# crud-menu

May 21, 2025

[1]:
```
print("Hello Guys,We are going to build an CRUD menu using mysql.connector.
  ↪connect")
```

```
Hello Guys,We are going to build an CRUD menu using mysql.connector.connect
```

## 0.1 Loading password from env file

[2]:
```
from dotenv import load_dotenv
import os

load_dotenv(dotenv_path=".env")
```

[2]:
```
True
```

### importing library to connect with MySQL

[3]:
```
import mysql.connector
```

### code to get inputs

[4]:
```
from datetime import datetime


#To get input and return it in  integer type
def get_int_input(prompt):
    while True:
        try:
            return int(input(prompt))
        except ValueError:
            print("Please enter valid value")

 #To get input of string type  (simple as all methods are getting it in string␣
  ↪format)
def get_str_input(prompt):
    value = input(prompt).strip()
    return value

#To get input and return it in  float type
```

```
def get_float_input(prompt):
     while True:
        try:
            return float(input(promt))
        except ValueError:
            print("Please enter valid value")

#To get input and return it in date type
def get_date_input(prompt):
    while True:
        try:
            date_str = input(prompt)
            date_obj = datetime.strtime(date_str,"%Y-%m-%d").date()
            return date_obj
        except ValueError:
            print("Please enter valid Date format (YYYY-MM-DD):")
```

## 0.2 Connecting to MySQL server

[5]:

```
def connect_db():
    return mysql.connector.connect(
    host="localhost",
    user="root",
    password = os.getenv("MYSQL_PASSWORD"),
    database = "ecommerce"
    )
```

## 0.3 =========== CRUD Functions for each table ============

============================ For ecom_customer ==========================

[6]:

```
def insert_customer(cursor):
    customer_id = get_int_input("Customer_ID :")
    name = get_str_input("Name :")
    email = get_str_input("Email :")
    city = get_str_input("City :")
    cursor.execute("Insert into ecom_customer␣
 ↪Values(%s,%s,%s,%s)",(customer_id,name,email,city))
    print("Customer added successfully\n")


def read_customer(cursor):
    cursor.execute("Select * from ecom_customer")
    for row in cursor.fetchall():
        print(row)
```

```python
def update_customer(cursor):
    cid = get_int_input(" Enter Customer_Id to be updated :")
    city = get_str_input("New city :")
    cursor.execute("Update ecom_customer SET city = %s WHERE customer_id = %s",(cid,city))
    print("Updated successfully\n")

def delete_customer(cursor):
    cid = get_int_input("Enter Customer_id to be deleted :")
    cursor.execute("DELETE FROM ecom_customer WHERE customer_id = %s",(cid))
    print("Deleted Successfully")
```

====================== For table Products ====================

[7]:

```python
def insert_products(cursor):
    Product_id = get_int_input("Product_ID :")
    name = get_str_input("Product Name :")
    price = get_float_input("Price :")
    stock = get_int_input("Stock :")
    cursor.execute("Insert into products Values(%s,%s,%s,%s)",(Product_id,name,price,stock))
    print("Product added successfully\n")


def read_products(cursor):
    cursor.execute("Select * from products")
    for row in cursor.fetchall():
        print(row)

def update_products(cursor):
    pid = get_int_input(" Enter Product_Id to be updated :")
    stock = get_str_input("New Stock :")
    cursor.execute("Update products SET stock = %s WHERE product_id = %s",(pid,stock))
    print("Updated successfully\n")

def delete_products(cursor):
    pid = get_int_input("Enter product_id to be deleted :")
    cursor.execute("DELETE FROM products WHERE product_id = %s",(pid))
    print("Deleted Successfully")
```

===================== For ecom_orders ======================

[8]:

```python
def insert_orders(cursor):
    order_id = get_int_input("Order_id :")
    customer_id = get_int_input("Customer_ID :")
    order_date = get_date_input("Date of Order (YYYY-MM-DD) :")
    cursor.execute("Insert into ecom_orders␣
 ↪Values(%s,%s,%s)",(order_id,customer_id,order_date))
    print("Customer added successfully\n")


def read_orders(cursor):
    cursor.execute("Select * from ecom_orders")
    for row in cursor.fetchall():
        print(row)

def update_orders(cursor):
    oid = get_int_input(" Enter Order_id to be updated :")
    date = get_date_input("New Date(YYYY-MM-DD) ")
    cursor.execute("Update ecom_orders SET date = %s WHERE order_id =␣
 ↪%s",(oid,date))
    print("Updated successfully\n")

def delete_orders(cursor):
    oid = get_int_input("Enter order_id to be delted :")
    cursor.execute("DELETE FROM ecom_orders WHERE order_id = %s",(oid))
    print("Deleted Successfully")
```

=============================== For table order_items =========================================

