

**T.C.**

**SAKARYA ÜNİVERSİTESİ**

**BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

Veritabanı Yönetim Sistemleri Proje Ödevi

Öğrencilerin Bilgileri

Adı: Emir  
Soyadı: VARDAR  
Öğrenci Numarası: G211210071  
Sınıf ve Şube: 2. Öğretim A  
E-posta Adresi: emir.vardar@ogr.sakarya.edu.tr

Adı: İsmail  
Soyadı: BAĞIŞ  
Öğrenci Numarası: B221210067  
Sınıf ve Şube: 1. Öğretim C  
E-posta Adresi: İsmail.bagis@ogr.sakarya.edu.tr

Uygulamanın Kısa Tanımı:

Yaptığımız uygulama c# form uygulaması üzerinde çalışır ve postgresql veri tabanını kullanır. Uygulamamız bir şirket yönetim uygulamasıdır. Yöneticilerin şirketleri ve şirketlerin de çalışanları, araçları, müşterileri vb. elemanları vardır.

İş Kuralları:

1. Yönetici(yonetici) bir veya birden fazla Şirkete(sirket) sahip olabilir.
2. Bir Şirket(sirket) bir ve yalnızca bir Yöneticiye(yonetici) ait olabilir.
3. Şirketler(sirket) bir veya birden fazla Şubeye(sube) sahip olabilir.
4. Bir Şube(sube) bir ve yalnızca bir Şirkete(sirket) ait olabilir.
5. Şirketler(sirket) bir veya birden fazla Otoparka(otopark) sahip olabilir.
6. Bir Otopark(otopark) bir ve yalnızca bir Yöneticiye(yonetici) sahip olabilir.
7. Şirketler(sirket) bir veya birden fazla Çalışana(calisan) sahip olabilir.
8. Bir Çalışan(calisan) bir ve yalnızca bir Şirkette(sirket) çalışabilir.
9. Bir Çalışan(calisan) ya tamirci ya temizlikelemani ya mudur ya da ofismemuru ya da sofor olmalıdır.
10. Çalışanların(calisan) maaşları, izin günleri, kilo ve boy bilgileri vardır.
11. Şirketler(sirket) bir veya birden fazla Tıra(tir) sahip olabilir.
12. Tırların(tir) kendilerine özel marka, model, plaka bilgileri vardır.
13. Bir Tır(tir) bir ve yalnızca bir Şirkete(sirket) ait olabilir.
14. Tırların(tir) bir veya birden fazla Şoförü(sofor) olabilir.
15. Bir Şoför(sofor) bir ve yalnızca bir Tırı(tir) sürebilir.
16. Şirketlerin(sirket) bir veya birden fazla Müşterisi(musteri) olabilir.
17. Bir Müşteri(musteri) bir ve yalnızca bir Şirkete (sirket) ait olabilir.
18. Müşteriler(musteri) bir veya daha fazla Sipariş(siprais) verebilir.
19. Bir Sipariş(siparis) bir ve yalnızca bir Müşteriye(musteri) ait olabilir.
20. Müşterilerin(musteri) bir veya daha fazla Faturası(fatura) olabilir.
21. Bir Fatura(fatura) bir ve yalnızca bir Müşteriye(müşteri) ait olabilir.
22. Müşterilerin(musteri) bir veya daha fazla İletişim(iletisim) olabilir.
23. Bir İletişim(iletisim) bir ve yalnızca bir Müşteriye(müşteri) ait olabilir.
24. Tırlar(tir) bir veya daha fazla Siparişi(siparis) taşıyabilir.
25. Bir Sipariş(siparis) bir ve yalnızca bir Tıra(tir) aittir.
26. Siparişlerin(siparis) sipariş tutarı, yükleme ve boşaltma yerleri bilgileri vardır.
27. Faturanın(fatura) fatura tutarı, fatura tarihi ve fatura etiket bilgileri vardır.
28. İletişimin(iletisim) iletişim tipi ve iletişim adresi bilgileri vardır.
29. Siparişler(siparis) bir ve yalnızca bir Faturaya(fatura) sahiptir.
30. Fatura(fatura) bir ve yalnızca bir Siparişe(siparis) ait olabilir.

Varlık Bağıntı Diyagramı

metin, diyagram, plan, teknik çizim içeren bir resim

Açıklama otomatik olarak oluşturuldu

İlişkisel Şema

* **yonetici (yonetici : integer**, yonetici\_adi : charvar(50)**)**
* **sirket(sirket\_id : integer**, sirket\_adi : charvar(50), sube\_sayisi : integer, otopark\_sayisi : integer, yonetici\_id : integer **)**
* **otopark (otopark\_id : integer**, otopark\_adi : charvar(50), sirket\_id : integer**)**
* **sube(sube\_id : integer**, sube\_adi : charvar(50),sirket\_id : integer**)**
* **calisan (calisan\_id : integer**, calisan\_adi : charvar(50), yas : integer, kilo : integer, sirket\_id : integer**)**
* **tamirci (calisan\_id : integer**, calisan\_adi : charvar(50), yas : integer, kilo : integer, sirket\_id : integer, tamir\_maas : integer, tamir\_izin : charvar(50)**)**
* **temizlikelemani (calisan\_id : integer**, calisan\_adi : charvar(50), yas : integer, kilo : integer, sirket\_id : integer, temizlik\_elemani\_maas : integer, temizlik\_elemani\_izin : charvar(50)**)**
* **mudur (calisan\_id : integer**, calisan\_adi : charvar(50), yas : integer, kilo : integer, sirket\_id : integer, mudur\_maas : integer, mudur\_izin : charvar(50)**)**
* **ofismemuru (calisan\_id : integer**, calisan\_adi : charvar(50), yas : integer, kilo : integer, sirket\_id : integer, ofis\_maas : integer, ofis\_izin : charvar(50)**)**
* **sofor (calisan\_id : integer**, calisan\_adi : charvar(50), yas : integer, kilo : integer, sirket\_id : integer, sofor\_maas : integer, sofor\_izin : charvar(50), tir\_id : integer**)**
* **tir (tir\_id : integer**, tir\_marka : charvar(50), tir\_model : charvar(50), tir\_plaka : charvar(50), sirket\_id : integer**)**
* **musteri(musteri\_id : integer**, musteri\_adi : charvar(50), sirket\_id : integer**)**
* **siparis (siparis\_id : integer**, siparis\_tarihi : text, musteri\_id : integer, siparis\_tutari : integer, yukleme : charvar(50), bosaltma : charvar(50), tir\_id : integer, musteri\_id : integer**)**
* **fatura (fatura\_id : integer**, fatura\_tutari : integer, fatura\_tarihi : text, musteri\_id : integer, fatura\_etiket : charvar(15),siparis\_id : integer**)**
* **iletisim (iletisim\_id : integer**, iletişim\_tipi : text, iletişim\_adresi : charvar(50),musteri\_id : integer**)**

