

Proje Başlığı: Araç Yedek Parça Tedarik ve Pazarlama Otomasyonu

Proje Ekibindeki Kişiler:

210260303-Cem Davrak

1.Proje Amacı: Bir ticaret iş akışını desteklemek,iş sürecinde oluşabilecek yanlışları önlemek için stok yönetimi,tedarikçi yönetimi,depo yönetimi,stok yönetimi ve kargo takibi yapabilmemizi sağlar.

Müşteriler sipariş verebilmeli,depoları ürün tedarik edilebilmeli ve depo içeriği bilinip ona göre stok takibi yapılabilmesi.

2.Kullanıcı Türleri ve Gereksinimler:

1.Müşteri Rolü:

- Sipariş Verebilme ve sipariş detayını görebilme
- Kişisel bilgilerini güncelleme
- Kargo takip numarası üzerinden sipariş durumunu sorgulama

2.Depo Sorumlusu Rolü:

- Depoya gelen ürünlerin giriş işlemlerini kaydetme.
- Ürünlerin stok miktarını görme ve eksik ürünler için tedarik siparişi oluşturma.

3.Tedarikçi Rolü:

- Depodan gelen taleplere göre ürün sağlama.
- Ürün bazlı tedarik işlemlerinin miktar ve teslim tarihlerini görme.
- Sipariş durumunu takip ederek sevkiyat planı yapma.

4. Kargo Firması Rolü

- Sipariş ve tedarik siparişlerine ait işlem bilgilerini güncel tutma.
- Sipariş durumunu takip ederek sevkiyat planı yapma.

5. Yönetici Rolü

- Müşteri ve Tedarikçi kayıtlarını kontrol etme ve düzenleme.

- Ürün kategorileri,depo bilgileri ve stok durumlarına erişim.
- Kargo firmalarıyla iş birliği süreçlerini yönetme.

3.Varlık ve Nitelikleri

1.Müşteri: Müsteri_ID(Primary Key), Ad_Soyad, Telefon, E_Posta, Adres

2.Tedarikçi: Tedarikci_ID(Primary Key),Firma_Adı,Firma_Telefon,Firma_Adres

3.Sipariş: Siparis_ID(Primary Key),Müşteri_ID(Foreign Key),Siparis_Tarihi, Toplam_Tutar

4.Ürün: Parca_ID (Primary Key), ParcaAdı_ID, Stok_Miktarı, Birim_Fiyat, Satis_Fiyat, Ürün_Kategorisi_ID, Tedarikci_ID

5.Depo: Depo_ID (Primary Key), Depo_Adı, Konum, Depo_Sorumlusu_ID

6. Kargo: Kargo_ID (Primary Key), Kargo_Firma_ID(Foreign Key), Takip_No, Durum, Siparis_ID(Foreign Key)

7.Ürün Kategorisi: Ürün_Kategorisi_ID(PrimaryKey),Bulunduğu_Depo(Foreign Key), Ürün_Türü

8.Parça Adı: Parca_ID(Primary Key), ParcaAdı_ID

9. Depo Sorumlusu: Depo_Sorumlusu_ID(Primary Key),Ad_Soyad,Telefon,Ev Adresi

10. Kargo Firması: Kargo_Firma_ID (Primary Key), Kargo_Firma_Adı

11. Sipariş Ürün: Siparis_ID(Foreign Key), Parça_ID(Foreign Key),Miktar,Toplam_Fiyat

12.Tedarik Siparişi: Tedarik_Siparis_ID (Primary Key), Tedarikci_ID(Foreign Key), Depo_ID(Foreign Key), Parca_ID(Foreign Key), Miktar, Siparis_Tarihi, Tahmini_Teslim_Tarihi

13.Kargo-Müşteri: Kargo_ID(Foreign Key), Müsteri_ID (Foreign Key)

