

**Project:** **E –Commerce Management System**

**Database Management System**

Submitted To:

Sir Adeeb ur Rehman

Submitted By:

* Musahrib Rehman(032)
* Abdullah Arsalan(305)
* Mohid Arsalan(307)
* Ali Jamil(147)

Submission Date: 14/05/2024

E –Commerce Management System

# INTRODUCTION

An **E-Commerce Management System** is a powerful solution designed to facilitate online retail operations. It serves as the backbone for managing various aspects of an e-commerce platform, including customer interactions, product management, order processing, and payment handling.

In today's digital age, e-commerce has revolutionized the way businesses operate and customers shop. With the rise of online shopping, managing vast amounts of data efficiently has become crucial for businesses to stay competitive.

This project aims to design and develop a comprehensive e-commerce database system that can effectively manage customer information, product catalogs, orders, and payments. By leveraging the power of database technology, this system will provide a scalable, secure, and efficient solution for businesses to streamline their online operations and enhance customer experiences.

The system will enable businesses to track inventory levels, process transactions, and analyze sales trends, ultimately leading to improved decision-making and increased revenue. Furthermore, the system will provide a user-friendly interface for customers to browse and purchase products, making online shopping a seamless and enjoyable experience. By developing a robust and reliable e-commerce database system, this project aims to contribute to the growth and success of online businesses in the digital marketplace.

# Components of the E-Commerce Management System

The E-Commerce Management System is composed of several key components, each designed to manage a specific aspect of the e-commerce operation. These components include:

* **Customers**: Manage customer information, including contact details, addresses, and registration dates.
* **Products**: Maintain product details, including product names, prices, quantities, brands, and descriptions.
* **Categories**: Organize products into categories for easy navigation and management.
* **Sellers**: Manage information about sellers, including their contact details and sales performance.
* **Orders**: Track orders from customers, including order dates, shipping dates, order amounts, and order statuses.
* **Order Items**: Manage the individual items within each order, including item prices and quantities.
* **Payments**: Handle payment transactions, including payment amounts, methods, statuses, and payment dates.
* **Reviews**: Manage customer reviews and ratings for products, providing valuable feedback to the business.

Each component is implemented as a table in the SQL Server database, with relationships established between the tables to ensure data integrity and consistency. For example, the Orders table references the Customers table to link each order to a specific customer, and the Order Items table references the Orders table to link each item to a specific order.

# NEED:

As E-commerce continues to expand globally, businesses require efficient systems to manage various aspects of their operations. Here, we will delve into the essential needs that an E-Commerce Management System fulfills, highlighting its importance in modern business environments.

## **1. Efficient Product Management:**

One of the primary needs of an e-commerce business is to manage its products efficiently. This includes adding new products, updating existing ones, managing inventory levels, and categorizing products appropriately. An E-Commerce Management System provides a centralized platform where businesses can handle all aspects of product management with ease.

## **2. Customer Management**

Managing customer information is crucial for providing personalized services and maintaining customer satisfaction. An E-Commerce Management System stores detailed customer data, including contact information. Access to detailed customer information helps in building and maintaining strong relationships with customers.

## **3. Order Processing and Management**

Efficient order processing is vital for ensuring timely deliveries and customer satisfaction. An E-Commerce Management System manages the entire order lifecycle, from order placement to delivery. Allows customers to track their orders, enhancing their shopping experience.

## **4. Payment Processing**

Handling payments securely and efficiently is a critical need for any e-commerce business. An E-Commerce Management System integrates with various payment gateways, allowing customers to pay using their preferred methods. Provides customers with a variety of payment options.

## **5. Shipping Management**

Managing shipping is essential for ensuring that products reach customers on time. An E-Commerce Management System helps businesses coordinate with shipping carriers, calculate shipping costs, and manage delivery schedules.

# Procedure:

### **Database Creation and Setup**

1. **Create and Select Database:**
   * Create a database named E\_commerce.
   * Select the E\_commerce database for use.

