**E-COMMERCE DATABASE SYSTEM**

1. ABSTRACT

The rapid growth of online shopping has led to an increasing demand for small clothing sellers to transition from traditional offline models to online platforms. However, this transition can be daunting, especially when it comes to managing and storing large amounts of product data, customer information, and sales transactions. To address this challenge, there is a need for a robust and scalable database system that can efficiently manage and support the online sales process for small clothing sellers.

The objectives of this project are to:

* design a comprehensive database system for small clothing sellers to manage their online sales process.
* develop a user-friendly interface for sellers to easily add, update, and manage their product catalog.
* ensure the database system is scalable, secure, and can handle increasing amounts of data and traffic.
* provide features for sellers to track sales, customer information, and product performance.

1. MODULE SPECIFICATION

This project uses the following modules:

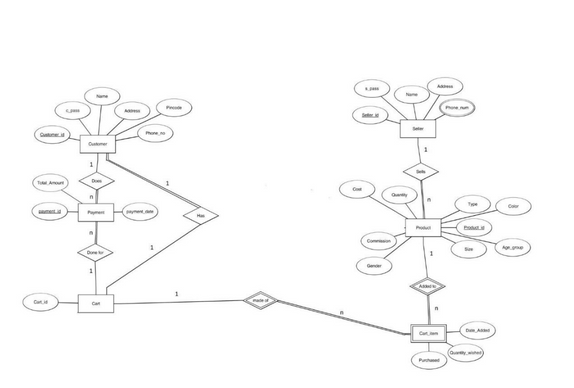
* mysql.connector

This module helps the app with the Python code connect to the one with the database and tables, written in SQL language.

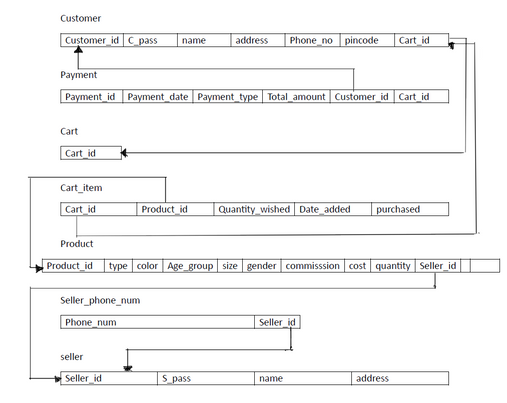
* tabulate

This module helps to display the data acquired in a neat tabular form.

