VERİTABANI PROJE ÖDEVİ

AD: MUSTAFA MELİH

SOYAD: TÜFEKCİOĞLU

NUMARA: B191210004

GRUP: 1.ÖĞRETİM / C GRUBU

ÖĞRETİM GÖREVLİSİ: Prof. Dr. CELAL ÇEKEN

UYGULAMANIN TANITIMI:

Yapmış olduğum bu uygulamada bir futbol klübünün personel bilgilerini ve mali durum bilgilerini saklayan ve bu bilgiler üzerinde düzenleme, ekleme, silme gibi işlemleri gerçekleştiren veri tabanı ve masaüstü uygulamasını gerçekledim. Bu uygulamanın amacı bir spor klübünün personel ve finansal işlemlerini kolay ve hızlı bir şekilde gerçekleştirmesidir. Bu doğrultuda basit bir ara yüzü olan programı Windows form uygulaması şeklinde gerçekledim. Uygulamanın veri tabi kısmını postgresql kullanarak, ara yüz ve kullanıcı işlemleri kısmını da c# programlama dili kullanarak yaptım.

İŞ KURALLARI:

- Sistemdeki her personel id'siyle temsil edilir.
- Bu id'ler benzersiz olmalıdır.
- Sistemdeki her personelin kişisel bilgileri yer almalıdır.
  - Ad
  - Soyad
  - Maas
  - Yas
  - Personel Tipi
- Yukarıdaki bilgiler boş geçilemez.
- Sisteme yeni bir personel eklendiği zaman id'si otomatik artan olmalıdır.
- Personeller ile ilgili ekleme, silme, güncelleme ve arama işlemleri için fonksiyonlar yazılmalıdır.
- Sistemden personel silme işlemi id'ler üzerinden yapılmalıdır.
- Personel tipine göre her personelin ayrıca tutulduğu birer personel tipi tablosu olmalıdır.
- Bir personelin yalnız bir personel tipi olabilir.
- Sistemdeki personeller listelenirken personel tipine göre de ayrıca listelenmelidir.
- Sisteme yeni bir personel girildiği zaman bu personelin bilgileri aynı zamanda personelin personel tipine göre olan tabloya da işlenmelidir.
- Bu işlenme trigger'ler aracılığıyla yapılmalıdır.
- Genel mali durumu tutan bir tablo olmalıdır.
- Mali durum gelir ve gider diye 2 ayrı tabloda da gösterilmelidir ve bu tablolarında(gelir-gider) alt tabloları yani gelir türleri ve/veya gider türlerine ait tablolar olmalıdır.
- Bu tablolarda tutulan gelir veya gider türlerinin id 'leri olmalıdır.
- Bu id'ler benzersiz olmalıdırlar.
- Bu id'ler üzerinden gelir veya giderlerin tek tek başlık halinde olduğu tablolar olmadılır.
- Bu tablolar da istenildiği zaman ayrı ayrı ekrana getirilip incelenebilir olmalıdır.

- Gelir veya gider tablolarına ve alt tablolarına veri girişi ara yüz programından yapılmamalıdır.
- Gelir veya gider tablolarına veri girişi sadece veri tabanından yapılan değişiklikle gerçekleşmelidir.

İLİŞKİSEL ŞEMA:

MTSporKlübü(personel:string, malidurum:integer)

personel(id:int,ad:string,soyad:string,maas:int,yas:int,personeltipi:string)

yonetim(id:int,ad:string,soyad:string,maas:int,yas:int,personeltipi:string)

teknikekip(id:int,ad:string,soyad:string,maas:int,yas:int,personeltipi:string)

futbolcu(id:int,ad:string,soyad:string,maas:int,yas:int,personeltipi:string)

saglikcilar(id:int,ad:string,soyad:string,maas:int,yas:int,personeltipi:string)

digerpersonel(id:int,ad:string,soyad:string,maas:int,yas:int,personeltipi:string)

malidurum**(id:int**,geliradi:string,gelir:int,gideradi:string,gider:int)

gelir(**id:int**,gelirturu:string,gelir:int)

yayingeliri(**id:int**,yayingeliri:int)

stadgeliri(**id:int**,stadgeliri:int)

mtstoregeliri(**id:int**,mtstoregeliri:int)

sponsorgeliri(**id:int**,sponsorgeliri:int)

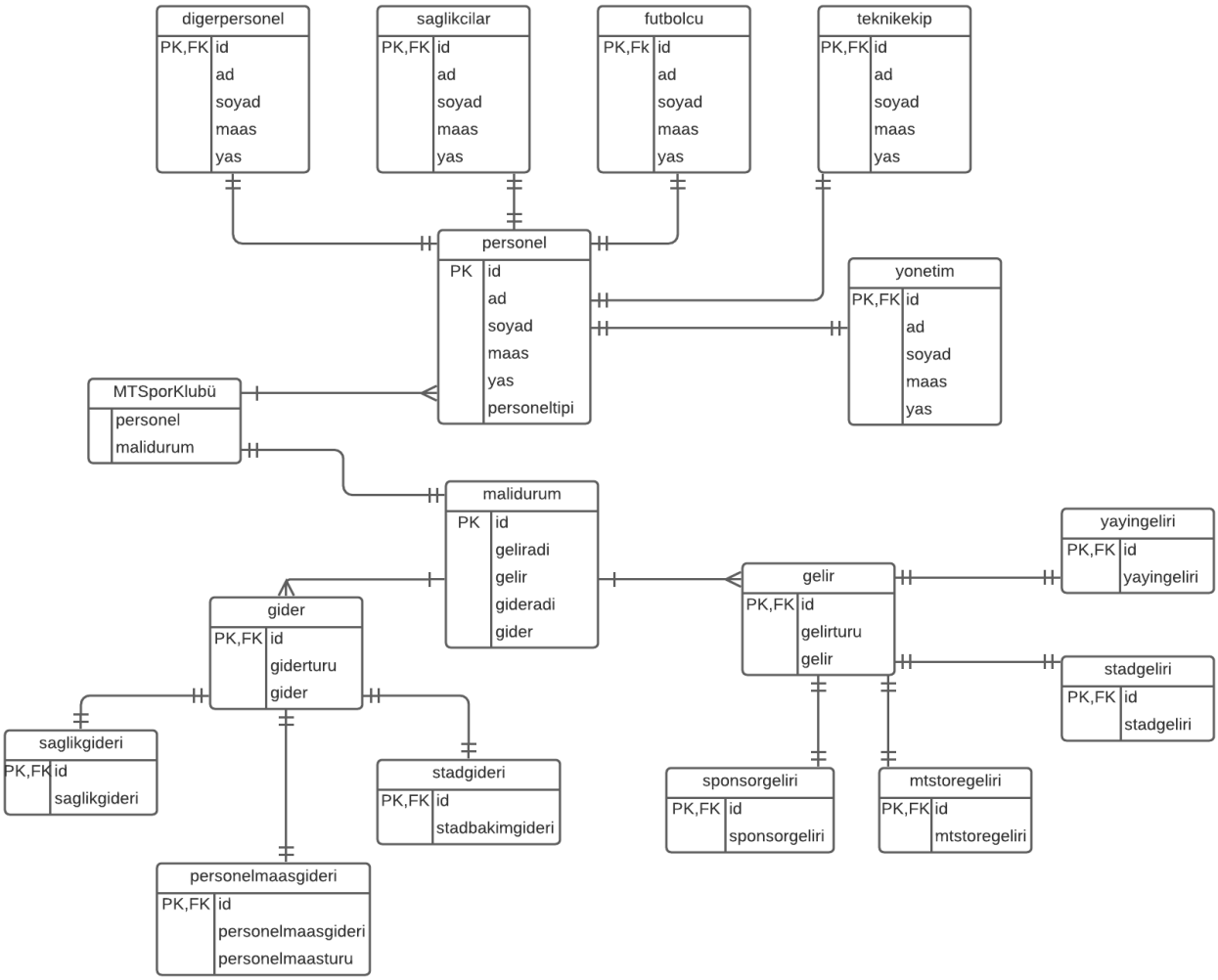
gider(**id:int**,giderturu:string,gider:int)

stadgideri(id:int,stadbakimgideri:int)

personelmaasgideri(**id:int**,personelmaasgider:int,personelmaasturu:string)

saglikgideri(**id:int**,saglikgideri:int)

VARLIK BAĞINTI MODELİ:



SQL İFADELİR:

--

-- PostgreSQL database dump

--

-- Dumped from database version 13.1

-- Dumped by pg_dump version 13.1

SET statement_timeout = 0;

```
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SELECT pg_catalog.set_config('search_path', '', false);
SET check_function_bodies = false;
SET xmloption = content;
SET client_min_messages = warning;
SET row_security = off;


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

CREATE FUNCTION public.digerpersonelekle() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
declare
idsi integer;
adi varchar;
soyadi varchar;
maasi integer;
yasi integer;
personeltipii varchar;
begin
idsi:= (select id from personel order by id desc limit 1 );
adi:= (select ad from personel order by id desc limit 1 );
soyadi:= (select soyad from personel order by id desc limit 1 );
maasi:= (select maas from personel order by id desc limit 1 );
```

```
yasi:= (select yas from personel order by id desc limit 1 );
personeltipii:= (select personeltipi from personel order by id desc limit 1 );
        if personeltipii='digerpersonel' then
        insert into digerpersonel(id,ad,soyad,maas,yas) values(idsi,adi,soyadi,maasi,yasi) ;
        end if;
return new ;
end;
$$;


ALTER FUNCTION public.digerpersonelekle() OWNER TO postgres;


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

CREATE FUNCTION public.digerpersonelgetir(personelno integer) RETURNS TABLE(adi
character varying, soyadi character varying, maasi integer, yasi integer)
   LANGUAGE plpgsql
   AS $$
begin
return query select ad,soyad,maas,yas from digerpersonel where id=personelNO;
end;
$$;


ALTER FUNCTION public.digerpersonelgetir(personelno integer) OWNER TO postgres;


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

