**PYTHON DATABASE ACCESS:**

The python standard database interface is the python DB-API. (Application Program Interface). Most python database interfaces adhere to this standard.

The python database API supply with database neutral programming interface for different databases.

These are : MYSQL, SQLite, MSSQL, Oracle, PostgreSQL, Informix, Sybase, Interbase etc. The API supports all these database servers.

The API includes:

* 1. Importing the API Module
  2. Acquiring a connection with the database
  3. Issuing SQL commands and stored procedures.
  4. Closing the connection.

Benefits of Python Database Programming:

1. Programming is simple, efficient compare to the other languages.
2. Python database and python programs both are portable.
3. API of python for the databases is compatible with other databases also.

Some common database connectivity to learn:

1. connecting using mysql using python3 default interface
2. connecting using pymysql module in jupiter notebook
3. connect with SQLite
4. connect using cx\_Oracle module in python
5. Connect with SQL server using pyodbc
6. connect with MS-Access using pyodbc

**Python with MYSQL connectivity using default interface:**

To connect with the MYSQL database we must have MYSQL software and then we need to import mysql.connector module in the program.

The mysql.connector is not a built-in module. So, we need to install it using pip installer.

>>>python -m pip install mysql-connector

Or we can download the file:

mysql-connector-python-8.0.13.tar.gz

and then extract the archived file.

Then open the folder of mysql-connector and execute the code:

$python setup.py install

Once installation is over , we can import the file using statement:

>>>import mysql.connector

**Creating the connection:**

Connect(): To create a connection between the MYSQL database and the python application this method is used. This method is available in the mysql.connector module.

Syntax:

Connectionobject=mysql.connector.connect(host=”hostname”, user=”username”,passwd=”password”, database=”databasename”)

Here the host name is local host or ip of the host.

Example:

import mysql.connector

myconn=mysql.connector.connect(host=”localhost”,user=”root”,passwd=””, database=”db1”)

print(myconn)

o/p:

mysql.connector.connection….

#object created

**Creating a Cursor Object:**

The MySQLCursor of mysql-connector-python (and similar libraries) is used to execute statements to communicate with the MySQL database.

**cursor():** This method is used to create a cursor object for the mysqlconnector object.

Syntax/example:

mycursor\_object=myconnection\_object.cursor()

mycursor=myconn.cursor()

Now the cursor object mycursor is created and can be used to execute mysql commands.

**fetchall()**:

The method fetches all rows of a query result set and returns a **list of tuples**. If no more rows are available, it returns an empty list.

You must **fetch all** rows for the current query before executing new statements using the same connection.

**fetchone():**

**fetchall()** fetches all the rows of a query result. An empty list is returned if there is no record to fetch the cursor. **fetchone()** method returns one row or a single record at a time. It will return None if no more rows / records are available.

**Ex1: To see the list of databases: (p1.py)**

import mysql.connector

myconn=mysql.connector.connect(host=”localhost”,user=”root”,passwd=””)

mycursor=myconn.cursor()

mycursor.execute(“show databases”)

dbs=mycursor.fetchall()

for x in dbs:

print(x)

myconn.close()

o/p:

it will show all the list of databases in MYSQL databases.

**Ex1.1: to show tables from a database(p1.1.py)**

**shows the tables list from database db1**

#TO SHOW TABLES FROM DATABASE db1

import mysql.connector

myconn=mysql.connector.connect(host="localhost",user="root",passwd="")

mycursor=myconn.cursor()

mycursor.execute("show databases")

for i in mycursor.fetchall():

print(i)

x=input("enter a database name")

mycursor.execute("use "+x)

mycursor.execute("show tables")

dbs=mycursor.fetchall()

print("tables are...............")

for x in dbs:

print(x)

y= input("enter a table name")

mycursor.execute("select \* from "+y)

dbs=mycursor.fetchall()

print("content is...............")

for x in dbs:

print(x)

myconn.close()

o/p:

**Ex2: to show the content of a table employee from database db1 (p2.py)**

Solution:

import mysql.connector

myconn=mysql.connector.connect(host=”localhost”, user=”root”, passwd=””, database=”db1”)

mycursor=myconn.cursor()

mycursor.execute(“select \* from employee”)

rows=mycursor.fetchall()

for x in rows:

print(x)

o/p:

(1, 'sree', 'lakshmi', 'nbihar', 'null')

(2, 'ree', 'hmi', 'nihar', 'null')

(3, 'reena', 'hunmi', 'vinihar', 'null')

