Name - Smeet Kathiria | Class - IT 635:852 | Midterm Project

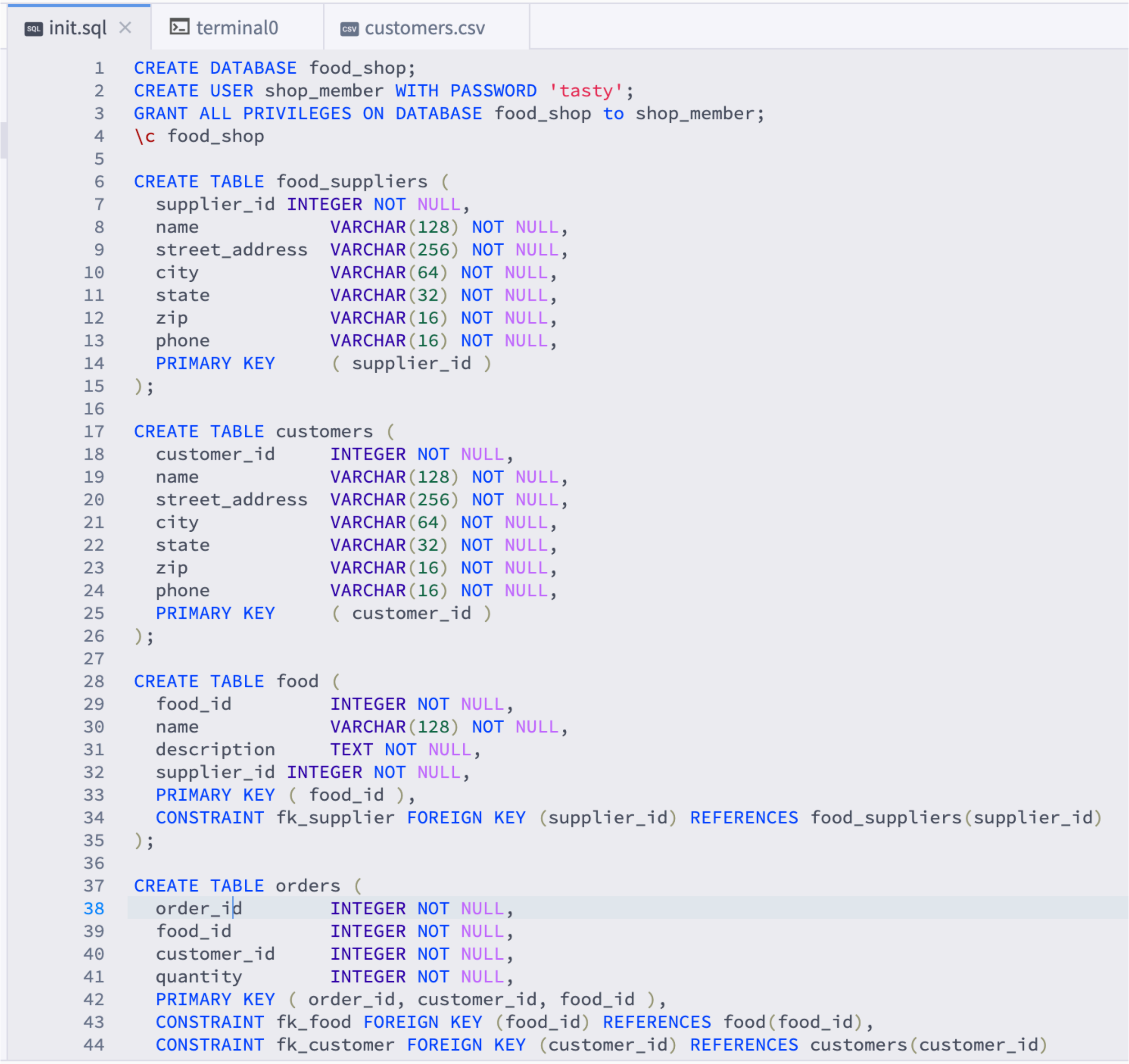
All the code file for this project can be found at :- Link

Step 1: food shop database was created which has four tables

1) food_suppliers
2) customers
3) food
4) orders

User with name shop_member is created which has access to the database.

