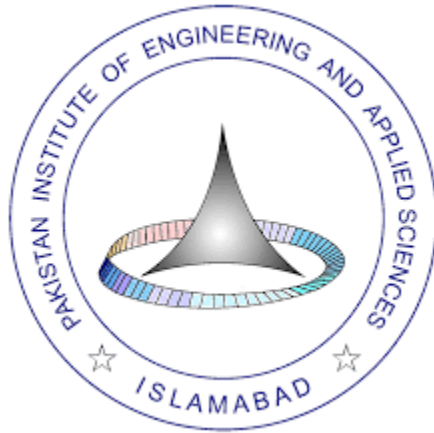




Internet Application Development

Project Title: Grocery Store Management System

Submitted By: Bisma Tahir

Submitted to: Sir Irfan Hameed

Submitted on: May 22,2025

Project Description:

This project involves developing a data-driven web application using ASP.NET Web Forms for managing an online grocery store. The system will allow customers to register, browse products, add items to cart, place orders, and make payments. Admins will manage categories, products, and orders, while suppliers can manage product stock. The project ensures secure transactions, proper form validation, role-based functionalities, and session-based state management.

Key Features:

- **Customer Interface** – Registration, Login, View Products, Add to Cart, Checkout, Payment
- **Admin Interface** – Manage Products, Categories, Orders, Users
- **Supplier Interface** – Manage Inventory, Update Stock
- **Security Features** – Password hashing, query parameter validation, ViewState encryption
- **State Management** – ViewState, Session, Cookies
- **Form Validation** – Client-side & Server-side validation using ASP.NET Validation Controls
- **Role-Based Access** – Admin, Customer, and Supplier roles with access control

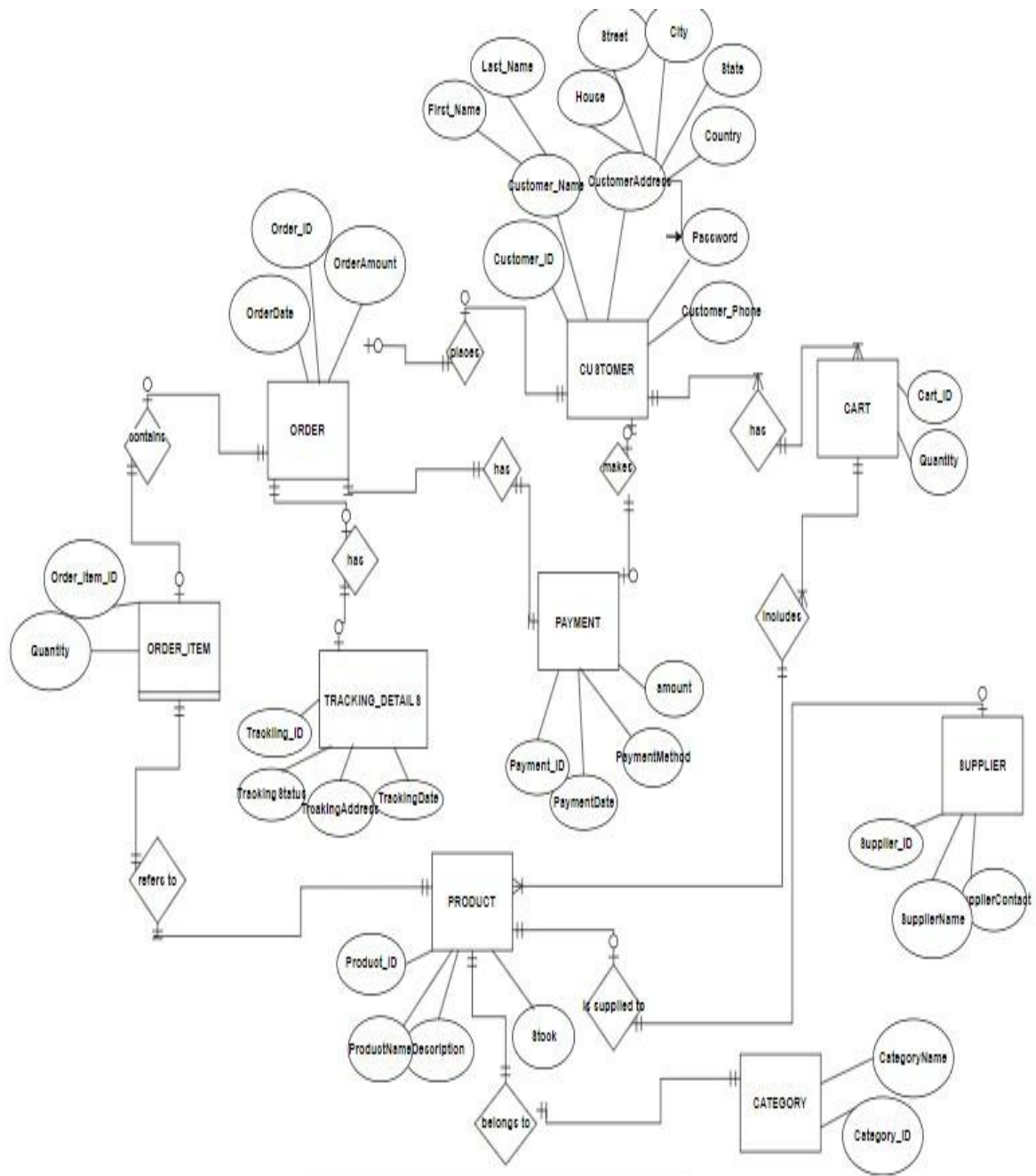
Entities Involved:

Following entities are involved in this system:

1. Customer
2. Admin
3. Order
4. Order
5. Product
6. Tracking_Details
7. Cart
8. Category

9. Supplier
10. Order_Item
11. Payment

Entity Relationship Model:



Relational Model:

CUSTOMER:

<u>Customer_ID</u>	First_Name	Last_Name	Password	Customer_Phone	Street	House	City	State	Country
--------------------	------------	-----------	----------	----------------	--------	-------	------	-------	---------

ORDER:

<u>Order_ID</u>	OrderDate	Amount
-----------------	-----------	--------

PRODUCT:

<u>Product_ID</u>	Product_name
-------------------	--------------

PRODUCT PRICE HISTORY:

<u>Product_ID</u>	Price	price_history	Date
-------------------	-------	---------------	------

SUPPLIER:

<u>Supplier_ID</u>	SupplierName	SupplierContact
--------------------	--------------	-----------------

PAYMENT:

<u>Payment_ID</u>	PaymentDate	PaymentAmount	Online	Cash_on_delivery
-------------------	-------------	---------------	--------	------------------

TRACKING DETAILS:

Tracking_ID	Order_ID	TrackingStatus	TrackingAddress
-------------	----------	----------------	-----------------

ORDER_ITEM TABLE:

<u>Order_Item_ID</u>	Order_ID	Quantity
----------------------	----------	----------

