**T.C.**

**SAKARYA ÜNİVERSİTESİ**

**BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**Ders:** Veri Tabanı Yönetim Sistemleri

**Ödev:** Dönem Sonu Proje Ödevi

**Dönem:** 2020-2021 GÜZ Dönemi

**Adı Soyadı:** Arslancan Sarıkaya

**Okul No:** B181210052

**E-mail:** arslancan.sarikaya@ogr.sakarya.edu.tr

**Konu:** Bir Bilgisayar Şirketinin Çalışan ve Stok Otomasyonu

**Uygulama Tanıtımı:**

Bilgisayar donanımları satan bir firmanın ürün, çalışan, depo, mağaza kayıtlarını tutan bir program geliştirdim. Projede istenen bütün gereklilikleri yerine getirdim.

Projede 13 fonksiyon, 4 Gizli yordam , 4 Tetikleyici kullandım.

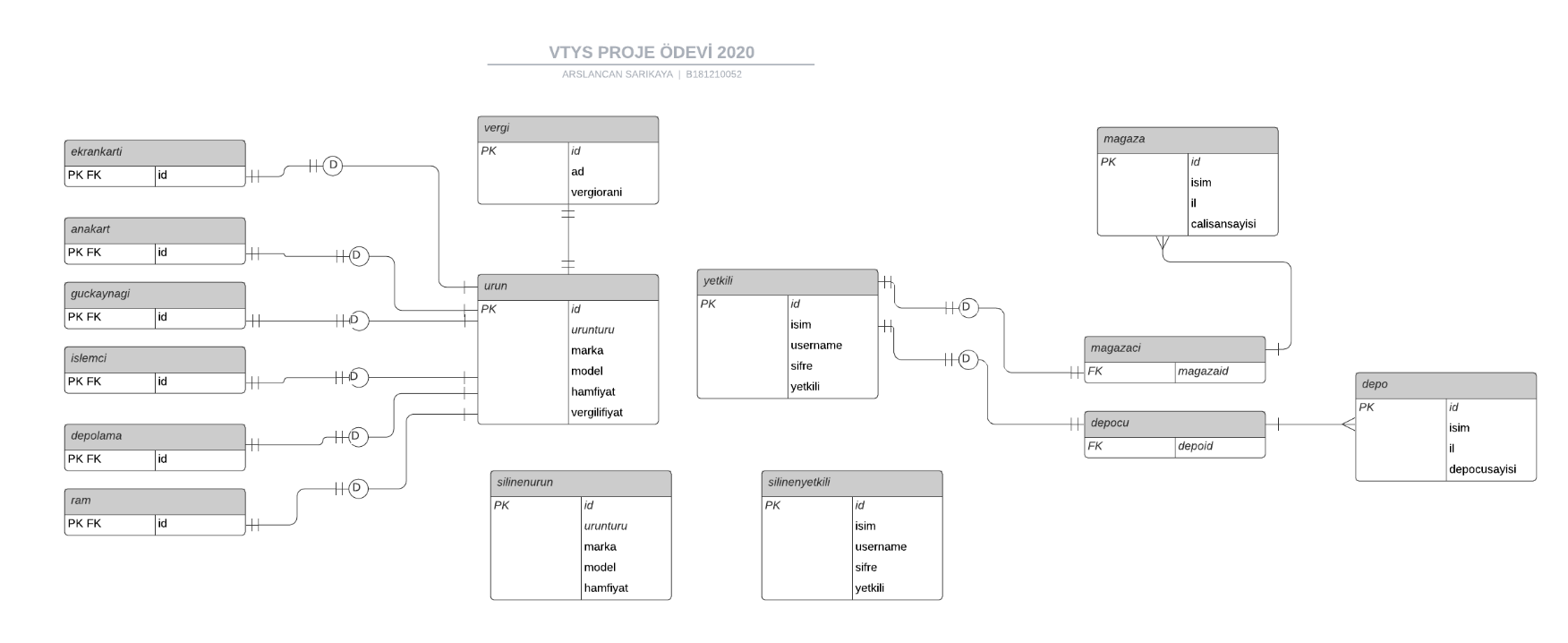
Programı geliştirirken .NET Framework ve postgresql’den yararlandım.

