



VERİ TABANI VE YÖNETİM SİSTEMLERİ

Dr. Öğretim Üyesi

Alper Talha KARADENİZ





GENEL
DERS
İÇERİĞİ

1. Join

2. Primary Key-Foreign Key

3. Intersect

4. Except

5. Union

6. Select Top



JOIN (BİRLEŞTİRME)

Bazı veritabanları sadece bir tablodan oluşmaz. Bunun sebebi genellikle veri tekrarını önlemektir.

Bu şekilde oluşturulmuş veri tabanlarında tablolardaki değerleri birbirleriyle ilişkili bir şekilde kullanabilmek için **join** işlemi yapılır.

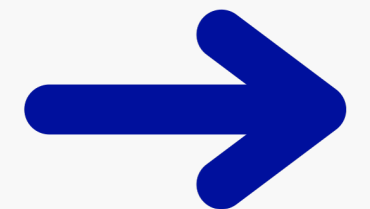
Dikkat edilmesi gereken nokta, tabloların Primary Key ve Foreign Key'ler ile birbirlerine bağlanmış olması gerektiğidir.



PRIMARY KEY



- Bir tabloda her satırı benzersiz şekilde tanımlayan sütun ya da sütunlar bütünüdür.
- NULL olamaz ve tekrar edemez.
- Genellikle ID (kimlik) gibi sütunlar primary key olarak seçilir
- Her tabloda en fazla bir tane primary key olabilir.
- Birden fazla sütundan oluşan bileşik (composite) primary key de tanımlanabilir.





PRIMARY KEY



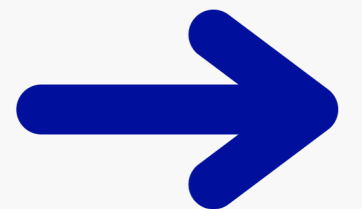
```
CREATE TABLE Ogrenciler (  
  ogrenci_id INT PRIMARY KEY,  
  ad VARCHAR(50),  
  soyad VARCHAR(50),  
  dogum_tarihi DATE  
);
```





FOREIGN KEY

- Bir tablodaki sütunun, başka bir tablonun primary key sütununa bağlantı (referans) oluşturmastır.
- Tablolar arasında ilişki kurmak için kullanılır.
- Referans aldığı değerin tablosunda mevcut olması gerekir.
- FOREIGN KEY, birden fazla olabilir.
- Veritabanında bütünlük sağlar.
- Ekleme, silme ve güncelleme işlemlerinde ilişkili kayıtları kontrol eder.





FOREIGN KEY



```
CREATE TABLE Ogrenciler (  
  ogrenci_id INT PRIMARY KEY,  
  ad VARCHAR(50)  
);  
CREATE TABLE Notlar (  
  not_id INT PRIMARY KEY,  
  ogrenci_id INT,  
  ders VARCHAR(50),  
  notu INT,  
  FOREIGN KEY (ogrenci_id)  
  REFERENCES Ogrenciler(ogrenci_id)  
);
```





JOIN (BİRLEŞTİRME)

Birleştirilen iki tablodan hangisinin baskın olacağı veya hangisinden hangi alanların okunacağı tercih edilen JOIN türüne göre değişkenlik arz eder.

Bu amaçla SQL Server aşağıdaki üç temel JOIN türlerini destekler.

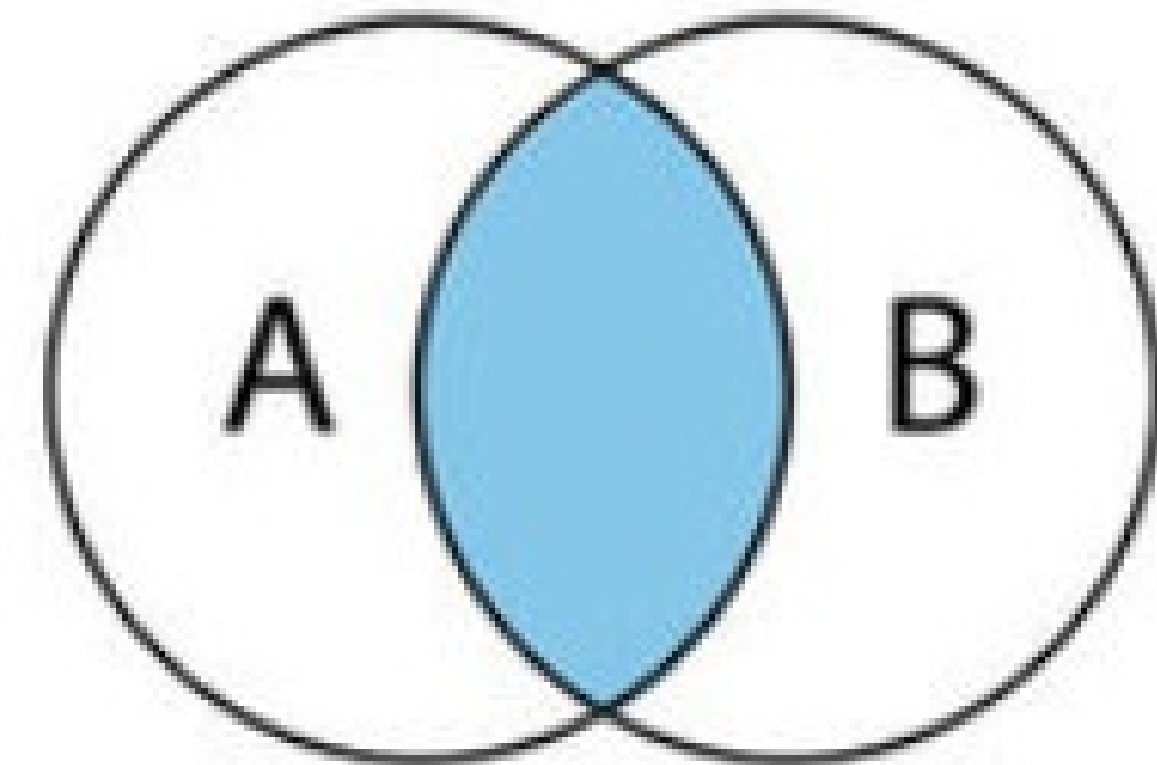
1. Inner Join
2. Outer Join
 - a. Left Outer Join
 - b. Right Outer Join
 - c. Full Outer Join
3. Cross Join



