



**EGE UNIVERSITY**  
**FACULTY OF ENGINEERING**  
**COMPUTER ENGINEERING DEPARTMENT**

**INTRODUCTION TO DATABASES**  
**2024–2025 FALL SEMESTER**

**BIM302-2024-2025-Project**

**DELIVERY DATE**

17/01/2025

**PREPARED BY**

05210000004, Amin Azizzade

05210000194, Melih Kumru

05210000273, Muhammed Orhantekin

05210000297, Ömer Karaduman

**TEAM NO**

# ANALYSIS

## QUESTION;

(4 pts for each website) Examine the websites below and write the data requirements for each website that fit the main scenario above.

## ANSWER;

### 1- Amazon Data Requirement

The Amazon e-commerce platform is designed to enhance user experience and provide flexibility. There are two main user types on this platform: Sellers and Customers. A user can be both a customer and a seller. All users are required to provide basic information during registration, including a unique UserID, email address, password, phone number, and registration date. These details are mandatory regardless of the user type.

After registration, users who wish to become sellers must provide additional information. This includes whether the seller is an individual or a business, the business type, company name (even if individual), tax number, IBAN, store location, store name, and business address. Sellers also manage their inventory, which consists of attributes such as product name, active status, barcode, quantity, and storage location. Sellers can add or update their stock as necessary, ensuring that product information is accurate and up-to-date.

For users registering as customers, it is sufficient to create a username. Customers can manage their shipping details, which include attributes like full name, address lines, city, zip code, country, neighborhood, phone number, and district. Customers also have the option to save multiple addresses for convenience, such as home and workplace.

Amazon also offers a subscription system available to all users. Subscription plans include attributes like subscription type, start date, renewal date, and status. By subscribing, users gain special privileges, such as exclusive campaigns, discounts, and gifts.

The platform enables users to save and manage their payment details, including card number, card expiry date, cardholder name, and CVV. Additionally, customers can use gift cards and promotional codes during payment to save more. Promotional codes have attributes such as code type, description, start date, end date, and value, making the shopping experience more appealing and cost-effective.

Amazon provides users with the ability to create their own custom lists. Each list is identified by attributes like list name and quantity. Users can name these lists as they wish and add their favorite products to them, allowing for a more organized and personalized shopping experience.

The platform hosts hundreds of thousands of products, which are added to the system by sellers. Each product is assigned a unique ASIN number, product name, brand, price, currency, and release date. Products are categorized using attributes like category name and subcategory, making searches and filtering faster and more practical for customers. Sellers can stock their products in any quantity they wish and update inventory information, which includes barcode and storage location.

Every product is added to the system with detailed descriptions and features. The same product or products with the same barcode can be offered by different sellers. Customers can evaluate sellers based on their ratings and product reviews. Reviews consist of attributes such as rating number, rating date, and comments, enabling customers to make informed purchasing decisions.

Customers can easily search for their desired products from the wide range available on the platform and add them to their cart. Shopping carts are identified by attributes such as cart ID, quantity, and total amount. Once customers confirm their cart, they must provide payment information and delivery address details to create

an order. Orders are identified by attributes such as order date, total amount, and order status. Each order consists of multiple order items, which include attributes like order item ID, quantity, and price.

Once orders are confirmed, shipments are prepared with details like shipment number, shipping company, and tracking number. Shipments include attributes such as delivery status, delivery date, and estimated delivery time. Orders are then carefully delivered to customers by shipping companies.

To enhance customer satisfaction, Amazon regularly organizes campaigns, promotions, and discounts. Discounts have attributes like type, start date, end date, and value, and they apply to specific products or entire categories. These opportunities ensure both economical and enjoyable shopping experiences for customers.

## **2. Hepsiburada Data Requirement**

The Hepsiburada e-commerce platform is designed to provide its users with a modern shopping experience.

All users on the platform are uniquely identified by a UserID and have basic information such as gender, date of birth, IBAN, email, and phone number. Users can add personal data, including shipment details and payment information. They can also benefit from additional services through the subscription system and Hepsipay wallet accounts. Users can operate on the platform as both sellers and customers.

Shipment information includes data such as an address title, address type (e.g., invoice or delivery), full address details (city, district, street), postal code, country, and phone number, allowing users to save their addresses in detail. For payment details, users can add information such as PaymentID, card number, expiration date, CVV, and cardholder name. User subscriptions are tracked with SubscriptionID, start date, end date, renewal date, and fee information.

Sellers are identified with details such as store name, type, rating, total sales, and tax identification number. Sellers can list their products for sale by managing their inventory and participate in campaigns. On the other hand, customers can create product lists and favorite lists, manage their shopping carts, and place orders. Additionally, customers can follow sellers and earn discounts with coupons.

Products on the platform are recorded with details such as ProductID, product name, brand, price, barcode, and detailed descriptions. Products can belong to one or more categories, while categories are grouped under CategoryID and can be divided into subcategories. Inventory and stock management enable sellers to track and update product stock. The inventory is identified by InventoryID, while stock information consists of data such as product quantity, barcode, and price.

Campaigns and promotions can be applied to products or shopping carts with PromotionID. Promotions are offered in the form of coupons or discounts and are detailed with usage conditions. Shopping carts are composed of one or more products and identified by CartID, and customers can convert these carts into orders. Orders are defined by OrderNo, order date, total amount, status, and address details. Each order can include multiple order items, containing information such as product quantity and status. Order items are identified by OrderItemNo.

Logistics processes are tracked with LogisticID, including carrier name, delivery status, and tracking number to monitor the delivery of orders. Shipment details, identified by TrackingNumber, allow for tracking delivery times and statuses of orders. Payments are linked to PaymentID, payment method, amount, and payment details. Users can manage their balances and make payments using HepsipayID. Customers can also leave reviews for the products they have ordered. These reviews include information such as ReviewNumber, rating, date, and description.

This data model of Hepsiburada enables detailed tracking of critical operations such as user management, order processes, payment transactions, and logistics, designed to optimize the customer experience.

### **3. Trendyol Data Requirement**

Trendyol e-commerce platform is designed to simplify user experience and provide flexibility. The platform has two main user types: sellers and customers. A user can act as both a customer and a seller.

Each user is identified by unique information such as UserID, password, date of birth, multiple last login records, registration date, email, and phone number. Users have one or more shipment details and payment information, which are mandatory regardless of the user type.

Sellers are identified by a unique SellerID and details such as name, store name, tax identification number, store rating, store opening date, total sales, IBAN, and sales category type. Sellers manage inventory details, including product names, activity status, barcode numbers, quantity, and storage location. They can increase or update their stock as needed, ensuring product information is always accurate and up to date. Additionally, sellers are associated with campaigns.

Customers are identified by a unique CustomerID and name information and can create collections. Each collection has unique attributes such as CollectionNo, collection name, and type. Customers can also add favorite products, fill their shopping carts, and place orders.

Shipment details include unique attributes such as Shipping DetailID, address title, city, district, neighborhood, address line, name, phone number, and invoice type. These details can be associated with multiple orders.

Payment details include a unique Payment DetailID and information such as card number, expiration date, cardholder name, CVV, and billing country. These can be linked to multiple payment records.

The platform contains hundreds of thousands of products added by sellers. Each product is recorded with unique details such as ProductID, product name, description, brand, price, and features. Products are also classified with CategoryID, category name, and subcategory information, making search and filtering faster and more practical for customers. Sellers can stock their products in any quantity and update inventory information.

Each product is added to the system with detailed descriptions and features. Products with the same barcode can also be offered for sale by different sellers. Customers can shop based on seller ratings and product reviews. Reviews include unique attributes such as review number, review date, and comment, helping customers make informed decisions.

Orders have features such as order date, total amount, and order status. Each order consists of multiple order items, and these items include unique order item IDs, quantity, and price information.

Campaigns are defined by unique CampaignID, name, description, start and end dates. A campaign can be either a promotion or a discount. Promotions are detailed with codes or coupons, while discounts have unique DiscountID, start date, end date, type, and conditions.

Shopping carts have a unique ShoppingCartID and are defined by the derived total amount and quantity, eventually forming a complete order.

Payments include payment method, amount, and card information and are linked to orders.

Shipments are tracked with unique attributes such as TrackingNumber, delivery dates, package weight, and carrier name, and are processed by logistic units. Logistic units have a unique LogisticID.

These structures are designed to comprehensively track and manage all processes within the Trendyol e-commerce system.

## DESIGN-CONCEPTUAL DESIGN

### QUESTION1;

(5 pts for each EER) Create separate EER diagrams for each website according to your data requirements.

### ANSWER;

All EER files are available in the other PDF documents.

### QUESTION2

(23) Combine all E2/R diagrams. Explain your method before starting the assembly process.

### ANSWER;

As a result of a comprehensive analysis conducted among e-commerce platforms, we observed that the data requirements of Amazon, Trendyol, and Hepsiburada are shaped around similar fundamental structures. Although all three platforms share common functionalities, each also has unique and distinctive features that stand out. These differences play a key role in shaping user experiences and providing competitive advantages.

In our analysis, we identified the common entities and relationships across all platforms and integrated these elements into our new EER diagram. At the same time, we incorporated the unique features offered by each platform into the structure to create a robust and functional model.

The following steps were taken to develop our new unified EER diagram:

#### 1. Common Entities and Relationships

We identified the entities and relationships shared across all three platforms and used them as the foundation for the new structure.

#### 2. Cross-Platform Differences and Integration

While some entity names differed across platforms, their functionalities were identical. Therefore, we unified them under a single structure.

#### 3. Inclusion of Unique Features

We included entities and attributes unique to each platform that we considered beneficial:

- Added the subscription system from Amazon and Hepsiburada.
- Integrated gift cards and promotional codes, available during the payment process, from Amazon.
- Included the campaign system from Hepsiburada and Trendyol (although campaigns also exist on Amazon, they were observed to be less detailed than on the other two platforms).
- Added the "LastLogin" attribute from Trendyol.
- Incorporated the feature of customers being able to follow sellers from Trendyol and Hepsiburada into the new structure.

#### 4. Excluded Features

Certain features present on some platforms were excluded from the new EER structure:

- Did not include the coupon system or the direct coupon structure, which can be entered at the cart stage on Trendyol.
- Excluded Hepsiburada's "HepsiPay" digital wallet feature.

#### Conclusion

In creating our new EER diagram, we combined the entities and attributes present in the three analyzed platforms that we deemed useful without deviating from the main scenario. By integrating the strengths of different platforms, we aimed to achieve a functional and comprehensive structure.

## DESIGN-LOGICAL MODEL

#### QUESTION;

(18) Convert EER diagram into a relational model using the methodology that will be introduced in your course.

#### ANSWER;

All mapping tables are available in the other files.

## IMPLEMENTATION-PHYSICAL MODEL

#### QUESTION;

(4 pts) Write down the appropriate SQL scripts (DDL statements) for creating the database and its relational model. You can select any of the DBMS you wish.

