

Endüstriyel Otomasyon ve Haberleşme Ürünleri Yönetim Sistemi

1. Cihaz ve Kategori Yönetimi

- Categories (Kategoriler) -

- Cihazların ait olduğu ana kategorileri tanımlamak için kullanılır. Her kategori aşağıdaki bilgileri içerir:

- **CategoryID:** Benzersiz kimlik numarası
- **Adı:** Kategorinin adı
- **Tanımı:** İsteğe bağlı açıklama bilgisi

- DeviceSeries (Cihaz Serileri) -

- Cihazlar, bir kategoriye ait belirli serilere ayrılır. Bu tabloda bulunur:

- **SeriesID:** Serinin benzersiz kimliği
- **Adı:** Seri adı
- **Açıklama:** Seriye dair açıklama bilgileri
- **CategoryID:** Kategorilerle bağlantılıdır - Yabancı Anahtar
-

- Devices (Cihazlar) -

- Sistemde yer alan tüm cihazları tanımlar. Her cihaz için:

- **CihazKodu :** Birincil Anahtar
- **SeriID:** İlgili seriyle ilişkili
- **Adı:** Cihazın adı
- **Özellikleri:** Teknik ve donanımsal özellikler
- **Üretim Tarihi:** Cihazın üretim tarihi
- **Açıklama:** Detaylı açıklama bilgileri

2. Kullanıcı ve Destek Yönetimi

- Users (Kullanıcılar) -

- Sistemdeki tüm kullanıcıları temsil eder. Kullanıcı bilgileri:

- **UserID:** Birincil Anahtar
- **Kullanıcı Adı:** Kullanıcının sisteme kayıtlı adı
- **E-posta:** Kullanıcının e-posta adresi
- **Şifre:** Sisteme erişim için kullanılan şifre
- **Son Giriş Tarihi:** Sisteme en son erişim tarihi

- CustomerSupport (Müşteri Destek) -

Kullanıcıların sistemde oluşturduğu destek kayıtları. Her kayıt içerir:

- **DestekID: Birincil Anahtar**
- **UserID:** Destek talebini oluşturan kullanıcı - Yabancı Anahtar
- **Destek Tarihi:** Talebin tarihi
- **Talep Açıklaması:** Destek isteğine dair detaylar

- CustomerSupportTechnician (Destek Teknisyenleri) -

Destek taleplerini yöneten teknisyenlerin bilgilerini içerir:

- **SupportID: Birincil Anahtar**
- **Teknisyen Adı:** Teknisyenin adı
- **Pozisyonu:** Teknisyenin pozisyonu
- **Tecrübesi:** Teknisyenin deneyim detayları

3. Cihaz Teknik ve Donanım Yetenekleri

- DeviceFeatures (Cihaz Özellikleri) -

Cihazların yazılım ve donanım yeteneklerini belirtir:

- **CihazKodu** : Cihazlar Tablosuyla Bağlantılıdır. - Yabancı Anahtar
- **IEC 61131-3 Standardına Uygunluk**
- **Gerçek Zamanlı Olarak Kontrol Edilebilirlik**
- **ÇevrimDışı Simülasyonlarda Kullanılabilirlik İmkamı**
- **USB Bağlantısıyla Cihaz Kullanımı**
- **TCP/IP ile Program Yükleme / İndirme**
- **Yazılımın ilgili Şifreyle Korunması**
- **Kullanıcı Bilgisayarının IP Koruması**
- **IoT Desteği**
- **MODBUS Modülünde GATEWAY Kullanımı**
- **Kontrol Blok Kütüphanesiyle Eksen Kontrolü Sağlanması**
- **Kontrol Blok Kütüphanesiyle PID Modülünün Ayarlanması**
- **Kontrol Blok Kütüphanesiyle Matematiksel İşlemlerde Float Düzeyinin Ayarlanılabilirliği**
- **Kontrol Blok Kütüphanesiyle Zamanlayıcı Değerlerinin Kontrolü**
- **Blokların Makro Düzeyde Ayarlanabilirliği**

- MotionControlFunctions (Hareket Kontrol Fonksiyonları) -

Cihazların hareket kontrol yeteneklerini içerir:

- **Cihaz Kodu** : Cihazlar Tablosuyla Bağlantılıdır. - Yabancı Anahtar
- **Hızlı Sayıcı Girişi**
- **Darba Genişliği ve Modülasyon Çıkışı (PWM Çıkışı)**
- **Darbe Tren Çıkışı (PTO)**

