**Öğrenci Adı Soyadı :** Burak Akça

**Öğrenci Numarası:** 170422005

**T.C. MARMARA ÜNİVERSİTESİ TEKNOLOJİ FAKÜLTESİ**

**2024-2025 EĞİTİM ÖĞRETİM YILI**

**GÜZ DÖNEMİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**VERİTABANI YÖNETİM SİSTEMLERİ DERSİ**

**PROJE-FİNAL DÖKÜMANI**

1. **Gerçekleştirdiğiniz veri tabanı projesi için grup arkadaşlarınızın isimlerini yazınız ve projenize ait veri tabanı/diğer yazılım bileşenleri hakkında bilgi veriniz. (5 p)**

Grup çalışmamızda Hüseyin Karakaya ve Ozan Deste ile birlikte çalışarak bir Araç Bakım ve Servis Takip Sistemi geliştirdik. Bu proje, araçların bakım süreçlerini ve servis kayıtlarını daha etkili bir şekilde yönetmek için tasarlandı. Projemiz, masaüstü bir uygulama olarak **JavaFX** kullanılarak geliştirildi. Arka planda, veri yönetimi için **Microsoft SQL Server (MSSQL)** tercih edildi.

1. **Gerçekleştirdiğiniz veri tabanı projesi için proje dokümanınızı ve dosyalarınızı içeren herkese açık github bağlantılarınızı paylaşınız. (5 p)**

**<https://github.com/karakayahuseyin/arac-servis-bakim>**

**<https://github.com/Burak-Akca/Servis_takip_sistemi>**

1. **Gerçekleştirdiğiniz projenin amacını detaylı bir şekilde açıklayınız. (10 p)**

Bu proje, bir araç servis işletmesinin yedek parça ve servis operasyonlarını verimli bir şekilde takip edebilmesi ve yönetebilmesi için bir veritabanı tasarımı ve uygulamasını amaçlamaktadır. Servis süreçlerinin kayıt altına alınması, yedek parça stoklarının yönetimi ve müşterilerle ilgili detaylı bilgilerin saklanması için bir çözüm sunar. Projenin amaçlarını detaylandırıcak olursak:

**Servis Süreçlerini Takip Etmek:**

* Araçların muayene ve tamir süreçleriyle ilgili tüm bilgileri bir arada tutarak düzenli bir kayıt sistemi oluşturmak.
* Servis teknisyenlerinin hangi muayenelerde görev aldığını ve tamir işlemlerinde kullanılan işçilik ve yedek parça bilgilerini saklamak.
* Hangi araçların hangi tarihlerde servise geldiğini ve tamir masraflarını takip etmek.

**Yedek Parça Stok Yönetimi:**

* Araç tamirlerinde kullanılan yedek parçaların stok durumlarını ve maliyetlerini kontrol altında tutmak.
* Hangi yedek parçaların hangi araç modellerinde kullanıldığını kaydetmek.
* Yedek parça stoklarında eksikliklerin önüne geçerek zamanında tedarik edilmesini sağlamak.

**Müşteri ve Araç Bilgilerinin Yönetimi:**

* Müşterilerin kişisel bilgilerini (isim, soyisim, iletişim bilgileri vb.) ve sahip oldukları araçları kayıt altına almak.
* Müşterilerin hangi araçlarıyla ne zaman servise geldiklerini ve toplamda ne kadar masraf yaptıklarını görmek.
* Müşteri odaklı bir veri yönetimi sağlayarak müşteri memnuniyetini artırmak.

1. **Tasarladığınız veri tabanı mimarisinde hangi tablo ve ilişkileri kullanıldığınızı açıklayınız. (10 p)**

**Tablolar**

**1-Müşteri:**

· Müşterilerin bilgilerini tutar.

· Alanlar: id, Ad, Soyad, Mail, Yaş, Password, Kullanıcı\_Adı.

**2**-**MüşteriAraba**:

· Müşterilere ait araç bilgilerini içerir.

· Alanlar: Plaka, Araba\_Modeli, Ruhsat\_Sahibi.

**3-ArabaModeli:**

· Araç modelleri ile ilgili bilgileri barındırır.

· Alanlar: id, Marka, Modeli, Üretim\_Yılı, Periyodik\_Bakım\_Süresi.

**4-YedekParça:**

· Araç yedek parçalarını içerir.

· Alanlar: id, Yedek\_Parça\_Adı, Stok, Fiyat, Araba\_Modeli.

**5-Muayene:**

· Araçların muayene kayıtlarını tutar.

· Alanlar: id, Servis\_Teknikeri, Tarih, Plaka.

**6-Tamir:**

· Araç tamir bilgilerini barındırır.

