

Veritabanı Yönetim Sistemleri Proje Raporu

Ad-Soyad : Abdülmuttalib GÜLER

Numara : G181210011

Grup : 1. Öğretim A Grubu

E-posta: abdulmuttalib.guler@ogr.sakarya.edu.tr

Kısa Tanımı

Uygulamam bir banka uygulamasıdır.En baş kısmında eğer bankaya üyeliğiniz varsa giriş yapabilirsiniz. Eğer üye değilseniz kayıt olabilirsiniz. Girişiniz başarılı olursa borçlarınızı görüntüleyebilirsiniz. Para gönderebilirsiniz. Ödemelerde bulunabilirsiniz ve kredi çekebilirsiniz.

İŞ KURALLARI

Bir kullanıcı bir hesap açabilir.

Bir kullanıcının bir kullanıcı tipi olabilir.

Bir ödemenin bir türü olabilir.

Bir hesabın bir türü olur.

Bir hesabın bir ödemesi olabilir.

Bir hesabın birçok dövizli olabilir.

Bir hesabın bir faizi olabilir.

Bir hesabın bir emri olabilir.

Bir faizin bir türü olabilir.

Bir kredinin birçok dövizli olabilir.

Bir kredinin bir faizi olur.

Bir kredinin bir tipi olur.

Bir hesabın birçok borcu olabilir.

İLİŞKİSEL ŞEMA

KULLANICI(Id:int,HesapId:int,KullaniciTipiId:int,KullaniciAdi:char(250),Sifre:int,Email:char(50),TC:int, İsim:char(250),Soyisim:char(250))

KULLANICI TIPI(Id:int,İsim:char(250))

ODEME TURU(Id:int,İsim:char(250))

FAIZ TURU(Id:int,İsim:char(250))

KREDİ TIPI(Id:int,İsim:char(250))

HESAP TURU(Id:int,İsim:char(250))

FAİZ(Id:int,FaizMiktari:int,FaizturuId:int)

ODEME(Id:int,OdemeTuruId:int,BaseId:int,TargetId:int,ParaMiktari:float)

DOVİZ(Id:int,TL:float,Euro:float,Dolar:float,Sterlin:float)

EMİRLER(Id:int,BaseId:int,TargetId:int,DovizId:int,ParaMiktari:float)

KREDİ(Id:int,DovizId:int,FaizId:int,KrediTipiId:int,SonOdemeTarihi:Date,ParaMiktari:float)

