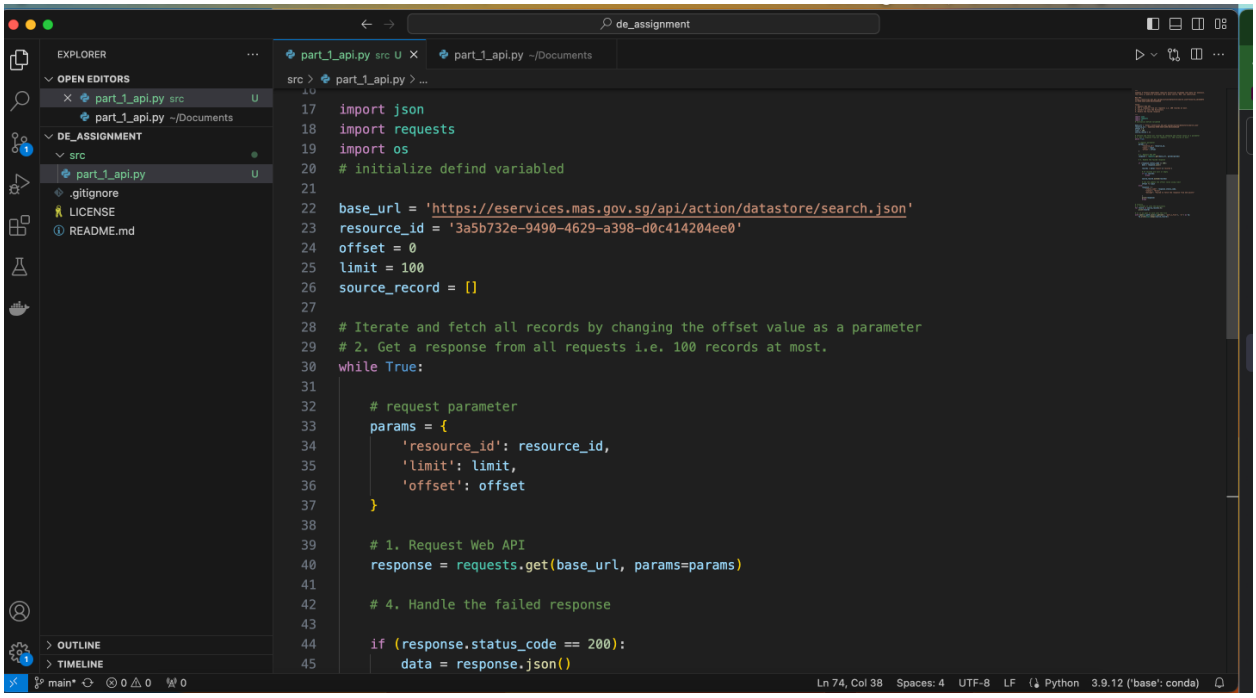
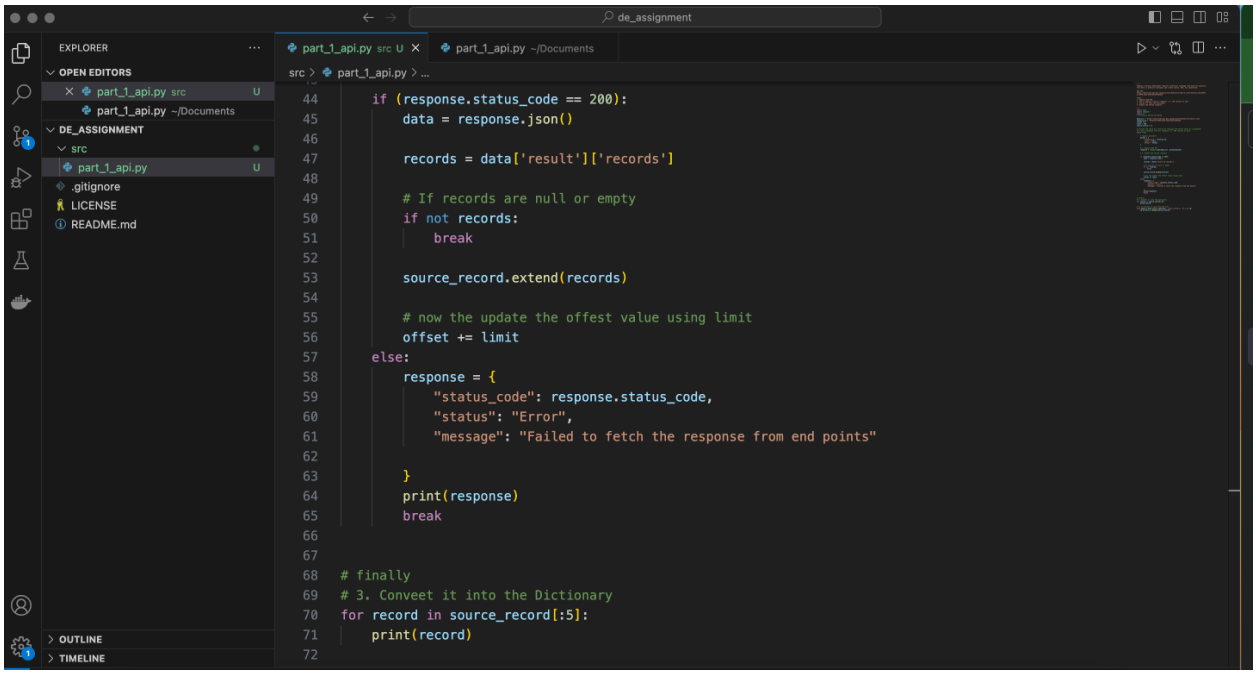
Ajay Kumar Maurya
Data Engineer-Assignment

