

BMÜ329 VERİ TABANI SİSTEMLERİ DERSİ DÖNEM PROJESİ

Proje Adı: Otel Otomasyon Sistemi

Proje Ekibi

- Aleyna YILMAZ
- Ömer Murat KARACA
- Ahmet El RECEP

Proje Gereksinimleri

Otel Otomasyon Sistemi veri tabanı projesinin gereksinim detaylarını alt kısımda açıkladık. Projemizde yer alan varlıklar, bu varlıkların sahip olduğu nitelikler ve aralarındaki ilişkiler belirtilmiştir.

Müşteri Gereksinimleri

Müşteri, oda tipine göre rezervasyon yapabilir. Belirlenmiş hizmetlerden faydalanabilir (havuz vb.) Müşteri, konaklama ve hizmet dökümü olan Check Out'u görüntüleyebilir. Müşteri, kendisine sunulmuş kampanyalardan yararlanabilir. Müşteri, araç plaka bilgisiyle otopark hizmetinden yararlanabilir.

Varlıklar ve İlişkiler

1. Otel Tablosu

- **Özellikler:**
- OtelID (PK): Otel kimlik numarası
- OtelAdi: Otel ismi
- Kapasite: Toplam oda kapasitesi
- YildizSayisi: Otel yıldız sayısı
- **İlişkiler:**
- Sahip Olur (Oda): Bir otel birden fazla odaya sahiptir (1:N)

2. Oda Tipi Tablosu

- **Özellikler:**
- OdaTipiID (PK): Oda kimlik numarası
- TipAdi: Odanın kategorisi
- GunlukFiyat: Odanın bir günlük fiyatı
- **İlişkiler:**
- Bağlıdır (Oda): Bir OdaTipi birden fazla odayı temsil edebilir (1:N)

3. Oda Tablosu

- **Özellikler:**
- OdaID (PK): Oda kimlik numarası
- OdaNo: Oda numarası
- OdaTipiID (FK): Oda türünün kimlik numarası
- Durum: Odanın doluluk durumu
- OtelID (FK): Bağlı olduğu otel
- **İlişkiler:**
- Ait Olur (Otel): Her oda bir otele aittir (N:1)
- Sahip Olur (OdaTipi): Her oda bir tipe sahiptir (N:1)
- Konaklama (Müşteri): Bir odada birden fazla müşteri kalabilir (1:N)

4. Müşteri Tablosu

- **Özellikler:**
- MusteriID (PK): Müşteri kimlik numarası
- Ad: Müşteri adı
- Soyad: Müşteri soyadı
- DogumTarihi: Doğum tarihi
- Cinsiyet: Cinsiyet bilgisi
- TCKimlikNo: TC kimlik numarası
- TelefonNo: İletişim numarası
- Plaka: Araç plakası
- GirişTarihi: Müşterinin otele giriş tarihi
- ÇıkışTarihi: Müşterinin otelden ayrılma tarihi
- OdaID: Müşterinin kaldığı odanın kimlik numarası
- OdaTipiID: Müşterinin kaldığı odanın türünün kimlik numarası
- KampanyaID: Müşteriye tanınan indirim kodunun kimlik numarası
- **İlişkiler:**
- Konaklama (Oda): Müşteri bir odada konaklayabilir (N:1)
- Kullanır (Hizmet): Müşteri birden fazla hizmet kullanabilir (N:M)
- Yararlanır (Kampanya): Müşteri kampanyalardan yararlanabilir (N:M)
- Kullanır (Otopark): Müşteri otopark hizmetinden yararlanabilir (N:1)

5. Hizmet Tablosu

- **Özellikler:**
- HizmetID (PK): Hizmet kimlik numarası
- HizmetAdi: Hizmet adı
- Ücret: Hizmet ücreti
- **İlişkiler:**
- Kullanılır (Müşteri): Bir hizmet birden fazla müşteri tarafından kullanılabilir (M:N)

6. Kampanya Tablosu

- **Özellikler:**

- KampanyaID (PK): Kampanya kimlik numarası
- KampanyaAdi: Kampanya adı
- IndirimYuzdesi: İndirim oranı

- **İlişkiler:**

- Yararlanılır (Müşteri): Bir kampanyadan birden fazla müşteri yararlanabilir (M:N)

7. Otopark Tablosu

- **Özellikler:**

- OtoparkID (PK): Otoparkın kimlik numarası
- ToplamKapasite: Otoparka toplam girebilecek araç sayısı
- DoluYer: Otoparkta bulunan araç sayısı
- BosYer: Otoparkın anlık boş yer sayısı

- **İlişkiler:**

- Yararlanılır (Müşteri): Otoparktan birden fazla müşteri yararlanabilir (M:N)

İlişki Özeti

İlişki	Varlık 1	Varlık 2	Kardinalite
Sahip Olur	Otel	Oda	1:N
Bağlıdır	OdaTipi	Oda	1:N
Ait Olur	Oda	Otel	N:1
Sahip Olur	Oda	OdaTipi	N:1
Konaklama	Oda	Müşteri	1:N
Konaklama	Müşteri	Oda	N:1
Kullanır	Müşteri	Hizmet	M:N
Kullanılır	Hizmet	Müşteri	M:N
Yararlanır	Müşteri	Kampanya	M:N
Yararlanılır	Kampanya	Müşteri	M:N
Kullanır	Müşteri	Otopark	N:1
Yararlanılır	Otopark	Müşteri	1:N

Veri Bütünlüğü ve Kısıtlamalar

- Tüm tablolarda Primary Key tanımlanmıştır
- Foreign Key ilişkileri ile referans bütünlüğü sağlanmıştır
- CHECK kısıtlamaları ile veri doğruluğu kontrol edilmektedir
- UNIQUE kısıtlamalar ile tekrar eden kayıtlar engellenmektedir

1. Trigger Sistemi

1.1 Müşteri İşlemleri Trigger'ları

- tr_Musteri_Insert: Müşteri eklendiğinde oda durumunu günceller
- tr_Musteri_Delete: Müşteri silindiğinde oda durumunu günceller
- tr_Musteri_Update: Müşteri bilgileri güncellendiğinde oda durumlarını kontrol eder

1.2 Otopark Trigger'ları

- trg_MusteriEkle: Yeni müşteri kaydında otopark durumunu günceller
- trg_MusteriSil: Müşteri silindiğinde otopark durumunu günceller

1.3 Ücretlendirme Trigger'ları

- trg_ToplamUcretHesapla: CheckOut işleminde toplam ücreti hesaplar
- trg_HizmetEklendiNetUcret: Yeni hizmet eklendiğinde net ücreti günceller

