# Load

## Methodology and Tools

- Since the tables are related to each other, PostgreSQL database was used to store the data. PostgreSQL was chosen since it is a relational database, which allows users to control table structures, datatypes, and relationships between the tables within the database.

- Python module SQLAlchemy was used in conjunction with Pandas to interact with the PostgreSQL and import data to the database. There are multiple ways to import data to database as following:

1. Create relational tables using pgAdmin4 and manually import csv files from the platform
2. Create relational tables using pgAdmin4 and import csv files using SQLAlchemy on Python
3. Create relational tables and import files directly using SQLAlchemy on Python

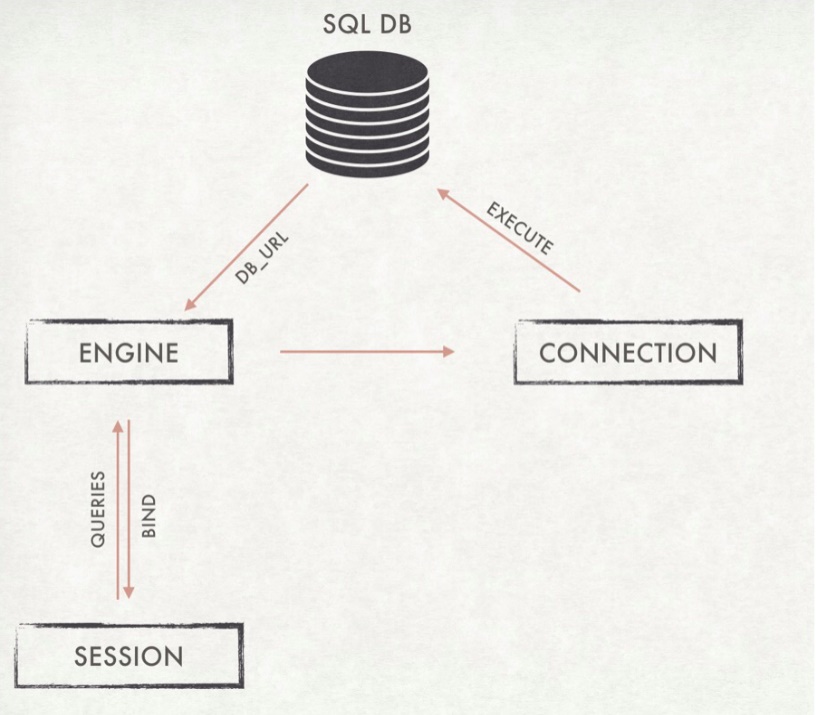
- Our team chose the last method as it required the least manual works and only required one tool, Python. The steps taken to import csv files to database are listed in order as following:

1. Read csv files and created data frames using Pandas.
2. Created engine, which acts as an interface to PostgreSQL local database named Project\_2\_ETL.
3. Created a Declarative Base, which allows user to create classes and tables relative to that base.
4. Defined mapped classes on the Base and tables for each class. Table columns and data types are specified, as well as the relationships between the tables using keywords: Primary Key, Foreign Key, or ForeignKeyConstraint for composite key.
5. Since PostgreSQL only allows 5 connections simultaneously, we created session to talk to the database instead of connecting and executing every time. A session is a workspace for the objects, and at this point we has not opened any connections yet until we commit all changes and close the session.
6. Described database using Metadata. Metadata.create\_all(engine) is to emit the set of schemata created above to the database once committed.
7. Used Pandas to\_sql() function to commit data frames to the database.
8. Committed and closed session.
9. Executed tables to checked if they are inserted to the database successfully.

- PostgreSQL username and password are stored in a file called config.py. This file was not pushed to the Github to hide the confidential information from the public users.

## Challenges

- Even though SQLAlchemy is a powerful module to interact with a databse, users need to have a good understand of what each step does and that they need to be performed to make it work. The following figure helped us to visualize the concept better.



## Future considerations

- Due to time constraint, our team only store data on local host PostgreSQL database. In the future projects, we will consider cloud-based, PostgreSQL-hosted database called ElephantSQL or another cloud-base database. Cloud database has the benefits of storing data on cloud thus data can be accessed from anywhere on the globe. Cloud database also allows access through a web interface or vendor’s API, is easily scalable, and most importantly provide a security of the data through backups on remote servers.