**CIS 550 Group Project Report**

Team member:

Chuqi He

Danmeng Chen

Zhuoyu Ding

Date:

12/13/2017

# Abstract

The topic of our project is guidance of Time Square, which contains the part of food, hotel, entertainment, signage and traffic. It is running by node.js with html files. The sql database is based on MySQL hosted in AWS and no sql database is based on neo4j hosted in GRAPHENE. It will then introduce our database design and application features in detail.

# Introduction

As the crossroads of the world, the Time Square is always attracted people all around the world. It is not only a tourist attraction, but also a commercial financial center. In order to provide a guidance for these people, our group has researched on this topic and present a friendly view of such information which is easily to be queried by users.

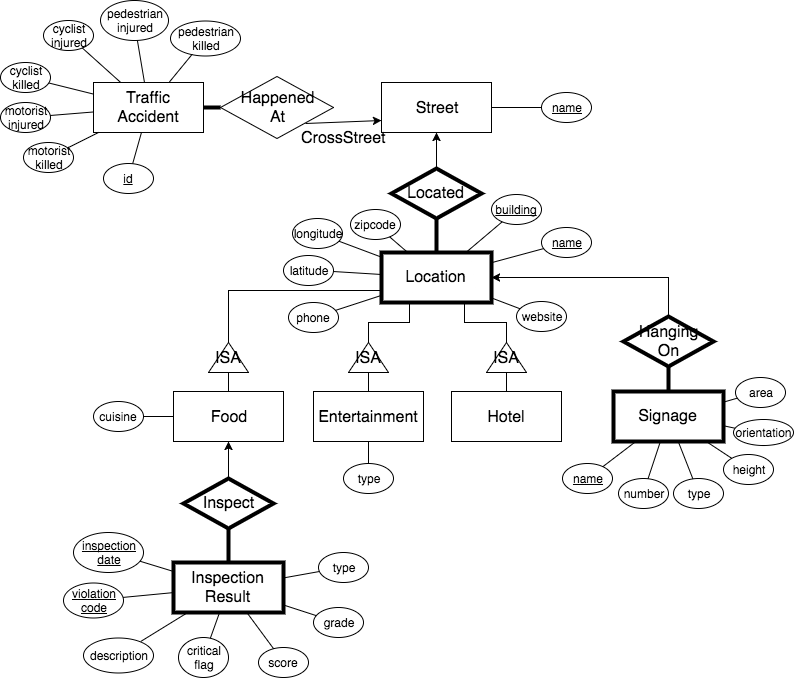
For the tourists, we have integrated the datasets about hotels, food, entertainment and. And for the businessmen, we would like to provide them with the full information about the signage including the location and size. Also, the datasets of all the building in the time square will be included which also provides the illegal records of all the restaurants. It will give a warning to the users who would like to use our datasets to find their restaurant. In addition, it contains the database of traffic, so that the users can refer to the subway by our website.

# Modules and Architecture

This project is running on Node.js with html, including embedded SQL and No SQL. We use AWS’s RDS to host our SQL database based on MySQL. In addition, the part of no SQL graph database is set up by Neo4j and use GRAPHENE as host. This project also associates with the third party: Google map API, Facebook API etc.

Entity relational diagram is used to design the schema, which consist 6 entity sets: traffic accident, street, food, entertainment, hotel and signage, 3 of them (food, entertainment and hotel) are the members of one entity: location. So that they have the same properties. The location is a weak entity of Street: each street has a unique name and each location has a unique building and name. For the food entity, we also researched the database of inspection result which is a weak entity of food, it is identified by the properties of inspection date and violation code. The signage is also a weak entity of location of which primary key is name. In this ER diagram, it can be noticed that there are many one-to-many relationships: one street can hold many locations (Food, Entertainment, Hotel and Signage), and one food can have many inspection results. In addition, we also have the data about traffic which may be completed by No Sql.

* ER Diagram:



* Relational Schema:

Note that the entities Street and Location are decentralized into more specific tables.