**SQL KODLARI**

--

-- PostgreSQL database dump

--

-- Dumped from database version 15.5

-- Dumped by pg\_dump version 15.5

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: acilis(); Type: FUNCTION; Schema: public; Owner: postgres

--

CREATE FUNCTION public.acilis() RETURNS trigger

    LANGUAGE plpgsql

    AS $$

BEGIN

    INSERT INTO yonetici (yonetici\_id,yonetici\_adi, sirket\_acilis\_tarihi)

    VALUES (yonetici\_id,NEW.yonetici\_adi, CURRENT\_TIMESTAMP);

    RETURN NULL;

END;

$$;

ALTER FUNCTION public.acilis() OWNER TO postgres;

--

-- Name: calisan\_say(); Type: FUNCTION; Schema: public; Owner: postgres

--

CREATE FUNCTION public.calisan\_say() RETURNS integer

    LANGUAGE plpgsql

    AS $$

DECLARE

    total\_count INTEGER;

BEGIN

    -- Kişi sayısını bul

    SELECT COUNT(\*) INTO total\_count FROM calisan;

    -- Sonucu döndür

    RETURN total\_count;

END;

$$;

ALTER FUNCTION public.calisan\_say() OWNER TO postgres;

--

-- Name: mudur\_say(); Type: FUNCTION; Schema: public; Owner: postgres

--

CREATE FUNCTION public.mudur\_say() RETURNS integer

    LANGUAGE plpgsql

    AS $$

DECLARE

    total\_count INTEGER;

BEGIN

    -- Kişi sayısını bul

    SELECT COUNT(\*) INTO total\_count FROM mudur;

    -- Sonucu döndür

    RETURN total\_count;

END;

$$;

ALTER FUNCTION public.mudur\_say() OWNER TO postgres;

--

-- Name: ofismemuru\_say(); Type: FUNCTION; Schema: public; Owner: postgres

--

CREATE FUNCTION public.ofismemuru\_say() RETURNS integer

    LANGUAGE plpgsql

    AS $$

DECLARE

    total\_count INTEGER;

BEGIN

    -- Kişi sayısını bul

    SELECT COUNT(\*) INTO total\_count FROM ofismemuru;

    -- Sonucu döndür

    RETURN total\_count;

END;

$$;

ALTER FUNCTION public.ofismemuru\_say() OWNER TO postgres;

--

-- Name: otopark\_sayisi(); Type: FUNCTION; Schema: public; Owner: postgres

--

CREATE FUNCTION public.otopark\_sayisi() RETURNS trigger

    LANGUAGE plpgsql

    AS $$

BEGIN

    IF TG\_OP = 'INSERT' THEN

        UPDATE sirket

        SET otopark\_sayisi = otopark\_sayisi + 1

        WHERE sirket\_id = NEW.sirket\_id;

    ELSIF TG\_OP = 'DELETE' THEN

        UPDATE sirket

        SET otopark\_sayisi = otopark\_sayisi - 1

        WHERE sirket\_id = OLD.sirket\_id;

    END IF;

    RETURN NULL;

END;

$$;

ALTER FUNCTION public.otopark\_sayisi() OWNER TO postgres;

--

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

--

CREATE FUNCTION public.say() RETURNS trigger

    LANGUAGE plpgsql

    AS $$

begin

update "sirket" set sayi=sayi+1 from "sube" where "sirket"."sirket\_id"= public."sube"."sirket\_id";

return new;

end;

$$;

ALTER FUNCTION public.say() OWNER TO postgres;

--

-- Name: siparis\_say(); Type: FUNCTION; Schema: public; Owner: postgres

--

CREATE FUNCTION public.siparis\_say() RETURNS integer

    LANGUAGE plpgsql

    AS $$

DECLARE

    total\_count INTEGER;

BEGIN

    -- Kişi sayısını bul

    SELECT COUNT(\*) INTO total\_count FROM siparis;

    -- Sonucu döndür

    RETURN total\_count;

END;

$$;

ALTER FUNCTION public.siparis\_say() OWNER TO postgres;

--

-- Name: sirket\_acilis\_tarihi\_guncelle(); Type: FUNCTION; Schema: public; Owner: postgres

--

CREATE FUNCTION public.sirket\_acilis\_tarihi\_guncelle() RETURNS trigger

    LANGUAGE plpgsql

    AS $$

BEGIN

  UPDATE Yonetici

  SET sirket\_acilis\_tarihi = NEW.sirket\_acilis\_tarihi

  WHERE yonetici\_id = NEW.yonetici\_id;

  RETURN NEW;

END;

$$;

ALTER FUNCTION public.sirket\_acilis\_tarihi\_guncelle() OWNER TO postgres;

--

-- Name: sirket\_ekleme\_fonksiyonu(); Type: FUNCTION; Schema: public; Owner: postgres

--

CREATE FUNCTION public.sirket\_ekleme\_fonksiyonu() RETURNS trigger

    LANGUAGE plpgsql

    AS $$

BEGIN

    -- Yonetici tablosundan ilgili yonetici\_adi'nin sirket\_acilis\_tarihi alınıyor

    SELECT sirket\_acilis\_tarihi INTO NEW.sirket\_acilis\_tarihi

    FROM yonetici

    WHERE yonetici.yonetici\_id = NEW.yonetici\_id;

    RETURN NEW;

END;

$$;

ALTER FUNCTION public.sirket\_ekleme\_fonksiyonu() OWNER TO postgres;

--

-- Name: sofor\_say(); Type: FUNCTION; Schema: public; Owner: postgres

--

CREATE FUNCTION public.sofor\_say() RETURNS integer

    LANGUAGE plpgsql

    AS $$

DECLARE

    total\_count INTEGER;