PART 1: FETCH RECORD AND CONVERT INTO THE DICTIONARY

Project Code (GitHub): https://github.com/ajay9889/de_assignment/



```python
import json
import requests
import os
# initialize defind variabled

base_url = 'https://eservices.mas.gov.sg/api/action/datastore/search.json'
resource_id = '3a5b732e-9490-4629-a398-d0c414204ee0'
offset = 0
limit = 100
source_record = []

# Iterate and fetch all records by changing the offset value as a parameter
# 2. Get a response from all requests i.e. 100 records at most.
while True:

    # request parameter
    params = {
        'resource_id': resource_id,
        'limit': limit,
        'offset': offset
    }

    # 1. Request Web API
    response = requests.get(base_url, params=params)

    # 4. Handle the failed response

    if (response.status_code == 200):
        data = response.json()
```

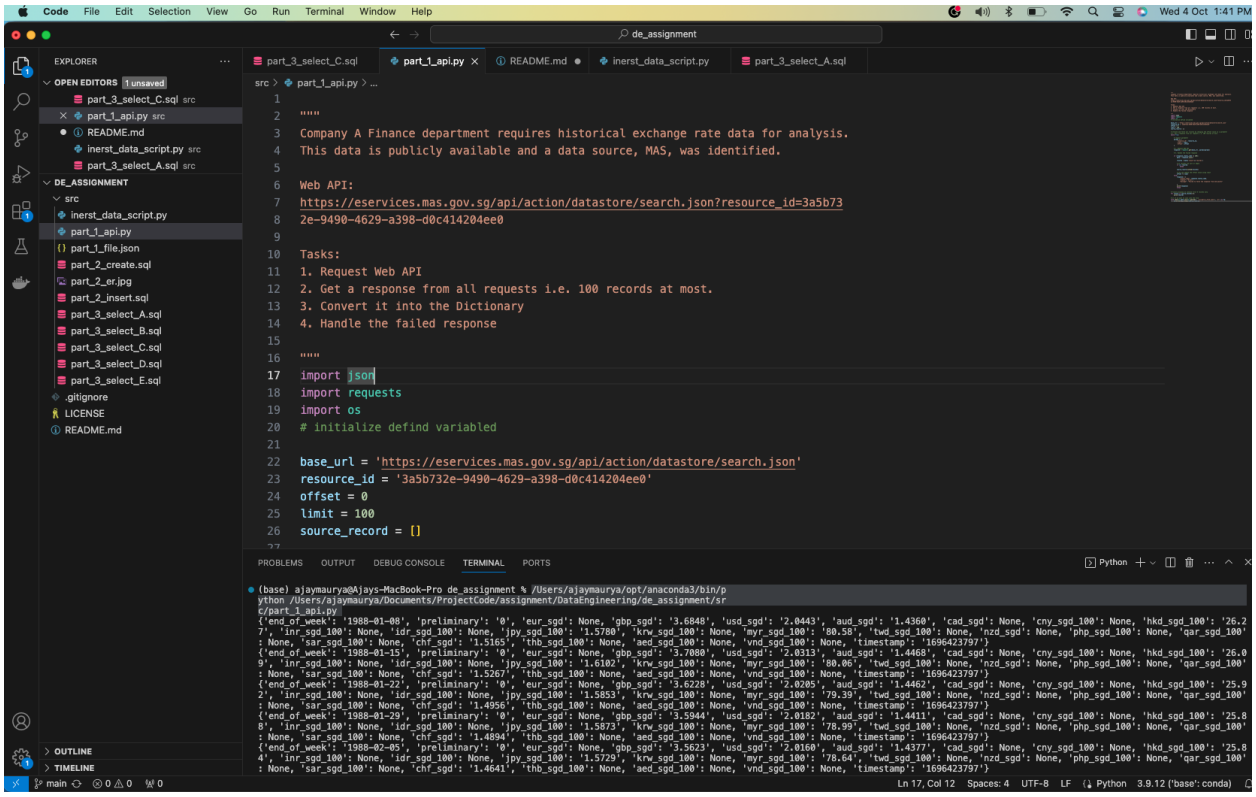


```python
    if (response.status_code == 200):
        data = response.json()

        records = data['result']['records']

        # If records are null or empty
        if not records:
            break

        source_record.extend(records)

        # now the update the offest value using limit
        offset += limit
    else:
        response = {
            "status_code": response.status_code,
            "status": "Error",
            "message": "Failed to fetch the response from end points"

        }
        print(response)
        break


# finally
# 3. Conveet it into the Dictionary
for record in source_record[:5]:
    print(record)
```