INNER JOIN

SQL Inner Join komutu kullanımı 2 tabloyu belirlenen alanlar üzerinden birleştirmek ve bu alanlarda eşleşen değerlere ait kayıtları sorgulama için kullanılır. Her iki tarafta bulunan ortak kolonlar sayesinde birleştirme işlemi yapılır. Birebir ilişkiler için tercih edilir. Yazım biçimi olarak INNER JOIN yazılabileceği gibi sadece JOIN de yazılabilir.

```
SELECT alan_adlari  
FROM tablo1  
INNER JOIN tablo2  
ON tablo1.kolonadi = tablo2.kolanadi
```



INNER JOIN

Results

Messages

	personel...	ADI	SOYADI	TELEFON	TCKIMLIKNO	departmanID	DOGUMTARIHI
1	3	Oğuzhan	TAŞ	02127777777	12345678912	1	1976-02-15
2	4	Sinan	Engin	02127778899	12345678913	2	2000-03-15
3	5	Erhan	Can	02128889900	12345678914	2	2001-04-20
4	6	Furkan	Namlı	03129998877	12345678915	2	2000-08-19
5	7	Ceydan	Şen	02128889988	12345678916	3	1985-03-15
6	8	Sinan	Kurban	021233344499	12345678917	1	2001-07-25
7	9	Faruk	Tınaz	02128886677	12345678918	5	2001-07-16
8	10	Hatice	Can	04241112233	12345678919	5	2014-07-16
9	11	Pakize	Şener	02128882200	12345678921	5	2001-10-16
10	12	Şaheser	Soydan	02128883322	12345678922	5	2001-11-16
11	13	Anıl	Can	02128889911	12345678923	5	2001-08-16
12	14	Anıl	Kurban	02126662211	12345678924	5	2003-07-16
13	15	Defne	Dertli	NULL	NULL	NULL	NULL

Results		Messages	
	departmanID	DEPARTMAN	
1	1	Yönetim	
2	2	Bilişim	
3	3	Muhasebe	
4	5	Pazarlama	
5	6	İletişim	
6	7	Destek	

```
SELECT p.ADI, p.SOYADI, d.DEPARTMAN
FROM tblpersonel p INNER JOIN tbldepartman d
ON p.departmanID = d.departmanID
```

Results		Messages	
	ADI	SOYADI	DEPARTMAN
1	Oğuzhan	TAŞ	Yönetim
2	Sinan	Engin	Bilişim
3	Erhan	Can	Bilişim
4	Furkan	Namlı	Bilişim
5	Ceydan	Şen	Muhasebe
6	Sinan	Kurban	Yönetim
7	Faruk	Tınaz	Pazarlama
8	Hatice	Can	Pazarlama
9	Pakize	Şener	Pazarlama
10	Şaheser	Soydan	Pazarlama
11	Anıl	Can	Pazarlama
12	Anıl	Kurban	Pazarlama



OUTER JOIN

Outer join iki tablodaki ilişkili olmayan kayıtları da döndürmemizi sağlar.

Üç farklı çeşidi vardır:

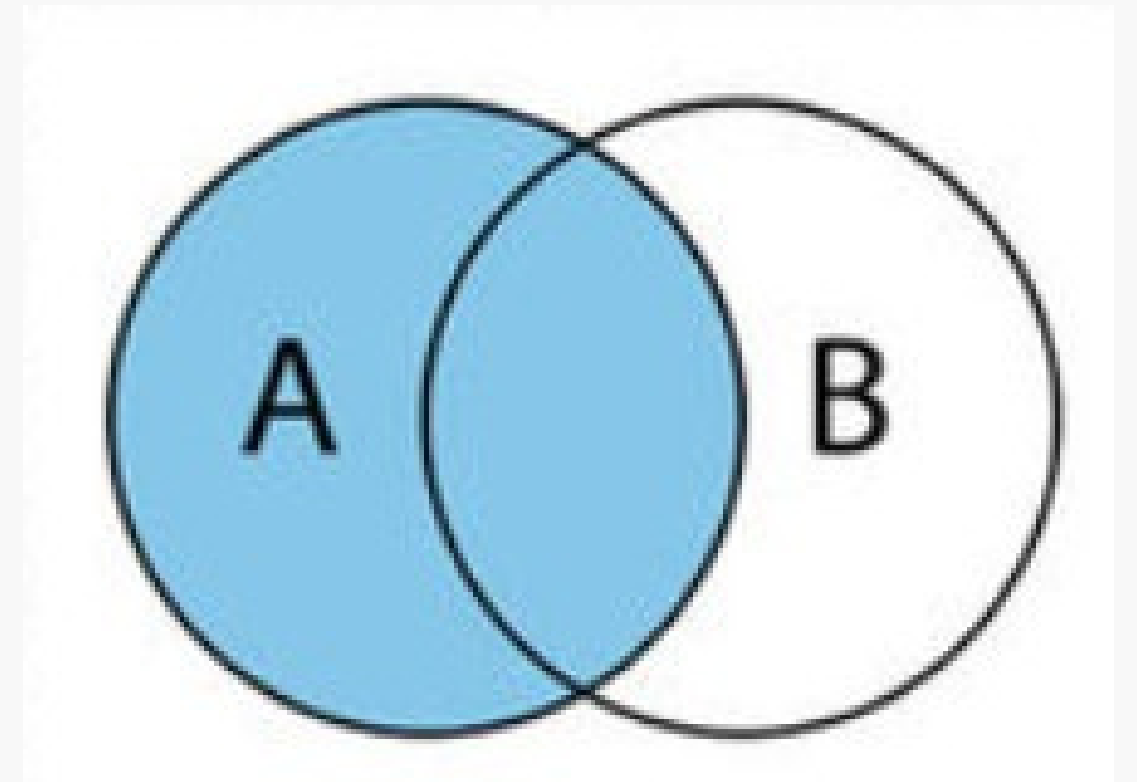
- 1) Left Outer Join
- 2) Right Outer Join
- 3) Full Outer Join



LEFT OUTER JOIN

Birinci tablodaki tüm kayıtlar getirilir. İkinci tabloda birinci tablo ile ortak bir kayıt varsa onlar da getirilir.

```
SELECT * FROM  
Tablo1 t1 LEFT OUTER Tablo2 t2  
ON  
t1.kolon1 = t2.kolon1
```



LEFT OUTER JOIN

Results		Messages					
	personel...	ADI	SOYADI	TELEFON	TCKIMLIKNO	departmanID	DOGUMTARIHI
1	3	Oğuzhan	TAŞ	0212777777	12345678912	1	1976-02-15
2	4	Sinan	Engin	02127778899	12345678913	2	2000-03-15
3	5	Erhan	Can	02128889900	12345678914	2	2001-04-20
4	6	Furkan	Namlı	03129998877	12345678915	2	2000-08-19
5	7	Ceydan	Şen	02128889988	12345678916	3	1985-03-15
6	8	Sinan	Kurban	021233344499	12345678917	1	2001-07-25
7	9	Faruk	Tınaz	02128886677	12345678918	5	2001-07-16
8	10	Hatice	Can	04241112233	12345678919	5	2014-07-16
9	11	Pakize	Şener	02128882200	12345678921	5	2001-10-16
10	12	Şaheser	Soydan	02128883322	12345678922	5	2001-11-16
11	13	Anıl	Can	02128889911	12345678923	5	2001-08-16
12	14	Anıl	Kurban	02126662211	12345678924	5	2003-07-16
13	15	Defne	Dertli	NULL	NULL	NULL	NULL

Results		Messages	
	departmanID	DEPARTMAN	
1	1	Yönetim	
2	2	Bilişim	
3	3	Muhasebe	
4	5	Pazarlama	
5	6	İletişim	
6	7	Destek	

```
SELECT p.ADI, p.SOYADI, d.DEPARTMAN
FROM tblpersonel p LEFT JOIN tbldepartman d
ON p.departmanID=d.departmanID
```