Below code does the above mentioned task



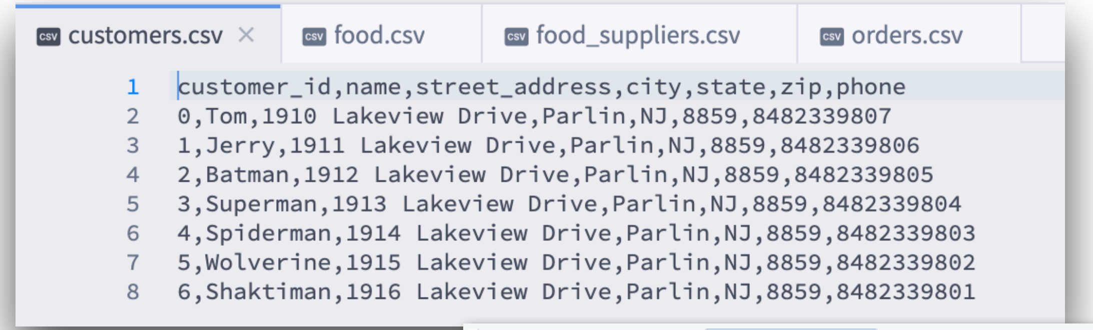
```sql
CREATE DATABASE food_shop;
CREATE USER shop_member WITH PASSWORD 'tasty';
GRANT ALL PRIVILEGES ON DATABASE food_shop to shop_member;
\c food_shop

CREATE TABLE food_suppliers (
  supplier_id INTEGER NOT NULL,
  name            VARCHAR(128) NOT NULL,
  street_address  VARCHAR(256) NOT NULL,
  city            VARCHAR(64) NOT NULL,
  state           VARCHAR(32) NOT NULL,
  zip             VARCHAR(16) NOT NULL,
  phone           VARCHAR(16) NOT NULL,
  PRIMARY KEY     ( supplier_id )
);

CREATE TABLE customers (
  customer_id     INTEGER NOT NULL,
  name            VARCHAR(128) NOT NULL,
  street_address  VARCHAR(256) NOT NULL,
  city            VARCHAR(64) NOT NULL,
  state           VARCHAR(32) NOT NULL,
  zip             VARCHAR(16) NOT NULL,
  phone           VARCHAR(16) NOT NULL,
  PRIMARY KEY     ( customer_id )
);

CREATE TABLE food (
  food_id         INTEGER NOT NULL,
  name            VARCHAR(128) NOT NULL,
  description     TEXT NOT NULL,
  supplier_id INTEGER NOT NULL,
  PRIMARY KEY ( food_id ),
  CONSTRAINT fk_supplier FOREIGN KEY (supplier_id) REFERENCES food_suppliers(supplier_id)
);

CREATE TABLE orders (
  order_id        INTEGER NOT NULL,
  food_id         INTEGER NOT NULL,
  customer_id     INTEGER NOT NULL,
  quantity        INTEGER NOT NULL,
  PRIMARY KEY ( order_id, customer_id, food_id ),
  CONSTRAINT fk_food FOREIGN KEY (food_id) REFERENCES food(food_id),
  CONSTRAINT fk_customer FOREIGN KEY (customer_id) REFERENCES customers(customer_id)
```

