In this exercise you will be required to define and create tables in MySQL, download data from an API using rest calls, and insert them into the tables that are created. We will be using the Tribune Media Services (TMS) API to download data about Movies. The documentation for the API can be reached at <http://developer.tmsapi.com/docs/read/data_v1_1>.

First of all, register for a public plan on this page <https://developer.tmsapi.com/Getting_Started>

And follow the steps, that will generate an API key for accessing the below API’s

1. Get data for movies playing in local theatres in US for a particular zip code and start date:

http://data.tmsapi.com/v1.1/movies/showings?startDate=<start\_date>&zip=<zip\_code>&api\_key=<api\_secret>

2. Get data for movies airing on TV for a particular line up and date and time :

http://data.tmsapi.com/v1.1/movies/airings?lineupId=<line\_up\_id> &startDateTime=<date\_time>&api\_key=<api\_secret>

Corresponding to the 2 APIs listed above design the following 2 tables using SQLAlchemy models

1. A table to store movie data playing in theatres
2. A table to store movie data airing on TV

Mandatory fields for the tables (Rest can be added as per choice):

* Title
* Release year
* Genres
* Description
* Theatre / Channel (Based on the API)

Once you have the necessary models in place, generate the tables in your local MySQL database.

The code would have 2 functions :

A function to get the data from the API’s using the following parameters :

* <api\_secret> : The API key generated while registering
* <zip\_code> : 78701
* <start\_date> : Current date in yyyy-mm-dd format eg. 2020-10-09
* <line\_up\_id> : USA-TX42500-X
* <date\_time> : Current date with time (ISO 8601) eg. 2020-10-09T09:30Z

Another function that would :

1. Group both the Movie lists based on ‘Genre’
2. Combine the movie lists (Theatre and Channel movies) based on the Genres
3. Return the Top 5 Genres with the highest movie count along with the movie details

Read the data from the database and use pandas to join the different data sets into memory and filter and return the data.

Share the code (along with database dump from MySQL) on GitHub.