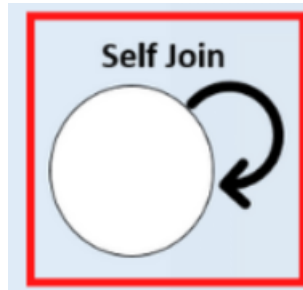
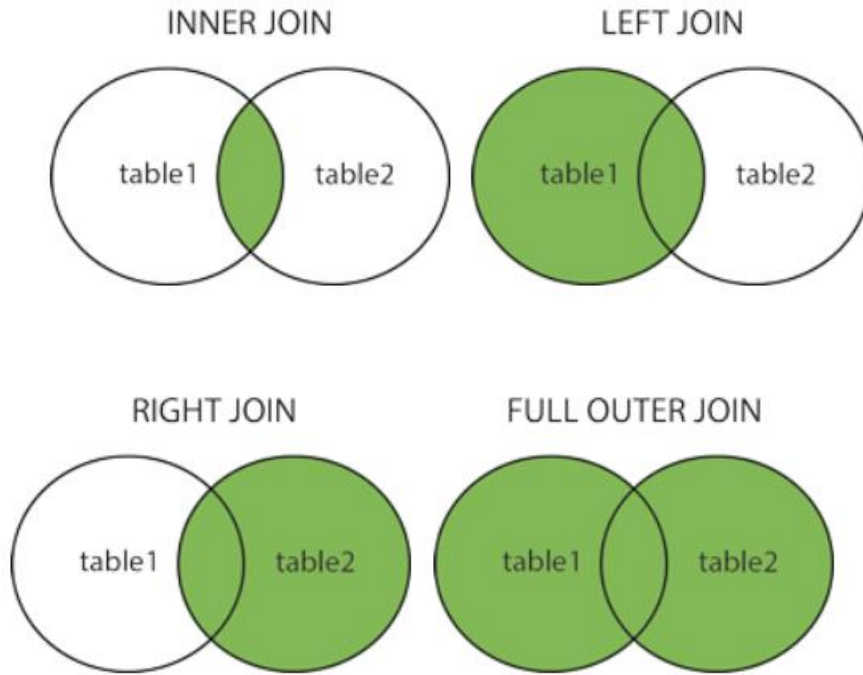


## SQL KATILIM İŞLEMLERİ (JOIN)

İlişkisel veri tabanları ile çalışırken veriler tek bir tabloda olabildiği gibi birden fazla tabloda da bulunabilir. JOIN işlemleri ile birden fazla tablo, ortak bulunan kolonlar yardımı ile birleştirilerek sorgulama işlemleri yapılabilmektedir.

Tablolardaki ilgili sütunu temel alarak iki veya daha fazla tablodaki satırları birleştirmek için bir **JOIN** yan tümce kullanılır.

- **(INNER) JOIN** : Her iki tabloda da eşleşen değerlere sahip kayıtları döndürür
- **LEFT (OUTER) JOIN** : Sol tablodaki tüm kayıtları ve sağ tablodaki eşleşen kayıtları döndürür
- **RIGHT (OUTER) JOIN** : Sağ tablodaki tüm kayıtları ve sol tablodaki eşleşen kayıtları döndürür
- **FULL (OUTER) JOIN** : Sol veya sağ tabloda bir eşleşme olduğunda tüm kayıtları döndürür



OZALFirma - dbo.Musteriler		OZALFirma - dbo.Siparisler		
	MusId	AD	SOYAD	SEHIR
	1	MURAT	AY	ELAZIĞ
	2	NURDAN	ÇALI	TUNCELİ
	3	YUSUF	OKAN	İSTANBUL
	4	AYŞE	ALÇI	TUNCELİ
	5	VELİ	AŞAR	ELAZIĞ
	37	NURAY	AY	KASTAMON...
	77	SEZGİN	ACAR	İSTANBUL
▶*	NULL	NULL	NULL	NULL

OZALFirma - dbo.Siparisler			OZALogrenci
	SipId	MusId	SipTar
	10308	2	2022-09-18
	10309	NULL	2022-09-19
	10310	9	2022-09-20
	10311	3	2022-10-01
	10312	4	2022-10-02
	10313	22	2022-10-03
	10314	37	2022-10-04
▶*	NULL	NULL	NULL