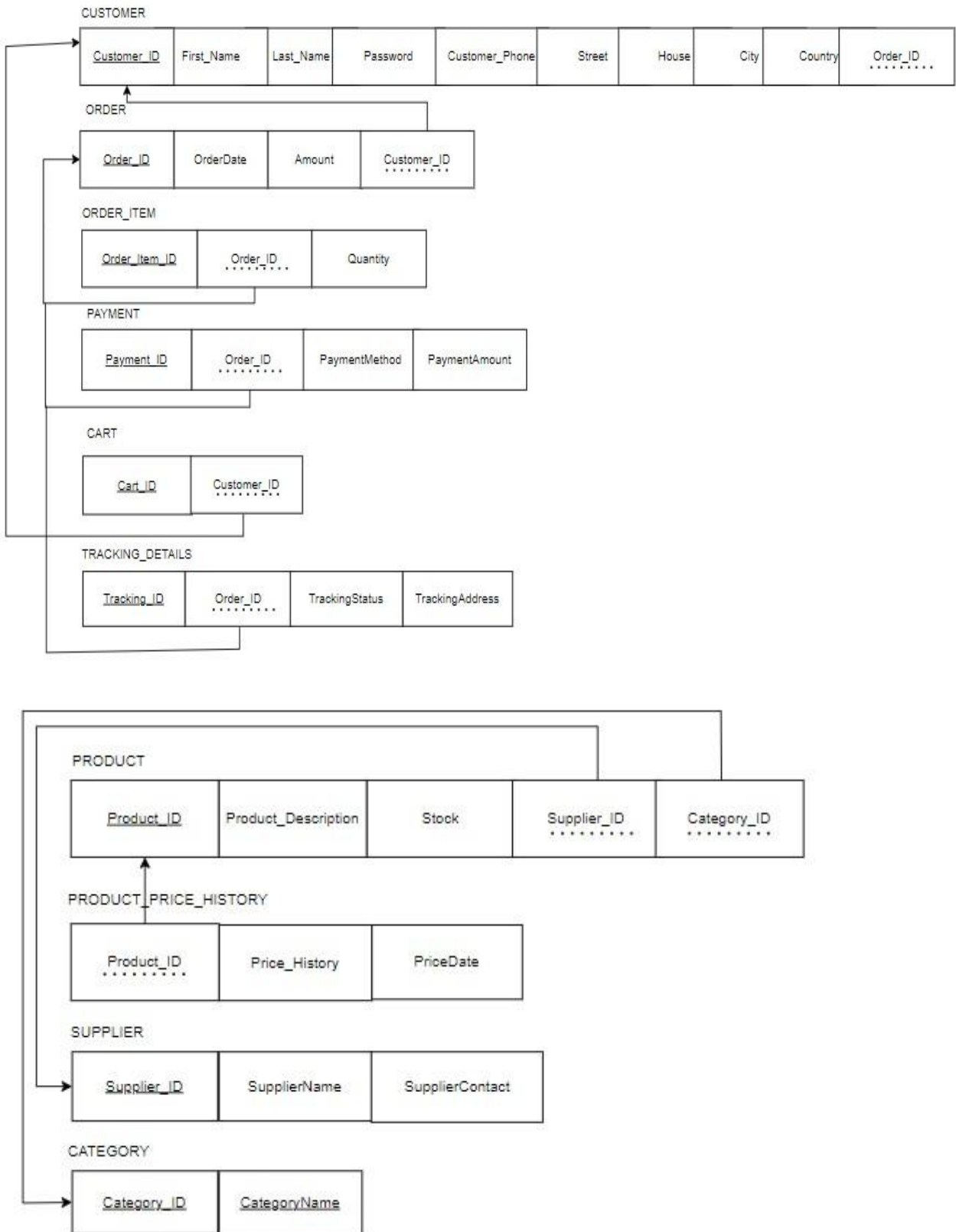
CART:

<u>Cart_ID</u>	Customer_ID
----------------	-------------

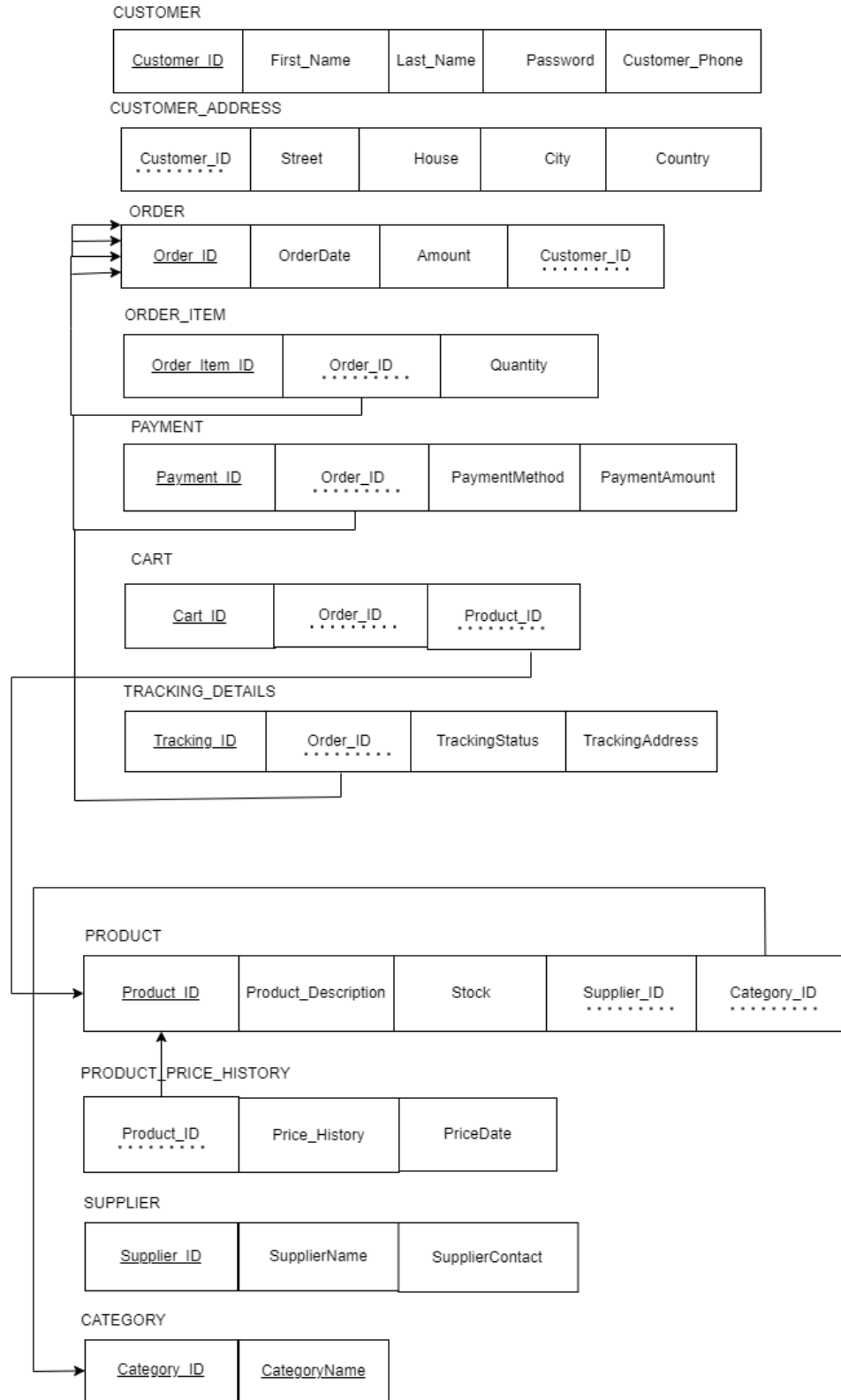
PRODUCT CATEGORY:

<u>Category_ID</u>	Category_name
--------------------	---------------

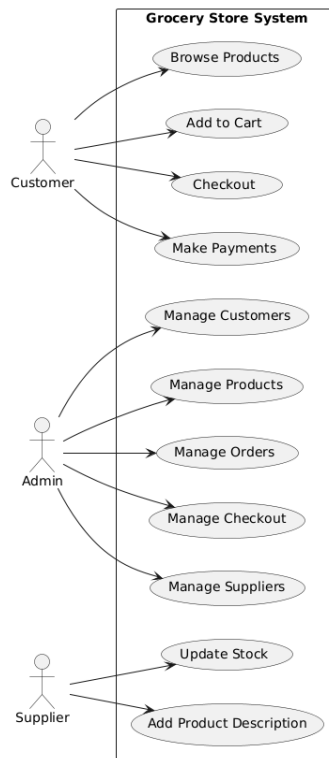
Complete Relational Model:



Normalized Schema:

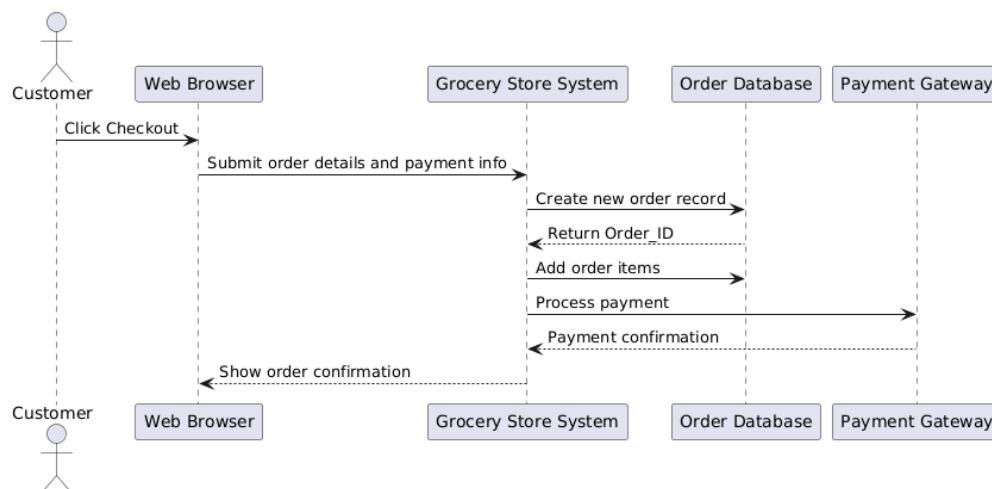


Use Case Model

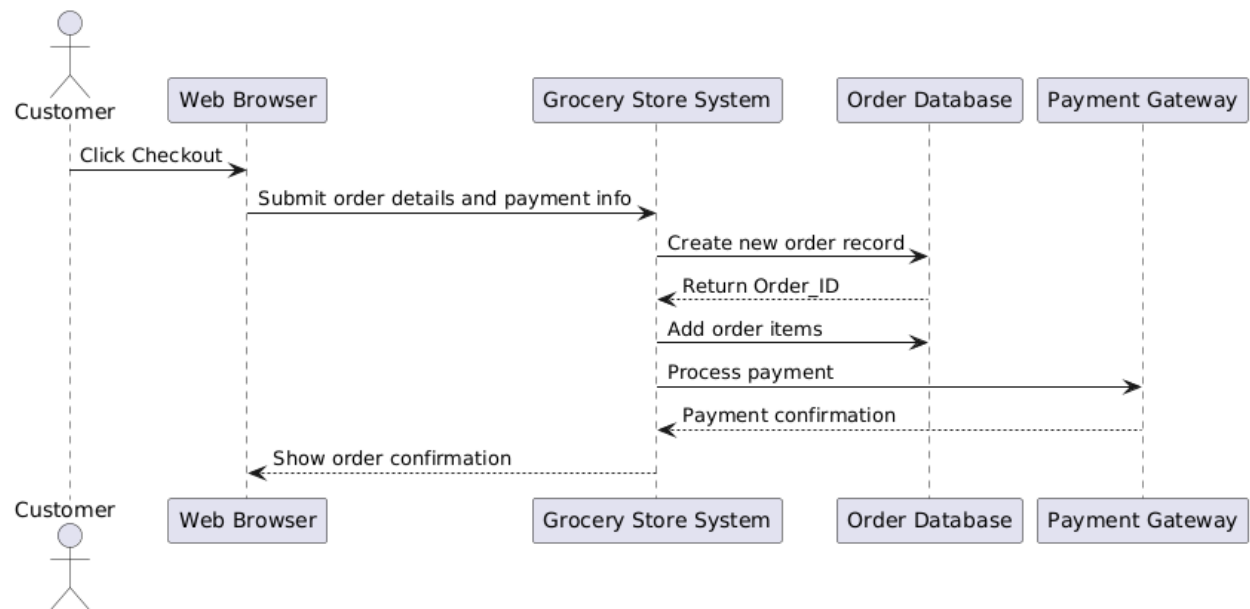


Sequence Diagrams:

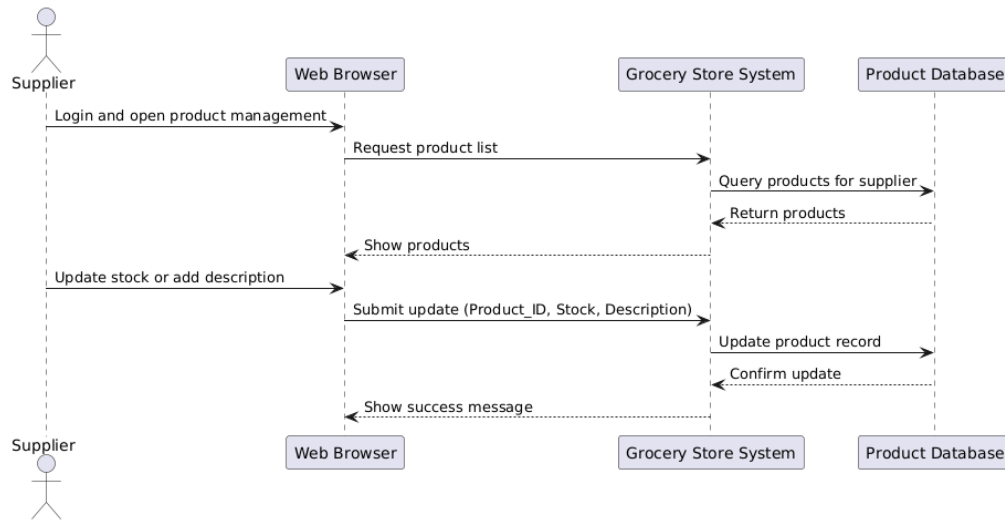
1. Customer Browses and Adds Product to Cart



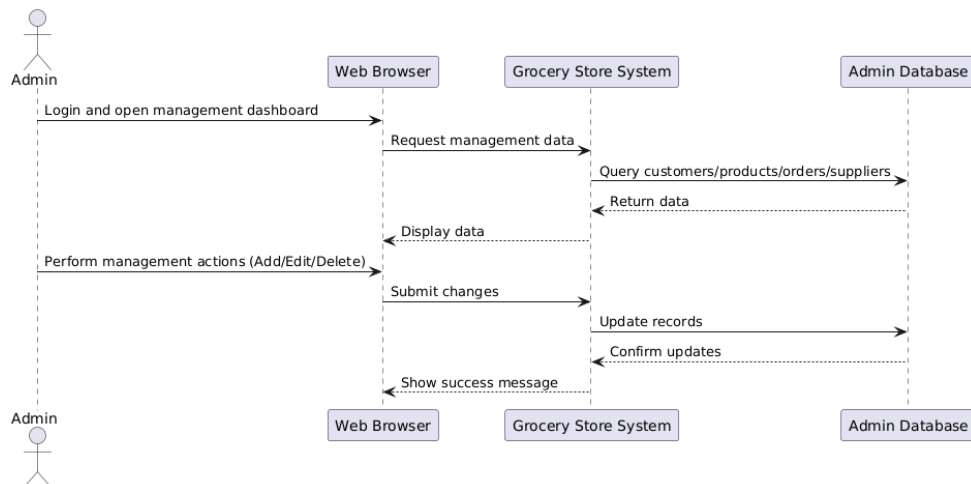
2. Customer Checkout and Make Payment



3. Supplier Updates Stock and Adds Product Description

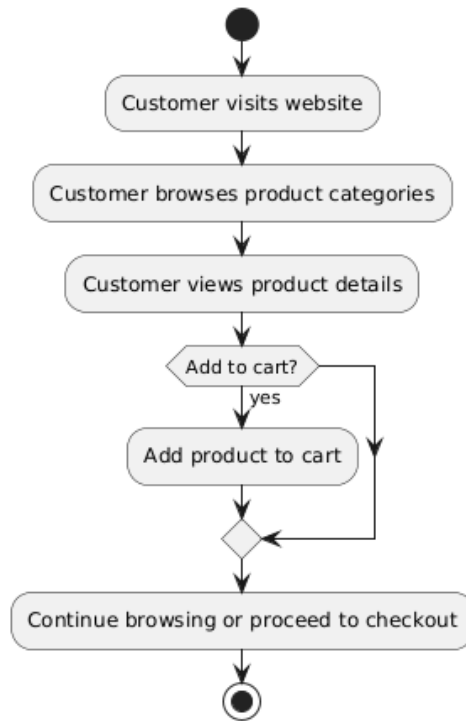


4. Admin Manages Customers, Products, Orders, Suppliers

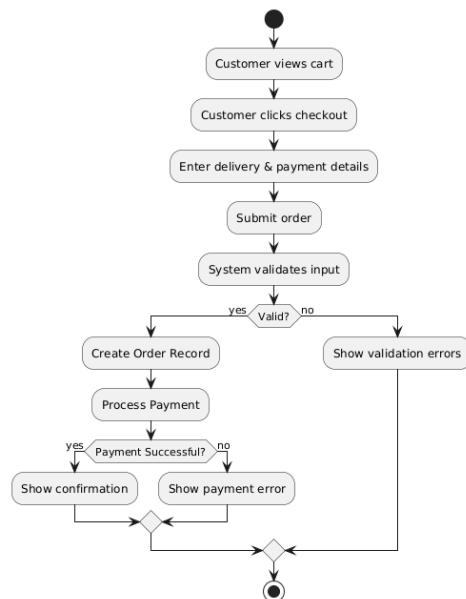


Activity Diagrams:

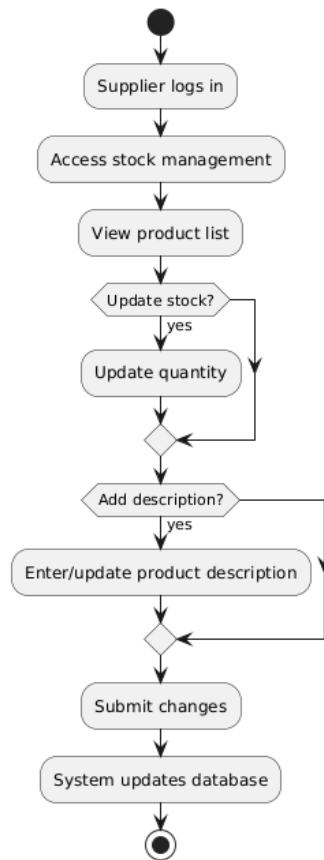
1. Customer Browses Products and Adds to Cart



