

ASSIGNMENT-7

Module 16: Building Database Apps with PostgreSQL & Python:

Creating a database: create database database_name;

```
SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Password for user postgres:

psql (18.1)
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 | Locale Provider | Collate | Ctype | Locale | ICU Rules | Access privileges
-----+-----+-----+-----+-----+-----+-----+-----+-----
 postgres | postgres | UTF8 | libc | English_United States.1252 | English_United States.1252 |  |  | =c/postgres
 template0 | postgres | UTF8 | libc | English_United States.1252 | English_United States.1252 |  |  | postgres=Ctc/postgres
 template1 | postgres | UTF8 | libc | English_United States.1252 | English_United States.1252 |  |  | =c/postgres
 postgres=Ctc/postgres
(3 rows)

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

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

Deleting a database: drop database database_name;

```
SQL Shell (psql)
You are now connected to database "demodb" as user "postgres".
demodb=# create database testdb;
CREATE DATABASE
demodb=# \l
      List of databases
  Name | Owner | Encoding | Locale Provider | Collate | Ctype | Locale | ICU Rules | Access privileges
-----+-----+-----+-----+-----+-----+-----+-----+-----
 demodb | postgres | UTF8 | libc | English_United States.1252 | English_United States.1252 |  |  | =c/postgres
 postgres | postgres | UTF8 | libc | English_United States.1252 | English_United States.1252 |  |  | postgres=Ctc/postgres
 template0 | postgres | UTF8 | libc | English_United States.1252 | English_United States.1252 |  |  | =c/postgres
 template1 | postgres | UTF8 | libc | English_United States.1252 | English_United States.1252 |  |  | postgres=Ctc/postgres
 testdb | postgres | UTF8 | libc | English_United States.1252 | English_United States.1252 |  |  | 
(5 rows)

demodb=# drop database testdb;
DROP DATABASE
demodb=# \l
      List of databases
  Name | Owner | Encoding | Locale Provider | Collate | Ctype | Locale | ICU Rules | Access privileges
-----+-----+-----+-----+-----+-----+-----+-----+-----
 demodb | postgres | UTF8 | libc | English_United States.1252 | English_United States.1252 |  |  | =c/postgres
 postgres | postgres | UTF8 | libc | English_United States.1252 | English_United States.1252 |  |  | postgres=Ctc/postgres
 template0 | postgres | UTF8 | libc | English_United States.1252 | English_United States.1252 |  |  | =c/postgres
 template1 | postgres | UTF8 | libc | English_United States.1252 | English_United States.1252 |  |  | postgres=Ctc/postgres
(4 rows)

demodb=#
```

Creating table and adding data:

CREATE TABLE TABLE_NAME(COLUMN_NAMES);

INSERT INTO TABLE_NAME VALUES(COLUMN_DATA);

```
postgres=# create database student;
CREATE DATABASE
postgres=# \c student
You are now connected to database "student" as user "postgres".
student=# CREATE TABLE students(name text,number int,age int);
CREATE TABLE
student=# \d
          List of relations
  Schema | Name      | Type  | Owner
  -----+-----+-----+-----
 public | students | table | postgres
(1 row)

student=# INSERT INTO students(name,number,age) VALUES('Pragathi',528,24);
INSERT 0 1
student=# INSERT INTO students(name,number,age) VALUES('Nithish',514,25);
INSERT 0 1
student=#
```

Retrieving data from database and deleting contents in the table:

SELECT * FROM TABLE_NAME;

TRUNCATE TABLE TABLE_NAME;

```
SQL Shell (psql)
student=#
student=# SELECT * FROM students;
   name | number | age
  -----+-----+-----
 Pragathi | 528 | 24
  Nithish | 514 | 25
(2 rows)

student=# SELECT name FROM students;
 name
  -----
 Pragathi
  Nithish
(2 rows)

student=# SELECT * FROM students WHERE number=528;
   name | number | age
  -----+-----+-----
 Pragathi | 528 | 24
(1 row)

student=# SELECT * FROM students WHERE age=24;
   name | number | age
  -----+-----+-----
 Pragathi | 528 | 24
(1 row)

student=# SELECT number FROM students where age=24;
 number
  -----
    528
(1 row)

student=# SELECT number FROM students where name='Nithish';
 number
  -----
```

```
SQL Shell (psql)
(1 row)

student=# SELECT number FROM students where name='Nithish';
 number
-----
    514
(1 row)

