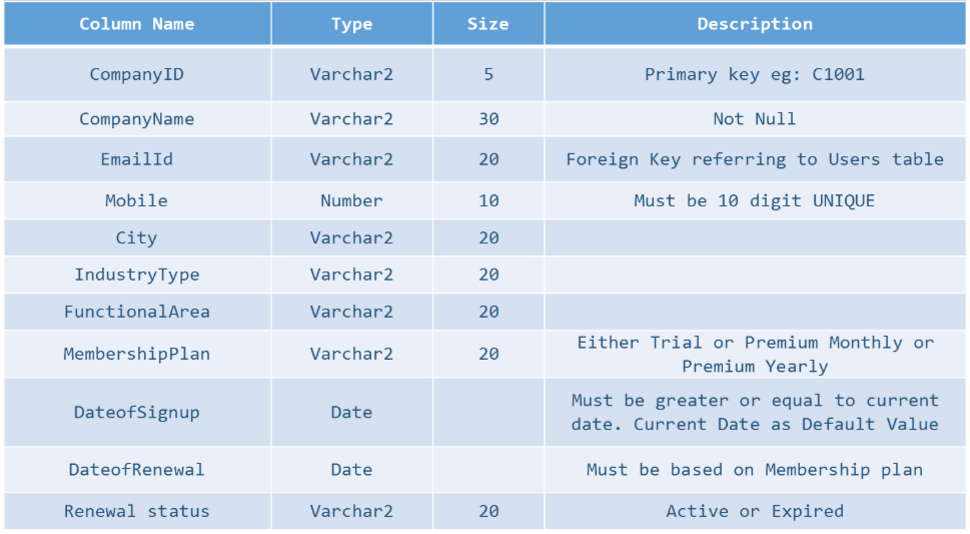
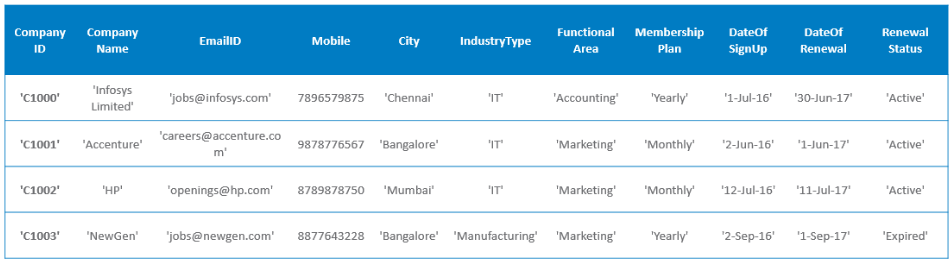
**ASSIGNMENT-1**

**Question:-**

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





**Ques1:-** 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

**Solution:-**

import cx\_Oracle

con = cx\_Oracle.connect(*"SYSTEM/user"*)

cur = con.cursor()

statement = *"select \* from employer"*

cur.execute(statement)

res = cur.fetchall()

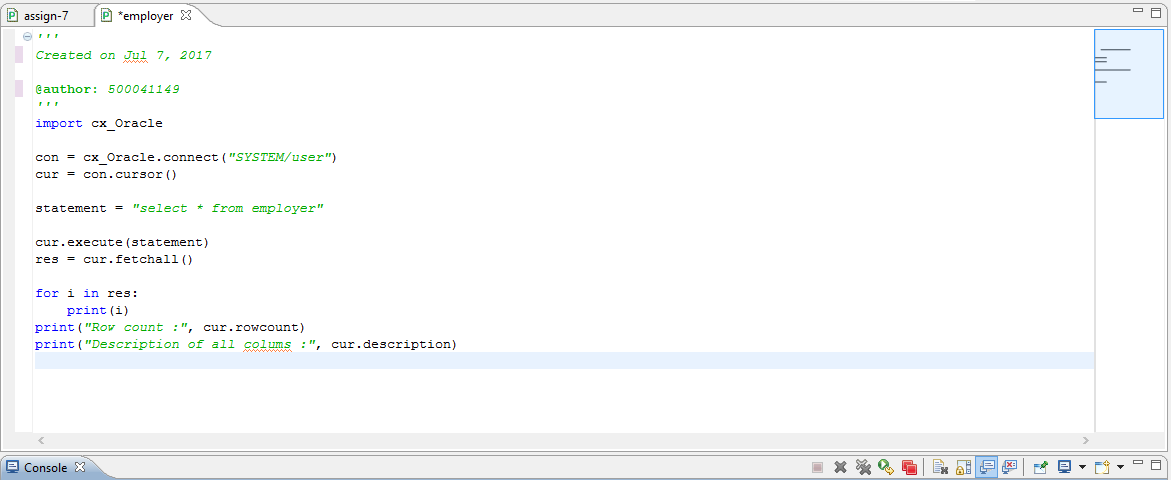
for i in res:

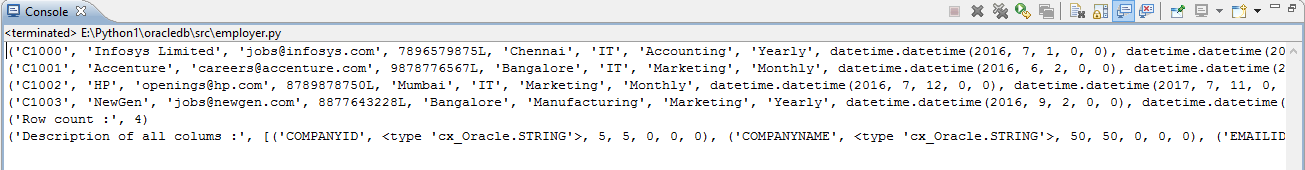
print(i)

