

Assignment-7

CREATING A DATABASE:-

```
Select SQL Shell (psql)
Server: [localhost]:
Database: [postgres]:
Port: [5433]:
Username: [postgres]:
Password for user postgres:
psql (13.21)
WARNING: Console code page (437) differs from Windows code page (1252)
8-bit characters might not work correctly. See psql reference
page "Notes for Windows users" for details.
Type "help" for help.

postgres=# \l
      List of databases
  Name | Owner | Encoding | Collate | Ctype | Access privileges
-----|-----|-----|-----|-----|-----
postgres | postgres | UTF8 | English_United States.1252 | English_United States.1252 | 
template0 | postgres | UTF8 | English_United States.1252 | English_United States.1252 | =c/postgres +
template1 | postgres | UTF8 | English_United States.1252 | English_United States.1252 | =c/postgres +
(3 rows)

postgres=# create database demodb;
CREATE DATABASE
postgres=# \l
      List of databases
  Name | Owner | Encoding | Collate | Ctype | Access privileges
-----|-----|-----|-----|-----|-----
demodb | postgres | UTF8 | English_United States.1252 | English_United States.1252 | 
postgres | postgres | UTF8 | English_United States.1252 | English_United States.1252 | =c/postgres +
template0 | postgres | UTF8 | English_United States.1252 | English_United States.1252 | postgres=Ctc/postgres +
template1 | postgres | UTF8 | English_United States.1252 | English_United States.1252 | =c/postgres +
(4 rows)

postgres=# \c demodb
You are now connected to database "demodb" as user "postgres".
demodb=#
```

DELETING A DATABASE:-

```
Select SQL Shell (psql)

postgres=# \c demodb
You are now connected to database "demodb" as user "postgres".
demodb=# create database test;
CREATE DATABASE
demodb=# \l
      List of databases
  Name | Owner | Encoding | Collate | Ctype | Access privileges
-----|-----|-----|-----|-----|-----
 demodb | postgres | UTF8 | English_United_States.1252 | English_United_States.1252 | 
 postgres | postgres | UTF8 | English_United_States.1252 | English_United_States.1252 | 
 template0 | postgres | UTF8 | English_United_States.1252 | English_United_States.1252 | =c/postgres +
 template1 | postgres | UTF8 | English_United_States.1252 | English_United_States.1252 | postgres=Ctc/postgres +
 test | postgres | UTF8 | English_United_States.1252 | English_United_States.1252 | postgres=Ctc/postgres
(5 rows)

demodb=# drop database test;
DROP DATABASE
demodb=# \l
      List of databases
  Name | Owner | Encoding | Collate | Ctype | Access privileges
-----|-----|-----|-----|-----|-----
 demodb | postgres | UTF8 | English_United_States.1252 | English_United_States.1252 | 
 postgres | postgres | UTF8 | English_United_States.1252 | English_United_States.1252 | 
 template0 | postgres | UTF8 | English_United_States.1252 | English_United_States.1252 | =c/postgres +
 template1 | postgres | UTF8 | English_United_States.1252 | English_United_States.1252 | postgres=Ctc/postgres +
(4 rows)

demodb=#
```

CREATING AND DATA INSERTING INTO TABLE:-

```
Select SQL Shell (psql)

demodb=# CREATE TABLE students(name text, number int, age int);
CREATE TABLE
demodb=# \d
      List of relations
 Schema | Name | Type | Owner
-----|-----|-----|-----
 public | students | table | postgres
(1 row)

demodb=# INSERT INTO students(name,number,age) VALUES('Aarya', 1, 25);
INSERT 0 1
demodb=# INSERT INTO students(name,number,age) VALUES('Ruchita', 2, 22);
INSERT 0 1
demodb=# INSERT INTO students(name,number,age) VALUES('Issue', 3, 20);
INSERT 0 1
demodb=#
```

RETRIVING DATA AND DELETING CONTENT IN TABLE:-

```
Select SQL Shell (psql)
demo=# SELECT * FROM students;
 name | number | age
-----+-----+----
Aarya |      1 |  25
Ruchita |    2 |  22
Issue |    3 |  20
(3 rows)

demo=# SELECT name FROM students;
 name
-----
Aarya
Ruchita
Issue
(3 rows)

demo=# SELECT * FROM students WHERE age=22;
 name | number | age
-----+-----+----
Ruchita |    2 |  22
(1 row)

demo=# SELECT name FROM students WHERE number=2;
 name
-----
Ruchita
(1 row)

demo=# TRUNCATE TABLE students;
TRUNCATE TABLE
demo=# \d
List of relations
Schema | Name | Type | Owner
-----+-----+-----+-----
public | students | table | postgres
(1 row)

demo=# SELECT * FROM students;
 name | number | age
-----+-----+----
(0 rows)

demo=#
```

ACTIVATE AND DEACTIVATE ENV:-

```
Select C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5965]
(c) Microsoft Corporation. All rights reserved.

D:\Python\postgres_db>pip install virtualenv
Requirement already satisfied: virtualenv in c:\users\admin\appdata\local\programs\python\python313\lib\site-packages (20.31.2)
Requirement already satisfied: distlib<1,>=0.3.7 in c:\users\admin\appdata\local\programs\python\python313\lib\site-packages (from virtualenv) (0.3.9)
Requirement already satisfied: filelock<4,>=3.12.2 in c:\users\admin\appdata\local\programs\python\python313\lib\site-packages (from virtualenv) (3.18.0)
Requirement already satisfied: platformdirs<5,>=3.9.1 in c:\users\admin\appdata\local\programs\python\python313\lib\site-packages (from virtualenv) (4.3.8)

D:\Python\postgres_db>virtualenv env
'virtualenv' is not recognized as an internal or external command,
operable program or batch file.

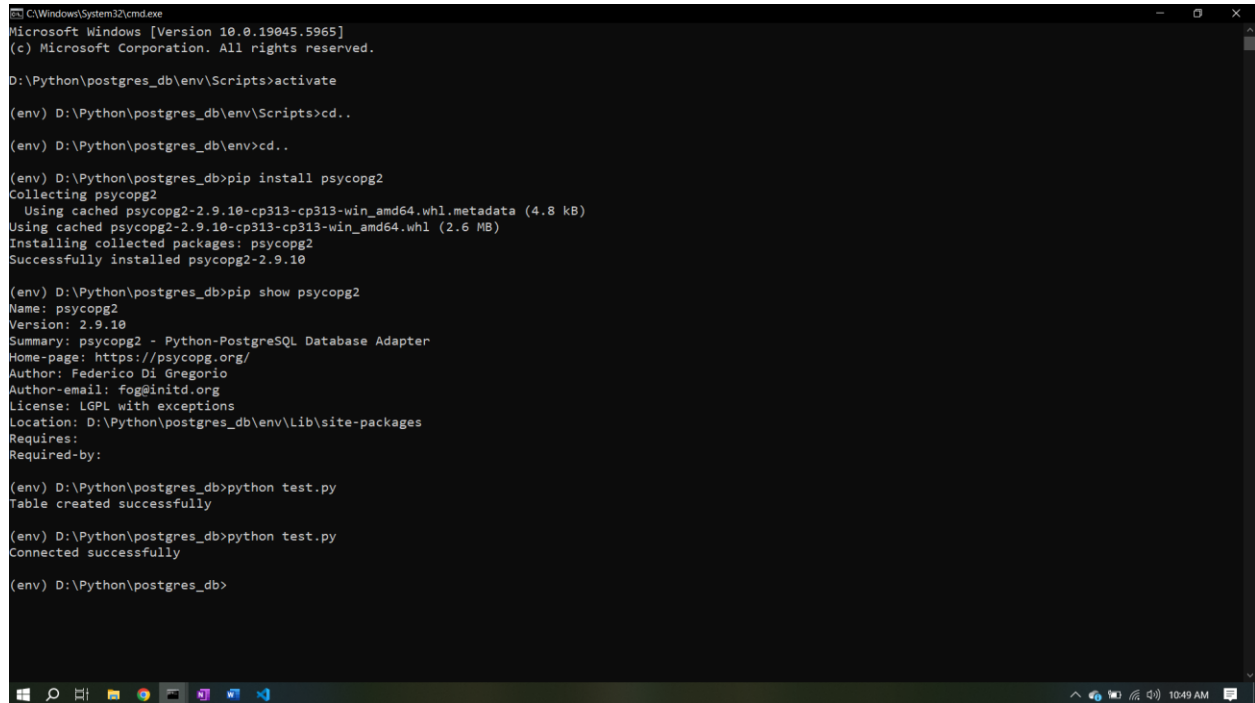
D:\Python\postgres_db>virtualenv env
created virtual environment CPython3.13.2.final.0-64 in 697ms
creator CPython3Windows(dest=D:\Python\postgres_db\env, clear=False, no_vcs_ignore=False, global=False)
seeder FromAppData(download=False, pip=25.1.1, via=copy, app_data_dir=C:\Users\Admin\AppData\Local\pip\virtualenv)
added seed packages: pip==25.1.1
activators BashActivator,BatchActivator,FishActivator,PowerShellActivator,PythonActivator

D:\Python\postgres_db>cd env
D:\Python\postgres_db\env>cd scripts
D:\Python\postgres_db\env\Scripts>activate
(env) D:\Python\postgres_db\env\Scripts>cd..
(env) D:\Python\postgres_db\env>cd..
(env) D:\Python\postgres_db>python test.py
python
(env) D:\Python\postgres_db>deactivate
D:\Python\postgres_db>cd env
D:\Python\postgres_db\env>cd scripts
D:\Python\postgres_db\env\Scripts>activate
(env) D:\Python\postgres_db\env\Scripts>
```

CONNECTING TO DATABASE:-

```
import psycopg2
# def table():
connect = psycopg2.connect(
```

```
    dbname= "postgres",
    user = "postgres",
    password = "Aashu@191010",
    host = "localhost",
    port = "5433"
)
print("Connected successfully")
```



The screenshot shows a Windows command prompt window with the following text:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5965]
(c) Microsoft Corporation. All rights reserved.

D:\Python\postgres_db\env\Scripts>activate

(env) D:\Python\postgres_db\env\Scripts>cd..

(env) D:\Python\postgres_db\env>cd..

(env) D:\Python\postgres_db>pip install psycopg2
Collecting psycopg2
  Using cached psycopg2-2.9.10-cp313-cp313-win_amd64.whl.metadata (4.8 kB)
Using cached psycopg2-2.9.10-cp313-cp313-win_amd64.whl (2.6 MB)
Installing collected packages: psycopg2
Successfully installed psycopg2-2.9.10

(env) D:\Python\postgres_db>pip show psycopg2
Name: psycopg2
Version: 2.9.10
Summary: psycopg2 - Python-PostgreSQL Database Adapter
Home-page: https://psycopg.org/
Author: Federico Di Gregorio
Author-email: fog@initd.org
License: LGPL with exceptions
Location: D:\Python\postgres_db\env\Lib\site-packages
Requires:
Required-by:

(env) D:\Python\postgres_db>python test.py
Table created successfully

(env) D:\Python\postgres_db>python test.py
Connected successfully

(env) D:\Python\postgres_db>
```

CREATING TABLE USING PYTHON:-

```
import psycopg2
def table():
    connect = psycopg2.connect(
        dbname= "postgres",
        user = "postgres",
        password = "Aashu@191010",
        host = "localhost",
        port = "5433"
    )
    cursor = connect.cursor()
    # Test the connection
    cursor.execute('''create table Employees(Name text,ID int,age int);''')
    print("Table created successfully")

    connect.commit()
```

```
connect.close()
table()
```

```
SQL Shell (psql)
Username [postgres]:
Password for user postgres:
psql (13.21)
WARNING: Console code page (437) differs from Windows code page (1252)
8-bit characters might not work correctly. See psql reference
page "Notes for Windows users" for details.
Type "help" for help.

postgres=# \l
      List of databases
  Name | Owner  | Encoding | Collate | Ctype | Access privileges
-----+-----+-----+-----+-----+-----
 demodb | postgres | UTF8     | English_United States.1252 | English_United States.1252 |
 postgres | postgres | UTF8     | English_United States.1252 | English_United States.1252 |
 template0 | postgres | UTF8     | English_United States.1252 | English_United States.1252 | =c/postgres +
 template1 | postgres | UTF8     | English_United States.1252 | English_United States.1252 | postgres=Ctc/postgres +
(4 rows)

postgres=# \d
      List of relations
 Schema | Name      | Type  | Owner
-----+-----+-----+-----
 public | employees | table | postgres
(1 row)

postgres=#
```

INSERTING THE DATA USING PYTHON:-

```
def data():
    connect = psycopg2.connect(
        dbname= "postgres",
        user = "postgres",
        password = "Aashu@191010",
        host = "localhost",
        port = "5433"
    )

    cursor = connect.cursor()

    # Test the connection
    # Insert data into the table
    cursor.execute('''insert into Employees(Name, ID, age)
values('sam',1,27);''')
    print("data added successfully")

    connect.commit()
    connect.close()
```

data()

The screenshot shows a development environment with VS Code and a SQL Shell. In VS Code, a Python file named `test.py` is open, containing a `table()` function that connects to a PostgreSQL database, creates an `employees` table, and inserts a record for 'sam' with ID 1 and age 27. The terminal shows the command `python test.py` being executed, resulting in the output `data added successfully`. Below the VS Code window, the SQL Shell (psql) is open, showing the database structure and the result of a `select` query on the `employees` table, which returns one row: 'sam' with ID 1 and age 27.

```
test.py
1 def table():
2     cursor = connect.cursor()
3     # Test the connection
4     cursor.execute('create table Employees(Name text, ID int, age int);')
5     print("Table created successfully")
6     # Insert data into the table
7     cursor.execute('insert into Employees(Name, ID, age) values('sam',1,27);')
8     print("data added successfully")
9
10    connect.commit()
11    connect.close()
12    table()
13
14
15
16
17
18
19
20
21
```

```
PS D:\Python\postgres_db> & D:\Python\postgres_db\env\Scripts\Activate.ps1
(env) PS D:\Python\postgres_db> python test.py
data added successfully
(env) PS D:\Python\postgres_db>
```

```
SQL Shell (psql)
Username [postgres]:
Password for user postgres:
psql (13.21)
WARNING: Console code page (437) differs from Windows code page (1252)
8-bit characters might not work correctly. See psql reference
page "Notes for Windows users" for details.
Type "help" for help.

postgres=# \d
          List of relations
 Schema | Name      | Type  | Owner
-----|-----|-----|-----
 public | employees | table | postgres
(1 row)

postgres=# select * from employees;
 name | id | age
-----+---+----
(0 rows)

postgres=# select * from employees;
 name | id | age
-----+---+----
 sam  | 1  | 27
(1 row)

postgres=#
```

EXTRACTING DATA FROM DATABASE:-

```
def extract():
```

```

connect = psycopg2.connect(
    dbname= "postgres",
    user = "postgres",
    password = "Aashu@191010",
    host = "localhost",
    port = "5433"
)

cursor = connect.cursor()
# Test the connection
# extract data from the table
cursor.execute('''select * from Employees;''')
# print(cursor.fetchone())
show = cursor.fetchone()
print("Name:", show[0])
print("ID:", show[1])
print("Age:", show[2])

# print(cursor.fetchall())
# print("data added successfully")

connect.commit()
connect.close()
extract()

```

The screenshot shows a Visual Studio Code editor window with a file named `test.py` open. The file contains a Python script that connects to a PostgreSQL database using `psycopg2`, tests the connection, and extracts data from a table named `Employees`. The script prints the first row of data and then prints "data added successfully".

The terminal output shows the command `python test.py` being executed in a PowerShell environment. The output displays the extracted data: Name: sam, ID: 1, Age: 27. The terminal also shows the command `python test.py` being executed again, which prints "data added successfully".

```

def extract():
    connect = psycopg2.connect(
        dbname= "postgres",
        user = "postgres",
        password = "Aashu@191010",
        host = "localhost",
        port = "5433"
    )

    cursor = connect.cursor()
    # Test the connection
    # extract data from the table
    cursor.execute('''select * from Employees;''')
    # print(cursor.fetchone())
    show = cursor.fetchone()
    print("Name:", show[0])
    print("ID:", show[1])
    print("Age:", show[2])

    # print(cursor.fetchall())
    # print("data added successfully")

    connect.commit()
    connect.close()
extract()

```

```

PS D:\Python\postgres_db> & D:\Python\postgres_db\env\Scripts\Activate.ps1
(env) PS D:\Python\postgres_db> python test.py
data added successfully
(env) PS D:\Python\postgres_db> python test.py
Name: sam
ID: 1
Age: 27
(env) PS D:\Python\postgres_db>

```

ADDING INPUT FROM USER:-

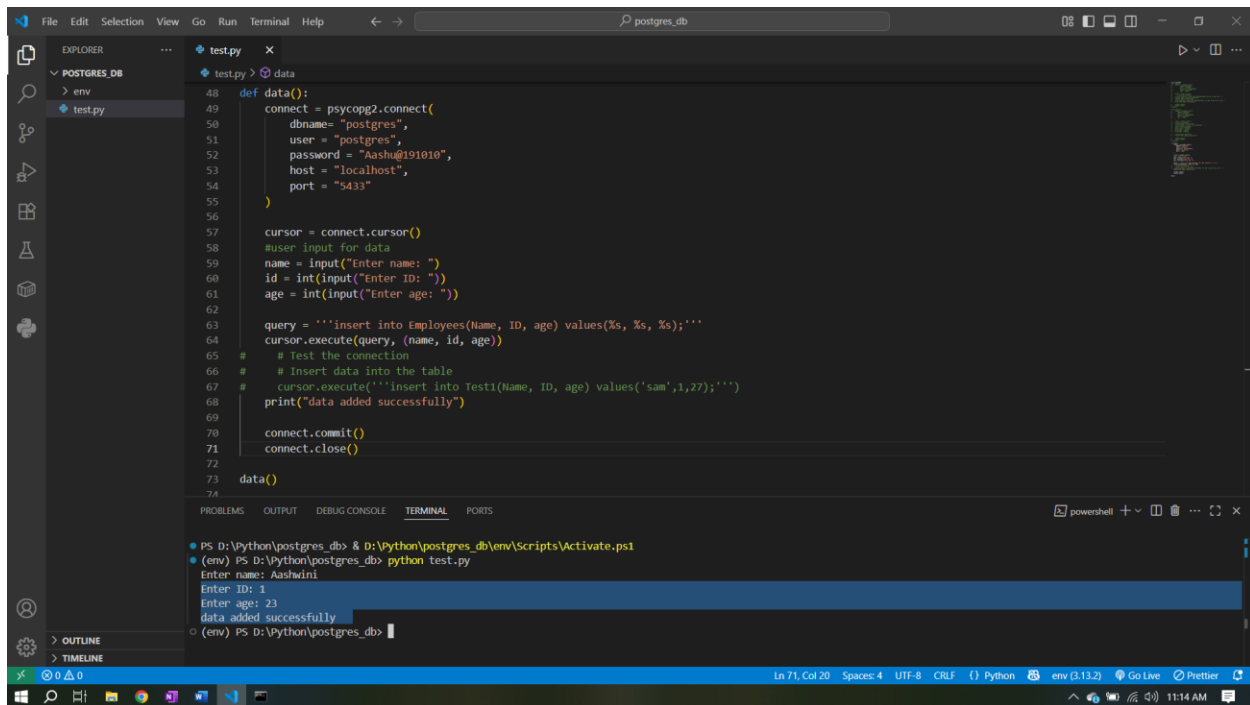
```
def data():
    connect = psycopg2.connect(
        dbname= "postgres",
        user = "postgres",
        password = "Aashu@191010",
        host = "localhost",
        port = "5433"
    )

    cursor = connect.cursor()
    #user input for data
    name = input("Enter name: ")
    id = int(input("Enter ID: "))
    age = int(input("Enter age: "))

    query = '''insert into Employees(Name, ID, age) values(%s, %s, %s);'''
    cursor.execute(query, (name, id, age))
#    # Test the connection
#    # Insert data into the table
#    cursor.execute('''insert into Test1(Name, ID, age) values('sam',1,27);''')
    print("data added successfully")

    connect.commit()
    connect.close()

data()
```

```
File Edit Selection View Go Run Terminal Help postgres_db
EXPLORER
  postgres_db
    env
    test.py
test.py
48 def data():
49     connect = psycopg2.connect(
50         dbname="postgres",
51         user="postgres",
52         password="Aashu@191010",
53         host="localhost",
54         port="5433"
55     )
56
57     cursor = connect.cursor()
58     #user input for data
59     name = input("Enter name: ")
60     id = int(input("Enter ID: "))
61     age = int(input("Enter age: "))
62
63     query = '''insert into Employees(Name, ID, age) values(%s, %s, %s);'''
64     cursor.execute(query, (name, id, age))
65     # Test the connection
66     # Insert data into the table
67     cursor.execute('insert into Test1(Name, ID, age) values('sam',1,27);')
68     print("data added successfully")
69
70     connect.commit()
71     connect.close()
72
73 data()
74
TERMINAL
PS D:\Python\postgres_db> & D:\Python\postgres_db\env\Scripts\Activate.ps1
(env) PS D:\Python\postgres_db> python test.py
Enter name: Aashwini
Enter ID: 1
Enter age: 23
data added successfully
(env) PS D:\Python\postgres_db>
```

EXTRACTING DATA IN TEST1 TABLE:-

```
def extract():
    connect = psycopg2.connect(
        dbname="postgres",
        user="postgres",
        password="Aashu@191010",
        host="localhost",
        port="5433"
    )

    cursor = connect.cursor()
    # Test the connection
    # extract data from the table
    cursor.execute('select * from Employees;')
    show = cursor.fetchall()
    for i in show:
        print(i)
    print("data extracted successfully")
    connect.commit()
    connect.close()
extract()
```

The image shows a Visual Studio Code editor window with a file explorer on the left and a terminal at the bottom. The file explorer shows a project named 'POSTGRES_DB' with a file 'test.py'. The terminal shows the execution of 'python test.py' in a PowerShell environment, displaying the output of the script.

```
File Edit Selection View Go Run Terminal Help postgres_db
EXPLORER
POSTGRES_DB
  env
  test.py
test.py
74
75 def extract():
76     connect = psycopg2.connect(
77         dbname="postgres",
78         user="postgres",
79         password="Aashu@191010",
80         host="localhost",
81         port="5433"
82     )
83
84     cursor = connect.cursor()
85     # Test the connection
86     # extract data from the table
87     cursor.execute('select * from Employees;')
88     show = cursor.fetchall()
89     for i in show:
90         print(i)
91     print("data extracted successfully")
92     connect.commit()
93     connect.close()
94     extract()
95
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Enter ID: 1
Enter age: 23
data added successfully
(env) PS D:\Python\postgres_db> python test.py
('sam', 1, 27)
('Aashwini', 1, 23)
('Aashwini', 1, 23)
data extracted successfully
(env) PS D:\Python\postgres_db>
```

Ln 90, Col 17 Spaces: 4 UTF-8 CRLF Python env (3.13.2) Go Live Prettier 11:17 AM