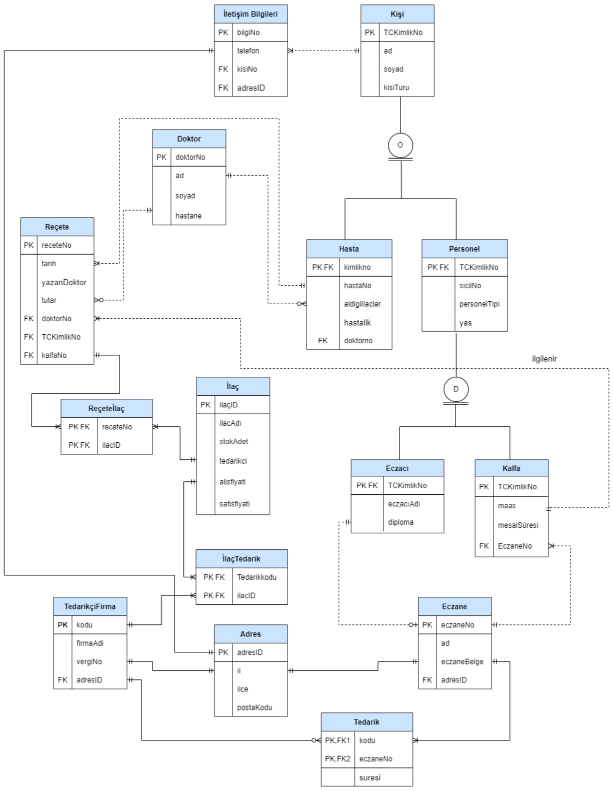


VERİTABANI YÖNETİM SİSTEMLERİ

KONU: Eczane veritabanı yönetimi

Esra kızılelma

B191210040 1A

SENERYO

Eczaneye gelen hastaların bilgilerini, ilaç stokları, depolanması ve yönetilmesi için bir veri tabanı tasarlanması isteniyor. Tasarlanan veri tabanında hasta bilgileri, ilaçlar, reçeteler, eczacılar, ilaçların tedarik edildiği firma gibi bilgilerin saklanması beklenmektedir.

İŞ KURALLARI

1. Bir eczacı yalnız bir eczane açabilir.
2. Bir eczanede birçok kalfa çalışır.
3. Eczanenin yalnız bir adresi olabilir.
4. İlaçlar ilaç firmasında tedarik edilmektedir.
5. Bir eczane en az bir ilaç firmasıyla çalışmak zorundadır, çok sayıda firma ile de çalışabilir.
6. Bir ilaç firması hiçbir eczane ile çalışmayabilir (tarım ilacı yapan firmalar), çok eczane ile de çalışabilir.
7. İlaç firmalarının kodu, adı, vergi dairesi, vergi numarası bilgileri bulunur.
8. İlaçların adı, ilaçID, stok miktarı, birimFiyatı, tedarikçi firma bilgileri saklanır.
9. Bu veri tabanında her hastaya ait bir numara verilerek hastanın TC kimlik numarası, adı, soyadı saklanılması düşünülüyor.
10. Bir reçeteye bakan bir kalfa mevcuttur. Bir reçetede en fazla bir kalfa bakabilir. Kalfaların TCKimlikNo, ad, soyadı bilgileri mevcuttur.
11. İlaçların kodu, adı, fiyatı, stok miktarı ve tedarik edildiği firma bilgilerinin saklanması gerekir.
12. Reçetelerin reçete numarası, tarih, tutar, reçeteyi yazan doktor bilgileri saklanmalıdır
13. Bir reçetede en az bir ilaç bulunabilir ancak çok sayıda ilaç da bulunabilir. Bir ilaç çok sayıda reçetede bulunabilir.
14. Bir hasta çok sayıda reçete alabilir fakat bir reçetenin yalnızca bir hastası olabilir. Bir reçetede çok sayıda ilaç olabilir.
15. Eczanede çalışan personel ya eczacı ya da kalfa olur ikisini birlikte olamaz.
16. Bir hasta sadece bir doktor tarafından muayene olabilir (aile hekimi).
17. Bir eczanede bir eczacı yönetebilir. (bir diplomayla bir eczane açılır)
18. Bir hastanın hiç reçetesi de olmayabilir doktor ilaç yazmamıştır.
19. İlaçlar reçetesiz alınamaz.

İLİŞKİSEL ŞEMA(METİNSEL GÖSTERİM)

-Iletişim Bilgileri (**bilgiNo:integer**, telefon:varchar, kisiNo:int, adresID:text)

-Kişi (**TCKimlikNo:bigint**, ad:varchar, soyad:varchar, kisiTürü:varchar)

-Hasta (**TCKimlikNo:bigint,** hastaNo:int, aldiğiIlaçlar:text, hastalık:text, doktorNo:int)

-Personel (**TCKimlikNo:bigint,** sicilNo:int, personelTipi:text, yas:int)

-Eczacı (**TCKimlikNo:bigint** eczacıAdı:varchar, diploma:text,)

-Kalfa (**TCKimlikNo:bigint** maas:int, mesaiÜcreti:int, eczaneNo:int)

-Eczane (**eczaneNo:char** ad:varchar, eczaneBelge:text, adresID:varchar)

-Doktor (**doktorNo:int,** ad:varchar, soyad:varchar, hastane:varchar)

-Recete (**receteNo:int**, tarih:date, yazanDoktor:text, tutar:modey, doktorNo:int, TCKimlikNo:bigint, kalfaNo:bigint)

-Ilaç (**İlaçID:char,** ilacAdi:varchar, skt:varchar, stokAdet:int, tedarikçi:varchar, birimFiyati: modey)

-IlacTedarik (**tedarikKodu:int, ilacID:int**)

-İlaçEczane (**İlaçID:int, receteNo:int**)