**Food**(StreetName, Building, Name, ZipCode, Longitude, Latitude, PhoneNumber, WebSite, Cuisine);

**Entertainment**(StreetName, Building, Name, ZipCode, Longitude, Latitude, PhoneNumber, WebSite, Type);

**Hotel**(StreetName, Building, Name, ZipCode, Longitude, Latitude, PhoneNumber, WebSite);

**Signage**(StreetName, Building, BuildingName, Name, NumberOfScreens, Height, Type, Orientation, Area);

**InspectionResult**(StreetName, Building, Name, InspectionDate, ViolationCode, Description, CriticalFlag, Score, Grade, Type);

**TrafficAccident**(Id, OnStreet, CrossStreet, PedestrianInjured, PedestrianKilled, CyclistInjured, CyclistKilled, MotoristInjured, MotoristKilled);

**SubwayEntrance**(Id, CrossStreet1, CrossStreet2, Corner, Longitude, Latitude);

* Neo4j graph database:

The data about traffic is set up by NoSQL database. We use the dataset of subway entrance to form a neo4j graph database. There are two kinds of nodes: subway line and subway station then there is an edge (800+ nodes and 1000+ edges).

For this part, GRAPHENE worked as host. For creating the databases, we connected to GRAPHENE using python’s connector and sent my neo4j queries to create all the nodes and edges.

# Data Instance Used

The database used in this project are all comes from the website named NYC OpenData: <https://opendata.cityofnewyork.us/>, which contains 5 main part: signage, entertainment, hotel, food and traffic. The following table lists the database and the relevant application in detail:

1. Time Square Signage

(<https://data.cityofnewyork.us/Business/Times-Square-Signage/6bzx-emuu>)

The database provides the basic information about signage in the time square, including its name, location, size and type. It can be designed to give a guidance for the businessman to use.

1. Time Square Entertainment Venues

(<https://data.cityofnewyork.us/Business/Times-Square-Entertainment-Venues/jxdc-hnze>)

The database provides the basic information about entertainment in the time square, including its name, location and type. It can be designed to give a guidance for the traveler to choose.

1. Time Square Hotels

(<https://data.cityofnewyork.us/Business/Times-Square-Hotels/v8qe-fx6p>)

The database provides the basic information about hotel in the time square, including its name, location, postcode, phone and website. It can be designed to give a guidance for the traveler to choose.

1. Time Square Food & Beverage Locations

(<https://data.cityofnewyork.us/Business/Times-Square-Food-Beverage-Locations/kh2m-kcyz>)

The database provides the basic information about food in the time square, including its name, location and cuisine. It can be designed to give a guidance for the traveler to choose.

1. DOHMH New York City Restaurant Inspection Results

(<https://data.cityofnewyork.us/Health/DOHMH-New-York-City-Restaurant-Inspection-Results/xx67-kt59>)

The database provides the inspection results in detail of most restaurants in the New York. We can match them with the restaurants in the Time Square and show them to users.

1. Subway Entrances

(<https://data.cityofnewyork.us/browse?q=subway%20Entrances&provenance=official&anonymous=true&sortBy=relevance>)

The database provides the basic information of subway, including its name and route. It can be designed to give a route navigation for the users to look up.

# Data Cleaning and Import Mechanism

Our datasets are all downloaded as csv files (hotel, food, entertainment, signage, subway) from one website which is intended to provide various databases about New York. And the following work should be cleaning and formatting them into new csv files:

Cleaning:

* Extract useful columns (or fields)
* Remove redundant records
* Remove records with a lot of empty fields

Formatting:

* Make all addresses into a standard format (easy for searching)
* Make sure all fields are in their right type
* Merge detailed records that sharing the same key word into a single record (easy for searching)

# Application introduction

Our design is main comprised of 6 pages: one homepage and 5 parallel pages (food, hotel, entertainment, hotel, signage and subway).

## Homepage:

1. Search a specific name:

There is a search bar in the homepage in which we can enter the keyword of the name you want to search. In this Lsearch query, we merge the datasets of food, entertainment, hotel and signage. If you enter the keyword of the objective location, it will show all relevant results and also tell you what type it is (food or hotel, etc). For the result, you can access to the website of each location if the url is provided.

1. Google map API:

Our page was designed a section of Google map in the bottom. There is a button of “Google maps” in each search result which can be turned you to the google map and show the location in detail.

