

VERİTABANI YÖNETİM SİSTEMLERİ

Onur etlioğlu s191210515

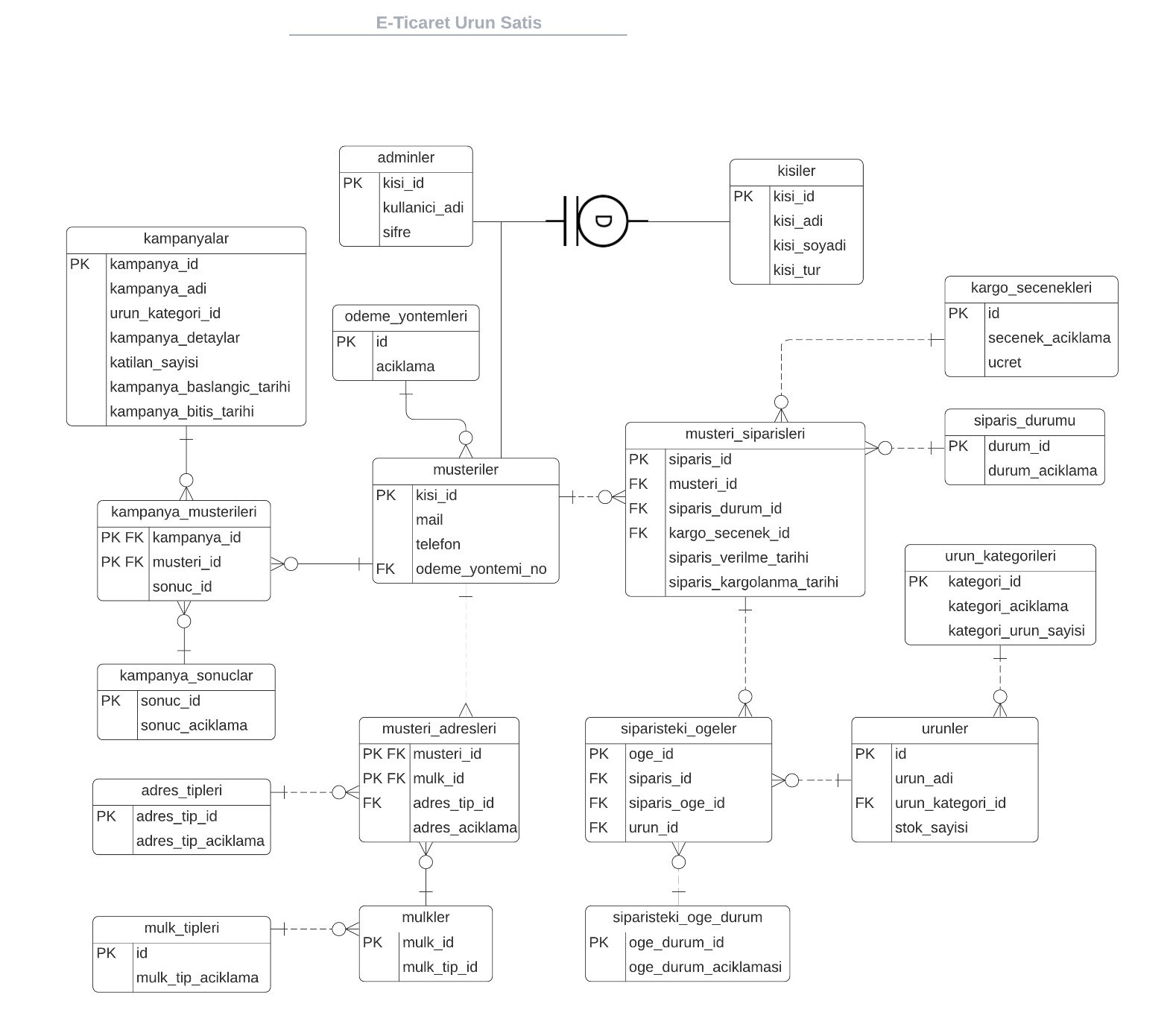
İŞ KURALLARI

* Uygulama kullanici adi sifre sorgusuyla admin yetkisiyle başlar.
* Adminler urunleri ekleyip guncelleyip silebilirler.
* Adminler ve musteriler birer kisidir.
* Musteriler urunleri siparis edebilir.
* Urunler kargolanabilir, onaylanabilir veya iptal olur.
* Urunlerin kategorileri vardir.
* Kategorilerde urun sayilari da tutulur.
* Musterilerin bazılarına özel kampanya yapılabilir.
* Kampanyalar düşük ücretli ürünleri içerir.
* Kampanyaya katılan sayısı tutulur.
* Kampanyanın son tarihi vardır.
* Musterilerin secimine göre kampanya katıldı reddetti gibi sonuclar vardır.
* Musterilerin adresleri vardır.
* Adreslerin kargo fatura gibi türleri vardır.
* Adreslere ait mülk tipleri vardır.
* Musterilern farklı ödeme yöntemleri olabilir.

İLİŞKİSEL ŞEMA

* musteriler(kisi\_id:Integer, mail:Varchar, odeme\_yontemi\_no:Integer, telefon:Varchar)
* kisiler(kisi\_id:Integer, kisi\_adi:Varchar, kisi\_soyadi:Varchar, kisi\_tur:Char)
* kampanyalar(kampanya\_id:Integer, kampanya\_adi:Varchar, kampanya\_detaylar:Varchar, kampanya\_baslangic\_tarihi:Datetime, kampanya\_bitis\_tarihi:Datetime, katilan\_sayisi:Integer, urun\_kategori\_id: Integer)
* musteri\_siparisleri(siparis\_id:Integer, siparis\_durum\_id:Integer, kargo\_secenek\_id:Integer, musteri\_id:Integer, siparis\_kargolanma\_tarihi:Datetime, siparis\_verilme\_tarihi:DateTime)
* urun\_kategorileri(kategori\_id:Integer, kategori\_aciklama:Varchar, kategori\_urun\_sayisi:Integer)
* mulk\_tipleri(id:Integer, mulk\_tip\_aciklama:Varchar)
* urunler(id:Integer, urun\_adi:Varchar, urun\_kategori\_id:Integer, stok\_sayisi:Integer)
* musteri\_adresleri(mulk\_id:Integer, musteri\_id:Integer, adres\_tip\_id:Integer, adres\_aciklamasi:Varchar)
* adres\_tipleri(adres\_tip\_id:Integer, adres\_tip\_aciklama: Varchar)
* siparisteki\_ogeler(oge\_id:Integer, siparis\_id:Integer, oge\_durum\_id:Integer, urun\_id:Integer)
* kargo\_secenekleri(id:Integer, secenek\_aciklama:Varchar, ucret:Integer)
* odeme\_yontemleri(id:Integer, aciklama:Varchar)
* siparisteki\_oge\_durum(oge\_durum\_id:Integer, oge\_durum\_aciklama:Varchar)
* kampanya\_musterileri(kampanya\_id:Integer, musteri\_id:Integer, sonuc\_id:Integer)
* adminler(kisi\_id:Integer, kullanici\_adi:Varchar, sifre:Varchar)
* siparis\_durumu(durum\_id:Integer, durum\_aciklama:Varchar)
* mulkler(mulk\_id:Integer, mulk\_tip\_id:Integer)