Response Handl:

```
$# finally display on console first 5 records only
for record in source_record[:5]:
print(record)


# or if we want to save into the file
with open(os.path.join(os.getcwd(), 'src/part_1_file.json'), 'w+') as
fp:
fp.write(json.dumps(source_record))
```

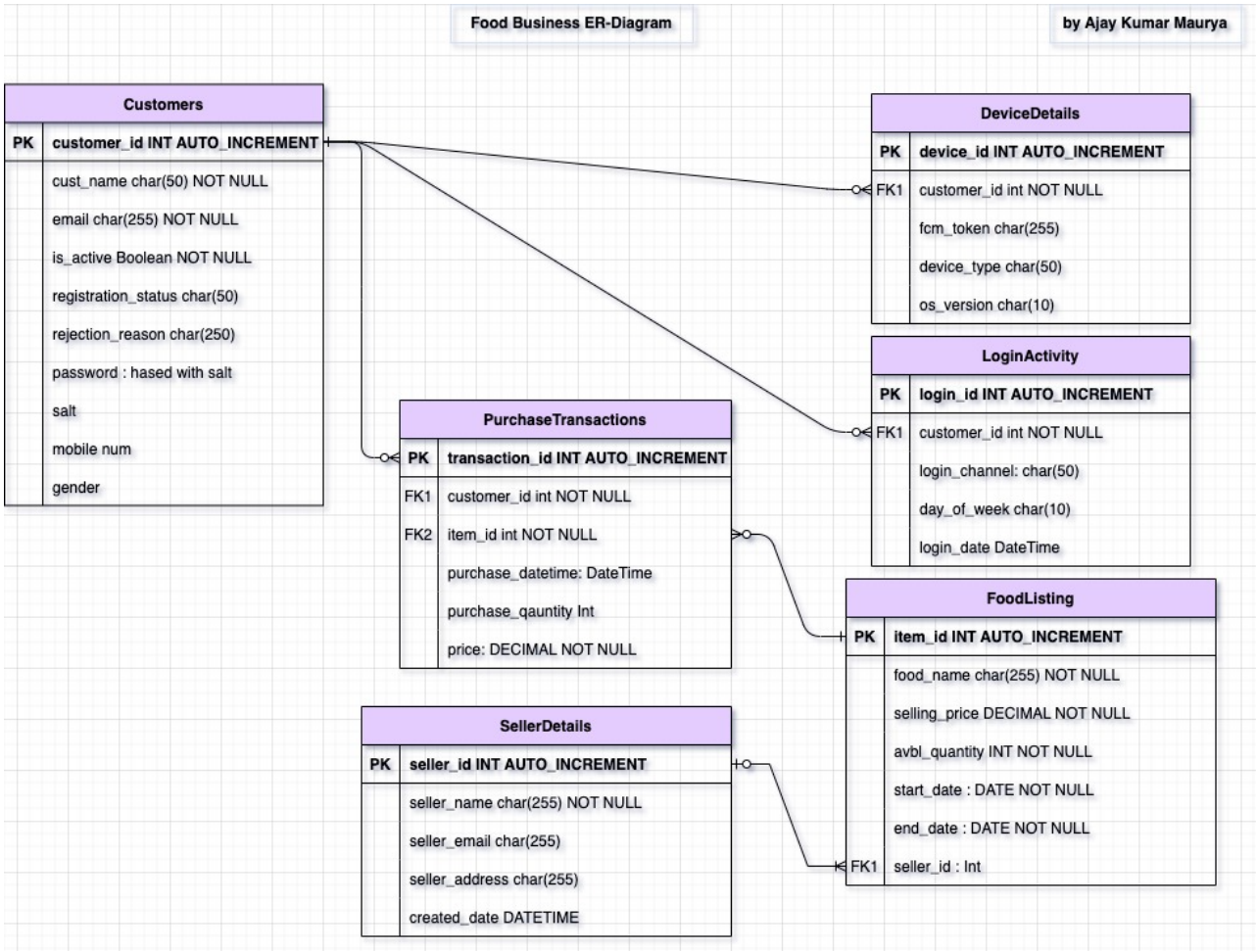
Display 5 records on the console and dump data into the the file in same src/**part_1_file** folder



