**Design of Customer Review Database in Tourism Website**

Wenqing Shi

for CIS 556 - Database Systems:

Instructor: Bogdan Capatina

Winter 2018

# 1.introduction

Tourist industry is flourishing with the development of economic. More and more companies consider to use big data in travelling websites to analyze these users’ information. The users’ personal information and reviews about the attraction are most representative data about the attraction. Database system is an excellent tool to store and use the data.

# 2. Applications

This travelling website review database can support tourism websites, tourism companies and government to make targeted decisions. The applications are shown as follows:

1. For visitors, it supports them to find which and when got to the attractions.
2. For tourism websites, it supports them run specific advertisements on websites according to different users’ preferences.
3. For tourism companies, it supports them make more effective tourism products.
4. For governments, it supports them distinguish their tourism industry situation and use more scientific tourism management system.

# 3. Design Process / Principle / Tools

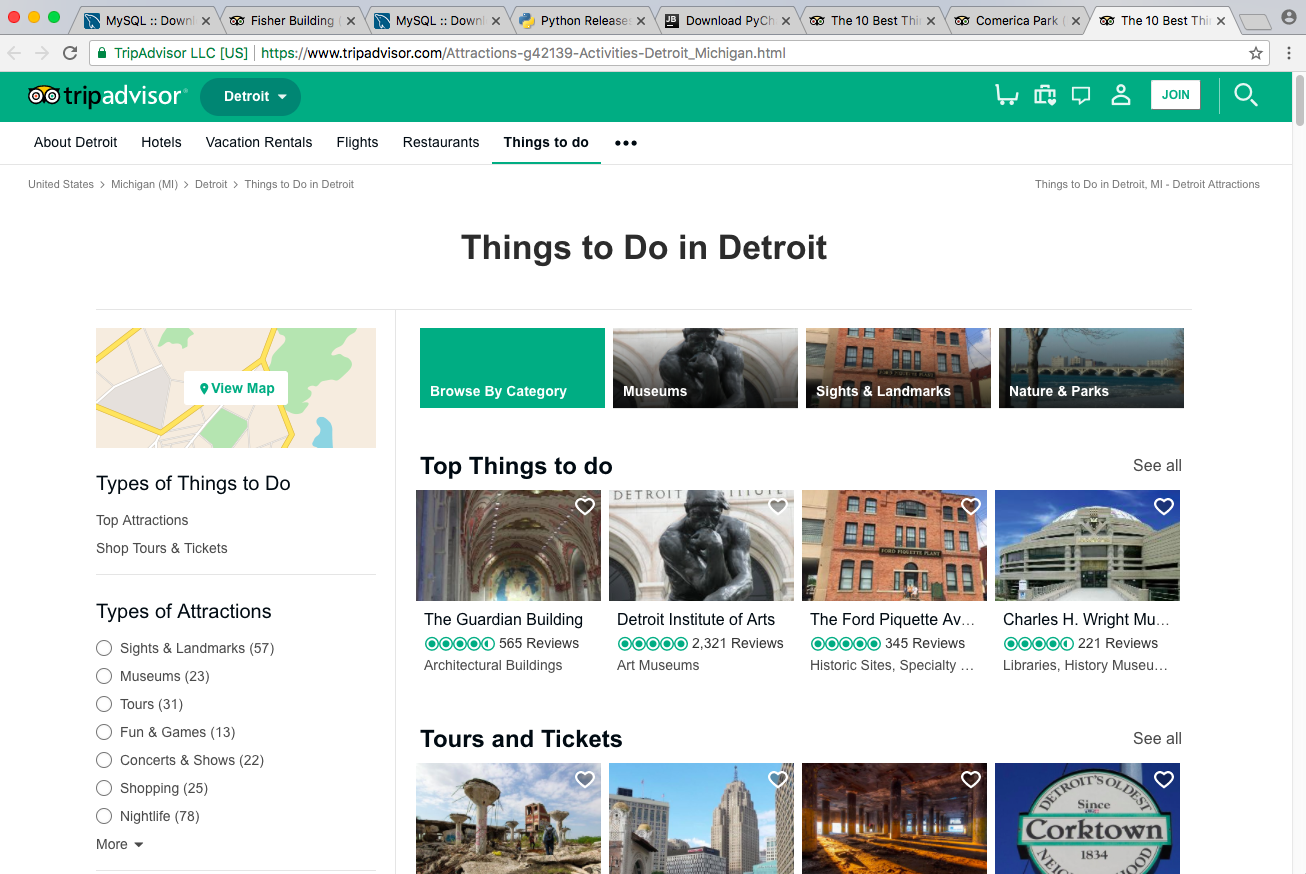
## 3.1 Design Process

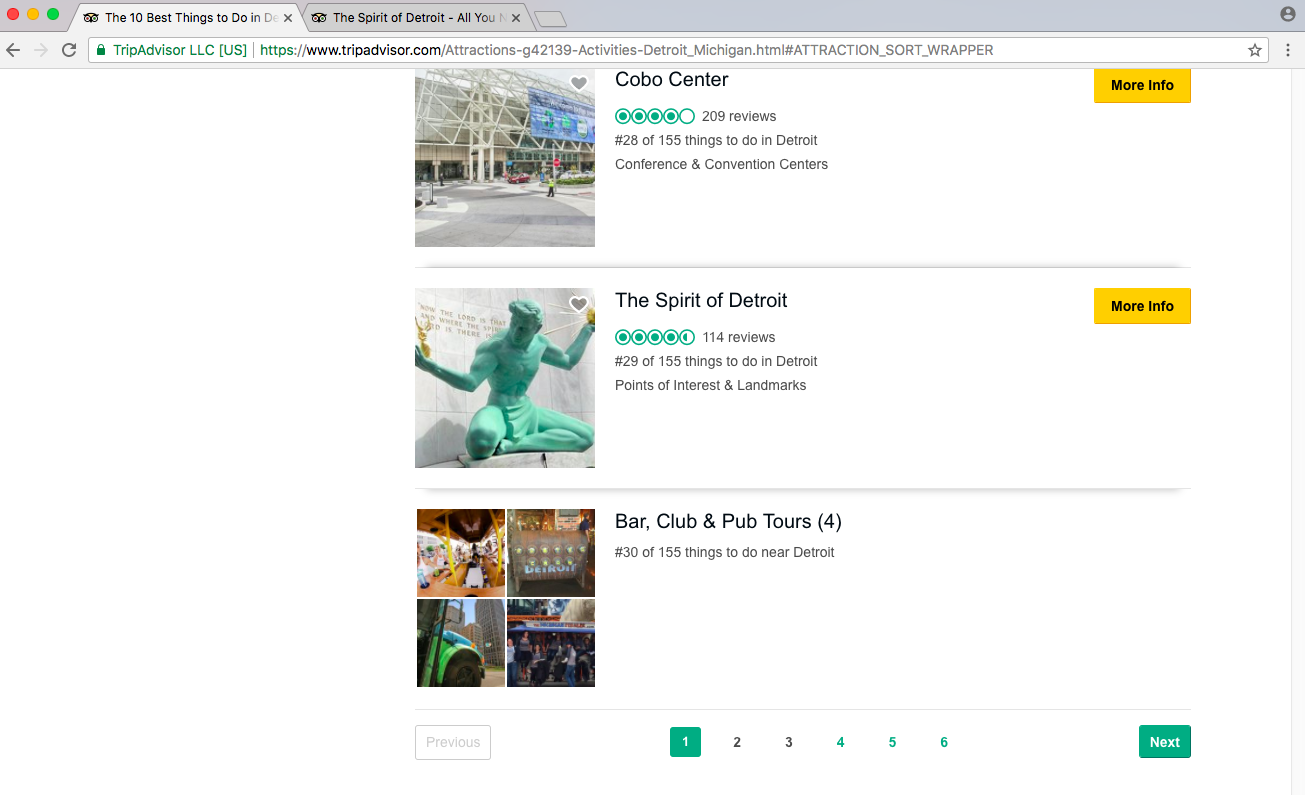
In my project, I choose to record attractions’ and users’ information from a popular travelling website TripAdvisor.com. There are billions of data in this website. Crawlers needs much time to grab information, to spend more time on designing not running code, I just choose Detroit in Michigan as the source of my data. The website link is (“Top things to do”): <https://www.tripadvisor.com/Attractions-g42139-Activities-Detroit_Michigan.html>.