· Alanlar: id, Yedek\_Parça, Tamir\_Süresi, İşçilik\_Ücreti, Toplam\_Ücret, Muayene\_id.

**7-ServisTeknikeri:**

· Servis teknikerlerinin bilgilerini tutar.

· Alanlar:id, Ad, Soyad, Mail, Password, Kullanıcı\_Adı, Yaş.

**İlişkiler**

**1-Müşteri - MüşteriAraba:**

· Bir müşteri birden fazla araca sahip olabilir. Ruhsat\_Sahibi alanı ile ilişki kuruldu.

**2-MüşteriAraba - ArabaModeli:**

· Her aracın modeli vardır. Araba\_Modeli alanı bu ilişkiyi ifade eder.Bir araba modeli birden fazla kullanıcıda olabilir.

**3-ArabaModeli - YedekParça:**

· Her araç modeline uygun yedek parçalar vardır. Bir arabada birden çok yedek parça kullanılabilir.

**4-Muayene - ServisTeknikeri:**

· Muayene işlemi bir tekniker tarafından gerçekleştirilir. Bir servis teknikeri birden fazla muayene yapabilir

**5-Muayene - MüşteriAraba:**

· Muayene edilen araçlar Plaka alanı üzerinden ilişkilendirildi. Bir araba birden fazla muayenede yer alabilir

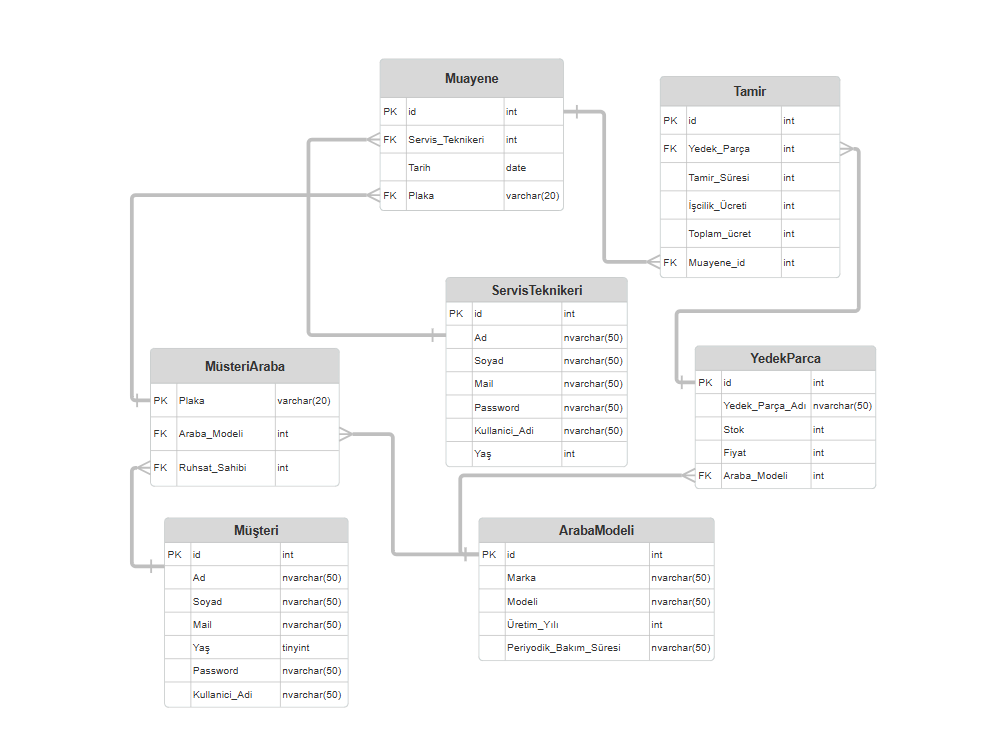
**6-Tamir - Muayene:**

· Tamir işlemleri bir muayeneye bağlıdır. Muayene\_id ile ilişki kuruldu. Bir muayenede birden fazla tamir yer alabilir

**7-Tamir - YedekParça:**

· Tamir işlemlerinde kullanılan yedek parçalar Yedek\_Parça alanı üzerinden ilişkilendirilmiştir.Bir yedek\_parça birden fazla tamirde kullanılabilir.

