



SAKARYA
ÜNİVERSİTESİ

Veri Tabanı Yönetim Sistemleri Projesi
2024/2025 Güz Yarıyılı

Ad: Umut

Soyad: Uslu

Öğrenci Numara: B221210012

Grup: 1.Öğretim C grubu

Github reposu: <https://github.com/umutuslu17/vtys>

Eposta: umut.uslu@ogr.sakarya.edu.tr

- **Uygulama Tanıtımı:** Diş Kliniği Veri Tabanı Yönetim Sistemi yaptım. Projemde odaklandığım nokta personel işlemlerini yapabiliyor olması.Ekleme,silme,arama ve listeleme işlemlerini yaptım ve 5 tane saklıyordam oluşturdum, onların çalışmalarını kanıtlamak amacıyla arayüze saklıyordamların çalışmasını sağlayan araçlar koydum.4 tane trigger oluşturdum ve birçok işlemde düzenleme sağlayarak veri işlemede önemli rol aldılar. Veritabanı işlemlerini Postgresql ile yaptım, C# ve Windows Form kullanarak arayüz tasarladım ve personel işlemlerini C# kullanarak yaptım.

İş Kuralları:

- **Personel ve Roller**
- Personel tablosunda her personel için ad, soyad, telefon, ve meslek bilgileri doldurulmalıdır.
- Uzmanlık_alani, personel türüne (diş hekimi, hemşire, sekreter) göre belirlenir ve ilişkili dishekim, hemsire, sekreter tablolarında yer alır.
- PersonelVardiya tablosunda her personel için vardiya_id ve baslangic_saati atanmalıdır.
- Personel tablosu ebeveyn tablosudur ve disHekimi,sekreter ve hemşirelik olmak üzere üç tane çocuk tablosu vardır.Aralarında bu şekilde kalıtım vardır.
- Bir personelden birden fazla hastayla ilgilenilebilir ve birden fazla randevusu olabilir.
- **Hastalar**
- Hasta tablosunda her hasta için ad, soyad, doğum_tarihi, telefon ve adres bilgileri bulunmalıdır.
- Hasta tablosu ebeveyn tablosudur ve yetiskinHasta ve cocukHasta tabloları ile kalıtım ilişkisi vardır ve bu iki tablo çocuk tablolarıdır.
- Çocuk hastalar için veli_adi_soyadi bilgisi zorunludur.
- Bir hasta hiç veya bir ve birden fazla randevu alabilir ve farklı personellerden tedavi olabilir.
- **Randevular**
- Randevu tablosunda hasta_id, personel_id, oda_id, ve randevu_tarihi bilgileri olmalıdır.

- Bir randevu sadece bir hastaya ait olabilir.
- **Odalar**
 - Oda tablosunda her oda için oda_tipi, kapasite bilgileri bulunmalı ve aynı zaman diliminde bir randevuya atanmalıdır.
 - Bir odada birden fazla randevu olabilir ancak aynı anda sadece bir randevu için kullanılabilir.
 - Bir odada birden fazla ekipman bulunabilir.
- **Tedaviler**
 - Tedavi tablosunda her tedavi için tedavi_tipi ve maliyet belirtilmelidir.
 - TedaviEkipman tablosu ile her tedavi için kullanılan ekipman_id ve adet bilgisi kaydedilmelidir.
 - Tedavi tablosu ebeveyn tablodur ve 4 tane çocuk tablo ile kalıtım ilişkisi vardır.
 - Tedavi tablosundaki her tedavi türü (örneğin, disDolgu, kanalTedavisi) kendi özel tablosunda (kalıtım ilişkisi ile) tutulur.
 - Bir tedavi birden fazla hastaya uygulanabilir
 - Bir tedavide birden fazla ekipman kullanılabilir
- **Ekipman**
 - Ekipman tablosunda her ekipman için ad, marka, model, ve maliyet bilgileri yer almalıdır.
 - Bir ekipman birden fazla tedavide kullanılabilir.
 - Bir ekipman birden fazla odada kullanılabilir.
- **İlişkiler ve Kısıtlar**
 - Tüm tablolar arasında 1:N veya N:N ilişkileri doğrulanmalı; ilişkilerde foreign key kısıtlamaları kullanılmalıdır.
- **Ortak Tablolar**
 - Proje içerisinde birden fazla çok-çok bağıntısı olduğu için toplam 4 tane ara tablo vardır
- Bunlar personelHasta, tedaviEkipman, odaEkipman ve hastaTedavi ara tablolarıdır.

İlişkisel Şema:

personel(personel_id: integer, ad: varchar(50), soyad: varchar(50), telefon: varchar(20), meslek: varchar(50))

dishekimi(personel_id: integer (FK), uzmanlik_alani: varchar(50))

hemsire(personel_id: integer (FK), uzmanlik_alani: varchar(50))

sekreter(personel_id: integer (FK), uzmanlik_alani: varchar(50))

personelVardiya(vardiya_id: integer, personel_id: integer (FK), baslangic_saati: time)

hasta(hasta_id: integer, ad: varchar(50), soyad: varchar(50), dogum_tarihi: date, telefon: varchar(20), adres: varchar(100))

cocukhasta(hasta_id: integer (FK), veli_adi_soyadi: varchar(100))

yetiskinhasta(hasta_id: integer (FK), meslek: varchar(50))

oda(oda_id: integer, kapasite: integer, oda_tipi: varchar(50))

randevu(randevu_id: integer, hasta_id: integer (FK), personel_id: integer (FK), oda_id: integer (FK), randevu_tarihi: date)

tedavi(tedavi_id: integer, tedavi_tipi: varchar(50), maliyet: numeric)

disCekimi(tedavi_id: integer (FK), cekim_turu: varchar(50))

disDolgu(tedavi_id: integer (FK), malzeme_turu: varchar(50))

dentalImplant(tedavi_id: integer (FK), implant_turu: varchar(50))

kanalTedavisi(tedavi_id: integer (FK), disin_konumu: varchar(50), malzeme_turu: varchar(50))

hastaTedavi(hasta_id: integer (FK), tedavi_id: integer (FK), tedavi_baslangic: date)

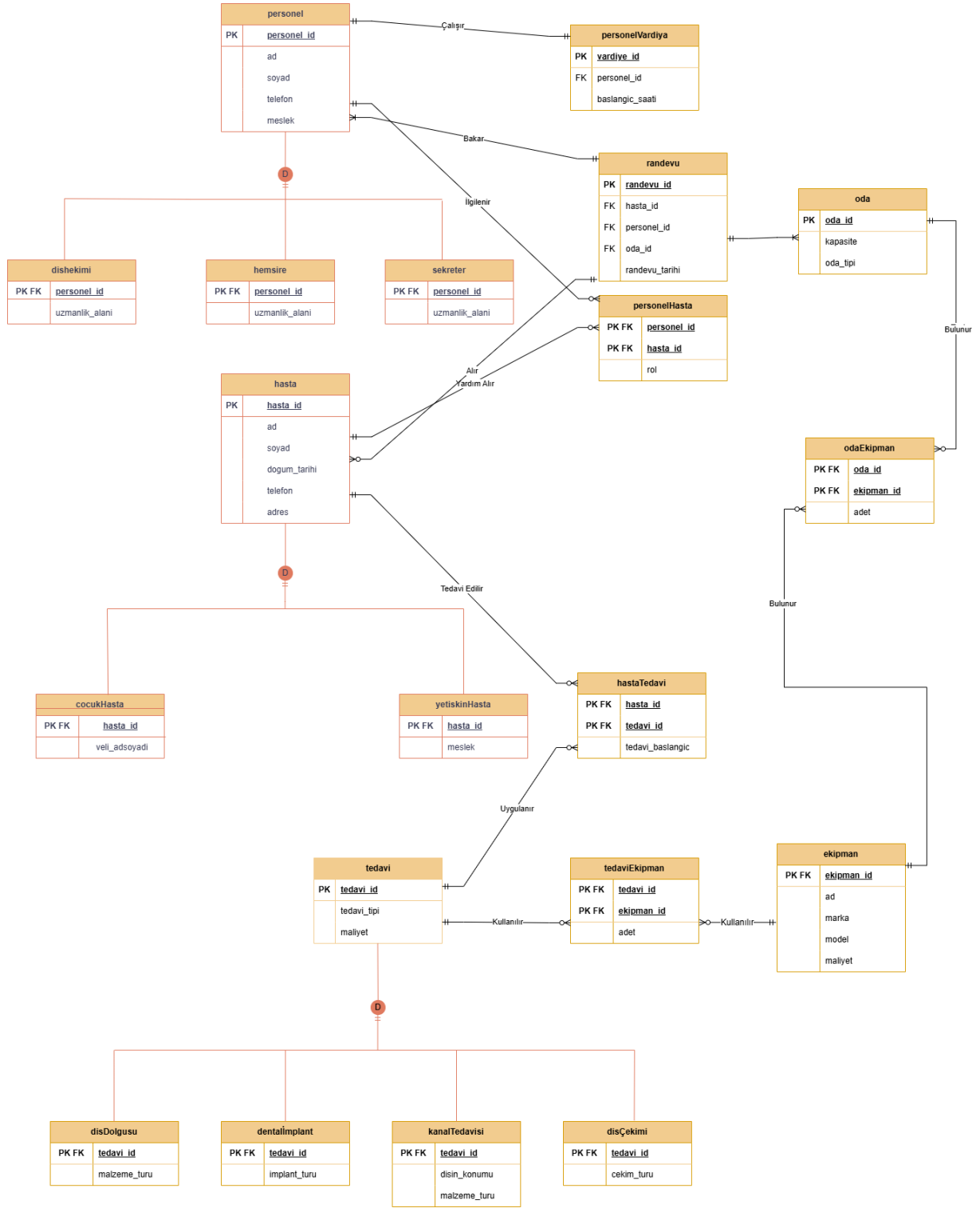
ekipman(ekipman_id: integer, ad: varchar(50), marka: varchar(50), model: varchar(50), maliyet: numeric)

tedaviEkipman(tedavi_id: integer (FK), ekipman_id: integer (FK), adet: integer)

odaEkipman(oda_id: integer (FK), ekipman_id: integer (FK), adet: integer)

personelHasta(personel_id: integer (FK), hasta_id: integer (FK), rol: varchar(50))

VARLIK BAĞINTI MODELİ:



VERİTABANI OLUŞTURMAYI SAĞLAYAN SQL KODLARI

--

-- PostgreSQL database dump

--

-- Dumped from database version 15.10

-- Dumped by pg_dump version 15.10

-- Started on 2024-12-21 01:30:41

SET statement_timeout = 0;

SET lock_timeout = 0;

SET idle_in_transaction_session_timeout = 0;

SET client_encoding = 'UTF8';