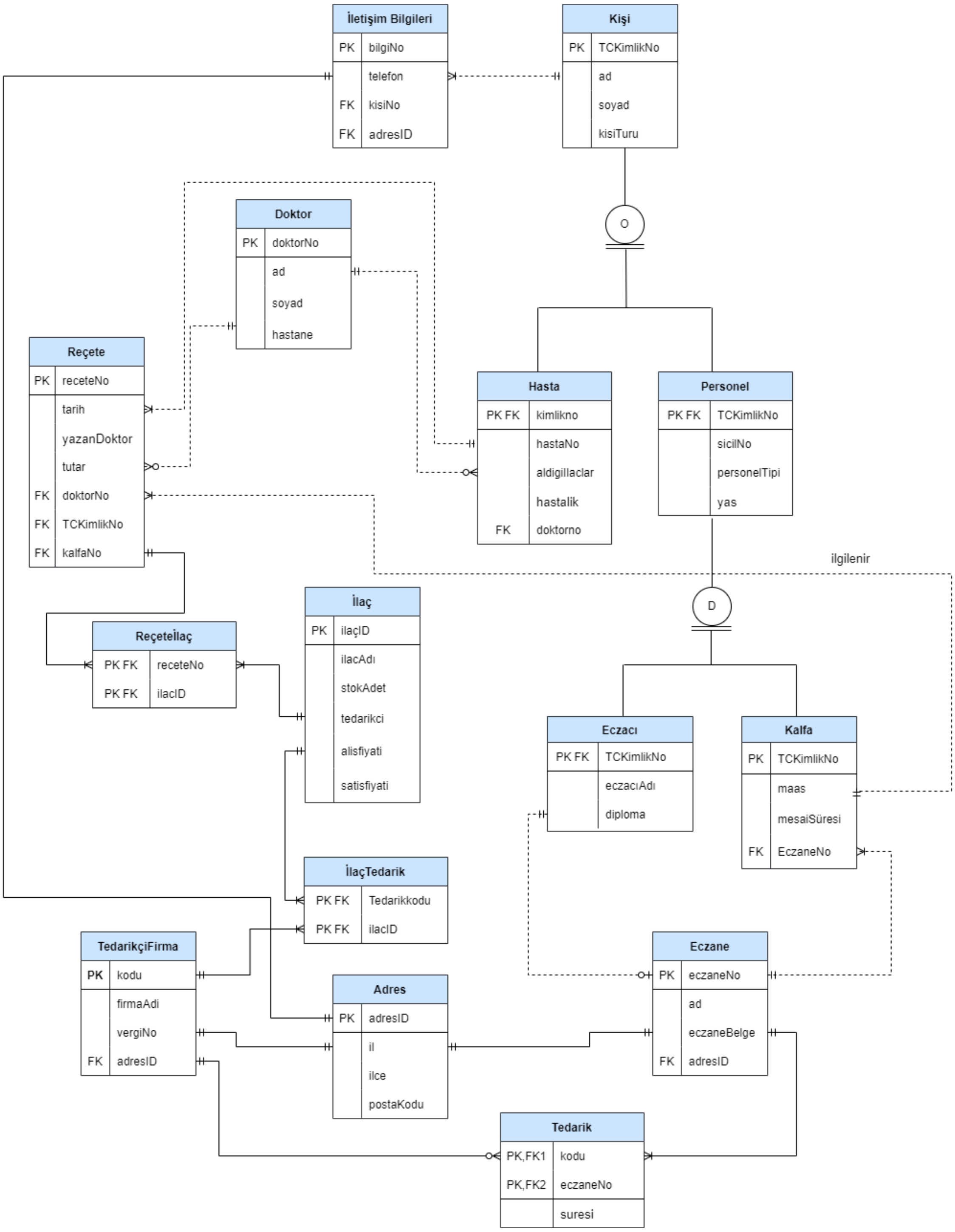
-Reçeteİlaç (**İlaçID:int, ilacID:int**)

-Tedarik (**kodu:int, eczaneNo:int,** suresi:time)

-Adres (**adresID:varchar**, il:varchar, ilçe:varchar, postaKodu:varchar)

-TedarikçiFirma (**kodu:int**, firmaAdı:text, vergiNo:int, adresID:varchar)

VARLIK BAĞINTI MODELİ



SQL KODLARI

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

-- Dumped from database version 14.1
-- Dumped by pg_dump version 14.0

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: public; Type: SCHEMA; Schema: -; Owner: postgres
--

CREATE SCHEMA public;


ALTER SCHEMA public OWNER TO postgres;

--
-- Name: SCHEMA public; Type: COMMENT; Schema: -; Owner: postgres
--

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

Fonksiyon

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

```
CREATE FUNCTION public.ilacara(idilac integer) RETURNS TABLE(iladid
integer, aliacadi character varying, skt character varying, stokadet integer, tedarikci
character varying, alisfiyati integer, satisfiyati integer)
    LANGUAGE plpgsql
    AS $$
begin
	return query select ilacid,ilacadi,skt,stokadet,tedarikci,alisfiyati,satisfiyati from
ilac
	where ilacid = idilac;
end;
$$;


ALTER FUNCTION public.ilacara(idilac integer) OWNER TO postgres;

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

CREATE FUNCTION public.ilacgetir() RETURNS TABLE(iladid integer, aliacadi
character varying, skt date, stokadet integer, tedarikci character varying, alisfiyati
integer, satisfiyati integer)
    LANGUAGE plpgsql
    AS $$
begin
return query select *from ilac;
end;
$$;


ALTER FUNCTION public.ilacgetir() OWNER TO postgres;

--
-- Name: indirim(integer, integer); Type: FUNCTION; Schema: public; Owner:
postgres
--

CREATE FUNCTION public.indirim(alisfiyati integer, satisfiyati integer) RETURNS
integer
    LANGUAGE plpgsql
    AS $$
declare
yuzde integer;
```

```
karoran integer;
kar integer;
begin
kar=satisfiyati-alisfiyati;
karoran=kar/alisfiyati;
yuzde=(karoran*100)/alisfiyati;
return yuzde;
end;
$$;


ALTER FUNCTION public.indirim(alisfiyati integer, satisfiyati integer) OWNER TO postgres;

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

Trigger

```
CREATE FUNCTION public.kapasite() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
begin
if((select count(*)from "ilac" where"stokadet" = new."stokadet")>999)
then
        raise exception 'bu ilacın stoğu dolu !!!';
        end if;
        return new;
end;
$$;


ALTER FUNCTION public.kapasite() OWNER TO postgres;

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

CREATE FUNCTION public.kayitrecete() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
```

```
declare
tarih date := current_date;
begin
insert into recete (receteno,tarih,yazandoktor,tutar,doktorno,kimlikno,kalfano)
values(new.receteno,tarih,new.yazandoktor,new.tutar,new.doktorno,new.kimlikno,new.kalfano);
	return new;
end;
$$;


