**DDL:**-- Create ServisPersonel Table

CREATE TABLE ServisPersonel (

PersonelID INT PRIMARY KEY,

TamAd NVARCHAR(100),

Pozisyon NVARCHAR(50),

TelefonNumarasi NVARCHAR(15)

);

-- Create ServisTipleri Table

CREATE TABLE ServisTipleri (

ServisTipiID INT PRIMARY KEY,

ServisAdi NVARCHAR(100),

ServisAciklamasi NVARCHAR(MAX),

TabanUcret DECIMAL(10, 2)

);

-- Create Musteriler Table

CREATE TABLE Musteriler (

MusteriID INT PRIMARY KEY,

TamAd NVARCHAR(100) NOT NULL,

TelefonNumarasi NVARCHAR(15),

Email NVARCHAR(100),

Adres NVARCHAR(250)

);

-- Create Araclar Table

CREATE TABLE Araclar (

AracID INT PRIMARY KEY,

MusteriID INT NOT NULL FOREIGN KEY REFERENCES Musteriler(MusteriID),

Plaka NVARCHAR(15) NOT NULL,

Marka NVARCHAR(50),

Model NVARCHAR(50),

Yil INT,

SasiNumarasi NVARCHAR(20)

);

-- Create ServisKayitlari Table (with ServisID as an identity column)

CREATE TABLE ServisKayitlari (

ServisID INT IDENTITY(1,1) PRIMARY KEY, -- Make ServisID an identity column

AracID INT NOT NULL FOREIGN KEY REFERENCES Araclar(AracID),

ServisTarihi DATE NOT NULL,

PersonelID INT NOT NULL FOREIGN KEY REFERENCES ServisPersonel(PersonelID),

ServisTipiID INT NOT NULL FOREIGN KEY REFERENCES ServisTipleri(ServisTipiID),

Notlar NVARCHAR(MAX),

ToplamUcret DECIMAL(10, 2)

);

-- Create YedekParcalar Table

CREATE TABLE YedekParcalar (

ParcaID INT PRIMARY KEY,

ParcaAdi NVARCHAR(100),

StokMiktari INT NOT NULL,

BirimFiyat DECIMAL(10, 2)

);

-- Create ServisParcalari Table

CREATE TABLE ServisParcalari (

ServisParcaID INT PRIMARY KEY,

ServisID INT NOT NULL FOREIGN KEY REFERENCES ServisKayitlari(ServisID),

ParcaID INT NOT NULL FOREIGN KEY REFERENCES YedekParcalar(ParcaID),

KullanilanMiktar INT

);

-- Create Faturalar Table

CREATE TABLE Faturalar (

FaturaID INT PRIMARY KEY,

ServisID INT NOT NULL FOREIGN KEY REFERENCES ServisKayitlari(ServisID),

FaturaTarihi DATE NOT NULL,

ToplamTutar DECIMAL(10, 2)

);

**DML:**

-- Insert into Musteriler

INSERT INTO Musteriler VALUES (1, 'Ali Veli', '05341234567', 'ali.veli@gmail.com', 'Istanbul');

INSERT INTO Musteriler VALUES (2, 'Ayşe Yılmaz', '05349876543', 'ayse.yilmaz@gmail.com', 'Ankara');

INSERT INTO Musteriler VALUES (3, 'Fatma Gül', '05441234567', 'fatma.gul@gmail.com', 'Izmir');

INSERT INTO Musteriler VALUES (4, 'Emre Can', '05445566778', 'emre.can@gmail.com', 'Bursa');

-- Insert into Araclar

INSERT INTO Araclar VALUES (1, 1, '34ABC123', 'Toyota', 'Corolla', 2018, '1ABCDEFG123456789');

INSERT INTO Araclar VALUES (2, 2, '06DEF456', 'Honda', 'Civic', 2020, '2HIJKLMN987654321');

INSERT INTO Araclar VALUES (3, 3, '35XYZ678', 'Ford', 'Focus', 2015, '3ABCDEFGHI1234567');

INSERT INTO Araclar VALUES (4, 4, '16QWE999', 'Renault', 'Clio', 2019, '4JKLMNOPQ9876543');

-- Insert into ServisTipleri

INSERT INTO ServisTipleri VALUES (1, 'Yağ Değişimi', 'Motor yağının değiştirilmesi.', 300.00);

