**Relational Model**

1. CUSTOMER (`CUSTOMERID`, FirstName, LastName, Email, Phone)

2. PRODUCT(`ITEMID`, ItemName, Description, Quantity, Price)

3. OrderDetails(`ORDERID`, Date, Status, TotalAmount, `CUSTOMERID`)

**FK NN**

4. PRODUCT\_ORDER (`ITEMID`, `ORDERID`, Quantity)

**FK NN FK NN**

5. DEPARTMENT (`DEPARTMENTID`, DepartmentName)

6. DEPARTMENT\_PRODUCT (`ITEMID`, `DEPARTMENTID`)

**FK NN FK NN**

7. DISCOUNT (`ID`, Name, Percentage, StartDate, EndDate)

8. PAYMENTINFO (`PAYMENTID`, CreditCardType, CreditCardNum, Expiry, CVVnum, BillingAddr, `CUSTOMERID`)

**FK NN**

**Create Database and tables**

CREATE DATABASE IF NOT EXISTS OnlineRetail;

USE OnlineRetail;

-- Customer Table

CREATE TABLE Customer (

CustomerID INT AUTO\_INCREMENT PRIMARY KEY,

FirstName VARCHAR(255) NOT NULL,

LastName VARCHAR(255) NOT NULL,

Email VARCHAR(255) NOT NULL,

Phone VARCHAR(20)

);

-- Product Table

CREATE TABLE Product (

ItemID INT AUTO\_INCREMENT PRIMARY KEY,

ItemName CHAR(255) NOT NULL,

Description TEXT,

Quantity INT,

Price FLOAT

);

-- Order Table

CREATE TABLE OrderDetails (

OrderID INT AUTO\_INCREMENT PRIMARY KEY,

DATE DATETIME,

STATUS VARCHAR(50),

TotalAmount FLOAT,

CustomerID INT NOT NULL,

FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)

);

-- Product\_Order Join Table

CREATE TABLE Product\_Order (

ItemID INT NOT NULL,

OrderID INT NOT NULL,

Quantity INT NOT NULL,

PRIMARY KEY (ItemID, OrderID),

FOREIGN KEY (ItemID) REFERENCES Product(ItemID),

FOREIGN KEY (OrderID) REFERENCES OrderDetails(OrderID)

);

-- Department Table

CREATE TABLE Department (

DepartmentID INT AUTO\_INCREMENT PRIMARY KEY,

DepartmentName VARCHAR(255) NOT NULL

);

-- Department\_Product Join Table

CREATE TABLE Department\_Product (

ItemID INT NOT NULL,

DepartmentID INT NOT NULL,

PRIMARY KEY (ItemID, DepartmentID),

FOREIGN KEY (ItemID) REFERENCES Product(ItemID),

FOREIGN KEY (DepartmentID) REFERENCES Department(DepartmentID)

);

-- Discount Table

CREATE TABLE Discount (

ID VARCHAR(50) PRIMARY KEY,

NAME VARCHAR(255) NOT NULL,

Percentage FLOAT NOT NULL,

STARTDATE DATETIME,

ENDDATE DATETIME

);

-- PaymentInfo Table

CREATE TABLE PaymentInfo (

PaymentID INT AUTO\_INCREMENT PRIMARY KEY,

CreditCardType VARCHAR(50),

CreditCardNum CHAR(16),

Expiry DATE,

CVVnum CHAR(3),

BillingAddr TEXT,

CustomerID INT NOT NULL,

FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)

);

**Insert Statements**

INSERT INTO Customer (FirstName, LastName, Email, Phone) VALUES

('John', 'Doe', 'johndoe@example.com', '555-1234'),

('Jane', 'Smith', 'janesmith@example.com', '555-5678'),

('Emily', 'Jones', 'emilyjones@example.com', '555-9012'),

('Michael', 'Brown', 'michaelbrown@example.com', '555-3456'),

('Chloe', 'Davis', 'chloedavis@example.com', '555-7890'),

('Lucas', 'Miller', 'lucasmiller@example.com', '555-6543'),

('Emma', 'Wilson', 'emmawilson@example.com', '555-6789'),

('Olivia', 'Moore', 'oliviamoore@example.com', '555-1230'),

('William', 'Taylor', 'williamtaylor@example.com', '555-4567'),

('Sophia', 'Anderson', 'sophiaanderson@example.com', '555-8912');

INSERT INTO Product (ItemName, Description, Quantity, Price) VALUES

('Laptop', '15-inch screen, 8GB RAM', 10, 1200.00),

('Smartphone', '64GB storage, 12MP camera', 20, 299.99),

('Headphones', 'Noise-cancelling, over-ear', 15, 149.99),

('Tablet', '10-inch screen, 32GB storage', 30, 450.00),

('Smartwatch', 'Water-resistant, GPS', 25, 199.99),

('Mouse', 'Wireless, ergonomic design', 50, 29.99),

('Keyboard', 'Mechanical, RGB lighting', 40, 89.99),

('Monitor', '27-inch, 4K UHD', 20, 329.99),

('Printer', 'All-in-one, wireless', 15, 109.99),

('Webcam', '1080p, autofocus', 30, 69.99);

INSERT INTO OrderDetails (Date, Status, TotalAmount, CustomerID) VALUES

('2023-01-15 14:25:00', 'Shipped', 1200.00, 1),

('2023-01-16 09:40:00', 'Processing', 449.98, 2),

('2023-01-17 11:15:00', 'Delivered', 149.99, 3),

('2023-01-18 16:45:00', 'Cancelled', 199.99, 4),

('2023-01-19 13:30:00', 'Processing', 450.00, 5);