**INNER JOIN:** Her iki tabloda da eşleşen değerlere sahip kayıtları seçen aşağıdaki SQL deyimidir.

```
SELECT sütun_ad(ları)
FROM tablo1
INNER JOIN tablo2
ON tablo1.sütunadı=tablo2.sütunadı
```

Örnek:

SELECT AD,SOYAD,SipID,SipTar FROM Musteriler
INNER JOIN Siparisler ON Musteriler.MusId=Siparisler.MusId

AD	SOYAD	SipID	SipTar
1	NURDAN	ÇALI	10308 2022-09-18
2	YUSUF	OKAN	10311 2022-10-01
3	AYŞE	ALÇI	10312 2022-10-02
4	NURAY	AY	10314 2022-10-04

**Not:** Sütunlar arasında bir eşleşme olduğu sürece her iki tablodaki tüm satırları seçer. "Siparişler" tablosunda "Müşteriler"de eşleşmeyen kayıtlar varsa bu siparişler gösterilmeyecektir!

**LEFT JOIN:** Sol tablodaki (tablo1) tüm kayıtları ve sağ tablodaki (tablo2) eşleşen kayıtları döndürür. Eşleşme yoksa sonuç sağ taraftan 0 kayıttır.

SELECT AD,SOYAD,SipID,SipTar FROM Musteriler
LEFT JOIN Siparisler ON Musteriler.MusId=Siparisler.MusId

AD	SOYAD	SipID	SipTar
1	MURAT	AY	NULL NULL
2	NURDAN	ÇALI	10308 2022-09-18
3	YUSUF	OKAN	10311 2022-10-01
4	AYŞE	ALÇI	10312 2022-10-02
5	VELİ	AŞAR	NULL NULL
6	NURAY	AY	10314 2022-10-04
7	SEZGİN	ACAR	NULL NULL

**Not:** **LEFT JOIN**, sağ tabloda (Siparişler) eşleşme olmasa bile sol tablodaki (Müşteriler) tüm kayıtları döndürür.

**RIGHT JOIN:** Sağ tablodaki (tablo2) tüm kayıtları ve sol tablodaki (tablo1) eşleşen kayıtları döndürür. Eşleşme yoksa sonuç sol taraftan 0 kayıttır.

```
SELECT AD,SOYAD,SipID,SipTar FROM Musteriler  
RIGHT JOIN Siparisler ON Musteriler.MusId=Siparisler.MusId
```

	AD	SOYAD	SipID	SipTar
1	NURDAN	ÇALI	10308	2022-09-18
2	NULL	NULL	10309	2022-09-19
3	NULL	NULL	10310	2022-09-20
4	YUSUF	OKAN	10311	2022-10-01
5	AYŞE	ALÇI	10312	2022-10-02
6	NULL	NULL	10313	2022-10-03
7	NURAY	AY	10314	2022-10-04

**Not:** **RIGHT JOIN**, sol tabloda (Siparişler) eşleşme olmasa bile sağ tablodaki (Çalışanlar) tüm kayıtları döndürür.

**FULL OUTER JOIN (FULL JOIN):** Sol (tablo1) veya sağ (tablo2) tablo kayıtlarında bir eşleşme olduğunda tüm kayıtları döndürür.

```
SELECT AD,SOYAD,SipID,SipTar FROM Musteriler  
FULL JOIN Siparisler ON Musteriler.MusId=Siparisler.MusId
```

	AD	SOYAD	SipID	SipTar
1	MURAT	AY	NULL	NULL
2	NURDAN	ÇALI	10308	2022-09-18
3	YUSUF	OKAN	10311	2022-10-01
4	AYŞE	ALÇI	10312	2022-10-02
5	VELİ	AŞAR	NULL	NULL
6	NURAY	AY	10314	2022-10-04
7	SEZGİN	ACAR	NULL	NULL
8	NULL	NULL	10309	2022-09-19
9	NULL	NULL	10310	2022-09-20
10	NULL	NULL	10313	2022-10-03

### SELF JOIN:

Self Join ifadesi SQL'de bulunan özel bir JOIN işlemidir. Birbirinden farklı iki veya daha çok tablonun birleştirildiği diğer JOIN metotlarının aksine, Self Join işleminde tek bir tablo vardır ve bu tablo kendisi ile birleştirilir. Join işlemi tablo ve tablonun bir kopyası ile gerçekleşir.

