

Veri Tabanı Modelleme ve Tasarımı BLM5225



Öğr. Grv. Furkan ÇAKMAK

Ders İşleyiş Bilgilendirmesi

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 1

- Teorik Ders Saatleri: **Pazartesi 19:00 - 21:50**
- Dersler şimdilik online ortamda olacak.
- Duyurular için: <https://avesis.yildiz.edu.tr/fcakmak/dokumanlar>
- Google Classroom Sınıf Kodu: **nuhk4hb**
 - Dersle ilgili sorularınızı Classroom üzerinden duvar paylaşımı yaparak sorabilirsiniz.
- Zoom Personal Room:
 - <https://us04web.zoom.us/j/3752287039?pwd=TTFKNittZWJTUEhHREovckl0VTVYUT09>
 - Meeting ID: 375 228 7039
 - Passcode: 5AXMNd
- Dersle ilgili soru ve danışmalarınız için EN AZ 1 (BİR) GÜN ÖNCEDEN linkten randevu alınız.
 - <https://fcakmak.simplybook.it/v2/>
 - Mail ile randevu verilmeyecektir.
 - Alınan randevu saatinde lütfen yukarıda verilen Zoom odasında bulununuz.

Öğr. Grv. Furkan Çakmak

Ders Tanıtım Formu ve Konular

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 1

Hafta	Tarih	Konular
1	27.02.2023	Ders İçeriği Tanıtımı ve Kavram Açıklamaları
2	6.03.2023	SQL'e Giriş; DDL ve DML Komutlarına Giriş, company-db'nin Tanıtımı
3	13.03.2023	PostgreSQL Ortamının Tanıtımı ve Sorgulama Örnekleri
4	20.03.2023	Tablolarda Kısıt, View ve Sequence İşlemleri; Union, Intersect, Except İşlemleri
5	27.03.2023	Tablolarda Gruplama ve Sıralama Fonksiyonları
6	3.04.2023	PL/pgSQL Fonksiyon Tanıtımı
7	10.04.2023	PL/pgSQL Alias, Record/Cursor ve Trigger Tanımları
8	17.04.2023	JDBC ile Veri Tabanına Bağlanıp Sorgu Yapma Uygulamaları
9	24.04.2023	Ara Sınav - 1
10	1.05.2023	Tatil - İşçi Bayramı
11	8.05.2023	Hibernate Uygulamaları
12	15.05.2023	Xpath ve Xquery Yapısı
13	22.05.2023	Proje Gösterimleri - 1
14	29.05.2023	Proje Gösterimleri - 2

Öğr. Grv. Furkan ÇAKMAK

Ders Tanıtım Formu - Ders İçin Kaynaklar

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 1

	Kaynaklar
1	PostgreSQL Tutorials: https://wiki.postgresql.org/wiki/PostgreSQL_Tutorials
2	Veri Tabanı Sistemleri, Ünal Yarımağan

Öğr. Grv. Furkan ÇAKMAK

Ders Tanıtım Formu - Değerlendirme

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 1

Başarı Değerlendirme Sistemi	Yöntem	Adedi	Etki Oranı (%)
	Ara Sınavlar	1	30
	Kısa Sınavlar	-	-
	Ödevler	-	-
	Projeler	1	30
	Dönem Ödevi	-	-
	Laboratuvar	-	-
	Diğer	-	-
	Final Sınavı	1	40

Öğr. Grv. Furkan ÇAKMAK

Önce Kavramlar

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 1

- Veri Tabanı Nedir?
- Veri Tabanı Yönetim Sistemi (VTYS) Nedir?
- Veri Nasıl Temsil Edilir?
- Tablo Nedir?
- Kolon Nedir?

MEYVE		
meyveld	integer	primary key
meyveAdi	varchar(50)	not null
Bolgesi	smallint	not null
Mevsimi	smallint	
ort_agirligi	float	
uretim_miktari	bigint	
ital_mi	boolean	default true
rengi	char(40)	



Öğr. Grv. Furkan ÇAKMAK

Sabırla Dinlediğiniz İçin Teşekkürler

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 1



Öğr. Grv. Furkan Çakmak

Veri Tabanı Modelleme ve Tasarımı BLM5225



Öğr. Grv. Furkan ÇAKMAK

Ders Tanıtım Formu ve Konular

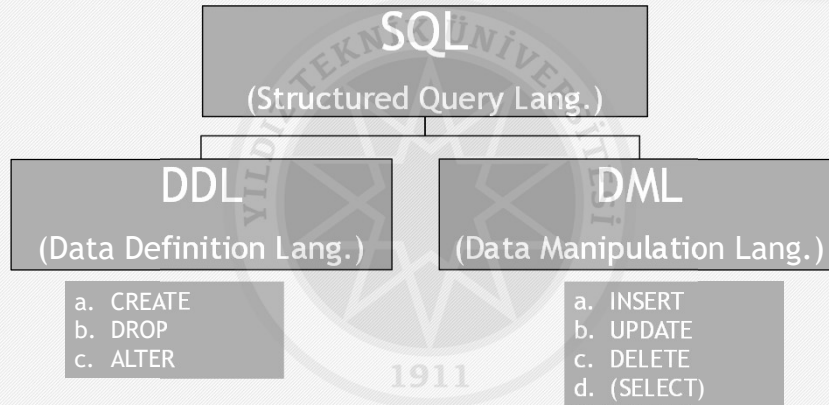
BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 2

Hafta	Tarih	Konular
1	27.02.2023	Ders İçeriği Tanıtımı ve Kavram Açıklamaları
2	6.03.2023	SQL'e Giriş; DDL ve DML Komutlarına Giriş; company-db'nin Tanıtımı
3	13.03.2023	PostgreSQL Ortamının Tanıtımı ve Sorgulama Örnekleri
4	20.03.2023	Tablolarda Kısıt, View ve Sequence İşlemleri; Union, Intersect, Except İşlemleri
5	27.03.2023	Tablolarda Gruplama ve Sıralama Fonksiyonları
6	3.04.2023	PL/pgSQL Fonksiyon Tanımı
7	10.04.2023	PL/pgSQL Alias, Record/Cursor ve Trigger Tanımları
8	17.04.2023	JDBC ile Veri Tabanına Bağlanıp Sorgu Yapma Uygulamaları
9	24.04.2023	Ara Sınav - 1
10	1.05.2023	Tatil - İşçi Bayramı
11	8.05.2023	Hibernate Uygulamaları
12	15.05.2023	Xpath ve Xquery Yapısı
13	22.05.2023	Proje Gösterimleri - 1
14	29.05.2023	Proje Gösterimleri - 2

Öğr. Grv. Furkan ÇAKMAK

SQL

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 2



DDL - CREATE Komutu

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 2

```
CREATE TABLE ( Kolonİsmi  
    KolonTipi(Boyutu) PrimaryKey,  
    Kolonİsmi KolonTipi(Boyutu) Default (Default value),  
    Kolonİsmi KolonTipi(Boyutu) not null,  
    Kolonİsmi KolonTipi(Boyutu), Kolonİsmi KolonTipi,  
    Primary key (PkeyField1,PkeyField2)  
);
```

MEYVE		
meyveld	integer	primary key
meyveAdi	varchar(50)	not null
Bolgesi	smallint	not null
Mevsimi	smallint	
ort_agirligi	float	
uretim_miktari	bigint	
ital_mi	boolean	default true
rengi	char(40)	

```
CREATE TABLE Meyve(  
    meyveld int [primary key],  
    meyveAdi varchar(50) not null,  
    bolgesi smallint not null,  
    mevsimi smallint,  
    ort_agirligi float,  
    uretim_miktari bigint,  
    ital_mi boolean default true,  
    rengi varchar(50)  
    [, primary key (meyveld)]
```

```
);  
Öğr. Grv. Furkan ÇAKMAK
```

DDL - CREATE Komutu (Devam)

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 2

OGRENCI		
girisYili	numeric(4)	not null, default 2023
bolumKodu	numeric(3)	not null
siralamasi	Numeric	not null
ismi	Varchar	not null
eposta	char	not null unique essiz_posta

```
CREATE TABLE Ogresnci(  
  girisYili numeric(4) not null default 2018,  
  bolumKodu numeric(3) not null,  
  siralamasi numeric not null,  
  isim varchar not null,  
  email char not null [unique],  
  primary key(girisYili, bolumKodu, siralamasi)  
  [, constraint essiz_mail unique(email)]  
);
```