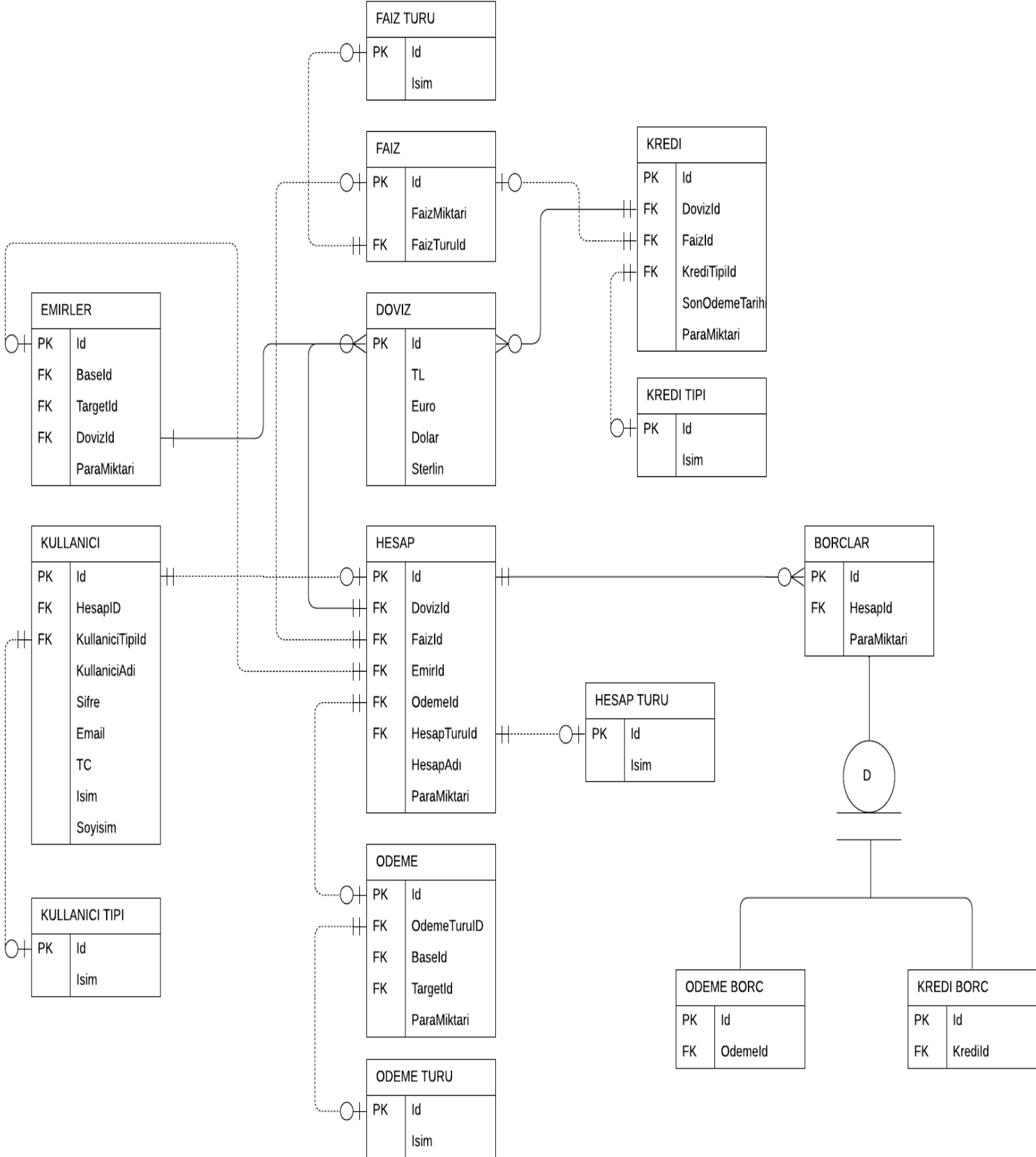
HESAP(Id:int,DovizId:int,FaizId:int,EmirId:int,OdemeId:int,HesapTuruId:int,HesapAdi:char(250), ParaMiktari:float)

BORCLAR(Id:int,ParaMiktari:float,BorcTipi:char(250))

ODEME BORC(Id:int,OdemeId:int)

KREDİ BORC(Id:int,KrediId:int)

Varlık Bağını Modeli



Saklı Yordamlar

1-)

```
CREATE OR REPLACE FUNCTION public.aktar(base INTEGER, target
INTEGER, deger NUMERIC)
  RETURNS void
  LANGUAGE plpgsql
AS $function$
    #variable_conflict use_variable
    DECLARE
    BEGIN
        UPDATE "HESAP" SET "ParaMiktari" = "ParaMiktari"-deger
        WHERE "HESAP"."Id" = base;
        UPDATE "HESAP" SET "ParaMiktari" = "ParaMiktari"+deger
        WHERE "HESAP"."Id" = target;
    END;
$function$
```

2-)

```
CREATE OR REPLACE FUNCTION public."Borclandır"(id INTEGER,
deger NUMERIC, hesapid INTEGER, borctipi CHARACTER)
  RETURNS void
  LANGUAGE plpgsql
AS $function$
    #variable_conflict use_variable
    DECLARE
    BEGIN
        INSERT INTO "public"."BORCLAR" VALUES (id,HesapID,deger,
BorcTipi);
    END;
$function$
```

3-)

```
CREATE OR REPLACE FUNCTION public."FaizHesabi"(id INTEGER,
deger INTEGER, fm NUMERIC)
  RETURNS INTEGER
  LANGUAGE plpgsql
AS $function$
    #variable_conflict use_variable
    DECLARE
    .....
    BEGIN
        RETURN deger*fm;
    .....
    END;
$function$
```

4-)

```
CREATE OR REPLACE FUNCTION public."KrediBorclandir"(id INTEGER
, krediid INTEGER)
  RETURNS void
  LANGUAGE plpgsql
AS $function$
    #variable_conflict use_variable
    DECLARE
    .....
    BEGIN
        INSERT INTO "public"."KREDI BORC" VALUES ('id', 'OdemeID');
    .....
    END;
$function$
```

