

DATABASE SYSTEM



SalesSystem

Assigned By: HOD Amir Jamshaid

Assigned To:

Muhammad Faizan(GL) 1010

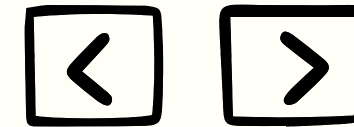
Abdullah Ramay 1028

Noman Khalid 1026

Aymen Malik 1017



Contents



01

Introduction

02

ER Diagram

03

ER Details

04

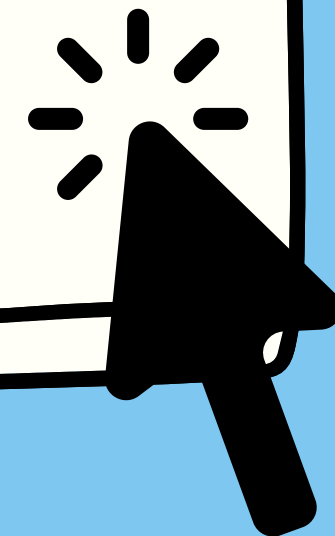
Data Flow Diagram

05

SQL Code

06

SQL Tables

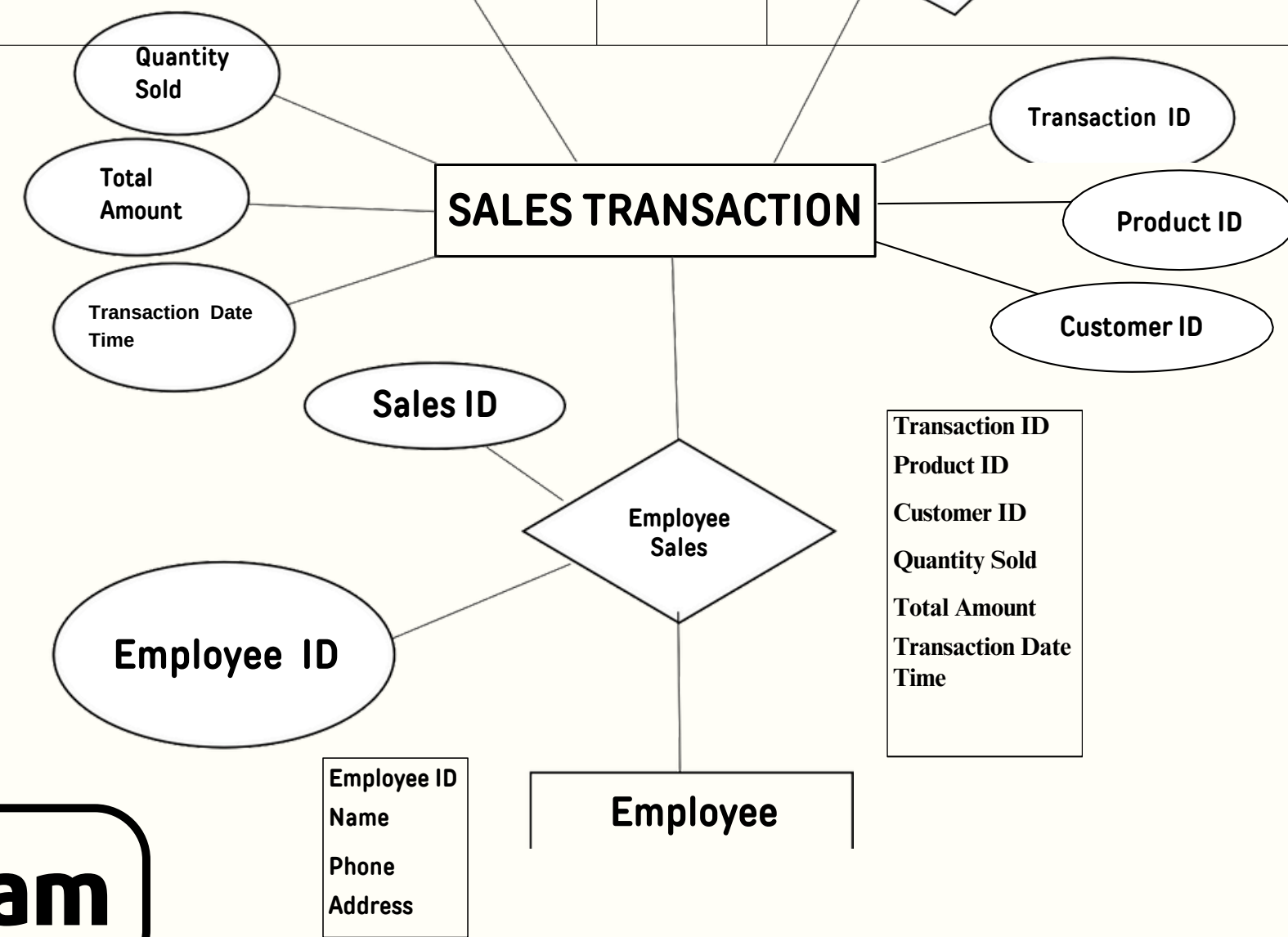
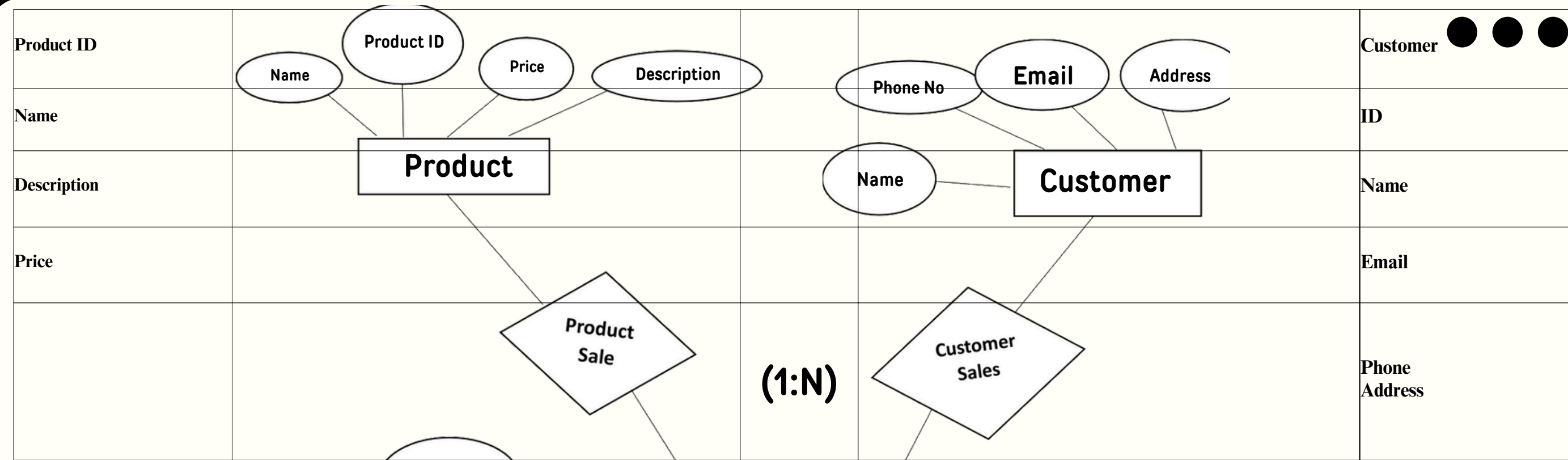




