In [*]:
```
import psycopg2

con = psycopg2.connect(database="OBAE", user="cindy", password="Flamingosis01.", host="localhost", port="5432")

print("Database opened successfully")
```

In [*]:
```
cur = con.cursor()
```

In [*]:
```
import psycopg2 as pg
import pandas as pd
import pandas.io.sql as psql
from IPython import display
import matplotlib.image as mpimg
from matplotlib import rcParams
```

In [*]:
```
pd.DataFrame(psql.read_sql("SELECT * FROM oba_locations", con)) #Displaying raw data from dable 'stores'
```

In [*]:
```
# join libraries with weekdays opening hours

pd.DataFrame(psql.read_sql("""
SELECT oba_locations.name ,
    open_weekdays.opening_weekdays,
    open_weekdays.opening_hours_weekdays
FROM open_weekdays
INNER JOIN oba_locations ON oba_locations.id = open_weekdays.id"""
,con))
```

In [*]:
```
# join libraries with weekend opening hours

pd.DataFrame( psql.read_sql("""

SELECT oba_locations.name ,
    open_weekends.opening_weekends,
    open_weekends.opening_hours
FROM open_weekends
INNER JOIN oba_locations ON oba_locations.id = open_weekends.id"""
,con))
```

In [*]:
```
# join opening weekdays and weekend hours

pd.DataFrame(psql.read_sql("""

SELECT  oba_locations.name,
    oba_locations.lat,
    oba_locations.lng ,
    open_weekends.opening_weekends,
    open_weekends.opening_hours
FROM open_weekends,
    oba_locations
WHERE open_weekends.opening_weekends = 'Sunday'
GROUP BY  oba_locations.name ,
    oba_locations.lat,
    oba_locations.lng,
    open_weekends.opening_weekends,
    open_weekends.opening_hours"""
,con))
```

In [*]:
```
df.to_csv('Sunday_opening_libraries.csv', index=False, header=True)
```

In [*]:
```
#find item in stores
Sunday_opening_libraries = pd.read_csv('/Users/cindymendoncapaez/opt/anaconda3/lib/python3.8/site-packages/folium/OBA/Sunday_opening_libr

# Drop rows with missing locations
Sunday_opening_libraries.dropna(subset=['lat','lng'], inplace=True)
```

In [*]:
```
import folium
from folium import Choropleth, Circle, Marker
from folium.plugins import MarkerCluster
import csv
```

In [*]:
```
m_1 = folium.Map(location=[52.379189, 4.899431], tiles='openstreetmap', zoom_start=12)
```

In [*]:
```
for index,row in Sunday_opening_libraries.iterrows():
    lat = row["lat"]
    lon = row["lng"]
    name = row["name"]
    opening_days = row ["opening_weekends"]
    opening_hours = row ["opening_hours"]
    map_displayed_info = '{} : {} : {}'.format(name, opening_days, opening_hours)
    folium.Marker([lat,lon],popup=map_displayed_info).add_to(m_1)
m_1
```

In [*]:
```
pd.DataFrame(psql.read_sql("""

SELECT oba_locations.name ,
    restaurants_oba.name,
    restaurants_oba.address,
    restaurants_oba.opening_hours,
    restaurants_oba.lat,
    restaurants_oba.lng
FROM   join_restaurant_libraries
INNER JOIN oba_locations ON join_restaurant_libraries.restaurants_id = join_restaurant_libraries.restaurants_id
INNER JOIN restaurants_oba ON oba_locations.id = join_restaurant_libraries.libraries_id
LIMIT 5"""
,con))
```

In [*]:
```
#find item in stores
find_restaurants = pd.read_csv('/Users/cindymendoncapaez/opt/anaconda3/lib/python3.8/site-packages/folium/OBA/find_restaurants.csv')

# Drop rows with missing locations
find_restaurants.dropna(subset=['lat','lng'], inplace=True)
```

In [*]:
```
m_2 = folium.Map(location=[52.379189, 4.899431], tiles='openstreetmap', zoom_start=13)
```

In [*]:
```
for index,row in find_restaurants.iterrows():
    lat = row["lat"]
    lon = row["lng"]
    name = row["name"]
    address= row ["address"]
    opening_hours = row ["opening_hours"]
    map_displayed_info = '{} : {} : {}'.format(name, address, opening_hours)
    folium.Marker([lat,lon],popup=map_displayed_info).add_to(m_2)
m_2
```

In [*]:
```
#join books with libraries

pd.DataFrame(psql.read_sql("""

SELECT
  books.id,
  books.title,
  oba_locations.name,
  oba_locations.lat,
  oba_locations.lng,
  join_oba_books.book_id
FROM join_oba_books
JOIN books
ON  books.id = join_oba_books.book_id
JOIN oba_locations
ON oba_locations.id = join_oba_books.library_id"""
, con))
```

