



KAN BAĞIŞI SİSTEMİ VERİTABANI PROJESİ

DR. ÖĞR. ÜYESİ AYŞE NUR ALTINTAŞ TANKÜL

1. Aml Rizk
2. Yağmur İnci
3. Asmaa Moustafa Ahmed



Senaryo:

Bu proje, bir kan bankası veya sağlık kuruluşu için kapsamlı bir kan bağısı yönetim sistemi oluşturmayı amaçlamaktadır. Sistem, bağışçıların bilgilerini etkin bir şekilde yönetmeyi, kan stoklarını detaylı bir şekilde takip etmeyi, kan bağısı etkinliklerini organize etmeyi, kan ihtiyacı olan hastaların taleplerini karşılamayı ve kan transfüzyonu süreçlerini güvenli bir şekilde izlemeyi hedeflemektedir.

Kan bankası, farklı coğrafi bölgelerde birden fazla Kan Bankası Merkezi'ne sahip olabilir.

Her merkezin kendine özgü bilgileri (adı, adresi, iletişim vb.) bulunur. Kan bankasında çalışan Personel (doktorlar, hemşireler, teknisyenler, idari personel vb.) bu merkezlerde görev yapar. Bir personel aynı anda yalnızca birincil bir merkezde çalışsa da, geçmişte farklı merkezlerde görevlendirilmiş olabilir.

Kan bağışının temelini Bağışçılar oluşturur. Her bağışçının kişisel ve iletişim bilgileri, kan grubu ve sağlık geçmişi detaylı olarak kaydedilir. Bir bağışçı, hayatı boyunca birden fazla kez kan bağışında bulunabilir. Kan bağışları, kan bankası merkezlerinde düzenli olarak yapılabildiği gibi, çeşitli lokasyonlarda organize edilen Kan Bağısı Etkinlikleri aracılığıyla da gerçekleştirilebilir. Bir bağışçı birden fazla etkinliğe katılabilir ve bir etkinlikte birden fazla bağışçı bulunabilir.

Her bir Kan Bağısı İşlemi, bağışçının kimliğini, bağışın yapıldığı tarihi ve saati, alınan kan miktarını ve bağışla ilgili diğer tıbbi verileri içerir. Bir kan bağışından elde edilen kan, farklı Kan Ürünleri'ne (tam kan, eritrosit süspansiyonu, plazma, trombosit vb.) işlenebilir. Her kan ürünü, benzersiz bir barkod numarasıyla tanımlanır ve kan grubu, üretim tarihi, son kullanma tarihi gibi bilgileri içerir. Kan ürünlerinin stok takibi, bulundukları kan bankası merkezine göre yapılır. Bir merkezde farklı kan gruplarından ve farklı türlerde birden fazla kan ürünü bulunabilir.

Bağışlanan her kan birimine, güvenliği sağlamak amacıyla çeşitli Kan Testleri (kan grubu tayini, enfeksiyon taraması vb.) uygulanır. Bir kan ürününe birden fazla test yapılabilir ve her testin sonucu ayrı ayrı kaydedilir.

Kan ihtiyacı olan Hastaneler, sistem üzerinden kan talebinde bulunabilirler. Her hastanenin bilgileri sistemde kayıtlıdır. Bir hastane, birden fazla Kan Talebi oluşturabilir. Her talep, belirli bir Hasta için yapılır ve ihtiyaç duyulan kan ürünü türünü ve kan grubunu belirtir. Bir hasta için birden fazla kan talebi açılabilir (farklı zamanlarda veya farklı ihtiyaçlar için).

Karşılanan her kan talebi için bir veya birden fazla Stok birimi (belirli barkod numarasına sahip kan ürünü) kullanılır.

Gerçekleşen her Kan Transfüzyonu, hangi hastaya, hangi kan ürününün verildiği, transfüzyonun yapıldığı tarih ve saatı ve transfüzyonu gerçekleştiren veya denetleyen personel bilgisini içerir. Bir hasta, tedavi sürecinde birden fazla kan transfüzyonu geçirebilir. Her transfüzyonda belirli bir stoktaki kan ürünü kullanılır.