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

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

student=# |
```

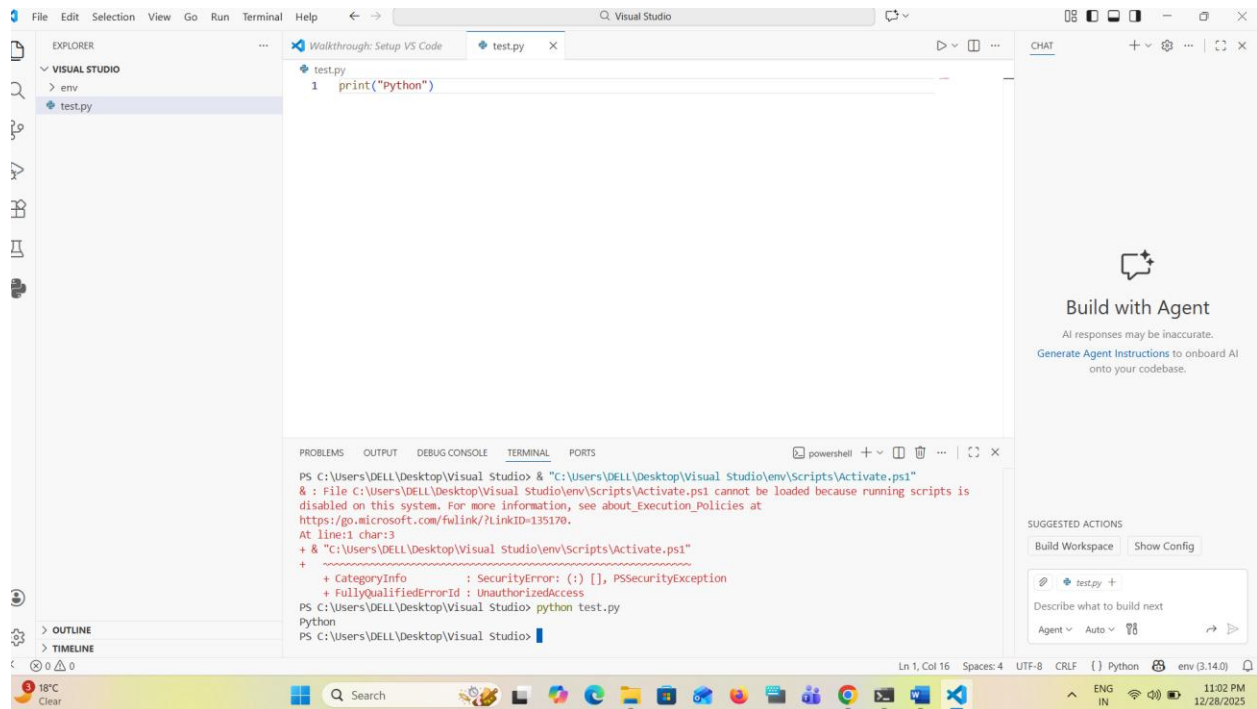
Setting up virtualenv:

```
Command Prompt
Downloading virtualenv-20.35.4-py3-none-any.whl (6.0 MB)
6.0/6.0 MB 8.7 MB/s 0:00:00
Downloading distlib-0.4.0-py2.py3-none-any.whl (469 kB)
Downloading filelock-3.20.1-py3-none-any.whl (16 kB)
Installing collected packages: distlib, filelock, virtualenv
Successfully installed distlib-0.4.0 filelock-3.20.1 virtualenv-20.35.4

