**SOLENT UNIVERSITY**

**Unit Title:** Databases (COM711)

**Report 1 (Python Code)**

**Student Name:** Ajaz *Ali Saiyed*

**Date:** *4th November*

***Full Python code for Part-3***

**import** sqlite3 **as** sq  
  
con = sq.connect(**'solent.db'**)  
obj = con.cursor()  
  
*# Store the option value and display the categories menu or option***def** opt\_lst(function, header, category):  
 serial = 1  
lst = []  
print(**"\n"**, header, **"\n"**)  
**for** x **in** function:  
 num = x[0]  
val = x[1]  
print(**"{0}.\t{1}"**.format(serial, val))  
 serial = serial + 1  
lst.append(num)  
 i = 0  
**while** i >len(lst) **or** i == 0:  
 display = **"Enter the number against the "** + category + **" you want to choose: "**i = int(input(display))  
**return** lst[i - 1]  
  
*# take input from user about the shopper ID*id = int(input(**"Please enter your shopper ID: "**))  
  
  
**def** prod\_cat(): *# Stores category\_id and category\_description*obj.execute(**"SELECT category\_id, category\_description FROM categories"**)  
 j = obj.fetchall()  
**return** j  
  
  
**def** shopper\_name(): *# stores shopper name*obj.execute(**"SELECT shopper\_first\_name, shopper\_surname FROM shoppers WHERE shopper\_id = ?"**, (id,))  
 result = obj.fetchall()  
**for** i **in** result:  
f\_name = i[0]  
l\_name = i[1]  
 name = f\_name + **" "** + l\_name  
**return** name  
  
  
  
**def** id\_lst(): *# stores list of shopper IDs*obj.execute(**"SELECT shopper\_id FROM shoppers"**)  
 temp = obj.fetchall()  
 x = []  
**for** val**in** temp:  
x.append(val[0])  
**return** x  
  
  
**def** menu\_ui(): *# UI for shopper main menu*print(**"\nORINOCO-SHOPPER MAIN MENU"**)  
print(**"-"** \* 50)  
print(**"1. Display Your Order History"**)  
print(**"2. Add and Item to Basket"**)  
print(**"3. View Your Basket"**)  
print(**"4. Check Out"**)  
print(**"5. Exit"**)  
  
  
**def** add\_address\_and\_card(): *# add new payment method and address*print(**"\nAs you have not placed order, you will need to enter a delivery address\n"**)  
 add1 = input(**"Enter the delivery address line 1: "**)  
 add2 = input(**"Enter the delivery address line 2: "**)  
 add3 = input(**"Enter the delivery address line 3: "**)  
 add4 = input(**"Enter the delivery county: "**)  
 pin = input(**"Enter the delivery post code: "**)  
*# insert a new row into the shopper\_delivery\_addresses*query = **"""  
 INSERT INTO shopper\_delivery\_addresses (delivery\_address\_line\_1, delivery\_address\_line\_2,  
 delivery\_address\_line\_3, delivery\_county, delivery\_post\_code)  
 VALUES(?,?,?,?,?)  
 """**obj.execute(query, (add1, add2, add3, add4, pin))  
con.commit() *# commit changes*print(**"\nAs you have not placed order, you will need to enter a your payment card details \n"**)  
card\_type = input(**"Enter the card type (Visa, Mastercard or AMEX) : "**)  
card\_number = input(**"Enter the 16-digit card number: "**)  
*# insert a new row into the shopper\_payment\_cards*obj.execute(**"""  
 INSERT INTO shopper\_payment\_cards (card\_type, card\_number )  
 VALUES (?, ?)  
 """**, (card\_type, card\_number))  
print(**"\nCheckout complete, your order has been placed\n"**)  
con.commit() *# commit changes***if** id **in** id\_lst():  
print(**"\nWelcome "** + shopper\_name())  
  
**while True**:  
**if** id **not in** id\_lst():  
print(**"\nThe shopper does not exist\n"**)  
**break  
 else**:  
menu\_ui()  
  
 temp = int(input(**"\nPlease enter your choice: "**))  
  
**if** temp == 1:  
obj.execute(**""" SELECT so.order\_id, order\_date, product\_description, seller\_name, price, quantity,   
ordered\_product\_status FROM shoppers s   
 INNER JOIN shopper\_orders so  
 ON s.shopper\_id = so.shopper\_id  
 INNER JOIN ordered\_products op   
 ON so.order\_id = op.order\_id  
 INNER JOIN sellers s   
 ON op.seller\_id = s.seller\_id  
 INNER JOIN products p   
 ON p.product\_id = op.product\_id  
 WHERE s.shopper\_id = ?   
 ORDER BY order\_date DESC """**, (id,))  
 j = obj.fetchall()  
**if** len(j) == 0: *# check if no orders*print(**"\nNo orders placed by this customer\n"**)  
**else**:  
print(**"\n{0} {1:10} {2:70} {3:18} {4:7} {5:3} {6}\n"**.format(**"Order ID"**, **"Order Date"**, **"Description"**, **"Seller"**, **"Price"**, **"Qty"**, **"Status"**))  
**for** val**in** j: *# Assigning values  
 # Print table*print(**"{0} {1:10} {2:70} {3:18} £{4:7.2f} {5:3} {6}"**.format(val[0], val[1], val[2], val[3], val[4], val[5], val[6]))  
  
