

# Maden Şantiyesi Yönetim Sistemi Raporu

## Proje Gereksinimleri:

*Bu proje, iş yerinde çalışanların vardiya ve malzeme takibini dijital bir sistemle kolaylaştırmayı amaçlamaktadır. Çalışanların puantajları, sipariş işlemleri, malzeme stok durumu ve vardiya bilgileri gibi süreçlerin dijital ortamda izlenebilmesi hedeflenmiştir. Böylece işletme içindeki operasyonel süreçler hem daha düzenli olacak hem de bilgiye erişim daha kolay hale gelecektir.*

## Varlıklar ve Nitelikler:

*Proje kapsamında veritabanında yer alacak ana varlıklar ve bu varlıklara ait nitelikler şöyle tanımlanmıştır:*

## Varlıklar ve Nitelikler:

### 1. Çalışan:

*İş yerinde çalışan kişilere ait bilgileri içerir.*

#### Temel Bilgiler:

*id(PK),*

*ad,*

*soyad,*

*tc\_no,*

*sicil\_no*

#### İletişim ve Demografik Bilgiler:

*telefon,*

*e\_posta,*

*adres,*

*dogum\_tarihi*

***İş Bilgileri: mezuniyet, maas, meslek, gorev, gorev\_2, giris\_tarihi, cikis\_tarihi***

***SGK ve Kurum Bilgileri: sgk\_isyeri\_sicil\_no, iskol\_u, biriken\_izin, kurum***

***Çalışan Türü: tur alanı ile çalışanlar "yönetici" veya "işçi" olarak ayrılmaktadır.***

## **2.Yöneticiler:**

***Yöneticilere ait özel giriş kartı bilgilerini tutar.***

***Nitelikler: kart\_id(PK),***

***calisan\_id (FK),***

***kart\_no,***

***kart\_turu,***

***verilme\_tarihi***

## **3.İşçiler:**

***İşçilerin kullandığı malzemelerin takibini sağlar.***

***Nitelikler: malzeme\_id(PK),***

***calisan\_id (Çalışan tablosuyla ilişkili,FK),***

***malzeme\_adi,***

***miktar***

## **4. Puantaj:**

***id(PK),***

***calisan\_id (Çalışan ile ilişki,FK),***

***tarih,***

***mesai\_tipi,***

*mesai\_saat,*  
*fazla\_mesai,*  
*vardiya\_id(FK)*  
*aciklama.*

## 5. Vardiya:

*id(PK),*  
*tarih,*  
*vardiyaBaslangic,*  
*vardiyaBitis,*  
*calisan\_id(FK).*

*Çalışma alanı bilgileri: lokasyonAdi, kuyuNumarasi, egim,*  
*makinaNo, seriNo.*

*Diğer detaylar: toplamIlerleme, toplamKarot, matkapNo,*  
*havaDurumu, aciklama.*

## 6. Malzeme:

*malzeme\_id(PK),*  
*malzeme\_adi,*  
*stok\_miktar.*  
*aciklama.*  
*siparis\_no(FK).*

## 7. Hafif Malzemeler:

*Malzeme tablosundan türetilmiş olup, hafif malzemelere ait bilgileri içerir.*

*Nitelikler: malzeme\_id(PK),  
malzeme\_adi,  
stok\_miktar,  
aciklama*

## **8.Ağır Malzemeler:**

*Malzeme tablosundan türetilmiş olup, ağır malzemelere ait bilgileri içerir.*

*Nitelikler: malzeme\_id(PK),  
malzeme\_adi,  
stok\_miktar,  
aciklama*

## **9.Siparişler:**

*Temel bilgiler: siparis\_no(PK),  
kategori,  
malzeme(FK),  
firma(FK),  
adet\_miktar.  
Ek bilgiler: firma\_adi, aciliyet, tarih, onay.  
calisan\_id(FK).*

## **Varlıklar Arasındaki İlişkiler ve Kısıtlamalar:**

**Projede varlıklar arasında çeşitli ilişkiler ve kısıtlamalar bulunmaktadır. Bu ilişkiler sayesinde sistemdeki verilerin birbiriyle uyumlu ve doğru şekilde yönetilmesi sağlanır.**

## **Çalışan - Vardiya İlişkisi:**

**Bir alıřanın birden fazla vardiyası olabilir, fakat her vardiyada yalnızca bir alıřan yer alabilir. (1) ----> (N) iliřkisi vardır.**

