**1.Function and Views**

/\*==========================================================

|| Functions ||

==========================================================\*/

-- ==========================================

-- Function Name: dbo.fn\_CalculateTotalSales

-- Description: Calculates the total amount spent by a customer on all transactions

-- Inputs:

-- @CustomerID (INT) - The ID of the customer

-- Returns:

-- DECIMAL - The total amount spent by the customer

-- ==========================================

CREATE FUNCTION dbo.fn\_CalculateTotalSales (@CustomerID INT)

RETURNS DECIMAL(10, 2)

AS

BEGIN

DECLARE @TotalSales DECIMAL(10, 2);

-- Calculate the total sales for the customer by summing the FinalAmount from SalesTransaction

SELECT @TotalSales = SUM(T.FinalAmount)

FROM SalesTransaction T

WHERE T.CustomerID = @CustomerID;

-- Return the result

RETURN ISNULL(@TotalSales, 0); -- Return 0 if no sales are found

END

-- ==========================================

-- Function Name: dbo.fn\_CalculateProductSales

-- Description: Calculates the total sales for a given product

-- Inputs:

-- @ProductID (INT) - The ID of the product

-- Returns:

-- DECIMAL - The total sales for the product

-- ==========================================

CREATE FUNCTION dbo.fn\_CalculateProductSales (@ProductID INT)

RETURNS DECIMAL(10, 2)

AS

BEGIN

DECLARE @TotalSales DECIMAL(10, 2);

-- Calculate the total sales for the product by summing the LineTotal from SalesTransactionDetail

SELECT @TotalSales = SUM(D.LineTotal)

FROM SalesTransactionDetail D

WHERE D.ProductID = @ProductID;

-- Return the result

RETURN ISNULL(@TotalSales, 0); -- Return 0 if no sales are found

END

-- ==========================================

-- Function Name: dbo.fn\_GetProductAvailability

-- Description: Returns the available quantity of a product in a specific warehouse

-- Inputs:

-- @ProductID (INT) - The ID of the product

-- @WarehouseID (INT) - The ID of the warehouse

-- Returns:

-- INT - The available quantity of the product in the warehouse

-- ==========================================

CREATE FUNCTION dbo.fn\_GetProductAvailability (@ProductID INT, @WarehouseID INT)

RETURNS INT

AS

BEGIN

DECLARE @QuantityAvailable INT;

-- Get the quantity of the product in the specified warehouse

SELECT @QuantityAvailable = QuantityAvailable

FROM Stock

WHERE ProductID = @ProductID AND WarehouseID = @WarehouseID;

-- Return the result

RETURN ISNULL(@QuantityAvailable, 0); -- Return 0 if the product is not found

END

-- ==========================================

-- Function Name: dbo.fn\_GetDiscountedPrice

-- Description: Calculates the price of a product after applying a discount

-- Inputs:

-- @ProductID (INT) - The ID of the product

-- @DiscountID (INT) - The ID of the discount to be applied

-- Returns:

-- DECIMAL - The discounted price of the product

-- ==========================================

CREATE FUNCTION dbo.fn\_GetDiscountedPrice (@ProductID INT, @DiscountID INT)

RETURNS DECIMAL(10, 2)

AS

BEGIN

DECLARE @DiscountPercentage DECIMAL(5, 2);

DECLARE @UnitPrice DECIMAL(10, 2);

DECLARE @DiscountedPrice DECIMAL(10, 2);

-- Get the unit price of the product

SELECT @UnitPrice = UnitPrice

FROM Product

WHERE ProductID = @ProductID;

-- Get the discount percentage for the given discount ID

SELECT @DiscountPercentage = DiscountPercentage

FROM Discount

WHERE DiscountID = @DiscountID;

-- Calculate the discounted price

SET @DiscountedPrice = @UnitPrice - (@UnitPrice \* @DiscountPercentage / 100);

-- Return the discounted price

RETURN @DiscountedPrice;

END

/\*==========================================================

|| Views ||

==========================================================\*/

-- ==========================================

-- View Name: vw\_CustomerSalesSummary

-- Description: Summary view of total sales and transaction count for each customer

-- ==========================================

CREATE VIEW vw\_CustomerSalesSummary AS

SELECT

C.CustomerID,

C.FirstName + ' ' + C.LastName AS CustomerName,

COUNT(T.TransactionID) AS TotalTransactions,

SUM(T.FinalAmount) AS TotalSpent,

MAX(T.TransactionDate) AS LastTransactionDate

FROM

Customer C

LEFT JOIN SalesTransaction T ON C.CustomerID = T.CustomerID

GROUP BY

C.CustomerID, C.FirstName, C.LastName;

-- ==========================================

-- View Name: vw\_ProductSalesOverview

-- Description: Provides an overview of products with total sales and current stock

-- ==========================================

CREATE VIEW vw\_ProductSalesOverview AS

SELECT

P.ProductID,

P.ProductName,

ISNULL(SUM(D.LineTotal), 0) AS TotalSales,

ISNULL(S.QuantityAvailable, 0) AS StockAvailable

FROM

Product P

LEFT JOIN SalesTransactionDetail D ON P.ProductID = D.ProductID

LEFT JOIN Stock S ON P.ProductID = S.ProductID

GROUP BY

P.ProductID, P.ProductName, S.QuantityAvailable;

-- ==========================================

-- View Name: vw\_EmployeeRoleSummary

-- Description: Provides employee details along with their roles and contact info

-- ==========================================

CREATE VIEW vw\_EmployeeRoleSummary AS

SELECT

E.EmployeeID,

E.FirstName + ' ' + E.LastName AS EmployeeName,

R.RoleName,

E.Email,

E.PhoneNumber,

E.HireDate

FROM

Employee E

JOIN Role R ON E.RoleID = R.RoleID;

-- ==========================================

-- View Name: vw\_SalesTransactionDetails

-- Description: Provides details of each sales transaction, including product info

-- ==========================================

CREATE VIEW vw\_SalesTransactionDetails AS

SELECT

T.TransactionID,

C.FirstName + ' ' + C.LastName AS CustomerName,

P.ProductName,

D.Quantity,

D.UnitPrice,

D.LineTotal,

T.TransactionDate

FROM

SalesTransaction T

JOIN Customer C ON T.CustomerID = C.CustomerID

JOIN SalesTransactionDetail D ON T.TransactionID = D.TransactionID

JOIN Product P ON D.ProductID = P.ProductID;  
  
  
  
  
  
  
**2) Procedures & Triggers**  
  
/\*==========================================================

|| Procedure ||

==========================================================\*/

-- ==========================================

-- Procedure Name: sp\_InsertCustomer

-- Description: Inserts a new customer into the Customer table and logs the action in AuditLog

-- Inputs:

-- @FirstName (NVARCHAR) - First name of the customer

-- @LastName (NVARCHAR) - Last name of the customer

-- @Email (NVARCHAR) - Email of the customer (must be unique)

-- @PhoneNumber (NVARCHAR) - Phone number of the customer (optional)

-- @AddressLine1, @AddressLine2, @City, @PostalCode - Customer address details (optional)

-- ==========================================

CREATE PROCEDURE sp\_InsertCustomer

@FirstName NVARCHAR(100),

@LastName NVARCHAR(100),

@Email NVARCHAR(255),

@PhoneNumber NVARCHAR(15) = NULL,

@AddressLine1 NVARCHAR(255) = NULL,

@AddressLine2 NVARCHAR(255) = NULL,

@City NVARCHAR(100) = NULL,

@PostalCode NVARCHAR(20) = NULL

AS

BEGIN

-- Begin error handling block

BEGIN TRY

-- Insert new customer into the Customer table

INSERT INTO Customer

(FirstName, LastName, Email, PhoneNumber, AddressLine1, AddressLine2, City, PostalCode, RegistrationDate)

VALUES

(@FirstName, @LastName, @Email, @PhoneNumber, @AddressLine1, @AddressLine2, @City, @PostalCode, GETDATE());

-- Get the CustomerID of the newly inserted record

DECLARE @CustomerID INT = SCOPE\_IDENTITY();