VARLIK BAĞINTI MODELİ



SQL KOMUTLARI

--

-- PostgreSQL database dump

--

-- Dumped from database version 12.3

-- Dumped by pg\_dump version 12rc1

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: kampanya\_katilan(); Type: FUNCTION; Schema: public; Owner: postgres

--

CREATE FUNCTION public.kampanya\_katilan() RETURNS trigger

LANGUAGE plpgsql

AS $$

BEGIN

UPDATE kampanyalar

SET katilan\_sayisi = katilan\_sayisi + 1 WHERE kampanya\_id = (SELECT kampanya\_id FROM kampanya\_musterileri WHERE sonuc\_id = 2 AND kampanya\_id = NEW.kampanya\_id);

RETURN NEW;

END;

$$;

ALTER FUNCTION public.kampanya\_katilan() OWNER TO postgres;

--

-- Name: kategori\_azalt(); Type: FUNCTION; Schema: public; Owner: postgres

--

CREATE FUNCTION public.kategori\_azalt() RETURNS trigger

LANGUAGE plpgsql

AS $$

BEGIN

UPDATE urun\_kategorileri

SET kategori\_urun\_sayisi = kategori\_urun\_sayisi - 1

where kategori\_id = (SELECT urun\_kategori\_id from urunler WHERE id = NEW.id AND OLD.stok\_sayisi - 1 = NEW.stok\_sayisi);

--- (SELECT urun\_id from siparisteki\_ogeler WHERE siparis\_oge\_durum\_id = 2 AND urun\_id = NEW.urun\_id);---

RETURN NEW;

END;

$$;

ALTER FUNCTION public.kategori\_azalt() OWNER TO postgres;

--

-- Name: oge\_durum\_degistir(); Type: FUNCTION; Schema: public; Owner: postgres

--

CREATE FUNCTION public.oge\_durum\_degistir() RETURNS trigger

LANGUAGE plpgsql

AS $$

BEGIN

UPDATE siparisteki\_ogeler

SET siparis\_oge\_durum\_id = NEW."siparis\_durum\_id" WHERE siparisteki\_ogeler.siparis\_id = NEW.siparis\_id;

RETURN NEW;

END;

$$;

ALTER FUNCTION public.oge\_durum\_degistir() OWNER TO postgres;

--

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

--

CREATE FUNCTION public.siparisara(aranansiparis integer) RETURNS TABLE(siparisid integer, siparisdurum integer, kargosecenek integer, siparistarih timestamp without time zone)

LANGUAGE plpgsql

AS $$

BEGIN

RETURN QUERY SELECT "siparis\_id", "siparis\_durum\_id", "kargo\_secenek\_id", "siparis\_verilme\_tarihi" FROM musteri\_siparisleri

WHERE "siparis\_id" = aranansiparis;

END;

$$;

ALTER FUNCTION public.siparisara(aranansiparis integer) OWNER TO postgres;

--

-- Name: stok\_azalt(); Type: FUNCTION; Schema: public; Owner: postgres

--

CREATE FUNCTION public.stok\_azalt() RETURNS trigger

LANGUAGE plpgsql

AS $$

BEGIN

UPDATE urunler

SET stok\_sayisi = stok\_sayisi - 1

where id = (SELECT urun\_id from siparisteki\_ogeler WHERE siparis\_oge\_durum\_id = 2 AND siparis\_id = NEW.siparis\_id);

RETURN NEW;

END;

$$;

ALTER FUNCTION public.stok\_azalt() OWNER TO postgres;

--

-- Name: tum\_urunler(); Type: FUNCTION; Schema: public; Owner: postgres

--

CREATE FUNCTION public.tum\_urunler() RETURNS TABLE(urunid integer, urunkategori character varying, urunadi character varying, stoksayisi integer)

LANGUAGE plpgsql

AS $$

BEGIN

RETURN QUERY SELECT "id", urun\_kategorileri."kategori\_aciklama", "urun\_adi", "stok\_sayisi" FROM urunler

INNER JOIN urun\_kategorileri ON urunler.urun\_kategori\_id = urun\_kategorileri.kategori\_id;

END;

$$;

ALTER FUNCTION public.tum\_urunler() OWNER TO postgres;

--

-- Name: urun\_ara(integer); Type: FUNCTION; Schema: public; Owner: postgres

--

CREATE FUNCTION public.urun\_ara(arananurun integer) RETURNS TABLE(urunid integer, urunkategori integer, urunadi character varying, stoksayisi integer)

LANGUAGE plpgsql

AS $$

BEGIN

RETURN QUERY SELECT "id", "urun\_kategori\_id", "urun\_adi", "stok\_sayisi" FROM urunler

WHERE "id" = arananurun;

END;

$$;

ALTER FUNCTION public.urun\_ara(arananurun integer) OWNER TO postgres;

--

-- Name: urunler\_toplami(integer); Type: FUNCTION; Schema: public; Owner: postgres

--

CREATE FUNCTION public.urunler\_toplami(kategoriid integer) RETURNS text

LANGUAGE plpgsql

AS $$

DECLARE

toplam NUMERIC;

urunler RECORD;

BEGIN

urunler := urun\_ara(2);

toplam := (SELECT SUM(stok\_sayisi) FROM urunler WHERE urunler.urunkategori = kategoriid);

RETURN urunler."urunkategori" || E'\t' || toplam;

END

$$;

ALTER FUNCTION public.urunler\_toplami(kategoriid integer) OWNER TO postgres;

SET default\_tablespace = '';

SET default\_table\_access\_method = heap;

--

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

--

CREATE TABLE public.adminler (

kisi\_id integer NOT NULL,

kullanici\_adi character varying NOT NULL,

sifre character varying NOT NULL

);

ALTER TABLE public.adminler OWNER TO postgres;

--

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

--

CREATE TABLE public.adres\_tipleri (

adres\_tip\_id integer NOT NULL,

adres\_tip\_aciklama character varying NOT NULL

);

ALTER TABLE public.adres\_tipleri OWNER TO postgres;

--

-- Name: adres\_tipleri\_adres\_tip\_id\_seq; Type: SEQUENCE; Schema: public; Owner: postgres