To run the init.sql file connect to Postgres using below command

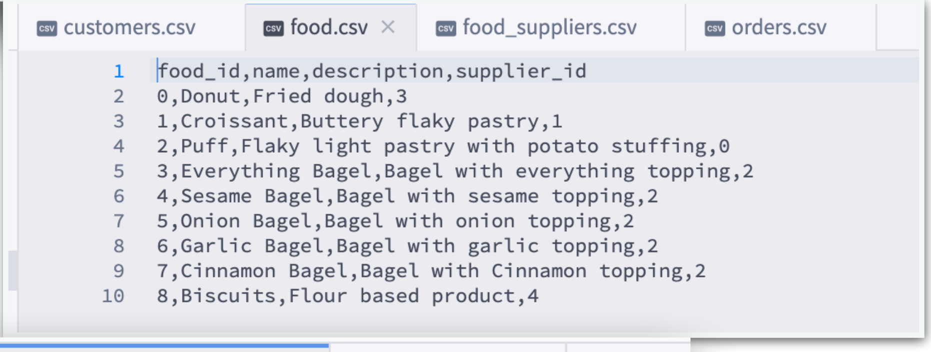
sudo -u postgres -psql

once connected to Postgres, run the command -> **\i init.sql**

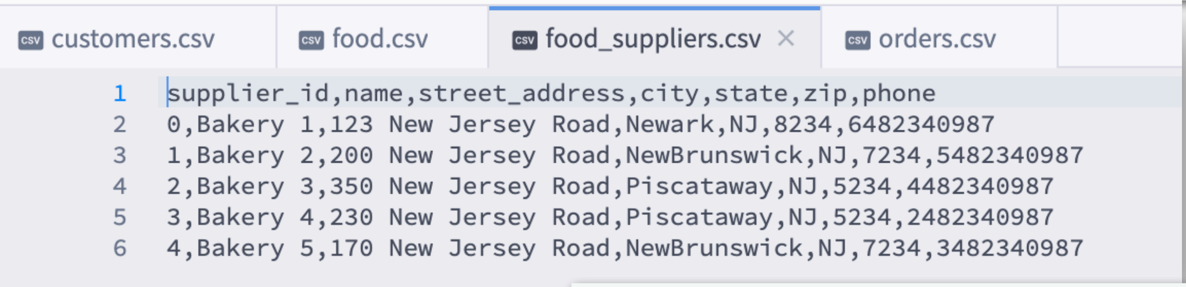
Step 2:- Once the database is created with respective tables, data was inserted from CSV file into the tables. Below images show how data looks like for each of the table.



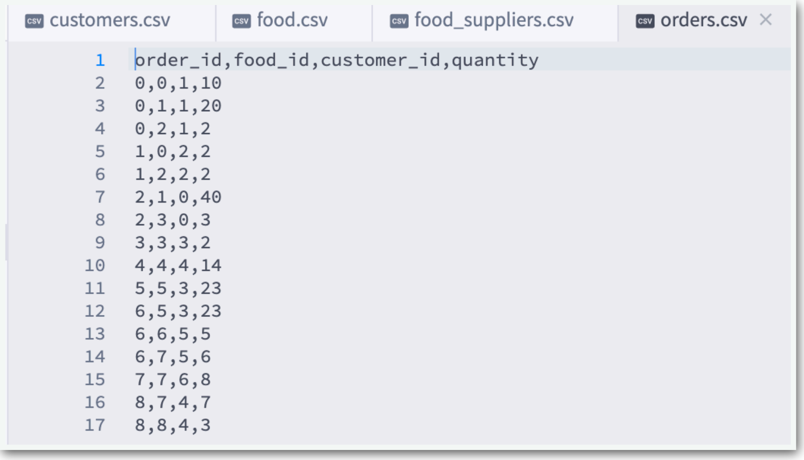
```
customer_id,name,street_address,city,state,zip,phone
0,Tom,1910 Lakeview Drive,Parlin,NJ,8859,8482339807
1,Jerry,1911 Lakeview Drive,Parlin,NJ,8859,8482339806
2,Batman,1912 Lakeview Drive,Parlin,NJ,8859,8482339805
3,Superman,1913 Lakeview Drive,Parlin,NJ,8859,8482339804
4,Spiderman,1914 Lakeview Drive,Parlin,NJ,8859,8482339803
5,Wolverine,1915 Lakeview Drive,Parlin,NJ,8859,8482339802
6,Shaktiman,1916 Lakeview Drive,Parlin,NJ,8859,8482339801
```



```
food_id,name,description,supplier_id
0,Donut,Fried dough,3
1,Croissant,Buttery flaky pastry,1
2,Puff,Flaky light pastry with potato stuffing,0
3,Everything Bagel,Bagel with everything topping,2
4,Sesame Bagel,Bagel with sesame topping,2
5,Onion Bagel,Bagel with onion topping,2
6,Garlic Bagel,Bagel with garlic topping,2
7,Cinnamon Bagel,Bagel with Cinnamon topping,2
8,Biscuits,Flour based product,4
```



```
supplier_id,name,street_address,city,state,zip,phone
0,Bakery 1,123 New Jersey Road,Newark,NJ,8234,6482340987
1,Bakery 2,200 New Jersey Road,NewBrunswick,NJ,7234,5482340987
2,Bakery 3,350 New Jersey Road,Piscataway,NJ,5234,4482340987
3,Bakery 4,230 New Jersey Road,Piscataway,NJ,5234,2482340987
4,Bakery 5,170 New Jersey Road,NewBrunswick,NJ,7234,3482340987
```



```
order_id,food_id,customer_id,quantity
0,0,1,10
0,1,1,20
0,2,1,2
1,0,2,2
1,2,2,2
2,1,0,40
2,3,0,3
3,3,3,2
4,4,4,14
5,5,3,23
6,5,3,23
6,6,5,5
6,7,5,6
7,7,6,8
8,7,4,7
8,8,4,3
```

