

# Lab 5: Normalization and Aggregation

Deadline: Friday, Feb. 23rd at 11:59pm.

## Goals:

The goals for this lab are to continue normalizing the Airbnb schema and write some aggregate queries to explore the data.

## Inputs:

-Airbnb database based on the schema changes made in Lab 4.

## Part 1: Normalization

### Desired Outputs:

- The Calendar.price field converted to a numeric type
- The Host.response\_rate field converted to a numeric type. Hint: remove "%" and update "N/A" to NULL.
- The Host.location field split into 3 fields: Host.city, Host.state, Host.country
- The Host.location field removed
- The Listing.amenities field unnested into new relation Amenity(listing\_id, amenity\_name) such that each amenity is stored individually along with its associated listing id.
- Primary key created on Amenity.listing\_id and Amenity.amenity\_name
- Foreign key created on Amenity.listing\_id
- Listing.amenities field removed
- ER diagram updated to reflect new schema

## **Part 2: Aggregation**

### Desired outputs:

-6 aggregate-group-by queries that satisfy the following minimum criteria:

- 3 queries must have a join of which 1 must be an outer join
- 2 queries must have a having clause
- 2 queries must have a where clause
- 2 queries must have an order by clause

-Queries must span at least 5 tables in the Airbnb database.

-A short comment above each query that explains what the query does in 1-2 sentences.

### **Tools You Need:**

- GitHub
- Cloud SQL for Postgres
- psql client
- Lucidchart

### **Code Organization:**

-The SQL related to the Calendar table changes should be stored in a file named `update_calendar.sql`.

-The SQL related to the Host table changes should be stored in a file named `update_host.sql`.

-The SQL related to the new Amenity table should be stored in a file named `create_amenity.sql`.

-The SQL for the aggregate-group-by queries should be stored in the file `aggregate_queries.sql`.

### **Implementation Hints:**

-Use `regexp_split_to_table()` to split a varchar based on a regular expression and convert each resulting split into its own

record. Find an appropriate delimiter in the value and use "+" to specify 1 or more splits.

-`regexp_replace()` to remove multiple characters from a string based a regular expression. Use "|" to separate each character to be removed (e.g. '{|}|"').

### **Reference Documentation:**

Pattern matching functions:

<https://www.postgresql.org/docs/9.3/static/functions-matching.html>

Postgres & regular expressions:

<http://www.postgresqlonline.com/journal/archives/152-Regular-Expressions-in-PostgreSQL.html>

### **Snippets:**

Best Buy Aggregate Queries: [https://github.com/cs327e-spring2018/snippets/blob/master/aggregate\\_queries.sql](https://github.com/cs327e-spring2018/snippets/blob/master/aggregate_queries.sql)

### **Additional Notes:**

-Create a lab5 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.