--

CREATE SEQUENCE public.adres\_tipleri\_adres\_tip\_id\_seq

AS integer

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE public.adres\_tipleri\_adres\_tip\_id\_seq OWNER TO postgres;

--

-- Name: adres\_tipleri\_adres\_tip\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: postgres

--

ALTER SEQUENCE public.adres\_tipleri\_adres\_tip\_id\_seq OWNED BY public.adres\_tipleri.adres\_tip\_id;

--

-- Name: kampanya\_musterileri; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public.kampanya\_musterileri (

kampanya\_id integer NOT NULL,

musteri\_id integer NOT NULL,

sonuc\_id integer NOT NULL

);

ALTER TABLE public.kampanya\_musterileri OWNER TO postgres;

--

-- Name: kampanya\_sonuclar; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public.kampanya\_sonuclar (

sonuc\_id integer NOT NULL,

sonuc\_aciklama character varying NOT NULL

);

ALTER TABLE public.kampanya\_sonuclar OWNER TO postgres;

--

-- Name: kampanya\_sonuclar\_sonuc\_id\_seq; Type: SEQUENCE; Schema: public; Owner: postgres

--

CREATE SEQUENCE public.kampanya\_sonuclar\_sonuc\_id\_seq

AS integer

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE public.kampanya\_sonuclar\_sonuc\_id\_seq OWNER TO postgres;

--

-- Name: kampanya\_sonuclar\_sonuc\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: postgres

--

ALTER SEQUENCE public.kampanya\_sonuclar\_sonuc\_id\_seq OWNED BY public.kampanya\_sonuclar.sonuc\_id;

--

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

--

CREATE TABLE public.kampanyalar (

kampanya\_id integer NOT NULL,

urun\_kategori\_id integer NOT NULL,

kampanya\_adi character varying NOT NULL,

kampanya\_baslangic\_tarihi timestamp without time zone NOT NULL,

kampanya\_son\_tarihi timestamp without time zone NOT NULL,

"kampanya detaylar" character varying NOT NULL,

katilan\_sayisi integer DEFAULT 0 NOT NULL

);

ALTER TABLE public.kampanyalar OWNER TO postgres;

--

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

--

CREATE SEQUENCE public.kampanyalar\_kampanya\_id\_seq

AS integer

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE public.kampanyalar\_kampanya\_id\_seq OWNER TO postgres;

--

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

--

ALTER SEQUENCE public.kampanyalar\_kampanya\_id\_seq OWNED BY public.kampanyalar.kampanya\_id;

--

-- Name: kargo\_secenekleri; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public.kargo\_secenekleri (

id integer NOT NULL,

secenek\_aciklamasi character varying NOT NULL,

ucret integer NOT NULL

);

ALTER TABLE public.kargo\_secenekleri OWNER TO postgres;

--

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

--

CREATE SEQUENCE public.kargo\_secenekleri\_id\_seq

AS integer

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE public.kargo\_secenekleri\_id\_seq OWNER TO postgres;

--

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

--

ALTER SEQUENCE public.kargo\_secenekleri\_id\_seq OWNED BY public.kargo\_secenekleri.id;

--

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

--

CREATE TABLE public.kisiler (

kisi\_id integer NOT NULL,

kisi\_adi character varying NOT NULL,

kisi\_soyadi character varying NOT NULL,

kisi\_tur character(1) NOT NULL

);

ALTER TABLE public.kisiler OWNER TO postgres;

--

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

--

CREATE SEQUENCE public.kisiler\_kisi\_id\_seq

AS integer

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE public.kisiler\_kisi\_id\_seq OWNER TO postgres;

--

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

--

ALTER SEQUENCE public.kisiler\_kisi\_id\_seq OWNED BY public.kisiler.kisi\_id;

--

-- Name: mulk\_tipleri; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public.mulk\_tipleri (

id integer NOT NULL,

mulk\_tip\_aciklama character varying NOT NULL

);

ALTER TABLE public.mulk\_tipleri OWNER TO postgres;

--

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

--

CREATE SEQUENCE public.mulk\_tipleri\_id\_seq

AS integer

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE public.mulk\_tipleri\_id\_seq OWNER TO postgres;

--

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

--

ALTER SEQUENCE public.mulk\_tipleri\_id\_seq OWNED BY public.mulk\_tipleri.id;

--

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

--

CREATE TABLE public.mulkler (

mulk\_id integer NOT NULL,

mulk\_tip\_id integer NOT NULL

);

ALTER TABLE public.mulkler OWNER TO postgres;

--

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

--

CREATE SEQUENCE public.mulkler\_mulk\_id\_seq

AS integer

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE public.mulkler\_mulk\_id\_seq OWNER TO postgres;

--

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

--

ALTER SEQUENCE public.mulkler\_mulk\_id\_seq OWNED BY public.mulkler.mulk\_id;

--

-- Name: musteri\_adresleri; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public.musteri\_adresleri (

musteri\_id integer NOT NULL,

mulk\_id integer NOT NULL,

adres\_tip\_id integer NOT NULL,

adres\_aciklamasi character varying NOT NULL

);

ALTER TABLE public.musteri\_adresleri OWNER TO postgres;

--

-- Name: musteri\_siparisleri; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public.musteri\_siparisleri (

siparis\_id integer NOT NULL,

musteri\_id integer NOT NULL,

siparis\_durum\_id integer NOT NULL,

kargo\_secenek\_id integer NOT NULL,

siparis\_verilme\_tarihi timestamp without time zone DEFAULT now() NOT NULL,

siparis\_kargolanma\_tarihi timestamp without time zone DEFAULT now() NOT NULL

);

ALTER TABLE public.musteri\_siparisleri OWNER TO postgres;

--

-- Name: musteri\_siparisleri\_siparis\_id\_seq; Type: SEQUENCE; Schema: public; Owner: postgres

--

CREATE SEQUENCE public.musteri\_siparisleri\_siparis\_id\_seq

AS integer

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE public.musteri\_siparisleri\_siparis\_id\_seq OWNER TO postgres;

--

-- Name: musteri\_siparisleri\_siparis\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: postgres

--

ALTER SEQUENCE public.musteri\_siparisleri\_siparis\_id\_seq OWNED BY public.musteri\_siparisleri.siparis\_id;

--

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

--

CREATE TABLE public.musteriler (

kisi\_id integer NOT NULL,

mail character varying NOT NULL,

telefon character varying NOT NULL,

odeme\_yontemi\_no integer NOT NULL

);

ALTER TABLE public.musteriler OWNER TO postgres;

--

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

--

CREATE SEQUENCE public.musteriler\_musteri\_id\_seq

AS integer

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE public.musteriler\_musteri\_id\_seq OWNER TO postgres;

--

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

--

ALTER SEQUENCE public.musteriler\_musteri\_id\_seq OWNED BY public.musteriler.kisi\_id;