First, I need to grab the page’s link in the website. Then the web crawlers grab attractions in the page one by one. In each attraction, I need attraction’s and users’ review data.

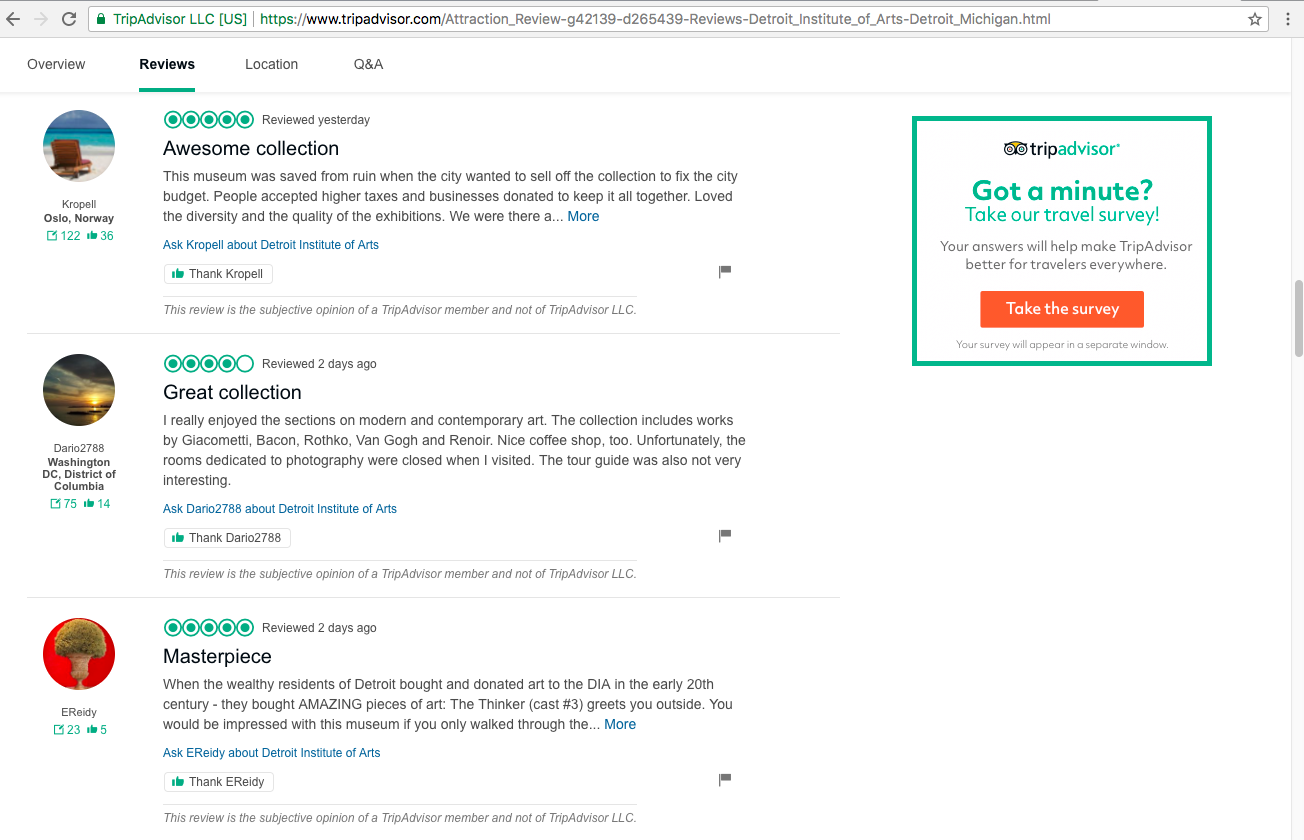
I write web crawlers by python to grab the top 115 attractions’ information and reviews in Detroit. In each attraction’s website, I grab top 9121 users’ personal information and 12629 reviews. Meanwhile, every attraction has nearby hotels, nearby restaurants and nearby attractions. All the information can support us to analyze whether attraction’s nearby infrastructures have an effect on attraction’s rating. Then I grab these information in my database.

In a word, my database ‘SceneryReview’ has 6 relations: have, upload, have, have, have, have. And 7 entities: Cities, Attractions, Users, Reviews, Nearby\_hotels, Nearby\_restaurants, Nearby\_attractions.



