

4th Assignment: Redesigning the Airbnb Database

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 2nd Assignment.

Desired Outputs:

Host Table - **host table.sql**:

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

Room table - **room table.sql**:

- Create a new Room table that has the following fields of Listings table: listing_id, accommodates, bathrooms, bedrooms, beds, bed_type, amenities, square_feet, price, weekly_price, monthly_price, security_deposit.
- Delete the fields mentioned above from the Listings table, **except the listing_id field**.
- Enter a suitable Foreign Key that references Listings table.\

Price Table - **price_table.sql**

- Create a new Price table that has the following fields of Listings table: listing_id, price, weekly_price, monthly_price, security_deposit, cleaning_fee, guests_included, extra_people, minimum_nights, maximum_nights, minimum_minimum_nights, maximum_minimum_nights, minimum_maximum_nights, maximum_maximum_nights, minimum_nights_avg_ntm, maximum_nights_avg_ntm.
- Change the decimal number and money field types to a new suitable type (you may need to remove some symbols, hint: replace() function).
- Delete the fields mentioned above from the Listings table, **except the listing_id field**.
- Enter a suitable Foreign Key that references Listings table.

Location Table - **location_table.sql**

- Create a new Location table that has the following fields of Listings table: listing_id, street, neighbourhood, neighbourhood_cleansed, city, state, zipcode, market, smart_location, country_code, country, latitude, longitude, is_location_exact.
- Delete the fields mentioned above from the Listings table, **except the listing_id field**.
- Enter a suitable Foreign Key that references Listings table.
- Delete the association of the listing table with neighborhood table and relate properly the location table with the neighborhood table.

Calendar Table - **calendar.sql**

- In a similar manner, remove the symbols needed so ALTER statement is allowed in decimal number field types for the price, adjustable_price fields. Convert **available** field to Boolean type.

For all Tables

- Rename all arrays from plural to singular. For example, Listing, Review, etc. All these commands must be in a sql file named all_tables.sql.
- Change the ER diagram from 2nd Assignment to match the changes you have made. Not all changes need to be displayed, but a subset of those. Deliver the ER as an image file.

Tools you need:

- Postgres psql ή/και PgAdmin
- [Draw.io](https://draw.io)

Implementation Hints:

- Before you start making changes to each table you should make a backup copy. In case something goes wrong and it is difficult for you to return to the previous state that your table was, you can work with the backup. If you make such copies of tables, however, when you complete the Task please delete the copies.
- Use the CREATE TABLE AS SELECT command to create a table from the results of a SELECT query.
- Use ALTER TABLE ... ADD PRIMARY/ FOREIGN KEY ...
- Use ALTER TABLE ADD/ DROP / RENAME COLUMN ...
- Use ALTER TABLE RENAME TO ...
- Use ALTER TABLE ALTER COLUMN TYPE ... and if necessary USING ...
<http://www.postgresqltutorial.com/postgresql-change-column-type/>
- Use the **split_part()** function to get a String part of a varchar field separated by delimiter `''`.
- Use the **replace()** function to remove a character from a varchar field.

Useful Links:

- CREATE TABLE AS: <https://www.postgresql.org/docs/9.6/static/sql-createtables.html>
- DISTINCT: <https://www.postgresql.org/docs/9.6/static/sql-select.html#SQL-DISTINCT>
- ALTER TABLE: <https://www.postgresql.org/docs/9.6/static/sql-altertable.html>
- String functions: <https://www.postgresql.org/docs/9.6/static/functions-string.html>