**How to Integrate PostgreSQL to Python**

We write a python program by using the **Psycopg2 module** to connect to the PostgreSQL database and then perform some database or SQL transactions such as creating a table and then insert data to the table, updating a table data and delete records from the table using python code. We shall also see how to fetch table data from PostgreSQL database to python program.

**To Install Psycopg2 module:**

For Windows,

***pip install psycopg2***

For Linux/Mac,

***pip3 install psycopg2***

Make sure to use an up-to-date version of **pip** (you can upgrade it using something like pip install

-U pip).

Now Once you installed the module the next thing is to identify all the different credentials that you need to connect with your Database i.e..., Hostname,Database,Username,Password,Port-Id you can get most of the information from PG-Admin. Just go to your PG\_Admin Right Click on the Server name go to properties and go to Connections

Graphical user interface, text, application

Description automatically generatedA picture containing graphical user interface

Description automatically generated

* We Use Connection and Cursor Functions.
* We Place Our Database Connection Command inside Try and Except Block incase if the Data base connection Fails it does not Effect our Program it throw an exception.

Exception block capturing the Error and Print the Error.

* Cursor helps you to perform SQL operations.
* To Create a Cursor We Use a variable **cur = conn.cursor()**
* At the End of the Program we need to End close the Cursor **cur.close()**

**Same as conn.close()**

* Initially, we assign none for both conn and cur variables
* We Use Finally block to close connection and cursor if and only if the value of conn and cur is not none .
* If your Database connection never succeeded then conn variable would never have the connection object and never have close method it throws an error. In order to avoid we use none value for both conn and cur variable in the beginning.
* In Order to commit the transaction we use conn.commit( ) . it save the transactions into the database

Syntax will be same as SQL command. We just need to place it in python program.

To execute the commands we use cur.execute( ) function.

We can fetch the data from database by using cur.fetchall( )

**Python to PostgreSQL**

import psycopg2

hostname = 'localhost'

database = 'demo'

username = 'postgres'

pwd = 'Sadhan@1'

port\_id = 5432

conn = None

cur = None

try:

    conn = psycopg2.connect(

                host = hostname,

                dbname = database,

                user = username,

                password = pwd,

                port = port\_id)

    cur = conn.cursor()

    cur.execute('DROP TABLE IF EXISTS employee')

create\_script = '''CREATE TABLE employee(id int ,

         name varchar(40),

salary int,

dept\_id varchar(30)) '''

    cur.execute(create\_script)

    insert\_script = '''INSERT INTO employee  VALUES

    (1,'Dinesh',12000,'D1'),

    (2,'Mahesh',15000,'D1'),

    (3,'SAtya',20000,'D2')'''

    cur.execute(insert\_script)

    cur.execute('select \* from employee')

    print(cur.fetchall())

    conn.commit()

except Exception as error:

    print(error)

finally:

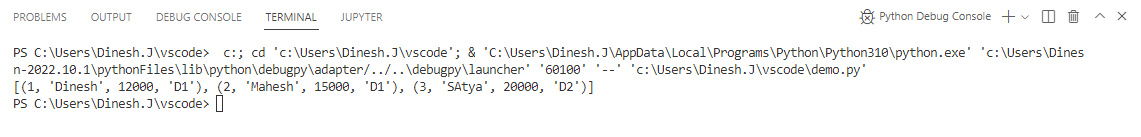
    if cur is not None:

        cur.close()

    if conn is not None:

        conn.close()

**Output in VS Code Terminal**:



**In PostgreSQL,**

Table

Description automatically generated with medium confidence

Graphical user interface, application, Word

Description automatically generated