ENTITIES

Varlık-1	Varlık-2	Varlık-3	Varlık-4	Varlık-5	Varlık-6	Varlık-7	Varlık-8
BAGISCI	KAN BAGISI ETKINLIK	KAN BAGISI	KAN URUNU	KAN TESTI	STOK	KAN TALABI	KAN TRANSFUZYONY
Varlık-9	Varlık-10	Varlık-11	Varlık-12	Varlık-13	Varlık-14	Varlık-15	Varlık-16
HASTA	HASTANE	KAN BANKASI MERKEZİ	PERSONEL	BAGCI KATILIM	DOKTOR	HIMSIRE	TEKNISYEN

ATTRIBUTES

	Entity-1	Entity-2	Entity-3	Entity-4	Entity-5	Entity-6	Entity-7	Entity-8	Entity-9	Entity-10	Entity-11	Entity-12	Entity-13	Entity-14	Entity-15	Entity-16
	BAGISCI	KAN BAGISI ETKINLIGI	KAN BAGISI	KAN URUNU	KAN TESTI	STOK	KAN TALEBİ	HASTA	KAN TRANSFUZYONU	HASTANE	KAN BANKASI MERKEZİ	BAGISCI KATILIM	PERSONEL	DOKTOR	HIMSIRE	TEKNISYEN
Attribute-1	bagisciID	etkinlikID	bagisID	urunID	testID	stokID	talepID	hastaID	transfuzyonID	hastaneID	merkezID	Katilimtarihi	personelID	ID	ID	ID
Attribute-2	TCKimlikNo	merkezID	bagisciID	bagisID	urunID	merkezID	hastaneID	tCKimlikNo	hastaID	hastaneAdi	merkezAdi		merkezID	uzmanlikAlani	SertifikaNo	ruhsatNo
Attribute-3	ad	etkinlikAdi	merkezID	stokID	testAdi	stokTuru	stokID	hastaneID	transfuzyonTarihi	adres	adres		tCKimlikNo		vardiyeTuru	
Attribute-4	soyadi	konum	bagisTarihi	barkodNumarası	testTarihi	miktar	hastaID	ad	transfuzyon Miktarı	telefon	telefon		ad			
Attribute-5	dogumTarihi	baslangicTarihi	tansiyon	urunAdi	sonuc	durumu	talepTarihi	soyadi	Notlar	email	email		soyad			
Attribute-6	cinsiyet	bitisTarihi	nabiz	rafomru	not		gerekliKanGrubu	kanGrubu			acilisTarihi		dogumTarihi			
Attribute-7	kanGrubu	HedefBagisci-Sayisi	notlar	aciklama			gerekliMiktar	telefon					cinsiyet			
Attribute-8	telefon		alinanKanMiktarı	miktar			oncelikDurumu	email					telefon			
Attribute-9	email		hemoglobinDegerleri	saklamaKosullari			talepDurumu	dogumTarihi					email			
Attribute-10	adres			uretimTarihi			gerekliKanTuru	cinsiyet					adres			
Attribute-11	kayitTarihi						AciliyetSeviyesi						unvan			
Attribute-12	saglikDurumu						AmbulansIhtiyac						yoneticid			
Attribute-13							OperasyonTarihi						doktorID			
Attribute-14													himsireID			
Attribute-15													teknisyen			

MATRIX DIAGRAM

Varlıklar Arasındaki İlişkiler

1.Kan Bankası Merkezi ↔ Personel:

Bir-Çok (1:M): Bir Kan Bankası Merkezi'nde birden fazla personel çalışabilir.

2.Bağışçı ↔ Kan Bağışı:

Bir-Çok (1:M): Bir bağışçı hayatı boyunca birden fazla kan bağışında bulunabilir.

3.Bağışçı ↔ Kan Bağışı Etkinliği:

Çok-Çok (M:M): Bir bağışçı birden fazla kan bağısı etkinliğine katılabilir ve bir kan bağısı etkinliğine birden fazla bağışçı katılabilir.

4.Kan Bağışı ↔ Kan Ürünü:

Bir-Çok (1:M): Bir kan bağıından birden fazla farklı kan ürünü (tam kan, plazma vb.) elde edilebilir.

5.Kan Bankası Merkezi ↔ Stok :

Bir-Çok (1:M): Bir Kan Bankası Merkezi'nde farklı kan gruplarından ve farklı türlerde birden fazla stoktan ürün ünabılır.

6.Kan Ürünü ↔ Stok:

Bir-Cok (1:M): Her bir kan ürünü farklı stoklarda bulanabilir.

7.Kan Ürünü ↔ Kan Testi:

Bir-Çok (1:M): Bir kan ürününe birden fazla farklı kan testi uygulanabilir.

8.Hastane ↔ Kan Talebi:

Bir-Çok (1:M): Bir hastane birden fazla kan talebinde bulunabilir.

9.Kan Talebi ↔ Stok (Kan Ürünü):

Bir-Çok (1:M): Bir Stoktan birden fazla kan talebi yabilabilir.

10.Hasta ↔ Kan Transfüzyonu:

Bir-Çok (1:M): Bir hasta tedavi sürecinde birden fazla kan transfüzyonu geçirebilir.

11.Kan Bağışı ↔ Kan Bankası Merkezi:

Bir-Çok (1:M): Bir kan bağısı, doğrudan bir kan bankası merkezinde gerçekleştirilir.

