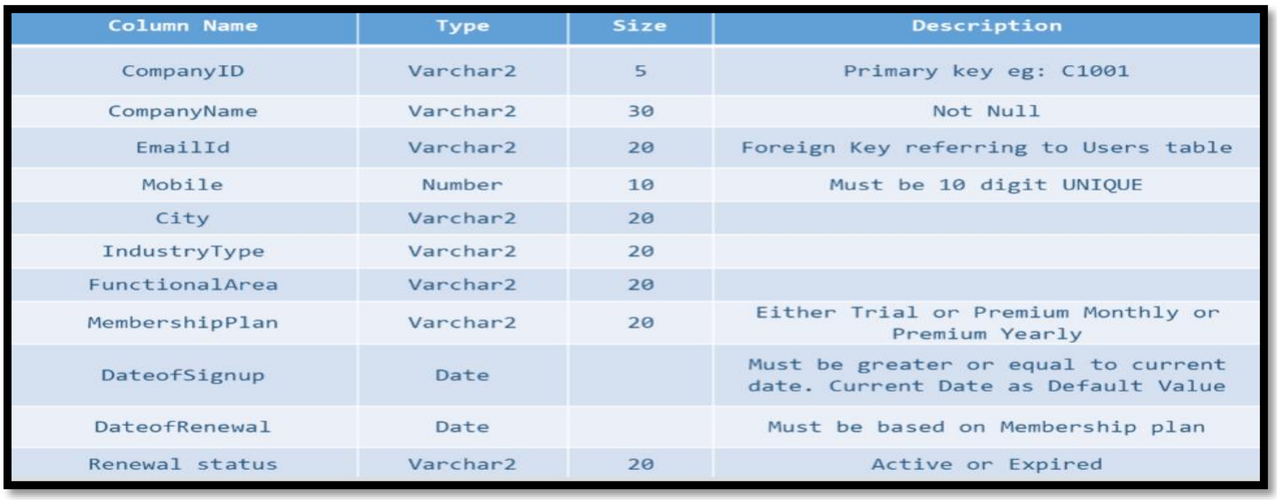
Batch Name: Summer Internship 2018

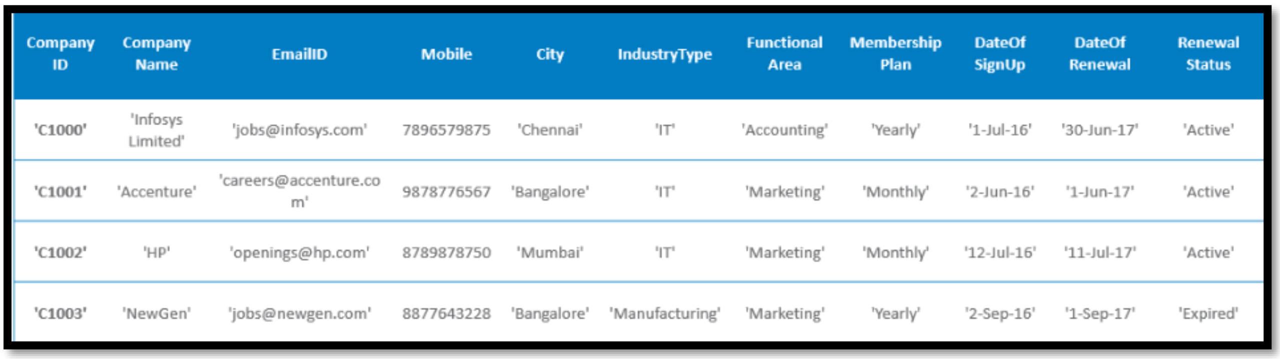
Enrolment No: **R610217007**
SAPID: **500062812**
Name: **DIVYANSHU SINGH**
Semester: **II**
Branch: **CSE –MAINFRAME TECHNOLOGY**

FP5.0 Module-3 Assignments 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

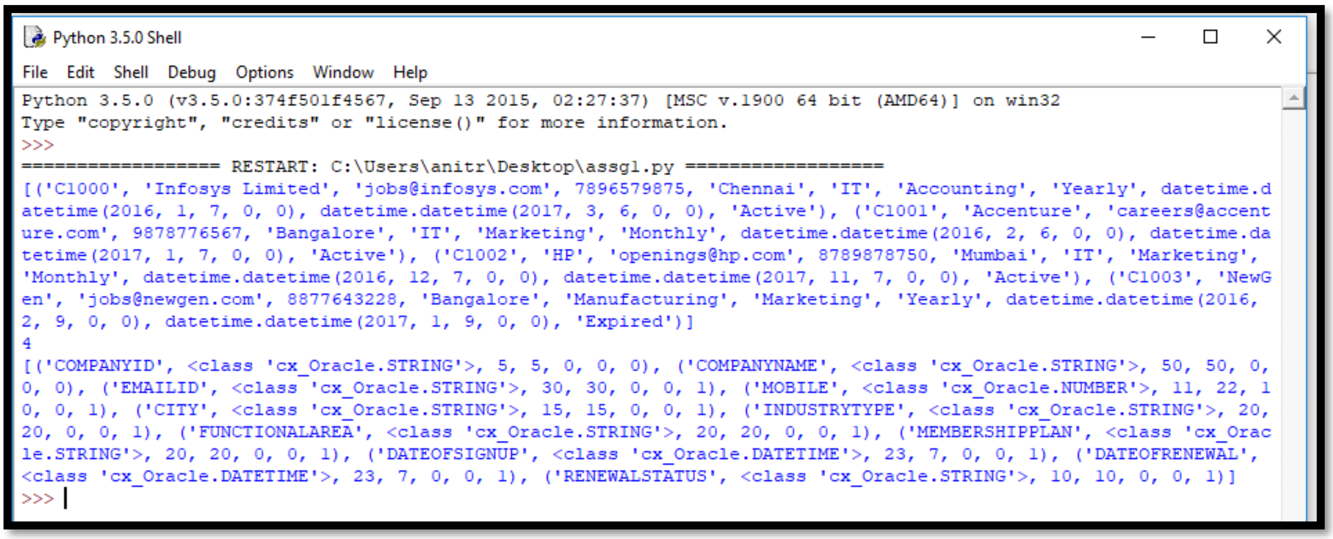


Company ID	Company Name	EmailID	Mobile	City	IndustryType	Functional Area	Membership Plan	DateOf SignUp	DateOf Renewal	Renewal Status
'C1000'	'Infosys Limited'	'jobs@infosys.com'	7896579875	'Chennai'	'IT'	'Accounting'	'Yearly'	'1-Jul-16'	'30-Jun-17'	'Active'
'C1001'	'Accenture'	'careers@accenture.com'	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