2. Stored Procedure'lar

2.1 Müşteri İşlemleri

- sp_MusteriEkle: Yeni müşteri kaydı oluşturur
- sp_MusteriSil: Müşteri kaydını siler
- sp_MusteriGuncelle: Müşteri bilgilerini günceller

3. Güvenlik Önlemleri

- TC Kimlik No ve telefon numarası benzersiz
- Oda kapasitesi ve müşteri sayısı kontrolleri

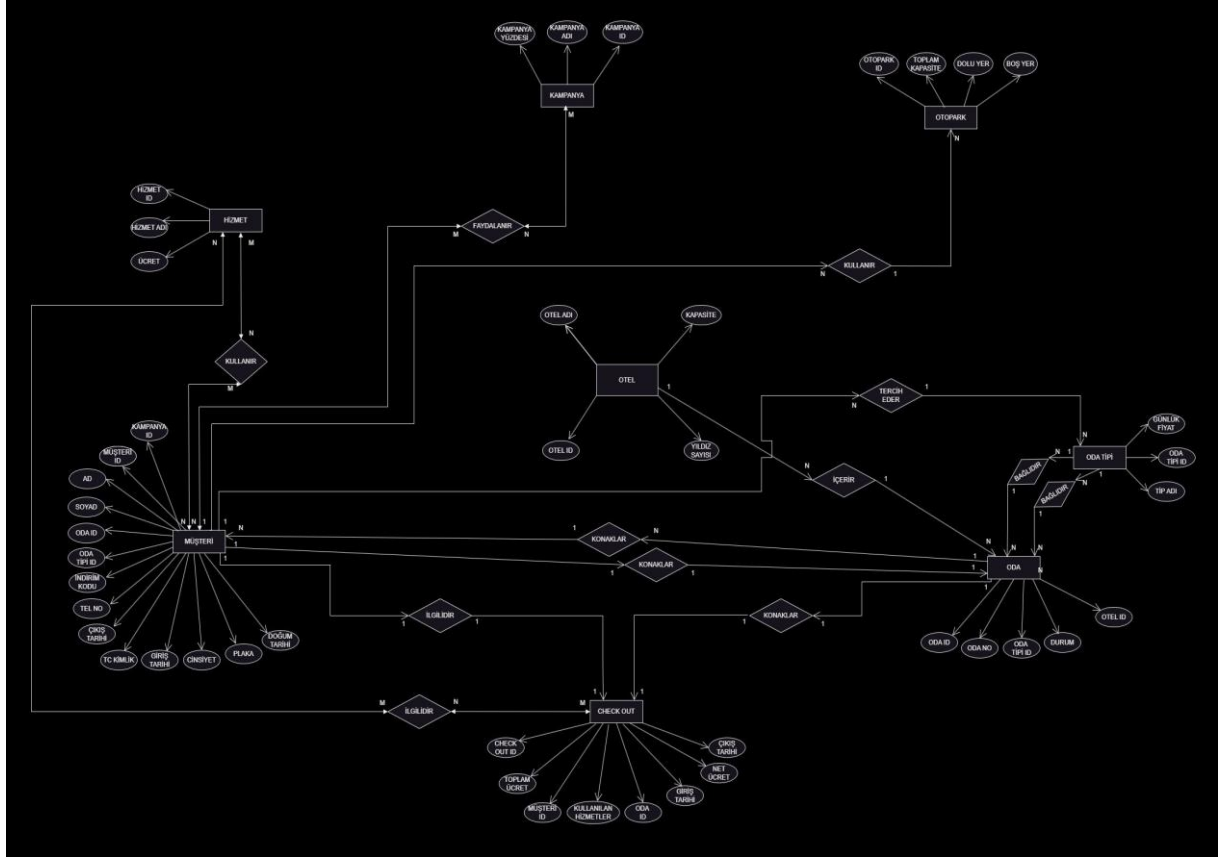
4. Optimizasyon

- Computed columns kullanımı (Otopark boş yer hesaplama)
- İndeksler ve ilişkiler ile performans optimizasyonu
- Transaction yönetimi ile veri tutarlılığı

5. Sonuç

Bu sistem, modern bir otelin tüm temel ihtiyaçlarını karşılayacak şekilde tasarlanmıştır. Trigger'lar ve stored procedure'lar ile otomatize edilmiş işlemler, veri bütünlüğü kontrolleri ve optimizasyon teknikleri ile güvenilir ve performanslı bir altyapı sunmaktadır.

E-R Diyagramı:



Proje Kodları:

--TABLOLARI OLUŞTURMA

-- OTEL TABLOSU

```
CREATE TABLE Otel (  
    OtelID INT PRIMARY KEY IDENTITY(1,1),  
    OtelAdi NVARCHAR(100) NOT NULL,  
    Kapasite INT NOT NULL CHECK (Kapasite > 0),  
    YildizSayisi INT NOT NULL CHECK (YildizSayisi BETWEEN 1 AND 5)  
);
```

-- ODA TİPLERİ TABLOSU

```
CREATE TABLE OdaTipi (  
    OdaTipiID INT PRIMARY KEY IDENTITY(1,1),  
    TipAdi NVARCHAR(50) NOT NULL UNIQUE,  
    GunlukFiyat DECIMAL(10,2) NOT NULL CHECK (GunlukFiyat > 0)  
);
```

-- ODALAR TABLOSU

```
CREATE TABLE Oda (  
    OdaID INT PRIMARY KEY IDENTITY(1,1),  
    OdaNo INT NOT NULL CHECK (OdaNo > 0),  
    OdaTipiID INT NOT NULL,  
    Durum BIT NOT NULL DEFAULT 0, -- 0: Boş, 1: Dolu  
    OtelID INT NOT NULL,  
    FOREIGN KEY (OdaTipiID) REFERENCES OdaTipi(OdaTipiID),  
    FOREIGN KEY (OtelID) REFERENCES Otel(OtelID)  
);
```

-- KAMPANYALAR TABLOSU

```
CREATE TABLE Kampanya (  
    KampanyaID INT PRIMARY KEY IDENTITY(1,1),  
    KampanyaAdi NVARCHAR(50) NOT NULL UNIQUE,  
    IndirimYuzdesi DECIMAL(5,2) NOT NULL CHECK (IndirimYuzdesi BETWEEN 0  
AND 100)  
);
```

