Lab 6: Query Optimization and Data Visualization

Deadline: Friday, Mar. 2nd at 11:59pm.

Goals:

The goals for this lab are twofold: 1) experiment with B-tree indexes to speed up query time; and 2) visualize the Airbnb data with simple charts.

Inputs:

-Aibnb schema with modifications from Lab 5.

-6 aggregate-group-by queries from Lab 5.

Part 1: Indexes

Desired Outputs:

-For each aggregate-group-by query:

- Provide the baseline EXPLAIN ANALYZE output for that query.
- Provide at least one suitable index definition to speed up query execution.
- Provide the new EXPLAIN ANALYZE output after index creation that shows an index scan in the query plan.

Part 2: Data Visualization

Desired outputs:

-A Dashboard that meets the following requirements:

- A Data Studio report that contains 6 charts.
- Each chart visualizes the results from an aggregate-group-by query.

- The report is shared with lab partner with full edit permissions, enabling both partners to collaborate on same report.
- The report is shared with cs327e.spring2018@gmail.com with view permissions only.

Tools You Need:

-GitHub

-Cloud SQL for Postgres

-psql client

-Data Studio

Code Organization:

-The create index statements for Part 1 should be stored in the file create_indexes.sql.

-The explain analyze output for Part 1 (before and after index creation) should be stored in the file explain.txt. Output must include the explain analyze statement in addition to the generated query plan. Add a short comment between queries that summarizes the results. For example:

/* Query 1: w/out index: 55 ms; w/index: 33 ms */

-A screenshot of the Airbnb Dashboard for Part 2 saved as file dashboard.png. Report should be in view mode when screenshot is taken.

-Any modified SQL for the aggregate-group-by queries, if applicable, stored in the file updated_aggregate_queries.sql.

Implementation Hints:

-Run VACUUM ANALYZE on all Airbnb tables prior to analyzing query plans.

-Run set enable_seqscan=off; to disable table scans while testing a candidate index.

-Whitelist the Data Studio IP ranges below to allow Data Studio to connect to your Postgres instance.

IP Ranges:
64.18.0.0/20
64.233.160.0/19
66.102.0.0/20
66.249.80.0/20
72.14.192.0/18
74.125.0.0/16
108.177.8.0/21
173.194.0.0/16
207.126.144.0/20
209.85.128.0/17
216.58.192.0/19
216.239.32.0/19

-Create 6 data sources in Data Studio, one for each aggregate query.

-Give each data source a descriptive name instead of keeping the default name.

Index References:

Vacuum command: https://www.postgresql.org/docs/9.6/static/sql-vacuum.html

Analyze command: https://www.postgresql.org/docs/9.6/static/sql-analyze.html

Basic create index command:
https://www.postgresql.org/docs/9.6/static/sql-createindex.html

Multi-column indexes:
https://www.postgresql.org/docs/9.6/static/indexes-multicolumn.html

Partial indexes:
https://www.postgresql.org/docs/9.6/static/indexes-partial.html

Data Studio References:

Data Studio Console:
https://datastudio.google.com/u/0/navigation/reporting

Video Tutorial "Report like a Boss using Google Data Studio":
https://www.youtube.com/watch?v=C1w-yuTDUeM

Postgres Connector for Data Studio:
https://support.google.com/datastudio/answer/7288010?hl=en&ref_topic=7332343

Snippets:

Tickit Example Indexes: https://github.com/cs327e-spring2018/snippets/blob/master/create_indexes.sql

Additional Notes:

-Create a lab6 folder in your git repo and place your work in this folder.

-Submission is done through Canvas with a submission.json file.

-The submission.json file should be in this format:

```
{
    "commit_id": "[commit id]"
}
```

-There should be one submission only per team.

-Lateness penalty is %10 reduction per late day.