1. **Veri tabanı ER (Entity Relationship) diagramının bilgisayar ortamında çizilmiş halini paylaşınız. (Ara raporda eksik kısımlar bu raporda giderilmelidir ve ER çizme programlarından faydalanıbilir. Elle çizim, çizip fotoğrafını çekme vb. kabul edilmeyecektir.) (10 p)**

****

1. **Herhangi iki tablonuz için DDL (create) kodları yazılmalıdır. (10 p)**

**Musteri Tablosu**

CREATE TABLE Musteri ( id INT PRIMARY KEY IDENTITY(1,1),

Ad NVARCHAR(50),

Soyad NVARCHAR(50) ,

Mail NVARCHAR(50),

Yas tinyint,

Password NVARCHAR(50) ,

Kullanici\_Adi NVARCHAR(50) );

**MusteriAraba Tablosu**

CREATE TABLE MusteriAraba (

Plaka VARCHAR(20) PRIMARY KEY,

Araba\_Modeli INT,

Ruhsat\_Sahibi INT,

FOREIGN KEY (Araba\_Modeli) REFERENCES ArabaModeli(id),

FOREIGN KEY (Ruhsat\_Sahibi) REFERENCES Musteri(id));

1. **5 adet DML (update, insert, delete) içeren kodları yazılmalıdır. (10 p)**

**1. Yeni Müşteri Ekleme (INSERT)**

INSERT INTO Musteri (Ad, Soyad, Mail, Yaş, Password, Kullanici\_Adi)

VALUES ('Ahmet', 'Yılmaz', 'ahmet.yilmaz@mail.com', 35, '12345', 'ahmet35');

**2. Yeni Araç Ekleme (INSERT)**

INSERT INTO MüsteriAraba (Plaka, Araba\_Modeli, Ruhsat\_Sahibi)

VALUES ('07p2316', 40, 4);

**3. Araç Modelinin Periyodik Bakım Süresini Güncelleme (UPDATE)**

UPDATE ArabaModeli

SET Periyodik\_Bakım\_Süresi = '12 ay'

WHERE id = 40;

**4. Bir Tamir Kaydını Silme (DELETE)**

DELETE FROM Tamir

WHERE id = 2;

**5. Müşteri Bilgilerini Güncelleme (UPDATE)**

UPDATE Musteri SET Mail = 'mehmet.kaya@mail.com', Yaş = 36 WHERE id = 4;

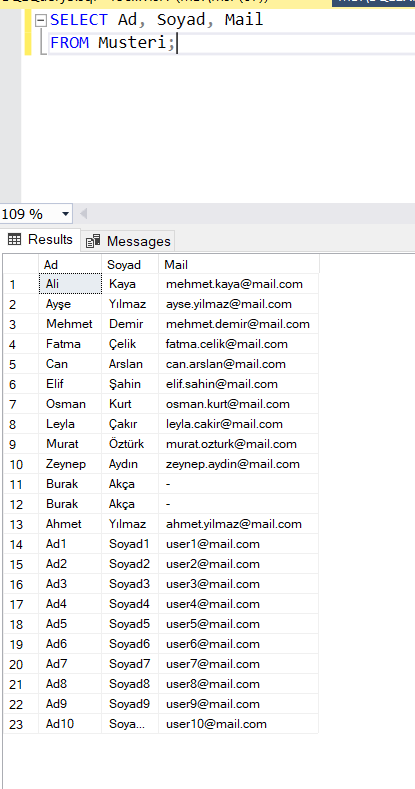
1. **Projenize ait kendi belirlediğiniz 10 adet SQL sorgusu yazınız, sorguların amacını ve sonuç çıktısını da lütfen ekleyiniz. (Açıklama: Sorgular Select deyimleri ve gruplama fonksiyonlarını HAVING deyimini (min, max, avg, count gibi) ve join deyimlerini (en az iki tablo ile birleştirme sorgusu) içerecek şekilde basitten karmaşığa doğru gitmelidir. Proje sunum anında veri tabanınıza ait sorular SQL ortamında gösterilecek ve açıklanacaktır. Raporunuzda ise sorgular, sorguların cevap ve sonuçlarının ekran görüntüsü olarak paylaşılması beklenmektedir. (10 p) Örnek:**

**1. Tüm Müşterileri Listeleme**

**Amaç**: Sistemde kayıtlı tüm müşterilerin adını, soyadını ve e-posta adreslerini listelemek.

SELECT Ad, Soyad, Mail

FROM Musteri;



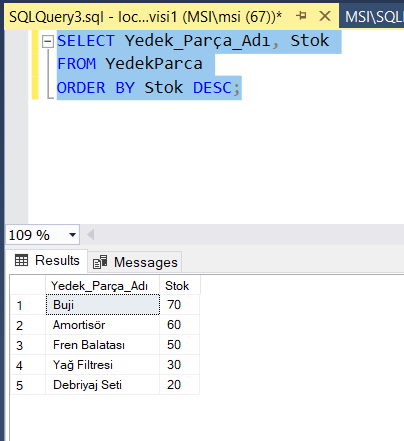
### **2. En Çok Stokta Olan Yedek Parçaları Listeleme**