1. Facebook API:

There is a Facebook log in button in the top right corner. There will be a link connecting to Facebook for user to log in and then get their head portrait and name in our homepage.

1. Share, print, and e-mail:

There is a share button in the top right corner. By clicking on it, it will bring the user to the bottom page. It provides various icons of social platforms like ins, twitter, google etc. The users can share our website by click on these icons. In addition, there are also icons which directed to print page or e-mail page.

## Food page:

1. Search for your restaurant

The search is based on the location and cuisine. In this page, we list all the street and all the cuisine of restaurants in the time square. User can search for their restaurant based on their need and preference.

1. Show inspection results

In the search results, there is a link bring users to see if this restaurant has inspection results. If so, it will show it in detail. So that users can choose their restaurant better.

1. Get route from google map or link to subway

In the search results, there is a link bring users to the google map routing. It will automatically enter the location name and street, so that users can plan a route towards it.

In addition, we also give a link to our subway page. It will generate the latitude and longitude of you location automaticly.

## Hotel page:

1. Search for your hotel

The search is based on the location and number of restaurants and entertainments nearby. User can search for their hotel based on their need.

1. Show the restaurants and entertainments

The search results will show the number of restaurants and entertainments. Furthermore, by click on the number, the users can access to the list of restaurants and entertainments. Therefore, it provides a detailed guidance to users if that want to live near certain restaurant or theater.

1. Show the distribution of restaurants and entertainments

There will be a button appear after searching, which can be used to show pie chart of the search results. It provides an obvious view for the users to look for which hotel has the most restaurants or entertainments nearby.

## Entertainment page:

1. Search for your entertainment:

The search is based on the location and type. In this page, we list all the street and all the type of entertainment in the time square such as theater and night club. User can search for their restaurant based on their need and preference.

## Signage page:

1. Search for your signage:

The search is based on the location and size. User can search for their restaurant based on their need and preference.

## Subway page:

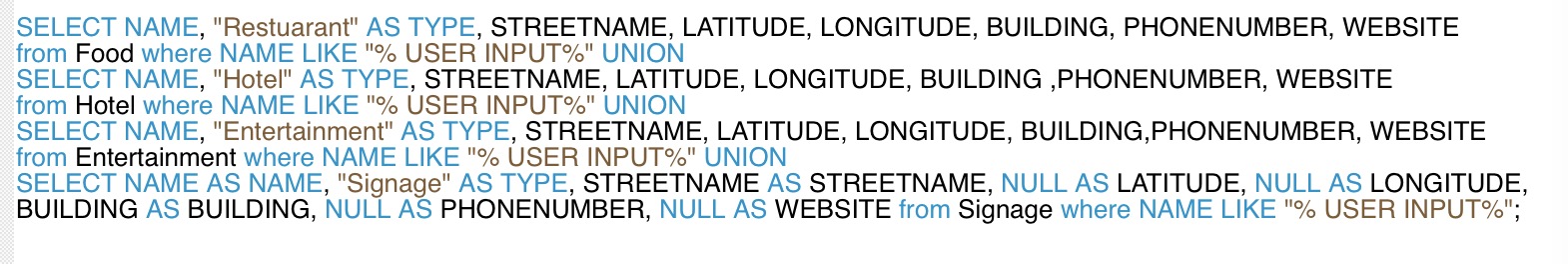
1. Plan route:

In this page, it can locate user’s current location (in longitude and latitude) and get the location information using queries to other datasets (hotel, food etc). After getting the source and destination information as inputs, can tell the user all the possible ways riding subways starting at the source and ending at the destination.

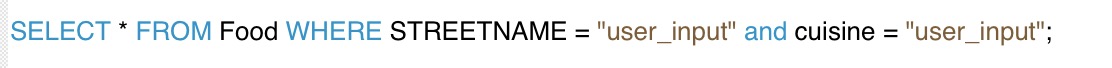
# Query and optimization

1. KEY WORD SEARCH QUERY

In these query, I union the 4 table of Food, Hotel, Entertainment and Signage together and also do the projection and selection according to the key word user input