  
**elif**temp == 2:  
 var1 = opt\_lst(prod\_cat(), **"Product Categories"**, **"product category"**)  
obj.execute(**"SELECT product\_id, product\_description FROM products WHERE category\_id = ?"**, (var1,))  
 res1 = obj.fetchall()  
 var2 = opt\_lst(res1, **"Products"**, **"product"**)  
obj.execute(**""" SELECT sellers.seller\_id, seller\_name || ' ' || '(' || '£' || price || ')' FROM products   
 INNER JOIN product\_sellers ON product\_sellers.product\_id = products.product\_id  
 INNER JOIN sellers ON sellers.seller\_id = product\_sellers.seller\_id  
 WHERE products.product\_id = ? """**, (var2,))  
 res2 = obj.fetchall()  
 var3 = opt\_lst(res2, **"Sellers who sell this product"**, **"seller"**)  
obj.execute(**"SELECT price FROM product\_sellers WHERE product\_id = ? AND seller\_id = ?"**, (var2, var3))  
 price = obj.fetchall()[0][0]  
 qty = int(input(**"Enter the quantity of selected product you want to buy: "**))  
obj.execute(**"SELECT seq FROM sqlite\_sequence WHERE name = 'shopper\_baskets' "**)  
 basket = obj.fetchall()[0][0]  
obj.execute(**"INSERT INTO shopper\_baskets(basket\_id, shopper\_id, basket\_created\_date\_time) "  
 "VALUES(?, ?, DATE('NOW'))"**, (basket, id))  
con.commit()  
obj.execute(**"INSERT INTO basket\_contents(basket\_id, product\_id, seller\_id, quantity, price) "  
 "VALUES (? ,? , ?, ?, ?)"**, (basket, var2, var3, qty, price))  
con.commit()  
print(**"\nItem added to basket"**)  
  
**if** temp == 3:  
obj.execute(**"SELECT product\_description, seller\_name, quantity, price, quantity \* price AS total "  
 "FROM basket\_contentsbc "  
 "INNER JOIN sellers s ON s.seller\_id = bc.seller\_id "  
 "INNER JOIN products p ON p.product\_id = bc.product\_id"**)  
 j = obj.fetchall()  
**if** len(j) == 0: *# check if the basket is empty*print(**"Error! The basket is empty"**)  
**else**:  
print(**"\nBasket Contents\n"  
 "----------------\n"**)  
print(**"{0:70} {1:20} {2:4} {3:7} {4}\n"**.format(**"Description"**, **"Seller"**, **"Qty"**, **"Price"**,  
**"Total"**))  
**for** i **in** j:  
print(**"{0:70} {1:20} {2:4} £{3:7.2f} £{4}"**.  
 format(i[0], i[1], i[2], i[3], i[4]))  
  
  
**elif**temp == 4:  
obj.execute(**"SELECT sda.delivery\_address\_id, delivery\_address\_line\_1 || ', ' || "  
 "delivery\_address\_line\_2 || ', ' || COALESCE(delivery\_address\_line\_3,' ') || ', ' || delivery\_county "  
 "|| ', ' || delivery\_post\_code FROM shopper\_delivery\_addressessda "  
 "INNER JOIN shopper\_orders so ON so.delivery\_address\_id = sda.delivery\_address\_id "  
 "WHERE shopper\_id = ?"**, (id,))  
full\_address = list(dict.fromkeys(obj.fetchall()))  
  
**if** len(full\_address) == 0:  
add\_address\_and\_card()  
**else**:  
add\_id = opt\_lst(full\_address, **"Delivery Addresses"**, **"delivery address"**)  
obj.execute(**"""  
 SELECT  
shopper\_payment\_cards.payment\_card\_id,  
card\_type || ' ending in ' || card\_number  
 FROM shopper\_payment\_cards  
 INNER JOIN shopper\_orders  
 ON shopper\_payment\_cards.payment\_card\_id = shopper\_orders.payment\_card\_id  
 WHERE shopper\_orders.delivery\_address\_id = ? """**, (add\_id,))  
payment\_cards = list(dict.fromkeys(obj.fetchall()))  
id\_of\_payment\_cards = opt\_lst(payment\_cards, **"Payment cards"**, **"payment card"**)  
  
obj.execute(**"INSERT INTO shopper\_orders (shopper\_id, delivery\_address\_id, payment\_card\_id, order\_date, order\_status) "  
 "VALUES (?, ?, ?, DATE('NOW'), 'Placed')"**, (id, add\_id, id\_of\_payment\_cards))  
con.commit() *# commit changes*obj.execute(**"INSERT INTO ordered\_products (product\_id, seller\_id, quantity, price, ordered\_product\_status) "  
 "VALUES (?, ?, ?, ?, 'Placed')"**, (var2, var3, qty, price))  
print(**"\nCheckout complete, your order has been placed\n"**)  
con.commit() *# commit changes***break  
  
elif**temp == 5:  
obj.executescript(**"DELETE FROM shopper\_baskets;"  
 "DELETE FROM basket\_contents;"**)  
**break  
  
 else**:  
print(**"\nWrong option, input the correct option again to continue\n"**)  
  
con.close()