- Programming (Programlama) -

Cihazların desteklediği yazılım özellikleri:

- **Cihaz Kodu** : Cihazlar Tablosuyla Bağlantılıdır. - Yabancı Anahtar
- **Programlama Dilleri**
- **Editör Türleri**
- **USB Port Bilgileri**

- Environmental Conditions (Çevre Koşulları) -

Cihazların desteklendiği uygun çalışma koşulları :

- **Cihaz Kodu** : Cihazlar Tablosuyla Bağlantılıdır. - Yabancı Anahtar
- **Çalışma Sıcaklığı**
- **Cihazın Depolanma / Saklanma Sıcaklığı**
- **Cihaz Kullanımında Uygun Nem Değeri**
- **Cihazın Korunma Standartları**
- **Cihazın Çalışma İrtifası**

- **Card Type (Kart Tipi) -**

- **Cihaz Kodu** : Cihazlar Tablosuyla Bağlantılıdır. - Yabancı Anahtar
- **Dijital Giriş**
- **Dijital Çıkış**
- **Analog Giriş**

- **LED Signal (Led Sinyaller)-**

- **Cihaz Kodu** : Cihazlar Tablosuyla Bağlantılıdır. - Yabancı Anahtar
- **Kullanılan Led Sinyalin Beslediği Güç Seviyesi**
- **Led Sinyalin Çalışma Durumu**
- **Transmitter Haberleşme - Vericiyle Haberleşme Kullanımı (TX)**
- **Receiver Haberleşme - Alıcıyla Haberleşme Kullanımı (RX)**
- **Dijital Girişler**
- **Dijital Çıkışlar**

- **Electrical Wiring -**

- **Cihaz Kodu** : Cihazlar Tablosuyla Bağlantılıdır. - Yabancı Anahtar
- **Kablolama Yapımında Input ve Output Birimlerinin Durumu**
- **Kablolamada gerekli elektriğin cihaz ile etkileşimi (Güç Beslenme Girişi)**

- **Keeping Real-Time Clock Usage in a Calendar (Gerçek Zaman Saat Takvimi) :**

Gerçek Zamanlı olarak çalışan cihazın zaman ayarları kendisine özgüdür.

- **Cihaz Kodu** : Cihazlar Tablosuyla Bağlantılıdır. - Yabancı Anahtar
- **Gerçek Zaman Saati Verileri**
- **NTP (Network Time Protocol) client desteğiyle birlikte cihazla etkileşimli birimlerin aynı saat biriminde çalışabilmesi sağlanır.**
- **Cihaz (Modbus master) gerçek zamanlı saat bilgilerini (RTC) alarak bu bilgileri bağlı diğer cihazlara (Modbus slave) gönderebilir.**
- **Pil Durumu Seviyesi**
- **RTC Pil Kullanımı , bir cihazın gerçek zamanlı saati (RTC) doğru bir şekilde çalıştırabilmesi için gerekli olan pil veya bataryadır.**
- **Cihazlar arasındaki zaman kayması kontrolü**

4.Haberleşme ve Protokol Yönetimi

- **CommunicationProtocols (Haberleşme Protokolleri) -**

Cihazların desteklediği haberleşme protokolleri:

- **Cihaz Kodu** : Cihazlar Tablosuyla Bağlantılıdır. - Yabancı Anahtar
- **MasterHaberleşme Protokolleri**
- **SlaveHaberleşmeProtokolleri**
- **IoT Protokolleri**

- **CommunicationPorts (Haberleşme Portları) -**

Cihazların bağlantı port bilgileri:

- **Cihaz Kodu** : Cihazlar Tablosuyla Bağlantılıdır. - Yabancı Anahtar
- **Ethernet Portları**
- **Entegre Edilen GSM Haberleşme Portları**

- Special GSM Functions (Özel GSM Fonksiyonları) -

Cihazların GSM ile ilgili özellikleri:

- **Cihaz Kodu** : Cihazlar Tablosuyla Bağlantılıdır. - Yabancı Anahtar
- **SMS Oluşturma/Gönderme**
- **SMS Mesajlarını Alma /Ayrıştırma**
- **DTMF Çağrılarını Kabul Etme / Kod Gönderme Fonksiyonları**
- **DTMF Çağrılarını Kabul Etme / Kodlarını Ayrıştırma Fonksiyonları**

5. Garanti ve Sertifikasyon

- Warranty_Certification (Garanti ve Sertifikasyon) -

Cihazların garanti ve sertifikasyon bilgileri:

- **Cihaz Kodu** : Cihazlar Tablosuyla Bağlantılıdır. - Yabancı Anahtar
- **Garanti Süresi**
- **Sertifika**

6. Sipariş Yönetimi

Orders (Siparişler)

Kullanıcıların verdiği sipariş bilgileri:

- **Sipariş Durumu**
- **Toplam Tutar**
- **Oluşturulma Tarihi**

OrderDetails (Sipariş Detayları)

Siparişlerde yer alan cihazların bilgileri:

- **Cihaz Adedi**
- **Sipariş ile Cihazların Bağlantısı**

7. Çözüm ve Otomasyon Yönetimi

- Solutions (Çözümler) -

Cihazlarla ilgili geliştirilen çözümleri içerir:

- **Cihaz İlişkisi**
- **Oluşturulma Tarihi**

Veritabanı İlişkileri ve Normalizasyon

1-N (Çoktan Bire) İlişkiler

- **Categories - DeviceSeries**: Her kategori birden fazla cihaz serisi içerebilir.
Anahtarlar: Categories (1) → DeviceSeries (N)
- **DeviceSeries - Devices**: Her cihaz serisi birden fazla cihazı kapsar.
Anahtarlar: DeviceSeries (1) → Devices (N)
- **Users - Orders**: Her kullanıcı birden fazla sipariş verebilir.
Anahtarlar: Users (1) → Orders (N)
- **Users - CustomerSupport**: Her kullanıcı birden fazla destek talebi oluşturabilir.

Anahtarlar: Users (1) → CustomerSupport (N)

- **Orders - OrderDetails:** Bir sipariş birden fazla sipariş detayı içerebilir.

Anahtarlar: Orders (1) → OrderDetails (N)

- **Devices - OrderDetails:** Her cihaz birden fazla sipariş detayında yer alabilir.

Anahtarlar: Devices (1) → OrderDetails (N)

- **CustomerSupport - CustomerSupportTechnician:** Bir destek talebi birden fazla teknisyen tarafından işlenebilir.

Anahtarlar: CustomerSupport (1) → CustomerSupportTechnician (N)

N-N (Çoktan Çoğa) İlişkiler

- **Devices ↔ Orders (OrderDetails aracılığıyla):**

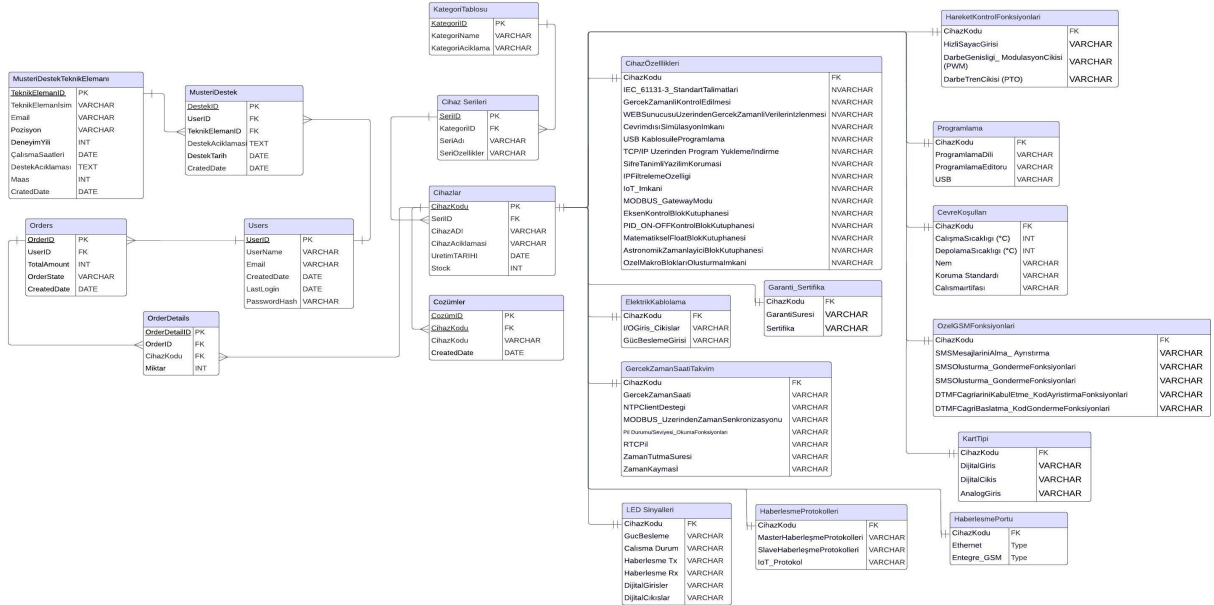
- **Detaylar:**

- Bir cihaz birden fazla siparişin parçası olabilir (birden fazla müşteri aynı cihazı sipariş edebilir).
- Bir sipariş birden fazla cihazı içerebilir (bir müşteri tek bir siparişte birden fazla cihaz alabilir).