1. Food Search Query



1. Restaurant Inspection Result Query

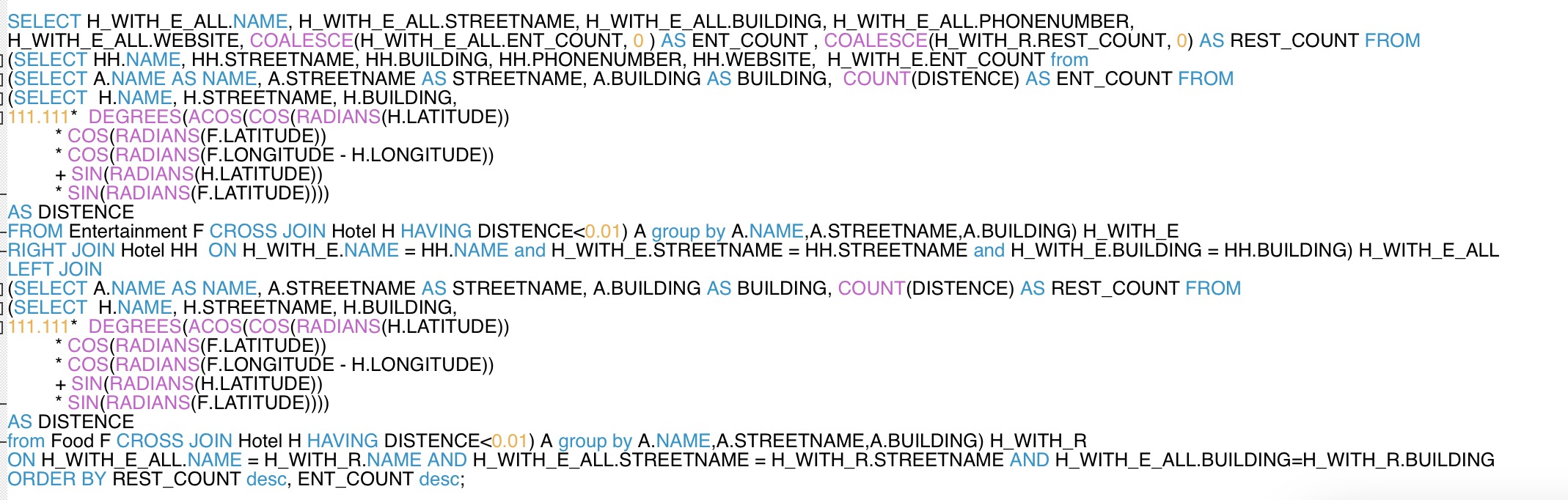
/Users/katiechen/Library/Containers/com.tencent.xinWeChat/Data/Library/Application Support/com.tencent.xinWeChat/2.0b4.0.9/adc8037f36056bde871680042b31714d/Message/MessageTemp/adc8037f36056bde871680042b31714d/Image/1221513198250_.pic.jpg

1. Entertainment Search Query



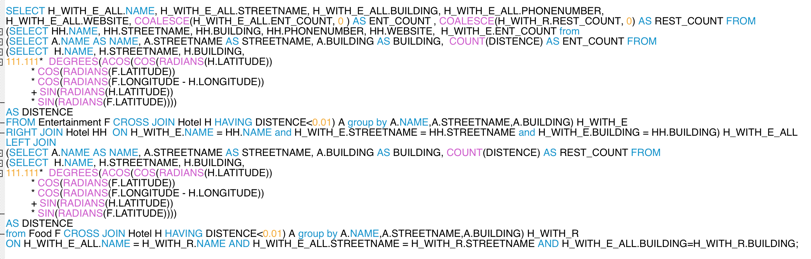
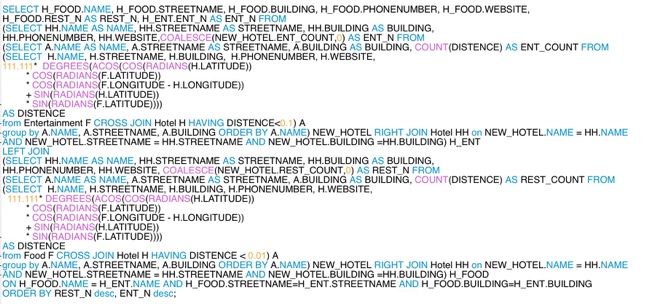
1. Hotel Search Query (Complex Query)