SOURCE CODE AND OUTPUT

```
#
import cx_Oracle
con=cx_Oracle.connect('ani/mycycle.com')
cur=con.cursor()
cur.execute("SELECT * From employer")
print(cur.fetchall())
print(cur.rowcount)
print(cur.description)
con.close()
#
```



```
Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
================== RESTART: C:\Users\anitr\Desktop\assg1.py ==================
[('C1000', 'Infosys Limited', 'jobs@infosys.com', 7896579875, 'Chennai', 'IT', 'Accounting', 'Yearly', datetime.d
atetime(2016, 1, 7, 0, 0), datetime.datetime(2017, 3, 6, 0, 0), 'Active'), ('C1001', 'Accenture', 'careers@accent
ure.com', 9878776567, 'Bangalore', 'IT', 'Marketing', 'Monthly', datetime.datetime(2016, 2, 6, 0, 0), datetime.da
tetime(2017, 1, 7, 0, 0), 'Active'), ('C1002', 'HP', 'openings@hp.com', 8789878750, 'Mumbai', 'IT', 'Marketing',
'Monthly', datetime.datetime(2016, 12, 7, 0, 0), datetime.datetime(2017, 11, 7, 0, 0), 'Active'), ('C1003', 'NewG
en', 'jobs@newgen.com', 8877643228, 'Bangalore', 'Manufacturing', 'Marketing', 'Yearly', datetime.datetime(2016,
2, 9, 0, 0), datetime.datetime(2017, 1, 9, 0, 0), 'Expired')]
4
[('COMPANYID', <class 'cx_Oracle.STRING'>, 5, 5, 0, 0, 0), ('COMPANYNAME', <class 'cx_Oracle.STRING'>, 50, 50, 0,
0, 0), ('EMAILID', <class 'cx_Oracle.STRING'>, 30, 30, 0, 0, 1), ('MOBILE', <class 'cx_Oracle.NUMBER'>, 11, 22, 1
0, 0, 1), ('CITY', <class 'cx_Oracle.STRING'>, 15, 15, 0, 0, 1), ('INDUSTRYTYPE', <class 'cx_Oracle.STRING'>, 20,
20, 0, 0, 1), ('FUNCTIONALAREA', <class 'cx_Oracle.STRING'>, 20, 20, 0, 0, 1), ('MEMBERSHIPPLAN', <class 'cx_Orac
le.STRING'>, 20, 20, 0, 0, 1), ('DATEOFSIGNUP', <class 'cx_Oracle.DATETIME'>, 23, 7, 0, 0, 1), ('DATEOFRENEWAL',
<class 'cx_Oracle.DATETIME'>, 23, 7, 0, 0, 1), ('RENEWALSTATUS', <class 'cx_Oracle.STRING'>, 10, 10, 0, 0, 1)]
>>>
```

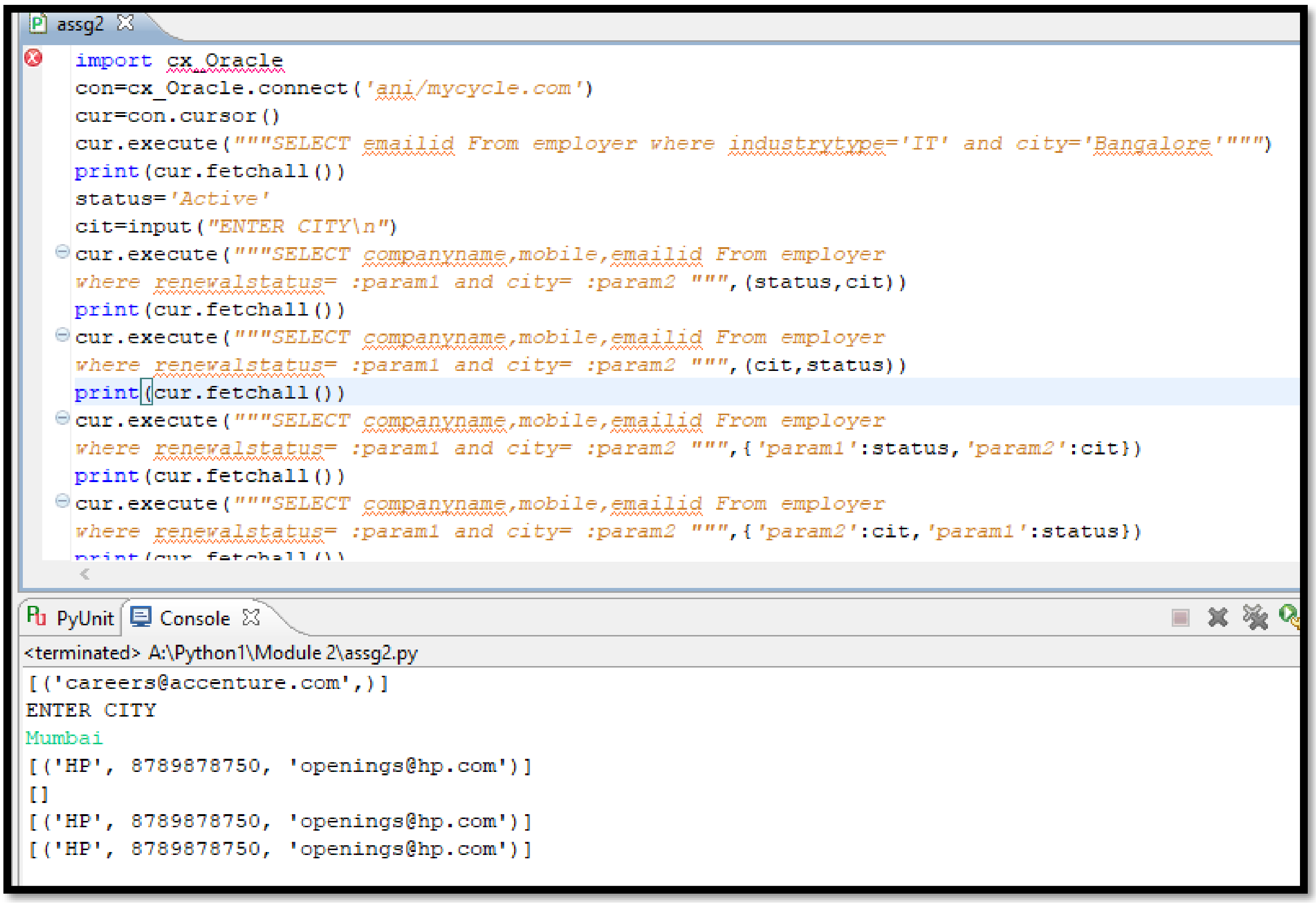
FP5.0 Module-3 Assignments 2

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 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?

SOURCE CODE AND OUTPUT

```
#
import cx_Oracle
con=cx_Oracle.connect('ani/mycycle.com')
cur=con.cursor()
cur.execute("""SELECT emailid From employer
 where industrytype='IT' and city='Bangalore'""")
print(cur.fetchall())
status='Active'
cit=input("ENTER CITY\n")
cur.execute("""SELECT companyname,mobile,emailid From employer
where renewalstatus= :param1 and city= :param2 """,(status,cit))
print(cur.fetchall())
cur.execute("""SELECT companyname,mobile,emailid From employer
where renewalstatus= :param1 and city= :param2 """,(cit,status))
print(cur.fetchall())
cur.execute("""SELECT companyname,mobile,emailid From employer
where renewalstatus= :param1 and city= :param2 """,{'param1':status,'param2':cit})
print(cur.fetchall())
cur.execute("""SELECT companyname,mobile,emailid From employer
where renewalstatus= :param1 and city= :param2 """,{'param2':cit,'param1':status})
print(cur.fetchall())
con.close()
#
```

We are still getting the same result as variables are being bound to the name.