SET standard_conforming_strings = on;

SELECT pg_catalog.set_config('search_path', '', false);

SET check_function_bodies = false;

SET xmloption = content;

SET client_min_messages = warning;

SET row_security = off;

--

-- TOC entry 4 (class 2615 OID 2200)

-- Name: public; Type: SCHEMA; Schema: -; Owner: pg_database_owner

--

CREATE SCHEMA public;


```
ALTER SCHEMA public OWNER TO pg_database_owner;
```

```
--
```

```
-- TOC entry 3516 (class 0 OID 0)
```

```
-- Dependencies: 4
```

```
-- Name: SCHEMA public; Type: COMMENT; Schema: -; Owner: pg_database_owner
```

```
--
```

```
COMMENT ON SCHEMA public IS 'standard public schema';
```

```
--
```

```
-- TOC entry 247 (class 1255 OID 17316)
```

```
-- Name: ad_baslayan_personel(character); Type: FUNCTION; Schema: public; Owner: postgres
```

```
--
```

```
CREATE FUNCTION public.ad_baslayan_personel(p_harf character) RETURNS  
TABLE(personel_id integer, ad character varying, soyad character varying)
```

```
LANGUAGE plpgsql
```

```
AS $$
```

```
BEGIN
```

```
RETURN QUERY
```

```
SELECT personel.personel_id, personel.ad, personel.soyad
```

```
FROM personel
```

```
WHERE personel.ad LIKE p_harf || '%';
```

```
END;
```

```
$$;
```

```
ALTER FUNCTION public.ad_baslayan_personel(p_harf character) OWNER TO postgres;
```

```

--

-- TOC entry 258 (class 1255 OID 17318)

-- Name: ayni_ad_soyad_kontrol(); Type: FUNCTION; Schema: public; Owner: postgres

--

CREATE FUNCTION public.ayni_ad_soyad_kontrol() RETURNS trigger

    LANGUAGE plpgsql

    AS $$

BEGIN

    -- Personel tablosunda aynı ad ve soyadı kontrol et

    IF EXISTS (

        SELECT 1

        FROM personel

        WHERE ad = NEW.ad AND soyad = NEW.soyad

    ) THEN

        -- Eğer aynı ad ve soyad varsa, bir hata fırlat

        RAISE EXCEPTION 'Bu isim ve soyada sahip bir personel zaten mevcut: % %', NEW.ad,
NEW.soyad;

    END IF;

    RETURN NEW;

END;

$$;

ALTER FUNCTION public.ayni_ad_soyad_kontrol() OWNER TO postgres;

--

-- TOC entry 244 (class 1255 OID 17313)

-- Name: ayni_meslekte_calisanlar(character varying); Type: FUNCTION; Schema: public;
Owner: postgres

```

--

```
CREATE FUNCTION public.ayni_meslekte_calisanlar(p_meslek character varying) RETURNS
TABLE(personel_id integer, ad character varying, soyad character varying)
```

```
LANGUAGE plpgsql
```

```
AS $$
```

```
BEGIN
```

```
RETURN QUERY
```

```
SELECT
```

```
personel.personel_id,
```

```
personel.ad,
```

```
personel.soyad
```

```
FROM personel
```

```
WHERE meslek = p_meslek;
```

```
END;
```

```
$$;
```

```
ALTER FUNCTION public.ayni_meslekte_calisanlar(p_meslek character varying) OWNER TO
postgres;
```

--

-- TOC entry 260 (class 1255 OID 17321)

-- Name: eksik_bilgi_kontrol(); Type: FUNCTION; Schema: public; Owner: postgres

--

```
CREATE FUNCTION public.eksik_bilgi_kontrol() RETURNS trigger
```

```
LANGUAGE plpgsql
```

```
AS $$
```

```
BEGIN
```

```
-- Eğer herhangi bir alan boş ise hata fırlat
```

```
IF NEW.ad IS NULL OR NEW.ad = '' THEN

    RAISE EXCEPTION 'Ad alanı boş bırakılamaz.';

END IF;


IF NEW.soyad IS NULL OR NEW.soyad = '' THEN

    RAISE EXCEPTION 'Soyad alanı boş bırakılamaz.';

END IF;


IF NEW.telefon IS NULL OR NEW.telefon = '' THEN

    RAISE EXCEPTION 'Telefon alanı boş bırakılamaz.';

END IF;


IF NEW.meslek IS NULL OR NEW.meslek = '' THEN

    RAISE EXCEPTION 'Meslek alanı boş bırakılamaz.';

END IF;

RETURN NEW;

END;

$$;


ALTER FUNCTION public.eksik_bilgi_kontrol() OWNER TO postgres;


--
-- TOC entry 262 (class 1255 OID 17325)
-- Name: maksimum_sekreter_kontrol(); Type: FUNCTION; Schema: public; Owner: postgres
--

CREATE FUNCTION public.maksimum_sekreter_kontrol() RETURNS trigger

    LANGUAGE plpgsql

    AS $$

DECLARE
```

```
sekreter_sayisi INT;

BEGIN

SELECT COUNT(*)

INTO sekreter_sayisi

FROM personel

WHERE meslek = 'Sekreter';

IF sekreter_sayisi >= 5 THEN

    RAISE EXCEPTION 'Maksimum 5 sekreter olabilir. Yeni kayıt eklenemez.';

END IF;

RETURN NEW;

END;

$$;
```

```
ALTER FUNCTION public.maksimum_sekreter_kontrol() OWNER TO postgres;
```

```
--
-- TOC entry 246 (class 1255 OID 17312)
-- Name: meslek_grup_personel_sayisi(); Type: FUNCTION; Schema: public; Owner: postgres
--
```

```
CREATE FUNCTION public.meslek_grup_personel_sayisi() RETURNS TABLE(meslek character
varying, toplam integer)
```

```
LANGUAGE plpgsql

AS $$

BEGIN

RETURN QUERY

SELECT personel.meslek, COUNT(*)::INT

FROM personel

GROUP BY personel.meslek;

END;
```

```
$$;
```

```
ALTER FUNCTION public.meslek_grup_personel_sayisi() OWNER TO postgres;
```

```
--
```

```
-- TOC entry 242 (class 1255 OID 17310)
```

```
-- Name: personel_sayisi(); Type: FUNCTION; Schema: public; Owner: postgres
```

```
--
```

```
CREATE FUNCTION public.personel_sayisi() RETURNS integer
```

```
    LANGUAGE plpgsql
```

```
    AS $$
```

```
DECLARE
```

```
    sayi INT;
```

```
BEGIN
```

```
    SELECT COUNT(*) INTO sayi FROM personel;
```

```
    RETURN sayi;
```

```
END;
```

```
$$;
```

```
ALTER FUNCTION public.personel_sayisi() OWNER TO postgres;
```

```
--
```

```
-- TOC entry 243 (class 1255 OID 17309)
```

```
-- Name: personel_soyada_gore_ara(character varying); Type: FUNCTION; Schema: public;  
Owner: postgres
```

```
--
```

```
CREATE FUNCTION public.personel_soyada_gore_ara(p_soyad character varying) RETURNS  
TABLE(personel_id integer, ad character varying, soyad character varying, telefon character varying)
```

```
LANGUAGE plpgsql

AS $$

BEGIN

    RETURN QUERY

    SELECT personel.personel_id, personel.ad, personel.soyad, personel.telefon

    FROM personel

    WHERE personel.soyad = p_soyad;

END;

$$;
```

```
ALTER FUNCTION public.personel_soyada_gore_ara(p_soyad character varying) OWNER TO
postgres;
```

```
--

-- TOC entry 245 (class 1255 OID 17311)

-- Name: son_eklenen_personel(); Type: FUNCTION; Schema: public; Owner: postgres

--
```

```
CREATE FUNCTION public.son_eklenen_personel() RETURNS TABLE(personel_id integer, ad
character varying, soyad character varying)
```

```
LANGUAGE plpgsql

AS $$

BEGIN

    RETURN QUERY

    SELECT personel.personel_id, personel.ad, personel.soyad

    FROM personel

    ORDER BY personel.personel_id DESC

    LIMIT 1;

END;

$$;
```

```
ALTER FUNCTION public.son_eklenen_personel() OWNER TO postgres;
```

```
--
```

```
-- TOC entry 261 (class 1255 OID 17323)
```

```
-- Name: soyad_buyuk_harf_yap(); Type: FUNCTION; Schema: public; Owner: postgres
```

```
--
```

```
CREATE FUNCTION public.soyad_buyuk_harf_yap() RETURNS trigger
```

```
    LANGUAGE plpgsql
```

```
    AS $$
```

```
BEGIN
```

```
    -- Eğer soyadı NULL değilse büyük harfe çevir
```

```
    IF NEW.soyad IS NOT NULL THEN
```

```
        NEW.soyad = UPPER(NEW.soyad);
```

```
    END IF;
```

```
    RETURN NEW;
```

```
END;
```

```
$$;
```

```
ALTER FUNCTION public.soyad_buyuk_harf_yap() OWNER TO postgres;
```

```
--
```

```
-- TOC entry 241 (class 1255 OID 17327)
```

```
-- Name: telefon_numarasi_benzersizlik_kontrol(); Type: FUNCTION; Schema: public; Owner: postgres
```

```
--
```

```
CREATE FUNCTION public.telefon_numarasi_benzersizlik_kontrol() RETURNS trigger
```

```
    LANGUAGE plpgsql
```



```

AS $$
BEGIN
    -- Yeni eklenen veya güncellenen telefon numarasının varlığını kontrol et
    IF EXISTS (
        SELECT 1
        FROM personel
        WHERE telefon = NEW.telefon AND personel_id <> NEW.personel_id
    ) THEN
        RAISE EXCEPTION 'Bu telefon numarası başka bir personel tarafından kullanılıyor: %',
NEW.telefon;
    END IF;
    RETURN NEW;
END;
$$;

```

```

ALTER FUNCTION public.telefon_numarasi_benzersizlik_kontrol() OWNER TO postgres;

```

```

SET default_tablespace = '';

```

```

SET default_table_access_method = heap;

```

```

--

```

```

-- TOC entry 223 (class 1259 OID 17136)

```

```

-- Name: cocukhasta; Type: TABLE; Schema: public; Owner: postgres

```

```

--

```

```

CREATE TABLE public.cocukhasta (
    hasta_id integer NOT NULL,
    veli_adi_soyadi character varying(100) NOT NULL
);