**İş Kuralları:**

* Üyelerin uygulamaya giriş yapabilmesi için kayıtlarının olması ve yetkilendirilmiş olması gerekir.
* Standart girişi hem müşteriler hem de çalışanlar kullanabilir hesap bilgileri girmeye gerek yoktur.
* Her bir ürünün kendine ait eşsiz bir id numarası vardır.
* Girilen ürünler ilk önce ürün tablosuna daha sonra ürün tiplerine göre tablolarına kaydedilirler.
* Bir mağazacı, 1 mağazada çalışabilir.
* Bir depocu, 1 depoda çalışabilir.
* Bir mağazanın birden fazla çalışanı olabilir.
* Bir deponun birden fazla çalışanı olabilir.
* Silinen ürünlerin kaydı ‘silinenurun’ tablosunda tutulur.
* Silinen çalışanların kaydı ‘silinenyetkili’ tablosunda tutulur.
* Ürün numaraları sistem tarafından atanır.
* Bir mağazacı iş yeri değiştirirse kaydının baştan yapılması gerekir.
* Bir depocu iş yeri değiştirirse kaydının baştan yapılması gerekir.
* İş yerlerinde çalışan sayısını program yeni kayıt alındıkça otomatik atar.

**İlişkisel Şema:**

* urun (id: bigserial, urunturu: character varying,marka: character varying, model: character varying, hamfiyat: bigint, vergilifiyat: bigint )
* ekrankarti (id: bigint)
* anakart (id: bigint)
* guckaynagı (id: bigint)
* işlemci (id: bigint)
* depolama (id: bigint)
* ram (id: bigint)
* yetkili(id: bigserial, isim: character varying, username: character varying, sifre: character varying, yetkili: boolean )
* magazaci (magazaid: bigint)
* depocu (magazaid: bigint)
* magaza (id: bigserial, isim: character varying, il: character varying, calisansayisi: bigint )
* depo (id: bigserial, isim: character varying, il: character varying, depocusayisi: bigint)
* silinenyetkili(id: bigserial, urunturu: character varying,marka: character varying, model: character varying, hamfiyat: bigint, vergilifiyat: bigint )
* silinenurun
* vergi (id: bigserial, urunturu: character varying,marka: character varying, model: character varying, hamfiyat: bigint)

**Varlık Bağıntı Modeli:**

**POSTRESQL KODLARI:**

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: vtysproje; Type: DATABASE; Schema: -; Owner: postgres

--

CREATE DATABASE vtysproje WITH TEMPLATE = template0 ENCODING = 'UTF8' LOCALE = 'Turkish\_Turkey.1254';

ALTER DATABASE vtysproje OWNER TO postgres;

\connect vtysproje

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: depo\_ekle(character varying, character varying); Type: PROCEDURE; Schema: public; Owner: postgres

--

CREATE PROCEDURE public.depo\_ekle(girisim character varying, giril character varying)

LANGUAGE sql

AS $$

insert into depo (isim,il,depocusayisi) values (girisim,giril,0);

$$;

ALTER PROCEDURE public.depo\_ekle(girisim character varying, giril character varying) OWNER TO postgres;

--

-- Name: depo\_sil(bigint); Type: PROCEDURE; Schema: public; Owner: postgres

--

CREATE PROCEDURE public.depo\_sil(\_girid bigint)

LANGUAGE sql

AS $$

delete from depo where id=\_girid

$$;

ALTER PROCEDURE public.depo\_sil(\_girid bigint) OWNER TO postgres;

--

-- Name: depocu\_login(character varying, character varying); Type: FUNCTION; Schema: public; Owner: postgres

--

CREATE FUNCTION public.depocu\_login(\_username character varying, \_password character varying) RETURNS integer

LANGUAGE plpgsql

AS $$

begin

if (select count(\*) from depocu where username = \_username and sifre = \_password and yetkili='t') >0 then

return 1;

else

return 0;

end if;

end;

$$;

ALTER FUNCTION public.depocu\_login(\_username character varying, \_password character varying) OWNER TO postgres;

--

-- Name: depocuekle(character varying, character varying, character varying, boolean, bigint); Type: FUNCTION; Schema: public; Owner: postgres

--

CREATE FUNCTION public.depocuekle(\_isim character varying, \_username character varying, \_sifre character varying, \_yetkili boolean, \_depoid bigint) RETURNS void

LANGUAGE plpgsql

AS $$

DECLARE

BEGIN

insert into depocu (isim,username,sifre,yetkili,depoid) values (\_isim,\_username,\_sifre,\_yetkili,\_depoid);

END

$$;

ALTER FUNCTION public.depocuekle(\_isim character varying, \_username character varying, \_sifre character varying, \_yetkili boolean, \_depoid bigint) OWNER TO postgres;