**Amaç:** Stokta bulunan yedek parçaları azalan stok miktarına göre listeleme.

SELECT Yedek\_Parça\_Adı, Stok

FROM YedekParca

ORDER BY Stok DESC;



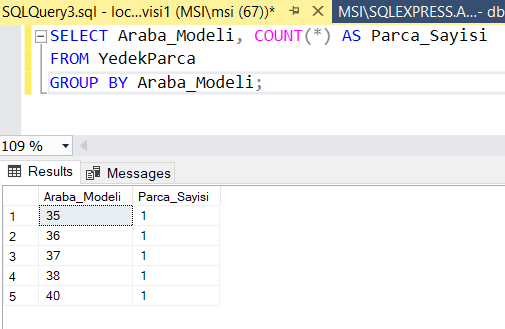
### **3. Araba Modellerine Göre Yedek Parça Sayısı**

**Amaç:** Her araba modeli için kaç farklı yedek parça bulunduğunu listeleme.

SELECT Araba\_Modeli, COUNT(\*) AS Parca\_Sayisi

FROM YedekParca

GROUP BY Araba\_Modeli;



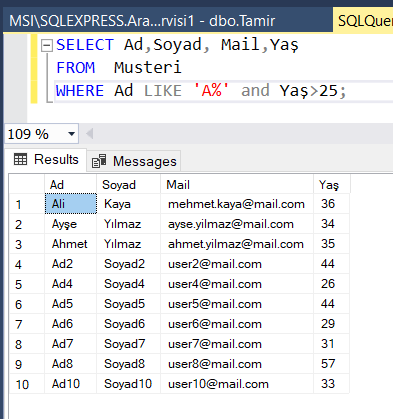
### **4. Verilen Koşula Göre Müşteri Bilgilerini Getirme**

**Amaç:** İsmi A ile başlayan ve yaşı 25 den büyük olan müşteri bilgileri listeleme.

SELECT Ad,Soyad, Mail,Yaş

FROM Musteri

WHERE Ad LIKE 'A%' and Yaş>25



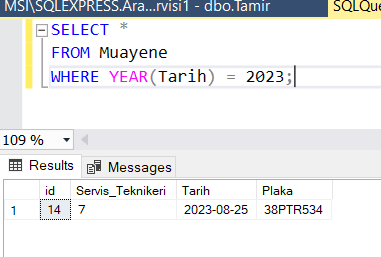
**5. 2023 Yılında Yapılan Muayeneler**

**Amaç:** 2023 yılında yapılan tüm muayeneleri listeleme.

SELECT \*

FROM Muayene

WHERE YEAR(Tarih) = 2023;



**6. Bir Plakaya Ait Müşteri Bilgilerini Getirme**

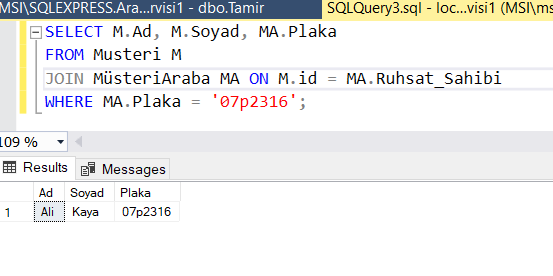
**Amaç:** Belirli bir plakaya ait müşteri bilgilerini getirme.

SELECT M.Ad, M.Soyad, MA.Plaka

FROM Musteri M

JOIN MüsteriAraba MA ON M.id = MA.Ruhsat\_Sahibi

WHERE MA.Plaka = '07p2316';



### 

### **7. En Az İki Muayeneye Sahip Müşteriler**

**Amaç:** Her müşterinin sahip olduğu araçlar için toplam muayene sayısını hesapla ve en az 2 muayenesi olan müşterileri listeleme.

SELECT M.Ad AS MusteriAdı,M.Soyad AS MusteriSoyadı,

COUNT(Mu.id) AS Toplam\_Muayene\_Sayisi

FROM Musteri M

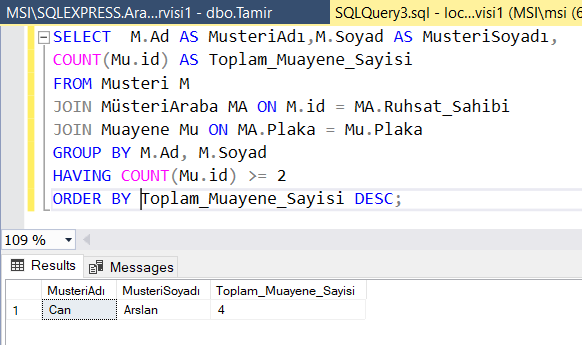
JOIN MüsteriAraba MA ON M.id = MA.Ruhsat\_Sahibi

