**Database in Python**

Just like we did in Java, we can handle DB related work & run SQL queries in python.

We need some way that connect our code & Oracle. That’s when cx\_Oracle comes in picture.

To **install** cx\_Oracle, run this pip command in cmd.

“pip install cx\_Oracle”

**#Connection Establishment and verification**

(Username & password = ce1 for our lab.)

import cx\_Oracle # inheritance.

con = cx\_Oracle.connect('username/password@192.168.29.152/xe')

#here 192.168.29.152 is the IP of the database computer with which we want to connect.

print (con.version)

for info in con.version.split('.'):

print(info)

con.close()

**#DDL/DML**

import cx\_Oracle

con = cx\_Oracle.connect('username/password@192.168.29.152/xe')

cur = con.cursor()

cur.execute('select sysdate from dual')

for result in cur:

print (result)

cur.close()

con.close()

**#Create table**

import cx\_Oracle

con = cx\_Oracle.connect('username/password@192.168.29.152/xe')

cur = con.cursor()

cur.execute("""CREATE TABLE "ADVERTISERS"

( "ADVERTISER\_ID" NUMBER(10,0),

"ADVERTISER\_NAME" VARCHAR2(20 BYTE),

"CONTACT\_LNAME" VARCHAR2(20 BYTE),

"CONTACT\_FNAME" VARCHAR2(20 BYTE),

"PHONE\_NUMBER" VARCHAR2(20 BYTE),

"CITY" VARCHAR2(10 BYTE),

"ST" CHAR(2 BYTE),

"COUNTRY" CHAR(20 BYTE),

"REGISTRY\_DATE" DATE

)""")

cur.execute("""

Insert into ADVERTISERS (ADVERTISER\_ID,ADVERTISER\_NAME,CONTACT\_LNAME,CONTACT\_FNAME,PHONE\_NUMBER,CITY,ST,COUNTRY) values (1,'ROLEY COMPANIES','SMITH','ERIC','713-882-0456','HOUSTON','TX','USA')""")

con.commit()

con.close()

**#Fetching Data**

3 methods

method fetchone()

method fetchmany(numRow=3)

method fetchall()

In the query.py, after execute command, you may run fetchone, fetchmany or fetchall as per requirement. Practice using below code:

res = cur.fetchall()

for r in res:

print r

**#Update**

import cx\_Oracle

con = cx\_Oracle.connect('username/password@192.168.29.152/xe')

cur = con.cursor()

cur.execute("""update ADVERTISERS set REGISTRY\_DATE=sysdate where ADVERTISER\_ID=1""")

con.commit()

con.close()