12.Hasta ↔ Hastane:

Bir-Çok (1:M): Bir hastane ise zaman içinde birden fazla hastaya hizmet verebilir.

13.Hasta↔KanTalebi:

Bir-Çok (1:M): Bir hasta birden fazla kan talebinde bulunabilir.

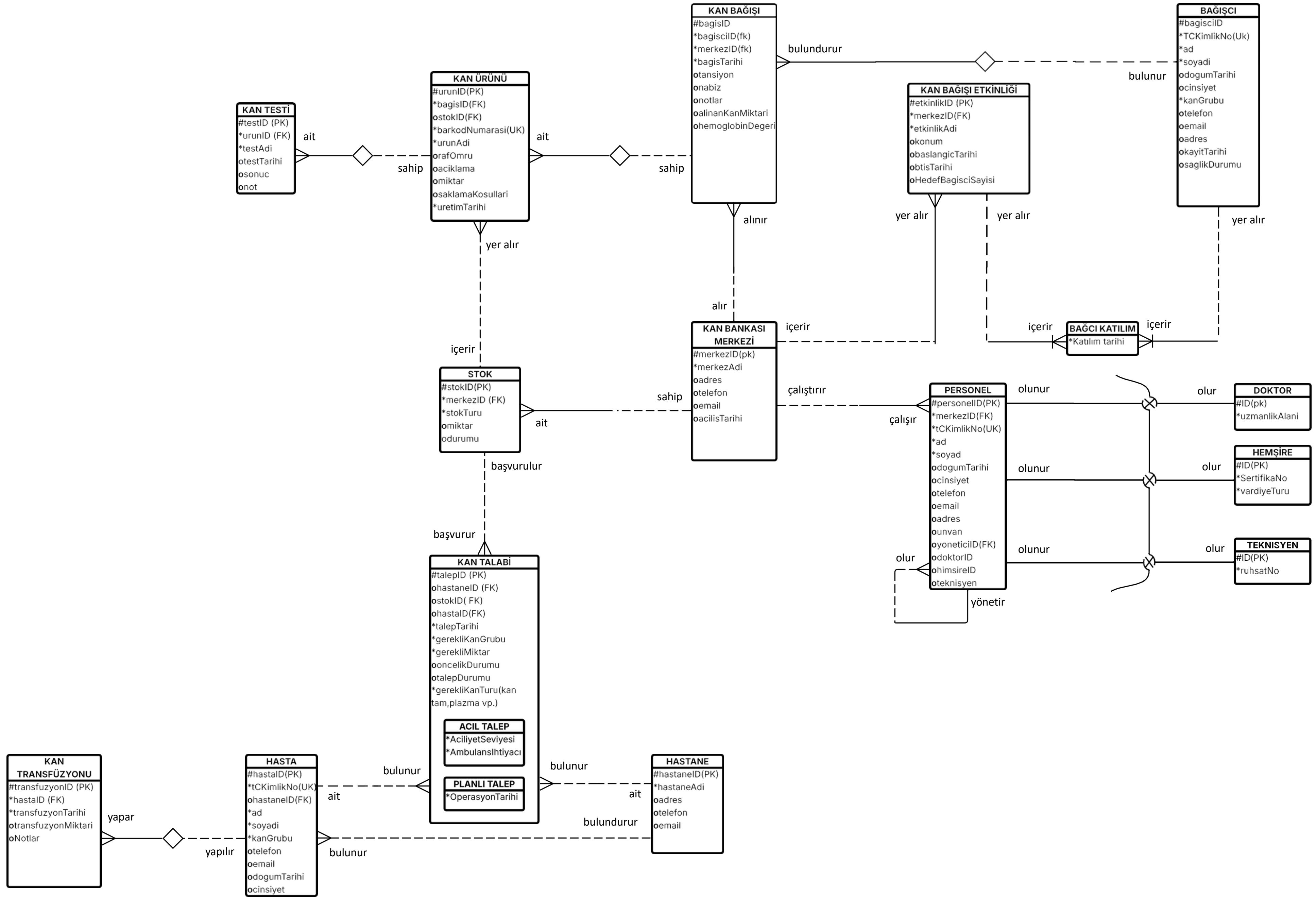


TABLE INSTANCE CHART

KAN_TESTI		
Anahtar	Secmelik	Sutun Adı
PK	*	testID
FK	*	urunID
	*	testAdi
O		sonuc
O		testTarihi
O		notlar

KAN_URUNU		
Anahtar	Secmelik	Sutun Adı
PF	*	urunID
FK	*	bagisID
FK	O	stokID
UK	*	barkodNo
	*	urunID
O		rafOmru
O		aciklama
O		miktar
O		saklamaKosul
*		uretimTarihi