INSERT INTO ServisTipleri VALUES (2, 'Fren Kontrolü', 'Fren sisteminin kontrolü ve bakımı.', 500.00);

INSERT INTO ServisTipleri VALUES (3, 'Lastik Değişimi', 'Aracın lastiklerinin değiştirilmesi.', 400.00);

INSERT INTO ServisTipleri VALUES (4, 'Akü Kontrolü', 'Aracın aküsünün kontrol edilmesi.', 200.00);

-- Insert into ServisPersonel

INSERT INTO ServisPersonel VALUES (1, 'Ahmet Kaya', 'Teknisyen', '05342223344');

INSERT INTO ServisPersonel VALUES (2, 'Mehmet Demir', 'Usta Baş', '05345556677');

INSERT INTO ServisPersonel VALUES (3, 'Zeynep Acar', 'Teknisyen', '05345566789');

INSERT INTO ServisPersonel VALUES (4, 'Kerem Uzun', 'Danışman', '05347778899');

-- Insert into YedekParcalar

INSERT INTO YedekParcalar VALUES (1, 'Yağ Filtresi', 100, 50.00);

INSERT INTO YedekParcalar VALUES (2, 'Fren Diskleri', 50, 200.00);

INSERT INTO YedekParcalar VALUES (3, 'Lastik', 200, 500.00);

INSERT INTO YedekParcalar VALUES (4, 'Akü', 50, 800.00);

-- Insert into ServisKayitlari (ServisID is automatically generated)

INSERT INTO ServisKayitlari (AracID, ServisTarihi, PersonelID, ServisTipiID, Notlar, ToplamUcret)

VALUES (1, '2023-12-01', 1, 1, 'Yağ değişimi yapıldı.', 350.00);

INSERT INTO ServisKayitlari (AracID, ServisTarihi, PersonelID, ServisTipiID, Notlar, ToplamUcret)

VALUES (2, '2023-12-10', 2, 3, 'Lastik değişimi yapıldı.', 900.00);

INSERT INTO ServisKayitlari (AracID, ServisTarihi, PersonelID, ServisTipiID, Notlar, ToplamUcret)

VALUES (3, '2023-12-15', 3, 4, 'Akü kontrolü yapıldı.', 250.00);

-- Insert into ServisParcalari (Using the ServisID generated in ServisKayitlari)

-- (1, 1, 1, 1) means 1 Yağ Filtresi Kullanıldı

INSERT INTO ServisParcalari (ServisID, ParcaID, KullanilanMiktar)

VALUES (1, 1, 1); -- 1 Yağ Filtresi Kullanıldı

-- (2, 2, 3, 4) means 4 Lastik Kullanıldı

INSERT INTO ServisParcalari (ServisID, ParcaID, KullanilanMiktar)

VALUES (2, 3, 4); -- 4 Lastik Kullanıldı

-- (3, 3, 4, 1) means 1 Akü Kullanıldı

INSERT INTO ServisParcalari (ServisID, ParcaID, KullanilanMiktar)

VALUES (3, 4, 1); -- 1 Akü Kullanıldı

-- Insert into Faturalar

INSERT INTO Faturalar VALUES (1, 1, '2023-12-02', 350.00);

INSERT INTO Faturalar VALUES (2, 2, '2023-12-11', 900.00);

INSERT INTO Faturalar VALUES (3, 3, '2023-12-16', 250.00);

**SQL Queries:**SELECT \*

FROM Musteriler;

**Purpose**: Lists all customers in the database, displaying their details like name, phone, email, and address.

SELECT A.Plaka, A.Marka, A.Model

FROM Araclar A

WHERE A.MusteriID = 1;

**Purpose**: Displays all vehicles owned by a specific customer (with MusteriID = 1).

SELECT M.TamAd AS Musteri, A.Plaka, A.Marka, A.Model

FROM Musteriler M

JOIN Araclar A ON M.MusteriID = A.MusteriID;

**Purpose**: Retrieves a list of vehicles along with their owner's name.

SELECT ST.ServisAdi, SUM(SK.ToplamUcret) AS TotalRevenue

FROM ServisTipleri ST

JOIN ServisKayitlari SK ON ST.ServisTipiID = SK.ServisTipiID

GROUP BY ST.ServisAdi;