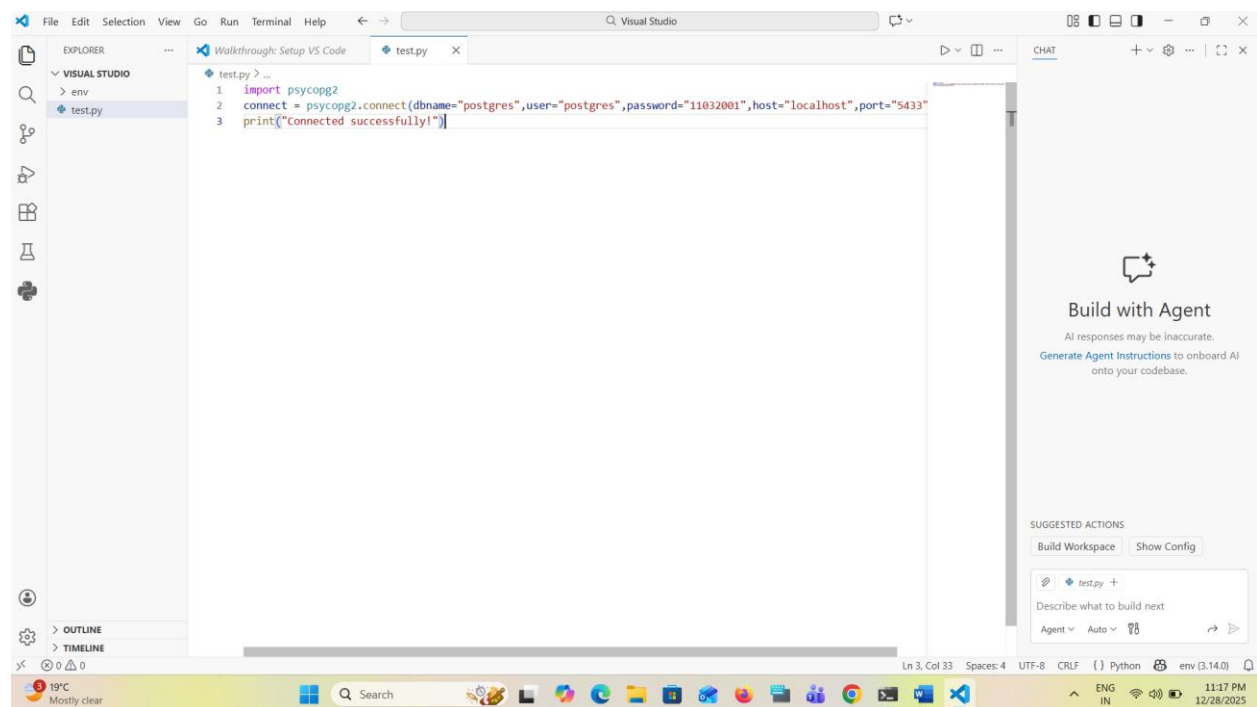
[notice] A new release of pip is available: 25.2 -> 25.3
[notice] To update, run: python.exe -m pip install --upgrade pip

C:\Users\DELL\Desktop\Visual Studio>virtualenv env
created virtual environment CPython3.14.0.final.0-64 in 19460ms
creator CPython3Windows(dest=C:\Users\DELL\Desktop\Visual Studio\env, clear=False, no_vcs_ignore=False, global=False)
seeder FromAppData(download=False, pip=bundle, via=copy, app_data_dir=C:\Users\DELL\AppData\Local\pypa\virtualenv)
added seed packages: pip==25.3
activators BashActivator,BatchActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator

C:\Users\DELL\Desktop\Visual Studio>cd env
C:\Users\DELL\Desktop\Visual Studio\env>cd scripts
C:\Users\DELL\Desktop\Visual Studio\env\Scripts>activate
(env) C:\Users\DELL\Desktop\Visual Studio\env\Scripts>cd..
(env) C:\Users\DELL\Desktop\Visual Studio\env>cd..
(env) C:\Users\DELL\Desktop\Visual Studio>python test.py
Python
(env) C:\Users\DELL\Desktop\Visual Studio>python test.py
Python
(env) C:\Users\DELL\Desktop\Visual Studio>deactivate
C:\Users\DELL\Desktop\Visual Studio>cd env
C:\Users\DELL\Desktop\Visual Studio\env>cd scripts
C:\Users\DELL\Desktop\Visual Studio\env\Scripts>activate
(env) C:\Users\DELL\Desktop\Visual Studio\env\Scripts>
```



Connecting to the database:



```
Command Prompt
Microsoft Windows [Version 10.0.26200.7462]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DELL>cd desktop
C:\Users\DELL\Desktop>cd visual studio
C:\Users\DELL\Desktop\Visual Studio>cd env
C:\Users\DELL\Desktop\Visual Studio\env>cd scripts
C:\Users\DELL\Desktop\Visual Studio\env\Scripts>activate
(env) C:\Users\DELL\Desktop\Visual Studio\env\Scripts>cd..
(env) C:\Users\DELL\Desktop\Visual Studio\env>cd..
(env) C:\Users\DELL\Desktop\Visual Studio>python test.py
Connected successfully!
(env) C:\Users\DELL\Desktop\Visual Studio>
```

Creating table using Python:

Python code:

```
import psycopg2
```

```
connect =  
psycopg2.connect(dbname="postgres",user="postgres",password="11032001",host="localhost",port="5433")
```

```
cursor = connect.cursor()
```

```
cursor.execute("""create table employees(Name text, ID int, Age int);""")
```

```
print("Table created successfully")
```

```
connect.commit()
```

```
connect.close()
```

```
Command Prompt
Microsoft Windows [Version 10.0.26200.7462]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DELL>cd desktop
C:\Users\DELL\Desktop>cd visual studio
C:\Users\DELL\Desktop\Visual Studio>cd env
C:\Users\DELL\Desktop\Visual Studio\env>cd scripts
C:\Users\DELL\Desktop\Visual Studio\env\Scripts>activate
(env) C:\Users\DELL\Desktop\Visual Studio\env\Scripts>cd..
(env) C:\Users\DELL\Desktop\Visual Studio\env>cd..
(env) C:\Users\DELL\Desktop\Visual Studio>python test.py
Connected successfully!
(env) C:\Users\DELL\Desktop\Visual Studio>python test.py
Table created successfully
(env) C:\Users\DELL\Desktop\Visual Studio>
```

Inserting the data using python:

Python code:

```
import psycopg2
```

```
def table():
```

```
    connect =
    psycopg2.connect(dbname="postgres",user="postgres",password="11032001",host="localhost",port="5433")
```

```
    cursor = connect.cursor()
```

```
    cursor.execute("""create table employees(Name text, ID int, Age int);""")
```

```
    print("Table created successfully")
```

```
    connect.commit()
```

```
    connect.close()
```

```
def data():
```

```
connect =
psycopg2.connect(dbname="postgres",user="postgres",password="11032001",host="localhost",port="5
433")

cursor = connect.cursor()

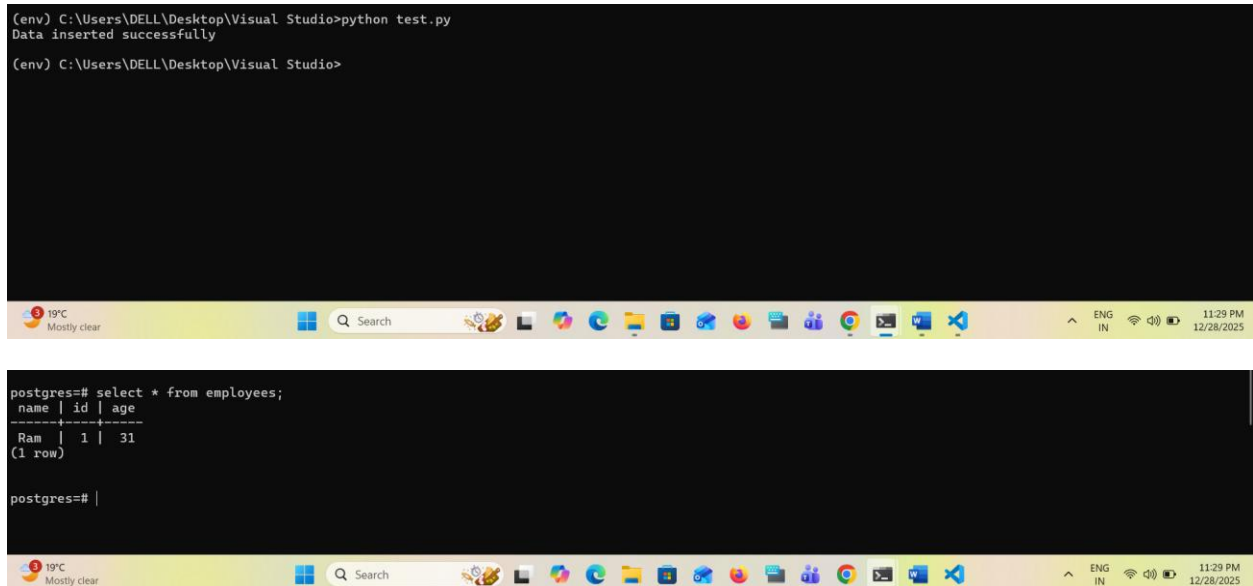
cursor.execute("""insert into employees(Name,ID,Age) values('Ram',01,31);""")

print("Data inserted successfully")

connect.commit()

connect.close()