ALTER FUNCTION public.kayitrecete() OWNER TO postgres;

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

CREATE FUNCTION public.kisicinsiyet() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
begin
if
new.cinsiyet = 'e' then
update kisicinsiyet set erkek =erkek +1;
else
update kisicinsiyet set kadin =kadin +1;
end if;
return new;
end;
$$;


ALTER FUNCTION public.kisicinsiyet() OWNER TO postgres;

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

CREATE FUNCTION public.tkisi() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
begin
update toplamkisi set sayi=sayi+1;
```

```
return new;
end;
$$;


ALTER FUNCTION public.tkisi() OWNER TO postgres;

--
-- Name: toplamkar(integer, integer, integer); Type: FUNCTION; Schema: public; Owner: postgres
--

CREATE FUNCTION public.toplamkar(alisfiyati integer, satisfiyati integer, stokadet integer) RETURNS integer
    LANGUAGE plpgsql
    AS $$
declare
toplamkar integer;
begin
toplamkar=(satisfiyati-alisfiyati)*stokadet;
return toplamkar;
end;
$$;


ALTER FUNCTION public.toplamkar(alisfiyati integer, satisfiyati integer, stokadet integer) OWNER TO postgres;

SET default_tablespace = '';

SET default_table_access_method = heap;

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

CREATE TABLE public.eczane (
    eczaneno integer NOT NULL,
    ad character varying(20),
    eczanebelge text NOT NULL,
    adresid character varying(10) NOT NULL
);
```

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

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

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


ALTER TABLE public."Eczane_eczaneNo_seq" OWNER TO postgres;

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

ALTER SEQUENCE public."Eczane_eczaneNo_seq" OWNED BY public.eczane.eczaneno;



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

CREATE TABLE public.hasta (
    kimlikno bigint NOT NULL,
    hastano integer NOT NULL,
    aldigiilaclar text,
    hastalik text,
    doktorno integer NOT NULL
);


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

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

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


ALTER TABLE public."Hasta_doktorNo_seq" OWNER TO postgres;

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

ALTER SEQUENCE public."Hasta_doktorNo_seq" OWNED BY
public.hasta.doktorno;



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

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


ALTER TABLE public."Hasta_hastaNo_seq" OWNER TO postgres;

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

```
ALTER SEQUENCE public."Hasta_hastaNo_seq" OWNED BY public.hasta.hastano;


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

CREATE TABLE public.ilac (
    ilacid integer NOT NULL,
    ilacadi character varying(20) NOT NULL,
    stokadet integer NOT NULL,
    tedarikci character varying(30),
    alisfiyati integer NOT NULL,
    satisfiyati integer,
    skt character varying DEFAULT CURRENT_DATE
);


ALTER TABLE public.ilac OWNER TO postgres;

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

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


ALTER TABLE public."Ilac_ilacID_seq" OWNER TO postgres;

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

ALTER SEQUENCE public."Ilac_ilacID_seq" OWNED BY public.ilac.ilacid;
```

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

CREATE TABLE public.kalfa (
    kimlikno bigint NOT NULL,
    maas integer,
    eczaneno integer NOT NULL,
    mesaisuressi real
);


ALTER TABLE public.kalfa OWNER TO postgres;

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

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


ALTER TABLE public."Kalfa_eczaneNo_seq" OWNER TO postgres;

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

ALTER SEQUENCE public."Kalfa_eczaneNo_seq" OWNED BY
public.kalfa.eczaneno;



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

```
CREATE TABLE public.personel (
    kimlikno bigint NOT NULL,
    sicilno integer NOT NULL,
    personeltipi text NOT NULL,
    yas integer NOT NULL
);


ALTER TABLE public.personel OWNER TO postgres;

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

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


ALTER TABLE public."Personel_sicilNo_seq" OWNER TO postgres;

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

ALTER SEQUENCE public."Personel_sicilNo_seq" OWNED BY public.personel.sicilno;



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

CREATE TABLE public.tedarik (
    kodu integer NOT NULL,
    eczaneno integer NOT NULL,
    suresi time(6) with time zone
);
```

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

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

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


ALTER TABLE public."Tedarik_eczaneNo_seq" OWNER TO postgres;

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

ALTER SEQUENCE public."Tedarik_eczaneNo_seq" OWNED BY
public.tedarik.eczaneno;



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

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


ALTER TABLE public."Tedarik_kodu_seq" OWNER TO postgres;
```

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

ALTER SEQUENCE public."Tedarik_kodu_seq" OWNED BY public.tedarik.kodu;


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

CREATE TABLE public.adres (
    adresid character varying(10) NOT NULL,
    il character varying(15),
    ilce character varying(20),
    postakodu character varying(5) NOT NULL
);


ALTER TABLE public.adres OWNER TO postgres;

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

CREATE TABLE public.cinsiyet (
    kadin text,
    erkek text
);


ALTER TABLE public.cinsiyet OWNER TO postgres;

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

CREATE TABLE public.doktor (
    doktorno integer NOT NULL,
    ad character varying(20) NOT NULL,
    soyad character varying(20) NOT NULL,
    hastane character varying(30)
);


ALTER TABLE public.doktor OWNER TO postgres;

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

CREATE TABLE public."eczacı" (
    kimlikno bigint NOT NULL,
    "eczacıadi" character varying(20) NOT NULL,
    diploma text NOT NULL
);


ALTER TABLE public."eczacı" OWNER TO postgres;

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

CREATE TABLE public.ilactedarik (
    tedarikkodu integer NOT NULL,
    ilacid integer NOT NULL
);


ALTER TABLE public.ilactedarik OWNER TO postgres;

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

CREATE TABLE public."iletişimbilgileri" (
    bilgino integer NOT NULL,
    telefon character varying(11) DEFAULT 0 NOT NULL,
    kisino integer,
    adresid text NOT NULL
);
```

```
ALTER TABLE public."iletişimbilgileri" OWNER TO postgres;

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

CREATE TABLE public.kapasite (
    toplam integer
);


ALTER TABLE public.kapasite OWNER TO postgres;

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

CREATE TABLE public.kisi (
    kimlikno bigint NOT NULL,
    ad character varying(20) NOT NULL,
    soyad character varying(20) NOT NULL,
    kisituru character varying(20) NOT NULL
);


ALTER TABLE public.kisi OWNER TO postgres;

