ONLINE SHOPPING DATABASE

STUDENT NAME : Romil Moradiya

STUDENT ID : GH1041738

GITHUB LINK : <https://github.com/romil45/online_shopdtb>

Video link :

https://drive.google.com/file/d/1pG7rrUndIdH03iciUc0ug0C8kdByAmq5/view?usp=drive\_link

**INTRODUCTION**

I have opt online shopping database since every e- commerce website have this database and it is very vital for every companies to Handal database and it decide functionality for user and stockholder and it also help to check which product is popular for the costumer and companies take opportunity for booming their product As well as , transaction is important for every companies since it cost to them lost money if transaction fail . and it take to note for evidence for the transaction. For example amazon, when user opt one product and make the transaction so that time database check the quantity of product and then make the transaction complete otherwise user will be get notification for fail transaction so how this database work

In addition to these essential components the database also supports transaction handling, ensuring that operation are atomic,consistant ensure and ACID when a user place an order the database ensure the payment is recorded the stock is updates and the users is notified about the transaction status if any failure happens during the process the system can roll back for maintain data security

This project contains to understand operation including CRUD function data functionality, data normalization and efficient transaction management

**Why I have choose Database for online shopping?**

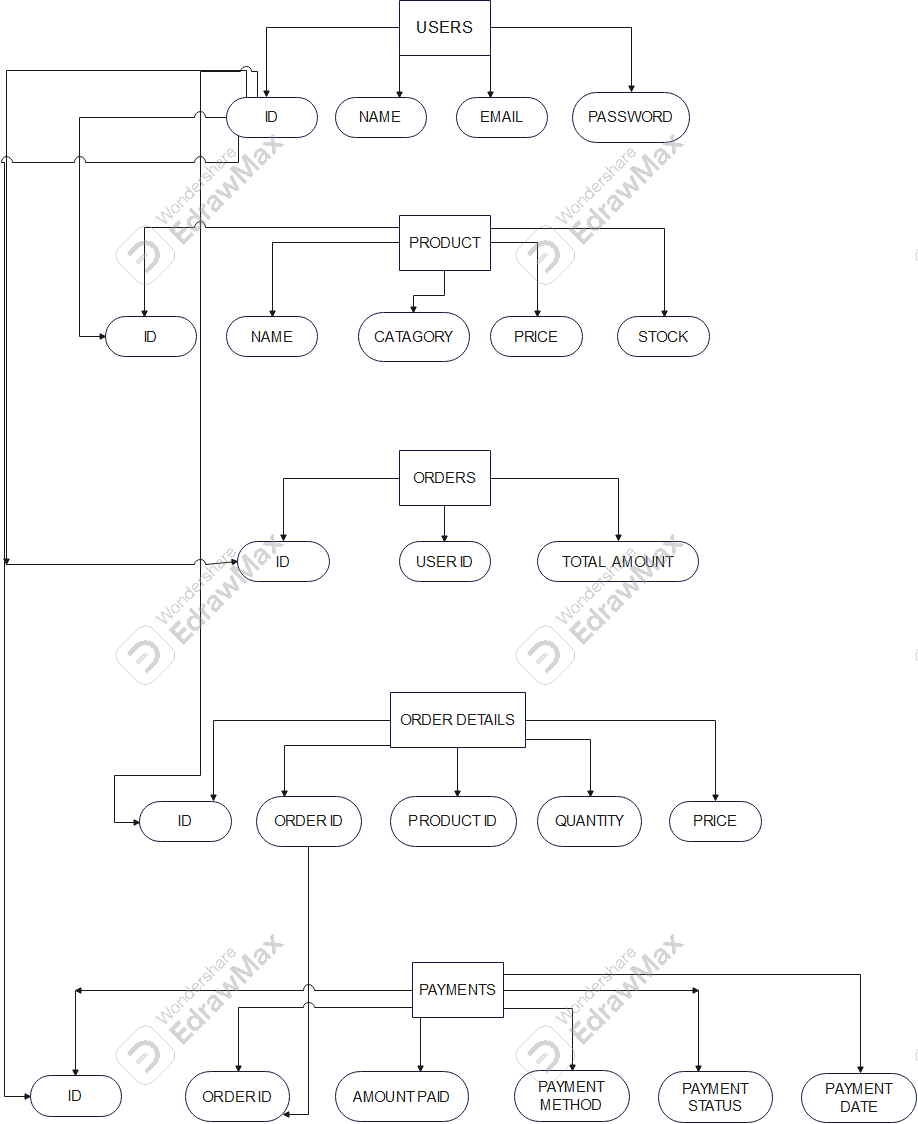
E commerce is best way to present project and learn as all the function will be used in this project such as a create, delete , and update followed by read which is required for the database and make the perfect database. Apart from that handling the user id and password also essential for the companies because it will be matter for the future if any kind query occur that time user id will help to identify this errors.