data()
```



The image consists of two screenshots of a Windows 11 desktop. The top screenshot shows a terminal window with the command prompt running a Python script. The output of the script is "Data inserted successfully". The bottom screenshot shows a terminal window with a PostgreSQL query result. The query is "select * from employees;" and the result is a table with columns "name", "id", and "age". The table contains one row with the values "Ram", "1", and "31".

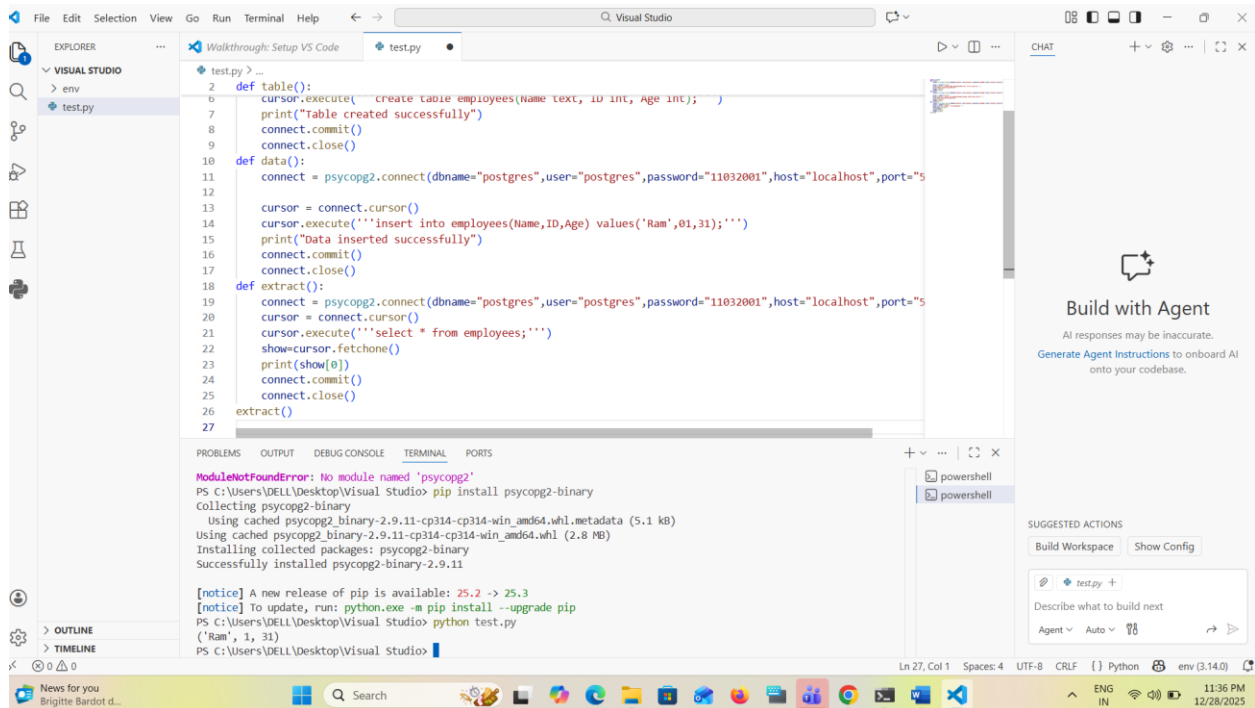
```
(env) C:\Users\DELL\Desktop\Visual Studio>python test.py
Data inserted successfully

(env) C:\Users\DELL\Desktop\Visual Studio>
```

```
postgres=# select * from employees;
 name | id | age 
-----+---+----
  Ram |  1 |  31 
(1 row)

postgres=# |
```

Extracting data from database:



The screenshot shows the Visual Studio Code interface with a Python file named `test.py` and a terminal window. The Python code defines functions for creating a table, inserting data, and extracting data from a PostgreSQL database. The terminal window shows the installation of the `psycopg2` binary using `pip install psycopg2-binary`.

