



Database Design for Online Shopping Platform (SHEIN - Beauty Items).

Prepared By: Mayar Yousef Al-Saeed

ID: 461210731

Course Name: CYS131.



PROJECT OVERVIEW & OBJECTIVES

- **Objective** : To design and implement a robust relational database specifically modeled after SHEIN Beauty Section.
- **Scope** : The system focuses on managing inventory for beauty categories: Skincare, Makeup, Haircare, and Fragrances.
- **Goal** : To simulate backend operations like tracking products, processing orders, and managing user reviews efficiently.

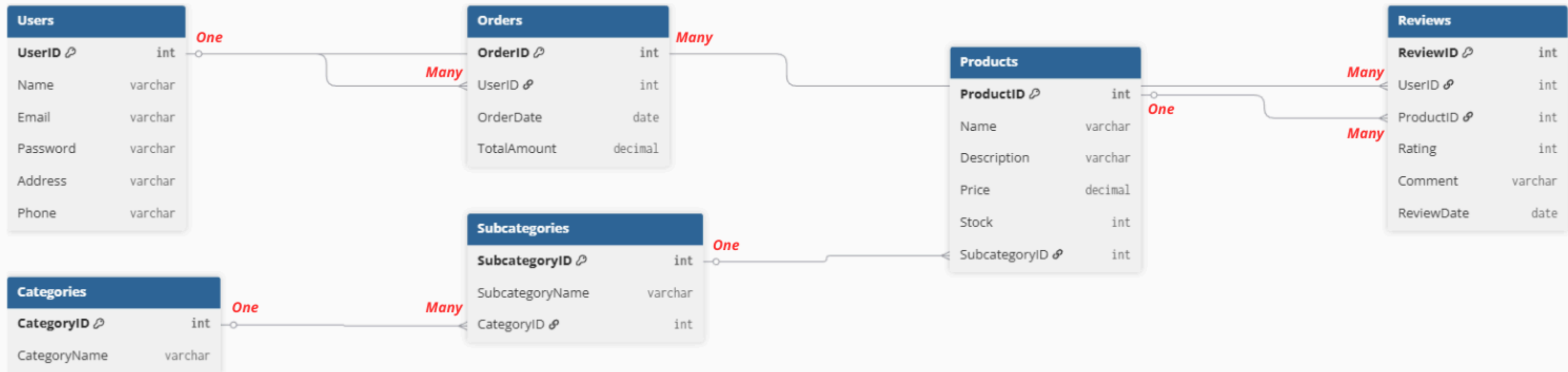
SYSTEM DESIGN (ENTITIES)

The database consists of 6 Core Entities:

1. **Users** : Stores customer information (Name , Email, Address).
2. **Products** : Manages item details (Price, Stock, Description).
3. **Orders** : Tracks transaction details (Total Amount, Date).
4. **Categories** : Represents major product groups (e.g., Skincare, Makeup).
5. **Subcategories**: Defines specific product types within categories(e.g., Cleansers, Lipsticks).
6. **Reviews**: Captures user ratings and feedback.

ER DIAGRAM

- Visual representation of database schema and relationships.
- Relationships:
 1. Users → Orders : One to Many.
 2. Orders → Products : Many to Many.
 3. Categories → Subcategories : One to Many.
 4. Subcategories → Products : One to Many.
 5. Users → Reviews : One to Many.



IMPLEMENTATION TOOLS

- **DBMS** : Oracle Database 19c (Selected for high performance and security).
- **Language** : SQL (Structured Query Language) used for :
 1. **DDL** : Creating tables with Primary & Foreign keys.
 2. **DML** : Data manipulation and complex querying.

SQL Plus

SQL*Plus: Release 23.0.0.0.0 - Production on Wed Dec 10 16:02:14 2025
Version 23.9.0.25.07

Copyright (c) 1982, 2025, Oracle. All rights reserved.

Enter user-name: sys as sysdba
Enter password:

Connected to:
Oracle Database 23ai Free Release 23.0.0.0.0 - Develop, Learn, and Run for Free
Version 23.9.0.25.07

SQL> create table Users(UserId number(20) primary key, username varchar2(20),emaile varchar2(20),password number(20),adress varchar2(20),phone number(20));

Table created.

SQL> create table categories(categoryid number(20) primary key,CategoryName varchar2(20));

Table created.

SQL> create table subcategories(subcategoryid number(20) primary key, subcategoryname varchar2(20), categoryid number(20),constraint sub_fk foreign key (categoryid) references categories(categoryid));

Table created.



```
SQL> create table products(productid number(20) primary key,productname varchar2(20),description varchar2(100),price number(20),stock number(20),Subcategory  
id number(20), constraint p_fk foreign key (subcategoryid) references subcategories(subcategoryid));
```

Table created.

```
SQL> create table orders(orderid number(20) primary key,orderdate date, totalamount number(20),userid number(20), constraint or_fk foreign key (userid) refe  
rences users(userid));
```

Table created.

```
SQL> create table reviews(reviewid number(20) primary key,rating number(10),comments varchar2(100),reviewdate date,userid number(20),productid number(20) ,c  
onstraint ru_fk foreign key (userid) references users(userid),constraint re_fk foreign key (productid) references products(productid));
```

Table created.

ADVANCED QUERIES

We implemented complex SQL queries to extract meaningful insights:

- ***Query 1 : category filtering:***
 - Listing all products under the “ Skincare” category using **Join** across 3 tables.
- ***Query 2 : Data analysis (Subquery):***
 - Identifying Top Rated Products (Rating>Average) in the “ Makeup” section.
 - This demonstrates the use of nested queries and aggregate functions(AVG).

QUERY RESULTS

- ***Skincare Products Result*** : Show accurate retrieval of skincare items like “ Hydrate cleanser”.
- ***Top Rated Product Result*** : Successfully identified “Matt velvet lipstick” (Rating 5) as top product in Makeup category.

```
SQL> SELECT p.PRODUCTNAME, p.Price FROM Products p JOIN Subcategories s ON p.SubcategoryID = s.SubcategoryID JOIN Categories c ON s.CategoryID = c.CategoryID WHERE c.CategoryName = 'Skincare';
```

PRODUCTNAME	PRICE
-------------	-------

Hydrate cleanser	110
Charcoal peel mask	17

```
SQL> SELECT p.PRODUCTNAME, r.Rating FROM Products p JOIN Reviews r ON p.ProductID = r.ProductID JOIN Subcategories s ON p.SubcategoryID = s.SubcategoryID JOIN Categories c ON s.CategoryID = c.CategoryID WHERE c.CategoryName = 'Makeup' AND r.Rating > (SELECT AVG(r2.Rating) FROM Reviews r2 JOIN Products p2 ON r2.ProductID = p2.ProductID JOIN Subcategories s2 ON p2.SubcategoryID = s2.SubcategoryID JOIN Categories c2 ON s2.CategoryID = c2.CategoryID WHERE c2.CategoryName = 'Makeup');
```

PRODUCTNAME	RATING
-----	-----
matt velvet lipstick	5

CONCLUSION

- **Summary** :Successfully designed and implemented a functional database for SHEIN Beauty.
- **Achievements** :
 1. Built a normalized schema with proper integrity constraints.
 2. Executed SQL queries to solve real world business questions (e.g., finding best-selling items).
- **Future Scope** : The system is scalable and ready for future enhancements