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

CREATE TABLE public.recete (
    recerteno integer NOT NULL,
    tarih date,
    yazandoktor text,
    tutar money,
    doktorno integer NOT NULL,
    kimlikno bigint NOT NULL,
    kalfano bigint NOT NULL
);


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

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

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


ALTER TABLE public.recete_doktorno_seq OWNER TO postgres;

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

ALTER SEQUENCE public.recete_doktorno_seq OWNED BY
public.recete.doktorno;



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

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


ALTER TABLE public.recete_kalfano_seq OWNER TO postgres;

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

```
--

ALTER SEQUENCE public.recete_kalfano_seq OWNED BY public.recete.kalfano;


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

CREATE TABLE public.receteilac (
    receteno integer NOT NULL,
    "iladID" integer NOT NULL
);


ALTER TABLE public.receteilac OWNER TO postgres;

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

CREATE SEQUENCE public.sayac
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


ALTER TABLE public.sayac OWNER TO postgres;

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

CREATE TABLE public.tedarikcifirma (
    kodu integer NOT NULL,
    firmaadi text NOT NULL,
    vergino integer NOT NULL,
    adresid character varying(10) NOT NULL
);
```

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

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

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


ALTER TABLE public."tedarikciFirma_kodu_seq" OWNER TO postgres;

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

ALTER SEQUENCE public."tedarikciFirma_kodu_seq" OWNED BY
public.tedarikcifirma.kodu;


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

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


ALTER TABLE public."tedarikciFirma_vergiNo_seq" OWNER TO postgres;
```

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

ALTER SEQUENCE public."tedarikciFirma_vergiNo_seq" OWNED BY
public.tedarikcifirma.vergino;



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

CREATE TABLE public.toplamkisi (
    sayi integer
);


ALTER TABLE public.toplamkisi OWNER TO postgres;

--
-- Name: eczane eczaneno; Type: DEFAULT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.eczane ALTER COLUMN eczaneno SET DEFAULT
nextval('public."Eczane_eczaneNo_seq"'::regclass);


--
-- Name: hasta hastano; Type: DEFAULT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.hasta ALTER COLUMN hastano SET DEFAULT
nextval('public."Hasta_hastaNo_seq"'::regclass);


--
-- Name: hasta doktorno; Type: DEFAULT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.hasta ALTER COLUMN doktorno SET DEFAULT
nextval('public."Hasta_doktorNo_seq"'::regclass);
```

```
--
-- Name: ilac ilacid; Type: DEFAULT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.ilac ALTER COLUMN ilacid SET DEFAULT
nextval('public."Ilac_ilacID_seq"'::regclass);


--
-- Name: kalfa eczaneno; Type: DEFAULT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.kalfa ALTER COLUMN eczaneno SET DEFAULT
nextval('public."Kalfa_eczaneNo_seq"'::regclass);


--
-- Name: personel sicilno; Type: DEFAULT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.personel ALTER COLUMN sicilno SET DEFAULT
nextval('public."Personel_sicilNo_seq"'::regclass);


--
-- Name: recete doktorno; Type: DEFAULT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.recete ALTER COLUMN doktorno SET DEFAULT
nextval('public.recete_doktorno_seq'::regclass);


--
-- Name: recete kalfano; Type: DEFAULT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.recete ALTER COLUMN kalfano SET DEFAULT
nextval('public.recete_kalfano_seq'::regclass);


--
-- Name: tedarik kodu; Type: DEFAULT; Schema: public; Owner: postgres
```

```
--

ALTER TABLE ONLY public.tedarik ALTER COLUMN kodu SET DEFAULT
nextval('public."Tedarik_kodu_seq"'::regclass);


--
-- Name: tedarik eczaneno; Type: DEFAULT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.tedarik ALTER COLUMN eczaneno SET DEFAULT
nextval('public."Tedarik_eczaneNo_seq"'::regclass);


--
-- Name: tedarikcifirma kodu; Type: DEFAULT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.tedarikcifirma ALTER COLUMN kodu SET
DEFAULT nextval('public."tedarikciFirma_kodu_seq"'::regclass);


--
-- Name: tedarikcifirma vergino; Type: DEFAULT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.tedarikcifirma ALTER COLUMN vergino SET
DEFAULT nextval('public."tedarikciFirma_vergiNo_seq"'::regclass);


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

INSERT INTO public.adres VALUES
	('1', 'izmir', 'bornova', '35000'),
	('2', 'hatay', 'dörtyol', '31000'),
	('3', 'bilecik', 'merkez', '11000'),
	('4', 'sakarya', 'serdivan', '54050');


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

```
--



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

INSERT INTO public.doktor VALUES
	(1, 'esra', 'kızılelma', 'acıbadem'),
	(3, 'tuğba', 'mercan', 'ankara şehir hastanesi'),
	(2, 'şevval', 'sönmez', 'bilecik devlet hastanesi'),
	(4, 'yusuf', 'sönmez', 'adana devlet hastanesi');


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



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

INSERT INTO public.eczane VALUES
	(1, 'kızılelma', '1', '1');


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



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

INSERT INTO public.ilac VALUES
	(14, 'klomax', 120, 'deva', 56, 100, '2025.02.03'),
	(15, 'gripin', 89, 'bayer', 23, 89, '2024.05.06'),
	(10, 'parol', 123, 'eczacıbaşı', 12, 19, '2024.09.09'),
	(123456789, 'aksef', 99, 'bayer', 13, 33, '2023.06.07'),
	(456, 'metpamid', 4, 'abdiibrahim', 4, 45, '2022.01.01'),
	(45, 'emedur', 4, 'abdiibrahim', 5, 45, '2022.10.12'),
	(787894522, 'dolorex', 78, 'bayer', 35, 85, '2022.10.12'),
	(4, 'dolven', 3, 'deva', 9, 29, '2021.12.12'),
	(89, 'pedigen', 65, 'bayer', 12, 89, '2022.05.08'),
	(1, '11', 1, '1', 1, 1, '1');


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



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

INSERT INTO public."iletişimbilgileri" VALUES
	(1, '1', 1, '1');


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

INSERT INTO public.kalfa VALUES
	(3, 4500, 1, 40),
	(5, 4500, 1, 40);