Introduction

Introducing the Smart Shop sales system, an innovative platform meticulously designed to optimize and streamline retail operations. This system features key entities such as **Product, Customer, Employee, and SalesTransaction**, each enriched with essential attributes like **product specifications, customer details, employee roles, and transaction data**.

Products and customers are seamlessly linked through sales transactions, capturing the essence of each sale with precise quantity and pricing information. Employees are pivotal in facilitating these transactions, ensuring a smooth and efficient sales process. This harmonious integration guarantees real-time tracking and management of all sales activities, propelling Smart Shop towards unparalleled efficiency and customer satisfaction.



ER Diagram

ER DETAILS

- **Entities and Attributes:**

- **Product:**

- **Attributes:** ProductID, Name, Description, Price, QuantityAvailable

- **Customer:**

- **Attributes:** CustomerID, Name, Email, Phone, Address

- **Employee:**

- **Attributes:** EmployeeID, Name, Role, Shift

- **SalesTransaction:**

- **Attributes:** TransactionID, ProductID (FK), CustomerID (FK), EmployeeID (FK), QuantitySold, TotalAmount, TransactionDateTime

- **Relationships and Connections:**

- **Product and SalesTransaction:**

- **Relationship:** A product is linked to sales transactions to record the items sold.
 - **Connection:** ProductID is a foreign key in SalesTransaction.

