

ASSIGNMENT 7:

Module 16: Building Database Apps with PostgreSQL & Python

Description: For this assignment, you are required to:

1. Complete all course lectures within the module, ensuring you follow the concepts taught in each session.
2. Implement all practical exercises provided in the course lectures. **Each practical task should be executed and the results properly documented.**
3. **Take complete, unedited screenshots of each practical output along with the corresponding code.** Ensure that the following are visible in your screenshots:
 - The **full code** you executed.
 - The **output displayed** on your system, including the **system timestamp**.
 - **No cropping or editing** should be done on the screenshots. Any attempt to crop or modify the screenshots will result in your submission being considered invalid, and you will be required to resubmit it.

Submission: You should either:

- **Zip the folder** containing all your practical screenshots, **or**
- **Prepare a PDF** containing all the screenshots and code.

Ensure **the screenshots are organized clearly and in the correct order** according to the sequence of lectures and practicals. Lastly, upload it to drive and then submit the **accessible** link (**make sure to enable sharing access**).

1. Installed PostgreSQL

The screenshot shows a Windows desktop environment. In the background, a web browser is open to a TuteDude course page for Python. The page has a purple header with the course name and a progress bar indicating 45% completion. A yellow banner at the top of the page contains a WhatsApp contact number. In the foreground, a file explorer window is open, displaying the 'SQL Shell (psql)' app. The taskbar at the bottom shows various icons, including the Start button, task view, and several open applications. The system clock in the bottom right corner shows the date as Monday, November 24, 2025, and the time as 11:39:14 AM.

Python

14h 20m 57s WATCHED

Old Chat

Course Progress 45%

If you are facing any issues with lectures, mentors or technical, WhatsApp at +91 9296214803

Module 15: CALCULATOR USING TKINTER
28m 49s | 3 / 3 lectures

Assignment 6
Tkinter calculator Under Review

Module 16: Building Database Apps with PostgreSQL & Python
1h 13m 4s | 15 / 15 lectures

- 1. Introduction to data
- 2. Introduction to database
- 3. Introduction to PostgreSQL
- 4. Installing PostgreSQL
- 5. Creating a database
- 6. Deleting a database
- 7. Creating table and adding data
- 8. Retrieving data from database and deleting contents in the table
- 9. Setting up virtualenv
- 10. Installing psycopg2
- 11. Connecting to the database
- 12. Creating table using python
- 13. Inserting the data using python
- 14. Extracting the data from the database
- 15. Adding the input from the user

Live 1:1 Mentorship

Chat with Mentor

Mon, Nov 24 2025 | 11:39:14 AM

2. Creating a database

```
SQL Shell (psql)
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_India.1252 | English_India.1252 |          |           |
 student  | postgres | UTF8     | libc            | English_India.1252 | English_India.1252 |          |           | =c/postgres +
 template0 | postgres | UTF8     | libc            | English_India.1252 | English_India.1252 |          |           | postgres=CTc/postgres
 template1 | postgres | UTF8     | libc            | English_India.1252 | English_India.1252 |          |           | =c/postgres +
                                           |                               |           |           | postgres=CTc/postgres
(4 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_India.1252 | English_India.1252 |          |           |
 postgres | postgres | UTF8     | libc            | English_India.1252 | English_India.1252 |          |           |
 student  | postgres | UTF8     | libc            | English_India.1252 | English_India.1252 |          |           | =c/postgres +
 template0 | postgres | UTF8     | libc            | English_India.1252 | English_India.1252 |          |           | postgres=CTc/postgres
 template1 | postgres | UTF8     | libc            | English_India.1252 | English_India.1252 |          |           | =c/postgres +
                                           |                               |           |           | postgres=CTc/postgres
(5 rows)

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

3. Deleting a database

```
SQL Shell (psql)
```

Name	Owner	Encoding	Locale Provider	Collate	Ctype	Locale	ICU Rules	Access privileges
demodb	postgres	UTF8	libc	English_India.1252	English_India.1252			
postgres	postgres	UTF8	libc	English_India.1252	English_India.1252			
student	postgres	UTF8	libc	English_India.1252	English_India.1252			
template0	postgres	UTF8	libc	English_India.1252	English_India.1252			=c/postgres +
template1	postgres	UTF8	libc	English_India.1252	English_India.1252			postgres=CTc/postgres +

(5 rows)