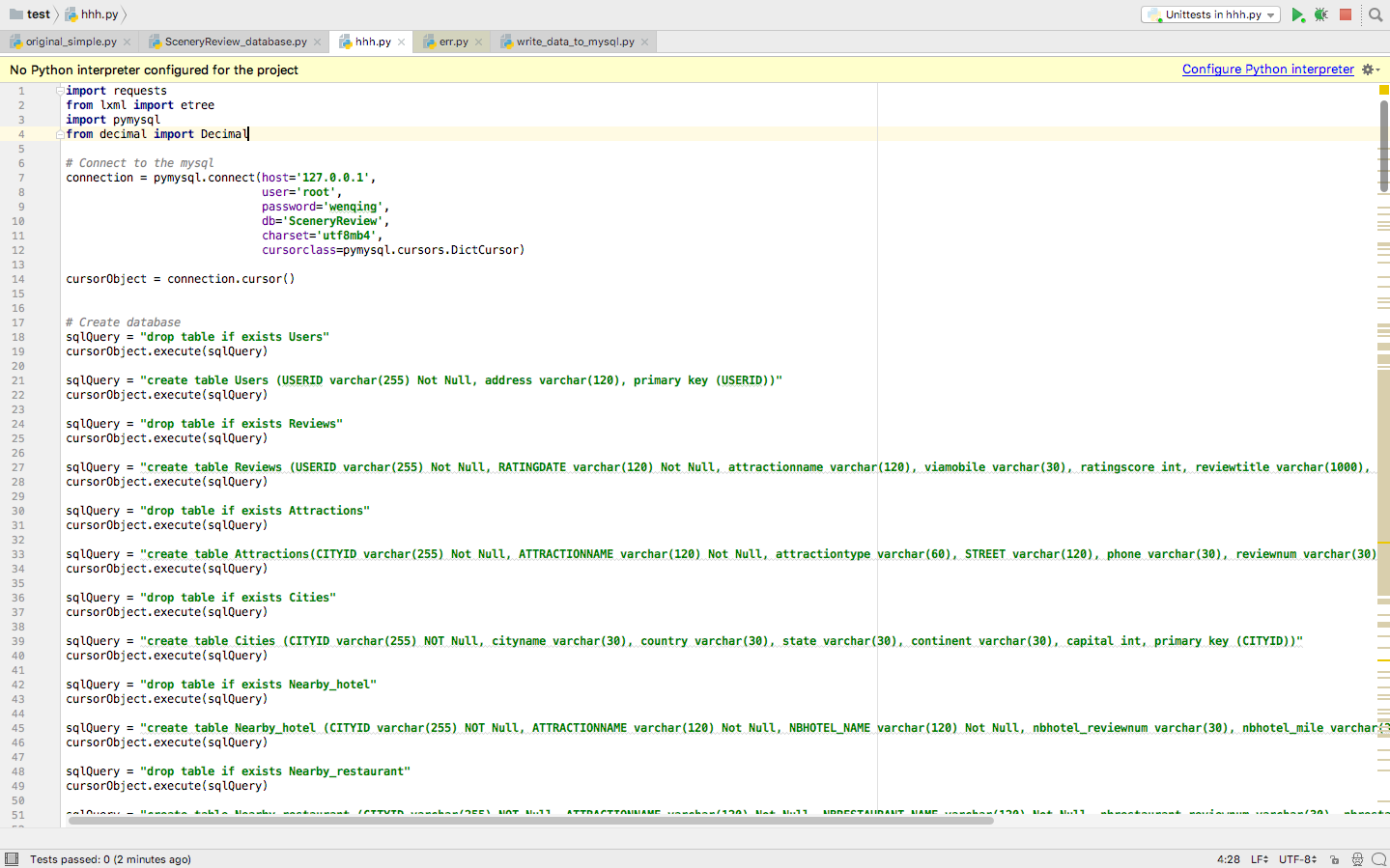




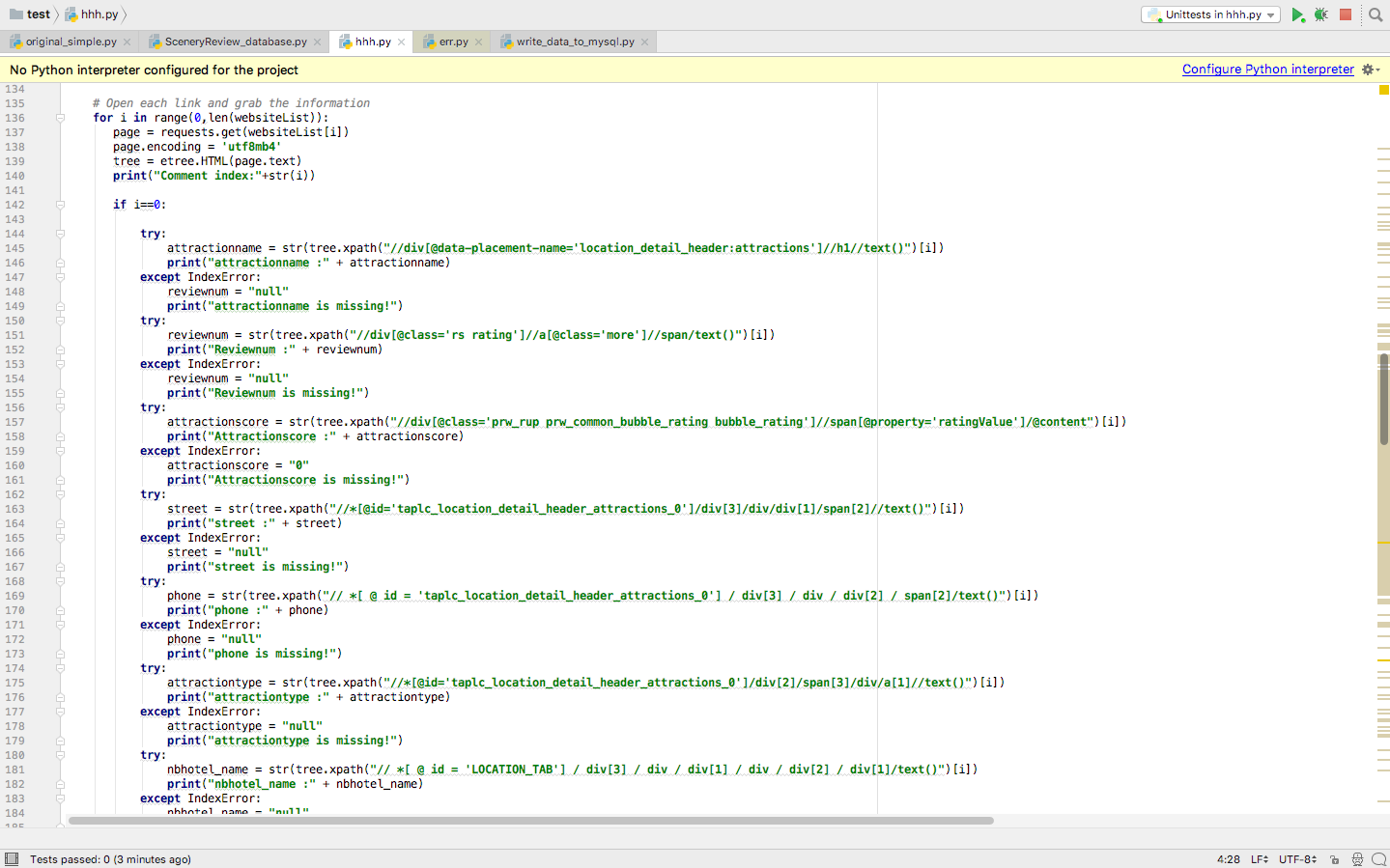


## 3.2 Design Principle

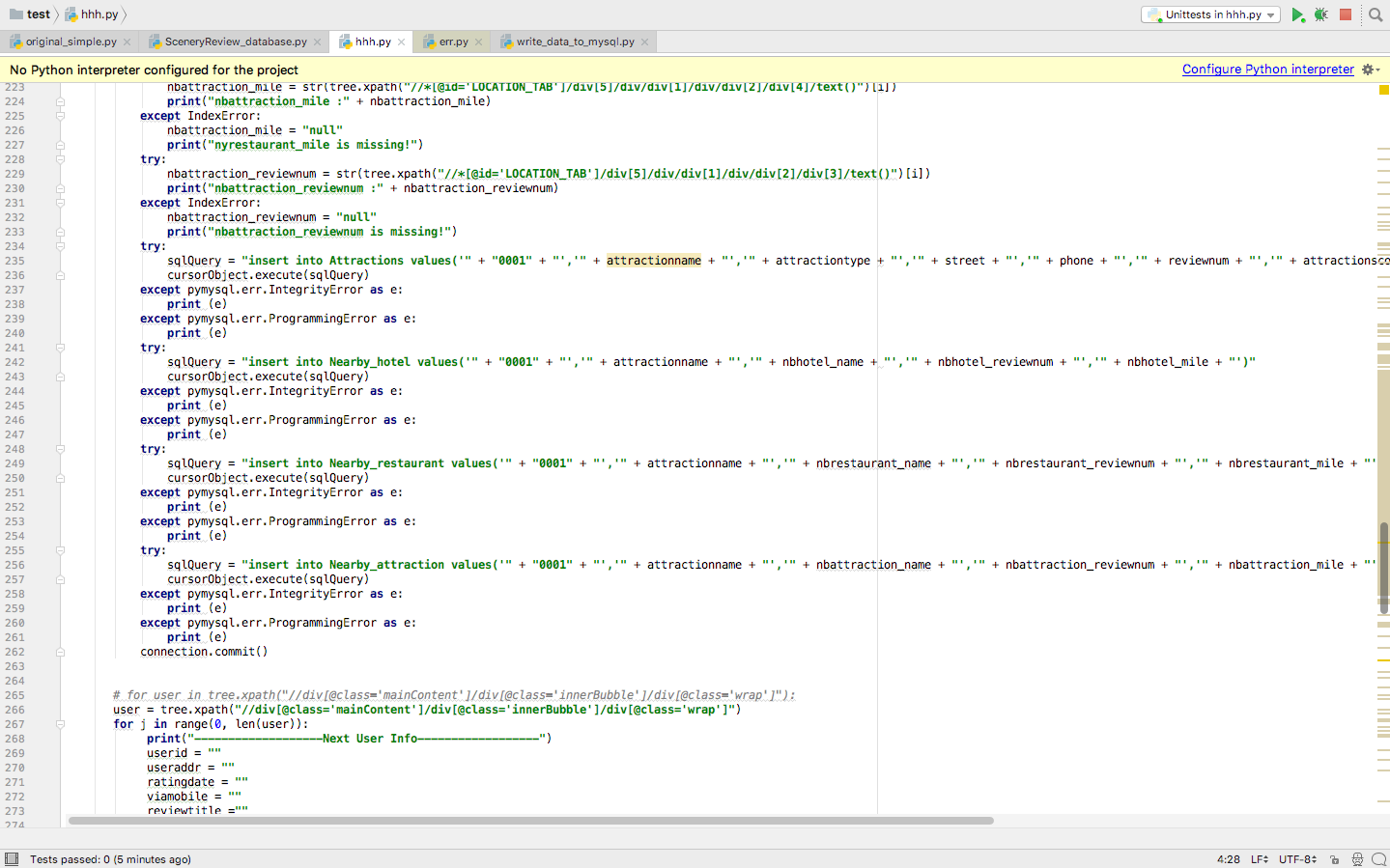
1. Create database by Python.



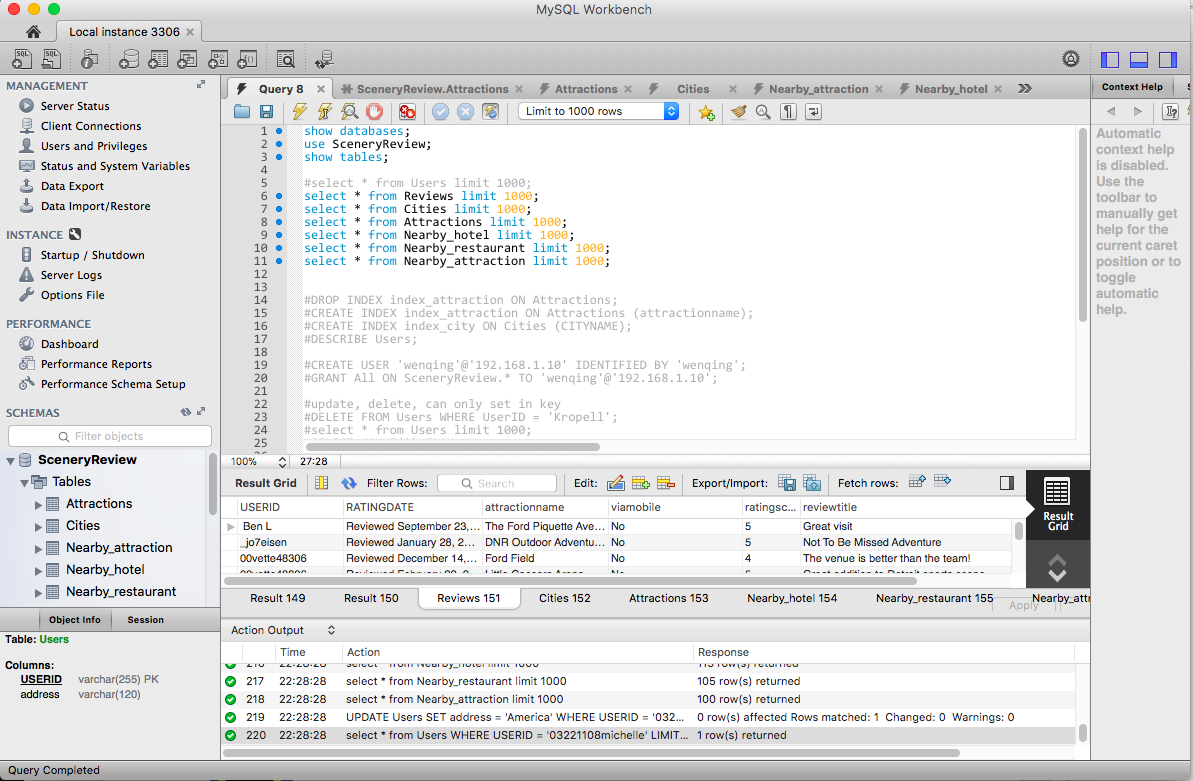
1. Write web crawlers to grab website data by Python.



1. Insert data into tables by Python.



1. Use MySQL Workbench as front end to implement functions. Though PyCharm can implement the same functions as MySQL, MySQL Workbench is a free and easy-used software. It is a suitable front end to database users.