JOIN Muayene Mu ON MA.Plaka = Mu.Plaka

GROUP BY M.Ad, M.Soyad

HAVING COUNT(Mu.id) >= 2

ORDER BY Toplam\_Muayene\_Sayisi DESC;



**8. Her Servis Teknisyeninin Tamir Sayısı**

**Amaç:** Her servis teknisyeninin kaç tamir işlemi yaptığını bulma.

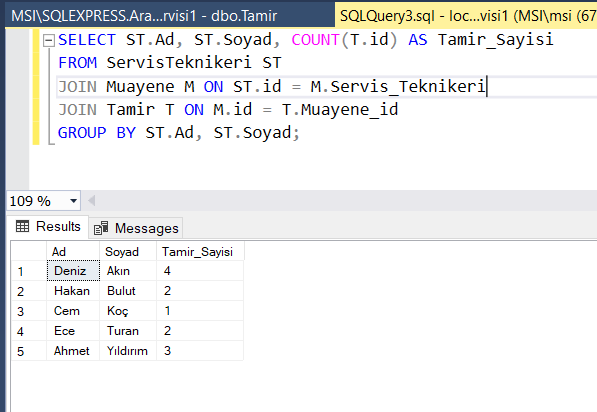
SELECT ST.Ad, ST.Soyad, COUNT(T.id) AS Tamir\_Sayisi

FROM ServisTeknikeri ST

JOIN Muayene M ON ST.id = M.Servis\_Teknikeri

JOIN Tamir T ON M.id = T.Muayene\_id

GROUP BY ST.Ad, ST.Soyad;



### **9. Müşterilerin Sahip Olduğu Araçlara Göre Toplam Muayene Sayısı**

**Amaç:** Bu sorgu, her bir müşterinin sahip olduğu araçlarla ilgili toplam muayene sayısını hesaplar.

SELECT M.Ad AS MusteriAdı,M.Soyad AS MusteriSoyadı,COUNT(Mu.id) AS Toplam\_Muayene\_Sayisi

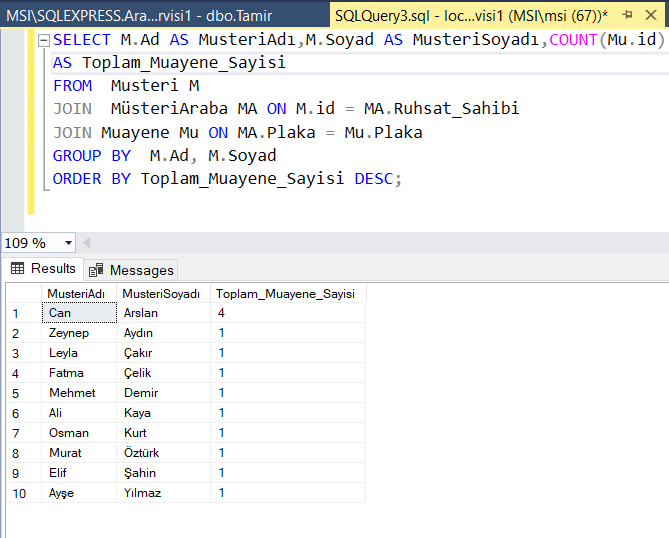
FROM Musteri M

JOIN MüsteriAraba MA ON M.id = MA.Ruhsat\_Sahibi

JOIN Muayene Mu ON MA.Plaka = Mu.Plaka

GROUP BY M.Ad, M.Soyad

ORDER BY Toplam\_Muayene\_Sayisi DESC;



**10. En Fazla Yedek Parça Kullanan Araç Modellerini Bulma**

**Amaç:** Bu sorgu, araç modellerine göre kullanılan toplam yedek parça sayısını hesaplar ve sıralar.

SELECT AM.Marka,AM.Modeli,COUNT(YP.id) AS Toplam\_Yedek\_Parca

FROM ArabaModeli AM

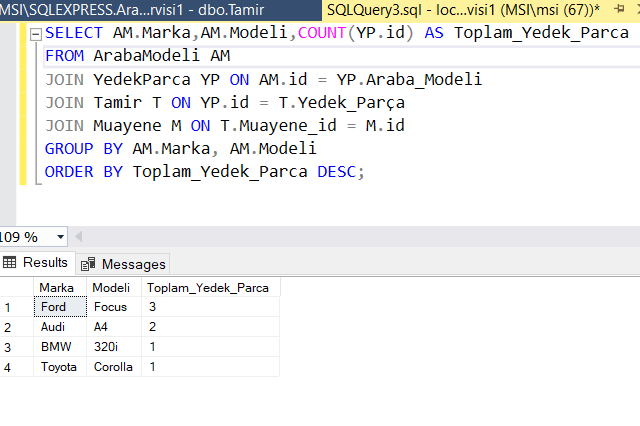
JOIN YedekParca YP ON AM.id = YP.Araba\_Modeli

