FP5.0 Module-3 Assignments

Batch Name:	**Infosys FP5.0 Summer Internship 2018**
Enrollment No:	**R110215062**
SAP ID:	**500044606**
NAME:	**Kanwaljit Singh**
Semester:	**VI**
Branch:	**CSE CCVT**

NAME - KANWALJIT SINGH (62)
SAP ID – 500044606, Roll - 62
CCVT – 6th SEM

Submitted To: Mr. Deepak Kumar Sharma
SIGNATURE/REMARKS

Assignment-1

Consider a table "Employer" in Oracle database. Structure and sample data for this table is given below.

Table Structure:

Column Name	Type	Size	Description
CompanyID	Varchar2	5	Primary key eg: C1001
CompanyName	Varchar2	30	Not Null
EmailId	Varchar2	20	Foreign Key referring to Users table
Mobile	Number	10	Must be 10 digit UNIQUE
City	Varchar2	20	
IndustryType	Varchar2	20	
FunctionalArea	Varchar2	20	
MembershipPlan	Varchar2	20	Either Trial or Premium Monthly or Premium Yearly
DateofSignup	Date		Must be greater or equal to current date. Current Date as Default Value
DateofRenewal	Date		Must be based on Membership plan
Renewal status	Varchar2	20	Active or Expired

Company ID	Company Name	EmailID	Mobile	City	IndustryType	Functional Area	Membership Plan	DateOf SignUp	DateOf Renewal	Renewal Status
'C1000'	'Infosys Limited'	'jobs@infosys.com'	7896579875	'Chennai'	'IT'	'Accounting'	'Yearly'	'1-Jul-16'	'30-Jun-17'	'Active'
'C1001'	'Accenture'	'careers@accenture.com'	9878776567	'Bangalore'	'IT'	'Marketing'	'Monthly'	'2-Jun-16'	'1-Jun-17'	'Active'
'C1002'	'HP'	'openings@hp.com'	8789878750	'Mumbai'	'IT'	'Marketing'	'Monthly'	'12-Jul-16'	'11-Jul-17'	'Active'
'C1003'	'NewGen'	'jobs@newgen.com'	8877643228	'Bangalore'	'Manufacturing'	'Marketing'	'Yearly'	'2-Sep-16'	'1-Sep-17'	'Expired'

Write a Python program for the following:

1)Connect to Oracle database
2)Fetch all the rows from the table Employer
3)Display all the rows
4)Display the count of rows fetched
5)Display the description of all columns of the table
6)Close the connection

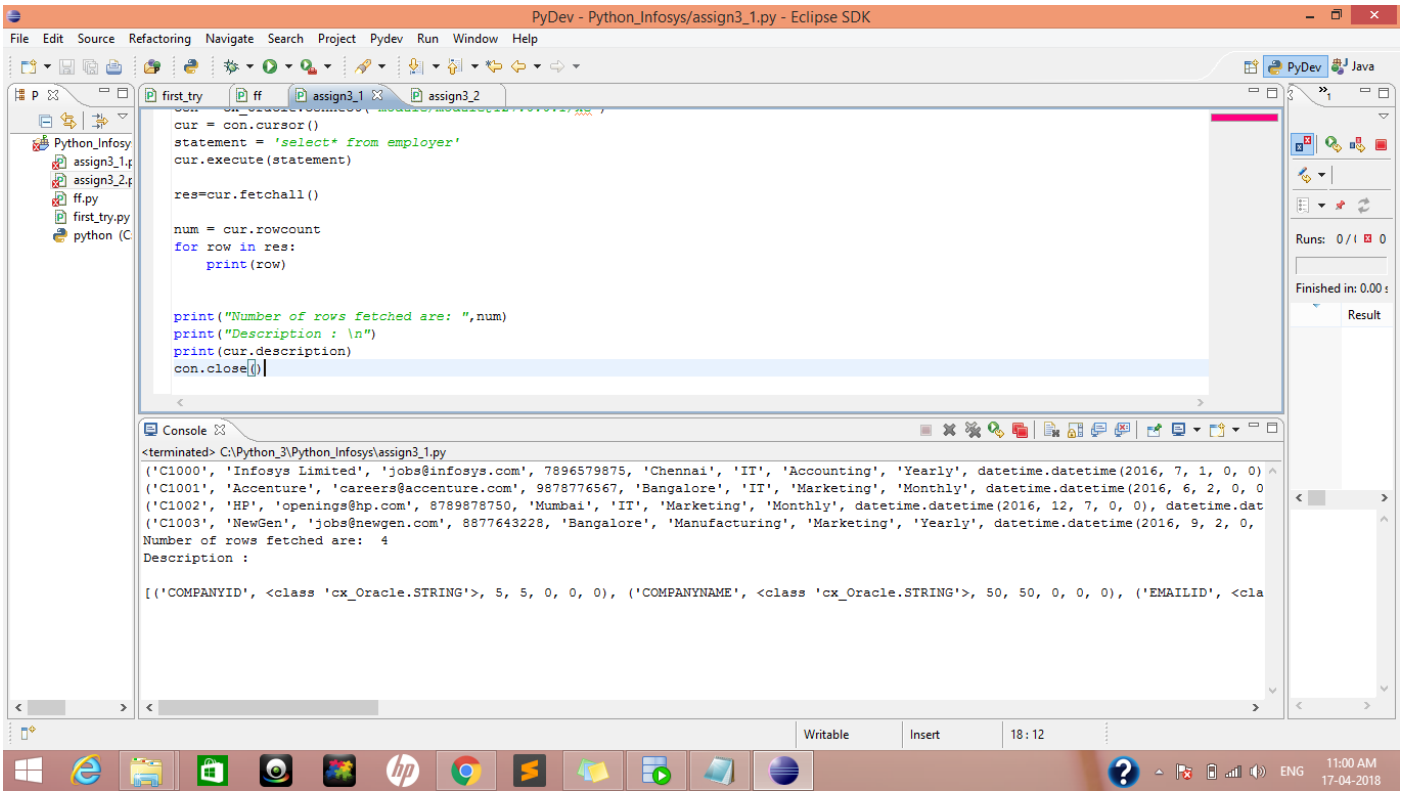
NAME - KANWALJIT SINGH (62)
SAP ID – 500044606, Roll - 62
CCVT – 6th SEM

Submitted To: Mr. Deepak Kumar Sharma
SIGNATURE/REMARKS

Code-

```
import cx_Oracle

con = cx_Oracle.connect('module/module@127.0.0.1/xe')
cur = con.cursor()
statement = 'select* from employer'
cur.execute(statement)

res=cur.fetchall()

num = cur.rowcount
for row in res:
   print(row)


print("Number of rows fetched are: ",num)
print("Description : \n")
print(cur.description)
con.close()
```

Output-



NAME - KANWALJIT SINGH (62)
SAP ID – 500044606, Roll - 62
CCVT – 6th SEM

Submitted To: Mr. Deepak Kumar Sharma
SIGNATURE/REMARKS

Assignment-2

InfoTech Systems wants to retrieve certain information regarding their employers. Help them implement the following business requirements:

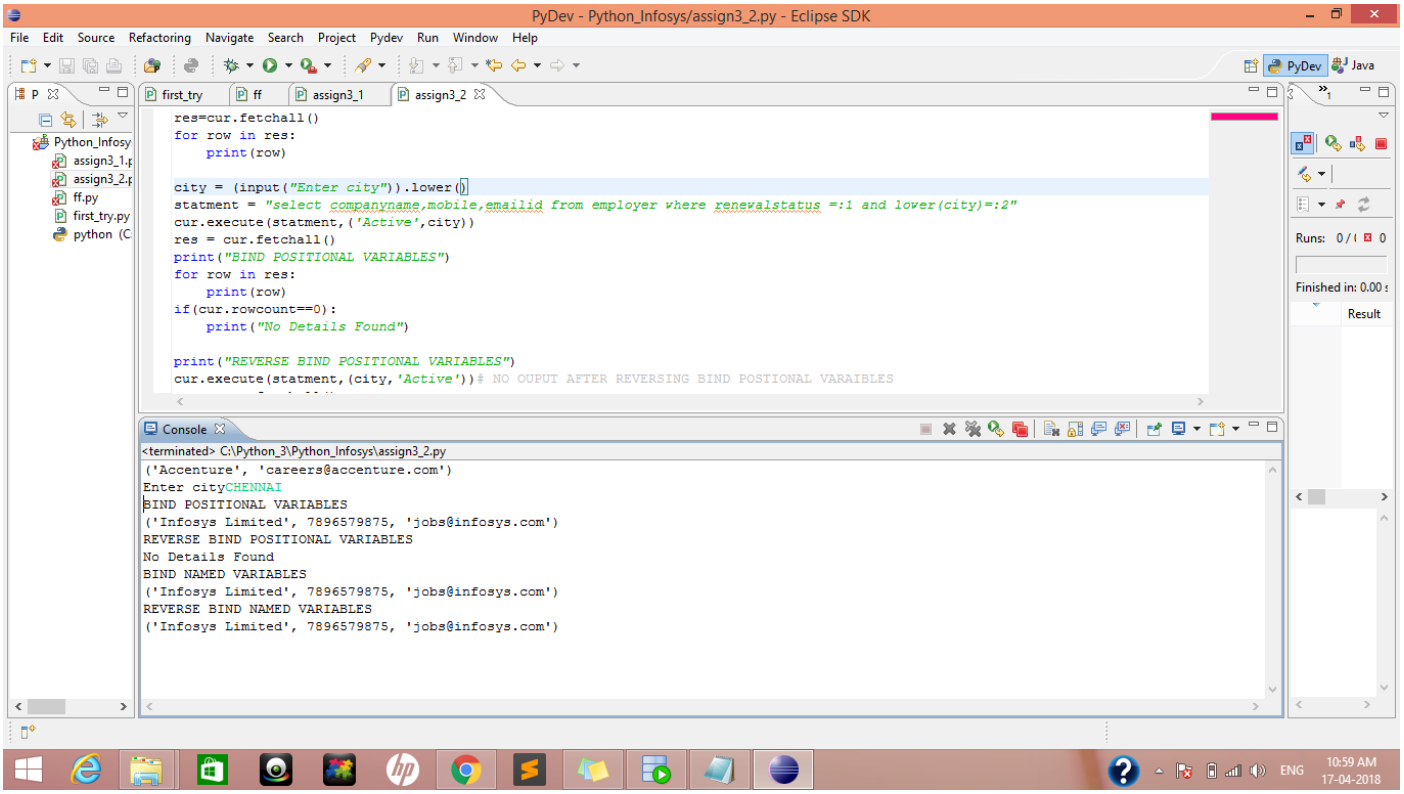
1)Retrieve the name and email id of all 'IT' companies in 'Bangalore'.
2)Retrieve the name, mobile number and email id of all companies in a given city whose Renewal Status is 'Active'. Accept 'city' and 'functionalarea' as an input from user. Use positional bind variables.
3)Reverse the order of passing the parameter values in the above program and observe the output.
4)Implement the scenario in question# 2 using named bind variables.
5)Reverse the order of passing of the bind variables in the above program and observe the output. Are you still getting the same result?

Code-

```
import cx_Oracle

con = cx_Oracle.connect('module/module@127.0.0.1/xe')
cur = con.cursor()
statment = "select companyname,emailid from employer where INDUSTRYTYPE='IT' and CITY='Bangalore'"
cur.execute(statment)
res=cur.fetchall()
for row in res:
   print(row)

city = (input("Enter city")).lower()
statment = "select companyname,mobile,emailid from employer where renewalstatus =:1 and lower(city)=:2"
cur.execute(statment,('Active',city))
res = cur.fetchall()
print("BIND POSITIONAL VARIABLES")
for row in res:
   print(row)
if(cur.rowcount==0):
   print("No Details Found")

print("REVERSE BIND POSITIONAL VARIABLES")
cur.execute(statment,(city,'Active'))# NO OUPUT AFTER REVERSING BIND POSTIONAL VARAIBLES
res = cur.fetchall()
for row in res:
   print(row)
if(cur.rowcount==0):
   print("No Details Found")

print("BIND NAMED VARIABLES")
statment = "select companyname,mobile,emailid from employer where renewalstatus =:par1 and
lower(city)=:par2"
cur.execute(statment,{'par1':'Active','par2':city})
res = cur.fetchall()
for row in res:
   print(row)
```

```
if(cur.rowcount==0):
   print("No Details Found")

print("REVERSE BIND NAMED VARIABLES")
statment = "select companyname,mobile,emailid from employer where renewalstatus =:par1 and
lower(city)=:par2"
cur.execute(statment,{'par2':city,'par1':'Active'})# SAME OUTPUT OUPUT AFTER REVERSING BIND POSTIONAL
VARAIBLES
res = cur.fetchall()
for row in res:
   print(row)
if(cur.rowcount==0):
   print("No Details Found")
con.close()
```

Output-



NAME - KANWALJIT SINGH (62)
SAP ID – 500044606, Roll - 62
CCVT – 6th SEM

Submitted To: Mr. Deepak Kumar Sharma
SIGNATURE/REMARKS

Assignment-3

InfoTech Systems is creating an online application for automating the task of job search between employer and job seekers.

1.Create a table 'Users' from Python code. The column details are given below:

Column Name	Type	Size	Description
UserId	Number	10	Primary key, Must be a digit
UserName	Varchar2	30	Cannot be null
Password	Varchar2	20	Cannot be null
UserType	Varchar2	20	Value can be either 'Employer 'or 'Jobseeker'

2. Insert the following data intoUserstable using cx_Oracle as per the specifications provided below:

UserId	Username	Password	UserType
1	jobs@infosys.com	jobs@infosys	Employer
2	careers@accenture.com	Acc1	Employer
3	rahulitsme@gmaill.com	rahulindia93	Jobseeker
4	careers@amazon.com	amazonindia	Employer

