ASSIGNMENT – 3

NAME:A.RAKCHA

REG NO:192311421

1. Create a database for manage product stock, suppliers and orders.

-Model tables for products stock, suppliers and orders.

-Write stored procedures for list low stock products.

-Implement triggers to update products and transaction status.

-Write SQL queries to analyze top selling products based on order quantity.

ANSWER:

Queries:

CREATE DATABASE StockManagement;

USE StockManagement;

CREATE TABLE Suppliers (

SupplierID INT PRIMARY KEY,

SupplierName VARCHAR(255) NOT NULL,

Address VARCHAR(255) NOT NULL,

Phone VARCHAR(20) NOT NULL,

Email VARCHAR(100) NOT NULL

);

CREATE TABLE Products (

ProductID INT PRIMARY KEY,

ProductName VARCHAR(255) NOT NULL,

Description VARCHAR(500),

UnitPrice DECIMAL(10, 2) NOT NULL,

StockQuantity INT NOT NULL,

ReorderLevel INT NOT NULL,

SupplierID INT,

FOREIGN KEY (SupplierID) REFERENCES Suppliers(SupplierID)

);

CREATE TABLE Orders (

OrderID INT PRIMARY KEY,

OrderDate DATE NOT NULL,

TotalCost DECIMAL(10, 2) NOT NULL,

Status VARCHAR(20) NOT NULL CHECK (Status IN ('Pending', 'Shipped', 'Delivered', 'Cancelled'))

);

CREATE TABLE OrderItems (

OrderItemID INT PRIMARY KEY,

OrderID INT,

ProductID INT,

Quantity INT NOT NULL,

UnitPrice DECIMAL(10, 2) NOT NULL,

FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),

FOREIGN KEY (ProductID) REFERENCES Products(ProductID)

);

Stored Procedure: List Low Stock Products

CREATE PROCEDURE sp\_ListLowStockProducts

AS

BEGIN

SELECT p.ProductName, p.StockQuantity, p.ReorderLevel

FROM Products p

WHERE p.StockQuantity <= p.ReorderLevel

ORDER BY p.StockQuantity ASC;

END;

Triggers

CREATE TRIGGER tr\_UpdateProductStock

ON OrderItems

AFTER INSERT, UPDATE

AS

BEGIN

UPDATE p

SET p.StockQuantity = p.StockQuantity - i.Quantity

FROM Products p

INNER JOIN inserted i ON p.ProductID = i.ProductID;

END;

CREATE TRIGGER tr\_UpdateOrderStatus

ON Orders

AFTER UPDATE

AS

BEGIN

IF UPDATE(Status)

BEGIN

IF EXISTS (SELECT 1 FROM inserted WHERE Status = 'Shipped')

BEGIN

UPDATE o

SET o.Status = 'Shipped'

FROM Orders o

INNER JOIN inserted i ON o.OrderID = i.OrderID;

END;

ELSE IF EXISTS (SELECT 1 FROM inserted WHERE Status = 'Delivered')

BEGIN

UPDATE o

SET o.Status = 'Delivered'

EGIN

UPDATE o

SET o.Status = 'Delivered'

FROM Orders o

INNER JOIN inserted i ON o.OrderID = i.OrderID;

END;

END;

END;

SQL Query: Top Selling Products

SELECT p.ProductName, SUM(oi.Quantity) AS TotalQuantity

FROM OrderItems oi

INNER JOIN Products p ON oi.ProductID = p.ProductID

GROUP BY p.ProductName

ORDER BY TotalQuantity DESC

LIMIT 10;

CONCLUSION:

The database structure provided offers a comprehensive solution for managing product stock, suppliers, and customer orders. Here's a summary of the key components and their functionalities:

|  |  |
| --- | --- |