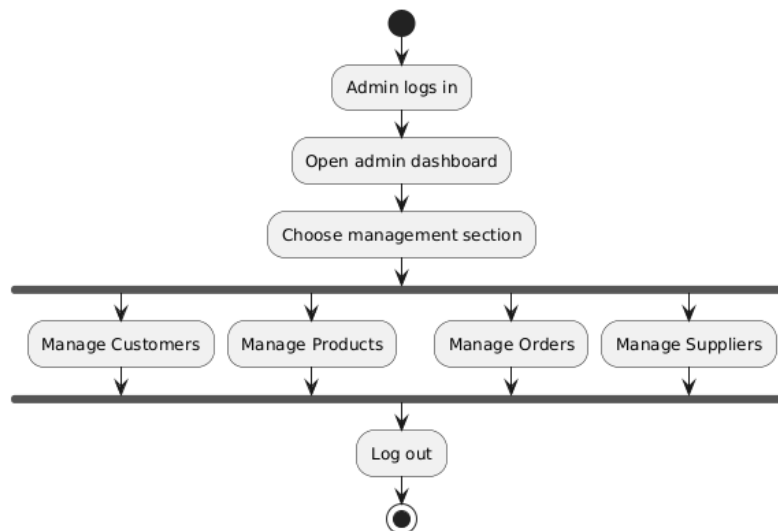
2. Customer Checkout and Payment



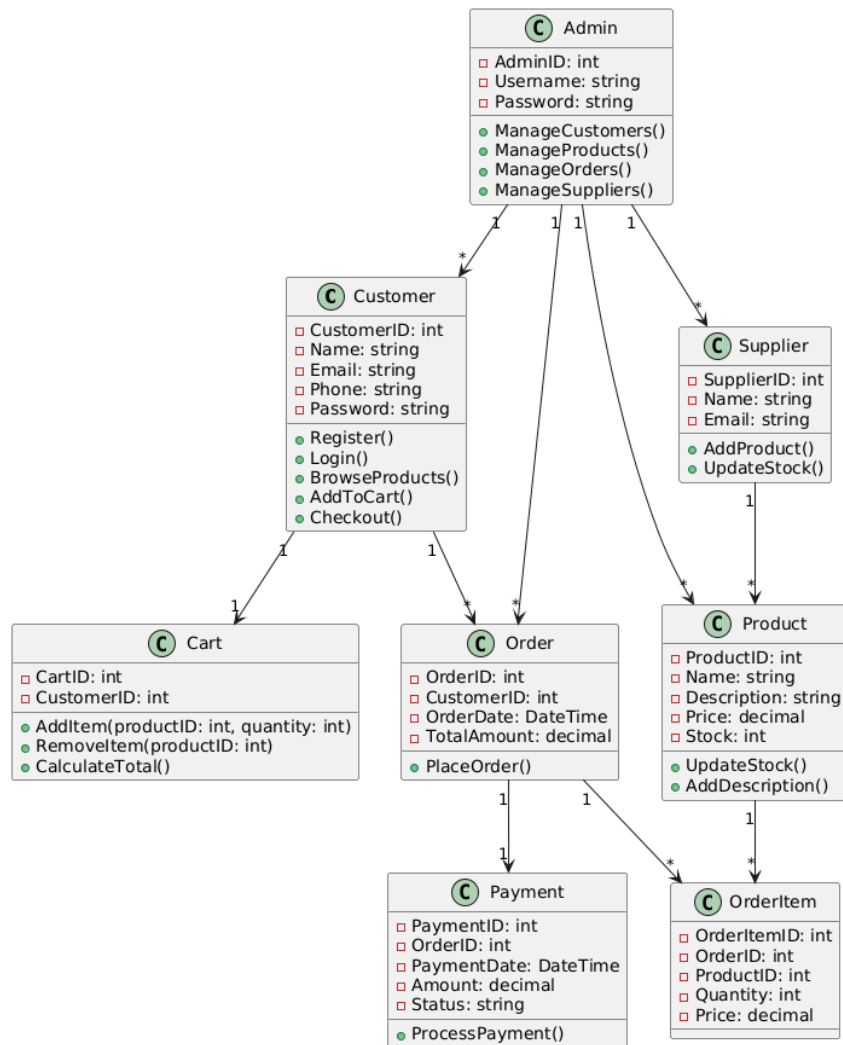
3. Supplier Updates Stock and Adds Description



