

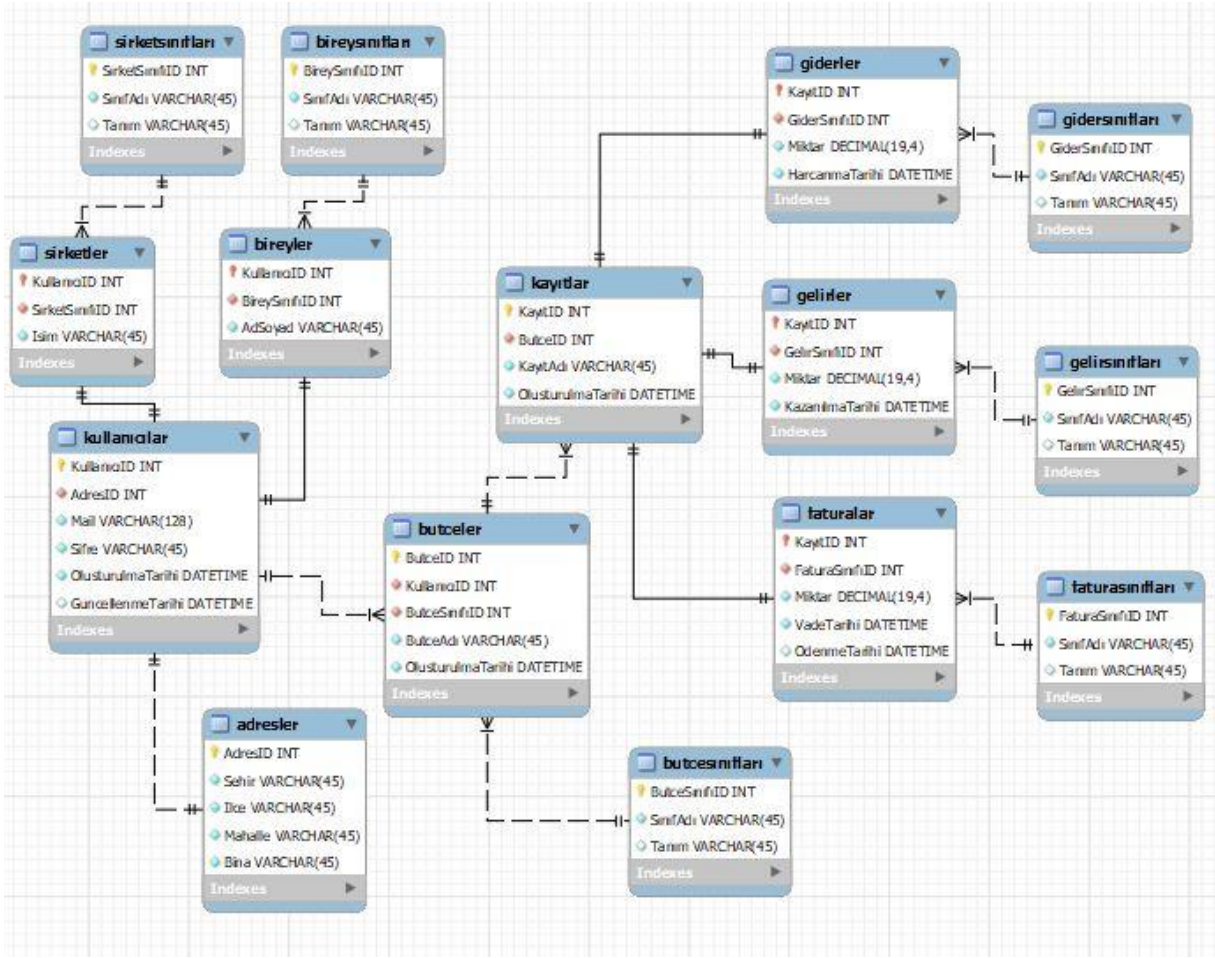


BLM19303 VERİ TABANI YÖNETİM  
SİSTEMLERİ DERSİ  
FİNAL ÖDEVİ

BÜTÇE TAKİBİ

Muhammed Yahya Avar  
1921221004

## Bütçe Takibi Veri Tabanı



**kullanıcılar:** Uygulamayı kullanacak olan kullanıcıların temel bilgileri bulunan tablodur. Her kullanıcının bütçeleri ve bütçelerinin kayıtları (gider-gelir-fatura) vardır. Uygulamanın çalışması için kullanıcının adresini vermesi zorunlu değildir. İki tip kullanıcı için alt tablo vardır.

