

**Department of Computer Science and Engineering Course Code: CSE-209**

**Database Management Systems II B.Tech - 4th Semester**

**eCommerce website**

|  |  |
| --- | --- |
| **Registration Number** | **Student Name** |
| AP22110010068 | Velaga Subhash |
| AP22110010069 | S. Aditya Raj |
| AP22110010093 | G. Satya Krishna |
| AP22110010105 | Yarra Krishna Teja |
| AP22110010122 | P. Chinni Krishna |

TABLE OF CONTENTS:

[AIM… 3](#_TOC_250005)

[INTRODUCTION… 4](#_TOC_250004)

[PROJECT DESCRIPTION 5](#_TOC_250003)

LIST OF ENTITES AND ATTRIBUTES… 6

[ER DIAGRAM 7](#_TOC_250002)

Description of ER diagram … . 8

CONVERSION OF ER DIAGRAMS TO TABLES 9

Creation of Tables………………………………………………… 10

Description of Tables ……………………………………….. 12

[NORMALIZATION](#_TOC_250001) 14

Creation of Data into the Tables……………………………..15

Queries on the tables……………………………………………...18

Creation of views using the tables………………………..20

[CONCLUSION ..2](#_TOC_250000)3

# AIM:

The aim of this project is to design and implement a comprehensive and scalable e-commerce database system that efficiently manages product information, customer details, transaction data, inventory management, and sales tracking.

The goal is to ensure seamless operations, enhance user experience, provide valuable insights for business growth, and enable effective inventory and sales management.

This project contains Introduction to the e-Commerce website. It is the computerized system for online shopping. It is mainly used to deliver the user requirements at door.

This project contains entity relationship model diagram based on e-Commerce website and relation model.

# INTRODUCTION:

Database is an organized collection of data. The data is typically organized to model aspects of reality in a way that supports processes requiring information. A DBMS makes it possible for end users to create, read, update, and delete data in a database. The DBMS essentially serves as an interface between the database and end users or application programs, ensuring that data is consistently organized and remains easily accessible.

The DBMS manages three important things: the data, the database engine that allows data to be accessed, locked, and modified and the database schema, which defines the database’s logical structure. These three foundational elements help provide concurrency, security, data integrity and uniform administration procedures. The DBMS can offer both logical and physical data independence. That means it can protect users and applications from needing to know where data is stored or having to be concerned about changes to the physical structure of data.

The primary objective of maintaining a database for an eCommerce website is to streamline the management of product inventory, customer information, and transaction details. This makes it easier for both the customers and the business to track orders, update product listings, and handle customer service inquiries.

Automation of these processes eliminates many of the inefficiencies and errors that can occur with manual record-keeping. This leads to improved customer satisfaction, more efficient operations, and ultimately, increased sales and profitability for the business.

# PROJECT DESCRIPTION:

This project is about creating a database about e-Commerce website.

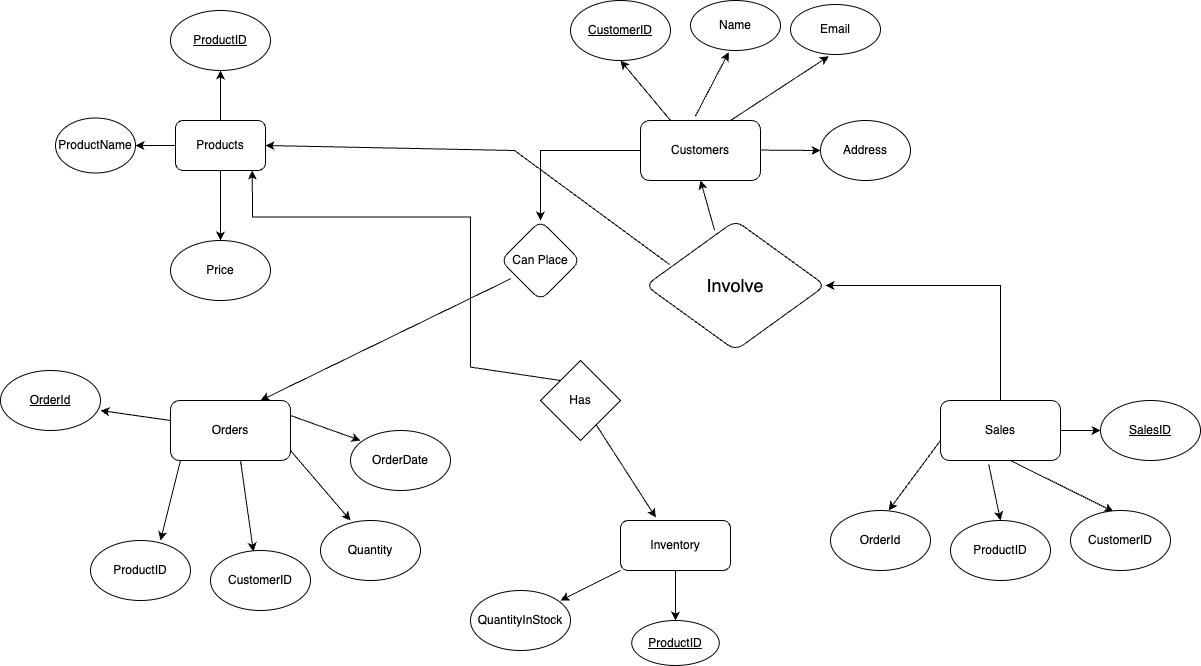
The eCommerce website is designed to provide a seamless shopping experience for customers and an efficient management system for the business. The database is structured to handle various aspects of the online store, including product listings, customer information, and transaction records.