Öğr. Grv. Furkan ÇAKMAK

DDL - DROP ve ALTER

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 2

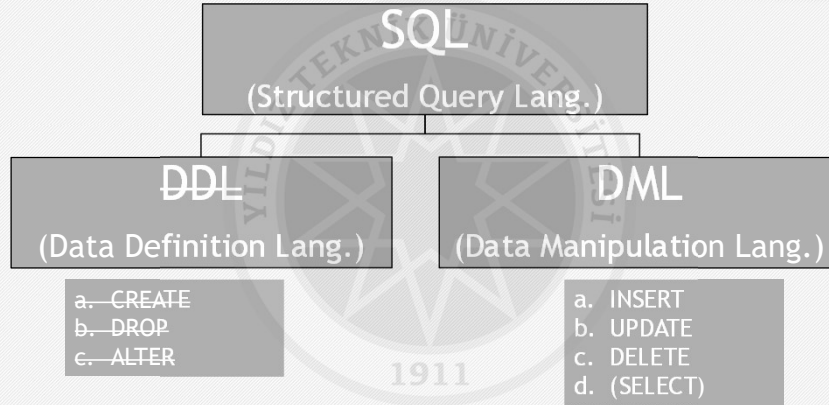
- Öğrenci tablosuna TCKimlikNo isimlik bir kolon ekleyelim.
- ALTER TABLE Ogresnci ADD [COLUMN] tckimlikno numeric(11) unique not null;
- TCKimlikNo kolonunu silelim.
- ALTER TABLE Ogresnci DELETE COLUMN tckimlikno;
- Öğrenci tablosundan email kolonunun unique olan kısıtını silelim.
- ALTER TABLE Ogresnci DROP CONSTRAINT essiz_mail;
- Öğrenci tablosundaki ismi kolonunu ad_soyad olarak değiştirilelim.
- ALTER TABLE Ogresnci RENAME ismi TO ad_soyad;
- Öğrenci tablosundaki ad_soyad kolonunun tipini char olarak değiştirilelim.
- ALTER TABLE Ogresnci ALTER COLUMN ad_soyad TYPE char;
- Öğrenci tablosunun ismini Talebe tablosu olarak değiştirilelim.
- ALTER TABLE Ogresnci RENAME TO Talebe;
- Talebe tablosunu silelim.
- DROP TABLE Talebe;

OGRENCI		
girisYili	numeric(4)	not null, default 2023
bolumKodu	numeric(3)	not null
siralamasi	Numeric	not null
ismi	Varchar	not null
eposta	char	not null unique essiz_posta
TCKimlikNo	numeric(11)	unique not null

Öğr. Grv. Furkan ÇAKMAK

SQL

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 2



DML - INSERT, DELETE, UPDATE

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 2

- Meyve tablosuna girdi yapalım.
- INSERT INTO Meyve VALUES (0, 'Elma', 3, 2, 0.2, 120000, false, 'Muhtelif');
- INSERT INTO Meyve(meyveld, meyveAdi, bölgesi) VALUES (1, 'Armut', 2);
- meyveld'si 1 olan meyveyi siliniz.
- DELETE FROM Meyve WHERE meyveld = 1;
- Ortalama ağırlığı 0.1'den küçük ve ital olan meyveleri siliniz.
- DELETE FROM Meyve WHERE ort_agirligi < 0.1 AND ital_mi = true;
- Numarası 68 olan meyvenin ortalama ağırlığını 0.6, üretim miktarını 123123 olarak değiştiriniz.
- UPDATE Meyve SET ort_agirligi = 0.6, uretim_miktari = 123123 WHERE meyveld = 68;
- Kış mevsiminde (4) yetişen meyvelerin rengini yeşil yapınız.
- UPDATE Meyve SET rengi = 'Yeşil' WHERE mevsimi = 4;

MEYVE		
meyveld	integer	primary key
meyveAdi	varchar(50)	not null
Bolgesi	smallint	not null
Mevsimi	smallint	
ort_agirligi	float	
uretim_miktari	bigint	
ital_mi	boolean	default true
rengi	char(40)	

DML - SELECT

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 2

- Öğrencilerin isim ve giriş yıllarını listeleyen sorguyu yazınız.
- SELECT ismi, girişYılı FROM Öğrenci;
- Bölümüne 3. sırada girmiş Mehmet ismindeki öğrencileri bulunuz.
- SELECT * FROM Öğrenci WHERE siralamasi = 3 AND ismi = 'Mehmet'

MEYVE		
meyvelid	integer	primary key
meyveAdi	varchar(50)	not null
Bolgesi	smallint	not null
Mevsimi	smallint	
ort_agirligi	float	
uretim_miktari	bigint	
ital_mi	boolean	default true
rengi	char(40)	

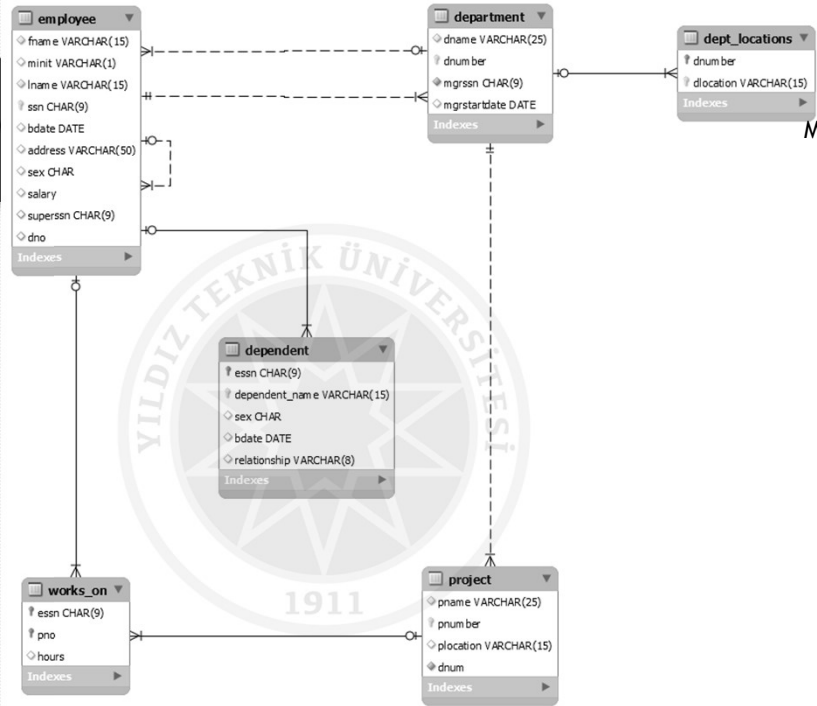
OGRENCI		
girisYili	numeric(4)	not null, default 2023
bolumKodu	numeric(3)	not null
siralamasi	Numeric	not null
ismi	Varchar	not null
eposta	char	not null unique essiz_posta

- Adında 'm' harfi geçen meyvelerin ortalama ağırlığını listeleyiniz.
- SELECT ort_agirligi FROM Meyve WHERE meyveAdi LIKE '%m%' [OR meyveAdi LIKE '%M%'];
- 2010 yılından sonra Bilişim Teknolojileri Tezsiz Yüksek Lisans Programına (574) ilk 20'de girenlerin mail listesini bulunuz.
- SELECT email FROM Öğrenci WHERE girişYılı > '2010' AND siralamasi <= '20' AND bolumKodu = 574;

Öğr. Grv. Furkan ÇAKMAK

Company Database

- Employee
- Department
- Dept_location
- Dependent
- Project
- Works_on



BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 2

Sabırla Dinlediğiniz İçin Teşekkürler

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 2



Öğr. Grv. Furkan Çakmak

Veri Tabanı Modelleme ve Tasarımı BLM5225



Öğr. Grv. Furkan ÇAKMAK

Ders Tanıtım Formu ve Konular

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 3

Hafta	Tarih	Konular
1	27.02.2023	Ders İçeriği Tanıtımı ve Kavram Açıklamaları
2	6.03.2023	SQL'e Giriş; DDL ve DML Komutlarına Giriş, company-db'nin Tanıtımı
3	13.03.2023	PostgreSQL Ortamının Tanıtımı ve Sorgulama Örnekleri
4	20.03.2023	Tablolarda Kısıt, View ve Sequence İşlemleri; Union, Intersect, Except İşlemleri
5	27.03.2023	Tablolarda Gruplama ve Sıralama Fonksiyonları
6	3.04.2023	PL/pgSQL Fonksiyon Tanımı
7	10.04.2023	PL/pgSQL Alias, Record/Cursor ve Trigger Tanımları
8	17.04.2023	JDBC ile Veri Tabanına Bağlanıp Sorgu Yapma Uygulamaları
9	24.04.2023	Ara Sınav - 1
10	1.05.2023	Tatil - İşçi Bayramı
11	8.05.2023	Hibernate Uygulamaları
12	15.05.2023	Xpath ve Xquery Yapısı
13	22.05.2023	Proje Gösterimleri - 1
14	29.05.2023	Proje Gösterimleri - 2

Öğr. Grv. Furkan ÇAKMAK

Geçtiğimiz hafta

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 3

SQL

DDL

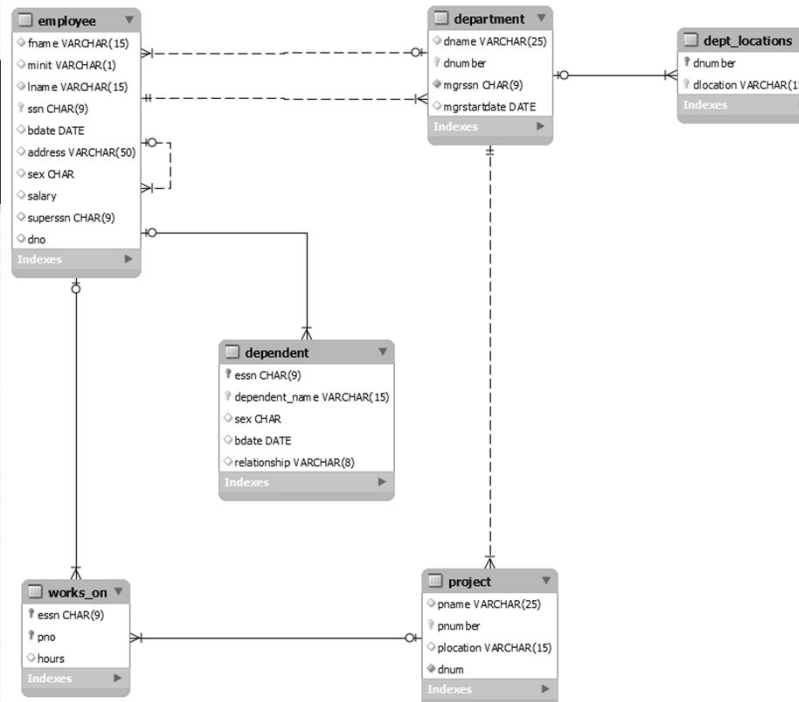
- a. CREATE
- b. DROP
- c. ALTER

DML

- a. INSERT
- b. UPDATE
- c. DELETE
- d. (SELECT)

Company DB

- Employee
- Department
- Dept_location
- Dependent
- Project
- Works_on



BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 3

Company_db Kurulumu

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 3

Sorgu Örnekleri 1

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 3

- Soru : 'Seattle' da oturan işçinin/işçilerin adını ve doğum tarihini listeleyen SQL sorgusunu yazınız.