```
--


CREATE FUNCTION public.futbolcuekle() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
declare
idsi integer;
adi varchar;
soyadi varchar;
maasi integer;
yasi integer;
personeltipii varchar;
begin
idsi:= (select id from personel order by id desc limit 1 );
adi:= (select ad from personel order by id desc limit 1 );
soyadi:= (select soyad from personel order by id desc limit 1 );
maasi:= (select maas from personel order by id desc limit 1 );
yasi:= (select yas from personel order by id desc limit 1 );
personeltipii:= (select personeltipi from personel order by id desc limit 1 );
        if personeltipii='futbolcu' then
        insert into futbolcu(id,ad,soyad,maas,yas) values(idsi,adi,soyadi,maasi,yasi) ;
        end if;
return new ;
end;
$$;


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

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

CREATE FUNCTION public.futbolcugetir(personelno integer) RETURNS TABLE(adi character varying, soyadi character varying, maasi integer, yasi integer)
    LANGUAGE plpgsql
    AS $$
begin
return query select ad,soyad,maas,yas from futbolcu where id=personelNO;
end;
$$;


ALTER FUNCTION public.futbolcugetir(personelno integer) OWNER TO postgres;

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

CREATE FUNCTION public.personelgetir(personelno integer) RETURNS TABLE(adi character varying, soyadi character varying, maasi integer, yasi integer, personeltipii character varying)
    LANGUAGE plpgsql
    AS $$
begin
return query select ad,soyad,maas,yas,personeltipi from personel where id=personelNO;
end;
$$;
```

```
ALTER FUNCTION public.personelgetir(personelno integer) OWNER TO postgres;


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


CREATE FUNCTION public.saglikciekle() RETURNS trigger
   LANGUAGE plpgsql
   AS $$
declare
idsi integer;
adi varchar;
soyadi varchar;
maasi integer;
yasi integer;
personeltipii varchar;
begin
idsi:= (select id from personel order by id desc limit 1 );
adi:= (select ad from personel order by id desc limit 1 );
soyadi:= (select soyad from personel order by id desc limit 1 );
maasi:= (select maas from personel order by id desc limit 1 );
yasi:= (select yas from personel order by id desc limit 1 );
personeltipii:= (select personeltipi from personel order by id desc limit 1 );
        if personeltipii='doktor' then
        insert into saglikcilar(id,ad,soyad,maas,yas) values(idsi,adi,soyadi,maasi,yasi) ;
        end if;
return new ;
end;
$$;
```

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


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


CREATE FUNCTION public.saglikcigetir(personelno integer) RETURNS TABLE(adi character varying, soyadi character varying, maasi integer, yasi integer)
    LANGUAGE plpgsql
    AS $$
begin
return query select ad,soyad,maas,yas from saglikcilar where id=personelNO;
end;
$$;


ALTER FUNCTION public.saglikcigetir(personelno integer) OWNER TO postgres;


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


CREATE FUNCTION public.teknikekipekle() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
declare
idsi integer;
adi varchar;
```

```
soyadi varchar;
maasi integer;
yasi integer;
personeltipii varchar;
begin
idsi:= (select id from personel order by id desc limit 1 );
adi:= (select ad from personel order by id desc limit 1 );
soyadi:= (select soyad from personel order by id desc limit 1 );
maasi:= (select maas from personel order by id desc limit 1 );
yasi:= (select yas from personel order by id desc limit 1 );
personeltipii:= (select personeltipi from personel order by id desc limit 1 );
        if personeltipii='teknikekip' then
        insert into teknikekip(id,ad,soyad,maas,yas) values(idsi,adi,soyadi,maasi,yasi) ;
        end if;
return new ;
end;
$$;


ALTER FUNCTION public.teknikekipekle() OWNER TO postgres;


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


CREATE FUNCTION public.teknikekipgetir(personelno integer) RETURNS TABLE(adi character varying, soyadi character varying, maasi integer, yasi integer)
   LANGUAGE plpgsql
   AS $$
begin
```

```
return query select ad,soyad,maas,yas from teknikekip where id=personelNO;
end;
$$;


ALTER FUNCTION public.teknikekipgetir(personelno integer) OWNER TO postgres;

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

CREATE FUNCTION public.yonetimekle() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
declare
idsi integer;
adi varchar;
soyadi varchar;
maasi integer;
yasi integer;
personeltipii varchar;
begin
idsi:= (select id from personel order by id desc limit 1 );
adi:= (select ad from personel order by id desc limit 1 );
soyadi:= (select soyad from personel order by id desc limit 1 );
maasi:= (select maas from personel order by id desc limit 1 );
yasi:= (select yas from personel order by id desc limit 1 );
personeltipii:= (select personeltipi from personel order by id desc limit 1 );
            if personeltipii='yonetim' then
```

```
        insert into yonetim(id,ad,soyad,maas,yas) values(idsi,adi,soyadi,maasi,yasi) ;
        end if;
return new ;
end;
$$;


ALTER FUNCTION public.yonetimekle() OWNER TO postgres;


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

CREATE FUNCTION public.yonetimgetir(personelno integer) RETURNS TABLE(adi character varying, soyadi character varying, maasi integer, yasi integer)
   LANGUAGE plpgsql
   AS $$
begin
return query select ad,soyad,maas,yas from yonetim where id=personelNO;
end;
$$;


ALTER FUNCTION public.yonetimgetir(personelno integer) OWNER TO postgres;

SET default_tablespace = '';

SET default_table_access_method = heap;

--
```

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

CREATE TABLE public.digerpersonel (
    id integer NOT NULL,
    ad character varying,
    soyad character varying,
    maas integer,
    yas integer
);


ALTER TABLE public.digerpersonel OWNER TO postgres;

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

CREATE TABLE public.futbolcu (
    id integer NOT NULL,
    ad character varying,
    soyad character varying,
    maas integer,
    yas integer
);


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

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

CREATE TABLE public.gelir (
    id integer NOT NULL,
    gelirturu character varying,
    gelir integer
);


ALTER TABLE public.gelir OWNER TO postgres;

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

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


ALTER TABLE public.gelir_id_seq OWNER TO postgres;

--
```

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

ALTER SEQUENCE public.gelir_id_seq OWNED BY public.gelir.id;



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

CREATE TABLE public.gider (
    id integer NOT NULL,
    giderturu character varying(20),
    gider integer
);


ALTER TABLE public.gider OWNER TO postgres;

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

CREATE SEQUENCE public.gider_id_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
```

```
    CACHE 1;


ALTER TABLE public.gider_id_seq OWNER TO postgres;

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

ALTER SEQUENCE public.gider_id_seq OWNED BY public.gider.id;


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

CREATE TABLE public.malidurum (
    id integer NOT NULL,
    geliradi character varying(25),
    gelir integer,
    gideradi character varying(25),
    gider integer
);


ALTER TABLE public.malidurum OWNER TO postgres;

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

```
--

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


ALTER TABLE public.malidurum_id_seq OWNER TO postgres;

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

ALTER SEQUENCE public.malidurum_id_seq OWNED BY public.malidurum.id;


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

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

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

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

CREATE TABLE public.personel (
    id integer NOT NULL,
    ad character varying,
    soyad character varying,
    maas integer,
    yas integer,
    personeltipi character varying
);


ALTER TABLE public.personel OWNER TO postgres;

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

CREATE SEQUENCE public.personel_id_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
```

```
    CACHE 1;


ALTER TABLE public.personel_id_seq OWNER TO postgres;

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

ALTER SEQUENCE public.personel_id_seq OWNED BY public.personel.id;


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

CREATE TABLE public.personelmaasgideri (
    id integer NOT NULL,
    personelmaasgideri integer,
    personelmaasturu character varying(25)
);


ALTER TABLE public.personelmaasgideri OWNER TO postgres;

