



SAKARYA
ÜNİVERSİTESİ

SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
VERİTABANI YÖNETİM SİSTEMLERİ DERSİ

AD SOYAD: BARIŞ YILMAZ

ÖĞRENCİ NO: G191210303

ÖĞRENİM TÜRÜ/ŞUBE: 2/B

E-POSTA: baris.yilmaz5@ogr.sakarya.edu.tr

İş Kuralları

- Kullanıcının kullanıcı adı, e-mail, şifre, isim, lokasyon, şirket adı, biyografi, url, durum mesajı ve hesap oluşturulma tarihi bulunur.
- Kullanıcılar takımlı veya takımsız kullanıcı olarak ayrılırlar.
- Takımlı kullanıcı takım lideri olabilir.
- Takım lideri takımdan üye çıkarabilme ve ekleyebilme paneline sahiptir.
- Kullanıcıların takipçileri ve takip ettiği kullanıcılar bulunur.
- Kullanıcılar organizasyon oluşturabilir veya dahil olabilir.
- Organizasyonun isim, iletişim e-mail, açıklama, url, lokasyon, tip ve oluşturulma tarihi bulunur.
- Organizasyonlarda takımlar oluşturulabilir veya dahil olunabilir.
- Takımın isim, açıklama, gizlilik durumu ve oluşturulma tarihi bulunur.
- Kullanıcılar repository oluşturabilirler.
- Repositorynin isim, açıklama, url, dil, oluşturulma tarihi ve son push tarihi bulunur.
- Repositorylerin watcher, starrer ve forkerları bulunur.
- Repositorylere pull request, proje, discussion, wiki, branch ve issue açılabilir.
- Pull requestin başlık, mesaj, oluşturulma tarihi ve kapanış tarihi bulunur.
- Projenin isim, açıklama, oluşturulma tarihi ve kapanış tarihi bulunur.
- Wikinin başlık, metin, mesaj ve oluşturulma tarihi bulunur.
- Discussion başlık, metin, kategori ve oluşturulma tarihi bulunur.
- Branchin başlık ve güncellenme tarihi bulunur.
- Issuenun başlık, metin, oluşturulma tarihi ve kapanış tarihi bulunur.
- Pull requestin yorumları bulunur.
- Projenin columnları bulunur.
- Discussionun yorumları bulunur.
- Branchin commitleri bulunur.
- Commitin yorumları bulunur.
- Issuenun yorumları, milestoneları ve labelları bulunur.
- Milestonun başlık, açıklama, tarih, güncellenme tarihi, oluşturulma tarihi ve kapanış tarihi bulunur.
- Labelın isim, açıklama ve renk kodu bulunur.
- Bir kullanıcı hiç veya çok sayıda organizasyona sahip olabilir. Bir organizasyonun da bir veya daha çok sayıda kullanıcısı olabilir.
- Bir organizasyon hiç veya çok sayıda takıma sahip olabilir. Bir takımın yalnızca bir organizasyonu olabilir.
- Bir kullanıcı hiç veya çok sayıda repositoryye sahip olabilir. Bir organizasyonun yalnızca bir kullanıcısı olabilir.
- Bir repository hiç veya çok sayıda watch sahip olabilir. Bir watchun yalnızca bir repositorysi bulunur.
- Bir repository hiç veya çok sayıda star sahip olabilir. Bir starın yalnızca bir repositorysi bulunur.
- Bir repository hiç veya çok sayıda fork sahip olabilir. Bir forkun yalnızca bir repositorysi bulunur.

- Bir repository hiç veya çok sayıda wikiye sahip olabilir. Bir wikinin yalnızca bir repositorysi bulunur.
- Bir repository hiç veya çok sayıda pull requeste sahip olabilir. Bir pull requestin yalnızca bir repositorysi bulunur.
- Bir pull request hiç veya çok sayıda yoruma sahip olabilir. Bir yorumun yalnızca bir pull requesti bulunur.
- Bir repository hiç veya çok sayıda discussiona sahip olabilir. Bir discussionın yalnızca bir repositorysi bulunur.
- Bir discussion hiç veya çok sayıda yoruma sahip olabilir. Bir yorumun yalnızca bir repositorysi bulunur.
- Bir repository hiç veya çok sayıda projeye sahip olabilir. Bir projenin yalnızca bir repositorysi bulunur.
- Bir proje hiç veya çok sayıda columna sahip olabilir. Bir columnun yalnızca bir projesi bulunur.
- Bir repository hiç veya çok sayıda branche sahip olabilir. Bir branchin yalnızca bir repositorysi bulunur.
- Bir branchin hiç veya çok sayıda commite sahip olabilir. Bir commitin yalnızca bir branchi bulunur.
- Bir commit hiç veya çok sayıda yoruma sahip olabilir. Bir yorumun yalnızca bir commiti bulunur.
- Bir repository hiç veya çok sayıda issueya sahip olabilir. Bir issuenun yalnızca bir repositorysi bulunur.
- Bir issue hiç veya çok sayıda yoruma sahip olabilir. Bir yorumun yalnızca bir issuesu bulunur.
- Bir issue hiç veya çok sayıda milestonea sahip olabilir. Bir milestoneun yalnızca bir issuesu bulunur.
- Bir issue hiç veya çok sayıda labela sahip olabilir. Bir labelın yalnızca bir issuesu bulunur.