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



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

```
INSERT INTO public.kisi VALUES
	(1, 'esra', 'kızılelma', 'hasta'),
	(2, 'şevval', 'kızılelma', 'eczacı'),
	(3, 'melike', 'sönmez', 'kalfa'),
	(4, 'tuğba', 'mercan', 'hasta'),
	(5, 'yusuf', 'kaya', 'kalfa'),
	(6, 'mustafa', 'turk', 'hasta'),
	(7, 'ayşe', 'kamer', 'hasta'),
	(8, 'kemal', 'yıldız', 'hasta');



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

INSERT INTO public.personel VALUES
	(3, 3, 'kalfa', 28),
	(2, 2, 'eczacı', 23),
	(5, 5, 'kalfa', 30);



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




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




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




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

```
INSERT INTO public.tedarikcifirma VALUES
	(1, 'bayer', 1, '1'),
	(2, 'eczacıbaşı', 2, '2'),
	(3, 'deva', 3, '3'),
	(4, 'abdiibrahim', 4, '4');


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

INSERT INTO public.toplamkisi VALUES
	(2),
	(8);


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

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


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

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


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

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


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

```
--

SELECT pg_catalog.setval('public."Ilac_ilacID_seq"', 1, true);



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

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



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

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



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

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



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

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



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

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

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

SELECT pg_catalog.setval('public.recete_kalfano_seq', 1, true);


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

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


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

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


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

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


--
-- Name: adres Adres_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.adres
    ADD CONSTRAINT "Adres_pkey" PRIMARY KEY (adresid);


--
```

```
-- Name: eczane Eczane_pkey; Type: CONSTRAINT; Schema: public; Owner:
postgres
--

ALTER TABLE ONLY public.eczane
    ADD CONSTRAINT "Eczane_pkey" PRIMARY KEY (eczaneno);


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

ALTER TABLE ONLY public.hasta
    ADD CONSTRAINT "Hasta_pkey" PRIMARY KEY (kimlikno);


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

ALTER TABLE ONLY public.ilac
    ADD CONSTRAINT "Ilac_pkey" PRIMARY KEY (ilacid);


--
-- Name: kalfa Kalfa_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.kalfa
    ADD CONSTRAINT "Kalfa_pkey" PRIMARY KEY (kimlikno);


--
-- Name: kisi Kisi_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.kisi
    ADD CONSTRAINT "Kisi_pkey" PRIMARY KEY (kimlikno);


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

```
--

ALTER TABLE ONLY public.personel
    ADD CONSTRAINT "Personel_pkey" PRIMARY KEY (kimlikno);


--
-- Name: receteilac ReceteIlac_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.receteilac
    ADD CONSTRAINT "ReceteIlac_pkey" PRIMARY KEY (receteno);


--
-- Name: tedarik Tedarik_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.tedarik
    ADD CONSTRAINT "Tedarik_pkey" PRIMARY KEY (kodu);


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

ALTER TABLE ONLY public.doktor
    ADD CONSTRAINT doktor_pkey PRIMARY KEY (doktorno);


--
-- Name: ilactedarik ilactedarik_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.ilactedarik
    ADD CONSTRAINT ilactedarik_pkey PRIMARY KEY (tedarikkodu);


--
```

```
-- Name: iletişimbilgileri iletişimbilgileri_pkey; Type: CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public."iletişimbilgileri"
   ADD CONSTRAINT "iletişimbilgileri_pkey" PRIMARY KEY (bilgino);



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

ALTER TABLE ONLY public.recete
   ADD CONSTRAINT recete_pkey PRIMARY KEY (recerteno);



--
-- Name: tedarikcifirma tedarikciFirma_pkey; Type: CONSTRAINT; Schema: public;
Owner: postgres
--

ALTER TABLE ONLY public.tedarikcifirma
   ADD CONSTRAINT "tedarikciFirma_pkey" PRIMARY KEY (kodu);



--
-- Name: fki_eczacı_personel; Type: INDEX; Schema: public; Owner: postgres
--

CREATE INDEX "fki_eczacı_personel" ON public."eczacı" USING btree (kimlikno);



--
-- Name: kisi kapasiyetrig; Type: TRIGGER; Schema: public; Owner: postgres
--

CREATE TRIGGER kapasiyetrig AFTER INSERT ON public.kisi FOR EACH ROW
EXECUTE FUNCTION public.kapasite();



--
-- Name: hasta kisicinsiyet; Type: TRIGGER; Schema: public; Owner: postgres
--
```

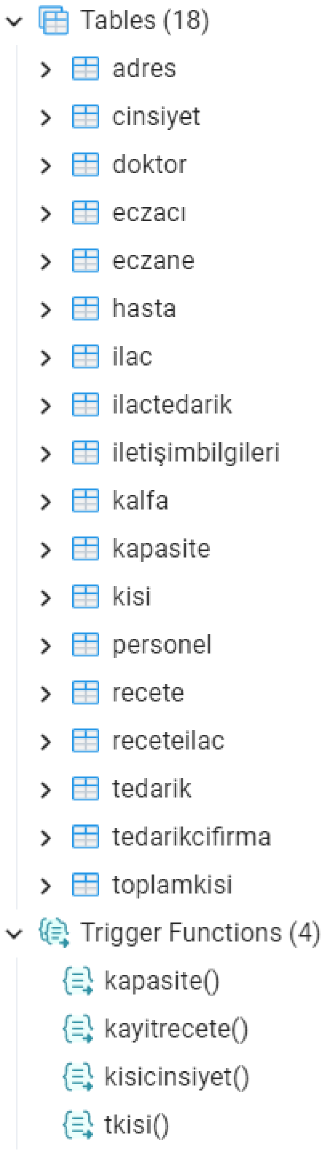
```
CREATE TRIGGER kisicinsiyet AFTER INSERT ON public.hasta FOR EACH
ROW EXECUTE FUNCTION public.kisicinsiyet();


--
-- Name: kisi tkisitrig; Type: TRIGGER; Schema: public; Owner: postgres
--

CREATE TRIGGER tkisitrig AFTER INSERT ON public.kisi FOR EACH ROW
EXECUTE FUNCTION public.tkisi();


--
-- Name: eczane adresFK; Type: FK CONSTRAINT; Schema: public; Owner:
postgres
--

ALTER TABLE ONLY public.eczane
   ADD CONSTRAINT "adresFK" FOREIGN KEY (adresid) REFERENCES
public.adres(adresid);


--
-- Name: eczacı eczacı_personel_fk; Type: FK CONSTRAINT; Schema: public;
Owner: postgres
--

