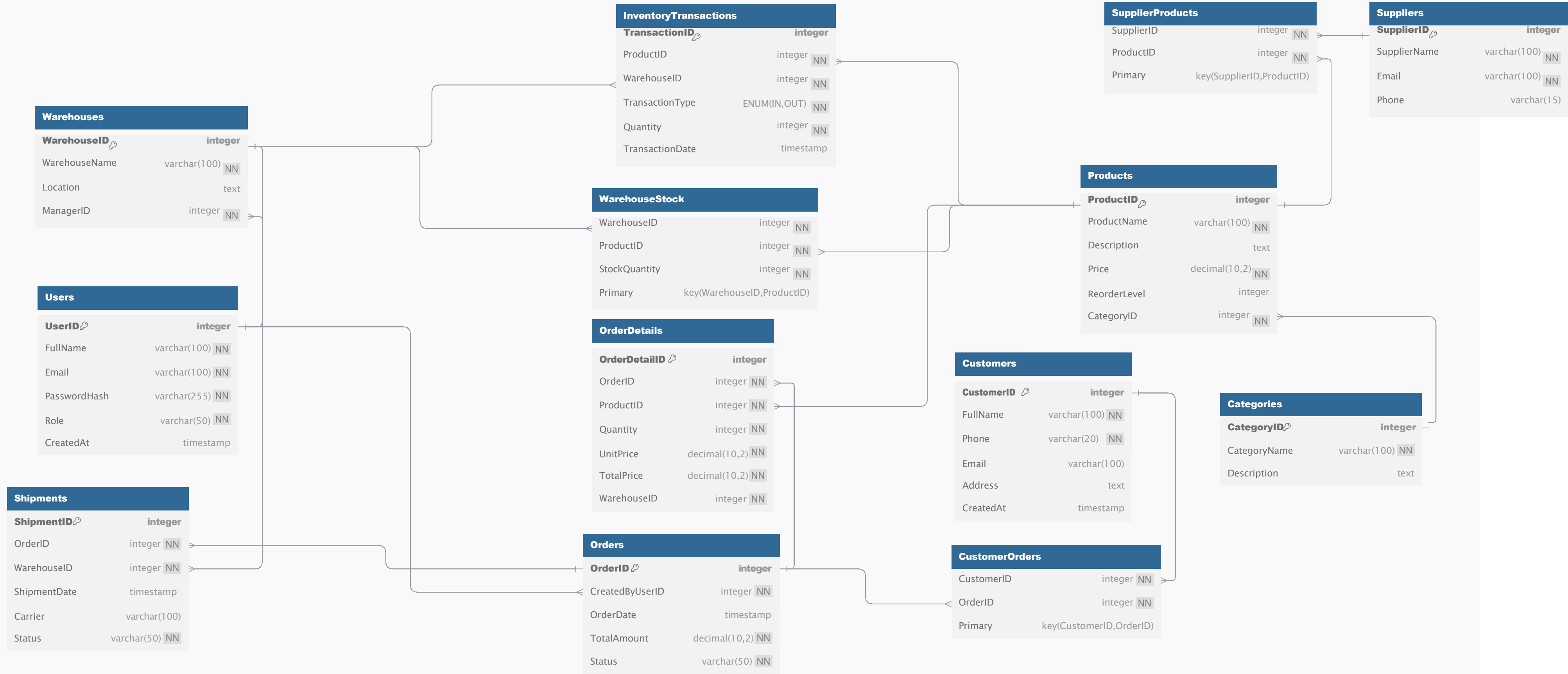
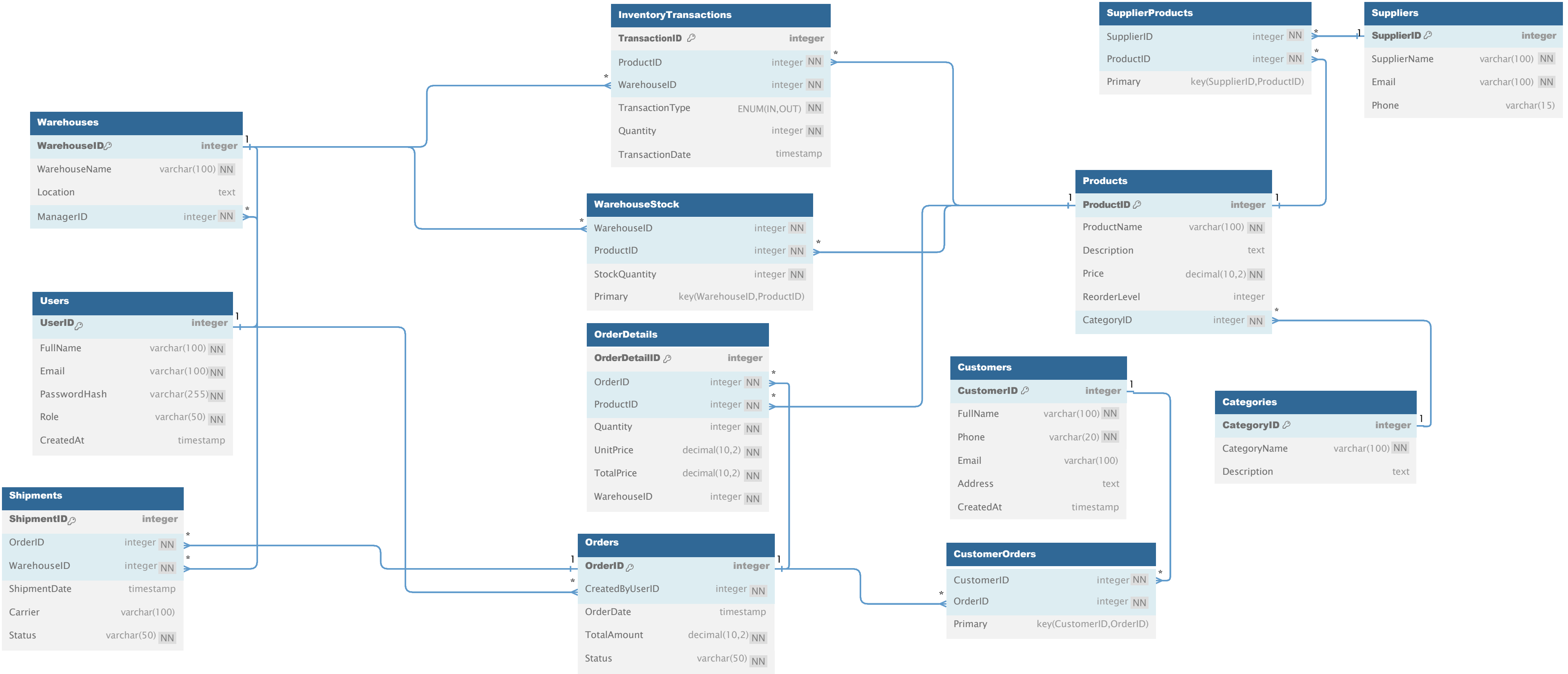


Logical & Physical Schema





-- SQL Server Schema for Inventory Management System

```
CREATE TABLE Users (  
    UserID INT PRIMARY KEY IDENTITY(1,1),  
    FullName VARCHAR(100) NOT NULL,  
    Email VARCHAR(100) NOT NULL UNIQUE,  
    PasswordHash VARCHAR(255) NOT NULL,  
    Role VARCHAR(50) NOT NULL, -- Admin, Manager, Staff  
    CreatedAt DATETIME DEFAULT GETDATE()  
);
```

```
CREATE TABLE Customers (  
    CustomerID INT PRIMARY KEY IDENTITY(1,1),  
    FullName VARCHAR(100) NOT NULL,  
    Phone VARCHAR(20) NOT NULL,  
    Email VARCHAR(100) UNIQUE,  
    Address TEXT,  
    CreatedAt DATETIME DEFAULT GETDATE()  
);
```

```
CREATE TABLE Categories (  
    CategoryID INT PRIMARY KEY IDENTITY(1,1),  
    CategoryName VARCHAR(100) NOT NULL UNIQUE,  
    Description TEXT  
);
```

```
CREATE TABLE Suppliers (  
    SupplierID INT PRIMARY KEY IDENTITY(1,1),  
    SupplierName VARCHAR(100) NOT NULL,  
    Email VARCHAR(100) NOT NULL UNIQUE,  
    Phone VARCHAR(15)  
);
```

```
CREATE TABLE Warehouses (  
    WarehouseID INT PRIMARY KEY IDENTITY(1,1),  
    WarehouseName VARCHAR(100) NOT NULL,  
    Location TEXT,  
    ManagerID INT NOT NULL,  
    FOREIGN KEY (ManagerID) REFERENCES Users(UserID)  
);
```

```
CREATE TABLE Products (  
    ProductID INT PRIMARY KEY IDENTITY(1,1),  