- **Anahtarlar:**

- Devices (N) → OrderDetails → Orders (N)

E-R Diyagramı:



3NF ve BCNF İNCELEMESİ

Anahtar Tablolar

1. Categories

- CategoryIDbirincil anahtar.
- Geçişli bağımlılık bulunmuyor. BCNF ve 3NF'e uygundur.

2. DeviceSeries

- SeriesIDbirincil anahtar.
- CategoryID, Categoristablosuna dış anahtar olarak bağlı.
- Geçişli bağımlılık yok. BCNF ve 3NF'e uygundur.

3. Devices

- DeviceCodebirincil anahtar.

- SeriesID, DeviceSeri tablosuna ikincil anahtar olarak bağlı.
- Özellikler (DeviceName, ProductionDate, Stock) yalnızca birincil anahtara bağlı. BCNF ve 3NF'e uygundur.

Bağımlılık Analizleri

- Örneğin, DeviceFeatures tablosunda:
 - DeviceCode birincil anahtar.
 - Diğer kolonlar (IEC_61131_3_StandardInstructions, RealTimeControlvs.) yalnızca DeviceCode'a bağlı.
 - Geçişli bağımlılık bulunmuyor. 3NF ve BCNF'e uygun.
- Programming tablosunda:
 - DeviceCode birincil anahtar.
 - ProgrammingLanguage, ProgrammingEditor ve USBPortType yalnızca DeviceCode'a bağlı.
 - BCNF ve 3NF'e uygun.

Müşteri ve Sipariş Tabloları

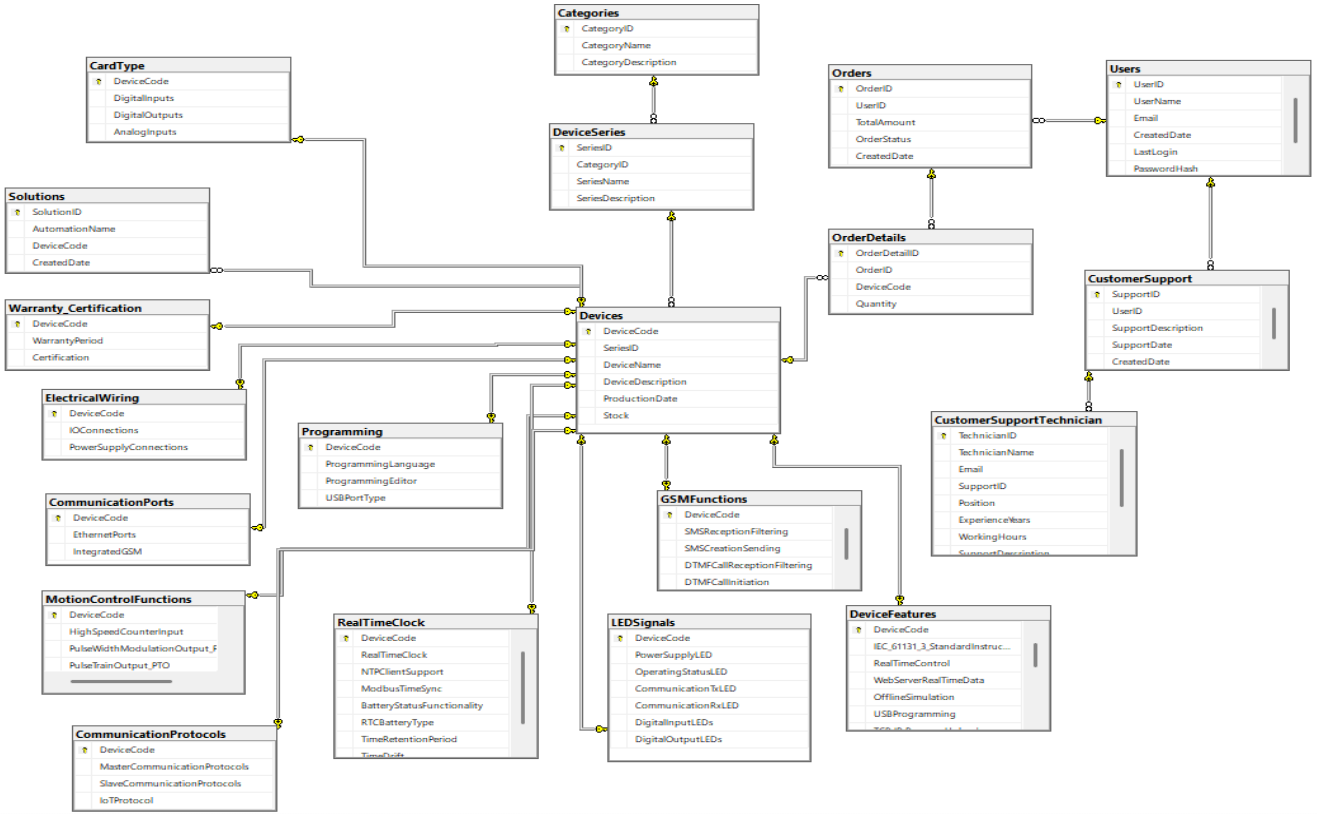
- **Users**
 - UserID birincil anahtar.
 - Geçişli bağımlılık bulunmuyor. BCNF ve 3NF'e uygun.
- **Orders**
 - OrderID birincil anahtar.
 - UserID ikincil anahtar olarak tanımlı.
 - Tabloda Geçişli bağımlılık bulunmuyor.
- **OrderDetails**
 - OrderDetailID birincil anahtar.
 - OrderID ve DeviceCode, ikincil anahtar olarak tanımlanmış.
 - BCNF ve 3NF'e uygun.