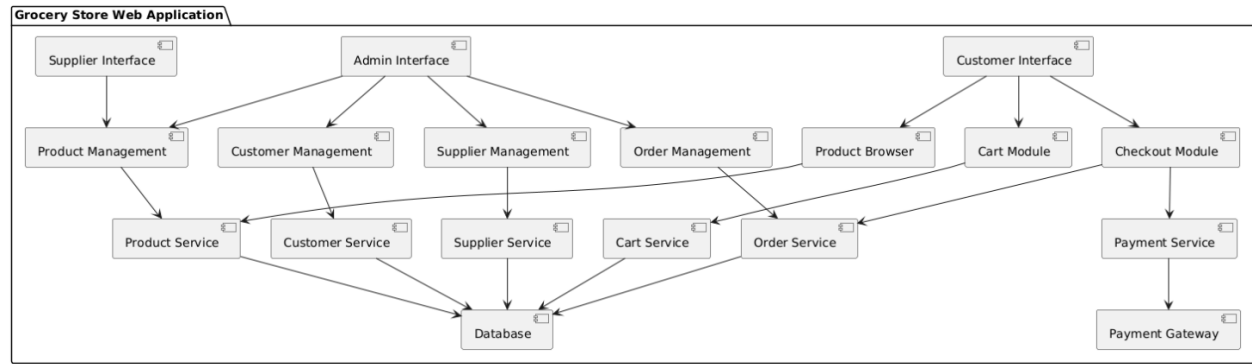
4. Admin Manages System



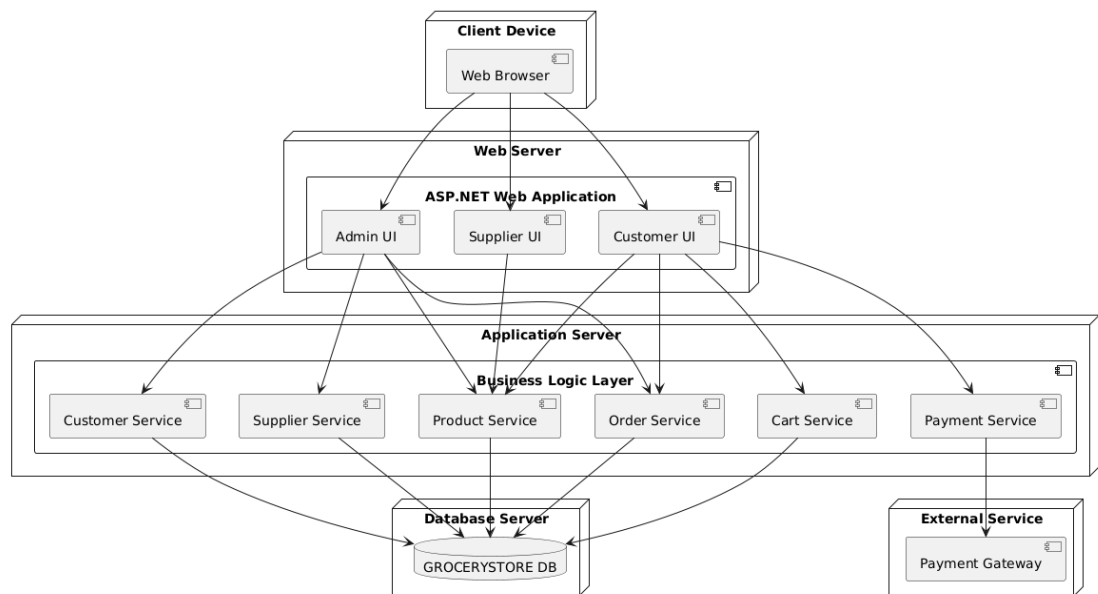
Class Diagram



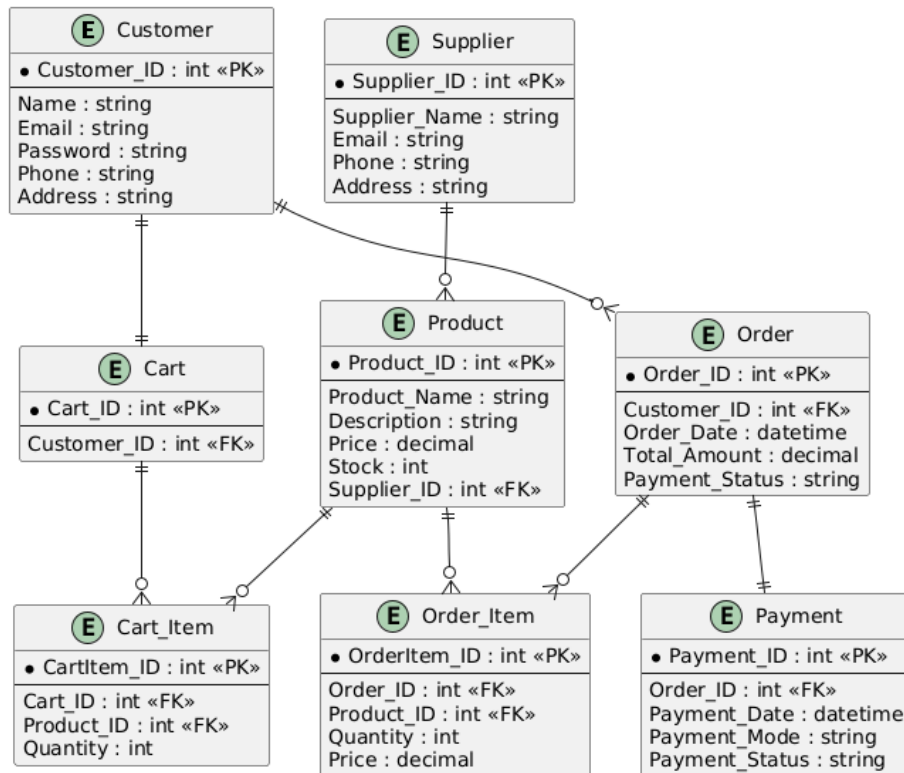
Component Diagram



Deployment Diagram



ER Model



SQL Implementation

Creating Tables:

- Admin T:

```

CREATE TABLE Admin_T (
    Admin_ID INT PRIMARY KEY,
    Admin_Username NVARCHAR(100),
    Admin_Password NVARCHAR(100),
    Admin_Email NVARCHAR(100),
    CreatedAt DATETIME,
    LastLogin DATETIME

```

);

- Customer_T

```
CREATE TABLE Customer_T (  
  
Customer_ID INT PRIMARY KEY,  
  
First_Name VARCHAR(100),  
  
Last_Name VARCHAR(100),  
  
Street VARCHAR(255),  
  
House INT,  
  
City VARCHAR(100),  
  
State VARCHAR(100),  
  
Country VARCHAR(100),  
  
Customer_Phone VARCHAR(20),  
  
Customer_Password VARCHAR(100),  
  
Customer_Email VARCHAR(100)  
);
```

- Category_T

```
CREATE TABLE Category_T (  
  
Category_ID BIGINT PRIMARY KEY,  
  
Category_Name VARCHAR(100)  
);
```

- Supplier_T

```
CREATE TABLE Supplier_T (  
  
Supplier_ID BIGINT PRIMARY KEY,  
  
Supplier_Name VARCHAR(100),  
  
Supplier_Contact VARCHAR(50),  
  
Supplier_Email VARCHAR(100),  
  
Supplier_Password VARCHAR(100)  
  
);
```

- Product_T

```
CREATE TABLE Product_T (  
  
Product_ID BIGINT PRIMARY KEY,  
  
SKU VARCHAR(50),  
  
Product_Description VARCHAR(255),  
  
Stock BIGINT,  
  
Price DECIMAL(10,2),  
  
Category_ID BIGINT,  
  
Supplier_ID BIGINT,  
  
FOREIGN KEY (Category_ID) REFERENCES Category_T(Category_ID),  
  
FOREIGN KEY (Supplier_ID) REFERENCES Supplier_T(Supplier_ID)  
  
);
```

- Order_T

```
CREATE TABLE Order_T (  
  
Order_ID BIGINT PRIMARY KEY,Order_Date DATE,  
  
Customer_ID BIGINT,  
  
Total_Price DECIMAL(10,2),  
  
FOREIGN KEY (Customer_ID) REFERENCES Customer_T(Customer_ID)  
  
);
```

- Order Item T

```
CREATE TABLE Order_Item_T (Order_Item_ID BIGINT PRIMARY KEY,  
  
Order_ID BIGINT,  
  
Product_ID BIGINT,  
  
Quantity BIGINT,  
  
FOREIGN KEY (Order_ID) REFERENCES Order_T(Order_ID),  
  
FOREIGN KEY (Product_ID) REFERENCES Product_T(Product_ID)  
  
);
```

- Payment_Method_T

```
CREATE TABLE Payment_Method_T (  
  
Payment_Method_ID BIGINT PRIMARY KEY,Payment_Method VARCHAR(100),  
  
TransactionID VARCHAR(100),  
  
Payment_Status VARCHAR(50)  
  
);
```

- Payment_T

```
CREATE TABLE Payment_T (  
  
Payment_ID BIGINT PRIMARY KEY,  
  
Payment_Date DATE,  
  
Amount DECIMAL(10,2),  
  
Payment_Method_ID BIGINT,  
  
FOREIGN KEY (Payment_Method_ID) REFERENCES Payment_Method_T(Payment_Method_ID)  
  
);
```

- Cart_T

```
CREATE TABLE Cart_T (  
  
Cart_ID BIGINT PRIMARY KEY,  
  
Customer_ID BIGINT,  
  
Quantity BIGINT,  
  
FOREIGN KEY (Customer_ID) REFERENCES Customer_T(Customer_ID)  
  
);
```

- Tracking_Details_T

```
CREATE TABLE Tracking_Details_T (  
  
Tracking_ID BIGINT PRIMARY KEY,  
  
Tracking_Date DATE,
```

Tracking_Status VARCHAR(100),
Street VARCHAR(255),
House INT,
City VARCHAR(100),
State CHAR(2),
Country VARCHAR(100)
);

- ContactUs_T

```
CREATE TABLE ContactUs_T (  
ContactUs_ID INT PRIMARY KEY,  
Name NVARCHAR(100),  
Email NVARCHAR(100),  
Message NVARCHAR(MAX),  
DateSubmitted DATETIME  
);
```

- Product Price History_T

```
CREATE TABLE Product_Price_History_T (  
Product_ID BIGINT,  
Price DECIMAL(10,2),Price_Date DATE,  
Price_History VARCHAR(255),  
PRIMARY KEY (Product_ID, Price_Date),
```

FOREIGN KEY (Product_ID) REFERENCES Product_T(Product_ID)

);

Inserting Values Into the Tables:

- Customer T:

INSERT INTO Customer_T

(Old_Customer_ID, First_Name, Last_Name, Street, House, City, State, Country,
Customer_Phone, Customer_Password, Customer_ID, Customer_Email)

VALUES

(0, 'Bisma', 'Tahir', '1', '119', 'rwp', 'punjab', 'pakistan', '1234567890',
'b041c0aeb35bb0fa4aa668ca5a920b590196fdaf9a00eb852c9b7f4d123cc6d6', 27,
'bisma.tahir@example.com'),

(1, 'John', 'Doe', '123 Oak St', '101', 'Springfield', 'IL', 'USA', '555-0101',
'b041c0aeb35bb0fa4aa668ca5a920b590196fdaf9a00eb852c9b7f4d123cc6d6', 1,
'john.doe@example.com'),

(2, 'Jane', 'Smith', '456 Maple Ave', '102', 'Greenville', 'TX', 'USA', '555-0102',
'b041c0aeb35bb0fa4aa668ca5a920b590196fdaf9a00eb852c9b7f4d123cc6d6', 2,
'jane.smith@example.com'),

(3, 'Alice', 'Johnson', '789 Birch Rd', '103', 'Clinton', 'IA', 'USA', '555-0103',
'b041c0aeb35bb0fa4aa668ca5a920b590196fdaf9a00eb852c9b7f4d123cc6d6', 3,
'alice.johnson@example.com'),

(4, 'Bob', 'Williams', '101 Pine St', '104', 'Madison', 'WI', 'USA', '555-0104',
'b041c0aeb35bb0fa4aa668ca5a920b590196fdaf9a00eb852c9b7f4d123cc6d6', 4,
'bob.williams@example.com'),

(5, 'Charlie', 'Brown', '202 Cedar Blvd', '105', 'Lincoln', 'NE', 'USA', '555-0105',
'b041c0aeb35bb0fa4aa668ca5a920b590196fdaf9a00eb852c9b7f4d123cc6d6', 5,
'charlie.brown@example.com'),