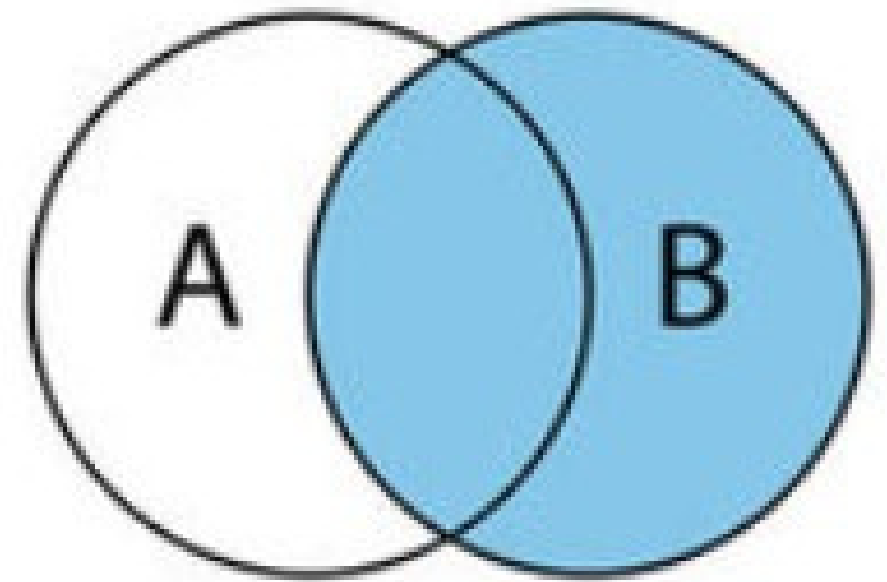
Results		Messages	
	ADI	SOYADI	DEPARTMAN
1	Oğuzhan	TAŞ	Yönetim
2	Sinan	Engin	Bilişim
3	Erhan	Can	Bilişim
4	Furkan	Namlı	Bilişim
5	Ceydan	Şen	Muhasebe
6	Sinan	Kurban	Yönetim
7	Faruk	Tınaz	Pazarlama
8	Hatice	Can	Pazarlama
9	Pakize	Şener	Pazarlama
10	Şaheser	Soydan	Pazarlama
11	Anıl	Can	Pazarlama
12	Anıl	Kurban	Pazarlama
13	Defne	Dertli	NULL



RIGHT OUTER JOIN

İkinci tablodaki tüm kayıtlar getirilir. Birinci tabloda ikinci tablo ile ortak bir kayıt varsa onlar da getirilir.

```
SELECT * FROM Tablo1 t1 RIGHT OUTER Tablo2 t2  
ON t1.kolon1 = t2.kolon1
```





RIGHT OUTER JOIN

Results

Messages

	personel...	ADI	SOYADI	TELEFON	TCKIMLIKNO	departmanID	DOGUMTARIHI
1	3	Oğuzhan	TAŞ	02127777777	12345678912	1	1976-02-15
2	4	Sinan	Engin	02127778899	12345678913	2	2000-03-15
3	5	Erhan	Can	02128889900	12345678914	2	2001-04-20
4	6	Furkan	Namlı	03129998877	12345678915	2	2000-08-19
5	7	Ceydan	Şen	02128889988	12345678916	3	1985-03-15
6	8	Sinan	Kurban	021233344499	12345678917	1	2001-07-25
7	9	Faruk	Tınaz	02128886677	12345678918	5	2001-07-16
8	10	Hatice	Can	04241112233	12345678919	5	2014-07-16
9	11	Pakize	Şener	02128882200	12345678921	5	2001-10-16
10	12	Şaheser	Soydan	02128883322	12345678922	5	2001-11-16
11	13	Anıl	Can	02128889911	12345678923	5	2001-08-16
12	14	Anıl	Kurban	02126662211	12345678924	5	2003-07-16
13	15	Defne	Dertli	NULL	NULL	NULL	NULL

Results		Messages	
	departmanID	DEPARTMAN	
1	1	Yönetim	
2	2	Bilişim	
3	3	Muhasebe	
4	5	Pazarlama	
5	6	İletişim	
6	7	Destek	

```
SELECT p.ADI, p.SOYADI, d.DEPARTMAN
FROM tblpersonel p RIGHT JOIN tbldepartman d
ON p.departmanID=d.departmanID
```

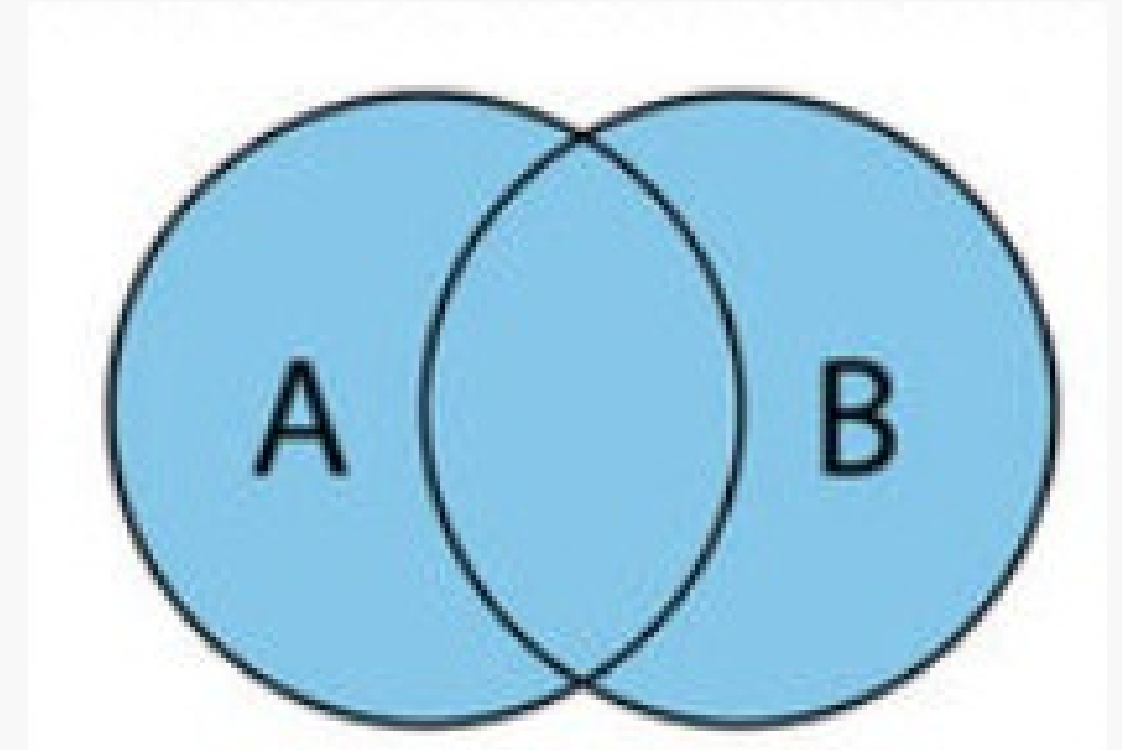
Results		Messages	
	ADI	SOYADI	DEPARTMAN
1	Oğuzhan	TAŞ	Yönetim
2	Sinan	Kurban	Yönetim
3	Sinan	Engin	Bilişim
4	Erhan	Can	Bilişim
5	Furkan	Namlı	Bilişim
6	Ceydan	Şen	Muhasebe
7	Faruk	Tınaz	Pazarlama
8	Hatice	Can	Pazarlama
9	Pakize	Şener	Pazarlama
10	Şaheser	Soydan	Pazarlama
11	Anıl	Can	Pazarlama
12	Anıl	Kurban	Pazarlama
13	NULL	NULL	İletişim
14	NULL	NULL	Destek



