

Laboratuvar 3

KÜME OPERATÖRLERİ

Kartezyen Çarpımı

İki tablo içerisinde bir birleştirme koşulu yoksa, o zaman kartezyen çarpımı sözkonusudur. Bunun dışında birleştirme koşulu geçersiz ise ve birinci tablodaki tüm satırlar ile birleştirilemiyorsa, yine kartezyen çarpımından söz edilebilir.

```
SQL> SELECT Employee_Id, Department_Name  
       FROM Employees, Departments;
```

Yukarıdaki sorguda, FROM cümleciğinde 2 tablo kullanılmıştır. Daha sonra işleyeceğimiz JOIN konusunda da göreceğimiz gibi, eğer sorguda birden fazla tabloyu herhangi bir birleştirme işlemi yapmadan kullanırsak sonuç kartezyen çarpımı olur. Yani Employees tablosundaki satırlar ile Departments tablosundaki satırların çarpımı. Employees tablosunda 15 satır, Departments tablosunda da 5 satır olduğunu varsayarsak, sonuçta $15 \times 5 = 75$ satır olacaktır.

Küme Operatörleri ile Sorgu sonuçlarını birleştirme

UNION ve UNION ALL: Union, verilen koşul ifadesine uygun olarak, çift kayıtların gözardı edilerek her iki kümedeki tüm kayıtları seçer. Union All, çift kayıtları da dahil eden birleştirme işlemidir.

INTERSECT: Her iki koşulda da uygun varlıkları elde etmede kullanılır. Küme kesişimi olarak da algılanabilir.

MINUS: Birinci tabloda olan ikinci tabloda olmayan kayıtları ayıklamada kullanılmaktadır. Kümesel fark işlemi olarak düşünülebilir.

Aşağıdaki örnekleri inceleyelim.

```
SQL> desc employees
```

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(6)
FIRST_NAME		VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
EMAIL	NOT NULL	VARCHAR2(25)
PHONE_NUMBER		VARCHAR2(20)
HIRE_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR2(10)
SALARY		NUMBER(8,2)
COMMISSION_PCT		NUMBER(2,2)
MANAGER_ID		NUMBER(6)
DEPARTMENT_ID		NUMBER(4)

```
SQL> desc certified_emps
```

Name	Null?	Type
EMPLOYEE_ID		NUMBER(6)
FIRST_NAME		VARCHAR2(20)
LAST_NAME	NOT NULL	CHAR(25)
PHONE_NUMBER		VARCHAR2(20)
HIRE_DATE	NOT NULL	DATE
SALARY		NUMBER(8,2)
DEPARTMENT_ID		NUMBER(4)
CERT_TYPE		VARCHAR2(15)

Örnek 1:

```
SQL> select first_name  
2 from employees  
3 UNION  
4 select first_name  
5 from certified_emps;
```

Yukarıdaki sorguda employees ve certifie_emps tablolarındaki first name sütunlarının sonuçları UNION operatörü ile birleştirilir. Buna göre sorgu sonucunda tekrarlanan isimler sadece bir kere görüntüleniyor ve satırlar Adan Zye sıralanır.

Örnek 2:

```
SQL> select first_name  
2 from employees  
3 UNION ALL  
4 select first_name  
5 from certified_emps;
```

Yukarıdaki sorguda da tabloların first_name sütunları birleştirilir ancak UNION ALL operatörü kullanıldığı için tekrarlanan isimler görüntülenir ve sorgu sonucunda sıralama olmaz.

Örnek 3:

```
SQL> select first_name  
2 from employees  
3 INTERSECT  
4 select first_name  
5 from certified_emps;
```

Yukarıdaki sorguda ise employees ve certified_emps tablolarındaki first_name sütunundaki ortak isimler görüntülenir. Yani her employees tablosunda hem de certified_emps tablosunda ortak olan isimler listelenir.

Örnek 4:

```
SQL> select first_name  
2 from employees  
3 MINUS  
4 select first_name  
5 from certified_emps;
```

MINUS operatörünü kullanacak olursak, yukarıdaki sorgu sonucunda sadece employees tablosundaki isimler görüntülenecektir. Eğer employees tablosundaki isimler, certified_emps tablosunda varsa sorgu sonucundan çıkarılır.

Örnek 5:

```
SQL> select first_name  
2 from certified_emps  
3 MINUS  
4 select first_name  
5 from employees;
```

Son olarak, MINUS operatörünü kullanırken eğer sorguların yerini değiştirirsek, sorgu sonucu da değişecektir. Şöyle ki, sadece certified_emps tablosundaki isimler görüntülenecektir.