(6, 'David', 'Miller', '303 Elm St', '106', 'Aurora', 'CO', 'USA', '555-0106',
'b041c0aeb35bb0fa4aa668ca5a920b590196fdaf9a00eb852c9b7f4d123cc6d6', 6,
'david.miller@example.com'),

(7, 'Eva', 'Davis', '404 Spruce Ln', '107', ' Fargo', 'ND', 'USA', '555-0107',
'b041c0aeb35bb0fa4aa668ca5a920b590196fdaf9a00eb852c9b7f4d123cc6d6', 7,
'eva.davis@example.com'),

(8, 'Frank', 'Garcia', '505 Walnut St', '108', 'Salem', 'OR', 'USA', '555-0108',
'b041c0aeb35bb0fa4aa668ca5a920b590196fdaf9a00eb852c9b7f4d123cc6d6', 8,
'frank.garcia@example.com'),

(9, 'Grace', 'Martinez', '606 Chestnut Rd', '109', 'Helena', 'MT', 'USA', '555-0109',
'b041c0aeb35bb0fa4aa668ca5a920b590196fdaf9a00eb852c9b7f4d123cc6d6', 9,
'grace.martinez@example.com'),

(10, 'Hank', 'Rodriguez', '707 Fir St', '110', 'Boise', 'ID', 'USA', '555-0110',
'b041c0aeb35bb0fa4aa668ca5a920b590196fdaf9a00eb852c9b7f4d123cc6d6', 10,
'hank.rodriguez@example.com'),

(11, 'Ivy', 'Wilson', '808 Cypress Ave', '111', 'Albany', 'NY', 'USA', '555-0111',
'b041c0aeb35bb0fa4aa668ca5a920b590196fdaf9a00eb852c9b7f4d123cc6d6', 11,
'ivy.wilson@example.com'),

(12, 'Jack', 'Anderson', '909 Poplar St', '112', 'Reno', 'NV', 'USA', '555-0112',
'b041c0aeb35bb0fa4aa668ca5a920b590196fdaf9a00eb852c9b7f4d123cc6d6', 12,
'jack.anderson@example.com'),

(13, 'Kate', 'Thomas', '1010 Dogwood Dr', '113', 'Tucson', 'AZ', 'USA', '555-0113',
'b041c0aeb35bb0fa4aa668ca5a920b590196fdaf9a00eb852c9b7f4d123cc6d6', 13,
'kate.thomas@example.com'),

(14, 'Leo', 'Taylor', '1111 Magnolia Ln', '114', 'Olympia', 'WA', 'USA', '555-0114',
'b041c0aeb35bb0fa4aa668ca5a920b590196fdaf9a00eb852c9b7f4d123cc6d6', 14,
'leo.taylor@example.com'),

(15, 'Mia', 'Moore', '1212 Willow St', '115', 'Richmond', 'VA', 'USA', '555-0115',
'b041c0aeb35bb0fa4aa668ca5a920b590196fdaf9a00eb852c9b7f4d123cc6d6', 15,
'mia.moore@example.com'),

(16, 'Nick', 'Jackson', '1313 Aspen Ct', '116', 'Bismarck', 'ND', 'USA', '555-0116',
'b041c0aeb35bb0fa4aa668ca5a920b590196fdaf9a00eb852c9b7f4d123cc6d6', 16,
'nick.jackson@example.com'),

(17, 'Olivia', 'White', '1414 Beech Rd', '117', 'Dover', 'DE', 'USA', '555-0117',
'b041c0aeb35bb0fa4aa668ca5a920b590196fdaf9a00eb852c9b7f4d123cc6d6', 17,
'olivia.white@example.com'),

(18, 'Paul', 'Harris', '1515 Hickory St', '118', 'Augusta', 'ME', 'USA', '555-0118',
'b041c0aeb35bb0fa4aa668ca5a920b590196fdaf9a00eb852c9b7f4d123cc6d6', 18,
'paul.harris@example.com'),

(19, 'Quinn', 'Martin', '1616 Sycamore Ave', '119', 'Concord', 'NH', 'USA', '555-0119',
'b041c0aeb35bb0fa4aa668ca5a920b590196fdaf9a00eb852c9b7f4d123cc6d6', 19,
'quinn.martin@example.com');

- Admin_T:

INSERT INTO Admin_T

(Admin_ID, Admin_Username, Admin_Password, Admin_Email, CreatedAt, LastLogin)

VALUES

```
(1, 'adminuser', '240be518fabd2724ddb6f04eeb1da5967448d7e831c08c8fa822809f74c720a9',  
'admin@example.com', '2025-05-22 08:56:05', NULL);
```

- Cart_T:

INSERT INTO Cart_T (Cart_ID, Customer_ID, Quantity) VALUES

(1, 1, 2),

(2, 2, 1),

(3, 3, 3),

(4, 4, 4),

(5, 5, 2),

(6, 6, 5),

(7, 7, 3),

(8, 8, 6),

(9, 9, 1),

(10, 10, 4),

(11, 11, 2),

(12, 12, 3),

(13, 13, 5),

(14, 14, 6),

(15, 15, 4),

(16, 16, 1),

(17, 17, 3),

(18, 18, 5),

(19, 19, 2),

(20, 20, 4);

- Category T;

INSERT INTO Category_T (Category_ID, Category_Name) VALUES

(1, 'Fruits'),

(2, 'Vegetables'),

(3, 'Beverages'),

(4, 'Dairy'),

(5, 'Snacks'),

(6, 'Meat'),

(7, 'Seafood'),

(8, 'Frozen Foods'),

(9, 'Bakery'),

(10, 'Cereals'),

(11, 'Condiments'),

(12, 'Baking'),

(13, 'Health & Wellness'),

(14, 'Cleaning Supplies'),

(15, 'Pet Supplies'),

(16, 'Baby Products'),

(17, 'Organic'),

(18, 'Gluten-Free'),

(19, 'Vegan'),

(20, 'Kosher');

- Order_T;

INSERT INTO Order_T (Order_ID, Order_Date, Customer_ID, Total_Price) VALUES

(1, '2024-12-01 00:00:00', 1, 250.00),

(2, '2024-12-02 00:00:00', 2, 300.00),

(3, '2024-12-03 00:00:00', 3, 150.00),

(4, '2024-12-04 00:00:00', 4, 500.00),

(5, '2024-12-05 00:00:00', 5, 200.00),

(6, '2024-12-06 00:00:00', 6, 350.00),

(7, '2024-12-07 00:00:00', 7, 450.00),

(8, '2024-12-08 00:00:00', 8, 100.00),

(9, '2024-12-09 00:00:00', 9, 600.00),

(10, '2024-12-10 00:00:00', 10, 250.00),

(11, '2024-12-11 00:00:00', 11, 700.00),

(12, '2024-12-12 00:00:00', 12, 380.00),

(13, '2024-12-13 00:00:00', 13, 320.00),

(14, '2024-12-14 00:00:00', 14, 550.00),

(15, '2024-12-15 00:00:00', 15, 600.00),
(16, '2024-12-16 00:00:00', 16, 420.00),
(17, '2024-12-17 00:00:00', 17, 190.00),
(18, '2024-12-18 00:00:00', 18, 520.00),
(19, '2024-12-19 00:00:00', 19, 250.00),
(20, '2024-12-20 00:00:00', 20, 450.00);

- Order Item T;

INSERT INTO Order_Item_T (Order_Item_ID, Order_ID, Product_ID, Quantity) VALUES

(1, 1, 1, 2),
(2, 1, 2, 1),
(3, 2, 3, 3),
(4, 2, 4, 2),
(5, 3, 5, 1),
(6, 3, 6, 4),
(7, 4, 7, 3),
(8, 4, 8, 2),
(9, 5, 9, 1),
(10, 5, 10, 5),
(11, 6, 11, 2),
(12, 6, 12, 4),
(13, 7, 13, 3),

(14, 7, 14, 2),
(15, 8, 15, 1),
(16, 8, 16, 6),
(17, 9, 17, 2),
(18, 9, 18, 3),
(19, 10, 19, 4),
(20, 10, 20, 1);

- Payment T;

INSERT INTO Payment_T (Payment_ID, Payment_Date, Amount, Payment_Method_ID) VALUES

(1, '2024-11-01 00:00:00', 100.50, 1),
(2, '2024-11-02 00:00:00', 200.75, 2),
(3, '2024-11-03 00:00:00', 150.00, 3),
(4, '2024-11-04 00:00:00', 80.25, 4),
(5, '2024-11-05 00:00:00', 250.00, 5),
(6, '2024-11-06 00:00:00', 180.40, 6),
(7, '2024-11-07 00:00:00', 120.90, 7),
(8, '2024-11-08 00:00:00', 160.60, 8),
(9, '2024-11-09 00:00:00', 95.50, 9),
(10, '2024-11-10 00:00:00', 220.75, 10),
(11, '2024-11-11 00:00:00', 300.20, 11),
(12, '2024-11-12 00:00:00', 130.00, 12),