**şirketler:** Şirket tipi kullanıcıların ek bilgilerini tutar.

**sirketsınıfları:** Şirketlerin kategorik bilgisini tutar. Örn: Anonim Şirketi, Start-Up vs.

**bireyler:** Birey tipi kullanıcıların ek bilgilerini tutar.

**bireysınıfları:** Bireylerin kategorik bilgisini tutar. Örn: Öğrenci, Emekli vs.

**adresler:** Kullanıcıların adres bilgilerini bulundurmaktadır. Bir kullanıcının yalnızca bir adresi olabilir. Bu adres şirket adresi, ev adresi vs. olabilir.

**butceler:** Kullanıcıların kendilerine ayırdığı bütçe bilgilerini tutar. Bir kullanıcının birden çok bütçesi olabilir.

**butcesınıfları:** Butcelerin kategorik bilgisini tutar. Örn: Okul bütçesi, Ticaret bütçesi vs.

**kayıtlar:** Veritabanının omurgası olan tablodur, herhangi bir mali işlemde update edilir. Kayıt tipleri üçe ayrılır.

**giderler:** Kullanıcının elinden çıkan paranın bilgilerini tutan tablodur.

**gidersınıfları:** Giderlerin kategorik bilgisini tutar. Örn. Aylık Gider, Borç Ödemesi vs.

**gelirler:** Kullanıcının eline geçen paranın bilgilerini tutan tablodur.

**gelirsınıfları:** Gelirlerin kategorik bilgisini tutar. Örn. Kira Geliri vs.

**faturalar:** Belirli bir tarihe kadar ödenmesi gereken paranın bilgilerini tutan tablodur.

**faturasınıfları:** Faturaların kategorik bilgisini tutar. Örn. İnternet Faturası, Kredi Kartı Borcu vs.

## Tablo Kuralları

- 1- Id, Gelir Miktarı, Gider Miktarı, Fatura Miktarı gibi nümerik veriler negatif verilemez.
- 2- Bütçe Sınıfı Adı ile Bütçe Adı aynı olamaz, farklı kavramları belirtirler.
- 3- İsim ya da tanım bilgisi olabildiğince kısa tutulmalıdır.
- 4- "kullanıcılar" tablosundaki "Sifre" değişkeni kullanıcının hash edilmiş şifresini gösterir.

## SQL ve İlişkisel Cebir Sorguları

1-2009 yılında oluşturulan ve 5000'in üzerinde gideri olan şirketlerin hangi şehirlerden olduklarını bulunuz.

```
SELECT şehir from adresler adr
WHERE adr.AdresID IN
  (SELECT AdresID from kullanıcılar kul
   WHERE DATEPART(yy, kul.OlustrulmaTarihi) = 2009
   AND kul.kullanıcıID IN
    (SELECT kullanıcıID FROM butceler b
     JOIN kayıtlar k ON b.ButceID=k.ButceID
     WHERE k.KayıtID IN
      (SELECT KayıtID FROM giderler g
       WHERE g.miktar>5000)))
```

bir =  $\pi$  kayıtlar.kayıtid  $\sigma$  giderler.miktar>5000

iki =  $\pi$  kayıtlar. butceid ( kullanıcı  $\bowtie$  kullanıcı.kayıtid= bir.giderler.kayıtid bir)

uc =  $\pi$  butceler.kullanıcıid ( butceler  $\bowtie$  butceler.butceid = iki.kayıtlar.butceid iki)

dort =  $\pi$  adresid (kullanıcılar  $\bowtie$  kullanıcılar.kullanıcıid = uc.butceler.kullanıcıid  $\wedge$  kullanıcılar.olustrulmayili= 2009 uc)

$\pi$  adresler.sehir (adresler.adresid = dort.kullanici.adresid dort)

2- 5'ten fazla bireyin bulunduğu ve her bireyin 2022 vade yılı faturası olan birey sınıflarının adlarını bulunuz.

```
SELECT SınıfAdı FROM bireysınıfları bs
JOIN bireyler br ON bs.BireySınıfID =br.BireySınıfID
GROUP BY BireySınıfID
HAVING COUNT (BireySınıfID>5)
AND br.KullanıcıID IN
  (SELECT KullanıcıID FROM butceler b
   JOIN kayıtlar k ON b.ButceID=k.ButceID
   WHERE k.KayıtID IN
    (SELECT KayıtID FROM faturalar f
     WHERE DATEPART(yy, f.VadeTarihi) = 2022))
```

bir=  $\pi$  kayıtlar.kayıtid  $\sigma$  faturalar.vadetarihi=2022

iki=  $\pi$  kayıtlar. butceid( kullanıcı  $\bowtie$  kullanıcı.kayıtid= bir.giderler.kayıtid bir)

uc =  $\pi$  butceler.kullanıcıid ( butceler  $\bowtie$  butceler.butceid = iki.kayıtlar.butceid iki)