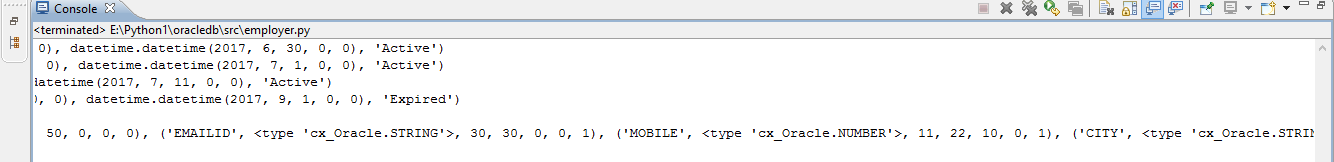
print(*"Row count :"*, cur.rowcount)

print(*"Description of all colums :"*, cur.description)

**Output:-**

****

****

****

**ASSIGNMENT-5**

**Question:-**

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@gmail.com | rahulindia93 | jobseeker |
| 4 | Careers@amazon.com | amaozonindia | employer |

**Ques1:-**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.

**ASSIGNMENT-6**

**Question:-**

**Ques1:-**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.

**Solution:-**

import cx\_Oracle

con = cx\_Oracle.connect(*"SYSTEM/user"*)

cur = con.cursor()

cur.execute(*"select \* from vechile"*)

res = cur.fetchall()

print(res)

a = (cur.rowcount)

v\_idp=2001

v\_idc=1001

for i in range(0,6):

cur.execute(*"update vechile set vechileid=:1 where vechileid=:2"*,(v\_idc,v\_idp))

v\_idp=v\_idp+1

v\_idc=v\_idc+1

cur.execute(*"select \* from vechile"*)

res = cur.fetchall()

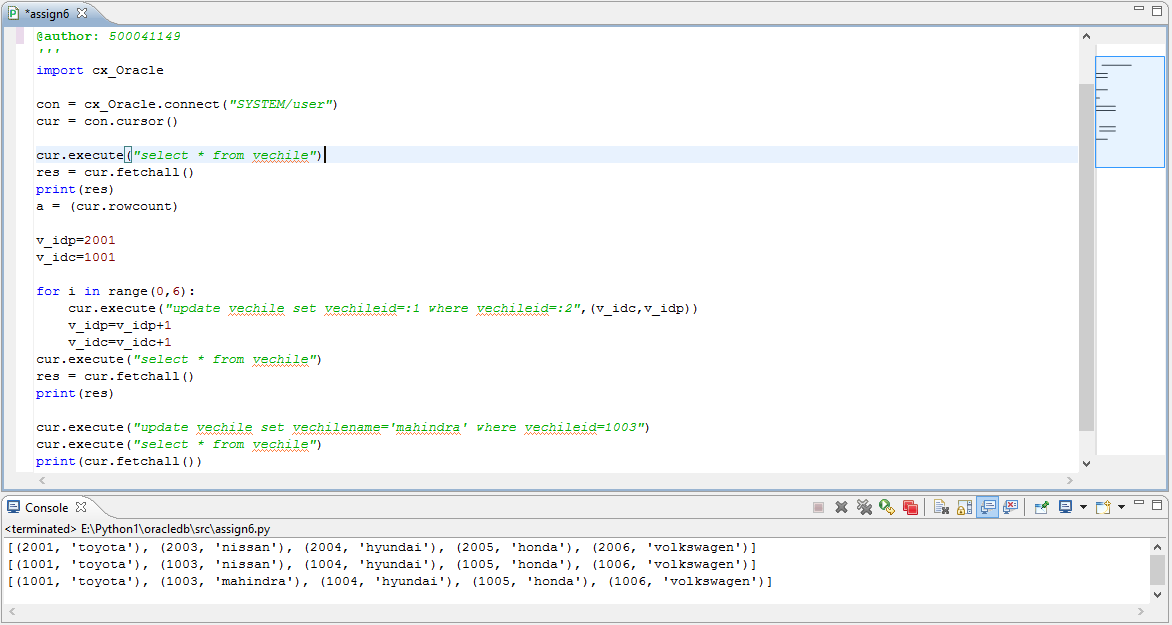
print(res)

cur.execute(*"update vechile set vechilename='mahindra' where vechileid=1003"*)

cur.execute(*"select \* from vechile"*)

print(cur.fetchall())

**Output:-**



**ASSIGNMENT-7**

**Question:-**

**Ques1:-**Consider 'users' table. Delete the record of user with userid = 1.

**Solution:-**

import cx\_Oracle

con = cx\_Oracle.connect(*"SYSTEM/user"*)

cur = con.cursor()

cur.execute(*"select \* from users"*)

res = cur.fetchall()

for i in res:

print(i)

cur.execute(*"delete from users where userid=1"*)

cur.execute(*"select \* from users"*)

print

print(*"after deletion:"*)

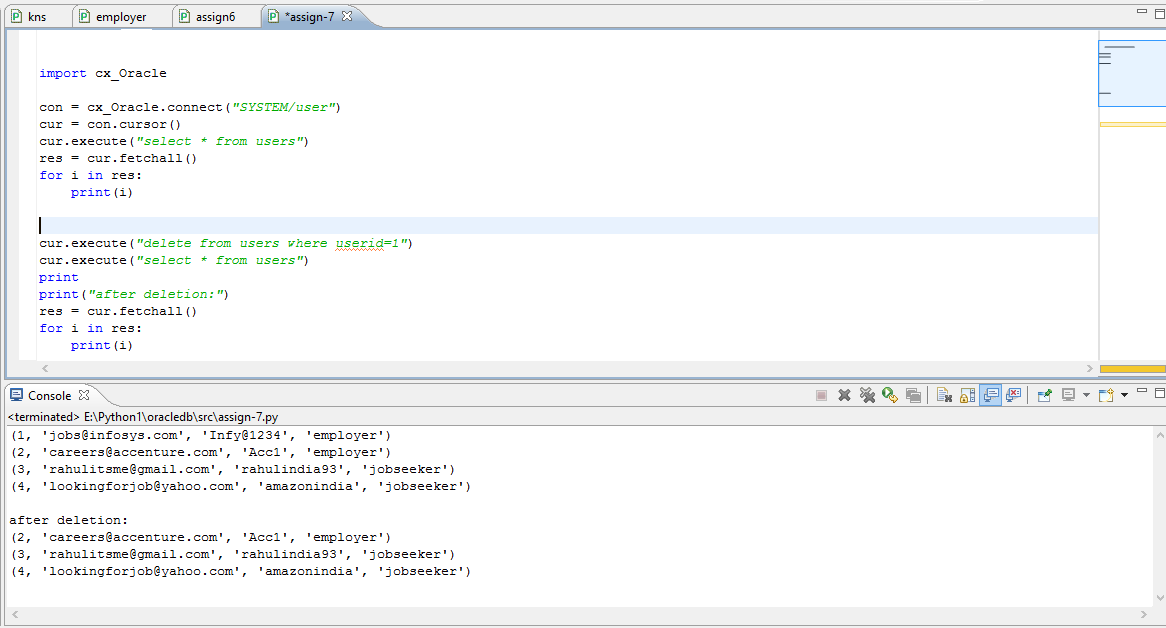
res = cur.fetchall()