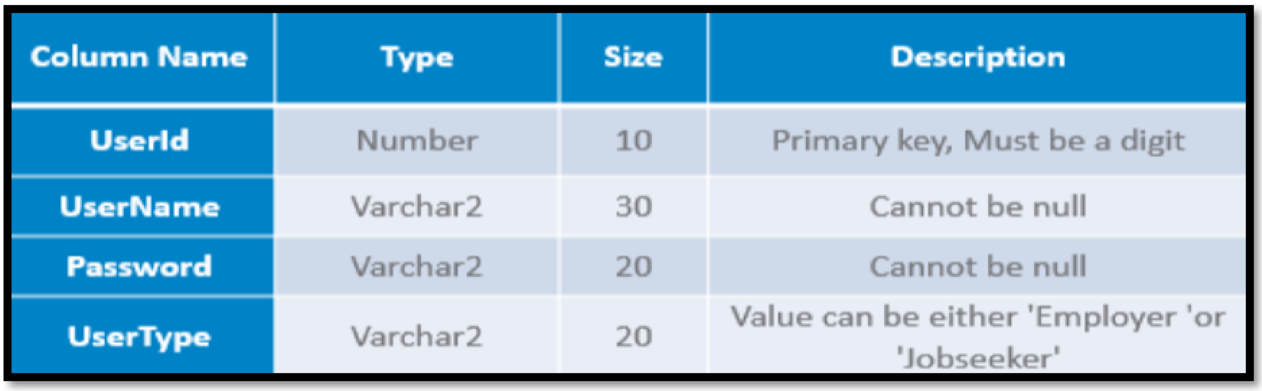
```
import cx_Oracle
con=cx_Oracle.connect('ani/mycycle.com')
cur=con.cursor()
cur.execute("""SELECT emailid From employer where industrytype='IT' and city='Bangalore'""")
print(cur.fetchall())
status='Active'
cit=input("ENTER CITY\n")
cur.execute("""SELECT companyname,mobile,emailid From employer
where renewalstatus= :param1 and city= :param2 """,(status,cit))
print(cur.fetchall())
cur.execute("""SELECT companyname,mobile,emailid From employer
where renewalstatus= :param1 and city= :param2 """,(cit,status))
print(cur.fetchall())
cur.execute("""SELECT companyname,mobile,emailid From employer
where renewalstatus= :param1 and city= :param2 """,{'param1':status,'param2':cit})
print(cur.fetchall())
cur.execute("""SELECT companyname,mobile,emailid From employer
where renewalstatus= :param1 and city= :param2 """,{'param2':cit,'param1':status})
```

```
<terminated> A:\Python1\Module 2\assg2.py
[('careers@accenture.com',)]
ENTER CITY
Mumbai
[('HP', 8789878750, 'openings@hp.com')]
[]
[('HP', 8789878750, 'openings@hp.com')]
[('HP', 8789878750, 'openings@hp.com')]
```

FP5.0 Module-3 Assignments 3

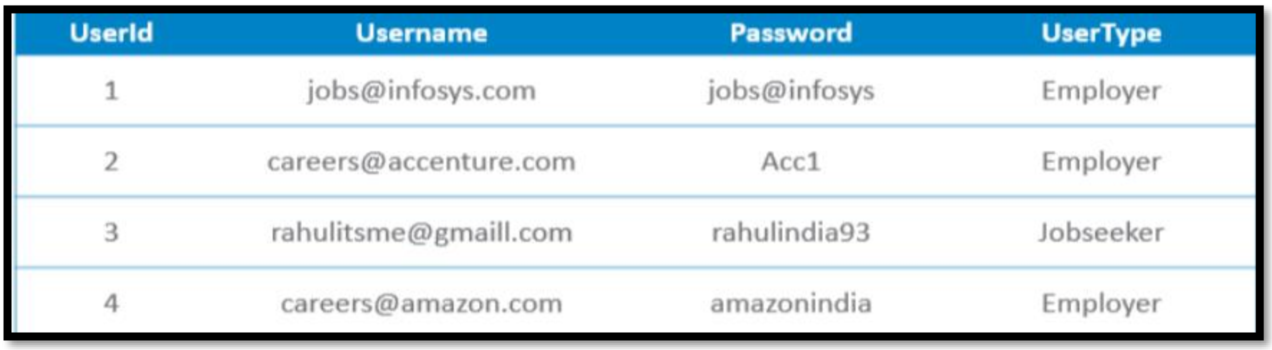
InfoTech Systems is creating an online application for automating the task of job search between employer and job
seekers.

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 into Users table 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.

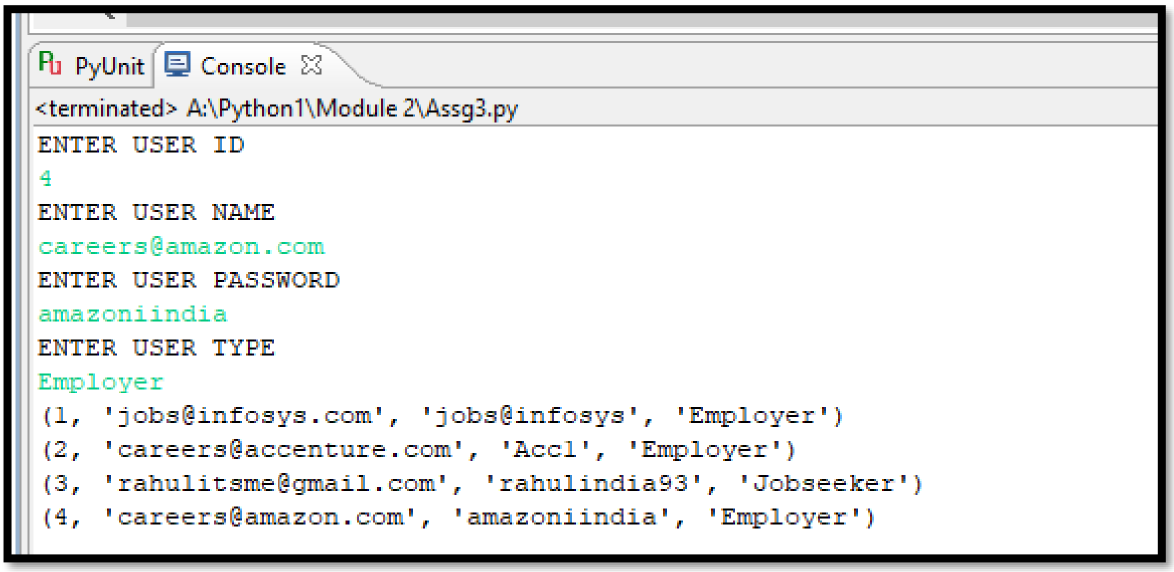
SOURCE CODE AND OUTPUT

```
#

import cx_Oracle
con=cx_Oracle.connect('ani/mycycle.com')
cur=con.cursor()
cur.execute("""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("""insert into users
values(1,'jobs@infosys.com','jobs@infosys','Employer')""")
uid1=2
uname1='careers@accenture.com'
upwd1='Acc1'
utype1='Employer'
cur.execute("""insert into users values(:A,:B,:C,:D)""",(uid1,uname1,upwd1,utype1))
uid2=3
uname2='rahulitsme@gmail.com'
upwd2='rahulindia93'
utype2='Jobseeker'
cur.execute("""insert into users
values(:par1,:par2,:par3,:par4)\n""",{'par1':uid2,'par2':uname2,'par3':upwd2,'par4':utyp
e2})
uid3=int(input("ENTER USER ID\n"))
uname3=input("ENTER USER NAME\n")
upwd3=input("ENTER USER PASSWORD\n")
utype3=input("ENTER USER TYPE\n")
cur.execute("""insert into users
values(:var1,:var2,:var3,:var4)""",{'var1':uid3,'var2':uname3,'var3':upwd3,'var4':utype3})
cur.execute("SELECT * From users")
data=cur.fetchall()
for line in data:
    print(line)
con.commit()
con.close()
```

#

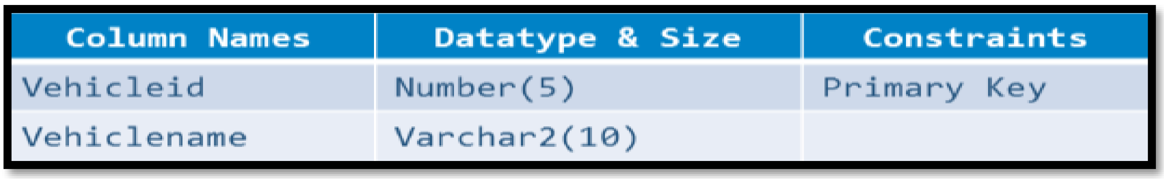


```
<terminated> A:\Python1\Module 2\Assg3.py
ENTER USER ID
4
ENTER USER NAME
careers@amazon.com
ENTER USER PASSWORD
amazoniindia
ENTER USER TYPE
Employer
(1, 'jobs@infosys.com', 'jobs@infosys', 'Employer')
(2, 'careers@accenture.com', 'Acc1', 'Employer')
(3, 'rahulitsme@gmail.com', 'rahulindia93', 'Jobseeker')
(4, 'careers@amazon.com', 'amazoniindia', 'Employer')
```