1. ER DIAGRAM



1. RELATIONAL MAPPING



1. SQL QUERIES

* Creation of Tables

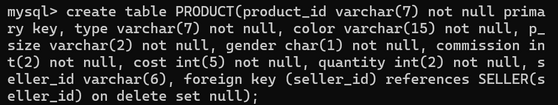
5edc1f1c751de1932a38027397d5bb21.png

f53b177637bf51c1eba909cedafc324e.png

4d2238c46d708253175b032992e05a6a.png

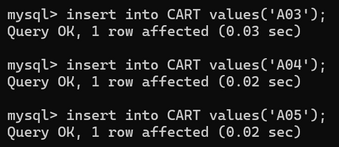
b914284b0026d1c74a42e2bfc6660339.png

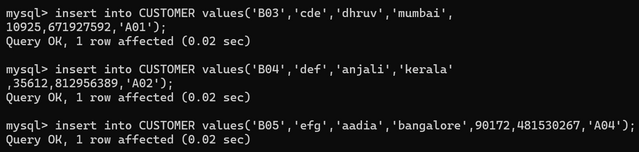
a9b9cfaea5b4e8d37d43dbc075e4da0b.png

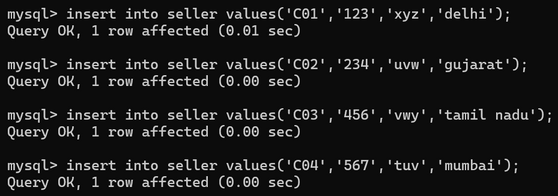


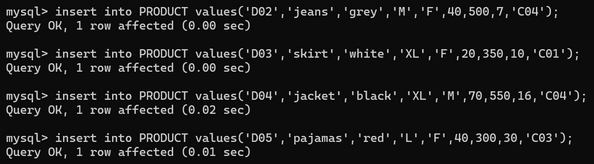
1f92081c4499dd2634935638dee4627b.png

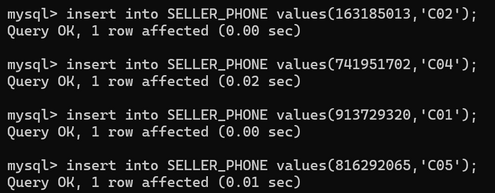
* Insertion of Values

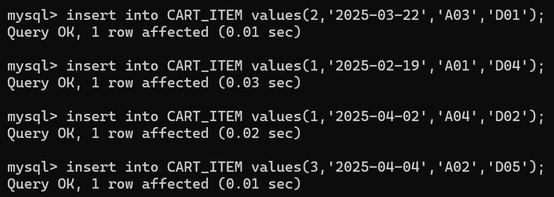


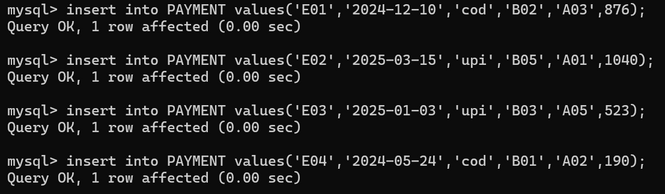












1. SAMPLE CODE

import mysql.connector as mc

import tabulate

def display\_cart():

try:

mycon=mc.connect(host="localhost", user="root", password="anjali24", database="project")

mycur=mycon.cursor()

mycur.execute("select \* from CART")

rs=mycur.fetchall()

print("CART TABLE")

print(tabulate.tabulate(rs,headers=['CART\_ID'],tablefmt='fancy\_grid'))

except Exception as e:

print(e)

mycur.close()

mycon.close()

def display\_customer():

try:

mycon=mc.connect(host="localhost", user="root", password="anjali24", database="project")

mycur=mycon.cursor()

mycur.execute("select \* from CUSTOMER order by customer\_id")

rs=mycur.fetchall()

print("CUSTOMER TABLE")

print(tabulate.tabulate(rs,headers=['CUSTOMER\_ID','C\_PASS','NAME','ADDRESS','PINCODE','PHONE','CART\_ID'],tablefmt='fancy\_grid'))

except Exception as e:

print(e)

mycur.close()

mycon.close()

def display\_seller():

try:

mycon=mc.connect(host="localhost", user="root", password="anjali24", database="project")

mycur=mycon.cursor()

mycur.execute("select \* from SELLER order by seller\_id")

rs=mycur.fetchall()

print("SELLER TABLE")

print(tabulate.tabulate(rs,headers=['SELLER\_ID','S\_PASS','NAME','ADDRESS'],tablefmt='fancy\_grid'))

except Exception as e:

print(e)

mycur.close()

mycon.close()

def display\_product():

try:

mycon=mc.connect(host="localhost", user="root", password="anjali24", database="project")

mycur=mycon.cursor()

mycur.execute("select \* from PRODUCT order by product\_id")

rs=mycur.fetchall()

print("PRODUCT TABLE")

print(tabulate.tabulate(rs,headers=['PRODUCT\_ID','TYPE','COLOR','P\_SIZE','GENDER','COMMISSION','COST','QUANTITY','SELLER\_ID'],tablefmt='fancy\_grid'))

except Exception as e:

print(e)

mycur.close()

mycon.close()

def display\_sellerphone():

try:

mycon=mc.connect(host="localhost", user="root", password="anjali24", database="project")

mycur=mycon.cursor()

mycur.execute("select \* from SELLER\_PHONE order by phone\_num")

rs=mycur.fetchall()

print("SELLER\_PHONE TABLE")

print(tabulate.tabulate(rs,headers=['PHONE\_NUM','SELLER\_ID'],tablefmt='fancy\_grid'))

except Exception as e:

print(e)

mycur.close()

mycon.close()

def display\_cartitem():

try:

mycon=mc.connect(host="localhost", user="root", password="anjali24", database="project")

mycur=mycon.cursor()

mycur.execute("select \* from CART\_ITEM order by cart\_id")

rs=mycur.fetchall()

print("CART\_ITEM TABLE")

print(tabulate.tabulate(rs,headers=['QUANTITY\_WISHED','DATE\_ADDED','CART\_ID','PRODUCT\_ID'],tablefmt='fancy\_grid'))

except Exception as e:

print(e)

mycur.close()

mycon.close()

def display\_payment():

try:

mycon=mc.connect(host="localhost", user="root", password="anjali24", database="project")

mycur=mycon.cursor()

mycur.execute("select \* from PAYMENT order by payment\_id")

rs=mycur.fetchall()

print("PAYMENT TABLE")

print(tabulate.tabulate(rs,headers=['PAYMENT\_ID','PAYMENT\_DATE','PAYMENT\_TYPE','ÇUSTOMER\_ID','CART\_ID','TOTAL\_AMOUNT'],tablefmt='fancy\_grid'))

except Exception as e:

print(e)

mycur.close()

mycon.close()

def insert():

try:

mycon=mc.connect(host="localhost", user="root", password="anjali24", database="project")

mycur=mycon.cursor()

table=input("Enter table you would like to add a record to: ")

if table.upper()=="CART":

display\_cart()

icart\_id=input("Enter cart ID: ")

q="insert into CART values('{}')".format(icart\_id)

mycur.execute(q)

mycon.commit()

print()

print("Updated table is: ")

display\_cart()

elif table.upper()=="CUSTOMER":

display\_customer()

icust\_id=input("Enter customer ID: ")

ic\_pass=input("Enter password: ")

iname=input("Enter name: ")

iaddress=input("Enter address: ")

ipincode=int(input("Enter pincode: "))

iphone=int(input("Enter phone number: "))

icart\_id=input("Enter cart ID: ")

q="insert into CUSTOMER values('{}','{}','{}','{}',{},{},'{}')".format(icust\_id,ic\_pass,iname,iaddress,ipincode,iphone,icart\_id)

mycur.execute(q)

mycon.commit()

print()

print("Updated table is: ")

display\_customer()

elif table.upper()=="SELLER":

display\_seller()

iseller\_id=input("Enter seller ID: ")

is\_pass=input("Enter password: ")

iname=input("Enter name: ")

iaddress=input("Enter address: ")

q="insert into SELLER values('{}','{}','{}','{}')".format(iseller\_id,is\_pass,iname,iaddress)

mycur.execute(q)

mycon.commit()

print()

print("Updated table is: ")

display\_seller()

elif table.upper()=="PRODUCT":

display\_product()

iproduct\_id=input("Enter product ID: ")

itype=input("Enter product type: ")

icolor=input("Enter product color: ")

ip\_size=input("Enter size: ")

igender=input("Enter gender: ")

icom=int(input("Enter commission: "))

icost=int(input("Enter product cost: "))

iqty=int(input("Enter quantity left: "))

iseller\_id=input("Enter seller ID: ")

q="insert into PRODUCT values('{}','{}','{}','{}','{}',{},{},{},'{}'".format(iproduct\_id,itype,icolor,ip\_size,igender,icom,icost,iqty,iseller\_id)

mycur.execute(q)

mycon.commit()

print()

print("Updated table is: ")

display\_product()

elif table.upper()=="SELLER\_PHONE":

display\_sellerphone()

iphone\_num=int(input("Enter phone number: "))

iseller\_id=input("Enter seller ID: ")

q="insert into SELLER\_PHONE values({},'{}')".format(iphone\_num,iseller\_id)

mycur.execute(q)

mycon.commit()

print()

print("Updated table is: ")

display\_sellerphone()

elif table.upper()=="CART\_ITEM":

display\_cartitem()

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

idate=input("Enter date (YYYY-MM-DD): ")

icart\_id=input("Enter cart ID: ")

iproduct\_id=input("Enter product ID: ")

q="insert into CART\_ITEM values({},'{}','{}','{}')".format(iqtywished,idate,icart\_id,iproduct\_id)

mycur.execute(q)

mycon.commit()

print()

print("Updated table is: ")

display\_cartitem()

elif table.upper()=="PAYMENT":

display\_payment()

ipayment\_id=input("Enter payment ID: ")

ipayment\_date=input("Enter payment date (YYYY-MM-DD): ")

ipayment\_type=input("Enter payment type: ")

icustomer\_id=input("Enter customer ID: ")

icart\_id=input("Enter cart ID: ")

itotal\_amount=int(input("Enter total amount: "))

q="insert into PAYMENT values('{}','{}','{}','{}','{}',{})".format(ipayment\_id,ipayment\_date,ipayment\_type,icustomer\_id,icart\_id,itotal\_amount)

mycur.execute(q)

mycon.commit()

print()

print("Updated table is: ")

display\_payment()

else:

print("Table does not exist")

pass

except Exception as e:

print(e)

mycur.close()

mycon.close()

def update():

try:

mycon=mc.connect(host="localhost", user="root", password="anjali24", database="project")

mycur=mycon.cursor()

table=input("Enter table you would like to update: ")

if table.upper()=="CART":

print("Cannot update table as table has only one field")

pass

elif table.upper()=="CUSTOMER":

display\_customer()

ids=input("Enter customer ID: ")

op=input("Enter if you would like to change phone number(P) or address(A): ")

if op.upper()=='P':

iphone=input("Enter new phone number: ")

q="update CUSTOMER set phone={} where customer\_id='{}'".format(iphone,ids)

mycur.execute(q)

mycon.commit()

elif op.upper()=='A':

iaddress=input("Enter new address: ")

ipincode=int(input("Enter new pincode: "))

q="update CUSTOMER set address='{}',pincode={} where customer\_id='{}'".format(iaddress,ipincode,ids)

mycur.execute(q)

mycon.commit()

print()

print("Updated table is: ")

display\_customer()

elif table.upper()=="SELLER":

print("Cannot modify any fields in this table")

pass

elif table.upper()=="PRODUCT":

display\_product()

ids=input("Enter product ID: ")

op=input("Enter if you would like to change cost(C) or quantity(Q): ")

if op.upper()=='C':

icost=int(input("Enter new product cost: "))

q="update PRODUCT set cost={} where product\_id='{}'".format(icost,ids)

mycur.execute(q)

mycon.commit()

elif op.upper()=='Q':

iqty=int(input("Enter new quantity: "))

q="update PRODUCT set quantity={} where product\_id='{}'".format(iqty,ids)

mycur.execute(q)

mycon.commit()

print()

print("Updated table is: ")

display\_product()

elif table.upper()=="SELLER\_PHONE":

print("Cannot modify any fields in this table")

pass

elif table.upper()=="CART\_ITEM":

display\_cartitem()

ids=input("Enter product ID: ")

iqwished=int(input("Enter new quantity of item: "))

q="update CART\_ITEM set quantity\_wished={} where product\_id='{}'".format(iqwished,ids)

mycur.execute(q)

mycon.commit()

print()

print("Updated table is: ")

display\_cartitem()

elif table.upper()=="PAYMENT":

display\_payment()

ids=input("Enter payment ID: ")

ipayment\_type=input("Enter new payment type: ")

q="update PAYMENT set payment\_type='{}' where payment\_id='{}'".format(ipayment\_type,ids)

mycur.execute(q)

mycon.commit()

print()

print("Updated table is: ")

display\_payment()

else:

print("Table does not exist")

pass

except Exception as e:

print(e)

mycur.close()

mycon.close()

def delete():

try:

mycon=mc.connect(host="localhost", user="root", password="anjali24", database="project")

mycur=mycon.cursor()

table=input("Enter table you would like to delete records from: ")

if table.upper()=="CART\_ITEM":

display\_cartitem()

ids=input("Enter cart ID: ")

q="delete from CART\_ITEM where cart\_id='{}'".format(ids)

mycur.execute(q)

mycon.commit()

print()

print("Updated table is: ")

display\_cartitem()

elif table.upper()=="PAYMENT":

display\_payment()

iyear=input("Enter year of all old payment records that must be deleted: ")

q="delete from payment where payment\_date like '{}-\_\_-\_\_'".format(iyear)

mycur.execute(q)

mycon.commit()

print()

print("Updated table is: ")

display\_payment()

else:

print("Invalid option")

pass

except Exception as e:

print(e)

mycur.close()

mycon.close()

def main():

print("Welcome to Online Shopping Database Management !!")

choice='Y'

while choice.upper()=='Y':

print("Kindly pick your choice of action from the following: ")

print("1. Insertion")

print("2. Deletion")

print("3. Updation")

print("4. Viewing of All Available Tables")

print("5. Exit")

ch=int(input("Enter choice: "))

if ch==1:

insert()

print()

elif ch==2:

delete()

print()

elif ch==3:

update()

print()

elif ch==4:

display\_cart()

print()

display\_customer()

print()

display\_seller()

print()

display\_product()

print()

display\_sellerphone()

print()

display\_cartitem()

print()

display\_payment()

print()

elif ch==5:

break

choice=input("Would you like to continue? (Y/N): ")

print()

main()

1. OUTPUT