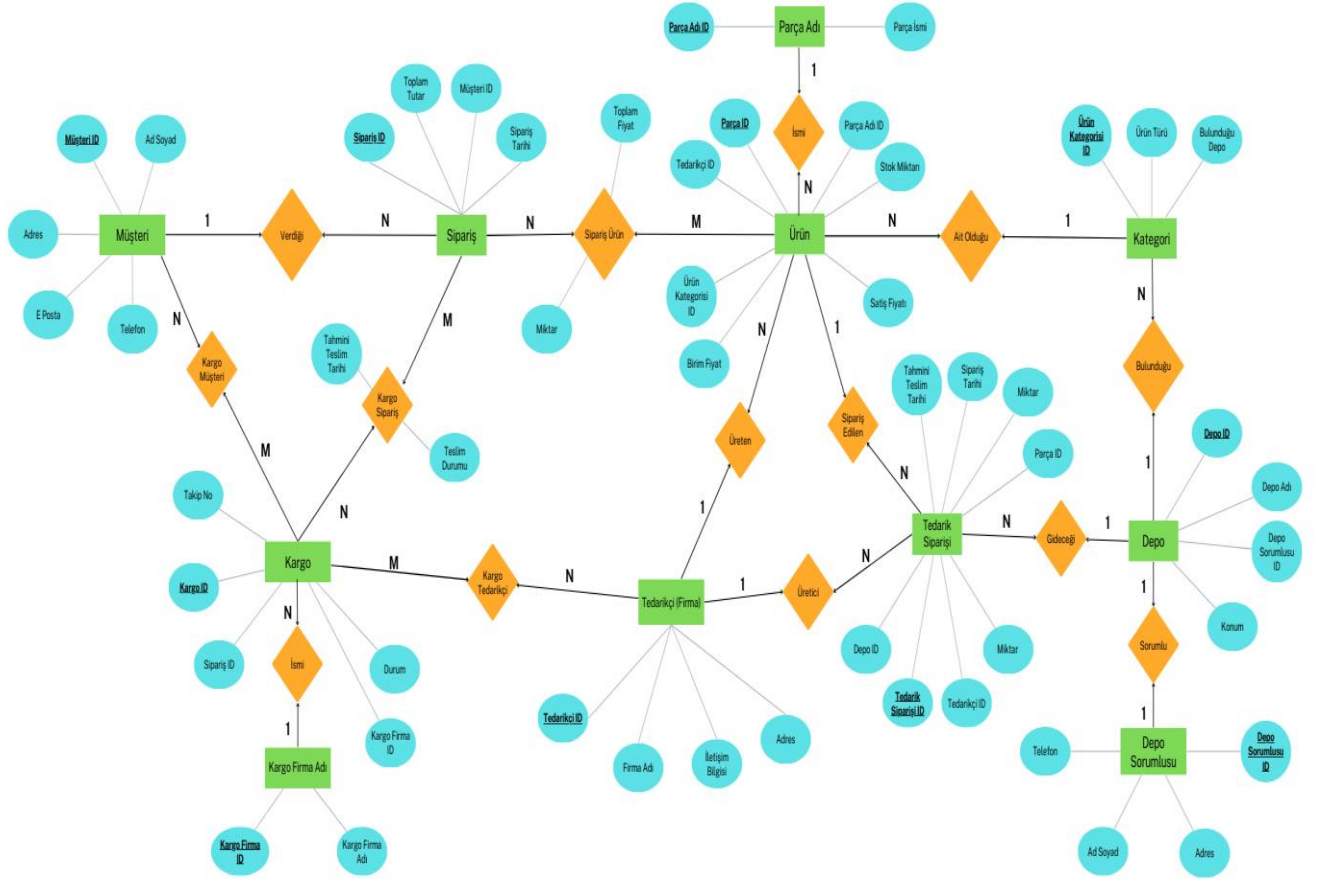
14.Sipariş-Kargo: Kargo_ID (Primary Key), Siparis_ID(Foreign Key), Tahmini_Teslim_Tarihi, Teslim_Durumu