```
postgres=# \c demodb
You are now connected to database "demodb" as user "postgres".
demodb=# create database test;
ERROR:  syntax error at or near "database"
LINE 1: create database test;
               ^
demodb=# create database test;
CREATE DATABASE
demodb=# \l
```

Name	Owner	Encoding	Locale Provider	Collate	Ctype	Locale	ICU Rules	Access privileges
demodb	postgres	UTF8	libc	English_India.1252	English_India.1252			
postgres	postgres	UTF8	libc	English_India.1252	English_India.1252			
student	postgres	UTF8	libc	English_India.1252	English_India.1252			
template0	postgres	UTF8	libc	English_India.1252	English_India.1252			=c/postgres +
template1	postgres	UTF8	libc	English_India.1252	English_India.1252			postgres=CTc/postgres +
test	postgres	UTF8	libc	English_India.1252	English_India.1252			=c/postgres +

(6 rows)

```
demodb=# drop database test;
DROP DATABASE
demodb=# \l
```

Name	Owner	Encoding	Locale Provider	Collate	Ctype	Locale	ICU Rules	Access privileges
demodb	postgres	UTF8	libc	English_India.1252	English_India.1252			
postgres	postgres	UTF8	libc	English_India.1252	English_India.1252			
student	postgres	UTF8	libc	English_India.1252	English_India.1252			
template0	postgres	UTF8	libc	English_India.1252	English_India.1252			=c/postgres +
template1	postgres	UTF8	libc	English_India.1252	English_India.1252			postgres=CTc/postgres +

(5 rows)

```
demodb=#
```

4. Creating table and adding data

```
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=# \
invalid command \
Try \? for help.
postgres=# \l

      List of databases
  Name | Owner  | Encoding | Locale Provider | Collate | Ctype | Locale | ICU Rules | Access privileges
-----+-----+-----+-----+-----+-----+-----+-----+-----
 postgres | postgres | UTF8 | libc | English_India.1252 | English_India.1252 |  |  | =c/postgres +
 template0 | postgres | UTF8 | libc | English_India.1252 | English_India.1252 |  |  | postgres=CTc/postgres
 template1 | postgres | UTF8 | libc | English_India.1252 | English_India.1252 |  |  | =c/postgres +
          |          |      |      |                  |                  |  |  | postgres=CTc/postgres
(3 rows)