-- MÜŞTERİ TABLOSU

```
CREATE TABLE Musteri (  
    MusteriID INT PRIMARY KEY IDENTITY(1,1),  
    Ad NVARCHAR(50) NOT NULL,  
    Soyad NVARCHAR(50) NOT NULL,
```

```

DogumTarihi DATE NOT NULL CHECK (DATEDIFF(YEAR, DogumTarihi,
GETDATE()) >= 18),
Cinsiyet NVARCHAR(10) NOT NULL CHECK (Cinsiyet IN ('Erkek', 'Kadın')),
TCKimlikNo NVARCHAR(11) UNIQUE NOT NULL,
TelefonNo NVARCHAR(15) NOT NULL,
Plaka NVARCHAR(20) NULL,
GirisTarihi DATE NOT NULL,
CikisTarihi DATE NOT NULL,
OdaID INT NULL,
OdaTipiID INT NULL, -- OdaTipiID sütunu eklendi
KampanyaID INT NULL, -- KampanyaID sütunu eklendi
FOREIGN KEY (OdaID) REFERENCES Oda(OdaID) ON DELETE SET NULL,
FOREIGN KEY (OdaTipiID) REFERENCES OdaTipi(OdaTipiID), -- OdaTipiID sütunu
ile ilişki
FOREIGN KEY (KampanyaID) REFERENCES Kampanya(KampanyaID) --
KampanyaID ile ilişki
);

```

```

-- HİZMETLER TABLOSU
CREATE TABLE Hizmet (
HizmetID INT PRIMARY KEY IDENTITY(1,1),
HizmetAdi NVARCHAR(50) NOT NULL UNIQUE,
Ucret DECIMAL(10,2) NOT NULL CHECK (Ucret >= 0)
);

```

```

-- MÜŞTERİ HİZMETLERİ TABLOSU
CREATE TABLE MusteriHizmet (
MusteriID INT NOT NULL,
HizmetID INT NOT NULL,
HizmetTarihi DATE NOT NULL,
PRIMARY KEY (MusteriID, HizmetID, HizmetTarihi),
FOREIGN KEY (MusteriID) REFERENCES Musteri(MusteriID),
FOREIGN KEY (HizmetID) REFERENCES Hizmet(HizmetID)
);

```

```

-- OTO PARK TABLOSU
CREATE TABLE Otopark (
OtoparkID INT PRIMARY KEY IDENTITY(1,1),
ToplamKapasite INT NOT NULL CHECK (ToplamKapasite > 0),
DoluYer INT NOT NULL DEFAULT 0 CHECK (DoluYer >= 0),
BosYer AS (ToplamKapasite - DoluYer) PERSISTED
);

```

```

-- CHECKOUT TABLOSU
CREATE TABLE CheckOut (
CheckOutID INT PRIMARY KEY IDENTITY(1,1),
MusteriID INT NOT NULL,
OdaID INT NOT NULL,
GirisTarihi DATE NOT NULL,
CikisTarihi DATE NOT NULL,

```

```

ToplamUcret DECIMAL(10,2) NOT NULL,
KullanilanHizmetler NVARCHAR(MAX) NULL,
NetUcret DECIMAL(10,2) NOT NULL,
OlusturmaTarihi DATETIME DEFAULT GETDATE(),
FOREIGN KEY (MusteriID) REFERENCES Musteri(MusteriID),
FOREIGN KEY (OdaID) REFERENCES Oda(OdaID)
);

--
-----
--

-- VERİ EKLEME

-- Otel Bilgileri
INSERT INTO Otel (OtelAdi, Kapasite, YildizSayisi) VALUES ('Lüks Otel', 200, 5);

-- Oda Tipleri
INSERT INTO OdaTipi (TipAdi, GunlukFiyat) VALUES
('Tek Kişilik', 300),
('Çift Kişilik', 500),
('Aile Odası', 700),
('Kral Dairesi', 2000);

-- Odalar
DECLARE @OdaNo INT = 1;
DECLARE @TipID INT;

-- Oda Tipi 1 (Tek Kişilik - 1-50)
SET @TipID = 1;
WHILE @OdaNo <= 50
BEGIN
    INSERT INTO Oda (OdaNo, OdaTipiID, Durum, OtelID) VALUES (@OdaNo, @TipID,
0, 1);
    SET @OdaNo = @OdaNo + 1;
END;

-- Oda Tipi 2 (Çift Kişilik - 51-100)
SET @TipID = 2;
WHILE @OdaNo <= 100
BEGIN
    INSERT INTO Oda (OdaNo, OdaTipiID, Durum, OtelID) VALUES (@OdaNo, @TipID,
0, 1);
    SET @OdaNo = @OdaNo + 1;
END;

-- Oda Tipi 3 (Aile Odası - 101-150)
SET @TipID = 3;
WHILE @OdaNo <= 150

```



```

BEGIN
    INSERT INTO Oda (OdaNo, OdaTipiID, Durum, OtelID) VALUES (@OdaNo, @TipID,
0, 1);
    SET @OdaNo = @OdaNo + 1;
END;

-- Oda Tipi 4 (Kral Dairesi - 151-200)
SET @TipID = 4;
WHILE @OdaNo <= 200
BEGIN
    INSERT INTO Oda (OdaNo, OdaTipiID, Durum, OtelID) VALUES (@OdaNo, @TipID,
0, 1);
    SET @OdaNo = @OdaNo + 1;
END;

-- Hizmetler
INSERT INTO Hizmet (HizmetAdi, Ucret) VALUES
('Spa', 200),
('Havuz', 100),
('Oda Servisi', 150),
('Fitness', 75);

-- Kampanyalar
INSERT INTO Kampanya (KampanyaAdi, IndirimYuzdesi) VALUES
('Erken Rezervasyon', 10),
('Özel İndirim', 20);

-- Otopark
INSERT INTO Otopark (ToplamKapasite, DoluYer) VALUES (70, 0);

--
-----
-----
--

-- TRIGGERLAR

--DROP TRIGGER IF EXISTS tr_Musteri_Insert

-- Müşteri Eklendiğinde Oda Durumu Güncelle

CREATE TRIGGER tr_Musteri_Insert
ON Musteri
AFTER INSERT
AS
BEGIN
    UPDATE Oda
    SET Durum = 1
    WHERE OdaID IN (SELECT OdaID FROM Inserted);