SQL'de Self Join işleminin syntax'ı şu şekildedir:

```
1  SELECT
2      t1.kolon_1,
3      t1.kolon_2,
4      t2.kolon_3,
5      .....
6      t2.kolon_n
7  FROM tablo1 t1,
8  INNER JOIN tablo1 t2
9  ON t1.ortak_kolon = t2.ortak_kolon;
```

Self Join ifadesi sık karşılaşılmasa da bilinmesi faydalı ve spesifik kullanımları olan bir yapıdır. İlk başta kulağa biraz farklı gelen bu konsept, bir tablo içerisinde yer alan hiyerarşik verileri sorgulamak veya satırları karşılaştırmak için kullanışlıdır.

Örnek: Aynı şehirdeki müşterileri bulalım!

OZALFirma - dbo.Musteriler		OZALFirma - dbo.Siparisler		
	MusId	AD	SOYAD	SEHIR
	1	MURAT	AY	ELAZIĞ
	2	NURDAN	ÇALI	TUNCELİ
	3	YUSUF	OKAN	İSTANBUL
	4	AYŞE	ALÇI	TUNCELİ
	5	VELİ	AŞAR	ELAZIĞ
	37	NURAY	AY	KASTAMON...
	77	SEZGİN	ACAR	İSTANBUL
▶*	NULL	NULL	NULL	NULL

OZALFirma - dbo.Siparisler		OZALogrenci	
	SipId	MusId	SipTar
	10308	2	2022-09-18
	10309	NULL	2022-09-19
	10310	9	2022-09-20
	10311	3	2022-10-01
	10312	4	2022-10-02
	10313	22	2022-10-03
	10314	37	2022-10-04
▶*	NULL	NULL	NULL

## SQL BAZI PROGRAMLAMA DEYİMLERİ

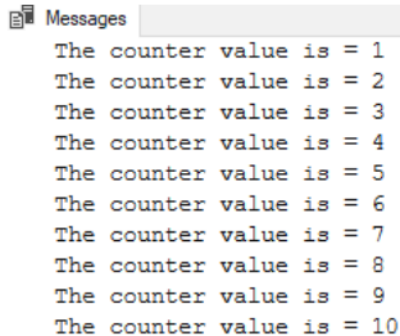
### WHILE DÖNGÜSÜ:

SQL WHILE döngüsü bize, belirtilen koşul sonucu yanlış olana kadar SQL deyimlerini tekrar tekrar yürütme avantajı sağlar.

SQL'deki WHILE döngüsünün sözdizimi aşağıdaki gibidir:

```
WHILE condition
BEGIN
    {...statements...}
END
```

```
DECLARE @Counter INT
SET @Counter=1
WHILE ( @Counter <= 10)
BEGIN
    PRINT 'The counter value is = ' + CONVERT(VARCHAR,@Counter)
    SET @Counter = @Counter + 1
END
```



The screenshot shows the 'Messages' window in SQL Server. It displays the output of the PRINT statement from the WHILE loop, showing the counter value increasing from 1 to 10. The text is as follows:

```
Messages
The counter value is = 1
The counter value is = 2
The counter value is = 3
The counter value is = 4
The counter value is = 5
The counter value is = 6
The counter value is = 7
The counter value is = 8
The counter value is = 9
The counter value is = 10
```

Bu adımda, WHILE döngüsü yardımıyla tüm verileri satır satır okuyacağız:

```
USE tempdb
GO

DECLARE @Counter INT , @MaxId INT,
        @CountryName NVARCHAR(100)
SELECT @Counter = min(Id) , @MaxId = max(Id)
FROM SampleTable

WHILE(@Counter IS NOT NULL
      AND @Counter <= @MaxId)
BEGIN
    SELECT @CountryName = CountryName
    FROM SampleTable WHERE Id = @Counter

    PRINT CONVERT(VARCHAR,@Counter) + '. country name is ' + @CountryName
    SET @Counter = @Counter + 1
END
```