KAN_BAGISI		
Anahtar	Secmelik	Sutun Adı
PK	*	bagisID
FK	*	bagisciID
FK	*	merkezID
	*	bagisTarihi
O		tansiyon
O		nabiz
O		notlar
O		alinanMiktarı
O		hemogobinDegeri

BAGISCI		
Anahtar	Secmelik	Sutun Adı
PK	*	bagiscilID
UK	*	TCKimlik
	*	ad
	*	soyadi
O		dogumTarihi
O		cinsiyet
O		kanGrubu
O		telefon
O		email
O		adres
O		kayitTarihi
O		saglikDurumu

KAN_BAGISI_ETKINLIGI		
Anahtar	Secmelik	Sutun Adı
PK	*	etkinlikID
FK	*	merkezID
	*	etkinlikAdi
O		konum
O		izlemeTarihi
O		bitisTarihi
O		hedefBagisSayisi

BAGCI_KATILIMI		
Anahtar	Secmelik	Sutun Adı
pk (fk)	*	etkinlikID
pk (fk)	*	bagisciID
	*	katilik_tarihi

KAN_BANKASI_MERKEZI		
Anahtar	Secmelik	Sutun Adı
pk	*	merkezID
	*	merkezAdi
O		adres
O		telefon
O		email
O		acilisTarihi

STOK		
Anahtar	Secmelik	Sutun Adı
PK	*	stokID
FK	*	merkezID
	*	stokTuru
	O	miktar
	O	durumu

KAN_TALABI		
Anahtar	Secmelik	Sutun Adı
PK	*	talapID
FK	O	hastalID
FK	O	stokID
	O	hastaneID
	*	talapTarihi
	*	gereklikGrubu
	O	oncelikDurumu
	O	talapDurumu
	*	gereklikTuru
	*	talapTuru
	*	gerekliMiktar
	O	aciliyetSeviyesi
	O	ambulansHijaci
	O	operasyonTarihi

PERSONAL		
Anahtar	Secmelik	Sutun Adı
PK	*	personalID
FK	*	merkezID
UK	*	TCKimlik
FK2,UK1	O	doktorID
FK3,UK2	O	himsireID
FK4,UK3	O	teknisyenID
	*	soyadi
	*	adi
O		cinsiyet
O		telefon
O		email
O		adres
O		unvan
FK4	O	yoneticid
	O	dogumTarihi

HASTA		
Anahtar	Secmelik	Sutun Adı
PK	*	hastalID
UK	*	TCKimlik
FK	O	hastaneID
	*	sd
	*	soyadi
	*	kanGrubu
O		telefon
O		email
O		dogumTarihi
O		cinsiyet

KAN-TRANSFUZYONU		
Anahtar	Secmelik	Sutun Adı
PK	*	transfuzyonID
FK	*	hastalID
	*	transfuzyonTarihi
O		transfuzyonMiktarı
O		notlar

HASTANE		
Anahtar	Secmelik	Sutun Adı
PK	*	hastaneID
	*	hastaneAdı
O		adres
O		telefon
O		email

DOKTOR		
Anahtar	Secmelik	Sutun Adı
PK	*	doktorID
	*	uzmanliklari

HIMSIRE		
Anahtar	Secmelik	Sutun Adı
PK	*	himsireID
	*	sertifikaNo
	*	vardiyeturu

TEKNISYEN		
Anahtar	Secmelik	Sutun Adı

<tbl_r cells="3" ix="2"

SQL- DDL

-- KAN BANKASI MERKEZI Tablosu

```
CREATE TABLE KAN_BANKASI_MERKEZI (
    MerkezID NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
    MerkezAdi VARCHAR2(100) NOT NULL,
    Adres VARCHAR2(200),
    Telefon VARCHAR2(15),
    Email VARCHAR2(100),
    AcilisTarihi DATE,
    CONSTRAINT PK_KAN_BANKASI_MERKEZI PRIMARY KEY (MerkezID));
```

-- PERSONEL Tablosu

```
CREATE TABLE PERSONEL (
    PersonelID NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
    MerkezID NUMBER,
    TCKimlikNo VARCHAR2(11) UNIQUE,
    Ad VARCHAR2(50) NOT NULL,
    Soyad VARCHAR2(50) NOT NULL,
    Cinsiyet VARCHAR2(10),
    DogumTarihi DATE,
    Unvan VARCHAR2(50),
    Telefon VARCHAR2(15),
    Email VARCHAR2(100),
    PersonelTuru VARCHAR2(20) NOT NULL, -- Doktor, Hemşire, Teknisyen, İdari gibi
    CONSTRAINT PK_PERSONEL PRIMARY KEY (PersonelID),
    CONSTRAINT FK_PERSONEL_MERKEZI FOREIGN KEY (MerkezID) REFERENCES KAN_BANKASI_MERKEZI(MerkezID));
```