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, rank 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, rank, age) values('Ronak', 04, 25);
INSERT 0 1
student=# insert into students(name, rank, age) values('Jatin', 06, 24);
INSERT 0 1
student=#
```

5. Retrieving data from database and deleting contents in the table

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

student=# select * from students where age=25;
 name | rank | age 
-----+-----+-----
 Ronak |    4 |   25 
(1 row)

student=# select rank from students where age=25;
 rank 
-----
    4 
(1 row)

student=# select rank from students where name=Jatin;
ERROR: column "jatin" does not exist
LINE 1: select rank from students where name=Jatin;
                                         ^
student=# select rank from students where name='Jatin';
 rank 
-----
    6 
(1 row)

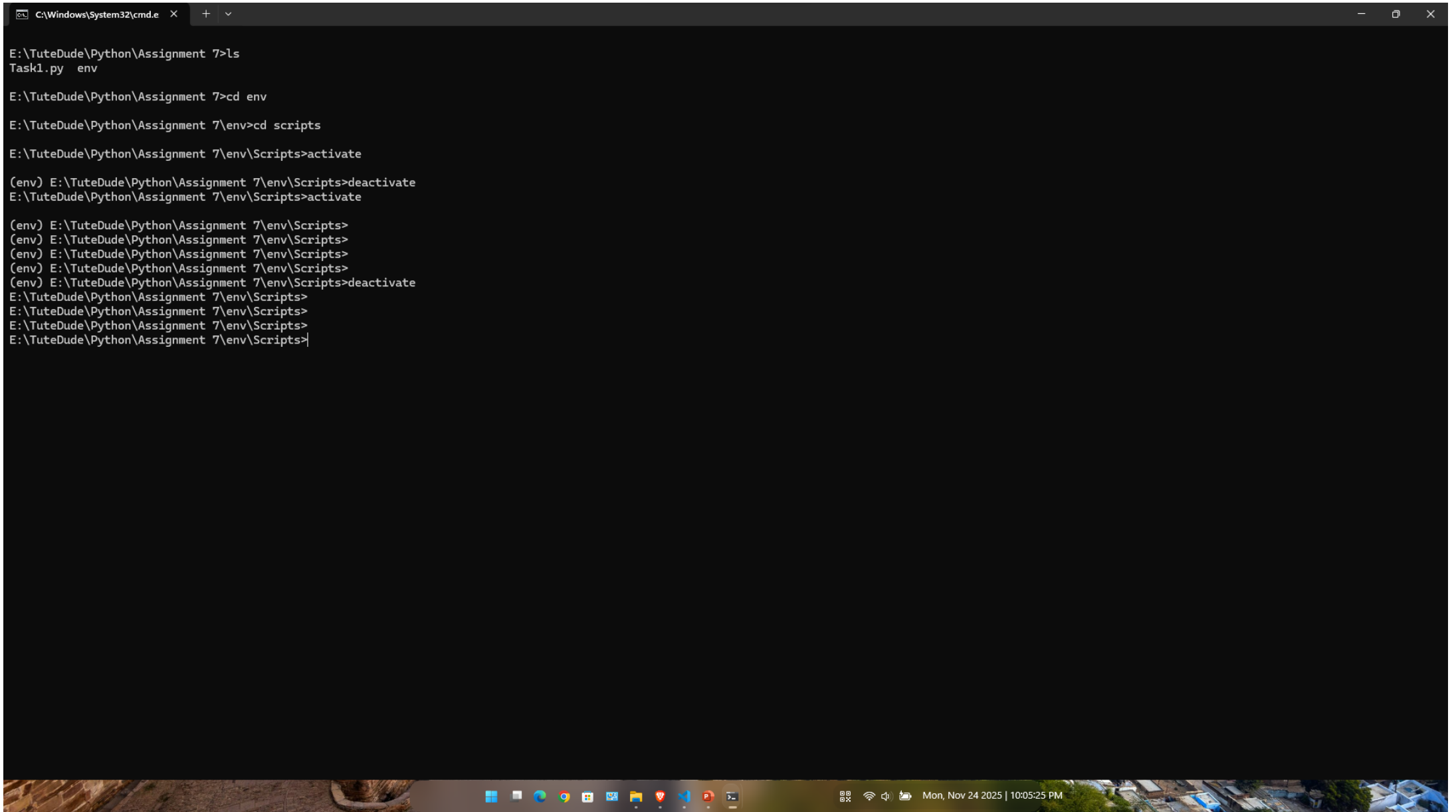
student=# truncate table students;
TRUNCATE TABLE
student=# \d
          List of relations
 Schema | Name   | Type  | Owner
-----+-----+-----+-----
 public | students | table | postgres
(1 row)

student=# select 8 from students;
?column?
-----
(0 rows)

student=# select * from students;
 name | rank | age 
-----+-----+-----
(0 rows)

student=#
```

6. Virtual Environment Setup



```
C:\Windows\System32\cmd.e  X + v

E:\TuteDude\Python\Assignment 7>ls
Task1.py  env

E:\TuteDude\Python\Assignment 7>cd env

E:\TuteDude\Python\Assignment 7\env>cd scripts

E:\TuteDude\Python\Assignment 7\env\Scripts>activate

(env) E:\TuteDude\Python\Assignment 7\env\Scripts>deactivate
E:\TuteDude\Python\Assignment 7\env\Scripts>activate

(env) E:\TuteDude\Python\Assignment 7\env\Scripts>
(env) E:\TuteDude\Python\Assignment 7\env\Scripts>
(env) E:\TuteDude\Python\Assignment 7\env\Scripts>
(env) E:\TuteDude\Python\Assignment 7\env\Scripts>
(env) E:\TuteDude\Python\Assignment 7\env\Scripts>deactivate
E:\TuteDude\Python\Assignment 7\env\Scripts>
E:\TuteDude\Python\Assignment 7\env\Scripts>
E:\TuteDude\Python\Assignment 7\env\Scripts>
E:\TuteDude\Python\Assignment 7\env\Scripts>
```