- **Customer and SalesTransaction:**

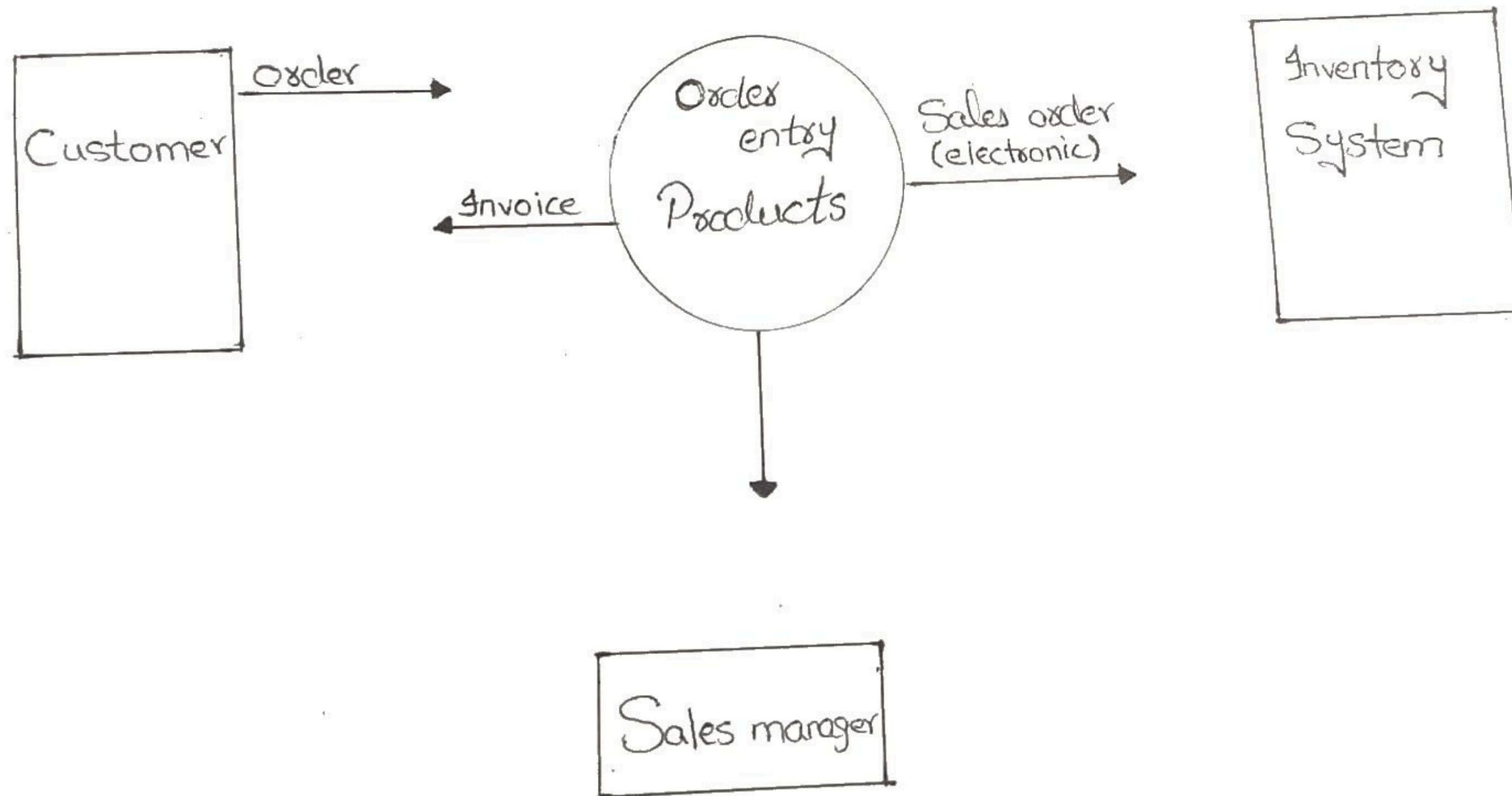
- **Relationship:** A customer is associated with sales transactions to track purchases.
 - **Connection:** CustomerID is a foreign key in SalesTransaction.

- **Employee and SalesTransaction:**

- **Relationship:** An employee facilitates sales transactions, linking sales to specific staff members.
 - **Connection:** EmployeeID is a foreign key in SalesTransaction.

