ATILIM UNIVERSITY

SCHOOL OF BUSINESS

DEPARTMENT OF ECONOMICS

Course: ECON485 – Introduction to Database Systems

Instructor: Bora GÜNGÖREN

SALES SHOP DATABASE PROJECT

Sude ÖZAYDIN

20232810027

Faculty of Management

Department of Economics

Ankara

Fall 2023

The table to be prepared is related to the activities of X A.Ş. in its sales store. This store sells small household appliances, glassware, white goods, electronics, home textiles, bed bases and furniture.

- When are these products delivered?

- If there is stock in the store warehouse, the product is delivered from there, if not, it is brought from the central warehouse.

Even if the product is not delivered, a payment plan is created with the order. The person's current account record is also opened. If the product is returned or cannot be delivered for any reason, the order is canceled, but the customer record remains in the system. If sales transactions are made for more than one product, only the payment plan for the returned product is canceled.

For each payment transaction in the payment plan, a counter record is opened and deducted from the total debt amount. Each payment in the payment plan has a date. The counter entry for cancelations and refunds is opened on the same date.

When payment is made, a separate record for the payment is opened.

Different IDs are defined for each product in the store. Orders contain quantities with these IDs. The prices of the products are fixed with the order and in case of cancelation/refund, the price at the time of order is valid. The price at the time of order is pulled from a product value table.

This table contains the ID of each product, the price, the date the price was set and whether it is valid or not. When a price is updated, a new record is entered and the last valid record is overridden. When a price is queried, the current record for that product ID is queried.

EVENT FLOW: Customer comes to the store. He finds the product he needs from the relevant aisle through the sales representative. The price of the product is queried through the system. If there is any change in the price of the product, this can also be detected through the system. The product ID is used when performing these operations.

It is checked how many of the product is available and whether it is available in the store. If the product is in the store, it is delivered from the store with the sales transaction. The customer goes to the cash register with the offer sheet. A current record, i.e. customer record, is also opened at the cash register. Even if the order will not be delivered at that moment, a customer record is created. If the customer makes the payment at that moment, a payment record is opened with the payment.

Products Table: Table with basic information for each product

```sql
CREATE TABLE Products (
    ProductID INT PRIMARY KEY,
    Category VARCHAR(50),
    Name VARCHAR(100),
    Description TEXT,
    -- Diğer gerekli alanlar
);
```

Warehouse Table: Table containing warehouse information such as store warehouse and central Warehouse

```sql
CREATE TABLE Warehouses (
    WarehouseID INT PRIMARY KEY,
    Name VARCHAR(100),
    Location VARCHAR(100),
    -- Diğer gerekli alanlar
);
```

Stock Table: Used to track the amount of product stock in each warehouse.

```sql
CREATE TABLE Stock (
    StockID INT PRIMARY KEY,
    ProductID INT,
    WarehouseID INT,
    Quantity INT,
    -- Diğer gerekli alanlar
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID),
    FOREIGN KEY (WarehouseID) REFERENCES Warehouses(WarehouseID)
);
```

Table of Orders: Used to track customer orders.

```sql
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    CustomerID INT,
    OrderDate DATE,
    Status VARCHAR(50), -- Sipariş durumu (ödendi, iade edildi vb.)
    -- Diğer gerekli alanlar
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);
```

Order Details Table: Contains the quantity and price information of the products in each order.

```sql
CREATE TABLE OrderDetails (
    OrderDetailID INT PRIMARY KEY,
    OrderID INT,
    ProductID INT,
    Quantity INT,
    Price DECIMAL(10, 2),
    -- Diğer gerekli alanlar
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);
```

Payment Plan Table: It is used to follow the payment plans of customers.

```sql
CREATE TABLE PaymentPlans (
    PaymentPlanID INT PRIMARY KEY,
    CustomerID INT,
    OrderID INT,
    PaymentDate DATE,
    Amount DECIMAL(10, 2),
    -- Diğer gerekli alanlar
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID)
);
```