-- Log the action in the AuditLog

INSERT INTO AuditLog (Action, PerformedBy, ActionTimestamp, TableName)

VALUES ('Insert New Customer', 1, GETDATE(), 'Customer'); -- Assuming PerformedBy = 1 for now

-- Print success message

PRINT 'Customer inserted successfully with CustomerID = ' + CAST(@CustomerID AS NVARCHAR(20));

END TRY

BEGIN CATCH

-- Print error message

PRINT 'Error: ' + ERROR\_MESSAGE();

END CATCH

END

-- ==========================================

-- Procedure Name: sp\_GenerateSalesReport

-- Description: Generate a sales report for a given date range

-- Inputs:

-- @StartDate (DATE) - The start date of the report

-- @EndDate (DATE) - The end date of the report

-- ==========================================

CREATE PROCEDURE sp\_GenerateSalesReport

@StartDate DATE,

@EndDate DATE

AS

BEGIN

-- Begin error handling block

BEGIN TRY

-- Select all transactions that happened in the date range

SELECT

T.TransactionID,

C.FirstName + ' ' + C.LastName AS CustomerName,

T.TransactionDate,

P.ProductName,

D.Quantity,

D.LineTotal

FROM

SalesTransaction T

JOIN Customer C ON T.CustomerID = C.CustomerID

JOIN SalesTransactionDetail D ON T.TransactionID = D.TransactionID

JOIN Product P ON D.ProductID = P.ProductID

WHERE

T.TransactionDate BETWEEN @StartDate AND @EndDate;

-- Log the report generation

INSERT INTO AuditLog (Action, PerformedBy, ActionTimestamp, TableName)

VALUES ('Generate Sales Report', 1, GETDATE(), 'SalesTransaction');

END TRY

BEGIN CATCH

-- Print error message

PRINT 'Error: ' + ERROR\_MESSAGE();

END CATCH

END

-- ==========================================

-- Procedure Name: sp\_AssignProductDiscount

-- Description: Assigns a discount to a product and ensures no duplicates

-- Inputs:

-- @ProductID (INT) - The ID of the product to which the discount will be assigned

-- @DiscountID (INT) - The ID of the discount to be applied

-- ==========================================

CREATE PROCEDURE sp\_AssignProductDiscount

@ProductID INT,

@DiscountID INT

AS

BEGIN

-- Begin error handling block

BEGIN TRY

-- Check if this product-discount combination already exists

IF NOT EXISTS (

SELECT 1

FROM ProductDiscount

WHERE ProductID = @ProductID AND DiscountID = @DiscountID

)

BEGIN

-- Insert a new product-discount association

INSERT INTO ProductDiscount (ProductID, DiscountID)

VALUES (@ProductID, @DiscountID);

-- Print success message

PRINT 'Product assigned to discount successfully.';

END

ELSE

BEGIN

-- Notify if the product-discount combo already exists

PRINT 'Product is already assigned to this discount.';

END

END TRY

BEGIN CATCH

-- Print error message

PRINT 'Error: ' + ERROR\_MESSAGE();

END CATCH

END

/\*==========================================================

|| Triggers ||

==========================================================\*/

-- ==========================================

-- Trigger Name: trg\_UpdateStockAfterSale

-- Description: After a new sale, reduce the stock of the sold product

-- Trigger Fires: AFTER INSERT on SalesTransactionDetail

-- ==========================================

CREATE TRIGGER trg\_UpdateStockAfterSale

ON SalesTransactionDetail

AFTER INSERT

AS

BEGIN

-- Turn off additional result sets for the trigger

SET NOCOUNT ON;

-- Reduce the stock quantity based on the product and quantity sold

UPDATE S

SET S.QuantityAvailable = S.QuantityAvailable - I.Quantity

FROM Stock S

INNER JOIN INSERTED I ON S.ProductID = I.ProductID

WHERE S.WarehouseID = 1; -- Adjust WarehouseID as required

-- Log the stock update action

INSERT INTO AuditLog (Action, PerformedBy, ActionTimestamp, TableName)

VALUES ('Stock Updated After Sale', 1, GETDATE(), 'Stock');

END

-- ==========================================

-- Trigger Name: trg\_ReorderNotification

-- Description: Triggers when stock is updated, logs an alert if quantity falls below reorder level

-- Trigger Fires: AFTER UPDATE on Stock

-- ==========================================

CREATE TRIGGER trg\_ReorderNotification

ON Stock

AFTER UPDATE

AS

BEGIN

-- Turn off additional result sets for the trigger

SET NOCOUNT ON;

-- Insert an alert into the AuditLog if stock is below reorder level

INSERT INTO AuditLog (Action, PerformedBy, ActionTimestamp, TableName)

SELECT

'Reorder Alert: Stock Low for ProductID ' + CAST(I.ProductID AS NVARCHAR(10)),

1,

GETDATE(),

'Stock'

FROM INSERTED I

JOIN Product P ON I.ProductID = P.ProductID

WHERE I.QuantityAvailable < P.ReorderLevel;

END

-- ==========================================

-- Trigger Name: trg\_LogCustomerDeletion

-- Description: Logs the deletion of a customer in the AuditLog

-- Trigger Fires: AFTER DELETE on Customer

-- ==========================================

CREATE TRIGGER trg\_LogCustomerDeletion

ON Customer

AFTER DELETE

AS

BEGIN

-- Turn off additional result sets for the trigger

SET NOCOUNT ON;

-- Insert log into AuditLog table

INSERT INTO AuditLog (Action, PerformedBy, ActionTimestamp, TableName)

SELECT

'Deleted Customer: ' + D.FirstName + ' ' + D.LastName + ' (Email: ' + D.Email + ')',

1, -- Assuming PerformedBy = 1 (can be adjusted based on user context)

GETDATE(),

'Customer'

FROM DELETED D;

END  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
**3)create tables**  
  
/\*==========================================================

|| CUSTOMER MANAGEMENT ||

==========================================================\*/

/\*-----------------------------------------------

-- Creating Customer table

-----------------------------------------------\*/

CREATE TABLE Customer (

CustomerID INT PRIMARY KEY IDENTITY(1,1),

FirstName NVARCHAR(100) NOT NULL CHECK (LEN(FirstName) > 1),

LastName NVARCHAR(100) NOT NULL CHECK (LEN(LastName) > 1),

Email NVARCHAR(255) UNIQUE CHECK (Email LIKE '%@%.%'),

PhoneNumber NVARCHAR(15) CHECK (PhoneNumber LIKE '[0-9]%'),

AddressLine1 NVARCHAR(255),

AddressLine2 NVARCHAR(255),

City NVARCHAR(100),

PostalCode NVARCHAR(20) CHECK (LEN(PostalCode) <= 20),

RegistrationDate DATE NOT NULL DEFAULT GETDATE()

);

/\*==========================================================

|| REPORTING AND ADMINISTRATIVE DATA ||

==========================================================\*/

/\*-----------------------------------------------

-- Creating Role table

-----------------------------------------------\*/

CREATE TABLE Role (

RoleID INT PRIMARY KEY IDENTITY(1,1),

RoleName NVARCHAR(100) NOT NULL CHECK (LEN(RoleName) > 0)

);

/\*-----------------------------------------------

-- Creating Employee table

-----------------------------------------------\*/

CREATE TABLE Employee (

EmployeeID INT PRIMARY KEY IDENTITY(1,1),

FirstName NVARCHAR(100) NOT NULL CHECK (LEN(FirstName) > 1),

LastName NVARCHAR(100) NOT NULL CHECK (LEN(LastName) > 1),

RoleID INT NOT NULL,

Email NVARCHAR(255) UNIQUE NOT NULL CHECK (Email LIKE '%@%.%'),

PhoneNumber NVARCHAR(15) CHECK (PhoneNumber LIKE '[0-9]%'),

HireDate DATE NOT NULL CHECK (HireDate <= GETDATE()),

FOREIGN KEY (RoleID) REFERENCES Role(RoleID) ON DELETE CASCADE

);

