

**T.C.
SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**VERİTABANI YÖNETİM SİSTEMLERİ DERSİ
PROJE ÖDEVİ**

G181210102

GÜRKAN KAYA

gurkan.kaya@ogr.sakarya.edu.tr

İçerik

- Uygulamanın kısa tanıtımı, iş kuralları, ilişkisel şema
- Varlık Bağintı modeli
- SQL ifadeleri
- Saklı yordamlar , tetikleyiciler
- Uygulamaya ait ekran görüntüleri
- Uygulamanın kaynak kodlarının bulunduğu Github adresi
- Çalışmamı anlattığım videonun adresi

Uygulamanın Tanıtımı

Bu projede bir hastanenin veritabanı gerçekleştirmesi yapılmıştır. Veritabanı yönetim sistemi olarak PostgreSQL , programlama dili olarak C# kullanılmıştır.

İŞ KURALLARI

Hasta tablosu TC No, ad, soyad, telefon, ilçe alanlarını içerir.

Bir hasta birçok kez muayene olabilir, hiç olmayabilir de.

Bir muayene yalnızca bir hastaya aittir.

Muayene tablosu muayene id, doktor id, hasta tc, teşhis alanlarını içerir.

Bir hastaya birçok reçete yazılabilir, hiçbir reçete yazılmayabilir de.

Bir reçete yalnızca bir hastaya yazılır.

Reçete tablosu id, hasta tc, doktor id alanlarını içerir.

Bir hasta yalnızca bir ilçede bulunabilir. En az bir ilçede bulunmalıdır.

Bir ilçede birçok hasta bulunabilir.

İlçe tablosu ilçe id, il id, ilçe adı alanlarını içerir.

İl tablosu il id ve ad alanlarını içerir.

Bir ilde en az bir ve daha fazla ilçe bulunabilir.

Bir ilçe yalnızca bir ile aittir.

Bir hasta için bir yatış işlemi bulunabilir. Hiç bulunmayabilir de.

Bir yatış işlemi yalnızca bir hastaya aittir.

Yatış tablosu id, hasta tc, oda no, yatılacak gün alanlarını içerir.

Bir hasta birçok randevuya sahip olabilir veya hiçbir randevusu olmayabilir.

Bir randevu yalnızca bir hastaya aittir.

Randevu tablosu randevu id, hasta tc, doktor id, tarih alanlarını içerir.

Doktor tablosu id, branş, ad, soyad, telefon alanlarını içerir.

Bir doktorun birçok randevusu bulunabilir veya hiçbir randevusu bulunmayabilir.

Bir randevu yalnızca bir hastaya aittir.

Bir doktorun yalnızca bir branşı olabilir.

Bir branşa sahip birçok doktor bulunabilir.

Branş tablosu branş adı ve branş numarası alanlarını içerir.

Her branшта en az bir doktor bulunmak zorundadır.

Bir doktor hiçbir reçete yazmayabileceği gibi birçok reçete de yazabilir.

Bir reçete yalnızca bir doktor tarafından yazılır.

Bir doktor hiçbir hastayı muayene etmeyebileceği gibi birçok hastayı da muayene edebilir.

Bir muayeneyi sadece bir doktor yapar.

Hemşire tablosu id, ad, soyad, branş, telefon alanlarını içerir. Burada bahsedilen branş hemşirenin çalışacağı servisi temsil eder.

Bir hemşire yalnızca bir branşa sahiptir yani yalnızca bir serviste çalışabilir.

Bir branшта birçok hemşire bulunabilir yani bir serviste birçok hemşire çalışabilir. Her serviste en az bir hemşire çalışmak zorundadır.

Bir reçete en az bir ilaç içerir, daha fazla ilaç da içerebilir.

Bir ilaç hiçbir reçetede bulunmayabileceği gibi birçok reçetede bulunabilir de.

İlaç tablosu ilaç adı ve kullanım bilgisi alanlarını içerir.

Bir odada yalnızca bir yatış olur.

Bir yatış yalnızca bir odaya aittir.

Oda tablosu oda no ve oda durumu alanlarını içerir.

Duyurular tablosu id, başlık ve açıklama alanlarını içerir.

