



Fırat Üniversitesi

Mühendislik Fakültesi Bilgisayar Mühendisliği

BMÜ329 VERİ TABANI SİSTEMLERİ

Proje Adı: Öğrenci Yurt Takip Otomasyonu

Proje Ekibi:

- 225260305 Efe Kocaefe
- 185260013 Serhat Yıldırım
- 215260087 Ahmet Bingöl

1.Proje Özeti

Projemiz öğrenci yurtlarında işleyişin kolaylaştırılmasını ve düzenli bir şekilde takip edilmesini sağlamaktadır. Projemiz yurtlarda kalan öğrenci ve çalışan personelin bilgileri tutulmak ve gerektiğinde düzenlenmesi hedeflemektedir. Projemiz ilişkisel veri tabanlarını kullanarak sistemi kolay bir hale getirmektedir

2. Proje Ayrıntıları:

2.1. Proje Amaçları:

- Öğrencilerin kayıtlarının takip edilmesini kolaylaştırmak
- Personel kayıtlarının ve yaptıkları işlerin düzenli ve sağlıklı bir şekilde kayıt altına almak
- Ödemeler ve kalan hesaplamaların kolaylaştırılması
- İşletmeyi zaman ve işleyiş bakımından hızlandırmak.

2.2. Kullanılan Teknolojiler:

- SQL Server Management Studio 2019
- İlişkisel Veritabanı Modelleri:

3. Proje Sonuçları:

Bu proje tamamlandığında, otomotiv tamirat ve tadilat işletmeleri aşağıdaki sonuçları elde edecektir:

- Yurttaki öğrenci kayıtlarının ve verilerinin düzenli takibi.
- Yurttaki öğrencilerin taksit ve kalan borç gibi maliyet hesaplamalarının takibi.
- Yurt ve öğrenci arasındaki ilişkilerin genel takibi. Örnek olarak çamaşır veya yemek çeşitleri gibi
- Personel ve yaptığı işlerin takibi

4.Tablolar

Toplamda 14 adet tablomuz vardır bunlar şöyledir.

Katlar

- KatId(PK)
- KatAdi

Odalar

- OdaId(PK)
- OdaNo
- KatId(FK)
- TipId(FK)

Oda Tipi

- TipId(PK)
- TipAdi

Bolumler

- BolumId(PK)
- BolumAdi

Kartlar

- KartId(PK)
- ÖğrenciId(FK)

Yemekler

- YemekId(PK)
- Cesit1
- Cesit2
- Cesit3
- Tarih

Veliler

- VeliId(PK)
- VeliAd
- VeliSoyad
- VeliTel

Tcler

- TcId(PK)
- Tc

Personel

- PersonelId(PK)
- PersonelAdi
- PersonelSoyadi
- PersonelTel
- TemizlikId(PK)

Temizlik

- PersonelId(FK)
- OdaId(FK)
- Tarih

Ogrenciler

- OgrenciId(PK)
- OgrenciAd
- OgrenciSoyad
- OdaId(FK)
- TcId(FK)
- BolumId(FK)
- Velid(FK)