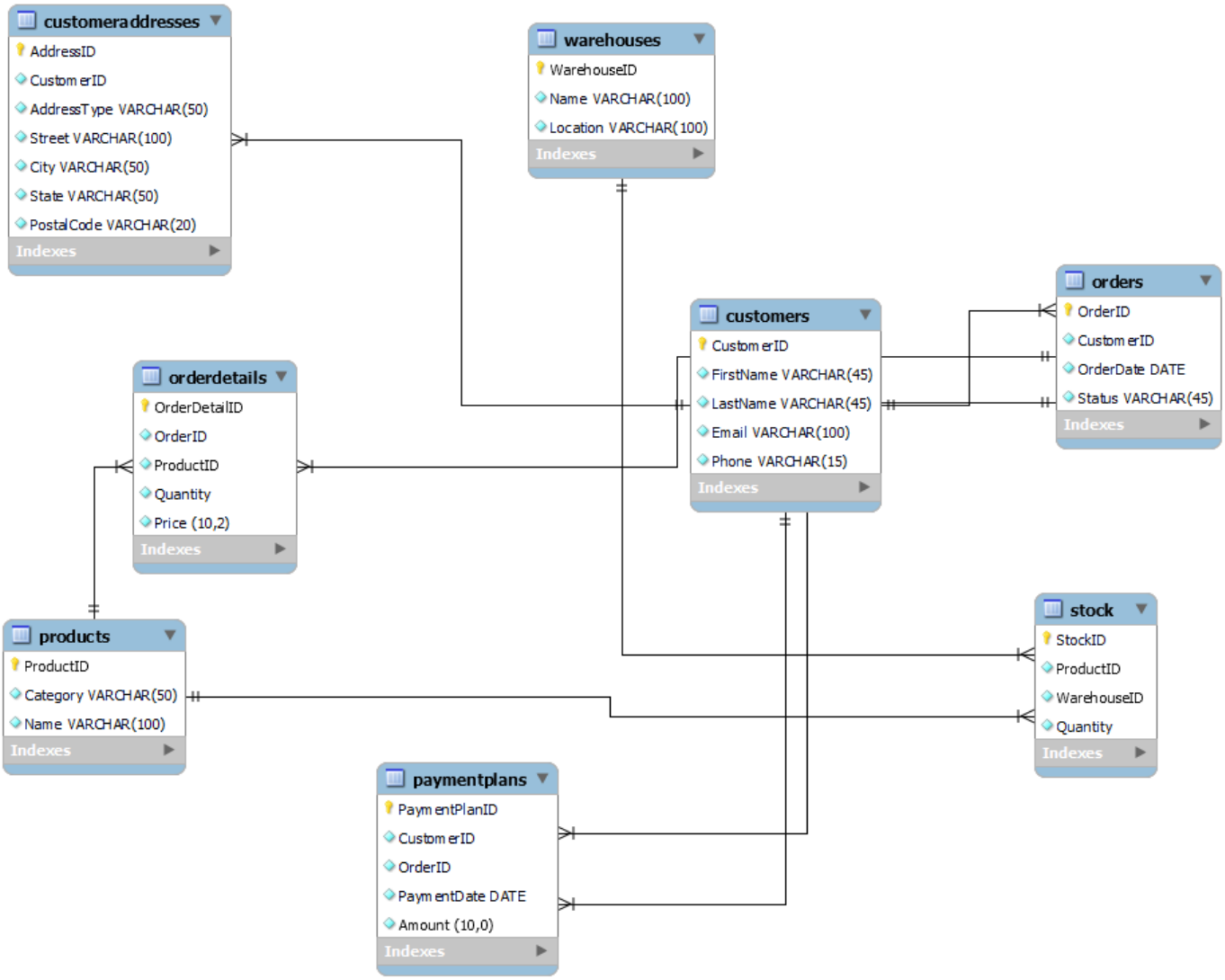
Customer Table: Table containing basic information about customers.

```sql
CREATE TABLE Customers (
    CustomerID INT PRIMARY KEY,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    Email VARCHAR(100),
    Phone VARCHAR(15),
    -- Diğer gerekli alanlar
);
```

Customer Addresses Table: Used to store multiple address information for customers.

```sql
CREATE TABLE CustomerAddresses (
    AddressID INT PRIMARY KEY,
    CustomerID INT,
    AddressType VARCHAR(50), -- Ev adresi, iş adresi vb.
    Street VARCHAR(100),
    City VARCHAR(50),
    State VARCHAR(50),
    PostalCode VARCHAR(20),
    -- Diğer gerekli alanlar
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);
```

The final table is on the last page.


customeraddresses
AddressID
CustomerID
AddressType VARCHAR(50)
Street VARCHAR(100)
City VARCHAR(50)
State VARCHAR(50)
PostalCode VARCHAR(20)
Indexes
warehouses
WarehouseID
Name VARCHAR(100)
Location VARCHAR(100)
Indexes
orders
OrderID
CustomerID
OrderDate DATE
Status VARCHAR(45)
Indexes
customers
CustomerID
FirstName VARCHAR(45)
LastName VARCHAR(45)
Email VARCHAR(100)
Phone VARCHAR(15)
Indexes
orderdetails
OrderDetailID
OrderID
ProductID
Quantity
Price (10,2)
Indexes
products
ProductID
Category VARCHAR(50)
Name VARCHAR(100)
Indexes
stock
StockID
ProductID
WarehouseID
Quantity
Indexes
paymentplans
PaymentPlanID
CustomerID
OrderID
PaymentDate DATE
Amount (10,0)
Indexes