```

```
ALTER TABLE public.cocukhasta OWNER TO postgres;
```

```
--
```

```
-- TOC entry 233 (class 1259 OID 17213)
```

```
-- Name: dentalimplant; Type: TABLE; Schema: public; Owner: postgres
```

```
--
```

```
CREATE TABLE public.dentalimplant (  
    tedavi_id integer NOT NULL,  
    implant_turu character varying(50) NOT NULL  
);
```

```
ALTER TABLE public.dentalimplant OWNER TO postgres;
```

```
--
```

```
-- TOC entry 231 (class 1259 OID 17193)
```

```
-- Name: discekimi; Type: TABLE; Schema: public; Owner: postgres
```

```
--
```

```
CREATE TABLE public.discekimi (  
    tedavi_id integer NOT NULL,  
    cekim_turu character varying(50) NOT NULL  
);
```

```
ALTER TABLE public.discekimi OWNER TO postgres;
```

```
--
```

```
-- TOC entry 232 (class 1259 OID 17203)
```

```
-- Name: disdolgu; Type: TABLE; Schema: public; Owner: postgres
```

--

```
CREATE TABLE public.disdolgu (  
    tedavi_id integer NOT NULL,  
    malzeme_turu character varying(50) NOT NULL  
);
```

```
ALTER TABLE public.disdolgu OWNER TO postgres;
```

--

```
-- TOC entry 216 (class 1259 OID 17085)  
-- Name: dishekimi; Type: TABLE; Schema: public; Owner: postgres  
--
```

```
CREATE TABLE public.dishekimi (  
    personel_id integer NOT NULL,  
    uzmanlik_alani character varying(50) NOT NULL  
);
```

```
ALTER TABLE public.dishekimi OWNER TO postgres;
```

--

```
-- TOC entry 237 (class 1259 OID 17249)  
-- Name: ekipman; Type: TABLE; Schema: public; Owner: postgres  
--
```

```
CREATE TABLE public.ekipman (  
    ekipman_id integer NOT NULL,  
    ad character varying(50) NOT NULL,  
    marka character varying(50) NOT NULL,
```

```
    model character varying(50),  
    maliyet money NOT NULL  
);
```

```
ALTER TABLE public.ekipman OWNER TO postgres;
```

```
--
```

```
-- TOC entry 236 (class 1259 OID 17248)
```

```
-- Name: ekipman_ekipman_id_seq; Type: SEQUENCE; Schema: public; Owner: postgres
```

```
--
```

```
CREATE SEQUENCE public.ekipman_ekipman_id_seq
```

```
    AS integer
```

```
    START WITH 1
```

```
    INCREMENT BY 1
```

```
    NO MINVALUE
```

```
    NO MAXVALUE
```

```
    CACHE 1;
```

```
ALTER TABLE public.ekipman_ekipman_id_seq OWNER TO postgres;
```

```
--
```

```
-- TOC entry 3517 (class 0 OID 0)
```

```
-- Dependencies: 236
```

```
-- Name: ekipman_ekipman_id_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner:  
postgres
```

```
--
```

```
ALTER SEQUENCE public.ekipman_ekipman_id_seq OWNED BY public.ekipman.ekipman_id;
```

```
--  
  
-- TOC entry 222 (class 1259 OID 17128)  
  
-- Name: hasta; Type: TABLE; Schema: public; Owner: postgres  
  
--
```

```
CREATE TABLE public.hasta (  
    hasta_id integer NOT NULL,  
    ad character varying(50),  
    soyad character varying(50),  
    dogum_tarihi date NOT NULL,  
    telefon character varying(15) NOT NULL,  
    adres text NOT NULL  
);
```

```
ALTER TABLE public.hasta OWNER TO postgres;
```

```
--  
  
-- TOC entry 221 (class 1259 OID 17127)  
  
-- Name: hasta_hasta_id_seq; Type: SEQUENCE; Schema: public; Owner: postgres  
  
--
```

```
CREATE SEQUENCE public.hasta_hasta_id_seq  
  
    AS integer  
  
    START WITH 1  
  
    INCREMENT BY 1  
  
    NO MINVALUE  
  
    NO MAXVALUE  
  
    CACHE 1;
```

```
ALTER TABLE public.hasta_hasta_id_seq OWNER TO postgres;
```

--

-- TOC entry 3518 (class 0 OID 0)

-- Dependencies: 221

-- Name: hasta_hasta_id_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: postgres

--

ALTER SEQUENCE public.hasta_hasta_id_seq OWNED BY public.hasta.hasta_id;

--

-- TOC entry 235 (class 1259 OID 17233)

-- Name: hastatedavi; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public.hastatedavi (

hasta_id integer NOT NULL,

tedavi_id integer NOT NULL,

tedavi_baslangic date NOT NULL

);

ALTER TABLE public.hastatedavi OWNER TO postgres;

--

-- TOC entry 217 (class 1259 OID 17095)

-- Name: hemsire; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public.hemsire (

personel_id integer NOT NULL,

uzmanlik_alani character varying(50) NOT NULL

```
);
```

```
ALTER TABLE public.hemsire OWNER TO postgres;
```

```
--
```

```
-- TOC entry 234 (class 1259 OID 17223)
```

```
-- Name: kanaltedavisi; Type: TABLE; Schema: public; Owner: postgres
```

```
--
```

```
CREATE TABLE public.kanaltedavisi (
```

```
    tedavi_id integer NOT NULL,
```

```
    disin_konumu character varying(50) NOT NULL,
```

```
    malzeme_turu character varying(50) NOT NULL
```

```
);
```

```
ALTER TABLE public.kanaltedavisi OWNER TO postgres;
```

```
--
```

```
-- TOC entry 226 (class 1259 OID 17157)
```

```
-- Name: oda; Type: TABLE; Schema: public; Owner: postgres
```

```
--
```

```
CREATE TABLE public.oda (
```

```
    oda_id integer NOT NULL,
```

```
    kapasite smallint NOT NULL,
```

```
    oda_tipi character varying(50) NOT NULL,
```

```
    CONSTRAINT oda_kapasite_check CHECK ((kapasite > 0))
```

```
);
```

```
ALTER TABLE public.oda OWNER TO postgres;
```

--

-- TOC entry 225 (class 1259 OID 17156)

-- Name: oda_oda_id_seq; Type: SEQUENCE; Schema: public; Owner: postgres

--

CREATE SEQUENCE public.oda_oda_id_seq

AS integer

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE public.oda_oda_id_seq OWNER TO postgres;

--

-- TOC entry 3519 (class 0 OID 0)

-- Dependencies: 225

-- Name: oda_oda_id_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: postgres

--

ALTER SEQUENCE public.oda_oda_id_seq OWNED BY public.oda.oda_id;

--

-- TOC entry 239 (class 1259 OID 17271)