for i in res:

print(i)

con.close()

**Output:-**



**Ques2:-**Delete a record from 'Vehicle' table using named bind variables. Accept VehicleId as an input from the user.

**Solution:-**

import cx\_Oracle

con = cx\_Oracle.connect(*"SYSTEM/user"*)

cur = con.cursor()

cur.execute(*"select \* from vechile"*)

print(cur.fetchall())

del\_id = int(raw\_input(*"enter the vechileid which you want to delete"*))

cur.execute(*"delete from vechile where vechileid=:v\_id"*,{*'v\_id'*:del\_id})

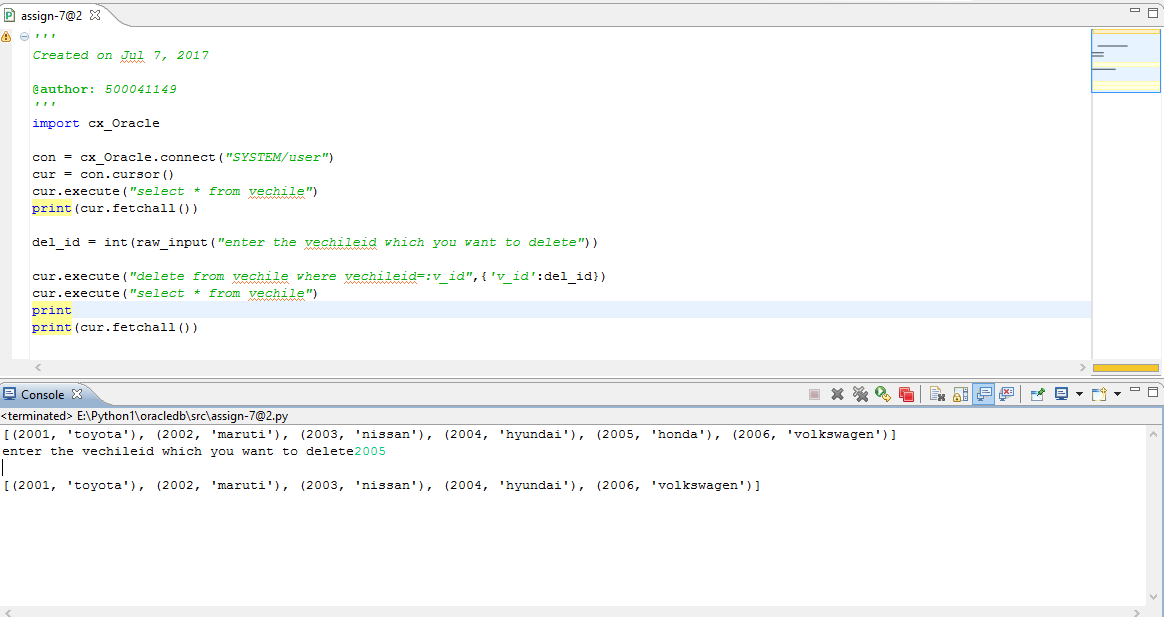
cur.execute(*"select \* from vechile"*)

print

print(cur.fetchall())

con.commit()

**Output:-**



**ASSIGNMENT-8**

**Question:-**

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)

**Ques1:-** 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 |

**Solution:-**

import cx\_Oracle

con = cx\_Oracle.connect(*"SYSTEM/user"*)

cur = con.cursor()

cur.execute(*'''create table account(*

*customerId number PRIMARY KEY,*

*accountNo varchar2(15),*

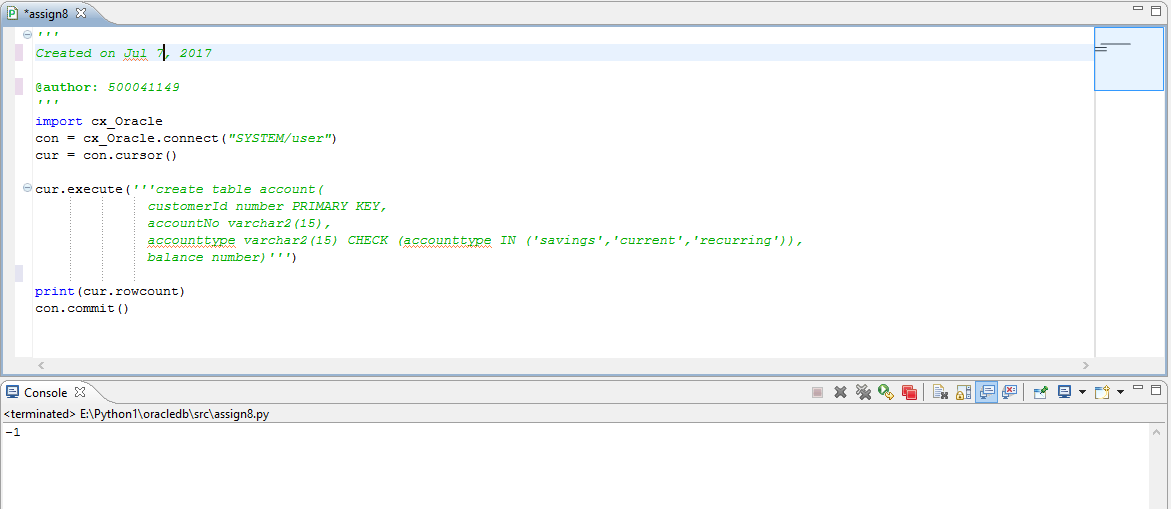
*accounttype varchar2(15) CHECK (accounttype IN ('savings','current','recurring')),*

