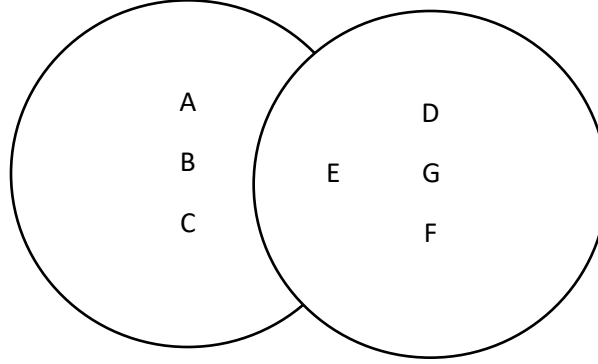


## BTEP104 – Lab Oturumu – 16-Nisan-2020

### Küme Operatörleri



Küme1={A,B,C,E}

Küme2={D,G,F,E}

**Birleşim Operatörü** -> Küme1UKüme2 = {A,B,C,E,D,G,F} = Küme2UKüme1 (?) ✓

**Kesişim Operatörü**-> Küme1 ∩ Küme2= {E} = Küme2 ∩ Küme1 (?) ✓

(Her iki kümenin ortak eleman/elemanları)

**Fark Operatörü** -> Küme1 – Küme2 = Küme2 – Küme1 (?) X

Küme1 – Küme2 = {A,B,C} , Küme2-Küme1={D,F,G}

Oracle-SQL Dilinde, Tablolar üzerinde Küme operasyonları uygulanabilir.

#### **Yazım Kuralı:**

Select  
From  
[Where] }  
Küme Operatörü  
Select  
From  
[Where] }  
Küme Operatörü  
Select  
From  
[Where] }

#### **ORDER BY**

.... Maximum 256 sorgu küme operatörleri kullanılarak yeni sorgu elde edilebilir.

UNION (Birleşim) : Kullanılan sorguların sonuçlarını birleştirir, ve bu işlemi yaparken, tekrar eden satırları sadece bir kere görüntüler, ayrıca ilk sorgunun ilk sütununa göre A—Z'ye sıralama yapar.

UNION ALL: Kullanılan sorguların sonuçlarını birleştirir, tekrar eden satırları gösterir ayrıca sıralama yapmaz.

INTERSECT (Kesişim): Kullanılan sorguların sonuçlarında geçen ortak satırları görüntüler, ve bu işlemi yaparken, tekrar eden satırları sadece bir kere görüntüler, ayrıca ilk sorgunun ilk sütununa göre A—Z'ye sıralama yapar.

MINUS (Fark): Kullanılan sorgulardan ilk sorgunun sonucunda olan ancak ikinci sorgunun sonucunda olmayan tüm satırları görüntüler, ve bu işlemi yaparken, tekrar eden satırları sadece bir kere görüntüler, ayrıca ilk sorgunun ilk sütununa göre A—Z'ye sıralama yapar.

### Örnekler:

```
select last_name from employees;
```

```
select last_name from certified_emps;
```

```
select last_name from employees
UNION
select last_name from certified_emps;
```

Yukarıdaki sorgunun sonucunda tekrar eden satırların bir kere görüntülenmesi bekleniyordu.

Ancak sorgu sonucunu gözlemlediğimizde tekrar eden satırların kaldırılmadığını farkettilik.

Her iki tablonun yapısına bakarak veri tiplerinin farklı olduğunu gördük

EMPLOYEES - last\_name varchar2(25)

CERTIFIED\_EMPS – last\_name char(25)

Veri tiplerinin farklı olmasından kaynaklı olarak tekrar eden satırlar sorgu sonucundan çıkarılmadı.

Varchar2 – karakter olan veriler için kullanılır. Saklanacak veriye göre Oracle sadece gereksiniz duyulan karakter sayısını kullanır, gerisini ise kullanmaz.

TabloX – isim varchar2(10) → A L I ~~-----~~

TabloY – isim char(10) → A L I b b b b b b (b=boşluk karakteri)

A L I (varchar2) = A L I b b b b b b (char)

length(isim) = length(isim)

3 ~~×~~ 10

varchar2 = rtrim(char)

3=3 ✓

**Yani Sorgumuz aşağıdaki gibi olmalı:**

```
select last_name from employees
UNION
select rtrim(last_name) from certified_emps;
```

ABEL=Abel (?) Harf/Karakter duyarlılığı olduğundan bu isimler aynı olsa dahi. Oracle tarafından aynı isimlermiş gibi algılanmazlar.

**Dolayısıyla aşağıdaki sorgu en doğru sorgu olacaktır.**

```
select upper(last_name) from employees
UNION
select rtrim(upper(last_name)) from certified_emps;
```

**UNION ALL kullanırsak eğer;**

```
select upper(last_name) from employees
UNION ALL
select rtrim(upper(last_name)) from certified_emps;
```

Sorgu sonucu tekrarları, sırasız bir biçimde görüntüleyecektir.

**INTERSECT operatörü:** Her iki sorgu sonucunda olan ORTAK satırları görüntüler...

```
select upper(last_name) from employees
INTERSECT
select rtrim(upper(last_name)) from certified_emps;
```

**MINUS operatörü:** employees tablosunda olup certified employees tablosunda olmayan satırları görüntüler.

```
select upper(last_name) from employees
MINUS
select rtrim(upper(last_name)) from certified_emps;
```

veya

```
select rtrim(upper(last_name)) from certified_emps
MINUS
select upper(last_name) from employees;
```

### Küme Operatörleri ile yapılmaması gereken işlemler:

1. Tüm satırlar (\*) işaretçiği ile seçilip sorguda kullanılmamalıdır. Tablolardaki sütunların birbiri ile eşleşmeme durumu yüksek bir olasılık taşır. Her iki sorguda kullanılan tablolardaki sütun sayısı farklı olabileceği gibi, aynı olsa dahi sütunlar farklı veri tiplerine sahip olabilir.

```
select * from employees
Union
select * form certified_emps;
```

2. Order by cümlecğiği mutlaka sorgunun en sonunda yer almalıdır.

```
select upper(first_name) from employees
order by first_name (yanlış yerde)
Union
select upper(first_name) form certified_emps;
```

```
select upper(first_name) from employees
Union
select upper(first_name) form certified_emps
order by first_name; (hata)
```

```
select upper(first_name),upper(last_name)
from employees
Union
select upper(first_name), upper(rtrim(last_name))
from certified_emps
order by 1; (first name sütununa göre A-Z'ye sıraladı)
```

```
select upper(first_name),upper(last_name)
from employees
Union
select upper(first_name), upper(rtrim(last_name))
from certified_emps
order by 2 desc; (last name sütununa göre Z-A'ya sıraladı)
```

3. Tüm sorgulardaki SELECT cümlesinde kullanılacak olan `sutun_adi/ifade` sayısı birbirine EŞİT olmalı. Ayrıca, tüm sorgulardaki sütunların birbiri ile aynı veya uyumlu veri tipine sahip olması gerekmektedir.

```
select upper(first_name),upper(last_name) from employees  
Union  
select upper(first_name) from certified_emps;
```

```
select upper(first_name),hire_date from employees  
Union  
select upper(first_name),cert_type from certified_emps;
```

yukarıdaki sorguyu aşağıda görüldüğü üzere düzelttik. Tarih verisi içeren `hire_date` sütununu `to_char` fonksiyonunu kullanarak karakter formatına dönüştürdük.

```
select upper(first_name),to_char(hire_date,'fmMonth') from employees  
Union  
select upper(first_name),cert_type from certified_emps;
```