

# SQL Server Birden Fazla Tabloyu Beraber Sorgulama

İlişkisel veri tabanı sistemlerinin (RDMS) hedeflerinden biri de veri tekrarını azaltmak ve performansı üst düzeyde tutmaktır. Bu sebeple tekrar eden veriler tablolara ayrılarak ilgili yerlere verilerin refansı verilir. Örneğin bir sipariş tablosunda veriler temel olarak aşağıdaki iki şekilde tutulabilir.

- Her sipariş satırına ürünün bilgileri ve müşteri bilgilerini girmek
- Müşteri bilgilerini ve ürün bilgilerini kendilerine has tablolarda tutup sipariş tablosuna sadece ilgili müşteri ve ürünün referans kolonlarını vermek.

İlk seçenekte oluşacak örnek bir sipariş tablosu aşağıdaki gibi olur.

Sipariş No	Müşteri Adı	Müşteri Soyadı	Müşteri Tel	Müşteri Adres	Ürün Adı	Sipariş Tarihi	Sipariş Miktarı
1	Tolga	Yalçın	5539152030	İpragaz Mah. Veli Çakmak Cad. No:72	Bebek Arabası	21.10.2018	1
2	Çetin	Songur	5475245555	Mevalana Mah. İkbal Cad. No:32	Tabak	22.12.2018	1

3	Tolga	Yalçın	5539152030	İpragaz Mah. Veli Çakmak Cad. No:72	Tava	25.12.2018	1
4	Tolga	Yalçın	5539152030	İpragaz Mah. Veli Çakmak Cad. No:72	Çanta	30.12.2018	2
5	Çetin	Songur	5475245555	Mevalana Mah. İkbâl Cad. No:32	Şemsiye	02.01.2019	3
6	Tolga	Yalçın	5539152030	İpragaz Mah. Veli Çakmak Cad. No:72	Kitap	05.01.2019	1

Tabloda da görülebileceği gibi birinci seçenekte müşteri bilgisi sürekli bir tekrarda dönüyor. Bunun yerine ikinci seçenekte belirtildiği gibi müşteri bilgileri ayrı tabloda tutulursa yapı aşağıdaki gibi olur.

No	Adı	Soyadı	Telefon	Adres
1	Tolga	Yalçın	5539152030	İpragaz Mah. Veli Çakmak Cad. No:72
2	Çetin	Songur	5475245555	Mevalana Mah. İkbâl Cad. No:32

Müşteri No	Ürün Adı	Tarih	Miktar
------------	----------	-------	--------

1	Bebek Arabası	21.10.20.18	1
2	Tabak	22.12.2018	1
1	Tava	25.12.2018	1
1	Çanta	30.12.2018	2
2	Şemsiye	02.01.2019	3
1	Kitap	05.01.2019	1

Görüldüğü üzere ilgili kayıtlar kendi tablolarında tutulup gerekli yerlere referans numaraları verilirse sipariş tablosunda veri tekrarı engellenmiş olur. Bu şekilde olduğunda müşteri bilgilerinde oluşacak bir değişiklik için bütün sipariş tablosunu dolaşmak yerine sadece müşteri tablosundan kaydı değiştirmek yeterli olacaktır. Referans konusunu [Tablo Seviyesinde Veri Bütünlüğü](#) isimli yazımızda bulabilirsiniz.

Veriler tek tabloda iken bir SELECT \* FROM ile alınabiliyor iken birden fazla tabloda tutulduğunda JOIN komutlarından faydalanmak gerekmektedir. JOIN komutları birden fazla tablo üzerinde sorgulama yaparak istenen sonuç kümesini sunar.

JOIN komutları üç başlıkta toplanır. Bunlar;

- INNER JOIN
- OUTER JOIN
  - LEFT OUTER JOIN
  - RIGHT OUTER JOIN
  - FULL OUTER JOIN
- CROSS JOIN

# JOIN TEMEL YAPISI

Join işleminin temel yapısı aşağıdaki gibidir:

```
SELECT kolon1, kolon2, kolon3...  
FROM Tablo1 [JOIN TÜRÜ] Tablo2 ON [JOIN ŞARFI]
```

## INNER JOIN

INNER JOIN sadece iki tabloda da eşleşen kayıtlar olması durumunda kaydı getirir. Örneğin müşteri ve sipariş tabloları INNER JOIN ile sorgulandığında siparişi olmayan müşteri, sipariş tablosunda eşleşen kaydı olmadığından sonuç ekranına gelmeyecektir. Örnek bir INNER JOIN sorgusu ve sonuç kümesi:

```
SELECT m.Ad, m.Soyad, m.Telefon, m.Adres, s.UrunAd, s.Adet,  
s.Tarih  
FROM Musteriler m INNER JOIN Siparisler s ON m.Id =  
s.MusteriId
```



INNER JOIN

## OUTER JOIN

## LEFT OUTER JOIN

LEFT OUTER JOIN, join kelimesinin solunda kalan tabloyu referans alarak bütün kayıtları getiri. Join kelimesinin sağında kalan tabloda eşleşen kayıt yok ise ilgili kolona NULL değer döner.

```
SELECT m.Ad, m.Soyad, m.Telefon, m.Adres, s.UrunAd, s.Adet,
s.Tarih
FROM Musteriler m LEFT JOIN Siparisler s ON m.Id = s.MusteriId
```



LEFT JOIN

Görüldüğü gibi son satırda ilgili müşteriye ait sipariş olmadığından değerler NULL olarak dönüyor.

## RIGHT OUTER JOIN

RIGHT OUTER JOIN, join kelimesinin sağında kalan tabloyu referans alarak bütün kayıtları getiri. Join kelimesinin solunda kalan tabloda eşleşen kayıt yok ise ilgili kolona NULL değer döner.

```
SELECT m.Ad, m.Soyad, m.Telefon, m.Adres, s.UrunAd, s.Adet,
s.Tarih
FROM Musteriler m right JOIN Siparisler s ON m.Id =
s.MusteriId
```



RIGHT JOIN

## FULL OUTER JOIN

FULL OUTER JOIN, her iki tablodan da bütün kayıtları getirir, eşleşmeyen kayıtların karşılıklarını NULL değer olarak döner.

```
SELECT m.Ad, m.Soyad, m.Telefon, m.Adres, s.UrunAd, s.Adet,
s.Tarih
FROM Musteriler m FULL JOIN Siparisler s ON m.Id = s.MusteriId
```



FULL JOIN

## CROSS JOIN

CROSS JOIN, matematikteki kartezyen çarpımı gibi çalışır ve sol tabloya karşılık sağdaki bütün kayıtları getirir. Diğer join türlerinden farklı olarak ON şartı kullanılmaz. Örneğin sol tabloda üç, sağ tabloda 8 kayıt var ise sonuç kümesinde  $3 * 8 = 24$  kayıt döner.

```
SELECT m.Ad, m.Soyad, m.Telefon, m.Adres, s.UrunAd, s.Adet,
s.Tarih
FROM Musteriler m cross JOIN Siparisler s
```



CROSS JOIN