-- BAGISCI Tablosu

```
CREATE TABLE BAGISCI (
    BagisciID NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
    TCKimlikNo VARCHAR2(11) UNIQUE,
    Ad VARCHAR2(50) NOT NULL,
    Soyad VARCHAR2(50) NOT NULL,
    DogumTarihi DATE,
    Cinsiyet VARCHAR2(10),
    KanGrubu VARCHAR2(3) NOT NULL,
    Telefon VARCHAR2(15),
    Email VARCHAR2(100),
    Adres VARCHAR2(200),
    KayitTarihi DATE,
    SaglikDurumuNotlari CLOB,
    CONSTRAINT PK_BAGISCI PRIMARY KEY (BagisciID));
```

-- KAN BAGISI ETKINLIGI Tablosu

```
CREATE TABLE KAN_BAGISI_ETKINLIGI (
    EtkinlikID NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
    EtkinlikAdi VARCHAR2(100) NOT NULL,
    Konum VARCHAR2(200),
    BaslangicTarihi DATE,
    BitisTarihi DATE,
    HedefBagisciSayisi NUMBER,
    OrganizeEdenPersonelID NUMBER,
    CONSTRAINT PK_KAN_BAGISI_ETKINLIGI PRIMARY KEY (EtkinlikID),
    CONSTRAINT FK_ETKINLIK_PERSONEL FOREIGN KEY (OrganizeEdenPersonelID) REFERENCES PERSONEL(PersonelID));
```

-- KAN BAGISI Tablosu

```
CREATE TABLE KAN_BAGISI (
    BagisID NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
    BagisciID NUMBER NOT NULL,
    EtkinlikID NUMBER,
    BagisTarihi DATE NOT NULL,
    AlinanKanMiktari NUMBER,
    HemoglobinDegeri NUMBER(4, 2),
    Tansiyon VARCHAR2(10),
    Nabiz NUMBER,
    Notlar CLOB,
    BagisiAlanPersonelID NUMBER,
    CONSTRAINT PK_KAN_BAGISI PRIMARY KEY (BagisID),
    CONSTRAINT FK_BAGISI_BAGISCI FOREIGN KEY (BagisciID) REFERENCES BAGISCI(BagisciID),
    CONSTRAINT FK_BAGISI_ETKINLIK FOREIGN KEY (EtkinlikID) REFERENCES KAN_BAGISI_ETKINLIGI(EtkinlikID),
    CONSTRAINT FK_BAGISI_PERSONEL FOREIGN KEY (BagisiAlanPersonelID) REFERENCES PERSONEL(PersonelID));
```

-- KAN URUNU Tablosu

```
CREATE TABLE KAN_URUNU (
    UrunID NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
    UrunAdi VARCHAR2(50) NOT NULL UNIQUE,
    SaklamaKosullari VARCHAR2(200),
    RafOmru NUMBER,
    Aciklama CLOB,
    CONSTRAINT PK_KAN_URUNU PRIMARY KEY (UrunID));
```

-- STOK Tablosu

```
CREATE TABLE STOK (
    StokID NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
    UrunID NUMBER NOT NULL,
    BagisID NUMBER NOT NULL,
    BarkodNumarası VARCHAR2(50) NOT NULL UNIQUE,
    KanGrubu VARCHAR2(3) NOT NULL,
    UretimTarihi DATE,
    SonKullanmaTarihi DATE NOT NULL,
    Miktar NUMBER,
    MerkezID NUMBER NOT NULL,
    Durumu VARCHAR2(20),
    CONSTRAINT PK_STOK PRIMARY KEY (StokID),
    CONSTRAINT FK_STOK_URUN FOREIGN KEY (UrunID) REFERENCES KAN_URUNU(UrunID),
    CONSTRAINT FK_STOK_BAGISI FOREIGN KEY (BagisID) REFERENCES KAN_BAGISI(BagisID),
    CONSTRAINT FK_STOK_MERKEZI FOREIGN KEY (MerkezID) REFERENCES KAN_BANKASI_MERKEZI(MerkezID));
```

-- KAN TESTİ Tablosu

```
CREATE TABLE KAN_TESTI (
    TestID NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
    StokID NUMBER NOT NULL,
    TestAdı VARCHAR2(50) NOT NULL,
    TestTarihi DATE,
    Sonuc VARCHAR2(20),
    Notlar CLOB,
    TestiYapanPersonelID NUMBER,
    CONSTRAINT PK_KAN_TESTI PRIMARY KEY (TestID),
    CONSTRAINT FK_TESTI_STOK FOREIGN KEY (StokID) REFERENCES STOK(StokID),
    CONSTRAINT FK_TESTI_PERSONEL FOREIGN KEY (TestiYapanPersonelID) REFERENCES PERSONEL(PersonelID));
```