(13, '2024-11-13 00:00:00', 215.30, 13),
(14, '2024-11-14 00:00:00', 110.15, 14),
(15, '2024-11-15 00:00:00', 175.60, 15),
(16, '2024-11-16 00:00:00', 90.40, 16),
(17, '2024-11-17 00:00:00', 130.25, 17),
(18, '2024-11-18 00:00:00', 200.00, 18),
(19, '2024-11-19 00:00:00', 140.75, 19),
(20, '2024-11-20 00:00:00', 180.00, 20);

- Payment Method T;

INSERT INTO Payment_Method_T (Payment_Method_ID, Payment_Method, TransactionID, Payment_Status) VALUES

(1, 'online', 'TXN001', 'successful'),
(2, 'cash on delivery', NULL, 'pending'),
(3, 'online', 'TXN003', 'failed'),
(4, 'cash on delivery', NULL, 'successful'),
(5, 'online', 'TXN005', 'successful'),
(6, 'cash on delivery', NULL, 'successful'),
(7, 'online', 'TXN007', 'pending'),
(8, 'cash on delivery', NULL, 'failed'),
(9, 'online', 'TXN009', 'successful'),
(10, 'cash on delivery', NULL, 'successful'),

(11, 'online', 'TXN011', 'failed'),
(12, 'cash on delivery', NULL, 'pending'),
(13, 'online', 'TXN013', 'successful'),
(14, 'cash on delivery', NULL, 'successful'),
(15, 'online', 'TXN015', 'pending'),
(16, 'cash on delivery', NULL, 'failed'),
(17, 'online', 'TXN017', 'successful'),
(18, 'cash on delivery', NULL, 'successful'),
(19, 'online', 'TXN019', 'failed'),
(20, 'cash on delivery', NULL, 'pending');

- Product T;

INSERT INTO Product_T (Product_ID, SKU, Product_Description, Stock, Category_ID, Price)
VALUES

(1, 'GRO001', 'Organic Apples - 1kg', 150, 1, 350.00),
(2, 'GRO002', 'Bananas - 1kg', 200, 1, 180.00),
(3, 'GRO003', 'Carrots - 1kg', 120, 2, 120.00),
(4, 'GRO004', 'Broccoli - 500g', 80, 2, 160.00),
(5, 'GRO005', 'Tomatoes - 1kg', 250, 2, 200.00),
(6, 'GRO006', 'Cucumbers - 1kg', 180, 2, 150.00),
(7, 'GRO007', 'Lettuce - 1 head', 220, 2, 100.00),
(8, 'GRO008', 'Potatoes - 5kg', 300, 2, 300.00),

(9, 'GRO009', 'Onions - 1kg', 250, 2, 220.00),
 (10, 'GRO010', 'Garlic - 500g', 200, 2, 260.00),
 (11, 'GRO011', 'Milk - 1L', 500, 4, 190.00),
 (12, 'GRO012', 'Eggs - 12 pack', 350, 4, 320.00),
 (13, 'GRO013', 'Cheese - 200g', 150, 4, 450.00),
 (14, 'GRO014', 'Yogurt - 500g', 300, 4, 280.00),
 (15, 'GRO015', 'Butter - 250g', 180, 4, 400.00),
 (16, 'GRO016', 'Chicken Breasts - 1kg', 130, 6, 850.00),
 (17, 'GRO017', 'Ground Beef - 1kg', 200, 6, 950.00),
 (18, 'GRO018', 'Pork Chops - 1kg', 150, 6, 880.00),
 (19, 'GRO019', 'Bacon - 500g', 250, 6, 600.00),
 (20, 'GRO020', 'Frozen Peas - 1kg', 300, 8, 240.00);

- Product Price History T:

INSERT INTO Price_History_T (Product_ID, Price, Price_Date, Price_History) VALUES

(1, 2.50, '2024-01-01', 'Initial price'),
 (1, 2.80, '2024-02-01', 'Previous price: 2.50, Price increase due to demand'),
 (2, 1.30, '2024-01-01', 'Initial price'),
 (2, 1.50, '2024-03-01', 'Previous price: 1.30, Price increase due to supplier cost'),
 (3, 1.00, '2024-01-01', 'Initial price'),
 (3, 1.20, '2024-02-15', 'Previous price: 1.00, Price increase for seasonal changes'),
 (4, 2.50, '2024-01-01', 'Initial price'),

(4, 2.75, '2024-04-01', 'Previous price: 2.50, Price increase due to scarcity'),

(5, 3.00, '2024-01-01', 'Initial price'),

(5, 3.25, '2024-02-10', 'Previous price: 3.00, Price increase for quality improvement'),

(6, 0.99, '2024-01-01', 'Initial price'),

(6, 1.10, '2024-03-20', 'Previous price: 0.99, Price increase due to inflation'),

(7, 1.50, '2024-01-01', 'Initial price'),

(7, 1.70, '2024-04-15', 'Previous price: 1.50, Price increase for packaging costs'),

(8, 4.00, '2024-01-01', 'Initial price'),

(8, 4.25, '2024-02-25', 'Previous price: 4.00, Price increase due to shipping costs'),

(9, 2.00, '2024-01-01', 'Initial price'),

(9, 2.25, '2024-03-10', 'Previous price: 2.00, Price increase due to market demand'),

(10, 0.50, '2024-01-01', 'Initial price'),

(10, 0.55, '2024-02-05', 'Previous price: 0.50, Price increase due to inflation'),

(11, 5.50, '2024-01-01', 'Initial price'),

(11, 5.75, '2024-02-15', 'Previous price: 5.50, Price increase due to new features'),

(12, 0.99, '2024-01-01', 'Initial price'),

(12, 1.10, '2024-02-25', 'Previous price: 0.99, Price increase due to demand'),

(13, 3.50, '2024-01-01', 'Initial price'),

(13, 3.80, '2024-04-05', 'Previous price: 3.50, Price increase due to increased material costs'),

(14, 6.00, '2024-01-01', 'Initial price'),

(14, 6.20, '2024-03-10', 'Previous price: 6.00, Price increase due to new packaging'),

(15, 2.20, '2024-01-01', 'Initial price'),

(15, 2.50, '2024-02-20', 'Previous price: 2.20, Price increase due to rising production costs'),
(16, 7.00, '2024-01-01', 'Initial price'),
(16, 7.25, '2024-03-15', 'Previous price: 7.00, Price increase due to higher demand'),
(17, 1.80, '2024-01-01', 'Initial price'),
(17, 1.90, '2024-03-30', 'Previous price: 1.80, Price increase due to supply chain issues'),
(18, 4.20, '2024-01-01', 'Initial price'),
(18, 4.50, '2024-02-28', 'Previous price: 4.20, Price increase due to market conditions'),
(19, 2.80, '2024-01-01', 'Initial price'),
(19, 3.00, '2024-03-15', 'Previous price: 2.80, Price increase due to supplier costs'),
(20, 3.50, '2024-01-01', 'Initial price');

- Supplier T;

INSERT INTO Supplier_T (Supplier_ID, Supplier_Name, Supplier_Contact, Supplier_Email,
Supplier_Password)

VALUES

(1, 'Fresh Farms', '123-456-7890', 'supplier@example.com',
'fe64f70054eb770bd151d84bf4425b356f6ba30d52c8958b65afefcdb4e02425'),
(2, 'Green Valley Supplies', '234-567-8901', 'supplier@example.com',
'fe64f70054eb770bd151d84bf4425b356f6ba30d52c8958b65afefcdb4e02425'),
(3, 'Organic Harvest', '345-678-9012', 'supplier@example.com',
'fe64f70054eb770bd151d84bf4425b356f6ba30d52c8958b65afefcdb4e02425'),
(4, 'Daily Essentials', '456-789-0123', 'supplier@example.com',
'fe64f70054eb770bd151d84bf4425b356f6ba30d52c8958b65afefcdb4e02425'),

(5, 'Quality Foods Co.', '567-890-1234', 'supplier@example.com',
'fe64f70054eb770bd151d84bf4425b356f6ba30d52c8958b65afefcdb4e02425'),
(201, 'Metro Traders', 'metro@traders.com', 'supplier@example.com',
'fe64f70054eb770bd151d84bf4425b356f6ba30d52c8958b65afefcdb4e02425');

- Tracking Details T;

INSERT INTO Tracking_T (Tracking_ID, Tracking_Date, Tracking_Status, Street, House, City, State, Country)