FP5.0 Module-3 Assignments 4

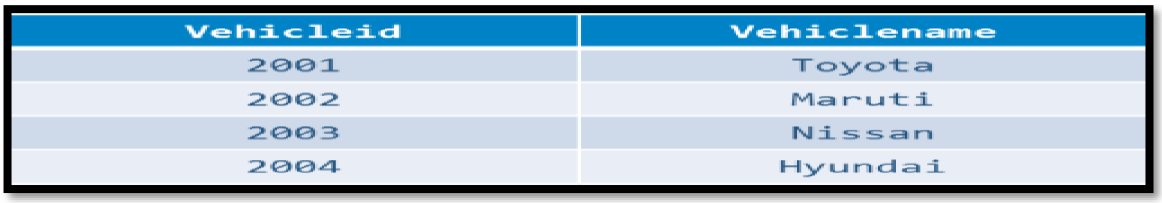
Bloom Technology wants to maintain their employee's vehicle details to make parking facility flexible to the employees.

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



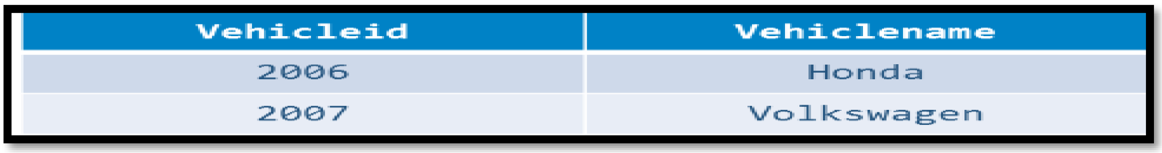
Column Names	Datatype & Size	Constraints
Vehicleid	Number(5)	Primary Key
Vehiclename	Varchar2(10)	

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 variable (use executemany() function)

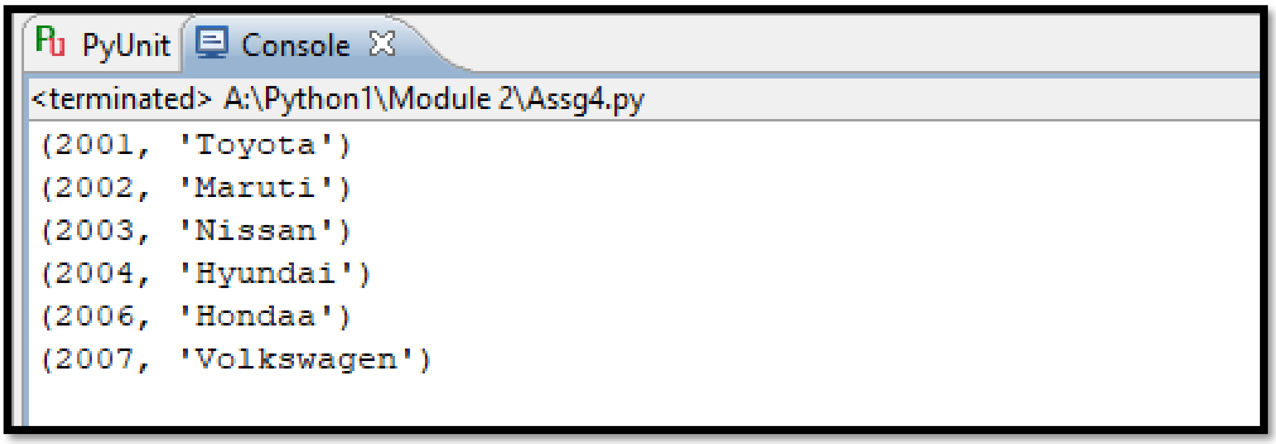


Vehicleid	Vehiclename
2006	Honda
2007	Volkswagen

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

```
#
import cx_Oracle
con=cx_Oracle.connect('ani/mycycle.com')
cur=con.cursor()
cur.execute("""Create Table vehicle(
          vehicleid number(5) primary key,
          vehiclename varchar2(10) not null)""")
vid=2001
cur.executemany("""insert into vehicle values(:var1,:var2)""",
             [(vid,'Toyota'),(vid+1,'Maruti'),(vid+2,'Nissan'),(vid+3,'Hyundai')])
cur.executemany("""insert into vehicle values(:par1,:par2)""",
             [{'par1':vid+5,'par2':'Hondaa'},
              {'par1':vid+6,'par2':'Volkswagen'},
              ])
cur.execute("SELECT * From vehicle")
data=cur.fetchall()
for line in data:
   print(line)
con.commit()
con.close()
#
```

# OUTPUT



```
<terminated> A:\Python1\Module 2\Assg4.py
(2001, 'Toyota')
(2002, 'Maruti')
(2003, 'Nissan')
(2004, 'Hyundai')
(2006, 'Hondaa')
(2007, 'Volkswagen')
```

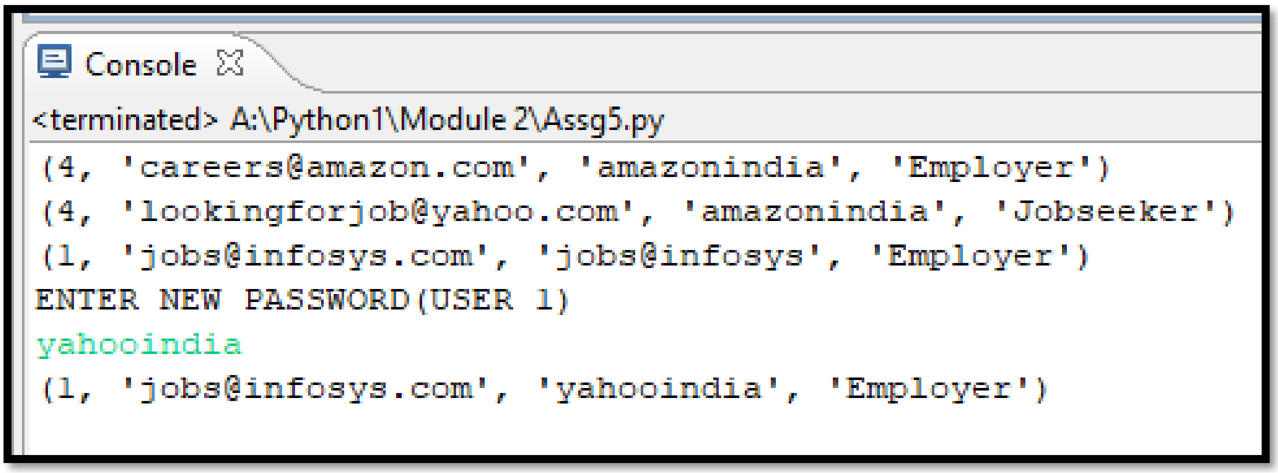
FP5.0 Module-3 Assignments 5

Refer to the table 'users' created earlier. The existing table data for “users” table is given below:
1.Modify the username and usertype of the user with userid = 4 with the following values:
•Username: lookingforjob@yahoo.com
•UserType: Jobseeker
Fetch 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 observe the value of 'password' of the user with 'userid = 1' before and after 'update' operation.

