Fırat Üniversitesi Bilgisayar Mühendisliği VERİ TABANI SİSTEMLERİ Dersi Dönem Projesi

Proje Adı : Kargo Takip Otomasyonu(CargoTracking)

Proje Amacı : Kargo Takip Otomasyonu, kargo gönderim süreçlerini yönetmek ve takip etmek için tasarlanmıştır. Sistem, kullanıcıların kargolarını oluşturma, adreslerini kaydetme, gönderimlerin durumlarını izlemeyi kolaylaştırmayı amaçlıyor.

Proje Ekibindeki Kişiler :

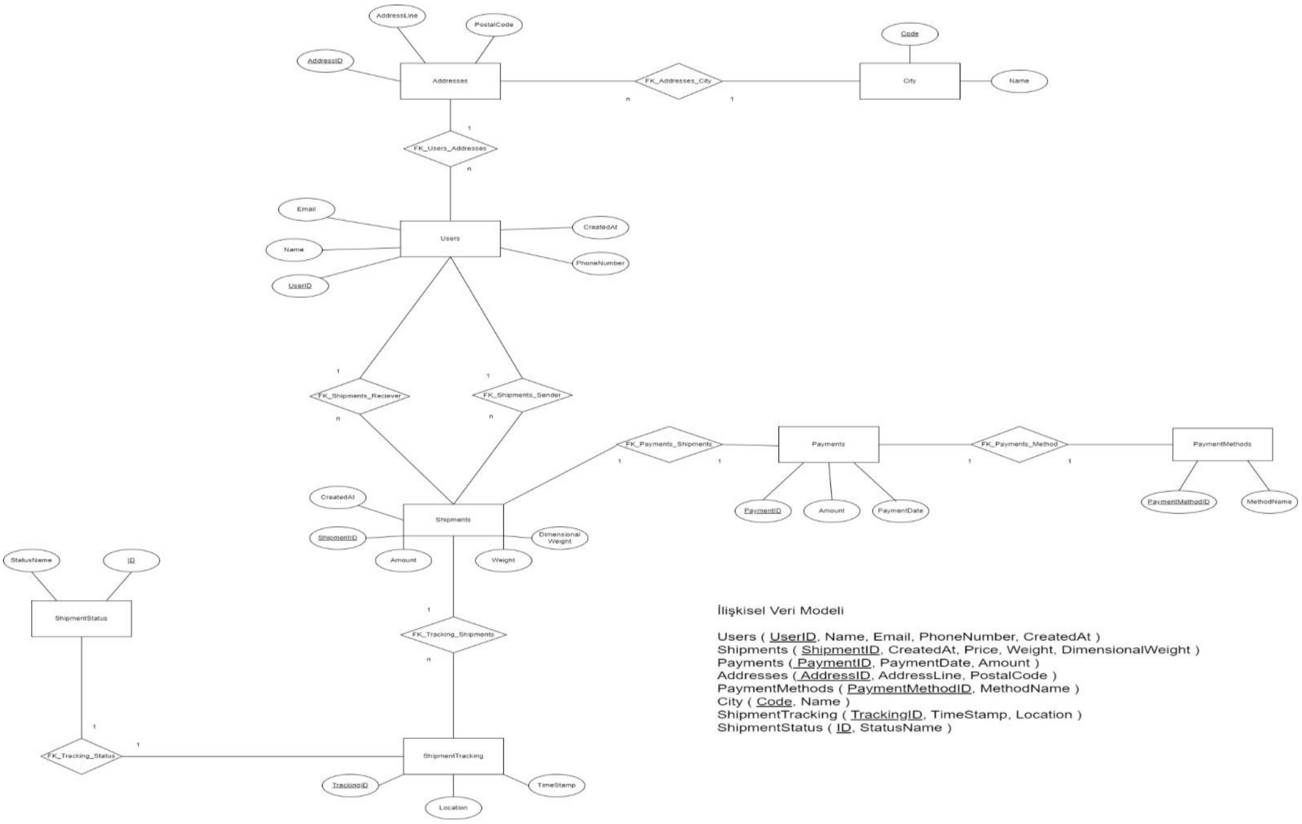
- Ömer Karaman 235260301
- Ahmet Yasin Durakcı 235260125
- Muhammed Bünyamin Altun 205260011