## 3.3 Tools/Languages:

Python

PyCharm (IDE) for Mac



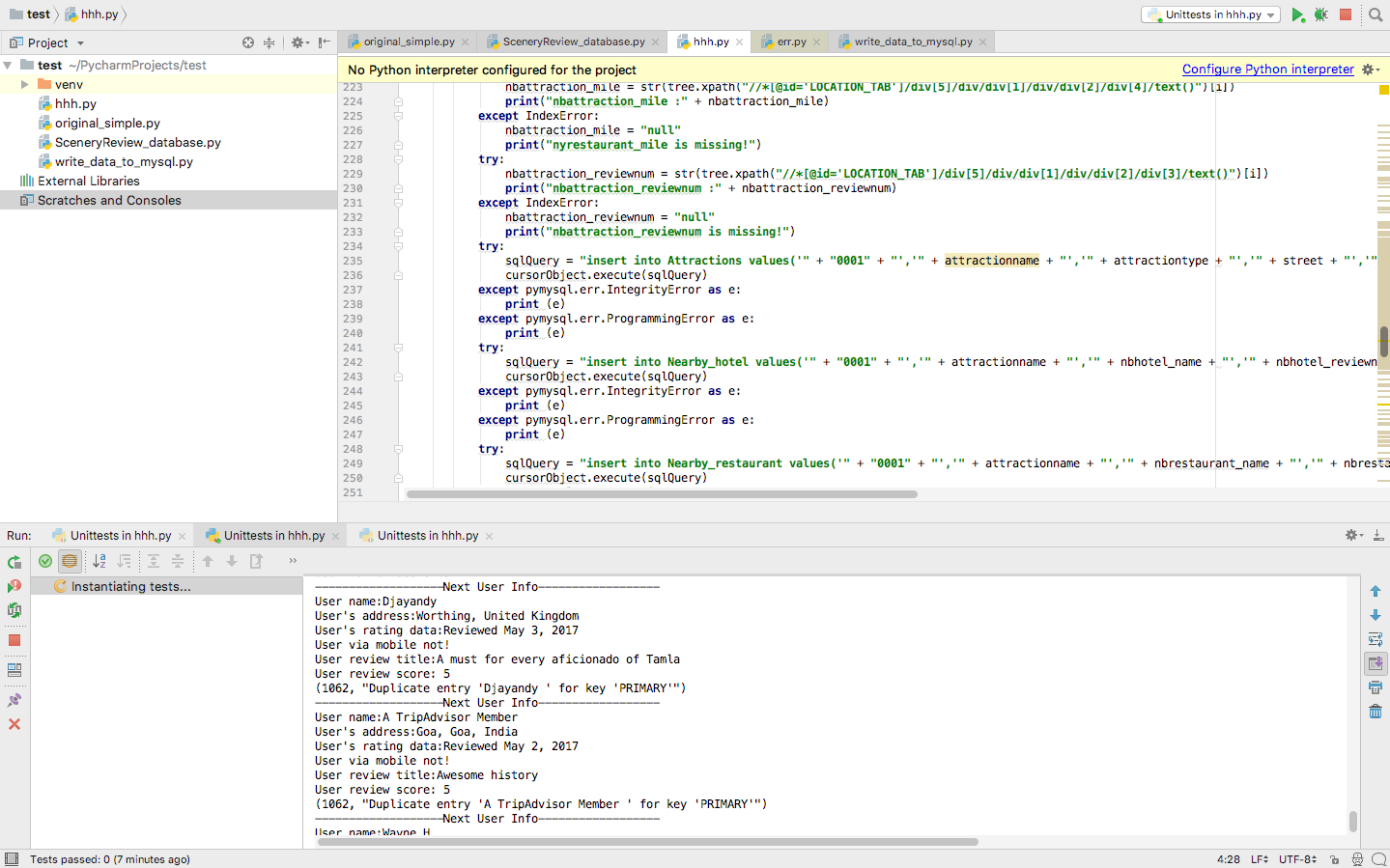
MySQL for Mac

MySQL Workbench for Mac



# 4. Functions

1) Automatic update online data at real time.

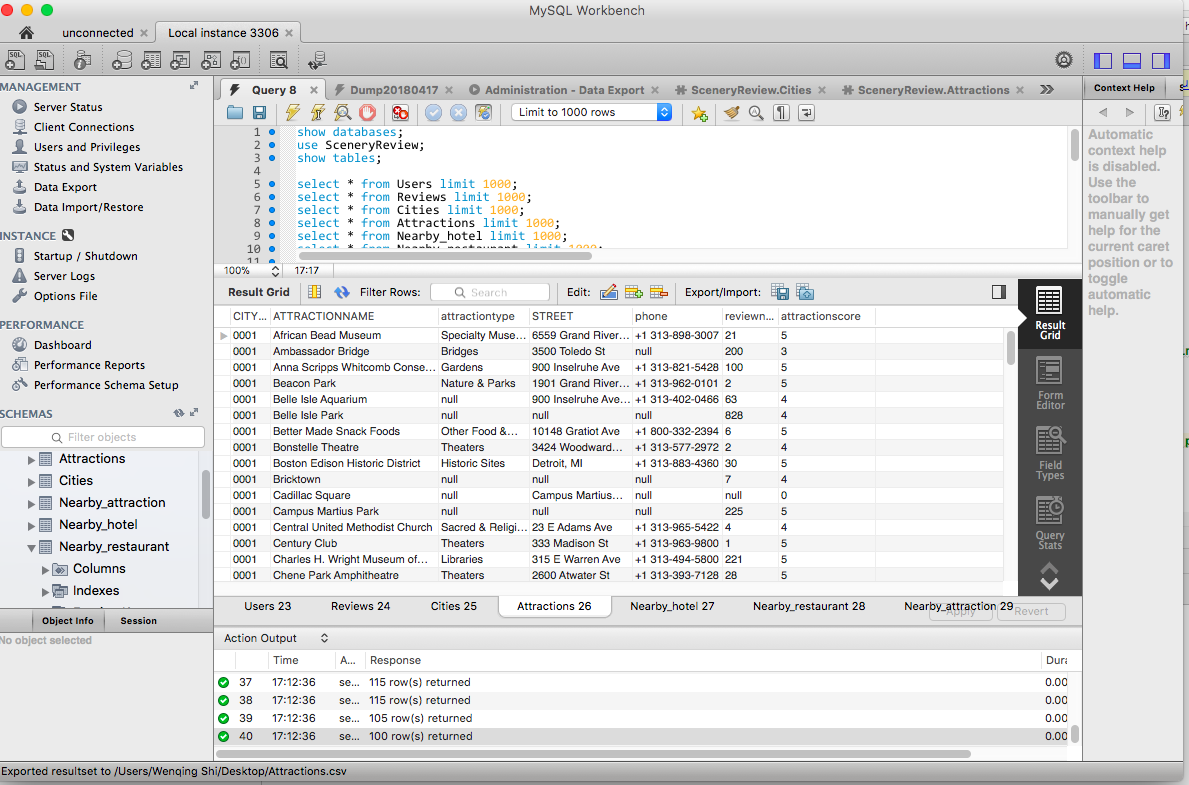


2) Query/Select:

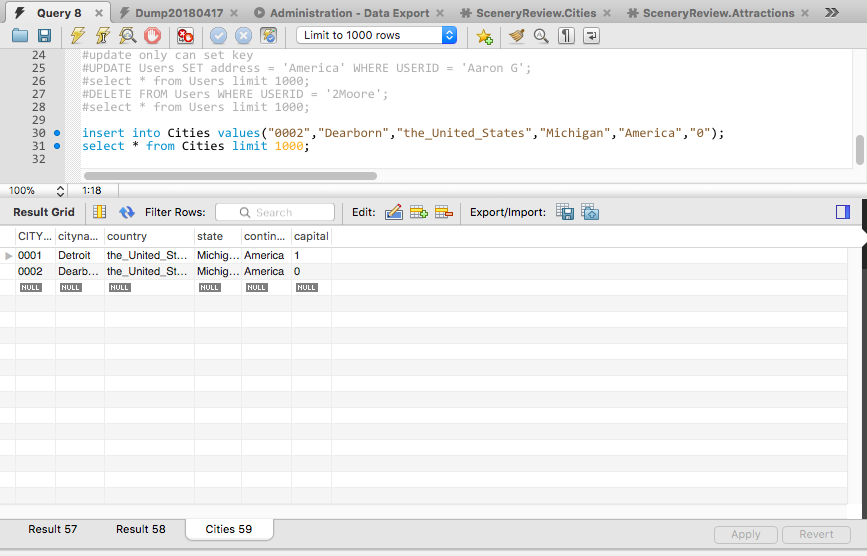
a. Query/Select any user’s ID and address, including their reviews’ records.

b. Query/Select any attraction’s basic introduction, reviews and nearby infrastructures.

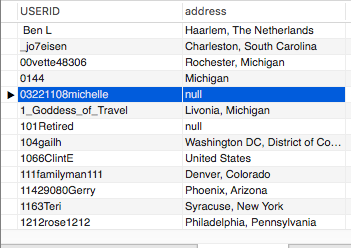
c. Query/Select any cities’ information and attractions.

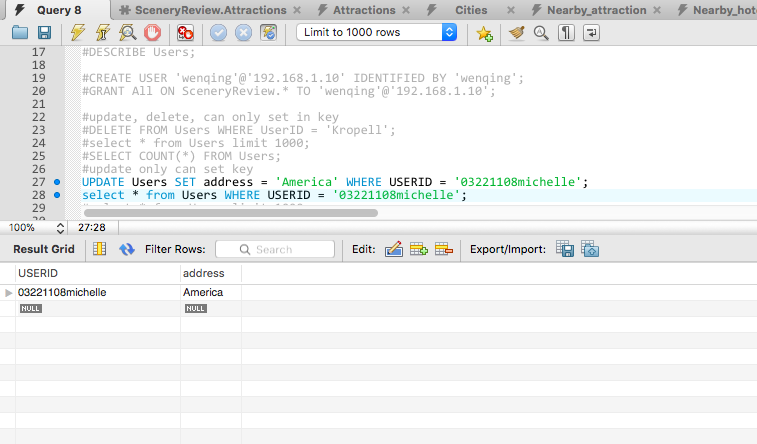


2) Insert: Insert any new record, such as city, attraction, review.

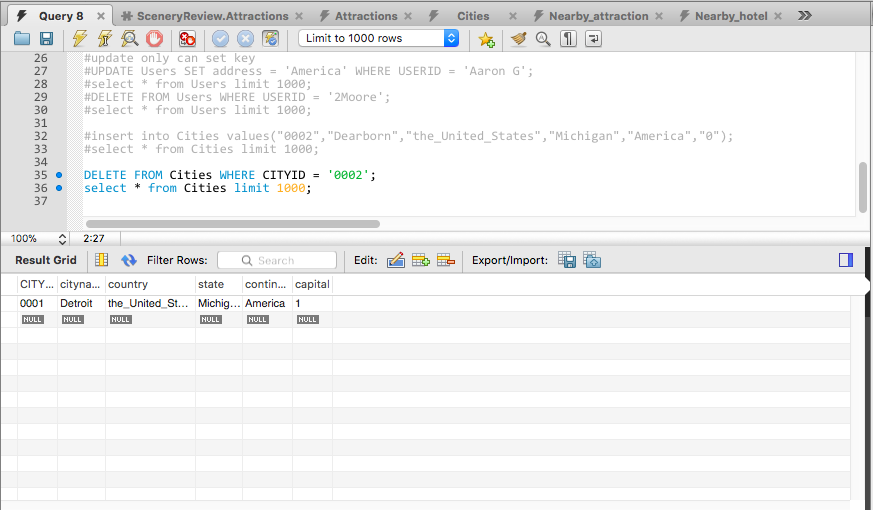


3) Alter/Update: Alter and update any record we need.