--

-- Name: depocuguncelle(bigint, character varying, character varying, character varying, boolean); Type: FUNCTION; Schema: public; Owner: postgres

--

CREATE FUNCTION public.depocuguncelle(\_girid bigint, \_isim character varying, \_username character varying, \_sifre character varying, \_yetkili boolean) RETURNS void

LANGUAGE plpgsql

AS $$

DECLARE

BEGIN

update depocu set isim=\_isim where id = \_girid;

update depocu set username=\_username where id= \_girid;

update depocu set sifre=\_sifre where id= \_girid;

update depocu set yetkili=\_yetkili where id= \_girid;

END

$$;

ALTER FUNCTION public.depocuguncelle(\_girid bigint, \_isim character varying, \_username character varying, \_sifre character varying, \_yetkili boolean) OWNER TO postgres;

--

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

--

CREATE FUNCTION public.depocusil(\_girid bigint) RETURNS void

LANGUAGE plpgsql

AS $$

DECLARE

\_isim character varying;

\_username character varying;

\_sifre character varying;

\_depoid bigint;

BEGIN

\_isim=(select isim from depocu where id =\_girid);

\_username = (select username from depocu where id =\_girid);

\_sifre =(select sifre from depocu where id =\_girid);

\_depoid = (select depoid from depocu where id =\_girid);

insert into silinenyetkili (isim,username,sifre,magazaid) values (\_isim,\_username,\_sifre,\_depoid);

delete from depocu where id=\_girid;

END

$$;

ALTER FUNCTION public.depocusil(\_girid bigint) OWNER TO postgres;

--

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

--

CREATE FUNCTION public.depoelemansayisiarttir() RETURNS trigger

LANGUAGE plpgsql

AS $$

declare

son bigint;

begin

son=currval('"public"."yetkili\_id\_seq"'::regclass);

update depo set depocusayisi = depocusayisi+1 where id =(select depoid from depocu where id=son) ;

return new;

end;

$$;

ALTER FUNCTION public.depoelemansayisiarttir() OWNER TO postgres;

--

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

--

CREATE FUNCTION public.depoelemansayisiazalt() RETURNS trigger

LANGUAGE plpgsql

AS $$

declare

son bigint;

begin

son=currval('"public"."silinenyetkili\_id\_seq"'::regclass);

update depo set depocusayisi = depocusayisi-1 where id =(select magazaid from silinenyetkili where id=son) ;

return new;

end;

$$;

ALTER FUNCTION public.depoelemansayisiazalt() OWNER TO postgres;

--

-- Name: depoguncelle(bigint, character varying, character varying); Type: FUNCTION; Schema: public; Owner: postgres

--

CREATE FUNCTION public.depoguncelle(\_girid bigint, \_isim character varying, \_il character varying) RETURNS void

LANGUAGE plpgsql

AS $$

DECLARE

BEGIN

update depo set isim=\_isim where id=\_girid;

update depo set il=\_il where id=\_girid;

END

$$;

ALTER FUNCTION public.depoguncelle(\_girid bigint, \_isim character varying, \_il character varying) OWNER TO postgres;

--

-- Name: kdvhesapla(bigint, bigint); Type: FUNCTION; Schema: public; Owner: postgres

--

CREATE FUNCTION public.kdvhesapla(hamfiyat bigint, vergiid bigint) RETURNS bigint

LANGUAGE plpgsql

AS $$

DECLARE

sonfiyat BIGINT;

vergiorani float;

BEGIN

vergiorani = (SELECT vergi.vergiorani FROM vergi where id =vergiid);

sonfiyat = (1+vergiorani)\* hamfiyat;

return sonfiyat;

END;

$$;

ALTER FUNCTION public.kdvhesapla(hamfiyat bigint, vergiid bigint) OWNER TO postgres;

--

-- Name: magaza\_ekle(character varying, character varying); Type: PROCEDURE; Schema: public; Owner: postgres

--

CREATE PROCEDURE public.magaza\_ekle(girisim character varying, giril character varying)

LANGUAGE sql

AS $$

insert into magaza (isim,il,calisansayisi) values (girisim,giril,0);

$$;

ALTER PROCEDURE public.magaza\_ekle(girisim character varying, giril character varying) OWNER TO postgres;

--

-- Name: magaza\_sil(bigint); Type: PROCEDURE; Schema: public; Owner: postgres

--

CREATE PROCEDURE public.magaza\_sil(girid bigint)