Camasirlar

- CamasirId(PK)
- OgrenciId(FK)
- Tarih
- Ucret

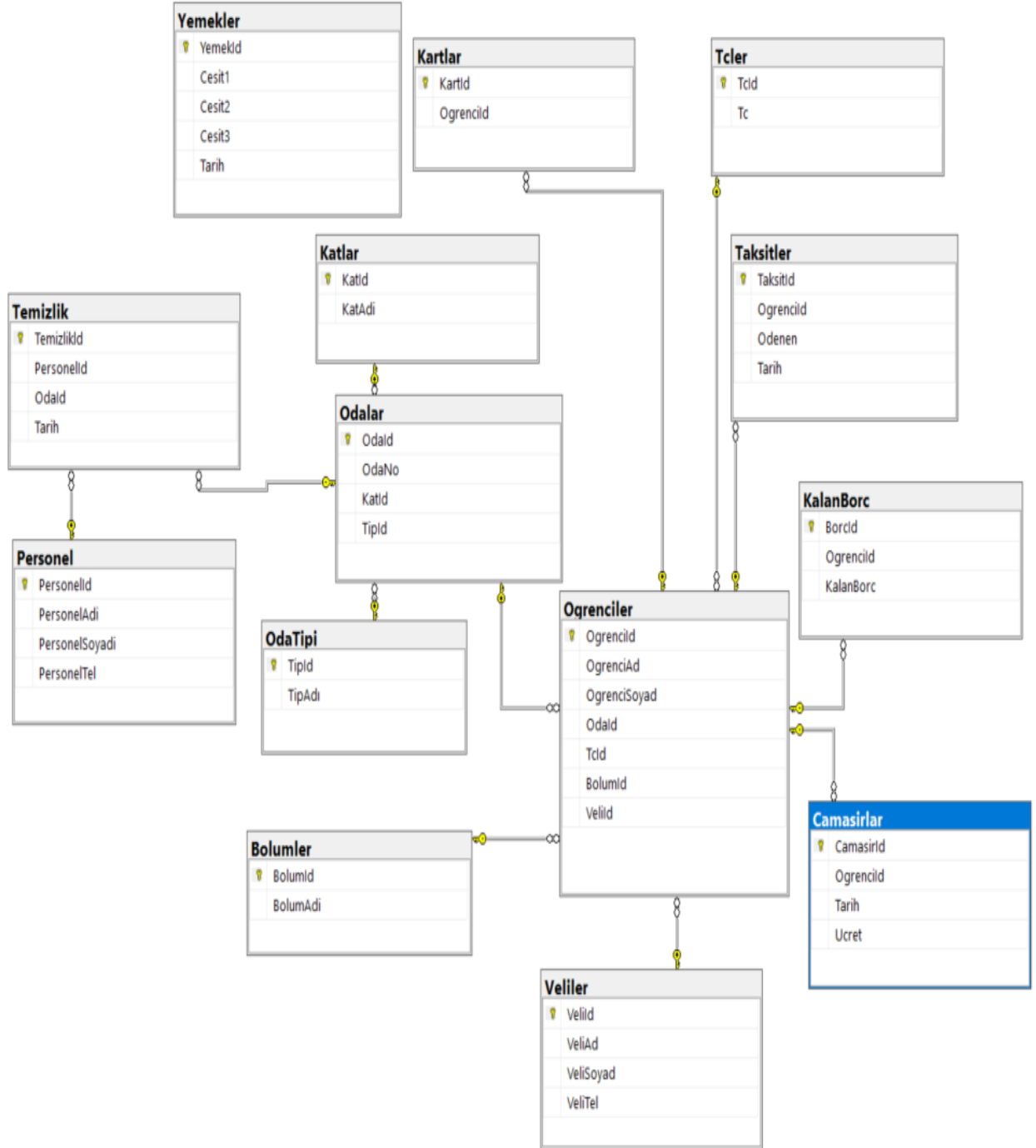
Taksitler

- TaksitId(PK)
- OgrenciId(FK)
- Miktar
- Tarih

KalanBorc

- Id(Pk)
- KalanBorc
- OgrenciId(FK)

5.E-R Diyagramı



6.Sql Komutları

CREATE KOMUTU

```
CREATE TABLE Veliler  
(  
  VeliId INT PRIMARY KEY,  
  VeliAd VARCHAR(50),  
  VeliSoyad VARCHAR(50),  
  VeliTel VARCHAR(11)  
);
```

SELECT KOMUTU

```
SELECT * FROM Veliler  
WHERE VeliAd="Mehmet";
```

UPDATE KOMUTU

```
Uptate Ogrenciler Set  
Odald=2 WHERE BolumId=3
```

DELETE KOMUTU

```
Delete from Ogrenciler  
WHERE Ogrencild=5;
```

VIEW KOMUTU

```
CREATE VIEW VW_TemizlikBilgisi
AS
SELECT
    T.TemizlikId,
    T.Tarih,
    P.PersonelAdi,
    P.PersonelSoyadi,
    P.PersonelTel,
    O.OdaNo,
    K.KatAdi,
    OT.TipAdi
FROM Temizlik AS T
INNER JOIN Personel AS P ON T.PersonelId =
P.PersonelId
INNER JOIN Odalar AS O ON T.OdaId =
O.OdaId
INNER JOIN Katlar AS K ON O.KatId = K.KatId
INNER JOIN OdaTipi AS OT ON O.TipId =
OT.TipId;
GO
```