/\*-----------------------------------------------

-- Creating AuditLog table

-----------------------------------------------\*/

CREATE TABLE AuditLog (

LogID INT PRIMARY KEY IDENTITY(1,1),

Action NVARCHAR(255) NOT NULL,

PerformedBy INT NULL, -- Allow NULL values for SET NULL action

ActionTimestamp DATETIME NOT NULL,

TableName NVARCHAR(100),

FOREIGN KEY (PerformedBy) REFERENCES Employee(EmployeeID) ON DELETE SET NULL

);

/\*-----------------------------------------------

-- Creating Report table

-----------------------------------------------\*/

CREATE TABLE Report (

ReportID INT PRIMARY KEY IDENTITY(1,1),

ReportName NVARCHAR(255) NOT NULL CHECK (LEN(ReportName) > 0),

GeneratedBy INT NOT NULL,

GeneratedDate DATETIME NOT NULL DEFAULT GETDATE(),

ReportType NVARCHAR(100) NOT NULL,

ReportData NVARCHAR(MAX),

FOREIGN KEY (GeneratedBy) REFERENCES Employee(EmployeeID) ON DELETE CASCADE

);

/\*==========================================================

|| PRODUCT AND STOCK MANAGEMENT MODULE ||

==========================================================\*/

/\*-----------------------------------------------

-- Creating Category table

-----------------------------------------------\*/

CREATE TABLE Category (

CategoryID INT PRIMARY KEY IDENTITY(1,1),

CategoryName NVARCHAR(100) NOT NULL UNIQUE CHECK (LEN(CategoryName) > 0)

);

/\*-----------------------------------------------

-- Creating Brand table

-----------------------------------------------\*/

CREATE TABLE Brand (

BrandID INT PRIMARY KEY IDENTITY(1,1),

BrandName NVARCHAR(100) NOT NULL UNIQUE CHECK (LEN(BrandName) > 0)

);

/\*-----------------------------------------------

-- Creating TaxRate table

-----------------------------------------------\*/

CREATE TABLE TaxRate (

TaxRateID INT PRIMARY KEY IDENTITY(1,1),

TaxRate DECIMAL(5, 2) NOT NULL CHECK (TaxRate >= 0 AND TaxRate <= 100)

);

/\*-----------------------------------------------

-- Creating Product table

-----------------------------------------------\*/

CREATE TABLE Product (

ProductID INT PRIMARY KEY IDENTITY(1,1),

ProductName NVARCHAR(150) NOT NULL CHECK (LEN(ProductName) > 0),

CategoryID INT NOT NULL,

BrandID INT NOT NULL,

UnitPrice DECIMAL(10, 2) NOT NULL CHECK (UnitPrice >= 0),

TaxRateID INT NOT NULL,

ReorderLevel INT NOT NULL CHECK (ReorderLevel >= 0),

IsActive BIT NOT NULL DEFAULT 1,

FOREIGN KEY (CategoryID) REFERENCES Category(CategoryID) ON DELETE CASCADE,

FOREIGN KEY (BrandID) REFERENCES Brand(BrandID) ON DELETE CASCADE,

FOREIGN KEY (TaxRateID) REFERENCES TaxRate(TaxRateID) ON DELETE CASCADE

);

/\*-----------------------------------------------

-- Creating Warehouse table

-----------------------------------------------\*/

CREATE TABLE Warehouse (

WarehouseID INT PRIMARY KEY IDENTITY(1,1),

WarehouseName NVARCHAR(150) NOT NULL CHECK (LEN(WarehouseName) > 0),

Location NVARCHAR(255),

ManagerID INT,

FOREIGN KEY (ManagerID) REFERENCES Employee(EmployeeID) ON DELETE SET NULL

);

/\*-----------------------------------------------

-- Creating Stock table

-----------------------------------------------\*/

CREATE TABLE Stock (

StockID INT PRIMARY KEY IDENTITY(1,1),

ProductID INT NOT NULL,

WarehouseID INT NOT NULL,

QuantityAvailable INT NOT NULL CHECK (QuantityAvailable >= 0),

LastRestockedDate DATE CHECK (LastRestockedDate <= GETDATE()),

FOREIGN KEY (ProductID) REFERENCES Product(ProductID) ON DELETE CASCADE,

FOREIGN KEY (WarehouseID) REFERENCES Warehouse(WarehouseID) ON DELETE CASCADE

);

/\*==========================================================

|| DISCOUNT MANAGEMENT MODULE ||

==========================================================\*/

/\*-----------------------------------------------

-- Creating Discount table

-----------------------------------------------\*/

CREATE TABLE Discount (

DiscountID INT PRIMARY KEY IDENTITY(1,1),

DiscountName NVARCHAR(150) NOT NULL,

StartDate DATE NOT NULL,

EndDate DATE NOT NULL,

DiscountPercentage DECIMAL(5, 2),

DiscountType NVARCHAR(50) NOT NULL,

CONSTRAINT CK\_Discount\_DateRange CHECK (EndDate >= StartDate) -- Table-level CHECK constraint

);

/\*-----------------------------------------------

-- Creating ProductDiscount table

-----------------------------------------------\*/

CREATE TABLE ProductDiscount (

ProductDiscountID INT PRIMARY KEY IDENTITY(1,1),

ProductID INT NOT NULL,

DiscountID INT NOT NULL,

FOREIGN KEY (ProductID) REFERENCES Product(ProductID) ON DELETE CASCADE,

FOREIGN KEY (DiscountID) REFERENCES Discount(DiscountID) ON DELETE CASCADE

);

/\*==========================================================

|| SALES MANAGEMENT MODULE ||

==========================================================\*/

/\*-----------------------------------------------

-- Creating PaymentMethod table

-----------------------------------------------\*/

CREATE TABLE PaymentMethod (

PaymentMethodID INT PRIMARY KEY IDENTITY(1,1),

PaymentType NVARCHAR(100) NOT NULL CHECK (LEN(PaymentType) > 0)

);

/\*-----------------------------------------------

-- Creating SalesTransaction table

-----------------------------------------------\*/

CREATE TABLE SalesTransaction (

TransactionID INT PRIMARY KEY IDENTITY(1,1),

CustomerID INT,

TransactionDate DATETIME NOT NULL DEFAULT GETDATE(),

PaymentMethodID INT NOT NULL,

TotalAmount DECIMAL(10, 2) NOT NULL CHECK (TotalAmount >= 0),

DiscountApplied DECIMAL(10, 2) CHECK (DiscountApplied >= 0),

FinalAmount DECIMAL(10, 2) NOT NULL CHECK (FinalAmount >= 0),

FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID) ON DELETE SET NULL,

FOREIGN KEY (PaymentMethodID) REFERENCES PaymentMethod(PaymentMethodID) ON DELETE CASCADE

);

/\*-----------------------------------------------

-- Creating SalesTransactionDetail table

-----------------------------------------------\*/

CREATE TABLE SalesTransactionDetail (

TransactionDetailID INT PRIMARY KEY IDENTITY(1,1),

TransactionID INT NOT NULL,

ProductID INT NOT NULL,

Quantity INT NOT NULL CHECK (Quantity > 0),

UnitPrice DECIMAL(10, 2) NOT NULL CHECK (UnitPrice >= 0),

LineTotal DECIMAL(10, 2) NOT NULL CHECK (LineTotal >= 0),

FOREIGN KEY (TransactionID) REFERENCES SalesTransaction(TransactionID) ON DELETE CASCADE,

FOREIGN KEY (ProductID) REFERENCES Product(ProductID) ON DELETE CASCADE

);

**4)sample data**

/\*==========================================================

|| CUSTOMER MANAGEMENT ||

==========================================================\*/

/\*-----------------------------------------------

-- Adding data to Customer table

-----------------------------------------------\*/

SET IDENTITY\_INSERT Customer ON;

INSERT INTO Customer (CustomerID, FirstName, LastName, Email, PhoneNumber, AddressLine1, AddressLine2, City, PostalCode, RegistrationDate)

VALUES

(1, 'Kumar', 'Perera', 'kumar.perera@example.com', '0771234567', '123 Temple Road', 'Apartment 4B', 'Colombo', '00100', '2023-01-15'),

