# T.C.

# SAKARYA ÜNİVERSİTESİ

## BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

## BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

VERİTABANI YÖNETİM SİSTEMLERİ RAPORU

**B181210103 - Muhammet Furkan BİNGÜL**
**1.Öğretim – A Grubu**
muhammet.bingul@ogr.sakarya.edu.tr

# Programın Özeti

Sistemimize kullanıcımız oluşturduğu kullanıcı adı ve şifresi ile giriş yapması gerekiyor biz admin olarak giriş yapacağız bulunduğumuz senaryoda. Kullanıcı adı ve şifreyi girdikten sonra alınan veriler sql database'imizde bulunan kullanıcı adı ve şifre bilgileri ile kıyaslanacak kıyaslanan bilgiler doğru ise menümüze erişilebilecek fakat bilgiler yanlış ise programımız kendini otomatik kapatacaktır.

Giriş yaptıktan sonra menümüz karşımıza gelecek gelen menüde kullanıcı numarası ile bilgi sorgulama , kullanıcı numarası değiştirme, kullanıcı ekleme ve silme gibi işlemler yapacağız ve aynı zamanda bu işlemler bizim oluşturduğumuz veri tabanında da gerçekleşecektir.

# İlişkisel Şema

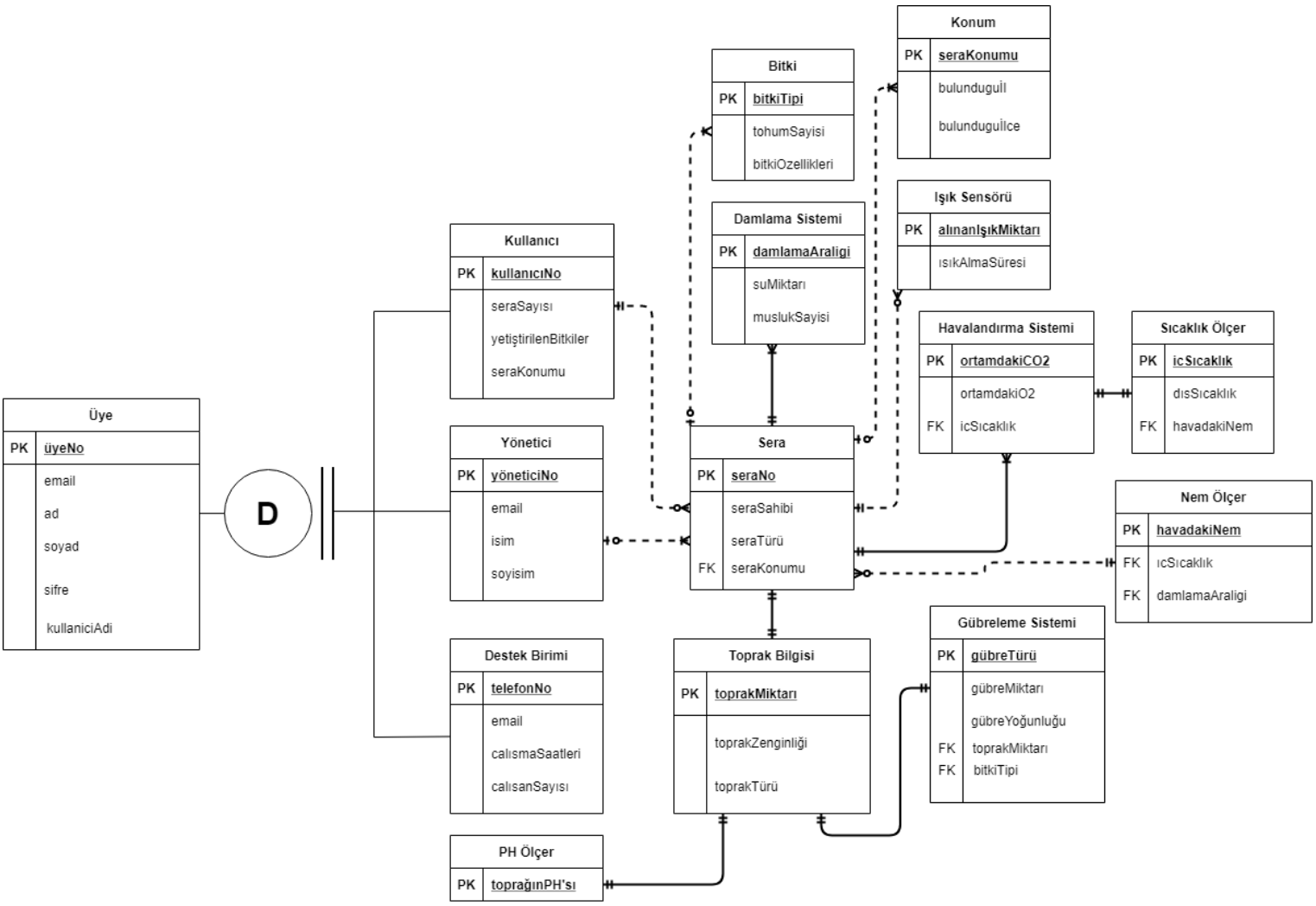
- Üye(**uyeNo : İnteger** , email : String , sifre : String , ad : String , soyad : String , kullaniciAdi : String )
- Kullanıcı(**kullanıcıNo : İnteger** , seraSayısı : İnteger , yetiştirilenBitkiler : String , seraKonumu : String )
- Yönetici(**yöneticiNo : İnteger** , email : String , isim : String , soyisim : String )
- Destek Birimi(**telefonNo : İnteger** , email : String , calismaSaatleri : Date , calisanSayisi : İnteger )
- Sera(**seraNo : İnteger** , seraSahibi : String , seraTürü : String , seraKonumu : String )
- Bitki(**bitkiTipi : String** , tohumSayısı : String , bitkiÖzellikler : String )
- Damlama Sistemi(**damlamaAraligi : İnteger** , suMiktarı : İnteger , muslukSayısı : İnteger )
- Toprak Bilgisi(**toprakMiktarı : İnteger** , toprakZenginliği : String , toprakTürü : String )
- PH Ölçer(**toprakPH'sı : İnteger**)

- Havalandırma Sistemi(**ortamdakiCO2 : String** , ortamdakiO2 : String , icSıcaklık : String )
- Sıcaklık Ölçer(**icSıcaklık : String** , dısSıcaklık : String , havadakiNem : String)
- Konum(**seraKonumu : String** , bulunduğuİl : String , bulunduğuİlce : String)
- Işık Sensörü(**alınanIsıkMiktarı : İnteger** , ısıkAlmaSüresi : İnteger)
- Gübreleme Sistemi(**gübreTürü : String** , gübreMiktarı : İnteger , gübreYoğunluğu : İnteger , toprakMiktarı : İnteger),
- Nem Ölçer(**havadakiNem : String** , icSıcaklık : İnteger , damlamaAralığı : İnteger)

# İş Kuralları

- Her üyenin bir üye numarası bulunur ve bu sadece o üyeye ait ve benzeri olmayan bir tanımlamadır.
- Üye olunabilmesi için kayıt olacak kişinin e posta bilgileri , isim , soyisim , kullanıcı adı ve şifre bilgileri eksiksiz ve doğru bir şekilde girilmelidir.
- Sisteme kullanıcı adı ve şifre bilgileri doğru bir şekilde girilerek giriş yapılır.
- Sistemimizde kullanıcı , yönetici ve destek birimi olmak üzere 3 üye tipi bulunur.
- Kullanıcılar sistemi seralarında kullanacak kişilerdir.
- Yöneticiler ise sistem denetimini gerçekleştiren şirketin çalışanlarıdır.
- Destek birimi ise kullanıcıların yaşadığı herhangibir sorun sonucunda ulaşmaları ve sorun hakkında bilgi alıp arıza kaydının yapılması için kurulmuş birimdir.
- Sisteme sahip olan kullanıcılar sera bilgilerini girmeleri sonucunda şirketimiz kullanıcının arazisine sera yapımına başlar.
- Birden fazla sera türümüz bulunmaktadır.

- Bir kullanıcı birden fazla seraya sahip olabilir.
- Bir seranın birden fazla sahibi olabilir (ortaklaşa kullanım).
- Seraların her birine özel olarak seraNo verilir ve gerekli işlemler bu numara üzerinden gerçekleştirilir.
- Seralar türüne göre birden fazla sisteme sahiptir.
- Bir sera birden fazla sisteme sahip olabilir.
- Bir sistem en az bir serada bulunmalıdır.
- Toprak Bilgisi sistemi gübreleme ve Ph sistemi ile güçlü bağ içerisindedir çünkü bu sistemler entegre çalışmaktadır hepsi bir bütündür.
- Damlama sistemi yetişen bitkinin türüne göre ayarlanır.
- Arazi büyüklüğüne göre su miktarı ve musluk sayısı değişkenlik gösterir.
- Havalandırma sistemi olmadan sıcaklık ölçer kurulabilir.
- Sıcaklık ölçer olmadan havalandırma sistemi kurulamaz.
- Nem ölçer sistemdeki sıcaklık ve damlama aralığına göre nem miktarını hesaplayarak sulama sistemi ile entegre çalışır bitkinin isteğine göre nem miktarı ayarlanır.
- Gübreleme sistemi ph ve toprak bilgisi olmadan kullanılamaz.
- Toprak bilgisi ve ph gübreleme sistemi olmadan kullanılamaz.
- Seralar bitki türleri bilgilerine göre düzenlenir.
- Bir sera bitki olmadan kurulamaz.
- Bir seraya bitki ekilmese de olur.
- Bütün sistem birbiri ile bileşke çalışır biri olmadan diğeri olabilir fakat bir işe yaramaz.
- Kullanıcılar sera sayılarına göre bilgilendirilir.
- Seraların gözetimini ve ayarlamasını yapan sistem mühendislerini yanı sıra kullanıcı isteği dahilinde kendi de kontrol edebilir.
- Uygulamanın sunduğu anlık bildirim sayesinde kullanıcı serasına ait bilgiyi an be an alabilir.




# SQL Kodları

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


-- Dumped from database version 12.3
-- Dumped by pg_dump version 12.3


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;

SET default_tablespace = '';

SET default_table_access_method = heap;

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

CREATE TABLE public."Bitki" (
    "bitkiTipi" text NOT NULL,
    "tohumSayısı" text NOT NULL,
    "bitkiÖzellikleri" text NOT NULL
);


ALTER TABLE public."Bitki" OWNER TO postgres;

--
-- Name: Damlama Sistemi; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."Damlama Sistemi" (
    "damlamaAraligi" integer NOT NULL,
    "suMiktarı" integer NOT NULL,
    "muslukSayisi" integer NOT NULL
);


ALTER TABLE public."Damlama Sistemi" OWNER TO postgres;

--
-- Name: Destek Birimi; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."Destek Birimi" (
    "telefonNo" integer NOT NULL,
    email character varying NOT NULL,
    "calismaSaatleri" date NOT NULL,
    "calisanSayisi" integer NOT NULL
);
```

```
ALTER TABLE public."Destek Birimi" OWNER TO postgres;

--
-- Name: Gübreleme Sistemi; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."Gübreleme Sistemi" (
    "gübreTürü" character varying NOT NULL,
    "gübreMiktarı" character varying NOT NULL,
    "gübreYoğunluğu" character varying NOT NULL,
    "toprakMiktarı" integer NOT NULL,
    "bitkiTipi" text NOT NULL
);


ALTER TABLE public."Gübreleme Sistemi" OWNER TO postgres;

--
-- Name: Havalandırma Sistemi; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."Havalandırma Sistemi" (
    "ortamdakiCO2" text NOT NULL,
    "ortamdakiO2" text NOT NULL,
    "icSıcaklık" text NOT NULL
);


ALTER TABLE public."Havalandırma Sistemi" OWNER TO postgres;

--
-- Name: Işık Sensörü; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."Işık Sensörü" (
    "alınanIşıkMiktarı" integer NOT NULL,
    "ışıkAlmaSüresi" integer NOT NULL
);
```

```
ALTER TABLE public."Işık Sensörü" OWNER TO postgres;

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

CREATE TABLE public."Konum" (
    "seraKonumu" text NOT NULL,
    "bulunduğuİl" text NOT NULL,
    "bulunduguİlce" text NOT NULL
);


ALTER TABLE public."Konum" OWNER TO postgres;

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

CREATE TABLE public."Kullanıcı" (
    "kullanıcıNo" character varying NOT NULL,
    "seraSayısı" character varying NOT NULL,
    "yetiştirilenBitkiler" character varying NOT NULL,
    "seraKonumu" character varying NOT NULL
);


ALTER TABLE public."Kullanıcı" OWNER TO postgres;

--
-- Name: Nem Ölçer; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."Nem Ölçer" (
    "havadakiNem" text NOT NULL,
    "icSıcaklık" text NOT NULL,
    "damlamaAralığı" integer NOT NULL
);


ALTER TABLE public."Nem Ölçer" OWNER TO postgres;
```

```
--
-- Name: PH Ölçer; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."PH Ölçer" (
    "toprağınPH'sı" integer NOT NULL
);


ALTER TABLE public."PH Ölçer" OWNER TO postgres;

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

CREATE TABLE public."Sera" (
    "seraNo" integer NOT NULL,
    "seraSahibi" text NOT NULL,
    "seraTürü" text NOT NULL,
    "seraKonumu" text NOT NULL
);


ALTER TABLE public."Sera" OWNER TO postgres;

--
-- Name: Sıcaklık Ölçer; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."Sıcaklık Ölçer" (
    "icSıcaklık" text NOT NULL,
    "dısSıcaklık" text NOT NULL,
    "havadakiNem" text NOT NULL
);


ALTER TABLE public."Sıcaklık Ölçer" OWNER TO postgres;

--
-- Name: Toprak Bilgisi; Type: TABLE; Schema: public; Owner: postgres
--
```

```
CREATE TABLE public."Toprak Bilgisi" (
    "toprakMiktarı" integer NOT NULL,
    "toprakZenginliği" text NOT NULL,
    "toprakTürü" text NOT NULL
);


ALTER TABLE public."Toprak Bilgisi" OWNER TO postgres;

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

CREATE TABLE public."Yönetici" (
    "yöneticiNo" integer NOT NULL,
    email text NOT NULL,
    isim text NOT NULL,
    soyisim text NOT NULL
);


ALTER TABLE public."Yönetici" OWNER TO postgres;

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

CREATE TABLE public."Üye" (
    "üyeNo" integer NOT NULL,
    ad text NOT NULL,
    soyad text NOT NULL,
    "kullaniciAdi" text NOT NULL,
    sifre text NOT NULL,
    email text
);


ALTER TABLE public."Üye" OWNER TO postgres;

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

```
COPY public."Bitki" ("bitkiTipi", "tohumSayısı", "bitkiÖzellikleri") FROM stdin;
\.


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

COPY public."Damlama Sistemi" ("damlamaAraligi", "suMiktarı", "muslukSayisi") FROM stdin;
\.


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

COPY public."Destek Birimi" ("telefonNo", email, "calismaSaatleri", "calisanSayisi") FROM stdin;
\.


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

COPY public."Gübreleme Sistemi" ("gübreTürü", "gübreMiktarı", "gübreYoğunluğu", "toprakMiktarı", "bitkiTipi") FROM stdin;
\.


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

COPY public."Havalandırma Sistemi" ("ortamdakiCO2", "ortamdakiO2", "icSıcaklık") FROM stdin;
\.


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

COPY public."Işık Sensörü" ("alınanIşıkMiktarı", "ışıkAlmaSüresi") FROM stdin;
```

\.

--

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

--

COPY public."Konum" ("seraKonumu", "bulunduğuİl", "bulunduguİlce") FROM stdin;

\.

--

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

--

COPY public."Kullanıcı" ("kullanıcıNo", "seraSayısı", "yetiştirilenBitkiler", "seraKonumu") FROM stdin;

1111	5	domates	erzurum

2222	10	salatal�k	erzurum

\.

--

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

--

COPY public."Nem Ölçer" ("havadakiNem", "icSıcaklık", "damlamaAralığı") FROM stdin;

\.

--

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

--

COPY public."PH Ölçer" ("toprağınPH'sı") FROM stdin;

\.

--

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

--

COPY public."Sera" ("seraNo", "seraSahibi", "seraTürü", "seraKonumu") FROM stdin;

```
\.


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

COPY public."Sıcaklık Ölçer" ("icSıcaklık", "dısSıcaklık", "havadakiNem") FROM stdin;
\.


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

COPY public."Toprak Bilgisi" ("toprakMiktarı", "toprakZenginliği", "toprakTürü") FROM stdin;
\.


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

COPY public."Yönetici" ("yöneticiNo", email, isim, soyisim) FROM stdin;
\.


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

COPY public."Üye" ("üyeNo", ad, soyad, "kullaniciAdi", sifre, email) FROM stdin;
123	furkan	bingül	furkan	1111	mfurkanbingul@gmail.com
111	sefa\n	bingül	sefa	2222	mfurkanbingul@gmail.com
222	admin	admin	admin	password	mfurkanbingul@gmail.com
\.


--
-- Name: Bitki Bitki_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--
```

```
ALTER TABLE ONLY public."Bitki"
    ADD CONSTRAINT "Bitki_pkey" PRIMARY KEY ("bitkiTipi");


--
-- Name: Damlama Sistemi Damlama Sistemi_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Damlama Sistemi"
    ADD CONSTRAINT "Damlama Sistemi_pkey" PRIMARY KEY ("damlamaAraligi");


--
-- Name: Destek Birimi Destek Birimi_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Destek Birimi"
    ADD CONSTRAINT "Destek Birimi_pkey" PRIMARY KEY ("telefonNo");


--
-- Name: Gübreleme Sistemi Gübreleme Sistemi_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Gübreleme Sistemi"
    ADD CONSTRAINT "Gübreleme Sistemi_pkey" PRIMARY KEY ("gübreTürü");


--
-- Name: Havalandırma Sistemi Havalandırma Sistemi_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Havalandırma Sistemi"
    ADD CONSTRAINT "Havalandırma Sistemi_pkey" PRIMARY KEY ("ortamdakiCO2");


--
-- Name: Işık Sensörü Işık Sensörü_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Işık Sensörü"
    ADD CONSTRAINT "Işık Sensörü_pkey" PRIMARY KEY ("alınanIşıkMiktarı");
```

```
--
-- Name: Konum Konum_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Konum"
    ADD CONSTRAINT "Konum_pkey" PRIMARY KEY ("seraKonumu");


--
-- Name: Kullanıcı Kullanıcı_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Kullanıcı"
    ADD CONSTRAINT "Kullanıcı_pkey" PRIMARY KEY ("kullanıcıNo");


--
-- Name: Nem Ölçer Nem Ölçer_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Nem Ölçer"
    ADD CONSTRAINT "Nem Ölçer_pkey" PRIMARY KEY ("havadakiNem");


--
-- Name: PH Ölçer PH Ölçer_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."PH Ölçer"
    ADD CONSTRAINT "PH Ölçer_pkey" PRIMARY KEY ("toprağınPH'sı");


--
-- Name: Sera Sera_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Sera"
    ADD CONSTRAINT "Sera_pkey" PRIMARY KEY ("seraNo");
```

```
--
-- Name: Sıcaklık Ölçer Sıcaklık Ölçer_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Sıcaklık Ölçer"
    ADD CONSTRAINT "Sıcaklık Ölçer_pkey" PRIMARY KEY ("icSıcaklık");


--
-- Name: Toprak Bilgisi Toprak Bilgisi_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Toprak Bilgisi"
    ADD CONSTRAINT "Toprak Bilgisi_pkey" PRIMARY KEY ("toprakMiktarı");


--
-- Name: Yönetici Yönetici_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Yönetici"
    ADD CONSTRAINT "Yönetici_pkey" PRIMARY KEY ("yöneticiNo");


--
-- Name: Üye Üye_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Üye"
    ADD CONSTRAINT "Üye_pkey" PRIMARY KEY ("üyeNo");


--
-- Name: Gübreleme Sistemi Gübreleme Sistemi_bitkiTipi_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Gübreleme Sistemi"
    ADD CONSTRAINT "Gübreleme Sistemi_bitkiTipi_fkey" FOREIGN KEY ("bitkiTipi") REFERENCES public."Bitki"("bitkiTipi");


--
-- Name: Gübreleme Sistemi Gübreleme Sistemi_toprakMiktarı_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres
```

```
--

ALTER TABLE ONLY public."Gübreleme Sistemi"
    ADD CONSTRAINT "Gübreleme Sistemi_toprakMiktarı_fkey" FOREIGN KEY ("toprakMiktarı") REFERENCES public."Toprak Bilgisi"("toprakMiktarı");


--
-- Name: Havalandırma Sistemi Havalandırma Sistemi_icSıcaklık_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Havalandırma Sistemi"
    ADD CONSTRAINT "Havalandırma Sistemi_icSıcaklık_fkey" FOREIGN KEY ("icSıcaklık") REFERENCES public."Sıcaklık Ölçer"("icSıcaklık") NOT VALID;


--
-- Name: Nem Ölçer Nem Ölçer_damlamaAralığı_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Nem Ölçer"
    ADD CONSTRAINT "Nem Ölçer_damlamaAralığı_fkey" FOREIGN KEY ("damlamaAralığı") REFERENCES public."Damlama Sistemi"("damlamaAraligi");


--
-- Name: Nem Ölçer Nem Ölçer_icSıcaklık_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Nem Ölçer"
    ADD CONSTRAINT "Nem Ölçer_icSıcaklık_fkey" FOREIGN KEY ("icSıcaklık") REFERENCES public."Sıcaklık Ölçer"("icSıcaklık");


--
-- Name: Sera Sera_seraKonumu_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Sera"
    ADD CONSTRAINT "Sera_seraKonumu_fkey" FOREIGN KEY ("seraKonumu") REFERENCES public."Konum"("seraKonumu");


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

```
run:
Kullanici Adi :
admin
Sifre :
password
```

**Kullanıcı adı ve şifre bilgilerimizi doğru bir şekilde giriyoruz.**

```
run:
Kullanici Adi :
admin
Sifre :
password
Veritabanına bağlandı!
KULLANICI VE ŞİFRE DOĞRU!
Veritabanına bağlandı!
Yapmak İstediğiniz İşlemi Seçiniz...
1 => KULLANICI NO SORGULAMA...
2 => KULLANICI NO DEĞİŞTİRME...
3 => KULLANICI SİLME...
4 => KULLANICI EKLEME...
5 => UYGULAMADAN ÇIKIŞ...
```

**Doğru girilen bilgilerden sonra menümüz açılıyor.**

**Menüden önce bir kullanıcı ekleyelim sonra onu sorgulatalım sonra güncelleyelim ve silelim.**

```
4
Kullanıcı No Giriniz : 2525
Sera Sayısı Giriniz : 20
Yetiştirilecek Bitkiyi Giriniz : havuc
Seranın Konumunu Giriniz : ısparta
Ekleme Başarılı...
Veritabanına bağlandı!
Yapmak İstediğiniz İşlemi Seçiniz...
1 => KULLANICI NO SORGULAMA...
2 => KULLANICI NO DEĞİŞTİRME...
3 => KULLANICI SİLME...
4 => KULLANICI EKLEME...
5 => UYGULAMADAN ÇIKIŞ...
```

**Ekleme işleminden sonra yeniden menümüz geliyor şimdi oluşturduğumuz kullanıcı no ile sorgulama yapalım.**

```
1
SORGULANACAK KULLANICI NO GİRİNİZ...
2525
SORGULANIYOR...
Kullanici NO : 2525
Sahip Olunan Sera Sayısı : 20
Yetistirilen Bitki Türleri : havuc
Seranın Bulunduğu Konum : □sparta
Veritabanına bağlandı!
Yapmak İstediğiniz İşlemi Seçiniz...
1 => KULLANICI NO SORGULAMA...
2 => KULLANICI NO DEĞİŞTİRME...
3 => KULLANICI SİLME...
4 => KULLANICI EKLEME...
5 => UYGULAMADAN ÇIKIŞ...
```

**Sorgulama işlemini yaptıktan sonra kullanıcı numarasını değiştirelim.**

```
2
Güncellenecek Kullanıcı No'yu Giriniz...
2525
Atanacak Yeni Kullanıcı No'yu Giriniz...
5454
Güncellendi...Veritabanına bağlandı!
Yapmak İstediğiniz İşlemi Seçiniz...
1 => KULLANICI NO SORGULAMA...
2 => KULLANICI NO DEĞİŞTİRME...
3 => KULLANICI SİLME...
4 => KULLANICI EKLEME...
5 => UYGULAMADAN ÇIKIŞ...
```

**2525 olan kullanıcı no muzu 5454 olarak değiştirdik şimdi 5454 olarak aratarak bilgilerin doğruluğunu sağlayalım.**

```
1
SORGULANACAK KULLANICI NO GİRİNİZ...
5454
SORGULANIYOR...
Kullanici NO : 5454
Sahip Olunan Sera Sayısı : 20
Yetistirilen Bitki Türleri : havuc
Seranın Bulunduğu Konum : □sparta
Veritabanına bağlandı!
Yapmak İstediğiniz İşlemi Seçiniz...
1 => KULLANICI NO SORGULAMA...
2 => KULLANICI NO DEĞİŞTİRME...
3 => KULLANICI SİLME...
4 => KULLANICI EKLEME...
5 => UYGULAMADAN ÇIKIŞ...
```

**Görüldüğü üzere kullanıcı numaramız başarılı bir şekilde değiştirildi son olarak kullanıcımızı siliyoruz.**

```
3
Silmek İstediğiniz Kullanıcı No'yu Giriniz...
5454
Silme Başarılı...
Veritabanına bağlandı!
Yapmak İstediğiniz İşlemi Seçiniz...
1 => KULLANICI NO SORGULAMA...
2 => KULLANICI NO DEĞİŞTİRME...
3 => KULLANICI SİLME...
4 => KULLANICI EKLEME...
5 => UYGULAMADAN ÇIKIŞ...
```

**Kullanıcımız başarılı bir şekilde silindi.**

**Yeniden arattığımız zaman bulunamayacaktır.**

```
1
SORGULANACAK KULLANICI NO GİRİNİZ...
5454
SORGULANIYOR...
Veritabanına bağlandı!
Yapmak İstediğiniz İşlemi Seçiniz...
1 => KULLANICI NO SORGULAMA...
2 => KULLANICI NO DEĞİŞTİRME...
3 => KULLANICI SİLME...
4 => KULLANICI EKLEME...
5 => UYGULAMADAN ÇIKIŞ...

1
SORGULANACAK KULLANICI NO GİRİNİZ...
2525
SORGULANIYOR...
Veritabanına bağlandı!
Yapmak İstediğiniz İşlemi Seçiniz...
1 => KULLANICI NO SORGULAMA...
2 => KULLANICI NO DEĞİŞTİRME...
3 => KULLANICI SİLME...
4 => KULLANICI EKLEME...
5 => UYGULAMADAN ÇIKIŞ...
```

**Görüldüğü üzere eski kullanıcı numarası ile de yeni kullanıcı numarası ile de arattığımızda kullanıcımız bulunamıyor.**

## Program Kodlarım:

## VTYSSeraSistemi :

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package vtys.sera.sistemi;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
```

```
import java.util.Scanner;
import java.sql.*;

/**
 *
 * @author Muhammet Furkan Bingül
 */
public class VTYSSeraSistemi {

   /**
    * @param args the command line arguments
    */
   public static void main(String[] args) {
      // TODO code application logic here
      String Sifre,
            Kullaniciadi,
            isimGirdisi,
            isimGirdi,
            veriAkisi,

            girisDeger;
      int kullanici_secim = 0;

      //kullanıcı adı ve şifreyi sorgulamak için DBsorgu sınıfını kullanıyoruz.

      DBsorgu sorgu = DBsorgu.getSorgu();

      //kullanıcı adı için
      Scanner KA = new Scanner(System.in);
      Scanner SF = new Scanner(System.in);

      //Kullanıcıdan değer aldığımız kısım.
      System.out.println("Kullanici Adi :");
      Kullaniciadi = KA.next();
      System.out.println("Sifre :");
      Sifre = SF.next();

      //Sql bilgilerini aldığımız kısım.
      sorgu.sqlSorgu();
```

```
 //Girilen değerlerle sql dekileri karşılaştırdığımız fonksiyonu çağırıyoruz.
   sorgu.sorgulama(Kullaniciadi,Sifre);



while (kullanici_secim != 5){


 try
 {  //Bağlantı kurulumu.
    Connection conn = DriverManager.getConnection("jdbc:postgresql://localhost:5432/vtysdb","postgres","2391a3a2343b58d9.");
    if (conn != null)
       System.out.println("Veritabanına bağlandı!");
    else
       System.out.println("Bağlantı girişimi başarısız!");



 System.out.println("Yapmak İstediğiniz İşlemi Seçiniz...");
 System.out.println("1 => KULLANICI NO SORGULAMA...");
 System.out.println("2 => KULLANICI NO DEĞİŞTİRME...");
 System.out.println("3 => KULLANICI SİLME...");
 System.out.println("4 => KULLANICI EKLEME...");
 System.out.println("5 => UYGULAMADAN ÇIKIŞ... \n");

 //kullanıcının seçimini uygulamamız için kullanıcıdan seçim için girdi alıyoruz.
 Scanner giris = new Scanner(System.in);
 kullanici_secim=giris.nextInt();

 //kullanıcının seçeneğine göre uygulamamızın işlemi uygulaması için switch case e giriyor.
 switch (kullanici_secim) {

    case 1 :

    System.out.println("SORGULANACAK KULLANICI NO GİRİNİZ...");
    Scanner girr = new Scanner(System.in);
    girisDeger = girr.nextLine();

    veriAkisi= "SELECT * FROM \"Kullanıcı\" WHERE \"kullanıcıNo\" = \'";
    veriAkisi = veriAkisi + girisDeger + "\'";
```

```
//Sorgu Çalıştırma.
Statement stmt = conn.createStatement();
ResultSet result = stmt.executeQuery(veriAkisi);

//Bağlantı Sonlandırma.
conn.close();

System.out.println("SORGULANIYOR...");



String
   kullanıcıNo,
   seraSayısı,
   yetiştirilenBitkiler,
   seraKonumu;

      while(result.next())
      {
         //KAYITTAKİ VERİLERİ DEĞİŞKENLERİNE ATAMA.
         kullanıcıNo  = result.getString("kullanıcıNO");
         seraSayısı = result.getString("seraSayısı");
         yetiştirilenBitkiler = result.getString("yetiştirilenBitkiler");
         seraKonumu = result.getString("seraKonumu");

         //EKRANA YAZDIRMA.
         System.out.println("Kullanici NO : "+ kullanıcıNo);
         System.out.println("Sahip Olunan Sera Sayısı : " + seraSayısı);
         System.out.println("Yetistirilen Bitki Türleri : " + yetiştirilenBitkiler);
         System.out.println("Seranın Bulunduğu Konum : " + seraKonumu);

      }
      result.close();
      stmt.close();

break;



case 2 :
   System.out.println("Güncellenecek Kullanıcı No'yu Giriniz...");
```

```
    Scanner giri = new Scanner(System.in);
    isimGirdisi = giri.nextLine();

    System.out.println("Atanacak Yeni Kullanıcı No'yu Giriniz...");
    Scanner giriş = new Scanner(System.in);
    isimGirdi = giriş.nextLine();

    veriAkisi= "UPDATE \"Kullanıcı\" SET \"kullanıcıNo\" = \'";
    veriAkisi = veriAkisi + isimGirdi + "\' WHERE \"kullanıcıNo\" = \'" + isimGirdisi + "\'";

    //Sorgu Çalıştırma.
    Statement stmt1 = conn.createStatement();
    stmt1.executeUpdate(veriAkisi);
    System.out.print("Güncellendi...");

    //Bağlantı Sonlandırma.
    conn.close();
    stmt1.close();
    break;


case 3 :
   System.out.println("Silmek İstediğiniz Kullanıcı No'yu Giriniz...");
   Scanner girri = new Scanner(System.in);
   isimGirdisi = girri.nextLine();
   veriAkisi = "DELETE FROM \"Kullanıcı\" WHERE \"kullanıcıNo\" =\'" + isimGirdisi + "\'";

   Statement stmt2 = conn.createStatement();
   stmt2.executeUpdate(veriAkisi);
   System.out.println("Silme Başarılı...");

   //Bağlantı Sonlandırma.
    conn.close();
    stmt2.close();
    break;


case 4 :
    //Girilecek değerler için tanımlanan değişkenler.
```

```
        String kullanıcıno ,
            serSay ,
            yetisBit ,
            konum ;

        System.out.print("Kullanıcı No Giriniz : ");
        Scanner d = new Scanner(System.in);
        kullanıcıno = d.nextLine();

        System.out.print("Sera Sayısı Giriniz : ");
        Scanner scan = new Scanner(System.in);
        serSay = scan.nextLine();

        System.out.print("Yetiştirilecek Bitkiyi Giriniz : ");
        Scanner t = new Scanner(System.in);
        yetisBit = t.nextLine();

        System.out.print("Seranın Konumunu Giriniz : ");
        Scanner a = new Scanner(System.in);
        konum = a.nextLine();

        veriAkisi ="INSERT INTO \"Kullanıcı\"\n(\"kullanıcıNo\", \"seraSayısı\","+ " \"yetiştirilenBitkiler\", \"seraKonumu\")"+ "\nVALUES ('" +
kullanıcıno + "', '" + serSay +"', '" + yetisBit + "', '" + konum + "')";

        Statement stmt3 = conn.createStatement();
        stmt3.executeUpdate(veriAkisi);

        System.out.println("Ekleme Başarılı...");

        //Bağlantı Sonlandırma.
        conn.close();
        stmt3.close();

        break;

      case 5 :
        //Kullanıcı uygulamadan çıkamak istediğinde uygulamamızı kapatıyoruz.
        System.out.println("!!!PROGRAM KAPATILIYOR!!!");
        System.exit(0);
```

```
            break;


        default :
            //eğer menü ışında bir değer girilirse hata mesajı yazdırıyoruz.
            System.out.println("Hatali secim! 1, 2 ya da 3'e basiniz.");
            break;
        }


    }


          catch (Exception e)
          {
             e.printStackTrace();
          }
       }

}
}
```

# DBsorgu Sınıfı:

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package vtys.sera.sistemi;
import java.sql.*;

/**
 *
 * @author Muhammet Furkn Bingül
 */
public class DBsorgu {

    private static final DBsorgu sorgu = new DBsorgu();

    public static DBsorgu getSorgu(){
```

```
    return sorgu;

}
    //sql deki kullanıcı adı ve şifrelerin atılacağı diziler bunlar.
        String kullaniciadi[];
        String kullanicisifresi[];

        int i = 0;
        int dSay = 0;
public void sqlSorgu(){

    try
    {   //Bağlantı kurulumu.
        Connection conn = DriverManager.getConnection("jdbc:postgresql://localhost:5432/vtysdb","postgres","2391a3a2343b58d9.");

        if (conn != null)
            System.out.println("Veritabanına bağlandı!");
        else
            System.out.println("Bağlantı girişimi başarısız!");


        String sql= "SELECT \"kullaniciAdi\" , \"sifre\" FROM \"Üye\"";

        //Sorgu çalıştırma.
        Statement stmt = conn.createStatement();
        ResultSet rs = stmt.executeQuery(sql);

        //Bağlantı sonlandırma.
        conn.close();

       String kullanici= null;
       String sifre=null;

       //Dizilerimizin uzunluğu.
        kullaniciadi = new String[150];
        kullanicisifresi = new String[150];

        while(rs.next())
        {
```

```
            //Kayda ait alan değerlerini değişkene atanıyor.
            kullanici  = rs.getString("kullaniciAdi");
            sifre = rs.getString("sifre");

            //Sql serverden çekilen kullanıcı adı ve şifreler diziye atanıyor.
            kullaniciadi[i] = kullanici;
            kullanicisifresi[i] = sifre;
            i++;

        }

        //Kaynakları serbest bırakıyor.
        rs.close();
        stmt.close();

    }

    catch (Exception e) {
        e.printStackTrace();
    }

}

//Sql den çektiğimiz değerleri kullanıcının girdileri ile karşılaştırma.
public void sorgulama(String x,String y){

    for(int z = 0 ; z<i; z++){

        //girilen değerlerle sql deki değerlerin karşılaştırılması.
            if(x.equals(kullaniciadi[z]) && y.equals(kullanicisifresi[z])){
              //eğer girilen değerler eşleşiyorsa sayacımız artıyor.
                dSay++;

            }

            }
    //sayacımız bir e eşit oluğu zaman demek ki girilen değerler doğru ve ekrana bildirimi yazdırılır.
            if(dSay == 1)
            System.out.println("KULLANICI VE ŞİFRE DOĞRU!");
```

```
  //eğer sayacımız 0 a eşitse demek ki girilen değerler eşleşmemiştir ekrana yanlış bildirimi yazdırılır.
            if(dSay==0){
              System.out.println("KULLANICI VE ŞİFRE YANLIŞ!");
              System.out.println("!!!PROGRAM KAPATILIYOR!!!");
              System.exit(0);
          }

          }

}
```

**Video Linki: https://youtu.be/enp-tG4_dJE**

**GitHUB Kodların Linki :** https://github.com/mfbingul/VTYS