SOURCE CODE AND OUTPUT

```
#
import cx_Oracle
con=cx_Oracle.connect('ani/mycycle.com')
cur=con.cursor()
cur.execute("SELECT * From user1 where userid=4")
data=cur.fetchall()
for line in data:
    print(line)
cur.execute("UPDATE user1 set username= :var1 , usertype= :var2 where
userid=4",{'var1':"lookingforjob@yahoo.com",'var2':"Jobseeker"})
cur.execute("SELECT * From user1 where userid=4")
data=cur.fetchall()
for line in data:
    print(line)
cur.execute("SELECT * From user1 where userid=1")
data=cur.fetchall()
```

```
for line in data:
    print(line)
psw=input("ENTER NEW PASSWORD(USER 1)\n")
cur.execute("UPDATE user1 set password= :par1 where userid=1",{'par1':psw})
cur.execute("SELECT * From user1 where userid=1")
data=cur.fetchall()
for line in data:
    print(line)
con.commit()
con.close()
#
```



```
<terminated> A:\Python1\Module 2\Assg5.py
(4, 'careers@amazon.com', 'amazonindia', 'Employer')
(4, 'lookingforjob@yahoo.com', 'amazonindia', 'Jobseeker')
(1, 'jobs@infosys.com', 'jobs@infosys', 'Employer')
ENTER NEW PASSWORD(USER 1)
yahooindia
(1, 'jobs@infosys.com', 'yahooindia', 'Employer')
```

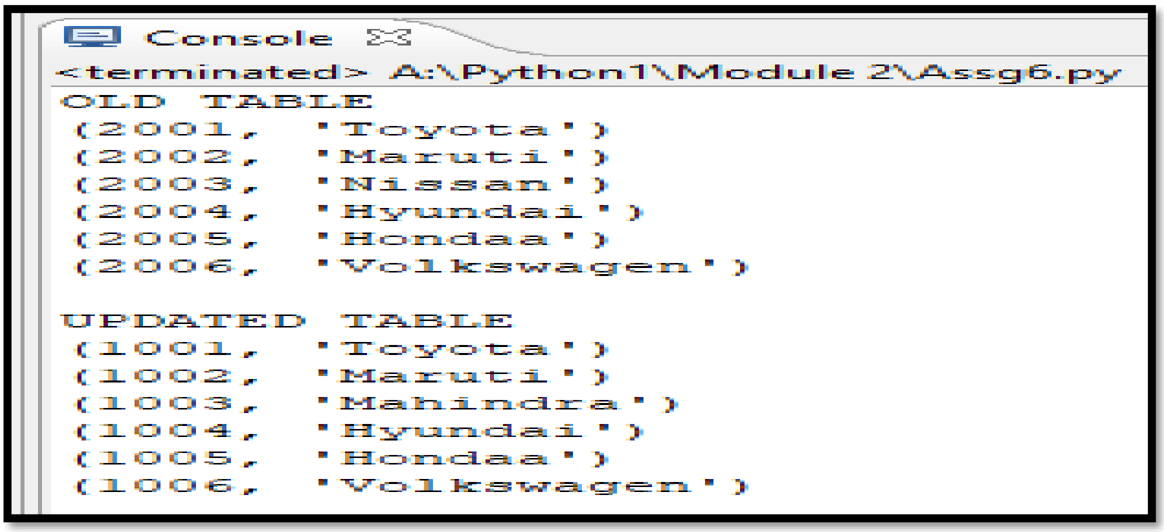
FP5.0 Module-3 Assignments 6

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.

SOURCE CODE AND OUTPUT

```
#
import cx_Oracle
con=cx_Oracle.connect('ani/mycycle.com')
cur=con.cursor()
print("OLD TABLE")
cur.execute("SELECT * From vehicle")
data=cur.fetchall()
for line in data:
    print(line)
oldid=2001
newid=1001
```

```
for line in data:
    cur.execute("update vehicle set vehicleid= :var1 where vehicleid=
:var2",{'var1':newid,'var2':oldid})
    oldid+=1
    newid+=1
cur.execute("update vehicle set vehiclename='Mahindra' where vehicleid=1003")
print("\nUPDATED TABLE")
cur.execute("SELECT * From vehicle")
data=cur.fetchall()
for line in data:
    print(line)
con.commit()
con.close()
#
```



```
Console
<terminated> A:\Python1\Module 2\Assg6.py
OLD TABLE
 (2001, 'Toyota')
 (2002, 'Maruti')
 (2003, 'Nissan')
 (2004, 'Hyundai')
 (2005, 'Hondaa')
 (2006, 'Volkswagen')

UPDATED TABLE
 (1001, 'Toyota')
 (1002, 'Maruti')
 (1003, 'Mahindra')
 (1004, 'Hyundai')
 (1005, 'Hondaa')
 (1006, 'Volkswagen')
```

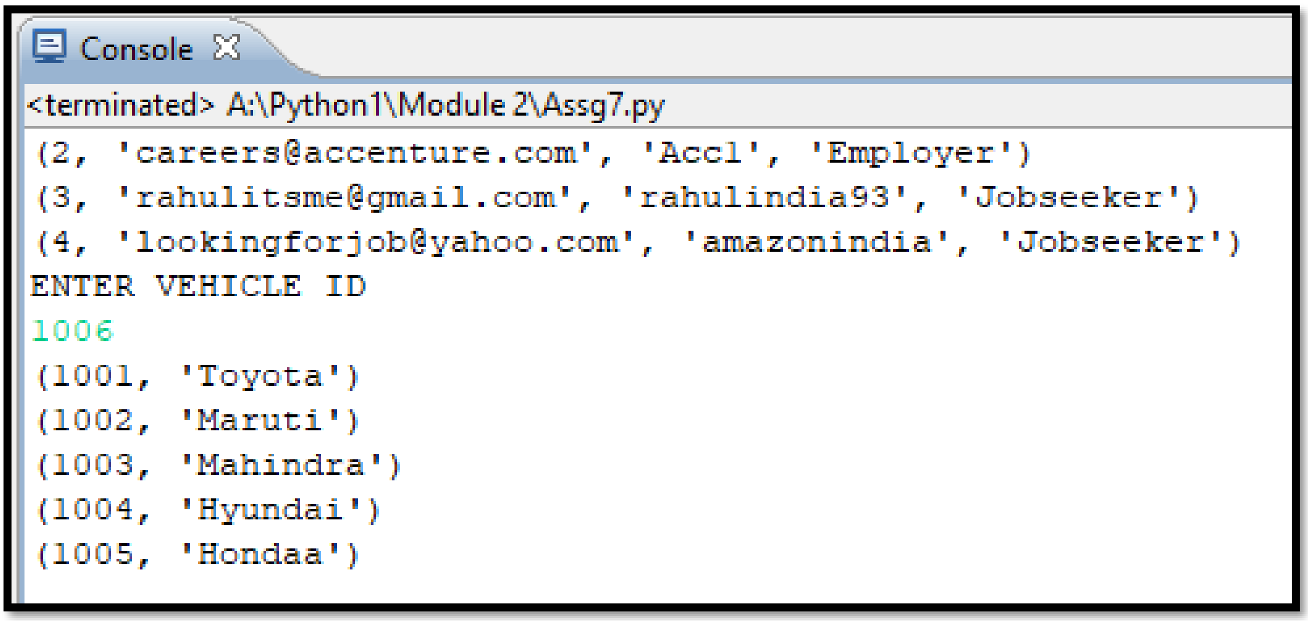
FP5.0 Module-3 Assignments 7

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.