This database design enables the eCommerce website to efficiently manage its operations, from updating product listings to processing orders and tracking sales. It also provides valuable data for analyzing sales trends and customer behavior, which can inform business strategies and decisions. The automation of these processes also enhances the shopping experience for customers, making it easier for them to browse products, place orders, and track their purchases.

# LIST OF ENTITIES AND ATTRIBUTES

|  |  |
| --- | --- |
| E N T I T E S | A T T R I B U T E S |
| Products | ProductID  ProductName  Price |
| Customers | CustomerID  Name  Email  Address |
| Orders | OrderID  CustomerID  ProductID (FK)  Quantity  OrderDate |
| Inventory | ProductID (FK)  QuantityInStock |
| Sales | SaleID OrderID (FK)  ProductID (FK)  CustomerID (FK) |

# ER-DIAGRAM:



Description of ER diagram:

The diagram contains total 5 tables based on e-commerce website. Products table is used to store the details of a particular product.It contains ProductID, ProductName, Price attributes. Customers table is used to store the details of customers purchasing products. It contains CustomerID, Name, Email, Address attributes. Orders table is used to store the order details. It contains OrderID, ProductID, Quantity, OrderDate attributes. Inventory table is used to store the number of products available. It contains ProductID and QuantityStock attributes. Sales table is used to store the details of which item is purchased by which customer. It contains SalesID, OrderId, ProductID and CustomerId attributes.

Conversion of ER diagram into Tables:

A screenshot of a computer

Description automatically generated

Creation of Tables:

* Products Table:

CREATE TABLE Products (

ProductID INT PRIMARY KEY,

ProductName VARCHAR(255),

Price DECIMAL(10, 2),

);

* Customers Table:

CREATE TABLE Customers (

CustomerID INT PRIMARY KEY,

Name VARCHAR(255),

Email VARCHAR(255),

Address VARCHAR(255)

);

* Orders Table:

CREATE TABLE Orders (

OrderID INT PRIMARY KEY,

CustomerID INT,

ProductID INT,

Quantity INT,

OrderDate DATE

);

* Sales Table:

CREATE TABLE Sales (

SaleID INT PRIMARY KEY,

OrderID INT,

ProductID INT,

CustomerID INT,

FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),

FOREIGN KEY (ProductID) REFERENCES Products(ProductID),

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

);

* Inventory Table:

CREATE TABLE Inventory (

ProductID INT PRIMARY KEY,

QuantityInStock INT,

FOREIGN KEY (ProductID) REFERENCES Products(ProductID)

)

Description of Tables:

Products Table:

A screenshot of a computer

Description automatically generated

Customers:

A screenshot of a computer

Description automatically generated

Orders:

A screenshot of a computer

Description automatically generated

Sales:

A screenshot of a computer

Description automatically generated

Inventory:

A screenshot of a computer

Description automatically generated

# NORMALIZATION:

Normalization is a process in which data is organized in a manner that reduces redundancy and dependency and allows easier access and manipulation of the data. This is typically done by breaking down a large table into smaller, more manageable tables, and then establishing relationships between these tables. Normalization typically involves dividing a database into two or more tables and defining relationships between the tables. The objective of normalization is to ensure that each table has a clearly defined role, and that data is stored in only one place, which reduces data redundancy and eliminates the potential for inconsistencies in the data.

The tables are already normalized to 3NF form becauseall tables don’t contain any multivalued attributes and no attribute has transitive dependency. All attributes are dependent on primary key attribute.