*balance number)'''*)

print(cur.rowcount)

con.commit()

**Output:-**



**Ques2:-** 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 |

**Solution:-**

import cx\_Oracle

con = cx\_Oracle.connect(*"SYSTEM/user"*)

cur = con.cursor()

cur.execute(*"insert into account values(101,'IBI1001','savings',0)"*)

cur.execute(*"insert into account values(102,'IBI1002','current',1200)"*)

cur.execute(*"insert into account values(103,'IBI1003','savings',6543)"*)

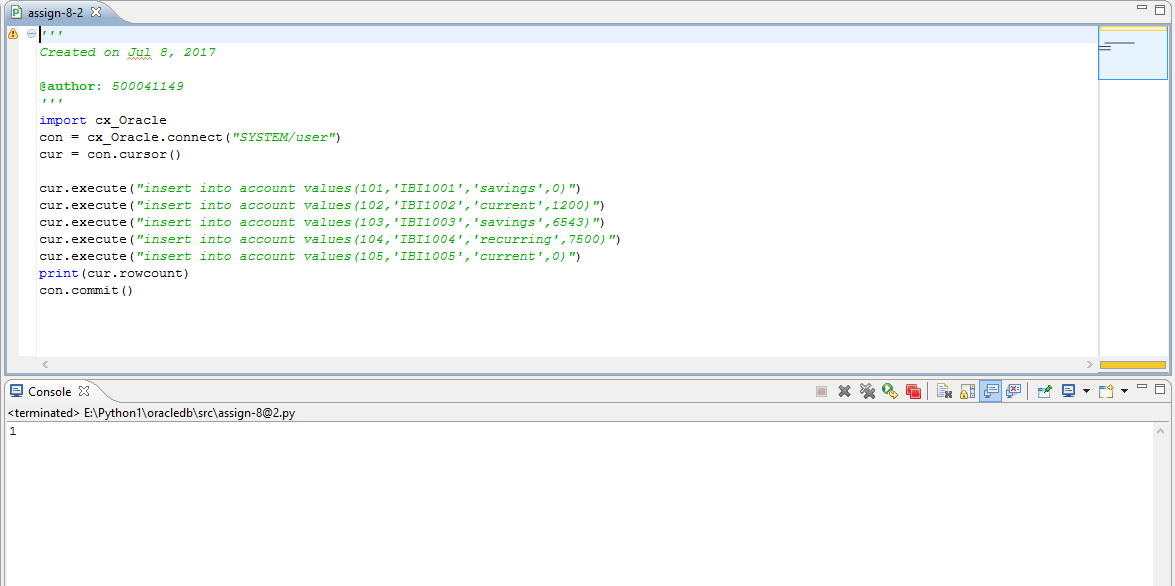
cur.execute(*"insert into account values(104,'IBI1004','recurring',7500)"*)

cur.execute(*"insert into account values(105,'IBI1005','current',0)"*)

print(cur.rowcount)

con.commit()

**Output:-**



**Ques3:-** Display the customer id and account balance of the customer with maximum account balance.

**Solution:-**

import cx\_Oracle

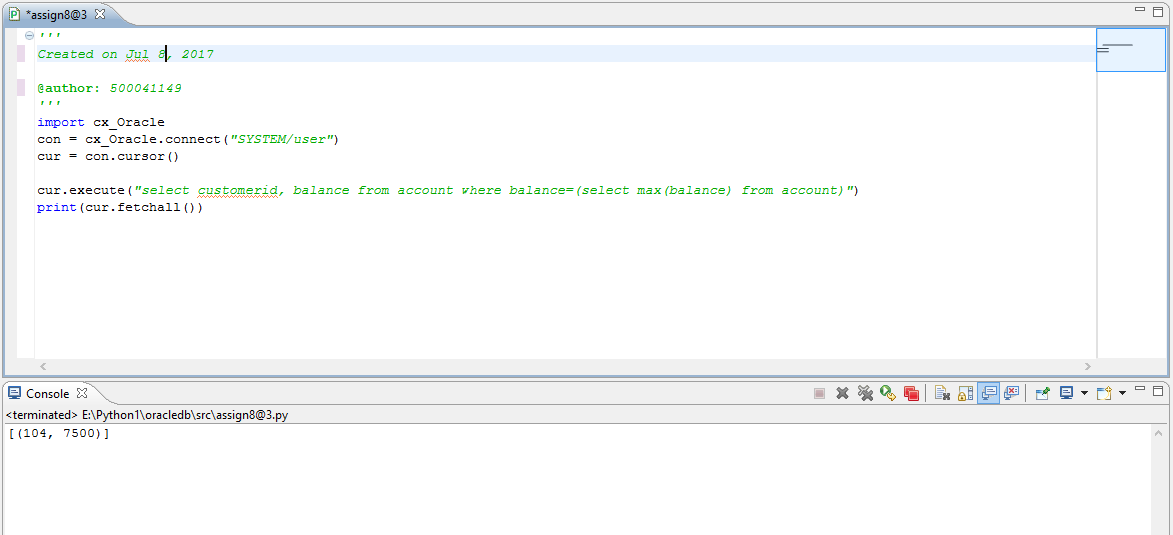
con = cx\_Oracle.connect(*"SYSTEM/user"*)

cur = con.cursor()

cur.execute(*"select customerid, balance from account where balance=(select max(balance) from account)"*)

print(cur.fetchall())