15.Kargo-Tedarikçi: Tedarikci_ID(Foreign Key), Kargo_ID(Foreign Key)

4.İlişkileri:

- 1- Müşteri->Sipariş:Verdiği(1-N)(Bir Müşteri birden fazla sipariş verebilir)
- 2- Sipariş->Ürün:Sipariş-Ürün(N-M)(Bir sipariş birden fazla ürün içerebilir;bir ürün farklı siparişlerde olabilir)
- 3- Ürün Kategorisi->Ürün:Ait Olduğu(1-N)(Bir ürün kategorisi birden fazla ürün içerebilir)
- 4- Sipariş->Kargo: Sipariş-Kargo (N-M)(Bir sipariş birden fazla kargo ile taşınabilir, bir kargo birden fazla sipariş taşıyabilir)
- 5- Depo->Ürün Kategorisi:Bulunduğu (N-M)(Her depo birden fazla kategori ürünü içerebilir)
- 6- Kargo->Müşteri: Kargo-Müşteri(N-M)(Bir kargo birden fazla müşteriye hizmet verebilir)
- 7- Depo Sorumlusu->Depo: Sorumlu (1-1)(Her depo bir depo sorumlusuna atanır)
- 8- Kargo->Tedarikçi:Kargo-Tedarikçi (N-M)(Bir kargo birden fazla tedarikçiyle çalışabilir)
- 9- Parça Adı->Ürün:İsmi(1-N)(Bir parça adı birden fazla ürüne atanabilir)
- 10- Ürün->Tedarik Siparişi:Sipariş Eden(1-N)(Bir ürün birden fazla tedarik siparişi ile talep edilebilir)
- 11- Tedarikçi->Ürün:Üreten(1-N)(Bir tedarikçi birden fazla ürün tedarik eder)
- 12- Tedarikçi->TedarikSiparişi:Üreten(1-N)(Bir tedarikçi birden fazla sipariş alabilir)
- 13- Kargo Firma Adı->Kargo:İsmi(Bir kargo firması birden fazla kargoya hizmet verebilir)
- 14- Depo-Tedarik Siparişi:Gideceği(1-N)(Bir depo birden fazla tedarik siparişi alabilir)

5.E-R Diyagramı:



6.Veritabanı Şeması:



Select

```
Select*From Sipariş
```

Sipariş_Ürün Tablosuna Veri Ekleme

```
INSERT INTO Sipariş_Ürün (Siparis_ID, Parça_ID, Miktar,  
Toplam_Fiyat)  
VALUES (10, 339, 6, 9900.00)
```

Ürün Tablosu Güncelleme (UPDATE)

```
UPDATE Ürün  
SET Stok_Miktari = Stok_Miktari + 20  
WHERE Parca_ID = 342;
```

Sipariş Silme (Delete)

```
DELETE FROM Sipariş_Ürün  
WHERE Siparis_ID = 5;
```

```
DELETE FROM Sipariş  
WHERE Siparis_ID = 5;
```

Kargo Takip Numarası Üzerinden Kargo Durumu Sorgulama (View)

```
CREATE VIEW KargoSiparisDurum_View AS  
SELECT  
    K.Takip_No AS Kargo_Takip_No,  
    K.Durum AS Kargo_Durumu,  
    S.Siparis_ID AS Siparis_Numara,  
    S.Siparis_Tarihi AS Siparis_Tarihi,  
    S.Toplam_Tutar AS Siparis_Tutari,  
    M.Ad_Soyad AS Musteri_Adi,  
    M.Telefon AS Musteri_Telefon,  
    M.Adres AS Musteri_Adresi,  
    SK.Tahmini_Teslim_Tarihi AS Tahmini_Teslim,  
    SK.Teslim_Durumu AS Teslim_Durumu  
  
FROM  
    Kargo K  
INNER JOIN  
    Sipariş S ON K.Siparis_ID = S.Siparis_ID
```

```
INNER JOIN
    Müsteri M ON S.Müsteri_ID = M.Müsteri_ID
INNER JOIN
    Siparis_Kargo SK ON K.Kargo_ID = SK.Kargo_ID AND
    S.Siparis_ID = SK.Siparis_ID;
```