İlişkisel Şema

- Kullanici(**kullaniciNo: int**, kullanıcıAdi: varchar, email: varchar, sifre: varchar, isim: varchar, lokasyon: varchar, şirketAdi: varchar, biyografi: text, url: varchar, durumMesaji: varchar, olusturulmaTarihi: date)
- TakimsizUye(**kullaniciNo: int**)
- TakimliUye(**kullaniciNo: int**, **takimNo: int**, **takimPanel: varchar**)
- TakimLideri(**kullaniciNo: int**, **takimKontrolPaneli: varchar**)
- NormalUye(**kullaniciNo: int**)
- Takip(**takipciNo: int**, **takipEdenNo: int**, **takipTarihi: date**)
- OrganizasyonUye(**orgNo: int**, **kullaniciNo: int**, **organizasyonPanel: varchar**)
- Organizasyon(**orgNo: int**, isim: varchar, iletisimMail: varchar, aciklama: text, url: varchar, lokasyon: varchar, tip: varchar, olusturulmaTarihi: date)
- Takim(**takimNo: int**, **orgNo: int**, isim: varchar, aciklama: text, gizlilikDurumu: varchar, olusturulmaTarihi: date)
- Repository(**repoNo: int**, **sahipNo: int**, **orgNo: int**, isim: varchar, aciklama: text, url: varchar, dil: varchar, olusturulmaTarihi: date, sonPushTarihi: date)
- Starrer(**repoNo: int**, **kullaniciNo: int**, **olusturulmaTarihi: date**)
- Watcher(**repoNo: int**, **kullaniciNo: int**, **olusturulmaTarihi: date**)
- Forker(**repoNo: int**, **kullaniciNo: int**, **olusturulmaTarihi: date**)
- Wiki(**wikiNo: int**, **repoNo: int**, **kullaniciNo: int**, baslik: varchar, metin: text, mesaj: text, olusturulmaTarihi: date)
- PullRequest(**pullReqNo: int**, **repoNo: int**, **kullaniciNo: int**, baslik: varchar, mesaj: text, olusturulmaTarihi: date, kapanisTarihi: date)
- PullRequestYorum(**pullReqYorumNo: int**, **pullReqNo: int**, **kullaniciNo: int**, mesaj: text, olusturulmaTarihi: date)
- Proje(**projeNo: int**, **repoNo: int**, isim: varchar, aciklama: text, olusturulmaTarihi: date, kapanisTarihi: date)
- Column(**columnNo: int**, **projeNo: int**, isim: varchar)
- Branch(**branchNo: int**, **repoNo: int**, baslik: varchar, guncellenmeTarihi: date)
- Commit(**commitNo: int**, **branchNo: int**, **kullaniciNo: int**, baslik: varchar, metin: text, olusturulmaTarihi: date)
- CommitYorum(**commitYorumNo: int**, **commitNo: int**, **kullaniciNo: int**, mesaj: text, olusturulmaTarihi: date)
- Discussion(**discussionNo: int**, **repoNo: int**, **kullaniciNo: int**, baslik: varchar, metin: text, kategori: varchar, olusturulmaTarihi: date)
- DiscussionYorum(**discussionYorumNo: int**, **discussionNo: int**, **kullaniciNo: int**, mesaj: text, olusturulmaTarihi: date)
- Issue(**issueNo: int**, **repoNo: int**, **sahipNo: int**, baslik: varchar, metin: text, olusturulmaTarihi: date, kapanisTarihi: date)
- IssueYorum(**issueYorumNo: int**, **issueNo: int**, **kullaniciNo: int**, mesaj: text, olusturulmaTarihi: date)
- Label(**labelNo: int**, **issueNo: int**, isim: varchar, aciklama: text, renkKodu: varchar)
- Milestone(**milestoneNo: int**, **issueNo: int**, baslik: varchar, aciklama: text, kapanisTarihi: date, guncellenmeTarihi: date, olusturulmaTarihi: date, tarih: date)

The diagram illustrates the database structure for a GitHub-like system. It includes the following entities and their attributes:

- Takip**: takipciNo, takipEdenNo, takipTarihi
- Kullanici**: kullaniciNo (PK), kullanicIAdi, email, sifre, isim, lokasyon, sirketAdi, biyografi, url, durumMesaji, olusturulmaTarihi
- Organizasyon Uye**: orgNo (PK FK), kullaniciNo (FK), organizasyonPanel
- Organizasyon**: orgNo (PK), isim, iletisimMail, aciklama, url, lokasyon, tip, olusturulmaTarihi
- Takim**: takimNo (PK), orgNo (FK), isim, aciklama, gizlilikDurumu, olusturulmaTarihi
- Takimsiz Uye**: kullaniciNo (FK)
- Takimli Uye**: kullaniciNo (FK), takimNo (FK), takimPanel
- Takim Lideri**: kullaniciNo (FK), takimKontrolPaneli
- Normal Uye**: kullaniciNo (FK)
- Repository**: repoNo (PK), sahipNo (FK), orgNo (FK), isim, aciklama, dil, olusturulmaTarihi, sonPushTarihi
- Watcher**: repoNo (FK), kullaniciNo (FK), olusturulmaTarihi
- Starrer**: repoNo (FK), kullaniciNo (FK), olusturulmaTarihi
- Forker**: repoNo (FK), kullaniciNo (FK), olusturulmaTarihi
- Branch**: branchNo (PK), repoNo (FK), baslik, guncellenmeTarihi
- Wiki**: wikiNo (PK), repoNo (FK), kullaniciNo (FK), baslik, metin, mesaj, olusturulmaTarihi
- Commit**: commitNo (PK), branchNo (FK), kullaniciNo (FK), baslik, metin, olusturulmaTarihi
- Commit Yorum**: commitYorumNo (PK), commitNo (FK), kullaniciNo (FK), mesaj, olusturulmaTarihi
- Issue**: issueNo (PK), repoNo (FK), sahipNo (FK), baslik, metin, olusturulmaTarihi, kapanisTarihi
- Issue Yorum**: issueYorumNo (PK), issueNo (FK), kullaniciNo (FK), mesaj, olusturulmaTarihi
- Milestone**: milestoneNo (PK), issueNo (FK), baslik, aciklama, tarih, guncellenmeTarihi, olusturulmaTarihi, kapanisTarihi
- Label**: labelNo (PK), issueNo (FK), isim, aciklama, renkKodu
- Pull Request**: pullReqNo (PK), repoNo (FK), kullaniciNo (FK), baslik, mesaj, olusturulmaTarihi, kapanisTarihi
- Pull Request Yorum**: pullReqYorumNo (PK), pullReqNo (FK), kullaniciNo (FK), mesaj, olusturulmaTarihi
- Proje**: projeNo (PK), repoNo (FK), isim, aciklama, olusturulmaTarihi, kapanisTarihi
- Column**: columnNo (PK), projeNo (FK), isim
- Discussion**: discussionNo (PK), repoNo (FK), kullaniciNo (FK), baslik, metin, kategori, olusturulmaTarihi
- Discussion Yorum**: discussionYorumNo (PK), discussionNo (FK), kullaniciNo (FK), mesaj, olusturulmaTarihi

Relationships are defined as follows:

- Kullanici** *olusturur* **Organizasyon Uye** (1:M)
- Organizasyon Uye** *olusturur* **Organizasyon** (1:M)
- Organizasyon** *sahiptir* **Takim** (1:M)
- Takim** *bulunur* **Repository** (1:M)
- Repository** *bulunur* **Issue** (1:M)
- Repository** *bulunur* **Branch** (1:M)
- Repository** *bulunur* **Wiki** (1:M)
- Repository** *bulunur* **Commit** (1:M)
- Repository** *bulunur* **Discussion** (1:M)
- Repository** *bulunur* **Proje** (1:M)
- Repository** *bulunur* **Column** (1:M)
- Repository** *bulunur* **Label** (1:M)
- Repository** *bulunur* **Milestone** (1:M)
- Repository** *bulunur* **Issue Yorum** (1:M)
- Repository** *bulunur* **Commit Yorum** (1:M)
- Repository** *bulunur* **Discussion Yorum** (1:M)
- Repository** *bulunur* **Pull Request Yorum** (1:M)
- Repository** *bulunur* **Starrer** (1:M)
- Repository** *bulunur* **Watcher** (1:M)
- Repository** *bulunur* **Forker** (1:M)
- Repository** *bulunur* **Normal Uye** (1:M)
- Repository** *bulunur* **Takim Lideri** (1:M)
- Repository** *bulunur* **Takimsiz Uye** (1:M)
- Repository** *bulunur* **Organizasyon Uye** (1:M)
- Repository** *bulunur* **Organizasyon** (1:M)
- Repository** *bulunur* **Takim** (1:M)
- Repository** *bulunur* **Issue** (1:M)
- Repository** *bulunur* **Branch** (1:M)
- Repository** *bulunur* **Wiki** (1:M)
- Repository** *bulunur* **Commit** (1:M)
- Repository** *bulunur* **Discussion** (1:M)
- Repository** *bulunur* **Proje** (1:M)
- Repository** *bulunur* **Column** (1:M)
- Repository** *bulunur* **Label** (1:M)
- Repository** *bulunur* **Milestone** (1:M)
- Repository** *bulunur* **Issue Yorum** (1:M)
- Repository** *bulunur* **Commit Yorum** (1:M)
- Repository** *bulunur* **Discussion Yorum** (1:M)
- Repository** *bulunur* **Pull Request Yorum** (1:M)
- Repository** *bulunur* **Starrer** (1:M)
- Repository** *bulunur* **Watcher** (1:M)
- Repository** *bulunur* **Forker** (1:M)
- Repository** *bulunur* **Normal Uye** (1:M)
- Repository** *bulunur* **Takim Lideri** (1:M)
- Repository** *bulunur* **Takimsiz Uye** (1:M)
- Repository** *bulunur* **Organizasyon Uye** (1:M)
- Repository** *bulunur* **Organizasyon** (1:M)
- Repository** *bulunur* **Takim** (1:M)
- Repository** *bulunur* **Issue** (1:M)
- Repository** *bulunur* **Branch** (1:M)
- Repository** *bulunur* **Wiki** (1:M)
- Repository** *bulunur* **Commit** (1:M)
- Repository** *bulunur* **Discussion** (1:M)
- Repository** *bulunur* **Proje** (1:M)
- Repository** *bulunur* **Column** (1:M)
- Repository** *bulunur* **Label** (1:M)
- Repository** *bulunur* **Milestone** (1:M)
- Repository** *bulunur* **Issue Yorum** (1:M)
- Repository** *bulunur* **Commit Yorum** (1:M)
- Repository** *bulunur* **Discussion Yorum** (1:M)
- Repository** *bulunur* **Pull Request Yorum** (1:M)
- Repository** *bulunur* **Starrer** (1:M)
- Repository** *bulunur* **Watcher** (1:M)
- Repository** *bulunur* **Forker** (1:M)
- Repository** *bulunur* **Normal Uye** (1:M)
- Repository** *bulunur* **Takim Lideri** (1:M)
- Repository** *bulunur* **Takimsiz Uye** (1:M)
- Repository** *bulunur* **Organizasyon Uye** (1:M)
- Repository** *bulunur* **Organizasyon** (1:M)
- Repository** *bulunur* **Takim** (1:M)
- Repository** *bulunur* **Issue** (1:M)
- Repository** *bulunur* **Branch** (1:M)
- Repository** *bulunur* **Wiki** (1:M)
- Repository** *bulunur* **Commit** (1:M)
- Repository** *bulunur* **Discussion** (1:M)
- Repository** *bulunur* **Proje** (1:M)
- Repository** *bulunur* **Column** (1:M)
- Repository** *bulunur* **Label** (1:M)
- Repository** *bulunur* **Milestone** (1:M)
- Repository** *bulunur* **Issue Yorum** (1:M)
- Repository** *bulunur* **Commit Yorum** (1:M)
- Repository** *bulunur* **Discussion Yorum** (1:M)
- Repository** *bulunur* **Pull Request Yorum** (1:M)
- Repository** *bulunur* **Starrer** (1:M)
- Repository** *bulunur* **Watcher** (1:M)
- Repository** *bulunur* **Forker** (1:M)
- Repository** *bulunur* **Normal Uye** (1:M)
- Repository** *bulunur* **Takim Lideri** (1:M)
- Repository** *bulunur* **Takimsiz Uye** (1:M)
- Repository** *bulunur* **Organizasyon Uye** (1:M)
- Repository** *bulunur* **Organizasyon** (1:M)</

