**POSTGRE SQL - DATABASE / CRUD IN PYTHON**

**Download from website**

A screenshot of a computer

Description automatically generated

**Download and Install the Exe (Windows x86-64)**

**Queries:**

create database employeedb;

\c employeedb;

create table employee (id serial primary key,name varchar(30),sal real);

create user test with encrypted password 'password';

grant privileges on database employeedb to test;

grant all privileges on table employee to test;

grant all privileges on sequence employee\_id\_seq to test;

drop database databasename / drop user test

**Execution Steps in Mac System**

A screenshot of a computer

Description automatically generated

A blue sign with black text

Description automatically generated

Create Database / Table / Role and assign it in Postgres (Windows)

A screenshot of a computer

Description automatically generated

**To connect and work with Postgres Databases from Python prgs, we need psycopyg2 package. Ie. Similar to MySQL Connector for MySQL database.**

A screen shot of a computer

Description automatically generated

**Google – psycopg2**

A screenshot of a computer

Description automatically generated

**Go to this page below:**

A blue screen with a white box and a white box with a white box and a white box with a black box with a white box with a white box with a black box with a white box

Description automatically generated

A screenshot of a computer

Description automatically generated

**Use this command : pip install psycopg2 in any other (Atom / Eclipse etc)**

**In PyCharm**

A screenshot of a computer

Description automatically generated

**Go to Python Interpreter, in the above list of tools, we should have psycopg2 interpreter. If not found, select the + symbol and search for psycopg2-binary and select install.**

A screenshot of a computer

Description automatically generated

**Once installed, this should be found in the list (as below)**

A screenshot of a computer

Description automatically generated

**DATABASE CREATED IN POSTGRES**

-- Database: PythonDB

-- DROP DATABASE IF EXISTS "PythonDB";

CREATE DATABASE "PythonDB"

WITH

OWNER = augusta

ENCODING = 'UTF8'

LC\_COLLATE = 'English\_Ireland.1252'

LC\_CTYPE = 'English\_Ireland.1252'

TABLESPACE = pg\_default

CONNECTION LIMIT = -1

IS\_TEMPLATE = False;

GRANT ALL ON DATABASE "PythonDB" TO augusta;

ALTER DEFAULT PRIVILEGES FOR ROLE postgres

GRANT ALL ON TABLES TO augusta;

**ROLE CREATED IN POSTGRES**

-- Role: augusta

-- DROP ROLE IF EXISTS augusta;

CREATE ROLE augusta WITH

LOGIN

NOSUPERUSER

INHERIT

NOCREATEDB

NOCREATEROLE

NOREPLICATION

ENCRYPTED PASSWORD 'SCRAM-SHA-256$4096:YAL/LfHVcV6BeDDjchCALg==$OnMCvsqODEM8v5/WcVJL+nkmlEwK9rw5EbvZUxBMW/Y=:tS6ybgkWeolnIJ5b5TbvKpXb9swU0jA6dYQteXbOR/4=';

**TABLE CREATED IN POSTGRES**

-- Table: public.volvocars

-- DROP TABLE IF EXISTS public.volvocars;

CREATE TABLE IF NOT EXISTS public.volvocars

(

carid integer NOT NULL DEFAULT nextval('volvocars\_carid\_seq'::regclass),

carname character varying(50) COLLATE pg\_catalog."default" NOT NULL,

carprice real NOT NULL,

cartype character varying(50) COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT volvocars\_pkey PRIMARY KEY (carid)

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.volvocars

OWNER to augusta;

GRANT ALL ON TABLE public.volvocars TO augusta;

**CODE TO INSERT RECORD IN POSTGRES USING PYTHON**

import psycopg2  
  
def postgresInsert(cn,cp,ct):  
 try:  
 conn = psycopg2.connect(database="PythonDB",user="postgres",password="Jazlyn@0910",host="localhost",port="5432")  
 print("Connected to Postgres SQL Server !!!!")  
 cursor = conn.cursor()  
 try:  
 insertString = "insert into volvocars(carname, carprice, cartype) values ('%s','%f','%s')"  
 args = (cn,cp,ct)  
  
 cursor.execute(insertString % args)  
 conn.commit()  
 finally:  
 conn.rollback()  
 print("Record Updated")  
 finally:  
 conn.close()  
  
  
cname = input("Please enter the Car Name : ")  
cprice = float(input("Enter the Car Price : "))  
ctype = input("Please enter the Car Type : ")  
postgresInsert(cname,cprice,ctype)

**INPUT IN PYTHON**

Please enter the Car Name : Volvo XC90

Enter the Car Price : 87580.05

Please enter the Car Type : Electric

Connected to Postgres SQL Server !!!!

Record Inserted !!!!!

**OUTPUT IN POSTGRES USING PYTHON**

**A screenshot of a computer

Description automatically generated**