$\pi$  bireysiniflari.sinifadi  $\sigma$  adet>5  $\gamma$  bireyler.bireysinifiid; count(bireyler.biersinifiid)  $\rightarrow$  adet(  
bireysiniflari  $\bowtie$  bireysiniflar.bireysinifiid = bireyler.bireysinifiid uc  $\wedge$  kullanicilar  $\bowtie$   
kullanicilar.kullanıcıid = uc.butceler.kullanıcıid)

3-İstanbul'da yaşayan, "Tasarruf" adlı bütçe sınıfı olan ve bu bütçede herhangi bir gelir kaydı bulunmayan kullanıcıların maillerini bulunuz.

```
SELECT Mail FROM kullanıcılar kul
WHERE kul.AdresID IN
  (SELECT AdresID FROM adresler a
   WHERE a.Sehir='İstanbul')
AND kul.KullanıcıID IN
  (SELECT kullanıcıID FROM butceler b
   WHERE b.ButceSınıfID IN
    (SELECT ButceSınıfID FROM butcesınıfları btcs
     WHERE btcs.SınıfAdı='Tasarruf')
   AND b.ButceID IN
    (SELECT ButceID FROM kayıtlar k
     MINUS
     SELECT ButceID FROM kayıtlar k
     WHERE k.KayıtID IN
      (SELECT KayıtID FROM gelirler g)))
```



Bir1=  $\pi$  gelirler.kayitid

İki1=  $\pi$  kayıtlar.butceid (kayıtlar  $\bowtie$  kayıtlar.kayitid  $\lt \gt$  iki.gelirler.kayitid bir1)

Bir2=  $\pi$  butcesiniflari.butcesinifiid  $\sigma$  butcesiniflari.sinifadi='Tasarruf'

uc =  $\pi$  butceler.kullaniciid ( butceler  $\bowtie$  butceler.butceid = iki.kayıtlar.butceid iki1)  $\wedge$  ( $\pi$  butceler.kullaniciid (butceler  $\bowtie$  butceler. butcesinifiid ==iki. Butcesiniflari. butcesinifiid bir2)

dort=  $\pi$  adresler. adresid( kullanicilar  $\bowtie$  kullanicilar.kullaniciid= uc.butceler.kullaniciid  $\wedge$  adresler.sehir='İstanbul' uc )

$\pi$  kullanicilar.mail (kullanicilar  $\bowtie$  kullanicilar.adresid= dort.adresler.adresid dort)

## PLSQL Tablo Trigger

Gelir tablosundan veri silinme işlemi yapıldığında gider olarak yansıtan trigger

```
CREATE OR REPLACE TRIGGER GELIR_TO_GIDER_REVERSE_UPDATE
BEFORE DELETE ON gelirler
FOR EACH ROW

BEGIN

INSERT INTO giderler
(KayıtID, GiderSınıfıID, Miktar, HarcanmaTarihi)
VALUES
(:new.KayıtID, :new.GiderSınıfıID, :new.Miktar, :new.HarcanmaTarihi)
END;

UPDATE gelirler
```

## PLSQL Fonksiyonu

Kullanıcının sahip olduğu bütçelerin sınıf adlarını gösteren fonksiyon

```
CREATE OR REPLACE FUNCTION kullanıcının_butcesınıf_adı (e_ID IN NUMBER)
RETURN VARCHAR2(45) AS

CURSOR SınıfAdı (kul_id IN NUMBER) IS
SELECT btcs.SınıfAdı FROM butcesiniflari btcs
WHERE btcs.ButceSınıfıID IN
(SELECT b.ButceSınıfıID FROM butceler b
WHERE b.KullanıcıID =kul_id))

gonderilen_sınıf_adı SınıfAdı%TYPE;

BEGIN

OPEN SınıfAdı(e_ID);
FETCH SınıfAdı; INTO gonderilen_sınıf_adı;
CLOSE SınıfAdı;

END
RETURN (gonderilen_sınıf_adı);
```

## Denormalizasyon

“kullanıcılar”, “butceler” ve “kayıtlar” tabloları birleştirilerek bir kullanıcının mali kayıtlarına daha kısa bir sürede daha iyi bir performansla ulaşması gerçekleştirilebilir. KullanıcıID ve ButceID kullanılarak bu üç tablo birbirine bağlanabilir.