LANGUAGE sql

AS $$

delete from magaza where id=girid

$$;

ALTER PROCEDURE public.magaza\_sil(girid bigint) OWNER TO postgres;

--

-- Name: magazaci\_login(character varying, character varying); Type: FUNCTION; Schema: public; Owner: postgres

--

CREATE FUNCTION public.magazaci\_login(\_username character varying, \_password character varying) RETURNS integer

LANGUAGE plpgsql

AS $$

begin

if (select count(\*) from magazaci where username = \_username and sifre = \_password and yetkili='t') >0 then

return 1;

else

return 0;

end if;

end;

$$;

ALTER FUNCTION public.magazaci\_login(\_username character varying, \_password character varying) OWNER TO postgres;

--

-- Name: magazaciekle(character varying, character varying, character varying, boolean, bigint); Type: FUNCTION; Schema: public; Owner: postgres

--

CREATE FUNCTION public.magazaciekle(\_isim character varying, \_username character varying, \_sifre character varying, \_yetkili boolean, \_magazaid bigint) RETURNS void

LANGUAGE plpgsql

AS $$

DECLARE

BEGIN

insert into magazaci (isim,username,sifre,yetkili,magazaid) values (\_isim,\_username,\_sifre,\_yetkili,\_magazaid);

END

$$;

ALTER FUNCTION public.magazaciekle(\_isim character varying, \_username character varying, \_sifre character varying, \_yetkili boolean, \_magazaid bigint) OWNER TO postgres;

--

-- Name: magazaciguncelle(bigint, character varying, character varying, character varying, boolean); Type: FUNCTION; Schema: public; Owner: postgres

--

CREATE FUNCTION public.magazaciguncelle(\_girid bigint, \_isim character varying, \_username character varying, \_sifre character varying, \_yetkili boolean) RETURNS void

LANGUAGE plpgsql

AS $$

DECLARE

BEGIN

update magazaci set isim=\_isim where id= \_girid;

update magazaci set username=\_username where id= \_girid;

update magazaci set sifre=\_sifre where id= \_girid;

update magazaci set yetkili=\_yetkili where id= \_girid;

END

$$;

ALTER FUNCTION public.magazaciguncelle(\_girid bigint, \_isim character varying, \_username character varying, \_sifre character varying, \_yetkili boolean) OWNER TO postgres;

--

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

--

CREATE FUNCTION public.magazacisil(\_girid bigint) RETURNS void

LANGUAGE plpgsql

AS $$

DECLARE

\_isim character varying;

\_username character varying;

\_sifre character varying;

\_magazaid bigint;

BEGIN

\_isim=(select isim from magazaci where id =\_girid);

\_username = (select username from magazaci where id =\_girid);

\_sifre =(select sifre from magazaci where id =\_girid);

\_magazaid = (select magazaid from magazaci where id =\_girid);

insert into silinenyetkili (isim,username,sifre,magazaid) values (\_isim,\_username,\_sifre,\_magazaid);

delete from magazaci where id=\_girid;

END

$$;

ALTER FUNCTION public.magazacisil(\_girid bigint) OWNER TO postgres;

--

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

--

CREATE FUNCTION public.magazaelemansayisiarttir() RETURNS trigger

LANGUAGE plpgsql

AS $$

declare

son bigint;

begin

son=currval('"public"."yetkili\_id\_seq"'::regclass);

update magaza set calisansayisi = calisansayisi+1 where id =(select magazaid from magazaci where id=son) ;

return new;

end;

$$;

ALTER FUNCTION public.magazaelemansayisiarttir() OWNER TO postgres;

--

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

--

CREATE FUNCTION public.magazaelemansayisiazalt() RETURNS trigger

LANGUAGE plpgsql

AS $$

declare

son bigint;

begin

son=currval('"public"."silinenyetkili\_id\_seq"'::regclass);

update magaza set calisansayisi = calisansayisi-1 where id =(select magazaid from silinenyetkili where id=son) ;

return new;

end;

$$;

ALTER FUNCTION public.magazaelemansayisiazalt() OWNER TO postgres;

--

-- Name: magazaguncelle(bigint, character varying, character varying); Type: FUNCTION; Schema: public; Owner: postgres

--

CREATE FUNCTION public.magazaguncelle(\_girid bigint, \_isim character varying, \_il character varying) RETURNS void