5-)

```
CREATE OR REPLACE FUNCTION public.ode(id INTEGER, deger
INTEGER)
  RETURNS void
  LANGUAGE plpgsql
AS $function$
    #variable_conflict use_variable
    DECLARE
    BEGIN
        UPDATE "HESAP" SET "ParaMiktari" = "ParaMiktari"-deger
        WHERE "HESAP"."Id" = id;
    END;
$function$
```

6-)

```
CREATE OR REPLACE FUNCTION public."OdemeBorclandir"(id INTEGER
, odemeid INTEGER)
  RETURNS void
  LANGUAGE plpgsql
AS $function$
    #variable_conflict use_variable
    DECLARE
    BEGIN
        INSERT INTO "public"."ODEME BORC" VALUES ('id', 'OdemeID');
    END;
$function$
```

Triggerlar

1-)

```
CREATE OR REPLACE FUNCTION public."KrediCek"()  
  RETURNS TRIGGER  
  LANGUAGE plpgsql  
AS $function$    DECLARE  
BEGIN  
  
  OLD."ParaMiktari" = NEW."ParaMiktari";  
  RETURN NEW;  
END;    $function$
```

```
CREATE TRIGGER arttirma BEFORE UPDATE  
ON public."HESAP" FOR EACH ROW EXECUTE  
  FUNCTION "KrediCek"()
```

2-)

```
CREATE OR REPLACE FUNCTION public."odemeAzaltFunc"()  
  RETURNS TRIGGER  
  LANGUAGE plpgsql  
AS $function$    DECLARE  
BEGIN  
  
  OLD."ParaMiktari" = NEW."ParaMiktari";  
  RETURN NEW;  
END;    $function$
```

```
CREATE TRIGGER azaltma BEFORE UPDATE  
ON public."HESAP" FOR EACH ROW EXECUTE  
  FUNCTION "odemeAzaltFunc"()
```


3-)

```
CREATE OR REPLACE FUNCTION public."idArttirFunc"()  
  RETURNS TRIGGER  
  LANGUAGE plpgsql  
AS $function$    DECLARE  
BEGIN  
NEW."ids" = OLD.ids+1;  
RETURN NEW;  
END;    $function$
```

```
CREATE TRIGGER idarttir BEFORE UPDATE  
ON public.idmaker FOR EACH ROW EXECUTE  
FUNCTION "idArttirFunc"()
```

EKLEME

Database: Banka Schema: public Table: KULLANICI Revert Tables Layout: Default

	Id	HesapId	KullaniciTipId	KullaniciAdi	Sifre
1	100	100	1	akin	12345
2	101	101	1	mertg	mert123
3	102	102	1	muratp	12345
4	103	103	1	jalek	123456
5	104	104	1	selenkk	123123
6	105	105	1	talibg	12345

Register

Ad :

Abdlmuttalib

Kullanıcı Adı :

abdl

Soyad :

Gler

Şifre :

1234

TC :

1111111111

Para Miktarı :

1500

E-Mail :

abdl@gmail.com

KAYIT OL

HesapAdi	HesapTuruId	ParaMiktari
akin Hesabi	1	3500
Mert Hesabi	1	7200
Murat Hesabi	1	6800
Jale Hesabi	1	40000
Selen Hesabi	1	7200
Talib Hesabi	1	30100
Abdlmuttalib Hesabi	1	1000

GÜNCELLEME

HesapAdi	HesapTuruId	ParaMiktari
akin Hesabi	1	3000
Mert Hesabi	1	7200
Murat Hesabi	1	6800
Jale Hesabi	1	40000
Selen Hesabi	1	7200
Talib Hesabi	1	30100
Abdülmuttalib Hesabi	1	1500

Gonderilecek Kişi :

Tutar :

Processing...

HesapAdi	HesapTuruId	ParaMiktari
akin Hesabi	1	3500
Mert Hesabi	1	7200
Murat Hesabi	1	6800
Jale Hesabi	1	40000
Selen Hesabi	1	7200
Talib Hesabi	1	30100
Abdülmuttalib Hesabi	1	1000

-Kaynak kodları

-Video Adresi

https://www.youtube.com/watch?v=GUZC2A_7pks