## Normalleştirme Analizi

KayıtID	KayıtSınıfID	SınıfAdı	DosyaYolu	GondericiID	GondericiAdı	AlıcıID	AlıcıAdı	OnaylayanID	OnaylayanAdı
1	1	borç	Borç/	1	Ahmet	2	Mehmet	1,3	Ahmet, Ali
2	1	borç	Borç/	2	Mehmet	1	Ahmet	2,3	Mehmet, Ali
3	2	alacak	Alacak/	3	Ali	4	Veli	3,5	Ali, Zeki
4	3	fatura	Fatura/	1	Ahmet	2	Mehmet	5,3	Zeki, Ali
5	2	alacak	Alacak/	5	Zeki	4	Veli	1,3	Ahmet, Ali

KayıtID, KayıtSınıfID → SınıfAdı

KayıtID, GöndericiID → GondericiAdı

KayıtID, AlıcıID → AlıcıAdı

KayıtID, OnaylayanID → OnaylayanAdı

SınıfAdı → DosyaYolu

{ KayıtID, GondericiID, AlıcıID, OnaylayanID } süper anahtarlardır.

### 1. Normalizasyon

Bir satırın bir özelliğinde iki adet bilgi bulunması 1. Normalizasyon’a aykırıdır. Bu sebeple OnaylayanID sütunundaki değerlerin ayrı satırlarda yer almaları gerekmektedir.

KayıtID	OnaylayanID	OnaylayanAdı
1	1	Ahmet
1	3	Ali
2	2	Mehmet
2	3	Ali
3	3	Ali
3	5	Zeki
4	5	Zeki
4	3	Ali
5	1	Mehmet
5	3	ali

## 2. Normalizasyon

KayıtID, GondericiID, AlıcıID ve OnaylayanID anahtarları prime özelliğe sahiptirler. Diğer özellikler ise prime değildir. Bu prime özelliklerin diğer prime olmayan özelliklere ulaşmamaları gerekmektedir. Bu durum 2. Normalizasyona aykırıdır. Bu sebeple bazı tabloların ayrılması gerekmektedir.

KayıtID → KayıtSınıfıID

KayıtSınıfıID → SınıfAdı

KayıtID → OnaylayanID

KayıtID → GondericiID

KayıtID → AlıcıID

OnaylayanID → OnaylayanAdı

GondericiID → GondericiAdı

AlıcıID → AlıcıAdı

SınıfAdı → DosyaYolu

Yeni fonksiyonel bağımlılıklarımızda süper anahtar KayıtID olur. Aday anahtarın alt kümesinde bir eleman bulunmaması da bu yapının 2. Normalizasyon formuna uygun hale geldiğini gösterir.

## 3. Normalizasyon

3. normalizasyon formunun uygunluğunu kontrol etmek için de prime olmayan özellikler arasında bir fonksiyonel bağımlılık olup olmaması duruma bakılır.

SınıfAdı → DosyaYolu

Bu ilişkiye baktığımızda prime olmayan DosyaYolu özelliğinin yine prime olmayan SınıfAdı özelliğine bağımlı olduğunu görüyoruz. Bu durum 3. Normalizasyona uymamaktadır. Bu durumu çözmek için ekstra tablolar oluşturulabilir.

KayıtID → KayıtSınıfıID

KayıtSınıfıID → SınıfAdı, DosyaYolu

KayıtID	OnaylayanID	OnaylayanAdı
1	1	Ahmet
1	3	Ali
2	2	Mehmet
2	3	Ali
3	3	Ali
3	5	Zeki
4	5	Zeki
4	3	Ali
5	1	Mehmet
5	3	ali

KayıtSınıfıID	SınıfAdı	DosyaYolu
1	Borç	Borç/
2	Alacak	Alacak/
3	Fatura	Fatura

AlıcıID	AlıcıAdı
2	Mehmet
1	Ahmet
4	Veli
2	Mehmet
4	Veli

GondericiID	GondericiAdı
1	Ahmet
2	Mehmet
3	Ali
1	Ahmet
5	Zeki