CustomerSupport ve Technician

- **CustomerSupport**
 - SupportID birincil anahtar.
 - UserID ikincil anahtar.
 - Geçişli bağımlılık yok. BCNF ve 3NF'e uygun.
- **CustomerSupportTechnician**
 - TechnicianID birincil anahtar.
 - SupportID ikincil anahtar.
 - Geçişli bağımlılık gözüküyor.

Sonuç olarak burdaki tabloların hepsi yukarıda gösterdiğimiz gibi BCNF ve 3NF formatına uygundur.

İlişkisel Veri Modeli



Veri Tabanında Kullanılan Komutlar

CREATE KOMUTLARI:

```
CREATE TABLE Categories (
  CategoryID INT PRIMARY KEY IDENTITY(1,1),
  CategoryName NVARCHAR(100) NOT NULL,
  CategoryDescription NVARCHAR(MAX),
);
```

```
CREATE TABLE DeviceSeries (
  SeriesID INT PRIMARY KEY IDENTITY(1,1),
  CategoryID INT,
  FOREIGN KEY (CategoryID) REFERENCES Categories(CategoryID),
  SeriesName NVARCHAR(100) NOT NULL,
  SeriesDescription NVARCHAR(MAX),
);
```

```
CREATE TABLE Devices (
  DeviceCode INT PRIMARY KEY,
  SeriesID INT NOT NULL,
  DeviceName VARCHAR(255) NOT NULL,
  DeviceFeatures VARCHAR(500),
  DeviceDescription VARCHAR(500),
  ProductionDate DATE,
```



```
FOREIGN KEY (SeriesID) REFERENCES DeviceSeries(SeriesID)
);
```

```
CREATE TABLE Users
( UserID INT PRIMARY KEY,
  UserName VARCHAR(255) NOT NULL, Email
  VARCHAR(255) NOT NULL,
  CreatedDate DATE, LastLogin
  DATE,
  PasswordHash VARCHAR(255) NOT NULL
);
```

```
CREATE TABLE Orders ( OrderID
INT PRIMARY KEY, UserID INT,
TotalAmount INT, OrderStatus
VARCHAR(255), CreatedDate
DATE,
FOREIGN KEY (UserID) REFERENCES Users(UserID)
);
```