#### ANSWER;

```
-- 1.iteration

CREATE TABLE IF NOT EXISTS `User` (
  UserID INT PRIMARY KEY AUTO_INCREMENT,
  `Password` VARCHAR(255) NOT NULL,
  BirthDate DATE NULL,
  RegistrationDate DATE NOT NULL,
  Email VARCHAR(255) UNIQUE NOT NULL,
  PhoneNumber VARCHAR(15) NOT NULL
```

```
);
```

```
CREATE TABLE IF NOT EXISTS Campaign (  
    CampaignID INT PRIMARY KEY AUTO_INCREMENT,  
    `Name` VARCHAR(255) NOT NULL,  
    `Description` TEXT,  
    StartDate DATE,  
    EndDate DATE  
);
```

```
CREATE TABLE if not exists ShippingDetail (  
    ShippingID INT PRIMARY KEY AUTO_INCREMENT,  
    AddressTitle VARCHAR(255) NOT NULL,  
    AddressLine VARCHAR(255) NOT NULL,  
    City VARCHAR(100) NOT NULL,  
    District VARCHAR(100) NOT NULL,  
    Neighborhood VARCHAR(100) NOT NULL,  
    BillType ENUM('Personal', 'Corporate') NOT NULL,  
    FirstName VARCHAR(100) NOT NULL,  
    LastName VARCHAR(100) NOT NULL,  
    PhoneNumber VARCHAR(15) NOT NULL  
);
```

```
CREATE TABLE if not exists Subscription (  
    SubscriptionID INT PRIMARY KEY AUTO_INCREMENT,  
    StartDate DATE NOT NULL,  
    EndDate DATE,  
    RenewalDate DATE NOT NULL,  
    Fee DECIMAL(10, 2) NOT NULL  
);
```

```
CREATE TABLE IF NOT EXISTS PaymentDetail (  
    PaymentDetailID INT PRIMARY KEY AUTO_INCREMENT,  
    CardNumber VARCHAR(16) NOT NULL,  
    CardExpiryDate DATE NOT NULL,  
    CardholderName VARCHAR(100) NOT NULL,  
    CVV VARCHAR(3) NOT NULL);
```

```
CREATE TABLE IF NOT EXISTS Inventory (  
    InventoryID INT PRIMARY KEY AUTO_INCREMENT,  
    AddressLine VARCHAR(255) NOT NULL,  
    `InventoryName` VARCHAR(255) NOT NULL  
);
```

```
CREATE TABLE IF NOT EXISTS Product (  
    ProductID INT PRIMARY KEY AUTO_INCREMENT,  
    ProductName VARCHAR(255) NOT NULL,  
    Information TEXT,  
    Brand VARCHAR(100),  
    Price DECIMAL(10, 2) NOT NULL,  
    Properties TEXT  
);
```

```
CREATE TABLE IF NOT EXISTS Category (  
    CategoryID INT PRIMARY KEY AUTO_INCREMENT,  
    CategoryName VARCHAR(100) NOT NULL  
);
```

```
CREATE TABLE IF NOT EXISTS Logistic (  
    LogisticID INT PRIMARY KEY AUTO_INCREMENT  
);
```

```
CREATE TABLE IF NOT EXISTS Shipment (  
    TrackingNumber VARCHAR(50) PRIMARY KEY,  
    PackageWeight DECIMAL(10, 2) NOT NULL,  
    PackageDimension VARCHAR(100) NOT NULL,  
    CarrierName VARCHAR(100) NOT NULL,  
    DeliveryStatus ENUM('In Transit', 'Out for Delivery', 'Delivered') NOT  
NULL,  
    DeliveryDate DATE ,  
    EstimatedDeliveryDate DATE NOT NULL,  
    ShipmentDate DATE NOT NULL  
);
```

```
CREATE TABLE IF NOT EXISTS Payment (  
    PaymentID INT PRIMARY KEY AUTO_INCREMENT,  
    PaymentAmount DECIMAL(10, 2) NOT NULL,  
    PaymentMethod ENUM('CreditCard', 'DebitCard') NOT NULL,  
    CardNumber VARCHAR(16) NULL,  
    CardExpiryDate DATE NULL,  
    CardholderName VARCHAR(100) NULL,  
    CVV VARCHAR(3) NULL  
);
```

```
CREATE TABLE IF NOT EXISTS PromotionCode (  
PromotionCodeID INT PRIMARY KEY AUTO_INCREMENT,  
    StartDate DATE NOT NULL,  
    EndDate DATE NOT NULL,  
    CodeType ENUM('Coupon', 'GiftCard', 'Code') NOT NULL,  
    Information TEXT  
);
```

```
CREATE TABLE IF NOT EXISTS ShoppingCart (  
    ShoppingCartID INT PRIMARY KEY AUTO_INCREMENT  
);
```

```
CREATE TABLE Stock (  
    InventoryID INT NOT NULL,  
    Barcode CHAR(13) NOT NULL,  
    Quantity INT NOT NULL,
```



```

        ListPrice DECIMAL(10, 2),
        PRIMARY KEY (InventoryID, Barcode),
        FOREIGN KEY (InventoryID) REFERENCES Inventory(InventoryID)
        ON DELETE CASCADE
        ON UPDATE CASCADE
    );

CREATE TABLE IF NOT EXISTS AppliedToPayment (
    PaymentID INT PRIMARY KEY,
    PromotionCodeID INT UNIQUE ,
    FOREIGN KEY (PromotionCodeID) REFERENCES
PromotionCode(PromotionCodeID)
        ON DELETE SET NULL ON UPDATE CASCADE,
    FOREIGN KEY (PaymentID) REFERENCES Payment(PaymentID)
        ON DELETE CASCADE ON UPDATE CASCADE
);

ALTER TABLE Subscription
ADD COLUMN UserID INT NOT NULL,
ADD CONSTRAINT FK_Subscription_User FOREIGN KEY (UserID) REFERENCES
User(UserID)
        ON DELETE RESTRICT
        ON UPDATE CASCADE,
ADD CONSTRAINT UC_Subscription_User UNIQUE (UserID);

CREATE TABLE ProductStock (
    ProductID INT ,
    Barcode CHAR(13) ,
    InventoryID INT ,
    PRIMARY KEY (ProductID,Barcode,InventoryID),
    FOREIGN KEY (ProductID) REFERENCES Product(ProductID)
        ON DELETE CASCADE
        ON UPDATE CASCADE,

        FOREIGN KEY (InventoryID, Barcode) REFERENCES Stock(InventoryID,
Barcode)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

```

```
ALTER TABLE ShippingDetail
ADD COLUMN UserID INT NOT NULL,
ADD FOREIGN KEY (UserID) REFERENCES User(UserID)
    ON DELETE CASCADE
    ON UPDATE CASCADE;
```

```
ALTER TABLE PaymentDetail
ADD COLUMN UserID INT NOT NULL,
ADD FOREIGN KEY (UserID) REFERENCES User(UserID)
    ON DELETE CASCADE
    ON UPDATE CASCADE;
```

```
ALTER TABLE Category
ADD COLUMN ParentCategoryID INT NULL,
ADD CONSTRAINT FK_Category_SubCategory FOREIGN KEY (ParentCategoryID)
REFERENCES Category(CategoryID)
    ON DELETE SET NULL
    ON UPDATE CASCADE;
```

```
ALTER TABLE Payment
ADD COLUMN PaymentDetailID INT,
ADD CONSTRAINT FK_Payment_PaymentDetail FOREIGN KEY (PaymentDetailID)
REFERENCES PaymentDetail(PaymentDetailID)
    ON DELETE SET NULL
    ON UPDATE CASCADE;
```

```
ALTER TABLE Shipment
ADD COLUMN LogisticID INT NOT NULL,
ADD CONSTRAINT FK_Shipment_Logistic FOREIGN KEY (LogisticID) REFERENCES
Logistic(LogisticID)
    ON DELETE RESTRICT
    ON UPDATE CASCADE;
```

```
CREATE TABLE IF NOT EXISTS CampaignCart (
    CampaignID INT NOT NULL,
    ShoppingCartID INT NOT NULL,
    PRIMARY KEY (CampaignID, ShoppingCartID),
    FOREIGN KEY (CampaignID) REFERENCES Campaign(CampaignID)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (ShoppingCartID) REFERENCES ShoppingCart(ShoppingCartID)
        ON DELETE CASCADE
        ON UPDATE CASCADE
```

```
);
```

```

CREATE TABLE IF NOT EXISTS CampaignProduct (
    CampaignID INT NOT NULL,
    ProductID INT NOT NULL,
    PRIMARY KEY (CampaignID, ProductID),
    FOREIGN KEY (CampaignID) REFERENCES Campaign(CampaignID)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (ProductID) REFERENCES Product(ProductID)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

```

```

CREATE TABLE IF NOT EXISTS CategoryProduct (
    CategoryID INT NOT NULL,
    ProductID INT NOT NULL,
    PRIMARY KEY (CategoryID, ProductID),
    FOREIGN KEY (CategoryID) REFERENCES Category(CategoryID)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (ProductID) REFERENCES Product(ProductID)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

```

```

CREATE TABLE IF NOT EXISTS CartItems (
    ProductID INT NOT NULL,
    ShoppingCartID INT NOT NULL,
    Quantity INT NOT NULL,
    PRIMARY KEY (ProductID, ShoppingCartID),
    FOREIGN KEY (ProductID) REFERENCES Product(ProductID)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (ShoppingCartID) REFERENCES ShoppingCart(ShoppingCartID)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

```

```

CREATE TABLE IF NOT EXISTS UserLastLogin (
    UserID INT NOT NULL,
    LastLogin DATETIME NOT NULL,
    PRIMARY KEY (UserID, LastLogin),
    FOREIGN KEY (UserID) REFERENCES User(UserID)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

```

```

CREATE TABLE IF NOT EXISTS Seller (
    SellerID INT PRIMARY KEY,
    FirstName VARCHAR(100) NOT NULL,
    LastName VARCHAR(100) NOT NULL,
    StoreName VARCHAR(255) NOT NULL,
    TaxID VARCHAR(50) NOT NULL,
    StoreRating DECIMAL(3, 2),
    SellerType ENUM('Individual', 'Corporate') NOT NULL,
    StoreDate DATE NOT NULL,
    TotalSellNumber INT,
    IbanNo VARCHAR(50),
    SalesCategoryType VARCHAR(255),
    FOREIGN KEY (SellerID) REFERENCES `User` (UserID)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

```

```

CREATE TABLE IF NOT EXISTS Customer (
    CustomerID INT PRIMARY KEY,
    FirstName VARCHAR(100) NOT NULL,
    LastName VARCHAR(100) NOT NULL,
    FOREIGN KEY (CustomerID) REFERENCES `User` (UserID)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

```

```

ALTER TABLE Campaign
    ADD COLUMN IsPromotion BOOLEAN NOT NULL DEFAULT FALSE,
    ADD COLUMN PromotionType ENUM('Seasonal', 'LimitedTime', 'NewArrival',
    'Clearance', 'BuyOneGetOne'),
    ADD COLUMN PromotionStartDate DATE,
    ADD COLUMN PromotionEndDate DATE,
    ADD COLUMN PromotionConditions TEXT,
    ADD COLUMN `Status` ENUM('Active', 'Upcoming', 'Expired',
    'Cancelled'),
    ADD COLUMN IsDiscount BOOLEAN NOT NULL DEFAULT FALSE,
    ADD COLUMN DiscountType ENUM('Flat', 'Percentage', 'FreeShipping'),
    ADD COLUMN DiscountStartDate DATE,
    ADD COLUMN DiscountEndDate DATE,
    ADD COLUMN DiscountValue DECIMAL(10, 2),
    ADD COLUMN DiscountConditions TEXT;

```

-- 2.iteration

```
CREATE TABLE IF NOT EXISTS `Order` (  
    OrderNo INT NOT NULL,  
    CustomerID INT NOT NULL,  
    ShoppingCartID INT NOT NULL,  
    OrderDate DATE NOT NULL,  
    OrderStatus ENUM('Pending', 'Processing', 'Shipped', 'Delivered',  
'Cancelled') NOT NULL,  
    TotalAmount DECIMAL(10, 2) NOT NULL,  
    PRIMARY KEY (OrderNo, CustomerID, ShoppingCartID),  
    FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)  
        ON DELETE RESTRICT  
        ON UPDATE CASCADE,  
    FOREIGN KEY (ShoppingCartID) REFERENCES ShoppingCart(ShoppingCartID)  
        ON DELETE RESTRICT  
        ON UPDATE CASCADE  
);
```

```
CREATE TABLE IF NOT EXISTS Collection (  
    CollectionName VARCHAR(255) NOT NULL,  
    CustomerID INT NOT NULL,  
    PRIMARY KEY (CollectionName, CustomerID),  
    FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
);
```

```
ALTER TABLE `Order`  
ADD COLUMN PaymentID INT NOT NULL,  
ADD COLUMN PaymentDate DATE ,  
ADD COLUMN PaymentStatus ENUM('Completed', 'Failed', 'Cancelled') NOT  
NULL,  
ADD CONSTRAINT FK_Order_Payment FOREIGN KEY (PaymentID) REFERENCES  
Payment(PaymentID)  
    ON DELETE RESTRICT  
    ON UPDATE CASCADE,  
ADD CONSTRAINT UC_Order_Payment UNIQUE (PaymentID);
```

```
ALTER TABLE `Order`  
ADD COLUMN LogisticID INT NOT NULL,  
ADD CONSTRAINT FK_Order_Logistic FOREIGN KEY (LogisticID) REFERENCES  
Logistic(LogisticID)  
    ON DELETE RESTRICT  
    ON UPDATE CASCADE,  
ADD CONSTRAINT UC_Order_Logistic UNIQUE (LogisticID);
```

```
ALTER TABLE Customer  
ADD COLUMN ShoppingCartID INT NOT NULL,  
ADD CONSTRAINT FK_Customer_ShoppingCart FOREIGN KEY (ShoppingCartID)  
REFERENCES ShoppingCart(ShoppingCartID)  
    ON DELETE RESTRICT  
    ON UPDATE CASCADE,  
ADD CONSTRAINT UQ_Customer_ShoppingCart UNIQUE (ShoppingCartID);
```

```
ALTER TABLE PromotionCode
ADD COLUMN CustomerID INT,
ADD FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)
    ON DELETE CASCADE
    ON UPDATE CASCADE;
```

```
ALTER TABLE Product
ADD COLUMN SellerID INT NOT NULL,
ADD COLUMN ShippingCost INT NOT NULL DEFAULT 0,
ADD FOREIGN KEY (SellerID) REFERENCES Seller(SellerID)
    ON DELETE CASCADE
    ON UPDATE CASCADE;
```

```
ALTER TABLE Payment
ADD COLUMN CustomerID INT NOT NULL,
ADD FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)
    ON DELETE CASCADE
    ON UPDATE CASCADE;
```

```
ALTER TABLE Inventory
ADD COLUMN SellerID INT NOT NULL,
ADD FOREIGN KEY (SellerID) REFERENCES Seller(SellerID)
    ON DELETE CASCADE
    ON UPDATE CASCADE;
```

```
ALTER TABLE `Order`
ADD COLUMN ShippingID INT NOT NULL,
ADD FOREIGN KEY (ShippingID) REFERENCES ShippingDetail(ShippingID)
    ON DELETE RESTRICT
    ON UPDATE CASCADE;
```

```
CREATE TABLE IF NOT EXISTS Follow (
    SellerID INT NOT NULL,
    CustomerID INT NOT NULL,
    PRIMARY KEY (SellerID, CustomerID),
    FOREIGN KEY (SellerID) REFERENCES Seller(SellerID)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```

```

CREATE TABLE IF NOT EXISTS SellerCampaign (
    SellerID INT NOT NULL,
    CampaignID INT NOT NULL,
    PRIMARY KEY (SellerID, CampaignID),
    FOREIGN KEY (SellerID) REFERENCES Seller(SellerID)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (CampaignID) REFERENCES Campaign(CampaignID)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

```