In the hotel search query, we need to return the restaurant amount and entertainment amount nearby each hotel. In order to get the restaurant amount nearby, we need to cross join the Food table and Hotel table and then convert the degree of LATITUDE and LONGITUDE to kilometers distance and also do selection to select all the distance is smaller than 0.1 kilometers and COUNT the restaurant number GROUP BY each hotel. However, some hotel might not have any restaurant nearby 0.1 kilometers. So I also need to right join the origin Hotel table to add back the hotels that has no restaurant nearby. So right now we get the full table of hotels with restaurant number nearby. Then we do the same things with Entertainment table and Hotel table. At last, we just join these two tables and user COALESCE to replace all the NULL with 0.



Query Optimization:

Before: After

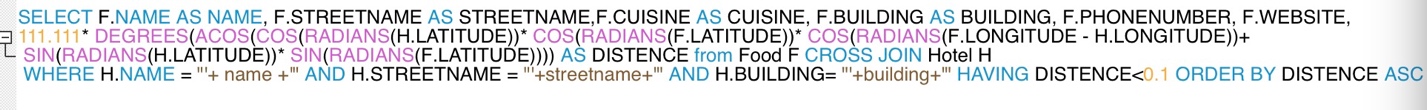


Before I optimize the query, I did some projection and selection at last. After I select and COUNT the restaurant number and entertainment number of each hotel. I right join the origin Hotel table to add back the hotels that has no restaurant nearby and entertainment nearby twice. I also did the COALESCE twice during right join the origin table.

After I optimize the query, I did some projection and selection at the beginning so we drop some useless columns and useless rows at the beginning. Moreover, because after I COUNT restaurant number and add back the hotels that has no restaurant nearby through right join the origin Hotel table, I already get the full hotel with restaurant number table. This table is complete. So when I try to get the table with Entertainment number, I don’t need to right join the origin Hotel table again. I just need to LEFT JOIN the full table with restaurant number. I did COALESCE at the last step so I only did the COALESCE once.

1. Restaurant Nearby Hotel Query

Cross join Hotel and Food table. Convert the degree of LATITUDE and LONGITUDE to kilometers distance and also do selection to select all the distance is smaller than 0.1 kilometers



1. Entertainment Nearby Hotel Query

Cross join Hotel and Entertainment table. Convert the degree of LATITUDE and LONGITUDE to kilometers distance and also do selection to select all the distance is smaller than 0.1 kilometers