-- Name: odaekipman; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public.odaekipman (


```
oda_id integer NOT NULL,  
ekipman_id integer NOT NULL,  
adet smallint,  
CONSTRAINT odaekipman_adet_check CHECK ((adet > 0))  
);
```

```
ALTER TABLE public.odaekipman OWNER TO postgres;
```

```
--
```

```
-- TOC entry 215 (class 1259 OID 17079)
```

```
-- Name: personel; Type: TABLE; Schema: public; Owner: postgres
```

```
--
```

```
CREATE TABLE public.personel (  
    personel_id integer NOT NULL,  
    ad character varying(50) NOT NULL,  
    soyad character varying(50) NOT NULL,  
    telefon character varying(15) NOT NULL,  
    meslek character varying(50) NOT NULL  
);
```

```
ALTER TABLE public.personel OWNER TO postgres;
```

```
--
```

```
-- TOC entry 214 (class 1259 OID 17078)
```

```
-- Name: personel_personel_id_seq; Type: SEQUENCE; Schema: public; Owner: postgres
```

```
--
```

```
CREATE SEQUENCE public.personel_personel_id_seq  
    AS integer
```

```
START WITH 1  
INCREMENT BY 1  
NO MINVALUE  
NO MAXVALUE  
CACHE 1;
```

```
ALTER TABLE public.personel_personel_id_seq OWNER TO postgres;
```

```
--  
-- TOC entry 3520 (class 0 OID 0)  
-- Dependencies: 214  
-- Name: personel_personel_id_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner:  
postgres
```

```
ALTER SEQUENCE public.personel_personel_id_seq OWNED BY public.personel.personel_id;
```

```
--  
-- TOC entry 240 (class 1259 OID 17287)  
-- Name: personelhasta; Type: TABLE; Schema: public; Owner: postgres  
--
```

```
CREATE TABLE public.personelhasta (  
    personel_id integer NOT NULL,  
    hasta_id integer NOT NULL,  
    rol character varying(50) NOT NULL  
);
```

```
ALTER TABLE public.personelhasta OWNER TO postgres;
```

```
--  
  
-- TOC entry 220 (class 1259 OID 17116)  
  
-- Name: personelvardiya; Type: TABLE; Schema: public; Owner: postgres  
  
--
```

```
CREATE TABLE public.personelvardiya (  
    vardiya_id integer NOT NULL,  
    personel_id integer NOT NULL,  
    baslangic_saati time without time zone NOT NULL  
);
```

```
ALTER TABLE public.personelvardiya OWNER TO postgres;
```

```
--  
  
-- TOC entry 219 (class 1259 OID 17115)  
  
-- Name: personelvardiya_vardiya_id_seq; Type: SEQUENCE; Schema: public; Owner: postgres  
  
--
```

```
CREATE SEQUENCE public.personelvardiya_vardiya_id_seq  
    AS integer  
    START WITH 1  
    INCREMENT BY 1  
    NO MINVALUE  
    NO MAXVALUE  
    CACHE 1;
```

```
ALTER TABLE public.personelvardiya_vardiya_id_seq OWNER TO postgres;
```

```
--  
  
-- TOC entry 3521 (class 0 OID 0)
```

```
-- Dependencies: 219

-- Name: personelvardiya_varদিয়া_id_seq; Type: SEQUENCE OWNED BY; Schema: public;
Owner: postgres
```

```
--
```

```
ALTER SEQUENCE public.personelvardiya_varদিয়া_id_seq OWNED BY
public.personelvardiya.varদিয়া_id;
```

```
--
```

```
-- TOC entry 228 (class 1259 OID 17165)

-- Name: randevu; Type: TABLE; Schema: public; Owner: postgres
```

```
--
```

```
CREATE TABLE public.randevu (
    randevu_id integer NOT NULL,
    hasta_id integer NOT NULL,
    personel_id integer NOT NULL,
    oda_id integer NOT NULL,
    randevu_tarihi date NOT NULL
);
```

```
ALTER TABLE public.randevu OWNER TO postgres;
```

```
--
```

```
-- TOC entry 227 (class 1259 OID 17164)

-- Name: randevu_randevu_id_seq; Type: SEQUENCE; Schema: public; Owner: postgres
```

```
--
```

```
CREATE SEQUENCE public.randevu_randevu_id_seq
    AS integer
```

```
START WITH 1  
INCREMENT BY 1  
NO MINVALUE  
NO MAXVALUE  
CACHE 1;
```

```
ALTER TABLE public.randevu_randevu_id_seq OWNER TO postgres;
```

```
--  
-- TOC entry 3522 (class 0 OID 0)  
-- Dependencies: 227  
-- Name: randevu_randevu_id_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner:  
postgres
```

```
ALTER SEQUENCE public.randevu_randevu_id_seq OWNED BY public.randevu.randevu_id;
```

```
--  
-- TOC entry 218 (class 1259 OID 17105)  
-- Name: sekreter; Type: TABLE; Schema: public; Owner: postgres  
--
```

```
CREATE TABLE public.sekreter (  
    personel_id integer NOT NULL,  
    uzmanlik_alani character varying(50)  
);
```

```
ALTER TABLE public.sekreter OWNER TO postgres;
```

```
--
```

```
-- TOC entry 230 (class 1259 OID 17187)
-- Name: tedavi; Type: TABLE; Schema: public; Owner: postgres
--
```

```
CREATE TABLE public.tedavi (
    tedavi_id integer NOT NULL,
    tedavi_tipi character varying(50) NOT NULL,
    maliyet money NOT NULL
);
```

```
ALTER TABLE public.tedavi OWNER TO postgres;
```

```
--
-- TOC entry 229 (class 1259 OID 17186)
-- Name: tedavi_tedavi_id_seq; Type: SEQUENCE; Schema: public; Owner: postgres
--
```

```
CREATE SEQUENCE public.tedavi_tedavi_id_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;
```

```
ALTER TABLE public.tedavi_tedavi_id_seq OWNER TO postgres;
```

```
--
-- TOC entry 3523 (class 0 OID 0)
-- Dependencies: 229
```

```
-- Name: tedavi_tedavi_id_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: postgres
```

```
--
```

```
ALTER SEQUENCE public.tedavi_tedavi_id_seq OWNED BY public.tedavi.tedavi_id;
```

```
--
```

```
-- TOC entry 238 (class 1259 OID 17255)
```

```
-- Name: tedaviekipman; Type: TABLE; Schema: public; Owner: postgres
```

```
--
```

```
CREATE TABLE public.tedaviekipman (  
    tedavi_id integer NOT NULL,  
    ekipman_id integer NOT NULL,  
    adet smallint,  
    CONSTRAINT tedaviekipman_adet_check CHECK ((adet > 0))  
);
```

```
ALTER TABLE public.tedaviekipman OWNER TO postgres;
```

```
--
```

```
-- TOC entry 224 (class 1259 OID 17146)
```

```
-- Name: yetiskinhasta; Type: TABLE; Schema: public; Owner: postgres
```

```
--
```

```
CREATE TABLE public.yetiskinhasta (  
    hasta_id integer NOT NULL,  
    meslek character varying(50) NOT NULL  
);
```

```
ALTER TABLE public.yetiskinhasta OWNER TO postgres;
```

```
--
```

```
-- TOC entry 3272 (class 2604 OID 17252)
```

```
-- Name: ekipman ekipman_id; Type: DEFAULT; Schema: public; Owner: postgres
```

```
--
```

```
ALTER TABLE ONLY public.ekipman ALTER COLUMN ekipman_id SET DEFAULT  
nextval('public.ekipman_ekipman_id_seq'::regclass);
```

```
--
```

```
-- TOC entry 3268 (class 2604 OID 17131)
```

```
-- Name: hasta hasta_id; Type: DEFAULT; Schema: public; Owner: postgres
```

```
--
```

```
ALTER TABLE ONLY public.hasta ALTER COLUMN hasta_id SET DEFAULT  
nextval('public.hasta_hasta_id_seq'::regclass);
```

```
--
```

```
-- TOC entry 3269 (class 2604 OID 17160)
```

```
-- Name: oda oda_id; Type: DEFAULT; Schema: public; Owner: postgres
```

```
--
```

```
ALTER TABLE ONLY public.oda ALTER COLUMN oda_id SET DEFAULT  
nextval('public.oda_oda_id_seq'::regclass);
```

```
--
```

```
-- TOC entry 3266 (class 2604 OID 17082)
```

```
-- Name: personel personel_id; Type: DEFAULT; Schema: public; Owner: postgres
```

```
--
```



```
ALTER TABLE ONLY public.personel ALTER COLUMN personel_id SET DEFAULT
nextval('public.personel_personel_id_seq'::regclass);
```

--

-- TOC entry 3267 (class 2604 OID 17119)

-- Name: personelvardiya vardiya_id; Type: DEFAULT; Schema: public; Owner: postgres

--

```
ALTER TABLE ONLY public.personelvardiya ALTER COLUMN vardiya_id SET DEFAULT
nextval('public.personelvardiya_vardiya_id_seq'::regclass);
```

--

-- TOC entry 3270 (class 2604 OID 17168)

-- Name: randevu randevu_id; Type: DEFAULT; Schema: public; Owner: postgres

--

```
ALTER TABLE ONLY public.randevu ALTER COLUMN randevu_id SET DEFAULT
nextval('public.randevu_randevu_id_seq'::regclass);
```

--

-- TOC entry 3271 (class 2604 OID 17190)

-- Name: tedavi tedavi_id; Type: DEFAULT; Schema: public; Owner: postgres

--

```
ALTER TABLE ONLY public.tedavi ALTER COLUMN tedavi_id SET DEFAULT
nextval('public.tedavi_tedavi_id_seq'::regclass);
```

--

-- TOC entry 3493 (class 0 OID 17136)

-- Dependencies: 223

-- Data for Name: cocukhasta; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public.cocukhasta (hasta_id, veli_adi_soyadi) FROM stdin;

1 Hasan Çetin

2 Fatma Kara

3 Ali Yılmaz

4 Emine Demir

\\.

--

-- TOC entry 3503 (class 0 OID 17213)

-- Dependencies: 233

-- Data for Name: dentalimplant; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public.dentalimplant (tedavi_id, implant_turu) FROM stdin;

\\.

--

-- TOC entry 3501 (class 0 OID 17193)

-- Dependencies: 231

-- Data for Name: discekimi; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public.discekimi (tedavi_id, cekim_turu) FROM stdin;

\\.

--

-- TOC entry 3502 (class 0 OID 17203)

-- Dependencies: 232

-- Data for Name: disdolgu; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public.disdolgu (tedavi_id, malzeme_turu) FROM stdin;

\.

--

-- TOC entry 3486 (class 0 OID 17085)

-- Dependencies: 216

-- Data for Name: dishekimi; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public.dishekimi (personel_id, uzmanlik_alani) FROM stdin;

\.

--

-- TOC entry 3507 (class 0 OID 17249)

-- Dependencies: 237

-- Data for Name: ekipman; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public.ekipman (ekipman_id, ad, marka, model, maliyet) FROM stdin;

\.

--

-- TOC entry 3492 (class 0 OID 17128)

-- Dependencies: 222

-- Data for Name: hasta; Type: TABLE DATA; Schema: public; Owner: postgres

--

```
COPY public.hasta (hasta_id, ad, soyad, dogum_tarihi, telefon, adres) FROM stdin;
```

```
1 Ali Çetin 2015-03-12 5441237890 Ankara
2 Ayşe Kara 2016-06-22 5336549870 İstanbul
3 Deniz Yılmaz 2014-09-18 5449871230 İzmir
4 Ece Demir 2017-11-02 5334563210 Bursa
5 Murat Şahin 1980-05-15 5551234567 Ankara
6 Selin Yıldız 1990-08-20 5339876543 İstanbul
7 Zeynep Ak 1985-12-01 5446547890 İzmir
8 Ahmet Kurt 1992-03-10 5442139870 Antalya
9 Elif Can 1987-07-25 5336541230 Adana
```

```
\.
```

```
--
```

```
-- TOC entry 3505 (class 0 OID 17233)
```

```
-- Dependencies: 235
```

```
-- Data for Name: hastatedavi; Type: TABLE DATA; Schema: public; Owner: postgres
```

```
--
```

```
COPY public.hastatedavi (hasta_id, tedavi_id, tedavi_baslangic) FROM stdin;
```

```
\.
```

```
--
```

```
-- TOC entry 3487 (class 0 OID 17095)
```

```
-- Dependencies: 217
```

```
-- Data for Name: hemsire; Type: TABLE DATA; Schema: public; Owner: postgres
```

```
--
```

```
COPY public.hemsire (personel_id, uzmanlik_alani) FROM stdin;
```

```
\.
```

--

-- TOC entry 3504 (class 0 OID 17223)

-- Dependencies: 234

-- Data for Name: kanaltedavisi; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public.kanaltedavisi (tedavi_id, disin_konumu, malzeme_turu) FROM stdin;

\\.

--

-- TOC entry 3496 (class 0 OID 17157)

-- Dependencies: 226

-- Data for Name: oda; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public.oda (oda_id, kapasite, oda_tipi) FROM stdin;

\\.

--

-- TOC entry 3509 (class 0 OID 17271)

-- Dependencies: 239

-- Data for Name: odaekipman; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public.odaekipman (oda_id, ekipman_id, adet) FROM stdin;

\\.

--

-- TOC entry 3485 (class 0 OID 17079)

-- Dependencies: 215

-- Data for Name: personel; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public.personel (personel_id, ad, soyad, telefon, meslek) FROM stdin;

1 Ahmet Yılmaz 5551234567 Dış Hekimliği

3 Mehmet Can 5449876543 Dış Hekimliği

4 Elif Demir 5336549872 Dış Hekimliği

5 Fatma Çelik 5442135487 Dış Hekimliği

7 Zeynep Ak 5334567890 Hemşirelik

9 Selin Yıldız 5339876541 Sekreter

2 Ayşe Yılmaz 5352036706 Sekreter

13 Umut Uslu 5423387643 Dış Hekimliği

18 Fahriye MUHTEREM 5837563521 Sekreter

22 Mümin KALEMCI 5425325333 Sekreter

6 Ali MUHTARZADE 5644444444 Hemşire

\\.

--

-- TOC entry 3510 (class 0 OID 17287)

-- Dependencies: 240

-- Data for Name: personelhasta; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public.personelhasta (personel_id, hasta_id, rol) FROM stdin;

\\.

--

-- TOC entry 3490 (class 0 OID 17116)

-- Dependencies: 220

-- Data for Name: personelvardiya; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public.personelvardiya (vardiya_id, personel_id, baslangic_saati) FROM stdin;

\.

--

-- TOC entry 3498 (class 0 OID 17165)

-- Dependencies: 228

-- Data for Name: randevu; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public.randevu (randevu_id, hasta_id, personel_id, oda_id, randevu_tarihi) FROM stdin;

\.

--

-- TOC entry 3488 (class 0 OID 17105)

-- Dependencies: 218

-- Data for Name: sekreter; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public.sekreter (personel_id, uzmanlik_alani) FROM stdin;

9 Ofis Yönetimi

\.

--

-- TOC entry 3500 (class 0 OID 17187)

-- Dependencies: 230

-- Data for Name: tedavi; Type: TABLE DATA; Schema: public; Owner: postgres

--

```
COPY public.tedavi (tedavi_id, tedavi_tipi, maliyet) FROM stdin;
```

```
\.
```

```
--
```

```
-- TOC entry 3508 (class 0 OID 17255)
```

```
-- Dependencies: 238
```

```
-- Data for Name: tedaviekipman; Type: TABLE DATA; Schema: public; Owner: postgres
```

```
--
```

```
COPY public.tedaviekipman (tedavi_id, ekipman_id, adet) FROM stdin;
```

```
\.
```

```
--
```

```
-- TOC entry 3494 (class 0 OID 17146)
```

```
-- Dependencies: 224
```

```
-- Data for Name: yetiskinhasta; Type: TABLE DATA; Schema: public; Owner: postgres
```

```
--
```

```
COPY public.yetiskinhasta (hasta_id, meslek) FROM stdin;
```

```
5 Mühendis
```

```
6 Avukat
```

```
7 Doktor
```

```
8 Öğretmen
```

```
9 Hemşire
```

```
\.
```

```
--
```

```
-- TOC entry 3524 (class 0 OID 0)
```

```
-- Dependencies: 236
```

```
-- Name: ekipman_ekipman_id_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres
```


--

```
SELECT pg_catalog.setval('public.ekipman_ekipman_id_seq', 1, false);
```

--

-- TOC entry 3525 (class 0 OID 0)

-- Dependencies: 221

-- Name: hasta_hasta_id_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

```
SELECT pg_catalog.setval('public.hasta_hasta_id_seq', 9, true);
```

--

-- TOC entry 3526 (class 0 OID 0)

-- Dependencies: 225

-- Name: oda_oda_id_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

```
SELECT pg_catalog.setval('public.oda_oda_id_seq', 1, false);
```

--

-- TOC entry 3527 (class 0 OID 0)

-- Dependencies: 214

-- Name: personel_personel_id_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

```
SELECT pg_catalog.setval('public.personel_personel_id_seq', 24, true);
```

--

-- TOC entry 3528 (class 0 OID 0)

```

-- Dependencies: 219

-- Name: personelvardiya_variya_id_seq; Type: SEQUENCE SET; Schema: public; Owner:
postgres

--

SELECT pg_catalog.setval('public.personelvardiya_variya_id_seq', 1, false);

--

-- TOC entry 3529 (class 0 OID 0)

-- Dependencies: 227

-- Name: randevu_randevu_id_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

SELECT pg_catalog.setval('public.randevu_randevu_id_seq', 1, false);

--

-- TOC entry 3530 (class 0 OID 0)

-- Dependencies: 229

-- Name: tedavi_tedavi_id_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

SELECT pg_catalog.setval('public.tedavi_tedavi_id_seq', 1, false);

--

-- TOC entry 3289 (class 2606 OID 17140)

-- Name: cocukhasta cocukhasta_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.cocukhasta

    ADD CONSTRAINT cocukhasta_pkey PRIMARY KEY (hasta_id);

```

```
--  
  
-- TOC entry 3303 (class 2606 OID 17217)  
  
-- Name: dentalimplant dentalimplant_pkey; Type: CONSTRAINT; Schema: public; Owner:  
postgres
```

```
--  
  
ALTER TABLE ONLY public.dentalimplant  
  
    ADD CONSTRAINT dentalimplant_pkey PRIMARY KEY (tedavi_id);
```

```
--  
  
-- TOC entry 3299 (class 2606 OID 17197)  
  
-- Name: discekimi discekimi_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres  
  
--
```

```
ALTER TABLE ONLY public.discekimi  
  
    ADD CONSTRAINT discekimi_pkey PRIMARY KEY (tedavi_id);
```

```
--  
  
-- TOC entry 3301 (class 2606 OID 17207)  
  
-- Name: disdolgu disdolgu_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres  
  
--
```

```
ALTER TABLE ONLY public.disdolgu  
  
    ADD CONSTRAINT disdolgu_pkey PRIMARY KEY (tedavi_id);
```

```
--  
  
-- TOC entry 3279 (class 2606 OID 17089)  
  
-- Name: dishekimi dishekimi_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres  
  
--
```

```
ALTER TABLE ONLY public.dishekimi
```

```
    ADD CONSTRAINT dishekimi_pkey PRIMARY KEY (personel_id);
```

```
--
```

```
-- TOC entry 3309 (class 2606 OID 17254)
```

```
-- Name: ekipman ekipman_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
```

```
--
```

```
ALTER TABLE ONLY public.ekipman
```

```
    ADD CONSTRAINT ekipman_pkey PRIMARY KEY (ekipman_id);
```

```
--
```

```
-- TOC entry 3287 (class 2606 OID 17135)
```

```
-- Name: hasta hasta_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
```

```
--
```

```
ALTER TABLE ONLY public.hasta
```

```
    ADD CONSTRAINT hasta_pkey PRIMARY KEY (hasta_id);
```

```
--
```

```
-- TOC entry 3307 (class 2606 OID 17237)
```

```
-- Name: hastatedavi hastatedavi_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
```

```
--
```

```
ALTER TABLE ONLY public.hastatedavi
```

```
    ADD CONSTRAINT hastatedavi_pkey PRIMARY KEY (hasta_id, tedavi_id);
```

```
--
```

```
-- TOC entry 3281 (class 2606 OID 17099)
```

```
-- Name: hemsire hemsire_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.hemsire

    ADD CONSTRAINT hemsire_pkey PRIMARY KEY (personel_id);

--

-- TOC entry 3305 (class 2606 OID 17227)

-- Name: kanaltedavisi kanaltedavisi_pkey; Type: CONSTRAINT; Schema: public; Owner:
postgres
--

ALTER TABLE ONLY public.kanaltedavisi

    ADD CONSTRAINT kanaltedavisi_pkey PRIMARY KEY (tedavi_id);

--

-- TOC entry 3293 (class 2606 OID 17163)

-- Name: oda oda_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.oda

    ADD CONSTRAINT oda_pkey PRIMARY KEY (oda_id);

--

-- TOC entry 3313 (class 2606 OID 17276)

-- Name: odaekipman odaekipman_pkey; Type: CONSTRAINT; Schema: public; Owner:
postgres
--

ALTER TABLE ONLY public.odaekipman
```

```
ADD CONSTRAINT odaekipman_pkey PRIMARY KEY (oda_id, ekipman_id);
```

```
--
```

```
-- TOC entry 3277 (class 2606 OID 17084)
```

```
-- Name: personel personel_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
```

```
--
```

```
ALTER TABLE ONLY public.personel
```

```
ADD CONSTRAINT personel_pkey PRIMARY KEY (personel_id);
```

```
--
```

```
-- TOC entry 3315 (class 2606 OID 17291)
```

```
-- Name: personelhasta personelhasta_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
```

```
--
```

```
ALTER TABLE ONLY public.personelhasta
```

```
ADD CONSTRAINT personelhasta_pkey PRIMARY KEY (personel_id, hasta_id);
```

```
--
```

```
-- TOC entry 3285 (class 2606 OID 17121)
```

```
-- Name: personelvardiya personelvardiya_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
```

```
--
```

```
ALTER TABLE ONLY public.personelvardiya
```

```
ADD CONSTRAINT personelvardiya_pkey PRIMARY KEY (vardiya_id);
```

```
--
```

```
-- TOC entry 3295 (class 2606 OID 17170)
```

-- Name: randevu randevu_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.randevu

ADD CONSTRAINT randevu_pkey PRIMARY KEY (randevu_id);

--

-- TOC entry 3283 (class 2606 OID 17109)

-- Name: sekreter sekreter_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.sekreter

ADD CONSTRAINT sekreter_pkey PRIMARY KEY (personel_id);

--

-- TOC entry 3297 (class 2606 OID 17192)

-- Name: tedavi tedavi_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.tedavi

ADD CONSTRAINT tedavi_pkey PRIMARY KEY (tedavi_id);

--

-- TOC entry 3311 (class 2606 OID 17260)

-- Name: tedaviekipman tedaviekipman_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.tedaviekipman

ADD CONSTRAINT tedaviekipman_pkey PRIMARY KEY (tedavi_id, ekipman_id);

```
--  
  
-- TOC entry 3291 (class 2606 OID 17150)  
  
-- Name: yetiskinhasta yetiskinhasta_pkey; Type: CONSTRAINT; Schema: public; Owner:  
postgres
```

```
--
```

```
ALTER TABLE ONLY public.yetiskinhasta  
  
    ADD CONSTRAINT yetiskinhasta_pkey PRIMARY KEY (hasta_id);
```

```
--
```

```
-- TOC entry 3337 (class 2620 OID 17319)  
  
-- Name: personel tetikleyici_ayni_ad_soyad; Type: TRIGGER; Schema: public; Owner:  
postgres
```

```
--
```

```
CREATE TRIGGER tetikleyici_ayni_ad_soyad BEFORE INSERT OR UPDATE ON public.personel  
FOR EACH ROW EXECUTE FUNCTION public.ayni_ad_soyad_kontrol();
```

```
--
```

```
-- TOC entry 3338 (class 2620 OID 17322)  
  
-- Name: personel tetikleyici_eksik_bilgi_kontrol; Type: TRIGGER; Schema: public; Owner:  
postgres
```

```
--
```

```
CREATE TRIGGER tetikleyici_eksik_bilgi_kontrol BEFORE INSERT ON public.personel FOR EACH  
ROW EXECUTE FUNCTION public.eksik_bilgi_kontrol();
```

```
--
```

```
-- TOC entry 3339 (class 2620 OID 17326)  
  
-- Name: personel tetikleyici_maksimum_sekreter; Type: TRIGGER; Schema: public; Owner:  
postgres
```


--

```
CREATE TRIGGER tetikleyici_maksimum_sekreter BEFORE INSERT ON public.personel FOR
EACH ROW WHEN (((new.meslek)::text = 'Sekreter'::text)) EXECUTE FUNCTION
public.maksimum_sekreter_kontrol();
```

--

-- TOC entry 3340 (class 2620 OID 17324)

-- Name: personel_tetikleyici_soyad_buyuk_harf; Type: TRIGGER; Schema: public; Owner:
postgres

--

```
CREATE TRIGGER tetikleyici_soyad_buyuk_harf BEFORE INSERT OR UPDATE ON
public.personel FOR EACH ROW EXECUTE FUNCTION public.soyad_buyuk_harf_yap();
```

--

-- TOC entry 3341 (class 2620 OID 17328)

-- Name: personel_tetikleyici_telefon_benzersizlik; Type: TRIGGER; Schema: public; Owner:
postgres

--

```
CREATE TRIGGER tetikleyici_telefon_benzersizlik BEFORE INSERT OR UPDATE ON
public.personel FOR EACH ROW EXECUTE FUNCTION public.telefon_numarasi_benzersizlik_kontrol();
```

--

-- TOC entry 3320 (class 2606 OID 17141)

-- Name: cocukhasta_cocukhasta_hasta_id_fkey; Type: FK CONSTRAINT; Schema: public;
Owner: postgres

--

```
ALTER TABLE ONLY public.cocukhasta
```

```
ADD CONSTRAINT cocukhasta_hasta_id_fkey FOREIGN KEY (hasta_id) REFERENCES
public.hasta(hasta_id) ON UPDATE CASCADE ON DELETE CASCADE;
```

```
--
```

```
-- TOC entry 3327 (class 2606 OID 17218)
```

```
-- Name: dentalimplant dentalimplant_tedavi_id_fkey; Type: FK CONSTRAINT; Schema: public;
Owner: postgres
```

```
--
```

```
ALTER TABLE ONLY public.dentalimplant
```

```
ADD CONSTRAINT dentalimplant_tedavi_id_fkey FOREIGN KEY (tedavi_id) REFERENCES
public.tedavi(tedavi_id) ON UPDATE CASCADE ON DELETE CASCADE;
```

```
--
```

```
-- TOC entry 3325 (class 2606 OID 17198)
```

```
-- Name: discekimi discekimi_tedavi_id_fkey; Type: FK CONSTRAINT; Schema: public; Owner:
postgres
```

```
--
```

```
ALTER TABLE ONLY public.discekimi
```

```
ADD CONSTRAINT discekimi_tedavi_id_fkey FOREIGN KEY (tedavi_id) REFERENCES
public.tedavi(tedavi_id) ON UPDATE CASCADE ON DELETE CASCADE;
```

```
--
```

```
-- TOC entry 3326 (class 2606 OID 17208)
```

```
-- Name: disdolgu disdolgu_tedavi_id_fkey; Type: FK CONSTRAINT; Schema: public; Owner:
postgres
```

```
--
```

```
ALTER TABLE ONLY public.disdolgu
```

```
ADD CONSTRAINT disdolgu_tedavi_id_fkey FOREIGN KEY (tedavi_id) REFERENCES
public.tedavi(tedavi_id) ON UPDATE CASCADE ON DELETE CASCADE;
```

```
--

-- TOC entry 3316 (class 2606 OID 17090)

-- Name: dishekimi dishekimi_personel_id_fkey; Type: FK CONSTRAINT; Schema: public;
Owner: postgres

--

ALTER TABLE ONLY public.dishekimi

    ADD CONSTRAINT dishekimi_personel_id_fkey FOREIGN KEY (personel_id) REFERENCES
public.personel(personel_id) ON UPDATE CASCADE ON DELETE CASCADE;

--

-- TOC entry 3329 (class 2606 OID 17238)

-- Name: hastatedavi hastatedavi_hasta_id_fkey; Type: FK CONSTRAINT; Schema: public;
Owner: postgres

--

ALTER TABLE ONLY public.hastatedavi

    ADD CONSTRAINT hastatedavi_hasta_id_fkey FOREIGN KEY (hasta_id) REFERENCES
public.hasta(hasta_id) ON UPDATE CASCADE ON DELETE CASCADE;

--

-- TOC entry 3330 (class 2606 OID 17243)

-- Name: hastatedavi hastatedavi_tedavi_id_fkey; Type: FK CONSTRAINT; Schema: public;
Owner: postgres

--

ALTER TABLE ONLY public.hastatedavi

    ADD CONSTRAINT hastatedavi_tedavi_id_fkey FOREIGN KEY (tedavi_id) REFERENCES
public.tedavi(tedavi_id) ON UPDATE CASCADE ON DELETE CASCADE;

--
```

```
-- TOC entry 3317 (class 2606 OID 17100)

-- Name: hemsire hemsire_personel_id_fkey; Type: FK CONSTRAINT; Schema: public; Owner:
postgres
```

```
--
```

```
ALTER TABLE ONLY public.hemsire
```

```
ADD CONSTRAINT hemsire_personel_id_fkey FOREIGN KEY (personel_id) REFERENCES
public.personel(personel_id) ON UPDATE CASCADE ON DELETE CASCADE;
```

```
--
```

```
-- TOC entry 3328 (class 2606 OID 17228)
```

```
-- Name: kanaltedavisi kanaltedavisi_tedavi_id_fkey; Type: FK CONSTRAINT; Schema: public;
Owner: postgres
```

```
--
```

```
ALTER TABLE ONLY public.kanaltedavisi
```

```
ADD CONSTRAINT kanaltedavisi_tedavi_id_fkey FOREIGN KEY (tedavi_id) REFERENCES
public.tedavi(tedavi_id) ON UPDATE CASCADE ON DELETE CASCADE;
```

```
--
```

```
-- TOC entry 3333 (class 2606 OID 17282)
```

```
-- Name: odaekipman odaekipman_ekipman_id_fkey; Type: FK CONSTRAINT; Schema: public;
Owner: postgres
```

```
--
```

```
ALTER TABLE ONLY public.odaekipman
```

```
ADD CONSTRAINT odaekipman_ekipman_id_fkey FOREIGN KEY (ekipman_id) REFERENCES
public.ekipman(ekipman_id);
```

```
--
```

```
-- TOC entry 3334 (class 2606 OID 17277)
```

```
-- Name: odaekipman odaekipman_oda_id_fkey; Type: FK CONSTRAINT; Schema: public;  
Owner: postgres
```

```
--
```

```
ALTER TABLE ONLY public.odaekipman
```

```
ADD CONSTRAINT odaekipman_oda_id_fkey FOREIGN KEY (oda_id) REFERENCES  
public.oda(oda_id);
```

```
--
```

```
-- TOC entry 3335 (class 2606 OID 17297)
```

```
-- Name: personelhasta personelhasta_hasta_id_fkey; Type: FK CONSTRAINT; Schema: public;  
Owner: postgres
```

```
--
```

```
ALTER TABLE ONLY public.personelhasta
```

```
ADD CONSTRAINT personelhasta_hasta_id_fkey FOREIGN KEY (hasta_id) REFERENCES  
public.hasta(hasta_id) ON UPDATE CASCADE ON DELETE CASCADE;
```

```
--
```

```
-- TOC entry 3336 (class 2606 OID 17292)
```

```
-- Name: personelhasta personelhasta_personel_id_fkey; Type: FK CONSTRAINT; Schema:  
public; Owner: postgres
```

```
--
```

```
ALTER TABLE ONLY public.personelhasta
```

```
ADD CONSTRAINT personelhasta_personel_id_fkey FOREIGN KEY (personel_id)  
REFERENCES public.personel(personel_id) ON UPDATE CASCADE ON DELETE CASCADE;
```

```
--
```

```
-- TOC entry 3319 (class 2606 OID 17122)
```

```
-- Name: personelvardiya personelvardiya_personel_id_fkey; Type: FK CONSTRAINT; Schema:  
public; Owner: postgres
```

```

--

ALTER TABLE ONLY public.personelvardiya

    ADD CONSTRAINT personelvardiya_personel_id_fkey FOREIGN KEY (personel_id)
REFERENCES public.personel(personel_id) ON UPDATE CASCADE ON DELETE CASCADE;

--

-- TOC entry 3322 (class 2606 OID 17171)
-- Name: randevu randevu_hasta_id_fkey; Type: FK CONSTRAINT; Schema: public; Owner:
postgres

--

ALTER TABLE ONLY public.randevu

    ADD CONSTRAINT randevu_hasta_id_fkey FOREIGN KEY (hasta_id) REFERENCES
public.hasta(hasta_id);

--

-- TOC entry 3323 (class 2606 OID 17181)
-- Name: randevu randevu_oda_id_fkey; Type: FK CONSTRAINT; Schema: public; Owner:
postgres

--

ALTER TABLE ONLY public.randevu

    ADD CONSTRAINT randevu_oda_id_fkey FOREIGN KEY (oda_id) REFERENCES
public.oda(oda_id);

--

-- TOC entry 3324 (class 2606 OID 17176)
-- Name: randevu randevu_personel_id_fkey; Type: FK CONSTRAINT; Schema: public; Owner:
postgres

--

```

```

ALTER TABLE ONLY public.randevu

    ADD CONSTRAINT randevu_personel_id_fkey FOREIGN KEY (personel_id) REFERENCES
public.personel(personel_id);

--

-- TOC entry 3318 (class 2606 OID 17110)

-- Name: sekreter sekreter_personel_id_fkey; Type: FK CONSTRAINT; Schema: public; Owner:
postgres

--

ALTER TABLE ONLY public.sekreter

    ADD CONSTRAINT sekreter_personel_id_fkey FOREIGN KEY (personel_id) REFERENCES
public.personel(personel_id) ON UPDATE CASCADE ON DELETE CASCADE;

--

-- TOC entry 3331 (class 2606 OID 17266)

-- Name: tedaviekipman tedaviekipman_ekipman_id_fkey; Type: FK CONSTRAINT; Schema:
public; Owner: postgres

--

ALTER TABLE ONLY public.tedaviekipman

    ADD CONSTRAINT tedaviekipman_ekipman_id_fkey FOREIGN KEY (ekipman_id)
REFERENCES public.ekipman(ekipman_id) ON UPDATE CASCADE ON DELETE CASCADE;

--

-- TOC entry 3332 (class 2606 OID 17261)

-- Name: tedaviekipman tedaviekipman_tedavi_id_fkey; Type: FK CONSTRAINT; Schema:
public; Owner: postgres

--

ALTER TABLE ONLY public.tedaviekipman

```

```
        ADD CONSTRAINT tedaviekipman_tedavi_id_fkey FOREIGN KEY (tedavi_id) REFERENCES
public.tedavi(tedavi_id) ON UPDATE CASCADE ON DELETE CASCADE;
```

```
--
```

```
-- TOC entry 3321 (class 2606 OID 17151)
```

```
-- Name: yetiskinhasta yetiskinhasta_hasta_id_fkey; Type: FK CONSTRAINT; Schema: public;
Owner: postgres
```

```
--
```

```
ALTER TABLE ONLY public.yetiskinhasta
```

```
        ADD CONSTRAINT yetiskinhasta_hasta_id_fkey FOREIGN KEY (hasta_id) REFERENCES
public.hasta(hasta_id) ON UPDATE CASCADE ON DELETE CASCADE;
```

```
-- Completed on 2024-12-21 01:30:42
```

```
--
```

```
-- PostgreSQL database dump complete
```

```
--
```


SAKLI YORDAMLAR:

```
-- Soyad üzerinden arama yapan sakli yordam
CREATE OR REPLACE FUNCTION personel_soyada_gore_ara(p_soyad VARCHAR)
RETURNS TABLE(
    personel_id INT,
    ad VARCHAR,
    soyad VARCHAR,
    telefon VARCHAR
)
AS $$
BEGIN
    RETURN QUERY
    SELECT personel.personel_id, personel.ad, personel.soyad, personel.telefon
    FROM personel
    WHERE personel.soyad = p_soyad;
END;
$$ LANGUAGE plpgsql;
```

Fonksiyon adı: personel_soyada_gore_ara ve parametresi: p_soyad.

Kullanıcıdan personelin soyadı alınır ve bu soyada sahip olan personel kayıtları yani id, ad, soyad, telefon sütunları da tablo olarak döndürülür.

Arama butonuna tıkladığımızda o soyada sahip olan çalışan veya çalışanların kayıtları DataGridView nesnesinde görüntülenir.

```
--Son Eklenen Elemanı Tablo Olarak Döndüren Sakli yordam
CREATE OR REPLACE FUNCTION son_eklenen_personel()
RETURNS TABLE(
    personel_id INT,
    ad VARCHAR,
    soyad VARCHAR
)
AS $$
BEGIN
    RETURN QUERY
    SELECT personel.personel_id, personel.ad, personel.soyad
    FROM personel
    ORDER BY personel.personel_id DESC
    LIMIT 1;
END;
$$ LANGUAGE plpgsql;
```

Fonksiyon adı: son_eklenen_personel

Son eklenilen personelin kaydını tablo olarak döner.DataGridView nesnesinde gözüktür.

```

--Personel Sayısını Hesaplayan Saklıyordam
CREATE OR REPLACE FUNCTION personel_sayisi()
RETURNS INT
AS $$
DECLARE
    sayi INT;
BEGIN
    SELECT COUNT(*) INTO sayi FROM personel;
    RETURN sayi;
END;
$$
LANGUAGE plpgsql;

```

Fonksiyon adı:personel_sayisi

Personel tablosunda kayıtlı olan toplam personel sayısını döndürür. Arayüzde butona bastığımda personel sayısı labelda gözükür.

```

--Her Meslek Alanında Çalışan Personel Sayısını Tablo Olarak Döndürür
CREATE OR REPLACE FUNCTION meslek_grup_personel_sayisi()
RETURNS TABLE(
    meslek VARCHAR,
    toplam INT
) AS $$
BEGIN
    RETURN QUERY
    SELECT personel.meslek, COUNT(*)::INT
    FROM personel
    GROUP BY personel.meslek;
END;
$$ LANGUAGE plpgsql;

```

Fonksiyon adı:meslek_grup_personel_sayisi

Personel tablosunda kayıtlı olan her meslek grubunda çalışan personel sayısını tablo olarak döndür.DataGridView nesnesinde görüntülenir.

```
--Girilen Harf İle Aynı Baş Harfe Sahip Olan Personelleri Listeleyen Saklıyordam  
CREATE OR REPLACE FUNCTION ad_baslayan_personel(p_harf CHAR)  
RETURNS TABLE(  
    personel_id INT,  
    ad VARCHAR,  
    soyad VARCHAR  
) AS $$  
BEGIN  
    RETURN QUERY  
    SELECT personel.personel_id, personel.ad, personel.soyad  
    FROM personel  
    WHERE personel.ad LIKE p_harf || '%';  
END;  
$$ LANGUAGE plpgsql;
```

Fonksiyon Adı: ad_baslayan_personel, Parametre: p_harf CHAR

Kullancıdan alınan harf ile başlayan adlara sahip olan personelleri kayıtları tablo şeklinde döndürüp gösteren saklıyordamdır.

Bir tane textbox koydum ve oradan kullanıcıdan harf aldım ve bunu parametre olarak saklıyordamda kullandım ve butona basınca o harfle başlayan personel kayıtlarını tablo olarak döndürür ve datagridview nesnesinde görüntülenir.

Tetikleyiciler

```
-- Trigger: Aynı ad ve soyadın girildiğinde tetiklenir.
CREATE OR REPLACE FUNCTION ayni_ad_soyad_kontrol()
RETURNS TRIGGER AS $$
BEGIN
    -- Personel tablosunda aynı ad ve soyadı kontrol et
    IF EXISTS (
        SELECT 1
        FROM personel
        WHERE ad = NEW.ad AND soyad = NEW.soyad
    ) THEN
        -- Eğer aynı ad ve soyad varsa, bir hata fırlat
        RAISE EXCEPTION 'Bu isim ve soyada sahip bir personel zaten mevcut: % %',
            NEW.ad, NEW.soyad;
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER tetikleyici_ayni_ad_soyad
BEFORE INSERT OR UPDATE ON personel
FOR EACH ROW
EXECUTE FUNCTION ayni_ad_soyad_kontrol();
```

Eğer aynı ada ve soyada sahip olan yeni bir personel eklenmesine engel olur ve uyarı verir. Aynı ada ve soyada sahip olan başka bir personel eklenmemiş olur.

```
CREATE OR REPLACE FUNCTION eksik_bilgi_kontrol()
RETURNS TRIGGER AS $$
BEGIN
    -- Eğer herhangi bir alan boş ise hata fırlat
    IF NEW.ad IS NULL OR NEW.ad = '' THEN
        RAISE EXCEPTION 'Ad alanı boş bırakılamaz.';
    END IF;
    IF NEW.soyad IS NULL OR NEW.soyad = '' THEN
        RAISE EXCEPTION 'Soyad alanı boş bırakılamaz.';
    END IF;
    IF NEW.telefon IS NULL OR NEW.telefon = '' THEN
        RAISE EXCEPTION 'Telefon alanı boş bırakılamaz.';
    END IF;
    IF NEW.meslek IS NULL OR NEW.meslek = '' THEN
        RAISE EXCEPTION 'Meslek alanı boş bırakılamaz.';
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
-- Insert komutu öncesinde çalışır
CREATE TRIGGER tetikleyici_eksik_bilgi_kontrol
BEFORE INSERT ON personel
FOR EACH ROW
```

Personel ekleme işlemi esnasında ad,soyad,telefon ve meslek alanlarından biri boş bırakılması durumunda hata mesajı yayınlar ve boş bırakılmasına engel olan bir trigger

```

-- Trigger:Soyadı büyük harfe çevirir
CREATE OR REPLACE FUNCTION soyad_buyuk_harf_yap()
RETURNS TRIGGER AS $$
BEGIN
    -- Eğer soyadı NULL değilse büyük harfe çevir
    IF NEW.soyad IS NOT NULL THEN
        NEW.soyad = UPPER(NEW.soyad);
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

--INSERT ve UPDATE işlemlerinde çalışır
CREATE TRIGGER tetikleyici_soyad_buyuk_harf
BEFORE INSERT OR UPDATE ON personel
FOR EACH ROW
EXECUTE FUNCTION soyad_buyuk_harf_yap();

```

Personel ekleme yapılırken soyadına yazılan veriyi büyük harfe çevirir ve veri tabanına bu şekilde kaydedilmesini sağlar.

```

-- Trigger:Telefon numarasının benzersizliğini kontrol eder
CREATE OR REPLACE FUNCTION telefon_numarasi_benzersizlik_kontrol()
RETURNS TRIGGER AS $$
BEGIN
    -- Yeni eklenen veya güncellenen telefon numarasının varlığını kontrol et
    IF EXISTS (
        SELECT 1
        FROM personel
        WHERE telefon = NEW.telefon AND personel_id <> NEW.personel_id
    ) THEN
        RAISE EXCEPTION 'Bu telefon numarası başka bir personel tarafından kullar
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

--INSERT ve UPDATE işlemlerinde çalışır
CREATE TRIGGER tetikleyici_telefon_benzersizlik
BEFORE INSERT OR UPDATE ON personel
FOR EACH ROW
EXECUTE FUNCTION telefon_numarasi_benzersizlik_kontrol();

```

Personel ekleme yaparken aynı telefon numaralarına sahip olan personellerin eklenmesine engel olur.

EKLEME,SİLME,ÇIKARMA VE GÜNCELLEME İŞLEMLERİNİN EKRAN GÖRÜNTÜLERİ

1-)Ekleme işlemi öncesi

Form1

Personel

personel_id	ad	soyad	telefon	meslek
1	Ahmet	Yılmaz	5551234567	Diş Hekimliği
3	Mehmet	Can	5449876543	Diş Hekimliği
4	Elif	Demir	5336549872	Diş Hekimliği
5	Fatma	Çelik	5442135487	Diş Hekimliği
7	Zeynep	Ak	5334567890	Hemşirelik
9	Selin	Yıldız	5339876541	Sekreter
25	Umut	USLU	1111112222	Hemşire

Ad:

Soyad:

Telefon:

Meslek:

EKLE SİL ARAMA GÜNCELLE LİSTELE

Her Meslekten Çalışan Sayısı Öğren

Harf Girin Belirtilen Harfla Başlayan Personeller

Personel Sayısı: Sayı

Son Eklenen Personel

Ekleme işlemi sonrası

Form1

Personel

personel_id	ad	soyad	telefon	meslek
1	Ahmet	Yılmaz	5551234567	Diş Hekimliği
3	Mehmet	Can	5449876543	Diş Hekimliği
4	Elif	Demir	5336549872	Diş Hekimliği
5	Fatma	Çelik	5442135487	Diş Hekimliği
7	Zeynep	Ak	5334567890	Hemşirelik
9	Selin	Yıldız	5339876541	Sekreter
25	Umut	USLU	1111112222	Hemşire
27	Ayşe	DEĞİRMENCI	5462533525	Sekreter

Ad:

Soyad:

Telefon:

Meslek:

EKLE SİL ARAMA GÜNCELLE LİSTELE

Her Meslekten Çalışan Sayısı Öğren

Harf Girin Belirtilen Harfla Başlayan Personeller

Personel Sayısı: Sayı

Son Eklenen Personel

2-)Silme İşleminin Öncesi

Form1

Personel

personel_id	ad	soyad	telefon	meslek
1	Ahmet	Yılmaz	5551234567	Diş Hekimliği
3	Mehmet	Can	5449876543	Diş Hekimliği
4	Elif	Demir	5336549872	Diş Hekimliği
5	Fatma	Çelik	5442135487	Diş Hekimliği
7	Zeynep	Ak	5334567890	Hemşirelik
9	Selin	Yıldız	5339876541	Sekreter
25	Umut	USLU	1111112222	Hemşire
27	Ayşe	DEĞİRME...	5462533525	Sekreter