```
def table():
    cursor.execute('create table employees(Name text, ID int, Age int);')
    print("Table created successfully")
    connect.commit()
    connect.close()

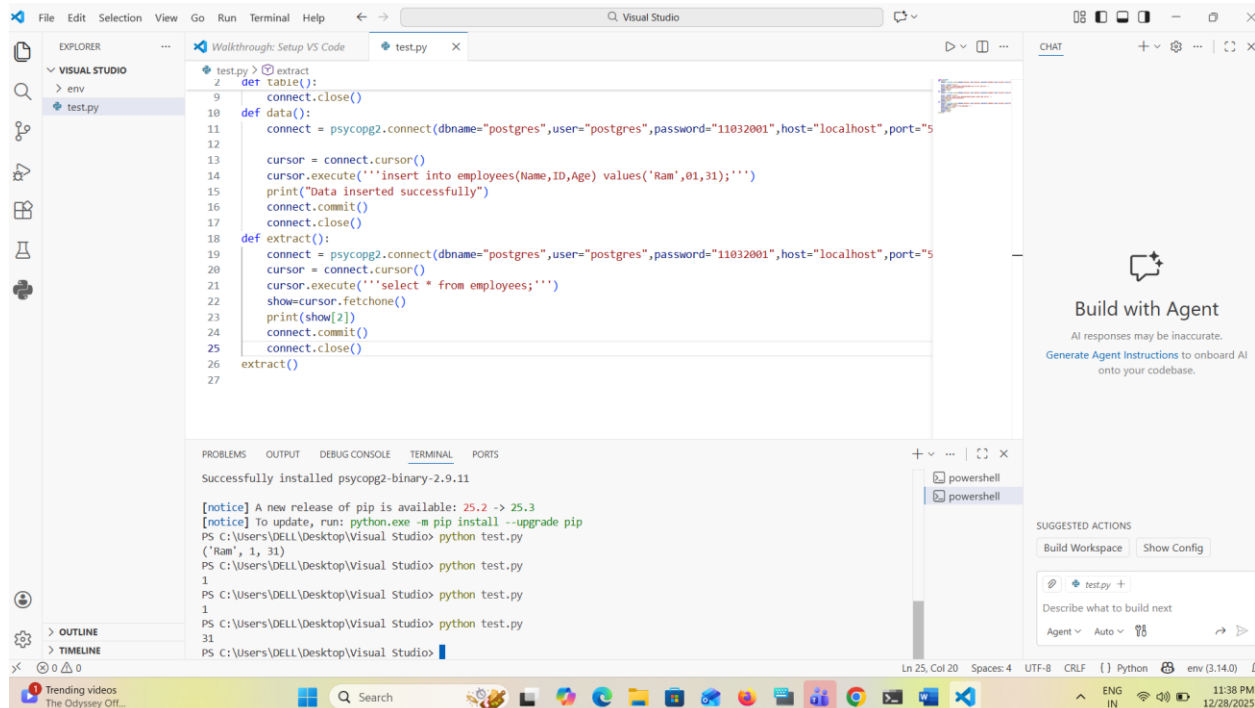
def data():
    connect = psycopg2.connect(dbname="postgres",user="postgres",password="11032001",host="localhost",port="5432")
    cursor = connect.cursor()
    cursor.execute('insert into employees(Name,ID,Age) values('Ram',01,31);')
    print("Data inserted successfully")
    connect.commit()
    connect.close()

def extract():
    connect = psycopg2.connect(dbname="postgres",user="postgres",password="11032001",host="localhost",port="5432")
    cursor = connect.cursor()
    cursor.execute('select * from employees;')
    show=cursor.fetchone()
    print(show[0])
    connect.commit()
    connect.close()

extract()
```

```
PS C:\Users\DELL\Desktop\Visual Studio> pip install psycopg2-binary
Collecting psycopg2-binary
  Using cached psycopg2_binary-2.9.11-cp314-cp314-win_amd64.whl.metadata (5.1 kB)
Using cached psycopg2_binary-2.9.11-cp314-cp314-win_amd64.whl (2.8 MB)
Installing collected packages: psycopg2-binary
Successfully installed psycopg2-binary-2.9.11

