Database Systems, CSCI 4380-01
Homework # 9
Due Thursday November 7, 2019 at 11:59:59 PM

Homework Statement. This homework is worth 5% of your total grade. If you choose to skip it, Midterm #2 will be worth 5% more. This homework is on using procedural SQL (PL/PGSQL).

Your homework will use the same dataset as Homework 7. Since you are changing data, you will use your own database (`db_username`), created already for you and populated with this data from Homework 8.

There are a few rules:

- Do not create tables or any other objects in any other database, other than your own.
- Do not change the already given tables as it will be costly for me to recreate them. Do all your work by creating new database objects as described below.
- Only create the tables described and nothing else. You can use more than one insert/update/delete statement for each expression, but do not use additional tables to simplify the expressions.

  If you need to create tables to simplify performance of your solution, use only `TEMPORARY` tables that are created within a function and deleted when the current session ends.

  `CREATE TEMPORARY TABLE ...`

Homework Description

In this homework you are asked to create a single PL/PGSQL function called `hw9()` that takes as input a number of parameters and returns a set of hotel and AirBnb listings ranked by goodness of fit to the given parameters. Basically, you are implementing a simple recommendation system that incorporates various considerations.

Before you start.

Assume the existence of a table given below:

```
create table results(id int, etype varchar(10), score float);
```

where `id` is the id of a hotel or listing, `etype` is either `'hotel'` or `'listing'`, and `score` is a floating point score of the match with high values being desirable.

In your function, assume that the table is already populated with some tuples that your function will use in its process. The template provided already creates this table and inserts two tuples but you will may have many more tuples in reality.

```
hw7=> select * from results ;
    id    |  etype  | score
----------+---------+-------
 18999839 | listing |     1
  2514184 | hotelid |     1
(2 rows)
```

These tuples indicate the id of listings and hotels that will be used as seed, showing properties that the user likes. Your objective is to find top k most similar and desirable properties to these in a given region (if any). The specifications are given below.

Function Specifications.

The function assumes the results table is already populated with tuples, and takes as input the following paramaters:

`hw9(iregion varchar, topk int, i1 int, i2 int, i3 int)`

where

- `iregion` is a region name, and
- `topk` is an integer, asking to return the `topk` results based on the ranking.
- `i1,i2,i3` are scaling parameters used in scoring as shown below.

Your function must return a set of tuples with the same schema as `results` that are ids of hotels and Airbnb listings such that:

- All returned hotels and listings (places) are in the given input region,
- Each returned place is reviewed by someone who also reviewed one of the hotels or listings that are currently in the `results` table,
- Each returned place has a score calculated using a number of conditions described below, and the function returns the `topk` highest scored listings only.

Scoring of Hotels.

For simplicity, we will identify hotels by their ID (even though the same hotel may have many IDs). The score of a hotel is computed based on the following measures:

- A: total number of reviews for this hotel,
- B: total number of reviewers that is common between this hotel and hotels currently stored in the `results` table (identified by author),
- C,D,E,F: Average values of reviews for this hotel for cleanliness (C), overall (D), value(E) and location (F).

Then, the score is given by $(A/i1) + B + (C + D + E + F)/4$.

Scoring of Listings.

The score of a listing is computed based on the following measures:

- A: total number of reviews for this listing,
- B: total number of reviewers that is common between this listing and listings currently stored in the `results` table (identified by reviewer_id),
- C: average number of reviews per month for this listing (0 if null),
- D: number of months since last review for this listing,
- E: calculated_host_listings_count for this listing,

Then, the score is given by $(A/i2) + B + (C/2) + (1/(D+1)) + i3/E$.

You can test the function as follows (test with the example given to you):

```
select hw9('Chinatown',10,1,1,1);
```

SUBMISSION INSTRUCTIONS. You will use SUBMITTY for this homework.

Use the template SQL file given to you that creates a single function named `hw9()`, but DOES NOT call it. The template shows a simple procedure with the expected input and output.

Test your procedure and submit the text file on SUBMITTY.