

SQL / 4 - 5 Aggregate Functions and Group By - İlişkisel Veri Tabanı(RDMS)

Aggregate Functions

AGGREGATE FUNCTIONS
(SUM,MIN,MAX,AVG,COUNT)

Nasıl Kullanılır?

```
SELECT SUM(PRICE),COUNT(ID),MIN(PRICE),MAX(PRICE),  
AVG(PRICE)  
FROM TABLOADI  
WHERE CITY='İSTANBUL'
```

Group By Functions

```
SELECT KOLON1,KOLON2, SUM(PRICE),COUNT(ID),MIN(PRICE),MAX(PRICE),  
AVG(PRICE)  
FROM TABLOADI  
GROUP BY KOLON1, KOLON2
```

İlişkisel Veri Tabanı Kavramı (RDMS)



- İlişkisel veritabanı kavramı (RDMS)
- Neden ilişkisel veritabanı kullanırız?
- Primary key
- Foreign key
- İlişkisel tablolardan sorgu çekme
- Join kavramı ve Join çeşitleri
- Subquery kavramı

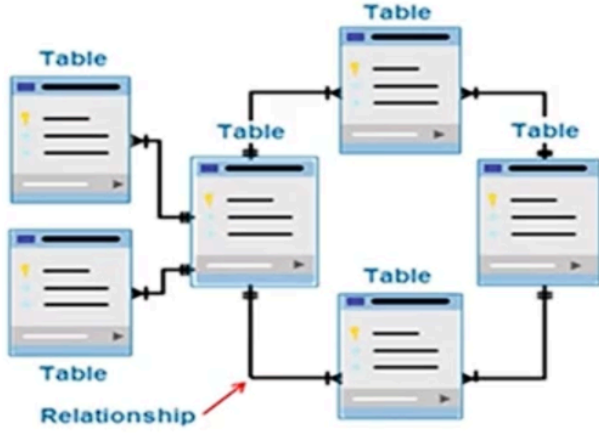
İlişkisel Veri Tabanı nedir?

Tekrar eden verileri tekilleştirmek amacıyla yapılandırılan veri tabanı sistemleridir.

İlişkisel veri tabanı, verilerin tablo halinde düzenlendiği ve bu tablolar arasında ilişkilerin kurulduğu bir veri tabanı modelidir. Bu model, verilerin kolayca erişilmesini ve manipüle edilmesini sağlar. İşte temel özellikleri:

- **Tablolar (Kayıtlar):** Veriler, satırlar (kayıtlar) ve sütunlar (alanlar) halinde düzenlenir.

- **İlişkiler:** Tablolar arasında belirli alanlar üzerinden ilişkiler kurulur. Örneğin, bir "Müşteriler" tablosu ile "Siparişler" tablosu arasında müşterinin kimlik numarası üzerinden ilişki kurulabilir.
- **SQL (Structured Query Language):** İlişkisel veri tabanları, verileri sorgulamak ve manipüle etmek için SQL dilini kullanır.
- **Veri Bütünlüğü:** İlişkisel veri tabanları, veri bütünlüğünü sağlamak için çeşitli kısıtlamalar ve kurallar uygular.



Avantajları:

- Tekrar eden verilerin sürekli girmekten kaynaklanan emek ve iş gücü kaybı.
- Tekrar eden verilerin gereksiz yere yer işgal etmesi ve kaynak israfı.
- Veri girerken insan hatası sebebiyle veri bütünlüğünün bozulması.
- Veride geçmişe dönük olarak güncelleme yapmanın zor olması.

Primary Key ve Foreign Key Kavramları

Primary Key

- **Tanım:** Birincil anahtar, bir tablodaki her kaydı benzersiz şekilde tanımlayan bir veya birden fazla sütundur.
- **Özellikler:**
 - Benzersizdir: Birincil anahtara atanan her değer eşsizdir.
 - Boş değer (NULL) içermez: Birincil anahtar sütununda NULL değer bulunamaz.
 - Tekdir: Her tabloda yalnızca bir birincil anahtar olabilir.

