

DATABASE PROJECT PART-3

By: Pratyusha Sanapathi

Screenshots that you successfully can import and use libraries

Checking Python Installation

```
C:\Users\Pratyusha Sanapathi>python
Python 3.6.4 (v3.6.4:d48ebeb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```

Installing psycopg2

```
C:\Users\Pratyusha Sanapathi>pip install psycopg2 --user
Collecting psycopg2
  Using cached psycopg2-2.9.5-cp36-cp36m-win32.whl (999 kB)
Installing collected packages: psycopg2
Successfully installed psycopg2-2.9.5
```

Installing pandas

```
C:\Users\Pratyusha Sanapathi>pip install pandas
Requirement already satisfied: pandas in c:\users\pratyusha sanapathi\appdata\local\programs\python\python36-32\lib\site-packages (1.1.5)
Requirement already satisfied: numpy>=1.15.4 in c:\users\pratyusha sanapathi\appdata\local\programs\python\python36-32\lib\site-packages (from pandas) (1.19.5)
Requirement already satisfied: pytz>=2017.2 in c:\users\pratyusha sanapathi\appdata\local\programs\python\python36-32\lib\site-packages (from pandas) (2017.3)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\users\pratyusha sanapathi\appdata\local\programs\python\python36-32\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\pratyusha sanapathi\appdata\local\programs\python\python36-32\lib\site-packages (from python-dateutil>=2.7.3->pandas) (1.11.0)
```

Screenshots that you successfully established your connection with Postgres

Connecting to Python

```
RESTART: C:\Users\Pratyusha Sanapathi\Desktop\Assignments\DB\movie_data\Phase-3\connection.py
PostgreSQL server information
('user': 'postgres', 'channel_binding': 'prefer', 'dbname': 'postgres', 'host': 'localhost', 'port': '5432', 'options': '', 'sslmode': 'prefer', 'sslcompression': '0', 'sslnoni': '1', 'ssl_min_protocol_version': 'TLSv1.2', 'gssencmode': 'disable', 'krbservername': 'postgres', 'target_session_attrs': 'any')
You are connected to - ('PostgreSQL 15.0, compiled by Visual C++ build 1914, 64-bit',)
```

2 New Questions

1. Write a SQL query to find the movies with year and genres. Return movie title, movie year and generic title.

ANS:

Code:

```
import psycopg2
import pandas as pd
import warnings

warnings.filterwarnings('ignore')

# connecting to db
con = psycopg2.connect(
    host="localhost",
    database="postgres",
    user="postgres",
    password="DBproject",
    port="5432"
)

# cursor
cur = con.cursor()

# query execution
cur.execute("SELECT mov_title, mov_year, gen_title FROM movie NATURAL JOIN movie_genre NATURAL JOIN genre;")
movie_details = pd.DataFrame(columns = ['mov_title', 'mov_year', 'gen_title'])

table = cur.fetchall()

for r in table:
    output_table_df = {'mov_title':r[0], 'mov_year':r[1], 'gen_title':r[2]}
    movie_details = movie_details.append(output_table_df, ignore_index = True)
print(movie_details)

# close the cursor
cur.close()

# close the connection
con.close()
```

Output using Python:

No. of rows returned =102

```
RESTART: C:\Users\Pratyusha Sanapathi\Desktop\Assignments\DB\movie_data\Phase-3\query_1.py
   mov_title  mov_year  gen_title
0         Vertigo    1958      Action
1    The Innocents    1961  Adventure
2  Lawrence of Arabia    1962  Animation
3    The Deer Hunter    1978  Biography
4        Amadeus    1984    Comedy
..         ...      ...      ...
97         Two Lovers    1999  Adventure|Comedy|Drama
98  Endgame: Blueprint for Global Enslavement    1992  Action|Drama|Romance
99         Blueprint for Murder, A    2002  Comedy|Drama|Thriller
100             Mickey    2006  Adventure|Romance
101             Escape    1993    Drama|Romance

[102 rows x 3 columns]
>>>
```

Query:

```
SELECT mov_title, mov_year, gen_title
FROM movie
NATURAL JOIN movie_genre
NATURAL JOIN genre;
```

Output of same query in pgAdmin:

```
1 SELECT mov_title, mov_year, gen_title
2     FROM movie
3     NATURAL JOIN movie_genre
4     NATURAL JOIN genre;
5
```

Data Output

	mov_title character varying	mov_year integer	gen_title character varying
1	Vertigo	1958	Action
2	The Innocents	1961	Adventure
3	Lawrence of Ara...	1962	Animation
4	The Deer Hunter	1978	Biography
5	Amadeus	1984	Comedy
6	Blade Runner	1982	Crime
7	Eyes Wide Shut	1999	Drama
8	The Usual Suspe...	1995	Horror
9	Chinatown	1974	Music
10	Boogie Nights	1997	Mystery
11	Annie Hall	1977	Romance
12	Princess Monono...	1997	Thriller
13	The Shawshank ...	1994	War
14	American Beauty	1999	Horror Sci-Fi
15	Titanic	1997	Adventure Come...
16	Good Will Hunting	1997	Action
17	Deliverance	1972	Documentary Dra...

Total rows: 102 of 102 Query complete 00:00:00.127

2. write a SQL query to find those movies, which were released before 1998 and language is Japanese.

ANS:

Code:

```
import psycopg2
import pandas as pd
import warnings

warnings.filterwarnings('ignore')

# connecting to db
con = psycopg2.connect(
    host="localhost",
    database="postgres",
    user="postgres",
    password="DBproject",
    port="5432"
)

# cursor
cur = con.cursor()

# query execution
cur.execute("SELECT mov_id, mov_title FROM movie WHERE mov_year<1998 AND mov_lang = 'Japanese';")
movie_info= pd.DataFrame(columns = ['mov_id', 'mov_title'])

table = cur.fetchall()

for r in table:
    output_table_df ={'mov_id':r[0], 'mov_title': r[1]}
    movie_info = movie_info.append(output_table_df, ignore_index = True)
print(movie_info)

# close the curesor
cur.close()

# close the connection
con.close()
```

Output using Python:

No. of rows returned =2

```
RESTART: C:\Users\Pratyusha Sanapathi\Desktop\Assignments\DB\movie_data\Phase-3\query_2.py 5
mov_id      mov_title
912  Princess Mononoke
926    Seven Samurai
```

Query:

```
SELECT mov_id, mov_title
FROM movie
WHERE mov_year<1998
AND mov_lang = 'Japanese'
```

Output of same query in pgAdmin:

Query

Query History

1

2

3

4

5

6

SELECT

mov_id, mov_title

FROM

movie

WHERE

mov_year<1998

AND

mov_lang = 'Japanese'

Data Output

	mov_id [PK] integer	mov_title character varying
1	912	Princess Mononoke
2	926	Seven Samurai

Total rows: 2 of 2

Query complete 00:00:00.410