



## **Mühendislik Fakültesi Bilgisayar Mühendisliği**

### **BMÜ329 VERİ TABANI SİSTEMLERİ**

Proje Adı: Tarım Kooperatifleri Veri Tabanı

Proje Ekibi:

- 225260050 - Ömer Faruk Mutlu
- 225260066 - Yasin Mert Acar

#### **1-Proje Gereksinimleri**

Bu proje, kooperatiflerin üyeleri, tedarikçileri, ürünleri ve sipariş süreçlerini yönetmek amacıyla geliştirilmiş bir veritabanı modelini içermektedir. Sistem, kooperatiflerin mali verilerini, ürün fiyatlandırmalarını, üyelerin verdikleri siparişleri, tedarikçi ilişkilerini ve diğer birçok işlevi takip etmeye yönelik olarak tasarlanmıştır.

## 2-Proje Özeti: Tarım Kooperatif Yönetim Sistemi

" Tarım Kooperatif Yönetim Sistemi" projesi, kooperatiflerin üyeleri, tedarikçileri, ürünleri ve sipariş süreçlerini modern bir şekilde yönetmelerini amaçlamaktadır. Proje, kullanıcı dostu bir arayüz ve güçlü bir ilişkisel veritabanı modeli ile kooperatiflerin işleyişini iyileştirmeyi hedefler. Sistem, mali işlemlerden etkinlik yönetimine kadar çeşitli iş süreçlerini kapsayarak operasyonel verimliliği artırır. Kooperatifler, gelir-gider takibi, ürün fiyatlandırma, stok yönetimi, sipariş kaydı, tedarikçi ilişkileri ve nakil işlemlerini organize edebilir. Ayrıca, sosyal sorumluluk süreçlerini de entegre ederek kapsamlı bir yönetim platformu sunar.

## 3-Proje Amaçları:

- Kooperatiflerin Operasyonel Verimliliğini Artırmak: Üyeler, tedarikçiler ve ürünler arasında bilgi akışını kolaylaştırarak kooperatiflerin iş süreçlerini daha etkili hale getirmek.
- Stok ve Sipariş Yönetimini İyileştirmek: Kooperatiflerin ürün stoklarını ve sipariş süreçlerini güvenli, hızlı ve sistematik bir şekilde yönetmelerini sağlamak.
- Maliyet ve Gelir Yönetimini Desteklemek: Gelir-gider takibini ve maliyet hesaplamalarını doğru ve şeffaf bir şekilde yaparak kooperatiflerin finansal performanslarını artırmak.
- Üye ve Tedarikçi İlişkilerini Güçlendirmek: Üye siparişlerini ve tedarikçi anlaşmalarını düzenli bir şekilde yöneterek taraflar arasındaki etkileşimi optimize etmek.
- Modern ve Kullanıcı Dostu Çözümler Sunmak: Tüm süreçlerin kolayca anlaşılabilir ve uygulanabilir bir platform üzerinden yürütülmesini sağlayarak zaman ve emekten tasarruf etmek.

#### 4-Proje Süreci:

- Kooperatif Veritabanının Tasarımı ve İlişkisel Modellerin Oluşturulması: Kooperatiflerin üyeleri, tedarikçileri, ürünleri ve finansal verilerini yönetmek için detaylı bir veritabanı yapısı geliştirilmesi.
- Üye ve Sipariş Yönetimi İçin Veri Modellerinin Geliştirilmesi: Üye kayıtlarını, sipariş süreçlerini ve tedarikçi ilişkilerini sistematik bir şekilde takip edebilecek bir mimari oluşturulması.
- Stok ve Fiyatlandırma Sisteminin Entegrasyonu: Kooperatif ürünlerinin stok durumunu ve fiyatlandırma verilerini tutacak yapının tasarlanması ve işlevsel hale getirilmesi.
- Kullanıcı Dostu Arayüz Geliştirme: Veri tabanı ile entegre, kooperatiflerin günlük işlemlerini kolayca yönetebilecekleri bir yazılım arayüzünün tasarlanması.
- Test ve İyileştirme Aşamaları: Geliştirilen sistemin kullanıcılar tarafından test edilmesi, alınan geri bildirimlere göre iyileştirme ve geliştirme süreçlerinin gerçekleştirilmesi.