### **alıřan – Puantaj İliřkisi:**

**Bir alıřanın birden fazla puantaj kaydı olabilir. (1) ----> (N) iliřkisi vardır.**

### **Vardiya – Malzeme İliřkisi:**

**Bir vardiyada birden fazla malzeme kullanılabilir, ancak her malzeme yalnızca bir vardiyaaya ait olur. (1) ----> (N) iliřkisi vardır.**

### **Malzeme – Sipariřler İliřkisi:**

**Bir malzeme birden fazla sipariřle iliřkili olabilir, ancak her sipariř tek bir malzemeye ait olur. iliřkisi vardır. (1)---->(N) iliřkisi vardır.**

### **alıřan – Sipariřler İliřkisi:**

**Bir alıřan birden fazla sipariř oluřturabilir, fakat her sipariř tek bir alıřan tarafından oluřturulmalıdır. (1) ----> (N) iliřkisi vardır.**

### **Hafif ve Ağır Malzemeler-Malzemeler İliřkisi:**

**Hafif ve ağır malzemeler, malzeme tablosunda farklı özellikler ve kullanım alanlarına göre ayrılır. Malzeme tablosunda, her bir malzeme türü için ağırlık, dayanıklılık, maliyet ve kullanım amacına göre sınıflandırmalar bulunur.**

**Malzeme tablosunda bu özelliklerin her biri belirli kriterler altında gösterilir. Hafif ve ağır malzemelerin seçimi projeye özel gereksinimlere göre yapılır.**

### **İř Kuralları ve Kısıtlamalar:**

**Bu projede verilerin tutarlılıđını sađlamak için belirli iş kuralları ve kısıtlamalar bulunmaktadır: • alıřan TC Kimlik Numarası benzersiz olmalı. • Malzeme stok miktarı, sıfırın altına düşmemeli. Stok**

yetersiz kaldığında uyarı sistemi devreye girebilir. • Sipariş aciliyet durumu, "Yüksek", "Orta" veya "Düşük" gibi belirli kategorilerde sınıflandırılmalı. • Puantaj tarihleri, çalışanın ilgili vardiya tarihleriyle uyumlu olmalı. • Hafif ve ağır malzemeler, malzeme türlerine göre ayrılmalı, böylece malzeme yönetimi daha etkin yapılmalıdır.

## Kullanıcı Türleri ve Gereksinimleri:

Projede üç temel kullanıcı türü bulunmaktadır ve her birinin farklı gereksinimleri vardır:

1. **Yönetici:** Tüm çalışan ve vardiya bilgilerini görebilir ve düzenleyebilir, malzeme ve sipariş durumunu kontrol edebilir, siparişleri onaylayabilir.
2. **Çalışan:** Kendi vardiya ve puantaj bilgilerini görüntüleyebilir, kendi adına sipariş oluşturabilir.
3. **Depo Görevlisi:** Malzeme stok durumunu takip eder, stok güncellemelerini yapar.

## Raporlama ve İzleme:

*Bu sistem, işletmenin operasyonel süreçlerini daha etkili bir şekilde yönetebilmesi için çeşitli raporlama ve izleme özellikleri de içermelidir: • Vardiya Raporları: Vardiya başına görevli çalışanlar ve yapılan işler raporlanmalı. • Malzeme Stok Raporu: Stok durumu periyodik olarak raporlanmalı ve eksik malzemeler hakkında bilgilendirme yapılmalı. Sipariş Durumu Raporu: Siparişlerin onay durumu, teslim edilme durumu gibi bilgiler raporlanmalıdır. Hafif ve Ağır Malzemeler Raporu: Malzemelerin türlerine göre stok durumları ve kullanımları düzenli olarak raporlanmalıdır.*

*Bu gereksinimler doğrultusunda, sistemin tasarlanması ve geliştirilmesi projenin hedeflerine ulaşması için büyük önem taşır. Sistemin kapsamlı, kullanıcı dostu ve iş akışlarına uygun olması için bu gereksinimlere uyulmalıdır.*

# Normalizasyon Süreci

## 1NF Ayırıştırma:

*Tablodaki bütün kolonlar atomik (tekil değerler içeriyor), dolayısıyla 1NF koşulu sağlanmıştır.*

## 2NF Ayırıştırma:

*Vardiya tablosundaki tüm kolonlar, birincil anahtar olan vardiya\_id ile tam bağımlıdır ve kısmi bağımlılık yoktur. Bu nedenle, 2NF koşulu sağlanmıştır.*