**Output:-**



**Ques4:-**Fetch the account balance of the customer with customer id 102 and store it in a Python variable – 'acct\_bal'

**Solution:-**

import cx\_Oracle

con = cx\_Oracle.connect(*"SYSTEM/user"*)

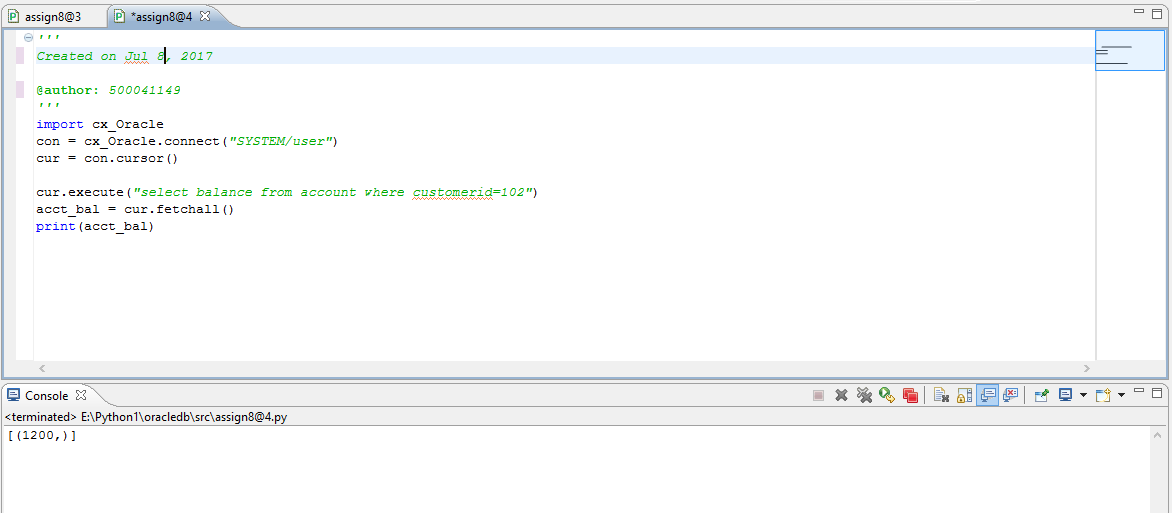
cur = con.cursor()

cur.execute(*"select balance from account where customerid=102"*)

acct\_bal = cur.fetchall()

print(acct\_bal)

**Output:-**

****

**Ques5:-**Increment 'acct\_bal' with 2000 and update the 'Balance' field of the table (for that particular customer) with the new value.

**Solution:-**

import cx\_Oracle

con = cx\_Oracle.connect(*"SYSTEM/user"*)

cur = con.cursor()

cur.execute(*"select balance from account where customerid=102"*)

acct\_bal = cur.fetchall()

print(acct\_bal)

print

print(*"after updation"*)

cur.execute(*"update account set balance = balance + 2000 where customerid=102"*)

con.commit()

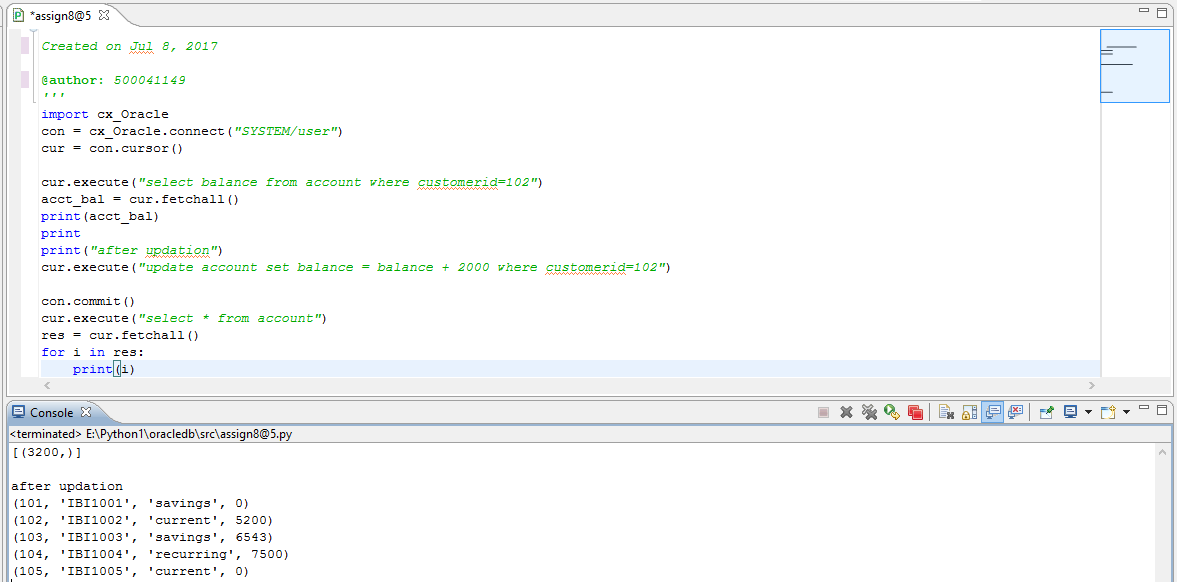
cur.execute(*"select \* from account"*)

res = cur.fetchall()

for i in res:

print(i)

**Output:-**



**Ques6:-**Fetch and observe the updated account balance of the customer with customer id 102.

**Solution:-**

import cx\_Oracle

con = cx\_Oracle.connect(*"SYSTEM/user"*)

