions to submit the project:
1) Place all six required files in a directory.
2) Name this directory using the format: <netID>_P4 (e.g.klklassy_P4).
3) Run:

```
tar -czvf <netID>_P4.tar.gz <netID>_P4
```

4) Submit the tar file.
5) To check, you can uncompress the tar file. (run: tar –xzvf <netID>_P4.tar.gz)