Hackathon : Food Ordering

-- Create Tables

-- Customers Table

CREATE TABLE Customers (

CustomerID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(100) NOT NULL,

Address VARCHAR(200),

Contact VARCHAR(15)

);

-- Menu Table

CREATE TABLE Menu (

ItemID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(100) NOT NULL,

Category VARCHAR(50),

Price DECIMAL(10, 2) NOT NULL

);

-- Orders Table

CREATE TABLE Orders (

OrderID INT PRIMARY KEY AUTO\_INCREMENT,

CustomerID INT,

Status VARCHAR(50),

OrderDate DATE,

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

);

-- Payments Table

CREATE TABLE Payments (

PaymentID INT PRIMARY KEY AUTO\_INCREMENT,

OrderID INT,

Amount DECIMAL(10, 2),

PaymentMethod VARCHAR(50),

FOREIGN KEY (OrderID) REFERENCES Orders(OrderID)

);

-- Inventory Table

CREATE TABLE Inventory (

IngredientID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(100),

Quantity INT

);

Insert Sample Data

INSERT INTO Customers (Name, Address, Contact)

VALUES

('Arun Kumar', 'Chennai', '9876543210'),

('Priya Ramesh', 'Coimbatore', '8765432109'),

('Karthik Raja', 'Madurai', '7654321098'),

('Deepa Suresh', 'Tiruchirapalli', '6543210987'),

('Vikram Anand', 'Salem', '5432109876'),

('Anjali Devi', 'Erode', '4321098765'),

('Ravi Chandran', 'Thanjavur', '3210987654'),

('Meera Venkat', 'Vellore', '2109876543'),

('Sundar Mohan', 'Tirunelveli', '1098765432'),

('Lakshmi Narayan', 'Kanyakumari', '1987654321');

INSERT INTO Menu (Name, Category, Price)

VALUES

('Dosa', 'South Indian', 2.99),

('Idli', 'South Indian', 1.99),

('Vada', 'South Indian', 1.50),

('Pongal', 'South Indian', 3.00),

('Biryani', 'Indian', 5.99),

('Parotta', 'South Indian', 2.50),

('Sambar', 'South Indian', 1.75),

('Rasam', 'South Indian', 1.50),

('Kothu Parotta', 'South Indian', 4.00),

('Curd Rice', 'South Indian', 2.00);

INSERT INTO Orders (CustomerID, Status, OrderDate)

VALUES

(1, 'Pending', '2025-03-10'),

(2, 'Delivered', '2025-03-09'),

(3, 'Pending', '2025-03-11'),

(4, 'Cancelled', '2025-03-08'),

(5, 'Delivered', '2025-03-07'),

(6, 'Pending', '2025-03-12'),

(7, 'Delivered', '2025-03-06'),

(8, 'Pending', '2025-03-05'),

(9, 'Cancelled', '2025-03-04'),

(10, 'Delivered', '2025-03-03');

INSERT INTO Payments (OrderID, Amount, PaymentMethod)

VALUES

(1, 5.99, 'Credit Card'),

(2, 8.99, 'Cash'),

(3, 2.99, 'UPI'),

(4, 3.00, 'Net Banking'),

(5, 5.99, 'Credit Card'),

(6, 2.50, 'Cash'),

(7, 1.75, 'Debit Card'),

(8, 1.50, 'UPI'),

(9, 4.00, 'Net Banking'),

(10, 2.00, 'Cash');

INSERT INTO Inventory (Name, Quantity)

VALUES

('Rice', 100),

('Lentils', 50),

('Flour', 75),

('Sugar', 40),

('Oil', 30),

('Spices', 20),

('Vegetables', 60),

('Milk', 80),

('Curd', 90),

('Bread', 50);

Fetch Orders with Customer Details

SELECT o.OrderID, c.Name, o.Status, o.OrderDate

FROM Orders o

JOIN Customers c ON o.CustomerID = c.CustomerID;

Track Pending Deliveries

SELECT o.OrderID, c.Name, o.Status

FROM Orders o

JOIN Customers c ON o.CustomerID = c.CustomerID

WHERE o.Status = 'Pending';

Generate Sales Report

SELECT m.Name, COUNT(o.OrderID) AS TotalOrders, SUM(p.Amount) AS TotalRevenue

FROM Orders o

JOIN Payments p ON o.OrderID = p.OrderID

JOIN Menu m ON o.OrderID = m.ItemID

GROUP BY m.Name;

--Calculate Total Order Amount (Using Stored Procedure)

DELIMITER

CREATE PROCEDURE CalculateTotalOrder(IN orderID INT)

BEGIN

DECLARE totalAmount DECIMAL(10, 2);

SELECT SUM(m.Price) INTO totalAmount

FROM Orders o

JOIN Menu m ON o.OrderID = m.ItemID

WHERE o.OrderID = orderID;

SELECT totalAmount;

END

DELIMITER ;

--Call Procedure:

CALL CalculateTotalOrder(1);

--List Top-Selling Items

SELECT m.Name, COUNT(o.OrderID) AS TotalOrders

FROM Orders o

JOIN Menu m ON o.OrderID = m.ItemID

GROUP BY m.Name

ORDER BY TotalOrders DESC

LIMIT 5;