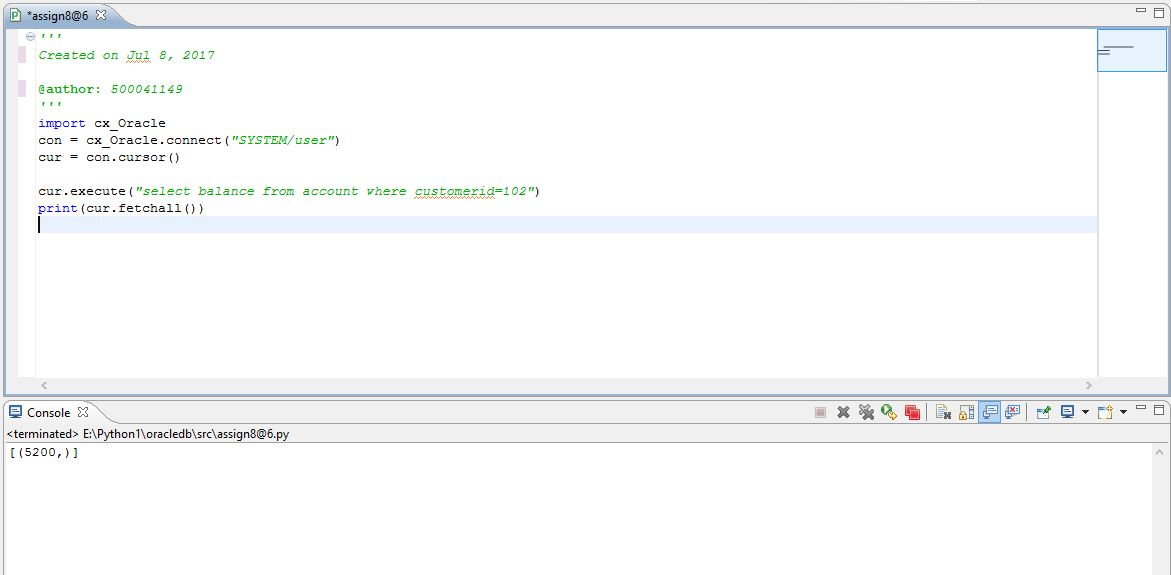
cur = con.cursor()

cur.execute(*"select balance from account where customerid=102"*)

print(cur.fetchall())

con.close()

**Output:-**



**Ques7:-**Delete the 'Current' accounts with zero balance.

**Solution:-**

import cx\_Oracle

con = cx\_Oracle.connect(*"SYSTEM/user"*)

cur = con.cursor()

cur.execute(*"delete from account where accounttype='current' and balance=0"*)

con.commit()

cur.execute(*"select \* from account"*)

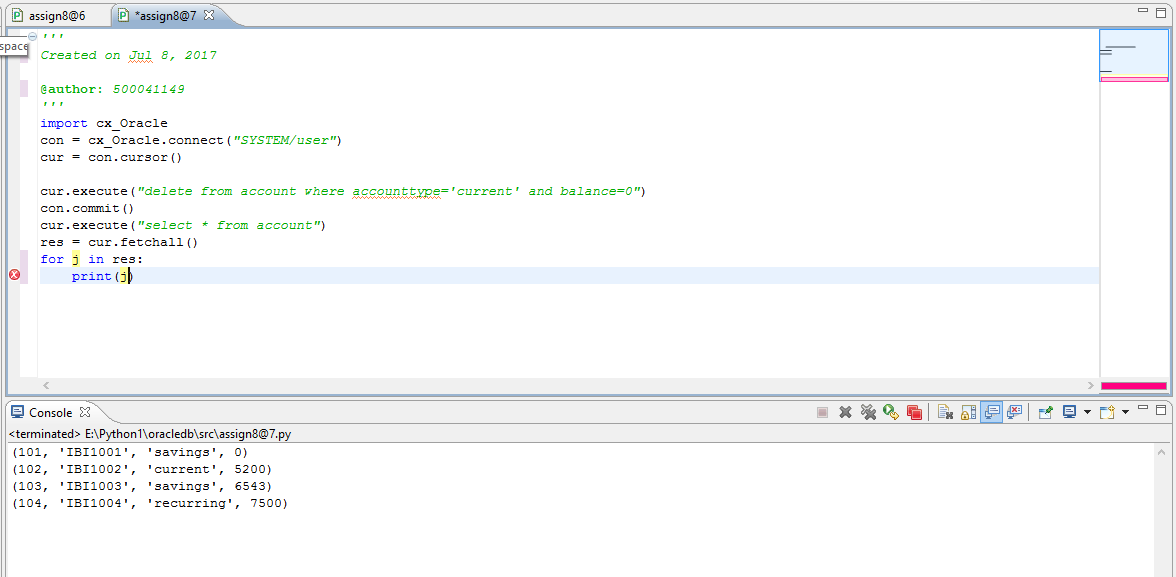
res = cur.fetchall()

for i in res:

print(i)

con.close()

**Output:-**



.

**ASSIGNMENT-9**

**Question:-**

Consider 'users' table already created. It has following data:

There is a requirement to delete the record of user with 'userid' 2.