#### 5-Kullanılan Teknolojiler:

- SQL Server Management Studio 2022: Kooperatiflerin verilerini güvenli ve etkili bir biçimde yönetmek amacıyla güçlü bir veritabanı yönetim aracı olarak kullanılacaktır.
- İlişkisel Veritabanı Modelleri: Üyeler, tedarikçiler, ürünler ve finansal veriler arasındaki karmaşık ilişkileri düzenlemek ve yönetmek için optimize edilmiş veri modelleri geliştirilecektir

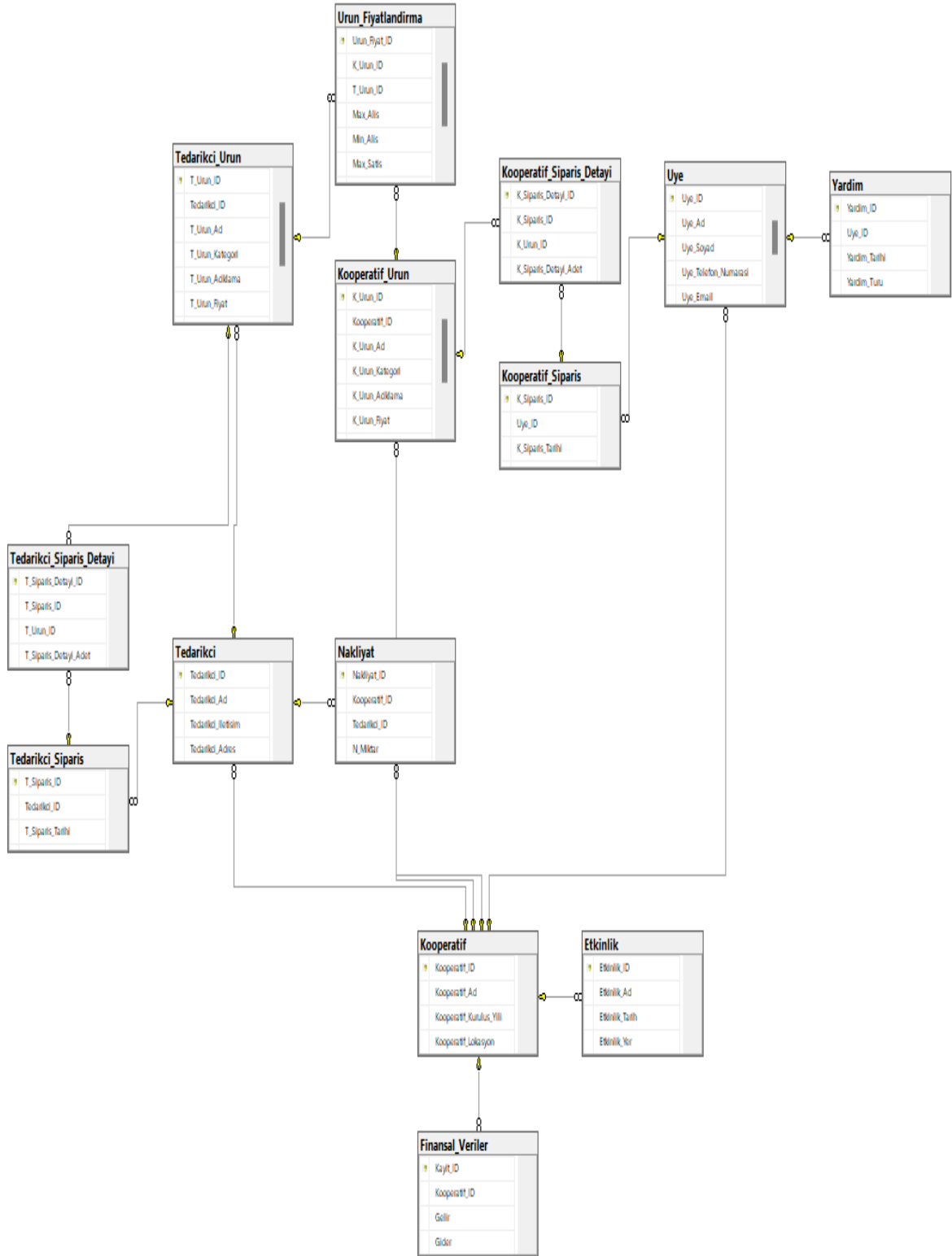
## 6-Proje Sonuçları:

- Proje tamamlandığında, kooperatifler ve üyeleri aşağıdaki faydaları elde edecektir:
- Etkin Stok Yönetimi: Ürün stokları daha düzenli ve etkili bir şekilde izlenecektir.
- Sistematik Üye ve Sipariş Yönetimi: Üye bilgileri, siparişler ve tedarikçi ilişkileri güvenli ve düzenli bir biçimde yönetilecektir.
- Doğru ve Kapsamlı Finansal İzleme: Kooperatiflerin gelir, gider ve fiyatlandırma işlemleri daha hızlı ve doğru bir şekilde gerçekleştirilecektir.
- Artırılmış Operasyonel Verimlilik: Kooperatiflerin günlük iş süreçleri daha kolay ve etkili bir şekilde yürütülerek karar alma süreçleri hızlanacaktır.
- Gelişmiş Rekabet Avantajı: Modern bir veri yönetim sistemi sayesinde kooperatiflerin rekabet gücü artacak ve sürdürülebilir bir yapı kazanacaktır.