FNAME	NOT NULL	VARCHAR(15)
MINIT		VARCHAR(1)
LNAME	NOT NULL	VARCHAR(15)
SSN	NOT NULL	CHAR(9)
BDATE		DATE
ADDRESS		VARCHAR(50)
SEX		CHAR(1)
SALARY		NUMERIC
SUPERSSN		CHAR(9)
DNO		NUMERIC

Employee Tablosu

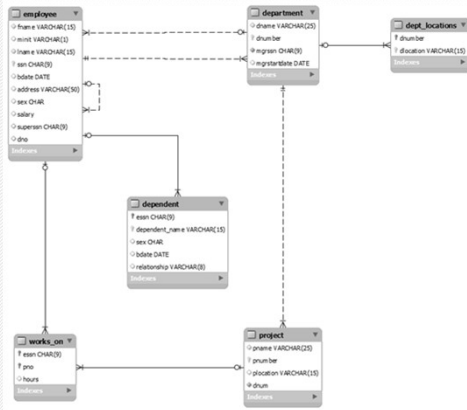
Sorgu Sonucu

	lname character varying(15)	bdate date
1	Kramer	1964-08-22
2	King	1949-08-16

Sorgu Örnekleri 2

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 3

- Soru : İsmi 'Franklin Wong' olan çalışanın çalıştığı projelerin numaralarını listeleyen SQL sorgusunu yazınız.



FNAME	NOT NULL	VARCHAR(15)
MINT		VARCHAR(1)
LNAME	NOT NULL	VARCHAR(15)
SSN	NOT NULL	CHAR(9)
BDATE		DATE
ADDRESS		VARCHAR(50)
SEX		CHAR(1)
SALARY		NUMERIC
SUPERSSN		CHAR(9)
DNO		NUMERIC

Employee Tablosu

ESSN	NOT NULL	CHAR(9)
PNO	NOT NULL	NUMERIC
HOURS		NUMERIC

Works_on Tablosu

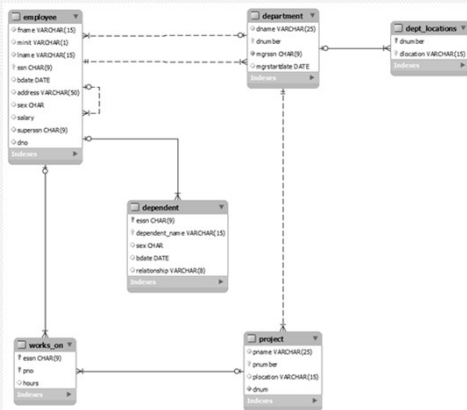
Sorgu Sonucu

	pno	smallint
1		2
2		3
3		10
4		20

Sorgu Örnekleri 3

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 3

- Soru : Şirketin satış departmanının ("Sales") hangi şehir(ler)de ofisi bulunduğunu bulan sorguyu yazınız.



DNAME	NOT NULL	VARCHAR(25)
DNUMBER	NOT NULL	NUMERIC
MGRSSN	NOT NULL	CHAR(9)
MGRSTARTDATE		DATE

Department Tablosu

DNUMBER	NOT NULL	NUMERIC
DLOCATION	NOT NULL	VARCHAR(15)

Dept_loaitons Tablosu

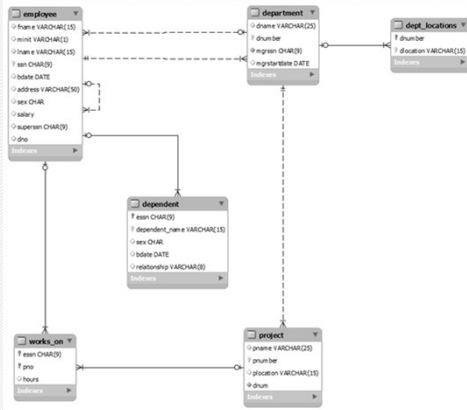
Sorgu Sonucu

	şehir	character varying(15)
1	Chicago	
2	Dallas	
3	Philadelphia	
4	Seattle	
5	Miami	

Sorgu Örnekleri 4

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 3

- Soru : 'Elizabeth' isminde akrabası olan çalışanın yöneticisinin (supervisor) adını ve soyadını bulan SQL sorgusunu yazınız.



FNAME	NOT NULL	VARCHAR(15)
MNAME		VARCHAR(1)
LNAME	NOT NULL	VARCHAR(15)
SSN	NOT NULL	CHAR(9)
BDATE		DATE
ADDRESS		VARCHAR(50)
SEX		CHAR(1)
SALARY		NUMERIC
SUPERSSN		CHAR(9)
DNO		NUMERIC

Employee Tablosu

ESSN	NOT NULL	CHAR(9)
DEPENDENT_NAME	NOT NULL	VARCHAR(15)
SEX		CHAR(1)
BDATE		DATE
RELATIONSHIP		VARCHAR(8)

Dependent Tablosu

Sorgu Sonucu

	fname character varying(15)	lname character varying(15)
1	Franklin	Wong

Sabırla Dinlediğiniz İçin Teşekkürler

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 3



Sorgu Örnekleri Cevapları

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 3

1. SELECT lname, bdate FROM employee WHERE address like '%Seattle%';
2. SELECT pno FROM employee, works_on WHERE fname = 'Franklin' AND lname = 'Wong' AND essn = ssn;
3. SELECT dlocation şehir FROM department d, dept_locations dl WHERE dname = 'Sales' AND d.dnumber=dl.dnumber;
4. SELECT e2.fname, e2.lname FROM employee e1, employee e2, dependent d WHERE d.dependent_name = 'Elizabeth' AND d.essn = e1.ssn AND e1.superssn = e2.ssn;

Veri Tabanı Modelleme ve Tasarımı BLM5225



Öğr. Grv. Furkan Çakmak

Ders Tanıtım Formu ve Konular

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 4

Hafta	Tarih	Konular
1	27.02.2023	Ders İçeriği Tanıtımı ve Kavram Açıklamaları
2	6.03.2023	SQL'e Giriş; DDL ve DML Komutlarına Giriş, company-db'nin Tanıtımı
3	13.03.2023	PostgreSQL Ortamının Tanıtımı ve Sorgulama Örnekleri
4	20.03.2023	Tablolarda Kısıt, View ve Sequence İşlemleri; Union, Intersect, Except İşlemleri
5	27.03.2023	Tablolarda Gruplama ve Sıralama Fonksiyonları
6	3.04.2023	PL/pgSQL Fonksiyon Tanımı
7	10.04.2023	PL/pgSQL Alias, Record/Cursor ve Trigger Tanımları
8	17.04.2023	JDBC ile Veri Tabanına Bağlanıp Sorgu Yapma Uygulamaları
9	24.04.2023	Ara Sınav - 1
10	1.05.2023	Tatil - İşçi Bayramı
11	8.05.2023	Hibernate Uygulamaları
12	15.05.2023	Xpath ve Xquery Yapısı
13	22.05.2023	Proje Gösterimleri - 1
14	29.05.2023	Proje Gösterimleri - 2

Öğr. Grv. Furkan ÇAKMAK

Alt Başlıklar

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 4

- «Team» ve «Team_employee» tablolarının oluşturulması
- Tablolar üzerindeki kısıtların açıklanması
- «View» tanımı ve oluşturulması
- «Sequence» tanımı ve oluşturulması
- «Union» işlemi tanımı ve örneği
- «Intersect» işlemi tanımı ve örneği
- «Except» işlemi tanımı ve örneği
- «Not Exist» işlemi tanımı ve örneği
- «In» işlemi tanımı ve örneği
- Çeşitli fonksiyon tanımları ve kullanımları

«TEAM» Tablosunun Oluşturulması

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 4

- Şirkette çalışan işçiler kendi aralarında basketbol tanımı kurmuştur.
- Her bir işçi belli hafta boyunca bu takımda oynayabilir.
- İşçilerin tek bir takımda oynama zorunluluğu yoktur.
- Takım tablosu sağdaki gibidir.

Kolon Adı	Özelliği	Tipi
TNUMBER (PK)	NOT NULL	NUMERIC(2)
TNAME	NOT NULL	VARCHAR(15)

```
CREATE TABLE TEAM (  
    TNUMBER NUMERIC(2) NOT NULL,  
    TNAME VARCHAR(15) NOT NULL,  
    PRIMARY KEY(TNUMBER)  
);
```

«TEAM_EMPLOYEE» Tablosunun Oluşturulması

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 4

- «TEAM» tablosu ve «EMPLOYEE» tablolarının ilişki tablosu olan «TEAM_EMPLOYEE» tablosu aşağıdaki özellikleri olacak şekilde oluşturulacaktır.
- Tablo nitelikleri: TNO (numeric(2)), ESSN (char(9)), PLAY_TIME (numeric(2))
 - «TNO» ve «ESSN» birlikte belirleyici nitelik olmalıdır.
 - «TNO», TEAM tablosundaki «TNUMBER»'ı referans almalıdır ve «TEAM» tablosundan bir satır silindiğinde bu satıra ait «TNO»'lu satırlar da silinmelidir.
 - «ESSN», «EMPLOYEE» tablosundaki «SSN»'i referans almalıdır ve «EMPLOYEE» tablosundan bir satır silindiğinde bu satıra ait «ESSN»'li satırlar da silinmelidir.
 - «PLAY_TIME» alanının 12'den çok olması engellenmelidir. (Bir çalışan, bir takımda 12 haftadan daha uzun süre oynayamaz.)

Kolon Adı	Tipi
TNO	NUMERIC(2)
ESSN	CHAR(9)
PLAY_TIME	NUMERIC(2)

```
CREATE TABLE TEAM_EMPLOYEE (  
    TNO NUMERIC(2), ESSN CHAR(9), PLAY_TIME NUMERIC(2),  
    CONSTRAINT PK_TEAM_EMP PRIMARY KEY(TNO,ESSN),  
    CONSTRAINT FK_TEAM FOREIGN KEY(TNO) REFERENCES TEAM(TNUMBER) ON DELETE CASCADE,  
    CONSTRAINT FK_EMP FOREIGN KEY(ESSN) REFERENCES EMPLOYEE(SSN) ON DELETE CASCADE,  
    CONSTRAINT PLAY_TIME_CK CHECK (PLAY_TIME<13)  
);
```

«View» Tanımı ve Oluşturulması

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 4

- View, sorgu sonucunda oluşan sanal tablolardır.
- VT'de fiziksel olarak bulunmazlar, sorgu olarak saklanırlar.
- View'lerin sorgularında adı geçen tabloların içeriği değiştirildiğinde View'lerin ürettiği sonuç da otomatik olarak değişir.
- Syntax'ı aşağıdaki gibidir:

```
CREATE VIEW viewname AS  
SELECT select-list  
FROM table-list  
[WHERE search-condition]
```

VIEW ÖRNEKLERİ

- Maaşı 20000 ile 40000 arasında olan çalışanların isimlerini ve maaşlarını gösteren bir view yazın.
 - «Sales» isimli departmanda çalışanların ad, soyad ve cinsiyetlerini gösteren bir view yazın.
- CREATE VIEW maas_kriteri AS SELECT fname, lname, salary FROM employee WHERE salary between 20000 and 40000;
 - CREATE VIEW sales_calisanlari AS SELECT fname, lname, sex FROM employee e, department d WHERE dname = 'Sales' AND d.dnumber = e.dno;

«Sequence» Tanımı ve Oluşturulması

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 4

- Sequence, belli bir sırada numeric değer üretilmesini sağlayan VT nesnesidir.
- Genelde birincil anahtar gibi eşsiz (unique) olan alanların değerlerinin otomatik arttırılması için kullanılır.
- Syntax'ı aşağıdaki gibidir:
- «SELECT * FROM sequence_name;» // bilgi alma

```
CREATE SEQUENCE sequence_name  
[INCREMENT BY #]  
[START WITH #]  
[MAXVALUE # | NOMAXVALUE]  
[MINVALUE # | NOMINVALUE]  
[CYCLE | NO CYCLE] // Son değerden sonra başa dönme durumu  
[CACHE #] // Hız için bellek kullanımı
```

SEQUENCE ÖRNEĞİ

1. 9'dan başlayıp 99'a kadar birer birer artan bir «sequence» oluşturun. Bu sequence'i «TEAM» tablosunun 'tnumber' larının numaralandırılması için kullanın.
 - A. CREATE SEQUENCE seq_tnumber MINVALUE 9 MAXVALUE 99 INCREMENT BY 1;
 - B. INSERT INTO team VALUES(nextval('seq_tnumber'), 'Yiğitler');

Tablolarda UNION, INTERSECT, EXCEPT İşlemleri

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 4

1. «OperatingSystems» isimli projede çalışanların ve «Software» departmanında çalışanların ad, soyad bilgilerini bulunuz. (UNION / INTERSECT / EXCEPT)
2. «OperatingSystems» isimli projede çalışanların veya «Software» departmanında çalışanların ad, soyad bilgilerini bulunuz. (UNION / INTERSECT / EXCEPT)
3. «OperatingSystems» isimli projede çalışan ama «Software» departmanında çalışmayanların ad, soyad bilgilerini bulunuz. (UNION / INTERSECT / EXCEPT)
4. Hiçbir departmanın veya hiçbir çalışanın yöneticisi olmayan çalışanların isimlerini bulunuz. (EXISTS / NOT EXISTS)
5. İsmi «John» olan işçilerin çalıştıkları departmanların isimlerini «IN» kullanarak bulunuz.
6. «Sales» departmanında kaç kişinin çalıştığını, en düşük, en yüksek, ortalama ve toplam maaşı bulunuz.

Sabırla Dinlediğiniz İçin Teşekkürler

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 4



UNION, INTERSECT, EXCEPT Sorguları Cevapları

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 4

1.

```
SELECT fname, lname FROM employee e, project p, works_on wo WHERE pname = 'OperatingSystems' AND p.pnumber = wo.pno AND wo.essn = e.ssn
INTERSECT
SELECT fname, lname FROM employee e, department d WHERE dname = 'Software' AND e.dno = d.dnumber;
```
2.

```
SELECT fname, lname FROM employee e, project p, works_on wo WHERE pname = 'OperatingSystems' AND p.pnumber = wo.pno AND wo.essn = e.ssn
UNION
SELECT fname, lname FROM employee e, department d WHERE dname = 'Software' AND e.dno = d.dnumber;
```
3.

```
SELECT fname, lname FROM employee e, project p, works_on wo WHERE pname = 'OperatingSystems' AND p.pnumber = wo.pno AND wo.essn = e.ssn
EXCEPT
SELECT fname, lname FROM employee e, department d WHERE dname = 'Software' AND e.dno = d.dnumber;
```
4.

```
SELECT fname, lname FROM employee e WHERE NOT EXISTS (SELECT null FROM department d WHERE d.mgrssn = e.ssn) AND NOT EXISTS (SELECT null
FROM employee s WHERE s.superssn = e.ssn);
```
5.

```
SELECT dname FROM department d WHERE dnumber IN (SELECT dno FROM employee WHERE fname = 'John');
```
6.

```
SELECT COUNT(*), min(salary), max(salary), avg(salary), sum(salary) FROM department, employee WHERE dname = 'Sales' AND dnumber = dno;
```

Veri Tabanı Modelleme ve Tasarımı BLM5225



Öğr. Grv. Furkan Çakmak

Ders Tanıtım Formu ve Konular

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 5

Hafta	Tarih	Konular
1	27.02.2023	Ders İçeriği Tanıtımı ve Kavram Açıklamaları
2	6.03.2023	SQL'e Giriş; DDL ve DML Komutlarına Giriş, company-db'nin Tanıtımı
3	13.03.2023	PostgreSQL Ortamının Tanıtımı ve Sorgulama Örnekleri
4	20.03.2023	Tablolarda Kısıt, View ve Sequence İşlemleri; Union, Intersect, Except İşlemleri
5	27.03.2023	Tablolarda Gruplama ve Sıralama Fonksiyonları
6	3.04.2023	PL/pgSQL Fonksiyon Tanıtımı
7	10.04.2023	PL/pgSQL Alias, Record/Cursor ve Trigger Tanımları
8	17.04.2023	JDBC ile Veri Tabanına Bağlanıp Sorgu Yapma Uygulamaları
9	24.04.2023	Ara Sınav - 1
10	1.05.2023	Tatil - İşçi Bayramı
11	8.05.2023	Hibernate Uygulamaları
12	15.05.2023	Xpath ve Xquery Yapısı
13	22.05.2023	Proje Gösterimleri - 1
14	29.05.2023	Proje Gösterimleri - 2

Öğr. Grv. Furkan ÇAKMAK

Sorgu Örneği 1

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 5

- 8 numaralı departmanda çalışan işçilerin ortalama maaşı ve toplam maaşlarını bulunuz.

Kolon Adı	Özelliği	Tipi
FNAME	NOT NULL	VARCHAR(15)
MINIT		VARCHAR(1)
LNAME	NOT NULL	VARCHAR(15)
SSN (PK)	NOT NULL	CHAR(9)
BDATE		DATE
ADDRESS		VARCHAR(50)
SEX		CHAR(1)
SALARY		NUMERIC
SUPERSSN (FK)		CHAR(9)
DNO (FK)		NUMERIC

Sorgu Örneği 2

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 5

- “Hardware” departmanında çalışanların en düşük ve en yüksek maaş miktarını bulunuz.

Kolon Adı	Özelliği	Tipi
FNAME	NOT NULL	VARCHAR(15)
MINIT		VARCHAR(1)
LNAME	NOT NULL	VARCHAR(15)
SSN (PK)	NOT NULL	CHAR(9)
BDATE		DATE
ADDRESS		VARCHAR(50)
SEX		CHAR(1)
SALARY		NUMERIC
SUPERSSN (FK)		CHAR(9)
DNO (FK)		NUMERIC

Kolon Adı	Özelliği	Tipi
DNAME	NOT NULL	VARCHAR(25)
DNUMBER (PK)	NOT NULL	NUMERIC
MGRSSN (FK)	NOT NULL	CHAR(9)
MGRSTARTDATE		DATE

Sorgu Örneği 3

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 5

- “Middleware” projesinde kaç kişinin çalıştığını ve bu çalışanların ortalama maaşını bulunuz.

Kolon Adı	Özelliği	Tipi
FNAME	NOT NULL	VARCHAR(15)
MINIT		VARCHAR(1)
LNAME	NOT NULL	VARCHAR(15)
SSN (PK)	NOT NULL	CHAR(9)
BDATE		DATE
ADDRESS		VARCHAR(50)
SEX		CHAR(1)
SALARY		NUMERIC
SUPERSSN (FK)		CHAR(9)
DNO (FK)		NUMERIC

Kolon Adı	Özelliği	Tipi
PNAME	NOT NULL	VARCHAR(25)
PNUMBER (PK)	NOT NULL	NUMERIC
PLOCATION		VARCHAR(15)
DNUM (FK)	NOT NULL	NUMERIC

Kolon Adı	Özelliği	Tipi
ESSN (PK)	NOT NULL	CHAR(9)
PNO (PK)	NOT NULL	NUMERIC
HOURS		NUMERIC

Sorgu Örneği 4

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 5

- Her bir projede çalışanların ortalama maaşını bulup **proje ismine** göre alfabetik olarak listeleyiniz.

Kolon Adı	Özelliği	Tipi
FNAME	NOT NULL	VARCHAR(15)
MINIT		VARCHAR(1)
LNAME	NOT NULL	VARCHAR(15)
SSN (PK)	NOT NULL	CHAR(9)
BDATE		DATE
ADDRESS		VARCHAR(50)
SEX		CHAR(1)
SALARY		NUMERIC
SUPERSSN (FK)		CHAR(9)
DNO (FK)		NUMERIC

Kolon Adı	Özelliği	Tipi
PNAME	NOT NULL	VARCHAR(25)
PNUMBER (PK)	NOT NULL	NUMERIC
PLOCATION		VARCHAR(15)
DNUM (FK)	NOT NULL	NUMERIC

Kolon Adı	Özelliği	Tipi
ESSN (PK)	NOT NULL	CHAR(9)
PNO (PK)	NOT NULL	NUMERIC
HOURS		NUMERIC

Sorgu Örneği 5

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 5

- Her bir departmanda, her bir cinsiyetten (M ve F) kaçar işçi olduğunu ve bu işçilerin ortalama maaşlarını bulunuz ve departman ismine göre ters alfabetik sırada sıralayınız.

Kolon Adı	Özelliği	Tipi
FNAME	NOT NULL	VARCHAR(15)
MINIT		VARCHAR(1)
LNAME	NOT NULL	VARCHAR(15)
SSN (PK)	NOT NULL	CHAR(9)
BDATE		DATE
ADDRESS		VARCHAR(50)
SEX		CHAR(1)
SALARY		NUMERIC
SUPERSSN (FK)		CHAR(9)
DNO (FK)		NUMERIC

Kolon Adı	Özelliği	Tipi
DNAME	NOT NULL	VARCHAR(25)
DNUMBER (PK)	NOT NULL	NUMERIC
MGRSSN (FK)	NOT NULL	CHAR(9)
MGRSTARTDATE		DATE

Sorgu Örneği 6

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 5

- 5 no'lu departman dışındaki departmanlar arasından, ortalama maaşı 40,000\$'dan fazla olan departmanların ismini ve o departmanlarda çalışanların ortalama maaşlarını bulunuz.

Kolon Adı	Özelliği	Tipi
FNAME	NOT NULL	VARCHAR(15)
MINIT		VARCHAR(1)
LNAME	NOT NULL	VARCHAR(15)
SSN (PK)	NOT NULL	CHAR(9)
BDATE		DATE
ADDRESS		VARCHAR(50)
SEX		CHAR(1)
SALARY		NUMERIC
SUPERSSN (FK)		CHAR(9)
DNO (FK)		NUMERIC

Kolon Adı	Özelliği	Tipi
DNAME	NOT NULL	VARCHAR(25)
DNUMBER (PK)	NOT NULL	NUMERIC
MGRSSN (FK)	NOT NULL	CHAR(9)
MGRSTARTDATE		DATE

Sorgu Örneği 7

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 5

- Şirkette en çok maaşı olan çalışanın hangi departmanda olduğunu bulunuz.

Kolon Adı	Özelliği	Tipi
FNAME	NOT NULL	VARCHAR(15)
MINIT		VARCHAR(1)
LNAME	NOT NULL	VARCHAR(15)
SSN (PK)	NOT NULL	CHAR(9)
BDATE		DATE
ADDRESS		VARCHAR(50)
SEX		CHAR(1)
SALARY		NUMERIC
SUPERSSN (FK)		CHAR(9)
DNO (FK)		NUMERIC

Kolon Adı	Özelliği	Tipi
DNAME	NOT NULL	VARCHAR(25)
DNUMBER (PK)	NOT NULL	NUMERIC
MGRSSN (FK)	NOT NULL	CHAR(9)
MGRSTARTDATE		DATE

Sabırla Dinlediğiniz İçin Teşekkürler

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 5



UNION, INTERSECT, EXCEPT Sorguları Cevapları

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 5

1. SELECT AVG(salary) ort_maas, SUM(salary) top_maas FROM employee e WHERE e.dno = 8;
2. SELECT MIN(salary) min_maas, MAX(salary) max_maas FROM employee e, department d WHERE e.dno = d.dnumber AND d.dname = 'Hardware';
3. SELECT COUNT(salary), AVG(salary) FROM project p, works_on wo, employee e WHERE e.ssn = wo.essn AND wo.pno = p.pnumber AND p.pname = 'Middleware';
4. SELECT pname, AVG(salary) FROM project p, works_on wo, employee e WHERE e.ssn = wo.essn AND wo.pno = p.pnumber GROUP BY pname ORDER BY pname;
5. SELECT dname, sex, COUNT(sex), AVG(salary) FROM employee e, department d WHERE e.dno = d.dnumber GROUP BY dname, sex ORDER BY dname DESC;
6. SELECT dname dep_ismi, AVG(salary) ort_maas FROM employee e, department d WHERE e.dno = d.dnumber AND d.dnumber <> 5 GROUP BY dname HAVING AVG(salary) > 40000;
7. A) SELECT dname FROM department WHERE dnumber IN (SELECT e2.dno FROM employee e2 WHERE e2.salary IN (SELECT MAX(e1.salary) FROM employee e1));
B) SELECT dname FROM department, employee WHERE dno = dnumber GROUP BY dname ORDER BY max(salary) DESC LIMIT 1;
C) SELECT dname FROM department WHERE dnumber IN (SELECT dno FROM employee ORDER BY salary DESC LIMIT 1);

Veri Tabanı Modelleme ve Tasarımı BLM5225



Öğr. Grv. Furkan Çakmak

Ders Tanıtım Formu ve Konular

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 6

Hafta	Tarih	Konular
1	27.02.2023	Ders İçeriği Tanıtımı ve Kavram Açıklamaları
2	6.03.2023	SQL'e Giriş; DDL ve DML Komutlarına Giriş, company-db'nin Tanıtımı
3	13.03.2023	PostgreSQL Ortamının Tanıtımı ve Sorgulama Örnekleri
4	20.03.2023	Tablolarda Kısıt, View ve Sequence İşlemleri; Union, Intersect, Except İşlemleri
5	27.03.2023	Tablolarda Gruplama ve Sıralama Fonksiyonları
6	3.04.2023	PL/pgSQL Fonksiyon Tanıtımı
7	10.04.2023	PL/pgSQL Alias, Record/Cursor ve Trigger Tanımları
8	17.04.2023	JDBC ile Veri Tabanına Bağlanıp Sorgu Yapma Uygulamaları
9	24.04.2023	Ara Sınav - 1
10	1.05.2023	Tatil - İşçi Bayramı
11	8.05.2023	Hibernate Uygulamaları
12	15.05.2023	Xpath ve Xquery Yapısı
13	22.05.2023	Proje Gösterimleri - 1
14	29.05.2023	Proje Gösterimleri - 2

Öğr. Grv. Furkan ÇAKMAK

PostgreSQL Fonksiyonel Dilleri ve PL/pgSQL

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 6

PostgreSQL'de 4 farklı fonksiyonel (prosedürel) dil var.

1. PL/pgSQL
2. PL/Tcl (C programlama dili)
3. PL/Perl (Perl Programlama Dili)
4. PL/Python (Python Programlama Dili)
5. «SELECT * FROM pg_pltemplate;»

PL/pgSQL Avantajları:

1. Fonksiyonlar ve trigger'lar oluşturulabilir.
2. Döngüsel ve koşula bağlı işlem adımları daha kolay yapılabilir. (while, for, if)
3. Karmaşık sorgulamalar ve hesaplamalar yapılabilir.
4. Kullanıcının kendi amacına yönelik fonksiyon yazması sağlanabilir.
5. PL/pgSQL, SQL'in tüm veri tipi, operatör ve hazır fonksiyonlarını tanıyabilir ve kullanabilir.

PL/pgSQL Fonksiyon Dönüş Tipleri

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 6

1. PL/pgSQL tek bir değer döndürmek zorunda değildir.
2. Birden fazla dönüş yapılacaksa «output» anahtar sözlüğü kullanılır.
3. PL/pgSQL fonksiyonları basit tipte veri döndürebilecekleri gibi birleşik (composit) bir veri de döndürebilirler.
4. Ya da bir sonuç kümesinin (tablosunun) adresini gösteren bir işaretçi (pointer) döndürebilir.
5. Bütün fonksiyonlar da değer döndürmek zorunda değildir (return, return void).

PL/pgSQL Sintaksı

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 6

PostgreSQL'de fonksiyonel bir dilin kullanılabilmesi için «CREATE LANGUAGE plpgsql» komutu kullanılarak bu dil öncelikle oluşturulmalıdır.

```
CREATE FUNCTION fonksiyon_adi (parametre1 tipi,  
parametre2 tipi,..., [out] parametreN tipi )  
[RETURNS çıktının veri tipi] AS [$$] ['']  
DECLARE  
    tanımlamalar;  
BEGIN  
    komutlar;  
[RETURN] [çıktı değeri];..  
EXCEPTION  
    kural dışı durumlar;  
END;  
[$$] [''] LANGUAGE plpgsql;
```

Fonksiyonu çağırmak için;

```
SELECT fonksiyon_adi(parametreler);
```

Düşürmek için;

```
DROP FUNCTION fonksiyon_adi(parametreler);
```

PL/pgSQL RETURN ve Dönüş Çeşitleri

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 6

- PL/pgSQL fonksiyonundan çıkış için «RETURN 'ifade'» anahtar sözcüğü kullanılır.
- Eğer parametre döndürülmeyecekse sadece «RETURN» yazılır. Veya birden çok parametre döndürülecekse «OUTPUT» anahtar sözcüğü kullanılabilir.
- «OUTPUT» veya «RETURN VOID» olmayan fonksiyonlarda «RETURN» ifadesi olmak zorundadır. Yoksa çalışma sırasında hata ile karşılaşılır.

PL/pgSQL Statement ve Loop Sintaksı

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 6

Değişken Tanımı:

- user_id integer;
- quantity numeric(5);
- url varchar;
- my_var
tablename.columnname
%TYPE;

«IF» Koşulu Tanımı:

IF koşul THEN yapılacaklar;
[ELSEIF koşul THEN yapılacaklar;]
[ELSE yapılacaklar;]
END IF;

«CASE» Tanımı:

CASE secici
WHEN secici_kosulu1 THEN yapılacaklar1;
WHEN secici_kosulu2 THEN yapılacaklar2;
...
WHEN secici_kosuluN THEN yapılacaklarN;
[ELSE secici_kosulu(N+1)]
END;

«LOOP» Tanımı:

LOOP
yapılacaklar...
EXIT [WHEN dongu_kosulu];
END LOOP;

«FOR» Tanımı:

FOR sayaç IN [REVERSE]
alt_limit...üst_limit LOOP
yapılacaklar...
END LOOP;

«WHILE» Tanımı:

WHILE dongu_kosulu LOOP
yapılacaklar...
END LOOP;

PL/pgSQL Örnekleri

BLM5225
Veri Tabanı
Modelleme ve
Tasarımı
Hafta 6

1. Girdi olarak verilen 2 sayının toplamını bulan fonksiyonu yazınız ve (22,63) parametreleri için çalıştırınız.
 2. Adı verilen bir departmandaki çalışanların ortalama maaşını bulan bir fonksiyon yazınız.
 3. Departman tablosundaki minimum ve maksimum departman numarasını bulup min_deptno ve max_deptno değişkenlerine atan fonksiyonu yazınız.
 4. Numarası verilen departmandaki çalışanların sayısını bulun, çalışan sayısı 10'dan azsa departmandaki tüm çalışanların maaşına %5 zam yapın.
 5. İsmi verilen bir departmanda çalışanların ortalama maaşı, verilen bir değerden düşük ve o departmandaki kadın çalışanların maaşlarının toplamı verilen bir limitin üstündeyse, o departmanda 1'den fazla projede çalışanların maaşlarına yine verilen bir oranda zam yapan fonksiyonu yazınız.
- ```
SELECT kosullu_zam_yap('Research', 50000, 20000, 5);
```

## Sabırla Dinlediğiniz İçin Teşekkürler

BLM5225  
Veri Tabanı  
Modelleme ve  
Tasarımı  
Hafta 6



## PL/pgSQL Örnekleri Cevap 1

BLM5225  
Veri Tabanı  
Modelleme ve  
Tasarımı  
Hafta 6

```
-- RETURN ile çözüm
CREATE FUNCTION toplama(sayi1 integer, sayi2 integer) RETURNS integer AS '
DECLARE
 toplam integer;
BEGIN
 toplam := sayi1 + sayi2;
 return toplam;
END;
' LANGUAGE 'plpgsql';

/*Çağırılması : */ SELECT toplama1(22,63);
/*Düşürülmesi: */ DROP FUNCTION toplama1(integer,integer);
```

```
-- OUT ile çözüm
CREATE FUNCTION toplama(sayi1 integer, sayi2 integer, OUT toplam integer) AS '
BEGIN
 toplam := sayi1 + sayi2;
END;
' LANGUAGE 'plpgsql';

SELECT toplama(22,63);
/*Çağırılması : */ SELECT toplama2(22,63);
/*Düşürülmesi: */ DROP FUNCTION toplama2(integer,integer,out integer); /*veya*/ DROP FUNCTION toplama1(integer,integer);
```

## PL/pgSQL Örnekleri Cevap 2

BLM5225  
Veri Tabanı  
Modelleme ve  
Tasarımı  
Hafta 6

```
CREATE FUNCTION dep_ort_maas(depname department.dname%TYPE, out ort_maas real) AS '
BEGIN
 SELECT AVG(salary) INTO ort_maas FROM employee e, department d WHERE e.dno = d.dnumber AND dname = depname;
END;
' LANGUAGE 'plpgsql';

/*Çağırılması : */ SELECT dep_ort_maas('Sales');
/*Düşürülmesi: */ DROP FUNCTION dep_ort_maas(department.dname%TYPE, out real); /*veya*/ DROP FUNCTION
dep_ort_maas(department.dname%TYPE);
```

## PL/pgSQL Örnekleri Cevap 3

BLM5225  
Veri Tabanı  
Modelleme ve  
Tasarımı  
Hafta 6

```
CREATE FUNCTION dep_no_bul(out min_deptno numeric, out max_deptno numeric) AS '
BEGIN
 SELECT MIN(dnumber),MAX(dnumber) INTO min_deptno, max_deptno FROM department;
END;
' LANGUAGE 'plpgsql';

/*Çağırılması : */ SELECT dep_no_bul();
/*Düşürülmesi: */ DROP FUNCTION dep_no_bul(out numeric, out numeric); /*veya*/ DROP FUNCTION dep_no_bul();
```

## PL/pgSQL Örnekleri Cevap 4

BLM5225  
Veri Tabanı  
Modelleme ve  
Tasarımı  
Hafta 6

```
CREATE FUNCTION dep6_emp(dnum numeric) RETURNS VOID AS '
DECLARE
 emp_sayac numeric;
BEGIN
 SELECT count(*) INTO emp_sayac FROM employee e WHERE e.dno = dnum;
 IF(emp_sayac < 10) THEN
 UPDATE employee e SET salary = salary*1.05 WHERE e.dno = dnum;
 END IF;
END;
' LANGUAGE 'plpgsql';

/*Çağırılması : */ SELECT dep6_emp(6);
/*Düşürülmesi: */ DROP FUNCTION dep6_emp(numeric);
```

## PL/pgSQL Örnekleri Cevap 5

BLM5225  
Veri Tabanı  
Modelleme ve  
Tasarımı  
Hafta 6

```
CREATE OR REPLACE FUNCTION kosullu_zam_yap(bolum_ismi department.dname%TYPE, ort_maas real, f_top_maas employee.salary%TYPE, zam_orani
integer) RETURNS VOID AS '
DECLARE
 ger_ort_maas real;
 kadin_maaslari integer;
 bolum_no department.dnumber%TYPE;
BEGIN
 SELECT dnumber INTO bolum_no FROM department WHERE dname = bolum_ismi;
 SELECT AVG(salary) INTO ger_ort_maas FROM employee WHERE dno = bolum_no;
 SELECT SUM(salary) INTO kadin_maaslari FROM employee WHERE dno = bolum_no AND sex = 'F';
 IF ger_ort_maas < ort_maas AND kadin_maaslari > f_top_maas THEN
 UPDATE employee SET salary = salary*zam_orani/100 + salary WHERE ssn IN (SELECT essn FROM employee, works_on WHERE
 ssn = essn AND dno = bolum_no GROUP BY essn HAVING COUNT(*) > 1);
 END IF;
END;
' LANGUAGE plpgsql;
```



# Veri Tabanı Modelleme ve Tasarımı BLM5225



Öğr. Grv. Furkan Çakmak

## Ders Tanıtım Formu ve Konular

BLM5225  
Veri Tabanı  
Modelleme ve  
Tasarımı  
Hafta 7

| Hafta | Tarih      | Konular                                                                          |
|-------|------------|----------------------------------------------------------------------------------|
| 1     | 27.02.2023 | Ders İçeriği Tanıtımı ve Kavram Açıklamaları                                     |
| 2     | 6.03.2023  | SQL'e Giriş; DDL ve DML Komutlarına Giriş, company-db'nin Tanıtımı               |
| 3     | 13.03.2023 | PostgreSQL Ortamının Tanıtımı ve Sorgulama Örnekleri                             |
| 4     | 20.03.2023 | Tablolarda Kısıt, View ve Sequence İşlemleri; Union, Intersect, Except İşlemleri |
| 5     | 27.03.2023 | Tablolarda Gruplama ve Sıralama Fonksiyonları                                    |
| 6     | 3.04.2023  | PL/pgSQL Fonksiyon Tanıtımı                                                      |
| 7     | 10.04.2023 | PL/pgSQL Alias, Record/Cursor ve Trigger Tanımları                               |
| 8     | 17.04.2023 | JDBC ile Veri Tabanına Bağlanıp Sorgu Yapma Uygulamaları                         |
| 9     | 24.04.2023 | Ara Sınav - 1                                                                    |
| 10    | 1.05.2023  | Tatil - İşçi Bayramı                                                             |
| 11    | 8.05.2023  | Hibernate Uygulamaları                                                           |
| 12    | 15.05.2023 | Xpath ve Xquery Yapısı                                                           |
| 13    | 22.05.2023 | Proje Gösterimleri - 1                                                           |
| 14    | 29.05.2023 | Proje Gösterimleri - 2                                                           |

Öğr. Grv. Furkan ÇAKMAK



## PL/pgSQL Record ve Cursor Tanımı

BLM5225  
Veri Tabanı  
Modelleme ve  
Tasarımı  
Hafta 7

1. Geçtiğimiz hafta bahsettiğimiz gibi, PL/pgSQL fonksiyonlarını sadece tek bir değer döndürmek zorunda değildi. Karmaşık sonuçları veya bir tabloyu da döndürebileceğimizden bahsetmiştik.
2. Bu tarz composit veri tiplerini döndürmek için RECORD tanımları kullanılmaktadır.
3. Eğer bir tablo döndürmek istiyorsak CURSOR tanımlayarak sonucu bunun üzerinden döndürebiliriz.

### RECORD Tanımı:

CREATE TYPE my\_record\_type AS (field1 type1, field2 type2, ...);

Ör: CREATE TYPE emp\_kisa\_bilgi AS (fname varchar(15), salary integer);

### CURSOR Tanımı:

cursor\_name [ [ NO ] SCROLL ] CURSOR [ ( arguments ) ] FOR sql\_query;

Ör: my\_cur CURSOR FOR select \* from employee;

### CURSOR kullanımı:

OPEN my\_cursor;

FETCH [ direction { FROM | IN } ] cursor  
INTO target;

Target: record ya da variable1, variable2,...  
şeklinde olabilir.

MOVE [ direction { FROM | IN } ] cursor;

CLOSE my\_cursor;

## PL/pgSQL RAISE Tanımı

BLM5225  
Veri Tabanı  
Modelleme ve  
Tasarımı  
Hafta 7

### RAISE Tanımı: (Ekran mesaj yazdırmak için kullanılır.)

RAISE [ level ] 'format' [, expression [, ...]] [ USING option = expression [, ... ] ];

RAISE [ level ] condition\_name [ USING option = expression [, ... ] ];

RAISE [ level ] SQLSTATE 'sqlstate' [ USING option = expression [, ... ] ];

RAISE [ level ] USING option = expression [, ... ];

RAISE ;

Ör: RAISE NOTICE 'Salary here is %', sal\_variable;

### "Level" seçeneği, ne tür mesaj yazdırmak istediğimizi söyler:

- DEBUG,
- LOG,
- INFO,
- NOTICE,
- WARNING,
- EXCEPTION
- Default'u EXCEPTION'dir.

## PL/pgSQL Record ve Cursor Örnekleri 1

BLM5225  
Veri Tabanı  
Modelleme ve  
Tasarımı  
Hafta 7

**Örnek 1:** SSN'i parametre olarak verilen çalışanın ismini, çalıştığı departmanın ismini ve maaşını ekrana yazdıran PL/pgSQL bloğunu yazın. Bir ssn vererek fonksiyonu çağırınız.

```
CREATE TYPE record1 AS (isim varchar(15), bolum varchar(25), maas integer);
CREATE OR REPLACE FUNCTION calisan_bilgiler(emp_ssn employee.ssn%TYPE) RETURNS record1 AS '
DECLARE
 emp_record record1;
BEGIN
 SELECT fname, dname, salary INTO emp_record FROM employee e, department d WHERE e.dno = d.dnumber AND
 e.ssn = emp_ssn;
 RAISE NOTICE "Calisan ismi: %, bolum ismi: %, maas: %", emp_record.isim, emp_record.bolum, emp_record.maas;
 RETURN emp_record;
END;
' LANGUAGE 'plpgsql';
/*Çağırılması: */ SELECT calisan_bilgiler('123456789');
/*Düşürülmesi: */ DROP FUNCTION calisan_bilgiler(employee.ssn%type); DROP TYPE record1;
```

## PL/pgSQL Record ve Cursor Örnekleri 2

BLM5225  
Veri Tabanı  
Modelleme ve  
Tasarımı  
Hafta 7

**Örnek 2:** Numarası verilen bir departmandaki çalışanların isimlerini bulan bir fonksiyon yazınız. Bir departman numarası vererek fonksiyonu çağırınız.

```
CREATE OR REPLACE FUNCTION bolum_calisanlari(dnum numeric) RETURNS void AS '
DECLARE
 emp_cursor CURSOR FOR SELECT fname, lname FROM employee WHERE dno = dnum;
BEGIN
 FOR emp_record IN emp_cursor LOOP
 RAISE INFO "Calisan ismi: % %", emp_record.fname, emp_record.lname;
 END LOOP;
END;
' LANGUAGE 'plpgsql';
/*Çağırılması: */ SELECT bolum_calisanlari(6);
/*Düşürülmesi: */ DROP FUNCTION bolum_calisanlari(numeric);
```

## PL/pgSQL Record ve Cursor Örnekleri 3

BLM5225  
Veri Tabanı  
Modelleme ve  
Tasarımı  
Hafta 7

**Örnek 3:** Numarası verilen bir projede çalışanların maaşları verilen bir değere tam bölünebiliyorsa, o kişilerin ad, soyad ve maaş bilgilerini HAVING fonksiyonu kullanmadan listeleyen fonksiyonu yazınız.

```
CREATE TYPE calisan AS (isim varchar(15), soyisim varchar(15), maas integer);
CREATE OR REPLACE FUNCTION calisan_listele(pnum project.pnumber%TYPE, bolen integer) RETURNS calisan[] AS '
DECLARE
 emp_cursor CURSOR FOR SELECT fname, lname, salary FROM employee, works_on WHERE ssn = essn AND pno = pnum;
 cal calisan[];
 i integer;
BEGIN
 i := 1;
 FOR emp_record IN emp_cursor LOOP
 IF emp_record.salary % bolen = 0 THEN
 cal[i] = emp_record;
 i := i + 1;
 END IF;
 END LOOP;
 RETURN cal;
END;
LANGUAGE 'plpgsql';
/*Çağırılması: */ SELECT calisan_listele('61',9);
/*Düşürülmesi: */ DROP FUNCTION calisan_listele(project.pnumber%TYPE, integer);
```

## PL/pgSQL Record ve Cursor Örnekleri 4

BLM5225  
Veri Tabanı  
Modelleme ve  
Tasarımı  
Hafta 7

**Örnek 4:** Departman numarası verilen bir departmandaki çalışanların toplam maaşını (SUM() fonksiyonundan yararlanmadan) bulan bir fonksiyon yazınız. (SELECT sum(salary) FROM employee WHERE dno = X;)

```
CREATE OR REPLACE FUNCTION dep_sum_salary(dnum numeric, OUT sum_sal numeric) AS '
DECLARE
 emp_cursor CURSOR FOR SELECT salary FROM employee WHERE dno = dnum;
BEGIN
 sum_sal := 0;
 FOR emp_record IN emp_cursor LOOP
 sum_sal := sum_sal + emp_record.salary;
 END LOOP;
END;
LANGUAGE 'plpgsql';
/*Çağırılması: */ SELECT dep_sum_salary(6);
/*Düşürülmesi: */ DROP FUNCTION dep_sum_salary(numeric);
```