The screenshot shows a Windows Command Prompt window with the following content:

- Tab: C:\Windows\System32\cmd.e
- Command: `E:\TuteDude\Python\Assignment 7>ls`
Output: `Task1.py env`
- Command: `E:\TuteDude\Python\Assignment 7>cd env`
- Command: `E:\TuteDude\Python\Assignment 7\env>cd scripts`
- Command: `E:\TuteDude\Python\Assignment 7\env\Scripts>activate`
Output: `(env) E:\TuteDude\Python\Assignment 7\env\Scripts>`
- Command: `(env) E:\TuteDude\Python\Assignment 7\env\Scripts>deactivate`
Output: `E:\TuteDude\Python\Assignment 7\env\Scripts>`
- Command: `E:\TuteDude\Python\Assignment 7\env\Scripts>activate`
Output: `(env) E:\TuteDude\Python\Assignment 7\env\Scripts>`
- Subsequent commands and outputs show the user alternating between the active virtual environment and the system prompt.

The taskbar at the bottom shows the date and time: Mon, Nov 24 2025 | 10:05:25 PM.

7. Installing psycopg2

```
C:\Windows\System32\cmd.e  X  +  v

E:\TuteDude\Python\Assignment 7>ls
Task1.py  env

E:\TuteDude\Python\Assignment 7>cd env

E:\TuteDude\Python\Assignment 7\env>cd scripts


E:\TuteDude\Python\Assignment 7\env\Scripts>activate

(env) E:\TuteDude\Python\Assignment 7\env\Scripts>deactivate
E:\TuteDude\Python\Assignment 7\env\Scripts>activate

(env) E:\TuteDude\Python\Assignment 7\env\Scripts>
(env) E:\TuteDude\Python\Assignment 7\env\Scripts>
(env) E:\TuteDude\Python\Assignment 7\env\Scripts>
(env) E:\TuteDude\Python\Assignment 7\env\Scripts>
(env) E:\TuteDude\Python\Assignment 7\env\Scripts>deactivate
E:\TuteDude\Python\Assignment 7\env\Scripts>
E:\TuteDude\Python\Assignment 7\env\Scripts>
E:\TuteDude\Python\Assignment 7\env\Scripts>
E:\TuteDude\Python\Assignment 7\env\Scripts>pip show psycopg2
Name: psycopg2
Version: 2.9.11
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: E:\TuteDude\Python\Assignment 7\env\Lib\site-packages
Requires:
Required-by:

E:\TuteDude\Python\Assignment 7\env\Scripts>pip show psycopg2-binary
WARNING: Package(s) not found: psycopg2-binary

E:\TuteDude\Python\Assignment 7\env\Scripts>
```

The image shows the bottom portion of a Windows desktop. The taskbar is visible with various application icons including the Start button, File Explorer, Google Chrome, and several other utility programs. The system tray on the right shows the date and time as 'Mon, Nov 24 2025 | 10:07:11 PM'. The desktop background features a scenic landscape with a body of water and trees.

8. Connecting to database

The screenshot displays the Visual Studio Code interface. The Explorer sidebar on the left shows a project structure with folders for Assignments 1 through 7, and files like Task1.py, Test.py, .gitignore, README.md, and Assignment 6.zip. The main editor area shows the content of 'Test.py' in the 'Assignment 7' folder, which contains the following Python code:

```
1 import psycopg2
2 conn = psycopg2.connect(dbname="postgres",user="postgres",password="admin",host="localhost",port="5432")
3
4 print('Connected Successfully')
```