(2, 'Nuwan', 'Fernando', 'nuwan.fernando@example.com', '0719876543', '45 Beach Drive', NULL, 'Galle', '80000', '2023-02-10'),

(3, 'Amali', 'Wijesinghe', 'amali.wijesinghe@example.com', '0723456789', '78 Lake View', 'Suite 3', 'Kandy', '20000', '2023-03-05'),

(4, 'Saman', 'Silva', 'saman.silva@example.com', '0751234560', '89 High Street', NULL, 'Matara', '81000', '2023-04-12'),

(5, 'Lakshmi', 'De Silva', 'lakshmi.desilva@example.com', '0762345671', '15 Palm Grove', 'Near Park', 'Negombo', '11500', '2023-05-18'),

(6, 'Ruwan', 'Jayasinghe', 'ruwan.jayasinghe@example.com', '0783456782', '52 Hill Crest', 'Flat 6A', 'Nuwara Eliya', '22200', '2023-06-07'),

(7, 'Chathura', 'Karunaratne', 'chathura.k@example.com', '0709876544', '102 Green Lane', NULL, 'Jaffna', '40000', '2023-07-02'),

(8, 'Dilini', 'Gunasekara', 'dilini.g@example.com', '0775671234', '56 River Side', 'Opposite School', 'Batticaloa', '30000', '2023-08-19'),

(9, 'Thilini', 'Ratnayake', 'thilini.ratnayake@example.com', '0743456789', '28 Mountain Pass', NULL, 'Badulla', '90000', '2023-09-03'),

(10, 'Prasanna', 'Abeywickrama', 'prasanna.abeywickrama@example.com', '0712349876', '16 Lotus Street', 'Near Temple', 'Kurunegala', '60000', '2023-10-01'),

(11, 'Harsha', 'Senanayake', 'harsha.senanayake@example.com', '0721234568', '9 Orchid Path', NULL, 'Ratnapura', '70000', '2023-10-15'),

(12, 'Shanika', 'Dias', 'shanika.d@example.com', '0784567890', '67 Cinnamon Drive', NULL, 'Trincomalee', '31000', '2023-11-20'),

(13, 'Anjana', 'Bandara', 'anjana.bandara@example.com', '0702345679', '34 Jasmine Avenue', 'Apartment 5C', 'Anuradhapura', '50000', '2023-12-01'),

(14, 'Kavindu', 'Ekanayake', 'kavindu.ekanayake@example.com', '0753456781', '77 Sunset Boulevard', 'Flat 2B', 'Polonnaruwa', '51000', '2023-12-05'),

(15, 'Nadeeka', 'Rajapakse', 'nadeeka.r@example.com', '0715678901', '101 Palm Court', 'Opposite Market', 'Hambantota', '82000', '2023-12-08'),

(16, 'Sunil', 'Herath', 'sunil.herath@example.com', '0767890123', '12 Garden Path', NULL, 'Kalutara', '12000', '2023-12-12'),

(17, 'Malith', 'Gamage', 'malith.gamage@example.com', '0778901234', '24 Cliff Road', 'Suite 101', 'Ampara', '32000', '2023-12-15'),

(18, 'Rashmi', 'Liyanage', 'rashmi.liyanage@example.com', '0759012345', '88 Oak Lane', NULL, 'Puttalam', '61000', '2023-12-20'),

(19, 'Janaka', 'Dissanayake', 'janaka.d@example.com', '0701237890', '40 Main Street', 'Building A', 'Vavuniya', '43000', '2023-12-25'),

(20, 'Ishara', 'Peiris', 'ishara.peiris@example.com', '0713456782', '17 Cherry Blossom', NULL, 'Kilinochchi', '44000', '2023-12-30');

SET IDENTITY\_INSERT Customer OFF;

/\*==========================================================

|| EMPLOYEE AND ROLE MANAGEMENT ||

==========================================================\*/

/\*-----------------------------------------------

-- Adding data to Role table

-----------------------------------------------\*/

SET IDENTITY\_INSERT Role ON;

INSERT INTO Role (RoleID, RoleName)

VALUES

(1, 'Admin'),

(2, 'Cashier'),

(3, 'Manager'),

(4, 'Inventory Supervisor'),

(5, 'Sales Representative'),

(6, 'Accountant'),

(7, 'Warehouse Manager'),

(8, 'Customer Service Representative'),

(9, 'IT Support Specialist'),

(10, 'Marketing Executive'),

(11, 'Delivery Coordinator');

SET IDENTITY\_INSERT Role OFF;

/\*-----------------------------------------------

-- Adding data to Employee table

-----------------------------------------------\*/

SET IDENTITY\_INSERT Employee ON;

INSERT INTO Employee (EmployeeID, FirstName, LastName, RoleID, Email, PhoneNumber, HireDate)

VALUES

-- Admins

(1, 'Samantha', 'Wijeratne', 1, 'samantha.wijeratne@pos.lk', '0711234567', '2022-01-05'),

(2, 'Dilshan', 'Edirisinghe', 1, 'dilshan.edirisinghe@pos.lk', '0741234567', '2022-05-15'),

-- Cashiers

(3, 'Dinesh', 'Kumarasinghe', 2, 'dinesh.kumarasinghe@pos.lk', '0772345678', '2022-03-12'),

(4, 'Sachini', 'Rajapaksa', 2, 'sachini.rajapaksa@pos.lk', '0750123456', '2022-04-08'),

-- Managers

(5, 'Tharindu', 'Jayasinghe', 3, 'tharindu.jayasinghe@pos.lk', '0703456789', '2021-05-10'),

(6, 'Ruwan', 'Senanayake', 3, 'ruwan.senanayake@pos.lk', '0717890123', '2023-01-10'),

-- Inventory Supervisors

(7, 'Chamodi', 'Fernando', 4, 'chamodi.fernando@pos.lk', '0784567890', '2022-06-15'),

(8, 'Isuru', 'Bandara', 4, 'isuru.bandara@pos.lk', '0789012345', '2023-03-01'),

-- Sales Representatives

(9, 'Kasun', 'Perera', 5, 'kasun.perera@pos.lk', '0755678901', '2022-07-20'),

(10, 'Thilini', 'Wijesinghe', 5, 'thilini.wijesinghe@pos.lk', '0785678901', '2022-09-12'),

-- Accountants

(11, 'Nishadi', 'De Silva', 6, 'nishadi.desilva@pos.lk', '0746789012', '2021-08-25'),

(12, 'Sanduni', 'Wickramasinghe', 6, 'sanduni.wickramasinghe@pos.lk', '0758901234', '2022-12-12'),

-- Warehouse Managers

(13, 'Ajith', 'Kumarasinghe', 7, 'ajith.kumarasinghe@pos.lk', '0771234561', '2023-01-15'),

(14, 'Kavinda', 'Perera', 7, 'kavinda.perera@pos.lk', '0719876523', '2022-05-18'),

(15, 'Nadeesha', 'Fernando', 7, 'nadeesha.fernando@pos.lk', '0703456729', '2022-08-25'),

(16, 'Priyanka', 'De Silva', 7, 'priyanka.desilva@pos.lk', '0742345671', '2021-10-10'),

(17, 'Sunil', 'Weerasinghe', 7, 'sunil.weerasinghe@pos.lk', '0784567888', '2022-03-30'),

(18, 'Rasika', 'Jayawardena', 7, 'rasika.jayawardena@pos.lk', '0775678998', '2022-06-12'),

(19, 'Tharaka', 'Senanayake', 7, 'tharaka.senanayake@pos.lk', '0746789001', '2022-09-17'),

(20, 'Chathura', 'Gunathilaka', 7, 'chathura.gunathilaka@pos.lk', '0787891234', '2023-04-22'),

(21, 'Janaka', 'Dissanayake', 7, 'janaka.dissanayake@pos.lk', '0758901239', '2022-02-14'),

(22, 'Dilini', 'Ranawaka', 7, 'dilini.ranawaka@pos.lk', '0719012346', '2023-07-19'),