```

CREATE TABLE IF NOT EXISTS CollectionHas (
    CollectionName VARCHAR(255) NOT NULL,
    CustomerID INT NOT NULL,
    ProductID INT NOT NULL,
    PRIMARY KEY (CollectionName, CustomerID, ProductID),
    FOREIGN KEY (CollectionName, CustomerID) REFERENCES
Collection(CollectionName, CustomerID)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (ProductID) REFERENCES Product(ProductID)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

```

-- 3.iteration

```

CREATE TABLE IF NOT EXISTS OrderItem (
    OrderNo INT NOT NULL,
    CustomerID INT NOT NULL,
    ShoppingCartID INT NOT NULL,
    ProductID INT NOT NULL,
    OrderItemNo INT NOT NULL,
    Quantity INT NOT NULL,
    ItemStatus ENUM('Pending', 'Processing', 'Shipped', 'Delivered',
'Cancelled') NOT NULL,
    PRIMARY KEY (OrderNo, CustomerID, ShoppingCartID, ProductID,
OrderItemNo),
    FOREIGN KEY (OrderNo, CustomerID, ShoppingCartID) REFERENCES
`Order`(OrderNo, CustomerID, ShoppingCartID)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (ProductID) REFERENCES Product(ProductID)
        ON DELETE RESTRICT
        ON UPDATE CASCADE
);

```

```
ALTER TABLE OrderItem
ADD COLUMN TrackingNumber VARCHAR(50) ,
ADD FOREIGN KEY (TrackingNumber) REFERENCES Shipment(TrackingNumber)
    ON DELETE RESTRICT
    ON UPDATE CASCADE;
```

-- 4.iteration

```
CREATE TABLE IF NOT EXISTS Review (
    ReviewNo INT NOT NULL,
    OrderNo INT NOT NULL,
    CustomerID INT NOT NULL,
    ShoppingCartID INT NOT NULL,
    ProductID INT NOT NULL,
    OrderItemNo INT NOT NULL,
    ReviewDate DATE NOT NULL,
    Rating INT NOT NULL ,
    `Comment` TEXT NULL,
    PRIMARY KEY (OrderNo, CustomerID, ShoppingCartID, ProductID,
OrderItemNo,ReviewNo),
    FOREIGN KEY (OrderNo, CustomerID, ShoppingCartID, ProductID,
OrderItemNo) REFERENCES OrderItem(OrderNo, CustomerID, ShoppingCartID,
ProductID, OrderItemNo)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```



## QUESTION;

(4 pts) Populate the database you just created again using SQL script file loaded with sample tuples. The initial database should have the tuples of our department. (The tables should have enough tuples for the SELECT statements to be run accordingly.)

## ANSWER;

```
INSERT INTO `User` (`Password`, BirthDate, RegistrationDate, Email,
PhoneNumber)
VALUES
('sifre123', '1990-05-15', '2023-06-01', 'mehmet.kaya@example.com',
'05321234567'),
('gucluSifre456', '1985-11-22', '2023-07-10', 'ayse.yilmaz@example.com',
'05329876543'),
('kolayPass789', '2000-03-10', '2023-08-20', 'ali.can@example.com',
'05445551234'),
('Sifre@2024', '1995-07-18', '2023-09-15', 'fatma.demir@example.com',
'05559876543'),
('GuvanlıKey999', NULL, '2023-10-01', 'hakan.oz@example.com',
'05667777888'),
('MySecretPass', '1992-02-28', '2023-11-01', 'zeynep.turan@example.com',
'05334445566'),
('Key1234', '1998-06-15', '2023-09-12', 'mustafa.dogan@example.com',
'05332221100'),
('TestPassword', NULL, '2023-06-01', 'elif.tas@example.com',
'05539988877'),
('StrongPassword1!', '1993-08-08', '2023-08-15', 'kemal.ak@example.com',
'05661122334'),
('SafeLogin2023', '1987-03-01', '2023-07-10', 'derya.uzun@example.com',
'05445566778'),
('SecureP@ss', '1996-12-15', '2023-09-05', 'selin.gunes@example.com',
'05449933445'),
('MyUniquePass', '1991-01-10', '2023-10-12', 'orhan.celik@example.com',
'05331129988'),
('Access2024!', NULL, '2023-08-19', 'deniz.akyuz@example.com',
'05338889900'),
('NewSecret1', '1994-11-05', '2023-06-15', 'ahmet.kilic@example.com',
'05341145577'),
('Secure!2023', '1989-07-25', '2023-09-05', 'baris.turan@example.com',
'05678887766'),
('P@ssword2024', '1990-04-22', '2023-07-18', 'melisa.korkmaz@example.com',
'05446678899'),
('SafePass!#', NULL, '2023-10-02', 'ipek.can@example.com', '05347778899'),
('StrongCode2023', '1986-02-11', '2023-09-25', 'kemal.ulus@example.com',
'05449977666'),
('MyUniqueKey', '1993-05-19', '2023-07-22', 'cansu.yildiz@example.com',
'05331112244'),
('SecureAccess', '1997-06-28', '2023-10-20', 'burak.gok@example.com',
'05557778822'),
('Unique@2023', '1995-05-05', '2023-07-30', 'eda.arslan@example.com',
'05327778899'),
('StrongPass@2024', '1992-10-10', '2023-07-18', 'emre.akyuz@example.com',
'05336678888'),
('LoginSecure', '1985-01-15', '2023-06-10', 'aylin.dikmen@example.com',
'05337765588'),
('MyKeyAccess', '1988-11-11', '2023-10-15', 'burak.yildirim@example.com',
```

```

'05559992233'),
('SuperPass2023', '1983-03-22', '2023-08-25', 'yasemin.deniz@example.com',
'05443321122'),
('Easy1234', '1994-12-01', '2023-07-10', 'ali.sahin@example.com',
'05441123344'),
('KeyLogin', NULL, '2023-10-01', 'zehra.yildiz@example.com',
'05339976655'),
('SafeLock', NULL, '2023-09-20', 'mert.celik@example.com', '05447778811'),
('SecureTest!1', '1996-07-14', '2023-09-12', 'ece.demir@example.com',
'05334455666'),
('Access@Code', '1999-01-25', '2023-08-12', 'can.kurt@example.com',
'05346678899');

```

```

INSERT INTO ShippingDetail (AddressTitle, AddressLine, City, District,
Neighborhood, BillType, FirstName, LastName, PhoneNumber, UserID)
VALUES
('Ev Adresi', 'Atatürk Cad. No:123', 'İstanbul', 'Kadıköy', 'Bostancı',
'Personal', 'Mehmet', 'Kaya', '05321234567', 1),
('Ofis Adresi', 'Cumhuriyet Cad. No:45', 'Ankara', 'Çankaya', 'Kızılay',
'Corporate', 'Mehmet', 'Kaya', '05321234567', 1),
('Ev Adresi', 'Barış Sokak No:67', 'İzmir', 'Bornova', 'Kazımdirik',
'Personal', 'Ayşe', 'Yılmaz', '05329876543', 2),
('Ev Adresi', 'Fatih Mah. Çiçek Sokak No:11', 'Bursa', 'Osmangazi',
'Altıparmak', 'Personal', 'Ali', 'Can', '05445551234', 3),
('Yazlık', 'Güzelbahçe Mah. Deniz Yolu No:22', 'Antalya', 'Konyaaltı',
'Liman', 'Personal', 'Ali', 'Can', '05445551234', 3),
('Ofis Adresi', 'Hürriyet Cad. No:99', 'İstanbul', 'Beşiktaş', 'Levent',
'Corporate', 'Ali', 'Can', '05445551234', 3),
('Ev Adresi', 'Yeni Mahalle No:3', 'Adana', 'Seyhan', 'Çukurova',
'Personal', 'Fatma', 'Demir', '05559876543', 4),
('Ev Adresi', 'Selçuklu Mah. Osman Gazi Cad. No:7', 'Konya', 'Selçuklu',
'Merkez', 'Personal', 'Hakan', 'Öz', '05667777888', 5),
('Depo Adresi', 'Sanayi Sitesi No:15', 'Gaziantep', 'Şahinbey', 'Organize
Sanayi', 'Corporate', 'Hakan', 'Öz', '05667777888', 5),
('Ev Adresi', 'Mimar Sinan Cad. No:88', 'Samsun', 'İlkadım', 'Atakum',
'Personal', 'Zeynep', 'Turan', '05334445566', 6),
('Ev Adresi', 'Hacı Bayram Sokak No:9', 'Eskişehir', 'Tepebaşı',
'Şirintepe', 'Personal', 'Mustafa', 'Doğan', '05332221100', 7),
('Ofis Adresi', 'Kurtuluş Mah. No:45', 'İzmir', 'Karşıyaka', 'Yalı',
'Corporate', 'Mustafa', 'Doğan', '05332221100', 7),
('Kargo Adresi', 'Demokrasi Sokak No:30', 'Muğla', 'Fethiye', 'Ölüdeniz',
'Personal', 'Mustafa', 'Doğan', '05332221100', 7),
('Ev Adresi', 'Atatürk Mah. İstiklal Sokak No:20', 'Mersin', 'Mezitli',
'Akdeniz', 'Personal', 'Elif', 'Taş', '05539988877', 8),
('Fatura Adresi', 'Çakmak Sokak No:77', 'Kocaeli', 'Gebze', 'Darıca',
'Corporate', 'Elif', 'Taş', '05539988877', 8),
('Ev Adresi', 'Sevgi Mah. Çamlık Cad. No:44', 'Trabzon', 'Ortahisar',
'Boztepe', 'Personal', 'Kemal', 'Ak', '05661122334', 9),
('Ev Adresi', 'Deniz Mah. Sahil Sokak No:10', 'Balıkesir', 'Edremit',
'Altınoluk', 'Personal', 'Derya', 'Uzun', '05445566778', 10),
('Ofis Adresi', 'Şehir Merkezi Mah. İş Bankası Sokak No:2', 'Kayseri',
'Kocasinan', 'Talas', 'Corporate', 'Derya', 'Uzun', '05445566778', 10),
('Ev Adresi', 'Şehitler Cad. No:23', 'Ankara', 'Etimesgut', 'Elvankent',
'Personal', 'Selin', 'Güneş', '05449933445', 11),
('Ofis Adresi', 'Cinnah Cad. No:19', 'Ankara', 'Çankaya', 'Gaziosmanpaşa',
'Corporate', 'Selin', 'Güneş', '05449933445', 11),
('Ev Adresi', 'Mevlana Sokak No:50', 'İstanbul', 'Pendik', 'Kaynarca',
'Personal', 'Orhan', 'Çelik', '05331129988', 12),

```

('Ev Adresi', 'Çiçek Cad. No:12', 'İzmir', 'Bayraklı', 'Bornova',  
 'Personal', 'Deniz', 'Akyüz', '05338889900', 13),  
 ('Yazlık', 'Sahil Cad. No:5', 'Muğla', 'Bodrum', 'Göltürkbükü',  
 'Personal', 'Deniz', 'Akyüz', '05338889900', 13),  
 ('Ofis Adresi', 'Çamlık Sokak No:40', 'Denizli', 'Pamukkale',  
 'Merkezefendi', 'Corporate', 'Deniz', 'Akyüz', '05338889900', 13),  
 ('Ev Adresi', 'Demirci Sokak No:8', 'Antalya', 'Kepez', 'Altınova',  
 'Personal', 'Ahmet', 'Kılıç', '05341145577', 14),  
 ('Ev Adresi', 'Zambak Mah. Yasemin Cad. No:22', 'Bursa', 'Yıldırım',  
 'Bağlarbaşı', 'Personal', 'Barış', 'Turan', '05678887766', 15),  
 ('Depo Adresi', 'Organize Sanayi No:7', 'Bursa', 'Osmangazi', 'Demirtaş',  
 'Corporate', 'Barış', 'Turan', '05678887766', 15),  
 ('Ev Adresi', 'Gümüş Sokak No:99', 'Samsun', 'Atakum', 'Çobanlı',  
 'Personal', 'Melisa', 'Korkmaz', '05446678899', 16),  
 ('Ev Adresi', 'Bağdat Cad. No:14', 'İstanbul', 'Kadıköy', 'Göztepe',  
 'Personal', 'İpek', 'Can', '05347778899', 17),  
 ('Fatura Adresi', 'Ceviz Sokak No:5', 'Tekirdağ', 'Çorlu', 'Hürriyet',  
 'Corporate', 'İpek', 'Can', '05347778899', 17),  
 ('Ev Adresi', 'Adnan Menderes Cad. No:10', 'Adana', 'Seyhan', 'Barajyolu',  
 'Personal', 'Kemal', 'Ulus', '05449977666', 18),  
 ('Ofis Adresi', 'Çınar Sokak No:1', 'Hatay', 'Antakya', 'Kurtuluş',  
 'Corporate', 'Kemal', 'Ulus', '05449977666', 18),  
 ('Kargo Adresi', 'Limonluk Sokak No:77', 'Mersin', 'Yenişehir',  
 'Portakal', 'Personal', 'Kemal', 'Ulus', '05449977666', 18),  
 ('Ev Adresi', 'Gül Sokak No:55', 'Eskişehir', 'Odunpazarı', 'Vişnelik',  
 'Personal', 'Cansu', 'Yıldız', '05331112244', 19),  
 ('Ev Adresi', 'İstiklal Mah. Çamlık Cad. No:23', 'Trabzon', 'Ortahisar',  
 'Çukurçayır', 'Personal', 'Burak', 'Gök', '05557778822', 20),  
 ('Yazlık', 'Deniz Sokak No:9', 'Sinop', 'Merkez', 'Karşıyaka', 'Personal',  
 'Burak', 'Gök', '05557778822', 20);