DFD

DFD



DFD

Data stores

customer information

Product inventory

Sales data

Data Flows

Customer order	→	Process order
Process order	→	Inventory updates
Process order	→	Sales data
Inventory updates	→	Manage inventory
Manage inventory	→	Inventory updates
Sales data	→	Generate reports

SQL TABLE & CODE

```
CREATE DATABASE SalesSystem  
USE SalesSystem
```

-- Product Table

```
CREATE TABLE Product (  
  ProductID INT PRIMARY KEY,  
  ProductName VARCHAR(50) NOT NULL,  
  Description TEXT,  
  Price DECIMAL(10, 2) NOT NULL,  
  QuantityAvailable INT NOT NULL  
);
```

-- Customer Table

```
CREATE TABLE Customer (  
  CustomerID INT PRIMARY KEY,  
  CustomerName VARCHAR(50) NOT NULL,  
  Email VARCHAR(100) UNIQUE NOT NULL,  
  Phone VARCHAR(15),  
  Address VARCHAR(250)  
);
```


SQL TABLE & CODE

-- Employee Table

```
CREATE TABLE Employee (  
  EmployeeID INT PRIMARY KEY,  
  EmployeeName VARCHAR(50) NOT NULL,  
  Role VARCHAR(50),  
  Shift VARCHAR(50)  
);
```

-- SalesTransaction Table

```
CREATE TABLE SalesTransaction (  
  TransactionID INT PRIMARY KEY,  
  ProductID INT,  
  CustomerID INT,  
  EmployeeID INT,  
  QuantitySold INT,  
  TotalAmount DECIMAL(10, 2) NOT NULL,  
  TransactionDatetime DATETIME NOT NULL,  
  FOREIGN KEY (ProductID) REFERENCES Product(ProductID),  
  FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),  
  FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID)  
);
```

SQL TABLE & CODE

-- To perform insertion Operations

```
INSERT INTO Product (ProductID, ProductName, Description, Price, QuantityAvailable)  
VALUES
```

```
(1, 'Laptop', 'High performance Laptop', 1199.99, 40),  
(2, 'Smartphone', 'Latest model smartphone with advanced features', 799.99, 100),  
(3, 'Tablet', 'Lightweight tablet with high-resolution display', 499.99, 75),  
(4, 'Headphones', 'Noise-cancelling over-ear headphones', 199.99, 200),  
(5, 'Smartwatch', 'Water-resistant smartwatch with heart rate monitor', 299.99, 150),  
(6, 'Camera', 'Digital camera with 20MP sensor, and 4K video recording', 1399.99, 50);
```

-- Insert data into Customer Table

```
INSERT INTO Customer (CustomerID, CustomerName, Email, Phone, Address)  
VALUES
```

```
(1, 'John Doe', 'john.doe@example.com', '123-456-7890', '123 Elm St, Springfield'),  
(2, 'Jane Smith', 'jane.smith@example.com', '234-567-8901', '456 Oak St, Springfield'),  
(3, 'Michael Johnson', 'michael.johnson@example.com', '345-678-9012', '789 Pine St, Springfield'),  
(4, 'Emily Davis', 'emily.davis@example.com', '456-789-0123', '101 Maple St, Springfield'),  
(5, 'David Wilson', 'david.wilson@example.com', '567-890-1234', '202 Birch St, Springfield'),  
(6, 'Sophia Martinez', 'sophia.martinez@example.com', '678-901-2345', '303 Cedar St, Springfield');
```

SQL TABLE & CODE

-- Insert data in Employee Table

INSERT INTO Employee (EmployeeID, EmployeeName, Role, Shift)

VALUES

(1, 'Jane Smith', 'Sales Associate', 'Morning'),

(2, 'Robert Brown', 'Cashier', 'Evening'),

(3, 'Emily Clark', 'Inventory Manager', 'Morning'),

(4, 'Michael Johnson', 'Customer Service Representative', 'Afternoon'),

(5, 'Sophia Martinez', 'Assistant Manager', 'Night'),

(6, 'David Wilson', 'Sales Manager', 'Day');

-- Insert data in SalesTransaction

INSERT INTO SalesTransaction (TransactionID, ProductID, CustomerID, EmployeeID, QuantitySold, TotalAmount, TransactionDateTime)

VALUES

(1, 1, 1, 1, 2, 1999.98, '2024-05-27 14:30:00'),

(2, 2, 2, 2, 1, 799.99, '2024-05-27 15:00:00'),

(3, 3, 3, 3, 3, 1499.97, '2024-05-27 16:00:00'),

(4, 4, 4, 4, 2, 399.98, '2024-05-27 17:30:00'),

(5, 5, 5, 5, 1, 299.99, '2024-05-27 18:00:00'),

(6, 6, 6, 6, 1, 649.99, '2024-05-27 19:00:00');

SQL TABLE & CODE

-- Select data to show from Table

```
SELECT * FROM Employee;
```

```
CREATE PROCEDURE ShowSales AS
```

```
BEGIN
```

```
    SELECT * FROM Product;
```

```
    SELECT * FROM Customer;
```

```
    SELECT * FROM Employee;
```

```
    SELECT * FROM SalesTransaction;
```

```
END;
```

```
GO;
```

```
EXEC ShowSales;
```



THANKS