--

-- Name: odeme\_yontemleri; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public.odeme\_yontemleri (

id integer NOT NULL,

aciklama character varying NOT NULL

);

ALTER TABLE public.odeme\_yontemleri OWNER TO postgres;

--

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

--

CREATE SEQUENCE public.odeme\_yontemleri\_id\_seq

AS integer

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE public.odeme\_yontemleri\_id\_seq OWNER TO postgres;

--

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

--

ALTER SEQUENCE public.odeme\_yontemleri\_id\_seq OWNED BY public.odeme\_yontemleri.id;

--

-- Name: siparis\_durumu; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public.siparis\_durumu (

durum\_id integer NOT NULL,

durum\_aciklama character varying NOT NULL

);

ALTER TABLE public.siparis\_durumu OWNER TO postgres;

--

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

--

CREATE SEQUENCE public.siparis\_durumu\_id\_seq

AS integer

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE public.siparis\_durumu\_id\_seq OWNER TO postgres;

--

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

--

ALTER SEQUENCE public.siparis\_durumu\_id\_seq OWNED BY public.siparis\_durumu.durum\_id;

--

-- Name: siparisteki\_oge\_durum; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public.siparisteki\_oge\_durum (

oge\_durum\_id integer NOT NULL,

oge\_durum\_aciklamasi character varying NOT NULL

);

ALTER TABLE public.siparisteki\_oge\_durum OWNER TO postgres;

--

-- Name: siparisteki\_oge\_durum\_siparis\_oge\_durum\_id\_seq; Type: SEQUENCE; Schema: public; Owner: postgres

--

CREATE SEQUENCE public.siparisteki\_oge\_durum\_siparis\_oge\_durum\_id\_seq

AS integer

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE public.siparisteki\_oge\_durum\_siparis\_oge\_durum\_id\_seq OWNER TO postgres;

--

-- Name: siparisteki\_oge\_durum\_siparis\_oge\_durum\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: postgres

--

ALTER SEQUENCE public.siparisteki\_oge\_durum\_siparis\_oge\_durum\_id\_seq OWNED BY public.siparisteki\_oge\_durum.oge\_durum\_id;

--

-- Name: siparisteki\_ogeler; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public.siparisteki\_ogeler (

oge\_id integer NOT NULL,

siparis\_oge\_durum\_id integer NOT NULL,

siparis\_id integer NOT NULL,

urun\_id integer NOT NULL

);

ALTER TABLE public.siparisteki\_ogeler OWNER TO postgres;

--

-- Name: siparisteki\_ogeler\_oge\_id\_seq; Type: SEQUENCE; Schema: public; Owner: postgres

--

CREATE SEQUENCE public.siparisteki\_ogeler\_oge\_id\_seq

AS integer

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE public.siparisteki\_ogeler\_oge\_id\_seq OWNER TO postgres;

--

-- Name: siparisteki\_ogeler\_oge\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: postgres

--

ALTER SEQUENCE public.siparisteki\_ogeler\_oge\_id\_seq OWNED BY public.siparisteki\_ogeler.oge\_id;

--

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

--

CREATE TABLE public.urun\_kategorileri (

kategori\_id integer NOT NULL,

kategori\_aciklama character varying NOT NULL,

kategori\_urun\_sayisi integer DEFAULT 0

);

ALTER TABLE public.urun\_kategorileri OWNER TO postgres;

--

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

--

CREATE SEQUENCE public.urun\_kategorileri\_kategori\_id\_seq

AS integer

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

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

--

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

--

ALTER SEQUENCE public.urun\_kategorileri\_kategori\_id\_seq OWNED BY public.urun\_kategorileri.kategori\_id;

--

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

--

CREATE TABLE public.urunler (

id integer NOT NULL,

urun\_kategori\_id integer NOT NULL,

urun\_adi character varying NOT NULL,

stok\_sayisi integer NOT NULL

);

ALTER TABLE public.urunler OWNER TO postgres;

--

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

--

CREATE SEQUENCE public.urunler\_id\_seq

AS integer

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

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

--

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

--

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

--

-- Name: adres\_tipleri adres\_tip\_id; Type: DEFAULT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.adres\_tipleri ALTER COLUMN adres\_tip\_id SET DEFAULT nextval('public.adres\_tipleri\_adres\_tip\_id\_seq'::regclass);

--

-- Name: kampanya\_sonuclar sonuc\_id; Type: DEFAULT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.kampanya\_sonuclar ALTER COLUMN sonuc\_id SET DEFAULT nextval('public.kampanya\_sonuclar\_sonuc\_id\_seq'::regclass);

--

-- Name: kampanyalar kampanya\_id; Type: DEFAULT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.kampanyalar ALTER COLUMN kampanya\_id SET DEFAULT nextval('public.kampanyalar\_kampanya\_id\_seq'::regclass);

--

-- Name: kargo\_secenekleri id; Type: DEFAULT; Schema: public; Owner: postgres

--

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

--

-- Name: kisiler kisi\_id; Type: DEFAULT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.kisiler ALTER COLUMN kisi\_id SET DEFAULT nextval('public.kisiler\_kisi\_id\_seq'::regclass);

--

-- Name: mulk\_tipleri id; Type: DEFAULT; Schema: public; Owner: postgres

--

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

--

-- Name: mulkler mulk\_id; Type: DEFAULT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.mulkler ALTER COLUMN mulk\_id SET DEFAULT nextval('public.mulkler\_mulk\_id\_seq'::regclass);

--

-- Name: musteri\_siparisleri siparis\_id; Type: DEFAULT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.musteri\_siparisleri ALTER COLUMN siparis\_id SET DEFAULT nextval('public.musteri\_siparisleri\_siparis\_id\_seq'::regclass);

--

-- Name: odeme\_yontemleri id; Type: DEFAULT; Schema: public; Owner: postgres

--

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

--

-- Name: siparis\_durumu durum\_id; Type: DEFAULT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.siparis\_durumu ALTER COLUMN durum\_id SET DEFAULT nextval('public.siparis\_durumu\_id\_seq'::regclass);

--

-- Name: siparisteki\_oge\_durum oge\_durum\_id; Type: DEFAULT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.siparisteki\_oge\_durum ALTER COLUMN oge\_durum\_id SET DEFAULT nextval('public.siparisteki\_oge\_durum\_siparis\_oge\_durum\_id\_seq'::regclass);

--

-- Name: siparisteki\_ogeler oge\_id; Type: DEFAULT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.siparisteki\_ogeler ALTER COLUMN oge\_id SET DEFAULT nextval('public.siparisteki\_ogeler\_oge\_id\_seq'::regclass);

--

-- Name: urun\_kategorileri kategori\_id; Type: DEFAULT; Schema: public; Owner: postgres