Overlaid on the editor is a terminal window titled 'C:\Windows\System32\cmd.exe'. It shows the following commands and their outputs:

```
E:\TuteDude\Python\Assignment 7\env\Scripts>
E:\TuteDude\Python\Assignment 7\env\Scripts>
E:\TuteDude\Python\Assignment 7\env\Scripts>
E:\TuteDude\Python\Assignment 7\env\Scripts>pip show psycopg2
Name: psycopg2
Version: 2.9.11
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: E:\TuteDude\Python\Assignment 7\env\Lib\site-packages
Requires:
Required-by:

E:\TuteDude\Python\Assignment 7\env\Scripts>pip show psycopg2-binary
WARNING: Package(s) not found: psycopg2-binary

E:\TuteDude\Python\Assignment 7\env\Scripts>python Test.py
python.exe: can't open file 'E:\\TuteDude\\Python\\Assignment 7\\env\\Scripts\\Test.py': [Errno 2] No such file or directory

E:\TuteDude\Python\Assignment 7\env\Scripts>cd..

E:\TuteDude\Python\Assignment 7\env>cd..

E:\TuteDude\Python\Assignment 7>python Test.py
Connected Successfully

E:\TuteDude\Python\Assignment 7>
```

The bottom status bar of VS Code indicates the current file is 'main', the encoding is 'UTF-8', and the language is 'Python'.

9. Creating table using Python

The screenshot displays a Visual Studio Code (VS Code) environment with a Python file named `Test.py` open. The code uses the `psycopg2` library to connect to a PostgreSQL database and create a table named `employees`.

```
1 import psycopg2
2 conn = psycopg2.connect(dbname="postgres",user="postgres",password="admin",host="localhost",port="5432")
3
4 cursor = conn.cursor()
5 cursor.execute('create table employees(Name Text, ID Int, Age Int);')
6 print('Table Created Successfully')
7
8 conn.commit()
9 conn.close
```

Below the code editor, a terminal window shows the execution of the Python script. The output indicates a successful connection and table creation.

```
E:\TuteDude\Python\Assignment 7>python Test.py
Connected Successfully

E:\TuteDude\Python\Assignment 7>python Test.py
Table Created Successfully

E:\TuteDude\Python\Assignment 7>
```

Overlaid on the terminal is a `SQL Shell (psql)` window. It shows the connection details and the result of the `\d` command, which lists the newly created `employees` table.

```
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=# \d
Did not find any relations.
postgres=# \d
          List of relations
 Schema | Name      | Type  | Owner
-----+-----+-----+-----
 public | employees | table | postgres
(1 row)

postgres=#
```

The bottom status bar of VS Code shows the current file is `main*` at line 8, column 14, with 4 spaces, UTF-8 encoding, CRLF line endings, Python 3.13.8, and a file icon.

10. Inserting the data using Python

The screenshot displays a Visual Studio Code editor with a Python file named `Test.py` open. The code defines two functions: `table()` and `data()`. `table()` connects to a PostgreSQL database, creates a table named `employees`, and commits the changes. `data()` connects to the same database and inserts a new record into the `employees` table. The code is as follows:

```
1 import psycopg2
2
3 def table():
4     conn = psycopg2.connect(dbname="postgres")
5
6     cursor = conn.cursor()
7     cursor.execute('create table employees (id int, name text, age int)')
8     print('Table Created Successfully')
9
10    conn.commit()
11    conn.close()
12
13 def data():
14     conn = psycopg2.connect(dbname="postgres")
15
16     cursor = conn.cursor()
17     cursor.execute('insert into employees (id, name, age) values (1, "Sam", 26)')
18     print('Data Added Successfully')
19
20    conn.commit()
21    conn.close()
22
23 data()
```

Below the code editor, a terminal window shows the execution of the Python script. The output is:

```
PS E:\TuteDude\Python> & C:/Users/User/AppData/Local/Programs/Python/Python38-6/python.exe Test.py
PS E:\TuteDude\Python> cd .\Assignment 7\
PS E:\TuteDude\Python\Assignment 7> cmd
Microsoft Windows [Version 10.0.26200.7171]
(c) Microsoft Corporation. All rights reserved.
```

Simultaneously, a SQL Shell (psql) window is open, showing the connection details and the results of the database operations. The output is:

```
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=# \d
Did not find any relations.
postgres=# \d
               List of relations
Schema | Name      | Type  | Owner
-----+-----+-----+-----
public | employees | table | postgres
(1 row)

postgres=# select * from employees;
 name | id | age
-----+---+----
 Sam  |  1 |  26
(1 row)

postgres=#
```

The bottom of the image shows the Windows taskbar with the system clock indicating Monday, November 24, 2025, at 10:35:45 PM.