(23, 'Chaminda', 'Senarath', 7, 'chaminda.senarath@pos.lk', '0779123450', '2022-11-05'),

(24, 'Sanjeewa', 'Wickramaratne', 7, 'sanjeewa.wickramaratne@pos.lk', '0701234568', '2023-01-25'),

(25, 'Ruwanthi', 'Karunathilaka', 7, 'ruwanthi.karunathilaka@pos.lk', '0782345679', '2022-12-01'),

(26, 'Dulanjali', 'Madushanka', 7, 'dulanjali.madushanka@pos.lk', '0713456789', '2022-07-15'),

(27, 'Lakshman', 'Gunasekara', 7, 'lakshman.gunasekara@pos.lk', '0744567890', '2022-04-09'),

(28, 'Shanika', 'Wijerathne', 7, 'shanika.wijerathne@pos.lk', '0755678902', '2022-08-20'),

(29, 'Supun', 'Ranasinghe', 7, 'supun.ranasinghe@pos.lk', '0706789013', '2023-02-11'),

(30, 'Harindra', 'Wijesooriya', 7, 'harindra.wijesooriya@pos.lk', '0787890124', '2023-03-27'),

(31, 'Menaka', 'Jayasekara', 7, 'menaka.jayasekara@pos.lk', '0758901235', '2022-06-30'),

(32, 'Ishani', 'Wimalasuriya', 7, 'ishani.wimalasuriya@pos.lk', '0719012347', '2022-12-15'),

-- Customer Service Representatives

(33, 'Amali', 'Gunaratne', 8, 'amali.gunaratne@pos.lk', '0708901234', '2023-02-05'),

(34, 'Lakshan', 'Fernando', 8, 'lakshan.fernando@pos.lk', '0709012345', '2023-01-20'),

-- IT Support Specialists

(35, 'Sameera', 'Ranasinghe', 9, 'sameera.ranasinghe@pos.lk', '0746789012', '2023-10-09'),

-- Marketing Executives

(36, 'Ashani', 'Mendis', 10, 'ashani.mendis@pos.lk', '0704567890', '2023-08-01'),

-- Delivery Coordinators

(37, 'Dilini', 'Ratnayake', 11, 'dilini.ratnayake@pos.lk', '0780123456', '2023-02-25'),

(38, 'Tharuka', 'Wijesooriya', 11, 'tharuka.wijesooriya@pos.lk', '0756783456', '2023-03-15'),

(39, 'Mahesh', 'Senarath', 11, 'mahesh.senarath@pos.lk', '0719234567', '2023-04-10');

SET IDENTITY\_INSERT Employee OFF;

/\*-----------------------------------------------

-- Adding data to AuditLog table

-----------------------------------------------\*/

SET IDENTITY\_INSERT AuditLog ON;

INSERT INTO AuditLog (LogID, Action, PerformedBy, ActionTimestamp, TableName)

VALUES

-- Admin Actions

(1, 'Added new employee: Amali Gunaratne', 1, '2023-02-05 09:15:30', 'Employee'),

(2, 'Updated product price for Ceylon Tea 500g', 2, '2023-06-12 10:22:15', 'Product'),

(3, 'Deleted inactive customer record', 1, '2023-08-01 14:45:05', 'Customer'),

-- Cashier Actions

(4, 'Processed sales transaction #10123', 3, '2023-05-25 13:34:50', 'SalesTransaction'),

(5, 'Applied discount on sales transaction #10256', 4, '2023-06-18 15:25:10', 'SalesTransactionDetail'),

(6, 'Updated payment method for transaction #10321', 4, '2023-07-02 16:40:00', 'PaymentMethod'),

-- Manager Actions

(7, 'Approved bulk order request for Lanka Spices Ltd.', 5, '2023-09-10 11:10:45', 'Stock'),

(8, 'Assigned new warehouse manager to Warehouse #3', 6, '2023-10-20 10:00:15', 'Warehouse'),

(9, 'Updated stock reorder level for Rice 10kg', 5, '2023-11-11 09:45:30', 'Product'),

-- Inventory Supervisor Actions

(10, 'Stock restocked for Product #205 (Milk Powder)', 7, '2023-08-23 14:15:55', 'Stock'),

(11, 'Updated supplier details for Product #109 (Coconut Oil)', 8, '2023-09-30 12:10:25', 'Product'),

(12, 'Performed monthly stock audit for Warehouse #1', 7, '2023-12-01 11:00:00', 'AuditLog'),

-- Sales Representative Actions

(13, 'Generated sales report for July 2023', 9, '2023-07-31 17:45:30', 'Report'),

(14, 'Generated promotional sales summary', 10, '2023-09-12 18:00:00', 'Report'),

(15, 'Submitted customer feedback report', 9, '2023-10-18 15:20:15', 'Report'),

-- Accountants Actions

(16, 'Processed supplier invoice #20435', 11, '2023-09-05 10:05:40', 'AuditLog'),

(17, 'Generated tax report for Q2 2023', 12, '2023-07-01 12:45:55', 'Report'),

(18, 'Corrected payment record for Invoice #20987', 11, '2023-08-20 14:30:25', 'PaymentMethod'),

-- Warehouse Manager Actions

(19, 'Added new stock location: Warehouse #4', 13, '2023-04-01 10:20:00', 'Warehouse'),

(20, 'Updated manager details for Warehouse #2', 14, '2023-05-18 15:15:35', 'Warehouse'),

(21, 'Approved stock transfer to Warehouse #3', 15, '2023-06-22 09:50:30', 'Stock'),

-- Customer Service Actions

(22, 'Resolved customer complaint for Order #23145', 33, '2023-07-25 11:30:15', 'Customer'),

(23, 'Updated customer phone number for account #12345', 34, '2023-08-10 14:05:50', 'Customer'),

-- IT Support Actions

(24, 'Performed system backup for Q3 2023', 35, '2023-10-15 20:00:00', 'AuditLog'),

(25, 'Restored customer database from backup', 35, '2023-11-20 22:10:05', 'Customer');

SET IDENTITY\_INSERT AuditLog OFF;

/\*==========================================================

|| REPORT GENERATION MODULE ||

==========================================================\*/

/\*-----------------------------------------------

-- Adding data to Report table

-----------------------------------------------\*/

SET IDENTITY\_INSERT Report ON;

INSERT INTO Report (ReportID, ReportName, GeneratedBy, GeneratedDate, ReportType, ReportData)

VALUES

-- Daily Sales Reports

(1, 'Daily Sales Report - 2023-09-15', 9, '2023-09-15 20:10:00', 'Daily', '{TotalSales: 456700, Transactions: 158}'),

(2, 'Daily Sales Report - 2023-09-16', 10, '2023-09-16 20:10:00', 'Daily', '{TotalSales: 512300, Transactions: 172}'),

(3, 'Daily Sales Report - 2023-09-17', 9, '2023-09-17 20:10:00', 'Daily', '{TotalSales: 478200, Transactions: 164}'),

-- Monthly Sales Summary

(4, 'Monthly Sales Summary - August 2023', 12, '2023-09-01 09:15:00', 'Monthly', '{TotalSales: 14567000, TotalTransactions: 4520, TopProduct: "Ceylon Tea 500g"}'),

(5, 'Monthly Sales Summary - September 2023', 12, '2023-10-01 09:15:00', 'Monthly', '{TotalSales: 15784000, TotalTransactions: 4805, TopProduct: "Rice 10kg"}'),

-- Stock and Inventory Reports

(6, 'Low Stock Alert - September 2023', 7, '2023-09-20 14:30:00', 'Monthly', '{Products: ["Milk Powder", "Sugar", "Flour"]}'),

(7, 'Stock Reorder Summary - October 2023', 8, '2023-10-05 14:30:00', 'Monthly', '{Products: ["Rice 5kg", "Coconut Oil"]}'),

-- Customer Reports

(8, 'New Customer Registrations - September 2023', 33, '2023-09-30 16:00:00', 'Monthly', '{TotalCustomers: 325, NewCustomers: 115}'),