SQL İfadeleri

```
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;
```

```
CREATE SCHEMA github;
```

```
ALTER SCHEMA github OWNER TO postgres;
```

```
CREATE SEQUENCE github.sayac12
  START WITH 1
  INCREMENT BY 1
  NO MINVALUE
  NO MAXVALUE
  CACHE 1;
```

```
ALTER TABLE github.sayac12 OWNER TO postgres;
```

```
SET default_tablespace = '';
```

```
SET default_table_access_method = heap;
```

```
CREATE TABLE github."Branch" (
  "branchNo" integer DEFAULT nextval('github.sayac12::regclass) NOT NULL,
  "repoNo" integer NOT NULL,
  "baslik" character varying(25) NOT NULL,
  "guncellenmeTarihi" date NOT NULL
);
```

```
ALTER TABLE github."Branch" OWNER TO postgres;
```

```
CREATE SEQUENCE github.sayac9  
  START WITH 1  
  INCREMENT BY 1  
  NO MINVALUE  
  NO MAXVALUE  
  CACHE 1;
```

```
ALTER TABLE github.sayac9 OWNER TO postgres;
```

```
CREATE TABLE github."Column" (  
  "columnNo" integer DEFAULT nextval('github.sayac9'::regclass) NOT NULL,  
  "projeNo" integer NOT NULL,  
  "isim" character varying(20) NOT NULL  
);
```

```
ALTER TABLE github."Column" OWNER TO postgres;
```

```
CREATE SEQUENCE github.sayac13  
  START WITH 1  
  INCREMENT BY 1  
  NO MINVALUE  
  NO MAXVALUE  
  CACHE 1;
```

```
ALTER TABLE github.sayac13 OWNER TO postgres;
```

```
CREATE TABLE github."Commit" (  
  "commitNo" integer DEFAULT nextval('github.sayac13'::regclass) NOT NULL,  
  "branchNo" integer NOT NULL,  
  "kullaniciNo" integer NOT NULL,  
  "baslik" character varying(45) NOT NULL,
```

```
"metin" text NOT NULL,  
"olusturulmaTarihi" date NOT NULL  
);
```

```
ALTER TABLE github."Commit" OWNER TO postgres;
```

```
CREATE SEQUENCE github.sayac14  
  START WITH 1  
  INCREMENT BY 1  
  NO MINVALUE  
  NO MAXVALUE  
  CACHE 1;
```

```
ALTER TABLE github.sayac14 OWNER TO postgres;
```

```
CREATE TABLE github."CommitYorum" (  
  "commitYorumNo" integer DEFAULT nextval('github.sayac14'::regclass) NOT NULL,  
  "commitNo" integer NOT NULL,  
  "kullaniciNo" integer NOT NULL,  
  "mesaj" text NOT NULL,  
  "olusturulmaTarihi" date NOT NULL  
);
```

```
ALTER TABLE github."CommitYorum" OWNER TO postgres;
```

```
CREATE SEQUENCE github.sayac10  
  START WITH 1  
  INCREMENT BY 1  
  NO MINVALUE  
  NO MAXVALUE  
  CACHE 1;
```

```
ALTER TABLE github.sayac10 OWNER TO postgres;
```



```
CREATE TABLE github."Discussion" (  
    "discussionNo" integer DEFAULT nextval('github.sayac10'::regclass) NOT NULL,  
    "repoNo" integer NOT NULL,  
    "kullaniciNo" integer NOT NULL,  
    "baslik" character varying(70) NOT NULL,  
    "metin" text NOT NULL,  
    "kategori" character varying(20) NOT NULL,  
    "olusturulmaTarihi" date NOT NULL  
);
```

```
ALTER TABLE github."Discussion" OWNER TO postgres;
```

```
CREATE SEQUENCE github.sayac11  
    START WITH 1  
    INCREMENT BY 1  
    NO MINVALUE  
    NO MAXVALUE  
    CACHE 1;
```

```
ALTER TABLE github.sayac11 OWNER TO postgres;
```

```
CREATE TABLE github."DiscussionYorum" (  
    "discussionYorumNo" integer DEFAULT nextval('github.sayac11'::regclass) NOT NULL,  
    "discussionNo" integer NOT NULL,  
    "kullaniciNo" integer NOT NULL,  
    "mesaj" text NOT NULL,  
    "olusturulmaTarihi" date NOT NULL  
);
```

```
ALTER TABLE github."DiscussionYorum" OWNER TO postgres;
```

```
CREATE TABLE github."Forker" (  
    "repoNo" integer NOT NULL,  
    "kullaniciNo" integer NOT NULL,  
    "olusturulmaTarihi" date NOT NULL
```

```
);
```

```
ALTER TABLE github."Forker" OWNER TO postgres;
```

```
CREATE SEQUENCE github.sayac15
```

```
START WITH 1
```

```
INCREMENT BY 1
```

```
NO MINVALUE
```

```
NO MAXVALUE
```

```
CACHE 1;
```

```
ALTER TABLE github.sayac15 OWNER TO postgres;
```

```
CREATE TABLE github."Issue" (
```

```
"issueNo" integer DEFAULT nextval('github.sayac15'::regclass) NOT NULL,
```

```
"repoNo" integer NOT NULL,
```

```
"sahipNo" integer NOT NULL,
```

```
"baslik" character varying(70) NOT NULL,
```

```
"metin" text,
```

```
"olusturulmaTarihi" date NOT NULL,
```

```
"kapanisTarihi" date
```

```
);
```

```
ALTER TABLE github."Issue" OWNER TO postgres;
```

```
CREATE SEQUENCE github.sayac16
```

```
START WITH 1
```

```
INCREMENT BY 1
```

```
NO MINVALUE
```

```
NO MAXVALUE
```

```
CACHE 1;
```

```
ALTER TABLE github.sayac16 OWNER TO postgres;
```

```
CREATE TABLE github."IssueYorum" (
```

```
"issueYorumNo" integer DEFAULT nextval('github.sayac16'::regclass) NOT NULL,  
"issueNo" integer NOT NULL,  
"kullaniciNo" integer NOT NULL,  
"mesaj" text NOT NULL,  
"olusturulmaTarihi" date NOT NULL  
);
```

```
ALTER TABLE github."IssueYorum" OWNER TO postgres;
```

```
CREATE SEQUENCE github.sayac  
START WITH 1  
INCREMENT BY 1  
NO MINVALUE  
NO MAXVALUE  
CACHE 1;
```

```
ALTER TABLE github.sayac OWNER TO postgres;
```

```
CREATE TABLE github."Kullanici" (  
"kullaniciNo" integer DEFAULT nextval('github.sayac'::regclass) NOT NULL,  
"kullaniciAdi" character varying(15) NOT NULL,  
"email" character varying(45) NOT NULL,  
"sifre" character varying(40) NOT NULL,  
"isim" character varying(20),  
"lokasyon" character varying(20),  
"sirketAdi" character varying(40),  
"biyografi" text,  
"url" character varying(45),  
"durumMesaji" character varying(45),  
"olusturulmaTarihi" date NOT NULL  
);
```

```
ALTER TABLE github."Kullanici" OWNER TO postgres;
```

```
CREATE SEQUENCE github.sayac18
```

```
START WITH 1
INCREMENT BY 1
NO MINVALUE
NO MAXVALUE
CACHE 1;
```

```
ALTER TABLE github.sayac18 OWNER TO postgres;
```

```
CREATE TABLE github."Label" (
    "labelNo" integer DEFAULT nextval('github.sayac18'::regclass) NOT NULL,
    "issueNo" integer NOT NULL,
    "isim" character varying(20) NOT NULL,
    "aciklama" text,
    "renkKodu" character varying(15) NOT NULL
);
```

```
ALTER TABLE github."Label" OWNER TO postgres;
```

```
CREATE SEQUENCE github.sayac17
START WITH 1
INCREMENT BY 1
NO MINVALUE
NO MAXVALUE
CACHE 1;
```

```
ALTER TABLE github.sayac17 OWNER TO postgres;
```

```
CREATE TABLE github."Milestone" (
    "milestoneNo" integer DEFAULT nextval('github.sayac17'::regclass) NOT NULL,
    "issueNo" integer NOT NULL,
    "baslik" character varying(40) NOT NULL,
    "aciklama" text,
    "kapanisTarihi" date,
    "guncellenmeTarihi" date,
    "olusturulmaTarihi" date NOT NULL,
```

```
"tarih" date  
);
```

```
ALTER TABLE github."Milestone" OWNER TO postgres;
```

```
CREATE TABLE github."NormalUye" (  
    "kullaniciNo" integer NOT NULL  
);
```

```
ALTER TABLE github."NormalUye" OWNER TO postgres;
```

```
CREATE SEQUENCE github.sayac2  
    START WITH 1  
    INCREMENT BY 1  
    NO MINVALUE  
    NO MAXVALUE  
    CACHE 1;
```

```
ALTER TABLE github.sayac2 OWNER TO postgres;
```

```
CREATE TABLE github."Organizasyon" (  
    "orgNo" integer DEFAULT nextval('github.sayac2'::regclass) NOT NULL,  
    "isim" character varying(20),  
    "iletisimMail" character varying(45),  
    "aciklama" text,  
    "url" character varying(45),  
    "lokasyon" character varying(20),  
    "tip" character varying(20),  
    "olusturulmaTarihi" date NOT NULL  
);
```

```
ALTER TABLE github."Organizasyon" OWNER TO postgres;
```

```
CREATE TABLE github."OrganizasyonUye" (  
    "orgNo" integer NOT NULL,
```

```
"kullaniciNo" integer NOT NULL,  
"organizasyonPanel" character varying(255)  
);
```

```
ALTER TABLE github."OrganizasyonUye" OWNER TO postgres;
```

```
CREATE SEQUENCE github.sayac8  
  START WITH 1  
  INCREMENT BY 1  
  NO MINVALUE  
  NO MAXVALUE  
  CACHE 1;
```

```
ALTER TABLE github.sayac8 OWNER TO postgres;
```

```
CREATE TABLE github."Proje" (  
  "projeNo" integer DEFAULT nextval('github.sayac8'::regclass) NOT NULL,  
  "repoNo" integer NOT NULL,  
  "isim" character varying(40) NOT NULL,  
  "aciklama" text,  
  "olusturulmaTarihi" date NOT NULL,  
  "kapanisTarihi" date  
);
```

```
ALTER TABLE github."Proje" OWNER TO postgres;
```

```
CREATE SEQUENCE github.sayac6  
  START WITH 1  
  INCREMENT BY 1  
  NO MINVALUE  
  NO MAXVALUE  
  CACHE 1;
```

```
ALTER TABLE github.sayac6 OWNER TO postgres;
```

```
CREATE TABLE github."PullRequest" (  
    "pullReqNo" integer DEFAULT nextval('github.sayac6'::regclass) NOT NULL,  
    "repoNo" integer NOT NULL,  
    "kullaniciNo" integer NOT NULL,  
    "baslik" character varying(65) NOT NULL,  
    "mesaj" text NOT NULL,  
    "olusturulmaTarihi" date NOT NULL,  
    "kapanisTarihi" date  
);
```

```
ALTER TABLE github."PullRequest" OWNER TO postgres;
```

```
CREATE SEQUENCE github.sayac7  
    START WITH 1  
    INCREMENT BY 1  
    NO MINVALUE  
    NO MAXVALUE  
    CACHE 1;
```

```
ALTER TABLE github.sayac7 OWNER TO postgres;
```

```
CREATE TABLE github."PullRequestYorum" (  
    "pullReqYorumNo" integer DEFAULT nextval('github.sayac7'::regclass) NOT NULL,  
    "pullReqNo" integer NOT NULL,  
    "kullaniciNo" integer NOT NULL,  
    "mesaj" text NOT NULL,  
    "olusturulmaTarihi" date NOT NULL  
);
```

```
ALTER TABLE github."PullRequestYorum" OWNER TO postgres;
```

```
CREATE SEQUENCE github.sayac4  
    START WITH 1  
    INCREMENT BY 1  
    NO MINVALUE
```