JOIN Tamir T ON YP.id = T.Yedek\_Parça

JOIN Muayene M ON T.Muayene\_id = M.id

GROUP BY AM.Marka, AM.Modeli

ORDER BY Toplam\_Yedek\_Parca DESC;



1. **Eğer gerçekleştirmiş iseniz, veri tabanı bağlama ve uygulama geliştirme aşamalarınızı kısaca açıklayarak, kullanıcı ara yüz ekranından bir örnek veriniz. Ve geliştirdiğiniz ara yüzü anlatınız. (10 p)**
2. SQL Server Configuration Manager üzerinde SQL Server Hizmetleri aktifleştirildi.
3. SQL Server Configuration Manager üzerinde SQL Server Network Configuration bölümünden TCP/IP etkinleştirildi ve çalıştırıldığı port adresi bulundu. (Projemiz için 1433 olacaktır)
4. MSSQL’e ait JDBC driver dosyaları indirildi ve indirilen driver dosyasındaki 12.8.1 sürümüne ait jar dosyası projeye kütüphane olarak eklendi.
5. module-info.java içerisinde SQL kütüphanesi import edildi module … { requires java.sql; }
6. Veritabanına bağlantıyı sağlamak için controller/VeritabaniBaglantisi sınıfı oluşturuldu.

public class VeritabaniBaglantisi {

private static Connection connection;

private static final String URL = “jdbc:sqlserver…”;

private static final String USERNAME = “...”;

private static final String PASSWORD = “...”;

private static Connection getConnection() throws SQLException {

if(connection == null || connection.isClosed) {

connection =DriverManager.getConnection(URL,USERNAME,PASSWORD);

System.out.println(“Veritabanına bağlanıldı.”);

}

return Connection;}}

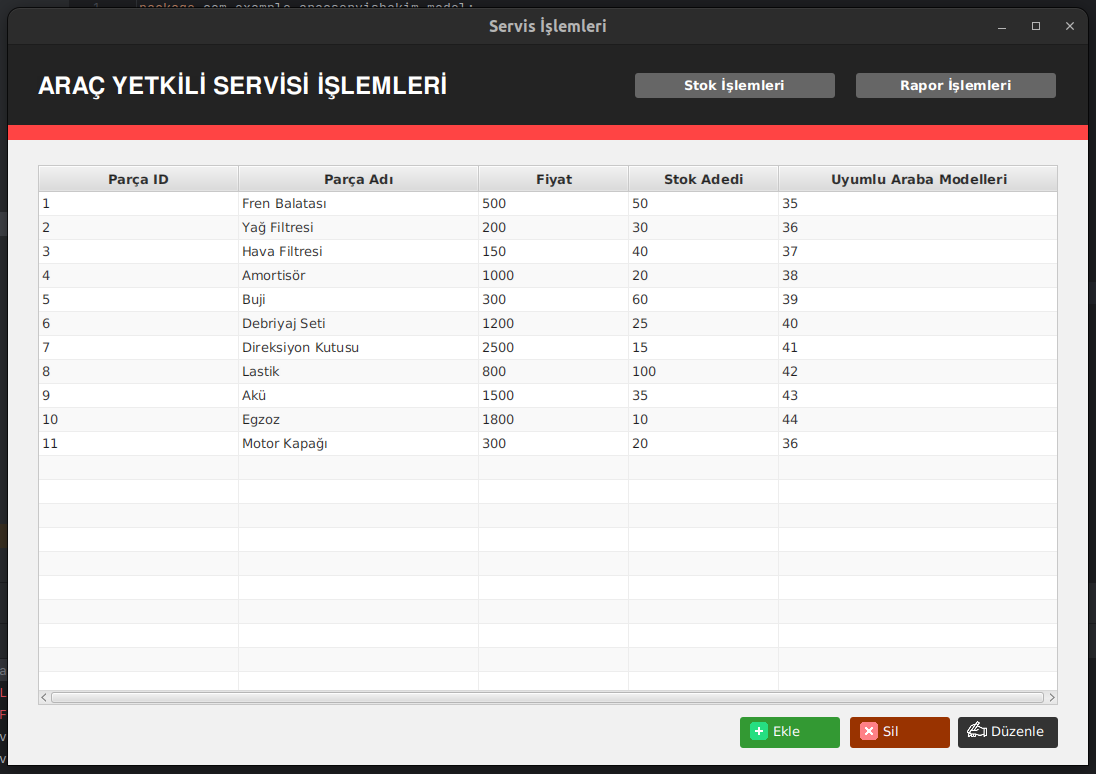
1. Hem müşteri kullanıcı hem servis teknikeri kullanıcısı için yetkilere göre giriş sağlandı.
2. Arayüz üzerinde görüntülenen tabloların her biri için model sınıfları oluşturuldu