Proje Github Linki

İlişkisel Veri Modeli

Users (<u>UserID</u>, Name, Email, PhoneNumber, CreatedAt **)**
Shipments (<u>ShipmentID</u>, CreatedAt, Price, Weight, DimensionalWeight **)**
Payments (<u>PaymentID</u>, PaymentDate, Amount **)**
Addresses (<u>AddressID</u>, AddressLine, PostalCode **)**
PaymentMethods (<u>PaymentMethodID</u>, MethodName **)**
City (<u>Code</u>, Name **)**
ShipmentTracking (<u>TrackingID</u>, TimeStamp, Location **)**
ShipmentStatus (<u>ID</u>, StatusName **)**

ER Diagramı



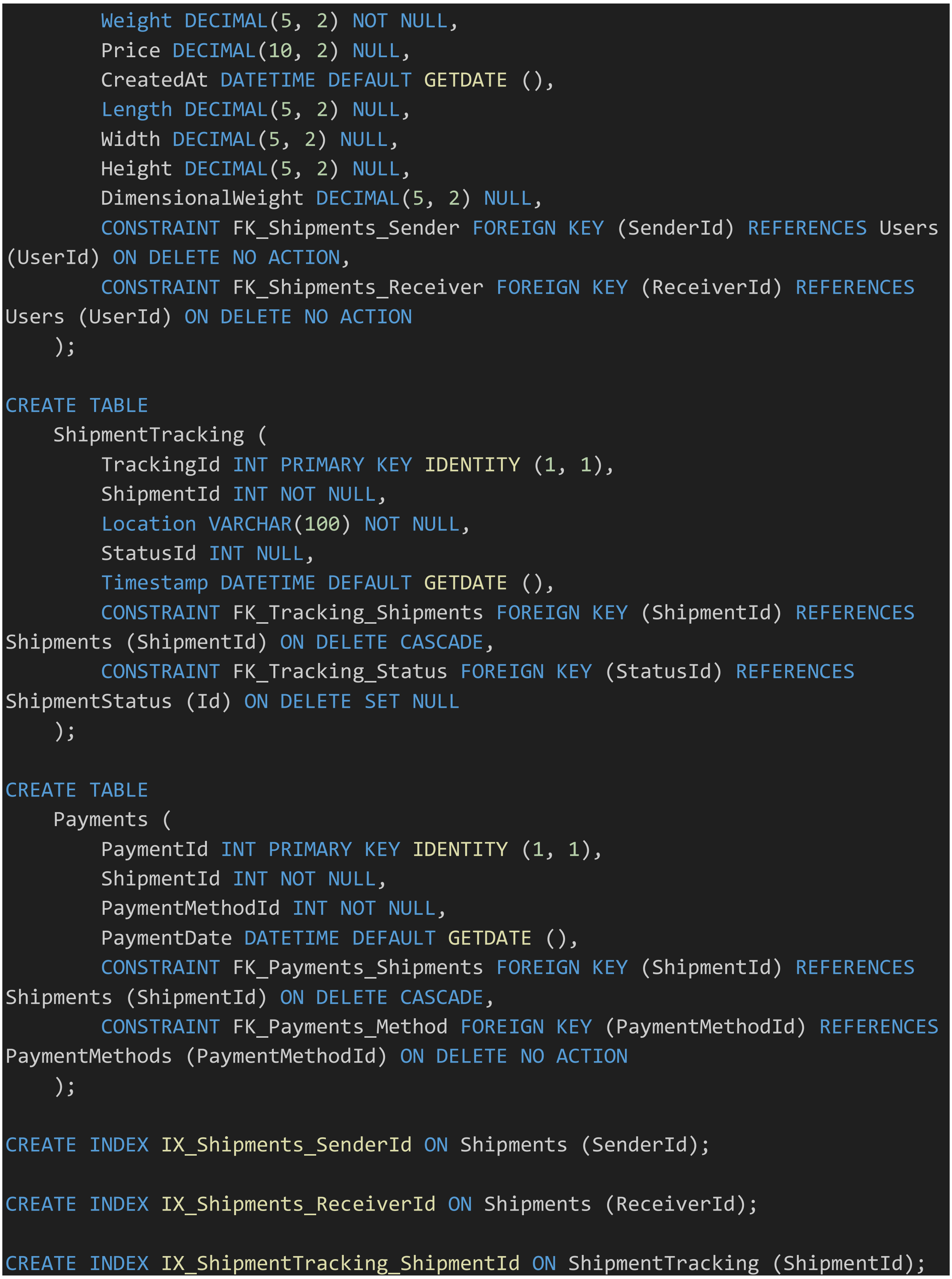
İlişkisel Veri Modeli

Users (<u>UserID</u>, Name, Email, PhoneNumber, CreatedAt)
Shipments (<u>ShipmentID</u>, CreatedAt, Price, Weight, DimensionalWeight)
Payments (<u>PaymentID</u>, PaymentDate, Amount)
Addresses (<u>AddressID</u>, AddressLine, PostalCode)
PaymentMethods (<u>PaymentMethodID</u>, MethodName)
City (<u>Code</u>, Name)
ShipmentTracking (<u>TrackingID</u>, TimeStamp, Location)
ShipmentStatus (<u>ID</u>, StatusName)

Scripts :

CreateTables.sql

```
CREATE TABLE
    City (
        code char(2) PRIMARY KEY NOT NULL,
        name nvarchar (50) NOT NULL
    )
CREATE TABLE
    ShipmentStatus (
        Id INT PRIMARY KEY IDENTITY (1, 1),
        StatusName VARCHAR(50) NOT NULL
    );

CREATE TABLE
    PaymentMethods (
        PaymentMethodId INT PRIMARY KEY IDENTITY (1, 1),
        MethodName VARCHAR(50) NOT NULL
    );

CREATE TABLE
    Addresses (
        AddressId INT PRIMARY KEY IDENTITY (1, 1),
        AddressLine VARCHAR(255) NOT NULL,
        CityId char(2) NOT NULL,
        PostalCode VARCHAR(10),
        CONSTRAINT FK_Addresses_City FOREIGN KEY (CityId) REFERENCES City
(code) ON DELETE NO ACTION
    );

CREATE TABLE
    Users (
        UserId INT PRIMARY KEY IDENTITY (1, 1),
        Name VARCHAR(100) NOT NULL,
        Surname VARCHAR(100) NOT NULL,
        PhoneNumber VARCHAR(15) UNIQUE,
        Email VARCHAR(100) UNIQUE,
        CreatedAt DATETIME DEFAULT GETDATE (),