NO MAXVALUE

CACHE 1;

ALTER TABLE github.sayac4 OWNER TO postgres;

```
CREATE TABLE github."Repository" (  
    "repoNo" integer DEFAULT nextval('github.sayac4::regclass) NOT NULL,  
    "sahipNo" integer NOT NULL,  
    "orgNo" integer,  
    "isim" character varying(50) NOT NULL,  
    "aciklama" text,  
    "url" character varying(45),  
    "dil" character varying(20),  
    "olusturulmaTarihi" date NOT NULL,  
    "sonPushTarihi" date  
);
```

ALTER TABLE github."Repository" OWNER TO postgres;

```
CREATE TABLE github."Starrer" (  
    "repoNo" integer NOT NULL,  
    "kullaniciNo" integer NOT NULL,  
    "olusturulmaTarihi" date NOT NULL  
);
```

ALTER TABLE github."Starrer" OWNER TO postgres;

```
CREATE SEQUENCE github.sayac3  
    START WITH 1  
    INCREMENT BY 1  
    NO MINVALUE  
    NO MAXVALUE  
    CACHE 1;
```

ALTER TABLE github.sayac3 OWNER TO postgres;


```
CREATE TABLE github."Takim" (  
    "takimNo" integer DEFAULT nextval('github.sayac3'::regclass) NOT NULL,  
    "orgNo" integer NOT NULL,  
    "isim" character varying(40) NOT NULL,  
    "aciklama" text,  
    "gizlilikDurumu" character varying(20) NOT NULL,  
    "olusturulmaTarihi" date NOT NULL  
);
```

```
ALTER TABLE github."Takim" OWNER TO postgres;
```

```
CREATE TABLE github."TakimLideri" (  
    "kullaniciNo" integer NOT NULL,  
    "takimKontrolPaneli" character varying(255)  
);
```

```
ALTER TABLE github."TakimLideri" OWNER TO postgres;
```

```
CREATE TABLE github."TakimliUye" (  
    "kullaniciNo" integer NOT NULL,  
    "takimNo" integer NOT NULL,  
    "takimPanel" character varying(255)  
);
```

```
ALTER TABLE github."TakimliUye" OWNER TO postgres;
```

```
CREATE TABLE github."TakimsizUye" (  
    "kullaniciNo" integer NOT NULL  
);
```

```
ALTER TABLE github."TakimsizUye" OWNER TO postgres;
```

```
CREATE TABLE github."Takip" (  
    "takipciNo" integer NOT NULL,
```

```
"takipEdenNo" integer NOT NULL,  
"takipTarihi" date NOT NULL  
);
```

```
ALTER TABLE github."Takip" OWNER TO postgres;
```

```
CREATE TABLE github."Watcher" (  
"repoNo" integer NOT NULL,  
"kullaniciNo" integer NOT NULL,  
"olusturulmaTarihi" date NOT NULL  
);
```

```
ALTER TABLE github."Watcher" OWNER TO postgres;
```

```
CREATE SEQUENCE github.sayac5  
START WITH 1  
INCREMENT BY 1  
NO MINVALUE  
NO MAXVALUE  
CACHE 1;
```

```
ALTER TABLE github.sayac5 OWNER TO postgres;
```

```
CREATE TABLE github."Wiki" (  
"wikiNo" integer DEFAULT nextval('github.sayac5::regclass) NOT NULL,  
"repoNo" integer NOT NULL,  
"kullaniciNo" integer NOT NULL,  
"baslik" character varying(50) NOT NULL,  
"metin" text NOT NULL,  
"mesaj" text,  
"olusturulmaTarihi" date NOT NULL  
);
```

```
ALTER TABLE github."Wiki" OWNER TO postgres;
```

