

Database and Python connection - <https://www.sqlalchemy.org/>

Postgress sql server driver- pip install psycopg2 (v2.8.5)

psycopg2.\_\_version\_\_

psycopg – Driver

Python – Database connection

**#Importing Generic API package/library for database connection**

import sqlalchemy

**#Importing postgres driver**

import psycopg2

**#Connection URL to postgres DB to python**

SrinuPostgresDB\_URI = 'postgres+psycopg2://postgres:admin@localhost:5432/postgres'

Coding

from sqlalchemy import create\_engine #Spend Decide API/Package/Library for accessing databases in generel

engine = create\_engine(ShivaPostgresDB\_URI) #Spend Plan API/Package/Library for accessing databases in generel

connection = engine.connect() #Spend the API/money API/Package/Library for accessing databases in generel

connection = connection.execution\_options(isolation\_level="READ COMMITTED") # just READ

from sqlalchemy.sql import text

MyQuery = text("SELECT address\_id FROM \"Customers\" WHERE address\_id = 5 AND customer\_id = 1000") #Enjoy the benfits of API API/Package/Library for accessing databases in generel

a = connection.execute(MyQuery).fetchone()

print()

Github(Source code repository)

1. Pwd– Presenting working directory
2. Create new repository in Github – Internet
3. sync between internet and computer
4. git clone - repository name(url)
5. cd repository name(pythonprograms)
6. Ls – list the files (cd introcution to python) – whenever master appears it means sync is completed
7. Make sure the present working directory is master
8. git add –A
9. git commit –m “sending my python code to internet repository
10. git push origin master

**API’s Used(libraries or packages or helper classes)**

1. sqlalchemy(libraries : create\_engine, Table, Column, Integer, String, MetaData,ForeignKey, Boolean)
2. psycopg2
3. math

**jupyter notebook path :** D:\Pysql\Pythonprograms>jupyter notebook

Python Calsses

**Understanding the operators**  
1. Single by(/) represents integer division and // represents decimal division.

2. Any string or character representation must be quoted (“”)

3. There are three types of operators

- Arithmetic Operator ( +,-,\*,/)  
 - Comparison Operator ( <,>,etc)  
 - Logical Operator(AND, OR, NOT )  
 - Assignment operator(=)

Variable or Data types

1. In coding all the left side entities are called variables
2. Boolean datatypes means TRUE or FALSE
3. Casting means converting original data type to desired data type

Conditional statements

1. If else (elif) is similar to case when in SQL

**x= 10**

**If** (x>0):

Print(“ x is greater than 0”)

Elif(x<0):

Print(“x is less than 0”)

Elif(x!=0):

E=mc2

Print(E)

Elif (x==0):

Print(“ x is equal to 0”)

**Else :**

Print (“x is negative”)

Function – is a reusable functionality across the program

**Python Sample: Load (ObjectRelationalMapping)**

metadata = MetaData()  
Users = Table('Users', metadata,  
        Column('Name', String, primary\_key=True),  
        Column('ID', Integer),  
        Column('Role', String),  
        Column('Dept', String),  
        Column('Company', String),  
        Column('Location', String))  
  
ins = Users.insert().values(Name='rosy', ID=7, Role='S.Manager', Dept='IT', Company='PK', Location='USA')  
  
connection.execute(ins)  
connection.close()

**Accessing from database to python shows like below(rowproxy or list)**

(524,1,'Jared', 'Ely', 'jared.ely@sakilacustomer.org', 530,true, 2006-02-14, '49:45.7', 1)

**Accessing from database to python in a dictionary model**

{customer\_id:524, store\_id:1, first\_name:'Jared', last\_name:'Ely', mail:'jared.ely@sakilacustomer.org',address\_id:530,

activebool:true, create\_date:2006-02-14, last\_update:'49:45.7', active:1}

Important:

1. Print() statements are rough notes and testing only
2. We need to use workers to transfer or load big data – for loops with ranges, list, dictionaries

Dropna – Drop Not available

Numpy – Numerical Python