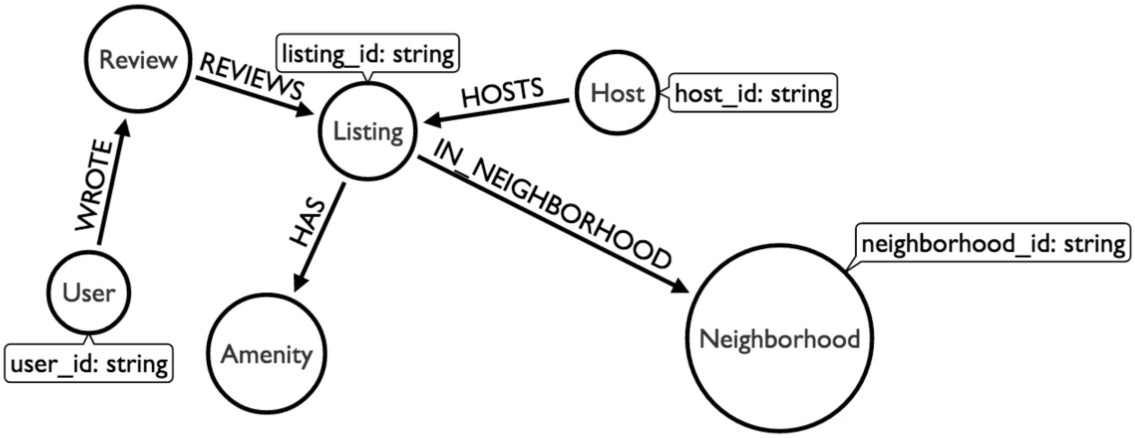
CS 327E Project 7, due Thursday 04/14.

This project makes use of an Airbnb dataset which is modeled as follows:



Your tasks are to load the Airbnb graph into your Neo4j database and then construct some queries on the resulting graph.

To start, open a new terminal window in JupyterLab and download and extract the airbnb assets from GCS:

```
gsutil cp gs://cs327e-open-access/airbnb.zip .
unzip airbnb.zip
```

The extracted folder contains 3 files:

- `listings.csv`
- `reviews.csv`
- `data_load.cypher`

Second, create a new Python Jupyter notebook and name it `project7.ipynb`. All subsequent instructions should be run through your notebook unless otherwise noted.

Before loading any data, be sure to empty your Neo4j database by running this command:

```
!$CONNECT "MATCH (n) DETACH DELETE n"
```

Of course, you'll need to set the `CONNECT` variable before running the above command!

Third, load the airbnb data into Neo4j as follows:

```
!cat /home/jupyter/airbnb/load_data.cypher | {CONNECT} --format plain
```

The script will take a few seconds to run. It should print the number of nodes that it loads for each node type:

```
COUNT(l)
5835
COUNT(a)
42
COUNT(n)
41
COUNT(h)
4633
COUNT(u)
55917
COUNT(r)
62976
```

Verify that all of the data has loaded correctly by retuning the node counts:

```
!{CONNECT} "MATCH (n) RETURN distinct labels(n), count(n)"
```

You should see this output:

```
+------------------------------+
| labels(n)         | count(n) |
+------------------------------+
| ["Listing"]       | 5835     |
| ["Amenity"]       | 42       |
| ["Neighborhood"] | 41       |
| ["Host"]          | 4633     |
| ["User"]          | 55917    |
| ["Review"]        | 62976    |
+------------------------------+
```

Very important: If you need to recreate the graph into your Neo4j database, you will need to drop the constraints and indexes before you can before you can re-run `load_data.cypher`. This is in addition to dropping the relationships and nodes from the database.

Use the following commands to drop the indexes and constraints:

```
DROP INDEX ON :Listing(listing_id);
DROP CONSTRAINT ON (u:User) ASSERT u.user_id IS UNIQUE;
DROP CONSTRAINT ON (n:Neighborhood) ASSERT n.neighborhood_id IS UNIQUE;
DROP CONSTRAINT ON (a:Amenity) ASSERT a.name IS UNIQUE;
DROP CONSTRAINT ON (h:Host) ASSERT h.host_id IS UNIQUE;
```

If you have an SSH tunnel, you can bring up the Neo4j Browser and explore the nodes in the graph and their relationships. This step is not required. If you didn't bring up the Neo4j browser, you can refer to the Airbnb diagram above to see which node labels are in the graph and how they are connected.

You are now ready to construct some cypher queries. Start by sampling the data in the graph by returning any 5 nodes for each unique node label.

Next, translate these questions into cypher and output the results for each one.

Q1. How many hosts are located in "Austin, Texas, United States"?

Q2. Which listings does host_id = "4641823" have? Return the listing name, property_type, price, and availability_365 sorted by price. Limit the result count to 10.

Q3. Which users wrote a review for listing_id = "5293632"? Return the user's id and name sorted alphabetically by the user's name. Limit the result count to 10.

Q4. Which users wrote a review for any listing which has the amenities "Washer / Dryer"? Return the user's id and name sorted alphabetically by name. Limit the result count to 10.

Q5. Which listings have 3 bedrooms and are located in the Clarksville neighborhood? Return the listing name, property_type, price, and availability_365 sorted by price. Limit the result count to 5.

Q6. Which amenities are the most common? Return the name of the amenity and its frequency. Sort the results by count in descending order. Limit the result count to 5.

Q7. Which neighborhoods have the highest number of listings? Return the neighborhood's name and zip code (i.e. neighborhood_id) along with the number of listings they have. Filter out any neighborhoods whose name is NULL from the query results. Sort the results by the number of listings in descending order. Limit the result count to 5.

CS 327E Project 7 Rubric
Due Date: 04/14/21

Download and extract the airbnb assets to your jupyter notebook instance. **-1** no airbnb assets found in Jupyter instance	1
Create a new Python Jupyter notebook named `project7.ipynb`. **-1** incorrect file name	1
Run the data loader script (`load_data.cypher)` to populate the airbnb graph. **-1** script not run or run incorrectly	1
Run a query that returns a count for each node label. **-1** for missing or incorrect counts	1
Run a query that returns any 5 nodes for each unique node type. **-2** for each missing or incorrect queries **-1** for each missing or incorrect output from queries	12
Implement queries Q1 - Q7. **-8** for each missing or incorrect query **-4** for each missing or incorrect output from query	84
`project7.ipynb` pushed to your group's private repo on GitHub. Your project **will not** be graded without this submission.	**Required**

`submission.json` submitted into Canvas. Your project **will not** be graded without this submission. The file should have the following schema:
`{`
`    "commit-id": "your most recent commit ID from GitHub",`
`    "project-id": "your project ID from GCP"`
`}`
Example:
`{`
`    "commit-id": "dab96492ac7d906368ac9c7a17cb0dbd670923d9",`
`    "project-id": "some-project-id"`
`}`
	Required

Total Credit:	**100**