(9, 'Inactive Customers Report - Q3 2023', 34, '2023-10-05 16:30:00', 'Quarterly', '{InactiveCustomers: 45}'),

-- Financial Reports

(10, 'Profit & Loss Statement - August 2023', 11, '2023-09-01 12:45:00', 'Monthly', '{Revenue: 15784000, Expenses: 10230000, NetProfit: 5554000}'),

(11, 'Tax Summary Report - Q3 2023', 12, '2023-10-10 13:00:00', 'Quarterly', '{VAT: 2045000, OtherTaxes: 150000}'),

-- Employee Performance Reports

(12, 'Employee Performance - Sales Reps Q3 2023', 5, '2023-10-12 10:45:00', 'Quarterly', '{TopEmployee: "Kasun Perera", TotalSales: 740500}'),

(13, 'Employee Attendance - September 2023', 6, '2023-09-30 17:00:00', 'Monthly', '{TotalAbsences: 12, BestAttendance: "Tharindu Jayasinghe"}'),

-- Promotional Campaigns and Marketing

(14, 'Promotion Impact Report - Mid-Year Sale 2023', 10, '2023-08-10 15:15:00', 'Campaign', '{SalesBoost: "25%", TopSelling: "Ceylon Tea"}'),

(15, 'Festival Campaign Summary - Sinhala New Year 2023', 10, '2023-04-20 16:00:00', 'Campaign', '{TotalSales: 9500000, TopCategory: "Food & Beverages"}'),

-- Supplier and Delivery Performance Reports

(16, 'Supplier Performance - Q3 2023', 8, '2023-10-15 14:30:00', 'Quarterly', '{TopSupplier: "Lanka Spices Ltd.", DeliverySuccessRate: "98%"}'),

(17, 'Delivery Timeliness Report - September 2023', 11, '2023-09-30 16:00:00', 'Monthly', '{OnTimeDeliveries: 350, LateDeliveries: 15}'),

-- System Logs and Security

(18, 'System Audit Log Summary - Q3 2023', 35, '2023-10-01 12:00:00', 'Quarterly', '{CriticalActions: 18, SystemErrors: 5}'),

(19, 'System Backup Report - October 2023', 35, '2023-10-31 21:30:00', 'Monthly', '{BackupStatus: "Successful", BackupSize: "120GB"}'),

-- Business Development Reports

(20, 'Store Expansion Plan - 2024', 1, '2023-11-01 10:30:00', 'Planning', '{NewStores: ["Galle", "Kandy", "Jaffna"]}'),

(21, 'Market Trend Analysis - Q3 2023', 10, '2023-10-25 14:15:00', 'Analysis', '{Trend: "Organic Food Demand Increase"}'),

-- Custom Reports

(22, 'Top Selling Products Report - Q3 2023', 9, '2023-10-10 11:00:00', 'Custom', '{Products: ["Rice 10kg", "Milk Powder", "Ceylon Tea 500g"]}'),

(23, 'Year-End Sales Forecast - 2023', 1, '2023-11-30 10:00:00', 'Forecast', '{ExpectedSales: 62000000, KeySeasons: ["Christmas", "New Year"]}'),

(24, 'Supplier Cost Summary - October 2023', 11, '2023-11-01 13:15:00', 'Monthly', '{TotalCost: 8400000, KeySuppliers: ["Lanka Spices Ltd."]}'),

(25, 'Customer Loyalty Program Impact - 2023', 34, '2023-12-01 15:45:00', 'Analysis', '{TotalMembers: 1250, ActiveMembers: 850}');

SET IDENTITY\_INSERT Report OFF;

/\*==========================================================

|| PRODUCT AND STOCK MANAGEMENT MODULE ||

==========================================================\*/

/\*-----------------------------------------------

-- Adding data to Category table

-----------------------------------------------\*/

SET IDENTITY\_INSERT Category ON;

INSERT INTO Category (CategoryID, CategoryName)

VALUES

(1, 'Dairy'),

(2, 'Tea'),

(3, 'Staples'),

(4, 'Snacks'),

(5, 'Frozen Food'),

(6, 'Condiments'),

(7, 'Household'),

(8, 'Beverages'),

(9, 'Personal Care'),

(10, 'Baby Care'),

(11, 'Pet Supplies'),

(12, 'Health & Wellness'),

(13, 'Bakery'),

(14, 'Breakfast Items'),

(15, 'Canned Goods'),

(16, 'Fresh Produce'),

(17, 'Meat & Seafood'),

(18, 'Spices'),

(19, 'Cleaning Supplies'),

(20, 'Electronics');

SET IDENTITY\_INSERT Category OFF;

/\*-----------------------------------------------

-- Adding data to Brand table

-----------------------------------------------\*/

SET IDENTITY\_INSERT Brand ON;

INSERT INTO Brand (BrandID, BrandName)

VALUES

(1, 'Nestlé'),

(2, 'Dilmah'),

(3, 'Prima'),

(4, 'Mlesna'),

(5, 'Maliban'),

(6, 'Keells'),

(7, 'Anchor'),

(8, 'Kist'),

(9, 'Sunlight'),

(10, 'LUX'),

(11, 'CIC'),

(12, 'Harpic'),

(13, 'Lanka Soy'),

(14, 'Elephant House'),

(15, 'Coca-Cola'),

(16, 'Munchee'),

(17, 'Ritzbury'),

(18, 'Dettol'),

(19, 'Kotmale'),

(20, 'Fonterra'),

(21, 'Elephant House Ice Cream'),

(22, 'Hela Bōjun'),

(23, 'Perera & Sons'),

(24, 'Hemas'),

(25, 'Pears'),

(26, 'Gold Leaf'),

(27, 'MD'),

(28, 'Maggi'),

(29, 'Samaposha'),

(30, 'Araliya'),

(31, 'Renuka'),

(32, 'Malwatta'),

(33, 'Ruhunu Foods'),

(34, 'Singer'),

(35, 'Milo'),

(36, 'Highland'),

(37, 'Harischandra'),

(38, 'Seven Seas'),

(39, 'Unilever');

SET IDENTITY\_INSERT Brand OFF;

/\*-----------------------------------------------

-- Adding data to TaxRate table

-----------------------------------------------\*/

SET IDENTITY\_INSERT TaxRate ON;

INSERT INTO TaxRate (TaxRateID, TaxRate)

VALUES

(1, 8.0),

(2, 12.0),

(3, 15.0),

(4, 5.0),

(5, 18.0),

(6, 22.0),

(7, 10.0),

(8, 20.0),

(9, 25.0),

(10, 30.0),

(11, 0.0),

(12, 7.5),

(13, 14.5),

(14, 17.0),

(15, 6.0),

(16, 9.0),

(17, 11.0),

(18, 19.0),

(19, 16.0),

(20, 13.0);

SET IDENTITY\_INSERT TaxRate OFF;

/\*-----------------------------------------------

-- Adding data to Product table

-----------------------------------------------\*/

SET IDENTITY\_INSERT Product ON;

INSERT INTO Product (ProductID, ProductName, CategoryID, BrandID, UnitPrice, TaxRateID, ReorderLevel, IsActive)