LANGUAGE plpgsql

AS $$

DECLARE

BEGIN

update magaza set isim=\_isim where id=\_girid;

update magaza set il=\_il where id=\_girid;

END

$$;

ALTER FUNCTION public.magazaguncelle(\_girid bigint, \_isim character varying, \_il character varying) OWNER TO postgres;

--

-- Name: urunekle(character varying, character varying, character varying, bigint); Type: FUNCTION; Schema: public; Owner: postgres

--

CREATE FUNCTION public.urunekle(urunturu character varying, marka character varying, model character varying, hamfiyat bigint) RETURNS void

LANGUAGE plpgsql

AS $$

DECLARE

son int;

kdvlifiyat int;

BEGIN

INSERT INTO "public"."urun" ("urunturu","marka","model","hamfiyat") VALUES (urunturu,marka,model,hamfiyat);

son=currval('"public"."urun\_id\_seq"'::regclass);

kdvlifiyat = KDVHesapla(hamfiyat,1);

if ( urunturu='anakart') then

INSERT INTO "public"."anakart" ("id","urunturu","marka","model","hamfiyat","vergilifiyat") VALUES (son,urunturu,marka,model,hamfiyat,kdvlifiyat);

end if;

if ( urunturu='ekrankarti') then

INSERT INTO "public"."ekrankarti" ("id","urunturu","marka","model","hamfiyat","vergilifiyat") VALUES (son,urunturu,marka,model,hamfiyat,kdvlifiyat);

end if;

if ( urunturu='guckaynagi') then

INSERT INTO "public"."guckaynagi" ("id","urunturu","marka","model","hamfiyat","vergilifiyat") VALUES (son,urunturu,marka,model,hamfiyat,kdvlifiyat);

end if;

if ( urunturu='islemci') then

INSERT INTO "public"."islemci" ("id","urunturu","marka","model","hamfiyat","vergilifiyat") VALUES (son,urunturu,marka,model,hamfiyat,kdvlifiyat);

end if;

if ( urunturu='ram') then

INSERT INTO "public"."ram" ("id","urunturu","marka","model","hamfiyat","vergilifiyat") VALUES (son,urunturu,marka,model,hamfiyat,kdvlifiyat);

end if;

if ( urunturu='depolama') then

INSERT INTO "public"."depolama" ("id","urunturu","marka","model","hamfiyat","vergilifiyat") VALUES (son,urunturu,marka,model,hamfiyat,kdvlifiyat);

end if;

END

$$;

ALTER FUNCTION public.urunekle(urunturu character varying, marka character varying, model character varying, hamfiyat bigint) OWNER TO postgres;

--

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

--

CREATE FUNCTION public.urunsil(girid bigint) RETURNS void

LANGUAGE plpgsql

AS $$

DECLARE

ttur character varying;

mmarka character varying;

mmodel character varying;

hhamfiyat BIGINT;

BEGIN

ttur = (SELECT urunturu from urun where id=girid limit 1 );

mmarka = (SELECT marka from urun where id=girid limit 1);

mmodel = (SELECT model from urun where id=girid limit 1);

hhamfiyat = (SELECT hamfiyat from urun where id=girid limit 1);

insert into silinenurun (id,urunturu,marka,model,hamfiyat) values (girid,ttur,mmarka,mmodel,hhamfiyat);

DELETE FROM urun WHERE id=girid;

END

$$;

ALTER FUNCTION public.urunsil(girid bigint) OWNER TO postgres;

SET default\_tablespace = '';

SET default\_table\_access\_method = heap;

--

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

--

CREATE TABLE public.urun (

id bigint NOT NULL,

urunturu character varying(50) NOT NULL,

marka character varying(50) NOT NULL,

model character varying(50) NOT NULL,

hamfiyat bigint NOT NULL,

vergilifiyat bigint

);

ALTER TABLE public.urun OWNER TO postgres;

--

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

--

CREATE TABLE public.anakart (

)

INHERITS (public.urun);

ALTER TABLE public.anakart OWNER TO postgres;

--

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

--

CREATE TABLE public.depo (

id bigint NOT NULL,

isim character varying(50) NOT NULL,

il character varying(50) NOT NULL,

depocusayisi bigint

);

ALTER TABLE public.depo OWNER TO postgres;

--

-- Name: depo\_id\_seq; Type: SEQUENCE; Schema: public; Owner: postgres

--

CREATE SEQUENCE public.depo\_id\_seq

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE public.depo\_id\_seq OWNER TO postgres;

