ECE:5995 Modern Databases – Fall 2020

Homework 1 – Relational databases – Postgresql – Users and product reviews

Due Thursday Sept 24, 2020 11:59pm.

Task 0. Create a new schema called **homework** in your postgres database (we'll use this schema for all postgresql homeworks).

Task 1. Create table users

```
create table users (
      userid char (20) primary key,
      firstname varchar (60), lastname varchar(60),
      email varchar(100), gender char(10), dob date);
```

Import the file users_select.csv into the users table (use the import utility from phppgadmin).

Task 2. Execute the reviews_sql.sql file to create table reviews
For your ref (included in reviews_sql.sql):

```
create table reviews (
      id serial primary key,
      productid char(20) NOT NULL,
      userid char(20) NOT NULL,
      helpfulnessnumerator  int, helpfulnessdenominator int,
      score int, dposted date,
      summary text, review text);
```

Task 3. Marketing would like you to generate some statistics for the selected users:
Q1. The number of users' birthdays per month (so marketing can plan birthday promotions)
Q2. The number of males/females users per age group.

Task 4. Create a `user_stats` table. The table will store userid, ageGroup, averageRating, reviewCount, lastPosted, helpfulNum, helpfulDenum, helpfulnessRate.
The userid is a primary key and there will be one row for each user in the users table.
The ageGroup is computed from the user's age as: Group 1 (<30), Group 2 (between 30 and 50) and Group 3 (over 50) (see postgresql age function).
The avgRating is the average score for all the reviews the user has posted
The reviewCount is the number of reviews posted by the user
LastPosted is the last date that this user posted a review
helpfulNum is the sum of the usefulness numerator
helpfulDenum is the sum of the usefulness denominator
helpfulnessRate is the helpfulNum divided by helpfulDenum for users whose helpfulDesum > 0 and 0 otherwise
Create a **stored procedure** named `update_user_stats` to populate/update the `user_stats` table.
Q3. Who had posted the most reviews? (Full name, email, gender, age, reviewcount)
Q4. Who writes the most helpful reviews? (Full name, email, gender, age, helfulnessrate, and helpfuldenum) - break the ties using the helpfulDenum.
Q5. Who posted last? (Full name, email, gender, age, and lastposted)

Task 5. Product reviews. For the following queries, we are only interested in these five productid: B004JRKEH4, B004X3VRLG, B0026RQTGE, B001BM3C0Q, B007K449CE

Q6. Find the review count and averageRating per ageGroup for each product. Show productid, age group, count, and average rating.
Q7. Find the review count and averageRating per gender for each product. Show productid, gender, count, and average rating.
Q8. Find the most helpful, highest rated review and the most helpful, lowest rated review for each product. Show the full review (productid, userid, helpfulnessnumerator, helpfulnessdenominator, score, dposted, summary, review).

Task 6. Indexing. Record the time it takes to execute each one of the previous queries without any additional indexes (with only the primary keys).
Now create AT MOST two index(es) that you believe would benefit this set of queries the most. At least one of the index(es) should be on the reviews table. Record the time that takes to execute the queries with the new index(es).
Now make the reviews index a clustered index (see CLUSTER command) and run the queries once more. If you created two indices over reviews, chose one to be the clustered index.
Present your results in a bar plot (three bars per query: No index, Index, Clustered Index) and three bars for the total performance (sum of the query execution times in ms).
Include the relevant query plans for the queries, and explain the performance changes, include the reason why you decided on this index(es) and why/why not it benefits each query.

Submission:

Prepare a report with the answers to the queries (Q1-Q8) and the analysis results (Task 6). Include all the sql statements and stored procedure as appendices at the end of the report.

Submit your complete report as a single file to the dropbox in ICON by **Thursday Sept 24 11:59pm.**