VILNIUS TECH
Faculty of Mechanics

DEPARTMENT OF MECHATRONICS, ROBOTICS AND DIGITAL MANUFACTURING

DATABASES

Warehouse Management System

COURSE PROJECT

Group: TDIFUC22

Lecturer: Ernestas Miknevičius
Made by: Moustafa Elsaid

Overview

This project involves designing, implementing, and manipulating a relational database for a warehouse management system using Microsoft SQL Server. It covers the core requirements outlined in the course, including database design, data manipulation, queries, and other advanced functionalities.

Database Schema

The database schema includes the following tables and relationships:

1. **Suppliers**
    - Columns:
        - `SupplierID` (Primary Key, INT, Auto-increment)
        - `SupplierName` (VARCHAR)
        - `ContactName` (VARCHAR)
        - `ContactEmail` (VARCHAR)
2. **Products**
    - Columns:
        - `ProductID` (Primary Key, INT, Auto-increment)
        - `ProductName` (VARCHAR)
        - `QuantityInStock` (INT)
        - `Price` (DECIMAL)
        - `SupplierID` (Foreign Key to `Suppliers.SupplierID`)
3. **Employees**
    - Columns:
        - `EmployeeID` (Primary Key, INT, Auto-increment)
        - `FirstName` (VARCHAR)
        - `LastName` (VARCHAR)
        - `Position` (VARCHAR)
        - `DateHired` (DATE)
4. **Orders**
    - Columns:
        - `OrderID` (Primary Key, INT, Auto-increment)
        - `OrderDate` (DATE)
        - `EmployeeID` (Foreign Key to `Employees.EmployeeID`)

5. **Order_Products**
    - Columns:
        - `OrderID` (Foreign Key to `Orders.OrderID`)
        - `ProductID` (Foreign Key to `Products.ProductID`)
        - `Quantity` (INT)
6. **Shipments**
    - Columns:
        - `ShipmentID` (Primary Key, INT, Auto-increment)
        - `OrderID` (Foreign Key to `Orders.OrderID`)
        - `ShipmentDate` (DATE)
        - `Status` (VARCHAR)

Scripts

Create Tables

```
CREATE TABLE Suppliers (
    SupplierID INT PRIMARY KEY IDENTITY(1,1),
    SupplierName VARCHAR(255),
    ContactName VARCHAR(255),
    ContactEmail VARCHAR(255)
);

CREATE TABLE Products (
    ProductID INT PRIMARY KEY IDENTITY(1,1),
    ProductName VARCHAR(255),
    QuantityInStock INT,
    Price DECIMAL(10, 2),
    SupplierID INT FOREIGN KEY REFERENCES Suppliers(SupplierID)
);

CREATE TABLE Employees (
    EmployeeID INT PRIMARY KEY IDENTITY(1,1),
    FirstName VARCHAR(255),
    LastName VARCHAR(255),
    Position VARCHAR(255),
    DateHired DATE
);

CREATE TABLE Orders (
    OrderID INT PRIMARY KEY IDENTITY(1,1),
    OrderDate DATE,
    EmployeeID INT FOREIGN KEY REFERENCES Employees(EmployeeID)
);

CREATE TABLE Order_Products (
    OrderID INT FOREIGN KEY REFERENCES Orders(OrderID),
    ProductID INT FOREIGN KEY REFERENCES Products(ProductID),
    Quantity INT,
    PRIMARY KEY (OrderID, ProductID)
```

```
);

CREATE TABLE Shipments (
    ShipmentID INT PRIMARY KEY IDENTITY(1,1),
    OrderID INT FOREIGN KEY REFERENCES Orders(OrderID),
    ShipmentDate DATE,
    Status VARCHAR(255)
);
```