--

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

--

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

--

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

--

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

--

INSERT INTO public.adminler VALUES

(2, 'hasan123', 'h123'),

(1, 'mehmet123', 'm123'),

(3, 'kemal123', 'k123');

--

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

--

INSERT INTO public.adres\_tipleri VALUES

(1, 'kargo adresi'),

(2, 'fatura adresi'),

(3, 'her ikisi ');

--

-- Data for Name: kampanya\_musterileri; Type: TABLE DATA; Schema: public; Owner: postgres

--

INSERT INTO public.kampanya\_musterileri VALUES

(1, 5, 2),

(2, 4, 2),

(3, 6, 1);

--

-- Data for Name: kampanya\_sonuclar; Type: TABLE DATA; Schema: public; Owner: postgres

--

INSERT INTO public.kampanya\_sonuclar VALUES

(5, 'tekrar hatırlat'),

(4, 'suresi doldu'),

(3, 'red etti'),

(2, 'kabul edildi'),

(1, 'beklemede');

--

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

--

INSERT INTO public.kampanyalar VALUES

(1, 3, 'Bebek bezi indirimi', '2020-03-22 00:00:00', '2020-05-22 00:00:00', '5 bebek bezine %10 indirim.', 2),

(2, 4, 'Kozmetikte 2 al 1 öde', '2020-03-22 00:00:00', '2020-05-22 00:00:00', '2 üründen birisi hediye.', 4),

(3, 2, 'Beyaz esyada firsat', '2020-03-22 00:00:00', '2020-05-22 00:00:00', 'Tüm beyaz esyalar %5 indirimli.', 1);

--

-- Data for Name: kargo\_secenekleri; Type: TABLE DATA; Schema: public; Owner: postgres

--

INSERT INTO public.kargo\_secenekleri VALUES

(1, 'aras-kargo', 11),

(2, 'yurtici-kargo', 14),

(3, 'mng-kargo', 9),

(4, 'xyz-kargo', 10);

--

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

--

INSERT INTO public.kisiler VALUES

(1, 'Mehmet', 'Demirkıran', 'A'),

(2, 'Hasan', 'Polat', 'A'),

(3, 'Kemal', 'Durmaz', 'A'),

(4, 'Ali', 'Palabıyık', 'M'),

(5, 'Alican', 'Aslan', 'M'),

(6, 'Serhat', 'Sönmez', 'M'),

(7, 'Salih', 'Babacan', 'M'),

(8, 'Kemal', 'Selim', 'M'),

(9, 'Murat', 'Sayılmış', 'M'),

(10, 'Rasim', 'Durdu', 'M'),

(11, 'Polat', 'Kadim', 'M');

--

-- Data for Name: mulk\_tipleri; Type: TABLE DATA; Schema: public; Owner: postgres

--

INSERT INTO public.mulk\_tipleri VALUES

(1, 'ev'),

(2, 'ofis'),

(3, 'depo'),

(4, 'işyeri');

--

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

--

INSERT INTO public.mulkler VALUES

(1, 2),

(2, 3),

(3, 2),

(4, 1),

(5, 2),

(6, 2),

(7, 4);

--

-- Data for Name: musteri\_adresleri; Type: TABLE DATA; Schema: public; Owner: postgres

--

INSERT INTO public.musteri\_adresleri VALUES

(4, 1, 1, 'Merkez Mahallesi'),

(9, 1, 1, 'Irfan Mahallesi'),

(7, 2, 3, 'Celal Mahallesi'),

(6, 2, 3, 'Burnaz Mahallesi'),

(8, 1, 3, 'Salim Mahallesi'),

(10, 1, 2, 'Atatürk Mahallesi'),

(11, 1, 3, 'Barbaros Mahallesi'),

(5, 2, 2, 'Merkez İşyeri');

--

-- Data for Name: musteri\_siparisleri; Type: TABLE DATA; Schema: public; Owner: postgres

--

INSERT INTO public.musteri\_siparisleri VALUES

(1, 4, 2, 2, '2020-05-13 00:00:00', '2020-08-13 14:00:00'),

(2, 6, 1, 1, '2020-09-13 17:00:00', '2020-08-15 14:00:00'),

(3, 7, 1, 3, '2020-11-13 17:30:00', '2020-08-17 14:00:00');

--

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

--

INSERT INTO public.musteriler VALUES

(4, 'mail1@gmail.com', '5459584578', 1),

(5, 'mail2@gmail.com', '5474512356', 2),

(6, 'mail2@gmail.com', '5455231654', 3),

(7, 'mail4@gmail.com', '5325641254', 1),

(8, 'mail5@gmail.com', '5342536598', 2),

(9, 'mail6@gmail.com', '5411523566', 1),

(10, 'mail7@gmail.com', '5541256326', 2),

(11, 'mail8@gmail.com', '5356512154', 3);

--

-- Data for Name: odeme\_yontemleri; Type: TABLE DATA; Schema: public; Owner: postgres

--

INSERT INTO public.odeme\_yontemleri VALUES

(1, 'kredi karti'),

(2, 'havale'),

(3, 'eft'),

(4, 'hediye kartı'),

(5, 'cüzdan');

--

-- Data for Name: siparis\_durumu; Type: TABLE DATA; Schema: public; Owner: postgres

--

INSERT INTO public.siparis\_durumu VALUES

(1, 'kargolandi'),

(2, 'ödendi'),

(3, 'iptal edildi');

--

-- Data for Name: siparisteki\_oge\_durum; Type: TABLE DATA; Schema: public; Owner: postgres

--

INSERT INTO public.siparisteki\_oge\_durum VALUES

(5, 'ulaştı'),

(4, 'iptal edildi'),

(3, 'kargolandi'),

(2, 'ödendi'),

(1, 'beklemede');

--

-- Data for Name: siparisteki\_ogeler; Type: TABLE DATA; Schema: public; Owner: postgres

--

INSERT INTO public.siparisteki\_ogeler VALUES

(1, 2, 1, 1),

(3, 1, 2, 1),

(2, 1, 3, 2);

--

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

--

INSERT INTO public.urun\_kategorileri VALUES

(3, 'cocuk', 1500),

(4, 'kozmetik', 4000),

(5, 'giyim', 1800),

(2, 'beyaz-esya', 2000),

(1, 'elektronik', 1697);

--

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

--

INSERT INTO public.urunler VALUES

(4, 4, 'Cocuk bezi', 1300),

(2, 2, 'Vestel no-frost buzdolabi', 400),

(1, 1, 'Acer 17" laptop', 147),

(5, 2, 'Arçelik bulaşık makinesi', 150),

(3, 2, 'Vestel ultra class çamaşır makinesi', 200),

(6, 2, 'Vestel deneme', 500);

--

-- Name: adres\_tipleri\_adres\_tip\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