INSERT KOMUTLARI:

```
INSERT INTO Categories (CategoryName, CategoryDescription) VALUES
('Otomasyon Sistemleri (DCS)', 'Endüstriyel süreçlerin otomatik kontrolü için dağıtık
çözümler.'),
('Programlanabilir Kontrol Üniteleri (PLC)', 'Endüstriyel uygulamalarda esnek ve verimli
kontrol sağlayan PLC tabanlı sistemler.'),
('Nesnelerin İnterneti (IoT) Bağlantı Çözümleri', 'Farklı IoT cihazları arasında güvenli ve
verimli iletişim sağlayan protokoller ve arayüzler.'),
('Süreç Görselleştirme ve Kontrol Sistemleri (SCADA)', 'Endüstriyel süreçlerin gerçek zamanlı
izlenmesi ve yönetimi için SCADA platformları.'),
('Ek Fonksiyon Modülleri', 'Mevcut sistemlerin kapasitesini artırmak ve yeni özellikler eklemek
için modüler çözümler.');
```

```
INSERT INTO DeviceSeries(CategoryID, SeriesName, SeriesDescription) VALUES (1,
'DM100 Serisi', 'Uzaktan erişim ve kontrol için ideal'),
(1, 'DM50 Serisi', 'Endüstriyel uygulamalar için uygun'), (1,
'DM500 Serisi', 'Rack tipi uzak terminal ünitesi'),
(1, 'RTU100 Serisi', 'Genel amaçlı uzaktan terminal ünitesi'),
(1, 'RTU300 Serisi', 'Yüksek performanslı uzaktan terminal ünitesi');
```

```
INSERT INTO DeviceSeries(CategoryID, SeriesName, SeriesDescription) VALUES (2,
'MP110 Serisi PLC', 'Mikro Düzey Kontrolör'),
(2, 'MP211 Serisi PLC', 'Kompakt Kontrolör');
```

```
INSERT INTO Devices (DeviceCode, SeriesID, DeviceName, DeviceDescription,
ProductionDate, Stock)
VALUES
```

```
('RTU301-E1L-GA1-B6001', 5, 'Cihaz 1', 'Açıklama A', '2025-01-01', 100),
('RTU301-E1L-GA1-B7001', 5, 'Cihaz 2', 'Açıklama B', '2025-01-02', 200),
('RTU301-E1L-GA2-B6001', 5, 'Cihaz 3', 'Açıklama C', '2025-01-03', 300),
('RTU301-E1L-GA2-B7001', 5, 'Cihaz 4', 'Açıklama D', '2025-01-04', 400);
```

```
INSERT INTO Orders (OrderID, UserID, TotalAmount, OrderStatus, CreatedDate) VALUES
(1, 4, 100, 'Tamamlandı', '2024-01-05'),
(2, 5, 150, 'Bekliyor', '2024-01-10'),
(3, 4, 1500, 'Tamamlandı', '2024-05-08'),
(4, 5, 2600, 'Tamamlandı', '2024-06-10');
```

STORED PROCEDURE (SAKLIYORDAM) KOMUTU

ManageDevices saklı yordamı, *Devices* tablosunda temel CRUD (Create, Read, Update, Delete) işlemlerini gerçekleştirmek için tasarlanmıştır. Saklı yordam, bir *@Action* parametresi olarak hangi işlemin yapılması gerektiğini belirler. Ayrıca, *@DeviceCode*, *@SeriesID*, *@DeviceName*, *@DeviceDescription*, *@ProductionDate*, ve *@Stock* gibi cihaz bilgilerini parametre olarak alır.

```
use ProductSystem;
go

CREATE PROCEDURE ManageDevices
    @Action NVARCHAR(10),
    @DeviceCode NVARCHAR(100) = NULL,
    @SeriesID INT = NULL,
    @DeviceName NVARCHAR(255) = NULL,
    @DeviceDescription NVARCHAR(500) = NULL,
    @ProductionDate DATE = NULL,
    @Stock INT = NULL
AS
BEGIN
    SET NOCOUNT ON;
    IF @Action = 'INSERT'
    BEGIN
        INSERT INTO Devices (DeviceCode, SeriesID, DeviceName, DeviceDescription,
ProductionDate, Stock)
        VALUES (@DeviceCode, @SeriesID, @DeviceName, @DeviceDescription, @ProductionDate,
@Stock);
    END

    ELSE IF @Action = 'UPDATE'
    BEGIN
        UPDATE Devices
        SET SeriesID = @SeriesID,
            DeviceName = @DeviceName,
            DeviceDescription = @DeviceDescription,
            ProductionDate = @ProductionDate,
            Stock = @Stock
        WHERE DeviceCode = @DeviceCode;
    END

    ELSE IF @Action = 'DELETE'
    BEGIN
        DELETE FROM Devices
        WHERE DeviceCode = @DeviceCode;
    END

    ELSE IF @Action = 'SELECT'
    BEGIN
        IF @DeviceCode IS NOT NULL
        BEGIN
            SELECT * FROM Devices
            WHERE DeviceCode = @DeviceCode;
        END
        ELSE
        BEGIN
            SELECT * FROM Devices;
        END
    END

    ELSE
    BEGIN
        PRINT 'Invalid Action. Use INSERT, UPDATE, DELETE, or SELECT.';
    END
END;

GO
```

```
EXEC ManageDevices
  @Action = 'INSERT',
  @DeviceCode = 'D12345',
  @SeriesID = 1,
  @DeviceName = 'Smart Device X',
  @DeviceDescription = 'A smart IoT-enabled device',
  @ProductionDate = '2025-01-04',
  @Stock = 100;
```

