Program: 12

Oct 11, 2022

Problem Statement:

Start with the publication, book, and CD classes of previous exercise. Add a base class sale that holds an array of three floats so that it can record the dollar sales of a particular publication for the last three months. Include a getdata() function to get three sales amounts from the user, and a putdata() function to display the salesfigures. Alter the book and CD classes so they are derived from both publication and sales. An object of class book or CD should input and output sales data along withits other data. Write a main() function to create a book object and a tape object and exercise their input/output capabilities.

Solution:

```
class Sale:
    l1=list()
class Publication(Sale):
    def getData(self):
        self.title=input("Enter Title: ")
        self.price=input("Enter Price: ")
    def putData(self):
        print("\nTitle:",self.title)
        print("Price:",self.price)
class Book(Publication):
    def getData(self):
        super().getData()
        self.pg=input("Enter page count: ")
    def putData(self):
        super().putData()
        print("Page Count:",self.pg)
class CD(Publication):
    def getData(self):
        super().getData()
        self.playT=input("Enter playing time: ")
    def putData(self):
        super().putData()
        print("Playing Time:",self.playT,"min")
def main():
    obj1=Book()
    obj1.get_book_data()
    obj1.put_book_data()
    obj2=CD()
    obj2.get_cd_data()
    obj2.put_cd_data()
    print("The sale",Sale.l1)
main()
```

Program: 13

Oct 11, 2022

Problem Statement:

Write A Program to create a table in oracle

Solution:

```
import cx_Oracle
try:
    conn=cx_Oracle.connect("scott/sistec")
    cursor=conn.cursor()
    print(conn.version)
    sql="create table student(eno number(10),ename varchar2(30))"
    cursor.execute(sql)
    print("Table created successfully")
except conn.DatabaseError as e:
    print("Not connected due to msg=",e)
finally:
    cursor.close()
    conn.close()

11.2.0.2.0
Table created successfully
import cx_Oracle
try:
    con=cx_Oracle.connect("scott/sistec")
    cursor=con.cursor()
    sql="insert into students values(:eno,:ename)"
    records=[(1,"Deepak"),(2,"Amit")]
    cursor.executemany(sql,records)
    con.commit()
    print("record inserted successfully")
except cx_Oracle.DatabaseError as e:
    if con:
        con.rollback()
        print("There is a problem with sql",e)
finally:
    if cursor:
        cursor.close()
    if con:
        con.close()
record inserted successfully
```

Program: 14

Nov 01, 2022

Problem Statement:

Write a program for update query in mysqlite.