```

```
END;  
GO
```

```
-- Müşteri Silindiğinde Oda Durumu Güncelle
```

```
CREATE TRIGGER tr_Musteri_Delete  
ON Musteri  
AFTER DELETE  
AS  
BEGIN  
    UPDATE Oda  
    SET Durum = 0  
    WHERE OdaID IN (SELECT OdaID FROM Deleted);  
END;  
GO
```

```
-- Müşteri Güncellendiğinde Oda Durumu Kontrolü
```

```
CREATE TRIGGER tr_Musteri_Update  
ON Musteri  
AFTER UPDATE  
AS  
BEGIN  
    -- Önce eski odanın durumunu güncelle  
    UPDATE Oda  
    SET Durum = 0  
    WHERE OdaID IN (SELECT OdaID FROM Deleted);  
  
    -- Yeni odanın durumunu güncelle  
    UPDATE Oda  
    SET Durum = 1  
    WHERE OdaID IN (SELECT OdaID FROM Inserted);  
END;  
GO
```

```
--  
-----  
-----  
--
```

```
--DROP TRIGGER IF EXISTS trg_MusteriEkle
```

```
--DROP TRIGGER IF EXISTS trg_MusteriSil
```

```
-- Otopark Durumunu Güncelleme: Müşteri Eklenmesi
```

```
CREATE TRIGGER trg_MusteriEkle
ON Musteri
AFTER INSERT
AS
BEGIN
    DECLARE @PlakaSayisi INT;

    SELECT @PlakaSayisi = COUNT(*)
    FROM Inserted
    WHERE Plaka IS NOT NULL;

    IF @PlakaSayisi > 0
    BEGIN
        UPDATE Otopark
        SET DoluYer = DoluYer + @PlakaSayisi
        WHERE OtoparkID = 1;
    END;
END;
GO

-- Otopark Durumunu Güncelleme: Müşteri Silinmesi

CREATE TRIGGER trg_MusteriSil
ON Musteri
AFTER DELETE
AS
BEGIN
    DECLARE @PlakaSayisi INT;

    SELECT @PlakaSayisi = COUNT(*)
    FROM Deleted
    WHERE Plaka IS NOT NULL;

    IF @PlakaSayisi > 0
    BEGIN
        UPDATE Otopark
        SET DoluYer = DoluYer - @PlakaSayisi
        WHERE OtoparkID = 1;
    END;
END;
GO

--
-----
-----
--

--DROP TRIGGER IF EXISTS trg_ToplamUcretHesapla
--DROP TRIGGER IF EXISTS trg_HizmetEklendiNetUcret
```

-- Toplam Ücret Hesaplama Trigger

CREATE TRIGGER trg_ToplamUcretHesapla

ON CheckOut

AFTER INSERT

AS

BEGIN

-- Değişkenler tanımlanır

DECLARE @CheckOutID INT;

DECLARE @MusteriID INT;

DECLARE @OdaID INT;

DECLARE @GunSayisi INT;

DECLARE @GunlukFiyat DECIMAL(10,2);

DECLARE @IndirimYuzdesi DECIMAL(5,2) = 0;

DECLARE @ToplamUcret DECIMAL(10,2);

-- Eklenen kayıttan müşteri ve oda bilgileri alınır

SELECT

@CheckOutID = i.CheckOutID,

@MusteriID = i.MusteriID,

@OdaID = i.OdaID,

@GunSayisi = DATEDIFF(DAY, i.GirisTarihi, i.CikisTarihi)

FROM inserted i;

-- Oda tipi ücretini bul

SELECT @GunlukFiyat = ot.GunlukFiyat

FROM Oda o

JOIN OdaTipi ot ON o.OdaTipiID = ot.OdaTipiID

WHERE o.OdaID = @OdaID;

-- Toplam Ücreti hesapla

SET @ToplamUcret = @GunSayisi * @GunlukFiyat;

-- Kampanya indirimi varsa hesapla

SELECT @IndirimYuzdesi = k.IndirimYuzdesi

FROM Musteri m

JOIN Kampanya k ON m.KampanyaID = k.KampanyaID

WHERE m.MusteriID = @MusteriID;

-- Toplam Ücret ve Net Ücret hesaplanır ve güncellenir

UPDATE CheckOut

SET

ToplamUcret = @ToplamUcret,

NetUcret = @ToplamUcret - (@ToplamUcret * @IndirimYuzdesi / 100)

WHERE CheckOutID = @CheckOutID;

END;

GO

--Net ücret güncelleme trigger

```

CREATE TRIGGER trg_HizmetEklendiNetUcret
ON MusteriHizmet
AFTER INSERT
AS
BEGIN
    -- Değişkenler tanımlanır
    DECLARE @MusteriID INT;
    DECLARE @HizmetID INT;
    DECLARE @HizmetUcreti DECIMAL(10,2);

    -- Eklenen kayıttan bilgiler alınır
    SELECT
        @MusteriID = i.MusteriID,
        @HizmetID = i.HizmetID
    FROM inserted i;

    -- Hizmetin ücretini bul
    SELECT @HizmetUcreti = h.Ucret
    FROM Hizmet h
    WHERE h.HizmetID = @HizmetID;

    -- Net Ücreti güncelle
    UPDATE CheckOut
    SET NetUcret = NetUcret + @HizmetUcreti
    WHERE MusteriID = @MusteriID;
END;
GO

```

```

--
-----
-----
--

```

```

DROP PROCEDURE IF EXISTS sp_MusteriEkle
DROP PROCEDURE IF EXISTS sp_MusteriSil
DROP PROCEDURE IF EXISTS sp_MusteriGuncelle

```

```

-- STORED PROCEDURES

```

```

--Müşteri ekleme prosedürü

```

```

CREATE PROCEDURE sp_MusteriEkle
    @Ad NVARCHAR(50),
    @Soyad NVARCHAR(50),
    @DogumTarihi DATE,
    @Cinsiyet NVARCHAR(10),
    @TCKimlikNo NVARCHAR(11),

```

```

@TelefonNo NVARCHAR(15),
@Plaka NVARCHAR(20),
@GirisTarihi DATE,
@CikisTarihi DATE,
@OdaID INT
AS
BEGIN
    BEGIN TRY
        BEGIN TRANSACTION;

        -- Oda var mı kontrol et
        IF NOT EXISTS (SELECT 1 FROM Oda WHERE OdaID = @OdaID)
        BEGIN
            ROLLBACK;
            THROW 50001, 'Böyle bir oda bulunmamaktadır.', 1;
        END;

        -- Oda dolu mu kontrol et
        IF EXISTS (SELECT 1 FROM Oda WHERE OdaID = @OdaID AND Durum = 1)
        BEGIN
            ROLLBACK;
            THROW 50002, 'Seçilen oda doludur.', 1;
        END;

        -- Müşteri ekle
        INSERT INTO Musteri (Ad, Soyad, DogumTarihi, Cinsiyet, TCKimlikNo, TelefonNo,
Plaka, GirisTarihi, CikisTarihi, OdaID)
        VALUES (@Ad, @Soyad, @DogumTarihi, @Cinsiyet, @TCKimlikNo, @TelefonNo,
@Plaka, @GirisTarihi, @CikisTarihi, @OdaID);

        -- Oda durumunu güncelle
        UPDATE Oda
        SET Durum = 1
        WHERE OdaID = @OdaID;

        COMMIT;
    END TRY
    BEGIN CATCH
        ROLLBACK;
        THROW;
    END CATCH;
END;
GO

--Müşteri silme prosedürü

CREATE PROCEDURE sp_MusteriSil
    @MusteriID INT
AS
BEGIN