    ProductName VARCHAR(100) NOT NULL,  
    Description TEXT,  
    Price DECIMAL(10,2) NOT NULL,  
    ReorderLevel INT,  
    CategoryID INT NOT NULL,  
    FOREIGN KEY (CategoryID) REFERENCES Categories(CategoryID)  
);
```

```
CREATE TABLE SupplierProducts (  
    SupplierID INT NOT NULL,  
    ProductID INT NOT NULL,  
    PRIMARY KEY (SupplierID, ProductID),  
    FOREIGN KEY (SupplierID) REFERENCES Suppliers(SupplierID),  
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)  
);
```

```
CREATE TABLE WarehouseStock (  
    WarehouseID INT NOT NULL,  
    ProductID INT NOT NULL,  
    StockQuantity INT NOT NULL DEFAULT 0,  
    PRIMARY KEY (WarehouseID, ProductID),  
    FOREIGN KEY (WarehouseID) REFERENCES Warehouses(WarehouseID),  
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)  
);
```

```
CREATE TABLE Orders (  
    OrderID INT PRIMARY KEY IDENTITY(1,1),  
    CreatedByUserID INT NOT NULL,  
    OrderDate DATETIME DEFAULT GETDATE(),  
    TotalAmount DECIMAL(10,2) NOT NULL,  
    Status VARCHAR(50) NOT NULL, -- Pending, Confirmed, Shipped, Delivered, Canceled  
    FOREIGN KEY (CreatedByUserID) REFERENCES Users(UserID)  
);
```

```
CREATE TABLE CustomerOrders (  
    CustomerID INT NOT NULL,  
    OrderID INT NOT NULL,  
    PRIMARY KEY (CustomerID, OrderID),  
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),  