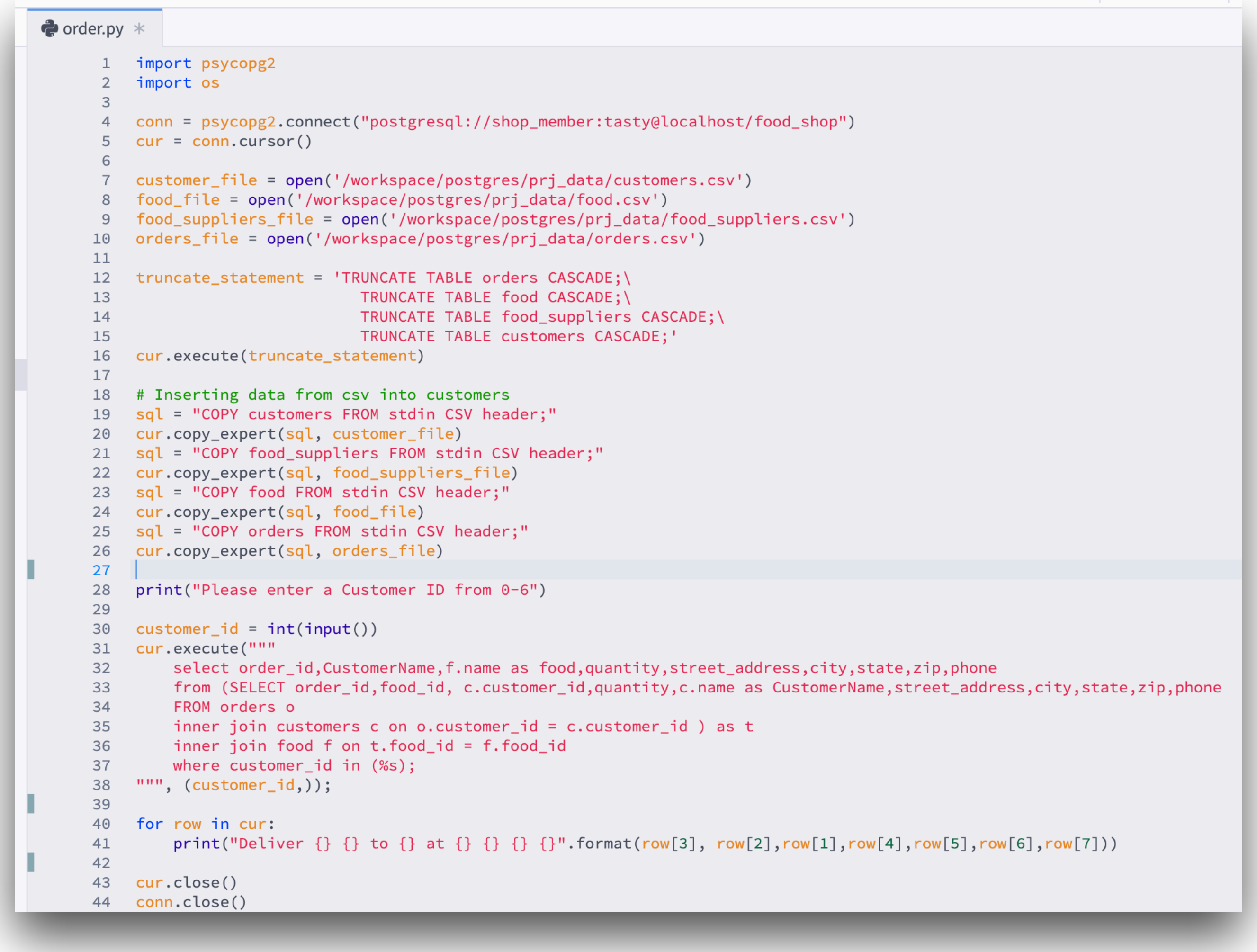
Step 3: Python program was used to populate the data into these tables, the functionality of the program is such that it makes a connection to the database with shop_member as user and then copy the data from each CSV file from a particular location and put that data into the tables inside the connected database.