Solution:

```
import sqlite3
try:
   conn=sqlite3.connect("emp.db")
   cursor=conn.cursor() #print(conn.version)
   sql="CREATE TABLE IF NOT EXISTS emp(eno INTEGER, ename TEXT, esal INTEGER,
eaddr TEXT)"
   cursor.execute(sql)
   print("Table created successfully")
except conn.DatabaseError as e:
   print("Not connected due to msg=",e)
finally:
   if cursor:
      cursor.close()
   if conn:
      conn.close()
Table created successfully

Enter the number of records:4
Enter Employee Number: 1
Enter Employee Name:Ankit
Enter Employee Salary:11000
Enter Employee Address:Bhopal
record inserted successfully

Enter Employee Number: 2
Enter Employee Name:Amit
Enter Employee Salary:12000
Enter Employee Address:Matasi
record inserted successfully

Enter Employee Number: 3
Enter Employee Name:Sumit
Enter Employee Salary:15000
Enter Employee Address:Jamui
record inserted successfully

Enter Employee Number: 4
Enter Employee Name:Sunil
Enter Employee Salary:21000
Enter Employee Address:Gaya
record inserted successfully
```

```
import sqlite3
try:
    con=sqlite3.connect("emp.db")
    cursor=con.cursor()
    cursor.execute("select * from emp")
    data=cursor.fetchall()
    for row in data:
        print(row)
        print('#'*31)
except sqlite3.DatabaseError as e:
    if con:
        con.rollback()
        print("There is a problem with sql:",e)
finally:
    if cursor:
        cursor.close()
    if con:
        con.close()
(1, 'Ankit', 11000, 'Bhopal')
###############################
(2, 'Amit', 12000, 'Matasi')
###############################
(3, 'Sumit', 15000, 'Jamui')
###############################
(4, 'Sunil', 21000, 'Gaya')
###############################

# Update Query
import sqlite3
try:
    conn=sqlite3.connect("emp.db")
    cursor=conn.cursor()
    increment=float(input("Enter increment salary:"))
    salrange=float(input("enter salary range:"))
    sql="update emp set esal=esal+{} where esal>{}".format(increment,salrange)
    cursor.execute(sql)
    print("record updated successfully")
    conn.commit()
except sqlite3.DatabaseError as e:
    if conn:
        conn.rollback()
        print("There is a problem with sql",e)
finally:
    if cursor:
        cursor.close()
    if conn:
        conn.close()
Enter increment salary:22000
enter salary range:11000
record updated successfully
```

```
import sqlite3
try:
    con=sqlite3.connect("emp.db")
    cursor=con.cursor()
    cursor.execute("select * from emp")
    data=cursor.fetchall()
    for row in data:
        print(row)
        print('#'*31)
except sqlite3.DatabaseError as e:
    if con:
        con.rollback()
        print("There is a problem with sql:",e)
finally:
    if cursor:
        cursor.close()
    if con:
        con.close()
```

```
(1, 'Ankit', 11000, 'Bhopal')
###############################
(2, 'Amit', 34000, 'Matasi')
###############################
(3, 'Sumit', 37000, 'Jamui')
###############################
(4, 'Sunil', 43000, 'Gaya')
###############################
```

Program: 15

Nov 01, 2022

Problem Statement:

Write a Program to insert fake data in database.

Solution:

```
import sqlite3
conn=sqlite3.connect("books.db")
cur=conn.cursor()
cur.execute("CREATE TABLE IF NOT EXISTS emp(eno INTEGER, ename TEXT, esal
INTEGER, eaddr TEXT)")
conn.commit()
conn.close()
from faker import Faker
import random
fake=Faker()
import sqlite3
con=sqlite3.connect("books.db")
cursor=con.cursor()
def populate(n):
    for i in range(n):
        eno=fake.random_int(min=1,max=999)
        ename=fake.name()
        esal=round(random.uniform(5000,25000),2)
        eaddr=fake.city()
        sql="insert into emp values(%d,'%s', %.2f, '%s' )"%(eno,ename,esal,eaddr)
        cursor.execute(sql)
        con.commit()
n=int(input("Enter the number of records:"))
populate(n)
print("record inserted successfully")
cursor.close()
con.close()
```

```
Enter the number of records:11
record inserted successfully
```

```
import sqlite3
try:
    con=sqlite3.connect("books.db")
    cursor=con.cursor()
    cursor.execute("select * from emp")
    n=int(input("Enter the number of required rows:"))
    data=cursor.fetchmany(n)
    for row in data:
        print(row)
        print('#'*51)
except cx_Oracle.DatabaseError as e:
    if con:
        con.rollback()
        print("There is a problem with sql:",e)
finally:
    if cursor:
        cursor.close()
    if con:
        con.close()
```

```
Enter the number of required rows:5
(504, 'Aaron Mullins', 13800.96, 'Burnettshire')
###################################################
(168, 'Andrew Reed', 5875.94, 'South Sherry')
###################################################
(254, 'Daniel Mcmahon', 14753.8, 'North Jeannebury')
###################################################
(377, 'Mary Morrison', 13773.5, 'Lake Tyler')
###################################################
(380, 'Jennifer Fisher', 11479.44, 'Lindaton')
###################################################
```