In [*]:
```
#search for book Orkael in the libraries

pd.DataFrame(psql.read_sql("""

SELECT
  join_books_lat_lng.id,
  join_books_lat_lng.title,
  join_books_lat_lng.name,
  join_books_lat_lng.book_id,
  join_books_lat_lng.lat,
  join_books_lat_lng.lng
FROM join_books_lat_lng
WHERE join_books_lat_lng.title = 'Orkael'"""

, con))
```

In [*]:
```
#find item in stores
find_book = pd.read_csv('/Users/cindymendoncapaez/opt/anaconda3/lib/python3.8/site-packages/folium/OBA/find_orkadel.csv')

# Drop rows with missing locations
find_book.dropna(subset=['lat','lng'], inplace=True)
```

In [*]:
```
m_3 = folium.Map(location=[52.379189, 4.899431], tiles='openstreetmap', zoom_start=12)
```

In [*]:
```
for index,row in find_book.iterrows():
    lat = row["lat"]
    lon = row["lng"]
    title= row ["title"]
    name = row["name"]
    map_displayed_info = '{} : {}'.format(name, title)
    folium.Marker([lat,lon],popup=map_displayed_info).add_to(m_3)
m_3
```

In [*]:
```
#join books and categories

pd.DataFrame(psql.read_sql(
"""
SELECT
  books.id,
  books.title,
  categories.genres,
  join_books_categories.genres_id,
  join_books_categories.item_id
FROM join_books_categories
JOIN books
ON  books.id = join_books_categories.item_id
JOIN categories
ON categories.id = join_books_categories.genres_id

"""
,con))
```

In [*]:
```
# Find book Something needs to change

pd.DataFrame(psql.read_sql(
"""
SELECT
  genres_books.title,
  genres_books.genres,
  oba_locations.name,
  oba_locations.lat,
  oba_locations.lng,
  join_books_genres_loc.book_id,
  join_books_genres_loc.location_id

FROM join_books_genres_loc
JOIN genres_books
ON  genres_books.id = join_books_genres_loc.book_id
JOIN oba_locations
ON oba_locations.id = join_books_genres_loc.location_id
WHERE genres_books.title = 'Something needs to change'

"""
,con))
```

In [*]:
```
#check for availability of books in OBA Olympic Quarter

books = []
books.append(["Something needs to change"])


def checkBook():
    book = str(input("Enter name of the book"))

    if book == 'Something needs to change':
        print("This book is available.")
    else:
        print("This book is not available")


# Main Program Starts Here
checkBook()
```

In [*]:
```
#find item in stores
find_book_category = pd.read_csv('/Users/cindymendoncapaez/opt/anaconda3/lib/python3.8/site-packages/folium/OBA/joined_books_genres_lat_

# Drop rows with missing locations
find_book_category.dropna(subset=['lat','lng'], inplace=True)
```

In [*]:
```
m_4 = folium.Map(location=[52.379189, 4.899431], tiles='openstreetmap', zoom_start=12)
```

In [*]:
```
for index,row in find_book_category.iterrows():
    lat = row["lat"]
    lon = row["lng"]
    title= row ["title"]
    name = row["name"]
    map_displayed_info = '{} : {}'.format(name, title)
    folium.Marker([lat,lon],popup=map_displayed_info).add_to(m_4)
m_4
```

In [*]:
```
# available seats

pd.DataFrame(psql.read_sql(
"""
SELECT
  oba_locations.name,
  floors.floors,
  floors.seat,
  floors.session,
  oba_locations.lat,
  oba_locations.lng,
  join_library_floors.libraries_id,
  join_library_floors.floors_id

FROM join_library_floors
JOIN floors
ON  floors.id = join_library_floors.libraries_id
JOIN oba_locations
ON oba_locations.id = join_library_floors.floors_id


"""
,con))
```

In [*]:
```
seat = []
seat.append(["AE-1"])
seat.append(["AE-2"])
seat.append(["AE-3"])
seat.append(["BT-5"])
seat.append(["BT-6"])
seat.append(["BT-1"])
seat.append(["PP-0"])
seat.append(["PP-5"])



def checkSeat():
    row = str(input("Enter seat name (two letters AE,BT or PP + one number from 0 to 5)"))

    if row == 'PP-0'or 'BT-5' or 'PP-0':
        print("This seat is already booked.")
    else:
        print("This seat is empty.")


# Main Program Starts Here
checkSeat()
```