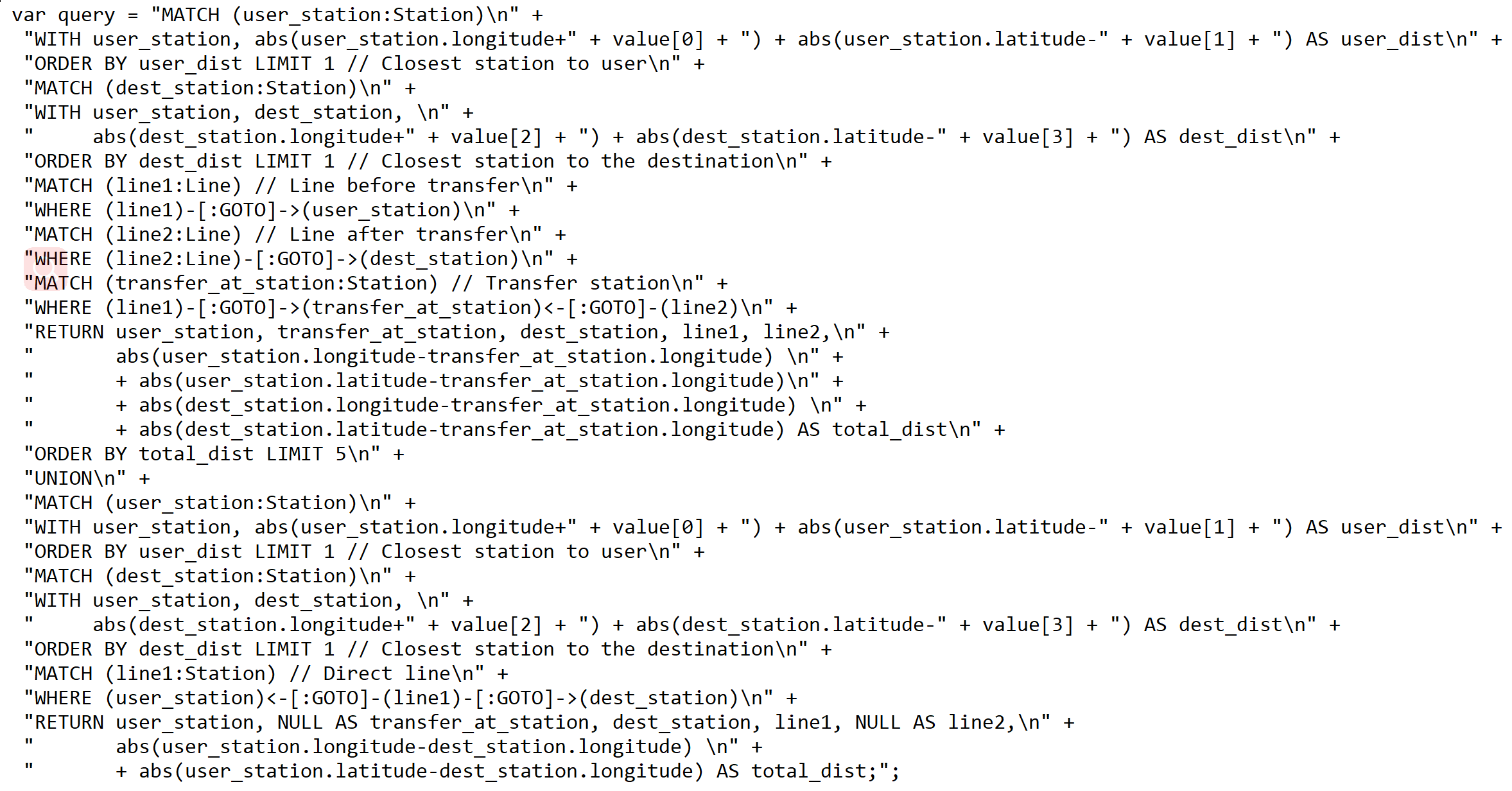


1. Signage Search Query

Because Signage doesn’t have LATITUDE and LONGITUDE, but it has STREETNAME and BUILDING number. So in order to get the longitude and latitude of Signage, we union the Food, Hotel and Entertainment tables and get all the LATITUDE AND LONGITUDE according to each STREETNAEM and BUILDING and assign the latitude and longitude to the Signage by LEFT JOIN Signage table.



1. Subway query (No sql)

Given the approximate longitude and latitude of the source and destination, this query returns the direct and one-stop ways of taking subways between source and destination. When writing this query, one difficulty is that among all the one-stop ways returned, there are unreasonable some results such as those detoured results. In order to get rid of those results, I computed the Manhattan distance for each one-stop way and only return the top 5 shortest ways. Also I did not use Euclidean distance as subways in NYC seldom use the shortest path and using Manhattan distance makes more sense.

# Special Features

1. Google map API: users can access to the location from google map directly. It provides a good visualization for users.
2. Share link to various social platforms: users can share, print or e-mail our website just by clicking on certain button, which is convenient and easy.
3. Piechart review: from the piechart of search result, users can get an obvious view of it.
4. Facebook API: it adds fun for users to show their picture and name on the page.

# Potential feature development

In the part of Facebook API, we just get the picture and name from Facebook. Due to the limitation of time, we have not implemented further effect of the login function. In the future, we plan to create a table for each user, and then they can save their favourite items such as restaurant, hotel or even a subway route. There are two ways to implement it: one is save the information in local cookie; the other is creating a new table in AWS. Such improvement will be more friendly to users.

# Member responsibility for project components

|  |  |
| --- | --- |
| Team Member | Division of Work |
| Chuqi He | Data cleaning, data integration , relational schema and subway no sql database development: neo4j query and relevant node.js |
| Danmeng Chen | Data visualization, web development, sql database development: Query data, optimization, validation and relevant node.js |
| Zhuoyu Ding | Data visualization, web development, sql database development: Query data, optimization, validation and relevant node.js |