## 7-Sonuç:

"Kooperatif Yönetim Sistemi" projesi, kooperatiflerin geleneksel yönetim yöntemlerinden modern, veri tabanı tabanlı bir sisteme geçişini sağlamayı hedeflemektedir. Proje ekibi, işbirliği içinde çalışarak etkili ve verimli iş süreçleri oluşturmayı amaçlamaktadır. Projenin uygulanması, kooperatiflerin

## 8-UML Diyagramı:



## İlişkiler

### Kooperatif - Üye

- **\*\*İlişki Türü:\*\*** 1 Kooperatif  $\Leftrightarrow$  N Üye
- **\*\*Açıklama:\*\*** Bir kooperatif birden fazla üye barındırabilir, ancak her üye yalnızca bir kooperatifle ilişkilidir.

### Kooperatif - Kooperatif\_Ürün

- **\*\*İlişki Türü:\*\*** 1 Kooperatif  $\Leftrightarrow$  N Kooperatif\_Ürün
- **\*\*Açıklama:\*\*** Bir kooperatif birçok ürün sunabilir.

### Kooperatif - Tedarikçi

- **\*\*İlişki Türü:\*\*** 1 Kooperatif  $\Leftrightarrow$  N Tedarikçi
- **\*\*Açıklama:\*\*** Bir kooperatif, birden fazla tedarikçiden ürün alabilir.

### Kooperatif\_Ürün - Ürün\_Fiyatlandırma

- **\*\*İlişki Türü:\*\*** 1 Kooperatif\_Ürün  $\Leftrightarrow$  1 Ürün\_Fiyatlandırma
- **\*\*Açıklama:\*\*** Her kooperatif ürününün belirli bir fiyatlandırma kaydı vardır.

### Tedarikçi - Tedarikçi\_Ürün

- **\*\*İlişki Türü:\*\*** 1 Tedarikçi  $\Leftrightarrow$  N Tedarikçi\_Ürün
- **\*\*Açıklama:\*\*** Her tedarikçi birden fazla ürünü tedarik edebilir.

### Tedarikçi\_Ürün - Ürün\_Fiyatlandırma

- **\*\*İlişki Türü:\*\*** 1 Tedarikçi\_Ürün  $\Leftrightarrow$  1 Ürün\_Fiyatlandırma
- **\*\*Açıklama:\*\*** Tedarikçi ürünlerinin de belirli bir fiyatlandırma kaydı vardır.

### Üye - Kooperatif\_Sipariş

- **\*\*İlişki Türü:\*\*** 1 Üye  $\Leftrightarrow$  N Kooperatif\_Sipariş
- **\*\*Açıklama:\*\*** Her üye birden fazla sipariş verebilir.

### Kooperatif\_Sipariş - Kooperatif\_Sipariş\_Detayı

- **\*\*İlişki Türü:\*\*** 1 Kooperatif\_Sipariş  $\Leftrightarrow$  N Kooperatif\_Sipariş\_Detayı
- **\*\*Açıklama:\*\*** Bir sipariş birden fazla ürün içerebilir.

### Tedarikçi\_Sipariş - Tedarikçi\_Sipariş\_Detayı

- **\*\*İlişki Türü:\*\*** 1 Tedarikçi\_Sipariş  $\Leftrightarrow$  N Tedarikçi\_Sipariş\_Detayı
- **\*\*Açıklama:\*\*** Bir tedarikçi siparişi birden fazla ürün içerebilir.

### Kooperatif - Finansal\_Veriler

- **\*\*İlişki Türü:\*\*** 1 Kooperatif  $\Leftrightarrow$  N Finansal\_Veriler

- **Açıklama:** Her kooperatif birden fazla finansal kayıt tutar.

#### Üye - Yardım

- **İlişki Türü:** 1 Üye  $\Leftrightarrow$  N Yardım
- **Açıklama:** Her üye birden fazla yardım alabilir.

#### Kooperatif - Etkinlik

- **İlişki Türü:** 1 Kooperatif  $\Leftrightarrow$  N Etkinlik
- **Açıklama:** Her kooperatif birden fazla etkinlik düzenleyebilir.

#### Kooperatif - Nakil

- **İlişki Türü:** 1 Kooperatif  $\Leftrightarrow$  N Nakil
- **Açıklama:** Bir kooperatif birden fazla nakil işlemi gerçekleştirebilir.

#### Tedarikçi - Nakil

- **İlişki Türü:** 1 Tedarikçi  $\Leftrightarrow$  N Nakil
- **Açıklama:** Her tedarikçi birden fazla nakil işlemi yapabilir.