Örn:

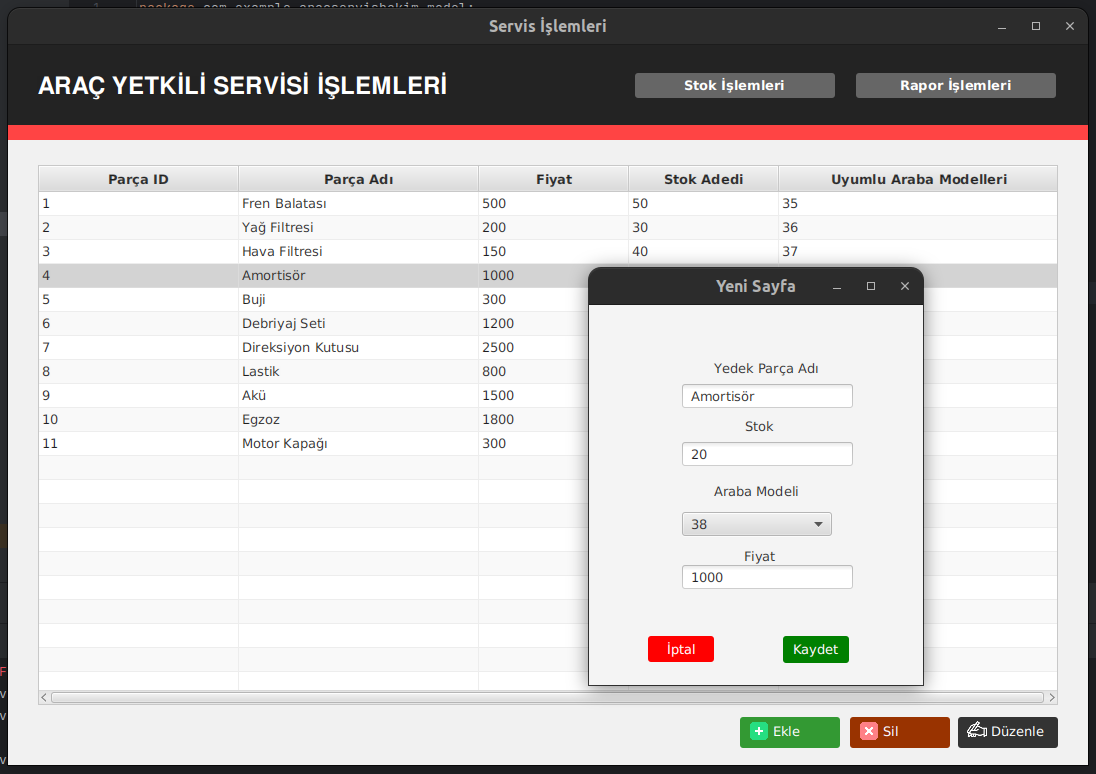
**AraclarımTabloSatır**(String markaModel, String plaka, String tamirSuresi, String periyodikBakimSuresi)