--

-- Name: depo\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: postgres

--

ALTER SEQUENCE public.depo\_id\_seq OWNED BY public.depo.id;

--

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

--

CREATE TABLE public.yetkili (

id bigint NOT NULL,

isim character varying(50) NOT NULL,

username character varying(50) NOT NULL,

sifre character varying(50) NOT NULL,

yetkili boolean NOT NULL

);

ALTER TABLE public.yetkili OWNER TO postgres;

--

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

--

CREATE TABLE public.depocu (

depoid bigint NOT NULL

)

INHERITS (public.yetkili);

ALTER TABLE public.depocu OWNER TO postgres;

--

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

--

CREATE TABLE public.depolama (

)

INHERITS (public.urun);

ALTER TABLE public.depolama OWNER TO postgres;

--

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

--

CREATE TABLE public.ekrankarti (

)

INHERITS (public.urun);

ALTER TABLE public.ekrankarti OWNER TO postgres;

--

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

--

CREATE TABLE public.guckaynagi (

)

INHERITS (public.urun);

ALTER TABLE public.guckaynagi OWNER TO postgres;

--

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

--

CREATE TABLE public.islemci (

)

INHERITS (public.urun);

ALTER TABLE public.islemci OWNER TO postgres;

--

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

--

CREATE TABLE public.magaza (

id bigint NOT NULL,

isim character varying(50) NOT NULL,

il character varying(50) NOT NULL,

calisansayisi bigint

);

ALTER TABLE public.magaza OWNER TO postgres;

--

-- Name: magaza\_id\_seq; Type: SEQUENCE; Schema: public; Owner: postgres

--

CREATE SEQUENCE public.magaza\_id\_seq

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE public.magaza\_id\_seq OWNER TO postgres;

--

-- Name: magaza\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: postgres

--

ALTER SEQUENCE public.magaza\_id\_seq OWNED BY public.magaza.id;

--

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

--

CREATE TABLE public.magazaci (

magazaid bigint NOT NULL

)

INHERITS (public.yetkili);

ALTER TABLE public.magazaci OWNER TO postgres;

--

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

--

CREATE TABLE public.ram (

)

INHERITS (public.urun);

ALTER TABLE public.ram OWNER TO postgres;

--

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

--

CREATE TABLE public.silinenurun (

id bigint NOT NULL,

urunturu character varying(50) NOT NULL,

marka character varying(50) NOT NULL,

model character varying(50) NOT NULL,

hamfiyat bigint NOT NULL

);

ALTER TABLE public.silinenurun OWNER TO postgres;

--

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

--

CREATE TABLE public.silinenyetkili (

id bigint NOT NULL,

isim character varying(50) NOT NULL,

username character varying(50) NOT NULL,

sifre character varying(50) NOT NULL,

magazaid bigint

);

ALTER TABLE public.silinenyetkili OWNER TO postgres;

--

-- Name: silinenyetkili\_id\_seq; Type: SEQUENCE; Schema: public; Owner: postgres

--

CREATE SEQUENCE public.silinenyetkili\_id\_seq

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE public.silinenyetkili\_id\_seq OWNER TO postgres;

--

-- Name: silinenyetkili\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: postgres

--

ALTER SEQUENCE public.silinenyetkili\_id\_seq OWNED BY public.silinenyetkili.id;

--

-- Name: urun\_id\_seq; Type: SEQUENCE; Schema: public; Owner: postgres

--

CREATE SEQUENCE public.urun\_id\_seq

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE public.urun\_id\_seq OWNER TO postgres;

--

-- Name: urun\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: postgres

--

ALTER SEQUENCE public.urun\_id\_seq OWNED BY public.urun.id;

--

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

--

CREATE TABLE public.vergi (

id bigint NOT NULL,

ad character varying(50) NOT NULL,

vergiorani double precision NOT NULL

);

ALTER TABLE public.vergi OWNER TO postgres;

--

-- Name: vergi\_id\_seq; Type: SEQUENCE; Schema: public; Owner: postgres

--

CREATE SEQUENCE public.vergi\_id\_seq

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE public.vergi\_id\_seq OWNER TO postgres;

--

-- Name: vergi\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: postgres

--

ALTER SEQUENCE public.vergi\_id\_seq OWNED BY public.vergi.id;

--

-- Name: yetkili\_id\_seq; Type: SEQUENCE; Schema: public; Owner: postgres

--

CREATE SEQUENCE public.yetkili\_id\_seq

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE public.yetkili\_id\_seq OWNER TO postgres;