```

```

BEGIN TRY
    BEGIN TRANSACTION;

    -- Silinecek müşterinin oda bilgisini al
    DECLARE @OdaID INT;
    SELECT @OdaID = OdaID FROM Musteri WHERE MusteriID = @MusteriID;

    -- Müşteri sil
    DELETE FROM Musteri WHERE MusteriID = @MusteriID;

    -- Oda durumunu güncelle (müşteri varsa dolu değilse boş)
    IF @OdaID IS NOT NULL
    BEGIN
        UPDATE Oda
        SET Durum = 0
        WHERE OdaID = @OdaID;
    END;

    COMMIT;
END TRY
BEGIN CATCH
    ROLLBACK;
    THROW;
END CATCH;
END;
GO

--Müşteri güncelleme prosedürü

CREATE PROCEDURE sp_MusteriGuncelle
    @MusteriID INT,
    @Ad NVARCHAR(50),
    @Soyad NVARCHAR(50),
    @DogumTarihi DATE,
    @Cinsiyet NVARCHAR(10),
    @TelefonNo NVARCHAR(15),
    @Plaka NVARCHAR(20),
    @GirisTarihi DATE,
    @CikisTarihi DATE,
    @OdaID INT
AS
BEGIN
    BEGIN TRY
        BEGIN TRANSACTION;

        -- Güncellenen müşterinin eski oda bilgisini al
        DECLARE @EskiOdaID INT;
        SELECT @EskiOdaID = OdaID FROM Musteri WHERE MusteriID = @MusteriID;

        -- Oda var mı kontrol et

```

```

IF NOT EXISTS (SELECT 1 FROM Oda WHERE OdaID = @OdaID)
BEGIN
    ROLLBACK;
    THROW 50001, 'Böyle bir oda bulunmamaktadır.', 1;
END;

-- Yeni oda dolu mu kontrol et
IF EXISTS (SELECT 1 FROM Oda WHERE OdaID = @OdaID AND Durum = 1 AND
OdaID != @EskiOdaID)
BEGIN
    ROLLBACK;
    THROW 50002, 'Seçilen oda doludur.', 1;
END;

-- Müşteri güncelle
UPDATE Musteri
SET Ad = @Ad,
    Soyad = @Soyad,
    DogumTarihi = @DogumTarihi,
    Cinsiyet = @Cinsiyet,
    TelefonNo = @TelefonNo,
    Plaka = @Plaka,
    GirisTarihi = @GirisTarihi,
    CikisTarihi = @CikisTarihi,
    OdaID = @OdaID
WHERE MusteriID = @MusteriID;

-- Eski oda boşalt
IF @EskiOdaID IS NOT NULL AND @EskiOdaID != @OdaID
BEGIN
    UPDATE Oda
    SET Durum = 0
    WHERE OdaID = @EskiOdaID;
END;

-- Yeni oda doldur
UPDATE Oda
SET Durum = 1
WHERE OdaID = @OdaID;

COMMIT;
END TRY
BEGIN CATCH
    ROLLBACK;
    THROW;
END CATCH;
END;
GO

```

--

--

--TABLOLARI GÖRÜNTÜLEME--

-- Otel Tablosu Verileri
SELECT * FROM Otel;

-- OdaTipi Tablosu Verileri
SELECT * FROM OdaTipi;

-- Oda Tablosu Verileri
SELECT * FROM Oda;

-- Musteri Tablosu Verileri
SELECT * FROM Musteri;

-- Hizmet Tablosu Verileri
SELECT * FROM Hizmet;

-- MusteriHizmet Tablosu Verileri
SELECT * FROM MusteriHizmet;

-- Kampanya Tablosu Verileri
SELECT * FROM Kampanya;

-- Otopark Tablosu Verileri
SELECT * FROM Otopark;

-- CheckOut Tablosu Verileri
SELECT * FROM CheckOut;

--

--

--MÜŞTERİ VERİ SETİMİZ--

INSERT INTO Musteri (Ad, Soyad, DogumTarihi, Cinsiyet, TCKimlikNo, TelefonNo, Plaka, GirişTarihi, ÇıkışTarihi, OdaID, OdaTipiID, KampanyaID) VALUES

-- Tek Kişilik Odalar (OdaTipiID = 1)
('Ahmet', 'Yılmaz', '1985-03-15', 'Erkek', '12344678901', '05321234567', '34ABC123', '2025-01-01', '2025-01-05', 1, 1, NULL),
('Ayşe', 'Kara', '1990-07-22', 'Kadın', '12345678902', '05332345678', '34DEF456', '2025-01-02', '2025-01-06', 2, 1, NULL),

('Mehmet', 'Çelik', '1992-11-11', 'Erkek', '12345678903', '05343456789', '34GHI789', '2025-01-03', '2025-01-07', 3, 1, NULL),
('Fatma', 'Demir', '1988-05-18', 'Kadın', '12345678904', '05354567890', '35JKL012', '2025-01-04', '2025-01-08', 4, 1, 2),
('Murat', 'Demir', '1992-02-15', 'Erkek', '12345678934', '05367890135', '34ABC130', '2025-01-06', '2025-01-10', 5, 1, NULL),
('Fatma', 'Kara', '1993-11-20', 'Kadın', '12345674935', '05378901240', '34DEF460', '2025-01-06', '2025-01-10', 6, 1, 1),
('Ali', 'Şahin', '1988-09-10', 'Erkek', '12345678936', '05389012357', '34GHI793', '2025-01-07', '2025-01-11', 7, 1, NULL),
('Ayşe', 'Yılmaz', '1991-04-05', 'Kadın', '12345678937', '05390123468', NULL, '2025-01-07', '2025-01-11', 8, 1, 2),
('Can', 'Çelik', '1987-03-12', 'Erkek', '12345678938', '05301234579', NULL, '2025-01-08', '2025-01-12', 9, 1, NULL),
('Eda', 'Ak', '1990-07-25', 'Kadın', '12345678939', '05312345680', '34PQR680', '2025-01-08', '2025-01-12', 10, 1, NULL),
('Berk', 'Öz', '1994-01-30', 'Erkek', '12345678940', '05323456791', '34STU903', '2025-01-09', '2025-01-13', 11, 1, NULL),
('Gamze', 'Koca', '1992-10-22', 'Kadın', '12345678941', '05334567802', '34VWX236', '2025-01-09', '2025-01-13', 12, 1, 2),
('İsmail', 'Yıldız', '1990-06-17', 'Erkek', '12345678942', '05345678913', '34XYZ012', '2025-01-10', '2025-01-14', 13, 1, NULL),
('Seda', 'Tunç', '1995-12-09', 'Kadın', '12341678943', '05356789024', NULL, '2025-01-10', '2025-01-14', 14, 1, NULL),

-- Çift Kişilik Odalar (OdaTipiID = 2)