(4, 'reema', 'hnmi', 'vainihar', 'london')

(5, 'rema', 'ahnmi', 'nihar', 'london')

**EX2.1:**

**#DISPLAYING THE CONTENT OF EMPLOYEE TABLE with formatting(p2.1.py)**

import mysql.connector

con = mysql.connector.connect( host="localhost",user="root",password="",database="db1")

mycursor=con.cursor()

mycursor.execute("desc employee")

mycursor.fetchall()

mycursor.execute("select \* from employee")

desc=mycursor.description

print('{:<7s} {:<7s} {:<7s} {:<7s}'. format(desc[0][0],desc[1][0],desc[2][0],desc[3][0]))

print('{:<7s} {:<7s} {:<7s} {:<7s}'.format('-'\*7, '-'\*7,'-'\*7, '-'\*7))

result= mycursor.fetchall()

for row in result:

print("%d\t%s\t%s\t%s"%(row[0],row[1],row[2],row[3]))

O/P:

empid fname lname address

------- ------- ------- -------

1 sree lakshmi nbihar

2 ree hmi nihar

3 reena hunmi vinihar

4 reema hnmi vainihar

5 rema ahnmi nihar

**EX2.2:**

**#DISPLAYING THE CONTENT OF EMPLOYEE TABLE with formatting(p2.2.py)**

#reading the content of table:employee with formatting from database: db1

import mysql.connector

myconn=mysql.connector.connect(host="localhost",user="root",passwd="",database="db1")

mycursor=myconn.cursor()

mycursor.execute("select empid,fname,lname from employee")

result=mycursor.fetchall()

print("employee details: \nid\t\tfname\t\tlname\n.......................................")

for row in result:

print("%d\t\t%s\t\t%s"%(row[0],row[1],row[2]))

myconn.close()

o/p:

employee details:

id fname lname

.......................................

1 sree lakshmi

2 ree hmi

3 reena hunmi

4 reema hnmi

5 rema ahnmi

**ex3: create a database using python mysql connector.(p3.py)**

Solution:

import mysql.connector

myconn=mysql.connector.connect(host="localhost", user="root", passwd="")

mycursor=myconn.cursor()

x=input("enter name for new database to create:")

mycursor.execute("create database "+x)

print("database created: "+x)

myconn.close()

o/p:

enter name for new database to create:hello

database created: hello

**ex4: to connect multiple databases using mysql connector.(p4.py)**

Solution:

import mysql.connector

myconn1=mysql.connector.connect(host=”localhost”, user=”root”, passwd=””, database=”db1”)

myconn2=mysql.connector.connect(host=”localhost”, user=”root”, passwd=””, database=”college”)

mycursor1=myconn1.cursor()

mycursor2=myconn2.cursor()

mycursor1.execute(“select \* from employee”)

rows1=mycursor1.fetchall()

for x in rows1:

print(x)

print(“end of employee”)

mycursor2.execute(“select \* from animals”)

rows2=mycursor2.fetchall()

for y in rows2:

print(y)

print(“end of animals”)

myconn1.close()

myconn2.close()

o/p:

(1, 'sree', 'lakshmi', 'nbihar', 'null')

(2, 'ree', 'hmi', 'nihar', 'null')

(3, 'reena', 'hunmi', 'vinihar', 'null')

(4, 'reema', 'hnmi', 'vainihar', 'london')

(5, 'rema', 'ahnmi', 'nihar', 'london')

end of employee

(1, 'dog')

(2, 'cat')

(3, 'penguin')

(4, 'lax')

(5, 'whale')

(6, 'ostrich')

end of animals

**ex5: creating a table P5.py)**

solution:

#CREATING A TABLE

import mysql.connector

#creating the connection object

myconn=mysql.connector.connect(host="localhost", user="root", passwd="", database="pythondb1")

#creating the cursor object

mycursor=myconn.cursor()

try:

#creating a table with name employee having four coulmns i.e: name, id, salary

x=input("enter name of employee table")

Dbs=mycursor.execute("create table "+x+ "(name varchar(20) not null, id int(20) not null primary key, salary float not null)")

mycursor.execute("show tables")

tab=mycursor.fetchall()

for i in tab:

print(i)

print("table created: ",x)

print("table structure is:")

mycursor.execute("desc "+x)

des=mycursor.fetchall()

for i in des:

print(i)

mycursor.execute("drop table " +x)

print("table dropped: ",x)

except:

myconn.rollback()

myconn.close()

o/p:

enter name of employee tableg

('a',)

('b',)

('c',)