[notice] A new release of pip is available: 25.2 -> 25.3
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\DELL\Desktop\Visual Studio> python test.py
('Ram', 1, 31)
PS C:\Users\DELL\Desktop\Visual Studio>
```

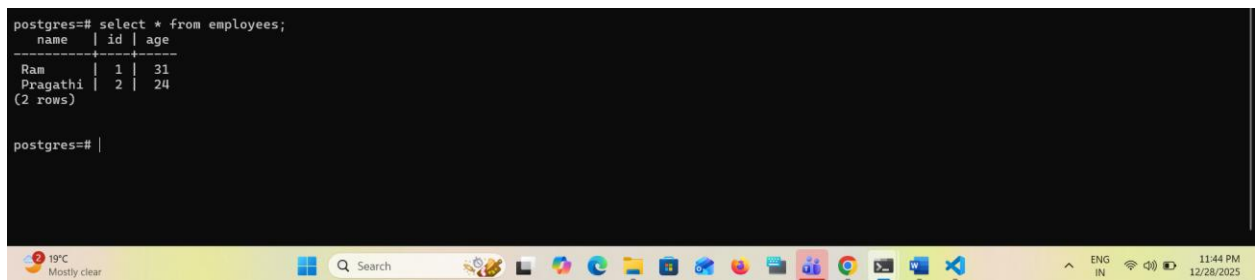
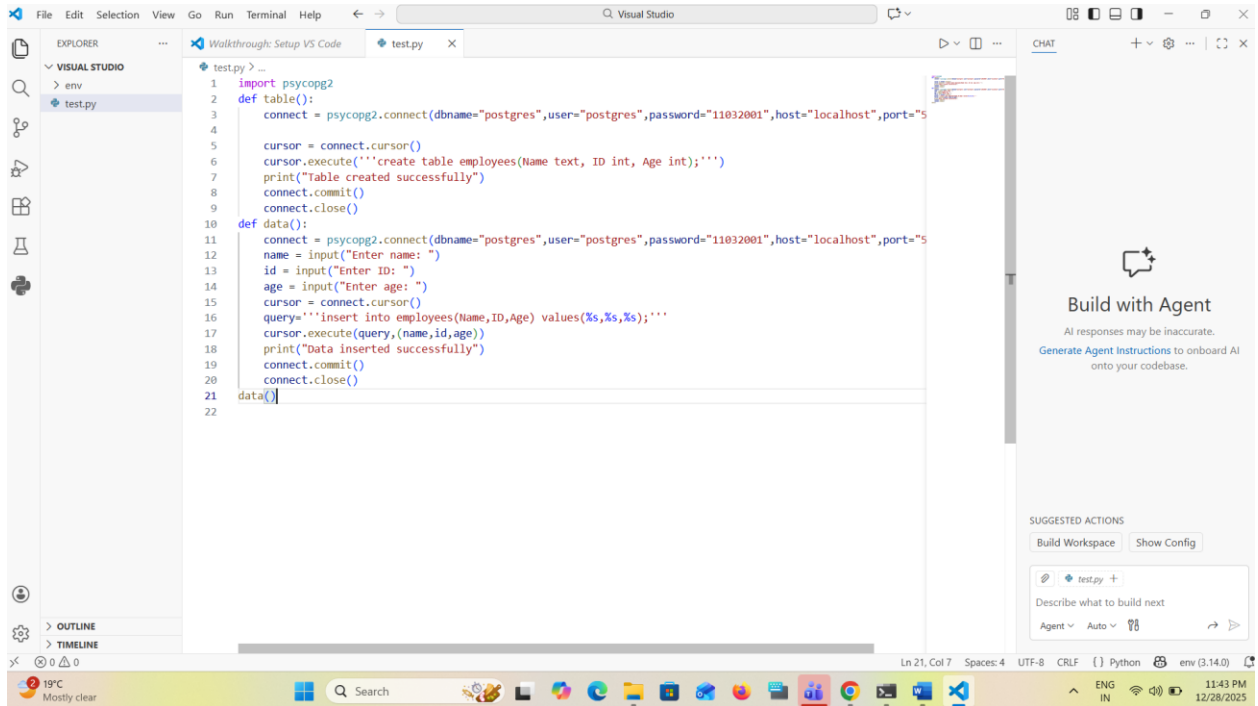


The screenshot shows the Visual Studio Code interface with the same Python file `test.py` and a terminal window. The terminal window now shows the successful installation of the `psycopg2` binary, followed by running the `python test.py` command, which outputs the result of the data extraction: `('Ram', 1, 31)`.

```
Successfully installed psycopg2-binary-2.9.11

[notice] A new release of pip is available: 25.2 -> 25.3
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\DELL\Desktop\Visual Studio> python test.py
('Ram', 1, 31)
PS C:\Users\DELL\Desktop\Visual Studio> python test.py
('Ram', 1, 31)
PS C:\Users\DELL\Desktop\Visual Studio> python test.py
('Ram', 1, 31)
PS C:\Users\DELL\Desktop\Visual Studio>
```


Adding input from user:



S