-- HASTANE Tablosu

```
CREATE TABLE HASTANE (
    HastaneID NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
    HastaneAdı VARCHAR2(100) NOT NULL UNIQUE,
    Adres VARCHAR2(200),
    Telefon VARCHAR2(15),
    Email VARCHAR2(100),
    CONSTRAINT PK_HASTANE PRIMARY KEY (HastaneID));
```

-- HASTA Tablosu

```
CREATE TABLE HASTA (
    HastaID NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
    TCKimlikNo VARCHAR2(11) UNIQUE,
    Ad VARCHAR2(50) NOT NULL,
    Soyad VARCHAR2(50) NOT NULL,
    DogumTarihi DATE,
    Cinsiyet VARCHAR2(10),
    KanGrubu VARCHAR2(3) NOT NULL,
    Telefon VARCHAR2(15),
    Email VARCHAR2(100),
    TedaviGorduguHastaneID NUMBER,
    Tani VARCHAR2(200),
    CONSTRAINT PK_HASTA PRIMARY KEY (HastaID),
    CONSTRAINT FK_HASTA_HASTANE FOREIGN KEY (TedaviGorduguHastaneID) REFERENCES HASTANE(HastaneID));
```

-- KAN TALEBI Tablosu

```
CREATE TABLE KAN_TALEBI (
    TalepID NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
    HastaneID NUMBER NOT NULL,
    HastaID NUMBER NOT NULL,
    TalepTarihi DATE NOT NULL,
    GerekliKanGrubu VARCHAR2(3) NOT NULL,
    GerekliUrunID NUMBER NOT NULL,
    GerekliMiktar NUMBER NOT NULL,
    OncelikDurumu VARCHAR2(20),
    TalepDurumu VARCHAR2(20),
    TalebiOluşturanPersonelID NUMBER,
    KarsilanStokID NUMBER, -- Birden fazla stok karşılayabilir, bu durumda ayrı bir tablo gerekebilir
    CONSTRAINT PK_KAN_TALEBI PRIMARY KEY (TalepID),
    CONSTRAINT FK_TALEBI_HASTANE FOREIGN KEY (HastaneID) REFERENCES HASTANE(HastaneID),
    CONSTRAINT FK_TALEBI_HASTA FOREIGN KEY (HastaID) REFERENCES HASTA(HastaID),
    CONSTRAINT FK_TALEBI_URUN FOREIGN KEY (GerekliUrunID) REFERENCES KAN_URUNU(UrunID),
    CONSTRAINT FK_TALEBI_PERSONEL FOREIGN KEY (TalebiOluşturanPersonelID) REFERENCES
    PERSONEL(PersonelID)
    -- CONSTRAINT FK_TALEBI_STOK FOREIGN KEY (KarsilanStokID) REFERENCES STOK(StokID) -- Tek bir stok ile
    ilişkilendirilmiş);
```

-- KAN TRANSFUZYONU Tablosu

```
CREATE TABLE KAN_TRANSFUZYONU (
    TransfuzyonID NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
    HastaID NUMBER NOT NULL,
    StokID NUMBER NOT NULL,
    TransfuzyonTarihi DATE NOT NULL,
    TransfuzyonMiktari NUMBER,
    TransfuzyonuYapanPersonelID NUMBER,
    Notlar CLOB,
    CONSTRAINT PK_KAN_TRANSFUZYONU PRIMARY KEY (TransfuzyonID),
    CONSTRAINT FK_TRANSFUZYON_HASTA FOREIGN KEY (HastaID) REFERENCES HASTA(HastaID),
    CONSTRAINT FK_TRANSFUZYON_STOK FOREIGN KEY (StokID) REFERENCES STOK(StokID),
    CONSTRAINT FK_TRANSFUZYON_PERSONEL FOREIGN KEY (TransfuzyonuYapanPersonelID) REFERENCES
PERSONEL(PersonelID)
);
```

ARA TABLOLAR (Çok-Çok İlişkiler İçin)

-- PERSONEL GEÇMİŞİ

(Personel ve Kan Bankası Merkezi arasındaki çok-çok ilişki)

```
CREATE TABLE KAN_TESTI (
    TestID NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
    StokID NUMBER NOT NULL,
    TestAdi VARCHAR2(50) NOT NULL,
    TestTarihi DATE,
    Sonuc VARCHAR2(20),
    Notlar CLOB,
    TestiYapanPersonelID NUMBER,
    CONSTRAINT PK_KAN_TESTI PRIMARY KEY (TestID),
    CONSTRAINT FK_TESTI_STOK FOREIGN KEY (StokID) REFERENCES STOK(StokID),
    CONSTRAINT FK_TESTI_PERSONEL FOREIGN KEY (TestiYapanPersonelID) REFERENCES PERSONEL(PersonelID)
);
```