FULL OUTER JOIN

Full outer joinde iki tablodaki tüm kayıtlar uyuşsun ya da uyuşmasın getirilir.

```
SELECT * FROM  
Tablo1 t1 FULL OUTER Tablo2 t2  
ON t1.kolon1 = t2.kolon1
```



FULL OUTER JOIN

Results

Messages

	personel...	ADI	SOYADI	TELEFON	TCKIMLIKNO	departmanID	DOGUMTARIHI
1	3	Oğuzhan	TAŞ	02127777777	12345678912	1	1976-02-15
2	4	Sinan	Engin	02127778899	12345678913	2	2000-03-15
3	5	Erhan	Can	02128889900	12345678914	2	2001-04-20
4	6	Furkan	Namlı	03129998877	12345678915	2	2000-08-19
5	7	Ceydan	Şen	02128889988	12345678916	3	1985-03-15
6	8	Sinan	Kurban	021233344499	12345678917	1	2001-07-25
7	9	Faruk	Tınaz	02128886677	12345678918	5	2001-07-16
8	10	Hatice	Can	04241112233	12345678919	5	2014-07-16
9	11	Pakize	Şener	02128882200	12345678921	5	2001-10-16
10	12	Şaheser	Soydan	02128883322	12345678922	5	2001-11-16
11	13	Anıl	Can	02128889911	12345678923	5	2001-08-16
12	14	Anıl	Kurban	02126662211	12345678924	5	2003-07-16
13	15	Defne	Dertli	NULL	NULL	NULL	NULL

Results Messages

	departmanID	DEPARTMAN
1	1	Yönetim
2	2	Bilişim
3	3	Muhasebe
4	5	Pazarlama
5	6	İletişim
6	7	Destek

Results		Messages	
	ADI	SOYADI	DEPARTMAN
1	Oğuzhan	TAŞ	Yönetim
2	Sinan	Engin	Bilişim
3	Erhan	Can	Bilişim
4	Furkan	Namlı	Bilişim
5	Ceydan	Şen	Muhasebe
6	Sinan	Kurban	Yönetim
7	Faruk	Tınaz	Pazarlama
8	Hatice	Can	Pazarlama
9	Pakize	Şener	Pazarlama
10	Şaheser	Soydan	Pazarlama
11	Anıl	Can	Pazarlama
12	Anıl	Kurban	Pazarlama
13	Defne	Dertli	NULL
14	NULL	NULL	İletişim
15	NULL	NULL	Destek

```
SELECT p.ADI, p.SOYADI, d.DEPARTMAN
FROM tblpersonel p Full JOIN tbldepartman d
ON p.departmanID=d.departmanID
```

