CMPE 138 Group Project Report

Group 6: Alex Richards, Yu Jung Yeh, Xiao Yi

Introduction

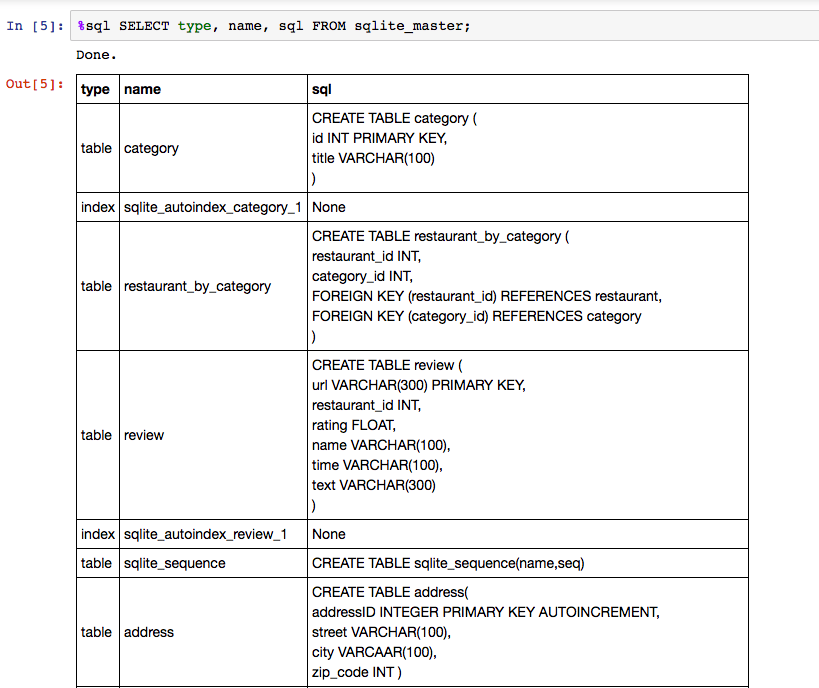
Web application, like Yelp, are the most common path to get information and reviews of restaurants. However, it is no easy to extract the high-end restaurants, like Micheline restaurants, from Yelp. Therefore, the purpose of this project is to develop an application which is specific to Micheline restaurants. To improve the users experience, a database of Micheline restaurants was build up and connected to Yelp web pages. Users can easily get information from the web application and connect to Yelp page for reviews.

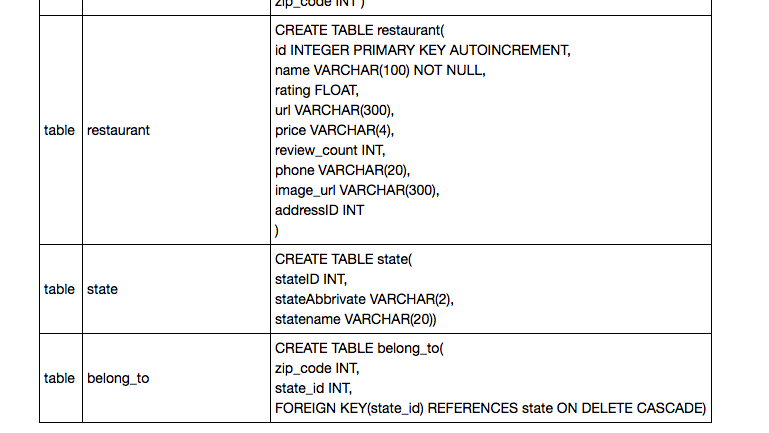
In this project, Yelp’s datasets were analyzed and the data were used to design a database schema that allows end-users to retrieve information about Michelin restaurants. (The user interface was designed very simple and similar to the one provided by Yelp.) The web application interface can provide end-users with the ability to filter search results by location, rating, or name. (more detail……)

Database Design

1. External Interfaces
   * Yelp API
     + Use python to access the Yelp API and build Micheline restaurants database relational tables.
   * Michelin restaurants data
     + The restaurants data were collected from viamichelin.com web page.
2. Software Requirements
   * User-Interface
   * (Photo of web page)
   * Backend model that contains all of tables, restaurant, review, category, address, and states.
   * Middleware can take user’s queries as a request to server and query database for filtered results.
   * (more detail)
3. Database

The dataset, which is maintained by yelp, was designed for our own database schema for this project. We provide an API that can utilize to create a customized client-facing interface. The dataset contains the following entity tables:



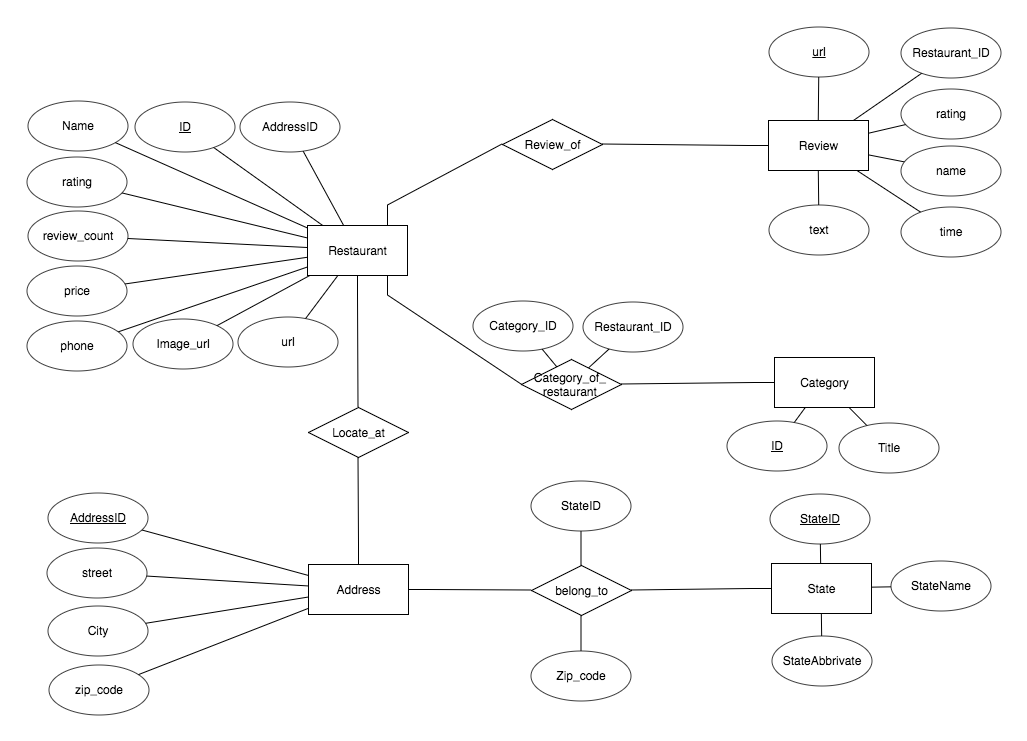


* Restaurant (id : integer, name: string, rating : number, url : string, price: string, review\_count : integer, phone : string, image\_url: string, addressID: integer)
  + There are 384 Michelin restaurants.
  + The primary key is id and set with auto increment so the id can generate a new id number for each new record insert.
* Address (addressID : integer, street : string, city : string, zip\_code: integer)
  + All Michelin restaurants located at 13 cities.
  + The primary key is addressID and set with auto increment so the addressID can generate a new ID number for each new record insert.
* State (stateID : integer, stateAbbrivate : string, statename : string)
  + Only consider Michelin restaurants within three states: Illinois, New York, California.
  + The primary key is stateID.
* Review (url : string, restaurant\_id : integer, rating : number, name : string, time: string, text : string)
  + Collect 1152 reviews record from Yelp.
  + The primary key is url
* Category ( id : integer, title : string)
  + There are 96 different of categories (types of food).
  + The primary key is id.

Relational table

* Restaurant\_by category ( restaurant\_id : integer, category\_id : integer)
  + The foreign keys are restaurant id and category id.
* Belong\_to (zip\_code : integer, state\_id : integer)
  + The foregin key is state\_id and set with “On Delete Cascade”.

1. ER-Diagram



Web Application

The web service was build up with Python Flask. (Flask will need install in advance)

(need more detail)

* The query for user request
  + Search restaurants by name or partial name
    - Either show address information or not
  + Search restaurants by zip\_code and sorting by rating
  + Search restaurants by zip\_code and filter by rating then sorting by price.
  + Search restaurants by city.
    - User can input either upper case or lower case.
    - User can filter the results with rating
    - User can filter the results with review\_count and sorting by rating and price
  + Search restaurants by street or partial information of street.
    - User can input either upper case or lower case.
    - User can filter the results with city name.
  + Search restaurants by state.
    - User can filter the results with rating and review\_count then sorting by rating and price.
    - User can filter the results with category
  + Search restaurants by category.
    - User can filter the results with city name.
  + Search the category (food type) of a restaurant by restaurant name or partial name.
  + (haven’t finish)

(more detail)