CREATE TABLE Kategoriler (

KategoriID INT PRIMARY KEY,

KategoriAdi VARCHAR(50),

Aciklama VARCHAR(255)

);

CREATE TABLE Tedarikciler (

TedarikciID INT PRIMARY KEY,

TedarikciAdi VARCHAR(50),

Adres VARCHAR(255),

Telefon VARCHAR(20)

);

CREATE TABLE Urunler (

UrunID INT PRIMARY KEY,

KategoriID INT,

TedarikciID INT,

UrunAdi VARCHAR(100),

Fiyat DECIMAL(10,2),

StokMiktari INT

);

CREATE TABLE Musteriler (

MusteriID INT PRIMARY KEY,

MusteriAdi VARCHAR(50),

Email VARCHAR(50),

Telefon VARCHAR(20),

Adres VARCHAR(255)

);

CREATE TABLE Siparisler (

SiparisID INT PRIMARY KEY,

MusteriID INT,

SiparisTarihi DATETIME,

SiparisDurumu VARCHAR(20)

);

CREATE TABLE SiparisDetaylari (

SiparisDetayID INT PRIMARY KEY,

SiparisID INT,

UrunID INT,

Adet INT,

Fiyat DECIMAL(10,2)

);

FOREIGN KEY

-- Urunler ve Kategoriler tabloları arasında ilişki

ALTER TABLE Urunler

ADD CONSTRAINT FK\_Urunler\_Kategoriler

FOREIGN KEY (KategoriID) REFERENCES Kategoriler(KategoriID);

-- Urunler ve Tedarikciler tabloları arasında ilişki

ALTER TABLE Urunler

ADD CONSTRAINT FK\_Urunler\_Tedarikciler

FOREIGN KEY (TedarikciID) REFERENCES Tedarikciler(TedarikciID);

-- Siparisler ve Musteriler tabloları arasında ilişki

ALTER TABLE Siparisler

ADD CONSTRAINT FK\_Siparisler\_Musteriler

FOREIGN KEY (MusteriID) REFERENCES Musteriler(MusteriID);

-- SiparisDetaylari ve Siparisler tabloları arasında ilişki

ALTER TABLE SiparisDetaylari

ADD CONSTRAINT FK\_SiparisDetaylari\_Siparisler

FOREIGN KEY (SiparisID) REFERENCES Siparisler(SiparisID);

-- SiparisDetaylari ve Urunler tabloları arasında ilişki

ALTER TABLE SiparisDetaylari

ADD CONSTRAINT FK\_SiparisDetaylari\_Urunler

FOREIGN KEY (UrunID) REFERENCES Urunler(UrunID);

CREATE PROCEDURE SiparisOlustur

@MusteriID INT,

@UrunID INT,

@Adet INT

AS

BEGIN

DECLARE @StokMiktari INT;

DECLARE @Fiyat DECIMAL(10,2);

-- Ürün stok miktarını kontrol et

SELECT @StokMiktari = StokMiktari, @Fiyat = Fiyat

FROM Urunler

WHERE UrunID = @UrunID;

IF (@StokMiktari >= @Adet)

BEGIN

-- Yeni sipariş oluştur

DECLARE @SiparisID INT;

INSERT INTO Siparisler (MusteriID, SiparisTarihi, SiparisDurumu)

VALUES (@MusteriID, GETDATE(), 'Hazırlanıyor');

SET @SiparisID = SCOPE\_IDENTITY();

-- Sipariş detayını ekle

INSERT INTO SiparisDetaylari (SiparisID, UrunID, Adet, Fiyat)

VALUES (@SiparisID, @UrunID, @Adet, @Fiyat);

-- Stok miktarını güncelle

UPDATE Urunler

SET StokMiktari = StokMiktari - @Adet

WHERE UrunID = @UrunID;

PRINT 'Sipariş başarıyla oluşturuldu.';

END

ELSE

BEGIN

PRINT 'Yetersiz stok miktarı.';

END

END

CREATE TRIGGER StokKontrol ON SiparisDetaylari

AFTER INSERT

AS

BEGIN

DECLARE @UrunID INT;

DECLARE @Adet INT;

SELECT @UrunID = UrunID, @Adet = Adet

FROM inserted;

IF EXISTS (SELECT 1 FROM Urunler WHERE UrunID = @UrunID AND StokMiktari < 0)

BEGIN

RAISERROR ('Stok miktarı yetersiz!', 16, 1);

ROLLBACK TRANSACTION;

END

END

CREATE VIEW KategorilereGoreSatislar AS

SELECT k.KategoriAdi, SUM(sd.Adet \* sd.Fiyat) AS ToplamSatis

FROM Kategoriler k

INNER JOIN Urunler u ON k.KategoriID = u.KategoriID

INNER JOIN SiparisDetaylari sd ON u.UrunID = sd.UrunID

GROUP BY k.KategoriAdi;

INSERT INTO Kategoriler (KategoriID, KategoriAdi, Aciklama)

VALUES

(1, 'Elektronik', 'Elektronik cihazlar'),

(2, 'Giyim', 'Kıyafet ve aksesuarlar'),

(3, 'Ev Gereçleri', 'Mutfak ve ev eşyaları'),

(4, 'Kitap', 'Kitaplar ve dergiler'),

(5, 'Spor', 'Spor malzemeleri');

INSERT INTO Tedarikciler (TedarikciID, TedarikciAdi, Adres, Telefon)

VALUES

(1, 'Teknosa', 'İstanbul', '0212 123 45 67'),

(2, 'MediaMarkt', 'Ankara', '0312 987 65 43'),

(3, 'Trendyol', 'Online', '-'),

(4, 'Hepsiburada', 'Online', '-'),

(5, 'Nike', 'İstanbul', '0212 555 55 55');

INSERT INTO Urunler (UrunID, KategoriID, TedarikciID, UrunAdi, Fiyat, StokMiktari)

VALUES

(1, 1, 1, 'iPhone 13', 25000.00, 50),

(2, 1, 2, 'Samsung Galaxy S22', 20000.00, 30),

(3, 2, 3, 'Gömlek', 150.00, 100),

(4, 2, 4, 'Pantolon', 200.00, 80),

(5, 3, 1, 'Buzdolabı', 8000.00, 10);

INSERT INTO Musteriler (MusteriID, MusteriAdi, Email, Telefon, Adres)

VALUES

(1, 'Hasan Yıldız', 'hasan.yildiz@gmail.com', '0544 561 21 60', 'Adana'),

(2, 'Emine Ayten', 'emine.ayten@gmail.com', '0551 444 19 10', 'Ankara'),

(3, 'Yarkın Taskın', 'yarkın.taskın@gmail.com', '0555 777 88 99', 'İzmir');

(4, 'Kutay Duran', 'kutay.duran@gmail.com', '0530 250 44 81', 'İstanbul');

(5, 'YunusEmre Karatas', 'yunusemre.karatas@gmail.com', '0530 250 44 81', 'Manisa');

INSERT INTO Siparisler (SiparisID, MusteriID, SiparisTarihi, SiparisDurumu)

VALUES

(1, 1, '2023-12-25 14:30:00', 'Hazırlanıyor'),

(2, 2, '2024-01-10 10:15:00', 'Kargolandı'),

(3, 1, '2024-02-18 16:45:00', 'Teslim Edildi'),

(4, 4, '2023-12-25 14:30:00', 'Stokta Yok'),

(5, 5, '2023-12-25 14:30:00', 'Tedarik Ediliyor');