**Understand real world database requirement** : online shopping system needs complex data handling across multiple entities making this an excellent project for learning the core principle of databe design and management

**Implement core database operation:** the project gives permission to me implement the four fundamental operation create read update and delete which are necessary for managing dynamic data like product listing and user transaction

**Simulate real world challenges:** from handling out of stock situations to managing payment failure the project allowed to me to estimate the situation and issues to handle very wisely in databse

**ER DIAGRAM**

**** SOURCE OF ER DIAGRAM MAKE : <https://www.edrawsoft.com/>

1. USER TABELE

|  |  |
| --- | --- |
| ID | PRIMARY KEY |
| NAME | VARCHAR |
| EMAIL | VARCHAR |
| PASSWORD | VARCHAR |

1. PRODUCT TABLE

|  |  |
| --- | --- |
| ID | PRIMARY KEY |
| NAME | VARCHAR |
| CATEGORY | VARCHAR |
| PRICE | DECIMAL |
| STOCK | INTEGER |

3.ORDER TABLE

|  |  |
| --- | --- |
| ID | PRIMARY KEY |
| USER\_ID | VARCHAR , FORIGN KEY |
| TOTAL\_AMOUNT | DECIMAL |

4.ORDER DETAILS

|  |  |
| --- | --- |
| ID | PRIMARY KEY |
| ORDER\_ID | INTEGER , FORIGN KEY |
| PRODUCT\_ID | INTEGER, FORIGN KEY |
| QUANTITY | INTEGER |
| PRICE | INTEGER |

5.PAYMENTS

|  |  |
| --- | --- |
| ID | PRIMARY KEY |
| ORDER\_ID | INTEGER , FOREGIN KEY |
| AMOUNT\_PAID | DECIMAL |
| PAYMENT\_METHOD | VARCHAR |
| PAYMENT\_STATUS | VARCHAR |
| PAYMENT\_DATE | TIMESTAMP |

EXPLANATION

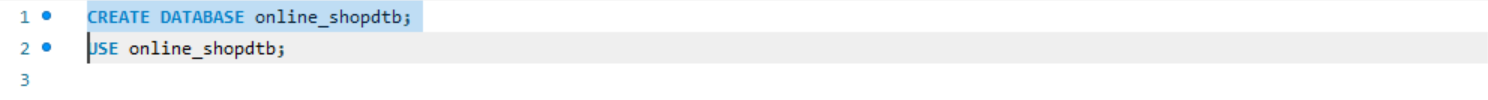
In this project I have sated primary key to the user id because user id is uniqe in this database and every table required user id in order to identify the recored so in this database I pin the primary key to user\_id .

So in USER table I have give the user id , name ,email and password , which will be submitted by the user that time user\_id will become the primary and unique

Product table I have created the id which will be primary key ,name of product , category, price ,stock now there are three table for the product , price and stock