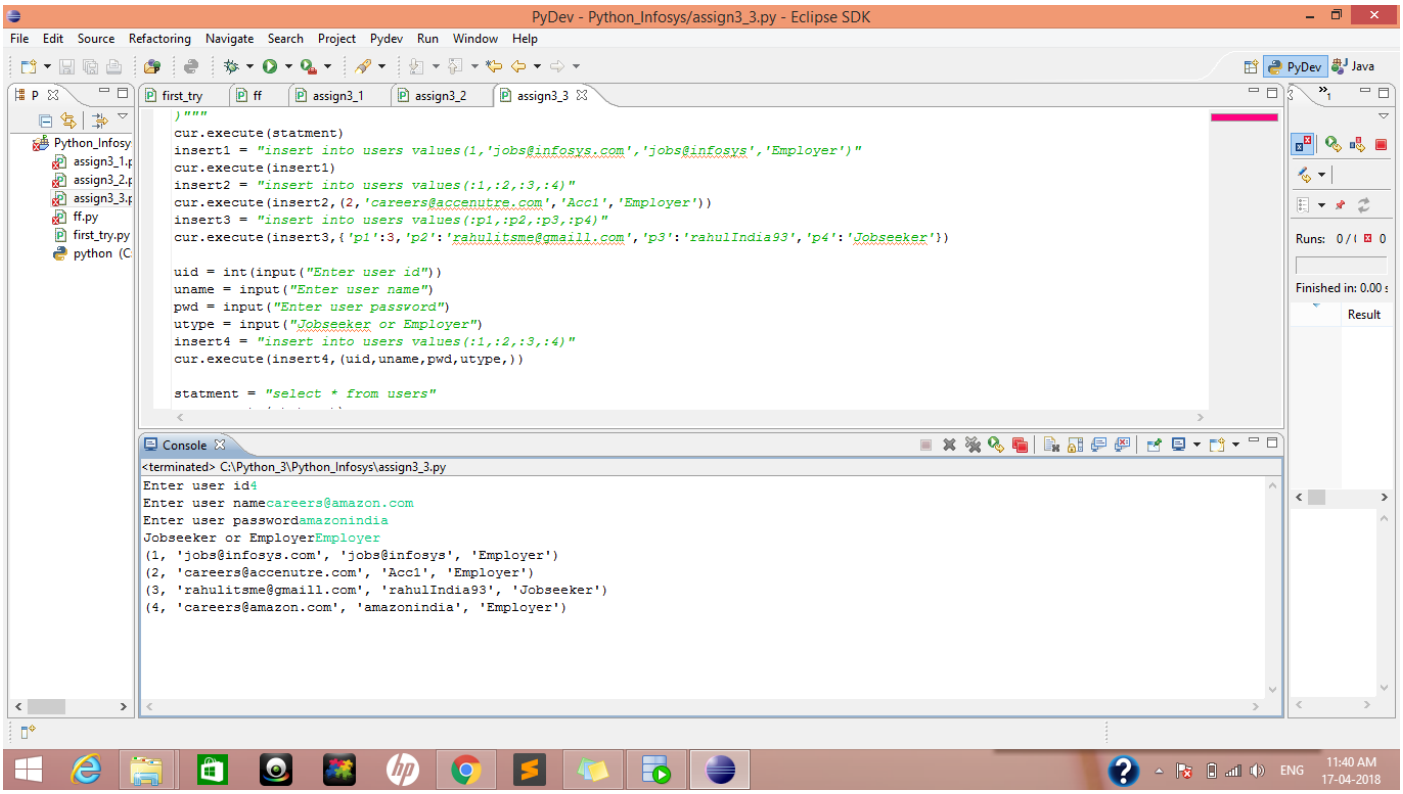
- Insert first row using hard-coded values in INSERT query.
- Insert second row using positional bind variables.
- Insert third row using named bind variables.
- Accept the values for fourth row from user and insert using bind variables.
- Fetch and display all the records from users table.

Code-

```
import cx_Oracle

con = cx_Oracle.connect('module/module@127.0.0.1/xe')
cur = con.cursor()
statment = """create table users(
userid number(10) primary key,
username varchar2(30) not null,
password varchar2(20) not null,
```

```
usertype varchar2(20) CHECK (usertype IN('Employer','Jobseeker'))
)"""
cur.execute(statment)
insert1 = "insert into users values(1,'jobs@infosys.com','jobs@infosys','Employer')"
cur.execute(insert1)
insert2 = "insert into users values(:1,:2,:3,:4)"
cur.execute(insert2,(2,'careers@accenutre.com','Acc1','Employer'))
insert3 = "insert into users values(:p1,:p2,:p3,:p4)"
cur.execute(insert3,{'p1':3,'p2':'rahulitsme@gmaill.com','p3':'rahulIndia93','p4':'Jobseeker'})

uid = int(input("Enter user id"))
uname = input("Enter user name")
pwd = input("Enter user password")
utype = input("Jobseeker or Employer")
insert4 = "insert into users values(:1,:2,:3,:4)"
cur.execute(insert4,(uid,uname,pwd,utype,))

statment = "select * from users"
cur.execute(statment)
res = cur.fetchall()
for row in res:
    print(row)

con.close()
```

Output-



NAME - KANWALJIT SINGH (62)
SAP ID – 500044606, Roll - 62
CCVT – 6th SEM

Submitted To: Mr. Deepak Kumar Sharma
SIGNATURE/REMARKS

Assignment-4

Bloom Technology wants to maintain their employee's vehicle details to make parking facility flexible to the employees.

1.Create the following Vehicle table as a part of the application. Specifications are provided below:

Column Names	Datatype & Size	Constraints
Vehicleid	Number(5)	Primary Key
Vehiclename	Varchar2(10)	

2.Insert the following records using executemany() function of cursor. Use positional bind variables.

Vehicleid	Vehiclename
2001	Toyota
2002	Maruti
2003	Nissan
2004	Hyundai

3.Insert two more rows using named bind variables(use executemany() function)

Vehicleid	Vehiclename
2006	Honda
2007	Volkswagen

4. Fetch and display all the records from Vehicle table.

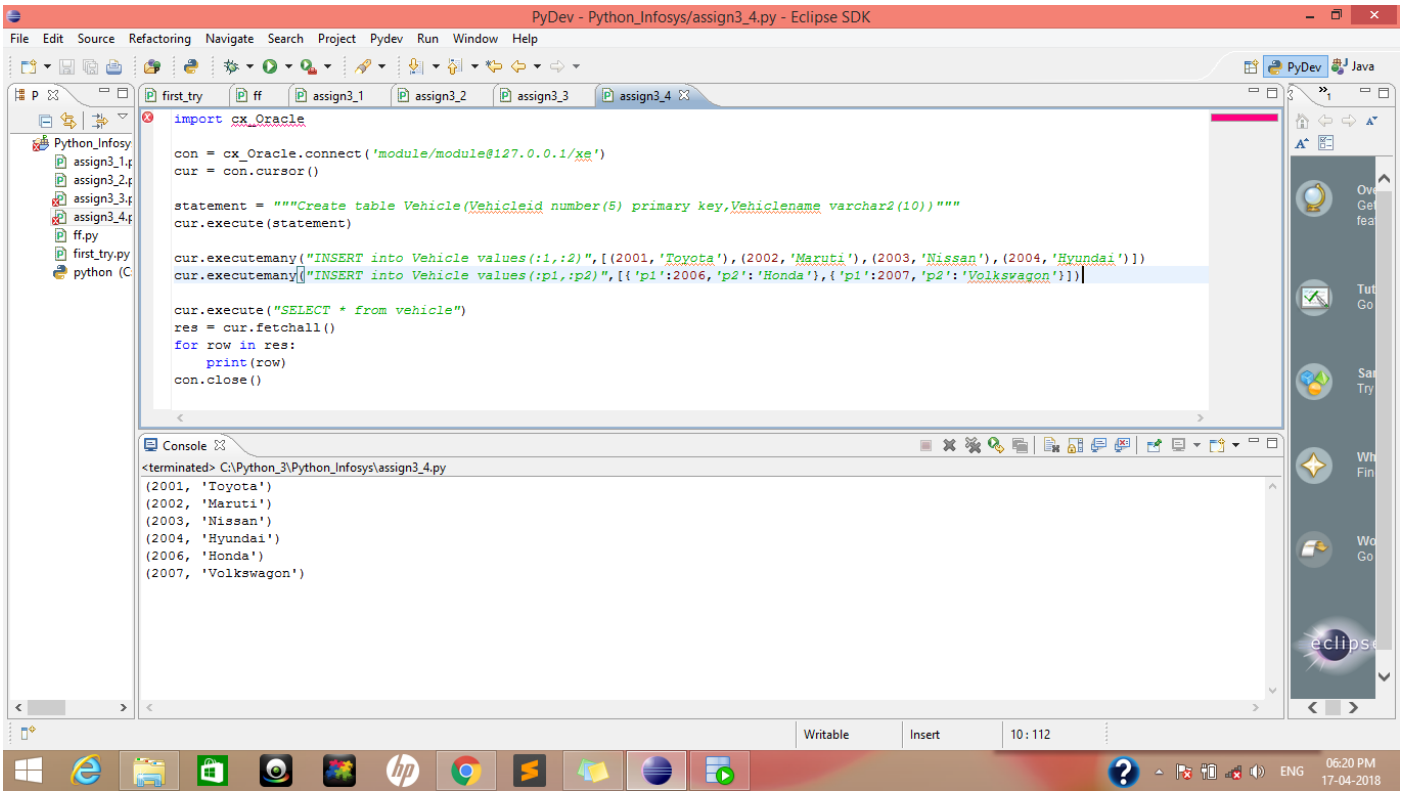
NAME - KANWALJIT SINGH (62)
SAP ID – 500044606, Roll - 62
CCVT – 6th SEM