VALUES

(1, '2024-12-01 00:00:00', 'Shipped', '123 Oak St', '101', 'Springfield', 'IL', 'USA'),
(2, '2024-12-02 00:00:00', 'In Transit', '456 Maple Ave', '102', 'Greenville', 'TX', 'USA'),
(3, '2024-12-03 00:00:00', 'Delivered', '789 Birch Rd', '103', 'Clinton', 'IA', 'USA'),
(4, '2024-12-04 00:00:00', 'Shipped', '101 Pine St', '104', 'Madison', 'WI', 'USA'),
(5, '2024-12-05 00:00:00', 'Out for Delivery', '202 Cedar Blvd', '105', 'Lincoln', 'NE', 'USA'),
(6, '2024-12-06 00:00:00', 'Shipped', '303 Elm Dr', '106', 'Portland', 'OR', 'USA'),
(7, '2024-12-07 00:00:00', 'In Transit', '404 Redwood Ln', '107', 'Denver', 'CO', 'USA'),
(8, '2024-12-08 00:00:00', 'Delivered', '505 Willow Way', '108', 'Chicago', 'IL', 'USA'),
(9, '2024-12-09 00:00:00', 'Out for Delivery', '606 Oakwood St', '109', 'Los Angeles', 'CA', 'USA'),
(10, '2024-12-10 00:00:00', 'Shipped', '707 Cedar Ave', '110', 'Miami', 'FL', 'USA'),
(11, '2024-12-11 00:00:00', 'Delivered', '808 Birch Rd', '111', 'Dallas', 'TX', 'USA'),
(12, '2024-12-12 00:00:00', 'In Transit', '909 Maple St', '112', 'Houston', 'TX', 'USA'),
(13, '2024-12-13 00:00:00', 'Out for Delivery', '1010 Pine St', '113', 'Atlanta', 'GA', 'USA'),

(14, '2024-12-14 00:00:00', 'Shipped', '1111 Oak Ln', '114', 'New York', 'NY', 'USA'),
(15, '2024-12-15 00:00:00', 'In Transit', '1212 Cedar Blvd', '115', 'San Francisco', 'CA', 'USA'),
(16, '2024-12-16 00:00:00', 'Delivered', '1313 Elm Dr', '116', 'Seattle', 'WA', 'USA'),
(17, '2024-12-17 00:00:00', 'Shipped', '1414 Redwood Rd', '117', 'Phoenix', 'AZ', 'USA'),
(18, '2024-12-18 00:00:00', 'In Transit', '1515 Willow Ln', '118', 'Tucson', 'AZ', 'USA'),
(19, '2024-12-19 00:00:00', 'Delivered', '1616 Oak Dr', '119', 'Sacramento', 'CA', 'USA'),
(20, '2024-12-20 00:00:00', 'Shipped', '1717 Birch Blvd', '120', 'San Diego', 'CA', 'USA');

SIZE ESTIMATION:

Attribute Size Estimation

General size for data types:

- **NUMBER:** ~8 bytes (for primary keys and quantities)
- **VARCHAR2(n):** n bytes for strings (variable size depending on content)
- **DATE:** 7 bytes

Tables and Row Size Calculations

***Customer Table* Attributes:**

- CustomerID (8 bytes)
- CustomerName (FirstName 30B + LastName 30B = 60 bytes)
- House (20 bytes)
- Street (50 bytes)
- City (30 bytes)
- State (20 bytes)

- Country (30 bytes)

$8 + 60 + 20 + 50 + 30 + 20 + 30 = 218$ Bytes

Orders Table

Attributes:

- OrderID (8 bytes)
- OrderDate (7 bytes)
- TotalPrice (8 bytes)

Total Row Size:

$8 + 7 + 8 = 23$ Bytes ***Tracking_DetailsTable*** Attributes:

- TrackingID (8 bytes)
- TrackingDate (7 bytes)
- Address (50 bytes)
- City (30 bytes)
- State (20 bytes)
- Country (30 bytes)
- TrackingStatus (20 bytes) **Total Row Size:**

$8 + 7 + 50 + 30 + 20 + 30 + 20 = 165$ bytes

Order_Item Table

Attributes:

- Order_itemID (8 bytes)
- Quantity (8 bytes)

- OrderDate (7 bytes)

Total Row Size:

$8 + 8 + 7 = 23$ Bytes

Payment Table

Attributes:

- PaymentID (8 bytes)
- PaymentDate (7
bytes)
- PaymentMethod (20
bytes)
- Amount (8 bytes)

Total Row Size:

$8 + 7 + 20 + 8 = 43$ Bytes

Cart Table

Attributes:

- CartID (8 bytes)
- Quantity (8 bytes)

Total Row Size:

$8 + 8 = 16$ Bytes

Product Table

Attributes:

- ProductID (8 bytes)

- SKU (20 bytes)
- ProductDescription (100 bytes) • Price (8 bytes) o

Price_date (7 bytes) o Price_history (100
bytes)

- Stock (8 bytes)

Total Row Size:

$8 + 20 + 100 + 8 + 7 + 100 + 8 = 251$

Bytes

Category Table

Attributes:

- CategoryID (8 bytes)
- CategoryName (50 bytes)

Total Row Size:

$8 + 50 = 58$ Bytes

Supplier Table

Attributes:

- SupplierID (8 bytes)
- SupplierName (50 bytes)
- SupplierContact (30 bytes)

Total Row Size:

$8 + 50 + 30 = 88$ Bytes

2. Table Size Estimation

The table size is calculated as:

Table Size = Row Size × Number of

Rows

Assuming an **estimated 1,000 rows** for each table (this can be adjusted as per your data volume):

Table Name	Row Size (Bytes)	Row Count	Table Size (KB)
Customer	218	1,000	218 KB
Orders	23	1,000	23 KB
Tracking_Details	165	1,000	165 KB
Order_Item	23	1,000	23 KB
Payment	43	1,000	43 KB
Cart	16	1,000	16 KB
Product	251	1,000	251 KB
Category	58	1,000	58 KB
Supplier	88	1,000	88 KB

3. Database Size Estimation

To estimate the total database size, sum up the individual table sizes: Total Database Size $\approx$ 218
+ 23 + 165 + 23 + 43 + 16 + 251 + 58 + 88 = 885 KB

If we assume indexes for each table (approximately 20–30% overhead):

Index Overhead: $885 \text{ KB} \times 0.3 \approx 266 \text{ KB}$

Total Database Size (including indexes):

$885 \text{ KB} + 266 \text{ KB} = \sim 1,151 \text{ KB}$ (1.15

MB)

4. Dynamic SQL Queries to Get Table Sizes

If you want to check table sizes dynamically in a database system like Oracle, use the following:

Table Sizes

```
SELECT segment_name AS table_name,  
       bytes/1024/1024 AS size_in_mb  
FROM user_segments  
WHERE segment_type = 'TABLE';
```

Home > SQL > **SQL Commands**

☒ Autocommit Display 30 ▼

```
SELECT segment_name AS table_name,
       bytes/1024/1024 AS size_in_mb
FROM user_segments
WHERE segment_type = 'TABLE';
```

Results Explain Describe Saved SQL History

TABLE_NAME	SIZE_IN_MB
PRODUCT_T	.0625
PRODUCT_PRICE_HISTORY_T	.0625
SUPPLIER_T	.0625
ORDER_ITEM_T	.0625
CUSTOMER_T	.0625
PAYMENT_METHOD_T	.0625
PAYMENT_T	.0625
CART_T	.0625
CATEGORY_T	.0625
ORDER_T	.0625
TRACKING_DETAILS_T	.0625

11 rows returned in 0.18 seconds [CSV Export](#)

Row Counts

SELECT table_name,

num_rows FROM

user_tables;

Database Size

SELECT SUM(bytes)/1024/1024 AS database_size_in_mb

FROM user_segments;

☒ Autocommit Display 30 ▼

```

SELECT segment_name AS table_name,
       bytes/1024/1024 AS size_in_mb
FROM user_segments
WHERE segment_type = 'TABLE';

SELECT table_name,
       num_rows
FROM user_tables;

SELECT SUM(bytes)/1024/1024 AS database_size_in_mb
FROM user_segments;
  
```

Results Explain Describe Saved SQL History

DATABASE_SIZE_IN_MB
1.375

1 rows returned in 0.05 seconds [CSV Export](#)

Conclusion:

The Grocery Store Management System successfully streamlines and automates key processes involved in managing a grocery store. By integrating customer registration, product catalog management, order processing, and payment handling into a single web-based platform, the system improves operational efficiency and enhances the shopping experience. Features such as role-based access, search functionality, and real-time inventory management ensure smooth daily operations. Overall, this system provides a scalable, user-friendly, and reliable solution for modern grocery stores to manage their business effectively and meet customer needs.