SCREENSHOT OF THE PROJECT

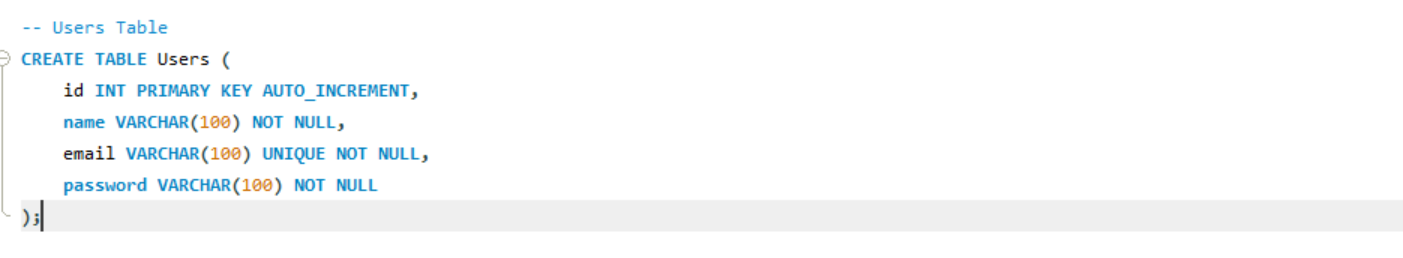
1.CREATE DATABSE



OUTPUT



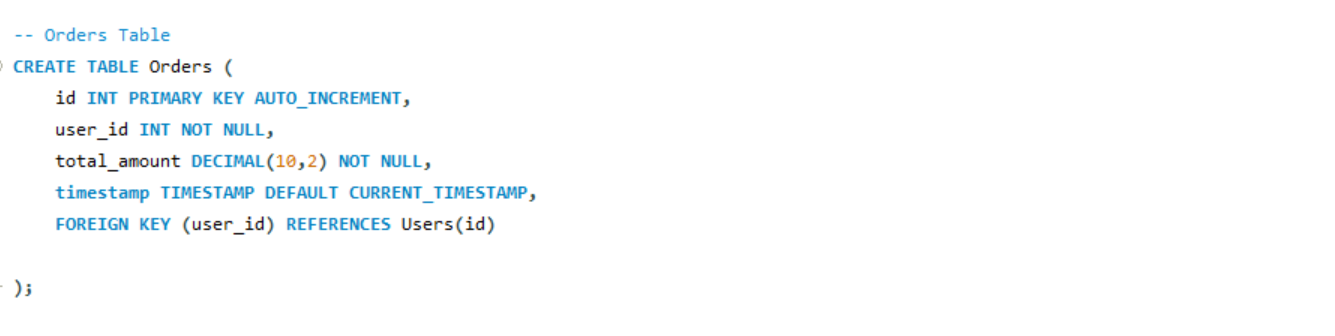
2.USER TABLE



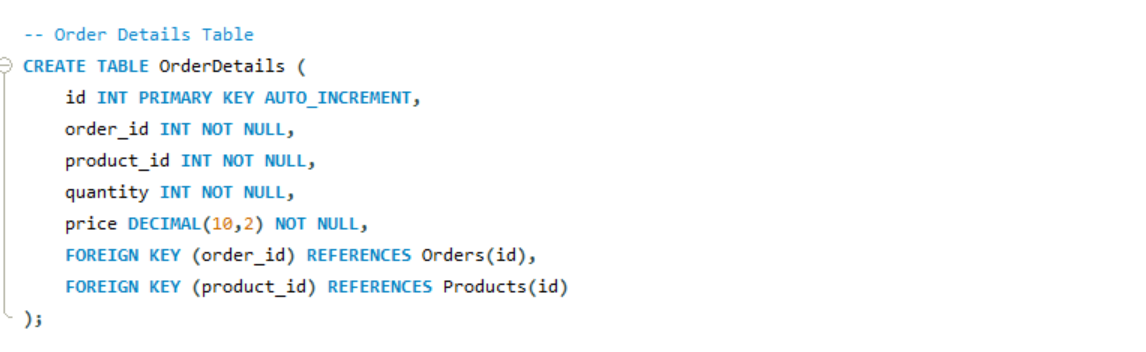
3.PRODUCT TABLE



4.ORDER TABLE



5.ORDER DETAILS TABLE



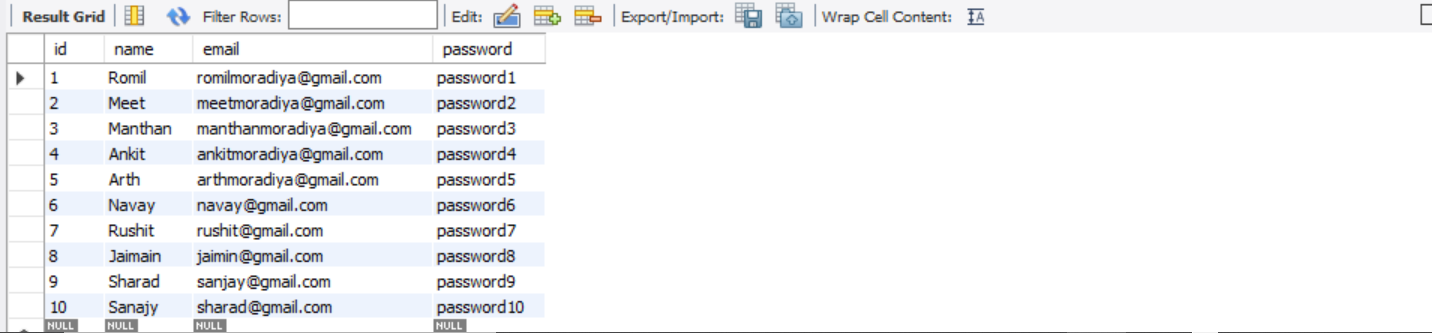
6.PAYMENT TABLE



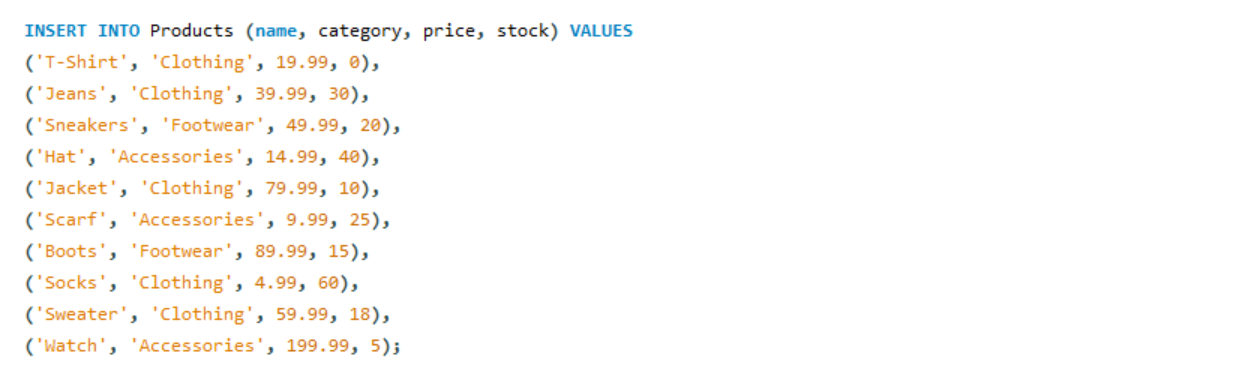
7.INSERT USER DATA



OUTPUT



8.INSERT PRODUCT DATA



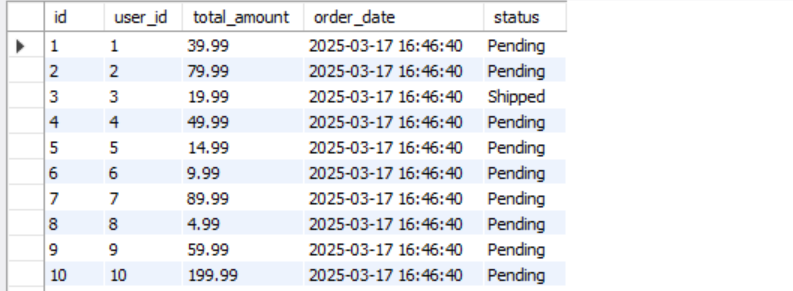
OUTPUT

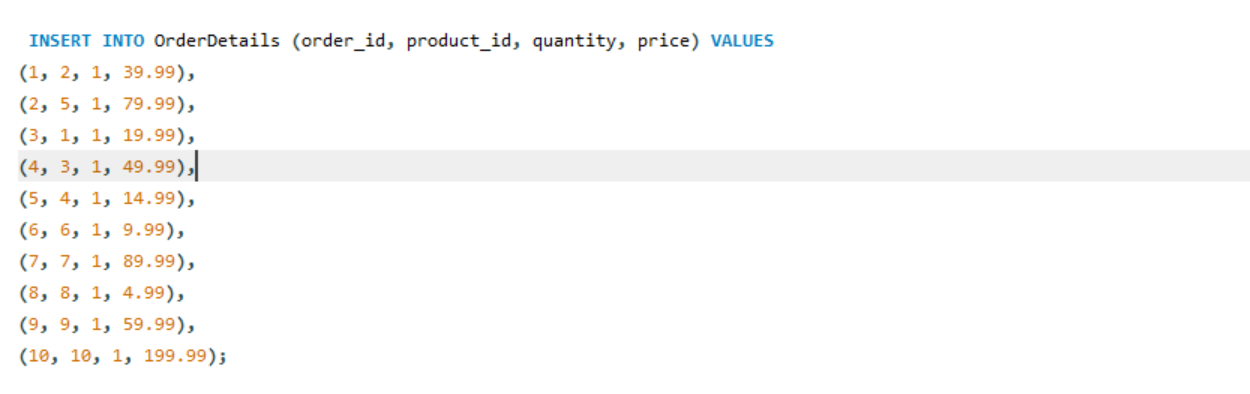


9.INSERT ORDERS DATA



OUTPUT

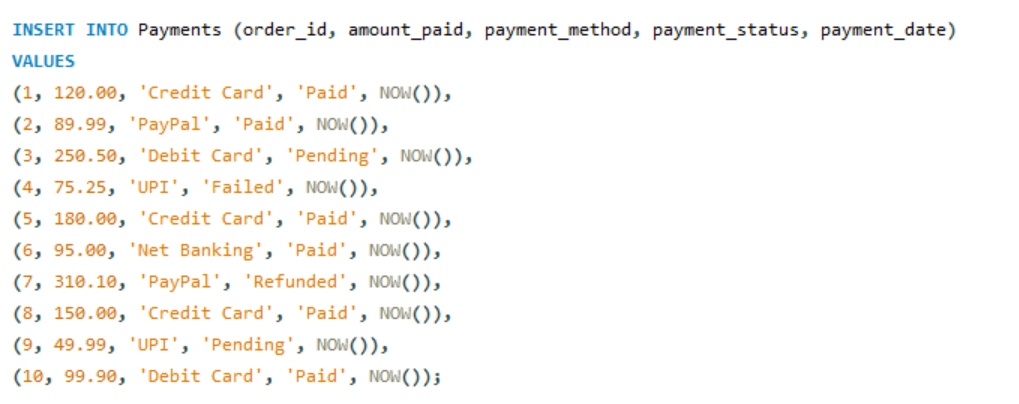


10.INSERT ORDERDETAILS