Submitted To: Mr. Deepak Kumar Sharma
SIGNATURE/REMARKS

Code-

```
import cx_Oracle

con = cx_Oracle.connect('module/module@127.0.0.1/xe')
cur = con.cursor()

statement = """Create table Vehicle(Vehicleid number(5) primary key,Vehiclename varchar2(10))"""
cur.execute(statement)

cur.executemany("INSERT into Vehicle
values(:1,:2)",[(2001,'Toyota'),(2002,'Maruti'),(2003,'Nissan'),(2004,'Hyundai')])
cur.executemany("INSERT into Vehicle values(:p1,:p2)",[{'p1':2006,'p2':'Honda'},{'p1':2007,'p2':'Volkswagon'}])

cur.execute("SELECT * from vehicle")
res = cur.fetchall()
for row in res:
   print(row)

con.close()
```

Output-



Assignment-5

Refer to the table 'users' created earlier. The existing table data for "users" table is given below:

UserId	Username	Password	UserType
1	jobs@infosys.com	jobs@infosys	Employer
2	careers@accenture.com	Acc1	Employer
3	rahulitsme@gmaill.com	rahulindia93	Jobseeker
4	careers@amazon.com	amazonindia	Employer

1.Modify the username and usertype of the user with userid = 4 with the following values:

- Username: lookingforjob@yahoo.com
- UserType: JobseekerFetch

and observe the values of 'username' and 'usertype' of the user with 'userid = 4' before and after'update' operation.

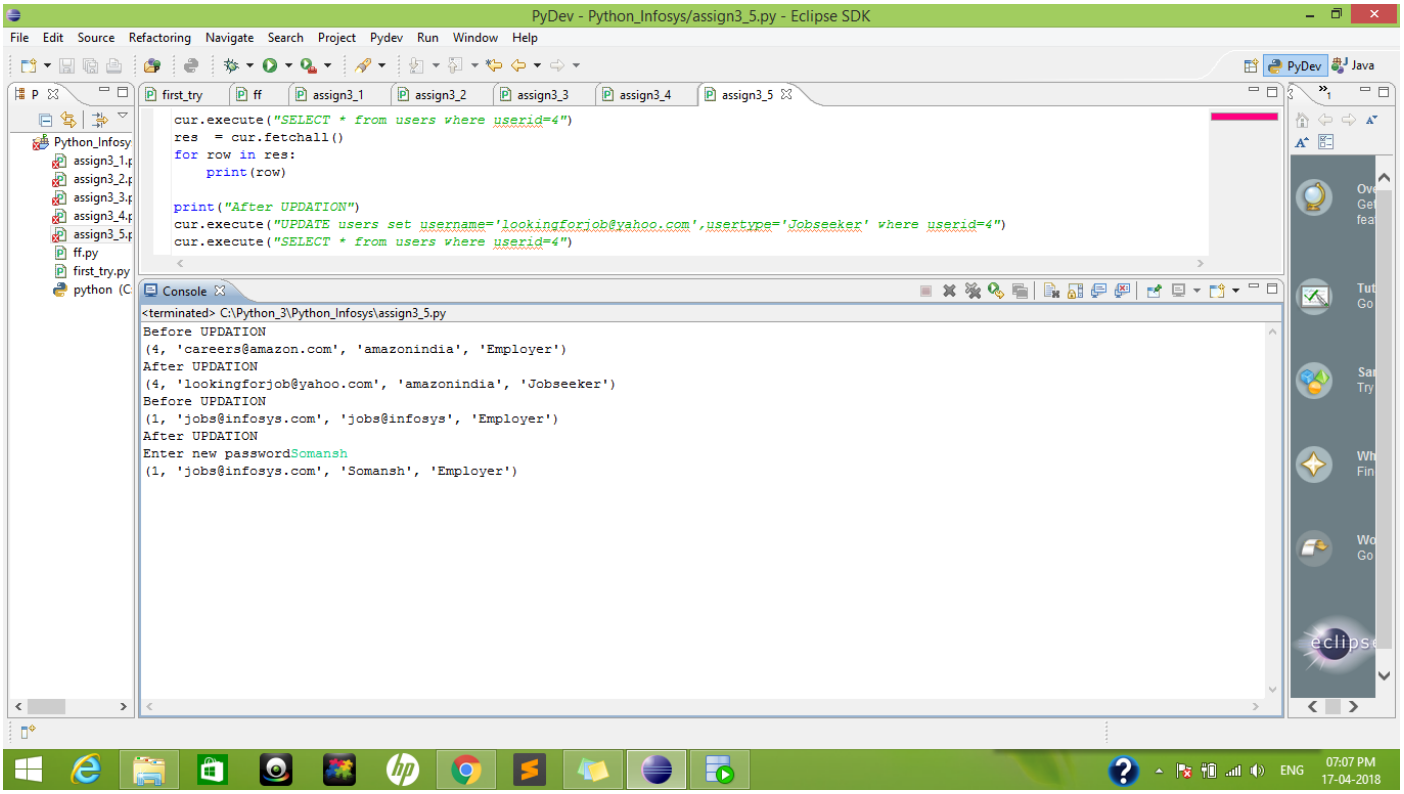
2.Change the password for userid = 1. Accept the new password as an input from user. Fetch and observethe value of 'password' of the user with 'userid = 1' before and after 'update' operation.

Code-

```
import cx_Oracle

con = cx_Oracle.connect('module/module@127.0.0.1/xe')
cur = con.cursor()
print("Before UPDATION")
cur.execute("SELECT * from users where userid=4")
res  = cur.fetchall()
for row in res:
   print(row)

print("After UPDATION")
cur.execute("UPDATE users set username='lookingforjob@yahoo.com',usertype='Jobseeker' where userid=4")
cur.execute("SELECT * from users where userid=4")
res  = cur.fetchall()
for row in res:
   print(row)

print("Before UPDATION")
cur.execute("SELECT * from users where userid=1")
res  = cur.fetchall()
```

```
for row in res:
   print(row)

print("After UPDATION")
pwd = input("Enter new password")
cur.execute("UPDATE users set password=:p1 where userid=1",{'p1':pwd})
cur.execute("SELECT * from users where userid=1")
res  = cur.fetchall()
for row in res:
   print(row)

con.close()
```