SELECT pg\_catalog.setval('public.adres\_tipleri\_adres\_tip\_id\_seq', 3, true);

--

-- Name: kampanya\_sonuclar\_sonuc\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

SELECT pg\_catalog.setval('public.kampanya\_sonuclar\_sonuc\_id\_seq', 5, true);

--

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

--

SELECT pg\_catalog.setval('public.kampanyalar\_kampanya\_id\_seq', 3, true);

--

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

--

SELECT pg\_catalog.setval('public.kargo\_secenekleri\_id\_seq', 4, true);

--

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

--

SELECT pg\_catalog.setval('public.kisiler\_kisi\_id\_seq', 11, true);

--

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

--

SELECT pg\_catalog.setval('public.mulk\_tipleri\_id\_seq', 4, true);

--

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

--

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

--

-- Name: musteri\_siparisleri\_siparis\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

SELECT pg\_catalog.setval('public.musteri\_siparisleri\_siparis\_id\_seq', 3, true);

--

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

--

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

--

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

--

SELECT pg\_catalog.setval('public.odeme\_yontemleri\_id\_seq', 5, true);

--

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

--

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

--

-- Name: siparisteki\_oge\_durum\_siparis\_oge\_durum\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

SELECT pg\_catalog.setval('public.siparisteki\_oge\_durum\_siparis\_oge\_durum\_id\_seq', 5, true);

--

-- Name: siparisteki\_ogeler\_oge\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

SELECT pg\_catalog.setval('public.siparisteki\_ogeler\_oge\_id\_seq', 4, true);

--

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

--

SELECT pg\_catalog.setval('public.urun\_kategorileri\_kategori\_id\_seq', 5, true);

--

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

--

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

--

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

--

ALTER TABLE ONLY public.adminler

ADD CONSTRAINT adminler\_pkey PRIMARY KEY (kisi\_id);

--

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

--

ALTER TABLE ONLY public.adres\_tipleri

ADD CONSTRAINT adres\_tipleri\_pkey PRIMARY KEY (adres\_tip\_id);

--

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

--

ALTER TABLE ONLY public.kampanya\_musterileri

ADD CONSTRAINT kampanya\_musterileri\_pkey PRIMARY KEY (kampanya\_id, musteri\_id);

--

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

--

ALTER TABLE ONLY public.kampanya\_sonuclar

ADD CONSTRAINT kampanya\_sonuclar\_pkey PRIMARY KEY (sonuc\_id);

--

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

--

ALTER TABLE ONLY public.kampanyalar

ADD CONSTRAINT kampanyalar\_pkey PRIMARY KEY (kampanya\_id);

--

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

--

ALTER TABLE ONLY public.kargo\_secenekleri

ADD CONSTRAINT kargo\_secenekleri\_pkey PRIMARY KEY (id);

--

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

--

ALTER TABLE ONLY public.kisiler

ADD CONSTRAINT kisiler\_pkey PRIMARY KEY (kisi\_id);

--

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

--

ALTER TABLE ONLY public.mulk\_tipleri

ADD CONSTRAINT mulk\_tipleri\_pkey PRIMARY KEY (id);

--

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

--

ALTER TABLE ONLY public.mulkler

ADD CONSTRAINT mulkler\_pkey PRIMARY KEY (mulk\_id);

--

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

--

ALTER TABLE ONLY public.musteri\_adresleri

ADD CONSTRAINT musteri\_adresleri\_pkey PRIMARY KEY (musteri\_id, mulk\_id);

--

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

--

ALTER TABLE ONLY public.musteri\_siparisleri

ADD CONSTRAINT musteri\_siparisleri\_pkey PRIMARY KEY (siparis\_id);

--

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

--

ALTER TABLE ONLY public.musteriler

ADD CONSTRAINT musteriler\_pkey PRIMARY KEY (kisi\_id);

--

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

--

ALTER TABLE ONLY public.odeme\_yontemleri

ADD CONSTRAINT odeme\_yontemleri\_pkey PRIMARY KEY (id);

--

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

--

ALTER TABLE ONLY public.siparis\_durumu

ADD CONSTRAINT siparis\_durumu\_pkey PRIMARY KEY (durum\_id);

--

-- Name: siparisteki\_oge\_durum siparisteki\_oge\_durum\_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.siparisteki\_oge\_durum

ADD CONSTRAINT siparisteki\_oge\_durum\_pkey PRIMARY KEY (oge\_durum\_id);

--

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

--

ALTER TABLE ONLY public.siparisteki\_ogeler

ADD CONSTRAINT siparisteki\_ogeler\_pkey PRIMARY KEY (oge\_id);

--

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

--

ALTER TABLE ONLY public.urun\_kategorileri

ADD CONSTRAINT urun\_kategorileri\_pkey PRIMARY KEY (kategori\_id);

--

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

--

ALTER TABLE ONLY public.urunler

ADD CONSTRAINT urunler\_pkey PRIMARY KEY (id);

--

-- Name: kampanya\_musterileri kampanya\_trigger; Type: TRIGGER; Schema: public; Owner: postgres

--

CREATE TRIGGER kampanya\_trigger AFTER UPDATE ON public.kampanya\_musterileri FOR EACH ROW EXECUTE FUNCTION public.kampanya\_katilan();

--

-- Name: urunler kategori\_trigger; Type: TRIGGER; Schema: public; Owner: postgres

--

CREATE TRIGGER kategori\_trigger AFTER UPDATE ON public.urunler FOR EACH ROW EXECUTE FUNCTION public.kategori\_azalt();

--

-- Name: musteri\_siparisleri oge\_durum\_trigger; Type: TRIGGER; Schema: public; Owner: postgres

--

CREATE TRIGGER oge\_durum\_trigger AFTER UPDATE ON public.musteri\_siparisleri FOR EACH ROW EXECUTE FUNCTION public.oge\_durum\_degistir();

--

-- Name: siparisteki\_ogeler stok\_trigger; Type: TRIGGER; Schema: public; Owner: postgres

--

CREATE TRIGGER stok\_trigger AFTER UPDATE ON public.siparisteki\_ogeler FOR EACH ROW EXECUTE FUNCTION public.stok\_azalt();

--

-- Name: musteri\_adresleri adres\_tip\_FK; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.musteri\_adresleri

ADD CONSTRAINT "adres\_tip\_FK" FOREIGN KEY (adres\_tip\_id) REFERENCES public.adres\_tipleri(adres\_tip\_id) MATCH FULL ON UPDATE RESTRICT ON DELETE RESTRICT;

--

-- Name: kampanya\_musterileri kampanya\_id\_FK; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.kampanya\_musterileri

ADD CONSTRAINT "kampanya\_id\_FK" FOREIGN KEY (kampanya\_id) REFERENCES public.kampanyalar(kampanya\_id) MATCH FULL ON UPDATE RESTRICT ON DELETE RESTRICT;

--

-- Name: musteri\_siparisleri kargo\_secenekleri\_FK; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.musteri\_siparisleri

ADD CONSTRAINT "kargo\_secenekleri\_FK" FOREIGN KEY (kargo\_secenek\_id) REFERENCES public.kargo\_secenekleri(id) MATCH FULL ON UPDATE RESTRICT ON DELETE RESTRICT;

--

-- Name: adminler kisi\_admin\_cs; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.adminler

ADD CONSTRAINT kisi\_admin\_cs FOREIGN KEY (kisi\_id) REFERENCES public.kisiler(kisi\_id) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: musteriler kisi\_musteri\_FK; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.musteriler

ADD CONSTRAINT "kisi\_musteri\_FK" FOREIGN KEY (kisi\_id) REFERENCES public.kisiler(kisi\_id) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: musteri\_adresleri mulk\_id\_FK; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.musteri\_adresleri

ADD CONSTRAINT "mulk\_id\_FK" FOREIGN KEY (mulk\_id) REFERENCES public.mulkler(mulk\_id) MATCH FULL ON UPDATE RESTRICT ON DELETE RESTRICT;

--

-- Name: mulkler mulk\_tip\_FK; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.mulkler

ADD CONSTRAINT "mulk\_tip\_FK" FOREIGN KEY (mulk\_tip\_id) REFERENCES public.mulk\_tipleri(id) MATCH FULL ON UPDATE RESTRICT ON DELETE RESTRICT;

--

-- Name: musteri\_adresleri musteri\_adres\_FK; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.musteri\_adresleri

ADD CONSTRAINT "musteri\_adres\_FK" FOREIGN KEY (musteri\_id) REFERENCES public.musteriler(kisi\_id) MATCH FULL ON UPDATE RESTRICT ON DELETE RESTRICT;

--

-- Name: musteri\_siparisleri musteri\_id\_FK; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.musteri\_siparisleri

ADD CONSTRAINT "musteri\_id\_FK" FOREIGN KEY (musteri\_id) REFERENCES public.musteriler(kisi\_id) MATCH FULL ON UPDATE RESTRICT ON DELETE RESTRICT;

--

-- Name: kampanya\_musterileri musteri\_kampanya\_FK; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.kampanya\_musterileri

ADD CONSTRAINT "musteri\_kampanya\_FK" FOREIGN KEY (musteri\_id) REFERENCES public.musteriler(kisi\_id) MATCH FULL ON UPDATE RESTRICT ON DELETE RESTRICT;

--

-- Name: siparisteki\_ogeler musteri\_siparis\_FK; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.siparisteki\_ogeler

ADD CONSTRAINT "musteri\_siparis\_FK" FOREIGN KEY (siparis\_id) REFERENCES public.musteri\_siparisleri(siparis\_id) MATCH FULL ON UPDATE RESTRICT ON DELETE RESTRICT;

--

-- Name: musteriler odeme\_musteriler\_FK; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.musteriler

ADD CONSTRAINT "odeme\_musteriler\_FK" FOREIGN KEY (odeme\_yontemi\_no) REFERENCES public.odeme\_yontemleri(id) MATCH FULL ON UPDATE RESTRICT ON DELETE RESTRICT;

--

-- Name: musteri\_siparisleri siparis\_durum\_id\_FK; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.musteri\_siparisleri

ADD CONSTRAINT "siparis\_durum\_id\_FK" FOREIGN KEY (siparis\_durum\_id) REFERENCES public.siparis\_durumu(durum\_id) MATCH FULL ON UPDATE RESTRICT ON DELETE RESTRICT;

--

-- Name: siparisteki\_ogeler siparis\_oge\_durum\_FK; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.siparisteki\_ogeler

ADD CONSTRAINT "siparis\_oge\_durum\_FK" FOREIGN KEY (siparis\_oge\_durum\_id) REFERENCES public.siparisteki\_oge\_durum(oge\_durum\_id) MATCH FULL ON UPDATE RESTRICT ON DELETE RESTRICT;

--

-- Name: kampanya\_musterileri sonuc\_id\_FK; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.kampanya\_musterileri

ADD CONSTRAINT "sonuc\_id\_FK" FOREIGN KEY (sonuc\_id) REFERENCES public.kampanya\_sonuclar(sonuc\_id) MATCH FULL ON UPDATE RESTRICT ON DELETE RESTRICT;

--

-- Name: urunler urun\_kategori\_FK; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.urunler

ADD CONSTRAINT "urun\_kategori\_FK" FOREIGN KEY (urun\_kategori\_id) REFERENCES public.urun\_kategorileri(kategori\_id) MATCH FULL ON UPDATE RESTRICT ON DELETE RESTRICT;

--

-- Name: siparisteki\_ogeler urunler\_id\_FK; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.siparisteki\_ogeler

ADD CONSTRAINT "urunler\_id\_FK" FOREIGN KEY (urun\_id) REFERENCES public.urunler(id) MATCH FULL ON UPDATE RESTRICT ON DELETE RESTRICT;

--

-- PostgreSQL database dump complete

--

UYGULAMA KODLARI

Baglanti.Java

package com.company;  
import java.sql.\*;  
import java.util.Scanner;  
  
public class Baglanti {  
 private Connection baglan() throws SQLException {  
 */\*\*\* Bağlantı kurulumu \*\*/* Connection conn = DriverManager.*getConnection*("jdbc:postgresql://localhost:5432/e-ticaret-db",  
 "postgres", "Kocaeli123");  
 if (conn != null)  
 System.*out*.println("Veritabanına bağlandı!");  
 else  
 System.*out*.println("Bağlantı girişimi başarısız!");  
  
 return conn;  
 }  
  
 public void dogrula() throws SQLException {  
 Connection conn = baglan();  
  
 String sql= "SELECT \* FROM adminler";  
  
  
 Statement stmt = conn.createStatement();  
 ResultSet rs = stmt.executeQuery(sql);  
  
  
 conn.close();  
  
 Scanner sc = new Scanner(System.*in*);  
  
 String kontrolKullaniciAdi="";  
 String kontrolSifre ="";  
  
 int kontrol = 0;  
  
 while(kontrol !=1) {  
  
 System.*out*.println("Kullanici Adi Giriniz:");  
 String kullaniciAdi = sc.next();  
 System.*out*.println("Sifre Giriniz:");  
 String sifre = sc.next();  
  
  
 while(rs.next()) {  
 kontrolKullaniciAdi = rs.getString("kullanici\_adi");  
 kontrolSifre = rs.getString("sifre");  
  
  
 if (kontrolKullaniciAdi.equals(kullaniciAdi) && kontrolSifre.equals(sifre)) {  
 System.*out*.println("Dogrulandi");  
 kontrol = 1;  
 }  
  