BEGIN

    -- Kişi sayısını bul

    SELECT COUNT(\*) INTO total\_count FROM sofor;

    -- Sonucu döndür

    RETURN total\_count;

END;

$$;

ALTER FUNCTION public.sofor\_say() OWNER TO postgres;

--

-- Name: sube\_say(); Type: FUNCTION; Schema: public; Owner: postgres

--

CREATE FUNCTION public.sube\_say() RETURNS trigger

    LANGUAGE plpgsql

    AS $$

begin

update sirket set sube\_sayisi=sube\_sayisi+1;

return new;

end;

$$;

ALTER FUNCTION public.sube\_say() OWNER TO postgres;

--

-- Name: sube\_sayisi(); Type: FUNCTION; Schema: public; Owner: postgres

--

CREATE FUNCTION public.sube\_sayisi() RETURNS trigger

    LANGUAGE plpgsql

    AS $$

BEGIN

    IF TG\_OP = 'INSERT' THEN

        UPDATE sirket

        SET sube\_sayisi = sube\_sayisi + 1

        WHERE sirket\_id = NEW.sirket\_id;

    ELSIF TG\_OP = 'DELETE' THEN

        UPDATE sirket

        SET sube\_sayisi = sube\_sayisi - 1

        WHERE sirket\_id = OLD.sirket\_id;

    END IF;

    RETURN NULL;

END;

$$;

ALTER FUNCTION public.sube\_sayisi() OWNER TO postgres;

--

-- Name: tamirci\_say(); Type: FUNCTION; Schema: public; Owner: postgres

--

CREATE FUNCTION public.tamirci\_say() RETURNS integer

    LANGUAGE plpgsql

    AS $$

DECLARE

    total\_count INTEGER;

BEGIN

    -- Kişi sayısını bul

    SELECT COUNT(\*) INTO total\_count FROM tamirci;

    -- Sonucu döndür

    RETURN total\_count;

END;

$$;

ALTER FUNCTION public.tamirci\_say() OWNER TO postgres;

--

-- Name: temizlikelemani\_say(); Type: FUNCTION; Schema: public; Owner: postgres

--

CREATE FUNCTION public.temizlikelemani\_say() RETURNS integer

    LANGUAGE plpgsql

    AS $$

DECLARE

    total\_count INTEGER;

BEGIN

    -- Kişi sayısını bul

    SELECT COUNT(\*) INTO total\_count FROM temizlikelemani;

    -- Sonucu döndür

    RETURN total\_count;

END;

$$;

ALTER FUNCTION public.temizlikelemani\_say() OWNER TO postgres;

--

-- Name: update\_sube\_say(); Type: FUNCTION; Schema: public; Owner: postgres

--

CREATE FUNCTION public.update\_sube\_say() RETURNS trigger

    LANGUAGE plpgsql

    AS $$

BEGIN

  UPDATE sube

  SET sube\_sayisi = sube\_sayisi + 1;

  RETURN NEW;

END;

$$;

ALTER FUNCTION public.update\_sube\_say() OWNER TO postgres;

SET default\_tablespace = '';

SET default\_table\_access\_method = heap;

--

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

--

CREATE TABLE public.calisan (

    calisan\_id integer NOT NULL,

    calisan\_adi character varying(50) NOT NULL,

    sirket\_id integer NOT NULL,

    yas integer,

    kilo integer

);

ALTER TABLE public.calisan OWNER TO postgres;

--

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

--

CREATE TABLE public.fatura (

    fatura\_id integer NOT NULL,

    fatura\_tutari integer NOT NULL,

    fatura\_etiket text NOT NULL,

    fatura\_tarihi text,

    musteri\_id integer NOT NULL,

    siparis\_id integer NOT NULL

);

ALTER TABLE public.fatura OWNER TO postgres;

--

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

--

CREATE TABLE public.iletisim (

    iletisim\_id integer NOT NULL,

    iletisim\_tipi text NOT NULL,

    iletisim\_adresi character varying NOT NULL,

    musteri\_id integer NOT NULL

);

ALTER TABLE public.iletisim OWNER TO postgres;

--

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

--

CREATE TABLE public.mudur (

    calisan\_id integer,

    mudur\_maas integer,

    mudur\_izin character varying

)

INHERITS (public.calisan);

ALTER TABLE public.mudur OWNER TO postgres;

--

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

--

CREATE TABLE public.musteri (

    musteri\_id integer NOT NULL,

    musteri\_adi character varying(50) NOT NULL,

    sirket\_id integer NOT NULL

);

ALTER TABLE public.musteri OWNER TO postgres;

--

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

--

CREATE TABLE public.ofismemuru (

    calisan\_id integer,

    ofis\_maas integer,

    ofis\_izin character varying

)

INHERITS (public.calisan);

ALTER TABLE public.ofismemuru OWNER TO postgres;

--

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

--

CREATE TABLE public.otopark (

    otopark\_id integer NOT NULL,

    otopark\_adi character varying(50) NOT NULL,

    sirket\_id integer NOT NULL

);

ALTER TABLE public.otopark OWNER TO postgres;

--

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

--

CREATE TABLE public.siparis (

    siparis\_id integer NOT NULL,

    musteri\_id integer NOT NULL,

    tir\_id integer NOT NULL,

    siparis\_tutari integer NOT NULL,

    yukleme character varying NOT NULL,

    bosaltma character varying NOT NULL,

    siparis\_tarihi text,

    siparis\_bilgi text

);

ALTER TABLE public.siparis OWNER TO postgres;

--

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

--

CREATE TABLE public.sirket (

    sirket\_id integer NOT NULL,

    sirket\_adi character varying(50) NOT NULL,

    yonetici\_id integer NOT NULL,

    sube\_sayisi integer DEFAULT 0,

    otopark\_sayisi integer DEFAULT 0

);

ALTER TABLE public.sirket OWNER TO postgres;

--

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

--

CREATE TABLE public.sofor (

    calisan\_id integer,

    tir\_id integer NOT NULL,

    sofor\_maas integer,

    sofor\_izin character varying(15)

)

INHERITS (public.calisan);

ALTER TABLE public.sofor OWNER TO postgres;

--

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

--

CREATE TABLE public.sube (

    sube\_id integer NOT NULL,

    sube\_adi character varying(50) NOT NULL,

    sirket\_id integer NOT NULL

);

ALTER TABLE public.sube OWNER TO postgres;

--

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

--

CREATE TABLE public.tamirci (

    calisan\_id integer,

    tamir\_maas integer,

    tamir\_izin character varying

)

INHERITS (public.calisan);

ALTER TABLE public.tamirci OWNER TO postgres;

--

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

--

CREATE TABLE public.temizlikelemani (

    calisan\_id integer,

    temizlik\_elemani\_maas integer,

    temizlik\_elemani\_izin character varying

)

INHERITS (public.calisan);

ALTER TABLE public.temizlikelemani OWNER TO postgres;

--

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

--

CREATE TABLE public.tir (

    tir\_id integer NOT NULL,

    tir\_marka character varying(50) NOT NULL,

    tir\_model character varying(50) NOT NULL,

    tir\_plaka character varying(50) NOT NULL,

    sirket\_id integer NOT NULL

);

ALTER TABLE public.tir OWNER TO postgres;

--

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

--

CREATE TABLE public.yonetici (

    yonetici\_id integer NOT NULL,

    yonetici\_adi character varying(50) NOT NULL

);

ALTER TABLE public.yonetici OWNER TO postgres;

--

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

--

COPY public.calisan (calisan\_id, calisan\_adi, sirket\_id, yas, kilo) FROM stdin;

\.

--

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

--

COPY public.fatura (fatura\_id, fatura\_tutari, fatura\_etiket, fatura\_tarihi, musteri\_id, siparis\_id) FROM stdin;

23  34525   ALİ 1   234523  23  3

24  4545    ALİ 1   545 23  3

\.

--

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

--

COPY public.iletisim (iletisim\_id, iletisim\_tipi, iletisim\_adresi, musteri\_id) FROM stdin;

\.

--

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

--

COPY public.mudur (calisan\_id, calisan\_adi, sirket\_id, yas, kilo, mudur\_maas, mudur\_izin) FROM stdin;

6   emir    1   12  12  12  12

7   emir    1   12  12  12  12

\.

--

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

--

COPY public.musteri (musteri\_id, musteri\_adi, sirket\_id) FROM stdin;

23  ALİ TAVUZ   1

233 HİHMET AYAZ 1

2334    AHMET AKAN  1

\.

--

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

--

COPY public.ofismemuru (calisan\_id, calisan\_adi, sirket\_id, yas, kilo, ofis\_maas, ofis\_izin) FROM stdin;

\.

--

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

--

COPY public.otopark (otopark\_id, otopark\_adi, sirket\_id) FROM stdin;

1   sad 1

2   dsadsa  2

3   sadasd  1

45  KUBİ PARK   5

\.

--

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

--

COPY public.siparis (siparis\_id, musteri\_id, tir\_id, siparis\_tutari, yukleme, bosaltma, siparis\_tarihi, siparis\_bilgi) FROM stdin;

3   23  2   23000   İSTANBUL    HATAY   30.10.2022  EŞYA

1   233 67  24554   ANKARA  MRDİN   30.10.2020  HALI

\.

--

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

--

COPY public.sirket (sirket\_id, sirket\_adi, yonetici\_id, sube\_sayisi, otopark\_sayisi) FROM stdin;

4   asas    9   0   0

3   VARDAR LOJİSTİK 6   1   0

2   x lojsitik  8   0   1

5   GREEN FAST  25  0   1

1   MARDEP LOJİSTİK 8   0   2

\.

--

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

--

COPY public.sofor (calisan\_id, calisan\_adi, sirket\_id, yas, kilo, tir\_id, sofor\_maas, sofor\_izin) FROM stdin;

8   ÜNAL BULUT  3   23  90  67  90000   SALI

\.

--

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

--

COPY public.sube (sube\_id, sube\_adi, sirket\_id) FROM stdin;

3   da  3

\.

--

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

--

COPY public.tamirci (calisan\_id, calisan\_adi, sirket\_id, yas, kilo, tamir\_maas, tamir\_izin) FROM stdin;

1   emir    1   12  12  12  12

3   emir    1   12  12  12  12

\.

--

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

--

COPY public.temizlikelemani (calisan\_id, calisan\_adi, sirket\_id, yas, kilo, temizlik\_elemani\_maas, temizlik\_elemani\_izin) FROM stdin;

2   emir    1   12  12  12  12

4   emir    1   12  12  12  12

5   emir    1   12  12  12  12

\.

--

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

--

COPY public.tir (tir\_id, tir\_marka, tir\_model, tir\_plaka, sirket\_id) FROM stdin;

2   MERCEDES    AXOR    47 ACH 6757 1

3   FORD    1848    47 AG 720   1

67  MERCEDES    AXOR    33 ADT 150  1

\.

--

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

--

COPY public.yonetici (yonetici\_id, yonetici\_adi) FROM stdin;

6   EMİR VARDAR

11  AHMET BAĞIŞ

9   ALİ VARDAR

7   MEHMET VARDAR

3   ABDULKADİR BAĞIŞ

1   MURAT BAĞIŞ

8   MUSTAFA VARDAR

25  KUBİLAY KÖZLEME

\.

--

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

--

ALTER TABLE ONLY public.calisan

    ADD CONSTRAINT calisan\_pkey PRIMARY KEY (calisan\_id);

--

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

--

ALTER TABLE ONLY public.fatura

    ADD CONSTRAINT fatura\_pkey PRIMARY KEY (fatura\_id);

--

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

--

ALTER TABLE ONLY public.iletisim

    ADD CONSTRAINT iletisim\_pkey PRIMARY KEY (iletisim\_id);

--

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

--

ALTER TABLE ONLY public.mudur

    ADD CONSTRAINT mudur\_pkey PRIMARY KEY (calisan\_id);

--

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

--

ALTER TABLE ONLY public.musteri

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

--

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

--

ALTER TABLE ONLY public.ofismemuru

    ADD CONSTRAINT ofismemuru\_pkey PRIMARY KEY (calisan\_id);

--

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

--

ALTER TABLE ONLY public.otopark

    ADD CONSTRAINT otopark\_pkey PRIMARY KEY (otopark\_id);

--

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

--

ALTER TABLE ONLY public.siparis

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

--

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

--

ALTER TABLE ONLY public.sirket

    ADD CONSTRAINT sirket\_pkey PRIMARY KEY (sirket\_id);

--

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

--

ALTER TABLE ONLY public.sofor

    ADD CONSTRAINT sofor\_pkey PRIMARY KEY (calisan\_id);

--

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

--

ALTER TABLE ONLY public.sube

    ADD CONSTRAINT sube\_pkey PRIMARY KEY (sube\_id);

--

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

--

ALTER TABLE ONLY public.tamirci

    ADD CONSTRAINT tamirci\_pkey PRIMARY KEY (calisan\_id);

--

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

--

ALTER TABLE ONLY public.temizlikelemani

    ADD CONSTRAINT temizlikelemani\_pkey PRIMARY KEY (calisan\_id);

--

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

--

ALTER TABLE ONLY public.tir

    ADD CONSTRAINT tir\_pkey PRIMARY KEY (tir\_id);

--

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

--

ALTER TABLE ONLY public.yonetici

    ADD CONSTRAINT yonetici\_pkey PRIMARY KEY (yonetici\_id);

--

-- Name: fki\_ALTER TABLE IF EXISTS public.fatura     ADD CONSTRAINT fatu; Type: INDEX; Schema: public; Owner: postgres

--

CREATE INDEX "fki\_ALTER TABLE IF EXISTS public.fatura     ADD CONSTRAINT fatu" ON public.sirket USING btree (yonetici\_id);

--

-- Name: fki\_c; Type: INDEX; Schema: public; Owner: postgres

--

CREATE INDEX fki\_c ON public.calisan USING btree (sirket\_id);

--

-- Name: fki\_fatura\_musteri\_id\_fkey; Type: INDEX; Schema: public; Owner: postgres

--

CREATE INDEX fki\_fatura\_musteri\_id\_fkey ON public.fatura USING btree (fatura\_id);

--

-- Name: fki\_fatura\_siparis\_id\_fkey; Type: INDEX; Schema: public; Owner: postgres

--

CREATE INDEX fki\_fatura\_siparis\_id\_fkey ON public.fatura USING btree (siparis\_id);

--

-- Name: fki\_iletisim\_musteri\_id\_fkey; Type: INDEX; Schema: public; Owner: postgres

--

CREATE INDEX fki\_iletisim\_musteri\_id\_fkey ON public.iletisim USING btree (musteri\_id);

--

-- Name: fki\_mudur\_calisan\_id\_fkey; Type: INDEX; Schema: public; Owner: postgres

--

CREATE INDEX fki\_mudur\_calisan\_id\_fkey ON public.mudur USING btree (calisan\_id);

--

-- Name: fki\_musteri\_sirket\_id\_fkey; Type: INDEX; Schema: public; Owner: postgres

--

CREATE INDEX fki\_musteri\_sirket\_id\_fkey ON public.musteri USING btree (sirket\_id);

--

-- Name: fki\_ofissmemuru\_calisan\_id\_fkey; Type: INDEX; Schema: public; Owner: postgres

--

CREATE INDEX fki\_ofissmemuru\_calisan\_id\_fkey ON public.ofismemuru USING btree (calisan\_id);

--

-- Name: fki\_otopark\_sirket\_id\_fkey; Type: INDEX; Schema: public; Owner: postgres

--

CREATE INDEX fki\_otopark\_sirket\_id\_fkey ON public.otopark USING btree (sirket\_id);

--

-- Name: fki\_siparis\_musteri\_id\_fkey; Type: INDEX; Schema: public; Owner: postgres

--

CREATE INDEX fki\_siparis\_musteri\_id\_fkey ON public.siparis USING btree (musteri\_id);

--

-- Name: fki\_siparis\_tir\_id\_fkey; Type: INDEX; Schema: public; Owner: postgres

--

CREATE INDEX fki\_siparis\_tir\_id\_fkey ON public.siparis USING btree (tir\_id);

--

-- Name: fki\_sofor\_calisan\_id\_fkey; Type: INDEX; Schema: public; Owner: postgres

--

CREATE INDEX fki\_sofor\_calisan\_id\_fkey ON public.sofor USING btree (calisan\_id);

--

-- Name: fki\_sofor\_tir\_id\_fkey; Type: INDEX; Schema: public; Owner: postgres

--

CREATE INDEX fki\_sofor\_tir\_id\_fkey ON public.sofor USING btree (tir\_id);

--

-- Name: fki\_sube\_sirket\_id\_fkey; Type: INDEX; Schema: public; Owner: postgres

--

CREATE INDEX fki\_sube\_sirket\_id\_fkey ON public.sube USING btree (sirket\_id);

--

-- Name: fki\_tamirci\_calisan\_id\_fkey; Type: INDEX; Schema: public; Owner: postgres

--

CREATE INDEX fki\_tamirci\_calisan\_id\_fkey ON public.tamirci USING btree (calisan\_id);

--

-- Name: fki\_temizlikelemani\_calisan\_id\_fkey; Type: INDEX; Schema: public; Owner: postgres

--

CREATE INDEX fki\_temizlikelemani\_calisan\_id\_fkey ON public.temizlikelemani USING btree (calisan\_id);

--

-- Name: fki\_tir\_sirket\_id\_fkey; Type: INDEX; Schema: public; Owner: postgres

--

CREATE INDEX fki\_tir\_sirket\_id\_fkey ON public.tir USING btree (sirket\_id);

--

-- Name: otopark otopark\_after\_delete; Type: TRIGGER; Schema: public; Owner: postgres

--

CREATE TRIGGER otopark\_after\_delete AFTER DELETE ON public.otopark FOR EACH ROW EXECUTE FUNCTION public.otopark\_sayisi();

--

-- Name: otopark otopark\_after\_insert; Type: TRIGGER; Schema: public; Owner: postgres

--

CREATE TRIGGER otopark\_after\_insert AFTER INSERT ON public.otopark FOR EACH ROW EXECUTE FUNCTION public.otopark\_sayisi();

--

-- Name: sube sube\_after\_delete; Type: TRIGGER; Schema: public; Owner: postgres

--

CREATE TRIGGER sube\_after\_delete AFTER DELETE ON public.sube FOR EACH ROW EXECUTE FUNCTION public.sube\_sayisi();

--

-- Name: sube sube\_after\_insert; Type: TRIGGER; Schema: public; Owner: postgres

--

CREATE TRIGGER sube\_after\_insert AFTER INSERT ON public.sube FOR EACH ROW EXECUTE FUNCTION public.sube\_sayisi();

--

-- Name: calisan calisan\_sirkt\_id\_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.calisan

    ADD CONSTRAINT calisan\_sirkt\_id\_fkey FOREIGN KEY (sirket\_id) REFERENCES public.sirket(sirket\_id) ON UPDATE CASCADE ON DELETE CASCADE NOT VALID;

--

-- Name: fatura fatura\_musteri\_id\_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.fatura

    ADD CONSTRAINT fatura\_musteri\_id\_fkey FOREIGN KEY (musteri\_id) REFERENCES public.musteri(musteri\_id) ON UPDATE CASCADE ON DELETE CASCADE NOT VALID;

--

-- Name: fatura fatura\_siparis\_id\_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.fatura

    ADD CONSTRAINT fatura\_siparis\_id\_fkey FOREIGN KEY (siparis\_id) REFERENCES public.siparis(siparis\_id) ON UPDATE CASCADE ON DELETE CASCADE NOT VALID;

--

-- Name: iletisim iletisim\_musteri\_id\_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.iletisim

    ADD CONSTRAINT iletisim\_musteri\_id\_fkey FOREIGN KEY (musteri\_id) REFERENCES public.musteri(musteri\_id) ON UPDATE CASCADE ON DELETE CASCADE NOT VALID;

--

-- Name: mudur mudur\_calisan\_id\_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.mudur

    ADD CONSTRAINT mudur\_calisan\_id\_fkey FOREIGN KEY (calisan\_id) REFERENCES public.mudur(calisan\_id) ON UPDATE CASCADE ON DELETE CASCADE NOT VALID;

--

-- Name: musteri musteri\_sirket\_id\_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.musteri

    ADD CONSTRAINT musteri\_sirket\_id\_fkey FOREIGN KEY (sirket\_id) REFERENCES public.sirket(sirket\_id) ON UPDATE CASCADE ON DELETE CASCADE NOT VALID;

--

-- Name: ofismemuru ofissmemuru\_calisan\_id\_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.ofismemuru

    ADD CONSTRAINT ofissmemuru\_calisan\_id\_fkey FOREIGN KEY (calisan\_id) REFERENCES public.ofismemuru(calisan\_id) ON UPDATE CASCADE ON DELETE CASCADE NOT VALID;

--

-- Name: otopark otopark\_sirket\_id\_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.otopark

    ADD CONSTRAINT otopark\_sirket\_id\_fkey FOREIGN KEY (sirket\_id) REFERENCES public.sirket(sirket\_id) ON DELETE CASCADE NOT VALID;

--

-- Name: siparis siparis\_musteri\_id\_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.siparis

    ADD CONSTRAINT siparis\_musteri\_id\_fkey FOREIGN KEY (musteri\_id) REFERENCES public.musteri(musteri\_id) ON UPDATE CASCADE ON DELETE CASCADE NOT VALID;

--

-- Name: siparis siparis\_tir\_id\_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.siparis

    ADD CONSTRAINT siparis\_tir\_id\_fkey FOREIGN KEY (tir\_id) REFERENCES public.tir(tir\_id) ON UPDATE CASCADE ON DELETE CASCADE NOT VALID;

--

-- Name: sirket sirket\_yonetici\_id\_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.sirket

    ADD CONSTRAINT sirket\_yonetici\_id\_fkey FOREIGN KEY (yonetici\_id) REFERENCES public.yonetici(yonetici\_id) ON UPDATE CASCADE ON DELETE CASCADE NOT VALID;

--

-- Name: sofor sofor\_calisan\_id\_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.sofor

    ADD CONSTRAINT sofor\_calisan\_id\_fkey FOREIGN KEY (calisan\_id) REFERENCES public.sofor(calisan\_id) ON UPDATE CASCADE ON DELETE CASCADE NOT VALID;

--

-- Name: sofor sofor\_tir\_id\_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.sofor

    ADD CONSTRAINT sofor\_tir\_id\_fkey FOREIGN KEY (tir\_id) REFERENCES public.tir(tir\_id) ON UPDATE CASCADE ON DELETE CASCADE NOT VALID;

--

-- Name: sube sube\_sirket\_id\_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.sube

    ADD CONSTRAINT sube\_sirket\_id\_fkey FOREIGN KEY (sirket\_id) REFERENCES public.sirket(sirket\_id) ON DELETE CASCADE NOT VALID;

--

-- Name: tamirci tamirci\_calisan\_id\_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.tamirci

    ADD CONSTRAINT tamirci\_calisan\_id\_fkey FOREIGN KEY (calisan\_id) REFERENCES public.tamirci(calisan\_id) ON UPDATE CASCADE ON DELETE CASCADE NOT VALID;

--

-- Name: temizlikelemani temizlikelemani\_calisan\_id\_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.temizlikelemani

    ADD CONSTRAINT temizlikelemani\_calisan\_id\_fkey FOREIGN KEY (calisan\_id) REFERENCES public.temizlikelemani(calisan\_id) ON UPDATE CASCADE ON DELETE CASCADE NOT VALID;

--

-- Name: tir tir\_sirket\_id\_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.tir

    ADD CONSTRAINT tir\_sirket\_id\_fkey FOREIGN KEY (sirket\_id) REFERENCES public.sirket(sirket\_id) ON UPDATE CASCADE ON DELETE CASCADE NOT VALID;

--

-- PostgreSQL database dump complete

--

**Fonksiyonlar ve Triggerlar**

--çalışan eleman sayısını buldurduğumuz fonksiyon--

CREATE FUNCTION public.calisan\_say() RETURNS integer

    LANGUAGE plpgsql

    AS $$

DECLARE

    total\_count INTEGER;

BEGIN

    -- Kişi sayısını bul

    SELECT COUNT(\*) INTO total\_count FROM calisan;

    -- Sonucu döndür

    RETURN total\_count;

END;

$$;

--müdür sayısını buldurduğumuz fonskiyon--

CREATE FUNCTION public.mudur\_say() RETURNS integer

    LANGUAGE plpgsql

    AS $$

DECLARE

    total\_count INTEGER;

BEGIN

    -- Kişi sayısını bul

    SELECT COUNT(\*) INTO total\_count FROM mudur;