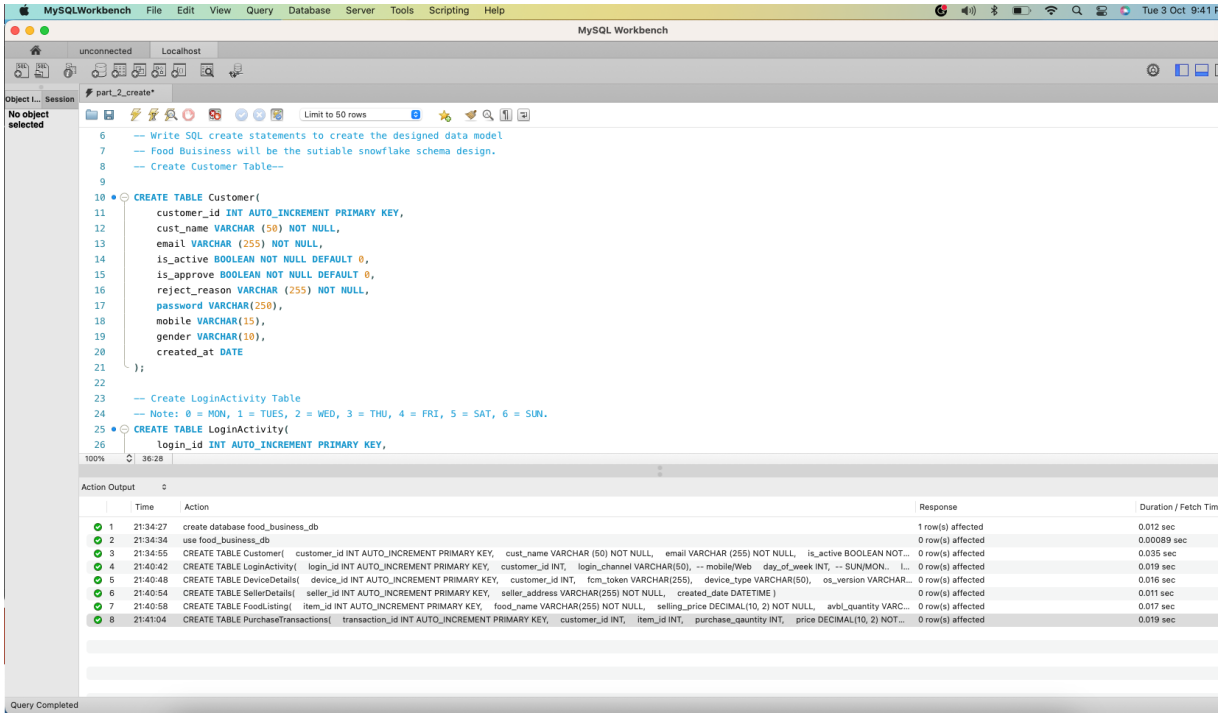
PART 2: Data Modeling, Creating Table and Insert Required Data into the respective Tables.