VALUES

-- Dairy Products

(1, 'Anchor Full Cream Milk Powder', 1, 7, 1000.00, 3, 50, 1),

(2, 'Kotmale Fresh Milk 1L', 1, 19, 240.00, 1, 30, 1),

(3, 'Highland Butter 200g', 1, 37, 950.00, 3, 20, 1),

-- Tea

(4, 'Dilmah Premium Tea 200g', 2, 2, 750.00, 2, 40, 1),

(5, 'Mlesna Black Tea 100g', 2, 4, 600.00, 2, 25, 1),

(6, 'Gold Leaf Tea 400g', 2, 26, 850.00, 3, 50, 1),

-- Staples

(7, 'Prima Wheat Flour 1kg', 3, 3, 180.00, 1, 100, 1),

(8, 'Ruhunu Red Rice 5kg', 3, 34, 1500.00, 2, 50, 1),

(9, 'CIC Basmati Rice 1kg', 3, 11, 1100.00, 2, 30, 1),

-- Snacks

(10, 'Munchee Marie Biscuits 400g', 4, 16, 240.00, 2, 70, 1),

(11, 'Ritzbury Chocolate Fingers 200g', 4, 17, 380.00, 3, 40, 1),

(12, 'Maliban Lemon Puff 300g', 4, 5, 300.00, 2, 60, 1),

-- Frozen Food

(13, 'Elephant House Chicken Sausages 1kg', 5, 14, 1500.00, 3, 20, 1),

(14, 'Keells Chicken Drumsticks 1kg', 5, 6, 1200.00, 3, 25, 1),

(15, 'Kotmale Ice Cream Vanilla 1L', 5, 19, 800.00, 3, 15, 1),

-- Condiments

(16, 'MD Mango Chutney 250g', 6, 27, 500.00, 2, 40, 1),

(17, 'Kist Tomato Sauce 750ml', 6, 8, 450.00, 2, 50, 1),

(18, 'Maggi Coconut Milk Powder 300g', 6, 28, 600.00, 2, 35, 1),

-- Household

(19, 'Sunlight Detergent Powder 2kg', 7, 9, 1500.00, 3, 20, 1),

(20, 'Harpic Toilet Cleaner 500ml', 7, 12, 550.00, 2, 25, 1),

(21, 'Dettol Antibacterial Soap 100g', 7, 18, 150.00, 2, 60, 1),

-- Beverages

(22, 'Coca-Cola Bottle 1.5L', 8, 15, 300.00, 2, 100, 1),

(23, 'Elephant House Cream Soda Can 330ml', 8, 14, 120.00, 2, 150, 1),

(24, 'Nestlé Milo 400g', 8, 35, 800.00, 3, 30, 1),

-- Personal Care

(25, 'LUX Body Wash 250ml', 9, 10, 450.00, 2, 20, 1),

(26, 'Pears Baby Lotion 200ml', 9, 25, 550.00, 2, 15, 1),

(27, 'Hemas Velvet Soap 100g', 9, 24, 100.00, 1, 50, 1);

SET IDENTITY\_INSERT Product OFF;

/\*-----------------------------------------------

-- Adding data to Warehouse table

-----------------------------------------------\*/

SET IDENTITY\_INSERT Warehouse ON;

INSERT INTO Warehouse (WarehouseID, WarehouseName, Location, ManagerID)

VALUES

(1, 'Central Warehouse', 'Colombo', 13),

(2, 'Kandy Distribution Center', 'Kandy', 14),

(3, 'Southern Storage', 'Galle', 15),

(4, 'Northern Depot', 'Jaffna', 16),

(5, 'Eastern Hub', 'Trincomalee', 17),

(6, 'Western Storage', 'Negombo', 18),

(7, 'Hill Country Depot', 'Nuwara Eliya', 19),

(8, 'North Western Facility', 'Kurunegala', 20),

(9, 'Uva Warehouse', 'Badulla', 21),

(10, 'Sabaragamuwa Depot', 'Ratnapura', 22),

(11, 'Eastern Highlands Center', 'Ampara', 23),

(12, 'Coastal Hub', 'Matara', 24),

(13, 'Dry Zone Depot', 'Anuradhapura', 25),

(14, 'Urban Storage', 'Batticaloa', 26),

(15, 'Central Highlands Depot', 'Hatton', 27),

(16, 'Industrial Hub', 'Kalutara', 28),

(17, 'Administrative Warehouse', 'Chilaw', 29),

(18, 'Port Storage', 'Hambantota', 30),

(19, 'Heritage Hub', 'Polonnaruwa', 31),

(20, 'Frontier Depot', 'Monaragala', 32);

SET IDENTITY\_INSERT Warehouse OFF;

/\*-----------------------------------------------

-- Adding data to Stock table

-----------------------------------------------\*/

SET IDENTITY\_INSERT Stock ON;

INSERT INTO Stock (StockID, ProductID, WarehouseID, QuantityAvailable, LastRestockedDate)

VALUES

(1, 1, 1, 500, '2024-12-01'),

(2, 2, 2, 300, '2024-12-02'),

(3, 3, 3, 200, '2024-12-03'),

(4, 4, 4, 600, '2024-12-04'),

(5, 5, 5, 450, '2024-12-05'),

(6, 6, 6, 700, '2024-12-06'),

(7, 7, 7, 800, '2024-12-01'),

(8, 8, 8, 400, '2024-12-02'),

(9, 9, 9, 350, '2024-12-03'),

(10, 10, 10, 500, '2024-12-04'),

(11, 11, 11, 550, '2024-12-05'),

(12, 12, 12, 250, '2024-12-06'),

(13, 13, 13, 600, '2024-12-01'),

(14, 14, 14, 300, '2024-12-02'),

(15, 15, 15, 450, '2024-12-03'),

(16, 16, 16, 700, '2024-12-04'),

(17, 17, 17, 500, '2024-12-05'),

(18, 18, 18, 350, '2024-12-06'),

(19, 19, 19, 400, '2024-12-01'),

(20, 20, 20, 600, '2024-12-02');

SET IDENTITY\_INSERT Stock OFF;

/\*==========================================================

|| DISCOUNT MANAGEMENT MODULE ||

==========================================================\*/

/\*-----------------------------------------------

-- Adding data to Discount table

-----------------------------------------------\*/

SET IDENTITY\_INSERT Discount ON;

INSERT INTO Discount (DiscountID, DiscountName, StartDate, EndDate, DiscountPercentage, DiscountType)

VALUES

(1, 'Avurudu Sale 2024', '2024-04-01', '2024-04-14', 15.00, 'Seasonal'),

(2, 'Vesak Poya Special', '2024-05-21', '2024-05-25', 10.00, 'Religious'),

(3, 'Sinhala Tamil New Year Bonanza', '2024-04-10', '2024-04-20', 20.00, 'Seasonal'),

(4, 'Back to School Offer', '2024-01-05', '2024-01-15', 12.50, 'Event'),

(5, 'Weekend Super Saver', '2024-06-01', '2024-06-02', 5.00, 'Weekend'),

(6, 'Independence Day Promo', '2024-02-01', '2024-02-07', 10.00, 'National Holiday'),

(7, 'Ramadan Festival Discount', '2024-03-11', '2024-03-30', 8.00, 'Religious'),

(8, 'Christmas Mega Sale', '2024-12-15', '2024-12-25', 25.00, 'Seasonal'),

(9, 'Black Friday Deals', '2024-11-29', '2024-11-30', 30.00, 'Event'),

(10, 'Poya Day Essentials Discount', '2024-06-21', '2024-06-21', 5.00, 'Religious'),

(11, 'Maha Shivaratri Offer', '2024-02-20', '2024-02-21', 12.00, 'Religious'),

(12, 'Mother’s Day Special', '2024-05-11', '2024-05-12', 15.00, 'Event'),

(13, 'Father’s Day Promotion', '2024-06-15', '2024-06-16', 15.00, 'Event'),

(14, 'Govi Sathiya Discounts', '2024-09-01', '2024-09-07', 10.00, 'Agricultural'),

(15, 'Back to Office Sale', '2024-07-01', '2024-07-10', 7.50, 'Event'),

(16, 'Deepavali Celebrations', '2024-10-15', '2024-10-25', 12.00, 'Religious'),

(17, 'End of Season Clearance', '2024-08-25', '2024-08-31', 20.00, 'Clearance'),

(18, 'School Holiday Snacks Promo', '2024-08-01', '2024-08-10', 5.00, 'Event'),

(19, 'Kandy Esala Perahera Special', '2024-07-20', '2024-07-30', 8.50, 'Cultural'),

(20, 'Poson Poya Discount', '2024-06-20', '2024-06-22', 10.00, 'Religious'),

(21, 'Pre-Avurudu Stock Clearance', '2024-03-25', '2024-03-31', 18.00, 'Clearance'),