-- BAGISCI KATILIM

(Bagisci ve Kan Bagisi Etkinliği arasındaki çok-çok ilişki)

```
CREATE TABLE BAGISCI_KATILIM (
    BagisciID NUMBER NOT NULL,
    EtkinlikID NUMBER NOT NULL,
    KatilimTarihi DATE,
    CONSTRAINT PK_BAGISCI_KATILIM PRIMARY KEY (BagisciID, EtkinlikID),
    CONSTRAINT FK_BK_BAGISCI FOREIGN KEY (BagisciID) REFERENCES BAGISCI(BagisciID),
    CONSTRAINT FK_BK_ETKINLIK FOREIGN KEY (EtkinlikID) REFERENCES KAN_BAGISI_ETKINLIGI(EtkinlikID)
);
```

-- DOKTOR, HEMSIRE, TEKNISYEN Tabloları

```
CREATE TABLE DOKTOR (
    PersonelID NUMBER PRIMARY KEY,
    UzmanlikAlani VARCHAR2(100),
    DiplomaNo VARCHAR2(50),
    CONSTRAINT FK_DOKTOR_PERSONEL FOREIGN KEY (PersonelID) REFERENCES
PERSONEL(PersonelID)
);
```

CREATE TABLE HEMSIRE (

```
    PersonelID NUMBER PRIMARY KEY,
    DiplomaNo VARCHAR2(50),
    Sertifikalar CLOB,
    CONSTRAINT FK_HEMSIRE_PERSONEL FOREIGN KEY (PersonelID) REFERENCES
PERSONEL(PersonelID)
);
```

CREATE TABLE TEKNISYEN (

```
    PersonelID NUMBER PRIMARY KEY,
    UzmanlikAlani VARCHAR2(100),
    Sertifikalar CLOB,
    CONSTRAINT FK_TEKNISYEN_PERSONEL FOREIGN KEY (PersonelID) REFERENCES PERSONEL(PersonelID)
);
```

SQL- DML

1. Alt sorgu içeren bir ifade (subquery):

```
SELECT H.Ad, H.Soyad
FROM HASTA H
WHERE H.HastaID IN (SELECT KT.HastaID
                     FROM KAN_TRANSFUZYONU KT
                     JOIN STOK S ON KT.StokID = S.StokID
                     JOIN KAN_BAGISI KB ON S.BagisID =
KB.BagisID
                     JOIN BAGISCI B ON KB.BagisciID = B.BagisciID
                     WHERE B.KanGrubu = 'A+');
```

The screenshot shows a SQL editor with the following code:

```

1  SELECT H.Ad, H.Soyad
2  FROM HASTA H
3  WHERE H.HastaID IN (SELECT KT.HastaID
4   FROM KAN_TRANSFUZYONU KT
5   JOIN STOK S ON KT.StokID = S.StokID
6   JOIN KAN_BAGISI KB ON S.BagisID = KB.BagisID
7   JOIN BAGISCI B ON KB.BagisciID = B.BagisciID
8   WHERE B.KanGrubu = 'A+' );
9

```

Below the code, there is a results table:

AD	SOYAD
Fatma	Yıldız
Canan	Deniz
Nazlı	Efe

3 rows returned in 0.03 seconds [Download](#)

2. Join içeren bir ifade:

```
SELECT B.Ad AS BagisciAdi, B.Soyad AS BagisciSoyadi,
KBE.EtkinlikAdi
FROM BAGISCI B
JOIN KAN_BAGISI KB ON B.BagisciID = KB.BagisciID
JOIN KAN_BAGISI_ETKINLIGI KBE ON KB.EtkinlikID =
KBE.EtkinlikID;
```

The screenshot shows a SQL editor with the following code:

```

1  SELECT B.Ad AS BagisciAdi, B.Soyad AS BagisciSoyadi, KBE.EtkinlikAdi
2  FROM BAGISCI B
3  JOIN KAN_BAGISI KB ON B.BagisciID = KB.BagisciID
4  JOIN KAN_BAGISI_ETKINLIGI KBE ON KB.EtkinlikID = KBE.EtkinlikID;

```

Below the code, there is a results table:

BAGISCIADI	BAGISCISOYADI	ETKINLIKADI
Zeynep	Kara	Karabuk Üniversitesi SosyalFest Kan Bağışı Etkinliği
Mehmet	Öztürk	Kızılay Kan Bağışı Kampanyası
Selin	Yıldırım	Karabuk Üniversitesi SosyalFest Kan Bağışı Etkinliği
Mehmet	Öztürk	Kızılay Kan Bağışı Kampanyası
Aml	Rizk	Karabuk Üniversitesi SosyalFest Kan Bağışı Etkinliği