*Malzeme tablosundaki kolonlar malzeme\_id ile doğrudan bağımlıdır. Kısmi bağımlılık olmadığından dolayı 2NF sağlanmıştır.*

## 3NF Ayırıştırma:

*Transitif bağımlılıkların olduğu tablolar belirlenmiş ve ayrıştırılmıştır:*

*Çalışan tablosu, transitif bağımlılığı ortadan kaldırmak için iki tabloya ayrıştırılır:*

*Çalışanlar (id(PK), ad, soyad, meslek, görevi, giriş\_tarihi, çıkış\_tarihi, tur)*

*ÇalışanBilgileri (id(PK), doğum\_tarihi, telefon, adres, mezuniyet, sgk\_no, birim\_izin, kurum)*

*Malzeme tablosu, malzemelerin ağırlıklarına göre iki tabloya ayrılmıştır:*

*Ağır Malzemeler (malzeme\_id, malzeme\_adi, stok\_miktar, açıklama)*

*Hafif Malzemeler (malzeme\_id, malzeme\_adi, stok\_miktar, açıklama)*

## BCNF Ayırıştırma:

***Tüm tablolar incelenmiş ve her determinant bir aday anahtarı olduğundan dolayı BCNF koşulu sağlanmıştır.***

## **Normalizasyon Sonrası İlişkiler**

***Çalışanlar 1:N Vardiyalar***

***(Bir çalışan birden fazla vardiyada çalışabilir.)***

***Çalışanlar 1:1 ÇalışanBilgileri***

***(Her çalışanın detaylı bilgileri ayrı bir tabloda tutulur.)***

***Vardiyalar 1:N Malzemeler***

***(Bir vardiya birden fazla malzeme kullanabilir.)***

***Malzemeler 1:1 Ağır Malzemeler ve 1:1 Hafif Malzemeler***

***(Her malzeme ya ağır ya da hafif olarak sınıflandırılır.)***

***Malzemeler 1:N Siparişler***

***(Bir malzeme birden fazla siparişte yer alabilir.)***

***Siparişler N:1 Çalışanlar***

***(Bir siparişi bir çalışan oluşturur.)***

***Çalışanlar 1:N Yöneticiler***

***(Her yönetici bir çalışandır.)***



## ***Çalışanlar 1:N İşçiler***

***(Her işçi de bir çalışandır.)***

### **Veritabanı Şemalarının Oluşturulması**

```
-- Tablo Oluşturma Komutları
-- Calisan Tablosu
CREATE TABLE Calisan (
    id INT PRIMARY KEY IDENTITY(1,1),
    ad NVARCHAR(50),
    soyad NVARCHAR(50),
    tc_no NVARCHAR(11),
    dogum_tarihi DATE,
    telefon NVARCHAR(15),
    adres NVARCHAR(100),
    mezuniyet NVARCHAR(50),
    meslek NVARCHAR(50),
    gorev NVARCHAR(50),
    aciklama NVARCHAR(MAX),
    giris_tarihi DATE,
    cikis_tarihi DATE,
    birim_izin INT,
    kurum NVARCHAR(50),
    isveren_id INT,
    sgk_isyeri_sicil_no NVARCHAR(20),
    istihdam NVARCHAR(50),
    tur NVARCHAR(10) CHECK (tur IN ('yonetici', 'isci'))
);
GO
CREATE TABLE Malzeme (
    malzeme_id INT PRIMARY KEY IDENTITY(1,1),
    malzeme_adi NVARCHAR(100),
    stok_miktar INT,
    aciklama NVARCHAR(MAX)
);
GO
-- Vardiya Tablosu
CREATE TABLE Vardiya (
    id INT PRIMARY KEY IDENTITY(1,1),
    vardiyaBaslangic DATETIME,
    vardiyaBitis DATETIME,
    lokasyonAdi NVARCHAR(100),
    kokuNumarasi NVARCHAR(50),
    egitim NVARCHAR(100),
    makineNo INT,
    seriNo NVARCHAR(50),
    toplamMeleme INT,
    toplamKarte INT,
    malzeme_id INT,
    sondaj NVARCHAR(50),
    yardimci NVARCHAR(50),
    basMotorSaati DECIMAL(10, 2),
```

```
bitMotorSaati DECIMAL(10, 2),
motorYagi NVARCHAR(50),
hidrolikYagi NVARCHAR(50),
havaDurumu NVARCHAR(50),
aciklama NVARCHAR(MAX),
calisan_id INT,
FOREIGN KEY (malzeme_id) REFERENCES Malzeme(malzeme_id),
FOREIGN KEY (calisan_id) REFERENCES Calisan(id)
);
GO
```