Output-



NAME - KANWALJIT SINGH (62)
SAP ID – 500044606, Roll - 62
CCVT – 6th SEM

Submitted To: Mr. Deepak Kumar Sharma
SIGNATURE/REMARKS

Assignment-6

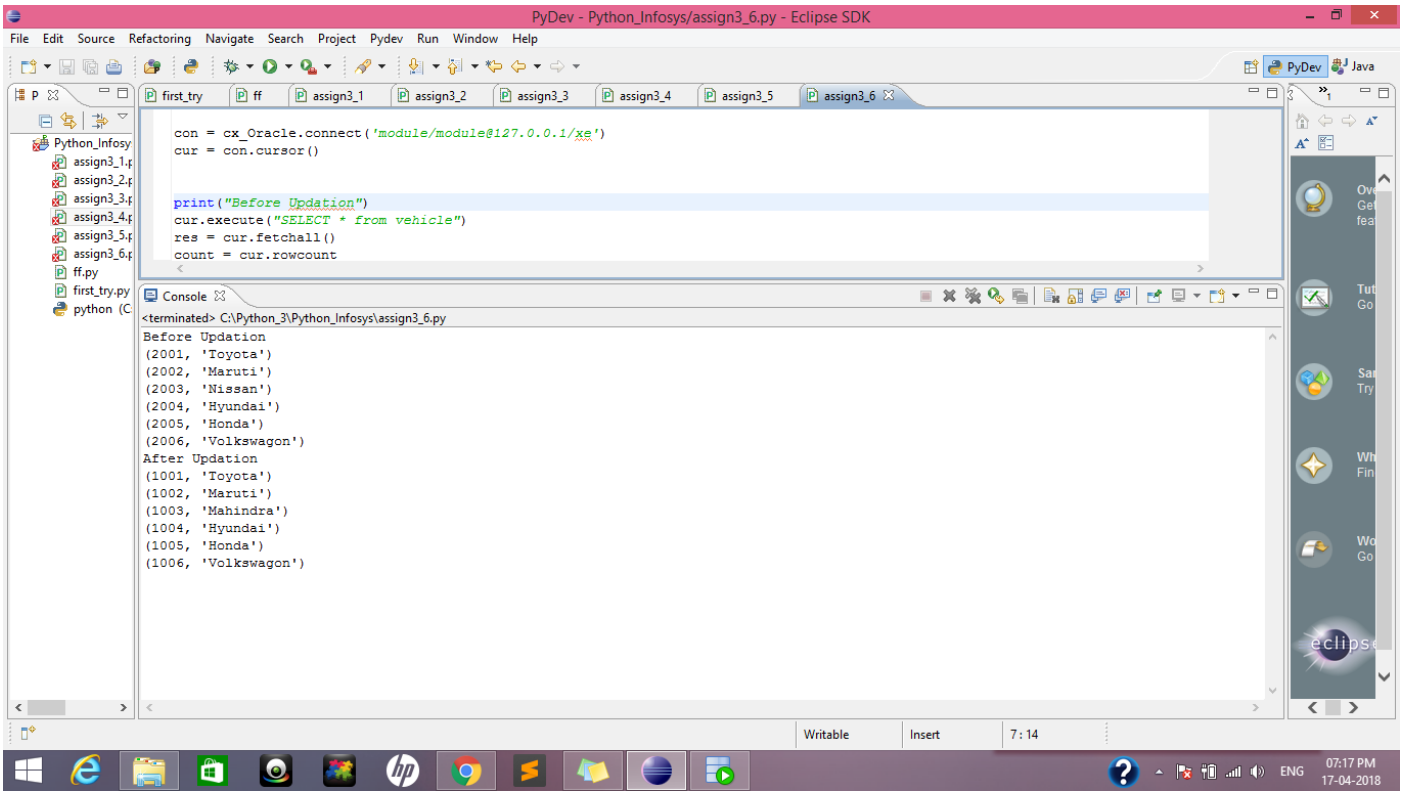
Consider the 'Vehicle' table created earlier. Currently 'Vehicleid' is an integer field with values starting from 2001 onwards.

- Update the values of 'Vehicleid' to start from 1001 onwards as shown below.-
  Hint – Use loops
- Update the Vehiclename to "Mahindra" for vehicle with vehicle id 1003.
- Fetch and display the values before and after the update operation.

Code-

```
import cx_Oracle

con = cx_Oracle.connect('module/module@127.0.0.1/xe')
cur = con.cursor()


print("Before Updation")
cur.execute("SELECT * from vehicle")
res = cur.fetchall()
count = cur.rowcount
for row in res:
   print(row)

old = 2001
new = 1001
i = 0
while(i<count):
   cur.execute("UPDATE vehicle set vehicleid=:1 where vehicleid=:2",(new,old))
   old=old+1
   new=new+1
   i=i+1
cur.execute("UPDATE vehicle set vehiclename='Mahindra' where vehicleid=1003")
print("After Updation")
cur.execute("SELECT * from vehicle")
res = cur.fetchall()
count = cur.rowcount
for row in res:
   print(row)
con.commit()
con.close()
```

Output-


PyDev - Python_Infosys/assign3_6.py - Eclipse SDK
File Edit Source Refactoring Navigate Search Project Pydev Run Window Help
first_try ff assign3_1 assign3_2 assign3_3 assign3_4 assign3_5 assign3_6
con = cx_Oracle.connect('module/module@127.0.0.1/xe')
cur = con.cursor()
print("Before Updation")
cur.execute("SELECT * from vehicle")
res = cur.fetchall()
count = cur.rowcount
Console
<terminated> C:\Python_3\Python_Infosys\assign3_6.py
Before Updation
(2001, 'Toyota')
(2002, 'Maruti')
(2003, 'Nissan')
(2004, 'Hyundai')
(2005, 'Honda')
(2006, 'Volkswagon')
After Updation
(1001, 'Toyota')
(1002, 'Maruti')
(1003, 'Mahindra')
(1004, 'Hyundai')
(1005, 'Honda')
(1006, 'Volkswagon')
Writable
Insert
7 : 14
ENG
07:17 PM
17-04-2018