INSERT INTO PaymentDetail (CardNumber, CardExpiryDate, CardholderName,  
 CVV, UserID)

VALUES

('4321432143214321', '2026-05-01', 'Mehmet Kaya', '123', 1),  
 ('5678567856785678', '2027-11-01', 'Mehmet Kaya', '456', 1),  
 ('9876543298765432', '2025-06-01', 'Ayşe Yılmaz', '789', 2),  
 ('8765876587658765', '2026-03-01', 'Ali Can', '654', 3),  
 ('3456345634563456', '2025-12-01', 'Ali Can', '321', 3),  
 ('1234123412341234', '2026-07-01', 'Ali Can', '987', 3),  
 ('4567456745674567', '2025-09-01', 'Fatma Demir', '111', 4),  
 ('6667666766676667', '2026-10-01', 'Hakan Öz', '222', 5),  
 ('7778777877787778', '2025-01-01', 'Hakan Öz', '333', 5),  
 ('8889888988898889', '2026-03-01', 'Zeynep Turan', '444', 6),  
 ('1111222233334444', '2027-02-01', 'Mustafa Doğan', '555', 7),  
 ('2222333344445555', '2025-04-01', 'Mustafa Doğan', '666', 7),  
 ('3333444455556666', '2026-09-01', 'Mustafa Doğan', '777', 7),  
 ('4444555566667777', '2026-06-01', 'Elif Taş', '888', 8),  
 ('5555666677778888', '2027-08-01', 'Elif Taş', '999', 8),  
 ('6666777788889999', '2025-12-01', 'Kemal Ak', '123', 9),  
 ('7777888899990000', '2026-11-01', 'Derya Uzun', '456', 10),  
 ('8888999900001111', '2027-03-01', 'Derya Uzun', '789', 10),  
 ('9999000011112222', '2026-05-01', 'Selin Güneş', '321', 11),  
 ('0000111122223333', '2025-10-01', 'Orhan Çelik', '654', 12),

```
('1111222233334445', '2026-07-01', 'Orhan Çelik', '987', 12),
('2222333344445556', '2025-06-01', 'Deniz Akyüz', '111', 13),
('3333444455556667', '2026-01-01', 'Deniz Akyüz', '222', 13),
('4444555566667778', '2027-12-01', 'Deniz Akyüz', '333', 13),
('5555666677778889', '2025-04-01', 'Ahmet Kılıç', '444', 14),
('6666777788889900', '2026-03-01', 'Barış Turan', '555', 15),
('7777888899990001', '2027-02-01', 'Barış Turan', '666', 15),
('8888999900001112', '2026-11-01', 'Melisa Korkmaz', '777', 16),
('9999000011112223', '2025-05-01', 'İpek Can', '888', 17),
('0000111122223334', '2026-08-01', 'İpek Can', '999', 17),
('1111222233334446', '2026-02-01', 'Kemal Ulus', '123', 18),
('2222333344445557', '2025-09-01', 'Kemal Ulus', '456', 18),
('3333444455556668', '2027-07-01', 'Kemal Ulus', '789', 18),
('4444555566667779', '2026-06-01', 'Cansu Yıldız', '321', 19),
('5555666677778880', '2025-10-01', 'Burak Gök', '654', 20),
('6666777788889901', '2026-12-01', 'Burak Gök', '987', 20);
```

```
( '2023-01-01', '2023-12-31', '2023-12-01', 1200.00, 1),
( '2023-09-01', NULL, '2023-10-01', 100.00, 2),
( '2022-06-01', '2023-05-31', '2023-05-01', 1200.00, 3),
( '2023-03-01', NULL, '2023-04-01', 100.00, 4),
( '2023-07-01', '2024-06-30', '2024-06-01', 1200.00, 5),
( '2023-11-01', NULL, '2023-12-01', 100.00, 6),
( '2022-01-01', '2022-12-31', '2022-12-01', 1200.00, 7),
( '2023-10-01', NULL, '2023-11-01', 100.00, 8),
( '2023-05-01', '2024-04-30', '2024-04-01', 1200.00, 10),
( '2023-08-01', NULL, '2023-09-01', 100.00, 12),
( '2023-01-01', '2023-12-31', '2023-12-01', 1200.00, 13),
( '2023-06-01', NULL, '2023-07-01', 100.00, 14),
( '2022-07-01', '2023-06-30', '2023-06-01', 1200.00, 15),
( '2023-12-01', NULL, '2024-01-01', 100.00, 17),
( '2023-09-01', '2024-08-31', '2024-08-01', 1200.00, 18);
```

[illegible]

```
(DEFAULT),
(DEFAULT),
(DEFAULT),
(DEFAULT),
(DEFAULT),
(DEFAULT),
(DEFAULT),
(DEFAULT);
```

```
INSERT INTO Customer (CustomerID, FirstName, LastName, ShoppingCartID)
VALUES
```

```
(1, 'Mehmet', 'Kaya', 1),
(2, 'Ayşe', 'Yılmaz', 2),
(3, 'Ali', 'Can', 3),
(4, 'Fatma', 'Demir', 4),
(5, 'Hakan', 'Öz', 5),
(6, 'Zeynep', 'Turan', 6),
(7, 'Mustafa', 'Doğan', 7),
(8, 'Elif', 'Taş', 8),
(9, 'Kemal', 'Ak', 9),
(10, 'Derya', 'Uzun', 10),
(11, 'Selin', 'Güneş', 11),
(12, 'Orhan', 'Çelik', 12),
(13, 'Deniz', 'Akyüz', 13),
(14, 'Ahmet', 'Kılıç', 14),
(15, 'Barış', 'Turan', 15),
(16, 'Melisa', 'Korkmaz', 16),
(17, 'İpek', 'Can', 17),
(18, 'Kemal', 'Ulus', 18),
(19, 'Cansu', 'Yıldız', 19),
(20, 'Burak', 'Gök', 20),
(21, 'Eda', 'Arslan', 21),
(22, 'Emre', 'Akyüz', 22),
(23, 'Aylin', 'Dikmen', 23),
(24, 'Burak', 'Yıldırım', 24),
(25, 'Yasemin', 'Deniz', 25);
```

```
INSERT INTO Seller (SellerID, FirstName, LastName, StoreName, TaxID,
StoreRating, SellerType, StoreDate, TotalSellNumber, IbanNo,
SalesCategoryType)
```

```
VALUES
(20, 'Burak', 'Gök', 'Gök Antiques', '12345678901', 4.5, 'Individual',
'2022-03-01', 150, 'TR200006789012345678901234', 'Antiques'),
(21, 'Eda', 'Arslan', 'Eda Handmade', '23456789012', 4.2, 'Individual',
'2021-05-15', 200, 'TR210007890123456789012345', 'Handmade Items'),
(22, 'Emre', 'Akyüz', 'Akyüz Art', '34567890123', 4.8, 'Individual',
'2023-01-10', 180, 'TR220008901234567890123456', 'Arts & Crafts'),
(23, 'Aylin', 'Dikmen', 'Dikmen Tech', '45678901234', 4.7, 'Individual',
'2022-07-20', 250, 'TR230009012345678901234567', 'Technology'),
(24, 'Burak', 'Yıldırım', 'Yıldırım Decor', '56789012345', 4.9,
'Individual', '2021-11-05', 300, 'TR240001234567890123456789', 'Home
Decor'),
```

```
(25, 'Yasemin', 'Deniz', 'Deniz Furniture', '12345678901', 4.5,
'Corporate', '2021-12-01', 500, 'TR250006789012345678901234',
'Furniture'),
(26, 'Ali', 'Şahin', 'Şahin Electronics', '23456789012', 4.3, 'Corporate',
'2020-06-15', 750, 'TR260007890123456789012345', 'Electronics'),
(27, 'Zehra', 'Yıldız', 'Yıldız Groceries', '34567890123', 4.6,
'Corporate', '2019-03-20', 1000, 'TR270008901234567890123456',
'Groceries'),
(28, 'Mert', 'Çelik', 'Çelik Tech', '45678901234', 4.8, 'Corporate',
'2022-01-10', 400, 'TR280009012345678901234567', 'Technology'),
(29, 'Ece', 'Demir', 'Demir Art', '56789012345', 4.7, 'Corporate', '2023-
04-25', 300, 'TR290001234567890123456789', 'Arts & Crafts'),
(30, 'Can', 'Kurt', 'Kurt Supplies', '67890123456', 4.9, 'Corporate',
'2021-08-05', 200, 'TR300002345678901234567890', 'Office Supplies');
```

```
INSERT INTO UserLastLogin (UserID, LastLogin)
VALUES
```

```
(1, '2025-01-01 09:00:00'),
(1, '2025-01-02 10:15:00'),
(2, '2025-01-01 09:30:00'),
(2, '2025-01-03 11:00:00'),
(3, '2025-01-02 14:00:00'),
(3, '2025-01-05 16:20:00'),
(4, '2025-01-03 11:45:00'),
(4, '2025-01-06 09:15:00'),
(5, '2025-01-03 12:30:00'),
(5, '2025-01-07 14:00:00'),
(6, '2025-01-04 13:00:00'),
(6, '2025-01-08 17:45:00'),
(7, '2025-01-04 13:30:00'),
(8, '2025-01-05 14:15:00'),
(9, '2025-01-05 15:00:00'),
(9, '2025-01-10 18:45:00'),
(10, '2025-01-06 15:30:00'),
(10, '2025-01-12 08:00:00'),
(11, '2025-01-06 16:15:00'),
(12, '2025-01-07 17:00:00'),
(13, '2025-01-07 18:30:00'),
(13, '2025-01-13 12:00:00'),
(14, '2025-01-08 18:00:00'),
(15, '2025-01-08 18:30:00'),
(16, '2025-01-09 19:00:00'),
(17, '2025-01-09 19:15:00'),
(18, '2025-01-10 20:00:00'),
(19, '2025-01-10 20:15:00'),
(20, '2025-01-11 14:45:00'),
(20, '2025-01-12 10:15:00'),
(21, '2025-01-12 11:00:00'),
(21, '2025-01-13 08:30:00'),
(22, '2025-01-13 09:45:00'),
(23, '2025-01-14 10:00:00'),
(23, '2025-01-14 12:00:00'),
(24, '2025-01-14 13:30:00'),
```

```
(24, '2025-01-15 14:15:00'),
(25, '2025-01-15 15:45:00'),
(26, '2025-01-16 16:00:00'),
(26, '2025-01-16 17:30:00'),
(27, '2025-01-16 18:00:00'),
(28, '2025-01-17 09:00:00'),
(28, '2025-01-17 10:45:00'),
(29, '2025-01-17 11:30:00'),
(30, '2025-01-17 12:15:00'),
(30, '2025-01-18 13:00:00');
```

```
INSERT INTO Inventory (InventoryID, AdressLine, InventoryName, SellerID)
VALUES
```

```
(101, 'Atatürk Cad. No:10', 'Deniz Warehouse 1', 25),
(102, 'Cumhuriyet Cad. No:20', 'Deniz Warehouse 2', 25),
(103, 'Sanayi Mah. No:30', 'Şahin Electronics Depot 1', 26),
(104, 'Teknoloji Cad. No:40', 'Şahin Electronics Depot 2', 26),
(105, 'Elektrik Sokak No:50', 'Şahin Electronics Depot 3', 26),
(106, 'Pazar Yolu No:1', 'Yıldız Depot 1', 27),
(107, 'Gıda Sokak No:2', 'Yıldız Depot 2', 27),
(108, 'Market Cad. No:3', 'Yıldız Depot 3', 27),
(109, 'Alışveriş Mah. No:4', 'Yıldız Depot 4', 27),
(110, 'Teknoloji Parkı No:100', 'Çelik Tech Depot 1', 28),
(111, 'Yazılım Sokak No:101', 'Çelik Tech Depot 2', 28),
(112, 'Donanım Mah. No:102', 'Çelik Tech Depot 3', 28),
(113, 'Elektronik Cad. No:103', 'Çelik Tech Depot 4', 28),
(114, 'Robotik Sokak No:104', 'Çelik Tech Depot 5', 28),
(115, 'Sanat Cad. No:1', 'Demir Art Depot', 29),
(116, 'Ofis Sokak No:10', 'Kurt Supplies Depot 1', 30),
(117, 'İş Mah. No:11', 'Kurt Supplies Depot 2', 30),
(118, 'Kırtasiye Cad. No:12', 'Kurt Supplies Depot 3', 30),
(119, 'El İşleri Cad. No:1', 'Eda Handmade Depot', 21),
(120, 'Sanatçılar Sokak No:1', 'Akyüz Art Depot', 22);
```

```
INSERT INTO Stock (InventoryID, Barcode, Quantity, ListPrice)
VALUES
```

```
(101, '1234500000001', 50, 1500.00),
(101, '1234500000002', 30, 2000.00),
(102, '1234500000001', 20, 1500.00),
(102, '1234500000002', 15, 2000.00),
(103, '5432100000001', 20, 500.00),
(103, '5432100000002', 15, 800.00),
(104, '6789100000003', 25, 1200.00),
(104, '5432100000001', 10, 500.00),
(105, '6789100000004', 10, 1500.00),
(105, '5432100000002', 5, 800.00),
(106, '1111100000001', 35, 50.00),
(107, '1111100000001', 40, 50.00),
(108, '1111100000002', 20, 100.00),
(109, '1111100000003', 25, 200.00),
(110, '6789100000003', 50, 1250.00),
(111, '6789100000004', 60, 1550.00),
(112, '6789100000003', 30, 1250.00),
(113, '6789100000006', 25, 4000.00),
```