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID)  
);
```

```
CREATE TABLE OrderDetails (  
    OrderDetailID INT PRIMARY KEY IDENTITY(1,1),  
    OrderID INT NOT NULL,  
    ProductID INT NOT NULL,  
    Quantity INT NOT NULL,  
    UnitPrice DECIMAL(10,2) NOT NULL,  
    TotalPrice DECIMAL(10,2) NOT NULL,  
    WarehouseID INT NOT NULL,  
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),  
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID),  
    FOREIGN KEY (WarehouseID) REFERENCES Warehouses(WarehouseID)  
);
```

```
CREATE TABLE InventoryTransactions (  
    TransactionID INT PRIMARY KEY IDENTITY(1,1),  
    ProductID INT NOT NULL,  
    WarehouseID INT NOT NULL,  
    TransactionType VARCHAR(10) NOT NULL CHECK (TransactionType IN ('IN', 'OUT')),  
    Quantity INT NOT NULL,  
    TransactionDate DATETIME DEFAULT GETDATE(),  
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID),  
    FOREIGN KEY (WarehouseID) REFERENCES Warehouses(WarehouseID)  
);
```

```
CREATE TABLE Shipments (  
    ShipmentID INT PRIMARY KEY IDENTITY(1,1),  
    OrderID INT NOT NULL,  
    WarehouseID INT NOT NULL,  
    ShipmentDate DATETIME DEFAULT GETDATE(),  
    Carrier VARCHAR(100) DEFAULT 'Unknown',  
    Status VARCHAR(50) NOT NULL DEFAULT 'In Transit',  
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),  
    FOREIGN KEY (WarehouseID) REFERENCES Warehouses(WarehouseID)  
);
```