Çalıştırılması

```
SELECT *FROM KargoSiparisDurum_View
WHERE Kargo_Takip_No = '10016';
```

Kargo Durumu Güncelleme (Trigger)

```
CREATE TRIGGER KargoDurumGuncelle
ON Kargo
AFTER UPDATE
AS
BEGIN
    UPDATE Siparis_Kargo
    SET Teslim_Durumu = INSERTED.Durum
    FROM
        INSERTED
    WHERE
        Siparis_Kargo.Kargo_ID = INSERTED.Kargo_ID;
END;
```

Ürün Kategorisi Ekleme(Stored Procedure)

```
CREATE PROCEDURE UrunKategorisiEkle
    @UrunTuru VARCHAR(100),
    @DepoID INT
AS
BEGIN
    INSERT INTO Ürün_Kategorisi (Ürün_Türü,
    Bulunduğu_Depo)
    VALUES (@UrunTuru, @DepoID);
END;

EXEC UrunKategorisiEkle @UrunTuru = 'Elektronik', @DepoID
= 19;
```

Stok Miktarı Güncelleme Hata Kontrolü(Transaction)

```
SELECT * FROM Ürün;

BEGIN TRANSACTION;

BEGIN TRY
    UPDATE Ürün
    SET Stok_Miktari = Stok_Miktari -3
    WHERE Parca_ID = 322;

    UPDATE Ürün
    SET Stok_Miktari = Stok_Miktari -15
    WHERE Parca_ID = 323;

    COMMIT TRANSACTION;
    PRINT 'Tüm işlemler başarıyla tamamlandı.';
END TRY

BEGIN CATCH
    ROLLBACK TRANSACTION;
    PRINT 'Hata oluştu: İşlemler geri alındı.';
END CATCH;
```

Sipariş Verirken Hata Kontrolü (Transaction)

```
BEGIN TRANSACTION;
```



```
BEGIN TRY
```

```
    DECLARE @YeniSiparisID INT;  
    INSERT INTO Sipariş (Müşteri_ID, Siparis_Tarihi, Toplam_Tutar)  
    VALUES ('12345678901', '2024-12-21', '1000');  
    SET @YeniSiparisID = SCOPE_IDENTITY();
```

```
    INSERT INTO Sipariş_Ürün (Siparis_ID, Parça_ID, Miktar,  
Toplam_Fiyat)  
    VALUES (@YeniSiparisID, 1, 2, 500);  
    INSERT INTO Sipariş_Ürün (Siparis_ID, Parça_ID, Miktar,  
Toplam_Fiyat)  
    VALUES (@YeniSiparisID, 2, 3, 500);
```

```
    UPDATE Ürün  
    SET Stok_Miktari = Stok_Miktari - 2  
    WHERE Parca_ID = 330;  
    UPDATE Ürün  
    SET Stok_Miktari = Stok_Miktari - 3  
    WHERE Parca_ID = 347;
```

```
    INSERT INTO Kargo (Kargo_Firma_ID, Takip_No, Durum, Siparis_ID)  
    VALUES (1, '10001', 'Hazırlanıyor', @YeniSiparisID);
```

```
    COMMIT TRANSACTION;  
    PRINT 'Sipariş başarıyla tamamlandı.';  
END TRY
```

```
BEGIN CATCH  
    ROLLBACK TRANSACTION;  
    PRINT 'Hata: Sipariş işlemi tamamlanamadı. Değişiklikler geri  
alındı.';  
END CATCH;
```

8.Tüm Tablo Bilgileri Ve Kısıtlamaları

1.Tedarikçi Tablosu

```
CREATE TABLE Tedarikci(  
Tedarikci_ID INT PRIMARY KEY IDENTITY(1,1) NOT NULL,  
Firma_Adı nvarchar(50),  
Firma_Telefon nchar(11),  
Firma_Adres nchar(100)  
)
```