('Elif', 'Taşdemir', '1989-03-30', 'Kadın', '12345678906', '05376789012', '06PQR678', '2025-01-01', '2025-01-05', 51, 2, 1),
('Hakan', 'Öztürk', '1995-06-14', 'Erkek', '12345678907', '05387890123', '34STU901', '2025-01-01', '2025-01-05', 51, 2, NULL),
('Fatih', 'Öztürk', '1985-07-14', 'Erkek', '12345678944', '05367890136', NULL, '2025-01-01', '2025-01-10', 53, 2, NULL),
('Hilal', 'Kara', '1994-08-25', 'Kadın', '12345678945', '05378901247', '34DEF463', '2025-01-01', '2025-01-10', 53, 2, 1),
('Zeynep', 'Ak', '1988-12-05', 'Kadın', '12345678946', '05389012358', NULL, '2025-01-07', '2025-01-11', 54, 2, NULL),
('Efe', 'Kar', '1991-01-15', 'Erkek', '12345678947', '05390123469', '34JKL015', '2025-01-07', '2025-01-11', 54, 2, NULL),
('Gamze', 'Yılmaz', '1990-06-02', 'Kadın', '12345678948', '05301234580', '34MNO350', '2025-01-08', '2025-01-12', 55, 2, NULL),
('Serkan', 'Demir', '1993-05-10', 'Erkek', '12345678949', '05312345681', '34PQR683', '2025-01-08', '2025-01-12', 55, 2, 1),
('Cansu', 'Çelik', '1992-03-17', 'Kadın', '12345678950', '05323456792', NULL, '2025-01-09', '2025-01-13', 56, 2, NULL),
('Canan', 'Yıldız', '1995-09-20', 'Kadın', '12345678951', '05334567803', NULL, '2025-01-09', '2025-01-13', 56, 2, 1),
('Kadir', 'Bursalı', '1989-11-18', 'Erkek', '12345678952', '05345678914', '34XYZ014', '2025-01-10', '2025-01-14', 57, 2, NULL),
('Buse', 'Güzel', '1990-12-02', 'Kadın', '12345678953', '05356789025', '34ABC136', '2025-01-10', '2025-01-14', 57, 2, 1),

('Ahmet', 'Yılmaz', '1980-03-15', 'Erkek', '12345678924', '05367890123', '34XYZ100', '2025-01-01', '2025-01-04', 58, 2, NULL),
('Seda', 'Öztürk', '1995-06-12', 'Kadın', '12345678925', '05378901234', '34ABC567', '2025-01-01', '2025-01-04', 58, 2, 2),
('Cem', 'Karaca', '1990-01-01', 'Erkek', '12345678926', '05389012345', NULL, '2025-01-03', '2025-01-06', 59, 2, 1),
('Yasemin', 'Duman', '1987-11-22', 'Kadın', '12345678927', '05390123456', '34GHI123', '2025-01-03', '2025-01-06', 59, 2, NULL),
('Emre', 'Can', '1985-07-07', 'Erkek', '12345678928', '05301234567', '34JKL456', '2025-01-05', '2025-01-07', 60, 2, 2),
('Elif', 'Polat', '1992-02-18', 'Kadın', '12345678929', '05312345678', '34MNO789', '2025-01-05', '2025-01-07', 60, 2, NULL),
('Mustafa', 'Güler', '1989-05-10', 'Erkek', '12345678930', '05323456789', '34PQR012', '2025-01-08', '2025-01-10', 61, 2, 1),
('Zeynep', 'Aydın', '1993-09-14', 'Kadın', '12345678931', '05334567890', '34STU345', '2025-01-08', '2025-01-10', 61, 2, NULL),
('Serkan', 'Kara', '1991-11-05', 'Erkek', '12345678932', '05345678901', '34VWX678', '2025-01-09', '2025-01-12', 62, 2, 2),
('Fatma', 'Yıldırım', '1994-04-09', 'Kadın', '12345678933', '05356789012', '34XYZ234', '2025-01-09', '2025-01-12', 62, 2, NULL),

-- Aile Odası (OdaTipiID = 3)

('Selim', 'Ak', '1980-01-12', 'Erkek', '11115678909', '05003123318', '41YZA567', '2025-01-01', '2025-01-06', 101, 3, 2),
('Aslı', 'Ak', '1983-04-19', 'Kadın', '12222278910', '05323856777', '41BCD890', '2025-01-01', '2025-01-06', 101, 3, 1),
('Zeynep', 'Ak', '2001-07-25', 'Kadın', '12345678911', '05334567890', NULL, '2025-01-01', '2025-01-06', 101, 3, NULL),
('Emre', 'Özdemir', '1987-06-25', 'Erkek', '12115673912', '05345677910', '41JKL123', '2025-01-02', '2025-01-07', 102, 3, 1),
('Ebru', 'Özdemir', '1991-09-15', 'Kadın', '12343378913', '05356782026', '41MNO456', '2025-01-02', '2025-01-07', 102, 3, NULL),
('Berk', 'Ak', '1985-05-20', 'Erkek', '12345677714', '05367890123', '41PQR789', '2025-01-03', '2025-01-08', 103, 3, NULL),
('Seda', 'Ak', '1992-11-05', 'Kadın', '12345678563', '05378221464', '41STU012', '2025-01-03', '2025-01-08', 103, 3, 2),
('Mehmet', 'Küçük', '1984-06-16', 'Erkek', '12396324934', '05367890123', '34ABC876', '2025-01-01', '2025-01-06', 104, 3, 1),
('Ayşe', 'Küçük', '1990-07-09', 'Kadın', '12345555555', '05378901234', NULL, '2025-01-01', '2025-01-06', 104, 3, NULL),
('Orhan', 'Küçük', '1986-08-25', 'Erkek', '12345675375', '05389012345', NULL, '2025-01-03', '2025-01-06', 104, 3, NULL),
('İrem', 'Küçük', '1993-11-11', 'Kadın', '12345678975', '05390223456', '34JKL989', '2025-01-04', '2025-01-06', 104, 3, 1),
('Cevdet', 'Büyük', '1988-03-18', 'Erkek', '12345778938', '05301114567', '34MNO012', '2025-01-06', '2025-01-09', 105, 3, NULL),
('Hülya', 'Büyük', '1992-12-25', 'Kadın', '12345649999', '05312885678', NULL, '2025-01-06', '2025-01-09', 105, 3, NULL),
('Burak', 'Büyük', '1989-04-10', 'Erkek', '12345677770', '05323456789', '34STU678', '2025-01-06', '2025-01-09', 105, 3, NULL),