**Purpose**: Shows the total revenue generated by each service type.

SELECT ST.ServisAdi, SUM(SK.ToplamUcret) AS TotalRevenue

FROM ServisTipleri ST

JOIN ServisKayitlari SK ON ST.ServisTipiID = SK.ServisTipiID

GROUP BY ST.ServisAdi

HAVING SUM(SK.ToplamUcret) > 500;

**Purpose**: Filters service types with total revenue greater than 500.

SELECT SK.ServisID, A.Plaka, ST.ServisAdi, SP.TamAd AS Personel, YP.ParcaAdi, SPAR.KullanilanMiktar

FROM ServisKayitlari SK

JOIN Araclar A ON SK.AracID = A.AracID

JOIN ServisTipleri ST ON SK.ServisTipiID = ST.ServisTipiID

JOIN ServisPersonel SP ON SK.PersonelID = SP.PersonelID

JOIN ServisParcalari SPAR ON SK.ServisID = SPAR.ServisID

JOIN YedekParcalar YP ON SPAR.ParcaID = YP.ParcaID;

**Purpose**: Displays detailed service records, including the parts used and the assigned personnel.

SELECT A.Plaka, A.Marka, A.Model

FROM Araclar A

WHERE A.AracID NOT IN (SELECT SK.AracID FROM ServisKayitlari SK);

**Purpose**: Finds vehicles that have not received any service.

SELECT M.TamAd AS Musteri, COUNT(A.AracID) AS VehicleCount

FROM Musteriler M

LEFT JOIN Araclar A ON M.MusteriID = A.MusteriID

GROUP BY M.TamAd;

**Purpose**: Shows the number of vehicles owned by each customer.

SELECT \*

FROM vw\_ServiceDetails; -- Assuming the view vw\_ServiceDetails has been created

**Purpose**: Retrieves a pre-defined view showing detailed service information with total costs.

EXEC AddService

@AracID = 4,

@ServisTarihi = '2023-12-20',

@PersonelID = 2,

@ServisTipiID = 3,

@Notlar = 'Yeni lastikler takıldı.',

@ToplamUcret = 1200.00;

**Purpose**: Adds a new service record using a stored procedure.

SELECT SK.ServisID, A.Plaka,

ST.ServisAdi,

CASE

WHEN SK.ToplamUcret > 1000 THEN 'Premium Service'

WHEN SK.ToplamUcret > 500 THEN 'Standard Service'

ELSE 'Basic Service'

END AS ServiceCategory

FROM ServisKayitlari SK

JOIN Araclar A ON SK.AracID = A.AracID

JOIN ServisTipleri ST ON SK.ServisTipiID = ST.ServisTipiID;

**Purpose**: Categorizes services into "Premium", "Standard", or "Basic" based on their total cost.

BEGIN TRANSACTION;

-- Insert a new service record

DECLARE @NewServisID INT;

INSERT INTO ServisKayitlari (AracID, ServisTarihi, PersonelID, ServisTipiID, Notlar, ToplamUcret)

VALUES (4, '2023-12-25', 3, 1, 'Motor yağı değişimi yapıldı.', 350.00);

SET @NewServisID = SCOPE\_IDENTITY();

-- Update stock for a part (Assuming ParcaID = 1 for Yağ Filtresi)

UPDATE YedekParcalar

SET StokMiktari = StokMiktari - 1

WHERE ParcaID = 1; -- Yağ Filtresi

-- Manually calculate a new ServisParcaID (Assuming you're generating it)

DECLARE @NewServisParcaID INT;

SET @NewServisParcaID = (SELECT ISNULL(MAX(ServisParcaID), 0) + 1 FROM ServisParcalari);

-- Add part usage to ServisParcalari (Use valid ParcaID here)

INSERT INTO ServisParcalari (ServisParcaID, ServisID, ParcaID, KullanilanMiktar)

VALUES (@NewServisParcaID, @NewServisID, 1, 1); -- Yağ Filtresi

SET @NewServisParcaID = (SELECT ISNULL(MAX(ServisParcaID), 0) + 1 FROM ServisParcalari);

INSERT INTO ServisParcalari (ServisParcaID, ServisID, ParcaID, KullanilanMiktar)

VALUES (@NewServisParcaID, @NewServisID, 3, 4); -- 4 Lastik Kullanıldı

COMMIT TRANSACTION;

**Purpose:** Demonstrates a transaction involving multiple operations, including adding a service and updating part stock.