#### Tedarikçi - Tedarikçi\_Sipariş



- **\*\*İlişki Türü:\*\*** 1 Tedarikçi  $\Leftrightarrow$  N Tedarikçi\_Sipariş
- **\*\*Açıklama:\*\*** Bir tedarikçi birden fazla sipariş alabilir.

## 9-SQL Kodları:

```
CREATE TABLE Kooperatif (  
    Kooperatif_ID INT PRIMARY KEY,  
    Kooperatif_Ad NVARCHAR(100) NOT NULL,  
    Kooperatif_Kurulus_Yili INT NOT NULL,  
    Kooperatif_Lokasyon NVARCHAR(255) NOT NULL,  
    Kooperatif_Uye_Sayisi INT  
);
```

```
CREATE TABLE Uye (  
    Uye_ID INT PRIMARY KEY,  
    Uye_Ad NVARCHAR(100) NOT NULL,  
    Uye_Soyad NVARCHAR(100) NOT NULL,  
    Uye_Telefon_Numarasi NVARCHAR(15),  
    Uye_Email NVARCHAR(100),  
    Kooperatif_ID INT,  
    FOREIGN KEY (Kooperatif_ID) REFERENCES  
Kooperatif(Kooperatif_ID)  
);
```

```
CREATE TABLE Kooperatif_Urun (  
    K_Urun_ID INT PRIMARY KEY,  
    Kooperatif_ID INT,  
    K_Urun_Ad NVARCHAR(100) NOT NULL,  
    K_Urun_Kategori NVARCHAR(50),  
    K_Urun_Aciklama NVARCHAR(255),  
    K_Urun_Fiyat DECIMAL(10, 2) NOT NULL,  
    K_Urun_Stok_Miktari INT NOT NULL,  
    Urun_Fiyat_ID INT,  
    FOREIGN KEY (Kooperatif_ID) REFERENCES  
Kooperatif(Kooperatif_ID)  
);
```

```
CREATE TABLE Tedarikci (  
    Tedarikci_ID INT PRIMARY KEY,  
    Tedarikci_Ad NVARCHAR(100) NOT NULL,  
    Tedarikci_Iletisim NVARCHAR(255),  
    Tedarikci_Adres NVARCHAR(255),  
    Kooperatif_ID INT,  
    FOREIGN KEY (Kooperatif_ID) REFERENCES  
Kooperatif(Kooperatif_ID)  
);
```

```
CREATE TABLE Tedarikci_Urun (  
    T_Urun_ID INT PRIMARY KEY,  
    Tedarikci_ID INT,  
    T_Urun_Ad NVARCHAR(100) NOT NULL,  
    T_Urun_Kategori NVARCHAR(50),  
    T_Urun_Aciklama NVARCHAR(255),  
    T_Urun_Fiyat DECIMAL(10, 2) NOT NULL,  
    T_Urun_Miktar INT NOT NULL,  
    Urun_Fiyat_ID INT,  
    FOREIGN KEY (Tedarikci_ID) REFERENCES  
Tedarikci(Tedarikci_ID)  
);
```

```
CREATE TABLE Kooperatif_Siparis (  
    K_Siparis_ID INT PRIMARY KEY,  
    Uye_ID INT,  
    K_Siparis_Tarihi DATE NOT NULL,  
    K_Toplam_Tutar DECIMAL(10, 2) NOT NULL,  
    FOREIGN KEY (Uye_ID) REFERENCES Uye(Uye_ID)  
);
```

```
CREATE TABLE Kooperatif_Siparis_Detayi (  
    K_Siparis_Detayi_ID INT PRIMARY KEY,  
    K_Siparis_ID INT,  
    K_Urun_ID INT,  
    K_Siparis_Detayi_Adet INT NOT NULL,  
    K_Siparis_Detayi_Fiyat DECIMAL(10, 2) NOT NULL,
```

```
        FOREIGN KEY (K_Siparis_ID) REFERENCES
Kooperatif_Siparis(K_Siparis_ID),
        FOREIGN KEY (K_Urun_ID) REFERENCES
Kooperatif_Urun(K_Urun_ID)
);
```

```
CREATE TABLE Tedarikci_Siparis (
    T_Siparis_ID INT PRIMARY KEY,
    Tedarikci_ID INT,
    T_Siparis_Tarihi DATE NOT NULL,
    T_Toplam_Tutar DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (Tedarikci_ID) REFERENCES
Tedarikci(Tedarikci_ID)
);
```

```
CREATE TABLE Tedarikci_Siparis_Detayi (
    T_Siparis_Detayi_ID INT PRIMARY KEY,
    T_Siparis_ID INT,
    T_Urun_ID INT,
    T_Siparis_Detayi_Adet INT NOT NULL,
    T_Siparis_Detayi_Fiyat DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (T_Siparis_ID) REFERENCES
Tedarikci_Siparis(T_Siparis_ID),
    FOREIGN KEY (T_Urun_ID) REFERENCES
Tedarikci_Urun(T_Urun_ID)
);
```

```
CREATE TABLE Finansal_Veriler (
    Kayit_ID INT PRIMARY KEY,
    Kooperatif_ID INT,
    Gelir DECIMAL(10, 2),
    Gider DECIMAL(10, 2),
    Finansal_Tarih NVARCHAR(50),
    FOREIGN KEY (Kooperatif_ID) REFERENCES
Kooperatif(Kooperatif_ID)
);
```

```
CREATE TABLE Yardim (
    Yardim_ID INT PRIMARY KEY,
```

```

    Uye_ID INT,
    Yardim_Tarihi DATE NOT NULL,
    Yardim_Turu NVARCHAR(100),
    Yardim_Miktari DECIMAL(10, 2),
    FOREIGN KEY (Uye_ID) REFERENCES Uye(Uye_ID)
);

```

```

CREATE TABLE Etkinlik (
    Etkinlik_ID INT PRIMARY KEY,
    Etkinlik_Ad NVARCHAR(100) NOT NULL,
    Etkinlik_Tarih DATE NOT NULL,
    Etkinlik_Yer NVARCHAR(255),
    Kooperatif_ID INT,
    FOREIGN KEY (Kooperatif_ID) REFERENCES
Kooperatif(Kooperatif_ID)
);

```

```

CREATE TABLE Urun_Fiyatlandirma (
    Urun_Fiyat_ID INT PRIMARY KEY,
    K_Urun_ID INT,
    T_Urun_ID INT,
    Max_Alis DECIMAL(10, 2),
    Min_Alis DECIMAL(10, 2),
    Max_Satis DECIMAL(10, 2),
    Min_Satis DECIMAL(10, 2),
    Gecerlilik_Tarihi DATE,
    FOREIGN KEY (K_Urun_ID) REFERENCES
Kooperatif_Urun(K_Urun_ID),
    FOREIGN KEY (T_Urun_ID) REFERENCES
Tedarikci_Urun(T_Urun_ID)
);

```

```

CREATE TABLE Nakliyat (
    Nakliyat_ID INT PRIMARY KEY,
    Kooperatif_ID INT,
    Tedarikci_ID INT,
    N_Miktar INT NOT NULL,
    N_Tarih DATE NOT NULL,

```

```
        FOREIGN KEY (Kooperatif_ID) REFERENCES
Kooperatif(Kooperatif_ID),
        FOREIGN KEY (Tedarikci_ID) REFERENCES
Tedarikci(Tedarikci_ID)
);
```

```
INSERT INTO Kooperatif (Kooperatif_ID, Kooperatif_Ad,
Kooperatif_Kurulus_Yili, Kooperatif_Lokasyon,
Kooperatif_Uye_Sayisi)
VALUES
```

```
(1, 'Sinop Kooperatifi', 2003, 'Sinop', 157),
(2, 'Mutlu Tarım', 2000, 'Malatya', 100),
(3, 'Bereket Kooperatifi', 1995, 'Elazığ', 200),
(4, 'Ayas Ticaret', 2015, 'Van', 80),
(5, 'Büyük Üreticiler', 1973, 'Rize', 20);
```

```
INSERT INTO Uye (Uye_ID, Uye_Ad, Uye_Soyad,
Uye_Telefon_Numarasi, Uye_Email, Kooperatif_ID)
VALUES
```

```
(1, 'Yasin Mert', 'Acar', '05432433517',
'acaryasinmert@gmail.com', 1),
(2, 'Ömer Faruk', 'Mutlu', '05489855463',
'mutluomerfaruk@gmail.com', 2),
(3, 'Onur', 'Bilici', '05232232323',
'bilicionur@gmail.com', 3),
(4, 'Ali', 'Ayas', '05676676766', 'ayasali@hotmail.com',
4),
(5, 'Emir', 'Bacanak', '05512345678',
'bacanakemir@hotmail.com', 5),
(6, 'Enes', 'Gökkaya', '05432433518',
'gokkayaenes@gmail.com', 1),
(7, 'Salih', 'Üstün', '05987456321',
'ustunsalih@hotmail.com', 2);
```

```
INSERT INTO Kooperatif_Urun (K_Urun_ID, Kooperatif_ID,
K_Urun_Ad, K_Urun_Kategori, K_Urun_Aciklama, K_Urun_Fiyat,
K_Urun_Stok_Miktari)
```

VALUES

```
(1, 1, 'Kestane', 'Meyve', 'Sinop Kestanesi', 650.00, 1500),
(2, 2, 'Kayısı', 'Meyve', 'Malatya Kayısı', 180.00, 20000),
(3, 3, 'Tereyağ', 'Süt Ürünleri', 'Organik Elazığ Tereyağı', 900.00, 750),
(4, 4, 'Otlı Peynir', 'Süt Ürünleri', 'Van Otlı Peyniri', 750.00, 600),
(5, 5, 'Çay', 'Çay Ürünleri', 'Rize Çayı', 800.00, 5000);
```

INSERT INTO Tedarikci (Tedarikci\_ID, Tedarikci\_Ad, Tedarikci\_Iletisim, Tedarikci\_Adres, Kooperatif\_ID)

VALUES

```
(1, 'Anadolu Tedarik', 'info@anadolutedarik.com', 'Konya', 1),
(2, 'Sivas Tarım', 'sivastarım@hotmail.com', 'Sivas', 2),
(3, 'Kardeşler Tedarik', 'crazyboy@gmail.com', 'Antalya', 3),
(4, 'Organik Tarım ve Tedarik', 'bilgi@organiktarım.com', 'Muğla', 4),
(5, 'Sinop Ürünleri', 'sinoplum@gmail.com', 'Sinop', 5);
```

INSERT INTO Tedarikci\_Urun (T\_Urun\_ID, Tedarikci\_ID, T\_Urun\_Ad, T\_Urun\_Kategori, T\_Urun\_Aciklama, T\_Urun\_Fiyat, T\_Urun\_Miktar)

VALUES

```
(1, 1, 'Un', 'Bakliyat', 'Esmer Un', 4.00, 500),
(2, 2, 'Sucuk', 'Et Ürünleri', 'Kangal Sucuk', 450.00, 300),
(3, 3, 'Muz', 'Meyve', 'Antalya Muz', 17.00, 2000),
(4, 4, 'Zeytin', 'Meyve', 'Siyah zeytin', 57.00, 4000),
(5, 5, 'Pirinç', 'Bakliyat', 'Baldo Pirinç', 10.00, 6000);
```

INSERT INTO Kooperatif\_Siparis (K\_Siparis\_ID, Uye\_ID, K\_Siparis\_Tarihi, K\_Toplam\_Tutar)

VALUES

```
(1, 1, '2026-01-01', 100.50),
(2, 2, '2026-01-02', 250.75),
```

```
(3, 3, '2026-01-03', 300.00),  
(4, 4, '2026-01-04', 50.00),  
(5, 5, '2026-01-05', 120.25);
```

```
INSERT INTO Kooperatif_Siparis_Detayi  
(K_Siparis_Detayi_ID, K_Siparis_ID, K_Urun_ID,  
K_Siparis_Detayi_Adet, K_Siparis_Detayi_Fiyat)  
VALUES
```

```
(1, 1, 1, 10, 3.50),  
(2, 2, 2, 20, 2.00),  
(3, 3, 3, 5, 20.00),  
(4, 4, 4, 2, 50.00),  
(5, 5, 5, 15, 10.00);
```

```
INSERT INTO Tedarikci_Siparis (T_Siparis_ID, Tedarikci_ID,  
T_Siparis_Tarihi, T_Toplam_Tutar)  
VALUES
```

```
(1, 1, '2026-01-01', 500.00),  
(2, 2, '2026-02-01', 1200.00),  
(3, 3, '2026-03-01', 300.00),  
(4, 4, '2026-04-01', 800.00),  
(5, 5, '2026-05-01', 400.00);
```

```
INSERT INTO Tedarikci_Siparis_Detayi (T_Siparis_Detayi_ID,  
T_Siparis_ID, T_Urun_ID, T_Siparis_Detayi_Adet,  
T_Siparis_Detayi_Fiyat)  
VALUES
```

```
(1, 1, 1, 50, 4.00),  
(2, 2, 2, 100, 6.00),  
(3, 3, 3, 20, 25.00),  
(4, 4, 4, 30, 12.00),  
(5, 5, 5, 40, 10.00);
```

```
INSERT INTO Finansal_Veriler (Kayit_ID, Kooperatif_ID,  
Gelir, Gider, Finansal_Tarih)  
VALUES
```

```
(1, 1, 5000.00, 2000.00, '2025-01'),  
(2, 2, 10000.00, 8000.00, '2025-02'),
```

```
(3, 3, 7000.00, 3000.00, '2025-03'),  
(4, 4, 2000.00, 1500.00, '2025-04'),  
(5, 5, 3000.00, 1000.00, '2025-05');
```

```
INSERT INTO Yardım (Yardım_ID, Uye_ID, Yardım_Tarihi,  
Yardım_Turu, Yardım_Miktari)  
VALUES
```

```
(1, 1, '2025-01-01', 'Nakit', 500.00),  
(2, 2, '2025-01-02', 'Burs', 4000.00),  
(3, 3, '2025-01-03', 'Gıda', 200.00),  
(4, 4, '2025-01-04', 'Eğitim', 300.00),  
(5, 5, '2025-01-05', 'Sağlık', 400.00);
```

```
INSERT INTO Etkinlik (Etkinlik_ID, Etkinlik_Ad,  
Etkinlik_Tarih, Etkinlik_Yer, Kooperatif_ID)  
VALUES
```

```
(1, 'Balıkesir Şenliği', '2026-06-01', 'Balıkesir', 1),  
(2, 'Kooperatif Buluşması', '2026-07-15', 'İzmir', 2),  
(3, 'Çiftçi Günü', '2026-08-20', 'Konya', 3),  
(4, 'Tarımsal Eğitim', '2026-09-10', 'Antalya', 4),  
(5, 'Köy Ürünleri Fuarı', '2026-10-05', 'Kahramanmaraş',  
5);
```

```
INSERT INTO Urun_Fiyatlandirma (Urun_Fiyat_ID, K_Urun_ID,  
T_Urun_ID, Max_Alis, Min_Alis, Max_Satis, Min_Satis,  
Gecerlilik_Tarihi)
```

```
VALUES
```

```
(1, 1, 1, 3.50, 3.00, 4.00, 3.50, '2026-01-01'),  
(2, 2, 2, 2.00, 1.80, 2.50, 2.00, '2026-02-01'),  
(3, 3, 3, 20.00, 18.00, 22.00, 20.00, '2026-03-01'),  
(4, 4, 4, 50.00, 45.00, 55.00, 50.00, '2026-04-01'),  
(5, 5, 5, 10.00, 9.00, 12.00, 10.00, '2026-05-01');
```

```
INSERT INTO Nakliyat (Nakliyat_ID, Kooperatif_ID,  
Tedarikci_ID, N_Miktar, N_Tarih)
```

```
VALUES
```

```
(1, 1, 1, 500, '2026-01-01'),  
(2, 2, 2, 400, '2026-02-01'),
```



```
(3, 3, 3, 300, '2026-03-01'),  
(4, 4, 4, 200, '2026-04-01'),  
(5, 5, 5, 100, '2026-05-01');
```

### Procedure Örneği:

```
CREATE PROCEDURE SiparisStokGuncelle (  
    @K_Urun_ID INT,  
    @Adet INT  
)  
AS  
BEGIN  
    UPDATE Kooperatif_Urun  
    SET K_Urun_Stok_Miktari = K_Urun_Stok_Miktari - @Adet  
    WHERE K_Urun_ID = @K_Urun_ID;  
END;
```

Kullanımı:

```
EXEC SiparisStokGuncelle @K_Urun_ID = 1, @Adet = 5;
```

### Transaction Örneği :

```
BEGIN TRANSACTION;  
  
BEGIN TRY  
    INSERT INTO Kooperatif_Siparis (K_Siparis_ID, Uye_ID,  
K_Siparis_Tarihi, K_Toplam_Tutar)  
    VALUES (101, 1, GETDATE(), 500.00);  
  
    INSERT INTO Kooperatif_Siparis_Detayi  
(K_Siparis_Detayi_ID, K_Siparis_ID, K_Urun_ID,  
K_Siparis_Detayi_Adet, K_Siparis_Detayi_Fiyat)  
    VALUES (201, 101, 1, 10, 50.00);  
  
    UPDATE Kooperatif_Urun  
    SET K_Urun_Stok_Miktari = K_Urun_Stok_Miktari - 10  
    WHERE K_Urun_ID = 1;
```

```

        IF EXISTS (SELECT * FROM Kooperatif_Urun WHERE
K_Urun_Stok_Miktari < 0)
        BEGIN
            THROW 50001, 'Stok yetersiz! İşlem iptal
ediliyor.', 1;
        END;

        COMMIT TRANSACTION;
        PRINT 'Sipariş başarıyla eklendi ve stok
güncellendi.';
    END TRY
    BEGIN CATCH
        ROLLBACK TRANSACTION;
        PRINT 'Hata oluştu: ' + ERROR_MESSAGE();
    END CATCH;

```

## Trigger Örneği:

### 1.ÖRNEK

```

CREATE TRIGGER Stok_Kontrol
ON Kooperatif_Siparis_Detayi
AFTER INSERT
AS
BEGIN
    SET NOCOUNT ON;
    DECLARE @Urun_ID INT, @Siparis_Adet INT;

    SELECT @Urun_ID = K_Urun_ID, @Siparis_Adet =
K_Siparis_Detayi_Adet
    FROM inserted;

    IF (SELECT K_Urun_Stok_Miktari FROM Kooperatif_Urun
WHERE K_Urun_ID = @Urun_ID) < @Siparis_Adet
    BEGIN

        ROLLBACK TRANSACTION;
        RAISERROR('Stok miktarı yetersiz, işlem iptal
edildi.', 16, 1);
        RETURN;
    END

```

```

    UPDATE Kooperatif_Urun
    SET K_Urun_Stok_Miktari = K_Urun_Stok_Miktari -
    @Siparis_Adet
    WHERE K_Urun_ID = @Urun_ID;
END;

```

## 2.ÖRNEK

```

CREATE TRIGGER Fiyat_Degisim_Finansal_Guncelle
ON Kooperatif_Urun
AFTER UPDATE
AS
BEGIN
    SET NOCOUNT ON;

    IF (UPDATE(K_Urun_Fiyat))
    BEGIN
        DECLARE @Kooperatif_ID INT, @Eski_Fiyat
        DECIMAL(10, 2), @Yeni_Fiyat DECIMAL(10, 2), @Degisim
        DECIMAL(10, 2);

        SELECT
            @Kooperatif_ID = I.Kooperatif_ID,
            @Eski_Fiyat = D.K_Urun_Fiyat,
            @Yeni_Fiyat = I.K_Urun_Fiyat
        FROM
            inserted I
        INNER JOIN
            deleted D ON I.K_Urun_ID = D.K_Urun_ID;

        SET @Degisim = @Yeni_Fiyat - @Eski_Fiyat;

        IF @Degisim > 0
        BEGIN
            UPDATE Finansal_Veriler
            SET Gelir = ISNULL(Gelir, 0) + @Degisim
            WHERE Kooperatif_ID = @Kooperatif_ID;
        END
    ELSE
    BEGIN

```

```

        UPDATE Finansal_Veriler
        SET Gider = ISNULL(Gider, 0) + ABS(@Degisim)
        WHERE Kooperatif_ID = @Kooperatif_ID;
    END
END
END;

```

## 10-Örnek Sorgu Kodları:

a)Her Kooperatifin Toplam Gelir ve Gider Farkını Getiren Sorgu

```

SELECT
    K.Kooperatif_Ad,
    SUM(FV.Gelir) AS ToplamGelir,
    SUM(FV.Gider) AS ToplamGider,
    (SUM(FV.Gelir) - SUM(FV.Gider)) AS KarZarar
FROM
    Kooperatif AS K
LEFT JOIN
    Finansal_Veriler AS FV
ON
    K.Kooperatif_ID = FV.Kooperatif_ID
GROUP BY
    K.Kooperatif_Ad;

```

b) Stok Miktarı Kritik Seviyede Olan Ürünlerin Listesi

```

SELECT
    KU.K_Urun_Ad AS UrunAdi,
    KU.K_Urun_Stok_Miktari AS StokMiktari,
    K.Kooperatif_Ad AS KooperatifAdi
FROM
    Kooperatif_Urun AS KU
INNER JOIN
    Kooperatif AS K
ON
    KU.Kooperatif_ID = K.Kooperatif_ID
WHERE
    KU.K_Urun_Stok_Miktari < 1000;

```

c) En Çok Satış Yapılan Ürünün Bilgisi

```

SELECT TOP 1
    KU.K_Urun_Ad AS UrunAdi,
    SUM(KSD.K_Siparis_Detayi_Adet) AS ToplamSatis,

```

```

        K.Kooperatif_Ad AS KooperatifAdi
FROM
        Kooperatif_Siparis_Detayi AS KSD
INNER JOIN
        Kooperatif_Urun AS KU
ON
        KSD.K_Urun_ID = KU.K_Urun_ID
INNER JOIN
        Kooperatif AS K
ON
        KU.Kooperatif_ID = K.Kooperatif_ID
GROUP BY
        KU.K_Urun_Ad, K.Kooperatif_Ad
ORDER BY
        SUM(KSD.K_Siparis_Detayi_Adet) DESC;

```

d) Tedarikçi Bazında Alım ve Satış Fiyatlarının Analizi

```

SELECT
        T.Tedarikci_Ad AS TedarikciAdi,
        TU.T_Urun_Ad AS UrunAdi,
        MAX(UF.Max_Alis) AS MaxAlisFiyati,
        MIN(UF.Min_Satis) AS MinSatisFiyati
FROM
        Tedarikci AS T
INNER JOIN
        Tedarikci_Urun AS TU
ON
        T.Tedarikci_ID = TU.Tedarikci_ID
INNER JOIN
        Urun_Fiyatlandirma AS UF
ON
        TU.T_Urun_ID = UF.T_Urun_ID
GROUP BY
        T.Tedarikci_Ad, TU.T_Urun_Ad;

```