

Veri Tabanı Dersi 3. Laboratuvarı

Grup 2 (UK)

Arş. Gör. Furkan Çakmak

1

Laboratuvar Programı

VT 20191
Lab 3

- Hafta 1 - SQL'e giriş; DDL ve DML komutlarına giriş
- Hafta 2 - Postgresql ortamının tanıtımı, Company-db'nin tanıtımı ve Sorgulama örnekleri
- Hafta 3 - Tablolarda Kısıt, View ve Sequence İşlemleri; Union, Intersect, Except İşlemleri
- Hafta 4 - Quiz 1
- Hafta 5 - Tablolarda Gruplama ve Sıralama Fonksiyonları
- Hafta 6 - JDBC ile Veri Tabanına Bağlanıp Sorgu Yapma Uygulamaları
- Hafta 7 - PL/pgSQL Fonksiyon Tanımı
- Hafta 8 - Quiz 2
- Hafta 9 - PL/pgSQL Alias, Record/Cursor ve Trigger Tanımları
- Hafta 10 - Xquery Yapısı ve Örnekleri
- Hafta 11 - Quiz 3

2

Alt Başlıklar

VT 20191
Lab 3

- «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

3

«TEAM» Tablosunun Oluşturulması

VT 20191
Lab 3

- 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)  
);
```

4

«TEAM_EMPLOYEE» Tablosunun Oluşturulması

VT 20191
Lab 3

- «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 «SSN»'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)  
);
```

5

«View» Tanımı ve Oluşturulması

VT 20191
Lab 3

- 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;

6

«Sequence» Tanımı ve Oluşturulması

VT 20191
Lab 3

- 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');

7

Tablolarda UNION, INTERSECT, EXCEPT İşlemleri

VT 20191
Lab 3

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.

8

Sabırla Dinlediğiniz İçin Teşekkürler

VT 20191
Lab 3



9

UNION, INTERSECT, EXCEPT Sorguları Cevapları

VT 20191
Lab 3

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;
```

10