INSERT INTO github."Branch" VALUES

(1, 1, 'a', '2020-12-01'),
(2, 2, 'b', '2020-12-02');

INSERT INTO github."Column" VALUES

(1, 1, 'a'),
(2, 1, 'b');

INSERT INTO github."Commit" VALUES

(1, 1, 1, 'a', 'b', '2020-12-01'),
(2, 2, 1, 'b', 'c', '2020-12-02');

INSERT INTO github."CommitYorum" VALUES

(1, 1, 1, 'a', '2020-12-01'),
(2, 2, 1, 'b', '2020-12-02');

INSERT INTO github."Discussion" VALUES

(1, 2, 1, 'a', 'b', 'c', '2020-12-01'),
(2, 2, 1, 'b', 'c', 'd', '2020-12-02');

INSERT INTO github."DiscussionYorum" VALUES

(1, 1, 1, 'a', '2020-12-01'),
(2, 2, 1, 'b', '2020-12-02');

INSERT INTO github."Forker" VALUES

(1, 1, '2020-12-01'),
(2, 1, '2020-12-02');

INSERT INTO github."Issue" VALUES

(1, 1, 1, 'a', 'b', '2020-12-01', NULL),
(2, 2, 1, 'b', 'c', '2020-12-02', '2020-12-03'),
(3, 3, 2, 'c', 'd', '2020-12-02', NULL);

INSERT INTO github."IssueYorum" VALUES

(1, 1, 1, 'a', '2020-12-01'),

(2, 2, 1, 'b', '2020-12-02');

INSERT INTO github."Kullanici" VALUES

(1, 'a', 'a@hotmail.com', '123456', 'a', 'b', 'c', 'd', 'e', 'f', '2020-12-01'),
(2, 'b', 'b@hotmail.com', '123456', 'a', 'b', 'c', 'd', 'e', 'f', '2020-12-01'),
(3, 'c', 'c@hotmail.com', '123456', 'a', 'b', 'c', 'd', 'e', 'f', '2020-12-02');

INSERT INTO github."Label" VALUES

(1, 1, 'a', 'b', '2323'),
(2, 2, 'f', NULL, '1313');

INSERT INTO github."Milestone" VALUES

(1, 1, 'a', 'b', NULL, NULL, '2020-12-01', NULL),
(2, 2, 'b', 'd', '2020-12-03', '2020-12-02', '2020-12-02', '2020-12-02');

INSERT INTO github."NormalUye" VALUES

(3),
(2);

INSERT INTO github."Organizasyon" VALUES

(1, 'a', 'a@hotmail.com', 'a', 'b', 'c', 'd', '2020-12-01'),
(2, 'b', 'b@hotmail.com', 'a', 'b', 'c', 'd', '2020-12-02');

INSERT INTO github."OrganizasyonUye" VALUES

(1, 2, NULL),
(2, 2, NULL),
(2, 3, NULL);

INSERT INTO github."Proje" VALUES

(1, 1, 'a', 'b', '2020-12-01', '2020-12-02'),
(2, 2, 'b', 'c', '2020-12-02', NULL);

INSERT INTO github."PullRequest" VALUES

(1, 1, 1, 'a', 'b', '2020-12-01', NULL),
(2, 3, 2, 'b', 'c', '2020-12-02', '2020-12-02');

INSERT INTO github."PullRequestYorum" VALUES

(1, 1, 1, 'a', '2020-12-01'),
(2, 2, 2, 'b', '2020-12-02');

INSERT INTO github."Repository" VALUES

(1, 1, NULL, 'a', 'b', 'c', 'd', '2020-12-01', NULL),
(2, 1, NULL, 'b', 'c', 'd', 'e', '2020-12-02', NULL),
(3, 2, 2, 'c', 'e', 'f', 'g', '2020-12-01', NULL);

INSERT INTO github."Starrer" VALUES

(1, 1, '2020-12-01'),
(2, 1, '2020-12-01');

INSERT INTO github."Takim" VALUES

(1, 1, 'a', 'b', 'c', '2020-12-01'),
(2, 2, 'b', 'c', 'd', '2020-12-02'),
(3, 1, 'g', 'f', 'r', '2020-12-01');

INSERT INTO github."TakimLideri" VALUES

(3, NULL);

INSERT INTO github."TakimliUye" VALUES

(2, 2, NULL),
(3, 1, NULL);

INSERT INTO github."TakimsizUye" VALUES

(1);

INSERT INTO github."Takip" VALUES

(1, 2, '2020-12-01'),
(1, 3, '2020-12-02'),
(3, 2, '2020-12-02');

INSERT INTO github."Watcher" VALUES

```
(1, 1, '2020-12-01'),  
(2, 1, '2020-12-02');
```

```
INSERT INTO github."Wiki" VALUES
```

```
(1, 1, 1, 'a', 'b', 'c', '2020-12-01'),  
(2, 3, 2, 'b', 'c', 'd', '2020-12-02');
```

```
SELECT pg_catalog.setval('github.sayac', 3, true);
```

```
SELECT pg_catalog.setval('github.sayac10', 2, true);
```

```
SELECT pg_catalog.setval('github.sayac11', 2, true);
```

```
SELECT pg_catalog.setval('github.sayac12', 2, true);
```

```
SELECT pg_catalog.setval('github.sayac13', 2, true);
```

```
SELECT pg_catalog.setval('github.sayac14', 2, true);
```

```
SELECT pg_catalog.setval('github.sayac15', 3, true);
```

```
SELECT pg_catalog.setval('github.sayac16', 2, true);
```

```
SELECT pg_catalog.setval('github.sayac17', 2, true);
```

```
SELECT pg_catalog.setval('github.sayac18', 2, true);
```

```
SELECT pg_catalog.setval('github.sayac2', 1, true);
```

```
SELECT pg_catalog.setval('github.sayac3', 1, true);
```

```
SELECT pg_catalog.setval('github.sayac4', 3, true);
```

```
SELECT pg_catalog.setval('github.sayac5', 2, true);
```

```
SELECT pg_catalog.setval('github.sayac6', 2, true);
```

```
SELECT pg_catalog.setval('github.sayac7', 2, true);
```

```
SELECT pg_catalog.setval('github.sayac8', 2, true);
```

```
SELECT pg_catalog.setval('github.sayac9', 3, true);
```

```
ALTER TABLE ONLY github."Branch"
```

```
ADD CONSTRAINT "branchPK" PRIMARY KEY ("branchNo");
```

```
ALTER TABLE ONLY github."Branch"
```

```
ADD CONSTRAINT "branchUnique1" UNIQUE (baslik);
```

```
ALTER TABLE ONLY github."Column"
```

```
ADD CONSTRAINT "columnPK" PRIMARY KEY ("columnNo");
```

```
ALTER TABLE ONLY github."Column"
```

```
ADD CONSTRAINT "columnUnique1" UNIQUE (isim);
```

```
ALTER TABLE ONLY github."Commit"
```

```
ADD CONSTRAINT "commitPK" PRIMARY KEY ("commitNo");
```

```
ALTER TABLE ONLY github."CommitYorum"
```

```
ADD CONSTRAINT "commitYorumPK" PRIMARY KEY ("commitYorumNo");
```

```
ALTER TABLE ONLY github."Discussion"
```

```
ADD CONSTRAINT "discussionPK" PRIMARY KEY ("discussionNo");
```

```
ALTER TABLE ONLY github."DiscussionYorum"
```

```
ADD CONSTRAINT "discussionYorumPK" PRIMARY KEY ("discussionYorumNo");
```

```
ALTER TABLE ONLY github."Forker"
```

```
ADD CONSTRAINT "forkerPK" PRIMARY KEY ("repoNo", "kullaniciNo");
```