(114, '67891000000007', 40, 4500.00),  
(115, '99999000000001', 10, 5000.00),  
(116, '12345000000001', 20, 1600.00),  
(116, '12345000000002', 15, 2100.00),  
(117, '12345000000002', 25, 2100.00),  
(118, '22222000000002', 30, 200.00),  
(119, '88888000000001', 5, 1500.00),  
(120, '77777000000001', 8, 1600.00);

INSERT INTO Product (ProductID, ProductName, Information, Brand, Price, Properties, SellerID, ShippingCost)

VALUES

(1, 'Ahşap Masa', 'Yüksek kaliteli, dayanıklı ahşap masa', 'Furniture Co.', 1500.00, 'Malzeme: Ahşap, Renk: Ceviz', 25, 40),  
(2, 'Modern Sandalye', 'Ergonomik, şık tasarımlı sandalye', 'Furniture Co.', 2000.00, 'Malzeme: Metal, Kumaş, Renk: Gri', 25, 40),  
(3, 'Oturma Grubu', 'Konforlu oturma odası seti', 'Furniture Co.', 4500.00, '3 Parça: Koltuk, Kanepe, Berjer', 25, 40),  
(4, 'Akıllı Telefon', '6.5 inç ekran, güçlü performans', 'TechBrand', 500.00, 'Ekran: 6.5 inç, Depolama: 128GB, Kamera: 48MP', 26, 50),  
(5, 'Tablet', '10 inç ekran, taşınabilir tasarım', 'TechBrand', 800.00, 'Ekran: 10 inç, Depolama: 64GB, Batarya: 7000mAh', 26, 50),  
(6, 'Kablosuz Kulaklık', 'Gelişmiş Bluetooth teknolojisi', 'TechBrand', 1200.00, 'Bluetooth 5.0, Pil Ömrü: 20 Saat', 26, 50),  
(7, 'Masaüstü Bilgisayar', '16GB RAM, güçlü işlemci', 'TechBrand', 3500.00, 'İşlemci: i7, Depolama: 512GB SSD, Ekran Kartı: GTX 1650', 26, 50),  
(8, 'Organik Elma', 'Doğal ortamda yetişmiş taze elmalar', 'FreshFarm', 50.00, 'Tür: Starking, Menşei: Türkiye', 27, 50),  
(9, 'Taze Süt', 'Günlük süt, taptaze lezzet', 'FreshFarm', 100.00, 'Yağ Oranı: %3, Paket: 1L', 27, 50),  
(10, 'Zeytinyağı', 'Sızma zeytinyağı, lezzetli ve doğal', 'FreshFarm', 200.00, 'Asit Oranı: %0.8, Paket: 1L', 27, 50),  
(11, 'Çikolatalı Kurabiye', 'El yapımı çikolatalı atıştırmalık', 'FreshFarm', 150.00, 'Paket: 500g, İçerik: Gerçek Çikolata', 27, 50),  
(12, 'Kablosuz Kulaklık', 'Yüksek kaliteli Bluetooth ses deneyimi', 'TechBrand', 1250.00, 'Bluetooth 5.0, Pil Ömrü: 20 Saat', 28, 40),  
(13, 'Oyuncu Monitörü', '144Hz yenileme hızına sahip ekran', 'GamingWorld', 4500.00, 'Ekran: 27 inç, Çözünürlük: 2560x1440, Yenileme Hızı: 144Hz', 28, 40),  
(14, 'Gaming Klavye', 'RGB ışıklı mekanik klavye', 'GamingWorld', 4000.00, 'Tür: Mekanik, Tuşlar: 104, Aydınlatma: RGB', 28, 40),  
(15, 'Gaming Mouse', 'Hassas kontrol için ergonomik fare', 'GamingWorld', 1500.00, 'DPI: 16000, Aydınlatma: RGB', 28, 40),  
(16, 'Sanat Tablosu', 'Özgün ve etkileyici bir yağlı boya eser', 'ArtMaster', 5000.00, 'Boyut: 50x70 cm, Malzeme: Tuval, Teknik: Yağlı Boya', 29, 40),  
(17, 'Ahşap Masa', 'Şık tasarımlı, dayanıklı ahşap masa', 'Furniture Co.', 1600.00, 'Malzeme: Ahşap, Renk: Meşe', 30, 35),  
(18, 'Modern Sandalye', 'Konforlu ve sağlam sandalye', 'Furniture Co.', 2100.00, 'Malzeme: Metal, Plastik, Renk: Siyah', 30, 35),  
(19, 'Ofis Malzemesi Seti', 'Kalem, zımba ve dosya içerir', 'OfficePro', 500.00, 'Set: 12 Kalem, 1 Zımba, 10 Dosya', 30, 35),



```
(20, 'El Yapımı Çanta', 'Şık ve kullanışlı örgü çanta', 'HandMadeCo',  
1500.00, 'Malzeme: Pamuk, Renk: Krem', 21, 35),  
(21, 'Seramik Vazo', 'Zarif ve dayanıklı seramik vazo', 'ArtMaster',  
1600.00, 'Boyut: 30x15 cm, Malzeme: Seramik', 22, 35),  
(22, 'Abajur', 'Dekoratif ve işlevsel abajur', 'ArtMaster', 2500.00,  
'Yükseklik: 40 cm, Malzeme: Seramik, Ampul: LED', 22, 35);
```

```
INSERT INTO ProductStock (ProductID, Barcode, InventoryID)  
VALUES
```

```
(1, '12345000000001', 101),  
(1, '12345000000001', 102),  
(2, '12345000000002', 101),  
(2, '12345000000002', 102),  
(4, '54321000000001', 103),  
(4, '54321000000001', 104),  
(5, '54321000000002', 103),  
(5, '54321000000002', 105),  
(6, '67891000000003', 104),  
(8, '11111000000001', 106),  
(8, '11111000000001', 107),  
(9, '11111000000002', 108),  
(10, '11111000000003', 109),  
(12, '67891000000003', 110),  
(12, '67891000000003', 112),  
(13, '67891000000004', 111),  
(14, '67891000000006', 113),  
(15, '67891000000007', 114),  
(16, '99999000000001', 115),  
(17, '12345000000001', 116),  
(18, '12345000000002', 116),  
(18, '12345000000002', 117),  
(19, '22222000000002', 118),  
(20, '88888000000001', 119),  
(21, '77777000000001', 120);
```

```
INSERT INTO Category (CategoryID, CategoryName, ParentCategoryID)  
VALUES
```

```
(1, 'Mobilya', NULL),  
(2, 'Elektronik', NULL),  
(3, 'Gıda', NULL),  
(4, 'Teknoloji', NULL),  
(5, 'Sanat ve Dekorasyon', NULL),  
(6, 'Oturma Odası Mobilyaları', 1),  
(7, 'Yatak Odası Mobilyaları', 1),  
(8, 'Ofis Mobilyaları', 1),  
(9, 'Telefonlar', 2),  
(10, 'Bilgisayarlar', 2),  
(11, 'Ev Elektroniği', 2),  
(12, 'Meyve ve Sebzeler', 3),  
(13, 'Süt ve Süt Ürünleri', 3),
```

```
(14, 'Kuru Gıda', 3),
(15, 'Bilgisayar Aksesuarları', 4),
(16, 'Oyun Ekipmanları', 4),
(17, 'Robotik ve Yapay Zeka', 4),
(18, 'Sanat Eserleri', 5),
(19, 'Dekoratif Ürünler', 5),
(20, 'El Yapımı Ürünler', 5);
```

```
INSERT INTO CategoryProduct (CategoryID, ProductID)
VALUES
```

```
(6, 1),
(8, 1),
(7, 2),
(8, 2),
(6, 17),
(9, 4),
(10, 5),
(10, 6),
(16, 6),
(12, 8),
(13, 9),
(15, 14),
(16, 14),
(15, 15),
(16, 15),
(15, 12),
(18, 16),
(19, 22),
(20, 20);
```

```
INSERT INTO Follow (SellerID, CustomerID)
VALUES
```

```
(25, 1),
(25, 3),
(25, 5),
(25, 6),
(25, 10),
(25, 12),
(26, 2),
(26, 6),
(26, 7),
(26, 3),
(26, 11),
(26, 13),
(27, 8),
(27, 9),
(27, 1),
(27, 10),
(27, 7),
(27, 14),
(28, 11),
(28, 12),
(28, 6),
(28, 13),
```

```
(28, 2),
(28, 15),
(29, 14),
(29, 15),
(29, 16),
(29, 11),
(29, 20),
(29, 19),
(30, 17),
(30, 18),
(30, 19),
(30, 1),
(21, 20),
(21, 8),
(21, 16),
(22, 14);
```

```
INSERT INTO Collection (CollectionName, CustomerID)
VALUES
```

```
('Modern Mobilya Koleksiyonu', 1),
('Teknolojik Ürünler Koleksiyonu', 2),
('Ofis Mobilyaları Koleksiyonu', 3),
('Teknoloji ve Elektronik Koleksiyonu', 3),
('Yatak Odası Mobilyaları Koleksiyonu', 5),
('Oyun Aksesuarları Koleksiyonu', 6),
('Kablosuz Cihazlar Koleksiyonu', 6),
('Sanat ve Dekorasyon Koleksiyonu', 8),
('Taze Gıda Koleksiyonu', 8),
('Kuru Gıda Koleksiyonu', 10),
('Gaming Ürünleri Koleksiyonu', 11),
('Bilgisayar Aksesuarları Koleksiyonu', 11),
('Elektronik Aletler Koleksiyonu', 12),
('Profesyonel Monitör Koleksiyonu', 13),
('Sanat Eserleri Koleksiyonu', 14),
('Dekoratif Ürünler Koleksiyonu', 15),
('El Yapımı Ürünler Koleksiyonu', 16),
('Modern Dekorasyon Koleksiyonu', 18),
('Gıda ve İçecek Koleksiyonu', 19),
('Ahşap Ürünler Koleksiyonu', 20),
('Ahşap ve El Yapımı Ürünler Koleksiyonu', 20),
('Ev Dekorasyonu Koleksiyonu', 24),
('Mobilya ve Aksesuar Koleksiyonu', 24);
```

```
INSERT INTO CollectionHas (CollectionName, CustomerID, ProductID)
VALUES
```

```
('Modern Mobilya Koleksiyonu', 1, 1),
('Modern Mobilya Koleksiyonu', 1, 2),
('Teknolojik Ürünler Koleksiyonu', 2, 4),
('Teknolojik Ürünler Koleksiyonu', 2, 5),
('Ofis Mobilyaları Koleksiyonu', 3, 17),
('Ofis Mobilyaları Koleksiyonu', 3, 18),
('Teknoloji ve Elektronik Koleksiyonu', 3, 5),
('Teknoloji ve Elektronik Koleksiyonu', 3, 6),
('Yatak Odası Mobilyaları Koleksiyonu', 5, 2),
```

```

('Yatak Odası Mobilyaları Koleksiyonu', 5, 3),
('Oyun Aksesuarları Koleksiyonu', 6, 14),
('Oyun Aksesuarları Koleksiyonu', 6, 15),
('Kablosuz Cihazlar Koleksiyonu', 6, 6),
('Sanat ve Dekorasyon Koleksiyonu', 8, 16),
('Sanat ve Dekorasyon Koleksiyonu', 8, 22),
('Taze Gıda Koleksiyonu', 8, 8),
('Taze Gıda Koleksiyonu', 8, 9),
('Kuru Gıda Koleksiyonu', 10, 10),
('Kuru Gıda Koleksiyonu', 10, 13),
('Gaming Ürünleri Koleksiyonu', 11, 14),
('Gaming Ürünleri Koleksiyonu', 11, 15),
('Bilgisayar Aksesuarları Koleksiyonu', 11, 13),
('Ahşap Ürünler Koleksiyonu', 20, 1),
('Ahşap Ürünler Koleksiyonu', 20, 17),
('Ahşap ve El Yapımı Ürünler Koleksiyonu', 20, 20),
('Ahşap ve El Yapımı Ürünler Koleksiyonu', 20, 22),
('Ev Dekorasyonu Koleksiyonu', 24, 16),
('Ev Dekorasyonu Koleksiyonu', 24, 22),
('Mobilya ve Aksesuar Koleksiyonu', 24, 3),
('Mobilya ve Aksesuar Koleksiyonu', 24, 18);

```

```

INSERT INTO CartItems (ProductID, ShoppingCartID, Quantity)
VALUES

```

```

(1, 1, 2),
(2, 1, 4),
(4, 2, 1),
(5, 2, 2),
(17, 3, 1),
(18, 3, 2),
(6, 3, 3),
(3, 5, 1),
(14, 6, 1),
(15, 6, 2),
(6, 6, 1),
(16, 8, 1),
(8, 8, 6),
(9, 8, 3),
(10, 10, 2),
(13, 10, 4),
(14, 11, 1),
(13, 11, 1),
(14, 13, 2),
(15, 13, 1),
(1, 20, 2),
(20, 20, 1),
(22, 20, 1),
(3, 24, 1),
(16, 24, 2);

```

```

INSERT INTO PromotionCode (PromotionCodeID, StartDate, EndDate, CodeType,
Information, CustomerID)
VALUES