Trigger Komutu

```
CREATE TRIGGER
TaksitEklendiğindeKalanBorçGüncelle
ON Taksitler
AFTER INSERT
AS
BEGIN
    -- Yeni eklenen değerleri al
    UPDATE KalanBorc
    SET KalanBorc = KalanBorc -
inserted.Odenen
    FROM inserted
    WHERE KalanBorc.OgrenciId =
inserted.OgrenciId;
END
```

Stored Procedure

```
CREATE PROCEDURE SP_OgrenciOdemeDetayGetir
    @Ogrencid INT
AS
BEGIN
    SET NOCOUNT ON;

    SELECT
        O.Ogrencid,
        O.OgrenciAd,
        O.OgrenciSoyad,
        T.TaksitId,
        T.Odenen AS 'OdenenTutar',
        T.Tarih AS 'OdemeTarihi',
        KB.KalanBorc AS 'KalanBorc'
    FROM Ogrenciler AS O
    LEFT JOIN Taksitler AS T ON O.Ogrencid =
    T.Ogrencid
    LEFT JOIN KalanBorc AS KB ON O.Ogrencid =
    KB.Ogrencid
    WHERE O.Ogrencid = @Ogrencid;
END;
GO
```

FUNCTION KOMUTU

```
CREATE FUNCTION FN_BorcluOgrenciler()
RETURNS TABLE
AS
RETURN(
    SELECT
        O.Ogrencid, O.OgrenciAd,
        O.OgrenciSoyad, KB.KalanBorc
    FROM Ogrenciler AS O
    INNER JOIN KalanBorc AS KB ON O.Ogrencid =
    KB.Ogrencid
    WHERE KB.KalanBorc > 0
);
GO
```

INDEX KOMUTU

```
CREATE NONCLUSTERED INDEX IX_Ogrenciler_TcId
ON Ogrenciler(TcId);
GO
```

TRANSACTION KOMUTU

```
BEGIN TRANSACTION;

BEGIN TRY

    INSERT INTO Taksitler (Ogrencid, Odenen, Tarih)
    VALUES (1, 500, GETDATE());

    UPDATE KalanBorc
    SET KalanBorc = KalanBorc - 500
    WHERE Ogrencid = 1;

    COMMIT TRANSACTION;

    PRINT 'Taksit ödeme ve borç güncelleme işlemi başarıyla
    tamamlandı.';

END TRY

BEGIN CATCH

    ROLLBACK TRANSACTION;

    SELECT
        ERROR_NUMBER() AS ErrorNumber,
        ERROR_MESSAGE() AS ErrorMessage;

    PRINT 'Bir hata oluştu! İşlemler geri alındı.';

END CATCH;
```

7.Sorgu Örnekleri

7.1)Ogrencilerin tüm bilgilerini yazdıran sql kodu

SELECT

O.OgrenciId,
O.OgrenciAd,
O.OgrenciSoyad,
OD.OdaNo AS OdaNumarasi,
K.KatAdi AS Kat,
B.BolumAdi AS Bolum,
V.VeliAd AS VeliAdi,
V.VeliSoyad AS VeliSoyadi,
V.VeliTel AS VeliTelefon,
P.PersonelAdi AS OdaSorumlusu,
OT.TipAdi AS OdaTipi

FROM

Ogrenciler O

LEFT JOIN Odalar OD ON O.OdaId = OD.OdaId

LEFT JOIN Katlar K ON OD.KatId = K.KatId

LEFT JOIN Bolumler B ON O.BolumId = B.BolumId

LEFT JOIN Veliler V ON O.VeliId = V.VeliId

LEFT JOIN Personel P ON OD.PersonelId = P.PersonelId

LEFT JOIN OdaTipi OT ON OD.TipId = OT.TipId;

