Portland State
UNIVERSITY

Project Part 3: “Pregnancy AND Health”

Submitted by:
Name: Niger Sultana Tahniat
PSU ID: 983114315
Instructor: Yeganeh Jalalpour

PART 1

QUERY: (2 NEW)

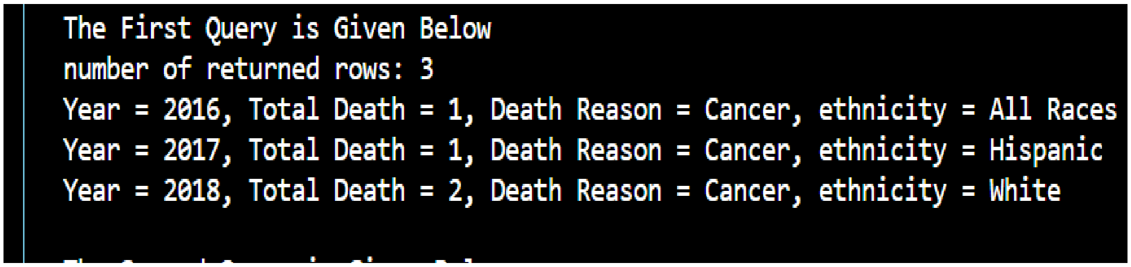
1. **In which year and how many women died of cancer during pregnancy and what was race ethnicity.**

   Response:

   SQL:

```
SELECT year, COUNT(death) AS total_death, underlying_cause,
race_ethnicity FROM
(SELECT d.year,d.underlying_cause,d.death, e.race_ethnicity FROM
"Pregnancy_and_Health".death_record
d JOIN "Pregnancy_and_Health".race_ethnicity e ON
d.ethnicity_id=e.id) AS death_record WHERE
underlying_cause='Cancer' GROUP BY
(year,underlying_cause,race_ethnicity);
```

OUTPUT SCREENSHOT:



```
The First Query is Given Below
number of returned rows: 3
Year = 2016, Total Death = 1, Death Reason = Cancer, ethnicity = All Races
Year = 2017, Total Death = 1, Death Reason = Cancer, ethnicity = Hispanic
Year = 2018, Total Death = 2, Death Reason = Cancer, ethnicity = White
```

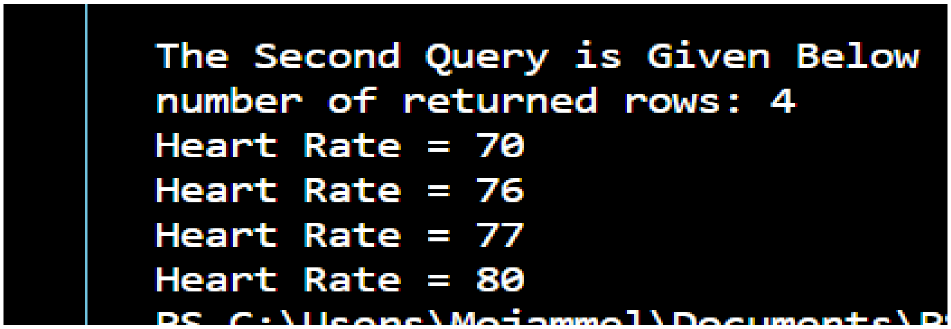
2. **What was the heart rate of women who have high risk level mental health conditions below age 20?**

   Response:

   SQL:

   SELECT m.heart_rate
   FROM "Pregnancy_and_Health".mental_health m JOIN "Pregnancy_and_Health".risk_level r ON m.risk_level_id=r.id AND r.risk_level='high risk' AND m.age<20 GROUP BY(m.heart_rate);

OUTPUT SCREENSHOT:

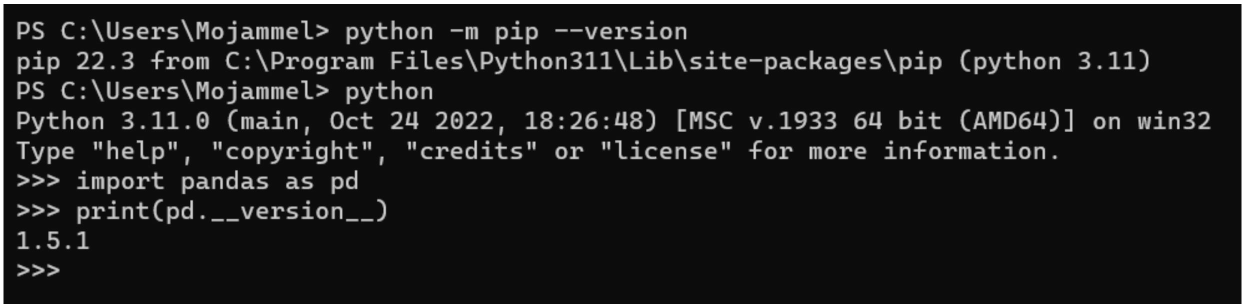


```
The Second Query is Given Below
number of returned rows: 4
Heart Rate = 70
Heart Rate = 76
Heart Rate = 77
Heart Rate = 80
PS C:\Users\Mojammel\Documents\P
```

PART 2
(CODE) : CONNECTION, IMPORT LIBRARIES & TWO QUERIES & OUTPUT

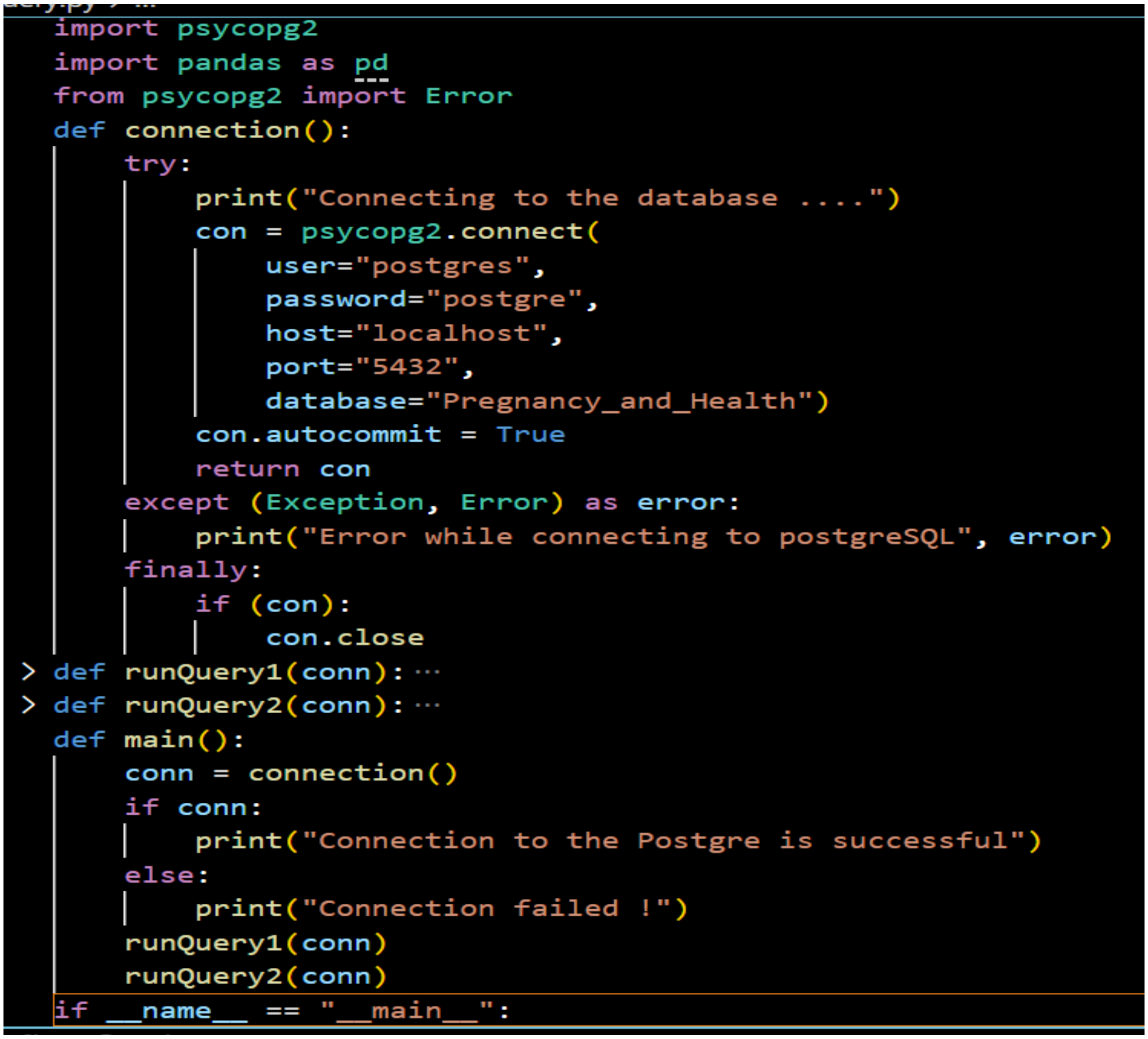
1) Connect python to postgreSQL

Response: I have successfully installed python, pip3 and pandas in my local machine. The screenshots are given below. Screenshots for library imports are-



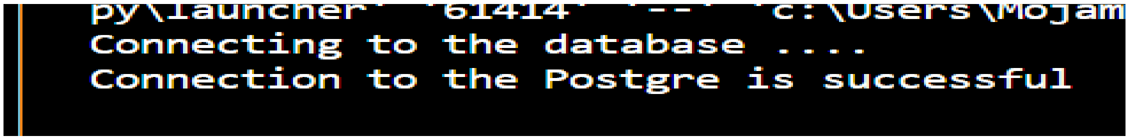
```
PS C:\Users\Mojammel> python -m pip --version
pip 22.3 from C:\Program Files\Python311\Lib\site-packages\pip (python 3.11)
PS C:\Users\Mojammel> python
Python 3.11.0 (main, Oct 24 2022, 18:26:48) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import pandas as pd
>>> print(pd.__version__)
1.5.1
>>>
```

2) python code to establish the connection between python and postgres and libraries imports



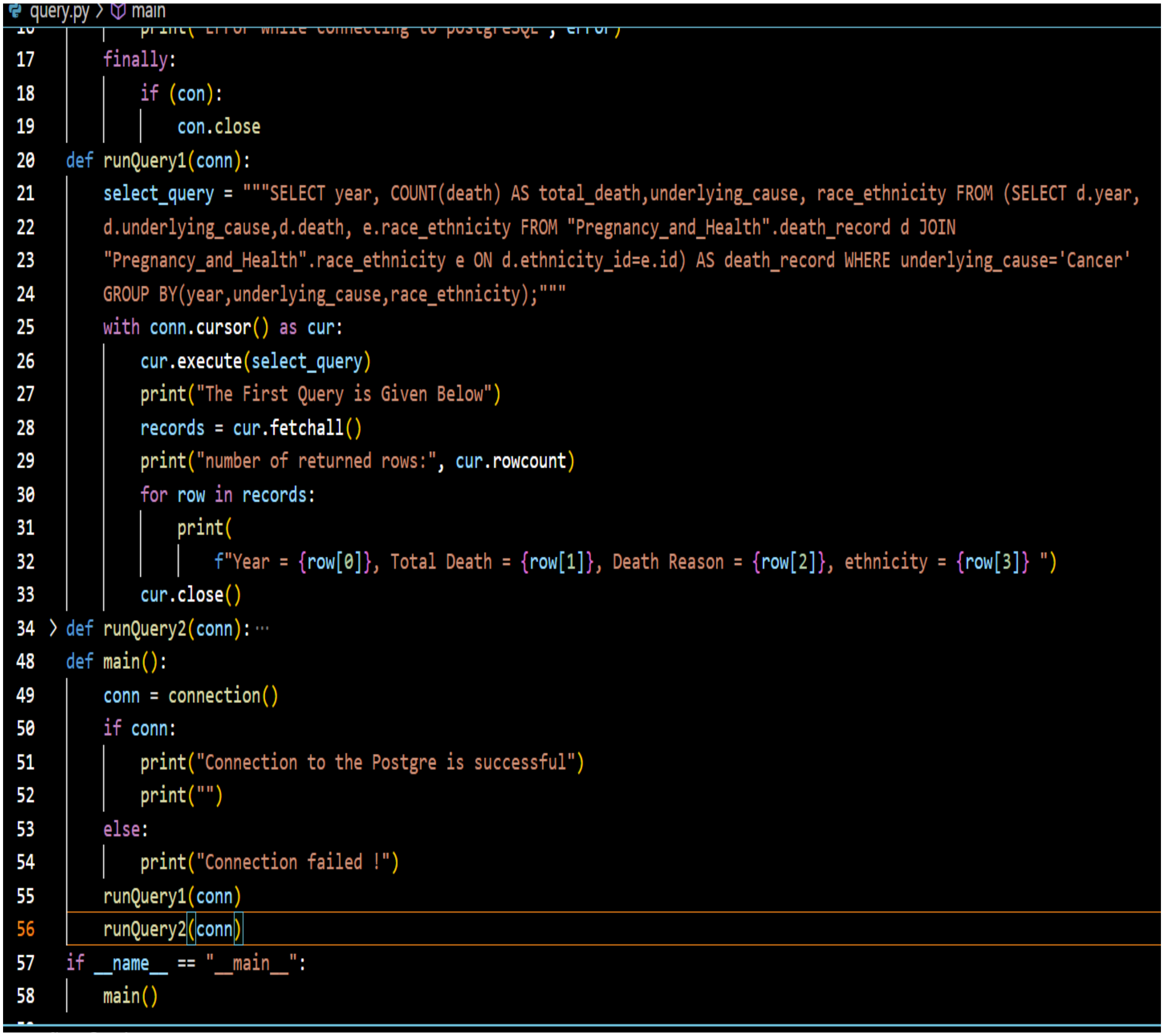
```
import psycopg2
import pandas as pd
from psycopg2 import Error
def connection():
    try:
        print("Connecting to the database ....")
        con = psycopg2.connect(
            user="postgres",
            password="postgre",
            host="localhost",
            port="5432",
            database="Pregnancy_and_Health")
        con.autocommit = True
        return con
    except (Exception, Error) as error:
        print("Error while connecting to postgreSQL", error)
    finally:
        if (con):
            con.close
def runQuery1(conn): ···
def runQuery2(conn): ···
def main():
    conn = connection()
    if conn:
        print("Connection to the Postgre is successful")
    else:
        print("Connection failed !")
    runQuery1(conn)
    runQuery2(conn)
if __name__ == "__main__":
```

As all my code is written in one .py file, cropping the output for the connection establishes a successful message from the whole output.



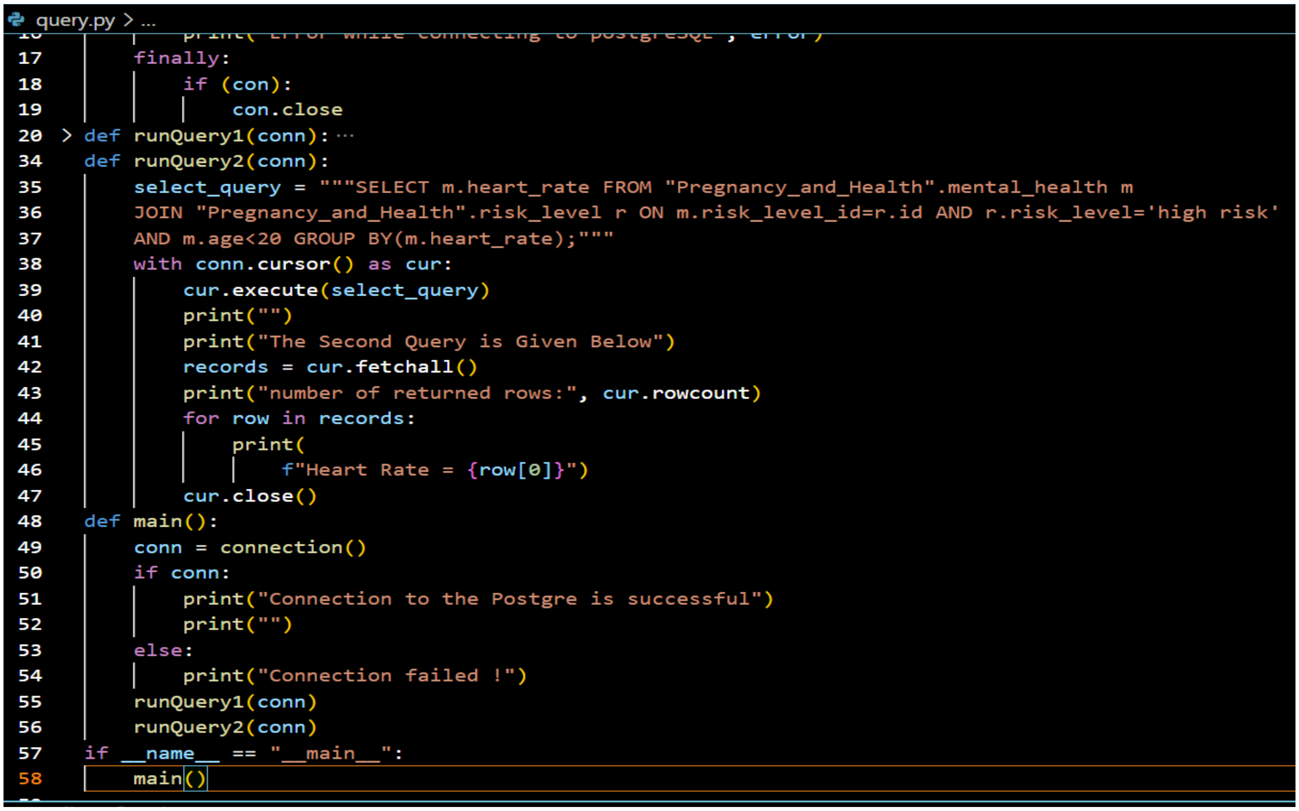
```
py\launcher' '61414' '--' 'c:\Users\Mojam
Connecting to the database ....
Connection to the Postgre is successful
```

2) Code for first query



```python
        print("Error while connecting to postgreSQL", error)
    finally:
        if (con):
            con.close
def runQuery1(conn):
    select_query = """SELECT year, COUNT(death) AS total_death,underlying_cause, race_ethnicity FROM (SELECT d.year,
    d.underlying_cause,d.death, e.race_ethnicity FROM "Pregnancy_and_Health".death_record d JOIN
    "Pregnancy_and_Health".race_ethnicity e ON d.ethnicity_id=e.id) AS death_record WHERE underlying_cause='Cancer'
    GROUP BY(year,underlying_cause,race_ethnicity);"""
    with conn.cursor() as cur:
        cur.execute(select_query)
        print("The First Query is Given Below")
        records = cur.fetchall()
        print("number of returned rows:", cur.rowcount)
        for row in records:
            print(
                f"Year = {row[0]}, Total Death = {row[1]}, Death Reason = {row[2]}, ethnicity = {row[3]} ")
        cur.close()
def runQuery2(conn):···
def main():
    conn = connection()
    if conn:
        print("Connection to the Postgre is successful")
        print("")
    else:
        print("Connection failed !")
    runQuery1(conn)
    runQuery2(conn)
if __name__ == "__main__":
    main()
```