INSERT INTO Product\_Order (ItemID, OrderID, Quantity) VALUES

(1, 1, 1),

(2, 1, 1),

(3, 2, 2),

(4, 3, 1),

(5, 4, 1),

(6, 5, 2),

(7, 5, 3),

(8, 1, 1),

(9, 2, 1),

(10, 3, 1);

INSERT INTO Department (DepartmentName) VALUES

('Electronics'),

('Accessories'),

('Home Appliances'),

('Gaming'),

('Fitness');

INSERT INTO Department\_Product (ItemID, DepartmentID) VALUES

(1, 1),

(2, 1),

(3, 2),

(4, 3),

(5, 4),

(6, 1),

(7, 1),

(8, 2),

(9, 3),

(10, 4);

INSERT INTO Discount (ID, NAME, Percentage, STARTDATE, ENDDATE) VALUES

('DIS10', 'New Year Discount', 10.00, '2023-01-01 00:00:00', '2023-01-31 23:59:59'),

('DIS15', 'Spring Sale', 15.00, '2023-03-01 00:00:00', '2023-03-31 23:59:59'),

('DIS20', 'Summer Special', 20.00, '2023-06-01 00:00:00', '2023-06-30 23:59:59'),

('DIS25', 'Fall Frenzy', 25.00, '2023-09-01 00:00:00', '2023-09-30 23:59:59'),

('DIS30', 'Winter Wonder', 30.00, '2023-12-01 00:00:00', '2023-12-31 23:59:59'),

('DIS35', 'Exclusive VIP', 35.00, '2023-02-01 00:00:00', '2023-02-28 23:59:59'),

('DIS40', 'Loyalty Reward', 40.00, '2023-04-01 00:00:00', '2023-04-30 23:59:59'),

('DIS45', 'Black Friday', 45.00, '2023-11-24 00:00:00', '2023-11-24 23:59:59');

INSERT INTO PaymentInfo (CreditCardType, CreditCardNum, Expiry, CVVnum, BillingAddr, CustomerID) VALUES

('Visa', '4111111111111111', '2025-12-01', '123', '123 Main St, Anytown, AT 12345', 1),

('MasterCard', '5500000000000004', '2024-11-01', '456', '456 Secondary Rd, Othertown, OT 67890', 2),

('Amex', '378282246310005', '2023-10-01', '987', '789 Tertiary Blvd, Thirdtown, TT 11111', 3),

('Discover', '6011111111111117', '2026-07-01', '741', '741 Quaternary Ave, Fourthtown, FT 22222', 4),

('Visa', '4111111111111111', '2028-08-01', '852', '852 Quinary St, Fifthtown, FT 33333', 5);

DELIMITER //

CREATE PROCEDURE ProcessOnlineOrder(

IN p\_CustomerID INT,

IN p\_ProductID INT,

IN p\_Quantity INT,

IN p\_DiscountID VARCHAR(50),

IN p\_CreditCardType VARCHAR(50),

IN p\_CreditCardNum CHAR(16),

IN p\_Expiry DATE,

IN p\_CVVNum CHAR(3),

IN p\_BillingAddr TEXT,

OUT p\_OrderID INT

)

BEGIN

DECLARE totalAmount FLOAT;

-- Start the transaction

START TRANSACTION;

-- Insert into OrderDetails table

INSERT INTO OrderDetails (DATE, STATUS, TotalAmount, CustomerID)

VALUES (NOW(), 'Pending', 0, p\_CustomerID);

-- Get the OrderID of the newly inserted order

SET p\_OrderID = LAST\_INSERT\_ID();

-- Calculate the total amount for the order

SELECT SUM(Quantity \* Price) INTO totalAmount

FROM Product

WHERE ItemID = p\_ProductID;

-- Apply discount if applicable

IF p\_DiscountID IS NOT NULL THEN

DECLARE discountPercentage FLOAT;

SELECT Percentage INTO discountPercentage

FROM Discount

WHERE ID = p\_DiscountID;

SET totalAmount = totalAmount - (totalAmount \* discountPercentage / 100);

END IF;

-- Update the TotalAmount in OrderDetails

UPDATE OrderDetails

SET TotalAmount = totalAmount

WHERE OrderID = p\_OrderID;

-- Update the Quantity in Product\_Order

INSERT INTO Product\_Order (ItemID, OrderID, Quantity)

VALUES (p\_ProductID, p\_OrderID, p\_Quantity);

-- Update the Quantity in the Product table

UPDATE Product

SET Quantity = Quantity - p\_Quantity

WHERE ItemID = p\_ProductID;

-- Insert into PaymentInfo table

INSERT INTO PaymentInfo (CreditCardType, CreditCardNum, Expiry, CVVnum, BillingAddr, CustomerID)

VALUES (p\_CreditCardType, p\_CreditCardNum, p\_Expiry, p\_CVVNum, p\_BillingAddr, p\_CustomerID);

-- Additional processing steps

-- Update inventory

UPDATE Inventory

SET Quantity = Quantity - p\_Quantity

WHERE ItemID = p\_ProductID;

-- Record shipment details

INSERT INTO ShipmentDetails (OrderID, ShipmentDate, Status)

VALUES (p\_OrderID, NOW(), 'Shipped');

-- Send notification to customer

INSERT INTO Notification (CustomerID, Message, SentDateTime)

VALUES (p\_CustomerID, 'Your order has been processed and shipped.', NOW());

-- Commit the transaction

COMMIT;

END //

DELIMITER ;