Foreign Key

- **Tanım:** Yabancı anahtar, bir tablodaki sütunun başka bir tablodaki birincil anahtara referans verdiği kısıtlamadır. İki tablo arasında bir ilişki oluşturur.
- **Özellikler:**
 - Bir ilişkisel bütünlük sağlar: Yabancı anahtar, ilgili olduğu anahtarın var olup olmadığını kontrol eder.
 - Tablolar arası bağlantıyı korur.

CUSTOMERS
ID
TCNUMBER
NAMESURNAME
TELNR
ADDRESS

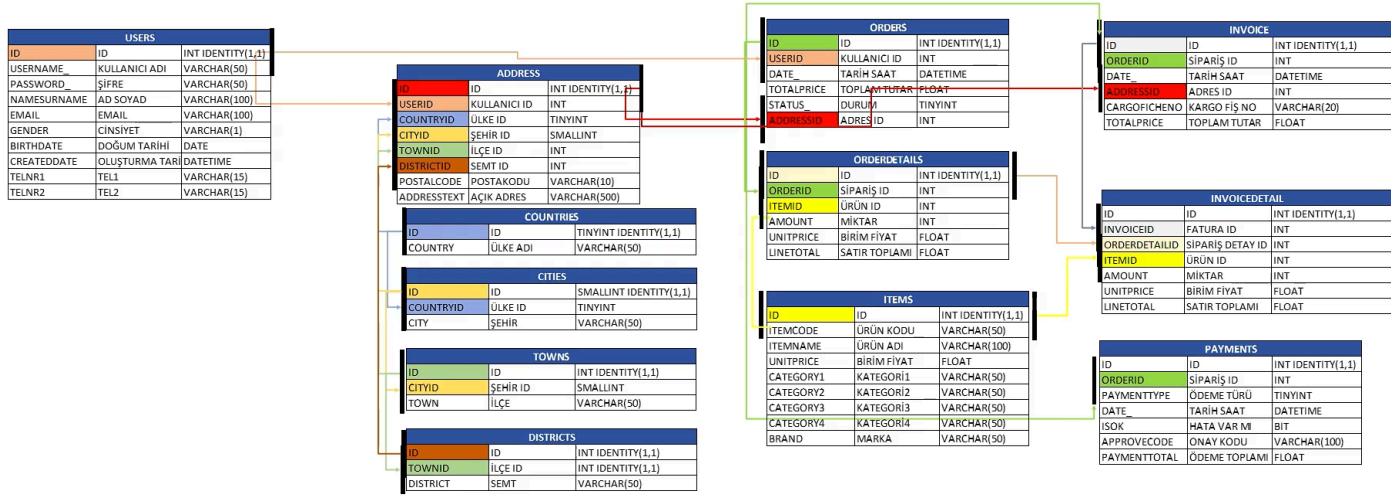
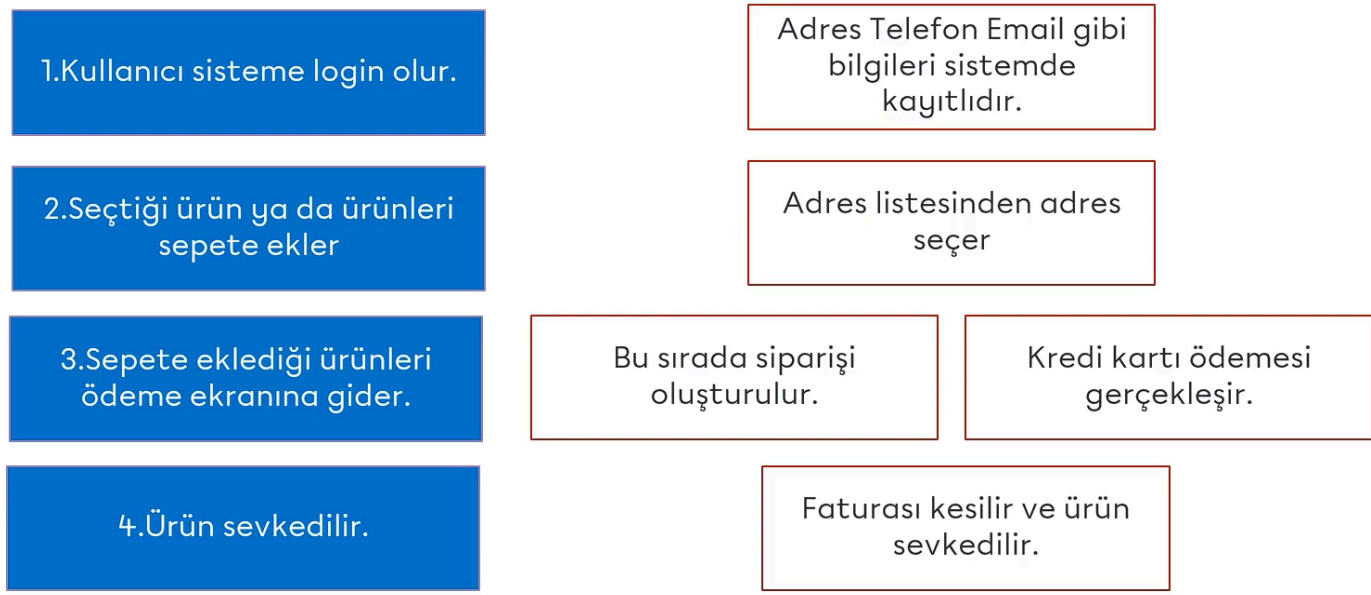
SALES
ID
DATE_
CUSTOMERID
ITEMID
PRICE
AMOUNT

ITEMS
ID
ITEMNAME
PRICE

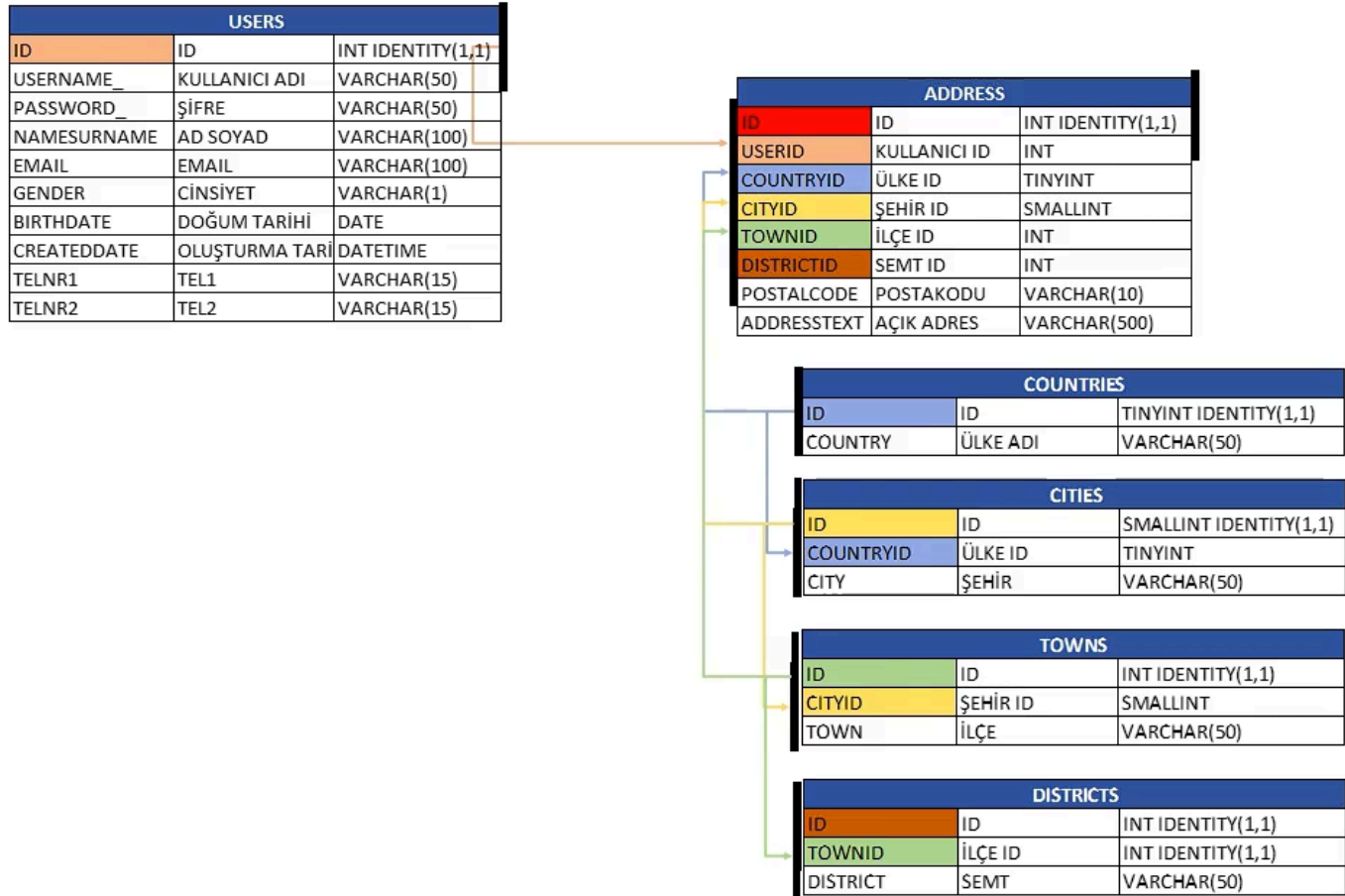
Primary Key

Foreign Key

Örnek bir E-Ticaret Sistemi



User ve Adres Tabloları



İki Tabloyu Birleştirme

1 - Geleneksel Yöntem

```

SELECT
A.KOLON1,A.KOLON2,B.KOLON3,B.KOLON4
FROM TABLO1 A,TABLO2 B
WHERE A.PK_KOLON=B.FK_KOLON
  
```

PK: Primary Key kolon

FK: Foreign Key kolon

```

SQLQuery15.sql - I...ADE_RDMS (sa (66))*  SQLQuery14.sql - I...ADE_RDMS (sa (67))*
SELECT U.USERNAME_, U.NAMESURNAME, A.POSTALCODE, A.ADDRESSTEXT,
C.COUNTRY, A.CITYID, CT.CITY, T.TOWN, D.DISTRICT
FROM USERS U, ADDRESS A, COUNTRIES C, CITIES CT, TOWNS T, DISTRICTS D
WHERE U.ID=A.USERID AND A.COUNTRYID=C.ID
AND CT.ID=A.CITYID
AND T.ID=A.TOWNID
AND D.ID=A.DISTRICTID
--AND U.ID=1

```

2 - Join Kullanımı ile

```

SELECT
A.KOLON1,A.KOLON2,B.KOLON3,B.KOLON4
FROM TABLO1 A
JOIN TABLO2 B ON A.PK_KOLON=B.FK_KOLON