Ad: Ayşe

Soyad: Değirmenci

Telefon: 5462533525

Meslek: Sekreter

EKLE SİL ARAMA GÜNCELLE LİSTELE

Her Meslekten Çalışan Sayısı Öğren

Harf Girin Belirtilen Harfla Başlayan Personeller

Personel Sayısı: Sayı

Son Eklenen Personel

Silme İşleminin Sonrası (Ad üzerinden silme işlemi gerçekleşir)

Form1

Personel

personel_id	ad	soyad	telefon	meslek
1	Ahmet	Yılmaz	5551234567	Diş Hekimliği
3	Mehmet	Can	5449876543	Diş Hekimliği
4	Elif	Demir	5336549872	Diş Hekimliği
5	Fatma	Çelik	5442135487	Diş Hekimliği
7	Zeynep	Ak	5334567890	Hemşirelik
9	Selin	Yıldız	5339876541	Sekreter
25	Umut	USLU	1111112222	Hemşire

Ad: Ayşe

Soyad:

Telefon:

Meslek:

EKLE SİL ARAMA GÜNCELLE LİSTELE

Her Meslekten Çalışan Sayısı Öğren

Harf Girin Belirtilen Harfla Başlayan Personeller

Personel Sayısı: Sayı

Son Eklenen Personel

3-)Arama İşlemi Öncesi

Form1

Personel

	personel_id	ad	soyad	telefon	meslek
▶	1	Ahmet	Yılmaz	5551234567	Dış Hekimliği
	3	Mehmet	Can	5449876543	Dış Hekimliği
	4	Elif	Demir	5336549872	Dış Hekimliği
	5	Fatma	Çelik	5442135487	Dış Hekimliği
	7	Zeynep	Ak	5334567890	Hemşirelik
	9	Selin	Yıldız	5339876541	Sekreter
	25	Umut	USLU	1111112222	Hemşire
*					

Ad:

Soyad:

Telefon:

Meslek:

EKLE SİL ARAMA GÜNCELLE LİSTELE

Her Meslekten Çalışan Sayısı Öğren

Harf Girin Belirtilen Harfla Başlayan Personeller

Personel Sayısı: Sayı

Son Eklenen Personel

Arama İşlemi Sonrası (Soyad üzerinden bu işlem yapılır)

Form1

Personel

	personel_id	ad	soyad	telefon
▶	1	Ahmet	Yılmaz	5551234567
*				

Ad:

Soyad:

Telefon:

Meslek:

EKLE SİL ARAMA GÜNCELLE LİSTELE

Her Meslekten Çalışan Sayısı Öğren

Harf Girin Belirtilen Harfla Başlayan Personeller

Personel Sayısı: Sayı

Son Eklenen Personel

4-)Güncelleme İşlemi Öncesi

Form1

Personel

personel_id	ad	soyad	telefon	meslek
1	Ahmet	Yılmaz	5551234567	Diş Hekimliği
3	Mehmet	Can	5449876543	Diş Hekimliği
4	Elif	Demir	5336549872	Diş Hekimliği
5	Fatma	Çelik	5442135487	Diş Hekimliği
7	Zeynep	Ak	5334567890	Hemşirelik
9	Selin	Yıldız	5339876541	Sekreter
25	Umut	USLU	1111112222	Hemşire

Ad:

Soyad:

Telefon:

Meslek:

EKLE SİL ARAMA GÜNCELLE LİSTELE

Her Meslekten Çalışan Sayısı Öğren

Harf Girin Belirtilen Harfla Başlayan Personeller

Personel Sayısı: Sayı

Son Eklenen Personel

Güncelleme İşlemi Sonrası(Ad üzerinde yapılıyor ve Zeynep ak ismini değiştirdik)

Form1

Personel

personel_id	ad	soyad	telefon	meslek
1	Ahmet	Yılmaz	5551234567	Diş Hekimliği
3	Mehmet	Can	5449876543	Diş Hekimliği
5	Fatma	Çelik	5442135487	Diş Hekimliği
25	Umut	USLU	1111112222	Hemşire
9	Selin	TANRIVERDİ	5467354234	Sekreter
7	Zeynep	ŞAHİN	5423387634	Diş Hekimliği

Ad:

Soyad:

Telefon:

Meslek:

EKLE SİL ARAMA GÜNCELLE LİSTELE

Her Meslekten Çalışan Sayısı Öğren

Harf Girin Belirtilen Harfla Başlayan Personeller

Personel Sayısı: Sayı

Son Eklenen Personel

Arayüz Kodlarım

```
using Npgsql;
using System;
using System.Collections;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace PostgresqlProjeArayuzu
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            NpgsqlConnection baglanti = new NpgsqlConnection("server=localhost; port=5432;
            Database=proje; user ID=postgres; password=12345");

            private void Form1_Load(object sender, EventArgs e)
            {
            }
        }
    }
}
```

```
private void listele_buton1_Click(object sender, EventArgs e)
```

```
{  
    string sorgu = "SELECT * FROM personel";  
    NpgsqlDataAdapter da = new NpgsqlDataAdapter(sorgu, baglanti);  
    DataSet ds = new DataSet();  
    da.Fill(ds);  
    dataGridView1.DataSource = ds.Tables[0];  
}
```

```
private void ekle_buton1_Click(object sender, EventArgs e)
```

```
{  
    baglanti.Open();  
    NpgsqlCommand komut1 = new NpgsqlCommand("insert into  
personel(ad,soyad,telefon,meslek) values (@p1,@p2,@p3,@p4)", baglanti);  
    komut1.Parameters.AddWithValue("@p1", ad_textbox1.Text);  
    komut1.Parameters.AddWithValue("@p2", soyad_textbox1.Text);  
    komut1.Parameters.AddWithValue("@p3", telefon_textbox1.Text);  
    komut1.Parameters.AddWithValue("@p4", meslek_textbox1.Text);  
    komut1.ExecuteNonQuery();  
    baglanti.Close();  
    MessageBox.Show("Kayıt İşlemi Başarılı Bir Şekilde Gerçekleşti", "Bilgi",  
MessageBoxButtons.OK, MessageBoxIcon.Information);  
}
```

```
private void sil_buton1_Click(object sender, EventArgs e)
```

```
{  
    baglanti.Open();  
    NpgsqlCommand komut = new NpgsqlCommand("delete from personel where  
ad=@p1", baglanti);  
    komut.Parameters.AddWithValue("@p1", ad_textbox1.Text);  
    komut.ExecuteNonQuery();
```

```

        baglanti.Close();

        MessageBox.Show("Silme İşlemi Başarılı Bir Şekilde Gerçekleşti", "Bilgi",
        MessageBoxButtons.OK, MessageBoxIcon.Stop);

    }

    private void guncelle_buton1_Click(object sender, EventArgs e)
    {
        baglanti.Open();

        NpgsqlCommand komut2 = new NpgsqlCommand("update personel set
        soyad=@p1,telefon=@p2,meslek=@p3 where ad=@p4", baglanti);

        komut2.Parameters.AddWithValue("@p1", soyad_textbox1.Text);
        komut2.Parameters.AddWithValue("@p2", telefon_textbox1.Text);
        komut2.Parameters.AddWithValue("@p3", meslek_textbox1.Text);
        komut2.Parameters.AddWithValue("@p4", ad_textbox1.Text);

        komut2.ExecuteNonQuery();

        MessageBox.Show("Güncelleme İşlemi Başarılı Bir Şekilde Gerçekleşti", "Bilgi",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);

        baglanti.Close();
    }

    private void arama_buton1_Click(object sender, EventArgs e)
    {

        baglanti.Open();

        string soyad = soyad_textbox1.Text;

```

```
        NpgsqlCommand komut = new NpgsqlCommand("SELECT * FROM  
personel_soyada_gore_ara(@p_soyad)", baglanti);
```

```
        komut.Parameters.AddWithValue("@p_soyad", soyad);
```

```
        NpgsqlDataAdapter da = new NpgsqlDataAdapter(komut);
```

```
        DataSet ds = new DataSet();
```

```
        da.Fill(ds);
```

```
        dataGridView1.DataSource = ds.Tables[0];
```

```
        baglanti.Close();
```

```
    }
```

```
private void personelsayisi_textbox1_Click(object sender, EventArgs e)
```

```
{
```

```
    baglanti.Open();
```

```
    NpgsqlCommand komut = new NpgsqlCommand("SELECT personel_sayisi()",  
baglanti);
```

```
    int toplamPersonelSayisi = Convert.ToInt32(komut.ExecuteScalar());
```

```
    personelsayisi_label1.Text = "Toplam Personel Sayısı: " + toplamPersonelSayisi;
```

```
    baglanti.Close();
```

```
}
```

```
private void soneklenenpersonel_button1_Click(object sender, EventArgs e)
```

```
{
```

```
    baglanti.Open();
```

```
    NpgsqlCommand komut = new NpgsqlCommand("SELECT * FROM  
son_eklenen_personel()", baglanti);
```

```
    NpgsqlDataAdapter da = new NpgsqlDataAdapter(komut);
```

```
    DataSet ds = new DataSet();
```

```
    da.Fill(ds);
```

```
    dataGridView1.DataSource = ds.Tables[0];
```

```
    baglanti.Close();
```

```
}
```

```
private void meslekgruppersonel_textbox1_Click_1(object sender, EventArgs e)
```

```
{
```

```
    string sorgu = "SELECT * FROM meslek_grup_personel_sayisi()";
```

```
    NpgsqlDataAdapter da = new NpgsqlDataAdapter(sorgu, baglanti);
```

```
    DataSet ds = new DataSet();
```

```
    da.Fill(ds);
```

```
    dataGridView1.DataSource = ds.Tables[0];
```

```
    baglanti.Close();
```

```
}
```

```
private void adlistele_button1_Click(object sender, EventArgs e)
```

```
{
```

```
    string girilenHarf = harfgirin_textbox1.Text;
```

```
    if (string.IsNullOrEmpty(girilenHarf) || girilenHarf.Length != 1)
```

```
    {
```

```
        MessageBox.Show("Lütfen yalnızca bir harf girin!", "Hata",  
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
```

```
        return;
```

```
    }
```

```
    string sorgu = "SELECT * FROM ad_baslayan_personel(@p_harf)";
```

```
    NpgsqlCommand komut = new NpgsqlCommand(sorgu, baglanti);
```

```
    komut.Parameters.AddWithValue("@p_harf", girilenHarf);
```

```
    NpgsqlDataAdapter da = new NpgsqlDataAdapter(komut);
```

```
    DataSet ds = new DataSet();
```

```
    da.Fill(ds);
```

```
    dataGridView1.DataSource = ds.Tables[0];
```

```
    baglanti.Close();
```

```
}
```

```
}
```

}