ALTER TABLE ONLY github."Issue"

ADD CONSTRAINT "issuePK" PRIMARY KEY ("issueNo");

ALTER TABLE ONLY github."IssueYorum"

ADD CONSTRAINT "issueYorumPK" PRIMARY KEY ("issueYorumNo");

ALTER TABLE ONLY github."Kullanici"

ADD CONSTRAINT "kullaniciPK" PRIMARY KEY ("kullaniciNo");

ALTER TABLE ONLY github."Kullanici"

ADD CONSTRAINT "kullaniciUnique1" UNIQUE (email);

ALTER TABLE ONLY github."Kullanici"

ADD CONSTRAINT "kullaniciUnique2" UNIQUE ("kullaniciAdi");

ALTER TABLE ONLY github."Label"

ADD CONSTRAINT "labelPK" PRIMARY KEY ("labelNo");

ALTER TABLE ONLY github."Label"

ADD CONSTRAINT "labelUnique1" UNIQUE (isim);

ALTER TABLE ONLY github."Milestone"

ADD CONSTRAINT "milestonePK" PRIMARY KEY ("milestoneNo");

ALTER TABLE ONLY github."Milestone"

ADD CONSTRAINT "milestoneUnique1" UNIQUE (baslik);

ALTER TABLE ONLY github."NormalUye"

ADD CONSTRAINT "normalUyePK" PRIMARY KEY ("kullaniciNo");

ALTER TABLE ONLY github."Organizasyon"

ADD CONSTRAINT "orgPK" PRIMARY KEY ("orgNo");

ALTER TABLE ONLY github."OrganizasyonUye"

ADD CONSTRAINT "orgUyePK" PRIMARY KEY ("kullaniciNo", "orgNo");


```
ALTER TABLE ONLY github."Proje"
```

```
ADD CONSTRAINT "projePK" PRIMARY KEY ("projeNo");
```

```
ALTER TABLE ONLY github."PullRequest"
```

```
ADD CONSTRAINT "pullReqPK" PRIMARY KEY ("pullReqNo");
```

```
ALTER TABLE ONLY github."PullRequestYorum"
```

```
ADD CONSTRAINT "pullReqYorumPK" PRIMARY KEY ("pullReqYorumNo");
```

```
ALTER TABLE ONLY github."Repository"
```

```
ADD CONSTRAINT "repoPK" PRIMARY KEY ("repoNo");
```

```
ALTER TABLE ONLY github."Repository"
```

```
ADD CONSTRAINT "repoUnique1" UNIQUE (isim);
```

```
ALTER TABLE ONLY github."Starrer"
```

```
ADD CONSTRAINT "starrerPK" PRIMARY KEY ("repoNo", "kullaniciNo");
```

```
ALTER TABLE ONLY github."TakimLideri"
```

```
ADD CONSTRAINT "takimLideriPK" PRIMARY KEY ("kullaniciNo");
```

```
ALTER TABLE ONLY github."Takim"
```

```
ADD CONSTRAINT "takimPK" PRIMARY KEY ("takimNo");
```

```
ALTER TABLE ONLY github."Takim"
```

```
ADD CONSTRAINT "takimUnique1" UNIQUE (isim);
```

```
ALTER TABLE ONLY github."TakimliUye"
```

```
ADD CONSTRAINT "takimliUyePK" PRIMARY KEY ("kullaniciNo");
```

```
ALTER TABLE ONLY github."TakimsizUye"
```

```
ADD CONSTRAINT "takimsizUyePK" PRIMARY KEY ("kullaniciNo");
```

```
ALTER TABLE ONLY github."Takip"
```

ADD CONSTRAINT "takipPK" PRIMARY KEY ("takipciNo", "takipEdenNo");

ALTER TABLE ONLY github."Watcher"

ADD CONSTRAINT "watcherPK" PRIMARY KEY ("repoNo", "kullaniciNo");

ALTER TABLE ONLY github."Wiki"

ADD CONSTRAINT "wikiPK" PRIMARY KEY ("wikiNo");

ALTER TABLE ONLY github."Wiki"

ADD CONSTRAINT "wikiUnique1" UNIQUE (baslik);

ALTER TABLE ONLY github."NormalUye"

ADD CONSTRAINT "NormalUye" FOREIGN KEY ("kullaniciNo") REFERENCES github."TakimliUye"("kullaniciNo") ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY github."TakimLideri"

ADD CONSTRAINT "TakimLideri" FOREIGN KEY ("kullaniciNo") REFERENCES github."TakimliUye"("kullaniciNo") ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY github."TakimliUye"

ADD CONSTRAINT "TakimliUye" FOREIGN KEY ("kullaniciNo") REFERENCES github."Kullanici"("kullaniciNo") ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY github."TakimsizUye"

ADD CONSTRAINT "TakimsizUye" FOREIGN KEY ("kullaniciNo") REFERENCES github."Kullanici"("kullaniciNo") ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY github."Branch"

ADD CONSTRAINT "branchFK" FOREIGN KEY ("repoNo") REFERENCES github."Repository"("repoNo") ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY github."Column"

ADD CONSTRAINT "columnFK" FOREIGN KEY ("projeNo") REFERENCES github."Proje"("projeNo") ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY github."Commit"