-- Malzeme Tablosu

```
CREATE TABLE AgirMalzeme (
    malzeme_id INT PRIMARY KEY,
    stok_miktar INT,
    aciklama NVARCHAR(MAX),
    FOREIGN KEY (malzeme_id) REFERENCES Malzeme(malzeme_id)
);
GO
```

```
CREATE TABLE HafifMalzeme (
    malzeme_id INT PRIMARY KEY,
    stok_miktar INT,
    aciklama NVARCHAR(MAX),
    FOREIGN KEY (malzeme_id) REFERENCES Malzeme(malzeme_id)
);
GO
```

-- Siparisler Tablosu

```
CREATE TABLE Siparisler (
    siparis_no INT PRIMARY KEY IDENTITY(1,1),
    kategori NVARCHAR(50),
    malzeme_id INT,
    firma NVARCHAR(50),
    adet_miktar INT,
    aciliyet NVARCHAR(10) CHECK (aciliyet IN ('düşük', 'orta', 'yüksek')),
    tarih DATE,
    onay BIT,
    calisan_id INT,
    FOREIGN KEY (malzeme_id) REFERENCES Malzeme(malzeme_id),
    FOREIGN KEY (calisan_id) REFERENCES Calisan(id)
);
GO
```

-- Puan Tablosu

```
CREATE TABLE Puan (
    id INT PRIMARY KEY IDENTITY(1,1),
    calisan_id INT,
    tarih DATE,
    mesai_tipi NVARCHAR(50),
    mesai_saat INT,
    fazla_mesai INT,
```

```

        aciklama NVARCHAR(MAX),
        vardiya_id INT,
        FOREIGN KEY (calisan_id) REFERENCES Calisan(id),
        FOREIGN KEY (vardiya_id) REFERENCES Vardiya(id)
    );
GO

```

-- Yoneticiler ve Isciler Tabloları

```

CREATE TABLE Yoneticiler (
    kart_id INT PRIMARY KEY IDENTITY(1,1),
    calisan_id INT,
    kart_turu NVARCHAR(50),
    kart_no NVARCHAR(50),
    verilme_tarihi DATE,
    FOREIGN KEY (calisan_id) REFERENCES Calisan(id)
);
GO

```

```

CREATE TABLE Isciler (
    kart_id INT PRIMARY KEY IDENTITY(1,1),
    calisan_id INT,
    malzeme_adi NVARCHAR(50),
    miktar INT,
    FOREIGN KEY (calisan_id) REFERENCES Calisan(id)
);

```

## Veritabanına Insert Into Kullanarak Örnek Veriler Ekleme

-- Tablolara INSERT INTO ile Örnek Veri Ekleme

-- 1. Calisan Tablosuna Veri Ekleme

```

INSERT INTO Calisan (ad, soyad, tc_no, dogum_tarihi, telefon, adres,
mezuniyet, meslek, gorev, aciklama, giris_tarihi, cikis_tarihi, birim_izin,
kurum, isveren_id, sgk_isyeri_sicil_no, istihdam, tur)
VALUES ('Ali', 'Kara', '12345678901', '1990-01-01', '5551112233', 'Adres
1', 'Lisans', 'Makine Operatörü', 'Üretim', 'Açıklama 1', '2025-01-01',
NULL, 10, 'Kurum A', 1, '12345', 'Tam Zamanlı', 'isci');

```

```

INSERT INTO Calisan (ad, soyad, tc_no, dogum_tarihi, telefon, adres,
mezuniyet, meslek, gorev, aciklama, giris_tarihi, cikis_tarihi, birim_izin,
kurum, isveren_id, sgk_isyeri_sicil_no, istihdam, tur)
VALUES ('Ayşe', 'Yılmaz', '98765432109', '1992-02-02', '5552223344', 'Adres
2', 'Yüksek Lisans', 'Muhasebeci', 'Ofis', 'Açıklama 2', '2025-01-02',
NULL, 15, 'Kurum B', 2, '67890', 'Yarı Zamanlı', 'yonetici');

```

-- 2. Vardiya Tablosuna Veri Ekleme

```

INSERT INTO Vardiya (vardiyaBaslangic, vardiyaBitis, lokasyonAdi,
kokuNumarasi, egitim, makineNo, seriNo, toplamMeleme, toplamKarte,
malzeme_id, sondaj, yardimci, basMotorSaati, bitMotorSaati, motorYagi,
hidrolikYagi, havaDurumu, aciklama, calisan_id)