### **Table Creation:**

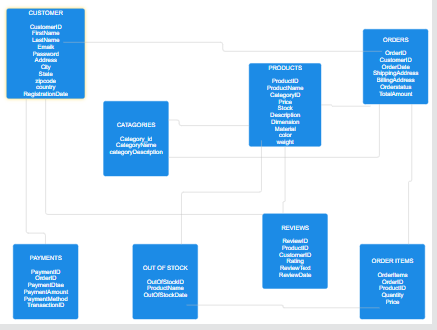
1. **Customers Table:**
   * Store customer details with columns like customer\_id, first\_name, last\_name, Phone\_number, Email, Address, Country, city, and R\_date.
2. **Product Table:**
   * Store product details with columns like product\_id, Seller\_id, Category\_id, product\_name, price, Quantity, Brand\_name, and Description.
   * Add foreign key constraints for Seller\_id and Category\_id.
3. **Category Table:**
   * Store product categories with columns like Category\_id , Category\_name, and description.
   * Add a foreign key constraint for product\_id.
4. **Orders Table:**
   * Store order details with columns like Order\_id, customer\_id, cart\_id, Shipping\_date, order\_date, Order\_amount, delivery\_charges, shipping\_address, discount\_amount, Order\_tracking, Order\_Status, and Order\_trackingNum.
   * Add foreign key constraints for customer\_id and cart\_id.
5. **Order Item Table:**
   * Store order item details with columns like item\_ID, Order\_id, Product\_id, Item\_Price, and Quantity.
   * Add foreign key constraints for Order\_id and Product\_id.
6. **Payments Table:**
   * Store payment details with columns like payment\_id, customer\_id, order\_id, payment\_date, payment\_amount, payment\_method, and payment\_status.
   * Add foreign key constraints for customer\_id and order\_id.
7. **Reviews Table:**
   * Store product reviews with columns like review\_id, customer\_id, product\_id, Review\_description, and Rating.
   * Add foreign key constraints for customer\_id and product\_id.
8. **OutOfStock Table:**
   * Store sometimes et out of stock on products that are seeling on high demand or its shortage in the market. Its uses column of OutOfStockID, ProductID, ProductName and OutOfStockDate
   * Add foreign key constraints for customer\_id and product\_id.

### **Data Insertion:**

**Insert Data into All 8 Table**

* + Insert review records with details for each table.

### **ER Diagram:**



1

### **Queries:**

**-- 1. Calculate value of each order and total value of all orders**

SELECT

o.OrderID,

CONCAT(c.FirstName, ' ', c.LastName) AS CustomerName,

o.OrderDate,

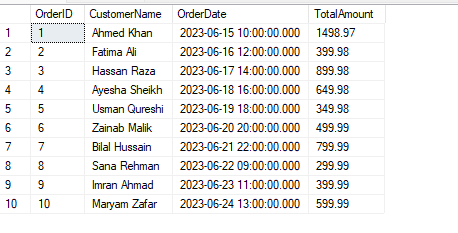
o.TotalAmount

FROM

Orders o

JOIN

Customers c ON o.CustomerID = c.CustomerID;



**-- 2. Total value of all orders.**

SELECT

SUM(TotalAmount) AS TotalValueOfAllOrders

FROM

Orders;

A close-up of a computer screen

Description automatically generated

**-- 3. Query to show data for a specific customer by CustomerID**

SELECT

CustomerID,

FirstName,

LastName,

Email,

PhoneNumber,

AddressLine1,

AddressLine2,

City,

State,

ZipCode,

Country,

RegistrationDate

FROM

Customers

WHERE

CustomerID = 1;



**-- 4. Query to show data for a specific order by OrderID.**

SELECT

o.OrderID,

o.CustomerID,

CONCAT(c.FirstName, ' ', c.LastName) AS CustomerName,

o.OrderDate,

o.ShippingAddress,

o.BillingAddress,

o.OrderStatus,

o.TotalAmount

FROM

Orders o

JOIN

Customers c ON o.CustomerID = c.CustomerID

WHERE

o.OrderID = 5;



