

# PANSİYON VERİ TABANI

Pansiyon veri tabanının içerisinde Pansiyon, müşteriler, odalar, personeller ve rezervasyon tabloları bulunmaktadır. Veri tabanının amacı, odaların bilgilerini, odalarda kalan müşterileri, kaç gün kaldıklarını, toplam ödenen tutarı, odaların temizlik personellerini ve maaşlarını görmek.

## 1- TABLOLARI OLUŞTURMAK

Create table komutu kullanarak tablo oluşturdum ve ilişkilendirdim.

- Pansiyon: (Id, Ad, İl, İlçe, Adres, Mail, Telefon)
- Personel: (Id, pansiyonID(foreign key), AdSoyad, Cinsiyet, TC, Telefon, Mail, Pozisyon, Maaş)
- Odalar: (Id, Kat, Fiyat, personelID(foreign key), Açıklama)
- Müşteriler: (Id, AdSoyad, Cinsiyet, TC, Telefon, Mail)
- Rezervasyon: (Id, musterIID(foreign key), odaID(foreign key), GirişTarihi, ÇıkışTarihi, ToplamTutar)

ID'leri identity(1,1) primary key verdim. ( OdaID identity (101,1) primary key)

## 2- PROCEDURE İLE VERİ EKLEME

Insert into komutu ile pansiyon tablosuna veri ekledikten sonra Odalar, Müşteriler ve Personel tablosu için procedure oluşturdum ve exec komutu yardımıyla çağırıp veri ekledim.

➔ create procedure addOda

```
@kat varchar(1), @fiyat int, @durum varchar(20), @personelID varchar(5),  
@aciklama varchar(200) as
```

```
begin
```

```
insert into Odalar(Kat,Fiyat,Durum,personelID,Aciklama)
```

```
values (@kat,@fiyat,@durum,@personelID,@aciklama)
```

```
end
```

➔ exec addOda

```
@kat = ", @fiyat = ", @personelID = ", @aciklama = '.'
```

### 3- TRIGGER

Müşteri ve personel tablosundaki insert, delete ve update işlemlerini takip edebilmek için ayrı ayrı audit\_log ve her bir işlem(insert,delete,update) için ayrı trigger oluşturdum.

```
➔ create table p_AUDIT_LOG ( ID int identity(1,1), personelID int,
Islem_Tipi varchar(10), Islem_Zamani datetime, Detay varchar(100)
constraint [PK_AUDIT_LOG] primary key CLUSTERED ([ID] asc) )

➔ create trigger trgAfterInsertOnPersonel on Personel
after insert as begin declare
@personelID int, @pansiyonID int, @AdSoyad varchar(100),
@Cinsiyet varchar(1), @TC varchar(11), @Telefon varchar(11),
@Pozisyon varchar(20), @Maas int
select @personelID= personelID , @pansiyonID=pansiyonID ,
@AdSoyad = AdSoyad , @Cinsiyet = Cinsiyet , @TC = TC,
@Telefon = Telefon , @Pozisyon= Pozisyon , @Maas = Maas
from inserted
insert into p_AUDIT_LOG(personelID,Islem_Tipi,Islem_Zamani,Detay)
values (@personelID,'INSERT',GETDATE(),'Yeni Personel Eklendi: ' + @AdSoyad )
end
```

### 4- WHILE DÖNGÜSÜ İLE RANDOM VERİ EKLEME

Rezervasyon tablosu için while kullanarak random 100.000 veri girdim.

Transaction kullanarak olası bir yanlışlıkta error mesajı döndürmesini sağladım.

Random tarihler girilirken @BaslangicTarihi ve @BitisTarihi adında iki değişken oluşturup hangi tarihler arasından seçilmesi gerektiğini belirttim.

Odaların fiyatlarını Odalar tablosundan ID'sine göre çekip @Fiyat değişkenine atadım.

@rezerveGun değişkeni oluşturup giriş ve çıkış tarihleri arasındaki gün ile odanın fiyatıyla çarparak sonucu @ToplamTutar değişkenine atadım.

Sonuçları Rezervasyon tablosuna eklettim.

```

begin transaction

declare @counter int = 0    while @counter < 100000

begin

declare @musteriID int = (select top 1 musterilD from Musteriler order by NEWID())

declare @odaID int = (select top 1 odaID from Odalar order by NEWID())

declare @BaslangicTarihi date = '2023-01-01';

declare @BitisTarihi date = '2024-12-31';

declare @ToplamGun int = DATEDIFF(DAY, @BaslangicTarihi, @BitisTarihi);

declare @RastgeleGirisTarihi date = DATEADD(DAY, ABS(CHECKSUM(NEWID())) %
@ToplamGun, @BaslangicTarihi);

declare @RastgeleCikisTarihi date = DATEADD(DAY, ABS(CHECKSUM(NEWID())) % 15,
@RastgeleGirisTarihi);

declare @rezerveGun int = DATEDIFF(DAY, @RastgeleGirisTarihi,
@RastgeleCikisTarihi);

declare @Fiyat decimal(10,2)

declare @ToplamTutar decimal(10,2)

set @Fiyat = (select Fiyat from Odalar where odaID = @odaID);

set @ToplamTutar = @rezerveGun * @Fiyat;

select @RastgeleGirisTarihi as GirisTarihi, @RastgeleCikisTarihi as CikisTarihi;

insert into Rezervasyon(musterilD,odaID,girisTarihi,cikistarihi,toplamTutar)

values(@musterilD, @odaID, @RastgeleGirisTarihi, @RastgeleCikisTarihi,
@ToplamTutar)

set @counter = @counter + 1

end

commit transaction

rollback transaction;

select ERROR_MESSAGE() as ErrorMessage;

```

## 5- VIEW

Güncel olarak pansiyonda kalan müşterileri, hangi odada kaldıklarını, ödenecek ücretleri, hangi tarihler arasında kalacaklarını gösteren view oluşturdum.