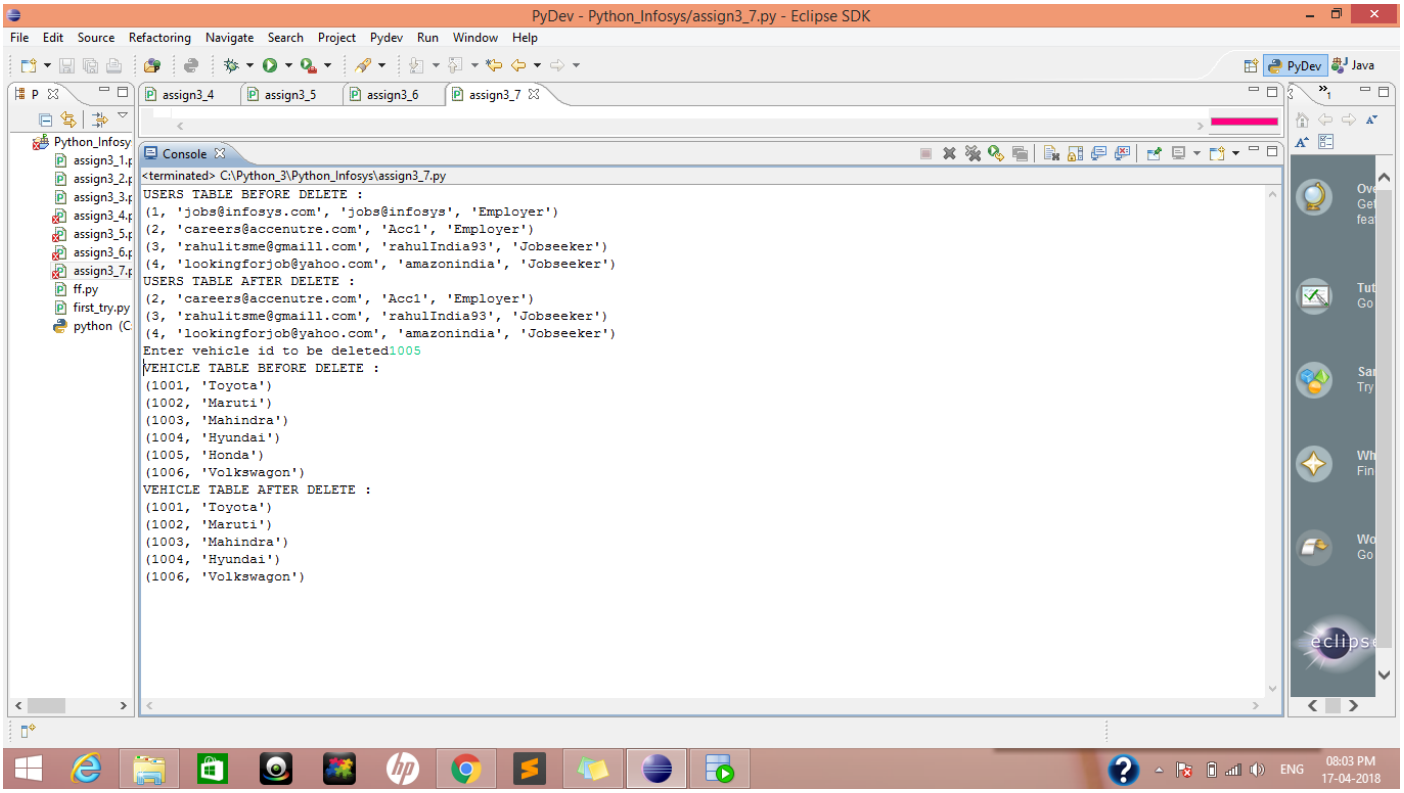
Assignment-7

1)Consider 'users' table. Delete the record of user with userid = 1.
2)Delete a record from 'Vehicle' table using named bind variables. Accept VehicleId as an input from the user.

Code-

```
import cx_Oracle

con = cx_Oracle.connect('module/module@127.0.0.1/xe')
cur = con.cursor()
print("USERS TABLE BEFORE DELETE :")
cur.execute("SELECT * from users")
res = cur.fetchall()
for row in res:
    print(row)
print("USERS TABLE AFTER DELETE :")
cur.execute("DELETE FROM users where userid=1")
cur.execute("SELECT * from users")
res = cur.fetchall()
for row in res:
    print(row)

vid = int(input("Enter vehicle id to be deleted"))
print("VEHICLE TABLE BEFORE DELETE :")
cur.execute("SELECT * from vehicle")
res = cur.fetchall()
for row in res:
    print(row)
cur.execute("DELETE FROM vehicle where vehicleid=:id",{'id':vid})
print("VEHICLE TABLE AFTER DELETE :")
cur.execute("SELECT * from vehicle")
res = cur.fetchall()
for row in res:
    print(row)
con.close()
```

Output-


PyDev - Python_Infosys/assign3_7.py - Eclipse SDK
File Edit Source Refactoring Navigate Search Project Pydev Run Window Help
assign3_4
assign3_5
assign3_6
assign3_7
Console
<terminated> C:\Python_3\Python_Infosys\assign3_7.py
USERS TABLE BEFORE DELETE :
(1, 'jobs@infosys.com', 'jobs@infosys', 'Employer')
(2, 'careers@accenutre.com', 'Acc1', 'Employer')
(3, 'rahulitsme@gmaill.com', 'rahulIndia93', 'Jobseeker')
(4, 'lookingforjob@yahoo.com', 'amazonindia', 'Jobseeker')
USERS TABLE AFTER DELETE :
(2, 'careers@accenutre.com', 'Acc1', 'Employer')
(3, 'rahulitsme@gmaill.com', 'rahulIndia93', 'Jobseeker')
(4, 'lookingforjob@yahoo.com', 'amazonindia', 'Jobseeker')
Enter vehicle id to be deleted1005
VEHICLE TABLE BEFORE DELETE :
(1001, 'Toyota')
(1002, 'Maruti')
(1003, 'Mahindra')
(1004, 'Hyundai')
(1005, 'Honda')
(1006, 'Volkswagon')
VEHICLE TABLE AFTER DELETE :
(1001, 'Toyota')
(1002, 'Maruti')
(1003, 'Mahindra')
(1004, 'Hyundai')
(1006, 'Volkswagon')
ENG
08:03 PM
17-04-2018

NAME - KANWALJIT SINGH (62)
SAP ID – 500044606, Roll - 62
CCVT – 6th SEM

Submitted To: Mr. Deepak Kumar Sharma
SIGNATURE/REMARKS

Assignment-8

Consider a scenario from a State Banking organization. The account table is created to store the account details of a customer (Assume every customer can have only one account). Use cx_Oracle module to implement the following requirements from Python code.(Do not execute the queries in database directly)

1.Create the table 'Account' as per below specifications:

Column Name	Column Type	Description
CustomerId	Number	Primary Key
AccountNo	Varchar2(15)	Alphanumeric
AccountType	Varchar2(15)	Can be Savings, Current or Recurring
Balance	Number	Account balance of the customer

2.Insert the following rows in the table:

CustomerId	AccountNo	AccountType	Balance
101	IBI1001	Savings	0
102	IBI1002	Current	1200
103	IBI1003	Savings	6543
104	IBI1004	Recurring	7500
105	IBI1005	Current	0

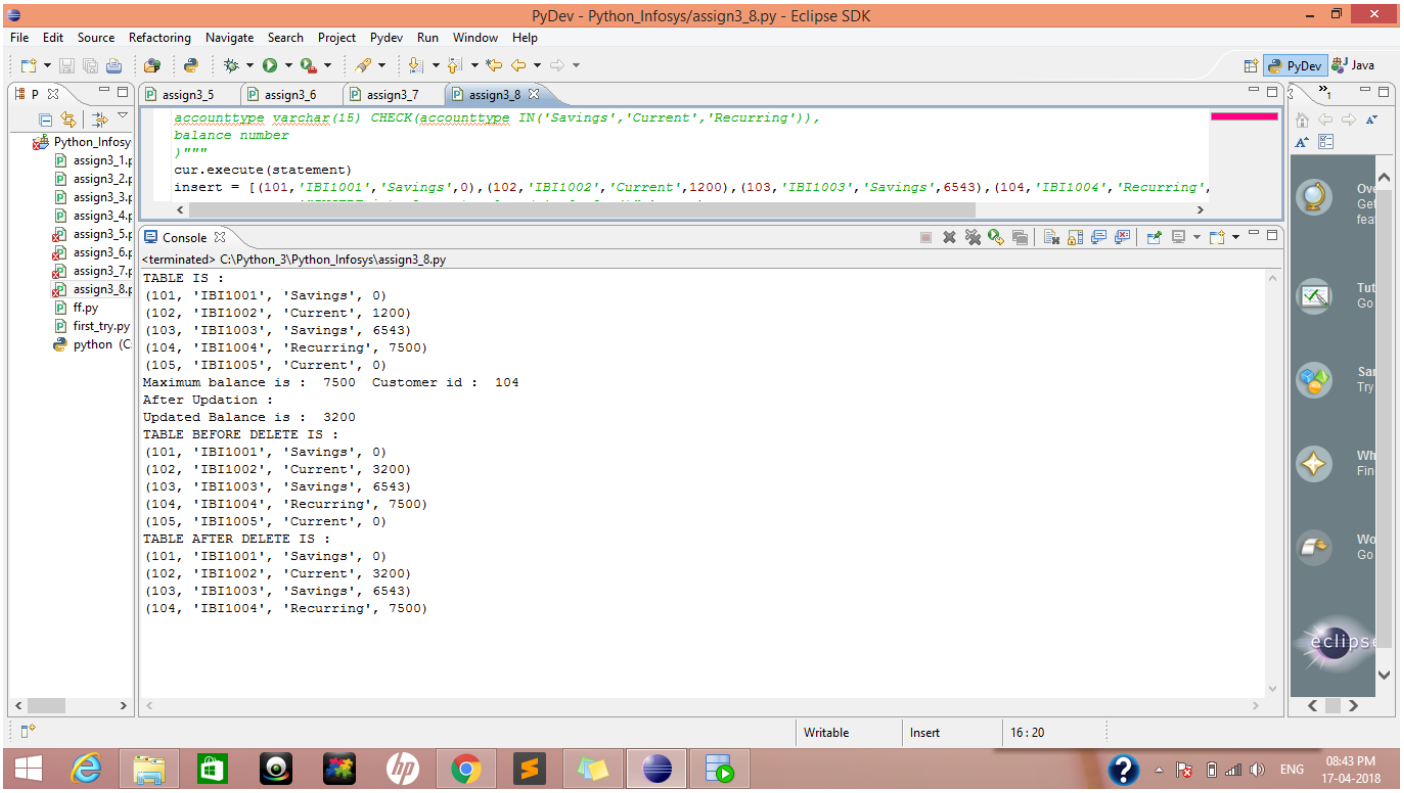
3.Display the customer id and account balance of the customer with maximum account balance.
4.Fetch the account balance of the customer with customer id 102 and store it in a Python variable – 'acct_bal'.
5.Increment 'acct_bal' with 2000 and update the 'Balance' field of the table (for that particular customer) with the new value.
6.Fetch and observe the updated account balance of the customer with customer id 102.
7.Delete the 'Current' accounts with zero balance.