[9]:

```
def insert_order_items(cursor):
    o_item_id = get_int_input("Order_item_id :")
    oid = get_int_input("Order_id :")
    pid = get_int_input("Product_id :")
    quantity = get_int_input("Quantity:")
    cursor.execute("Insert into order_items␣
 ↪Values(%s,%s,%s,%s)",(o_item_id,oid,pid,quantity))
    print("Order Items added successfully\n")


def read_order_items(cursor):
    cursor.execute("Select * from order_items")
    for row in cursor.fetchall():
        print(row)

def update_order_items(cursor):
    otid = get_int_input(" Enter order_item_id to be updated :")
    qty = get_int_input("New Quantity :")
    cursor.execute("Update order_items SET quantity = %s WHERE order_item_id =␣
 ↪%s",(otid,qty))
    print("Updated successfully\n")

def delete_order_items(cursor):
    otid = get_int_input("Enter order_item_id to be deleted :")
    cursor.execute("DELETE FROM order_items WHERE order_item_id = %s",(otid))
    print("Deleted Successfully")
```

### 0.3.1 Table Menu System for insert, read, update and delete

Menu System

[10]:
```
def table_menu(cursor, conn, table_name):
    while True:
        print(f"\n\n {table_name.upper()} Table Operations")
        print("1. Insert")
        print("2. Read")
        print("3. Update")
        print("4. Delete")
        print("5. Back to Main Menu")
        choice = get_int_input("Enter operation to be done: \n")

        try:
            if choice == 1:
                if table_name == "ecom_customer":
                    insert_customer(cursor)
                elif table_name == "products":
                    insert_products(cursor)
                elif table_name == "ecom_orders":
                    insert_orders(cursor)
                elif table_name == "order_items":
                    insert_order_items(cursor)

            elif choice == 2:
                if table_name == "ecom_customer":
                    read_customer(cursor)
                elif table_name == "products":
                    read_products(cursor)
                elif table_name == "ecom_orders":
                    read_orders(cursor)
                elif table_name == "order_items":
                    read_order_items(cursor)

            elif choice == 3:
                if table_name == "ecom_customer":
                    update_customer(cursor)
                elif table_name == "products":
                    update_products(cursor)
                elif table_name == "ecom_orders":
                    update_orders(cursor)
                elif table_name == "order_items":
                    update_order_items(cursor)

            elif choice == 4:
                if table_name == "ecom_customer":
                    delete_customer(cursor)
                elif table_name == "products":
                    delete_products(cursor)
                elif table_name == "ecom_orders":
                    delete_orders(cursor)
                elif table_name == "order_items":
                    delete_order_items(cursor)

            elif choice == 5:
                print(" Returning to main menu...")
                break

            else:
                print(" Invalid choice! Please enter a number between 1 and 5.")

            conn.commit()

        except mysql.connector.Error as err:
            print(" Error:", err)
            conn.rollback()
```

Main program

[ ]:

```
def main():
    try:
        conn = connect_db()
        cursor = conn.cursor()


        while True:
            print("\n ----Main Menu-----\n")
            print("1. ecom_customers")
            print("2. products")
            print("3. ecom_orders")
            print("4. order_items")
            print("5. Exit")
            main_choice = get_int_input("Select Table \n")

            if main_choice == 1:
                table_menu(cursor, conn, "ecom_customer")
            elif main_choice == 2:
                table_menu(cursor, conn, "products")
            elif main_choice == 3:
```

```python
                table_menu(cursor, conn, "ecom_orders")
            elif main_choice == 4:
                table_menu(cursor, conn, "order_items")
            elif main_choice == 5:
                break
            else:
                print(" Invalid option!")

            cursor.close()
            conn.close()
            print("Existed successfully")

    except mysql.connector.Error as err:
        print("Error",err)

main()
```

```
 ----Main Menu-----

1. ecom_customers
2. products
3. ecom_orders
4. order_items
5. Exit
Select Table
2


 PRODUCTS Table Operations
1. Insert
2. Read
3. Update
4. Delete
5. Back to Main Menu
Enter operation to be done:
2
(201, 'Bluetooth Speaker', Decimal('1499.99'), 50)
(202, 'wireless mouse', Decimal('799.99'), 100)
(203, 'Gaming Keyboard', Decimal('2999.99'), 40)
(204, 'USB-C Charger', Decimal('899.50'), 150)
(205, 'HD Webcam', Decimal('2499.00'), 35)
(206, 'Noise Cancelling Headphones', Decimal('5499.75'), 20)


 PRODUCTS Table Operations
1. Insert
```

```
2. Read
3. Update
4. Delete
5. Back to Main Menu
Enter operation to be done:
5
 Returning to main menu…
Existed successfully

 ----Main Menu-----

1. ecom_customers
2. products
3. ecom_orders
4. order_items
5. Exit
```

```
[ ]:
```