ADD CONSTRAINT "commitFK1" FOREIGN KEY ("kullaniciNo") REFERENCES
github."Kullanici"("kullaniciNo") ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY github."Commit"

ADD CONSTRAINT "commitFK2" FOREIGN KEY ("branchNo") REFERENCES
github."Branch"("branchNo") ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY github."CommitYorum"

ADD CONSTRAINT "commitYorumFK1" FOREIGN KEY ("kullaniciNo") REFERENCES
github."Kullanici"("kullaniciNo") ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY github."CommitYorum"

ADD CONSTRAINT "commitYorumFK2" FOREIGN KEY ("commitNo") REFERENCES
github."Commit"("commitNo") ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY github."Discussion"

ADD CONSTRAINT "discussionFK1" FOREIGN KEY ("kullaniciNo") REFERENCES
github."Kullanici"("kullaniciNo") ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY github."Discussion"

ADD CONSTRAINT "discussionFK2" FOREIGN KEY ("repoNo") REFERENCES
github."Repository"("repoNo") ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY github."DiscussionYorum"

ADD CONSTRAINT "discussionYorumFK1" FOREIGN KEY ("kullaniciNo") REFERENCES
github."Kullanici"("kullaniciNo") ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY github."DiscussionYorum"

ADD CONSTRAINT "discussionYorumFK2" FOREIGN KEY ("discussionNo") REFERENCES
github."Discussion"("discussionNo") ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY github."Forker"

ADD CONSTRAINT "forkerFK1" FOREIGN KEY ("kullaniciNo") REFERENCES
github."Kullanici"("kullaniciNo") ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY github."Forker"

ADD CONSTRAINT "forkerFK2" FOREIGN KEY ("repoNo") REFERENCES
github."Repository"("repoNo");

ALTER TABLE ONLY github."Issue"

ADD CONSTRAINT "issueFK1" FOREIGN KEY ("sahipNo") REFERENCES
github."Kullanici"("kullaniciNo") ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY github."Issue"

ADD CONSTRAINT "issueFK2" FOREIGN KEY ("repoNo") REFERENCES
github."Repository"("repoNo") ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY github."IssueYorum"

ADD CONSTRAINT "issueYorumFK1" FOREIGN KEY ("kullaniciNo") REFERENCES
github."Kullanici"("kullaniciNo") ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY github."IssueYorum"

ADD CONSTRAINT "issueYorumFK2" FOREIGN KEY ("issueNo") REFERENCES
github."Issue"("issueNo") ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY github."Label"

ADD CONSTRAINT "labelFK" FOREIGN KEY ("issueNo") REFERENCES github."Issue"("issueNo")
ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY github."Milestone"

ADD CONSTRAINT "milestoneFK" FOREIGN KEY ("issueNo") REFERENCES
github."Issue"("issueNo") ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY github."OrganizasyonUye"

ADD CONSTRAINT "orgUyeFK1" FOREIGN KEY ("kullaniciNo") REFERENCES
github."Kullanici"("kullaniciNo") ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY github."OrganizasyonUye"

ADD CONSTRAINT "orgUyeFK2" FOREIGN KEY ("orgNo") REFERENCES
github."Organizasyon"("orgNo") ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY github."Proje"

ADD CONSTRAINT "projeFK" FOREIGN KEY ("repoNo") REFERENCES
github."Repository"("repoNo") ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY github."PullRequest"

ADD CONSTRAINT "pullReqFK1" FOREIGN KEY ("kullaniciNo") REFERENCES
github."Kullanici"("kullaniciNo") ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY github."PullRequest"

ADD CONSTRAINT "pullReqFK2" FOREIGN KEY ("repoNo") REFERENCES
github."Repository"("repoNo") ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY github."PullRequestYorum"

ADD CONSTRAINT "pullReqYorumFK1" FOREIGN KEY ("kullaniciNo") REFERENCES
github."Kullanici"("kullaniciNo") ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY github."PullRequestYorum"

ADD CONSTRAINT "pullReqYorumFK2" FOREIGN KEY ("pullReqNo") REFERENCES
github."PullRequest"("pullReqNo") ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY github."Repository"

ADD CONSTRAINT "repoFK1" FOREIGN KEY ("sahipNo") REFERENCES
github."Kullanici"("kullaniciNo") ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY github."Repository"

ADD CONSTRAINT "repoFK2" FOREIGN KEY ("orgNo") REFERENCES
github."Organizasyon"("orgNo") ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY github."Starrer"

ADD CONSTRAINT "starrerFK1" FOREIGN KEY ("kullaniciNo") REFERENCES
github."Kullanici"("kullaniciNo") ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY github."Starrer"

ADD CONSTRAINT "starrerFK2" FOREIGN KEY ("repoNo") REFERENCES
github."Repository"("repoNo");

ALTER TABLE ONLY github."Takim"

ADD CONSTRAINT "takimFK" FOREIGN KEY ("orgNo") REFERENCES
github."Organizasyon"("orgNo") ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY github."TakimliUye"

ADD CONSTRAINT "takimliUyeFK" FOREIGN KEY ("takimNo") REFERENCES
github."Takim"("takimNo");

ALTER TABLE ONLY github."Takip"

```
ADD CONSTRAINT "takipFK1" FOREIGN KEY ("takipciNo") REFERENCES
github."Kullanici"("kullaniciNo") ON UPDATE CASCADE ON DELETE CASCADE;
```

```
ALTER TABLE ONLY github."Takip"
```

```
ADD CONSTRAINT "takipFK2" FOREIGN KEY ("takipEdenNo") REFERENCES
github."Kullanici"("kullaniciNo") ON UPDATE CASCADE ON DELETE CASCADE;
```

```
ALTER TABLE ONLY github."Watcher"
```

```
ADD CONSTRAINT "watcherFK1" FOREIGN KEY ("kullaniciNo") REFERENCES
github."Kullanici"("kullaniciNo") ON UPDATE CASCADE ON DELETE CASCADE;
```

```
ALTER TABLE ONLY github."Watcher"
```

```
ADD CONSTRAINT "watcherFK2" FOREIGN KEY ("repoNo") REFERENCES
github."Repository"("repoNo");
```

```
ALTER TABLE ONLY github."Wiki"
```

```
ADD CONSTRAINT "wikiFK1" FOREIGN KEY ("kullaniciNo") REFERENCES
github."Kullanici"("kullaniciNo") ON UPDATE CASCADE ON DELETE CASCADE;
```

```
ALTER TABLE ONLY github."Wiki"
```

```
ADD CONSTRAINT "wikiFK2" FOREIGN KEY ("repoNo") REFERENCES
github."Repository"("repoNo") ON UPDATE CASCADE ON DELETE CASCADE;
```