('d',)

('emp',)

('emp1',)

('emp2',)

('emp5',)

('emp6',)

('emplo',)

('g',)

('tab1',)

table created: g

table structure is:

('name', 'varchar(20)', 'NO', '', None, '')

('id', 'int(20)', 'NO', 'PRI', None, '')

('salary', 'float', 'NO', '', None, '')

table dropped: g

**ex5.1 dropping a table (p5.1.py)**

#DROPPING A TABLE

import mysql.connector

#creating the connection object

myconn=mysql.connector.connect(host="localhost", user="root", passwd="", database="pythondb1")

#creating the cursor object

mycursor=myconn.cursor()

try:

#ENTER TABLE name to drop

mycursor.execute("show tables")

s=mycursor.fetchall()

for i in s:

print(i)

x=input("enter name of table")

mycursor.execute("drop table " +x)

print("table dropped: ",x)

except:

myconn.rollback()

myconn.close()

o/p:

('b',)

('c',)

('d',)

('emp',)

('emp1',)

('emp2',)

('emp5',)

('emp6',)

('emplo',)

('tab1',)

enter name of tableb

table dropped: b

**Ex6: alter table(p6.py)**

#it is to alter a table emp in pythondb1

import mysql.connector

myconn=mysql.connector.connect(host="localhost",user="root",passwd="", database="pythondb1")

mycursor=myconn.cursor();

try:

mycursor.execute("show tables")

s=mycursor.fetchall()

for i in s:

print(i)

x=input("enter name of table")

mycursor.execute("alter table "+x+" add branch varchar(20) not null")

print("\n\ntable structure adding branch field is:")

mycursor.execute("desc "+x)

des=mycursor.fetchall()

for i in des:

print(i)

mycursor.execute("alter table "+x+" drop branch")

print("\n\ntable structure is after removing branch field:")

mycursor.execute("desc "+x)

des=mycursor.fetchall()

for i in des:

print(i)

except:

mycursor.rollback()

myconn.close()

o/p:

('c',)

('d',)

('emp',)

('emp1',)

('emp2',)

('emp5',)

('emp6',)

('emplo',)

('tab1',)

enter name of tablec

table structure adding branch field is:

('name', 'varchar(20)', 'NO', '', None, '')

('id', 'int(20)', 'NO', 'PRI', None, '')

('salary', 'float', 'NO', '', None, '')

('branch', 'varchar(20)', 'NO', '', None, '')

table structure is after removing branch field:

('name', 'varchar(20)', 'NO', '', None, '')

('id', 'int(20)', 'NO', 'PRI', None, '')

('salary', 'float', 'NO', '', None, '')

**Ex7:inserting data into a table (p7.py)**

#it is to insert into a table emp1 in pythondb1

import mysql.connector

myconn=mysql.connector.connect(host="localhost",user="root",passwd="", database="pythondb1")

mycursor=myconn.cursor();

q1="insert into emp1(name, id, salary, deptid, branchname) values(%s,%s,%s,%s,%s)"

nm=input("enter name")

iden=int(input("enter id"))

sal=float(input("enter salary"))

did=int(input("enter dept id"))

bname=input("enter branch name")

v1=(nm,iden,sal,did,bname)

try:

mycursor.execute(q1,v1)

myconn.commit()

except:

myconn.rollback()

print(mycursor.rowcount, "record inserted")

mycursor.execute("select \* from emp1")

tbs=mycursor.fetchall()

print("content in employee table is........")

for x in tbs:

print(x)

myconn.close()

o/p:

enter namea

enter id999

enter salary909090

enter dept id11

enter branch namecs

1 record inserted

content in employee table is........

('john', 110, 25000.0, 201, 'cse', '')

('sam', 111, 55000.0, 201, 'cse', '')

('a', 999, 909090.0, 11, 'cs', '')

**Ex8: reading the content of table:employee from database: db1 (p8.py)**

import mysql.connector

myconn=mysql.connector.connect(host="localhost",user="root",passwd="",database="db1")

mycursor=myconn.cursor()

try:

mycursor.execute("select \* from employee")

result=mycursor.fetchall()

print("employee details........................................")

for x in result:

print(x)

except:

myconn.rollback()

mycursor.execute("select empid,fname from employee")

result=mycursor.fetchall()

print("employee :......name.........id.......................")

for x in result:

print(x)

myconn.close()

o/p:

employee details........................................