**-- 5. Query to show all orders placed by a specific customer by CustomerID.**

SELECT

o.OrderID,

o.OrderDate,

o.ShippingAddress,

o.BillingAddress,

o.OrderStatus,

o.TotalAmount

FROM

Orders o

WHERE

o.CustomerID = 4;

A screenshot of a computer

Description automatically generated

**-- 6. Query to show detailed information for a specific order including order items.**

SELECT

o.OrderID,

CONCAT(c.FirstName, ' ', c.LastName) AS CustomerName,

o.OrderDate,

o.ShippingAddress,

o.BillingAddress,

o.OrderStatus,

o.TotalAmount,

oi.OrderItemID,

p.ProductName,

oi.Quantity,

oi.Price

FROM

Orders o

JOIN

Customers c ON o.CustomerID = c.CustomerID

JOIN

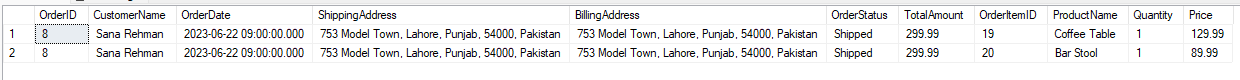
OrderItems oi ON o.OrderID = oi.OrderID

JOIN

Products p ON oi.ProductID = p.ProductID

WHERE

o.OrderID = 34;



**-- 7. Query to list the total number of orders and total order amount for each customer.**

SELECT

c.CustomerID,

c.FirstName,

c.LastName,

COUNT(o.OrderID) AS TotalOrders,

SUM(oi.Quantity \* oi.Price) AS TotalOrderAmount

FROM

Customers c

LEFT JOIN

Orders o ON c.CustomerID = o.CustomerID

LEFT JOIN

OrderItems oi ON o.OrderID = oi.OrderID

GROUP BY

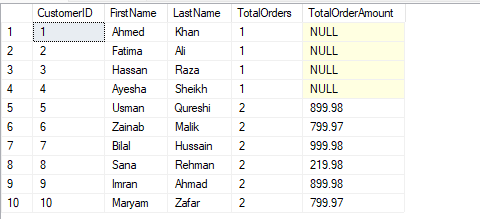
c.CustomerID,

c.FirstName,

c.LastName

ORDER BY

c.CustomerID;



**-- 8. Find All Products with Less Than a Specified Quantity in Stock.**

SELECT

ProductID,

ProductName,

Stock

FROM

Products

WHERE

Stock < 5;

A screenshot of a computer

Description automatically generated

**-- 9. List All Pending Orders with Customer and Shipping Details.**

SELECT

o.OrderID,

o.OrderDate,

o.ShippingAddress,

c.CustomerID,

c.FirstName,

c.LastName,

c.Email,

c.PhoneNumber,

c.AddressLine1,

c.AddressLine2,

c.City,

c.State,

c.ZipCode,

c.Country

FROM

Orders o

JOIN

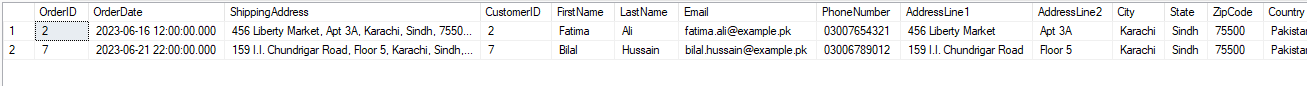
Customers c ON o.CustomerID = c.CustomerID

WHERE

o.OrderStatus = 'Pending'

ORDER BY

o.OrderDate;



**-- 10. Retrieve All Orders Made by a Specific Customer.**

SELECT

o.OrderID,

o.OrderDate,

o.ShippingAddress,

o.BillingAddress,

o.OrderStatus,

o.TotalAmount

FROM

Orders o

WHERE

o.CustomerID = 7

ORDER BY

o.OrderDate;