        AddressId INT NULL,
        CONSTRAINT FK_Users_Addresses FOREIGN KEY (AddressId) REFERENCES
Addresses (AddressId) ON DELETE SET NULL
    );

CREATE TABLE
    Shipments (
        ShipmentId INT PRIMARY KEY IDENTITY (1, 1),
        SenderId INT NOT NULL,
        ReceiverId INT NOT NULL,
```



```
        Weight DECIMAL(5, 2) NOT NULL,
        Price DECIMAL(10, 2) NULL,
        CreatedAt DATETIME DEFAULT GETDATE (),
        Length DECIMAL(5, 2) NULL,
        Width DECIMAL(5, 2) NULL,
        Height DECIMAL(5, 2) NULL,
        DimensionalWeight DECIMAL(5, 2) NULL,
        CONSTRAINT FK_Shipments_Sender FOREIGN KEY (SenderId) REFERENCES Users (UserId) ON DELETE NO ACTION,
        CONSTRAINT FK_Shipments_Receiver FOREIGN KEY (ReceiverId) REFERENCES Users (UserId) ON DELETE NO ACTION
    );

CREATE TABLE
    ShipmentTracking (
        TrackingId INT PRIMARY KEY IDENTITY (1, 1),
        ShipmentId INT NOT NULL,
        Location VARCHAR(100) NOT NULL,
        StatusId INT NULL,
        Timestamp DATETIME DEFAULT GETDATE (),
        CONSTRAINT FK_Tracking_Shipments FOREIGN KEY (ShipmentId) REFERENCES Shipments (ShipmentId) ON DELETE CASCADE,
        CONSTRAINT FK_Tracking_Status FOREIGN KEY (StatusId) REFERENCES ShipmentStatus (Id) ON DELETE SET NULL
    );

CREATE TABLE
    Payments (
        PaymentId INT PRIMARY KEY IDENTITY (1, 1),
        ShipmentId INT NOT NULL,
        PaymentMethodId INT NOT NULL,
        PaymentDate DATETIME DEFAULT GETDATE (),
        CONSTRAINT FK_Payments_Shipments FOREIGN KEY (ShipmentId) REFERENCES Shipments (ShipmentId) ON DELETE CASCADE,
        CONSTRAINT FK_Payments_Method FOREIGN KEY (PaymentMethodId) REFERENCES PaymentMethods (PaymentMethodId) ON DELETE NO ACTION
    );

CREATE INDEX IX_Shipments_SenderId ON Shipments (SenderId);

CREATE INDEX IX_Shipments_ReceiverId ON Shipments (ReceiverId);

CREATE INDEX IX_ShipmentTracking_ShipmentId ON ShipmentTracking (ShipmentId);
```