Creation of Data into the Tables:

1.Inserting data into products table:

INSERT INTO Products (ProductID, ProductName, Price)

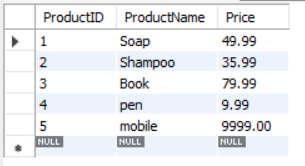
VALUES (1, 'Soap', 49.99),

(2, 'Shampoo', 35.99),

(3, 'Book', 79.99),

(4,'pen',9.99),

(5,'mobile',9999);



2. Inserting data into Customers table:

INSERT INTO Customers (CustomerID, Name, Email, Address)

VALUES (1, 'Subhash', 'subhash@gmail.com', 'guntur'),

(2, 'Aditya', 'aditya@gmail.com', 'hyderbad'),

(3, 'Satya Krishna', 'skrishna@gmail.com', 'Kakinada'),

(4,'Chinni Krishna','chinni@gmail.com','guntur'),

(5,'Krishna Teja','teja@gmail.com','guntur');

A screenshot of a computer

Description automatically generated

3. Inserting data into Orders table:

INSERT INTO Orders (OrderID, CustomerID, ProductID, Quantity, OrderDate)

VALUES (1, 1, 1, 2, '2024-04-23'),

(2, 2, 2, 1, '2024-04-24'),

(3, 3, 3, 3, '2024-04-25'),

(4, 2, 1, 4, '2024-04-23'),

(5, 3, 2, 2, '2024-04-24');

A screenshot of a computer

Description automatically generated

4. Inserting data into Sales table:

INSERT INTO Sales (SaleID, OrderID, ProductID, CustomerID)

VALUES (1, 1, 1, 1),

(2, 2, 2, 2),

(3, 3, 3, 4),

(4, 2, 5, 3),

(5, 3, 3, 2);

A screenshot of a computer

Description automatically generated

5. Inserting data into Inventory table:

INSERT INTO Inventory (ProductID, QuantityInStock)

VALUES (1, 100),

(2, 70),

(3, 150),

(4, 250),

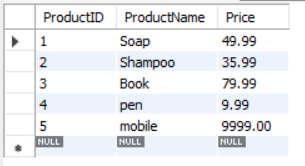
(5, 120);

A screenshot of a computer

Description automatically generated

Queries on the tables:

1. SELECT \* FROM Products;



2. SELECT \* FROM Customers WHERE Email = 'chinni@gmail.com';

A screenshot of a computer

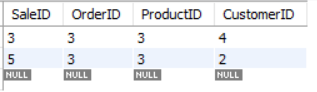
Description automatically generated

3.SELECT \* FROM Orders WHERE CustomerID = 2;

A screenshot of a computer

Description automatically generated

4.SELECT \* FROM Sales WHERE OrderID = 3;



5.SELECT \* FROM Inventory WHERE ProductID = 1;

A screenshot of a computer

Description automatically generated

Creation of views using the tables:

1.

CREATE VIEW SalesWithProductDetails AS

SELECT S.SaleID, S.OrderID, P.ProductID, P.ProductName, P.Price

FROM Sales S

JOIN Products P ON S.ProductID = P.ProductID;

SELECT \* FROM SalesWithProductDetails;

A screenshot of a computer

Description automatically generated

2.

CREATE VIEW ProductsWithInventory AS

SELECT P.ProductID, P.ProductName, P.Price, I.QuantityInStock

FROM Products P

JOIN Inventory I ON P.ProductID = I.ProductID;

A screenshot of a computer screen

Description automatically generated

3.

CREATE VIEW CustomersAndOrders AS

SELECT C.CustomerID, C.Name, O.OrderID, O.ProductID, O.Quantity, O.OrderDate

FROM Customers C

JOIN Orders O ON C.CustomerID = O.CustomerID;

A screenshot of a computer

Description automatically generated

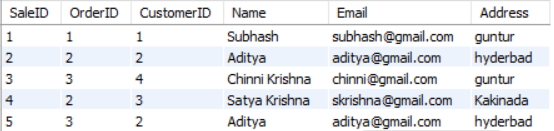
4.

CREATE VIEW SalesWithCustomerDetails AS

SELECT S.SaleID, S.OrderID, C.CustomerID, C.Name, C.Email, C.Address

FROM Sales S

JOIN Customers C ON S.CustomerID = C.CustomerID;



5.

CREATE VIEW OrdersWithDetails AS

SELECT O.OrderID, P.ProductID, P.ProductName, P.Price, C.CustomerID, C.Name

FROM Orders O

JOIN Products P ON O.ProductID = P.ProductID

JOIN Customers C ON O.CustomerID = C.CustomerID;

A screenshot of a product list

Description automatically generated

# Conclusion:

In our eCommerce website project, we have stored all the essential information about the products, customers, orders, sales, and inventory. This database is instrumental for the applications that facilitate customers to shop online, providing them with a convenient and user-friendly experience. Customers can browse and purchase products from the comfort of their own homes, avoiding the inconveniences of traditional brick-and-mortar shopping.

We have considered the most crucial requirements for an eCommerce website; however, many more features and details can be added to our project to create an even more robust and user-friendly application. Potential enhancements could include product reviews, shipping information, promotional codes, and more.

# 