➔ create view GuncelMusteriler as select

R.rezervasyonID, M.AdSoyad as Müşteri\_Adı, O.odaID,

R.girisTarihi, R.cikistarihi, R.toplamTutar

from Rezervasyon R

inner join Musteriler M on R.musteriID = M.musteriID

inner join Odalar O on R.odaID = O.odaID

where R.girisTarihi <= GETDATE() and R.cikistarihi >= GETDATE();

➔ select \* from GuncelMusteriler kodunu çalıştırarak aşağıdaki çıktıyı aldım.

	rezervasyonID	Müşteri_Adı	odaID	girisTarihi	cikistarihi	toplamTutar
1	22	Can Sevinç	107	2024-05-16	2024-05-30	13300
2	210	Cengiz Çakal	101	2024-05-20	2024-06-02	9750
3	245	Ahmet Balkara	107	2024-05-20	2024-05-31	10450
4	385	Büşra Çetin	102	2024-05-12	2024-05-25	8450
5	481	Ali Bilir	106	2024-05-14	2024-05-25	9900
6	842	Serdar Balıkcı	107	2024-05-24	2024-06-02	8550
7	882	Hanife Korkmaz	105	2024-05-21	2024-06-03	11050
8	913	Erdoğan Çiçek	108	2024-05-21	2024-05-30	9000
9	944	Raif Rota	108	2024-05-24	2024-06-06	13000
10	959	Doğan Gök	103	2024-05-22	2024-06-03	8400

## SORGULAR

1- Personelin çalıştığı odanın numarası ve kat bilgisini gösteren sorgu.

```
select
    P.adsoyad as Personel_Adı,
    O.odaID as Oda_No,
    O.kat
from Personel P
inner join Odalar O on P.personelID = O.personelID
```

	Personel_Adı	Oda_No	kat
1	Elçin Gürel	101	1
2	Mehmet Baytar	102	1
3	Halil Bilge	103	1
4	Selim Can	104	2
5	Derya Yılmaz	105	2
6	Gülâyşe Çetin	106	2
7	Can Değirmenci	107	3
8	Yunus Öncü	108	3
9	Hasan Altay	109	3

- 2- Bugün pansiyonda kalan ve toplam ödenecek tutarı 8.000 üzerinde olan, adı veya soyadında 'R' harfi bulunan müşterilerin tüm bilgisini getirecek sorgu.

```
select M.*, R.toplamTutar as ToplamTutar
from Rezervasyon R
inner join Musteriler M on R.musteriID = M.musteriID
where R.girisTarihi <= GETDATE()
and R.cikistarihi >= GETDATE()
and R.toplamTutar > 8000
and M.AdSoyad like '%R%';
```

	musteriID	AdSoyad	Cinsiyet	TC	Telefon	Mail	ToplamTutar
1	12	Ahmet Balkara	E	57493041201	05397896546	ahmtblkra@info.com	10450
2	27	Büşra Çetin	K	19856321458	05349877765	bsrctn@info.com	8450
3	9	Ali Bilir	E	90123698745	05344546511	aliblr@info.com	9900
4	6	Serdar Balıkçı	E	74445785596	05354684680	srdrblkc@info.com	8550
5	5	Hanife Korkmaz	K	88114477855	05384514878	hnfkrkmz@info.com	11050
6	20	Erdinç Çiçek	E	74102369587	05357896963	erdncck@info.com	9000
7	22	Raif Rota	E	48496415146	05326544412	raifrota@info.com	13000
8	19	Erdem Dal	E	91088012365	05347896556	erdmdl@info.com	12600
9	4	Dilara Ak	K	86511236578	05347237878	dlarak@info.com	9900
10	10	Ayşenur Güler	K	84764655134	05368978511	aysnrglmz@info.com	13650
11	2	Murat Karlıova	E	89464151456	05375695641	mrtkrlva@info.com	10800

- 3- Önümüzdeki 5 gün pansiyonda kalacak müşterilerden TC'si 3 ile başlayıp 6 ile bitenleri,giriş ve çıkış tarihlerini getirecek sorgu.

```
select M.AdSoyad,
M.TC,
R.girisTarihi as GirişTarihi,
R.cikistarihi as ÇıkışTarihi
from Rezervasyon R
inner join Musteriler M on R.musteriID = M.musteriID
where R.girisTarihi between GETDATE() and DATEADD(DAY, 5, GETDATE())
and M.TC like '2%' and M.TC like '%6'
```

	AdSoyad	TC	GirişTarihi	ÇıkışTarihi
1	Doğan Gök	25866521326	2024-05-25	2024-06-02
2	Çetin Kaçkın	21207845986	2024-05-29	2024-06-04