('Elif', 'Balci', '1991-02-20', 'Kadin', '23589641785', '05334567844', NULL, '2025-01-08', '2025-01-11', 106, 3, NULL),
('Alper', 'Balci', '1987-09-03', 'Erkek', '12345178222', '05345696901', '34XYZ345', '2025-01-08', '2025-01-11', 106, 3, 2),
('Büşra', 'Balci', '1994-10-05', 'Kadin', '11344589111', '05356782012', '34ABC123', '2025-01-08', '2025-01-11', 106, 3, NULL),

-- Kral Dairesi (OdaTipiID = 4)

('Cem', 'Kilic', '1985-03-30', 'Erkek', '12345678916', '05389012345', '34ABC145', '2025-01-02', '2025-01-06', 151, 4, 1),
('Aylin', 'Balci', '1992-04-18', 'Kadin', '12345678917', '05390123456', '34DEF470', '2025-01-02', '2025-01-06', 151, 4, NULL),
('Deniz', 'Çelik', '1988-11-11', 'Kadin', '12345678918', '05301234567', '34GHI799', '2025-01-03', '2025-01-08', 152, 4, NULL),
('Kaan', 'Özdemir', '1993-05-25', 'Erkek', '12345678919', '05312347878', NULL, '2025-01-04', '2025-01-09', 153, 4, 1),
('Eda', 'Duru', '1986-07-22', 'Kadin', '12345678920', '05323456789', '34MNO370', '2025-01-05', '2025-01-10', 154, 4, NULL),
('Gökhan', 'Arslan', '1990-12-02', 'Erkek', '12345678921', '05334567890', NULL, '2025-01-05', '2025-01-10', 154, 4, 2),
('Seda', 'Acar', '1991-08-17', 'Kadin', '12345678922', '05345678901', '34STU013', '2025-01-08', '2025-01-12', 155, 4, NULL),
('Baran', 'Erdem', '1989-02-25', 'Erkek', '12345600923', '05356789012', '34VWX242', '2025-01-08', '2025-01-12', 155, 4, 1),
('Gökhan', 'Demirtaş', '1990-11-11', 'Erkek', '10045678944', '05367890123', NULL, '2025-01-01', '2025-01-04', 156, 4, 1),
('Nihan', 'Demirtaş', '1985-02-28', 'Kadin', '12340078945', '05378901234', '34ABC654', '2025-01-01', '2025-01-04', 156, 4, NULL),
('Burak', 'Çelik', '1988-07-14', 'Erkek', '12345670046', '05389012345', '34DEF987', '2025-01-03', '2025-01-05', 157, 4, NULL),
('Meryem', 'Öztürk', '1993-03-08', 'Kadin', '12344256947', '05390123456', '34GHI345', '2025-01-03', '2025-01-05', 157, 4, 1),
('Serkan', 'Yazici', '1989-09-19', 'Erkek', '77345678948', '05301234567', '34JKL123', '2025-01-05', '2025-01-06', 158, 4, 2),
('Leyla', 'Erdem', '1991-12-02', 'Kadin', '12373678949', '05312345228', '34MNO567', '2025-01-05', '2025-01-06', 158, 4, NULL),
('Selim', 'Koç', '1992-06-25', 'Erkek', '12315888650', '05323456789', '34PQR890', '2025-01-07', '2025-01-09', 159, 4, 1),
('Hülya', 'Acar', '1987-01-15', 'Kadin', '10345678451', '05334562290', '34STU432', '2025-01-07', '2025-01-09', 159, 4, NULL),
('Volkan', 'Şahin', '1994-05-30', 'Erkek', '10745678952', '05345678901', NULL, '2025-01-10', '2025-01-12', 160, 4, 2),
('Büşra', 'Kara', '1993-08-17', 'Kadin', '10845644953', '05356789112', '34XYZ678', '2025-01-10', '2025-01-12', 160, 4, NULL);

INSERT INTO Musteri (Ad, Soyad, DogumTarihi, Cinsiyet, TCKimlikNo, TelefonNo, Plaka, GirişTarihi, CikisTarihi, OdaID, OdaTipiID, KampanyaID)
VALUES

```
('Büşra', 'Çınar', '1999-03-03', 'Kadın', '10101010101', '05106110112', '34BSR021', '2025-01-17', '2025-01-19', 200, 4, NULL);
```

```
INSERT INTO Musteri (Ad, Soyad, DogumTarihi, Cinsiyet, TCKimlikNo, TelefonNo, Plaka,
GirisTarihi, CikisTarihi, OdaID, OdaTipiID, KampanyaID)
```

```
VALUES
```

```
('Aleyna', 'Yılmaz', '2003-01-05', 'Kadın', '10552634000', '05431307811', '34SAGO005',
'2025-01-17', '2025-01-19', 199, 4, NULL);
```

```
DELETE FROM Musteri
```

```
--
```

```
-----
-----
```

```
--
```

```
--KONROLLER İÇİN ÖRNEKLER--
```

```
DELETE FROM Musteri
```

```
WHERE OdaID = 1;
```

```
DELETE FROM Musteri
```

```
WHERE Plaka = '34SAGO005';
```

```
INSERT INTO MusteriHizmet (MusteriID, HizmetID, HizmetTarihi)
```

```
VALUES (1, 2, GETDATE()); -- MusteriID 1, HizmetID 2 (Havuz), Bugün
```

```
-----
```

```
-----
```

```
SELECT mh.MusteriID, h.HizmetAdi, h.Ucret, mh.HizmetTarihi
```

```
FROM MusteriHizmet mh
```

```
JOIN Hizmet h ON mh.HizmetID = h.HizmetID
```

```
WHERE mh.MusteriID = 1;
```

```
SELECT MusteriID, ToplamUcret, NetUcret
```

```
FROM CheckOut
```

```
WHERE MusteriID = 1;
```

```
-- CheckOut tablosuna KampanyaID sütunu ekleme
```

```
ALTER TABLE CheckOut
```

```
ADD KampanyaID INT,
```

```
CONSTRAINT FK_CheckOut_Kampanya FOREIGN KEY (KampanyaID) REFERENCES
```

```
Kampanya(KampanyaID);
```

-- CheckOut tablosuna KampanyaID sütunu ekleme

ALTER TABLE CheckOut

ADD KampanyaID INT;

GO

-- KampanyaID için foreign key ekleme

ALTER TABLE CheckOut

ADD CONSTRAINT FK_CheckOut_Kampanya FOREIGN KEY (KampanyaID)

REFERENCES Kampanya(KampanyaID);

GO

-- Önce mevcut trigger'ı kaldıralım

DROP TRIGGER IF EXISTS tr_CheckOut_Insert;

GO

-- Mevcut checkout tablosunu yedekleyelim

SELECT * INTO CheckOut_Backup FROM CheckOut;

GO

-- Checkout tablosunu drop edelim

DROP TABLE CheckOut;

GO

-- Checkout tablosunu yeniden oluşturalım

CREATE TABLE CheckOut (

 CheckOutID INT PRIMARY KEY IDENTITY(1,1),

 MusteriID INT NOT NULL,

 OdaID INT NOT NULL,

 GirisTarihi DATE NOT NULL,

 CikisTarihi DATE NOT NULL,

 ToplamUcret DECIMAL(10,2) NOT NULL,

 KullanilanHizmetler NVARCHAR(MAX) NULL,

 NetUcret DECIMAL(10,2) NOT NULL,

 KampanyaID INT NULL,

 OlusturmaTarihi DATETIME DEFAULT GETDATE(),

 FOREIGN KEY (MusteriID) REFERENCES Musteri(MusteriID),

 FOREIGN KEY (OdaID) REFERENCES Oda(OdaID),

 FOREIGN KEY (KampanyaID) REFERENCES Kampanya(KampanyaID)

);

GO

-- Yedeklenen verileri geri yükleyelim

SET IDENTITY_INSERT CheckOut ON;

INSERT INTO CheckOut (CheckOutID, MusteriID, OdaID, GirisTarihi, CikisTarihi,
ToplamUcret, KullanilanHizmetler, NetUcret, OlusturmaTarihi)

SELECT CheckOutID, MusteriID, OdaID, GirisTarihi, CikisTarihi, ToplamUcret,

KullanilanHizmetler, NetUcret, OlusturmaTarihi

FROM CheckOut_Backup;

SET IDENTITY_INSERT CheckOut OFF;

GO

-- Yedek tabloyu silelim

DROP TABLE CheckOut_Backup;

GO

-- Trigger'ı oluşturalım

CREATE TRIGGER tr_CheckOut_Insert

ON CheckOut

AFTER INSERT

AS

BEGIN

SET NOCOUNT ON;

-- Eğer müşteri zaten checkout yapmışsa, yeni kaydı engelle

IF EXISTS (

SELECT 1

FROM CheckOut c

INNER JOIN inserted i ON c.MusteriID = i.MusteriID

WHERE c.CheckOutID < i.CheckOutID

)

BEGIN

RAISERROR ('Bu müşteri için zaten bir checkout kaydı mevcut!', 16, 1);

ROLLBACK TRANSACTION;

RETURN;

END;

-- Değişkenler

DECLARE @MusteriID INT;

DECLARE @OdaID INT;

DECLARE @GirisTarihi DATE;

DECLARE @CikisTarihi DATE;

DECLARE @GunSayisi INT;

DECLARE @OdaFiyati DECIMAL(10,2);

DECLARE @ToplamHizmetUcreti DECIMAL(10,2);

DECLARE @ToplamUcret DECIMAL(10,2);

DECLARE @NetUcret DECIMAL(10,2);

DECLARE @IndirimYuzdesi DECIMAL(5,2);

DECLARE @KullanilanHizmetler NVARCHAR(MAX);

DECLARE @KampanyaID INT;

DECLARE @CheckOutID INT;

-- Inserted'dan değerleri al

SELECT

@MusteriID = i.MusteriID,

@OdaID = i.OdaID,

@GirisTarihi = i.GirisTarihi,

@CikisTarihi = i.CikisTarihi,

@CheckOutID = i.CheckOutID

FROM inserted i;

```
-- Müşterinin kampanya bilgisini al
SELECT @KampanyaID = KampanyaID
FROM Musteri
WHERE MusteriID = @MusteriID;

-- Gün sayısını hesapla
SET @GunSayisi = DATEDIFF(DAY, @GirisTarihi, @CikisTarihi);
IF @GunSayisi = 0 SET @GunSayisi = 1;

-- Oda günlük fiyatını bul
SELECT @OdaFiyati = ot.GunlukFiyat
FROM Oda o
JOIN OdaTipi ot ON o.OdaTipiID = ot.OdaTipiID
WHERE o.OdaID = @OdaID;

-- Kullanılan hizmetlerin toplam ücretini hesapla
SELECT
    @ToplamHizmetUcreti = ISNULL(SUM(h.Ucret), 0),
    @KullanilanHizmetler = STRING_AGG(h.HizmetAdi, ', ')
FROM MusteriHizmet mh
JOIN Hizmet h ON mh.HizmetID = h.HizmetID
WHERE mh.MusteriID = @MusteriID;

-- Toplam ücreti hesapla (Oda ücreti + Hizmet ücretleri)
SET @ToplamUcret = (@OdaFiyati * @GunSayisi) + @ToplamHizmetUcreti;

-- Müşterinin kampanya indirimini kontrol et
SELECT @IndirimYuzdesi = ISNULL(k.IndirimYuzdesi, 0)
FROM Musteri m
LEFT JOIN Kampanya k ON m.KampanyaID = k.KampanyaID
WHERE m.MusteriID = @MusteriID;

-- Net ücreti hesapla (İndirim uygulanmış)
SET @NetUcret = @ToplamUcret - (@ToplamUcret * @IndirimYuzdesi / 100);

-- CheckOut kaydını güncelle
UPDATE CheckOut
SET
    ToplamUcret = @ToplamUcret,
    NetUcret = @NetUcret,
    KullanilanHizmetler = ISNULL(@KullanilanHizmetler, 'Hizmet Kullanılmadı'),
    KampanyaID = @KampanyaID
WHERE CheckOutID = @CheckOutID;

-- Odayı boşalt
UPDATE Oda
SET Durum = 0
WHERE OdaID = @OdaID;
END;
```


GO

-- Mevcut checkout kayıtlarını temizle

DELETE FROM CheckOut;

GO

-- Örnek bir checkout kaydı oluştur

INSERT INTO CheckOut (MusteriID, OdaID, GiriTarihi, CikisTarihi, ToplamUcret, NetUcret)

VALUES (14, 1, '2025-01-01', '2025-01-05', 0, 0);

INSERT INTO CheckOut (MusteriID, OdaID, GiriTarihi, CikisTarihi, ToplamUcret, NetUcret)

VALUES (13, 1, '2025-01-01', '2025-01-05', 0, 0);

INSERT INTO CheckOut (MusteriID, OdaID, GiriTarihi, CikisTarihi, ToplamUcret, NetUcret)

VALUES (15, 1, '2025-01-01', '2025-01-05', 0, 0);

INSERT INTO CheckOut (MusteriID, OdaID, GiriTarihi, CikisTarihi, ToplamUcret, NetUcret)

VALUES (16, 1, '2025-01-01', '2025-01-05', 0, 0);

GO

-- Sonuçları kontrol et

SELECT

c.*,

k.KampanyaAdi,

k.IndirimYuzdesi

FROM CheckOut c

LEFT JOIN Kampanya k ON c.KampanyaID = k.KampanyaID

WHERE c.MusteriID = 14;

GO