7.2)Bilgisayar mühendisliğinde okuyan öğrencilerin kalan borç bilgisini veren sql kodu

SELECT

O.OgrenciId,
O.OgrenciAd,
O.OgrenciSoyad,
B.BolumAdi,
KB.Kalanborc

FROM


```
Ogrenciler O
INNER JOIN Bolumler B ON O.BolumId = B.BolumId
INNER JOIN KalanBorc KB ON O.OgrenciId = KB.OgrenciId
WHERE
    O.BolumId = 1;
```

7.3) Adı a ile başlayan öğrencilerin veli bilgilerini getiren sql kodu

```
SELECT
    Veliler.VeliAd,
    Veliler.VeliSoyad,
    Veliler.VeliTel
FROM
    Ogrenciler
JOIN
    Veliler ON Ogrenciler.VeliId = Veliler.VeliId
WHERE
    Ogrenciler.OgrenciAd LIKE 'A%';
```

7.4) Bir ay içinde yapılan taksit ödemelerinin ortalama bilgisini getiren sql kodu

```
SELECT
    AVG(Taksitler.Odenen) AS OrtalamaOdeme
FROM
    Taksitler
WHERE
    Tarih >= DATEADD(MONTH, -1, GETDATE()) AND Tarih <= GETDATE();
```

7.5)3.kattaki odaları temizleyen personel bilgisini veren sql kodu

```
SELECT
    Personel.PersonelAdi,
    Personel.PersonelSoyadi,
    Personel.PersonelTel
FROM
    Temizlik
JOIN
    Odalar ON Temizlik.OdaId = Odalar.OdaId
JOIN
    Katlar ON Odalar.KatId = Katlar.KatId
JOIN
    Personel ON Temizlik.PersonelId = Personel.PersonelId
WHERE
    Katlar.KatAdi = '3';
```

7.6) Her bir bölümde okuyan öğrencilerden ne kadar borç toplanmış ve kaç öğrencinin borcu olduğunu gösteren sql kodu

```
SELECT
    B.BolumAdi,
    COUNT(DISTINCT O.OgrenciId) AS BorcuOlanOgrenciSayisi,
    SUM(KB.KalanBorc) AS ToplamKalanBorc
FROM Bolumler AS B
INNER JOIN Ogrenciler AS O ON B.BolumId = O.BolumId
INNER JOIN KalanBorc AS KB ON O.OgrenciId = KB.OgrenciId
WHERE KB.KalanBorc > 0
GROUP BY B.BolumAdi
HAVING SUM(KB.KalanBorc) > 0
ORDER BY SUM(KB.KalanBorc) DESC;
```

7.7) Borcu olan öğrencilerin veli ad, soyad ve telefon numaralarını veren sql kodu

```
SELECT
    O.OgrenciId,
    O.OgrenciAd,
    O.OgrenciSoyad,
    V.VeliAd,
    V.VeliSoyad,
    V.VeliTel
FROM Ogrenciler AS O
INNER JOIN Veliler AS V ON O.VeliId = V.VeliId
INNER JOIN KalanBorc AS KB ON O.OgrenciId = KB.OgrenciId
WHERE KB.KalanBorc > 0
ORDER BY O.OgrenciSoyad;
```

7.8) Belirli bir tarih aralığındaki temizlik kayıtlarını ve ilgili personel ad-soyadlarını veren sql kodu

```
SELECT
    T.TemizlikId,
    T.Tarih,
    P.PersonelAdi,
    P.PersonelSoyadi,
    Od.OdaNo,
    K.KatAdi,
    OT.TipAdi AS OdaTipi
FROM Temizlik AS T
INNER JOIN Personel AS P ON T.PersonelId = P.PersonelId
INNER JOIN Odalar AS Od ON T.OdaId = Od.OdaId
INNER JOIN Katlar AS K ON Od.KatId = K.KatId
INNER JOIN OdaTipi AS OT ON Od.TipId = OT.TipId
WHERE T.Tarih BETWEEN '2025-01-01' AND '2025-01-31'
ORDER BY T.Tarih;
```