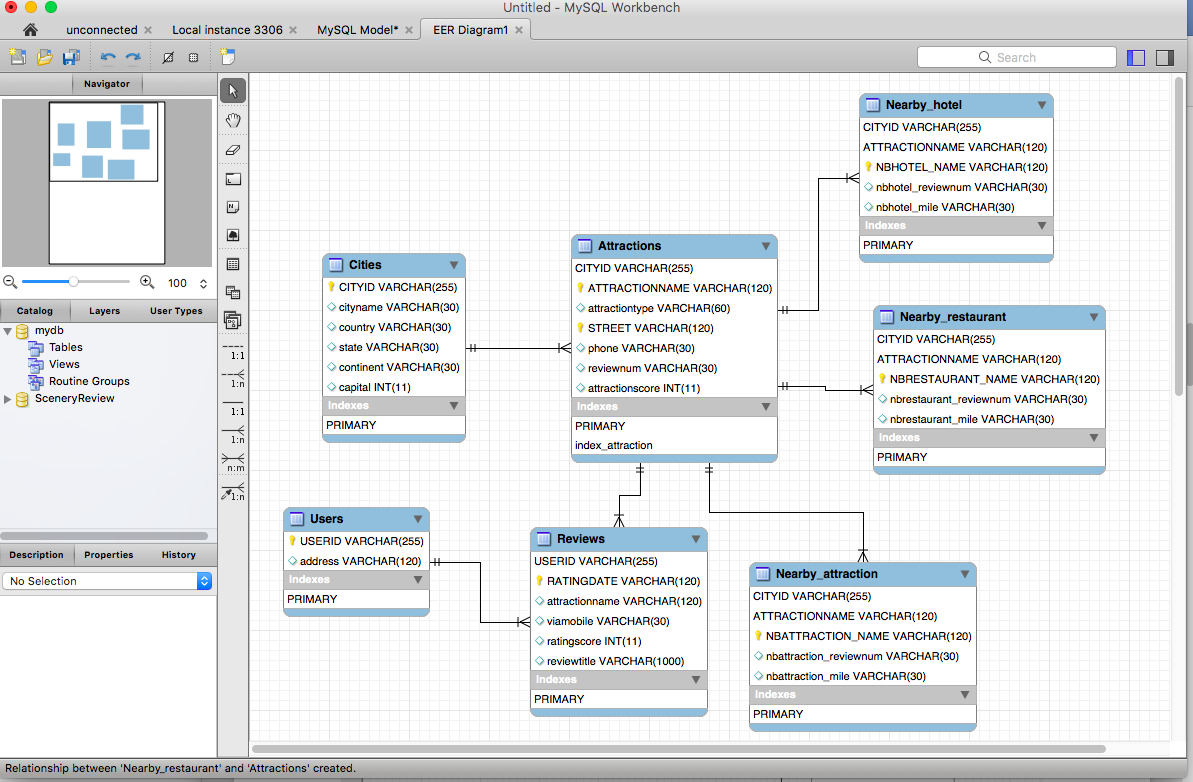




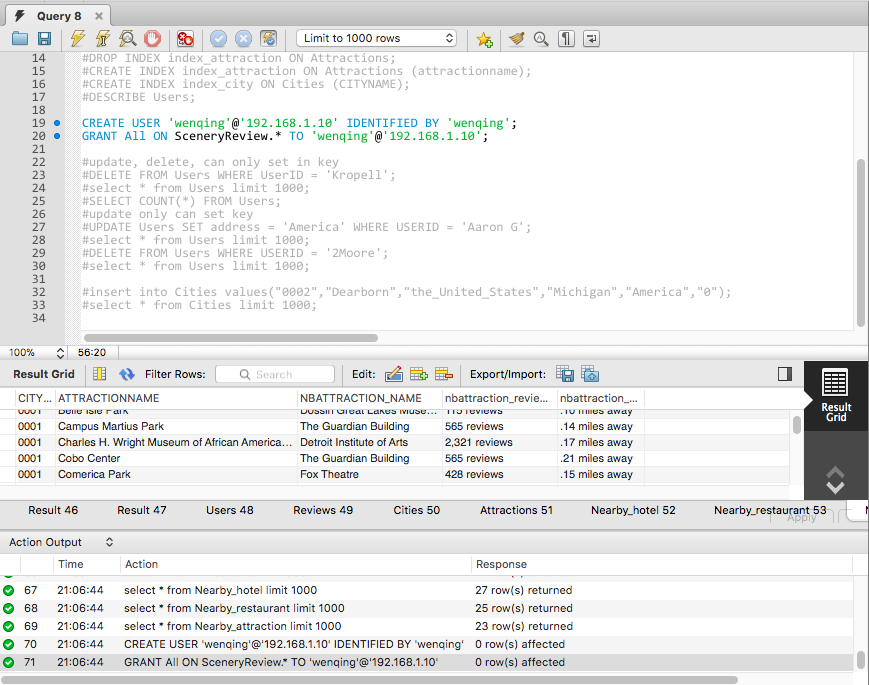
4) Delete: Delete any record do not need.



5) Automatic export EER diagram.



6) Grant.

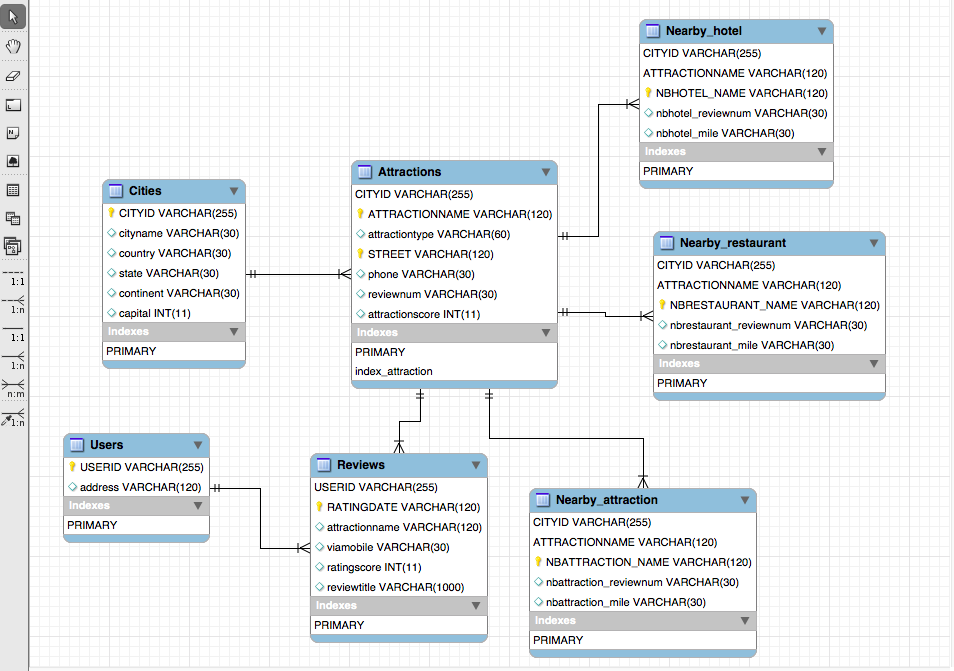


# 5. Database Design

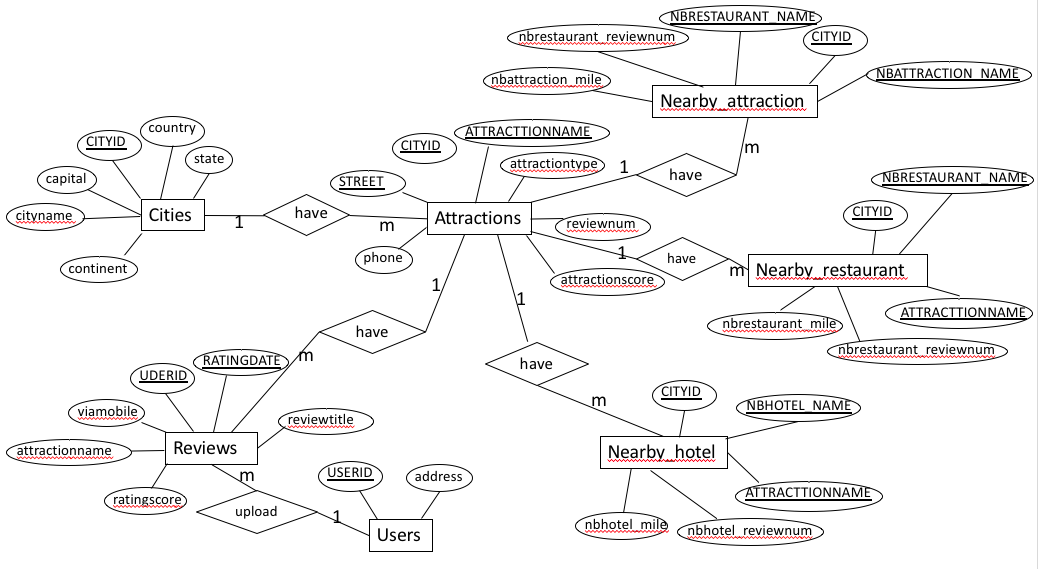
## 5.1 Assumptions

1. In table ‘Cities’, I just grab one city (Detroit), and assume the CITYID = 0001. Different database programmer can name this attribution as their way. For example, using Hash functions create ID. For research purpose, I name it as ‘0001’.
2. Users upload reviews in the website. The reviews have all kinds of formats, symbols, languages, typing habits. For these unstructured data in website, I did not cost much time in dealing with it.
3. Some information in the website, such as user’s level and visited cities, they are written by Java script. Then I did not grab these data.
4. Index will make the database’s speed lower, but it is convenient for search. I just choose one Index (‘attractionname’) on table Attractions.