```

```
VALUES ('2025-01-05 08:00:00', '2025-01-05 16:00:00', 'Fabrika A', 'Koku-1', 'Eğitim A', 101, 'Seri-001', 10, 5, 1, 'Sondaj-1', 'Yardımcı A', 12.5, 15.0, '10W-40', 'Hidrolik-1', 'Güneşli', 'Vardiya açıklaması 1', 1);
```

```
INSERT INTO Vardiya (vardiyaBaslangic, vardiyaBitis, lokasyonAdi, kokuNumarasi, egitim, makineNo, seriNo, toplamMeleme, toplamKarte, malzeme_id, sondaj, yordamci, basMotorSaati, bitMotorSaati, motorYagi, hidrolikYagi, havaDurumu, aciklama, calisan_id)
VALUES ('2025-01-06 08:00:00', '2025-01-06 16:00:00', 'Fabrika B', 'Koku-2', 'Eğitim B', 102, 'Seri-002', 15, 8, 2, 'Sondaj-2', 'Yardımcı B', 13.0, 16.0, '15W-40', 'Hidrolik-2', 'Bulutlu', 'Vardiya açıklaması 2', 2);
```

#### -- 3. Malzeme Tablosuna Veri Ekleme

```
INSERT INTO Malzeme (malzeme_adi, stok_miktar, aciklama)
VALUES ('Çelik', 100, 'Yüksek kaliteli çelik malzeme');
```

```
INSERT INTO Malzeme (malzeme_adi, stok_miktar, aciklama)
VALUES ('Plastik', 200, 'Dayanıklı plastik malzeme');
```

#### -- 4. Siparisler Tablosuna Veri Ekleme

```
INSERT INTO Siparisler (kategori, malzeme_id, firma, adet_miktar, aciliyet, tarih, onay, calisan_id)
VALUES ('Hammadde', 1, 'Firma A', 50, 'orta', '2025-01-10', 1, 1);
```

```
INSERT INTO Siparisler (kategori, malzeme_id, firma, adet_miktar, aciliyet, tarih, onay, calisan_id)
VALUES ('Parça', 2, 'Firma B', 30, 'yüksek', '2025-01-12', 1, 2);
```

#### -- 5. Puan Tablosuna Veri Ekleme

```
INSERT INTO Puan (calisan_id, tarih, mesai_tipi, mesai_saat, fazla_mesai, aciklama, vardiya_id)
VALUES (1, '2025-01-05', 'Gündüz', 8, 2, 'Yoğun vardiya', 1);
```

```
INSERT INTO Puan (calisan_id, tarih, mesai_tipi, mesai_saat, fazla_mesai, aciklama, vardiya_id)
VALUES (2, '2025-01-06', 'Gece', 10, 3, 'Ek mesai', 2);
```

## Veritabanına Saklı Yordam Kullanarak Örnek Veriler Ekleme

#### -- Saklı Yordamlar Kullanarak Örnek Veri Ekleme

##### -- Yeni Çalışan ekleme

```
CREATE PROCEDURE CalisanEkle
    @ad NVARCHAR(50),
    @soyad NVARCHAR(50),
    @meslek NVARCHAR(50),
    @gorev NVARCHAR(50),
    @tur NVARCHAR(10)
AS
BEGIN
    INSERT INTO Calisan (ad, soyad, meslek, gorev, tur)
    VALUES (@ad, @soyad, @meslek, @gorev, @tur);
END;
```

```

GO
EXEC YeniCalisanEkle 'Ali', 'Kara', 'Makine Operatörü', 'Üretim', 'isci';
EXEC YeniCalisanEkle 'Ayşe', 'Yılmaz', 'Muhasebeci', 'Ofis', 'yonetici';
GO
-- Yeni Vardiya ekleme
CREATE PROCEDURE YeniVardiyaEkle
    @vardiyaBaslangic DATETIME,
    @vardiyaBitis DATETIME,
    @lokasyonAdi NVARCHAR(100),
    @calisan_id INT
AS
BEGIN
    INSERT INTO Vardiya (vardiyaBaslangic, vardiyaBitis, lokasyonAdi,
calisan_id)
    VALUES (@vardiyaBaslangic, @vardiyaBitis, @lokasyonAdi, @calisan_id);
END;
GO

EXEC YeniVardiyaEkle '2025-01-05 08:00:00', '2025-01-05 16:00:00', 'Fabrika
A', 1;
EXEC YeniVardiyaEkle '2025-01-06 08:00:00', '2025-01-06 16:00:00', 'Fabrika
B', 2;
GO
-- Yeni Malzeme ekleme
CREATE PROCEDURE YeniMalzemeEkle
    @malzeme_adi NVARCHAR(100),
    @stok_miktar INT,
    @aciklama NVARCHAR(MAX)
AS
BEGIN
    INSERT INTO Malzeme (malzeme_adi, stok_miktar, aciklama)
    VALUES (@malzeme_adi, @stok_miktar, @aciklama);
END;
GO

EXEC YeniMalzemeEkle 'Çelik', 100, 'Yüksek kaliteli çelik malzeme';
EXEC YeniMalzemeEkle 'Plastik', 200, 'Dayanıklı plastik malzeme';
GO
-- Yeni Sipariş ekleme
CREATE PROCEDURE YeniSiparisEkle
    @kategori NVARCHAR(50),
    @malzeme_id INT,
    @firma NVARCHAR(50),
    @adet_miktar INT,
    @aciliyet NVARCHAR(10),
    @tarih DATE,
    @onay BIT,
    @calisan_id INT
AS
BEGIN
    INSERT INTO Siparisler (kategori, malzeme_id, firma, adet_miktar,
aciliyet, tarih, onay, calisan_id)
    VALUES (@kategori, @malzeme_id, @firma, @adet_miktar, @aciliyet,
@tarih, @onay, @calisan_id);

```

```

END;
GO

EXEC YeniSiparisEkle 'Hammadde', 1, 'Firma A', 50, 'orta', '2025-01-10', 1,
1;
EXEC YeniSiparisEkle 'Parça', 2, 'Firma B', 30, 'yüksek', '2025-01-12', 1,
2;
GO
-- Yeni Puan ekleme
CREATE PROCEDURE YeniPuanEkle
    @calisan_id INT,
    @tarih DATE,
    @mesai_tipi NVARCHAR(50),
    @mesai_saat INT,
    @fazla_mesai INT,
    @aciklama NVARCHAR(MAX),
    @vardiya_id INT
AS
BEGIN
    INSERT INTO Puan (calisan_id, tarih, mesai_tipi, mesai_saat,
fazla_mesai, aciklama, vardiya_id)
VALUES (@calisan_id, @tarih, @mesai_tipi, @mesai_saat, @fazla_mesai,
@aciklama, @vardiya_id);
END;
GO

EXEC YeniPuanEkle 1, '2025-01-05', 'Gündüz', 8, 2, 'Yoğun vardiya', 1;
EXEC YeniPuanEkle 2, '2025-01-06', 'Gece', 10, 3, 'Ek mesai', 2;

```

## Procedure Kullanımı

```

-- Prosedür Örnekleri
-- 1. Yeni Çalışan Ekleme
CREATE PROCEDURE YeniCalisanEkle
    @ad NVARCHAR(50),
    @soyad NVARCHAR(50),
    @meslek NVARCHAR(50),
    @gorev NVARCHAR(50),
    @tur NVARCHAR(10)
AS
BEGIN
    INSERT INTO Calisan (ad, soyad, meslek, gorev, tur)
VALUES (@ad, @soyad, @meslek, @gorev, @tur);
END;
GO

-- 2. Vardiya Bilgilerini Getir
CREATE PROCEDURE VardiyaBilgileriGetir
AS
BEGIN
    SELECT * FROM Vardiya;
END;
GO

```

```

-- 3. Stok Güncelleme
CREATE PROCEDURE StokGuncelle
    @malzeme_id INT,
    @yeni_stok INT
AS
BEGIN
    UPDATE Malzeme
    SET stok_miktar = @yeni_stok
    WHERE malzeme_id = @malzeme_id;
END;
GO

-- 4. Sipariş Detaylarını Getir
CREATE PROCEDURE SiparisDetaylari
    @siparis_no INT
AS
BEGIN
    SELECT * FROM Siparisler
    WHERE siparis_no = @siparis_no;
END;
GO

-- 5. Çalışan Performansı Görüntüle
CREATE PROCEDURE CalisanPerformansi
    @calisan_id INT
AS
BEGIN
    SELECT * FROM Puan
    WHERE calisan_id = @calisan_id;
END;

```

## Trigger Kullanımı

*Commit ve rollback kullanılarak oluşturulan kodlar:*

```

-- Trigger Örnekleri
-- 1. Malzeme Stok Kontrolü
CREATE TRIGGER StokKontrol
ON Malzeme
INSTEAD OF UPDATE
AS
BEGIN
    IF EXISTS (SELECT 1 FROM inserted WHERE stok_miktar < 0)
    BEGIN
        PRINT 'Stok miktarı sıfırdan küçük olamaz!';
    END
    ELSE
    BEGIN
        UPDATE Malzeme
        SET stok_miktar = inserted.stok_miktar
        FROM inserted
        WHERE Malzeme.malzeme_id = inserted.malzeme_id;
    END
END;
GO

```

```

-- 2. Siparis Tarih Kontrolü
CREATE TRIGGER SiparisTarihKontrol
ON Siparisler
INSTEAD OF INSERT
AS
BEGIN
    IF EXISTS (SELECT 1 FROM inserted WHERE tarih < CAST(GETDATE() AS
DATE))
    BEGIN
        PRINT 'Sipariş tarihi bugünden eski olamaz!';
    END
    ELSE
    BEGIN
        INSERT INTO Siparisler (kategori, malzeme_id, firma, adet_miktar,
aciliyet, tarih, onay, calisan_id)
        SELECT kategori, malzeme_id, firma, adet_miktar, aciliyet, tarih,
onay, calisan_id
        FROM inserted;
    END
END;
GO

-- 3. Vardiya Süre Kontrolü
CREATE TRIGGER VardiyaSureKontrol
ON Vardiya
INSTEAD OF INSERT
AS
BEGIN
    IF EXISTS (SELECT 1 FROM inserted WHERE vardiyaBaslangic >=
vardiyaBitis)
    BEGIN
        PRINT 'Vardiya başlangıç tarihi bitiş tarihinden küçük olmalıdır!';
    END
    ELSE
    BEGIN
        INSERT INTO Vardiya (vardiyaBaslangic, vardiyaBitis, lokasyonAdi,
kokuNumarasi, egitim, makineNo, seriNo, toplamMeleme, toplamKarte,
malzeme_id, sondaj, yardimci, basMotorSaati, bitMotorSaati, motorYagi,
hidrolikYagi, havaDurumu, aciklama, calisan_id)
        SELECT vardiyaBaslangic, vardiyaBitis, lokasyonAdi, kokuNumarasi,
egitim, makineNo, seriNo, toplamMeleme, toplamKarte, malzeme_id, sondaj,
yardimci, basMotorSaati, bitMotorSaati, motorYagi, hidrolikYagi,
havaDurumu, aciklama, calisan_id
        FROM inserted;
    END
END;

```

## Transaction Kullanımı

```

-- Transaction İşlemleri Örnekleri

-- Örnek 1: Calisan ve Vardiya Tablosunda Transaction Kullanımı
BEGIN TRANSACTION;
BEGIN TRY

```



```

-- Calisan Tablosuna Veri Ekle
INSERT INTO Calisan (ad, soyad, tc_no, dogum_tarihi, telefon, adres,
mezuniyet, meslek, gorev, aciklama, giris_tarihi, cikis_tarihi, birim_izin,
kurum, isveren_id, sgk_isyeri_sicil_no, istihdam, tur)
VALUES ('Mehmet', 'Demir', '11223344556', '1985-03-03', '5553334455',
'Adres 3', 'Lise', 'Tekniker', 'Bakım', 'Açıklama 3', '2025-01-03', NULL,
12, 'Kurum C', 3, '54321', 'Tam Zamanlı', 'isci');

-- Yeni eklenen çalışanın ID'sini al
DECLARE @CalisanID INT;
SET @CalisanID = SCOPE_IDENTITY();

-- Vardiya Tablosuna Veri Ekle
INSERT INTO Vardiya (vardiyaBaslangic, vardiyaBitis, lokasyonAdi,
kokuNumarasi, egitim, makineNo, seriNo, toplamMeleme, toplamKarte,
malzeme_id, sondaj, yordimci, basMotorSaati, bitMotorSaati, motorYagi,
hidrolikYagi, havaDurumu, aciklama, calisan_id)
VALUES ('2025-01-07 08:00:00', '2025-01-07 16:00:00', 'Fabrika C',
'Koku-3', 'Eğitim C', 103, 'Seri-003', 20, 10, 1, 'Sondaj-3', 'Yardımcı C',
14.0, 18.0, '5W-30', 'Hidrolik-3', 'Yağmurlu', 'Vardiya açıklaması 3',
@CalisanID);

COMMIT TRANSACTION;
PRINT 'Transaction başarıyla tamamlandı.';
END TRY
BEGIN CATCH
ROLLBACK TRANSACTION;
PRINT 'Transaction sırasında bir hata oluştu. İşlem geri alındı.';
THROW;
END CATCH;
GO

-- Örnek 2: Siparisler ve Malzeme Stok Güncelleme İşlemleri
BEGIN TRANSACTION;
BEGIN TRY
-- Sipariş Ekle
INSERT INTO Siparisler (kategori, malzeme_id, firma, adet_miktar,
aciliyet, tarih, onay, calisan_id)
VALUES ('Hammadde', 1, 'Firma C', 25, 'düşük', '2025-01-15', 1, 1);

-- Malzeme Stok Güncelle
UPDATE Malzeme
SET stok_miktar = stok_miktar - 25
WHERE malzeme_id = 1;

-- Kontrol: Stok miktarı sıfırdan küçük olursa hata fırlat
IF EXISTS (SELECT 1 FROM Malzeme WHERE stok_miktar < 0)
BEGIN
THROW 50000, 'Stok miktarı sıfırdan küçük olamaz!', 1;
END;

COMMIT TRANSACTION;
PRINT 'Sipariş ve stok güncelleme başarıyla tamamlandı.';
END