5 rows returned in 0.02 seconds [Download](#)

3. Group by içeren bir ifade:

```
SELECT KanGrubu, COUNT(*) AS BagisciSayisi
FROM BAGISCI
GROUP BY KanGrubu
ORDER BY KanGrubu;
```

The screenshot shows a SQL editor with the following code:

```

1  SELECT KanGrubu, COUNT(*) AS BagisciSayisi
2  FROM BAGISCI
3  GROUP BY KanGrubu
4  ORDER BY KanGrubu;

```

Below the code, there is a results table:

KANGRUBU	BAGISCISAYISI
0-	1
A+	2
AB	1
AB+	1
B+	1
B-	1

4. Date işlevi içeren bir ifade:

```
SELECT B.Ad, B.Soyad, KB.BagisTarihi  
FROM BAGISCI B  
JOIN KAN_BAGISI KB ON B.BagisciID = KB.BagisciID  
WHERE KB.BagisTarihi <  
TO_DATE('2025-05-08', 'YYYY-MM-DD');
```

The screenshot shows a SQL query results interface. At the top, the SQL code is displayed:

```
1 SELECT B.Ad, B.Soyad, KB.BagisTarihi  
2 FROM BAGISCI B  
3 JOIN KAN_BAGISI KB ON B.BagisciID = KB.BagisciID  
4 WHERE KB.BagisTarihi < TO_DATE('2025-05-08', 'YYYY-MM-DD');
```

Below the code, there is a navigation bar with tabs: Results, Explain, Describe, Saved SQL, and History. The Results tab is selected. The results table has three columns: AD, SOYAD, and BAGISTARIHI. One row is shown:

AD	SOYAD	BAGISTARIHI
Enes	Mustafa	02-May-2025

At the bottom left, it says "1 rows returned in 0.01 seconds". There is also a "Download" button.

5. Character işlevi içeren bir ifade:

```
SELECT EtkinlikAdi, Konum,  
BaslangicTarihi,  
BitisTarihi, HedefBagisciSayisi  
FROM KAN_BAGISI_ETKINLIGI  
WHERE UPPER (SUBSTR(EtkinlikAdi, 1, 1)) = 'K';
```

The screenshot shows a SQL query results interface. At the top, the SQL code is displayed:

```
1 SELECT EtkinlikAdi, Konum, BaslangicTarihi, BitisTarihi, HedefBagisciSayisi  
2 FROM KAN_BAGISI_ETKINLIGI  
3 WHERE UPPER(SUBSTR(EtkinlikAdi, 1, 1)) = 'K';  
4  
5
```

Below the code, there is a navigation bar with tabs: Results, Explain, Describe, Saved SQL, and History. The Results tab is selected. The results table has five columns: ETKINLIKADI, KONUM, BASLANGICTARIHI, BITISTARIHI, and HEDEFBAGISCISAYISI. Two rows are shown:

ETKINLIKADI	KONUM	BASLANGICTARIHI	BITISTARIHI	HEDEFBAGISCISAYISI
Karabuk Üniversitesi SosyalFest Kan Bağışı Etkinliği	Sosyal Yaşam Merkezi	30-May-2025	01-Jun-2025	300
Kızılay Kan Bağışı Kampanyası	Belediye Meydanı	10-May-2025	10-May-2025	100

6. Update içeren bir ifade:

```
UPDATE STOK
SET Durumu = 'Karantinada'
WHERE StokID IN (SELECT KT.StokID
                  FROM KAN_TESTI KT
                  WHERE KT.TestAdi = 'HIV Testi'
                        AND KT.Sonuc = 'Pozitif');
```

The screenshot shows a MySQL command-line interface. The SQL query is:

```
1 UPDATE STOK
2 SET Durumu = 'Karantinada'
3 WHERE StokID IN (SELECT KT.StokID
4                   FROM KAN_TESTI KT
5                   WHERE KT.TestAdi = 'HIV Testi'
6                         AND KT.Sonuc = 'Pozitif');
```

Below the query, there are tabs for Results, Explain, Describe, Saved SQL, and History. The Results tab is selected. The output is:

```
0 row(s) updated.
```

Execution time:

```
0.00 seconds
```

7.Alter table içeren bir ifade:

```
ALTER TABLE HASTA
ADD (KayitTarihi DATE,
     SonGuncellemeTarihi DATE);
```

The screenshot shows a MySQL command-line interface. The SQL query is:

```
1 ALTER TABLE HASTA
2 ADD (KayitTarihi DATE,
3      SonGuncellemeTarihi DATE);
```

Below the query, there are tabs for Results, Explain, Describe, Saved SQL, and History. The Results tab is selected. The output is:

```
Table altered.
```

Execution time:

```
0.03 seconds
```