**Ques1:-**Try to mention incorrect column name(e.g. user\_id) and observe the error.

**Solution:-**

import cx\_Oracle

con = cx\_Oracle.connect(*"SYSTEM/user"*)

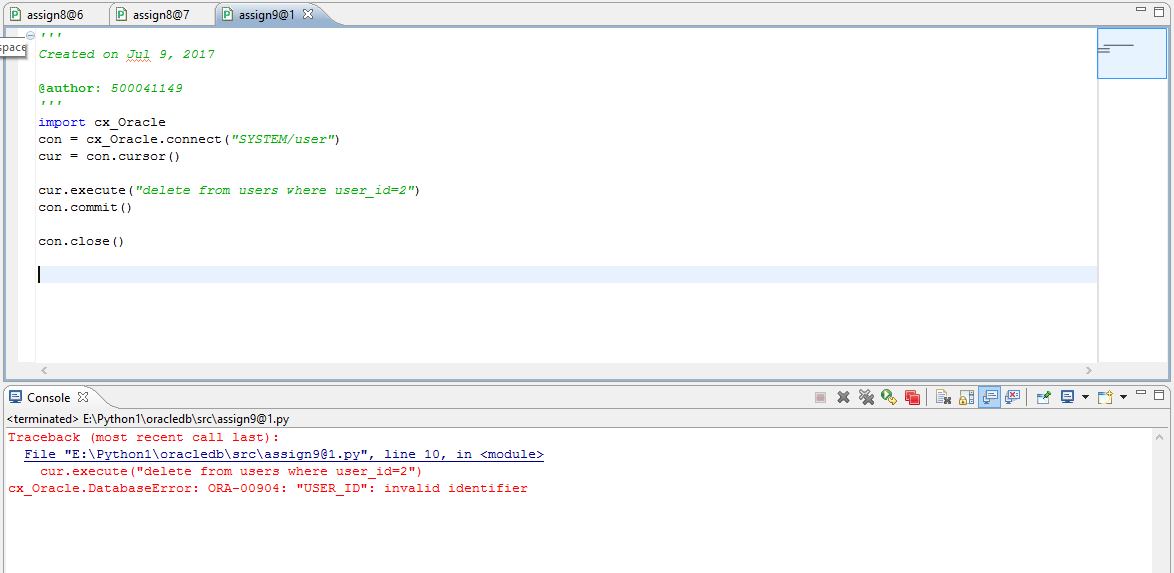
cur = con.cursor()

cur.execute(*"delete from users where user\_id=2"*)

con.commit()

con.close()

**Output:-**

****

**Ques2:-**Use exception handling to handle the exception appropriately. Display the error code and message.

**Solution:-**

import cx\_Oracle

con = cx\_Oracle.connect(*"SYSTEM/user"*)

cur = con.cursor()

try:

cur.execute(*"delete from users where user\_id=2"*)

con.commit()

except cx\_Oracle.DatabaseError as e:

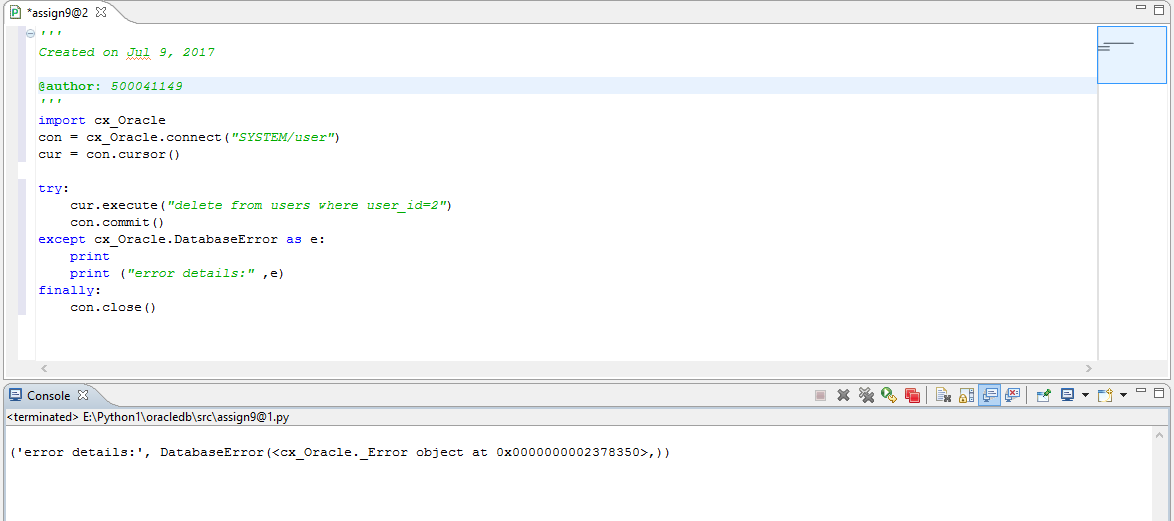
print

print (*"error details:"* ,e)

finally:

con.close()

**Output:-**



**Ques3:-**Try to give incorrect username for connection string and observe the error code and message.

**Solution:-**

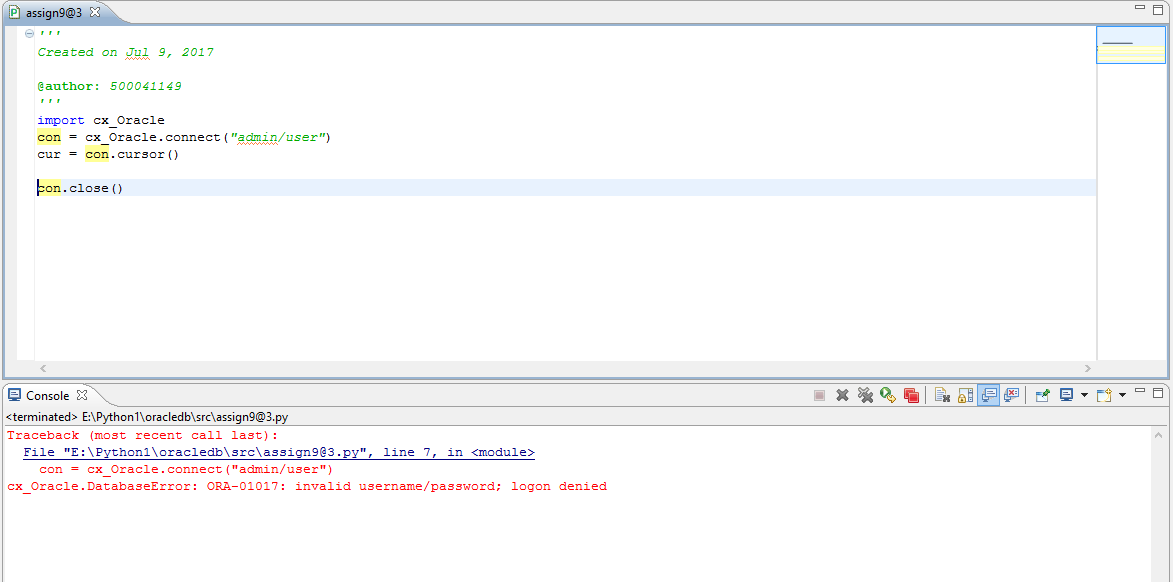
import cx\_Oracle

con = cx\_Oracle.connect(*"syste/user"*)

cur = con.cursor()

con.close()