The other functionality that was added to this program was to solve a particular business use case which is to get all the details for food delivery for a particular customer_id that has ordered.

How this other functionality works is, user is prompted for customer_id, once the application receives the customer_id from the user, it connects to the database and then performs an sql query operations for getting the required data for that customer_id.

The sql query used is combination of joins and filter condition for that specific customer_id.

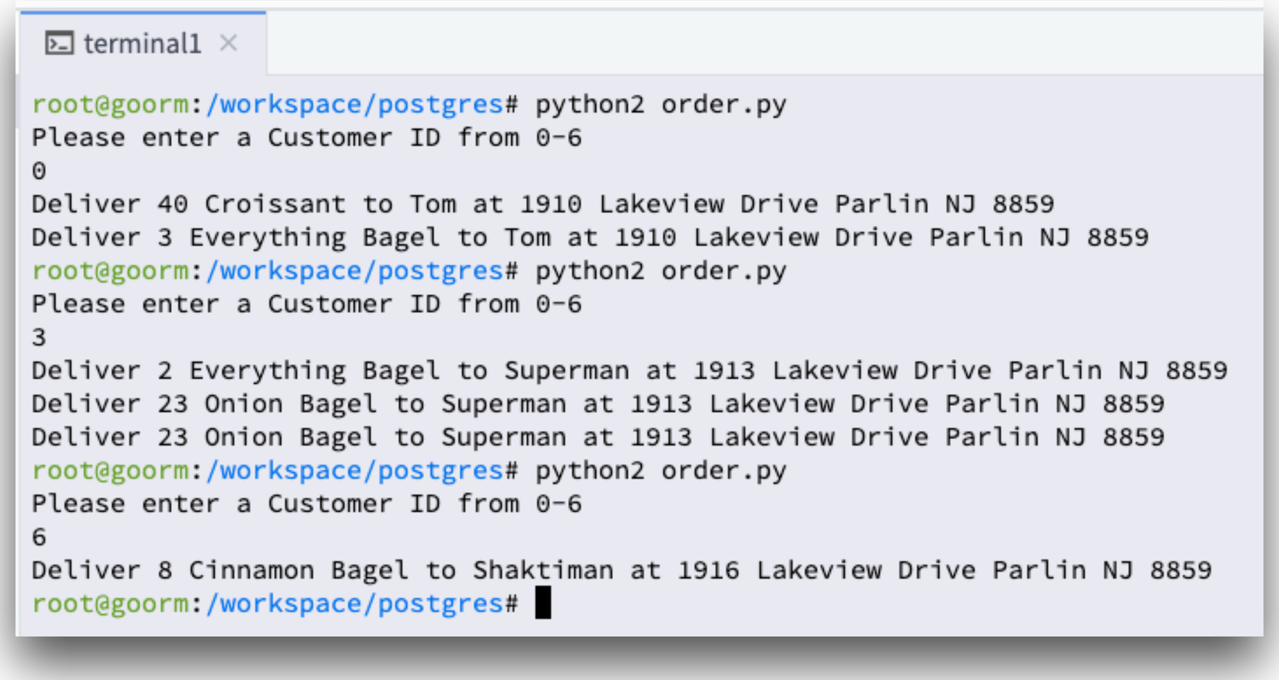
Below code shows what is described above



```
import psycopg2
import os

conn = psycopg2.connect("postgresql://shop_member:tasty@localhost/food_shop")
cur = conn.cursor()

customer_file = open('/workspace/postgres/prj_data/customers.csv')
food_file = open('/workspace/postgres/prj_data/food.csv')
food_suppliers_file = open('/workspace/postgres/prj_data/food_suppliers.csv')
orders_file = open('/workspace/postgres/prj_data/orders.csv')

truncate_statement = 'TRUNCATE TABLE orders CASCADE;\
                        TRUNCATE TABLE food CASCADE;\
                        TRUNCATE TABLE food_suppliers CASCADE;\
                        TRUNCATE TABLE customers CASCADE;'
cur.execute(truncate_statement)

# Inserting data from csv into customers
sql = "COPY customers FROM stdin CSV header;"
cur.copy_expert(sql, customer_file)
sql = "COPY food_suppliers FROM stdin CSV header;"
cur.copy_expert(sql, food_suppliers_file)
sql = "COPY food FROM stdin CSV header;"
cur.copy_expert(sql, food_file)
sql = "COPY orders FROM stdin CSV header;"
cur.copy_expert(sql, orders_file)

print("Please enter a Customer ID from 0-6")

customer_id = int(input())
cur.execute("""
    select order_id,CustomerName,f.name as food,quantity,street_address,city,state,zip,phone