**ArizaTamirSatir**(int yedekParçaId, String yedekParca, int tamirSuresi, int iscilikUcreti, int toplamUcret)

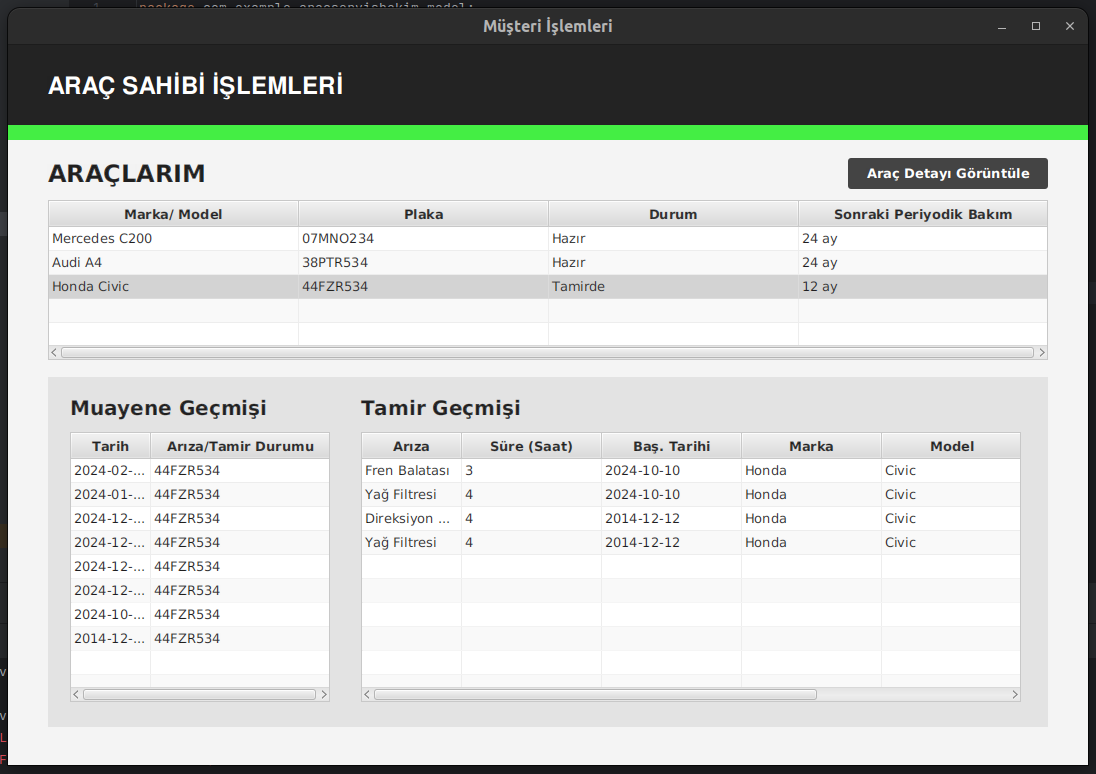
**Örn** Servis Teknikeri ekranı üzerindeki Stok İşlemleri Ekranı:



Yedek Parça bilgilerini görüntüleyen tablo ve ekleme, silme ve düzenleme ekranlarını açan butonlar yer almaktadır. Düzenle ekranı seçilen satır ile birlikte tıklandığında:



**Örn** Müşteri ekranı:



Bu arayüz üzerinde müşteri kendi araçlarına ait genel bilgileri ve istediği takdirde tablodan seçeceği aracın geçmiş muayene ve tamir detay bilgilerine erişebilmektedir.

1. **Eğer veri tabanı bağlama işlemini gerçekleştirmemiş iseniz VTYS sistemlerinde Transaction nedir açıklayınız ve çalışmanızdan bir Transaction örneği veriniz. (10 p)**
2. **View nedir açıklayınız ve bir adet view, bir adet saklı yordam (Stored Procedute) ifadesine ait SQL deyimlerinin sorgusunu ve cevabını yazınız.  
   (10 p)**

View (Görünüm), veritabanındaki bir veya birden fazla tablodan oluşturulan sanal bir tablodur. View, fiziksel olarak veri içermez, yalnızca verilerin bir sorgu sonucunda nasıl görüneceğini tanımlar. Genellikle karmaşık sorguları basitleştirmek, veri güvenliğini artırmak veya belirli verileri kolayca erişilebilir hale getirmek için kullanılır.

### 

Stored Procedure (Saklı Yordam) ise birden fazla SQL deyimini içerebilen ve belirli bir görevi yerine getiren, veritabanında saklanan ve gerektiğinde çağrılabilen bir program birimidir. Performansı artırmak ve tekrar eden işlemleri kolaylaştırmak için kullanılır.

#### **View Oluşturma (SQL Sorgusu):**

CREATE VIEW vw\_MusteriArabaDetay AS

SELECT

M.Ad AS MusteriAd,

M.Soyad AS MusteriSoyad,

M.Mail AS MusteriMail,

A.Marka AS ArabaMarka,

A.Modeli AS ArabaModeli,

A.Üretim\_Yılı AS UretimYili

FROM Musteri M

JOIN MüsteriAraba MA ON M.id = MA.Ruhsat\_Sahibi

JOIN ArabaModeli A ON MA.Araba\_Modeli = A.id;

SELECT \* FROM vw\_MusteriArabaDetay;

**Çıktı**:



**Stored Procedure Oluşturma (SQL Sorgusu):**

CREATE PROCEDURE sp\_MuayeneDetayByTarih

@BaslangicTarihi DATE

AS

BEGIN

SELECT

M.Ad AS MusteriAd,

M.Soyad AS MusteriSoyad,

MA.Plaka,

Mu.Tarih AS MuayeneTarihi,

Mu.Servis\_Teknikeri

FROM

Musteri M

JOIN

MüsteriAraba MA ON M.id = MA.Ruhsat\_Sahibi

JOIN

Muayene Mu ON MA.Plaka = Mu.Plaka

WHERE

Mu.Tarih > @BaslangicTarihi;

END;

EXEC sp\_MuayeneDetayByTarih @BaslangicTarihi = '2024-01-01';

**Çıktı**:

