

# **VERİ TABANI YÖNETİMİ**

## **PROJE ÖDEVİ**

**BLM 3041**

**Grup 29**

**İŞBUL – İŞ BAŞVURU SİSTEMİ**

Beyza FINDIK - 20011076

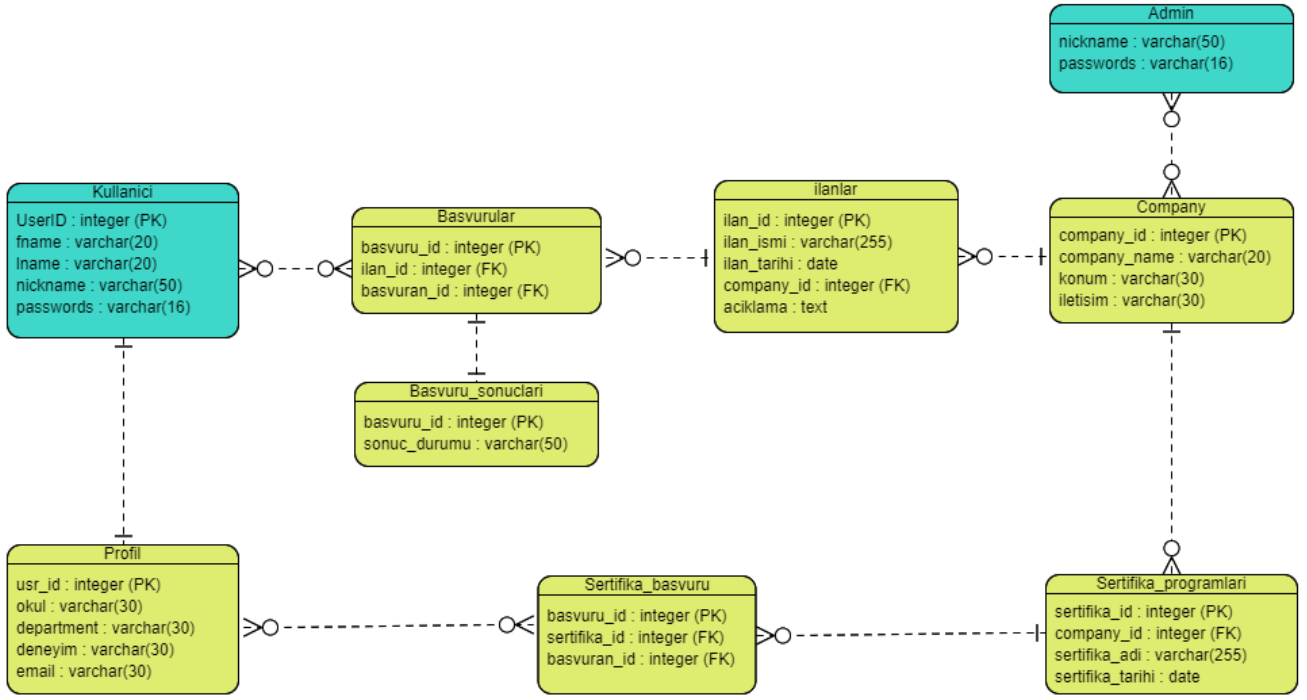
Ebru KILIÇ - 23011602

Merve ÇAKIR - 23011603

Sevda KARAHAN - 20011101

Ocak, 2024

## ER DİYAGRAMI



## TABLÖLAR

### Kullanici

	userid [PK] integer	fname character varying (20)	lname character varying (20)	nickname character varying (50)	passwords character varying (16)
1	253678	Stanford	Esley	StanfordEsley253678	12345esley
2	253679	Annice	Palk	AnnicePalk253679	12345palk
3	253680	Armin	Leall	ArminLeall253680	12345leall
4	253681	Trixy	Toville	TrixyToville253681	12345totrix
5	253682	Salaidth	Dart	SalaidthDart253682	dart56789
6	253683	North	Cannaway	NorthCannaway253683	22north22
7	253684	Caroljean	Leeder	CaroljeanLeeder253684	carol13jean
8	253685	Gabi	Brinsden	GabiBrinsden253685	gabi01sden
9	253686	Minette	Rank	MinetteRank253686	mine61rank
10	253687	Stefan	Farnall	StefanFarnall253687	stef17an11

### Admin

	nickname character varying (50)	passwords character varying (16)
1	BeyzaFındık	12345beyza
2	EbruKılıç	12345ebru
3	MerveÇakır	12345merve
4	SevdaKarahan	12345sevda

## Company

	company_id [PK] integer	company_name character varying (20)	konum character varying (30)	iletisim character varying (30)
1	1880	Linkbuzz	Ankara	linkbuzz@mail.com
2	4606	Centidel	Ankara	centidelAnkara@mail.com
3	4811	Jazzy	Istanbul	JazzyIst@mail.com
4	1453	Kazu	Istanbul	KazuIstanbul@mail.com
5	1077	Trudeo	Kocaeli	Trudeo@mail.com
6	6839	Kazu	Tokyo	KazuTokio@mail.com
7	3257	Limbo	Ankara	limboAnkara@mail.com
8	4657	Martı	Istanbul	MartıTR@mail.com
9	6020	Layo	Ankara	Layoo@mail.com
10	5522	Shuffledrive	Kocaeli	Shuffledrive@mail.com

## Sertifika\_programlari

	sertifika_id [PK] integer	company_id integer	sertifika_adi character varying (50)	sertifika_tarihi date
1	1	1880	Software Development Certificate	2022-05-20
2	2	4606	Data Analysis Certificate	2022-07-15
3	3	4811	Project Management Certification	2022-09-30
4	4	1453	Cloud Computing Certificate	2022-11-10
5	5	1077	Digital Marketing Certification	2023-01-05
6	6	6839	Cybersecurity Certificate	2023-03-20
7	7	3257	Machine Learning Certification	2023-05-15
8	8	4657	Network Administration Certificate	2023-07-01
9	9	6020	Web Development Certification	2023-09-10
10	10	5522	Database Management Certificate	2023-11-25

## Sertifika\_basvuru

	basvuru_id [PK] integer	sertifika_id integer	basvuran_id integer
1	100101	1	253678
2	100102	2	253679
3	100103	3	253680
4	100104	4	253681
5	100105	5	253682
6	100106	6	253683
7	100107	7	253684
8	100108	8	253685
9	100109	9	253686
10	100110	10	253687

## Profil

	usr_id [PK] integer	okul character varying (30)	department character varying (30)	deneyim character varying (30)	email character varying (30)
1	253678	Istanbul Technical University	Industrial Engineer	staj	esleystanford0101@mail.com
2	253679	Oxford University	Mechanical Engineer	part-time worker	palkannice11@mail.com
3	253680	Harvard University	Electrical Engineer	staj	leallarminn@mail.com
4	253681	Yildiz Technical University	Computer Engineer	part time developer	toville2rix@mail.com
5	253682	Yildiz Technical University	Electrical Engineer	internship	dartsalaidh13@mail.com
6	253683	Yildiz Technical University	Industrial Engineer	internship	northcannaway02@mail.com
7	253684	Yildiz Technical University	Computer Engineer	internship	carolLeeder6@mail.com
8	253685	Karadeniz Technical University	Electrical Engineer	internship at x	gabigabi6@mail.com
9	253686	Istanbul Technical University	Computer Engineer	internship	minetteRank7@mail.com
10	253687	Munich Technical University	Computer Engineer	internship	FarnallStefan01@mail.com

## İlanlar

	ilan_id [PK] integer	ilan_ismi character varying (255)	ilan_tarihi date	aciklama text	company_id integer
1	101	Veri Analisti	2023-07-30	Veri analizi konusunda deneyimli adaylar aranmaktadır. İstatistik bilgisi tercih sebebidir.	1880
2	102	Yazılım Mühendisi	2023-07-07	Yazılım geliştirme konusunda uzman adaylar aranmaktadır. Java, C++, Python bilenler tercih edilir.	4606
3	103	Grafik Tasarımcı	2023-11-12	Kreatif ve yenilikçi düşünebilen grafik tasarımcı aranmaktadır. Adobe Creative Suite bilgisi önemlidir.	4811
4	104	Sistem Analisti	2023-06-12	Bilgi sistemleri ve iş süreçleri konusunda deneyimli sistem analisti aranmaktadır.	1453
5	105	Finans Uzmanı	2023-12-16	Finans alanında uzman adaylar aranmaktadır. Muhasebe ve finans konularında bilgi sahibi olmak önemlidir.	1077
6	106	Mobil Uygulama Geliştirici	2023-03-29	iOS ve Android platformları için mobil uygulama geliştirme konusunda uzman adaylar aranmaktadır.	6839
7	107	Yazılım Test Mühendisi	2023-12-15	Yazılım test süreçlerine hakim adaylar aranmaktadır. Test senaryolarını oluşturabilme yeteneği önemlidir.	6839
8	108	İnsan Kaynakları Uzmanı	2023-10-17	İnsan kaynakları yönetimi konusunda deneyimli uzman aranmaktadır. İletişim becerileri yüksek adaylar tercih e...	4657
9	109	Veritabanı Yöneticisi	2023-10-12	Veritabanı yönetimi konusunda uzman adaylar aranmaktadır. SQL bilgisi önemlidir.	6020
10	110	Proje Yöneticisi	2023-05-13	Büyük ölçekli projeleri yönetebilecek deneyime sahip proje yöneticisi aranmaktadır.	5522

## Basvurular

	basvuru_id [PK] integer	ilan_id integer	basvuran_id integer
1	1	101	253678
2	2	102	253679
3	3	103	253680
4	4	104	253681
5	5	105	253682
6	6	106	253683
7	7	107	253684
8	8	108	253685
9	9	109	253686
10	10	110	253687

## Basvuru\_sonuc\_lari

	basvuru_id [PK] integer	sonuc_durumu character varying (50)
1	1	Kabul
2	2	Kabul
3	3	Kabul
4	4	Red
5	5	Beklemede
6	6	Red
7	7	Beklemede
8	8	Kabul
9	9	Red
10	10	Beklemede

## SQL KODLARI

### 1) Oluşturacağınız veritabanı en az 4 tablo içermelidir. Her tabloda en az 10 kayıt bulunmalıdır.

İş başvuru sistemi, kullanıcı, admin, şirket, sertifika programları, sertifika başvuruları, profil, ilanlar, başvurular ve başvuru sonuçları olmak üzere 9 tablodan oluşur. Admin olarak 4 kişinin erişimi vardır. Admin dışındaki 8 tablo 10 kayıt içerir.

### 2) Tablolarınızda primary key ve foreign key kısıtlarını kullanmalısınız.

UserID serial primary key

(Kullanıcı tablosunda her kullanıcı için benzersiz veri içeren UserID isimli sütun bulunur.)

FOREIGN KEY (basvuran\_id) REFERENCES Kullanici(UserID)

(Sertifika başvuruları tablosunda bulunan başvuru numarası, kullanıcı tablosunda bulunan kullanıcı numarasına referans eder.)

### 3) En az bir tabloda silme kısıtı ve sayı kısıtı olmalıdır.

FOREIGN KEY (company\_id) REFERENCES Company(company\_id) ON DELETE CASCADE

(Sertifika programları tablosundaki şirket numarasının silinmesi durumunda şirket tablosundaki şirket numarası sütununun silinmesi kısıtı konulmuştur.)

passwords varchar(16) check(char\_length(passwords) >= 8) NOT NULL

(Kullanıcı tablosundaki şifre sütunu verilerinin 8 ile 16 karakter arasında olması için sayı kısıtı konulmuştur.)

**4) Arayüzden en az birer tane insert, update ve delete işlemi gerçekleştirilebilmelidir.**

String query3 = "insert into users (fname, lname, passwords, nickname) values (?, ?, ?, ?)";  
(Arayüzdeki Register.java panelinden alınan isim, soy isim, şifre ve takma isim verilerinin query3 sql sorgusu sayesinde insert işlemi gerçekleşir.)

String query = "UPDATE profil SET okul = ?, department = ?, deneyim = ?, email = ? WHERE usr\_id = ?";

(Arayüzdeki ProfiliDuzenle.java panelinden alınan okul, departman, deneyim ve email verilerinin query sql sorgusu sayesinde update işlemi gerçekleşir. Aynı zamanda bu panel üzerinde profili silme ve güncelleme işlemleri de gerçekleştirilmektedir.)

query = "DELETE FROM sertifika\_programlari WHERE sertifika\_id = ?";

(Arayüzdeki Certificate.java panelinden alınan sertifika numarası verisinin query sql sorgusu sayesinde delete işlemi gerçekleşir.)

**5) Arayüzden girilecek bir değere göre ekrana sonuçların listelendiği bir sorgu yazmalısınız.**

String query = "SELECT \* FROM sertifika\_programlari WHERE sertifika\_id::text || company\_id::text || sertifika\_adi || sertifika\_tarihi ILIKE ?";

(Arayüzdeki Certificate.java panelinden belirli kriterlere uyan metin, sertifika adı ve sertifika tarihi verilerini döndürür.)

**6) Arayüzden çağrılan sorgulardan en az biri “view” olarak tanımlanmış olmalıdır.**

```
CREATE VIEW sertifika AS
SELECT cr.basvuran_id, cpr.sertifika_adi, cr.sertifika_id, cmp.company_name
FROM sertifika_basvuru cr, sertifika_programlari cpr, company cmp
WHERE cr.sertifika_id = cpr.sertifika_id and cpr.company_id = cmp.company_id;
```

String query = "SELECT sertifika\_adi, company\_name FROM sertifika WHERE basvuran\_id = ?";

(sertifika isminde oluşturulmuş view ProfilDeneme.java panelindeki sorgudan çağrılmıştır. Bu sayede kullanıcıya ait sertifika bilgileri profilde gösterilebilmektedir.)

**7) En az bir adet “sequence” oluşturmalı ve arayüzden yapılacak insert sırasında ilgili sütundaki değerlerin otomatik olarak atanmasını sağlamalısınız.**

```
CREATE OR REPLACE FUNCTION get_userID()
RETURNS INTEGER AS $$
DECLARE
    maxUserID INTEGER;
    sequenceName TEXT;
    nextValResult INTEGER;
    shouldIncrement BOOLEAN;
BEGIN
```

```

SELECT MAX(UserID) INTO maxUserID FROM Kullanici;
sequenceName := 'seq_' || (maxUserID + 1);
shouldIncrement := FALSE;
IF NOT EXISTS (SELECT 1 FROM information_schema.sequences WHERE sequence_name =
sequenceName) THEN
    EXECUTE 'CREATE SEQUENCE ' || sequenceName || ' START WITH ' || (maxUserID + 1) || '
INCREMENT BY 1';
    shouldIncrement := TRUE;
END IF;
IF shouldIncrement THEN
    EXECUTE 'SELECT nextval(' || sequenceName || ')' INTO nextValResult;
    RETURN nextValResult;
ELSE
    RETURN maxUserID + 1;
END IF;
END;
$$ LANGUAGE plpgsql;


```

String query2 = "select generate\_nickname(get\_userID(), ?, ?)";

String query3 = "insert into users (fname, lname, passwords, nickname) values (?, ?, ?, ?)";

(get\_userID fonksiyonunun içinde oluşturulan sequence kullanılarak öncelikle sırada bulunan numaranın UserID verisine atanması sağlanır ve ardından insert sütununda otomatik atanması sağlanır.)

## 8) Arayüzden çağrılan sorgulardan en az birinde union veya intersect veya except kullanmış olmalısınız.


 Veritabanından sertifika program bilgilerini çekip panellere ekleyen metod içerisinde union kullanılmıştır.

```

String query = "SELECT sertifika_id, sertifika_adi, company_id, sertifika_tarihi " + "FROM
sertifika_programlari " + "WHERE sertifika_adi ILIKE ? " + "UNION " + "SELECT sertifika_id,
sertifika_adi, company_id, sertifika_tarihi " + "FROM sertifika_programlari " + "WHERE company_id
IN (SELECT company_id FROM company WHERE company_name ILIKE ?)" + (yenidenEskiye ?
"ORDER BY sertifika_tarihi DESC" : "");

```

## 9) Sorgularınızın en az biri aggregate fonksiyonlar içermeli, having ifadesi kullanılmalıdır.

 Filtrelemede seçilen şehire göre kaç tane o konumda ilan olduğunu ekrana döndürür.

```

select count(*) as s
from company c, ilanlar i
where c.company_id = i.company_id and i.konum=?
group by c.konum
having c.konum=?;

```

**10) Arayüzden girilen değerleri parametre olarak alıp ekrana sonuç döndüren 3 farklı SQL fonksiyonu tanımlamış olmalısınız. Bu fonksiyonların en az birinde “record” ve “cursor” tanımı-kullanımı olmalıdır.**

🔗 Birinci fonksiyon, generate\_nickname fonksiyonu. Arayüzdeki Register.java panelinden girilen isim ve soy isim verilerini parametre olarak ekrana sonuç döndürür:

```
CREATE OR REPLACE FUNCTION generate_nickname(p_userID integer, p_fname varchar, p_lname
varchar)
RETURNS varchar AS $$
DECLARE
    nickname varchar;
BEGIN
    nickname := p_fname || p_lname || p_userID;
    RETURN nickname;
END;
$$ LANGUAGE plpgsql;
```

String query3 = "insert into users (fname, lname, passwords, nickname) values (?, ?, ?, ?)";

🔗 İkinci fonksiyon (record ve cursor içeriyor) profil bilgilerini düzenleme ekranından kişinin bilgilerini güncelleyebilmesi için oluşturuldu:

```
CREATE TYPE guncel1 AS (new_okul varchar(30), new_dept varchar(30), new_deneyim varchar(30),
new_email varchar(30));
```

```
CREATE OR REPLACE FUNCTION profil_guncelle2(kullanici_id INT, new_okul VARCHAR(30),
new_dept VARCHAR(30), new_deneyim VARCHAR(30), new_email VARCHAR(30))
RETURNS guncel1 AS $$
DECLARE
    profil_bilgi guncel1;
    profil_cursor CURSOR FOR SELECT usr_id, okul, department, deneyim, email FROM profil WHERE
usr_id = kullanici_id FOR UPDATE;
BEGIN
    FOR profil_row IN profil_cursor
    LOOP
        -- Her bir kayıt için güncelleme işlemi yap
        UPDATE profil
        SET okul = new_okul, department = new_dept, deneyim = new_deneyim, email = new_email
        WHERE usr_id = profil_row.usr_id
        RETURNING okul, department, deneyim, email INTO profil_bilgi;
        RETURN profil_bilgi; -- Güncellenen bilgiyi döndürür
    END LOOP;
END;
$$ LANGUAGE 'plpgsql';
```

🔗 Üçüncü fonksiyonda girilen company\_id(şirket id numarası) değerine göre şirketin ismi yazdırılıyor



```
CREATE OR REPLACE FUNCTION get_company_name(companyId int, OUT comp_name varchar(20))
RETURNS varchar(20) AS $$
BEGIN
    SELECT company_name INTO comp_name FROM company WHERE company_id = companyId;
END;
$$ LANGUAGE plpgsql;
-- Fonksiyonun çağırılması
SELECT get_company_name(4657);
```

**11) İki adet trigger tanımlamalı ve arayüzden girilecek değerlerle tetiklemelisiniz. Trigger'ın çalıştığına dair arayüze bilgilendirme mesajı döndürülmelidir.**

🚦 İlk trigger yeni bir şirket eklenmek istendiğinde company\_id eşsiz mi diye kontrol ediyor. Eğer bu company\_id'ye sahip başka bir şirket varsa bu kayıt eklenmiyor. company\_id eşsiz ise kayıt başarılı bir şekilde ekleniyor.


```
CREATE OR REPLACE FUNCTION sirket_id_kontrol()
RETURNS TRIGGER AS $$
DECLARE
    found_company integer := 0;
    companyID integer := new.company_id;
    cursor_companies CURSOR FOR SELECT company_id FROM company;
BEGIN
    FOR c_row IN cursor_companies LOOP
        IF c_row.company_id = companyID THEN
            found_company := 1;
            RAISE WARNING 'Şirket ID eşsiz olmalı. Bu şirket ID kayıtlı başka bir şirkete aittir.';
            RETURN NULL;
        END IF;
    END LOOP;
    IF found_company = 0 THEN
        RAISE INFO 'Yeni şirket eklendi.';
        RETURN NEW;
    END IF;
END;
```

```
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER sirket_id_kontrol_trigger  
BEFORE INSERT ON company  
FOR EACH ROW  
EXECUTE FUNCTION sirket_id_kontrol();
```

*Oluşturulan ilk triggerın tetiklenmesi:*

```
insert into Company (company_id, company_name, konum, iletisim) values (4606, 'Centidel',  
'Ankara', 'centidelAnkara@mail.com');
```

 *Oluşturulan ikinci triggerın işleyişi ise şu şekilde: Bir kayıt, sertifika\_programlari tablosuna eklenmek istendiğinde, eklemek istenen sertifika programına ait company\_id bilgisine sahip herhangi bir şirketin olup olmadığının kontrolü yapılır. Eğer şirket mevcutsa, eklenmek istenen kayıt başarılı bir şekilde kaydedilir. Eğer böyle bir company\_id'ye sahip bir şirket yoksa, kayıt işlemi başarısız olur ve ekrana sertifika ekleme işleminin başarısız olduğuna dair bir uyarı mesajı yazdırılır.*

```
CREATE OR REPLACE FUNCTION sertifika_sirket_id_kontrol()  
RETURNS TRIGGER AS $$  
BEGIN  
    IF (SELECT company_id FROM company WHERE company_id = new.company_id) IS NOT NULL  
    THEN  
        RAISE WARNING 'Kayıt başarılı.';  
        RETURN NEW;  
    ELSE  
        RAISE INFO 'Şirket ID kayıtlı değil!';  
        RAISE INFO 'Kayıtlı bir Şirket ID ile sertifika programı ekleyebilirsiniz.';  
        RETURN NULL;  
    END IF;  
END;  
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER sertifika_sirket_id_kontrol_trigger  
BEFORE INSERT ON sertifika_programlari
```

FOR EACH ROW

EXECUTE FUNCTION sertifika\_sirket\_id\_kontrol();

*Oluşturulan ikinci triggerın tetiklenmesi:*

insert into Sertifika\_programlari (company\_id, sertifika\_adi, sertifika\_tarihi) values (9876543, 'Software Development Certificate', '2022-05-20');