GoodReads

**key:**RdKI3z0BY23iNZuJuGCxw  
**secret:**8pDsmgmv9xXbOdpK0QKbSiZzkigN97XsGxA9pxITy4

First we insert our data into the DB, with a “create.sql” file we can create the tables needed for the project, for now, I just want to create the “books” table so I can insert the data from a “books.csv” file:

create.sql:

CREATE TABLE books (

    id SERIAL PRIMARY KEY,

    isbn VARCHAR NOT NULL,

    title VARCHAR NOT NULL,

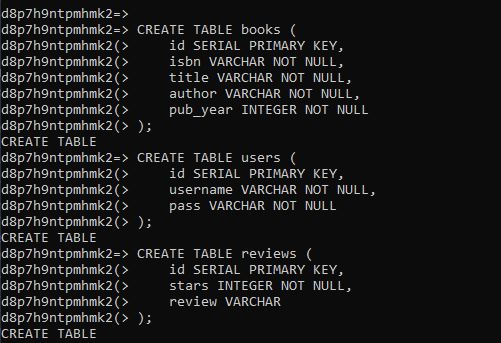
    author VARCHAR NOT NULL,

    year INTEGER NOT NULL

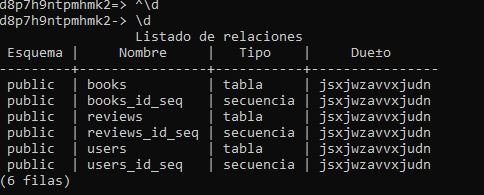
)

In order to create this table in my postgreDB which is hosted in heroku I’ll need to run in cmd:

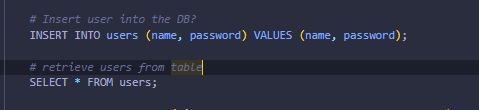
psql DATABASE\_URL (where this is my db uri) and then I can paste in the CREATE TABLE code.



**“\d”** would allow us to see all of the tables in our PostgreDB:



To insert or read values from the table we could run (in the console with psql):



At some point I needed to update a column in the reviews table, in order to associate reviews to users, so:

After the tables are created I can run “import.py” which looks like this:

*import* csv

*import* os

*from* sqlalchemy *import* create\_engine

*from* sqlalchemy.orm *import* scoped\_session, sessionmaker

engine = create\_engine(

    "postgres://jsxjwzavvxjudn:574e4c91abeea383f5d4881a0bf5f8cafbc0eecff2063a95c2be4c4c51e2f541@ec2-54-246-90-10.eu-west-1.compute.amazonaws.com:5432/d8p7h9ntpmhmk2")

db = scoped\_session(sessionmaker(bind=engine))

def main():

    f = open("books.csv")

    reader = csv.reader(f)

*for* isbn, title, author, year in reader:  *# loop gives each column a name*

        db.execute("INSERT INTO books (isbn, title, author, year) VALUES (:isbn, :title, :author, :year)",

                   {"isbn": isbn, "title": title, "author": author, "year": year})  *# substitute values from CSV line into SQL command, as per this dict*

        print(f"Added book {title} to the database")

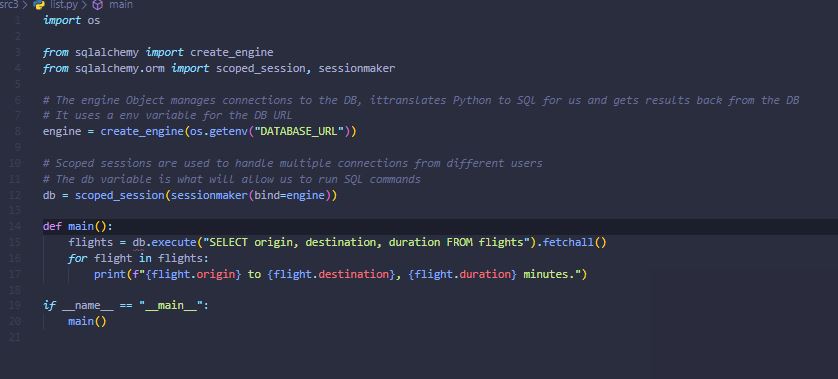
    db.commit()  *# transactions are assumed, so close the transaction finished*

*if* \_\_name\_\_ == "\_\_main\_\_":

    main()

**Connecting Flask and PostgreDB- SQLAlchemy**

SQLAlchemy is used to run SQL commands in a Flask application



I can use “Adminer” to inspect my DB or install PostgreSQL and run “psql URI” where the URI is the link provided in the Heroku credentials list. Then “\dt” will show the tables in the DB.

**Folder Structure**

**Templates** are inside of a “template” folder and **css** and **images** are inside of the “static” folder. Templates should use a “layout.html” which can be extended inside of the other templates.

Sessions

Sessions are a way to store data in a current Session, if the server is reset then that data would be lost:

To tie in together a Flask app with a SQl DB we wil use SQLAlchemy, a Python library to connect Python and SQL