ALTER TABLE ONLY public."eczacı"
   ADD CONSTRAINT "eczacı_personel_fk" FOREIGN KEY (kimlikno)
REFERENCES public.personel(kimlikno) ON UPDATE CASCADE ON DELETE
CASCADE NOT VALID;


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

ALTER TABLE ONLY public.hasta
   ADD CONSTRAINT hasta_doktor_fk FOREIGN KEY (doktorno) REFERENCES
public.doktor(doktorno) NOT VALID;
```

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

ALTER TABLE ONLY public.hasta
   ADD CONSTRAINT hasta_kisi_fk FOREIGN KEY (kimlikno) REFERENCES
public.kisi(kimlikno) ON UPDATE CASCADE ON DELETE CASCADE NOT
VALID;



--
-- Name: ilactedarik ilactedarik_ilac_fkpk; Type: FK CONSTRAINT; Schema: public;
Owner: postgres
--

ALTER TABLE ONLY public.ilactedarik
   ADD CONSTRAINT ilactedarik_ilac_fkpk FOREIGN KEY (ilacid)
REFERENCES public.ilac(ilacid);



--
-- Name: ilactedarik ilactedarik_tedarikcifirma_fkpk; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.ilactedarik
   ADD CONSTRAINT ilactedarik_tedarikcifirma_fkpk FOREIGN KEY
(tedarikkodu) REFERENCES public.tedarikcifirma(kodu);



--
-- Name: iletişimbilgileri iletisimbilgileri_adres_fk; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."iletişimbilgileri"
   ADD CONSTRAINT iletisimbilgileri_adres_fk FOREIGN KEY (kisino)
REFERENCES public.kisi(kimlikno);



--
```

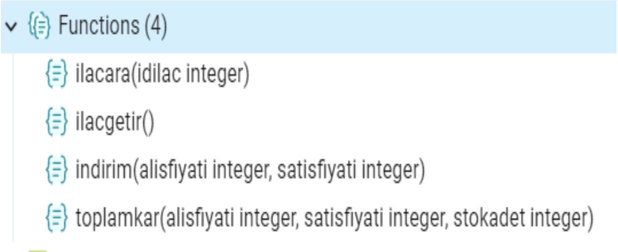
```
-- Name: iletişimbilgileri iletisimbilgileri_kisi_fk; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public."iletişimbilgileri"
   ADD CONSTRAINT iletisimbilgileri_kisi_fk FOREIGN KEY (adresid)
REFERENCES public.adres(adresid) NOT VALID;



--
-- Name: kalfa kalfa_eczane_fk; Type: FK CONSTRAINT; Schema: public; Owner:
postgres
--

ALTER TABLE ONLY public.kalfa
   ADD CONSTRAINT kalfa_eczane_fk FOREIGN KEY (eczaneno) REFERENCES
public.eczane(eczaneno) NOT VALID;



--
-- Name: kalfa kalfa_personel_fk; Type: FK CONSTRAINT; Schema: public; Owner:
postgres
--

ALTER TABLE ONLY public.kalfa
   ADD CONSTRAINT kalfa_personel_fk FOREIGN KEY (kimlikno)
REFERENCES public.personel(kimlikno) ON UPDATE CASCADE ON DELETE
CASCADE NOT VALID;



--
-- Name: personel personel_fkpk; Type: FK CONSTRAINT; Schema: public; Owner:
postgres
--

ALTER TABLE ONLY public.personel
   ADD CONSTRAINT personel_fkpk FOREIGN KEY (kimlikno) REFERENCES
public.kisi(kimlikno) ON UPDATE CASCADE ON DELETE CASCADE NOT
VALID;



--
```

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

ALTER TABLE ONLY public.recete
    ADD CONSTRAINT recete_doktor_fk FOREIGN KEY (doktorno) REFERENCES
public.doktor(doktorno);



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

ALTER TABLE ONLY public.recete
    ADD CONSTRAINT recete_hasta_fk FOREIGN KEY (kimlikno) REFERENCES
public.hasta(kimlikno);



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

ALTER TABLE ONLY public.recete
    ADD CONSTRAINT recete_kalfa_fk FOREIGN KEY (kalfano) REFERENCES
public.kalfa(kimlikno);



--
-- Name: receteilac receteilac_ilac_fk; Type: FK CONSTRAINT; Schema: public;
Owner: postgres
--

ALTER TABLE ONLY public.receteilac
    ADD CONSTRAINT receteilac_ilac_fk FOREIGN KEY ("iladID")
REFERENCES public.ilac(ilacid);



--
-- Name: receteilac receteilac_recete_fk; Type: FK CONSTRAINT; Schema: public;
Owner: postgres
--
```

```
ALTER TABLE ONLY public.receteilac
   ADD CONSTRAINT receteilac_recete_fk FOREIGN KEY (receteno)
REFERENCES public.recete(recerteno);



--
-- Name: tedarik tedarik_eczanePKFK; Type: FK CONSTRAINT; Schema: public;
Owner: postgres
--

ALTER TABLE ONLY public.tedarik
   ADD CONSTRAINT "tedarik_eczanePKFK" FOREIGN KEY (eczaneno)
REFERENCES public.eczane(eczaneno);



--
-- Name: tedarik tedarik_firmaPKFK; Type: FK CONSTRAINT; Schema: public;
Owner: postgres
--

ALTER TABLE ONLY public.tedarik
   ADD CONSTRAINT "tedarik_firmaPKFK" FOREIGN KEY (kodu)
REFERENCES public.tedarikcifirma(kodu);



--
-- Name: tedarikcifirma tedarikciFirmaFK; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public.tedarikcifirma
   ADD CONSTRAINT "tedarikciFirmaFK" FOREIGN KEY (adresid)
REFERENCES public.adres(adresid);