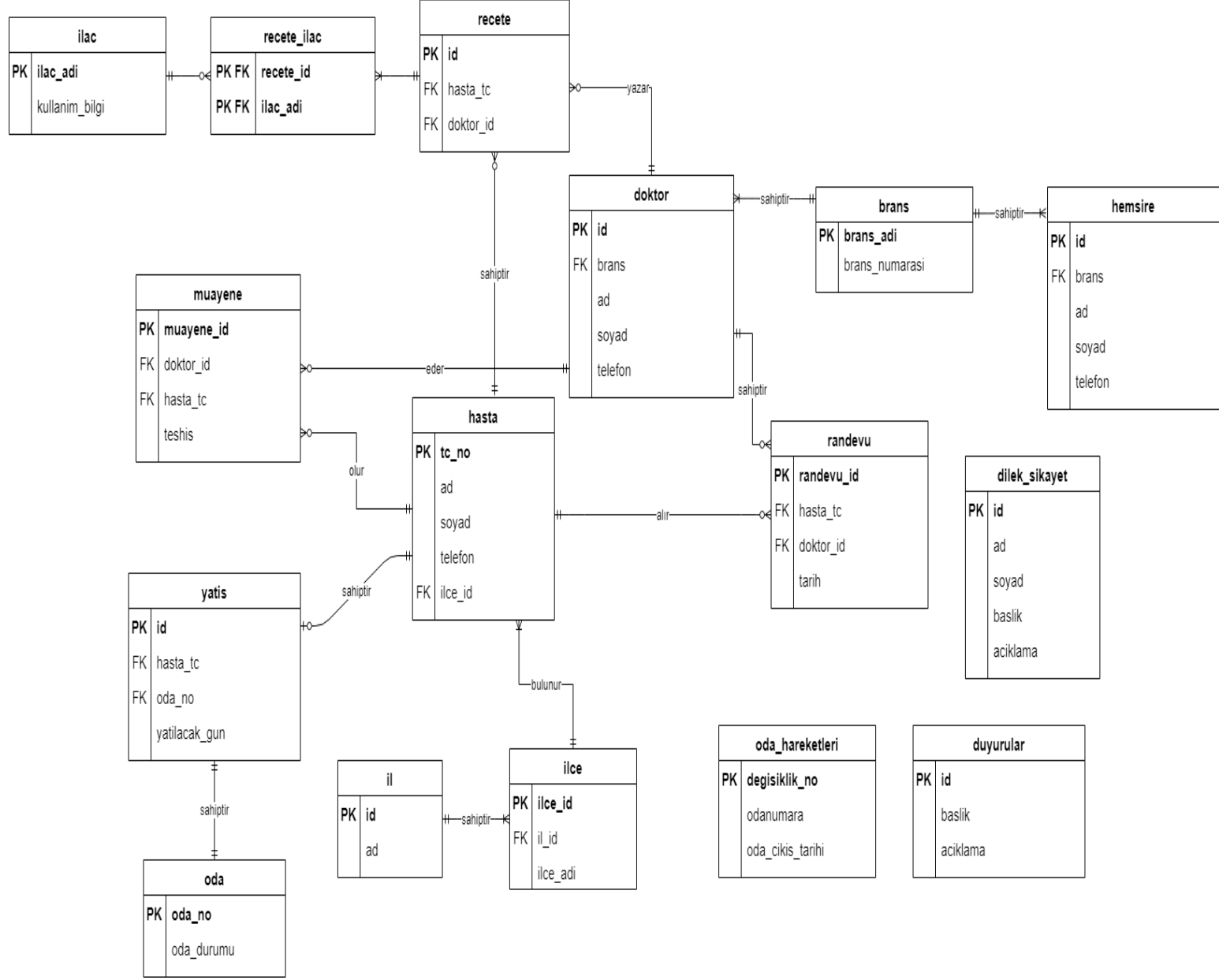
Oda hareketleri tablosu değişiklik no, oda numarası ve odadan çıkış tarihi alanlarını içerir.

Dilek şikayet tablosu id, ad, soyad, başlık, açıklama alanlarını içerir.

İLİŞKİSEL ŞEMA

- randevu (**randevu_id: integer** ,hasta_tc: bigint , doktor_id: integer, tarih: timestamp without time zone)
- recete (**id: integer** , hasta_tc: bigint , doktor_id: integer)
- oda_hareketleri (**degisiklik_no: integer** , odanumara: integer , oda_cikis_tarihi : timestamp without time zone)
- duyurular (**id: integer** , baslik: text , aciklama: text)
- doktor (**id: integer**, ad: text, soyad: text, brans: text , telefon: text)
- dilek_sikayet(**id: integer** , ad: text , aciklama: text , soyad: text, baslik: text)
- brans(**brans_adi: text** , brans_numarasi: integer)
- hasta (**tc_no: bigint** , ad: text , soyad: text , telefon: text , ilce_id: integer)
- il(**id: integer** , ad: character varying)
- hemsire (**id :integer** , ad: text , soyad: text , telefon: text , brans: text)
- ilac (**ilac_adi: text** , kullanim_bilgi: text)
- ilce (**ilce_id: integer** ,ilce_adi: text , il_id: integer)
- muayene (**muayene_id: integer** , doktor_id: integer ,hasta_tc: bigint, teşhis: text)
- oda (**oda_no: integer** , oda_durumu: text)
- recete_ilac (**recete_id: integer** , **ilac_adi: text**)
- yatis (**id: integer** , hasta_tc: bigint , oda_no: integer , yatilacak_gun: integer)

VARLIK BAĞINTI DİYAGRAMI



SQL İFADELERİ

--

-- PostgreSQL database dump

--

-- Dumped from database version 13.0

-- Dumped by pg_dump version 13.1

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;

--

-- Name: Hastane; Type: DATABASE; Schema: -; Owner: postgres

--

--CREATE DATABASE "Hastane" WITH TEMPLATE = template0 ENCODING = 'UTF8' LOCALE =
'Turkish_Turkey.1254';

ALTER DATABASE "Hastane" OWNER TO postgres;

--\connect "Hastane"

```

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;


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

CREATE FUNCTION public.bosodalarilistele() RETURNS TABLE(odano integer, odadurumu text)
    LANGUAGE plpgsql
    AS $$
BEGIN
    RETURN QUERY SELECT
        "oda_no",
        "oda_durumu"
    FROM
        "oda"
    WHERE
        "oda_durumu"='Boş'
    order by oda_no asc;
END; $$;


ALTER FUNCTION public.bosodalarilistele() OWNER TO postgres;

```

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

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

```
    LANGUAGE plpgsql
```

```
    AS $$
```

```
BEGIN
```

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

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

```
return NEW;
```

```
end;
```

```
$$;
```

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

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

```
CREATE FUNCTION public.doluodalariliste() RETURNS TABLE(odano integer, odadurumu text)
```

```
    LANGUAGE plpgsql
```

```
    AS $$
```

```
BEGIN
```

```
    RETURN QUERY SELECT
```

```
        "oda_no",
```

```
        "oda_durumu"
```

```
    FROM
```

```
        "oda"
```



```
WHERE
    "oda_durumu"='Dolu'
    order by oda_no asc ;
END; $$;
```

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

```
--
-- Name: duyuru_ekle(integer, text, text); Type: FUNCTION; Schema: public; Owner: postgres
--
```

```
CREATE FUNCTION public.duyuru_ekle(id integer, baslik text, aciklama text) RETURNS void
    LANGUAGE plpgsql
    AS $_$
begin
INSERT INTO duyurular(id, baslik,aciklama)
VALUES ($1,$2,$3);
    END;
    $_$;
```

```
ALTER FUNCTION public.duyuru_ekle(id integer, baslik text, aciklama text) OWNER TO postgres;
```

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

```
CREATE FUNCTION public.hasta_sayisi() RETURNS integer
    LANGUAGE plpgsql
    AS $$
```

```
DECLARE

    hasta_sayi integer;

BEGIN

    SELECT count("tc_no") into hasta_sayi from "hasta";

    RETURN hasta_sayi;

END;

$$;
```

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

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

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

    LANGUAGE plpgsql

    AS $$

        BEGIN

            insert into "oda_hareketleri"(odanumara,oda_cikis_tarihi) VALUES
(OLD.oda_no,current_timestamp::timestamp);

            return old;

        END;

    $$;
```

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

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

--

CREATE FUNCTION public.yatis_eklendiginde_dolu_yap() RETURNS trigger

LANGUAGE plpgsql

AS \$\$

DECLARE

odano int;

BEGIN

odano=NEW.oda_no ;

UPDATE oda set oda_durumu='Dolu' WHERE

oda_no=odano;

return NEW;

END;

\$\$;

ALTER FUNCTION public.yatis_eklendiginde_dolu_yap() OWNER TO postgres;

--

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

--

CREATE FUNCTION public.yatis_silindiginde_bos_yap() RETURNS trigger

LANGUAGE plpgsql

AS \$\$

DECLARE

odano int;

BEGIN

odano=OLD.oda_no ;

UPDATE oda set oda_durumu='Boş' WHERE

oda_no=odano;

```
return OLD;  
    END;  
    $$;
```

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

```
SET default_tablespace = '';
```

```
SET default_table_access_method = heap;
```

```
--
```

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

```
--
```

```
CREATE TABLE public.brans (  
    brans_adi text NOT NULL,  
    brans_numarasi integer NOT NULL  
);
```

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

```
--
```

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

```
--
```

```
CREATE TABLE public.dilek_sikayet (  
    ad text NOT NULL,  
    aciklama text NOT NULL,  
    id integer NOT NULL,
```

```
    soyad text NOT NULL,  
    baslik text NOT NULL  
);
```

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

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

```
CREATE TABLE public.doktor (  
    id integer NOT NULL,  
    ad text NOT NULL,  
    soyad text NOT NULL,  
    brans text NOT NULL,  
    telefon text NOT NULL  
);
```

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

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

```
CREATE TABLE public.uyurular (  
    id integer NOT NULL,  
    baslik text NOT NULL,  
    aciklama text NOT NULL  
);
```

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

```
--
```

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

```
--
```

```
CREATE TABLE public.hasta (
```

```
    tc_no bigint NOT NULL,
```

```
    ad text NOT NULL,
```

```
    soyad text NOT NULL,
```

```
    telefon text NOT NULL,
```

```
    ilce_id integer NOT NULL
```

```
);
```

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

```
--
```

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

```
--
```

```
CREATE TABLE public.hemsire (
```

```
    id integer NOT NULL,
```

```
    ad text NOT NULL,
```

```
    soyad text NOT NULL,
```

```
    telefon text NOT NULL,
```

```
    brans text NOT NULL
```

```
);
```

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

```
--
```

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

```
--
```

```
CREATE TABLE public.il (  
    ad character varying NOT NULL,  
    id integer NOT NULL  
);
```

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

```
--
```

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

```
--
```

```
CREATE TABLE public.ilac (  
    ilac_adi text NOT NULL,  
    kullanim_bilgi text NOT NULL  
);
```

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

```
--
```

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

```
--
```

```
CREATE TABLE public.ilce (  
    ilce_adi text NOT NULL,  
    il_id integer NOT NULL,  
    ilce_id integer NOT NULL  
);
```

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

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

```
CREATE TABLE public.muayene (  
    doktor_id integer NOT NULL,  
    hasta_tc bigint NOT NULL,  
    muayene_id integer NOT NULL,  
    teshis text NOT NULL  
);
```

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

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

```
CREATE TABLE public.oda (  
    oda_no integer NOT NULL,  
    oda_durumu text  
);
```



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

```
--
```

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

```
--
```

```
CREATE TABLE public.oda_hareketleri (  
    degisiklik_no integer NOT NULL,  
    odanumara integer,  
    oda_cikis_tarihi timestamp without time zone  
);
```

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

```
--
```

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

```
--
```

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

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

--

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

--

ALTER SEQUENCE public.oda_hareketleri_degisiklik_no_seq OWNED BY
public.oda_hareketleri.degisiklik_no;

--

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

--

CREATE TABLE public.randevu (
 hasta_tc bigint NOT NULL,
 doktor_id integer NOT NULL,
 tarih timestamp without time zone NOT NULL,
 randevu_id integer NOT NULL
);

ALTER TABLE public.randevu OWNER TO postgres;

--

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

--

CREATE TABLE public.recete (
 id integer NOT NULL,
 hasta_tc bigint NOT NULL,
 doktor_id integer NOT NULL

```
);
```

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

```
--
```

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

```
--
```

```
CREATE TABLE public.recete_ilac (
```

```
    recete_id integer NOT NULL,
```

```
    ilac_adi text NOT NULL
```

```
);
```

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

```
--
```

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

```
--
```

```
CREATE TABLE public.yatis (
```

```
    hasta_tc bigint NOT NULL,
```

```
    oda_no integer NOT NULL,
```

```
    yatilacak_gun integer NOT NULL,
```

```
    id integer NOT NULL
```

```
);
```

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

--

-- Name: oda_hareketleri_degisiklik_no; Type: DEFAULT; Schema: public; Owner: postgres

--

```
ALTER TABLE ONLY public.oda_hareketleri ALTER COLUMN degisiklik_no SET DEFAULT  
nextval('public.oda_hareketleri_degisiklik_no_seq'::regclass);
```

--

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

--

```
INSERT INTO public.brans VALUES
```

```
    ('Dahiliye', 1),
```

```
    ('Deri ve Zührevi Hastalıklar', 2),
```

```
    ('Fizyoterapi', 3),
```

```
    ('Göz Hastalıkları', 4),
```

```
    ('Kulak Burun Boğaz', 6),
```

```
    ('Ortopedi ve Travmatoloji', 7),
```

```
    ('Psikiyatri', 8),
```

```
    ('Nöroloji', 5);
```

--

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

--

```
INSERT INTO public.dilek_sikayet VALUES
```

```
    ('Ali', 'Hastanenin hijyen konusunda eksikleri bulunuyor', 1, 'Güven', 'Temizlik'),
```

```
    ('BURÇİN', 'Hastane yemekleri aldığı firmayı değiştirmeli!', 2, 'Kardeş', 'Yemekler');
```

--

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

--

INSERT INTO public.doktor VALUES

(1, 'AHMET', 'YILMAZ', 'Dahiliye', '5553211489'),
(5, 'TANJU', 'ÇELIKKAN', 'Psikiyatri', '5421526157'),
(4, 'MUSTAFA', 'ALDINÇ', 'Fizyoterapi', '5895741223'),
(3, 'NILGÜN', 'ÇELİK', 'Deri ve Zührevi Hastalıklar', '5245877413'),
(6, 'SU', 'CANDANER', 'Göz Hastalıkları', '5391520441'),
(7, 'GÖKCAN', 'AŞAN', 'Ortopedi ve Travmatoloji', '5165581918'),
(8, 'SARP', 'GÜLSAYIN', 'Nöroloji', '5825194921'),
(2, 'HÜSEYİN', 'AKIN', 'Kulak Burun Boğaz', '5712596381');

--

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

--

INSERT INTO public.duyurular VALUES

(1, 'Covid-19', 'Koronavirüs tedbirleri kapsamında randevular yalnızca internet üzerinden alınabilmektedir.'),
(2, 'Yemek Listesi', '29.12.2021 tarihli yemek listesi: Tarhana Çorbası, Sulu Köfte, Salata');

--

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

--

INSERT INTO public.hasta VALUES

```
(37758167991, 'RUKİYE ÖZDEN ', 'SARI', '5587419347', 2),  
(95327413957, 'NECLA', 'ALABAY', '5511712510', 6),  
(69287101233, 'CENGİZHAN', 'ERDEN', '5534574241', 5),  
(77201681510, 'DİLŞAH', 'ARSAL', '5423090121', 3),  
(88442647458, 'SENA', 'KOÇYİĞİT', '5955655013', 4),  
(67249922331, 'AÇELYA', 'YAKIŞAN', '5083722667', 1);
```

--

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

--

INSERT INTO public.hemsire VALUES

```
(3, 'ELİF', 'UYGUN', '5423659508', 'Dahiliye'),  
(4, 'ERKAN', 'DOĞA', '5040145974', 'Ortopedi ve Travmatoloji'),  
(5, 'AYŞE GÜL', 'GÜDER', '5032147896', 'Deri ve Zührevi Hastalıklar'),  
(1, 'AYŞE', 'TEKİN', '05587896985', 'Göz Hastalıkları'),  
(2, 'NURİYE', 'CANDAN', '05897462545', 'Psikiyatri');
```

--

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

--

INSERT INTO public.il VALUES

```
('Ankara', 1);
```

--

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

--

```
INSERT INTO public.ilac VALUES
```

```
    ('majezik', 'günde 2 defa '),  
    ('parol', 'günde 1 defa'),  
    ('aspirin', 'günde 3 defa'),  
    ('nurofen', '12 saate 1 kez'),  
    ('aferin', 'günde 2 defa'),  
    ('katarin', 'günde 1 kez');
```

```
--
```

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

```
--
```

```
INSERT INTO public.ilce VALUES
```

```
    ('Çankaya', 1, 1),  
    ('Elmadag', 1, 2),  
    ('Sincan', 1, 3),  
    ('Altındağ', 1, 4),  
    ('Kızılcahamam', 1, 5),  
    ('Etimesgut', 1, 6);
```

```
--
```

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

```
--
```

```
INSERT INTO public.muayene VALUES
```

```
    (1, 69287101233, 1, 'Guatr'),  
    (4, 77201681510, 2, 'Covid-19');
```

--

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

--

INSERT INTO public.oda VALUES

(7, 'Boş'),
(8, 'Boş'),
(15, 'Boş'),
(12, 'Boş'),
(13, 'Boş'),
(14, 'Boş'),
(1, 'Dolu'),
(9, 'Dolu'),
(10, 'Dolu'),
(2, 'Dolu'),
(11, 'Boş'),
(3, 'Boş'),
(4, 'Boş'),
(6, 'Boş'),
(5, 'Boş');

--

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

--

INSERT INTO public.oda_hareketleri VALUES

(1, 5, '2020-12-28 23:50:27.829352'),
(2, 11, '2020-12-28 23:51:02.580178'),
(3, 1, '2020-12-29 18:14:35.032395'),


```
(4, 2, '2020-12-29 18:14:35.032395'),  
(5, 8, '2020-12-29 18:14:35.032395'),  
(6, 15, '2020-12-29 18:14:35.032395'),  
(7, 6, '2020-12-29 21:00:20.785655'),  
(8, 11, '2020-12-29 21:05:07.298947'),  
(9, 6, '2020-12-29 21:12:37.87675'),  
(10, 6, '2020-12-29 21:20:21.77982'),  
(11, 6, '2020-12-29 21:27:02.477871'),  
(12, 5, '2020-12-29 22:05:12.65025'),  
(13, 5, '2020-12-29 23:10:27.559889'),  
(14, 5, '2020-12-29 23:37:38.264808');
```

```
--
```

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

```
--
```

```
INSERT INTO public.randevu VALUES
```

```
(69287101233, 3, '2020-12-16 00:00:00', 1),  
(95327413957, 4, '2021-02-17 00:00:00', 2);
```

```
--
```

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

```
--
```

```
INSERT INTO public.recete VALUES
```

```
(1, 95327413957, 6),  
(2, 67249922331, 7);
```

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

```
INSERT INTO public.recete_ilac VALUES
```

```
    (1, 'parol'),  
    (1, 'majezik'),  
    (2, 'katarin'),  
    (2, 'aspirin'),  
    (2, 'aferin');
```

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

```
INSERT INTO public.yatis VALUES
```

```
    (77201681510, 1, 3, 1),  
    (88442647458, 9, 12, 2),  
    (67249922331, 10, 7, 3),  
    (69287101233, 2, 15, 4);
```

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

```
SELECT pg_catalog.setval('public.oda_hareketleri_degisiklik_no_seq', 14, true);
```

```
--
```

-- Name: brans Brans_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.brans

ADD CONSTRAINT "Brans_pkey" PRIMARY KEY (brans_adi);

--

-- Name: doktor Doktor_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.doktor

ADD CONSTRAINT "Doktor_pkey" PRIMARY KEY (id);

--

-- Name: hasta Hasta_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.hasta

ADD CONSTRAINT "Hasta_pkey" PRIMARY KEY (tc_no);

--

-- Name: il Il_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.il

ADD CONSTRAINT "Il_pkey" PRIMARY KEY (id);

```
--  
-- Name: randevu Randevu_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres  
--
```

```
ALTER TABLE ONLY public.randevu  
    ADD CONSTRAINT "Randevu_pkey" PRIMARY KEY (randevu_id);
```

```
--  
-- Name: dilek_sikayet dilek_sikayet_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres  
--
```

```
ALTER TABLE ONLY public.dilek_sikayet  
    ADD CONSTRAINT dilek_sikayet_pkey PRIMARY KEY (id);
```

```
--  
-- Name: duyurular duyurular_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres  
--
```

```
ALTER TABLE ONLY public.duyurular  
    ADD CONSTRAINT duyurular_pkey PRIMARY KEY (id);
```

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

```
ALTER TABLE ONLY public.hemsire  
    ADD CONSTRAINT hemsire_pkey PRIMARY KEY (id);
```

--

-- Name: ilac ilac_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.ilac

ADD CONSTRAINT ilac_pkey PRIMARY KEY (ilac_adi);

--

-- Name: ilce ilce_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.ilce

ADD CONSTRAINT ilce_pkey PRIMARY KEY (ilce_id);

--

-- Name: muayene muayene_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.muayene

ADD CONSTRAINT muayene_pkey PRIMARY KEY (muayene_id);

--

-- Name: oda_hareketleri oda_hareketleri_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.oda_hareketleri

```
ADD CONSTRAINT oda_hareketleri_pkey PRIMARY KEY (degisiklik_no);
```

```
--
```

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

```
--
```

```
ALTER TABLE ONLY public.oda
```

```
ADD CONSTRAINT oda_pkey PRIMARY KEY (oda_no);
```

```
--
```

```
-- Name: recete_ilac recete_ilac_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
```

```
--
```

```
ALTER TABLE ONLY public.recete_ilac
```

```
ADD CONSTRAINT recete_ilac_pkey PRIMARY KEY (recete_id, ilac_adi);
```

```
--
```

```
-- Name: recete recete_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
```

```
--
```

```
ALTER TABLE ONLY public.recete
```

```
ADD CONSTRAINT recete_pkey PRIMARY KEY (id);
```

```
--
```

```
-- Name: brans unique_Brans_Ad; Type: CONSTRAINT; Schema: public; Owner: postgres
```

```
--
```

```
ALTER TABLE ONLY public.brans
```

```
ADD CONSTRAINT "unique_Brans_Ad" UNIQUE (brans_adi);
```

```
--
```

```
-- Name: doktor unique_Doktor_ID; Type: CONSTRAINT; Schema: public; Owner: postgres
```

```
--
```

```
ALTER TABLE ONLY public.doktor
```

```
ADD CONSTRAINT "unique_Doktor_ID" UNIQUE (id);
```

```
--
```

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

```
--
```

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

```
ADD CONSTRAINT "unique_Hasta_TCNo" UNIQUE (tc_no);
```

```
--
```

```
-- Name: il unique_Il_id; Type: CONSTRAINT; Schema: public; Owner: postgres
```

```
--
```

```
ALTER TABLE ONLY public.il
```

```
ADD CONSTRAINT "unique_Il_id" UNIQUE (id);
```

```
--
```

```
-- Name: randevu unique_Randevu_Tarih; Type: CONSTRAINT; Schema: public; Owner: postgres
```

```
--
```

```
ALTER TABLE ONLY public.randevu
```

```
ADD CONSTRAINT "unique_Randevu_Tarih" UNIQUE (tarih);
```

```
--
```

```
-- Name: brans unique_brans_brans_numarasi; Type: CONSTRAINT; Schema: public; Owner: postgres
```

```
--
```

```
ALTER TABLE ONLY public.brans
```

```
ADD CONSTRAINT unique_brans_brans_numarasi UNIQUE (brans_numarasi);
```

```
--
```

```
-- Name: dilek_sikayet unique_dilek_sikayet_id; Type: CONSTRAINT; Schema: public; Owner: postgres
```

```
--
```

```
ALTER TABLE ONLY public.dilek_sikayet
```

```
ADD CONSTRAINT unique_dilek_sikayet_id UNIQUE (id);
```

```
--
```

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

```
--
```

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

```
ADD CONSTRAINT unique_hemsire_id UNIQUE (id);
```

```
--
```

```
-- Name: ilac unique_ilac_ilac_adi; Type: CONSTRAINT; Schema: public; Owner: postgres
```


--

ALTER TABLE ONLY public.ilac

ADD CONSTRAINT unique_ilac_ilac_adi UNIQUE (ilac_adi);

--

-- Name: ilce unique_ilce_ilce_id; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.ilce

ADD CONSTRAINT unique_ilce_ilce_id UNIQUE (ilce_id);

--

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

--

ALTER TABLE ONLY public.oda

ADD CONSTRAINT unique_oda_oda_no UNIQUE (oda_no);

--

-- Name: recete unique_recete_id; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.recete

ADD CONSTRAINT unique_recete_id UNIQUE (id);

--

-- Name: yatis unique_yatis_id; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.yatis

ADD CONSTRAINT unique_yatis_id UNIQUE (id);

--

-- Name: yatis yatis_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.yatis

ADD CONSTRAINT yatis_pkey PRIMARY KEY (id);

--

-- Name: doktor doktor_eklerken_kontrol; Type: TRIGGER; Schema: public; Owner: postgres

--

CREATE TRIGGER doktor_eklerken_kontrol BEFORE INSERT OR UPDATE ON public.doktor FOR EACH
ROW EXECUTE FUNCTION public.doktor_eklerken();

--

-- Name: yatis oda_cikis_yapildiginda; Type: TRIGGER; Schema: public; Owner: postgres

--

CREATE TRIGGER oda_cikis_yapildiginda AFTER DELETE ON public.yatis FOR EACH ROW EXECUTE
FUNCTION public.oda_cikis_hareketleri();

--

-- Name: yatis yatis_eklendiginde_dolu; Type: TRIGGER; Schema: public; Owner: postgres

--

```
CREATE TRIGGER yatis_eklendiginde_dolu AFTER INSERT OR UPDATE ON public.yatis FOR EACH ROW
EXECUTE FUNCTION public.yatis_eklendiginde_dolu_yap();
```

--

-- Name: yatis yatis_silindiginde_bos; Type: TRIGGER; Schema: public; Owner: postgres

--

```
CREATE TRIGGER "yatis_silindiginde_bos" AFTER DELETE OR UPDATE ON public.yatis FOR EACH ROW
EXECUTE FUNCTION public.yatis_silindiginde_bos_yap();
```

--

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

--

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

```
    ADD CONSTRAINT brans_hemsire_fkey FOREIGN KEY (brans) REFERENCES public.brans(brans_adi)
    MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;
```

--

-- Name: muayene doktor-muayene-fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

```
ALTER TABLE ONLY public.muayene
```

```
    ADD CONSTRAINT "doktor-muayene-fkey" FOREIGN KEY (doktor_id) REFERENCES public.doktor(id)
    MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;
```

--

-- Name: doktor doktorbrans_pkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.doktor

ADD CONSTRAINT doktorbrans_pkey FOREIGN KEY (brans) REFERENCES public.brans(brans_adi)
MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: randevu doktorrandevu_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.randevu

ADD CONSTRAINT doktorrandevu_fkey FOREIGN KEY (doktor_id) REFERENCES public.doktor(id)
MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: recete doktorrecete-fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.recete

ADD CONSTRAINT "doktorrecete-fkey" FOREIGN KEY (doktor_id) REFERENCES public.doktor(id)
MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: muayene hasta-muayene-fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

```
ALTER TABLE ONLY public.muayene
```

```
    ADD CONSTRAINT "hasta-muayene-fkey" FOREIGN KEY (hasta_tc) REFERENCES public.hasta(tc_no)  
    MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;
```

```
--
```

```
-- Name: hasta hastailce-fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres
```

```
--
```

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

```
    ADD CONSTRAINT "hastailce-fkey" FOREIGN KEY (ilce_id) REFERENCES public.ilce(ilce_id) MATCH  
    FULL ON UPDATE CASCADE ON DELETE CASCADE;
```

```
--
```

```
-- Name: randevu hastarandevu_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres
```

```
--
```

```
ALTER TABLE ONLY public.randevu
```

```
    ADD CONSTRAINT hastarandevu_fkey FOREIGN KEY (hasta_tc) REFERENCES public.hasta(tc_no)  
    MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;
```

```
--
```

```
-- Name: yatis hastayatis_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres
```

```
--
```

```
ALTER TABLE ONLY public.yatis
```

```
    ADD CONSTRAINT hastayatis_fkey FOREIGN KEY (hasta_tc) REFERENCES public.hasta(tc_no)  
    MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;
```

--

-- Name: ilce illlce_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.ilce

ADD CONSTRAINT "illlce_fkey" FOREIGN KEY (il_id) REFERENCES public.il(id) MATCH FULL ON
UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: recete_ilac ilac-recete-fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.recete_ilac

ADD CONSTRAINT "ilac-recete-fkey" FOREIGN KEY (ilac_adi) REFERENCES public.ilac(ilac_adi)
MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: yatis odayatis_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.yatis

ADD CONSTRAINT odayatis_fkey FOREIGN KEY (oda_no) REFERENCES public.oda(oda_no) MATCH
FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: recete recete-hasta-fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.recete

```
    ADD CONSTRAINT "recete-hasta-fkey" FOREIGN KEY (hasta_tc) REFERENCES public.hasta(tc_no)
MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;
```

```
--
```

```
-- Name: recete_ilac recte-ilac-fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres
```

```
--
```

```
ALTER TABLE ONLY public.recete_ilac
```

```
    ADD CONSTRAINT "recte-ilac-fkey" FOREIGN KEY (recete_id) REFERENCES public.recete(id) MATCH
FULL ON UPDATE CASCADE ON DELETE CASCADE;
```

```
--
```

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

```
--
```

FONKSİYONLAR

```
CREATE OR REPLACE FUNCTION public.bosOdalariListele()  
  RETURNS TABLE(odaNo integer, odaDurumu text)  
  LANGUAGE plpgsql  
  AS $function$  
  BEGIN  
    RETURN QUERY SELECT  
      "oda_no",  
      "oda_durumu"  
    FROM  
      "oda"  
    WHERE  
      "oda_durumu"='Boş' ;  
  END; $function$
```

Fonksiyonun C# üzerinde kullanımı:

```
string sorgu = "select * from bosOdalariListele()";
```

```
CREATE OR REPLACE FUNCTION public.doluOdalariListele()  
  RETURNS TABLE(odaNo integer, odaDurumu text)  
  LANGUAGE plpgsql  
  AS $function$  
  BEGIN  
    RETURN QUERY SELECT  
      "oda_no",  
      "oda_durumu"
```



```
FROM
    "oda"
WHERE
    "oda_durumu"='Dolu' ;
END; $function$
```

Fonksiyonun C# üzerinde kullanımı:

```
string sorgu = "select * from doluOdalarListele()";
```

```
CREATE or replace FUNCTION duyuru_ekle(id int,baslik text,aciklama text)
RETURNS VOID
LANGUAGE plpgsql
AS $function$
BEGIN
INSERT INTO duyurular(id, baslik,aciklama)
VALUES ($1,$2,$3);
END;
$function$
```

Fonksiyonun C# üzerinde kullanımı:

```
NpgsqlCommand kmt = new NpgsqlCommand(@"duyuru_ekle", baglanti);
kmt.CommandType = CommandType.StoredProcedure;
kmt.Parameters.AddWithValue("id", int.Parse(textBox5.Text));
kmt.Parameters.AddWithValue("baslik", textBox1.Text);
kmt.Parameters.AddWithValue("aciklama", textBox2.Text);
kmt.ExecuteNonQuery();
```

```
CREATE OR REPLACE FUNCTION hasta_sayisi()
RETURNS integer
LANGUAGE plpgsql
AS $function$
    DECLARE
        hasta_sayi integer;
    BEGIN
        SELECT count("tc_no") into hasta_sayi from "hasta";
        RETURN hasta_sayi;
    END;
$function$
```

Fonksiyonun C# üzerinde kullanımı:

```
string sorgu2 = "select * from hasta_sayisi()";
NpgsqlCommand komut2 = new NpgsqlCommand(sorgu2, baglanti);
NpgsqlDataReader dr = komut2.ExecuteReader();
while (dr.Read())
{
    label8.Text = dr.GetValue(0).ToString();
}
```

TRİGGERLAR

```
CREATE OR REPLACE FUNCTION yatis_eklendiginde_dolu_yap()
```

```
RETURNS trigger
```

```
LANGUAGE plpgsql
```

```
AS $function$
```

```
    DECLARE
```

```
        odano int;
```

```
    BEGIN
```

```
        odano=NEW.oda_no ;
```

```
        UPDATE oda set oda_durumu='Dolu' WHERE
```

```
            oda_no=odano;
```

```
        return NEW;
```

```
    END;
```

```
    $function$
```

```
CREATE TRIGGER yatis_eklendiginde_dolu
```

```
after insert or update
```

```
ON yatis
```

```
for EACH row
```

```
execute PROCEDURE yatis_eklendiginde_dolu_yap();
```

```
CREATE OR REPLACE FUNCTION yatis_silindiginde_bos_yap()
```

```
RETURNS trigger
```

```
LANGUAGE plpgsql
```

```
AS $function$
```

```
    DECLARE
```

```
        odano int;
```

```
    BEGIN
```

```
        odano=OLD.oda_no ;
```

```
        UPDATE oda set oda_durumu='Boş' WHERE
```

```
            oda_no=odano;
```

```
return OLD;

    END;

$function$
```

```
CREATE TRIGGER yatis_silindiginde_boş
after delete or update
ON yatis
for EACH row
execute PROCEDURE yatis_silindiginde_bos_yap();
```

```
CREATE OR REPLACE FUNCTION doktor_eklerken()
returns TRIGGER
as
$function$
BEGIN
NEW.ad=UPPER(NEW.ad);
NEW.soyad=UPPER(NEW.soyad);
return NEW;
end;
$function$
language "plpgsql";
```

```
create TRIGGER doktor_eklerken_kontrol
before insert or update on doktor
for each row
execute procedure doktor_eklerken();
```

```
CREATE OR REPLACE FUNCTION oda_cikis_hareketleri()
RETURNS trigger
LANGUAGE plpgsql
AS $function$
    BEGIN
        insert into "oda_hareketleri"(odanumara,oda_cikis_tarihi) VALUES
(OLD.oda_no,current_timestamp::timestamp);
    return old;
    END;
$function$
```

```
CREATE TRIGGER "oda_cikis_yapildiginda"
AFTER delete ON yatis
FOR EACH ROW
EXECUTE FUNCTION oda_cikis_hareketleri();
```

UYGULAMANIN EKRAN GÖRÜNTÜLERİ

Form2

DOKTOR İŞLEMLERİ

RANDEVU İŞLEMLERİ

REÇETE İŞLEMLERİ

DUYURULAR

HASTA İŞLEMLERİ

MUAYENE İŞLEMLERİ

BRANŞ İŞLEMLERİ

DİLEK-ŞİKAYET

HEMŞİRE İŞLEMLERİ

YATIŞ İŞLEMLERİ

ODA DURUMU LİSTELE

ÇIKIŞ

DOKTOR

ID

AD

SOYAD

TELEFON

BRANŞ

Dahiliye

ID İle Doktor Ara:

Doktor Ara



EKLE

SİL

GÜNCELLE

	id	ad	soyad	brans	telefon
▶	1	AHMET	YILMAZ	Dahiliye	5553211489
	2	HÜSEYİN	AKIN	Kulak Burun Boğaz	5712596381
	3	NILGÜN	ÇELİK	Deri ve Zührevi Hastalı...	5245877413
	4	MUSTAFA	ALDINÇ	Fizyoterapi	5895741223
	5	TANJU	ÇELİKKAN	Psikiyatri	5421526157
	6	SU	CANDANER	Göz Hastalıkları	5391520441
	7	GÖKCAN	AŞAN	Ortopedi ve Travmatoloji	5165581918

RANDEVU

HASTA TC

DOKTOR ADI

TARİH

RANDEVU ID

AHMET YILMAZ

__/__/__

EKLE

SİL

GÜNCELLE

	hasta_tc	doktoradi	doktorsoyadi	randevuid	tarih
▶	95327413957	GÖKCAN	AŞAN	2	3.12.2021
	67249922331	HÜSEYİN	AKIN	3	17.08.2021
	69287101233	NILGÜN	ÇELİK	1	16.12.2020
•					

HASTA

TC KİMLİK NO

AD

SOYAD

TELEFON

İLÇE

TC ile Hasta Ara:

Hasta Ara

EKLE

SİL

GÜNCELLE

TOPLAM HASTA SAYISI : 7

	tc_no	ad	soyad	telefon	ilce_id
▶	45733958098	HÜSEYİN ON...	KEMİK	5574866521	1
	37758167991	RUKİYE ÖZD...	SARI	5587419347	2
	95327413957	NECLA	ALABAY	5511712510	6
	69287101233	CENGİZHAN	ERDEN	5534574241	5
	77201681510	DİLŞAH	ARSAL	5423090121	3
	88442647458	SENA	KOCYİĞİT	5955655013	4

HEMŞİRE

ID

AD

SOYAD

TELEFON

SERVİS

ID ile Hemşire Ara:

Hemşire Ara

EKLE

SİL

GÜNCELLE

	id	ad	soyad	telefon	brans
▶	1	AYŞE	TEKİN	05587896985	Göz Hastalıkları
	2	NURİYE	CANDAN	05897462545	Psikiyatri
	3	ELİF	UYGUN	5423659508	Dahiliye
	4	ERKAN	DOĞA	5040145974	Ortopedi ve Travmatoloji
	5	AYŞE GÜL	GÜDER	5032147896	Deri ve Zührevi Hastalıklar
*					

DUYURULAR

DUYURU NO

BAŞLIK

AÇIKLAMA

EKLE

SİL

GÜNCELLE

	id	başlık	açıklama
▶	1	Covid-19	Koronavirüs tedbirleri kapsamında ...
	2	Yemek Listesi	29.12.2021 tarihli yemek listesi: T...
*			

MUAYENE

HASTA TC

DOKTOR ADI

MUAYENE ID

TEŞHİS

EKLE

SİL

GÜNCELLE

	hasta_tc	doktoradi	doktorsoyadi	muayeneid	teshis
▶	69287101233	MUSTAFA	ALDINÇ	1	Grip
	88442647458	NILGÜN	ÇELİK	2	Covid-19
	77201681510	TANJU	ÇELİKKAN	3	Guatr
*					

DİLEK-ŞİKAYET

ŞİKAYET NO

AD

SOYAD

BAŞLIK

ŞİKAYETİNİZ:

EKLE

SİL

GÜNCELLE

YATIŞ

ID

HASTA TC

ODA NO

YATILACAK GÜN

EKLE

SİL

GÜNCELLE

	hasta_tc	oda_no	yatilacak_gun	id
▶	77201681510	1	3	1
	88442647458	9	12	2
	67249922331	10	7	3
	69287101233	2	15	4
	95327413957	11	2	5
*				

ODA

ODA NO

MÜSAİTLİK DURUMU

EKLE

SİL

GÜNCELLE

BOŞ ODALAR

	odano	odadurumu	
	6	Boş	^
	7	Boş	
	8	Boş	
	12	Boş	
	13	Boş	
	14	Boş	v

DOLU ODALAR

	odano	odadurumu	
▶	1	Dolu	
	2	Dolu	
	9	Dolu	
	10	Dolu	
	11	Dolu	
*			v

BRANŞ

BRANŞ ADI :

BRANŞ NO :

EKLE

SİL

GÜNCELLE

	brans_adi	brans_numarasi	
▶	Dahiliye	1	
	Deri ve Zührevi Hastalıklar	2	
	Fizyoterapi	3	
	Göz Hastalıkları	4	
	Nöroloji	5	
	Kulak Burun Boğaz	6	v

REÇETE

REÇETE ID

DOKTOR ADI

HASTA TC

İLAÇ ADI

Reçeteye İlaç Ekle

EKLE

SİL

GÜNCELLE

	hasta_tc	doktoradi	doktorsoyadi	receteid	ilac_adi	kullanım_bilgi
▶	95327413957	SU	CANDANER	1	parol	günde 1 defa
	95327413957	SU	CANDANER	1	majezik	günde 2 defa
	67249922331	GÖKCAN	AŞAN	2	katarin	günde 1 kez
	67249922331	GÖKCAN	AŞAN	2	aspirin	günde 3 defa
	67249922331	GÖKCAN	AŞAN	2	aferin	günde 2 defa

UYGULAMANIN KAYNAK KODLARI:

<https://github.com/gurkan-kaya/DatabaseManagementSystemsProject>

VİDEO ADRESİ

<https://youtu.be/gacDe4ga4jY>