(22, 'Valentine’s Day Promo', '2024-02-12', '2024-02-14', 20.00, 'Event'),

(23, 'New Year Kickoff Sale', '2024-01-01', '2024-01-05', 10.00, 'Event'),

(24, 'Thrift Thursday Offer', '2024-06-13', '2024-06-13', 5.00, 'Weekly'),

(25, 'Budget Sunday Promo', '2024-06-16', '2024-06-16', 6.00, 'Weekly');

SET IDENTITY\_INSERT Discount OFF;

/\*-----------------------------------------------

-- Adding data to ProductDiscount table

-----------------------------------------------\*/

SET IDENTITY\_INSERT ProductDiscount ON;

INSERT INTO ProductDiscount (ProductDiscountID, ProductID, DiscountID)

VALUES

-- Avurudu Sale 2024

(1, 1, 1), -- Anchor Full Cream Milk Powder

(2, 8, 1), -- Ruhunu Red Rice 5kg

(3, 12, 1), -- Maliban Lemon Puff 300g

(4, 7, 1), -- Prima Wheat Flour 1kg

-- Vesak Poya Special

(5, 2, 2), -- Kotmale Fresh Milk 1L

(6, 21, 2), -- Dettol Antibacterial Soap 100g

-- Back to School Offer

(7, 10, 4), -- Munchee Marie Biscuits 400g

(8, 11, 4), -- Ritzbury Chocolate Fingers 200g

(9, 26, 4), -- Pears Baby Lotion 200ml

-- Independence Day Promo

(10, 4, 6), -- Dilmah Premium Tea 200g

(11, 24, 6), -- Nestlé Milo 400g

-- Ramadan Festival Discount

(12, 9, 7), -- CIC Basmati Rice 1kg

(13, 18, 7), -- Maggi Coconut Milk Powder 300g

-- Christmas Mega Sale

(14, 13, 8), -- Elephant House Chicken Sausages 1kg

(15, 14, 8), -- Keells Chicken Drumsticks 1kg

(16, 3, 8), -- Highland Butter 200g

(17, 27, 8), -- Hemas Velvet Soap 100g

-- Black Friday Deals

(18, 5, 9), -- Mlesna Black Tea 100g

(19, 6, 9), -- Gold Leaf Tea 400g

(20, 20, 9), -- Harpic Toilet Cleaner 500ml

-- Poya Day Essentials Discount

(21, 15, 10), -- Kotmale Ice Cream Vanilla 1L

(22, 19, 10), -- Sunlight Detergent Powder 2kg

-- End of Season Clearance

(23, 22, 17), -- Coca-Cola Bottle 1.5L

(24, 23, 17), -- Elephant House Cream Soda Can 330ml

(25, 17, 17); -- Kist Tomato Sauce 750ml

SET IDENTITY\_INSERT ProductDiscount OFF;

/\*==========================================================

|| SALES MANAGEMENT MODULE ||

==========================================================\*/

/\*-----------------------------------------------

-- Adding data to PaymentMethod table

-----------------------------------------------\*/

SET IDENTITY\_INSERT PaymentMethod ON;

INSERT INTO PaymentMethod (PaymentMethodID, PaymentType)

VALUES

(1, 'Cash'),

(2, 'Debit Card - Commercial Bank'),

(3, 'Debit Card - HNB'),

(4, 'Credit Card - BOC'),

(5, 'Credit Card - Sampath Bank'),

(6, 'Mobile Payment - FriMi'),

(7, 'Mobile Payment - eZ Cash'),

(8, 'Mobile Payment - mCash'),

(9, 'Bank Transfer - NDB'),

(10, 'Bank Transfer - NSB Bank'),

(11, 'QR Payment - LankaQR');

SET IDENTITY\_INSERT PaymentMethod OFF;

/\*-----------------------------------------------

-- Adding data to SalesTransaction table

-----------------------------------------------\*/

SET IDENTITY\_INSERT SalesTransaction ON;

INSERT INTO SalesTransaction (TransactionID, CustomerID, TransactionDate, PaymentMethodID, TotalAmount, DiscountApplied, FinalAmount)

VALUES

(1, 1, '2024-01-01 10:15:00', 1, 1500.00, 100.00, 1400.00),

(2, 2, '2024-01-02 15:45:00', 2, 2500.00, 150.00, 2350.00),

(3, 3, '2024-01-03 12:30:00', 3, 4500.00, 200.00, 4300.00),

(4, 4, '2024-01-04 17:50:00', 6, 3200.00, 100.00, 3100.00),

(5, 5, '2024-01-05 09:40:00', 7, 1800.00, 50.00, 1750.00),

(6, 6, '2024-01-06 14:20:00', 4, 3700.00, 200.00, 3500.00),

(7, 7, '2024-01-07 11:00:00', 5, 5000.00, 300.00, 4700.00),

(8, 8, '2024-01-08 16:10:00', 8, 2200.00, 0.00, 2200.00),

(9, 9, '2024-01-09 13:25:00', 9, 6000.00, 500.00, 5500.00),

(10, 10, '2024-01-10 10:00:00', 10, 1200.00, 0.00, 1200.00),

(11, 11, '2024-01-11 18:15:00', 11, 3400.00, 150.00, 3250.00),

(12, 12, '2024-01-12 09:20:00', 1, 2800.00, 100.00, 2700.00),

(13, 13, '2024-01-13 14:35:00', 2, 1900.00, 50.00, 1850.00),

(14, 14, '2024-01-14 11:45:00', 3, 3100.00, 100.00, 3000.00),

(15, 15, '2024-01-15 16:00:00', 6, 4700.00, 200.00, 4500.00),

(16, 16, '2024-01-16 10:50:00', 7, 2100.00, 50.00, 2050.00),

(17, 17, '2024-01-17 13:10:00', 5, 4300.00, 300.00, 4000.00),

(18, 18, '2024-01-18 15:25:00', 4, 3800.00, 150.00, 3650.00),

(19, 19, '2024-01-19 17:00:00', 9, 5200.00, 300.00, 4900.00),

(20, 20, '2024-01-20 12:00:00', 10, 2500.00, 100.00, 2400.00);

SET IDENTITY\_INSERT SalesTransaction OFF;

/\*-----------------------------------------------

-- Adding data to SalesTransactionDetail table

-----------------------------------------------\*/

SET IDENTITY\_INSERT SalesTransactionDetail ON;

INSERT INTO SalesTransactionDetail (TransactionDetailID, TransactionID, ProductID, Quantity, UnitPrice, LineTotal)

VALUES

-- Transaction 1

(1, 1, 7, 3, 180.00, 540.00),

(2, 1, 10, 4, 240.00, 960.00),

-- Transaction 2

(3, 2, 2, 5, 240.00, 1200.00),

(4, 2, 21, 2, 150.00, 300.00),

(5, 2, 6, 1, 850.00, 850.00),

-- Transaction 3

(6, 3, 19, 2, 1500.00, 3000.00),

(7, 3, 12, 2, 300.00, 600.00),

(8, 3, 3, 1, 950.00, 950.00),

-- Transaction 4

(9, 4, 5, 2, 600.00, 1200.00),

(10, 4, 8, 1, 1500.00, 1500.00),

(11, 4, 25, 1, 450.00, 450.00),

-- Transaction 5

(12, 5, 1, 1, 1000.00, 1000.00),

(13, 5, 24, 1, 800.00, 800.00),

-- Transaction 6

(14, 6, 15, 2, 800.00, 1600.00),

(15, 6, 17, 2, 450.00, 900.00),

(16, 6, 13, 1, 1500.00, 1500.00),

-- Transaction 7

(17, 7, 11, 3, 380.00, 1140.00),

(18, 7, 9, 2, 1100.00, 2200.00),

(19, 7, 20, 1, 550.00, 550.00),

-- Transaction 8

(20, 8, 14, 2, 1200.00, 2400.00),

-- Transaction 9

(21, 9, 4, 2, 750.00, 1500.00),

(22, 9, 18, 3, 600.00, 1800.00),

(23, 9, 22, 2, 300.00, 600.00),

-- Transaction 10

(24, 10, 16, 1, 500.00, 500.00),

(25, 10, 27, 7, 100.00, 700.00);

SET IDENTITY\_INSERT SalesTransactionDetail OFF;