SOURCE CODE AND OUTPUT

```
#
import cx_Oracle
con=cx_Oracle.connect('ani/mycycle.com')
cur=con.cursor()
cur.execute("Delete From user1 where userid=1")
cur.execute("Select * from user1")
data=cur.fetchall()
for line in data:
    print(line)
vid=int(input("ENTER VEHICLE ID\n"))
cur.execute("Delete From vehicle where vehicleid= :var",(vid,))
cur.execute("Select * from vehicle")
```

```
data=cur.fetchall()
for line in data:
    print(line)
con.commit()
con.close()
#
```



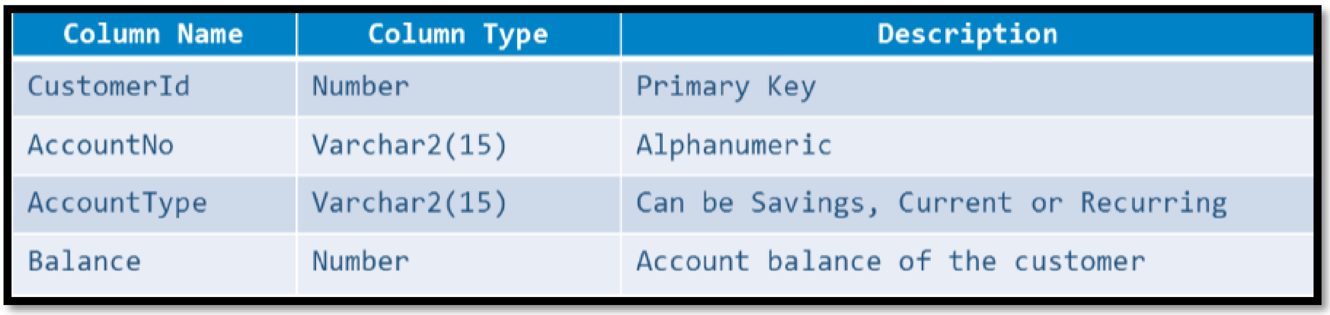
Console

```
<terminated> A:\Python1\Module 2\Assg7.py
(2, 'careers@accenture.com', 'Accl', 'Employer')
(3, 'rahulitsme@gmail.com', 'rahulindia93', 'Jobseeker')
(4, 'lookingforjob@yahoo.com', 'amazonindia', 'Jobseeker')
ENTER VEHICLE ID
1006
(1001, 'Toyota')
(1002, 'Maruti')
(1003, 'Mahindra')
(1004, 'Hyundai')
(1005, 'Hondaa')
```

FP5.0 Module-3 Assignments 8

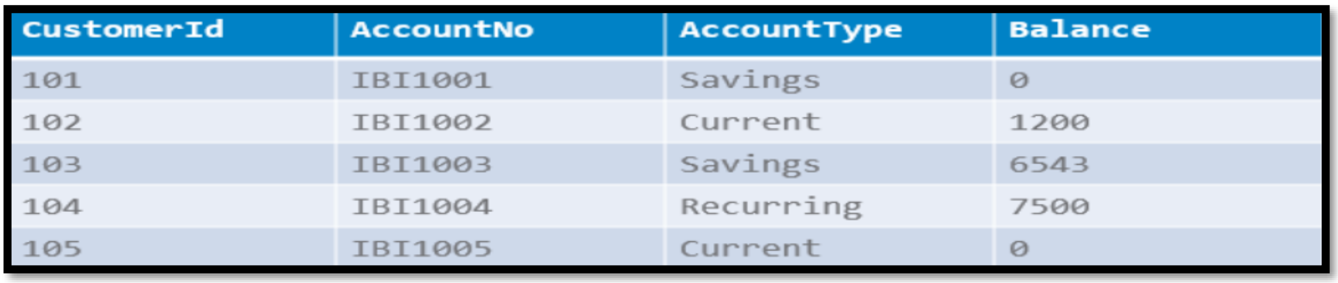
Consider a scenario from a State Banking organization. The account table is created to store the account details of a customer (Assume every customer can have only one account). Use cx_Oracle module to implement the following requirements from Python code.(Do not execute the queries in database directly)

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

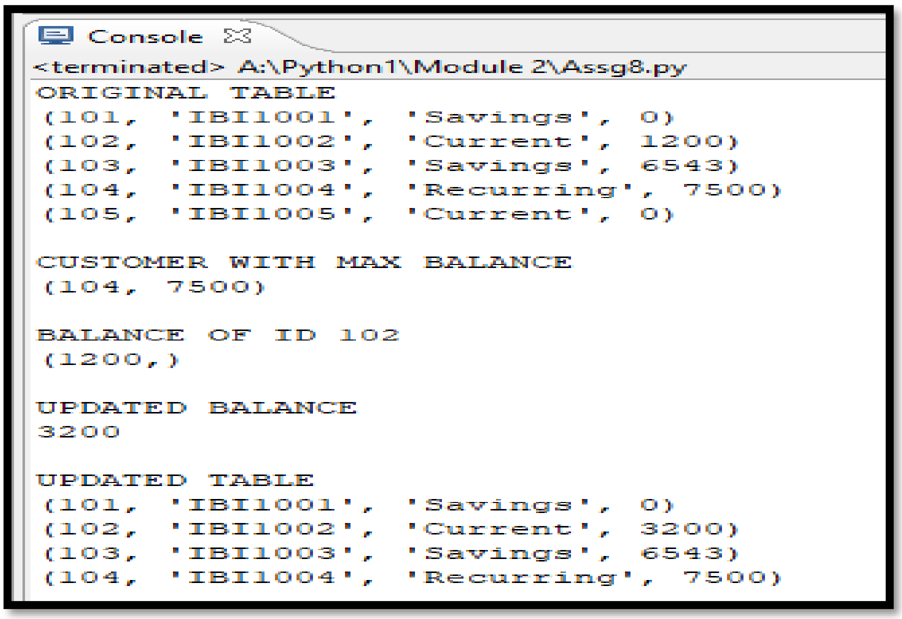
SOURCE CODE AND OUTPUT

```
#

import cx_Oracle
con=cx_Oracle.connect('ani/mycycle.com')
cur=con.cursor()
cur.execute("""Create Table account(
          customerid number(10) primary key,
          accountno varchar2(15) not null,
          accounttype varchar2(15) CHECK (accounttype IN
('Savings','Current','Recurring')),
          balance number(10) not null
          )""")
cid=101
acno='IBI100'
cur.executemany("""insert into account values(:par1,:par2,:par3,:par4)""",
             [{'par1':cid,'par2':acno+'1','par3':'Savings','par4':0},
              {'par1':cid+1,'par2':acno+'2','par3':'Current','par4':1200},
              {'par1':cid+2,'par2':acno+'3','par3':'Savings','par4':6543},
              {'par1':cid+3,'par2':acno+'4','par3':'Recurring','par4':7500},
              {'par1':cid+4,'par2':acno+'5','par3':'Current','par4':0}
              ])
print("ORIGINAL TABLE")
cur.execute("SELECT * from account")
data=cur.fetchall()
for line in data:
   print(line)
print("\nCUSTOMER WITH MAX BALANCE")
cur.execute("select customerid,balance from account where balance=(select
max(balance) from account)")
data=cur.fetchall()
for line in data:
   print(line)
print("\nBALANCE OF ID 102")
cur.execute("select balance from account where customerid=102")
data=cur.fetchall()
for acct_bal in data:
   print(acct_bal)
   acct_bal=acct_bal[0]+2000
cur.execute("update account set balance=:var where customerid=102",(acct_bal,))
print("\nUPDATED BALANCE")
cur.execute("select balance from account where customerid=102")
```

```
data=cur.fetchall()
for line in data:
    print(line[0])
cur.execute("DELETE FROM account where balance=0 and accounttype='Current'")
print("\nUPDATED TABLE")
cur.execute("SELECT * from account")
data=cur.fetchall()
for line in data:
    print(line)
con.commit()
con.close()
```

#



Console

```
<terminated> A:\Python1\Module 2\Assg8.py
ORIGINAL TABLE
(101, 'IBI1001', 'Savings', 0)
(102, 'IBI1002', 'Current', 1200)
(103, 'IBI1003', 'Savings', 6543)
(104, 'IBI1004', 'Recurring', 7500)
(105, 'IBI1005', 'Current', 0)

CUSTOMER WITH MAX BALANCE
(104, 7500)

BALANCE OF ID 102
(1200,)

UPDATED BALANCE
3200

UPDATED TABLE
(101, 'IBI1001', 'Savings', 0)
(102, 'IBI1002', 'Current', 3200)
(103, 'IBI1003', 'Savings', 6543)
(104, 'IBI1004', 'Recurring', 7500)
```

FP5.0 Module-3 Assignments 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.
•Use exception handling to handle the exception appropriately. Display the error code and message.
•Try to give incorrect username for connection string and observe the error code and message.
•Provide a wrong table name while writing the query and observe the error message.