Insert Data

```
INSERT INTO Suppliers (SupplierName, ContactName, ContactEmail) VALUES
('Supplier A', 'John Doe', 'john.doe@example.com'),
('Supplier B', 'Jane Smith', 'jane.smith@example.com');

INSERT INTO Products (ProductName, QuantityInStock, Price, SupplierID) VALUES
('Product X', 100, 10.99, 1),
('Product Y', 200, 15.99, 2);

INSERT INTO Employees (FirstName, LastName, Position, DateHired) VALUES
('Alice', 'Brown', 'Manager', '2020-01-15'),
('Bob', 'White', 'Clerk', '2021-06-20');

INSERT INTO Orders (OrderDate, EmployeeID) VALUES
('2024-01-01', 1),
('2024-02-01', 2);

INSERT INTO Order_Products (OrderID, ProductID, Quantity) VALUES
(1, 1, 10),
(2, 2, 20);

INSERT INTO Shipments (OrderID, ShipmentDate, Status) VALUES
(1, '2024-01-05', 'Delivered'),
(2, '2024-02-05', 'Pending');
```

Select Queries

```
-- Query 1: Aggregate Function
SELECT SupplierID, COUNT(*) AS ProductCount
FROM Products
GROUP BY SupplierID;

-- Query 2: Pagination
SELECT *
FROM Products
ORDER BY ProductID
OFFSET 0 ROWS FETCH NEXT 10 ROWS ONLY;

-- Query 3: Join Query
SELECT o.OrderID, p.ProductName, op.Quantity
FROM Orders o
JOIN Order_Products op ON o.OrderID = op.OrderID
JOIN Products p ON op.ProductID = p.ProductID;
```

Stored Procedures and Triggers

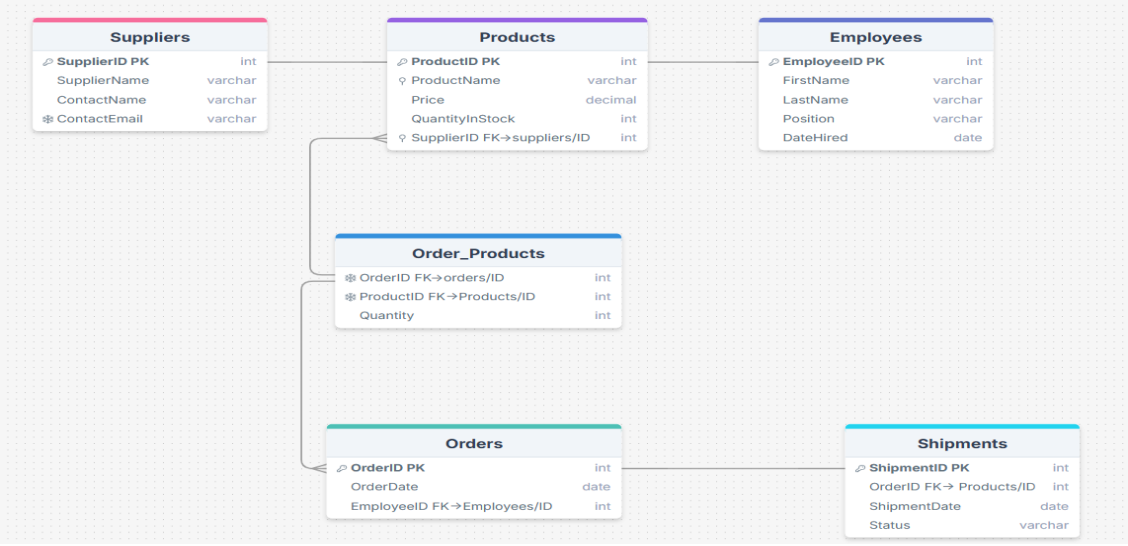
```
-- Stored Procedure Example
CREATE PROCEDURE GetProductsBySupplier (@SupplierID INT)
AS
BEGIN
    SELECT * FROM Products WHERE SupplierID = @SupplierID;
END;

-- Trigger Example
CREATE TRIGGER UpdateStock
ON Order_Products
AFTER INSERT
AS
BEGIN
    UPDATE Products
    SET QuantityInStock = QuantityInStock - i.Quantity
    FROM Products p
    INNER JOIN inserted i ON p.ProductID = i.ProductID;
END;
```

PowerPoint Presentation

Include a PowerPoint presentation summarizing:

1. Database design.
2. Key relationships and schema.
3. Examples of queries and their outputs.

ER Diagram



Conclusion

This project demonstrates the complete lifecycle of designing and implementing a relational database system for warehouse management, showcasing key functionalities and SQL techniques.