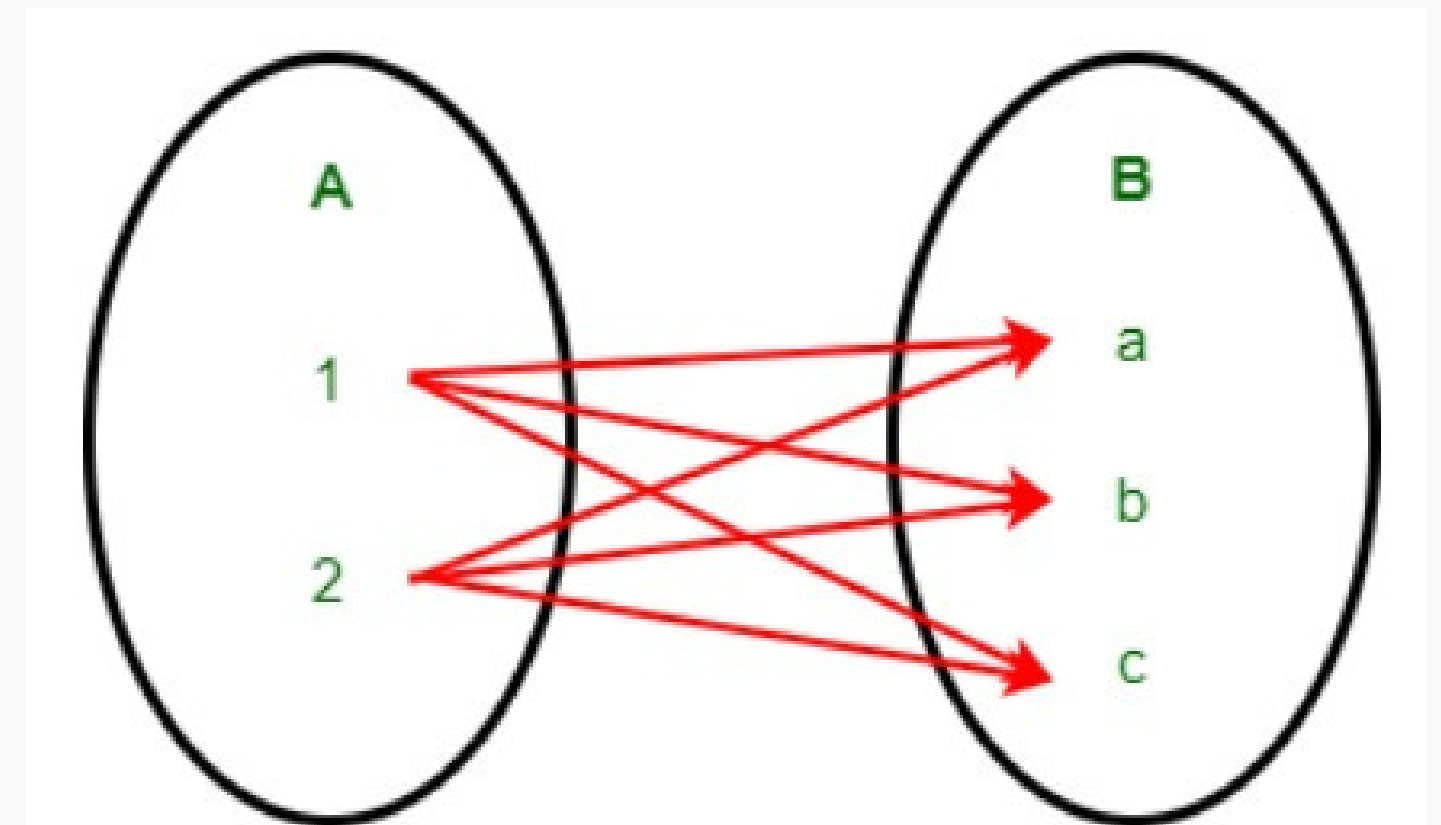


CROSS JOIN

CROSS JOIN, iki tabloyu birbiriyle her olasılık için eşleştirir. Yani bir tablodaki her satırı, diğer tablodaki her satırla eşleştirir. Bu teknik kartezyen çarpım olarak da bilinir.

- Eğer bir tabloda A kadar kayıt, diğerinde B kadar kayıt varsa,
- $A \times B = \text{Sonuç satır sayısı}$ olur.

```
SELECT O.ad, D.ders_adi  
FROM Ogrenciler O  
CROSS JOIN Dersler D;
```





CROSS JOIN

```
SELECT Car_model.Car_model, Color_name.Color_name  
FROM Car_model CROSS JOIN Color_name;
```