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

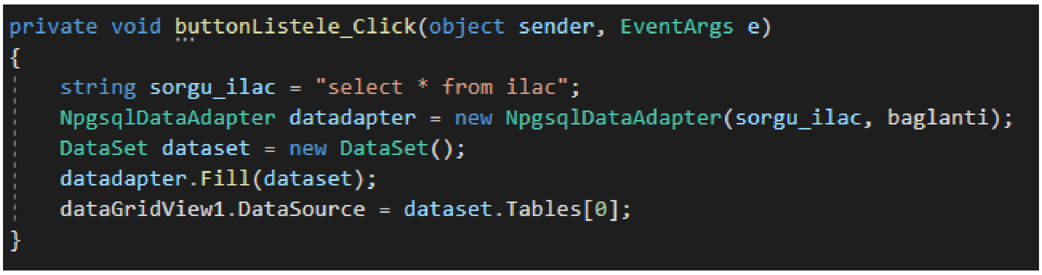


```
Tables (18)
  adres
  cinsiyet
  doktor
  eczacı
  eczane
  hasta
  ilac
  ilactedarik
  iletişimbilgileri
  kalfa
  kapasite
  kisi
  personel
  recete
  receteilac
  tedarik
  tedarikcifirma
  toplamkisi
Trigger Functions (4)
  kapasite()
  kayitrecete()
  kisicinsiyet()
  tkisi()
```



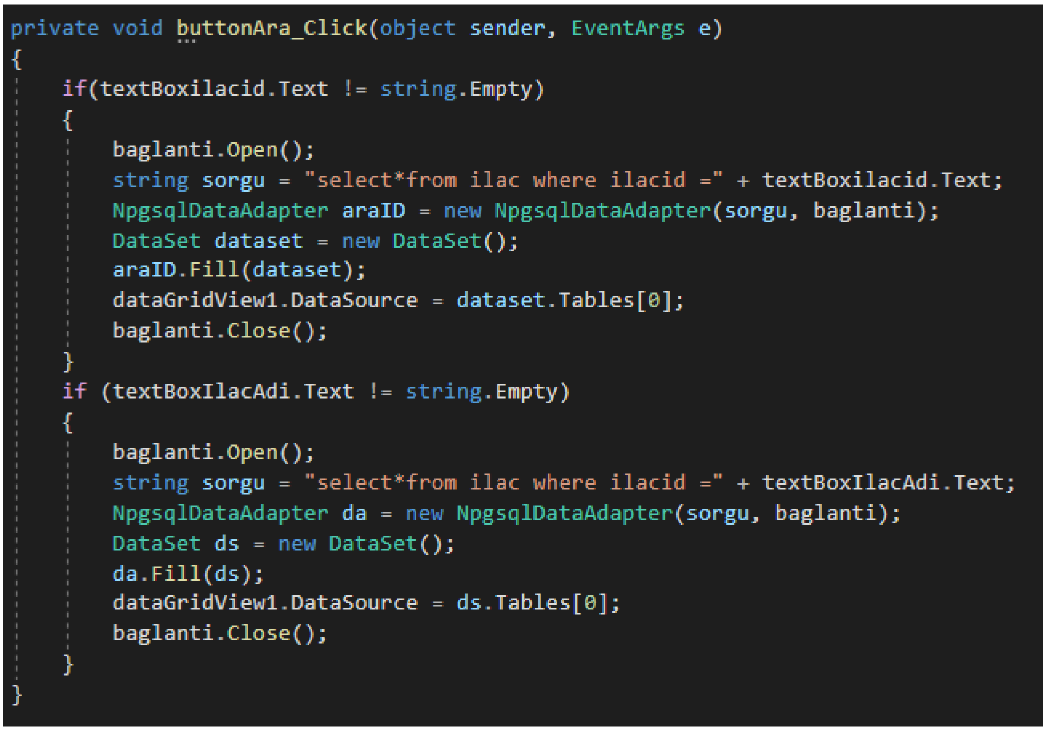
```
Functions (4)
  ilacara(idilac integer)
  ilacgetir()
  indirim(alisfiyati integer, satisfiyati integer)
  toplamkar(alisfiyati integer, satisfiyati integer, stokadet integer)
```

Listeleme:



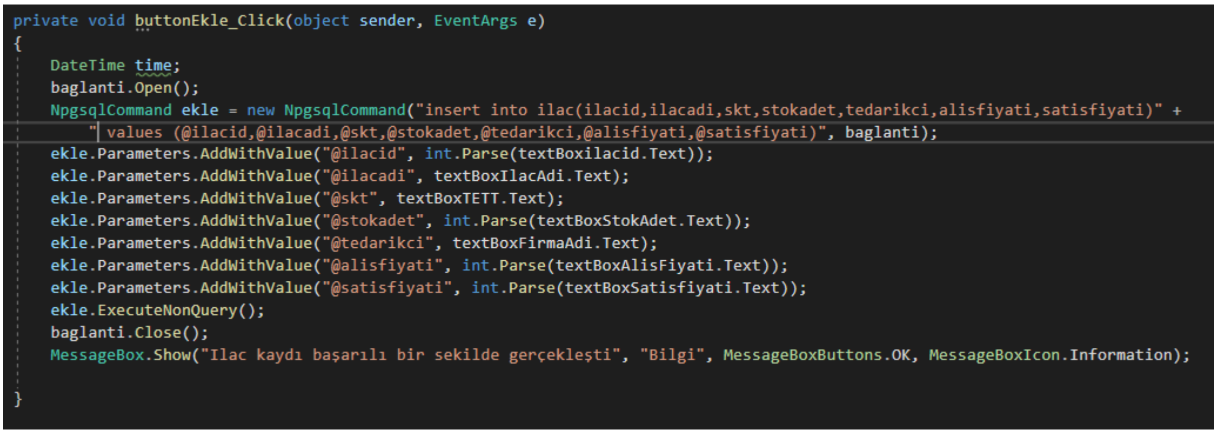
```
private void buttonListele_Click(object sender, EventArgs e)
{
    string sorgu_ilac = "select * from ilac";
    NpgsqlDataAdapter datadapter = new NpgsqlDataAdapter(sorgu_ilac, baglanti);
    DataSet dataset = new DataSet();
    datadapter.Fill(dataset);
    dataGridView1.DataSource = dataset.Tables[0];
}
```

Arama:



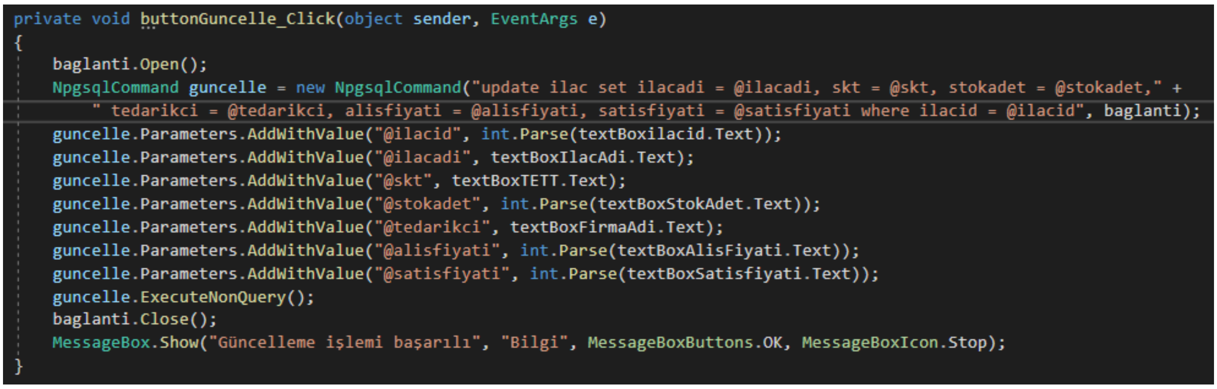
```
private void buttonAra_Click(object sender, EventArgs e)
{
    if(textBoxilacid.Text != string.Empty)
    {
        baglanti.Open();
        string sorgu = "select*from ilac where ilacid =" + textBoxilacid.Text;
        NpgsqlDataAdapter araID = new NpgsqlDataAdapter(sorgu, baglanti);
        DataSet dataset = new DataSet();
        araID.Fill(dataset);
        dataGridView1.DataSource = dataset.Tables[0];
        baglanti.Close();
    }
    if (textBoxIlacAdi.Text != string.Empty)
    {
        baglanti.Open();
        string sorgu = "select*from ilac where ilacid =" + textBoxIlacAdi.Text;
        NpgsqlDataAdapter da = new NpgsqlDataAdapter(sorgu, baglanti);
        DataSet ds = new DataSet();
        da.Fill(ds);
        dataGridView1.DataSource = ds.Tables[0];
        baglanti.Close();
    }
}
```

Ekleme:



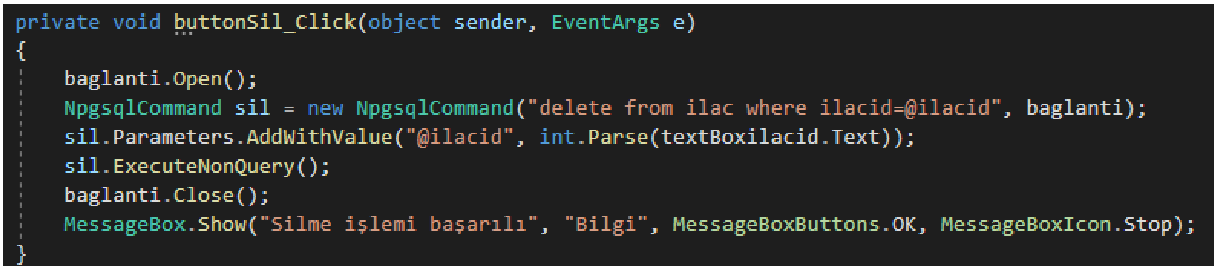
```
private void buttonEkle_Click(object sender, EventArgs e)
{
    DateTime time;
    baglanti.Open();
    NpgsqlCommand ekle = new NpgsqlCommand("insert into ilac(ilacid,ilacadi,skt,stokadet,tedarikci,alisfiyati,satisfiyati)" +
        " values (@ilacid,@ilacadi,@skt,@stokadet,@tedarikci,@alisfiyati,@satisfiyati)", baglanti);
    ekle.Parameters.AddWithValue("@ilacid", int.Parse(textBoxilacid.Text));
    ekle.Parameters.AddWithValue("@ilacadi", textBoxIlacAdi.Text);
    ekle.Parameters.AddWithValue("@skt", textBoxTETT.Text);
    ekle.Parameters.AddWithValue("@stokadet", int.Parse(textBoxStokAdet.Text));
    ekle.Parameters.AddWithValue("@tedarikci", textBoxFirmaAdi.Text);
    ekle.Parameters.AddWithValue("@alisfiyati", int.Parse(textBoxAlisFiyati.Text));
    ekle.Parameters.AddWithValue("@satisfiyati", int.Parse(textBoxSatisfiyati.Text));
    ekle.ExecuteNonQuery();
    baglanti.Close();
    MessageBox.Show("Ilac kaydı başarılı bir sekilde gerçekleşti", "Bilgi", MessageBoxButtons.OK, MessageBoxIcon.Information);

}
```

Güncelleme:



```
private void buttonGuncelle_Click(object sender, EventArgs e)
{
    baglanti.Open();
    NpgsqlCommand guncelle = new NpgsqlCommand("update ilac set ilacadi = @ilacadi, skt = @skt, stokadet = @stokadet," +
        " tedarikci = @tedarikci, alisfiyati = @alisfiyati, satisfiyati = @satisfiyati where ilacid = @ilacid", baglanti);
    guncelle.Parameters.AddWithValue("@ilacid", int.Parse(textBoxilacid.Text));
    guncelle.Parameters.AddWithValue("@ilacadi", textBoxIlacAdi.Text);
    guncelle.Parameters.AddWithValue("@skt", textBoxTETT.Text);
    guncelle.Parameters.AddWithValue("@stokadet", int.Parse(textBoxStokAdet.Text));
    guncelle.Parameters.AddWithValue("@tedarikci", textBoxFirmaAdi.Text);
    guncelle.Parameters.AddWithValue("@alisfiyati", int.Parse(textBoxAlisFiyati.Text));
    guncelle.Parameters.AddWithValue("@satisfiyati", int.Parse(textBoxSatisfiyati.Text));
    guncelle.ExecuteNonQuery();
    baglanti.Close();
    MessageBox.Show("Güncelleme işlemi başarılı", "Bilgi", MessageBoxButtons.OK, MessageBoxIcon.Stop);
}
```

Silme



```
private void buttonSil_Click(object sender, EventArgs e)
{
    baglanti.Open();
    NpgsqlCommand sil = new NpgsqlCommand("delete from ilac where ilacid=@ilacid", baglanti);
    sil.Parameters.AddWithValue("@ilacid", int.Parse(textBoxilacid.Text));
    sil.ExecuteNonQuery();
    baglanti.Close();
    MessageBox.Show("Silme işlemi başarılı", "Bilgi", MessageBoxButtons.OK, MessageBoxIcon.Stop);
}
```

Github

https://github.com/EsraKizilelma/Veritabani-Yonetim-Sistemleri-Proje