(1, 'sree', 'lakshmi', 'nbihar', 'null')

(2, 'ree', 'hmi', 'nihar', 'null')

(3, 'reena', 'hunmi', 'vinihar', 'null')

(4, 'reema', 'hnmi', 'vainihar', 'london')

(5, 'rema', 'ahnmi', 'nihar', 'london')

employee :......name.........id.......................

(1, 'sree')

(2, 'ree')

(3, 'reena')

(4, 'reema')

(5, 'rema')

**Ex9: updating a content of a table**

#updating the content of table:employee from database: db1

import mysql.connector

myconn=mysql.connector.connect(host="localhost",user="root",passwd="",database="db1")

mycursor=myconn.cursor()

mycursor.execute("select \* from employee")

t=mycursor.fetchall()

for i in t:

print(i)

x=input("enter 1st firstname")

y=input("enter city for the name")

#mycursor.execute("update employee set city=\"gunupur\" where fname=\"sam1\" ")

mycursor.execute("update employee set city=\""+y+"\" where fname=\""+x+"\"")

#mycursor.reset()

mycursor.execute("select \* from employee")

result=mycursor.fetchall()

print("employee details: \nempid\tfname\tlname\taddress\tcity\n......................................")

for row in result:

print("%d\t%s\t%s\t%s\t%s"%(row[0],row[1],row[2],row[3],row[4]))

myconn.close()

o/p:

(1, 'sree', 'lakshmi', 'nbihar', 'null')

(2, 'ree', 'hmi', 'nihar', 'null')

(3, 'reena', 'hunmi', 'vinihar', 'null')

(4, 'reema', 'hnmi', 'vainihar', 'london')

(5, 'rema', 'ahnmi', 'nihar', 'london')

enter 1st firstnamesree

enter city for the namevizag

employee details:

empid fname lname address city

....... .......... ......... . ...... .....

1 sree lakshmi nbihar vizag

2 ree hmi nihar null

3 reena hunmi vinihar null

4 reema hnmi vainihar london

5 rema ahnmi nihar london

**Python with pymysql module using Jupiter notebook:**

Ex 10#connecting using pymysql module in jupiter notebook(p10.py)

import pymysql

from pymysql import Error

con = pymysql.connect(host="localhost",user="root",password="", database="db1")

mycursor=con.cursor()

mycursor.execute("select \* from employee")

result=mycursor.fetchall()

for row in result:

print(row)

**Python with pyodbc module using MSACCESS:**

Ex11: #it is used to connect with MSACCESS

import sys

import pyodbc

#conn = pyodbc.connect(r'Driver={Microsoft Access Driver(\*.mdb,\*.accdb)}; DBQ=D:\DATABASE\db1.emp;')

conn=pyodbc.connect(dsn='test1') #to move control panel,administrative tools, data sources,systemDSN,add "test1"

mycursor = conn.cursor()

query="select \* from emp"

mycursor.execute(query)

result=mycursor.fetchall()

for row in result:

print(row)

**Python with sqlite3 module using SQLite3:**

Ex12:

#connect with SQLite

import sqlite3

conn = sqlite3.connect('db1')

mycursor=conn.cursor()

mycursor.execute("select \* from employee")

result=mycursor.fetchall()

for row in result:

print(row)

conn.close()

**Python with pyodbc module using SQLServer:**

Ex13:

#it is used to connect with SQL server

import sys

import pyodbc

conn = pyodbc.connect('Driver={SQL Server};'

'Server=DESKTOP-NTMMPFV;'

'Database=db1;'

'Trusted\_Connection=yes;')

mycursor = conn.cursor()

query="select \* from Student where Branch\_Id like 'CSE' and vcStatus='ACTIVE'"

mycursor.execute(query)

desc= mycursor.description

print('{:<10s} {:<20s} {:<20s} {:<6s} {:<20s}'.format(desc[0][0],desc[1][0], desc[2][0], desc[3][0],desc[4][0]))

print('{:<10s} {:<20s} {:<20s} {:<6s} {:<20s}'.format('-'\*20, '-'\*20, '-'\*20,'-'\*10, '-'\*6)) # Print divider

result=mycursor.fetchall()

for row in result:

print(row)

**Python with odbc module using Oracle:**

Ex14: connecting to oracle database

import cx\_Oracle

my\_dsn = cx\_Oracle.makedsn("localhost",port='8080',sid="apex")

myconn=cx\_Oracle.connect(user='system', password='manager', dsn='my\_dsn')

mycursor=myconn.cursor()

mycursor.execute(“show databases”)

dbs=mycursor.fetchall()

for x in dbs:

print(x)

myconn.close()