--

-- Name: yetkili\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: postgres

--

ALTER SEQUENCE public.yetkili\_id\_seq OWNED BY public.yetkili.id;

--

-- Name: anakart id; Type: DEFAULT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.anakart ALTER COLUMN id SET DEFAULT nextval('public.urun\_id\_seq'::regclass);

--

-- Name: depo id; Type: DEFAULT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.depo ALTER COLUMN id SET DEFAULT nextval('public.depo\_id\_seq'::regclass);

--

-- Name: depocu id; Type: DEFAULT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.depocu ALTER COLUMN id SET DEFAULT nextval('public.yetkili\_id\_seq'::regclass);

--

-- Name: depolama id; Type: DEFAULT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.depolama ALTER COLUMN id SET DEFAULT nextval('public.urun\_id\_seq'::regclass);

--

-- Name: ekrankarti id; Type: DEFAULT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.ekrankarti ALTER COLUMN id SET DEFAULT nextval('public.urun\_id\_seq'::regclass);

--

-- Name: guckaynagi id; Type: DEFAULT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.guckaynagi ALTER COLUMN id SET DEFAULT nextval('public.urun\_id\_seq'::regclass);

--

-- Name: islemci id; Type: DEFAULT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.islemci ALTER COLUMN id SET DEFAULT nextval('public.urun\_id\_seq'::regclass);

--

-- Name: magaza id; Type: DEFAULT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.magaza ALTER COLUMN id SET DEFAULT nextval('public.magaza\_id\_seq'::regclass);

--

-- Name: magazaci id; Type: DEFAULT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.magazaci ALTER COLUMN id SET DEFAULT nextval('public.yetkili\_id\_seq'::regclass);

--

-- Name: ram id; Type: DEFAULT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.ram ALTER COLUMN id SET DEFAULT nextval('public.urun\_id\_seq'::regclass);

--

-- Name: silinenyetkili id; Type: DEFAULT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.silinenyetkili ALTER COLUMN id SET DEFAULT nextval('public.silinenyetkili\_id\_seq'::regclass);

--

-- Name: urun id; Type: DEFAULT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.urun ALTER COLUMN id SET DEFAULT nextval('public.urun\_id\_seq'::regclass);

--

-- Name: vergi id; Type: DEFAULT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.vergi ALTER COLUMN id SET DEFAULT nextval('public.vergi\_id\_seq'::regclass);

--

-- Name: yetkili id; Type: DEFAULT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.yetkili ALTER COLUMN id SET DEFAULT nextval('public.yetkili\_id\_seq'::regclass);

--

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

--

INSERT INTO public.anakart VALUES

(3, 'anakart', 'MSI', 'BAZOOKA', 1100, 1298);

--

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

--

INSERT INTO public.depo VALUES

(1, 'TATANPC', 'ORDU', 2),

(2, 'VATANPC', 'ANKARA', 0);

--

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

--

INSERT INTO public.depocu VALUES

(5, 'Arslancan', 'admin', 'admin', true, 1),

(6, 'Eraycan', 'username', 'pass', true, 1);

--

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

--

INSERT INTO public.depolama VALUES

(4, 'depolama', 'SAMSUNG', '980PRO', 1400, 1652);

--

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

--

INSERT INTO public.ekrankarti VALUES

(1, 'ekrankarti', 'ASUS', 'RTX2080', 8000, 9440);

--

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

--

--

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

--

INSERT INTO public.islemci VALUES

(2, 'islemci', 'AMD', '3700X', 2500, 2950);

--

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

--

INSERT INTO public.magaza VALUES

(1, 'Arslan Ticaret', 'ordu', 1);

--

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

--

INSERT INTO public.magazaci VALUES

(7, 'Arslan', 'admin1', 'admin', true, 1);

--

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

--

INSERT INTO public.ram VALUES

(5, 'ram', 'KINGSTON', '16GB', 700, 826);

--

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

--

INSERT INTO public.silinenurun VALUES

(6, 'ekrankarti', 'PALIT', 'RTX3060', 4800);

--

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

--

INSERT INTO public.silinenyetkili VALUES

(1, 'Arslancan Sarıkaya', 'admin', 'admin', 2),

(2, 'Yiğit', 'admin1', 'admin', 1),

(3, 'Yiğit Sarıkaya', 'test', 'test', 1),

(4, 'Hüseyin', 'testyetki', 'test', 123),

(5, 'Hüseyin', 'testyetki', 'test', 123),

(6, 'eRAY', 'ASD', '123123', 1);