**Output:-**

****

**Ques4:-**Provide a wrong table name while writing the query and observe the error message.

**Solution:-**

import cx\_Oracle

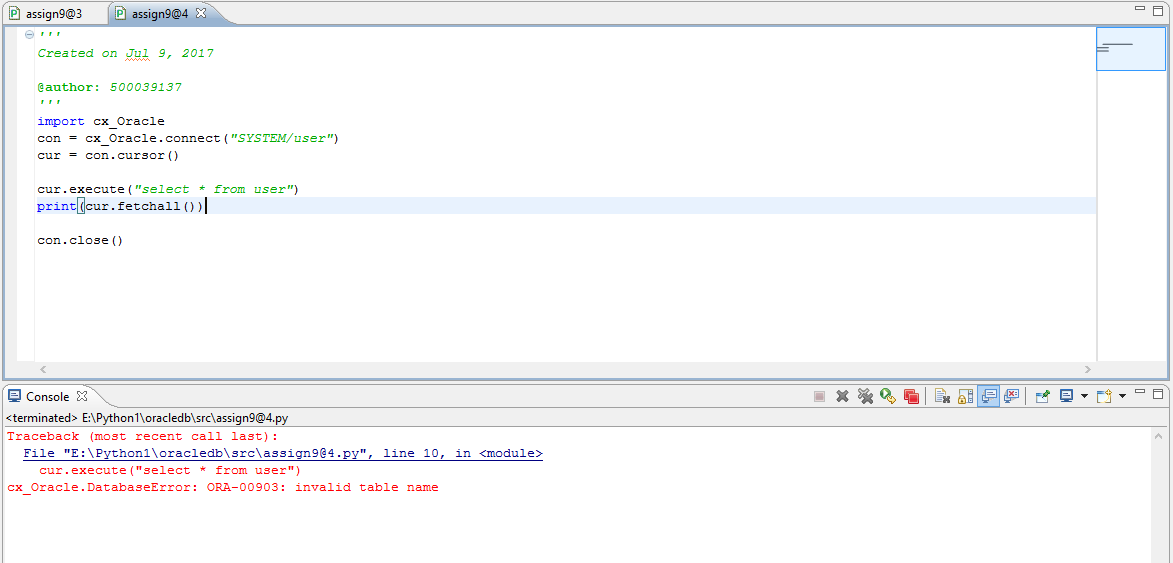
con = cx\_Oracle.connect(*"SYSTEM/user"*)

cur = con.cursor()

cur.execute(*"select \* from user"*)

con.close()

**Output:-**

****

**ASSIGNMENT-10**

**Question:-**

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

|  |  |  |  |
| --- | --- | --- | --- |
| **productid** | **type** | **price** | **quantity** |
| P106 | Jams | 150 | 30 |

**Ques1:-** 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.

**Solution:-**

import cx\_Oracle

con = cx\_Oracle.connect(*"SYSTEM/user"*)

cur = con.cursor()

try:

cur.execute(*"insert into product values('p106','Jams',150)"*)

except cx\_Oracle.DatabaseError as e:

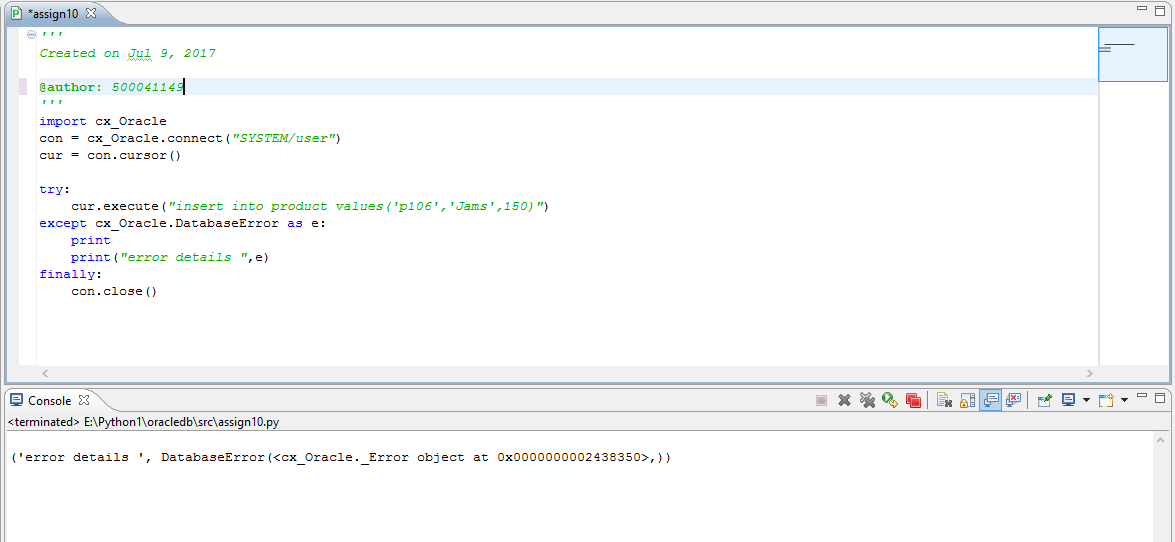
print

print(*"error details "*,e)

finally:

con.close()

**Output:-**

****