2.Müşteri Tablosu

```
CREATE TABLE Müsteri(  
Müsteri_ID NVARCHAR(11) PRIMARY KEY NOT NULL,  
Ad_Soyad varchar(50) NOT NULL,  
Telefon nvarchar(11) NOT NULL,  
E_Posta varchar(50),  
Adres nchar(100)  
)
```

3.Sipariş Tablosu

```
CREATE TABLE Sipariş(  
Siparis_ID INT PRIMARY KEY IDENTITY(1,1) NOT NULL,  
Müsteri_ID NVARCHAR(11)NOT NULL,  
Siparis_Tarihi NCHAR(10)NOT NULL,  
Toplam_Tutar NCHAR(10)NOT NULL  
  
FOREIGN KEY (Müsteri_ID) REFERENCES  
Müsteri(Müsteri_ID)  
)
```

4.Depo Sorumlusu Tablosu

```
CREATE TABLE DepoSorumlusu (  
Depo_Sorumlusu_ID INT PRIMARY KEY IDENTITY(1,1) NOT  
NULL,  
Ad_Soyad VARCHAR(50) NOT NULL,  
Telefon NVARCHAR(11) NOT NULL,  
[Ev Adresi] TEXT  
)
```

5.Depo Tablosu

```
CREATE TABLE Depo (  
    Depo_ID INT PRIMARY KEY IDENTITY(1,1) NOT NULL,  
    Depo_Adı VARCHAR(100) NOT NULL,  
    Konum TEXT NOT NULL,  
    Depo_Sorumlusu_ID INT NOT NULL,  
    FOREIGN KEY (Depo_Sorumlusu_ID) REFERENCES  
DepoSorumlusu(Depo_Sorumlusu_ID)  
);
```

6.Parça_Adı Tablosu

```
CREATE TABLE Parça_Adı (  
    ParçaAdı_ID INT PRIMARY KEY IDENTITY(1,1) NOT NULL,  
    Parça_Adı VARCHAR(100) NOT NULL  
);
```

7.Ürün_Kategorisi Tablosu

```
CREATE TABLE Ürün_Kategorisi (  
    Ürün_Kategorisi_ID INT PRIMARY KEY IDENTITY(1,1) NOT  
NULL,  
    Ürün_Türü VARCHAR(100) NOT NULL,  
    Bulunduğu_Depo INT,  
    FOREIGN KEY (Bulunduğu_Depo) REFERENCES Depo(Depo_ID)  
ON DELETE SET NULL  
);
```

8.Ürün Tablosu

```
CREATE TABLE Ürün (  
    Parca_ID INT PRIMARY KEY IDENTITY(1,1) NOT NULL,  
    ParcaAdı_ID INT NOT NULL,  
    Stok_Miktari INT NOT NULL,  
    Birim_Fiyat DECIMAL(10, 2) NOT NULL,  
    Satis_Fiyat DECIMAL(10, 2) NOT NULL,  
    Ürün_Kategorisi_ID INT NOT NULL,  
    Tedarikci_ID INT NOT NULL,  
  
    FOREIGN KEY (ParcaAdı_ID) REFERENCES Parça_Adı(ParçaAdı_ID),  
    FOREIGN KEY (Ürün_Kategorisi_ID) REFERENCES  
Ürün_Kategorisi(Ürün_Kategorisi_ID),  
    FOREIGN KEY (Tedarikci_ID) REFERENCES Tedarikci(Tedarikci_ID)  
);
```