    from (SELECT order_id,food_id, c.customer_id,quantity,c.name as CustomerName,street_address,city,state,zip,phone
    FROM orders o
    inner join customers c on o.customer_id = c.customer_id ) as t
    inner join food f on t.food_id = f.food_id
    where customer_id in (%s);
""", (customer_id,));

for row in cur:
    print("Deliver {} {} to {} at {} {} {} {}".format(row[3], row[2],row[1],row[4],row[5],row[6],row[7]))

cur.close()
conn.close()
```

Step 4: Application was tested by running the command -> python2 order.py

Once the application starts, it prompts for customer_id and then all the required delivery information is presented for that customer which can be seen in the below image.



```
root@goorm:/workspace/postgres# python2 order.py
Please enter a Customer ID from 0-6
0
Deliver 40 Croissant to Tom at 1910 Lakeview Drive Parlin NJ 8859
Deliver 3 Everything Bagel to Tom at 1910 Lakeview Drive Parlin NJ 8859
root@goorm:/workspace/postgres# python2 order.py
Please enter a Customer ID from 0-6
3
Deliver 2 Everything Bagel to Superman at 1913 Lakeview Drive Parlin NJ 8859
Deliver 23 Onion Bagel to Superman at 1913 Lakeview Drive Parlin NJ 8859
Deliver 23 Onion Bagel to Superman at 1913 Lakeview Drive Parlin NJ 8859
root@goorm:/workspace/postgres# python2 order.py
Please enter a Customer ID from 0-6
6
Deliver 8 Cinnamon Bagel to Shaktiman at 1916 Lakeview Drive Parlin NJ 8859
root@goorm:/workspace/postgres#
```

Overall it was very fun project, I learned a lot of things like,

1) Using python to connect to databases, granting user permission and privileges,
2) Creating separate user to connect to database.
3) Loading data from external file source into the database.
4) Running dynamic sql queries based on user given data.