Created ER-Diagram:

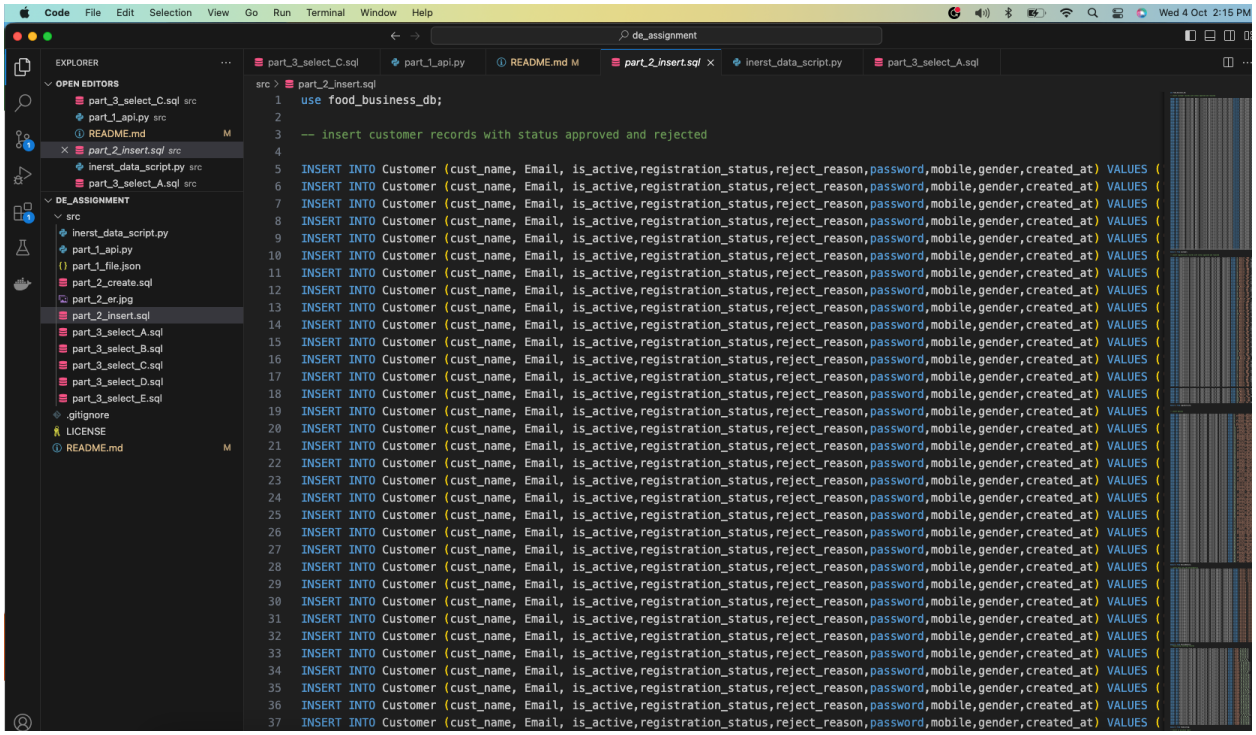
https://github.com/ajay9889/de_assignment/blob/main/src/part_2_er.jpg



Accordingly, run the create Tables command to create all required tables based on the above ER diagram in MySQL WorkBench.



Create “part_2_insert.sql”



Overview Summary and Running the Application Local Machine:

The project is uploaded on GitHub in a public repository.

> https://github.com/ajay9889/de_assignment
> Open it into the Visual Studio Code for Running the Python Script

```
# API Integrations and DataModeling.

Technical project for analysis.

# part_1_api.py

Company A’s Finance department requires historical exchange rate data
for analysis.
This data is publicly available and a data source, MAS, was
identified.

> src/part_1_api.py

How to run the Applications:

> Install Python by going to the official website >
https://www.python.org/downloads/.
> Verify the installation using command> python3 --version
> Run the App by using the command > python3 src/part_1_api.py

# Part 2: Data Modeling for Food Business

![alt
text](https://github.com/ajay9889/de_assignment/blob/main/src/part_2_
er.jpg)
```

```
# Create the Table
> Install MySQL WorkBench
> Create Database
> Run all CREATE TABLE Command which is mentioned in the file
src/part_2_create.sql
> Insert required data with respect to each created table > run
insert query one by one in the file src/part_2_insert.sql

# Part 3: Business Insight

A. How many accounts are pending approval, approved and rejected?
> src/part_3_select_A.sql
B. What are the total logins and purchases for each account?
> src/part_3_select_B.sql
C. How many accounts have purchases by the following buckets? In
percentage.
> src/part_3_select_C.sql
D. What is the third most purchased item in terms of quantity?
> src/part_3_select_D.sql
E. Which day (Mon - Sun) have the highest logins? If there are draws,
to provide all.
> src/part_3_select_E.sql
```