A screenshot of a computer

Description automatically generated

**-- 11. Query to show customers that gave the reviews.**

SELECT

c.CustomerID,

c.FirstName,

c.LastName,

r.ReviewID,

p.ProductName,

r.ReviewText AS Comment,

r.ReviewDate

FROM

Reviews r

JOIN

Customers c ON r.CustomerID = c.CustomerID

JOIN

Products p ON r.ProductID = p.ProductID

ORDER BY

r.ReviewDate DESC;

A table with text on it

Description automatically generated

**-- 12. Top 3 Product Sold.**

SELECT TOP 1

p.ProductID,

p.ProductName,

SUM(oi.Quantity) AS TotalQuantitySold

FROM

OrderItems oi

JOIN

Products p ON oi.ProductID = p.ProductID

GROUP BY

p.ProductID,

p.ProductName

ORDER BY

TotalQuantitySold DESC;



This summary captures the key steps and SQL operations needed to create and manage an E-commerce management system database.

# Benefits:

**Improved Customer Experience:** E-commerce systems often come with user-friendly interfaces, making it easy for customers to find products, compare prices, and complete purchases. Customers can track their orders in real-time, which enhances their shopping experience and builds trust in the brand.

**Cost Saving:** By reducing manual workload and minimizing errors, the system helps businesses save costs and allocate resources more effectively.

**Improve Security:** The e-commerce database system ensures the security and integrity of customer data, protecting sensitive information from unauthorized access.

**Multiple Payment Options:** Customers can choose from various payment methods, including credit cards, debit cards, and even cash on delivery in some cases.

### **Inventory Management:** E-commerce systems provide inventory tracking, helping businesses manage stock levels efficiently and avoid overstocking or stock outs.

### **24/7 Availability:** An E-commerce platform allows customers to shop anytime, anywhere. This round-the-clock availability enhances customer convenience and can lead to increased sales. Unlike traditional stores, which are limited by physical opening hours, an online store can operate 24/7 ,offering products and services at all times.

# Area of people:

#### **Customer**: Searches for products or services of interest. Adds items to the shopping cart and proceeds to checkout. Provides accurate billing and shipping information.

**SQL Developer:** Writes and optimizes SQL queries to retrieve and manipulate data. Develops stored procedures, triggers, and views to support application functionality.

#### **Shipping Partner:** Manages the shipping and delivery of products ordered through the e-commerce platform. Coordinates with sellers to fulfill orders and schedule pickups. Tracks shipments in and provides updates to customers.

# Conclusion:

Using Microsoft SQL Server to create and manage the database for an e-commerce system is a reliable choice. The project showcases the creation of various essential tables such as Customers, Products, Orders, Categories, Sellers, Payments, and Reviews. Each of these tables is meticulously designed to store relevant data, ensuring the integrity and organization of information. For instance, the Customers table holds personal and contact information, while the Products table includes details about each product available for sale. The provided SQL queries illustrate the powerful querying capabilities of Microsoft SQL Server. These queries enable the extraction of valuable information from the stored data. For instance, queries are used to retrieve order details, calculate total quantities of products sold, identify top-selling brands, and find orders with amounts greater than the average.

In conclusion, an E-commerce management system developed using Microsoft SQL Server offers numerous advantages for online businesses. The database creation and management capabilities ensure organized and efficient data storage. The structured data design supports complex queries providing valuable business insights. Comprehensive data handling and improving efficiency and customer satisfaction. Strong security measures protect sensitive information, while the system's scalability and flexibility support long-term growth.

# One of the most significant advantages of an e-commerce management system is its ability to integrate various business processes into a single platform. This integration means that inventory, sales, marketing, customer service, and financial data can all be accessed and managed from one central location. E-commerce management systems help improve the customer experience by providing features such as personalized shopping experiences, efficient order processing, and effective customer service.