İnit Data

```
insert into
    ShipmentStatus (StatusName)
values
    ('Kabul Edildi'),
    ('Taşımada'),
    ('Dağıtıcıya Verildi'),
    ('Teslim Edildi');

insert into
    PaymentMethods (MethodName)
values
    ('Gönderici Öder'),
    ('Alıcı Öder');

INSERT INTO
    City
VALUES
    ('01', 'Adana'),('02', 'Adıyaman'),('03', 'Afyon'),('04', 'Ağrı'),('05',
'Amasya'),('06', 'Ankara'),('07', 'Antalya'),
    ('08', 'Artvin'),('09', 'Aydın'),('10', 'Balıkesir'),('11',
'Bilecik'),('12', 'Bingöl'),('13', 'Bitlis'),('14', 'Bolu'),
    ('15', 'Burdur'),('16', 'Bursa'),('17', 'Çanakkale'),('18',
'Çankırı'),('19', 'Çorum'),('20', 'Denizli'),('21', 'Diyarbakır'),
    ('22', 'Edirne'),('23', 'Elazığ'),('24', 'Erzincan'),('25',
'Erzurum'),('26', 'Eskişehir'),('27', 'Gaziantep'),('28', 'Giresun'),
    ('29', 'Gümüşhane'),('30', 'Hakkari'),('31', 'Hatay'),('32',
'Isparta'),('33', 'Mersin'),('34', 'İstanbul'),('35', 'İzmir'),('36', 'Kars'),
    ('37', 'Kastamonu'),('38', 'Kayseri'),('39', 'Kırklareli'),('40',
'Kırşehir'),('41', 'Kocaeli'),('42', 'Konya'),('43', 'Kütahya'),
    ('44', 'Malatya'),('45', 'Manisa'),
    ('46', 'K.Maraş'),('47', 'Mardin'),('48', 'Muğla'),('49', 'Muş'),('50',
'Nevşehir'),('51', 'Niğde'),('52', 'Ordu'),
    ('53', 'Rize'),('54', 'Sakarya'),('55', 'Samsun'),('56', 'Siirt'),('57',
'Sinop'),('58', 'Sivas'),('59', 'Tekirdağ'),('60', 'Tokat'),('61', 'Trabzon'),
    ('62', 'Tunceli'),('63', 'Şanlıurfa'),('64', 'Uşak'),('65', 'Van'),('66',
'Yozgat'),('67', 'Zonguldak'),('68', 'Aksaray'),('69', 'Bayburt'),('70',
'Karaman'),
    ('71', 'Kırıkkale'),('72', 'Batman'),('73', 'Şırnak'),('74',
'Bartın'),('75', 'Ardahan'),('76', 'Iğdır'),('77', 'Yalova'),
    ('78', 'Karabük'),('79', 'Kilis'),('80', 'Osmaniye'),('81', 'Düzce')
```

Data Seed

```sql
DECLARE @AddressId INT;
DECLARE @ReceiverId INT;
DECLARE @SenderId INT;
DECLARE @ShipmentId INT;


INSERT INTO Addresses (AddressLine, CityId, PostalCode)
    VALUES
        ('Atatürk Mahallesi Cumhuriyet Caddesi 330.sokak no 13 daire 4 
Çayırova', '41', 00041);

SET @AddressId = SCOPE_IDENTITY();

INSERT INTO Users (Name,Surname, PhoneNumber, Email, CreatedAt, AddressId)
    VALUES
        ('ÖMER','Karaman', '+905060528330', 'drakken120@gmail.com', GETDATE(), 
@AddressId);

SET @ReceiverId =SCOPE_IDENTITY();

INSERT INTO Addresses (AddressLine, CityId, PostalCode)
    VALUES
        ('Sanat Mahallesi Nazım Hikme Caddesi 3 Haziran sokka no 60 daire 1', 
'06', 00006);

SET @AddressId = SCOPE_IDENTITY();

INSERT INTO Users (Name,Surname, PhoneNumber, Email, CreatedAt, AddressId)
    VALUES
        ('Nazım','Hikmet', '+905060526060', 'nazimhikmetci@gmail.com', 
GETDATE(), @AddressId);
SET @SenderId = SCOPE_IDENTITY();

INSERT INTO 
Shipments(SenderId,ReceiverId,Weight,Length,Height,Width,CreatedAt)
    VALUES
        (@ReceiverId,@SenderId,160,30.0,30.0,50.0,GETDATE());

SET @ShipmentId = SCOPE_IDENTITY();

INSERT INTO Payments(ShipmentId,PaymentMethodId,PaymentDate)
    VALUES
        (@ShipmentId,2,GETDATE());

INSERT INTO ShipmentTracking(ShipmentId,Location,StatusId,Timestamp)
    VALUES
        (@ShipmentId,'Kocaeli',1,GETDATE());
```

