Lab 4: Redesigning the Airbnb Database

Deadline: Friday, Feb. 16th at 11:59pm.

Goal:

The goal is to normalize the Airbnb schema in order to improve the integrity of the data.

Inputs:

-Airbnb database with the staging tables from Lab 2.

Desired Outputs:

Host Table:

-Create a new Host table that stores distinct host records.

-The table should have the following fields from the Listings table: host_id, host_url, host_name, host_since, host_location, host_about, host_response_time, host_response_rate, host_acceptance_rate, host_is_superhost, host_thumbnail_url, host_picture_url, host_neighbourhood, host_listings_count, host_total_listings_count, host_verifications, host_has_profile_pic, host_identity_verified, calculated_host_listings_count.

-The fields in the Host table should be renamed such that the 'host_' prefix is dropped from the name. For example, the field host_url should be renamed to url.

-The host fields should be dropped from the Listings table with the exception of host_id.

-The Host table should have a primary key on the id field.

-The Listings table should reference the Host table via the foreign key host_id.

Calendar Summary table:

-Create a new Calendar_Summary table that summarizes the availability for a listing over 30, 60, and 90 days.

-The table should have the following fields from the Listings table: id, calendar_updated, calendar_last_scraped, availability_30, availability_60, availability_90.

-The id field should be renamed to listing_id.

-The calendar_last_scraped field should be renamed to from_date.

-The primary key for the new table should be the combination of listing_id and from_date.

-The table should also reference the Listings table via a foreign key constraint on listing_id.

-The fields that were copied from the Listings table should be dropped from the Listings table with the exception of listing_id.

Neighborhood table:

-Create a new Neighborhood table which stores distinct neighbourhood and zip code pairs from the Listings table. For example, Rosedale, 78756.

-The field neighbourhood should be renamed to neighborhood_name in Neighborhood table.

-The records with NULL neighborhood_names should be deleted from the Neighborhood table.

-The records with NULL zipcodes should be deleted from the Neighborhood table.

-The primary key for the Neighborhood table should be the combination of neighborhood_name and zipcode.

- The fields neighbourhood_cleansed and neighbourhood_group_cleansed should be removed from the Listings table.

-The field neighbourhood_group should be removed from the Summary_Listings table.

-The old Neighbourhoods table should be dropped.

-The field neighbourhood should be renamed to neighborhood in the Listings table.

-The Listings table should reference the Neighborhood table via a foreign key constraint on the fields neighborhood and zipcode.

-The field neighbourhood should be renamed to zipcode in the Summary_Listings table.

Listings table:

-The city, state, zip and country values should be removed from the street field. For example, "Marathon Boulevard, Austin, TX 78756, United States" should be updated to "Marathon Boulevard".

-The dollar signs and commas should be removed from the values of the fields price, weekly_price, monthly_price, security_deposit, and cleaning_fee.

-The datatype for the price and fee fields mentioned above should be converted from varchar to numeric.

Additional Outputs:

-The table names should be renamed to singular form (e.g. Listing, Review, etc.).

-The ERD from Lab 2 should be updated to reflect the new schema.

Tools You Need:

-GitHub

-Postgres Cloud SQL instance

-Postgres psql client

Code Organization:

-The SQL for the new Host table should be stored in a file named create_host.sql.

-The SQL for the new Calendar_Summary table should be stored in a file named create_calendar_summary.sql.

-The SQL for the new Neighborhood table should be stored in a file named create_neighborhood.sql.

-The SQL for the changes to the Listings and Summary_Listings tables should be stored in a file named update_listings.sql.

-The SQL for the changes to the Reviews and Summary_Reviews tables should be stored in a file named update_reviews.sql.

Implementation Hints:

-Create a copy of a table as a backup before making any changes to it.

-Use the command **create table as select ...** to create a new table from the results of a query.

-Use the distinct clause, e.g. **select distinct ...** to eliminate duplicate records from a result set.

-Use the command **alter table add primary key ...** to add a primary key.

-Use the command **alter table add foreign key ...** to add a foreign key.

-Use the command **alter table add column ...** to add a new column.

-Use the command **alter table drop column ...** to delete a column.

-Use the command **alter table rename column to ...** to rename a column.

-Use the command **alter table rename to ...** to rename a table.

-Use the command **alter table alter column type ...** to change the type of a column.

-Use the **split_part** function to extract a substring from a varchar.

-Use the **replace** function to remove a character from a varchar.

Reference Documentation:

-Create Table As command:
https://www.postgresql.org/docs/9.6/static/sql-createtableas.html

-Distinct clause:
https://www.postgresql.org/docs/9.6/static/sql-select.html#SQL-DISTINCT

-Alter Table command:
https://www.postgresql.org/docs/9.6/static/sql-altertable.html

-String functions:
https://www.postgresql.org/docs/9.6/static/functions-string.html

Snippets:

Best Buy decomposition: https://github.com/cs327e-spring2018/snippets/blob/master/decompose_store.sql

Best Buy type conversion: https://github.com/cs327e-spring2018/snippets/blob/master/convert_price_shipping.sql

Additional Notes:

-Create a lab4 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.