Files

The Answers folder holds the output files of questions 1-3. It also includes answers.py which holds the python code for functions for questions 1-3 and the code to generate the outputs.

The jsonData folder holds the given json files.

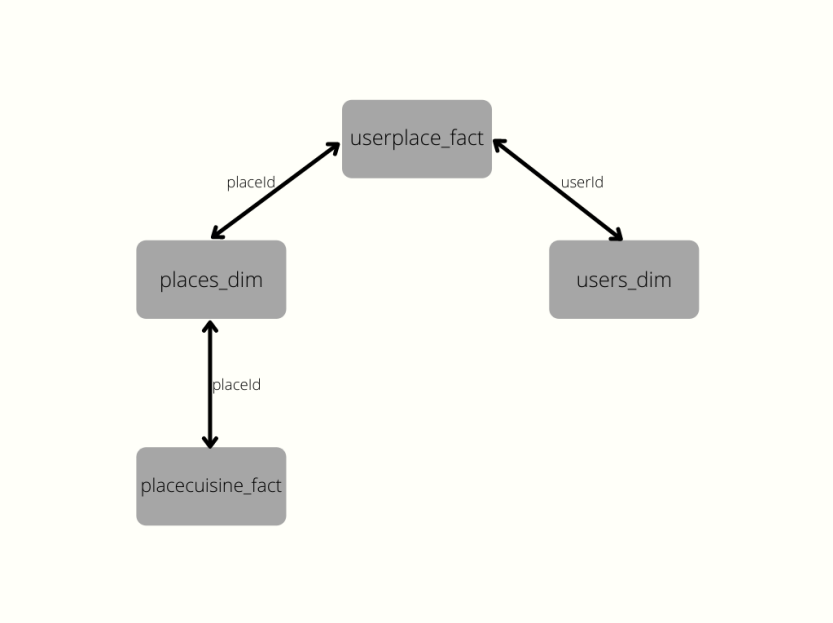
DataModelCreation.py is the one-time run file used to set up the database.

main.py is the file that generates the data for each table in the database. This file is set up so that it can trigger each time the json files are updated or a new one is given.

psDatabase.sqlite3 is the database file that holds the tables and data which were used for this project.

Data Model

For the Data Model, I have chosen to use SQLite to capture the data into a relational database using a fact-dimension data model. I have added a picture of the data model for clarification which I will explain briefly.



To allow for future scalability, I normalized the data following a traditional normalization process for relational database management systems. This involves splitting up the data so that each row in a table is unique. For example, the placeDetails.json file had certain attributes, such as favorite cuisine, which were list types so each of these list type attributes were split up into separate tables to prevent duplication of data and allow for scalability. **Please note that I only created the tables which were needed for the questions asked in the project**. More tables can be created easily using this model.

The foreign key constraints that are specified in the DataModelCreation.py file accounts for any data disparities between userDetails.json and placeDetails.json. For example, if a placeId is in the placeInteractionDetails attribute but not in placeDetails.json, the foreign key constraint in the userplace_fact table will be triggered and that record will not be entered. The constraint can also be modified in terms of what happens when it is triggered so it is malleable in that sense.

I also considered using another approach which involved parsing the json directly in python and storing each record into defined classes. However, the database approach above allows for storage of data for future analysis and minimized data inconsistency.

I chose to use SQLite as it is an easy to use database system for projects as it is self-contained and embeddable into the project itself. If given the tools, I would set up the data with an Apache Spark framework, utilizing Hadoop to store the data from the json files, using Scala or Python and SQL to generate the answers to the output from those files (similar to how I am doing in this project), etc.

Questions and Answers

I have added edge cases to note in the answers.py file as comments. I was debating on either using the pandas package to answer these questions or generating the output via sql. I chose to do the latter but either would have worked. Please note that there are some comments in the sql code, denoted with “--”.