```

```

(1, '2025-01-01', '2025-03-01', 'Coupon', 'İlk alışverişte %10 indirim',
1),
(2, '2025-02-01', '2025-04-01', 'GiftCard', '500 TL değerinde hediye
çeki', 1),
(3, '2025-01-10', '2025-03-10', 'Coupon', 'Elektronik ürünlerde %15
indirim', 2),
(4, '2025-01-15', '2025-03-15', 'Coupon', 'Ofis mobilyalarında %20
indirim', 3),
(5, '2025-02-01', '2025-04-01', 'GiftCard', '1000 TL değerinde hediye
çeki', 3),
(6, '2025-01-20', '2025-03-20', 'Coupon', 'Yatak odası mobilyalarında %10
indirim', 5),
(7, '2025-01-25', '2025-03-25', 'Coupon', 'Oyun ekipmanlarında %15
indirim', 6),
(8, '2025-02-05', '2025-04-05', 'GiftCard', '200 TL değerinde hediye
çeki', 6),
(9, '2025-01-30', '2025-03-30', 'Coupon', 'Sanat eserlerinde %10 indirim',
8),
(10, '2025-02-10', '2025-04-10', 'GiftCard', '500 TL değerinde hediye
çeki', 8),
(11, '2025-02-01', '2025-04-01', 'Coupon', 'Kuru gıda ürünlerinde %5
indirim', 10),
(12, '2025-02-15', '2025-04-15', 'Coupon', 'Gaming klavyelerde %20
indirim', 11),
(13, '2025-02-20', '2025-04-20', 'Coupon', 'Profesyonel monitörlerde %15
indirim', 13),
(14, '2025-03-01', '2025-05-01', 'Coupon', 'Ahşap ürünlerde %10 indirim',
20),
(15, '2025-03-10', '2025-05-10', 'GiftCard', '300 TL değerinde hediye
çeki', 20),
(16, '2025-03-15', '2025-05-15', 'Coupon', 'Ev dekorasyon ürünlerinde %15
indirim', 24),
(17, '2025-03-20', '2025-05-20', 'GiftCard', '1000 TL değerinde hediye
çeki', 24);

```

```

INSERT INTO Payment (PaymentAmount, PaymentMethod, CardNumber,
CardExpiryDate, CardholderName, CVV, PaymentDetailID, CustomerID)
VALUES
(1450.00, 'CreditCard', NULL, NULL, NULL, NULL, 1, 1),
(1900.00, 'DebitCard', NULL, NULL, NULL, NULL, 2, 1),
(500.00, 'CreditCard', NULL, NULL, NULL, NULL, 3, 2),
(1000.00, 'DebitCard', NULL, NULL, NULL, NULL, NULL, 2),
(800.00, 'CreditCard', '8765876587658765', '2026-03-01', 'Ali Can', '654',
NULL, 3),
(1200.00, 'CreditCard', '3456345634563456', '2025-12-01', 'Ali Can',
'321', NULL, 3),
(3500.00, 'CreditCard', '1234123412341234', '2026-07-01', 'Ali Can',
'987', NULL, 3),
(1500.00, 'CreditCard', NULL, NULL, NULL, NULL, 9, 6),
(2000.00, 'DebitCard', NULL, NULL, NULL, NULL, NULL, 6),
(180.00, 'DebitCard', NULL, NULL, NULL, NULL, 10, 8),
(600.00, 'CreditCard', NULL, NULL, NULL, NULL, 11, 8),
(300.00, 'CreditCard', '7777888899990000', '2026-11-01', 'Derya Uzun',
'456', NULL, 10),

```

```
(1800.00, 'DebitCard', '8888999900001111', '2027-03-01', 'Derya Uzun',
'789', NULL, 10),
(2500.00, 'CreditCard', NULL, NULL, NULL, NULL, 14, 11),
(3000.00, 'CreditCard', NULL, NULL, NULL, NULL, 15, 13),
(200.00, 'DebitCard', NULL, NULL, NULL, NULL, 16, 13),
(400.00, 'CreditCard', '5555666677778880', '2025-10-01', 'Burak Gök',
'654', NULL, 20),
(1200.00, 'CreditCard', '6666777788889901', '2026-12-01', 'Burak Gök',
'987', NULL, 20);
```

```
INSERT INTO Logistic
```

```
VALUES
```

```
(DEFAULT),
(DEFAULT),
(DEFAULT),
(DEFAULT),
(DEFAULT),
(DEFAULT),
(DEFAULT),
(DEFAULT),
(DEFAULT),
(DEFAULT),
(DEFAULT),
(DEFAULT),
(DEFAULT),
(DEFAULT),
(DEFAULT),
(DEFAULT),
(DEFAULT),
(DEFAULT),
(DEFAULT);
```

```
INSERT INTO Shipment (TrackingNumber, PackageWeight, PackageDimension,
CarrierName, DeliveryStatus, DeliveryDate, EstimatedDeliveryDate,
ShipmentDate, LogisticID)
```

```
VALUES
```

```
('TRK123456789', 2.5, '30x20x15', 'Aras Kargo', 'Delivered', '2025-01-01',
'2024-12-30', '2024-12-28', 1),
('TRK123456790', 1.5, '20x10x10', 'Yurtiçi Kargo', 'In Transit', NULL,
'2025-01-03', '2025-01-01', 1),
('TRK987654321', 1.2, '20x15x10', 'PTT Kargo', 'Delivered', '2025-01-02',
'2025-01-01', '2024-12-30', 2),
('TRK987654322', 3.0, '30x25x20', 'DHL', 'Out for Delivery', NULL, '2025-
01-05', '2025-01-02', 2),
('TRK112233445', 5.0, '50x40x30', 'MNG Kargo', 'Out for Delivery', NULL,
'2025-01-06', '2025-01-03', 3),
('TRK112233446', 3.5, '45x30x25', 'UPS', 'Delivered', '2025-01-05', '2025-
01-04', '2025-01-02', 3),
('TRK556677889', 3.8, '40x30x25', 'Sürat Kargo', 'Delivered', '2025-01-
03', '2025-01-02', '2024-12-31', 4),
('TRK556677890', 2.2, '35x25x15', 'Aras Kargo', 'In Transit', NULL, '2025-
01-07', '2025-01-04', 4),
('TRK998877665', 0.8, '15x10x5', 'Yurtiçi Kargo', 'In Transit', NULL,
'2025-01-08', '2025-01-04', 5),
```

```

('TRK998877666', 2.0, '25x15x10', 'DHL', 'Out for Delivery', NULL, '2025-01-10', '2025-01-06', 5),
('TRK667788990', 10.0, '60x50x40', 'PTT Kargo', 'Delivered', '2025-01-06', '2025-01-04', '2025-01-02', 6),
('TRK667788991', 7.5, '55x45x35', 'MNG Kargo', 'Out for Delivery', NULL, '2025-01-09', '2025-01-06', 6),
('TRK334455667', 7.2, '70x50x40', 'UPS', 'Delivered', '2025-01-07', '2025-01-05', '2025-01-03', 7),
('TRK334455668', 4.0, '45x35x25', 'DHL', 'In Transit', NULL, '2025-01-10', '2025-01-07', 7),
('TRK443322110', 4.5, '45x35x25', 'Sürat Kargo', 'In Transit', NULL, '2025-01-09', '2025-01-05', 8),
('TRK443322111', 2.3, '25x20x15', 'Aras Kargo', 'Delivered', '2025-01-08', '2025-01-06', '2025-01-04', 8),
('TRK998800112', 3.0, '30x30x20', 'Yurtiçi Kargo', 'Delivered', '2025-01-08', '2025-01-06', '2025-01-04', 9),
('TRK998800113', 1.8, '20x15x10', 'DHL', 'Out for Delivery', NULL, '2025-01-11', '2025-01-08', 9),
('TRK776655443', 2.1, '25x20x15', 'PTT Kargo', 'Out for Delivery', NULL, '2025-01-12', '2025-01-08', 10),
('TRK776655444', 4.0, '35x25x20', 'UPS', 'Delivered', '2025-01-11', '2025-01-09', '2025-01-07', 10),
('TRK112358132', 9.5, '80x70x60', 'Aras Kargo', 'Delivered', '2025-01-07', '2025-01-05', '2025-01-03', 11),
('TRK314159265', 0.7, '10x10x5', 'Sürat Kargo', 'Delivered', '2025-01-08', '2025-01-06', '2025-01-04', 12),
('TRK314159266', 1.0, '15x10x8', 'UPS', 'In Transit', NULL, '2025-01-09', '2025-01-07', 12),
('TRK123321456', 8.0, '60x40x30', 'DHL', 'In Transit', NULL, '2025-01-10', '2025-01-08', 13),
('TRK654987321', 6.3, '50x40x30', 'PTT Kargo', 'Delivered', '2025-01-09', '2025-01-07', '2025-01-05', 14),
('TRK654987322', 7.0, '70x50x40', 'MNG Kargo', 'Out for Delivery', NULL, '2025-01-11', '2025-01-09', 14),
('TRK445566778', 4.2, '40x30x20', 'Aras Kargo', 'Delivered', '2025-01-10', '2025-01-08', '2025-01-06', 15),
('TRK445566779', 3.0, '35x25x15', 'Yurtiçi Kargo', 'In Transit', NULL, '2025-01-12', '2025-01-10', 15),
('TRK223344556', 5.5, '55x45x35', 'PTT Kargo', 'Out for Delivery', NULL, '2025-01-13', '2025-01-11', 16),
('TRK998877112', 1.5, '20x15x10', 'DHL', 'Delivered', '2025-01-14', '2025-01-12', '2025-01-10', 17),
('TRK998877113', 2.0, '25x20x15', 'Aras Kargo', 'In Transit', NULL, '2025-01-15', '2025-01-13', 17),
('TRK445522113', 3.3, '35x25x20', 'UPS', 'Out for Delivery', NULL, '2025-01-16', '2025-01-14', 18),
('TRK445522114', 6.0, '50x40x30', 'MNG Kargo', 'Delivered', '2025-01-15', '2025-01-13', '2025-01-11', 18);

```

```

INSERT INTO `Order` (OrderNo, CustomerID, ShoppingCartID, OrderDate,
OrderStatus, TotalAmount, PaymentID, PaymentDate, PaymentStatus,
LogisticID, ShippingID)
VALUES

```

```

(1, 1, 1, '2025-01-01', 'Delivered', 1450.00, 1, '2025-01-01',
'Completed', 1, 1),
(2, 1, 1, '2025-01-02', 'Shipped', 1900.00, 2, '2025-01-02', 'Completed',
2, 2),
(3, 2, 2, '2025-01-03', 'Cancelled', 500.00, 3, NULL, 'Failed', 3, 3),
(4, 2, 2, '2025-01-04', 'Delivered', 1000.00, 4, '2025-01-04',
'Completed', 4, 3),
(5, 3, 3, '2025-01-05', 'Shipped', 800.00, 5, '2025-01-05', 'Completed',
5, 5),
(6, 3, 3, '2025-01-06', 'Delivered', 1200.00, 6, '2025-01-06',
'Completed', 6, 6),
(7, 3, 3, '2025-01-07', 'Processing', 3500.00, 7, '2025-01-07',
'Completed', 7, 4),
(8, 6, 6, '2025-01-08', 'Delivered', 1500.00, 8, '2025-01-08',
'Completed', 8, 10),
(9, 6, 6, '2025-01-09', 'Shipped', 2000.00, 9, '2025-01-09', 'Completed',
9, 10),
(10, 8, 8, '2025-01-10', 'Delivered', 180.00, 10, '2025-01-10',
'Completed', 10, 14),
(11, 8, 8, '2025-01-11', 'Processing', 600.00, 11, '2025-01-11',
'Completed', 11, 15),
(12, 10, 10, '2025-01-12', 'Shipped', 300.00, 12, '2025-01-12',
'Completed', 12, 17),
(13, 10, 10, '2025-01-13', 'Delivered', 1800.00, 13, '2025-01-13',
'Completed', 13, 18),
(14, 11, 11, '2025-01-14', 'Delivered', 2500.00, 14, '2025-01-14',
'Completed', 14, 19),
(15, 13, 13, '2025-01-15', 'Processing', 3000.00, 15, '2025-01-15',
'Completed', 15, 22),
(16, 13, 13, '2025-01-16', 'Shipped', 200.00, 16, '2025-01-16',
'Completed', 16, 23),
(17, 20, 20, '2025-01-17', 'Delivered', 400.00, 17, '2025-01-17',
'Completed', 17, 35),
(18, 20, 20, '2025-01-18', 'Delivered', 1200.00, 18, '2025-01-18',
'Completed', 18, 36);

```

```

INSERT INTO OrderItem (OrderNo, CustomerID, ShoppingCartID, ProductID,
OrderItemNo, Quantity, ItemStatus, TrackingNumber)
VALUES
(1, 1, 1, 1, 1, 2, 'Delivered', 'TRK123456789'),
(1, 1, 1, 2, 2, 4, 'Shipped', 'TRK123456790'),
(2, 1, 1, 4, 3, 1, 'Cancelled', NULL),
(2, 1, 1, 5, 4, 2, 'Delivered', 'TRK987654322'),
(3, 2, 2, 6, 5, 1, 'Shipped', 'TRK112233445'),
(3, 2, 2, 1, 6, 3, 'Delivered', 'TRK112233446'),
(4, 2, 2, 2, 7, 2, 'Processing', 'TRK556677889'),
(4, 2, 2, 3, 8, 1, 'Processing', 'TRK556677890'),
(5, 3, 3, 14, 9, 1, 'Delivered', 'TRK998877665'),
(5, 3, 3, 15, 10, 2, 'Shipped', 'TRK998877666'),
(6, 3, 3, 8, 11, 6, 'Delivered', 'TRK667788990'),
(6, 3, 3, 16, 12, 1, 'Delivered', 'TRK667788991'),

