

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | |
|  | | CSLR 51 : DBMS LAB-8 | | | | |  | |
|  |  | | | | | | |  |
|  | | | |  |  | | | |
|  | | | | Roll no. : 106119100Name : Rajneesh PandeySection : CSE-B |  | | | |
|  | | |  | | |  | | |

Design a simple database for Online Shopping Cart using Python to access the back-end MySQL database. The online shopping cart must contain the following modules.

import mysql.connector

from datetime import date, datetime

mydb = mysql.connector.connect(*host*="localhost", *user*='root', *password*='12345')

mycursor = mydb.cursor()

mycursor.execute("DROP DATABASE shopDB")

mycursor.execute("CREATE DATABASE shopDB")

mycursor.execute("USE shopDB")

mycursor.execute("Create table product(id int PRIMARY KEY, name varchar(20),category varchar(20), quantity int, price float, discount float, manf date,exp date)")

cont = "T"

opt = 0

while (cont == "t" or cont == "T"):

    opt = *int*(input("1. insert\n2.find\n3.search\_in\_category\n4.update\n5.provide\_discount\n6.delete\n7.notification\nEnter your choice: "))

    if (opt == 1):

        insert\_record()

    elif(opt == 2):

        find\_record()

    elif(opt == 3):

        search\_category()

    elif(opt == 4):

        update\_record()

    elif(opt == 5):

        provide\_discount()

    elif(opt == 6):

        delete\_record()

    elif(opt == 7):

        notify()

    else:

        print("invalid option")

    cont = input("do you want to continue(T/F): ")

1.The insert module must be able to accept the prod\_id(primary key),product name, category, quantity, price, discount, date of manufacture and date of expiry and store it in the database.

*def* insert\_record():

        prod\_id = *int*(input("Enter id: "))

        name = input("Enter name: ")

        category = input("Enter category: ")

        quantity = *int*(input("Enter quantity: "))

        price = *float*(input("Enter price in RS: "))

        discount = *float*(input("Enter discount in %: "))

        inputDate = input("Enter the date in format 'dd/mm/yy' : ")

        day1, month1, year1 = (inputDate.split('/'))

        manf = date(*int*(year1), *int*(month1), *int*(day1))

        inputDate = input("Enter the date in format 'dd/mm/yy' : ")

        day, month, year = inputDate.split('/')

        exp = date(*int*(year), *int*(month), *int*(day))

        sql = "INSERT INTO product(id, name, category, quantity, price, discount,manf, exp) values (%s, %s, %s, %s, %s, %s, %s, %s)"

        val = (prod\_id, name, category, quantity, price, discount, manf, exp)

        mycursor.execute(sql, val)

        mydb.commit()

        print("record inserted")

Input/ Output :

Text

Description automatically generated

2.The find module must be able to accept the name of the product and display all the details of the product.

*def* find\_record():

        name = input("Enter name: ")

        sql = *f*'SELECT \* FROM product WHERE name = "{name}"'

        mycursor.execute(sql)

        myresult = mycursor.fetchall()

        if (len(myresult) == 0):

            print("no records")

        else:

            for x in myresult:

                print(x)

Input/Output :

Text

Description automatically generated

3.The search\_in\_category module must be able to accept the name of the category and display all the details of the products in the categories.

*def* search\_category():

        category = input("Enter category: ")

        sql = *f*'SELECT \* FROM product WHERE category = "{category}"'

*# que = (category)*

        print(sql)

        mycursor.execute(sql)

        myresult = mycursor.fetchall()

        if (len(myresult) == 0):

            print("no records")

        else:

            for x in myresult:

                print(x)

Input/Output :

Text

Description automatically generated

4.The update module must be able to update the price of the products.

*def* update\_record():

        prod\_id = *int*(input("Enter id: "))

        price = *float*(input("Enter price: "))

        sql = *f*'UPDATE product SET price = "{price}" WHERE id = "{prod\_id}"'

*# val = (price, prod\_id)*

        mycursor.execute(sql)

        mydb.commit()

        print("record updated")

Input/Output :

Text

Description automatically generated

5.The provide\_discount module must be able to provide the discount of the products in a particular category.

*def* provide\_discount():

        category = input("Enter category: ")

        sql = *f*'SELECT name, discount FROM product WHERE category = "{category}"'

*# que = (category)*

        mycursor.execute(sql)

        myresult = mycursor.fetchall()

        if (len(myresult) == 0):

            print("no records")

        else:

            for x in myresult:

                print(x[0]+' : %d'%x[1])

Input/Output :

Text

Description automatically generated

6.The delete\_module must be able to delete/ cancel the product if the expiry date is not valid.

*def* delete\_record():

        mycursor.execute('SELECT \* FROM product')

        myresult = mycursor.fetchall()

        if (len(myresult) == 0):

            print("no records")

        else:

            for x in myresult:

                if (x[7] < datetime.now().date()):

                    print("record with id %d deleted" % x[0])

        sql = *f*"DELETE FROM product WHERE id = {x[0]}"

        mycursor.execute(sql)

        mydb.commit()

Input/Output :

Text

Description automatically generated

7.The notification\_module must indicate free shipping if the total product price exceeds Rs.1000.

*def* notify():

        prod\_id = *int*(input("Enter id: "))

        sql = *f*'SELECT price, discount FROM product WHERE id = {prod\_id}'

        que = (prod\_id)

        mycursor.execute(sql, que)

        myresult = mycursor.fetchall()

        if (len(myresult) == 0):

            print("no records")

        else:

             s = 0

             for x in myresult:

                s += x[0]

                s -= (x[0]\*x[1]\*0.01)

             if (s > 1000):

               print("Free shipping")

             else:

               print("no free shipping")

Input/Output :

Text

Description automatically generated