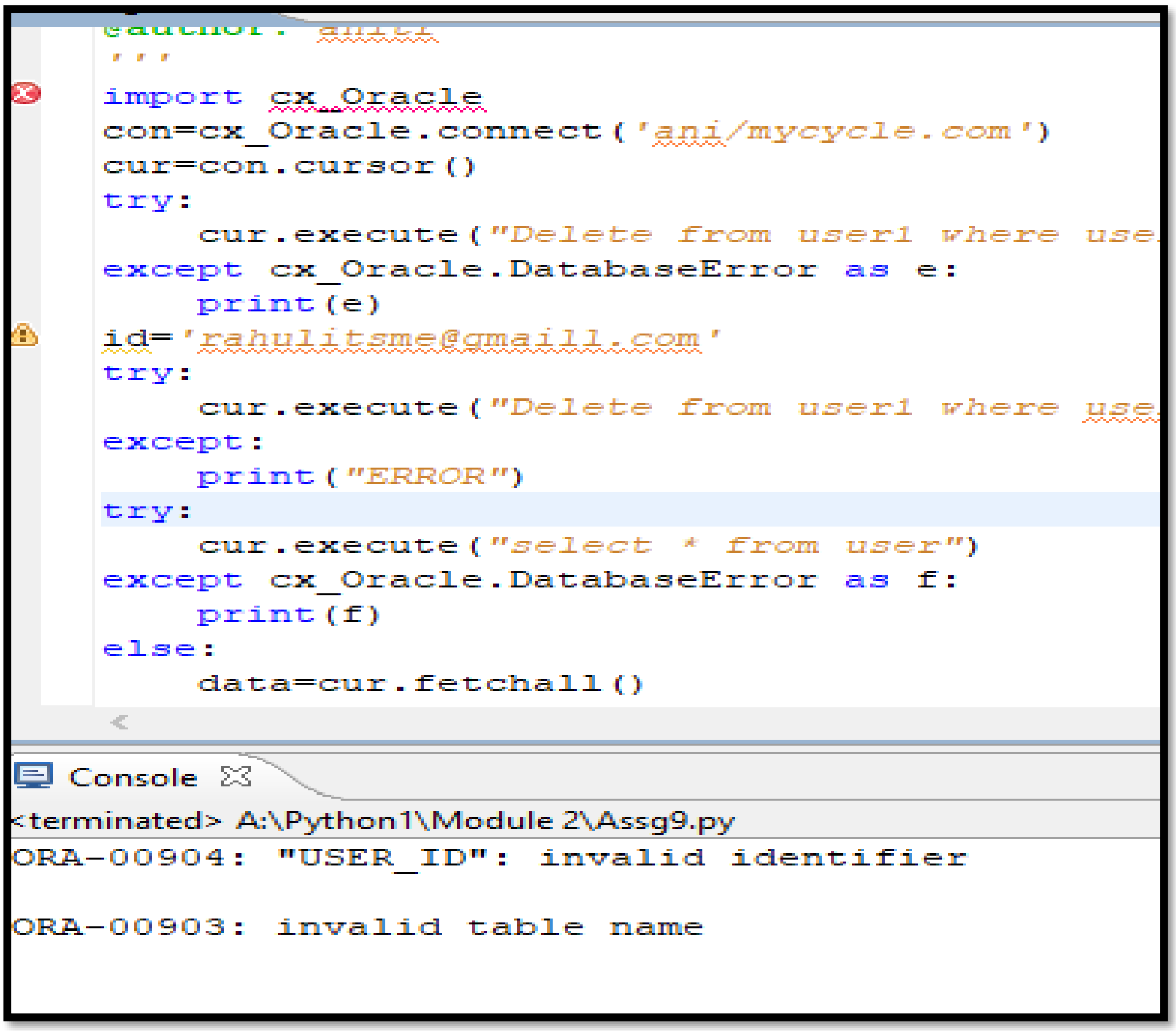
<u>SOURCE CODE AND OUTPUT</u>

#

```
import cx_Oracle
con=cx_Oracle.connect('ani/mycycle.com')
cur=con.cursor()
try:
    cur.execute("Delete from user1 where user_id=2")
except cx_Oracle.DatabaseError as e:
    print(e)
```

```
id='rahulitsme@gmaill.com'
try:
    cur.execute("Delete from user1 where username=:var",{'var':id})
except:
    print("ERROR")
try:
    cur.execute("select * from user")
except cx_Oracle.DatabaseError as f:
    print(f)
else:
    data=cur.fetchall()
    for line in data:
        print(line)
    con.commit()
con.close()
```

#



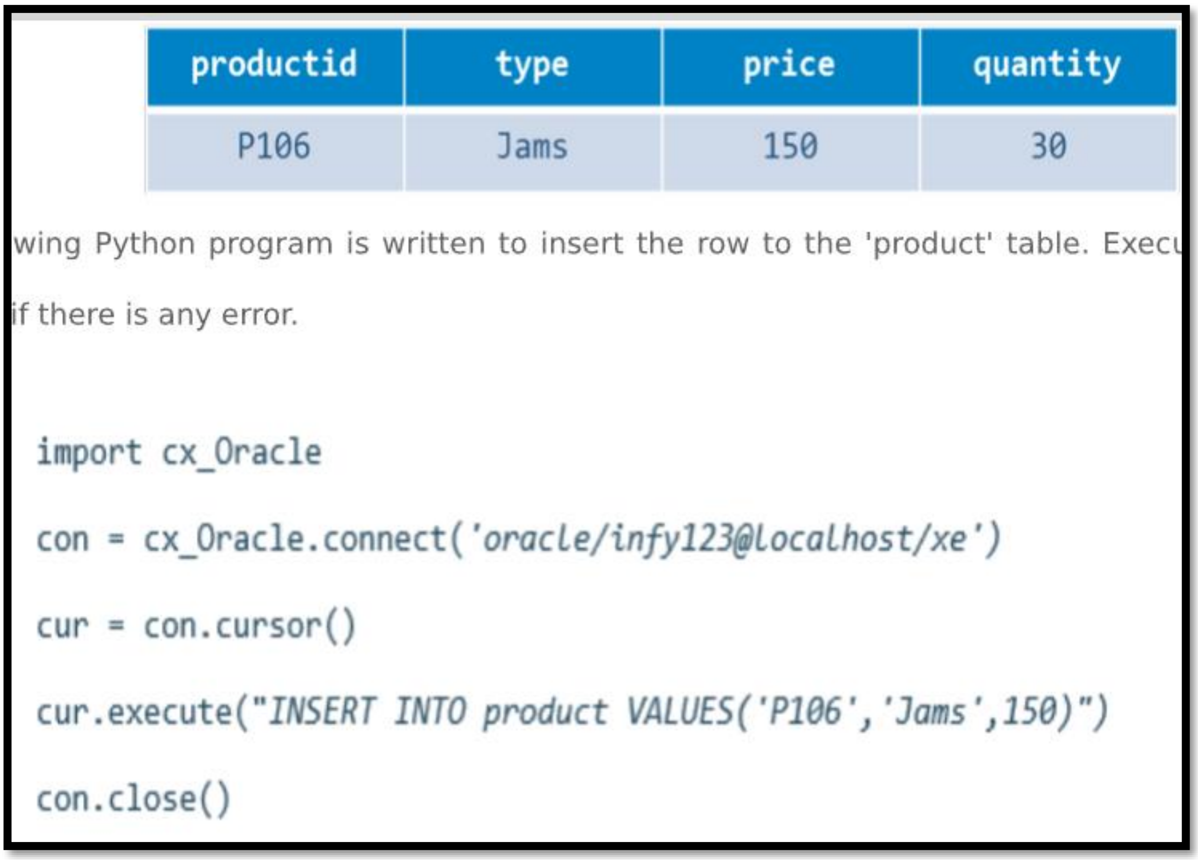
```
@author: aniti
'''
import cx_Oracle
con=cx_Oracle.connect('ani/mycycle.com')
cur=con.cursor()
try:
    cur.execute("Delete from user1 where use
except cx_Oracle.DatabaseError as e:
    print(e)
id='rahulitsme@gmaill.com'
try:
    cur.execute("Delete from user1 where use
except:
    print("ERROR")
try:
    cur.execute("select * from user")
except cx_Oracle.DatabaseError as f:
    print(f)
else:
    data=cur.fetchall()
```

```
<terminated> A:\Python1\Module 2\Assg9.py
ORA-00904: "USER_ID": invalid identifier

ORA-00903: invalid table name
```