```

PK: Primery Key

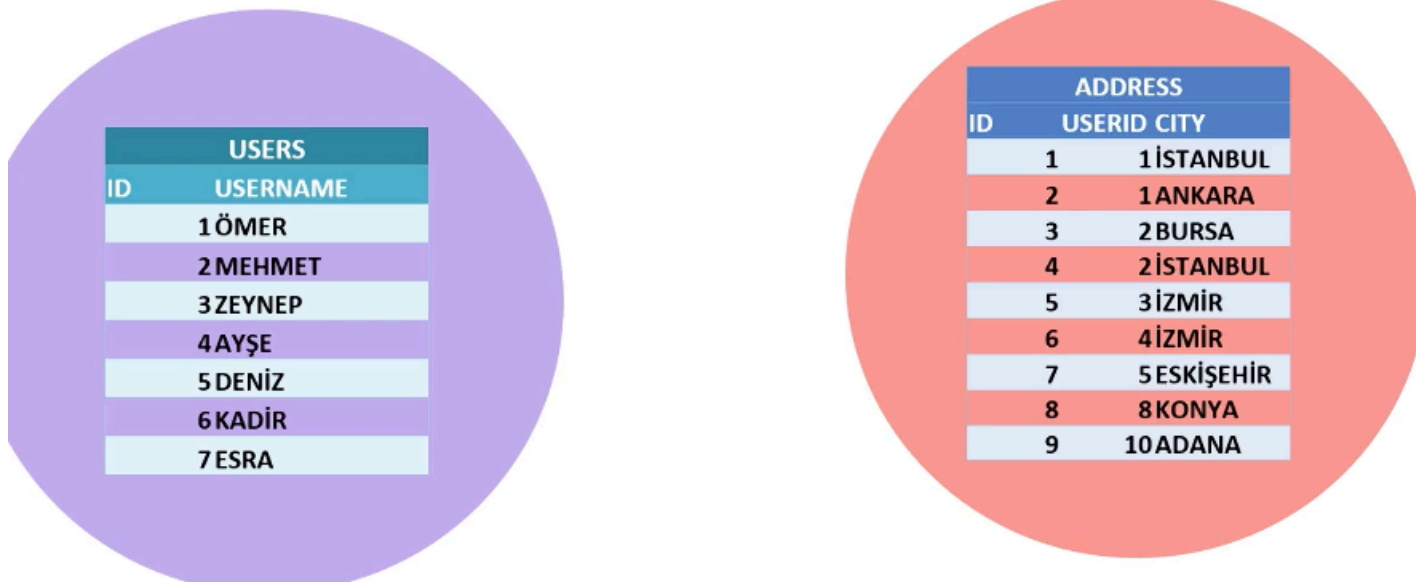
FK: Foreigne Key

```

SELECT U.ID, U.NAMESURNAME, A.POSTALCODE, A.ADDRESSTEXT,
C.COUNTRY, CT.CITY, T.TOWN, D.DISTRICT
FROM USERS U
JOIN ADDRESS A ON U.ID=A.USERID
JOIN COUNTRIES C ON C.ID=A.COUNTRYID
JOIN CITIES CT ON CT.ID=A.CITYID
JOIN TOWNS T ON T.ID=A.TOWNID
JOIN DISTRICTS D ON D.ID=A.TOWNID

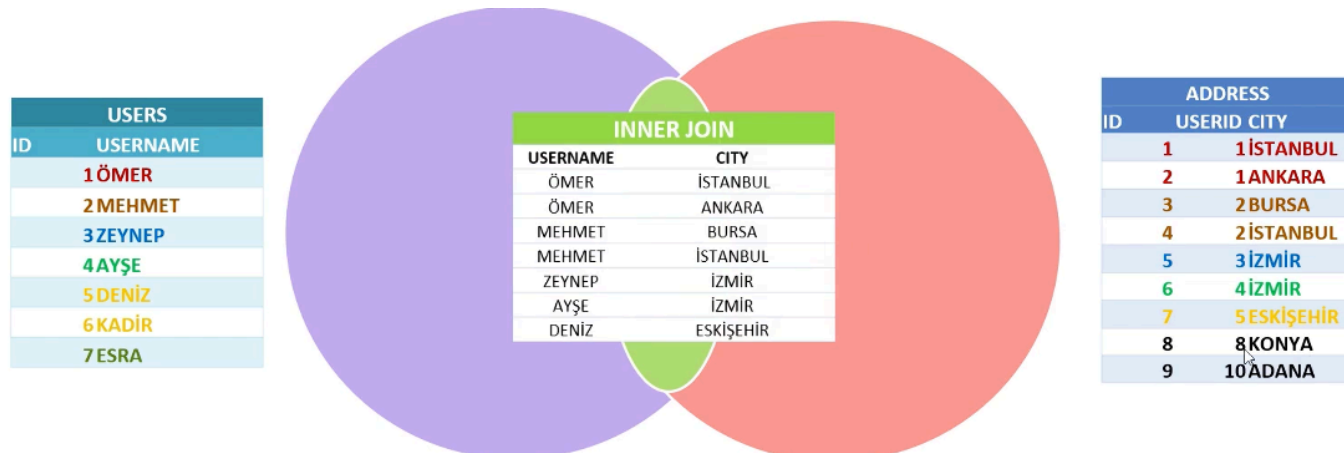
```