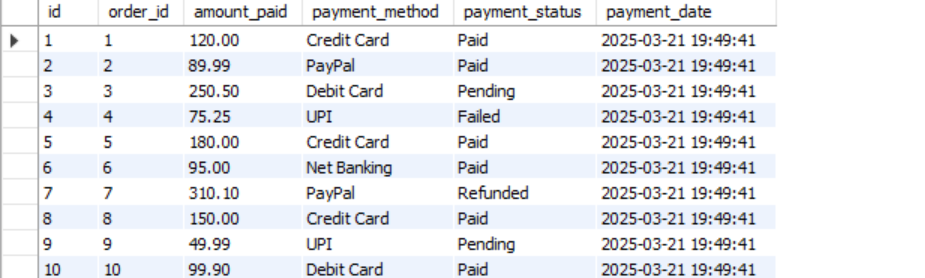
OUTPUT



11. INSERT PAYMENT DATA

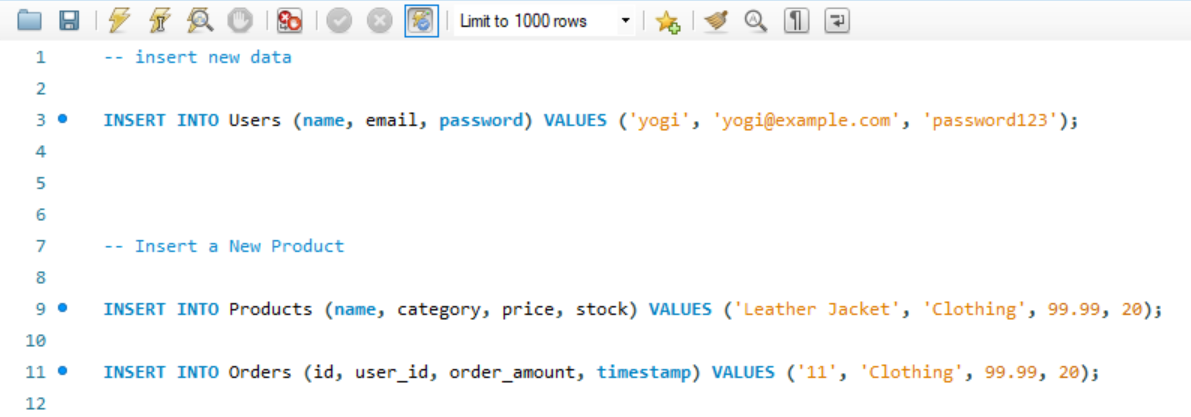


OUTPUT



CRUD OPRATION

1. CREATE DATABASE

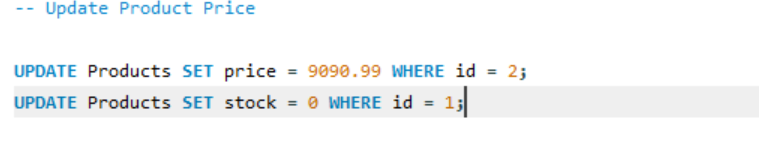


OUTPUT





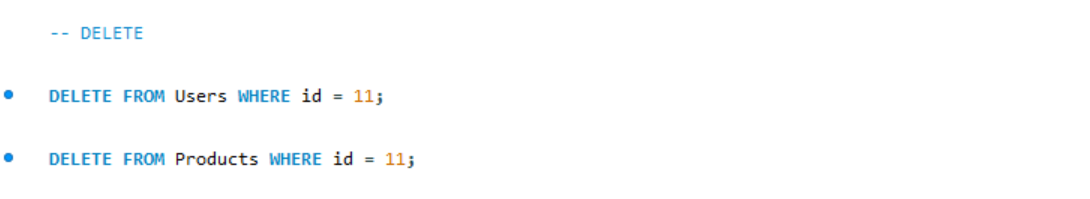
2.UPDATE



OUTPUT



3.DELETE



NOTE: I have not run this query since it more likely to be delete user and product However this query is runnable

**CHALLENGES IN TRANSACTION METHOD**

**1.Challenges**: transaction in e commerce systems often affect multiple related tables ensure that consistency across all the table when transaction is completed either can be fails when I was implementing the transaction method without ACID properties , I faced plenty of errors and it take more time to complete transaction and it had to do manually

**Solution:** so I have used the ACID function [atomicity , consistency , isolation , durability ] to ensure that each transaction is processed in a way that prevents partial updates for example if an order is place it could be finalized when both payment and stock update are completed successfully otherwise the entire transaction is rolled back

**Handling COMMIT and ROLLBACK correctly**

1. **Challenges:** one of the most vital factor of the transaction management is that commit or rollback changes. If a commit is issue early , changes can become parament even if an error occurs later in the process. In my project COMMIT was not working well since I have put wrong query in update stock as a result first step work well but when it comes to the ROLLBACK had given the errors for the stock

**Solution: I** was not sure that I I have use right squecnce of the COMMIT and ROLLBACK . I used propre exception handling to detect any issues during the transaction process and ensured that ROLLBACK was trigger when errors show to the code