## PL/pgSQL Trigger Tanımı

BLM5225  
Veri Tabanı  
Modelleme ve  
Tasarımı  
Hafta 7

1. Fonksiyonlar gibi veri tabanına kaydedilirler.
2. VTYS tarafından trigger'ın şartları oluştuğunda otomatik olarak çağırılırlar.
3. DML komutları trigger'ı başlatır. (INSERT, UPDATE, DELETE, vb.)

```
CREATE TRIGGER trigger_isim { BEFORE | AFTER }
{ event1 [OR event2 OR ...] } ON tablo_adi
[FOR [EACH] { ROW | STATEMENT }] [WHEN (condition)]
EXECUTE PROCEDURE trigger_fonk_adi(arguments);
```

```
CREATE OR REPLACE FUNCTION trig_fonk()
RETURNS TRIGGER AS '
BEGIN
Statements;
[RETURN [NULL | OLD | NEW];]
END;
' LANGUAGE 'plpgsql';
```

## PL/pgSQL Trigger Tanımı (Devam)

BLM5225  
Veri Tabanı  
Modelleme ve  
Tasarımı  
Hafta 7

| Part                                                                                                                                                                                                                                                                                           | Description                     | Possible Values          |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|--------------------------|
| Trigger timing                                                                                                                                                                                                                                                                                 | Trigger'ın harekete geçtiği an  | Before / After           |
| Trigger event                                                                                                                                                                                                                                                                                  | Trigger'ı tetikleyen DML        | Insert / Update / Delete |
| Trigger type                                                                                                                                                                                                                                                                                   | Trigger body'nin çalışma sayısı | Statement / Row          |
| <b>Trigger tipi</b> , trigger fonksiyonunun, bir SQL sorgusu için <b>sadece bir kez</b> mi, yoksa trigger olayından etkilenen <b>her bir satır</b> için mi çalışacağını belirler. Varsayılanı "FOR EACH STATEMENT"tır.                                                                         |                                 |                          |
| <b>NEW</b> : Tetikleyici prosedürün/fonksiyonun body bloğunda kullanılır. Row-level tetikleyiciler için insert/update olaylarında yeni eklenen satırın değerini tutan record yapısındaki değişkendir. Statement-level tetikleyicilerde ve Delete işlemlerinde NEW değişkeni NULL'dir.          |                                 |                          |
| <b>OLD</b> : Tetikleyici prosedürün/fonksiyonun body bloğunda kullanılır. Row-level tetikleyiciler için update/delete olaylarında, değişen/silinen eski satırın değerini tutan record yapısındaki değişkendir. Statement-level tetikleyicilerde ve Insert işlemlerinde OLD değişkeni NULL'dir. |                                 |                          |
| Trigger düşürülmesi: <b>DROP TRIGGER trigger_fonk_adi ON tablo_adi [ CASCADE   RESTRICT ]</b>                                                                                                                                                                                                  |                                 |                          |
| <b>CASCADE</b> : Tetikleyiciye bağlı olan nesneleri de otomatik olarak düşürür.                                                                                                                                                                                                                |                                 |                          |
| <b>RESTRICT</b> : Eğer tetikleyiciye bağlı nesneler varsa tetikleyici düşürülmez. Varsayılanı budur.                                                                                                                                                                                           |                                 |                          |



## PL/pgSQL Trigger Örnekleri 1

BLM5225  
Veri Tabanı  
Modelleme ve  
Tasarımı  
Hafta 7

**Örnek 1:** Sadece tatil günleri dışında ve mesai saatleri içinde employee tablosuna insert yapılmasına izin veren trigger'ı yazınız.

```
CREATE OR REPLACE FUNCTION trigger_fonkiyonu_1() RETURNS TRIGGER AS '
BEGIN
 IF (to_char(NOW(), "DY") IN ("SAT", "SUN") OR to_char(NOW(), "HH24") NOT BETWEEN "08" AND "18") THEN
 RAISE EXCEPTION "Sadece mesai gunlerinde ve mesai saatlerinde insert yapabilirsiniz.";
 RETURN NULL;
 ELSE
 RETURN NEW;
 END IF;
END;
' LANGUAGE 'plpgsql';
CREATE TRIGGER t_orn1 BEFORE INSERT ON employee FOR EACH ROW EXECUTE PROCEDURE trigger_fonkiyonu_1();
/*Düşürülmesi: Önce:*/ DROP TRIGGER t_orn1 on employee; /*Sonra:*/ DROP FUNCTION trigger_fonkiyonu_1();
/*Tetiklenmesi:*/ INSERT INTO employee VALUES('Vladimir', 'S', 'Putin', '666666666', '1952-10-07', '8975 Suriye', 'M', '125000', '333445555', '5');
```

## PL/pgSQL Trigger Örnekleri 2

BLM5225  
Veri Tabanı  
Modelleme ve  
Tasarımı  
Hafta 7

**Örnek 2:** Proje tablosunda pnumber kolonu değişince works\_on tablosunda da pno'nun aynı şekilde değişmesini sağlayan trigger'ı yazınız. (Öncelikle works\_on tablosundan pno'nun yabancı anahtar olma kısıtını kaldırınız.)

```
CREATE OR REPLACE FUNCTION trigger_fonkiyonu_2() RETURNS TRIGGER AS '
BEGIN
 UPDATE works_on SET pno = NEW.pnumber WHERE pno = OLD.pnumber;
 RETURN NEW;
END;
' LANGUAGE plpgsql;

CREATE TRIGGER t_orn2 AFTER UPDATE ON project FOR EACH ROW EXECUTE PROCEDURE trigger_fonkiyonu_2();
/*Düşürülmesi: Önce:*/ DROP TRIGGER t_orn2 on department; /*Sonra:*/ DROP FUNCTION trigger_fonkiyonu_2();
/*Tetiklenmesi:*/ UPDATE project SET pnumber = 999 WHERE pnumber = 61;
```



## PL/pgSQL Trigger Örnekleri 3

BLM5225  
Veri Tabanı  
Modelleme ve  
Tasarımı  
Hafta 7

**Örnek 3:** Maaş inişine ve %10'dan fazla maaş artışına izin vermeyen trigger'ı yazınız.

```
CREATE OR REPLACE FUNCTION trigger_fonkiyonu_3() RETURNS TRIGGER AS $
BEGIN
 IF (OLD.salary > NEW.salary OR NEW.salary > 1.1*OLD.salary) THEN
 RAISE EXCEPTION "Maasi dusuremezsiniz ve %10'dan fazla zam yapamazsiniz.";
 RETURN OLD;
 ELSE
 RETURN NEW;
 END IF;
END;
$ LANGUAGE 'plpgsql';
CREATE TRIGGER t_orn3 BEFORE UPDATE ON employee FOR EACH ROW EXECUTE PROCEDURE trigger_fonkiyonu_3();
/*Düşürülmesi: Önce:*/ DROP TRIGGER t_orn3 on employee; /*Sonra:*/ DROP FUNCTION trigger_fonkiyonu_3();
/*Tetiklenmesi*/ UPDATE employee SET salary = salary*1.12;
```

## PL/pgSQL Trigger Örnekleri 4

BLM5225  
Veri Tabanı  
Modelleme ve  
Tasarımı  
Hafta 7

**Örnek 4:** Departman tablonuza salary ile aynı tipte total\_salary kolonu ekleyin. Employee tablosunda bir işçinin maaşında maaş değişikliği olduğunda departman tablonuzdaki total\_salary kolonunda da gerekli güncellemeyi yapacak trigger'ı yazınız.

```
ALTER TABLE department ADD COLUMN total_salary; UPDATE department SET total_salary = (SELECT SUM(salary) FROM employee WHERE dno = dnumber);
CREATE OR REPLACE FUNCTION trigger_fonkiyonu_4() RETURNS TRIGGER AS $$
BEGIN
 IF (TG_OP = 'DELETE') THEN
 UPDATE department SET total_salary = total_salary - OLD.salary WHERE dnumber = OLD.dno;
 ELSEIF (TG_OP = 'UPDATE') THEN
 UPDATE department SET total_salary = total_salary - OLD.salary + NEW.salary WHERE dnumber = OLD.dno;
 ELSE
 UPDATE department SET total_salary = total_salary + NEW.salary WHERE dnumber = NEW.dno;
 END IF;
 RETURN NEW;
END;
$$ LANGUAGE 'plpgsql';
CREATE TRIGGER t_orn4 AFTER INSERT OR UPDATE OR DELETE ON employee FOR EACH ROW EXECUTE PROCEDURE trigger_fonkiyonu_4();
/*Tetiklenmesi 1*/ INSERT INTO employee VALUES('Vladimir', 'S', 'Putin', '666666667', '1952-10-07', '8975 Suriye', 'M', '100000000', '333445555', '1');
/*Tetiklenmesi 2,3*/ UPDATE employee SET salary = salary*1.07 WHERE dno = 1; DELETE FROM employee WHERE ssn = '1111111103';
```

# Sabırla Dinlediğiniz İçin Teşekkürler

BLM5225  
Veri Tabanı  
Modelleme ve  
Tasarımı  
Hafta 7