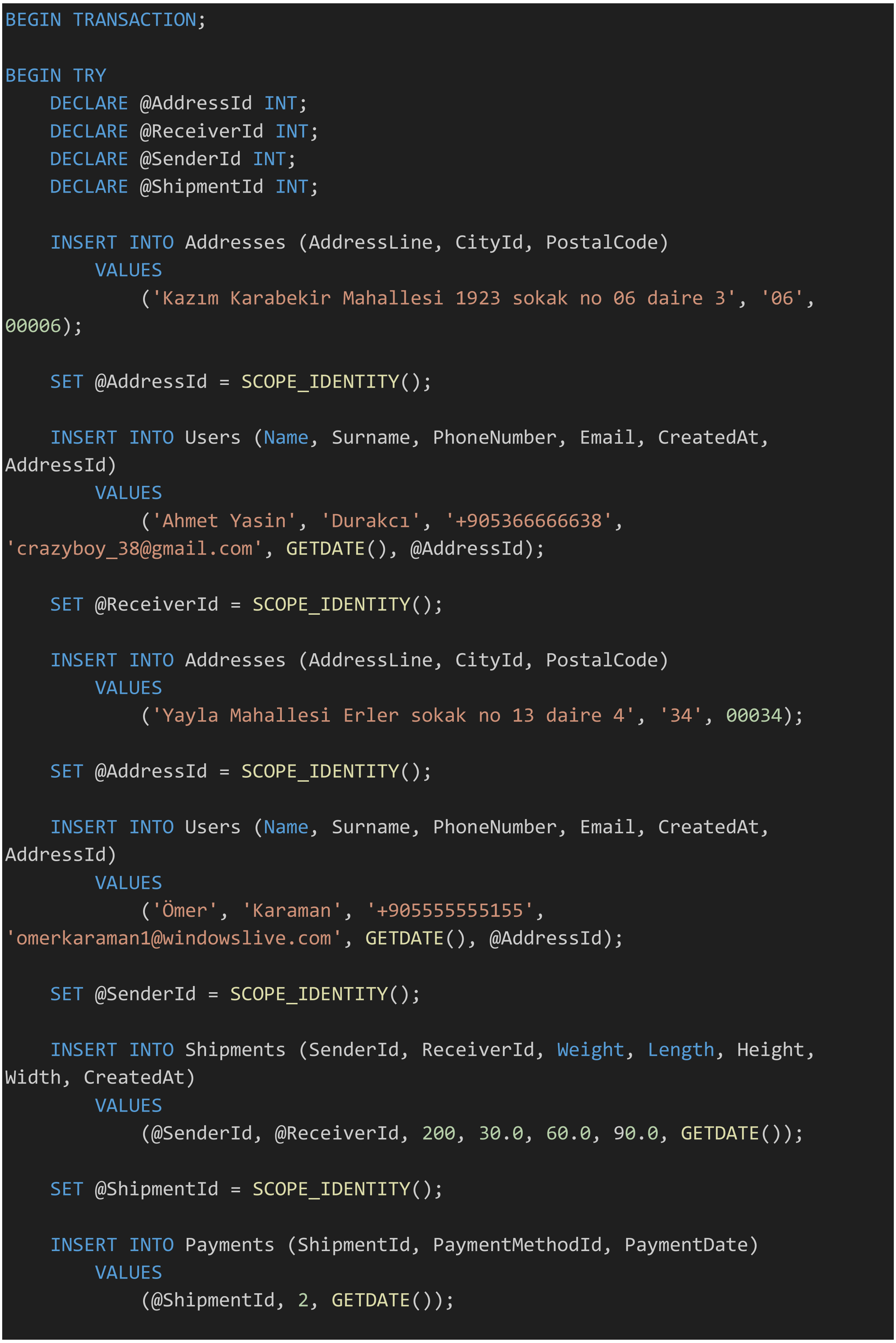
Procedures

```
CREATE PROCEDURE UpdateShipmentTracking
(   @ShipmentId int,
    @Location VARCHAR(100),
    @StatusId int
)
AS
BEGIN
    INSERT INTO ShipmentTracking(ShipmentId,Location,StatusId,Timestamp)
VALUES(@ShipmentId,@Location,@StatusId,GETDATE());
END


CREATE PROCEDURE CargoInformation
(
    @ShipmentId int
)
AS
BEGIN
    SELECT st.ShipmentId,st.TrackingId,st.Location,ss.StatusName,st.Timestamp
FROM ShipmentTracking as st
    inner join ShipmentStatus as ss
    on ss.Id = st.StatusId
    WHERE st.ShipmentId =@ShipmentId
END
```