```

```
BEGIN
```

```
    ROLLBACK TRANSACTION;
```

```
    PRINT 'Transaction sırasında bir hata oluştu. İşlem geri alındı.';
```

```
    THROW;
```

```
END CATCH;
```

```
GO
```

```
-- Örnek 3: Birden Fazla Tabloya Veri Ekleme ve Geri Alma
```

```
BEGIN TRANSACTION;
```

```
BEGIN TRY
```

```
    -- Calisan Ekle
```

```
    INSERT INTO Calisan (ad, soyad, tc_no, dogum_tarihi, telefon, adres,
mezuniyet, meslek, gorev, aciklama, giris_tarihi, cikis_tarihi, birim_izin,
kurum, isveren_id, sgk_isyeri_sicil_no, istihdam, tur)
```

```
    VALUES ('Zeynep', 'Ak', '99887766554', '1993-05-05', '5554445566',
'Adres 4', 'Doktora', 'Araştırmacı', 'AR-GE', 'Açıklama 4', '2025-01-04',
NULL, 20, 'Kurum D', 4, '98765', 'Tam Zamanlı', 'isci');
```

```
    -- Malzeme Ekle
```

```
    INSERT INTO Malzeme (malzeme_adi, stok_miktar, aciklama)
```

```
    VALUES ('Alüminyum', 50, 'Hafif ve dayanıklı malzeme');
```

```
    -- Yeni Çalışanın ID'sini ve Yeni Malzemenin ID'sini Al
```

```
    DECLARE @YeniCalisanID INT, @YeniMalzemeID INT;
```

```
    SET @YeniCalisanID = SCOPE_IDENTITY();
```

```
    INSERT INTO HafifMalzeme (malzeme_id, stok_miktar, aciklama) VALUES
(SCOPE_IDENTITY(), 50, 'Yeni hafif malzeme');
```

```
    -- Vardiya Ekle
```

```
    INSERT INTO Vardiya (vardiyaBaslangic, vardiyaBitis, lokasyonAdi,
kokuNumarasi, egitim, makineNo, seriNo, toplamMeleme, toplamKarte,
malzeme_id, sondaj, yardimci, basMotorSaati, bitMotorSaati, motorYagi,
hidrolikYagi, havaDurumu, aciklama, calisan_id)
```

```
    VALUES ('2025-01-08 08:00:00', '2025-01-08 16:00:00', 'Fabrika D',
'Koku-4', 'Eğitim D', 104, 'Seri-004', 25, 12, @YeniMalzemeID, 'Sondaj-4',
'Yardımcı D', 15.0, 19.0, '0W-20', 'Hidrolik-4', 'Karlı', 'Vardiya
açıklaması 4', @YeniCalisanID);
```

```
    COMMIT TRANSACTION;
```

```
    PRINT 'Birden fazla tabloya veri ekleme işlemi başarıyla tamamlandı.';
```

```
END TRY
```

```
BEGIN CATCH
```

```
    ROLLBACK TRANSACTION;
```

```
    PRINT 'Transaction sırasında bir hata oluştu. İşlem geri alındı.';
```

```
    THROW;
```

```
END CATCH;
```