**1. Introduction**

The E-commerce Database Project is a structured attempt to model the core operations of an online shopping platform using SQL. E-commerce is a fast-growing industry, and database management plays a crucial role in ensuring efficient handling of customers, products, and orders. The goal of this project is to simulate a simplified e-commerce environment that demonstrates how data is stored, retrieved, and manipulated in a relational database.

The project involves designing relational tables, establishing relationships using foreign keys, inserting sample data, and applying SQL operations to perform various business functions. Additionally, it incorporates advanced features such as triggers and stored procedures to automate tasks and enhance efficiency. Through this project, learners can understand the fundamentals of relational databases, Structured Query Language (SQL), and their application in real-world e-commerce systems.

**2. Database Design**

The database is structured around four primary tables, each of which plays a key role in maintaining the integrity and functionality of the system.

1. **Users Table**
   * Attributes: user\_id, name, email
   * Purpose: Stores details of customers using the platform.
   * Example: A record for a user named *Saranya* with email *saranya@gmail.com*.
2. **Products Table**
   * Attributes: product\_id, name, price, stock
   * Purpose: Contains the catalog of products available for purchase.
   * Example: A laptop priced at ₹50,000 with 10 units in stock.
3. **Orders Table**
   * Attributes: order\_id, user\_id, order\_date
   * Purpose: Records purchases made by customers and links them to users.
4. **Order\_Items Table**
   * Attributes: order\_item\_id, order\_id, product\_id, quantity
   * Purpose: Stores detailed order information, connecting orders with specific products and quantities purchased.

These four tables are connected using **primary keys** and **foreign keys**, ensuring **referential integrity**. For example, every order must be associated with a valid user, and every order item must link to an existing product.

**3. Data Insertion and SQL Operations**

After designing the database schema, the project demonstrates data insertion and SQL operations that mimic real e-commerce activities.

* **Data Insertion:** Sample records are added to the Users, Products, Orders, and Order\_Items tables. This helps simulate real-world interactions such as placing orders and updating stock.
* **Update Operation:** For example, updating a user’s email address from *saranya@gmail.com* to *new\_saranya@gmail.com*.
* **Delete Operation:** Products can be removed from the catalog, such as deleting headphones from the product list.
* **WHERE and LIKE Clauses:** These help in filtering data, such as retrieving all users with Gmail accounts or finding all products containing the letter “o” in their name.
* **Aggregate Functions (COUNT, SUM, AVG):** Used to generate business insights, such as calculating the total number of products, total stock, and average product price.
* **GROUP BY and HAVING Clauses:** Allow analysis of sales trends, such as identifying the total quantity ordered for each product and filtering products ordered more than once.
* **Subqueries:** Useful for retrieving advanced insights, such as listing users who have placed at least one order or finding products priced above the average.

Together, these operations illustrate how SQL can be used not only for storing data but also for extracting meaningful information that supports business decision-making.

**4. Advanced Features**

Beyond basic SQL operations, this project integrates advanced functionalities to simulate automation and real-world complexity.

**4.1 Triggers**

A trigger named **reduce\_stock** is implemented to automatically update stock levels when an order is placed. This ensures that whenever a product is purchased, its stock reduces by the ordered quantity, eliminating the need for manual updates.

**4.2 Stored Procedures**

The stored procedure **place\_order** simplifies the process of creating new orders. Instead of executing multiple SQL statements separately, this procedure automates the workflow by:

1. Creating a new order for a given user.
2. Inserting corresponding order items.
3. Automatically updating stock through the trigger.

This feature enhances efficiency and mimics the order placement process in real-world e-commerce systems.

**5. Query Demonstrations**

Several queries were executed to test the database functionality and retrieve insights:

* **Retrieve all users with Gmail accounts.**
* **Calculate total stock and average product price.**
* **Identify products ordered more than once.**
* **Find products with the letter “o” in their names.**
* **Check updated stock levels after an order is placed.**
* **View detailed customer orders** by joining Orders and Users tables.

These queries highlight how the database supports both operational tasks and analytical reporting.

**6. Conclusion**

This e-commerce database project successfully demonstrates the application of SQL in managing an online shopping platform. The project includes database design, data insertion, CRUD operations, filtering queries, aggregate functions, triggers, and stored procedures. Together, these features replicate the core functionalities of a real-world e-commerce system in a simplified manner.

The project highlights the importance of relational databases in business applications, especially in areas like inventory management, order tracking, and customer data management. By integrating both fundamental and advanced SQL techniques, this project provides a strong foundation for learners and practitioners aiming to work with databases in industry environments.