Trigger

```
CREATE TRIGGER shipment_calculate_dim
ON Shipments
AFTER INSERT
AS
BEGIN
    DECLARE @price INT;
    DECLARE @shipmentId INT;

    UPDATE Shipments
    SET
        DimensionalWeight = calc_dim,
        Price = calc_dim * 5
    FROM
        Shipments s
        INNER JOIN (
            SELECT
                ShipmentId,
                (Length * Width * Height) / 5000 AS calc_dim
            FROM
                inserted
        ) i ON s.ShipmentId = i.ShipmentId;
END;
```

Transaction



```
BEGIN TRANSACTION;

BEGIN TRY
    DECLARE @AddressId INT;
    DECLARE @ReceiverId INT;
    DECLARE @SenderId INT;
    DECLARE @ShipmentId INT;

    INSERT INTO Addresses (AddressLine, CityId, PostalCode)
        VALUES
            ('Kazım Karabekir Mahallesi 1923 sokak no 06 daire 3', '06', 
00006);

    SET @AddressId = SCOPE_IDENTITY();

    INSERT INTO Users (Name, Surname, PhoneNumber, Email, CreatedAt, 
AddressId)
        VALUES
            ('Ahmet Yasin', 'Durakcı', '+905366666638', 
'crazyboy_38@gmail.com', GETDATE(), @AddressId);

    SET @ReceiverId = SCOPE_IDENTITY();

    INSERT INTO Addresses (AddressLine, CityId, PostalCode)
        VALUES
            ('Yayla Mahallesi Erler sokak no 13 daire 4', '34', 00034);

    SET @AddressId = SCOPE_IDENTITY();

    INSERT INTO Users (Name, Surname, PhoneNumber, Email, CreatedAt, 
AddressId)
        VALUES
            ('Ömer', 'Karaman', '+905555555155', 
'omerkaraman1@windowslive.com', GETDATE(), @AddressId);

    SET @SenderId = SCOPE_IDENTITY();

    INSERT INTO Shipments (SenderId, ReceiverId, Weight, Length, Height, 
Width, CreatedAt)
        VALUES
            (@SenderId, @ReceiverId, 200, 30.0, 60.0, 90.0, GETDATE());

    SET @ShipmentId = SCOPE_IDENTITY();

    INSERT INTO Payments (ShipmentId, PaymentMethodId, PaymentDate)
        VALUES
            (@ShipmentId, 2, GETDATE());
    INSERT INTO ShipmentTracking (ShipmentId, Location, StatusId, Timestamp)
        VALUES
            (@ShipmentId, İstanbul, 1, GETDATE());

    COMMIT TRANSACTION;
    PRINT 'Transaction completed successfully.';
END TRY
BEGIN CATCH
    ROLLBACK TRANSACTION;
    PRINT 'Transaction failed. All changes have been rolled back.';
END CATCH;
```

Drop tables

```
DELETE FROM Payments;
DELETE FROM ShipmentTracking;
DELETE FROM Shipments;
DELETE FROM Users;
DELETE FROM Addresses;
DELETE FROM PaymentMethods;
DELETE FROM ShipmentStatus;
DELETE FROM City;

DROP TABLE IF EXISTS Payments;
DROP TABLE IF EXISTS ShipmentTracking;
DROP TABLE IF EXISTS Shipments;
DROP TABLE IF EXISTS Users;
DROP TABLE IF EXISTS Addresses;
DROP TABLE IF EXISTS PaymentMethods;
DROP TABLE IF EXISTS ShipmentStatus;
DROP TABLE IF EXISTS City;
```