FP5.0 Module-3 Assignments 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.
Use exception handling to handle the error (if any) and display error message appropriately



productid	type	price	quantity
P106	Jams	150	30

wing Python program is written to insert the row to the 'product' table. Execu

if there is any error.

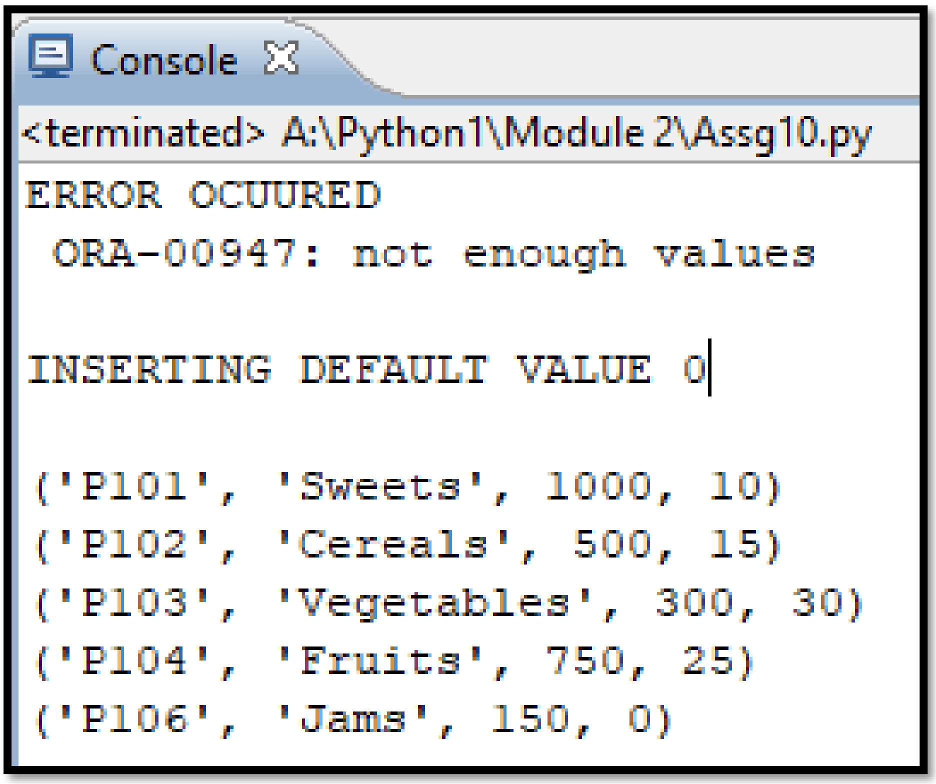
```
import cx_Oracle
con = cx_Oracle.connect('oracle/infy123@localhost/xe')
cur = con.cursor()
cur.execute("INSERT INTO product VALUES('P106','Jams',150)")
con.close()
```

SOURCE CODE AND OUTPUT

```
#

import cx_Oracle
con=cx_Oracle.connect('ani/mycycle.com')
cur=con.cursor()
try:
    cur.execute("INSERT INTO product VALUES('P106','Jams',150)")
except cx_Oracle.DatabaseError as e:
    print("ERROR OCUURED\n",e,"\nINSERTING DEFAULT VALUE 0\n")
    cur.execute("INSERT INTO product VALUES('P106','Jams',150,0)")
cur.execute("Select * from product")
data=cur.fetchall()
for line in data:
    print(line)
con.close()
```

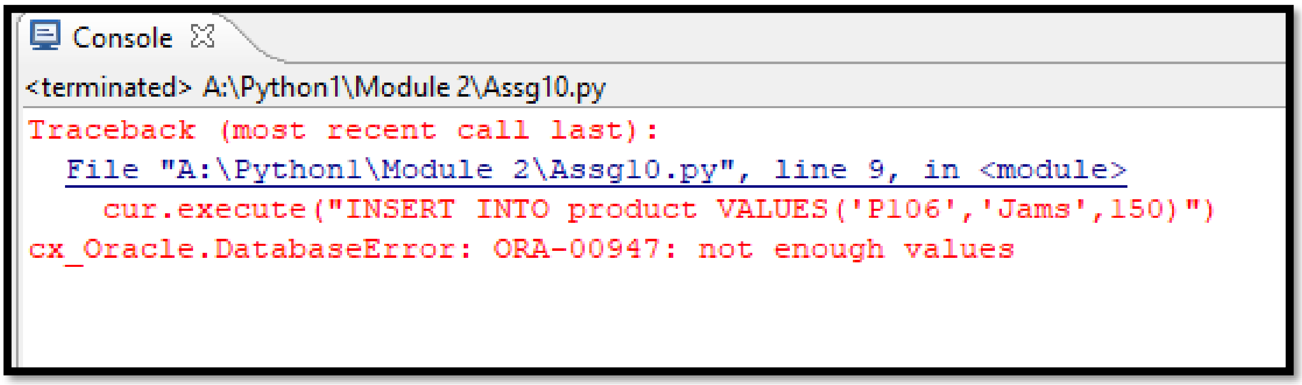
#



```
<terminated> A:\Python1\Module 2\Assg10.py
ERROR OCUURED
 ORA-00947: not enough values

INSERTING DEFAULT VALUE 0

('P101', 'Sweets', 1000, 10)
('P102', 'Cereals', 500, 15)
('P103', 'Vegetables', 300, 30)
('P104', 'Fruits', 750, 25)
('P106', 'Jams', 150, 0)
```



```
<terminated> A:\Python1\Module 2\Assg10.py
Traceback (most recent call last):
  File "A:\Python1\Module 2\Assg10.py", line 9, in <module>
    cur.execute("INSERT INTO product VALUES('P106','Jams',150)")
cx_Oracle.DatabaseError: ORA-00947: not enough values
```