```



```
(7, 3, 3, 10, 13, 2, 'Delivered', 'TRK334455667'),
(7, 3, 3, 9, 14, 1, 'Delivered', 'TRK334455668'),
(8, 6, 6, 11, 15, 4, 'Processing', 'TRK443322110'),
(8, 6, 6, 13, 16, 1, 'Delivered', 'TRK443322111'),
(9, 6, 6, 14, 17, 1, 'Delivered', 'TRK998800112'),
(9, 6, 6, 15, 18, 1, 'Processing', 'TRK998800113'),
(10, 8, 8, 14, 19, 1, 'Shipped', 'TRK776655443'),
(10, 8, 8, 1, 20, 1, 'Delivered', 'TRK776655444'),
(11, 8, 8, 20, 21, 1, 'Delivered', 'TRK112358132'),
(12, 10, 10, 9, 22, 1, 'Processing', 'TRK314159265'),
(12, 10, 10, 10, 23, 1, 'Processing', 'TRK314159266'),
(13, 10, 10, 15, 24, 1, 'Processing', 'TRK123321456'),
(14, 11, 11, 1, 25, 1, 'Delivered', 'TRK654987321'),
(14, 11, 11, 2, 26, 2, 'Delivered', 'TRK654987322'),
(15, 13, 13, 20, 27, 1, 'Delivered', 'TRK445566778'),
(15, 13, 13, 2, 28, 2, 'Delivered', 'TRK445566779'),
(16, 13, 13, 10, 29, 1, 'Processing', 'TRK223344556'),
(17, 20, 20, 9, 30, 1, 'Processing', 'TRK998877112'),
(17, 20, 20, 10, 31, 1, 'Processing', 'TRK998877113'),
(18, 20, 20, 19, 32, 1, 'Delivered', 'TRK445522113'),
(18, 20, 20, 22, 33, 1, 'Delivered', 'TRK445522114');
```

```
INSERT INTO Review (OrderNo, CustomerID, ShoppingCartID, ProductID,
OrderItemNo, ReviewNo, ReviewDate, Rating, Comment)
VALUES
(1, 1, 1, 1, 1, 1, '2025-01-02', 5, 'Ürün gerçekten kaliteli ve
dayanıklı.'),
(1, 1, 1, 2, 2, 2, '2025-01-04', 4, 'Rahat ve şık bir tasarım.
Memnunum.'),
(2, 1, 1, 5, 4, 3, '2025-01-06', 5, 'Performansı harika, şarj süresi
uzun.'),
(3, 2, 2, 6, 5, 4, '2025-01-07', 5, 'Ses kalitesi çok iyi, pil ömrü tatmin
edici.'),
(3, 2, 2, 1, 6, 5, '2025-01-09', 5, 'Masa çok şık ve sağlam.
Teşekkürler.'),
(5, 3, 3, 14, 9, 6, '2025-01-11', 4, 'Tuş hissiyatı çok iyi, ancak ışık
seviyesi biraz düşük.'),
(5, 3, 3, 15, 10, 7, '2025-01-12', 5, 'Beklentimi fazlasıyla karşıladı.'),
(6, 3, 3, 8, 11, 8, '2025-01-13', 5, 'Elmalar taze ve çok lezzetli.'),
(8, 6, 6, 11, 15, 9, '2025-01-15', 5, 'Lezzetli ve taptaze geldi. Çocuklar
bayıldı.'),
(8, 6, 6, 13, 16, 10, '2025-01-16', 5, 'Renk kalitesi ve yenileme hızı
harika. Tavsiye ederim.'),
(10, 8, 8, 1, 20, 11, '2025-01-17', 5, 'Tam istediğim gibi, sağlam ve
kullanışlı.'),
(11, 8, 8, 20, 21, 12, '2025-01-18', 4, 'Tasarıma bayıldım, biraz daha
büyük olabilirdi.'),
(12, 10, 10, 9, 22, 13, '2025-01-19', 5, 'Ürün taze ve güzel
paketlenmişti.');
```

```
INSERT INTO AppliedToPayment (PaymentID, PromotionCodeID)
VALUES
(1, 1),
(2, 2),
```

```
(3, 3),
(4, NULL),
(5, 4),
(6, 5),
(7, NULL),
(8, 7),
(9, NULL),
(10, 9),
(11, 10),
(12, 11),
(13, NULL),
(14, 12),
(15, 13),
(16, NULL),
(17, 14),
(18, 15);
```

```
INSERT INTO Campaign (CampaignID, Name, Description, StartDate, EndDate,
IsPromotion, PromotionType, PromotionStartDate, PromotionEndDate,
PromotionConditions, Status, IsDiscount, DiscountType, DiscountStartDate,
DiscountEndDate, DiscountValue, DiscountConditions)
VALUES
(1, 'Elektronik Bahar İndirimi', 'Elektronik ürünlerde %15 indirim.',
'2025-03-01', '2025-03-31', TRUE, 'LimitedTime', '2025-03-01', '2025-03-
31', 'Minimum alışveriş tutarı 500 TL gereklidir.', 'Active', TRUE,
'Percentage', '2025-03-01', '2025-03-31', 15.00, 'Elektronik ürünlerde
geçerlidir.'),
(2, 'Hediye Çeki Kampanyası', 'Alışverişlerde 500 TL hediye çeki.', '2025-
04-01', '2025-04-30', TRUE, 'LimitedTime', '2025-04-01', '2025-04-30',
'Minimum 1000 TL alışveriş gereklidir.', 'Upcoming', FALSE, NULL, NULL,
NULL, NULL, NULL),
(3, 'Yaz İndirimi', 'Yaz sezonu ürünlerinde %50'ye varan indirim.', '2025-
05-01', '2025-05-31', FALSE, NULL, NULL, NULL, NULL, NULL, TRUE,
'Percentage', '2025-05-01', '2025-05-31', 50.00, 'Yaz sezonu ürünleri.'),
(4, '1 Alana 1 Bedava Kampanyası', 'Seçili ürünlerde 1 alana 1 bedava.',
'2025-06-01', '2025-06-15', TRUE, 'BuyOneGetOne', '2025-06-01', '2025-06-
15', 'Seçili ürünlerde geçerlidir.', 'Active', FALSE, NULL, NULL, NULL,
NULL, NULL),
(5, 'Yeni Teknoloji Fırsatları', 'Yeni teknoloji ürünlerinde %20
indirim.', '2025-06-01', '2025-06-30', TRUE, 'NewArrival', '2025-06-01',
'2025-06-30', 'Teknoloji kategorisinde geçerlidir.', 'Upcoming', TRUE,
'Percentage', '2025-06-01', '2025-06-30', 20.00, 'Minimum alışveriş tutarı
1000 TL gereklidir.'),
(6, 'Doğum Günü Kampanyası', 'Doğum gününüzde %10 indirim.', '2025-01-01',
'2025-12-31', FALSE, NULL, NULL, NULL, NULL, NULL, FALSE, NULL, NULL, NULL,
NULL, NULL),
(7, 'Kış Sezonu Kampanyası', 'Kış sezonu ürünlerinde özel fırsatlar.',
'2025-01-15', '2025-01-31', TRUE, 'Seasonal', '2025-01-15', '2025-01-31',
'Kış sezonu kategorisindeki ürünler.', 'Expired', FALSE, NULL, NULL, NULL,
NULL, NULL),
(8, 'Ücretsiz Kargo Kampanyası', NULL, NULL, NULL, FALSE, NULL, NULL,
NULL, NULL, NULL, TRUE, 'FreeShipping', '2025-02-01', '2025-02-28', NULL,
'Alt limit olmaksızın geçerlidir.'),
;
```

```
INSERT INTO SellerCampaign (SellerID, CampaignID)
VALUES
(20, 2),
(20, 5),
(20, 8),
(21, 4),
(21, 6),
(21, 8),
(22, 4),
(22, 5),
(23, 1),
(23, 7),
(23, 3),
(24, 2),
(24, 5),
(24, 4),
(25, 6),
(25, 8),
(26, 1),
(26, 4),
(26, 7),
(27, 1),
(27, 3),
(28, 4),
(28, 8),
(29, 2),
(29, 5),
(29, 8),
(30, 1),
(30, 7),
(30, 4);
```

```
INSERT INTO CampaignProduct (CampaignID, ProductID)
VALUES
(1, 4),
(1, 5),
(1, 6),
(1, 13),
(1, 14),
(2, 1),
(2, 2),
(2, 3),
(2, 17),
(2, 18),
(3, 8),
(3, 9),
(3, 10),
(3, 15),
(3, 16),
(4, 5),
(4, 6),
```

```
(4, 12),  
(4, 13),  
(4, 21),  
(5, 2),  
(5, 3),  
(5, 9),  
(5, 10),  
(5, 20),  
(6, 8),  
(6, 9),  
(6, 10),  
(6, 19),  
(7, 4),  
(7, 6),  
(7, 14),  
(7, 15),  
(7, 22),  
(8, 1),  
(8, 2),  
(8, 8),  
(8, 9),  
(8, 10),  
(8, 19);
```

```
INSERT INTO CampaignCart (ShoppingCartID, CampaignID)  
VALUES
```

```
(1, 1),  
(1, 2),  
(2, 1),  
(2, 4),  
(3, 2),  
(3, 5),  
(3, 6),  
(4, 3),  
(5, 2),  
(5, 8),  
(6, 1),  
(6, 7),  
(7, 6),  
(7, 8),  
(8, 3),  
(8, 8),  
(9, 5),  
(10, 4),  
(10, 7),  
(11, 1),  
(11, 7),  
(12, 6),  
(13, 8),  
(14, 5),  
(15, 2),  
(15, 8),  
(16, 4),  
(16, 3),
```

```
(17, 8),  
(18, 1),  
(18, 6);
```

**QUESTION;**

(3 pts) Write down 3 triggers for 3 different tables. Triggers should be meaningful.

**ANSWER;**

```
-- Payment tablosu için ödeme detaylarını kontrol eden tetikleyici:
-- Bu tetikleyici, yeni bir ödeme eklenirken PaymentDetailID'nin
sağlanması durumunda diğer ödeme detaylarının NULL olmasını kontrol eder.
-- Eğer PaymentDetailID sağlanmazsa, CardNumber, CardExpiryDate, CVV ve
CardHolderName alanlarının NULL olmamasını sağlar.
DELIMITER //

CREATE TRIGGER ValidatePaymentDetail
BEFORE INSERT ON Payment
FOR EACH ROW
BEGIN
    -- Eğer PaymentDetails mevcutsa diğer alanlar NULL olmalı
    IF NEW.PaymentDetailID IS NOT NULL THEN
        IF NEW.CardNumber IS NOT NULL OR NEW.CardExpiryDate IS NOT NULL OR
NEW.CVV IS NOT NULL OR NEW.CardHolderName IS NOT NULL THEN
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'PaymentDetails sağlandıysa diğer ödeme
detayları NULL olmalı!';
        END IF;
    -- Eğer PaymentDetails mevcut değilse hiçbir alan NULL olmamalı
    ELSE
        IF NEW.CardNumber IS NULL OR NEW.CardExpiryDate IS NULL OR NEW.CVV
IS NULL OR NEW.CardHolderName IS NULL THEN
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'PaymentDetails yoksa diğer ödeme detayları
NULL olamaz!';
        END IF;
    END IF;
END;
//
DELIMITER ;
```

```
-- Review tablosu için aynı müşteri tarafından aynı ürüne yapılan tekrar
yorumları engelleyen tetikleyici:
-- Bu tetikleyici, yeni bir yorum eklenirken, aynı müşteri ve aynı ürün
için daha önce yapılmış bir yorumun olup olmadığını kontrol eder.
-- Eğer daha önce yapılmış bir yorum varsa hata fırlatır.
DELIMITER //

CREATE TRIGGER PreventDuplicateReview
BEFORE INSERT ON Review
FOR EACH ROW
BEGIN
    DECLARE review_count INT;

    -- Aynı müşterinin aynı ürüne yaptığı yorumların sayısını kontrol et
    SELECT COUNT(*)
    INTO review_count
    FROM Review
    WHERE CustomerID = NEW.CustomerID
       AND ProductID = NEW.ProductID;

    -- Eğer zaten bir yorum yaptıysa hata fırlat
    IF review_count > 0 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Bu müşteri bu ürüne zaten bir yorum yapmış!';
    END IF;
END;
//
DELIMITER ;
```

```

-- OrderItem tablosu için stok güncellemesini yapan tetikleyici:
-- Bu tetikleyici, yeni bir sipariş maddesi eklendiğinde stok miktarını
günceller.
-- İlgili ürünün stok miktarı sıfırın altına düşerse hata fırlatır ve
işlem tamamlanmaz.
DELIMITER //

CREATE TRIGGER UpdateStockOnOrder
AFTER INSERT ON OrderItem
FOR EACH ROW
BEGIN
    DECLARE inventory_id INT;
    DECLARE barcode CHAR(13);

    -- İlgili InventoryID ve Barcode değerlerini ProductStock tablosundan
al
    SELECT InventoryID, Barcode
    INTO inventory_id, barcode
    FROM ProductStock
    WHERE ProductID = NEW.ProductID
    LIMIT 1;

    -- Stok miktarını güncelle
    UPDATE Stock
    SET Quantity = Quantity - NEW.Quantity
    WHERE InventoryID = inventory_id
    AND Barcode = barcode;

    -- Stok miktarını kontrol et, negatif olmamalı
    IF (SELECT Quantity FROM Stock WHERE InventoryID = inventory_id AND
Barcode = barcode) < 0 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Stok yetersiz! Sipariş işlenemez.';
    END IF;
END;
//

DELIMITER ;

```



**QUESTION;**

(3 pts) Write down 3 meaningful check constraints.