Kodun İşlevi: Yeni bir cihaz ekler (Devices tablosuna yeni bir satır ekler)

```
EXEC ManageDevices
  @Action = 'UPDATE',
  @DeviceCode = 'D12345',
  @SeriesID = 4,
  @DeviceName = 'Updated DeviceE X',
  @DeviceDescription = 'Updated description',
  @ProductionDate = '2025-01-05',
  @Stock = 120;
```

Kodun İşlevi: DeviceCode ile eşleşen cihazın bilgilerini günceller.

```
EXEC ManageDevices
  @Action = 'DELETE',
  @DeviceCode = 'D12345';
```

Kodun İşlevi: DeviceCode ile eşleşen cihazı tablodan siler.

```
SELECT ROUTINE_NAME
FROM INFORMATION_SCHEMA.ROUTINES
WHERE ROUTINE_TYPE = 'PROCEDURE';
```

```
SELECT name
FROM sys.procedures;
```

Kodun İşlevi: Veri tabanımızda bulunan tüm saklı yordamları listeler

```
DROP PROCEDURE prosedür_adı;
```

Kodun İşlevi: Veri tabanımızda bulunan saklı yordamı silmemezi sağlar.

TRANSACTION KOMUTU

```
use ProductSystem;
GO
BEGIN TRANSACTION;
BEGIN TRY
    INSERT INTO CustomerSupport (UserID, SupportDescription, SupportDate, CreatedDate)
    VALUES (4, 'Cihaz ağ bağlantı sorunu', GETDATE(), GETDATE());

    DECLARE @SupportID INT = SCOPE_IDENTITY();

    INSERT INTO CustomerSupportTechnician (TechnicianID, TechnicianName, Email, SupportID,
    Position, ExperienceYears, WorkingHours, SupportDescription, Salary, CreatedDate)
    VALUES (9, 'Ertan', 'ErtanB.eleman@example.com', @SupportID, 'Teknik Destek Uzmanı', 5,
    GETDATE(), 'Cihaz ağ sorunu incelenecek.', 5000, GETDATE());

    COMMIT;
    PRINT 'İşlem başarıyla tamamlandı.';
END TRY

BEGIN CATCH
    ROLLBACK;
    PRINT 'Hata oluştu. İşlem geri alındı.';
    PRINT ERROR_MESSAGE();
END CATCH;
```

CustomerSupport ve ona bağlı CustomerSupportTechnician kaydını güvenli bir şekilde eklemeyi sağlar. İlk adımda, müşteri destek talebi CustomerSupport tablosuna eklenir ve bu işlem sonucunda oluşturulan birincil anahtar (SupportID) değeri alınır. Ardından, bu SupportID kullanılarak ilgili teknik destek uzmanı bilgileri CustomerSupportTechnician tablosuna eklenir. Eğer herhangi bir aşamada hata oluşursa, transaction işlemi rollback ile yapılan tüm değişiklikler geri alınır ve veritabanında hiçbir değişiklik yapılmaz. Ancak tüm işlemler sorunsuz bir şekilde tamamlanırsa, transaction commit işlemi ile sonuçlandırılır ve eklenen veriler kalıcı hale getirilir.