	Car_model
1	Camry
2	Corolla
3	Prius

	Color_name
1	Black
2	Red
3	Silver

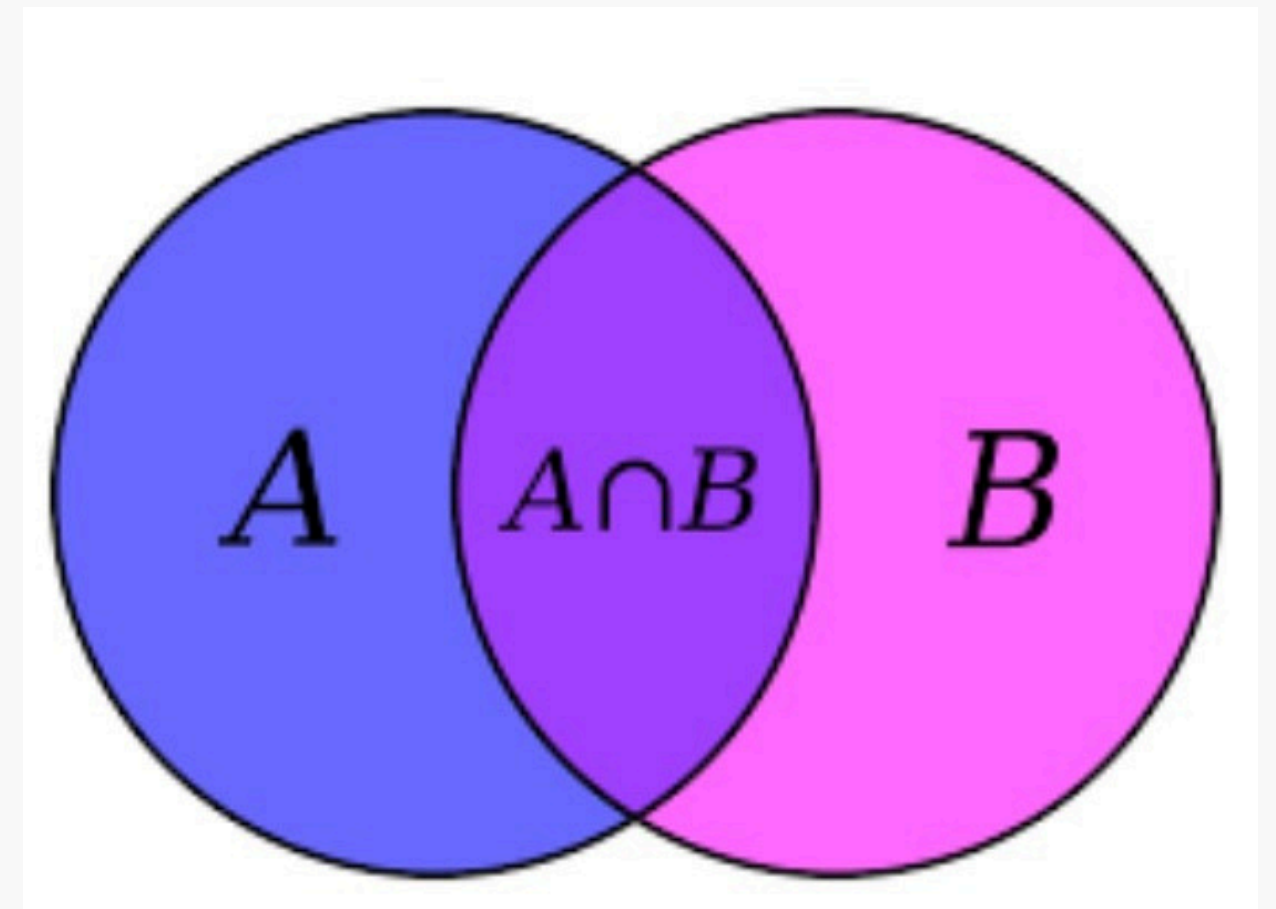
CROSS JOIN

	Car_model	Color_name
1	Camry	Black
2	Corolla	Black
3	Prius	Black
4	Camry	Red
5	Corolla	Red
6	Prius	Red
7	Camry	Silver
8	Corolla	Silver
9	Prius	Silver

INTERSECT İŞLEMİ

İki farklı sorgu sonucunun kesişimini elde etmek için kullanılır. Yani iki sorgu sonuç kümesinde de ortak olan verilerin gösterilmesi için kullanılır.
(MOR)

```
SELECT alan_adlari  
FROM tablo1  
INTERSECT  
SELECT alan_adlari  
FROM tablo2;
```