11. Extracting the data from database

The screenshot displays the Visual Studio Code interface with a Python project. The Explorer sidebar on the left shows a file tree for 'PYTHON' with folders 'Assignment 1' through 'Assignment 7'. Under 'Assignment 7', there are files 'env', 'Lib', 'Scripts', '.gitignore', 'CACHEDIR.TAG', 'pyvenv.cfg', 'Task1.py', 'Test.py', and 'Assignment 6.zip'. The 'Test.py' file is selected and open in the editor. The code in 'Test.py' defines two functions: 'data()' and 'extract()'. The 'data()' function inserts a record into the 'employees' table. The 'extract()' function connects to a PostgreSQL database, executes a 'select * from employees;' query, and prints the result. The terminal at the bottom shows the execution of 'python Test.py' three times, resulting in the output '('Sam', 1, 26)', 'Sam', and '1' respectively. The status bar at the bottom indicates the current position is Ln 29, Col 17, with 4 spaces, UTF-8 encoding, CRLF line endings, Python 3.13.8, and 0 errors/warnings.

```
def data():
    cursor.execute('insert into employees(Name, ID, Age) values('Sam', 1, 26);')
    print("Data Added Successfully")

    conn.commit()
    conn.close()

def extract():
    conn = psycopg2.connect(dbname="postgres",user="postgres",password="admin",host="localhost",port="5432")

    cursor = conn.cursor()
    cursor.execute('select * from employees;')
    show = cursor.fetchone()
    print(show[2])

    conn.commit()
    conn.close()

extract()
```

E:\TuteDude\Python\Assignment 7>python Test.py
('Sam', 1, 26)

E:\TuteDude\Python\Assignment 7>python Test.py
Sam

E:\TuteDude\Python\Assignment 7>python Test.py
1

E:\TuteDude\Python\Assignment 7>python Test.py
26

E:\TuteDude\Python\Assignment 7>

12. Adding the input from user

The screenshot shows a Visual Studio Code editor with a Python file named `Test.py` and a terminal window. The terminal shows the execution of `python Test.py` at `E:\TuteDude\Python\Assignment 7>`. The script prompts for employee details: name, ID, and age. The user enters 'John', '02', and '35', and the script outputs 'Data Added Successfully'.

Below the terminal, there is a section with the following commands and outputs:

```
E:\TuteDude\Python\Assignment 7>python Test.py ('Sam', 1, 26)
```

```
E:\TuteDude\Python\Assignment 7>python Test.py Sam
```

```
E:\TuteDude\Python\Assignment 7>python Test.py 1
```

```
E:\TuteDude\Python\Assignment 7>python Test.py 26
```

```
E:\TuteDude\Python\Assignment 7>
```

On the right, a PostgreSQL shell window (psql) is open, showing the following output:

```
8-bit characters might not work correctly. See psql reference
page "Notes for Windows users" for details.
Type "help" for help.

postgres=# \d
Did not find any relations.
postgres=# \d
List of relations
Schema | Name      | Type  | Owner
-----+-----+-----+-----
public | employees | table | postgres
(1 row)

postgres=# select * from employees;
 name | id | age
-----+---+---
 Sam  | 1  | 26
(1 row)

postgres=# select * from employees;
 name | id | age
-----+---+---
 Sam  | 1  | 26
 John | 2  | 35
(2 rows)

postgres=#
```

The bottom status bar shows the file is at `Ln 19, Col 48`, with `Spaces: 4`, `UTF-8` encoding, `CRLF` line endings, and `Python 3.13.8` interpreter.