#connect-psycopg2-python

import psycopg2

conn = psycopg2.connect(database="magd",

                        host="localhost",

                        user="postgres",

                        password="1234",

                        port="5432")

cursor = conn.cursor()

cursor.execute("CREATE TABLE emp (name VARCHAR(255), address VARCHAR(255))")

sql = "INSERT INTO emp (name, address) VALUES (%s, %s)"

val = ("John", "Highway 21")

cursor.execute(sql, val)

conn.commit()

print(cursor.rowcount, "record inserted.")

#sql = "DELETE FROM emp WHERE address = 'Highway 21'"

#cursor.execute(sql)

#conn.commit()

#print(cursor.rowcount, "record(s) deleted")

#sql = "DROP TABLE emp"

#cursor.execute(sql)

sql = "UPDATE emp SET address = 'Canyon 123' WHERE address = 'Highway 21'"

cursor.execute(sql)

conn.commit()

print(cursor.rowcount, "record(s) affected")

mycursor = conn.cursor()

mycursor.execute("SELECT \* FROM emp")

myresult = mycursor.fetchall()

for x in myresult:

  print(x)

sql = "SELECT \* FROM emp WHERE address ='Highway 21'"

mycursor.execute(sql)

myresult = mycursor.fetchall()

for x in myresult:

  print(x)

#connect- pymongo-python

import pymongo

myclient = pymongo.MongoClient("mongodb://localhost:27017/")

mydb = myclient["momen"]

mycol = mydb["iter"]

mydict = { "name": "John", "address": "Highway 37" }

x = mycol.insert\_one(mydict)

mydict = { "name": "Peter", "address": "Lowstreet 27" }

x = mycol.insert\_one(mydict)

print(x.inserted\_id)

for x in mycol.find():

  print(x)

  myquery = { "address": "Park Lane 38" }

mydoc = mycol.find(myquery)

for x in mydoc:

  print(x)

mydoc = mycol.find().sort("name")

for x in mydoc:

  print(x)

#myquery = { "address": "Mountain 21" }

# mycol.delete\_one(myquery)

"""

mycol.drop()

"""

myquery = { "address": "Valley 345" }

newvalues = { "$set": { "address": "Canyon 123" } }

mycol.update\_one(myquery, newvalues)

#print "customers" after the update:

for x in mycol.find():

  print(x)

  myresult = mycol.find().limit(5)

#print the result:

for x in myresult:

  print(x)

#connect-mysql-python

import mysql.connector

mydb = mysql.connector.connect(

  host="localhost",

  user="root",

  password="1234",

  database="momen"

)

"""

mycursor.execute("CREATE DATABASE momen")

mycursor.execute("CREATE TABLE customers (name VARCHAR(255), address VARCHAR(255))")

sql = "INSERT INTO customers (name, address) VALUES (%s, %s)"

val = ("John", "Highway 21")

mycursor.execute(sql, val)

mydb.commit()

print(mycursor.rowcount, "record inserted.")

sql = "DELETE FROM customers WHERE address = 'Highway 21'"

mycursor.execute(sql)

mydb.commit()

print(mycursor.rowcount, "record(s) deleted")

sql = "DROP TABLE customers"

mycursor.execute(sql)

sql = "UPDATE customers SET address = 'Canyon 123' WHERE address = 'Highway 21'"

mycursor.execute(sql)

mydb.commit()

print(mycursor.rowcount, "record(s) affected")

"""

mycursor = mydb.cursor()

mycursor.execute("SELECT \* FROM customers")

myresult = mycursor.fetchall()

for x in myresult:

  print(x)

sql = "SELECT \* FROM customers WHERE address ='Highway 21'"

mycursor.execute(sql)

myresult = mycursor.fetchall()

for x in myresult:

  print(x)

#connect-oracle-python

import cx\_Oracle

import db\_config

con = cx\_Oracle.connect(db\_config.user, db\_config.pw, db\_config.dsn)

cur = con.cursor()

# Create table

cur.execute("""begin

                 execute immediate 'drop table testgeometry';

                 exception when others then

                   if sqlcode <> -942 then

                     raise;

                   end if;

               end;""")

cur.execute("""create table testgeometry (

               id number(9) not null,

               geometry MDSYS.SDO\_GEOMETRY not null)""")

# Create and populate Oracle objects

typeObj = con.gettype("MDSYS.SDO\_GEOMETRY")

elementInfoTypeObj = con.gettype("MDSYS.SDO\_ELEM\_INFO\_ARRAY")

ordinateTypeObj = con.gettype("MDSYS.SDO\_ORDINATE\_ARRAY")

obj = typeObj.newobject()

obj.SDO\_GTYPE = 2003

obj.SDO\_ELEM\_INFO = elementInfoTypeObj.newobject()

obj.SDO\_ELEM\_INFO.extend([1, 1003, 3])

obj.SDO\_ORDINATES = ordinateTypeObj.newobject()

obj.SDO\_ORDINATES.extend([1, 1, 5, 7])

print("Created object", obj)

# Add a new row

print("Adding row to table...")

cur.execute("insert into testgeometry values (1, :objbv)", objbv = obj)

print("Row added!")

# Query the row

print("Querying row just inserted...")

cur.execute("select id, geometry from testgeometry");

for row in cur:

    print(row)