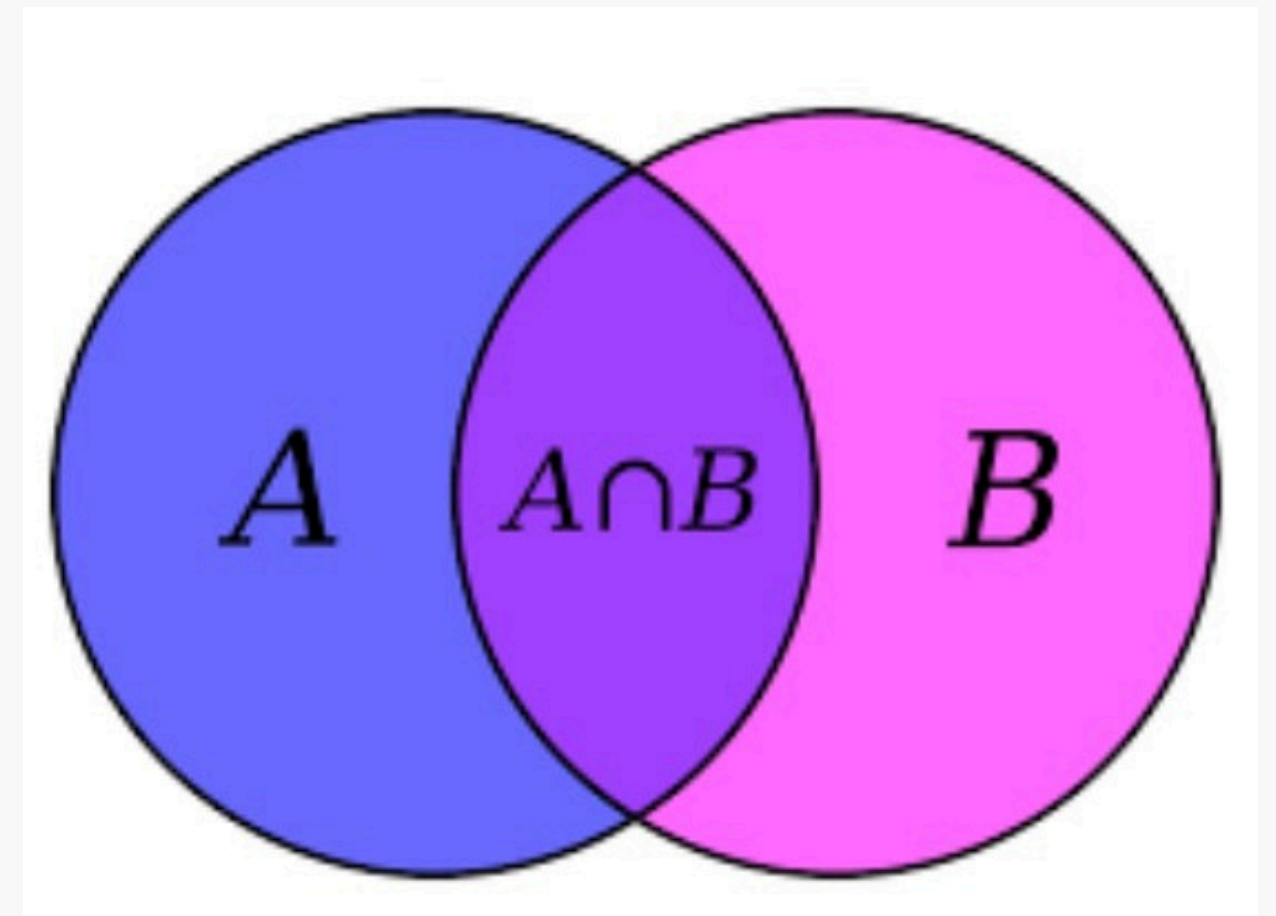




EXCEPT İŞLEMİ

İki farklı sorgu sonucunu karşılaştırır ve sadece ilk sonuç setinde olan, ikinci sorgu sonucunda olmayan kayıtları listelenmesini sağlar. (MAVİ)

```
SELECT alan_adlari  
FROM tablo1  
EXCEPT  
SELECT alan_adlari  
FROM tablo2;
```





UNION İŞLEMİ

- İki sorgunun sonuçlarını birleştirir.
- Aynı satırları (tekrarları) eleyerek sadece benzersiz satırları döner (DISTINCT gibi çalışır).

```
SELECT sütun1, sütun2  
FROM tablo1  
UNION SELECT  
sütun1, sütun2  
FROM tablo2;
```



UNION İŞLEMİ

Table: Teachers

id	name	age
1	Peter	32
2	Megan	43
3	Rose	29
4	Linda	30
5	Mary	41

Table: Students

id	name	age
1	Harry	23
2	Jack	45
3	Joe	32
4	Dent	23
5	Bruce	40

```
SELECT age  
FROM Teachers  
UNION  
SELECT age  
FROM Students;
```

age
23
29
30
32
40
41
43
45



UNION ALL İŞLEMİ

- SELECT sorgularının sonuçlarını tekrarlarıyla birlikte alt alta birleştirir.
- Hiçbir eleme yapmaz. Performans açısından daha hızlıdır.

```
SELECT sütun1, sütun2  
FROM tablo1  
UNION ALL  
SELECT sütun1, sütun2  
FROM tablo2;
```




UNION ALL İŞLEMİ

Table: Teachers

id	name	age
1	Peter	32
2	Megan	43
3	Rose	29
4	Linda	30
5	Mary	41

Table: Students

id	name	age
1	Harry	23
2	Jack	45
3	Joe	32
4	Dent	23
5	Bruce	40

```
SELECT age  
FROM Teachers  
UNION ALL  
SELECT age  
FROM Students;
```

age

32
43
29
30
41
23
42
32
23
40



SELECT TOP

SELECT TOP, bir tablodan ilk N kaydı almak için kullanılır.

Özellikle büyük veri tablolarında:

- Yalnızca en yüksek 5 maaş
- En son eklenen 10 kayıt
- İlk 3 öğrencinin bilgileri

gibi sorgular yazmak için kullanılır.

```
SELECT TOP n sütun_adı  
FROM tablo_adı;
```



SELECT TOP

id	Adi_soyadi	Sehir	Bolum	Sicil No
1	Salih ESKİOĞLU	İstanbul	Bilgi İşlem Sorumlusu	1234567
2	Ayhan ÇETİNKAYA	Kocaeli	İdari İşler Yöneticisi	2345678
3	Serkan ÖZGÜREL	Erzincan	Bilgi İşlem Sorumlusu	3456789
4	İlhan ÖZLÜ	İstanbul	Muhasebe	7677667

```
SELECT TOP 2 Adi_soyadi, Sehir FROM Personel /*2 tane kayıt getirir
```

Adi_soyadi	Sehir
Salih ESKİOĞLU	İstanbul
Ayhan ÇETİNKAYA	Kocaeli

```
SELECT TOP 25 Percent * FROM Personel /* %25lik kaydı getirir*/
```

id	Adi_soyadi	Sehir	Bolum	Sicil No
1	Salih ESKİOĞLU	İstanbul	Bilgi İşlem Sorumlusu	1234567