Code-

```
import cx_Oracle

con = cx_Oracle.connect('module/module@127.0.0.1/xe')
cur = con.cursor()
statement = """create table Account(
customerid number primary key,
accountno varchar2(15),
accounttype varchar(15) CHECK(accounttype IN('Savings','Current','Recurring')),
```

```
balance number
)"""
cur.execute(statement)
insert =
[(101,'IBI1001','Savings',0),(102,'IBI1002','Current',1200),(103,'IBI1003','Savings',6543),(104,'IBI1004','Recurring',75
00),(105,'IBI1005','Current',0)]
cur.executemany("INSERT into Account values(:1,:2,:3,:4)",insert)
cur.execute("SELECT *from account")
res = cur.fetchall()
print("TABLE IS :")
for row in res:
    print(row)
lid = 0
lba = 0
k = 0
acct_bal = 0
for row in res:
    if row[3]>lba:
        lba = row[3]
        lid = row[0]
    if row[0]==102:
        acct_bal = row[3]

print("Maximum balance is : ",lba," Customer id : ",lid)
acct_bal+=2000
cur.execute("UPDATE account set balance=:1 where customerid=:2",(acct_bal,102))
print("After Updation : ")
cur.execute("SELECT * from account where customerid=102")
res = cur.fetchall()
print("Updated Balance is : ",res[0][3])

cur.execute("SELECT * from account")
res = cur.fetchall()
print("TABLE BEFORE DELETE IS :")
for row in res:
    print(row)
for row in res:
    if row[2]=='Current' and row[3]==0:
        cur.execute("DELETE FROM account where customerid=:c1",{'c1':row[0]})
cur.execute("SELECT * from account")
res = cur.fetchall()
print("TABLE AFTER DELETE IS :")
for row in res:
    print(row)
con.close()
```

Output-


PyDev - Python_Infosys/assign3_8.py - Eclipse SDK
File Edit Source Refactoring Navigate Search Project Pydev Run Window Help
assign3_5
assign3_6
assign3_7
assign3_8
accounttype varchar(15) CHECK(accounttype IN('Savings','Current','Recurring')),
balance number
)"""
cur.execute(statement)
insert = [(101,'IBI1001','Savings',0),(102,'IBI1002','Current',1200),(103,'IBI1003','Savings',6543),(104,'IBI1004','Recurring',
Console
<terminated> C:\Python_3\Python_Infosys\assign3_8.py
TABLE IS :
(101, 'IBI1001', 'Savings', 0)
(102, 'IBI1002', 'Current', 1200)
(103, 'IBI1003', 'Savings', 6543)
(104, 'IBI1004', 'Recurring', 7500)
(105, 'IBI1005', 'Current', 0)
Maximum balance is : 7500 Customer id : 104
After Updation :
Updated Balance is : 3200
TABLE BEFORE DELETE IS :
(101, 'IBI1001', 'Savings', 0)
(102, 'IBI1002', 'Current', 3200)
(103, 'IBI1003', 'Savings', 6543)
(104, 'IBI1004', 'Recurring', 7500)
(105, 'IBI1005', 'Current', 0)
TABLE AFTER DELETE IS :
(101, 'IBI1001', 'Savings', 0)
(102, 'IBI1002', 'Current', 3200)
(103, 'IBI1003', 'Savings', 6543)
(104, 'IBI1004', 'Recurring', 7500)
Writable
Insert
16 : 20
ENG
08:43 PM
17-04-2018

Assignment-9

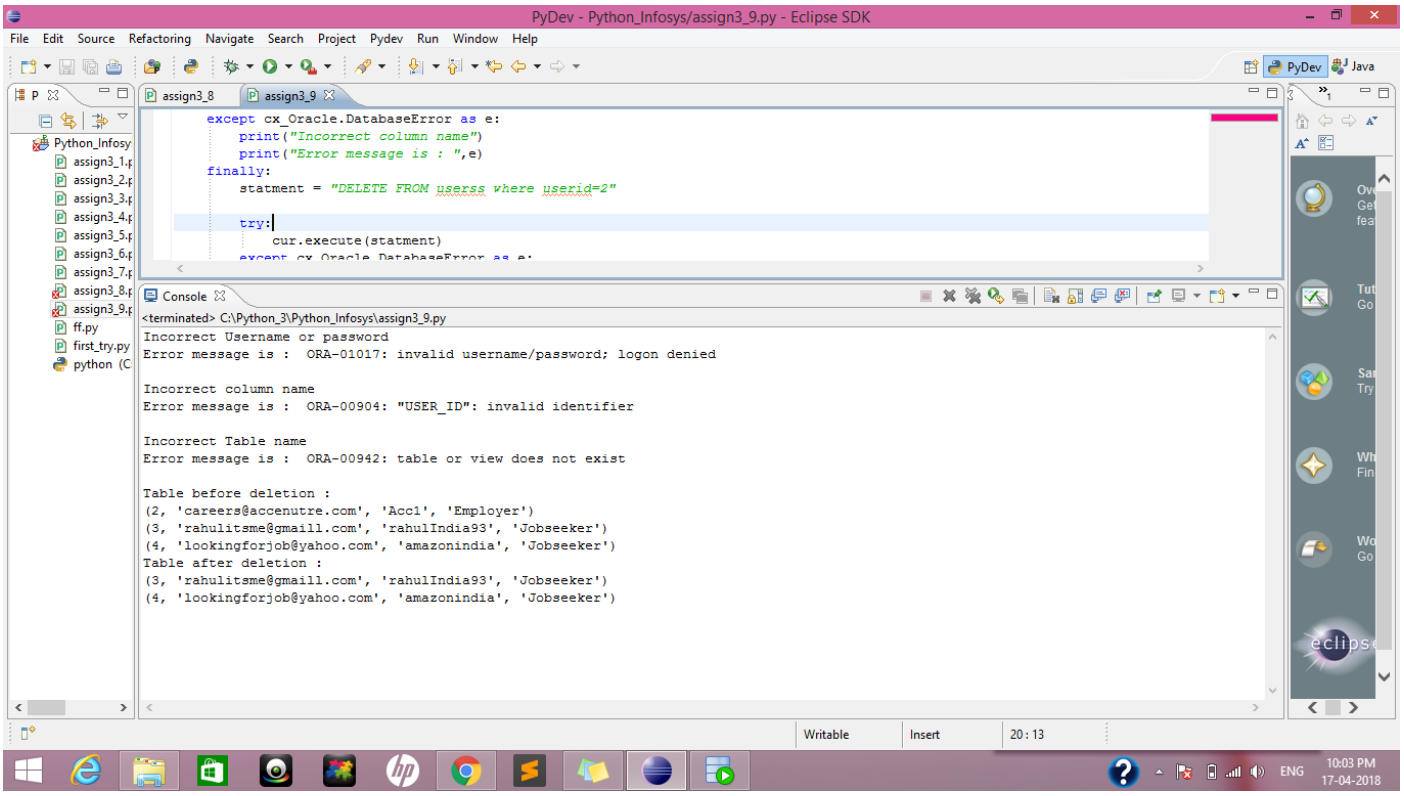
Consider 'users' table already created. It has following data:
There is a requirement to delete the record of user with 'userid' 2.