**ANSWER;**

```
-- Review tablosu için Check Constraint:
-- Bu kısıtlama, Rating (Puan) değerinin 1 ile 5 arasında olmasını sağlar.
ALTER TABLE Review
ADD CONSTRAINT CheckReviewRating
CHECK (Rating BETWEEN 1 AND 5);

-- PaymentDetail tablosu için Check Constraint:
-- Bu kısıtlama, CVV alanının uzunluğunun 3 haneli olmasını ve sadece
rakamlardan oluşmasını sağlar.
ALTER TABLE PaymentDetail
ADD CONSTRAINT CheckCardCVVLength
CHECK (LENGTH(CVV) = 3 AND CVV REGEXP '^[0-9]{3}$');

-- User tablosu için Check Constraint:
-- Bu kısıtlama, Email alanının geçerli bir e-posta formatında olmasını
sağlar.
-- Geçerli format örneği: "example@domain.com"
ALTER TABLE `User`
ADD CONSTRAINT CheckEmailFormat
CHECK (Email LIKE '%_@_%._%');
```

## QUESTION;

Write down the following SQL statements:

- a. (3 pts) Write sample INSERT, DELETE and UPDATE statements for 3 of the tables you have chosen.
- b. Write 5 SELECT statements for the database you have implemented.
  - i. (2 pts) 2 of them should use a minimum of 2 tables.
  - ii. (3 pts) 3 of them should use a minimum of 3 tables.
- c. (2 pts for each) Write 5 original SELECT statements that you think are critical.

## ANSWER;

- a. (3 pts) Write sample INSERT, DELETE and UPDATE statements for 3 of the tables you have chosen.

-- Product table

```
INSERT INTO Product (ProductID, ProductName, Information, Brand, Price,
Properties, SellerID,ShippingCost)
VALUES (23, 'Akıllı Saat', 'Su geçirmez, GPS özellikli', 'TeknoMark',
1500.00, 'Renk: Siyah, Ağırlık: 50g', 26,50);
```

```
DELETE FROM Product
WHERE ProductID = 21;
```

```
UPDATE Product
SET Properties = 'Malzeme: Ahşap, Renk: Sarı'
WHERE ProductID = 1;
```

-- User Table

```
INSERT INTO `User` (Password, BirthDate, RegistrationDate, Email,
PhoneNumber)
VALUES ('password123', '1995-06-15', CURDATE(), 'Amin@example.com',
'05551234567');
```

```
DELETE FROM `User`
WHERE UserID = 9;
```

```
UPDATE `User`
SET Email = 'newemail@example.com', PhoneNumber = '05557654321'
WHERE UserID = 2;
```

-- Campaign Table

```
INSERT INTO Campaign (CampaignID, Name, Description, StartDate, EndDate,
```

```
IsPromotion, PromotionType, PromotionStartDate, PromotionEndDate,  
PromotionConditions, Status, IsDiscount, DiscountType, DiscountStartDate,  
DiscountEndDate, DiscountValue, DiscountConditions)  
VALUES  
-- 1. Promotion ve Discount birlikte kullanılıyor  
(9, 'Ramazan Kampanyası', 'Gıda ürünlerinde 10%.', '2025-03-01', '2025-03-  
29', FALSE, NULL, NULL, NULL, NULL, NULL, TRUE, 'Percentage', '2025-03-  
01', '2025-03-29', 10.00, 'Gıda ürünlerinde geçerlidir.')  
;  
  
DELETE FROM Campaign  
WHERE CampaignID = 1;  
  
UPDATE Campaign  
SET EndDate = '2025-05-30'  
WHERE CampaignID = 2
```

b. Write 5 SELECT statements for the database you have implemented.

i. (2 pts) 2 of them should use a minimum of 2 tables.

-- 2 tabloyu ilgilendiren SELECT sorguları:

-- Satıcı ve stok bilgilerini listeleyen sorgu:

-- Bu sorgu, ürün adı, satıcının adı, mağaza adı ve stok miktarını listelemektedir.

```
SELECT P.ProductName, S.FirstName AS SellerName, S.StoreName, ST.Quantity
FROM Product P
JOIN Seller S ON P.SellerID = S.SellerID
JOIN ProductStock PS ON P.ProductID = PS.ProductID
JOIN Stock ST ON PS.InventoryID = ST.InventoryID AND PS.Barcode =
ST.Barcode;
```

-- Müşteri ürün incelemelerini listeleyen sorgu:

-- Bu sorgu, müşteri adını, ürün adını, mağaza adını, ürün puanını ve yorumu listelemektedir.

```
SELECT CUS.FirstName, CUS.LastName, P.ProductName, S.StoreName, R.Rating,
R.Comment
FROM Customer CUS
JOIN Review R ON CUS.CustomerID = R.CustomerID
JOIN Product P ON R.ProductID = P.ProductID
JOIN Seller S ON P.SellerID = S.SellerID;
```

ii. (3 pts) 3 of them should use a minimum of 3 tables.

-- 3 tabloyu ilgilendiren SELECT sorguları:

-- Sipariş, müşteri ve kargo detaylarını listeleyen sorgu:

-- Bu sorgu, müşterinin adı, sipariş bilgileri, kargo takip numarası ve teslimat durumunu listelemektedir.

```
SELECT
    C.FirstName AS CustomerFirstName,
    C.LastName AS CustomerLastName,
    O.OrderNo,
    O.OrderDate,
    O.TotalAmount,
    S.AddressTitle,
    S.City,
    S.District,
    S.Neighborhood,
    SH.TrackingNumber,
    SH.DeliveryStatus
FROM Customer C
JOIN `User` U ON C.CustomerID = U.UserID
JOIN `Order` O ON C.CustomerID = O.CustomerID
JOIN ShippingDetail S ON O.ShippingID = S.ShippingID
LEFT JOIN Shipment SH ON O.LogisticID = SH.LogisticID;
```

-- Satıcıların stok bilgileri ve ürün detaylarını listeleyen sorgu:

-- Bu sorgu, mağaza adı, ürün adı, stok miktarı, liste fiyatı ve satıcı türü bilgilerini döndürür.

```
SELECT S.StoreName, P.ProductName, ST.Quantity, ST.ListPrice,
S.SellerType, S.StoreRating
FROM Seller S
JOIN Inventory I ON S.SellerID = I.SellerID
JOIN Stock ST ON I.InventoryID = ST.InventoryID
JOIN ProductStock PS ON ST.InventoryID = PS.InventoryID AND ST.Barcode =
PS.Barcode
JOIN Product P ON PS.ProductID = P.ProductID;
```

-- Sipariş ürünleri ve müşteri detaylarını listeleyen sorgu:

-- Bu sorgu, müşteri adı, sipariş numarası, ürün adı ve ürün miktarını listelemektedir.

```
SELECT CUS.FirstName, CUS.LastName, O.OrderNo, P.ProductName, OI.Quantity
FROM Customer CUS
JOIN `Order` O ON CUS.CustomerID = O.CustomerID
JOIN OrderItem OI ON O.OrderNo = OI.OrderNo AND O.CustomerID =
OI.CustomerID
JOIN Product P ON OI.ProductID = P.ProductID;
```

c. (2 pts for each) Write 5 original SELECT statements that you think are critical.

-- 5 kritik SELECT sorgusu:

-- Aynı barkodlu ürünlerin satıcı ve fiyat detayları:

-- Bu sorgu, aynı barkoda sahip ürünlerin farklı satıcıdaki fiyatlarını ve açıklamalarını listelemektedir.

```
SELECT
    PS.Barcode,
    P.ProductName,
    S.SellerID,
    S.StoreName,
    S.StoreRating, -- Satıcı Puanı
    P.Information, -- Ürün Açıklaması
    P.ShippingCost,
    P.Price AS ProductPrice
FROM
    ProductStock PS
JOIN
    Product P ON PS.ProductID = P.ProductID
JOIN
    Seller S ON P.SellerID = S.SellerID
WHERE
    PS.Barcode IN (
        SELECT Barcode
        FROM ProductStock PS1
        JOIN Product P1 ON PS1.ProductID = P1.ProductID
        JOIN Seller S1 ON P1.SellerID = S1.SellerID
        GROUP BY PS1.Barcode
        HAVING COUNT(DISTINCT S1.SellerID) > 1 -- Farklı satıcı sayısını
kontrol eder
    )
GROUP BY
    PS.Barcode, P.ProductName, S.SellerID, S.StoreName, S.StoreRating,
P.Information, P.ShippingCost, P.Price
ORDER BY
    PS.Barcode, P.Price;
```

-- Kampanya ve indirim detayları:  
-- Bu sorgu, kampanyalara dahil olan ürünlerin indirim sonrası fiyatlarını  
ve kampanya detaylarını listelemektedir.

```
SELECT
    CU.FirstName,
    CU.LastName,
    P.ProductName,
    P.Price,
    C.Name AS CampaignName,
    C.IsDiscount,
    C.DiscountType,
    C.DiscountValue,
    C.IsPromotion,
    C.PromotionType,
    C.PromotionConditions,

    CASE
        WHEN C.IsDiscount = TRUE THEN
            CASE
                WHEN C.DiscountType = 'Percentage' THEN P.Price - (P.Price
* C.DiscountValue / 100)
                WHEN C.DiscountType = 'Flat' THEN P.Price -
C.DiscountValue
                ELSE P.Price
            END
        ELSE P.Price
    END AS PriceAfterDiscount,

    CASE
        WHEN C.IsPromotion = TRUE THEN
            CASE
                WHEN C.PromotionType = 'FreeShipping' THEN CONCAT('Free
Shipping Applied')
                WHEN C.PromotionType = 'BuyOneGetOne' THEN
CONCAT('Eligible for Buy 1 Get 1 Offer')
                ELSE 'Promotion Applied'
            END
        ELSE 'No Promotion'
    END AS PromotionDetails
FROM Customer CU
JOIN ShoppingCart SC ON CU.ShoppingCartID = SC.ShoppingCartID
JOIN CartItems CI ON SC.ShoppingCartID = CI.ShoppingCartID
JOIN Product P ON CI.ProductID = P.ProductID
JOIN CampaignCart CC ON SC.ShoppingCartID = CC.ShoppingCartID
JOIN Campaign C ON CC.CampaignID = C.CampaignID;
```

```

-- Teslimat gecikmelerini listeleyen sorgu:
-- Bu sorgu, siparişlerin teslimat durumlarını ve gecikme günlerini
hesaplamaktadır.
SELECT
    O.OrderNo,
    CU.FirstName,
    CU.LastName,
    S.TrackingNumber,
    S.CarrierName,
    S.DeliveryStatus,
    S.EstimatedDeliveryDate,
    S.DeliveryDate,
    DATEDIFF(S.DeliveryDate, S.EstimatedDeliveryDate) AS DelayDays
FROM `Order` O
JOIN Customer CU ON O.CustomerID = CU.CustomerID
JOIN Shipment S ON O.LogisticID = S.LogisticID
WHERE S.DeliveryStatus = 'Delivered';

-- Satıcıların en çok sattığı ürünler:
-- Bu sorgu, her satıcının en çok satış yaptığı ürünleri ve toplam satış
miktarlarını listeler.
SELECT S.StoreName, S.SellerType, S.SalesCategoryType, P.ProductName,
SUM(OI.Quantity) AS TotalSold
FROM Seller S
JOIN Product P ON S.SellerID = P.SellerID
JOIN OrderItem OI ON P.ProductID = OI.ProductID
GROUP BY S.SellerID, P.ProductID
ORDER BY S.StoreName, TotalSold DESC;

-- Düşük stok seviyesindeki ürünler:
-- Bu sorgu, depolardaki düşük stok seviyesine sahip ürünleri ve bu
ürünlerin detaylarını listeler.
SELECT P.ProductName, ST.Barcode, ST.Quantity, I.InventoryName,
I.AdressLine, S.StoreName
FROM Stock ST
JOIN ProductStock PS ON ST.InventoryID = PS.InventoryID AND ST.Barcode =
PS.Barcode
JOIN Product P ON PS.ProductID = P.ProductID
JOIN Inventory I ON ST.InventoryID = I.InventoryID
JOIN Seller S ON I.SellerID = S.SellerID
WHERE ST.Quantity < 20
ORDER BY ST.Quantity ASC;

```



#Her bir kullanıcı için en son yapılan sipariş ve siparişin toplam tutarı  
#Bu sorgu, her bir kullanıcı için en son yapılan sipariş bilgilerini ve  
sipariş toplam tutarını listeler.

```
SELECT
    CUS.FirstName,
    CUS.LastName,
    MAX(O.OrderDate) AS LastOrderDate,
    SUM(O.TotalAmount) AS TotalSpent,
    COUNT(DISTINCT OI.OrderItemNo) AS TotalOrderItems,
    GROUP_CONCAT(P.ProductName ORDER BY P.ProductName ASC SEPARATOR ', ')
AS ProductNames,
GROUP_CONCAT(OI.Quantity ORDER BY P.ProductName ASC SEPARATOR ', ') AS
Quantities,
    GROUP_CONCAT(OI.ItemStatus ORDER BY P.ProductName ASC SEPARATOR ', ')
AS ItemStatuses
FROM Customer CUS
JOIN `Order` O ON CUS.CustomerID = O.CustomerID
JOIN OrderItem OI ON O.OrderNo = OI.OrderNo
JOIN Product P ON OI.ProductID = P.ProductID
GROUP BY CUS.CustomerID
ORDER BY LastOrderDate DESC;
```