2) Code for second query

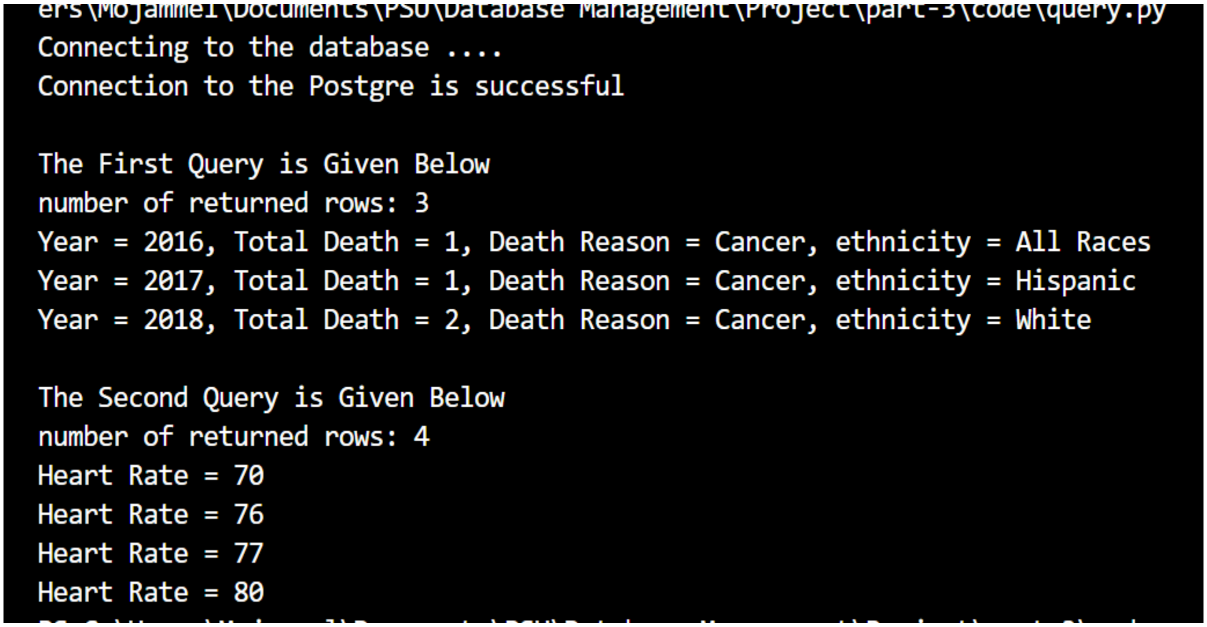


query.py > ...

```python
        print("Error while connecting to PostgreSQL", error)
    finally:
        if (con):
            con.close
def runQuery1(conn):···
def runQuery2(conn):
    select_query = """SELECT m.heart_rate FROM "Pregnancy_and_Health".mental_health m
    JOIN "Pregnancy_and_Health".risk_level r ON m.risk_level_id=r.id AND r.risk_level='high risk'
    AND m.age<20 GROUP BY(m.heart_rate);"""
    with conn.cursor() as cur:
        cur.execute(select_query)
        print("")
        print("The Second Query is Given Below")
        records = cur.fetchall()
        print("number of returned rows:", cur.rowcount)
        for row in records:
            print(
                f"Heart Rate = {row[0]}")
        cur.close()
def main():
    conn = connection()
    if conn:
        print("Connection to the Postgre is successful")
        print("")
    else:
        print("Connection failed !")
    runQuery1(conn)
    runQuery2(conn)
if __name__ == "__main__":
    main()
```

SCREENSHOT FOR OUTPUT :

All codes are in one .py file, so this is one output for all code requirements. Also added output screenshot for both queries above specifically cropping from this one.



```
ers\Mojammel\Documents\PSU\Database Management\Project\part-3\code\query.py
Connecting to the database ....
Connection to the Postgre is successful

The First Query is Given Below
number of returned rows: 3
Year = 2016, Total Death = 1, Death Reason = Cancer, ethnicity = All Races
Year = 2017, Total Death = 1, Death Reason = Cancer, ethnicity = Hispanic
Year = 2018, Total Death = 2, Death Reason = Cancer, ethnicity = White

The Second Query is Given Below
number of returned rows: 4
Heart Rate = 70
Heart Rate = 76
Heart Rate = 77
Heart Rate = 80
```