TRIGGER KOMUTU -1

```
use ProductSystem;
GO
CREATE TRIGGER trg_StockUpdate
ON OrderDetails
AFTER INSERT, DELETE, UPDATE
AS
BEGIN
    IF EXISTS (SELECT 1 FROM inserted WHERE NOT EXISTS (SELECT 1 FROM deleted WHERE
inserted.OrderDetailID = deleted.OrderDetailID))
    BEGIN
        UPDATE Devices
        SET Stock = Stock - i.Quantity
        FROM Devices d
        INNER JOIN inserted i
        ON d.DeviceCode = i.DeviceCode;

        PRINT 'Stok azaltma işlemi gerçekleştirildi (INSERT).';
    END;

    IF EXISTS (SELECT 1 FROM deleted WHERE NOT EXISTS (SELECT 1 FROM inserted WHERE
inserted.OrderDetailID = deleted.OrderDetailID))
    BEGIN
        UPDATE Devices
        SET Stock = Stock + del.Quantity
        FROM Devices d
        INNER JOIN deleted del
        ON d.DeviceCode = del.DeviceCode;

        PRINT 'Stok artırma işlemi gerçekleştirildi (DELETE).';
    END;

    IF EXISTS (SELECT 1 FROM inserted INNER JOIN deleted ON inserted.OrderDetailID =
deleted.OrderDetailID)
    BEGIN
        UPDATE Devices
        SET Stock = Stock + del.Quantity - i.Quantity
        FROM Devices d
        INNER JOIN inserted i
        ON d.DeviceCode = i.DeviceCode
        INNER JOIN deleted del
        ON d.DeviceCode = del.DeviceCode;

        PRINT 'Stok güncelleme işlemi gerçekleştirildi (UPDATE).';
    END;
END;
```

Yukarıdaki trg_StockUpdate tetikleyicisi, **OrderDetails** tablosunda gerçekleşen INSERT, DELETE, ve UPDATE işlemleri sonrasında **Devices** tablosundaki stok miktarını güncellemek için tasarlanmıştır. Bu tetikleyici, hangi işlemin yapıldığına göre stok miktarını artırır, azaltır veya günceller.

```
UPDATE OrderDetails
SET Quantity = 15
WHERE OrderDetailID = 1;
```

Sipariş miktarı değiştirildiğinde (UPDATE işlemi), Devices tablosundaki ilgili cihazın stok miktarını, eski ve yeni miktar arasındaki farka göre günceller.

```
DELETE FROM OrderDetails
WHERE OrderDetailID = 14;
```

Sipariş detayı OrderDetails tablosundan silindiğinde, Devices tablosundaki ilgili cihazın stok miktarını, silinen sipariş miktarı kadar artırır.

```
INSERT INTO OrderDetails (OrderDetailID, OrderID, DeviceCode, Quantity)
VALUES (1, 101, 'D001', 5);
```

Yeni bir sipariş detayı OrderDetails tablosuna eklendiğinde, Devices tablosundaki ilgili cihazın stok miktarını, eklenen sipariş miktarı kadar azaltır.

TRIGGER KOMUTU -2

```
CREATE TRIGGER trg_InsertDefaultDeviceFeatures
ON Devices
AFTER INSERT
AS
BEGIN
    INSERT INTO DeviceFeatures (
        DeviceCode,
        IEC_61131_3_StandardInstructions,
        RealTimeControl,
        WebServerRealTimeData,
        OfflineSimulation,
        USBProgramming,
        TCP_IP_ProgramUpload,
        EncryptedSoftwareProtection,
        IPFiltering,
        IoT_Feature,
        MODBUS_Gateway,
        AxialControlBlockLibrary,
        PID_ON_OFFBlockLibrary,
        MathematicalFloatBlockLibrary,
        AstronomicalTimeBlockLibrary,
        CustomMacroBlockLibrary
    )
    SELECT
        i.DeviceCode,
        'No' ,
        'No' ,
        'No' ,
        'No' ,
        'No' ,
        'No' ,
        'No' ,
        'No' ,
        'No' ,
        'No' ,
        'No' ,
        'No' ,
        'No' ,
        'No' ,
        'No'
    FROM
        inserted AS i;
END;
GO
```

Devices tablosuna yeni bir cihaz (kayıt) eklendiğinde, bu cihazın teknik özelliklerini varsayılan değerlerle otomatik olarak DeviceFeatures tablosuna eklemektir.

```
INSERT INTO Devices (
    DeviceCode,
    SeriesID,
    DeviceName,
    DeviceDescription,
    ProductionDate,
    Stock
)
VALUES (
    'DEV001' ,
    1, -- SeriesID
    'Example Device', -- DeviceName
    'This is a test device.', -- DeviceDescription
    '2025-01-05' , -- ProductionDate
    100 -- Stock
);
```

Devices Tablosuna yeni bir cihaz eklendiğinde, trg_InsertDefaultDeviceFeatures tetikleyicisi devreye girer. DeviceCode değerine göre eklenen cihazın DeviceCode değeri otomatik olarak DeviceFeatures tablosuna varsayılan değerlerle birlikte eklenir.