•Try to mention incorrect column name(e.g. user_id) and observe the error.

Code-

```
import cx_Oracle

try:
    con = cx_Oracle.connect('Incorrect/module@127.0.0.1/xe')
except cx_Oracle.DatabaseError as e:
    print("Incorrect Username or password")
    print("Error message is : ",e)
finally:
    con = cx_Oracle.connect('module/module@127.0.0.1/xe')
    cur = con.cursor()
    try:
        statment = "DELETE FROM users where user_id=2"
        cur.execute(statment)
    except cx_Oracle.DatabaseError as e:
        print("Incorrect column name")
        print("Error message is : ",e)
    finally:
        statment = "DELETE FROM users where userid=2"

        try:
            cur.execute(statment)
        except cx_Oracle.DatabaseError as e:
            print("Incorrect Table name")
            print("Error message is : ",e)
        finally:
            print("Table before deletion : ")
            cur.execute("SELECT * from users")
            res = cur.fetchall()
            for row in res:
                print(row)
            statment = "DELETE FROM users where userid=2"
            cur.execute(statment)
            print("Table after deletion : ")
            cur.execute("SELECT * from users")
            res = cur.fetchall()
            for row in res:
                print(row)
    con.close()
```

Output-


PyDev - Python_Infosys/assign3_9.py - Eclipse SDK
File Edit Source Refactoring Navigate Search Project Pydev Run Window Help
except cx_Oracle.DatabaseError as e:
print("Incorrect column name")
print("Error message is : ",e)
finally:
statment = "DELETE FROM userss where userid=2"
try:
cur.execute(statment)
Console
<terminated> C:\Python_3\Python_Infosys\assign3_9.py
Incorrect Username or password
Error message is : ORA-01017: invalid username/password; logon denied
Incorrect column name
Error message is : ORA-00904: "USER_ID": invalid identifier
Incorrect Table name
Error message is : ORA-00942: table or view does not exist
Table before deletion :
(2, 'careers@accenutre.com', 'Acc1', 'Employer')
(3, 'rahulitsme@gmaill.com', 'rahulIndia93', 'Jobseeker')
(4, 'lookingforjob@yahoo.com', 'amazonindia', 'Jobseeker')
Table after deletion :
(3, 'rahulitsme@gmaill.com', 'rahulIndia93', 'Jobseeker')
(4, 'lookingforjob@yahoo.com', 'amazonindia', 'Jobseeker')
Writable
Insert
20 : 13
ENG
10:03 PM
17-04-2018

Assignment-10

Consider the 'product' table already created. There is a requirement to insert one more row in the table.

•The following Python program is written to insert the row to the 'product' table. Execute the program and observe if there is any error.

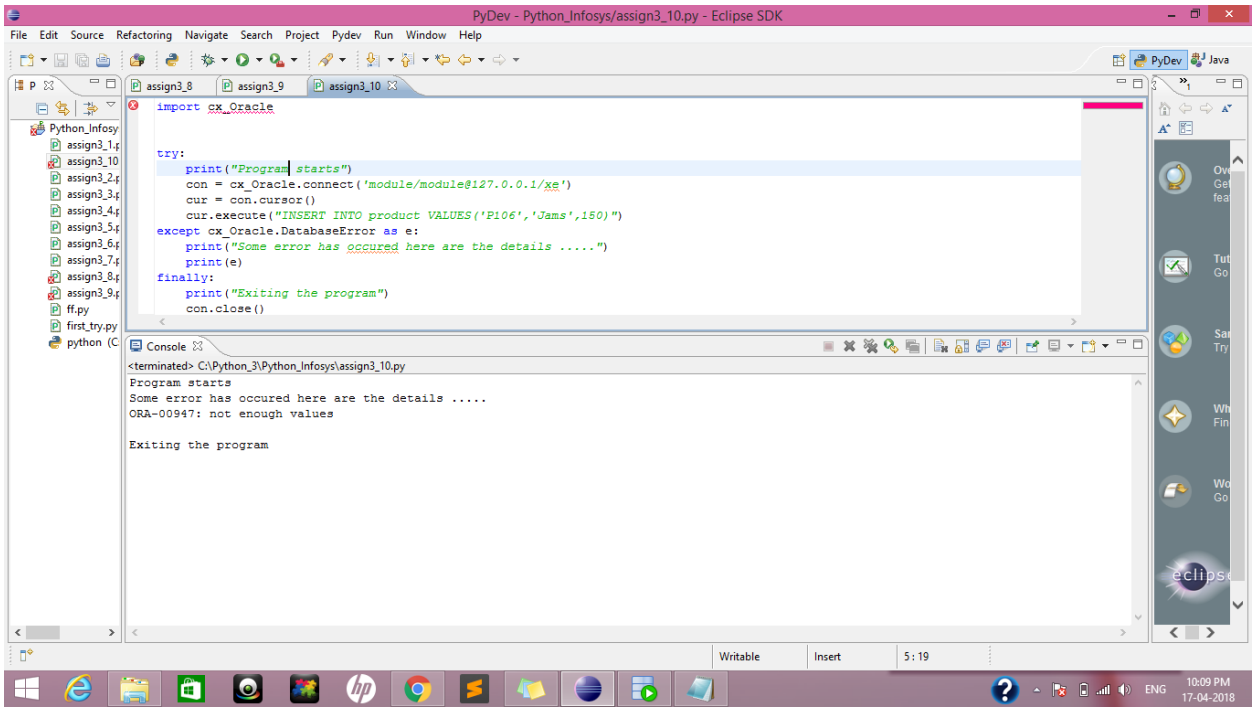
```
import cx_Oracle
con=cx_Oracle.connect("root/password@localhost")
cur=con.cursor()
cur.execute("INSERT INTO product VALUES('P106','Jams',150)")
con.close();
```

•Use exception handling to handle the error (if any) and display error message appropriately.

Code-

```
import cx_Oracle


try:
   print("Program starts")
   con = cx_Oracle.connect('module/module@127.0.0.1/xe')
   cur = con.cursor()
   cur.execute("INSERT INTO product VALUES('P106','Jams',150)")
except cx_Oracle.DatabaseError as e:
   print("Some error has occured here are the details .....")
   print(e)
finally:
   print("Exiting the program")
   con.close()
```

Output-



****END OF MODULE-3****