 }  
  
 if(kontrol == 0)  
 System.*out*.println("Hatali sifre");  
  
 }  
  
  
  
 rs.close();  
 stmt.close();  
 }  
  
  
 public void tumUrunler() throws SQLException {  
  
 Connection conn = baglan();  
  
  
 String sqll= "SELECT \* FROM \"urunler\"";  
  
  
 Statement stmt = conn.createStatement();  
 ResultSet rs = stmt.executeQuery(sqll);  
  
 conn.close();  
  
 while(rs.next())  
 {  
  
 int urun\_id = rs.getInt("id");  
 String urun\_kategori\_id = rs.getString("urun\_kategori\_id");  
 String urun\_adi = rs.getString("urun\_adi");  
 String stok\_sayisi = rs.getString("stok\_sayisi");  
  
 System.*out*.println("\n\turun\_id : " + urun\_id);  
 System.*out*.println("\turun\_kategori\_id : " + urun\_kategori\_id);  
 System.*out*.println("\turun\_adi : " + urun\_adi);  
 System.*out*.println("\tstok\_sayisi : " + stok\_sayisi);  
 }  
  
 rs.close();  
 stmt.close();  
  
 }  
  
 public void urunAra() throws SQLException {  
  
 Connection conn = baglan();  
  
 System.*out*.println("Aranan urun no");  
 Scanner sc = new Scanner(System.*in*);  
  
 String arananUrun = sc.next();  
  
 String sqll= "SELECT \* FROM \"urunler\" WHERE \"id\" = "+arananUrun;  
  
  
 Statement stmt = conn.createStatement();  
 ResultSet rs = stmt.executeQuery(sqll);  
  
  
 conn.close();  
  
  
 while(rs.next())  
 {  
  
 int urun\_id = rs.getInt("id");  
 String urun\_kategori\_id = rs.getString("urun\_kategori\_id");  
 String urun\_adi = rs.getString("urun\_adi");  
 String stok\_sayisi = rs.getString("stok\_sayisi");  
  
 System.*out*.println("\turun\_id : " + urun\_id);  
 System.*out*.println("\turun\_kategori\_id : " + urun\_kategori\_id);  
 System.*out*.println("\turun\_adi : " + urun\_adi);  
 System.*out*.println("\tstok\_sayisi : " + stok\_sayisi);  
 }  
  
 rs.close();  
 stmt.close();  
  
 }  
  
 public void urunGuncelle() throws SQLException {  
  
 String urun\_id, degisecek\_deger, yeni\_deger;  
 System.*out*.println("Kac nolu urun degisecek");  
 Scanner sc = new Scanner(System.*in*);  
 urun\_id = sc.next();  
  
 System.*out*.println("Degisecek sutun adi?");  
 degisecek\_deger = sc.next();  
 System.*out*.println("Yeni Deger: ");  
 yeni\_deger = sc.next();  
  
 Connection conn = baglan();  
  
 String sql= "UPDATE \"urunler\" SET \"" +degisecek\_deger+ "\" = '" +yeni\_deger+ "' WHERE \"id\" = " +urun\_id;  
  
  
 Statement stmt = conn.createStatement();  
 int sonuc = stmt.executeUpdate(sql);  
  
 if(sonuc == 1)  
 System.*out*.println("Güncelleme basarili");  
 else  
 System.*out*.println("islem basarisiz oldu");  
  
  
 conn.close();  
  
  
 stmt.close();  
  
 }  
  
 public void urunSil() throws SQLException {  
  
 String urun\_id;  
 System.*out*.println("Kac nolu urun silinecek?");  
  
 Scanner sc = new Scanner(System.*in*);  
 urun\_id = sc.next();  
  
 Connection conn = baglan();  
  
 String sql= "DELETE FROM \"urunler\" WHERE \"id\" = " +urun\_id;  
  
  
 Statement stmt = conn.createStatement();  
 int sonuc = stmt.executeUpdate(sql);  
  
 if(sonuc == 1)  
 System.*out*.println("Silme basarili");  
 else  
 System.*out*.println("islem basarisiz oldu");  
  
  
 conn.close();  
  
  
 stmt.close();  
  
 }  
  
 public void urunEkle() throws SQLException {  
  
 String urun\_kategori\_id,urun\_adi, stok\_sayisi;  
 Scanner sc = new Scanner(System.*in*);  
  
 System.*out*.println("urun\_adi giriniz :");  
 urun\_adi = sc.nextLine();  
 System.*out*.println("urun\_kategori\_id giriniz :");  
 urun\_kategori\_id = sc.nextLine();  
 System.*out*.println("stok sayisi :");  
 stok\_sayisi = sc.nextLine();  
  
  
 Connection conn = baglan();  
  
 String sql= "INSERT INTO \"urunler\" (\"urun\_adi\", \"urun\_kategori\_id\", \"stok\_sayisi\")" +  
 "VALUES ('"+urun\_adi+"', '"+urun\_kategori\_id+"' , '"+stok\_sayisi+" ')";  
  
  
 Statement stmt = conn.createStatement();  
 int sonuc = stmt.executeUpdate(sql);  
  
 if(sonuc == 1)  
 System.*out*.println("Ekleme basarili");  
 else  
 System.*out*.println("islem basarisiz oldu");  
  
  
 conn.close();  
  
  
 stmt.close();  
  
 }  
}

Main.Java

package com.company;  
  
import java.sql.SQLException;  
import java.util.Scanner;  
  
public class Main {  
  
 public static void main(String[] args) throws SQLException {  
 Baglanti baglanti = new Baglanti();  
  
  
 System.*out*.println("Hosgeldiniz lutfen giris yapiniz");  
 // kullaniciGiris.dogrula();  
  
  
 Scanner sc = new Scanner(System.*in*);  
 int menusecim = 0;  
  
 baglanti.dogrula();  
  
 while(menusecim != 6)  
 { System.*out*.println("\n1-Tum Urunler");  
 System.*out*.println("2-Urun Ekle");  
 System.*out*.println("3-Urun Guncelle");  
 System.*out*.println("4-Urun Ara");  
 System.*out*.println("5-Urun Sil");  
 System.*out*.println("6-Cikis");  
  
 menusecim = sc.nextInt();  
 switch (menusecim){  
 case 1:  
 baglanti.tumUrunler();  
 break;  
  
 case 2:  
 baglanti.urunEkle();  
 break;  
  
 case 3:  
 baglanti.urunGuncelle();  
 break;  
  
 case 4:  
 baglanti.urunAra();  
 break;  
  
 case 5:  
 baglanti.urunSil();  
 break;  
  
 case 6:  
 break;  
 }  
 }  
  
  
  
 }  
}