2 - A Join Türleri

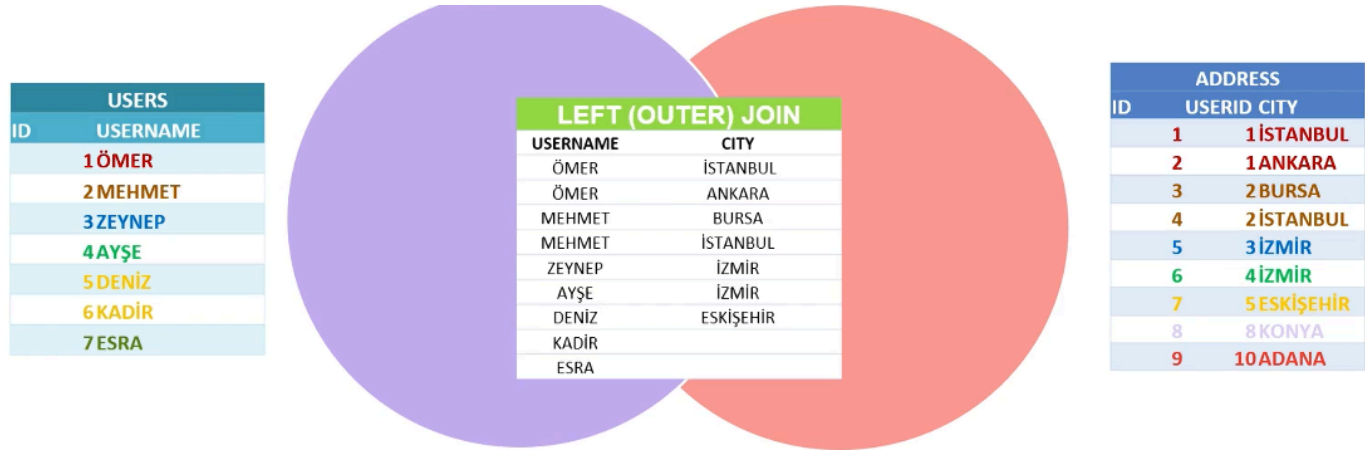


Users ve address isimli iki tabloyu birleştirmek istediğimizi düşünelim.

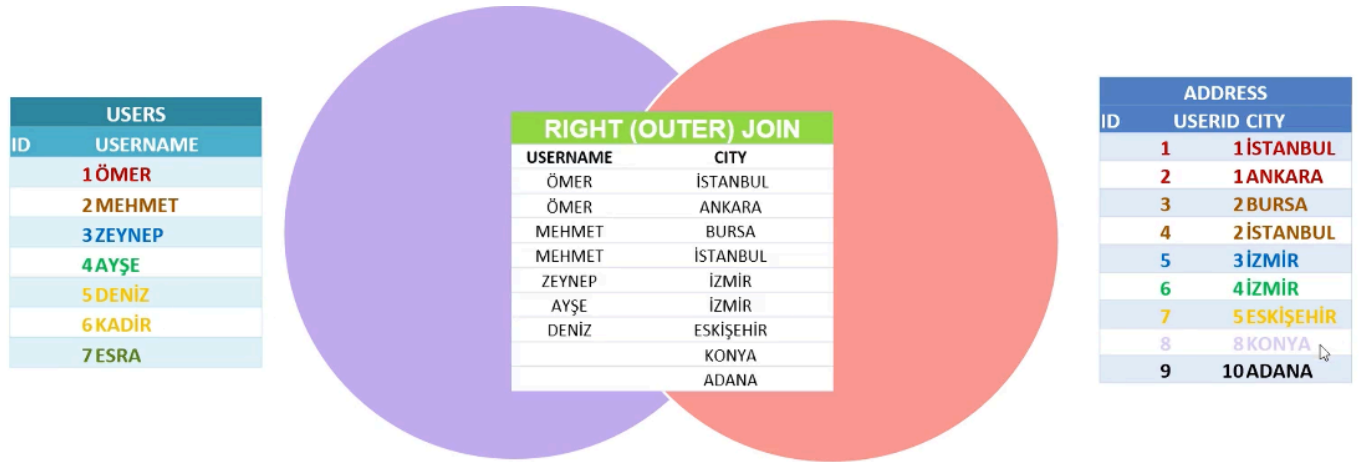
INNER JOIN: Her iki tablonun kesişim kümesini getirir.



LEFT (OUTHER) JOIN: Sol tarafı referans alır, sol taraftaki bütün kayıtları getirir.



RIGHT (OUTER) JOIN: Sağ taraftaki tabloyu referans alır, sağ taraftaki bütün kayıtları getirir.



FULL JOIN: Her iki tablonun bileşim kümesidir, yani tüm kayıtları getirir.

