Athens University of Economics and Business
Department: Computer Science
Course: Databases
Academic Year: 2019-2020
Instructor: Vasilis Vassalos

5th Assignment: Normalization, Joins and Group By Queries

Goal:

We will continue with more modifications in order to normalize our database, and then run some GROUP BY queries.

Input:

Airbnb database in 4th Assignment state.

Output:

Part A

- Notice that the values of the amenities field in the Room table, are not atomic.
- In order for the database to be normalized, create a new table that is called Amenity with fields called amenity_id and amenity_name. In amenity_name field, save all the unique atomic values of the amenities field after they have been modified (Use string functions of postgres, like regexp_split_to_table(), regexp_replace() etc). Declare amenity_id as an automatic increment field and as a Primary key.
- Take the necessary steps to relate Room table with Amenity. These actions may involve creation of new fields, setting Primary/Foreign Keys and even creating a new table.
- Delete amenities field from Room table.
- Update the ER Diagram.
- Is your Database in BCNF? If not, find the "bad" functional dependencies.

Write all the configuration commands of the Part A database to a part1.sql file and the answer to the BCNF question in a BCNF.pdf file. ER Diagram must be delivered in a .png file.

Part B

Create 5 Join Queries that are not trivial.

- All queries must use at least 1 join.
- 2 of the queries must use an outer join.
- At least 2 queries must use a WHERE statement.
- At least 2 queries must use a GROUP BY statement.
- At least 1 query must use the geolocation table.
- All tables that were created in the last assignment must be used (Host, Room, Price, Location tables).

All queries must be placed in 1 file called **part2.sql.** Add short descriptions of the queries and output results in comments.
Example:

```
/* Find all rentings from all houses that took place in the 31st of
December 2017
Output: 9663 rows
*/
SELECT listings.id, listings.listing_url, calendar.date_attr
FROM listings
INNER JOIN calendar ON listings.id = calendar.listing_id
WHERE calendar.date_attr = '2017-12-31';
```

Tools you need:

- Draw.io
- Postgres psql or PgAdmin

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.
- Run and test each query on your Airbnb database.
- Ensure that each query is not trivial.

Useful Links

- https://www.postgresql.org/docs/9.3/static/functions-matching.html
- http://www.postgresonline.com/journal/archives/152-Regular-Expressions-in-PostgreSQL.html