    -- Sonucu döndür

    RETURN total\_count;

END;

$$;

--ofismemuru sayısını buldurduğumuz fonksiyon--

CREATE FUNCTION public.siparis\_say() RETURNS integer

    LANGUAGE plpgsql

    AS $$

DECLARE

    total\_count INTEGER;

BEGIN

    -- Kişi sayısını bul

    SELECT COUNT(\*) INTO total\_count FROM siparis;

    -- Sonucu döndür

    RETURN total\_count;

END;

$$;

--sipariş sayısını buldurduğumuz --

CREATE FUNCTION public.siparis\_say() RETURNS integer

    LANGUAGE plpgsql

    AS $$

DECLARE

    total\_count INTEGER;

BEGIN

    -- Kişi sayısını bul

    SELECT COUNT(\*) INTO total\_count FROM siparis;

    -- Sonucu döndür

    RETURN total\_count;

END;

$$;

--sofor sayısını bulduğumuz fonksiyon--

CREATE FUNCTION public.sofor\_say() RETURNS integer

    LANGUAGE plpgsql

    AS $$

DECLARE

    total\_count INTEGER;

BEGIN

    -- Kişi sayısını bul

    SELECT COUNT(\*) INTO total\_count FROM sofor;

    -- Sonucu döndür

    RETURN total\_count;

END;

$$;

--tamirci sayısını bulduğumuz fonskiyon--

CREATE FUNCTION public.tamirci\_say() RETURNS integer

    LANGUAGE plpgsql

    AS $$

DECLARE

    total\_count INTEGER;

BEGIN

    -- Kişi sayısını bul

    SELECT COUNT(\*) INTO total\_count FROM tamirci;

    -- Sonucu döndür

    RETURN total\_count;

END;

$$;

--temizlikelemani sayıısını bulduruduğumuz fonksiyon--

CREATE FUNCTION public.temizlikelemani\_say() RETURNS integer

    LANGUAGE plpgsql

    AS $$

DECLARE

    total\_count INTEGER;

BEGIN

    -- Kişi sayısını bul

    SELECT COUNT(\*) INTO total\_count FROM temizlikelemani;

    -- Sonucu döndür

    RETURN total\_count;

END;

$$;

--Şube sayısını bulurduğumuz fonskiyon ve altında triggerleri--

-- Trigger oluştur

CREATE OR REPLACE FUNCTION sube\_sayisi()

RETURNS TRIGGER AS $$

BEGIN

    --insert işlemi yani ekleme yaptığımızda çalışacak kod

    IF TG\_OP = 'INSERT' THEN

        UPDATE sirket

        SET sube\_sayisi = sube\_sayisi + 1

        WHERE sirket\_id = NEW.sirket\_id;

    -- delete işlemi yani silme yaptığımızda çalışacak kod

    ELSIF TG\_OP = 'DELETE' THEN

        UPDATE sirket

        SET sube\_sayisi = sube\_sayisi - 1

        WHERE sirket\_id = OLD.sirket\_id;

    END IF;

    RETURN NULL;

END;

$$ LANGUAGE plpgsql;

--1. trigger

-- Trigger'ı Şube tablosuna bağla

CREATE TRIGGER sube\_after\_insert

AFTER INSERT ON sube -- sube tablosuna yapılan her eklemeden sonra

FOR EACH ROW

EXECUTE FUNCTION sube\_sayisi();--sube\_sayisi fonksiyonunu çalıştırır

--2.trigger

-- Trigger'ı Şube tablosuna bağla

CREATE TRIGGER sube\_after\_delete

AFTER DELETE ON sube --sube tablosunda yapılan her silme işleminden sonra

FOR EACH ROW

EXECUTE FUNCTION sube\_sayisi(); --sube\_sayisi fonksiyonunu fonksiyonunu çalıştırır

--otopark sayısını hespalattığımız fonskiyon ve triggerleri--

-- Trigger oluştur

CREATE OR REPLACE FUNCTION otopark\_sayisi()

RETURNS TRIGGER AS $$

BEGIN

    --insert işlemi yani ekleme yaptığımızda çalışacak kod

    IF TG\_OP = 'INSERT' THEN

        UPDATE sirket

        SET otopark\_sayisi = otopark\_sayisi + 1

        WHERE sirket\_id = NEW.sirket\_id;

    -- delete işlemi yani silme yaptığımızda çalışacak kod

    ELSIF TG\_OP = 'DELETE' THEN

        UPDATE sirket

        SET otopark\_sayisi = otopark\_sayisi - 1

        WHERE sirket\_id = OLD.sirket\_id;

    END IF;

    RETURN NULL;

END;

$$ LANGUAGE plpgsql;

--1. trigger

-- Trigger'ı Şube tablosuna bağla

CREATE TRIGGER otopark\_after\_insert

AFTER INSERT ON otopark --otopark tablosuna her ekleme işlemi yaptıktan sonra

FOR EACH ROW

EXECUTE FUNCTION otopark\_sayisi();--bu fonksiyonu çalıştırır

--2.trigger

-- Trigger'ı Şube tablosuna bağla

CREATE TRIGGER otopark\_after\_delete

AFTER DELETE ON otopark--otopark tablosuna her silme işlemiy aptığımızda

FOR EACH ROW

EXECUTE FUNCTION otopark\_sayisi();--bu fonksiyonu çalışıtırır

UYGULAMAYA AİT İŞLEMLERİN EKRAN GÖRÜNTÜLERİ

ADMİN PANELİ



metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturulduEKLEME (Bazı tablolar)

metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

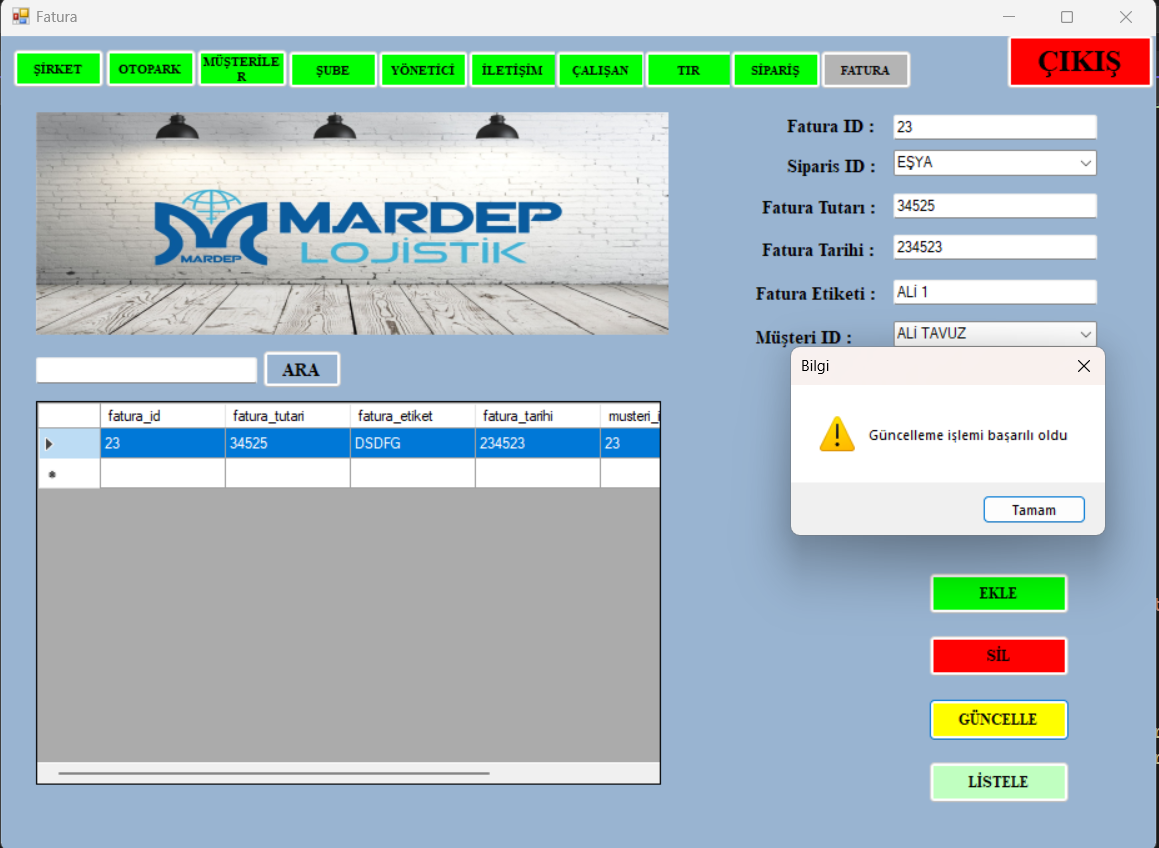
Açıklama otomatik olarak oluşturulduSİLME (Bazı tablolar)

metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

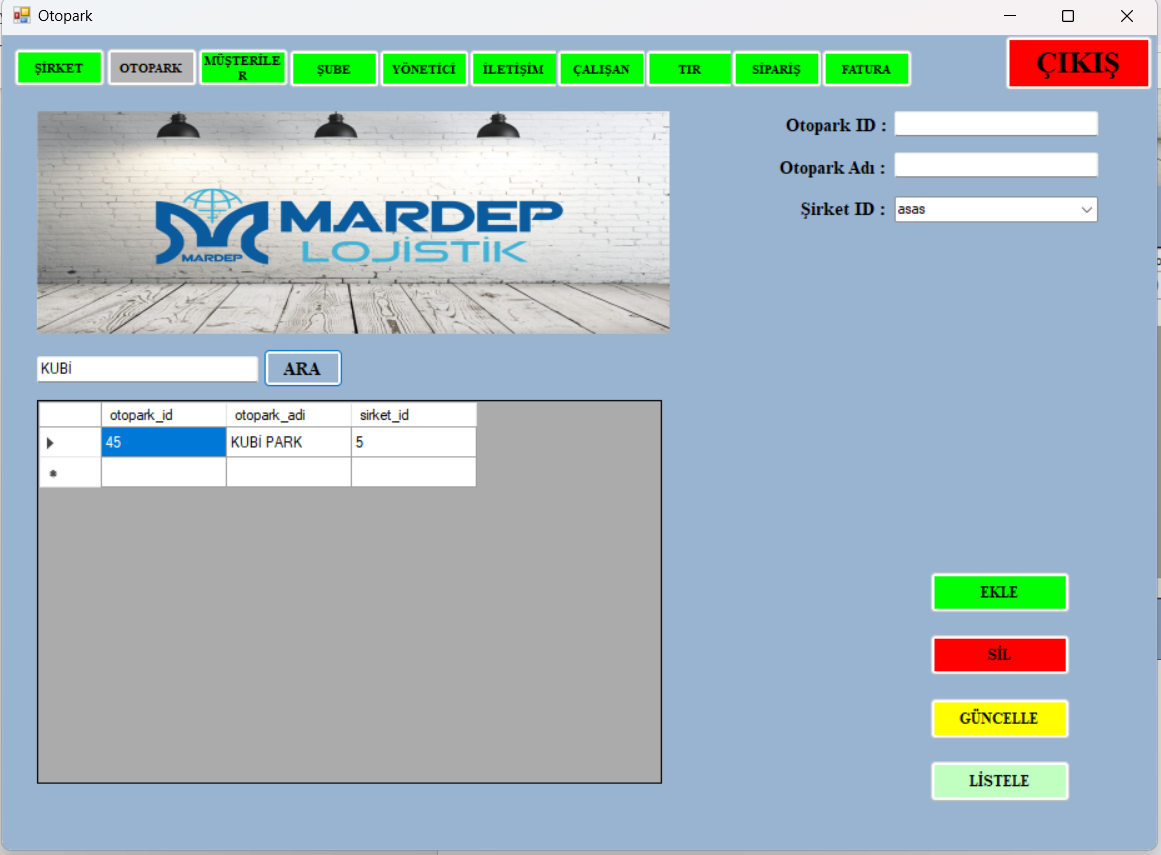
Açıklama otomatik olarak oluşturuldu

metin, yazılım, bilgisayar simgesi, web sayfası içeren bir resim

Açıklama otomatik olarak oluşturulduGÜNCELLEME (Bazı tablolar)

metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu ARAMA (Bazı tablolar)

Kaynak Kodlar

<https://github.com/ismailbagis/VTYS-ODEV>

https://github.com/EmirVardar/VTYS-ODEV