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

```
CREATE TABLE public.saglikcilar (
    id integer NOT NULL,
    ad character varying,
    soyad character varying,
    maas integer,
    yas integer
);


ALTER TABLE public.saglikcilar OWNER TO postgres;

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

CREATE TABLE public.saglikgideri (
    id integer NOT NULL,
    saglikgideri integer
);


ALTER TABLE public.saglikgideri OWNER TO postgres;

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

CREATE TABLE public.sponsorgeliri (
    id integer NOT NULL,
```

```
    sponsorgeliri character varying
);


ALTER TABLE public.sponsorgeliri OWNER TO postgres;

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

CREATE TABLE public.stadgeliri (
    id integer NOT NULL,
    stadgeliri character varying
);


ALTER TABLE public.stadgeliri OWNER TO postgres;

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

CREATE TABLE public.stadgideri (
    id integer NOT NULL,
    "stadbakımgideri" integer
);


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

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

CREATE TABLE public.teknikekip (
    id integer NOT NULL,
    ad character varying,
    soyad character varying,
    maas integer,
    yas integer
);


ALTER TABLE public.teknikekip OWNER TO postgres;

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

CREATE TABLE public.yayingeliri (
    id integer NOT NULL,
    yayingeliri character varying
);


ALTER TABLE public.yayingeliri OWNER TO postgres;

--
```

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

CREATE TABLE public.yonetim (
    id integer NOT NULL,
    ad character varying,
    soyad character varying,
    maas integer,
    yas integer
);


ALTER TABLE public.yonetim OWNER TO postgres;

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

ALTER TABLE ONLY public.gelir ALTER COLUMN id SET DEFAULT
nextval('public.gelir_id_seq'::regclass);


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

ALTER TABLE ONLY public.gider ALTER COLUMN id SET DEFAULT
nextval('public.gider_id_seq'::regclass);
```

```
--

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

--


ALTER TABLE ONLY public.malidurum ALTER COLUMN id SET DEFAULT
nextval('public.malidurum_id_seq'::regclass);



--

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

--


ALTER TABLE ONLY public.personel ALTER COLUMN id SET DEFAULT
nextval('public.personel_id_seq'::regclass);



--

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

--


INSERT INTO public.digerpersonel VALUES (10, 'emirhan', 'oksuz', 6000, 21);

INSERT INTO public.digerpersonel VALUES (2, 'feyza', 'akdogan', 6000, 24);

INSERT INTO public.digerpersonel VALUES (23, 'reha', 'aydin', 4500, 18);

INSERT INTO public.digerpersonel VALUES (22, 'fatih', 'turan', 4500, 33);

INSERT INTO public.digerpersonel VALUES (36, 'elif', 'ertuğrul', 9900, 22);



--

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

--
```

```
INSERT INTO public.futbolcu VALUES (18, 'enes', 'anil', 10000, 30);
INSERT INTO public.futbolcu VALUES (19, 'bahadir', 'ozcan', 10000, 26);
INSERT INTO public.futbolcu VALUES (7, 'burak', 'celik', 10000, 20);
INSERT INTO public.futbolcu VALUES (15, 'ahmet', 'dasdemir', 10000, 19);
INSERT INTO public.futbolcu VALUES (28, 'yasin ', 'yaz', 9000, 26);
INSERT INTO public.futbolcu VALUES (29, 'ömer', 'erkurt', 9500, 27);
INSERT INTO public.futbolcu VALUES (30, 'ali', 'keser', 11000, 32);
INSERT INTO public.futbolcu VALUES (31, 'enes ', 'kaynak', 9600, 26);
INSERT INTO public.futbolcu VALUES (32, 'faruk enes', 'ozcilingir', 2500, 17);
INSERT INTO public.futbolcu VALUES (33, 'serkan', 'yağlı', 3000, 19);
INSERT INTO public.futbolcu VALUES (34, 'sinan', 'baykuş', 9600, 24);
INSERT INTO public.futbolcu VALUES (35, 'enes', 'damar', 5600, 29);
INSERT INTO public.futbolcu VALUES (46, 'asmet', 'çıkık', 5693, 24);
INSERT INTO public.futbolcu VALUES (47, 'doğukan', 'özkader', 5469, 26);
INSERT INTO public.futbolcu VALUES (50, 'eren ', 'uğurlu', 4569, 21);
INSERT INTO public.futbolcu VALUES (51, 'ali', 'özdemir', 9635, 20);
INSERT INTO public.futbolcu VALUES (52, 'ricardo', 'quaresma', 20150, 35);



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


INSERT INTO public.gelir VALUES (1, 'yayingeliri', 12000);
INSERT INTO public.gelir VALUES (2, 'stadgeleri', 22000);
INSERT INTO public.gelir VALUES (3, 'mtstoregeliri', 30000);
INSERT INTO public.gelir VALUES (4, 'sponsorgeliri', 5000);
```

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

INSERT INTO public.gider VALUES (1, 'stadgideri', 1000);
INSERT INTO public.gider VALUES (2, 'saglikgideri', 1500);



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

INSERT INTO public.malidurum VALUES (1, 'yayingeliri', 12000, NULL, NULL);
INSERT INTO public.malidurum VALUES (2, 'stadgeleri', 22000, NULL, NULL);
INSERT INTO public.malidurum VALUES (3, 'mtstoregeliri', 30000, NULL, NULL);
INSERT INTO public.malidurum VALUES (4, 'sponsorgeliri', 5000, NULL, NULL);
INSERT INTO public.malidurum VALUES (5, NULL, NULL, 'stadgideri', 1000);
INSERT INTO public.malidurum VALUES (6, NULL, NULL, 'saglikgideri', 1500);



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

INSERT INTO public.mtstoregeliri VALUES (3, '30000');



--
```

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

INSERT INTO public.personel VALUES (1, 'melih', 'tufekcioglu', 6000, 22, 'yonetici');
INSERT INTO public.personel VALUES (2, 'feyza', 'akdogan', 6000, 24, 'digerPersonel');
INSERT INTO public.personel VALUES (3, 'menekse', 'keskin', 7000, 23, 'doktor');
INSERT INTO public.personel VALUES (5, 'akif', 'tufekcioglu', 7000, 30, 'doktor');
INSERT INTO public.personel VALUES (6, 'arif', 'kaynak', 7000, 29, 'doktor');
INSERT INTO public.personel VALUES (7, 'burak', 'celik', 10000, 20, 'futbolcu');
INSERT INTO public.personel VALUES (10, 'emirhan', 'oksuz', 6000, 21, 'digerPersonel');
INSERT INTO public.personel VALUES (15, 'ahmet', 'dasdemir', 10000, 19, 'futbolcu');
INSERT INTO public.personel VALUES (18, 'enes', 'anil', 10000, 30, 'futbolcu');
INSERT INTO public.personel VALUES (19, 'bahadir', 'ozcan', 10000, 26, 'futbolcu');
INSERT INTO public.personel VALUES (20, 'gani', 'kirlioglu', 5200, 35, 'teknikekip');
INSERT INTO public.personel VALUES (21, 'can', 'daldiran', 6900, 40, 'yonetim');
INSERT INTO public.personel VALUES (22, 'fatih', 'turan', 4500, 33, 'digerpersonel');
INSERT INTO public.personel VALUES (23, 'reha', 'aydin', 4500, 18, 'digerpersonel');
INSERT INTO public.personel VALUES (24, 'pınar', 'aydin', 4500, 25, 'saglikcilar');
INSERT INTO public.personel VALUES (27, 'salih', 'ovan', 7000, 25, 'doktor');
INSERT INTO public.personel VALUES (28, 'yasin ', 'yaz', 9000, 26, 'futbolcu');
INSERT INTO public.personel VALUES (29, 'ömer', 'erkurt', 9500, 27, 'futbolcu');
INSERT INTO public.personel VALUES (30, 'ali', 'keser', 11000, 32, 'futbolcu');
INSERT INTO public.personel VALUES (31, 'enes ', 'kaynak', 9600, 26, 'futbolcu');
INSERT INTO public.personel VALUES (32, 'faruk enes', 'ozcilingir', 2500, 17, 'futbolcu');
INSERT INTO public.personel VALUES (33, 'serkan', 'yağlı', 3000, 19, 'futbolcu');
INSERT INTO public.personel VALUES (34, 'sinan', 'baykuş', 9600, 24, 'futbolcu');
INSERT INTO public.personel VALUES (35, 'enes', 'damar', 5600, 29, 'futbolcu');
INSERT INTO public.personel VALUES (36, 'elif', 'ertuğrul', 9900, 22, 'digerpersonel');
INSERT INTO public.personel VALUES (37, 'fatih', 'karkınlı', 6000, 26, 'teknikekip');
```

```
INSERT INTO public.personel VALUES (38, 'tayip', 'usta', 1000, 29, 'yonetim');
INSERT INTO public.personel VALUES (39, 'emre ', 'önsöz', 6500, 30, 'yonetim');
INSERT INTO public.personel VALUES (40, 'berkay', 'bakis', 10000, 27, 'yonetim');
INSERT INTO public.personel VALUES (43, 'burcu', 'yigit', 6950, 34, 'yonetim');
INSERT INTO public.personel VALUES (44, 'berfin', 'gevşek', 9560, 40, 'teknikekip');
INSERT INTO public.personel VALUES (45, 'eda', 'sarıtaş', 1520, 39, 'teknikekip');
INSERT INTO public.personel VALUES (46, 'asmet', 'çıkık', 5693, 24, 'futbolcu');
INSERT INTO public.personel VALUES (47, 'doğukan', 'özkader', 5469, 26, 'futbolcu');
INSERT INTO public.personel VALUES (48, 'ahmet', 'karademir', 5963, 36, 'teknikekip');
INSERT INTO public.personel VALUES (49, 'emine', 'çimen', 19000, 45, 'teknikekip');
INSERT INTO public.personel VALUES (50, 'eren ', 'uğurlu', 4569, 21, 'futbolcu');
INSERT INTO public.personel VALUES (51, 'ali', 'özdemir', 9635, 20, 'futbolcu');
INSERT INTO public.personel VALUES (52, 'ricardo', 'quaresma', 20150, 35, 'futbolcu');
INSERT INTO public.personel VALUES (54, 'aslı', 'tufekcioglu', 15000, 15, 'yonetim');
INSERT INTO public.personel VALUES (56, 'yasin', 'nurlu', 6500, 32, 'teknikekip');



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




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

INSERT INTO public.saglikcilar VALUES (3, 'menekse', 'keskin', 7000, 23);
INSERT INTO public.saglikcilar VALUES (5, 'akif', 'tufekcioglu', 7000, 30);
```

```
INSERT INTO public.saglikcilar VALUES (6, 'arif', 'kaynak', 7000, 29);
INSERT INTO public.saglikcilar VALUES (24, 'pınar', 'aydin', 4500, 25);
INSERT INTO public.saglikcilar VALUES (27, 'salih', 'ovan', 7000, 25);


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

INSERT INTO public.saglikgideri VALUES (2, 1500);


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

INSERT INTO public.sponsorgeliri VALUES (4, '5000');


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

INSERT INTO public.stadgeliri VALUES (2, '22000');


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

```
INSERT INTO public.stadgideri VALUES (1, 1000);


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

INSERT INTO public.teknikekip VALUES (20, 'gani', 'kirlioglu', 5200, 35);
INSERT INTO public.teknikekip VALUES (37, 'fatih', 'karkınlı', 6000, 26);
INSERT INTO public.teknikekip VALUES (44, 'berfin', 'gevşek', 9560, 40);
INSERT INTO public.teknikekip VALUES (45, 'eda', 'sarıtaş', 1520, 39);
INSERT INTO public.teknikekip VALUES (48, 'ahmet', 'karademir', 5963, 36);
INSERT INTO public.teknikekip VALUES (49, 'emine', 'çimen', 19000, 45);
INSERT INTO public.teknikekip VALUES (56, 'yasin', 'nurlu', 9654, 31);


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

INSERT INTO public.yayingeliri VALUES (1, '12000');


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

INSERT INTO public.yonetim VALUES (1, 'melih', 'tufekcioglu', 6000, 22);
```

```
INSERT INTO public.yonetim VALUES (21, 'can', 'daldiran', 6900, 40);
INSERT INTO public.yonetim VALUES (38, 'tayip', 'usta', 1000, 29);
INSERT INTO public.yonetim VALUES (39, 'emre', 'önsöz', 6500, 30);
INSERT INTO public.yonetim VALUES (40, 'berkay', 'bakis', 10000, 27);
INSERT INTO public.yonetim VALUES (43, 'burcu', 'yigit', 6950, 34);
INSERT INTO public.yonetim VALUES (54, 'aslı', 'tufekcioglu', 15000, 15);


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

SELECT pg_catalog.setval('public.gelir_id_seq', 2, true);


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

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


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

SELECT pg_catalog.setval('public.malidurum_id_seq', 6, true);
```

```
--

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

--


SELECT pg_catalog.setval('public.personel_id_seq', 56, true);




--

-- Name: digerpersonel DIGERPERSONEL; Type: CONSTRAINT; Schema: public; Owner:
postgres

--


ALTER TABLE ONLY public.digerpersonel

   ADD CONSTRAINT "DIGERPERSONEL" PRIMARY KEY (id);




--

-- Name: futbolcu FUTBOLCU; Type: CONSTRAINT; Schema: public; Owner: postgres

--


ALTER TABLE ONLY public.futbolcu

   ADD CONSTRAINT "FUTBOLCU" PRIMARY KEY (id);




--

-- Name: saglikcilar SAGLIKCILAR; Type: CONSTRAINT; Schema: public; Owner: postgres

--


ALTER TABLE ONLY public.saglikcilar

   ADD CONSTRAINT "SAGLIKCILAR" PRIMARY KEY (id);
```

```
--
-- Name: teknikekip TeknikEkipPK; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.teknikekip
    ADD CONSTRAINT "TeknikEkipPK" PRIMARY KEY (id);



--
-- Name: gelir gelirPK; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.gelir
    ADD CONSTRAINT "gelirPK" PRIMARY KEY (id);



--
-- Name: gider giderPK; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.gider
    ADD CONSTRAINT "giderPK" PRIMARY KEY (id);



--
-- Name: malidurum malidurumPK; Type: CONSTRAINT; Schema: public; Owner: postgres
--
```

```
ALTER TABLE ONLY public.malidurum
    ADD CONSTRAINT "malidurumPK" PRIMARY KEY (id);


--
-- Name: mtstoregeliri mtstoregeliriPK; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.mtstoregeliri
    ADD CONSTRAINT "mtstoregeliriPK" PRIMARY KEY (id);


--
-- Name: personelmaasgideri personelmaasgideriPK; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.personelmaasgideri
    ADD CONSTRAINT "personelmaasgideriPK" PRIMARY KEY (id);


--
-- Name: saglikgideri saglikgideriPK; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.saglikgideri
    ADD CONSTRAINT "saglikgideriPK" PRIMARY KEY (id);
```

```
--
-- Name: sponsorgeliri sponsorgeliriPK; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.sponsorgeliri
    ADD CONSTRAINT "sponsorgeliriPK" PRIMARY KEY (id);


--
-- Name: stadgeliri stadgeliriPK; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.stadgeliri
    ADD CONSTRAINT "stadgeliriPK" PRIMARY KEY (id);


--
-- Name: stadgideri stadgiderPK; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.stadgideri
    ADD CONSTRAINT "stadgiderPK" PRIMARY KEY (id);


--
-- Name: yayingeliri yayıngeliriPK; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.yayingeliri
```

```
    ADD CONSTRAINT "yayıngeliriPK" PRIMARY KEY (id);


--
-- Name: personel yoneticiPK; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.personel
    ADD CONSTRAINT "yoneticiPK" PRIMARY KEY (id);


--
-- Name: yonetim yonetimPK; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.yonetim
    ADD CONSTRAINT "yonetimPK" PRIMARY KEY (id);


--
-- Name: personel digerpersonelekletrig; Type: TRIGGER; Schema: public; Owner: postgres
--

CREATE TRIGGER digerpersonelekletrig AFTER INSERT ON public.personel FOR EACH ROW
EXECUTE FUNCTION public.digerpersonelekle();


--
-- Name: personel futbolcuekletrig; Type: TRIGGER; Schema: public; Owner: postgres
--
```

CREATE TRIGGER futbolcuekletrig AFTER INSERT ON public.personel FOR EACH ROW EXECUTE FUNCTION public.futbolcuekle();

--

-- Name: personel saglikciekletrig; Type: TRIGGER; Schema: public; Owner: postgres

--

CREATE TRIGGER saglikciekletrig AFTER INSERT ON public.personel FOR EACH ROW EXECUTE FUNCTION public.saglikciekle();

--

-- Name: personel teknikekiptrig; Type: TRIGGER; Schema: public; Owner: postgres

--

CREATE TRIGGER teknikekiptrig AFTER INSERT ON public.personel FOR EACH ROW EXECUTE FUNCTION public.teknikekipekle();

--

-- Name: personel yonetimekletrig; Type: TRIGGER; Schema: public; Owner: postgres

--

CREATE TRIGGER yonetimekletrig AFTER INSERT ON public.personel FOR EACH ROW EXECUTE FUNCTION public.yonetimekle();

--

-- Name: gelir malidurumgelir; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.gelir

ADD CONSTRAINT malidurumgelir FOREIGN KEY (id) REFERENCES public.malidurum(id) ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: mtstoregeliri mtstoregelirigelir; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.mtstoregeliri

ADD CONSTRAINT mtstoregelirigelir FOREIGN KEY (id) REFERENCES public.gelir(id) ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: digerpersonel personelDIGERPERSONEL; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.digerpersonel

ADD CONSTRAINT "personelDIGERPERSONEL" FOREIGN KEY (id) REFERENCES public.personel(id) ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: futbolcu personelFutbolcu; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.futbolcu

ADD CONSTRAINT "personelFutbolcu" FOREIGN KEY (id) REFERENCES public.personel(id) ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: saglikcilar personelSAGLIKCI; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.saglikcilar

ADD CONSTRAINT "personelSAGLIKCI" FOREIGN KEY (id) REFERENCES public.personel(id) ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: teknikekip personelTeknikEkip; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.teknikekip

ADD CONSTRAINT "personelTeknikEkip" FOREIGN KEY (id) REFERENCES public.personel(id) ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: yonetim personelYonetim; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.yonetim

ADD CONSTRAINT "personelYonetim" FOREIGN KEY (id) REFERENCES public.personel(id) ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: personelmaasgideri personelmaasgiderigider; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.personelmaasgideri

ADD CONSTRAINT personelmaasgiderigider FOREIGN KEY (id) REFERENCES public.gelir(id) ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: saglikgideri saglikgiderigider; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.saglikgideri

ADD CONSTRAINT saglikgiderigider FOREIGN KEY (id) REFERENCES public.gelir(id) ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: sponsorgeliri sponsorgelirigelir; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.sponsorgeliri

ADD CONSTRAINT sponsorgelirigelir FOREIGN KEY (id) REFERENCES public.gelir(id) ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: stadgeliri stadgelirigelir; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.stadgeliri

    ADD CONSTRAINT stadgelirigelir FOREIGN KEY (id) REFERENCES public.gelir(id) ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: stadgideri stadgidergider; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.stadgideri

    ADD CONSTRAINT stadgidergider FOREIGN KEY (id) REFERENCES public.gelir(id) ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: yayingeliri yeyıngelirigelir; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.yayingeliri

    ADD CONSTRAINT "yeyıngelirigelir" FOREIGN KEY (id) REFERENCES public.gelir(id) ON UPDATE CASCADE ON DELETE CASCADE;

--

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

FONKSİYONLAR:

1)

```
create function personelgetir(personelNO int)
returns table (
	adi varchar,
	soyadi varchar,
	maasi int,
	yasi int,
	personeltipii varchar
)
as
$$
begin
return query select ad,soyad,maas,yas,personeltipi from personel where id=personelNO;
end;
$$
language plpgsql ;
```

2)

```
create function yonetimgetir(personelNO int)
returns table (
	adi varchar,
	soyadi varchar,
	maasi int,
	yasi int
)
as
```

```
$$
begin
return query select ad,soyad,maas,yas from yonetim where id=personelNO;
end;
$$
language plpgsql ;
```

3)

```
create function teknikekipgetir(personelNO int)
returns table (
        adi varchar,
        soyadi varchar,
        maasi int,
        yasi int
)
as
$$
begin
return query select ad,soyad,maas,yas from teknikekip where id=personelNO;
end;
$$
language plpgsql ;
```

4)

```
create function futbolcugetir(personelNO int)
returns table (
        adi varchar,
        soyadi varchar,
        maasi int,
        yasi int
)
```

```
as
$$
begin
return query select ad,soyad,maas,yas from futbolcu where id=personelNO;
end;
$$
language plpgsql ;
```

5)

```
create function saglikcigetir(personelNO int)
returns table (
        adi varchar,
        soyadi varchar,
        maasi int,
        yasi int
)
as
$$
begin
return query select ad,soyad,maas,yas from saglikcilar where id=personelNO;
end;
$$
language plpgsql ;
```

6)

```
create function digerpersonelgetir(personelNO int)
returns table (
        adi varchar,
        soyadi varchar,
        maasi int,
```

```
        yasi int
)
as
$$
begin
return query select ad,soyad,maas,yas from digerpersonel where id=personelNO;
end;
$$
language plpgsql ;
```

TETİKLEYİCİLER:

1)

```
create or replace function digerpersonelekle()
returns trigger
as
$$
declare
idsi integer;
adi varchar;
soyadi varchar;
maasi integer;
yasi integer;
personeltipii varchar;
begin
idsi:= (select id from personel order by id desc limit 1 );
adi:= (select ad from personel order by id desc limit 1 );
soyadi:= (select soyad from personel order by id desc limit 1 );
maasi:= (select maas from personel order by id desc limit 1 );
yasi:= (select yas from personel order by id desc limit 1 );
personeltipii:= (select personeltipi from personel order by id desc limit 1 );
```

```
        if personeltipii='digerPersonel' then
        insert into digerpersonel(id,ad,soyad,maas,yas) values(idsi,adi,soyadi,maasi,yasi) ;
        end if;
return new ;
end;
$$
language plpgsql ;
```

```
create trigger digerpersonelekletrig
after insert
on personel
for each row
execute procedure digerpersonelekle()
```

2)

```
create or replace function futbolcuekle()
returns trigger
as
$$
declare
idsi integer;
adi varchar;
soyadi varchar;
maasi integer;
yasi integer;
personeltipii varchar;
begin
idsi:= (select id from personel order by id desc limit 1 );
```

```
adi:= (select ad from personel order by id desc limit 1 );
soyadi:= (select soyad from personel order by id desc limit 1 );
maasi:= (select maas from personel order by id desc limit 1 );
yasi:= (select yas from personel order by id desc limit 1 );
personeltipii:= (select personeltipi from personel order by id desc limit 1 );
        if personeltipii=futbolcu then
        insert into futbolcu(id,ad,soyad,maas,yas) values(idsi,adi,soyadi,maasi,yasi) ;
        end if;
return new ;
end;
$$
language plpgsql ;
```

```
create trigger futbolcuekletrig
after insert
on personel
for each row
execute procedure futbolcuekle()
```

3)

```
create or replace function saglikciekle()
returns trigger
as
$$
declare
idsi integer;
adi varchar;
soyadi varchar;
```

```
maasi integer;
yasi integer;
personeltipii varchar;
begin
idsi:= (select id from personel order by id desc limit 1 );
adi:= (select ad from personel order by id desc limit 1 );
soyadi:= (select soyad from personel order by id desc limit 1 );
maasi:= (select maas from personel order by id desc limit 1 );
yasi:= (select yas from personel order by id desc limit 1 );
personeltipii:= (select personeltipi from personel order by id desc limit 1 );
        if personeltipii=doktor then
        insert into saglikcilar(id,ad,soyad,maas,yas) values(idsi,adi,soyadi,maasi,yasi) ;
        end if;
return new ;
end;
$$
language plpgsql ;


create trigger saglikciekletrigg
after insert
on personel
for each row
execute procedure saglikciekle()
```

4)

```
create or replace function teknikekipekle()
returns trigger
```

```
as
$$
declare
idsi integer;
adi varchar;
soyadi varchar;
maasi integer;
yasi integer;
personeltipii varchar;
begin
idsi:= (select id from personel order by id desc limit 1 );
adi:= (select ad from personel order by id desc limit 1 );
soyadi:= (select soyad from personel order by id desc limit 1 );
maasi:= (select maas from personel order by id desc limit 1 );
yasi:= (select yas from personel order by id desc limit 1 );
personeltipii:= (select personeltipi from personel order by id desc limit 1 );
        if personeltipii=teknikekip then
        insert into teknikekip(id,ad,soyad,maas,yas) values(idsi,adi,soyadi,maasi,yasi) ;
        end if;
return new ;
end;
$$
language plpgsql ;


create trigger dteknikekipekletrig
after insert
on personel
for each row
```

```
execute procedure teknikekipekle()
```

5)

```
create or replace function yonetimekle()
returns trigger
as
$$
declare
idsi integer;
adi varchar;
soyadi varchar;
maasi integer;
yasi integer;
personeltipii varchar;
begin
idsi:= (select id from personel order by id desc limit 1 );
adi:= (select ad from personel order by id desc limit 1 );
soyadi:= (select soyad from personel order by id desc limit 1 );
maasi:= (select maas from personel order by id desc limit 1 );
yasi:= (select yas from personel order by id desc limit 1 );
personeltipii:= (select personeltipi from personel order by id desc limit 1 );
        if personeltipii=yonetim then
        insert into yonetim(id,ad,soyad,maas,yas) values(idsi,adi,soyadi,maasi,yasi) ;
        end if;
return new ;
end;
$$
language plpgsql ;
```

create trigger yonetimekletrig

after insert

on personel

for each row

execute procedure yonetimekle()

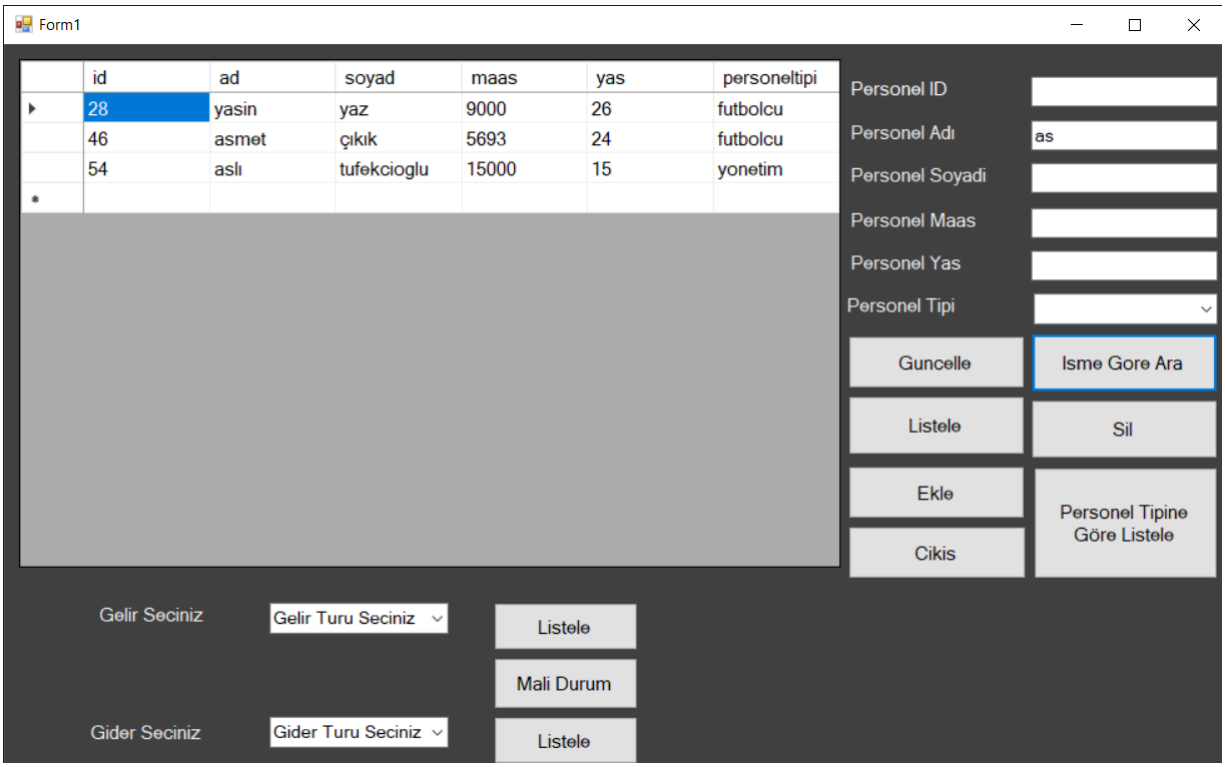
EKRAN GÖRÜNTÜLERİ:

Arama:

Aramadan önceki hali:

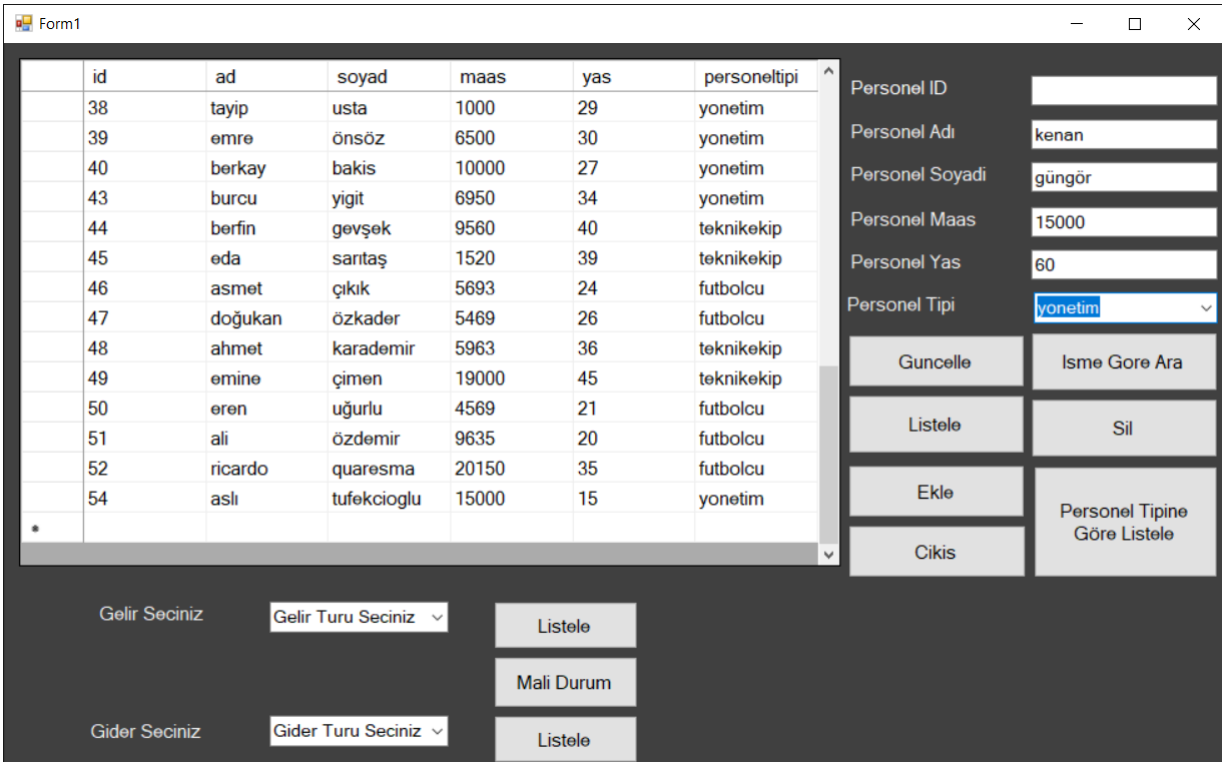
Ara dedikten sonraki hali:

Harfle arama yapma işlemi:

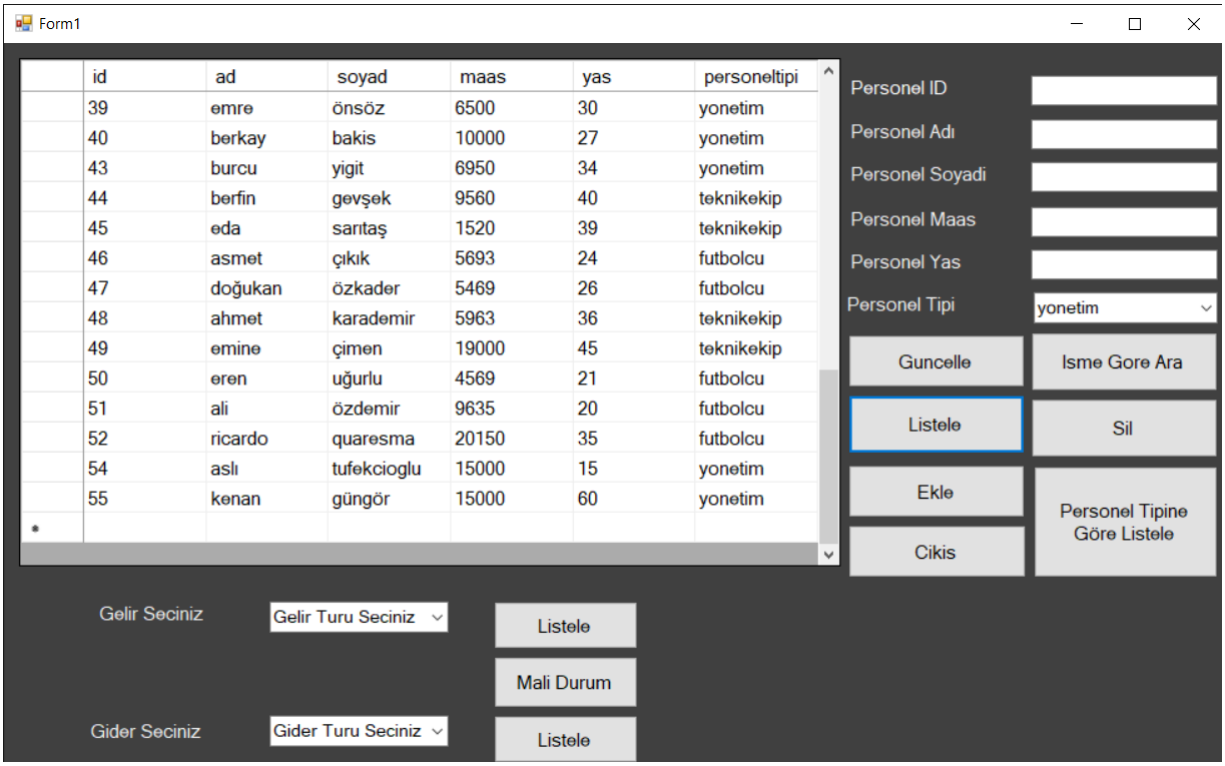


Ekleme:

Eklemden önceki hali:



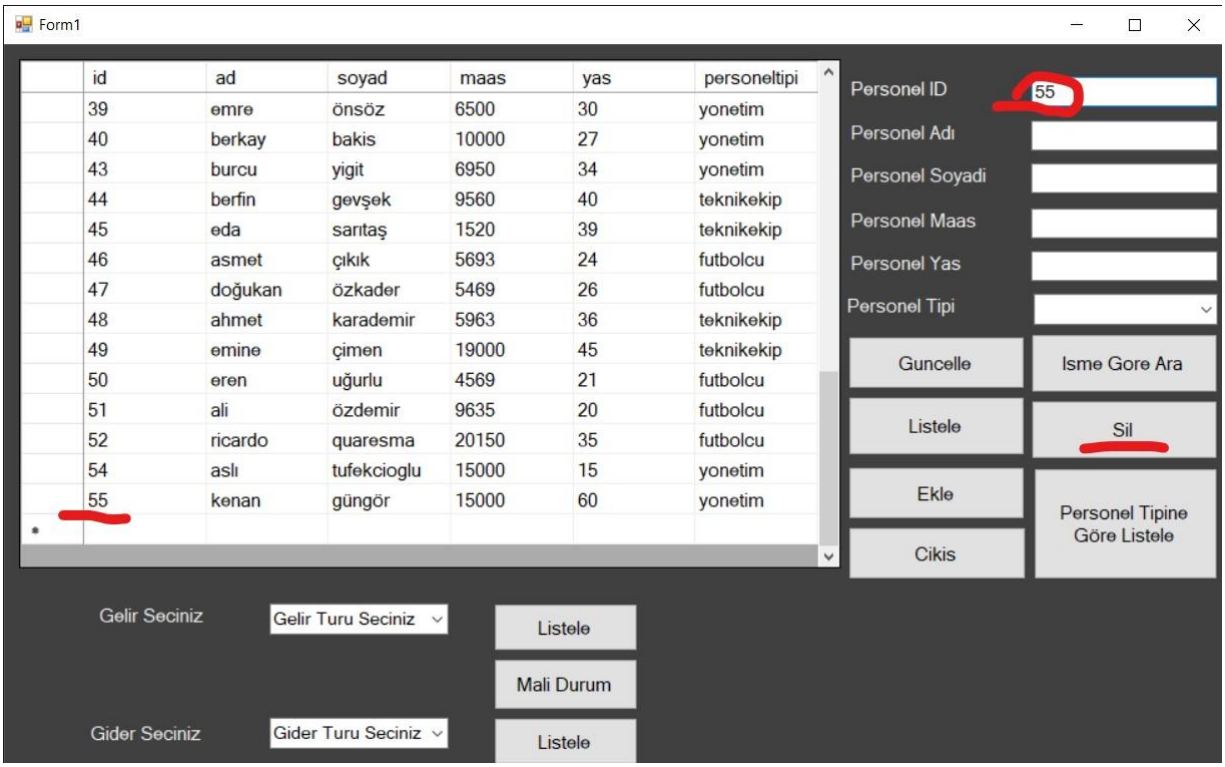
Ekledikten sonraki hali:



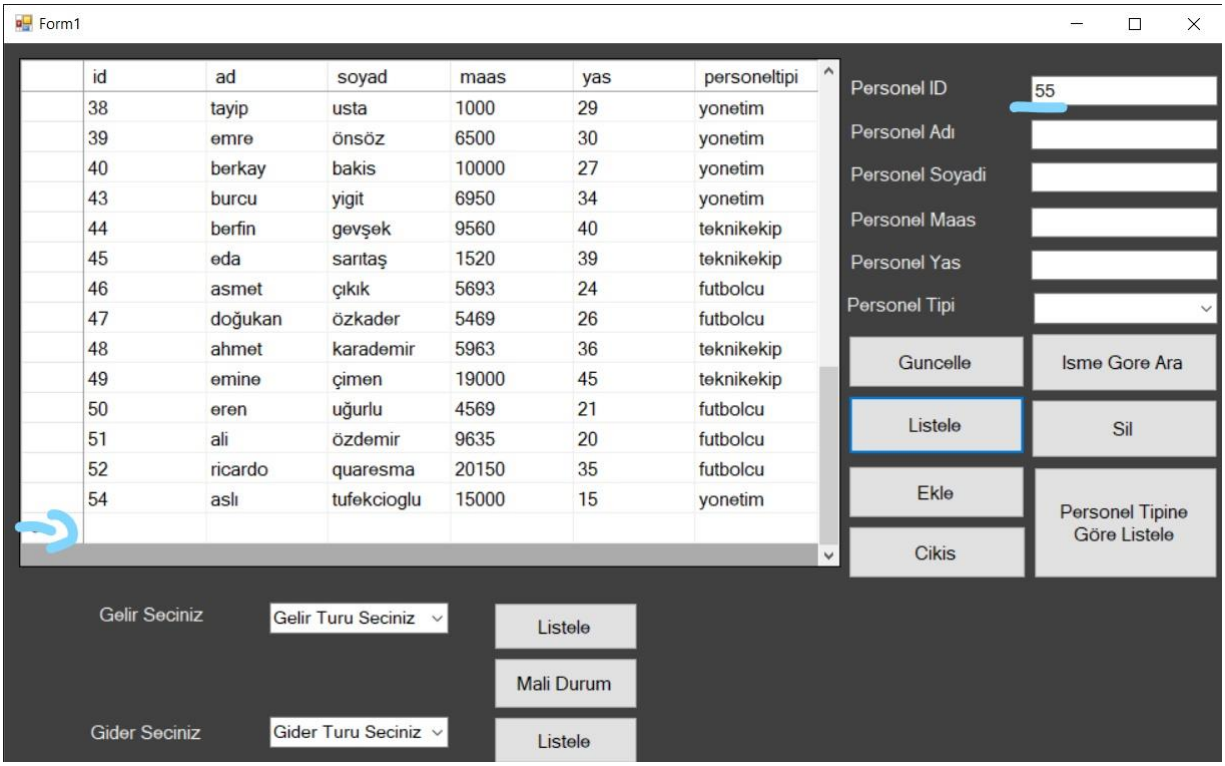
Silme:

-Silme işlemi id üzerinden yapılmaktadır.

Silmeden önceki hali:

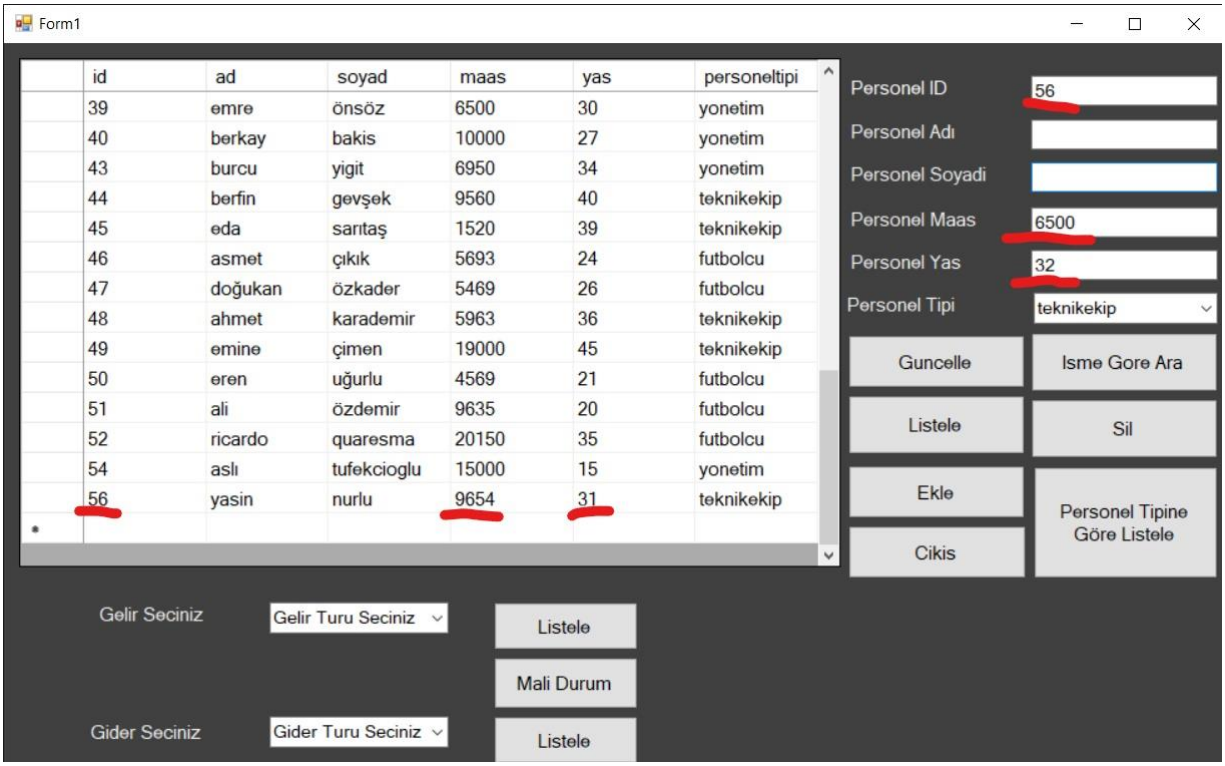


Sildikten sonraki hali:

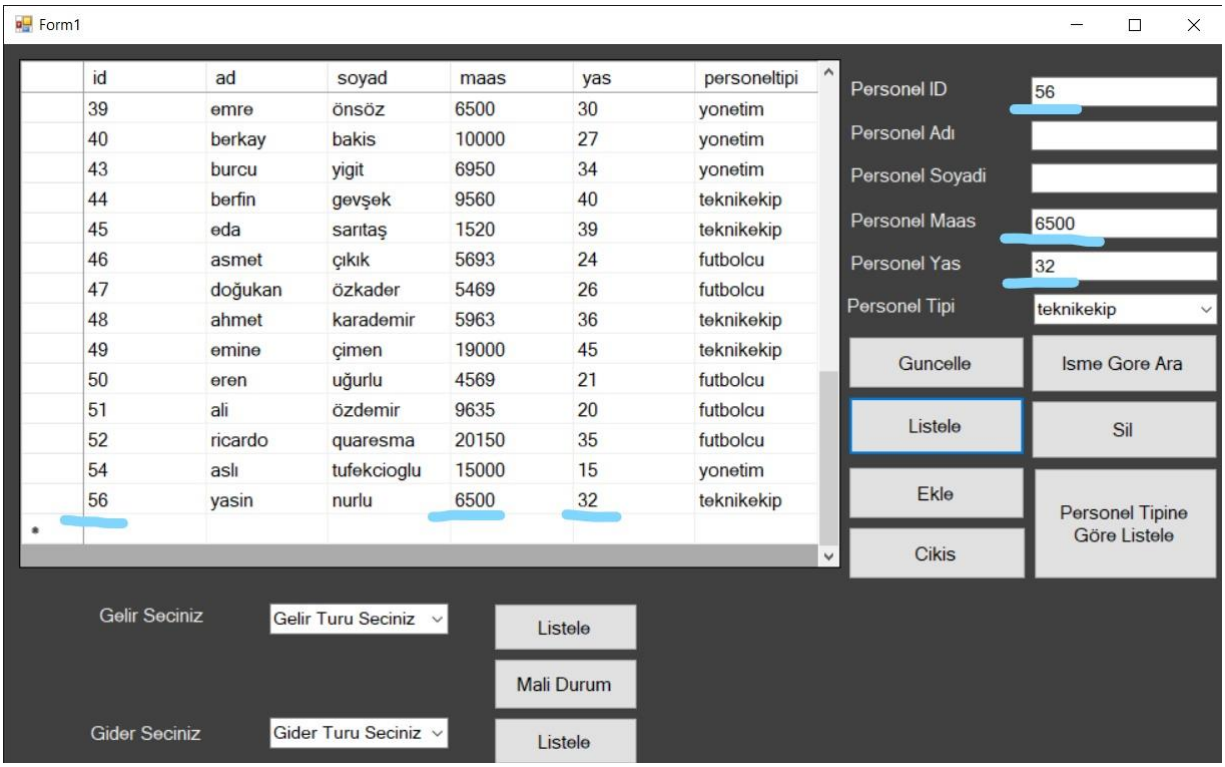


Güncelleme:

Güncellemeden önce:



Güncelledikten sonra:



KAYNAK KODLARI:

C# Programlama Dili:

```
using Npgsql;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace veri_tabani_proje
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void Form1_Load(object sender, EventArgs e)
        {
            cmbGelir.Items.Add("gelir");
            cmbGelir.Items.Add("yayingeliri");
            cmbGelir.Items.Add("stadgeliri");
            cmbGelir.Items.Add("mtstoregeliri");
            cmbGelir.Items.Add("sponsorgeliri");
            cmbGelir.Text = "Gelir Turu Seciniz";
            cmbGider.Items.Add("gider");
            cmbGider.Items.Add("stadgideri");
            cmbGider.Items.Add("saglikgideri");
            cmbGider.Text = "Gider Turu Seciniz";



            cmbPersonelTipi.Items.Add("yonetim");
            cmbPersonelTipi.Items.Add("teknikekip");
            cmbPersonelTipi.Items.Add("futbolcu");
            cmbPersonelTipi.Items.Add("doktor");
            cmbPersonelTipi.Items.Add("digerpersonel");
        }
        NpgsqlConnection baglanti = new NpgsqlConnection("server=localHost; port=5432;
Database=B191210004; user ID=postgres; password= Mmeliht1328.");
        private void btnListele_Click(object sender, EventArgs e)
        {
            string sorgu = "select * from personel";
            NpgsqlDataAdapter da = new NpgsqlDataAdapter(sorgu, baglanti);
            DataSet ds = new DataSet();
            da.Fill(ds);
            dataGridView1.DataSource = ds.Tables[0];
        }
        private void btnAra_Click(object sender, EventArgs e)
        {
            baglanti.Open();
            DataTable dt = new DataTable();
```

```
            NpgsqlDataAdapter ara = new NpgsqlDataAdapter("select * from
\"personel\"where \"ad\" like '%" + txtPersonelAdi.Text + "%' ", baglanti);
            ara.Fill(dt);
            baglanti.Close();
            dataGridView1.DataSource = dt;
        }
        private void btnPersonelEkle_Click(object sender, EventArgs e)  // bu
fonksiyon çalışmıyor....
        {
            baglanti.Open();
            NpgsqlCommand komutEkle = new NpgsqlCommand("insert into
personel(ad,soyad,maas,yas,personeltipi) values (@p1,@p2,@p3,@p4,@p5)", baglanti);
            komutEkle.Parameters.AddWithValue("@p1", txtPersonelAdi.Text);
            komutEkle.Parameters.AddWithValue("@p2", txtPersonelSoyadi.Text);
            komutEkle.Parameters.AddWithValue("@p3", int.Parse(txtPersonelMaas.Text));
            komutEkle.Parameters.AddWithValue("@p4",
Convert.ToInt32(txtPersonelYas.Text));
            komutEkle.Parameters.AddWithValue("@p5", cmbPersonelTipi.Text);
            komutEkle.ExecuteNonQuery();
            baglanti.Close();
            MessageBox.Show("Personel Ekleme Basariyla Gerceklesti");
            txtPersonelAdi.Text = null;
            txtPersonelMaas.Text = null;
            txtPersonelSoyadi.Text = null;
            txtPersonelYas.Text = null;
        }
        private void btnSilme_Click(object sender, EventArgs e)
        {
            baglanti.Open();
            NpgsqlCommand komutSil = new NpgsqlCommand("delete from personel where id=
@p1", baglanti);
            komutSil.Parameters.AddWithValue("@p1", int.Parse(txtPersonelID.Text));
            komutSil.ExecuteNonQuery();
            baglanti.Close();
            MessageBox.Show("Personel Silme Basariyla Gerceklesti");
        }
        private void btnGuncelle_Click(object sender, EventArgs e)
        {
            baglanti.Open();
            NpgsqlCommand komutGuncelle = new NpgsqlCommand("update personel set
maas=@p1 ,yas=@p2 ,personeltipi=@p3 where id=@p4", baglanti);
            komutGuncelle.Parameters.AddWithValue("@p1",
int.Parse(txtPersonelMaas.Text));
            komutGuncelle.Parameters.AddWithValue("@p2",
int.Parse(txtPersonelYas.Text));
            komutGuncelle.Parameters.AddWithValue("@p3",
cmbPersonelTipi.Text.ToString());
            komutGuncelle.Parameters.AddWithValue("@p4",
int.Parse(txtPersonelID.Text));
            komutGuncelle.ExecuteNonQuery();
            MessageBox.Show("Guncelleme Basariyla Gerceklesti");
            baglanti.Close();
        }
        private void btnGelirListele_Click(object sender, EventArgs e)
        {
            baglanti.Open();
            DataTable dt = new DataTable();
            NpgsqlDataAdapter ara = new NpgsqlDataAdapter("select * from " +
cmbGelir.Text , baglanti);
            ara.Fill(dt);
            baglanti.Close();
            dataGridView1.DataSource = dt;
```

```
        }
        private void btnGiderListele_Click(object sender, EventArgs e)
        {
            baglanti.Open();
            DataTable dt = new DataTable();
            NpgsqlDataAdapter ara = new NpgsqlDataAdapter("select * from " +
cmbGider.Text, baglanti);
            ara.Fill(dt);
            baglanti.Close();
            dataGridView1.DataSource = dt;
        }
        private void btnMaliDurum_Click(object sender, EventArgs e)
        {
            baglanti.Open();
            DataTable dt = new DataTable();
            NpgsqlDataAdapter ara = new NpgsqlDataAdapter("select * from malidurum",
baglanti);
            ara.Fill(dt);
            baglanti.Close();
            dataGridView1.DataSource = dt;
        }

        private void btnCikis_Click(object sender, EventArgs e)
        {
            Close();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            if (cmbPersonelTipi.Text == "yonetim")
            {
                baglanti.Open();
                DataTable dt = new DataTable();
                NpgsqlDataAdapter ara = new NpgsqlDataAdapter("select * from
"+cmbPersonelTipi.Text , baglanti);
                ara.Fill(dt);
                baglanti.Close();
                dataGridView1.DataSource = dt;
            }
            else if (cmbPersonelTipi.Text == "teknikekip")
            {
                baglanti.Open();
                DataTable dt = new DataTable();
                NpgsqlDataAdapter ara = new NpgsqlDataAdapter("select * from " +
cmbPersonelTipi.Text, baglanti);
                ara.Fill(dt);
                baglanti.Close();
                dataGridView1.DataSource = dt;
            }
            else if (cmbPersonelTipi.Text == "futbolcu")
            {
                baglanti.Open();
                DataTable dt = new DataTable();
                NpgsqlDataAdapter ara = new NpgsqlDataAdapter("select * from " +
cmbPersonelTipi.Text, baglanti);
                ara.Fill(dt);
                baglanti.Close();
                dataGridView1.DataSource = dt;
            }
            else if (cmbPersonelTipi.Text == "doktor")
            {
                baglanti.Open();
```

```
                DataTable dt = new DataTable();
                NpgsqlDataAdapter ara = new NpgsqlDataAdapter("select * from
saglikcilar" , baglanti);
                ara.Fill(dt);
                baglanti.Close();
                dataGridView1.DataSource = dt;
            }
            else if (cmbPersonelTipi.Text == "digerpersonel")
            {
                baglanti.Open();
                DataTable dt = new DataTable();
                NpgsqlDataAdapter ara = new NpgsqlDataAdapter("select * from " +
cmbPersonelTipi.Text, baglanti);
                ara.Fill(dt);
                baglanti.Close();
                dataGridView1.DataSource = dt;
            }
            else
                MessageBox.Show("Lutfen Personel Tipi seciniz");
        }
    }
}
```

YOUTUBE LİNKİ:

https://www.youtube.com/watch?v=LOICe5CN6Aw