## 5.2 EER diagram



## 5.3 ER diagram

****

## 5.4 Schema

Cities(

CITYID varchar(255) NOT Null,

cityname varchar(30),

country varchar(30),

state varchar(30),

continent varchar(30),

capital int,

primary key (CITYID)

);

Attractions;

create table Attractions(

CITYID varchar(255) Not Null,

ATTRACTIONNAME varchar(120) Not Null,

STREET varchar(120) Not Null,

attractiontype varchar(60),

phone varchar(30),

reviewnum int CHECK (reviewnum >=1),

Attractionscore int,

primary key (CITYID, ATTRACTIONNAME, STREET)

);

Users(

USERID varchar(255) Not Null,

address varchar(120),

primary key (USERID)

);

Reviews(

USERID varchar(255) Not Null,

RATINGDATE varchar(120)Not Null,

attractionname varchar(120),

viamobile int,

ratingscore int,

reviewtitle varchar(255),

primary key (USERID, RATINGDATE)

);

nearby\_hotel(

CityID varchar(255) Not Null,

ATTRACTIONNAME varchar(120) Not Null,

NBHOTEL\_NAME varchar(30) Not null,

nbhotel\_reviewnum int,

nbhotel\_mile varchar(30),

primary key (NBHOTELNAME, CITYID, ATTRACTIONNAME)

);

nearby\_restaurant(

CityID varchar(255) Not Null,

ATTRACTIONNAME varchar(120) Not Null,

NBRESTAURANTNAME varchar(120) Not Null,

nbrestaurant\_name int,

nbrestaurant\_mile varchar(30),

primary key (NBRESTAURANTNAME, CITYID, ATTRACTIONNAME)

);

nearby\_attraction(

CityID varchar(255) Not Null,

ATTRACTIONNAME varchar(120) Not Null,

NBATTRACTION\_NAME varchar(120) Not Null,

nbattraction\_reviewnum int,

nbattraction\_mile varchar(30),

primary key (NBATTRACTION\_NAME, CITYID, ATTRACTIONNAME)

);

## 5.5 Tables

1) Cities: city in the website (Detroit in this data sample)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | Description | Type | Key | Nullable | Others |
| CITYID | unique ID num which represent city | varchar(255) | PK | No |  |
| cityname | city’s name | varchar(30) |  | Yes |  |
| country | which country is the city | varchar(30) |  | Yes |  |
| state | which state is the city | varchar(30) |  | Yes |  |
| continent | which continent is the ciy | varchar(30) |  | Yes |  |
| capital | whether the capital is capital city in the state(I is yes, 0 is not) | int |  | Yes |  |

2) Attractions: city’s attractions in the website (Detroit’s attractions in this data sample)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | Description | Type | Key | Nullable | Others |
| CITYID | unique ID num which represent city | varchar(225) | PK | No |  |
| ATTRACTIONNAME | attraction’s name | varchar(120) | PK | No | INDEX |
| STREET | attraction’s address street | varchar(129) | PK | No |  |
| attractiontype | type of the attraction such as museum, library | varchar(60) |  | Yes |  |
| phone | attraction’s phone | varchar(30) |  | Yes |  |
| reviewnum | attraction’s review in website | varchar(30) |  | Yes |  |
| attractionscore | attraction’s score in website | int |  | Yes | CHECK |

3) Users: users that upload reviews in the website

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | Description | Type | Key | Nullable | Others |
| USERID | user’s account ID in website | varchar(225) | PK | No |  |
| address | user’s address in website | varchar(120) |  |  |  |

4) Reviews: users’ reviews about attractions in the website

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | Description | Type | Key | Nullable | Others |
| USERID | user’s account ID in website | varchar(225) | PK | No |  |
| RATINGDATE | when the user upload the review in website | varchar(120) | PK | No |  |
| attractionname | attraction’s name | varchar(60) | PK | Yes |  |
| viamonile | whether user upload the review via mobile | varchar(60) |  | Yes |  |
| ratingscore | the score that user give on website | int |  | Yes |  |
| reviewtitle | the review’s title that user upload on website | varchar(1000) |  | Yes |  |

5) Nearby\_hotel: attraction’s nearby hotels shown in the website

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | Description | Type | Key | Nullable | Others |
| CITYID | unique ID num which represent city | varchar(225) | PK | No |  |
| ATTRACTIONNAME | attraction’s name | varchar(120) | PK | No | INDEX |
| NBHOTEL\_NAME | attraction’s nearby hotel name | varchar(120) | PK | No |  |
| nbhotel\_reviewnum | review number of attraction’s nearby hotel | varchar(30) |  | Yes |  |
| nbhotel\_mile | distance between attraction and attraction’s nearby hotel | varchar(30) |  | Yes |  |

6) Nearby\_restaurant: attraction’s nearby restaurants shown in the website

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | Description | Type | Key | Nullable | Others |
| CITYID | unique ID num which represent city | varchar(225) | PK | No |  |
| ATTRACTIONNAME | attraction’s name | varchar(120) | PK | No |  |
| NBHOTEL\_NAME | attraction’s nearby restaurant name | varchar(120) | PK | No |  |
| nbrestaurant\_reviewnum | review number of attraction’s nearby restaurant | varchar(30) |  | Yes |  |
| nbrestaurant \_mile | distance between attraction and attraction’s nearby restaurant | varchar(30) |  | Yes |  |

7) Nearby\_attraction: attraction’s nearby attractions shown in the website

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | Description | Type | Key | Nullable | Others |
| CITYID | unique ID num which represent city | varchar(225) | PK | No |  |
| ATTRACTIONNAME | attraction’s name | varchar(120) | PK | No | INDEX |
| NBATTRACTION\_NAME | attraction’s nearby attraction name | varchar(120) | PK | No |  |
| nbattraction\_reviewnum | review number of attraction’s nearby attraction | varchar(30) |  | Yes |  |
| nbattraction \_mile | distance between attraction and attraction’s nearby attraction | varchar(30) |  | Yes |  |

# 6. Conclusion

In this project, I design a tourism review system, hope it can support visitors, tourism websites, tourism companies even governments to make sure which attractions to go, to advertise targeted customers, to make better tourism products and make scientific decisions.

I used web crawlers by python to grab information from a tourism website. Including attractions’, users’ information and reviews online. Meanwhile, I create a database ‘SceneryReview’ to record these data by python. MySQL Workbench is a front end for users. There are 6 relations and 7 entities in this database. I also show ER diagram, EER diagram and Schema in my report. The system can update online data at real time automatically. Also, it can provide query, update, grant, delete, alter, select and other functions.

There are also several limitations in my project. First, the reviews have all kinds of formats, symbols, languages, typing habits. The data I grab is not so clean. It will influence the effectiveness of my database. Second, every website has its own script, some data is so hard to grab. There are several meaningful attributes not including in my database. In future, once I want to finish the system, I need to cost much time in dealing with data.