9.Kargo Tablosu

```

CREATE TABLE Kargo (
    Kargo_ID INT PRIMARY KEY IDENTITY(1,1) NOT NULL,
    Kargo_Firma_ID INT NOT NULL,
    Takip_No NVARCHAR(5) NOT NULL,
    Durum NVARCHAR(12) NOT NULL,
    Siparis_ID INT NOT NULL,

    FOREIGN KEY (Kargo_Firma_ID) REFERENCES
Kargo_Firma(Kargo_Firma_ID),
    FOREIGN KEY (Siparis_ID) REFERENCES Sipariş(Siparis_ID)
);

```

10.Sipariş_Ürün Tablosu

```

CREATE TABLE Sipariş_Ürün (
    Siparis_ID INT NOT NULL,
    Parça_ID INT NOT NULL,
    Miktar INT NOT NULL CHECK (Miktar BETWEEN 1 AND 10),
    Toplam_Fiyat DECIMAL(10, 2) NOT NULL

    PRIMARY KEY (Siparis_ID,Parça_ID),
    FOREIGN KEY (Siparis_ID) REFERENCES Sipariş(Siparis_ID),
    FOREIGN KEY (Parça_ID) REFERENCES Ürün(Parca_ID)
);

```

11.Tedarik_Siparişi Tablosu

```

CREATE TABLE Tedarik_Siparisi (
    Tedarik_Siparis_ID INT PRIMARY KEY IDENTITY(1,1) NOT NULL,
    Tedarikci_ID INT NOT NULL,
    Depo_ID INT NOT NULL,
    Parca_ID INT NOT NULL,
    Miktar INT NOT NULL CHECK (Miktar > 0),
    Siparis_Tarihi NCHAR(10) NOT NULL,
    Tahmini_Teslim_Tarihi NCHAR(10) NOT NULL,

    FOREIGN KEY (Tedarikci_ID) REFERENCES Tedarikci(Tedarikci_ID),
    FOREIGN KEY (Depo_ID) REFERENCES Depo(Depo_ID),
    FOREIGN KEY (Parca_ID) REFERENCES Ürün(Parca_ID)
);

```

12.Kargo_Tedarikçi Tablosu

```
CREATE TABLE Kargo_Tedarikci (  
    Kargo_ID INT NOT NULL,  
    Tedarikci_ID INT NOT NULL,  
  
    PRIMARY KEY (Kargo_ID, Tedarikci_ID),  
    FOREIGN KEY (Kargo_ID) REFERENCES Kargo(Kargo_ID),  
    FOREIGN KEY (Tedarikci_ID) REFERENCES Tedarikci(Tedarikci_ID)  
);
```

13.Sipariş_Kargo Tablosu

```
CREATE TABLE Siparis_Kargo (  
    Siparis_ID INT NOT NULL,  
    Kargo_ID INT NOT NULL,  
    Tahmini_Teslim_Tarihi NCHAR(10) NOT NULL,  
    Teslim_Durumu NVARCHAR(12) NOT NULL,  
  
    PRIMARY KEY (Siparis_ID, Kargo_ID),  
    FOREIGN KEY (Siparis_ID) REFERENCES Siparis(Siparis_ID),  
    FOREIGN KEY (Kargo_ID) REFERENCES Kargo(Kargo_ID)  
);
```

14.Kargo_Müşteri Tablosu

```
CREATE TABLE Kargo_Musteri (  
    Kargo_ID INT NOT NULL,  
    Müsteri_ID NVARCHAR(11) NOT NULL,  
  
    PRIMARY KEY (Kargo_ID, Müsteri_ID),  
    FOREIGN KEY (Kargo_ID) REFERENCES Kargo(Kargo_ID),  
    FOREIGN KEY (Müsteri_ID) REFERENCES Müsteri(Müsteri_ID)  
);
```

15.Kargo_Firma Tablosu

```
CREATE TABLE Kargo_Firma (  
    Kargo_Firma_ID INT PRIMARY KEY IDENTITY(1,1) NOT NULL,  
    Kargo_Firma_Adi VARCHAR(50) NOT NULL UNIQUE  
);
```

VERİTABANINA GİTHUB SAYFASINDAN ERIŞEBİLİRSİNİZ