--

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

--

INSERT INTO public.urun VALUES

(1, 'ekrankarti', 'ASUS', 'RTX2080', 8000, NULL),

(2, 'islemci', 'AMD', '3700X', 2500, NULL),

(3, 'anakart', 'MSI', 'BAZOOKA', 1100, NULL),

(4, 'depolama', 'SAMSUNG', '980PRO', 1400, NULL),

(5, 'ram', 'KINGSTON', '16GB', 700, NULL);

--

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

--

INSERT INTO public.vergi VALUES

(1, 'KDV', 0.18);

--

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

--

--

-- Name: depo\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

SELECT pg\_catalog.setval('public.depo\_id\_seq', 2, true);

--

-- Name: magaza\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

SELECT pg\_catalog.setval('public.magaza\_id\_seq', 2, true);

--

-- Name: silinenyetkili\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

SELECT pg\_catalog.setval('public.silinenyetkili\_id\_seq', 6, true);

--

-- Name: urun\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

SELECT pg\_catalog.setval('public.urun\_id\_seq', 6, true);

--

-- Name: vergi\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

SELECT pg\_catalog.setval('public.vergi\_id\_seq', 1, false);

--

-- Name: yetkili\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

SELECT pg\_catalog.setval('public.yetkili\_id\_seq', 7, true);

--

-- Name: anakart anakart\_id\_key; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.anakart

ADD CONSTRAINT anakart\_id\_key UNIQUE (id);

--

-- Name: depo depo\_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.depo

ADD CONSTRAINT depo\_pkey PRIMARY KEY (id);

--

-- Name: depolama depolama\_id\_key; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.depolama

ADD CONSTRAINT depolama\_id\_key UNIQUE (id);

--

-- Name: ekrankarti ekrankarti\_id\_key; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.ekrankarti

ADD CONSTRAINT ekrankarti\_id\_key UNIQUE (id);

--

-- Name: guckaynagi guckaynagi\_id\_key; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.guckaynagi

ADD CONSTRAINT guckaynagi\_id\_key UNIQUE (id);

--

-- Name: islemci islemci\_id\_key; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.islemci

ADD CONSTRAINT islemci\_id\_key UNIQUE (id);

--

-- Name: magaza magaza\_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.magaza

ADD CONSTRAINT magaza\_pkey PRIMARY KEY (id);

--

-- Name: ram ram\_id\_key; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.ram

ADD CONSTRAINT ram\_id\_key UNIQUE (id);

--

-- Name: silinenurun silinenurun\_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.silinenurun

ADD CONSTRAINT silinenurun\_pkey PRIMARY KEY (id);

--

-- Name: silinenyetkili silinenyetkili\_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.silinenyetkili

ADD CONSTRAINT silinenyetkili\_pkey PRIMARY KEY (id);

--

-- Name: urun urun\_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.urun

ADD CONSTRAINT urun\_pkey PRIMARY KEY (id);

--

-- Name: vergi vergi\_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.vergi

ADD CONSTRAINT vergi\_pkey PRIMARY KEY (id);

--

-- Name: yetkili yetkili\_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.yetkili

ADD CONSTRAINT yetkili\_pkey PRIMARY KEY (id);

--

-- Name: depocu depoelemanarttirtrig; Type: TRIGGER; Schema: public; Owner: postgres

--

CREATE TRIGGER depoelemanarttirtrig AFTER INSERT ON public.depocu FOR EACH ROW EXECUTE FUNCTION public.depoelemansayisiarttir();

--

-- Name: depocu depoelemanazalttrig; Type: TRIGGER; Schema: public; Owner: postgres

--

CREATE TRIGGER depoelemanazalttrig AFTER DELETE ON public.depocu FOR EACH ROW EXECUTE FUNCTION public.depoelemansayisiazalt();

--

-- Name: magazaci magazaelemanazalttrig; Type: TRIGGER; Schema: public; Owner: postgres

--

CREATE TRIGGER magazaelemanazalttrig AFTER DELETE ON public.magazaci FOR EACH ROW EXECUTE FUNCTION public.magazaelemansayisiazalt();

--

-- Name: magazaci magazeelemantrig; Type: TRIGGER; Schema: public; Owner: postgres

--

CREATE TRIGGER magazeelemantrig AFTER INSERT ON public.magazaci FOR EACH ROW EXECUTE FUNCTION public.magazaelemansayisiarttir();