THE PYTHON CODE:

```
import psycopg2
import pandas as pd
from psycopg2 import Error

def connection():
    try:
        print("Connecting to the database ....")
        con = psycopg2.connect(
            user="postgres",
            password="postgre",
            host="localhost",
            port="5432",
            database="Pregnancy_and_Health")
        con.autocommit = True
        return con
    except (Exception, Error) as error:
        print("Error while connecting to postgreSQL", error)
    finally:
        if (con):
            Con.close

def runQuery1(conn):
    select_query = """SELECT year, COUNT(death) AS
total_death,underlying_cause, race_ethnicity FROM (SELECT d.year,
    d.underlying_cause,d.death, e.race_ethnicity FROM
"Pregnancy_and_Health".death_record d JOIN
    "Pregnancy_and_Health".race_ethnicity e ON d.ethnicity_id=e.id) AS
death_record WHERE underlying_cause='Cancer'
    GROUP BY(year,underlying_cause,race_ethnicity);"""
    with conn.cursor() as cur:
        cur.execute(select_query)
        print("The First Query is Given Below")
        records = cur.fetchall()
        print("number of returned rows:", cur.rowcount)
        for row in records:
            print(
                f"Year = {row[0]}, Total Death = {row[1]}, Death Reason = {row[2]},
ethnicity = {row[3]} ")
        cur.close()
```

```
def runQuery2(conn):
    select_query = """SELECT m.heart_rate FROM
"Pregnancy_and_Health".mental_health m
    JOIN "Pregnancy_and_Health".risk_level r ON m.risk_level_id=r.id AND
r.risk_level='high risk'
    AND m.age<20 GROUP BY(m.heart_rate);"""
    with conn.cursor() as cur:
        cur.execute(select_query)
        print("")
        print("The Second Query is Given Below")
        records = cur.fetchall()
        print("number of returned rows:", cur.rowcount)
        for row in records:
            print(
                f"Heart Rate = {row[0]}")
        cur.close()

def main():
    conn = connection()
    if conn:
        print("Connection to the Postgre is successful")
        print("")
    else:
        print("Connection failed !")
    runQuery1(conn)
    runQuery2(conn)
if __name__ == "__main__":
    main()
```

NOTES:

I have written my connection code and two queries in the same .py file. Called the connection initialize form my main once. And Added two different function calls for two queries. In my one terminal output there are all output for connection establishment, library imports, query one result and query two result. In above for each specific part have crop the outputs from the terminal's output. But at the end I uploaded a screenshot for the whole terminal's output.

Initially, I made 3 files, one for connection, second for first query, third for the second query. But later I liked to add everything in one so that I can add only one code in this pdf file as the page requirement was within 3 pages. Though it will cross 3 pages, sorry about it. I hope that's okay.

I also added two queries in one function initially and used cursor.execute twice for two different queries in one function. But to make it look clear I added it in two different functions for 2 queries later. And call the two different query functions from my main(). I hope that is okay too.

PROGRESS WITH PROJECT AND MODIFICATIONS:

1. No new modifications have been made in my project's database.. I already have 9 tables.I don't have plans to expand the database anymore.
2. I have added a few more new queries in my list for the final part.
3. I have installed python and libraries for the first time on my local machine. I took many steps to find out how to make everything work successfully, for example adding installation path to the system after installation, checking version with different type commands because python -v, pip -v or many others were giving errors like pip not recognized. Finally, I solved all issues and also I have added successful screenshots for every step that is required for this project part 3.
4. Finally, I am happy to connect python to postgres. I have learnt many things for this project. And I am happy that my connection, python code and queries are all working.

THANK YOU