4-1
Sql Veri Tipleri
ve
Örnek Veritabanı Tasarımı

Veritabanı 1

Ms SQL Server Veri Tipleri

- **Bigint:** -9,223,372,036,854,775,808 ile 9,223,372,036,854,775,807 arasındaki sayıları tutar. 8 baytlık yer kaplar.
- **Int:** -2,147,483,648 ile 2,147,483,648 arasındaki sayıları tutar. fiziksel olarak 4 baytlık yer kaplar.
- **Smallint:** -32,768 ile 32,767 arasındaki sayıları tutar. 2 baytlık yer kaplar.
- **Tinyint:** 0 ile 255 arasındaki sayıları tutar. 1 baytlık yer kaplar.

Ms SQL Server Veri Tipleri

- **Decimal (x,y):** Boyutu 5 ile 17 byte arasında değişir. ondalıklı sayıları tutmaya yarar. İki parametre alır. İlk parametre (x) sayının kaç karakter uzunluğunda olacağını, ikinci parametre (y) ise bu karakterlerin kaçının ondalık kısım olduğunu ifade eder. Örneğin decimal(5,2) şeklinde tanımlanan bir değişken 123,45 değeri alabilir.
- **Numeric (x,y): Kullanımı ve özellikleri decimal ile aynıdır.**
- ***Float:*** Çok büyük ve çok küçük kesirli sayılar için kullanılan veri tipidir. Boyutu 4 ile 8 byte arasındadır.
- Real: Float ile aynı özelliklere sahip.

Ms SQL Server Veri Tipleri

- **Money:** -922 337 203 685 47,5808 ile 922 337 203 685 477,5807 arası parasal veri tutabilir. 8 baytlık yer kaplar.
- **Smallmoney: -**214 748,3648 ile 214 748,3647 arası parasal veri tutabilir. 4 baytlık yer kaplar.
- **Datetime:** 1 Ocak 1753 ile 31 Aralık 9999 tarihleri arasındaki herhangi bir tarihi saatiyle birlikte tutar. 3.33 milisaniyelik bir hassasiyete sahiptir. 8 baytlık yer kaplar.
- **Smalldatetime :** 1 Ocak 1990 ile 6 Haziran 2079 tarihleri arasındaki herhangi bir tarihi saatiyle birlikte tutar. 10 dakikalık bir hassasiyete sahiptir. 4 baytlık yer kaplar.

Ms SQL Server Veri Tipleri

- ANSI standartlarına uyan herhangi bir karakter 1 byte yer kaplar. Ancak Unicode karakterler ise hafızada 2 byte yer kaplar. Unicode karakterler farklı dillere özgü karakterlerdir.Türkçe' de kullandığımız ç,ğ,ş,ö gibi harfler bizim dilimize özgüdür ve ANSI standartlarında yer almamaktadır.
- **Char(n):** Boyutu1 ile 8000 arasında değişir. Maksimum 8000 karakter tutar.
- **Nchar(n):** Boyutu 2 ile 8000 arasında değişir. Maksimum 4000 karakter tutar.
- **Varchar(n):** Boyutu1 ile 8000 arasında değişir. Maksimum 8000 karakter tutar.
- **Nvarchar(n):** Boyutu 2 ile 8000 arasında değişir. Maksimum 4000 karakter tutar.
- **Nvarchar(MAX):** Maksimum 2 gigabyte (536.870.912 karakter) veri saklar.
- **Text:** Maksimum 2 gigabyte (1.073.741.824 karakter) veri saklar.
- **Ntext:** Maksimum 2 gigabyte (536.870.912 karakter) veri saklar.

Ms SQL Server Veri Tipleri

- Örnek olarak char(10) olarak tanımladığımız değişkene “ali” değerini de yazsak “mehmet” değerini de yazsak kapladığı alan 10 byte dır.Fakat varchar(10) olarak tanımladığımız bir değişkene “ali” değerini yazarsak 3 byte “mehmet” değerini yazarsak 6 byte yer kaplar. yani veriye göre kapladığı alan değişir. char ve varchar arasındaki fark budur.

- nvarchar(10) olarak tanımladığımız değişkene “ali” değerini yazarsak kaç byte yer kaplar “mehmet” değerini yazarsak kaç byte yer kaplar?

- Nvarchar ile varchar arasındaki fark birisi Türkçe karakterleri desteklerken diğeri desteklemez.

- Nvarchar(5) olarak tanımladığımız bir değişkene “Mehmet” değerini atarsak ne olur.

Ms SQL Server Veri Tipleri

- **Binary(n):** Boyutu 1 ile 8000 byte arasında değişir. Sabit boyutta binary veri tutmak için kullanılır.
- **Varbinary(n):** Boyutu 1 ile 8000 byte arasında değişir. Değişken boyutta binary veri tutmak için kullanılır.
- **Varbinary(MAX):** Boyutu maksimum 2 gigabyte tır. Değişken boyutta binary veri tutmak için kullanılır.
- **Image:** Boyutu maksimum 2 gigabyte tır. Sadece resim dosyaları için değildir tüm dosyalar tutulabilir.

Ms SQL Server Veri Tipleri

- **Bit:** 1 bayt yer kaplar. 1 ya da 0 değeri depolar yani true yada false değeri döndürür. örnek olarak cinsiyet sütunarında erkek/kadın gibi iki değer alan durumlarda kullanılabilir.
- **Uniqueidentifier:** Bir satırı tek (aynı kopyası bulunmayan) yapmak için tanımlanan 16 bitlik kimliktir (GUID-Global Unique Identifier) .
- **Xml:** Boyutu 2 gigabyte a kadar olan Xml dokümanlarını tutmaya yarar. Seçenekler yardımıyla sadece belirtilen yapıdaki bir xml dokümanını saklamasını da sağlayabiliriz.

Null ve Not Null

- Bir alanın NULL olarak tanımlanması demek bu alana veri girilmeden yeni bir satır oluşturulabileceği anlamına gelir. Örneğin: iki isimli insanları düşünerek 2. isim diye bir yer açtık tablomuzda. Fakat her insanın 2. ismi olmak zorunda değil. Ozaman 2. isim özelliğini NULL yapabiliriz.
- Fakat bazı veriler varki bunları kesinlikle NULL yapamayız. Mesela TC kimlik numarası her insanda olmak zorundadır. O yüzden TC kimlik numarası değerini kesinlikle boş bırakılamaz.

IDENTITY
(Otomatik artan sayı)

- Kişi oluşturduğu tabloda verilerin ardışık sayılardan oluşan değerleri alması isteniyorsa identity özelliği kullanılır. Yani veri ekleyeceğimiz vakit ürün veya nesnesinin ID sayısı otomatik olarak artar.
- Identity alanlar güncellenemezler ve boş bırakılamazlar. identity alanlar genelde primary key olarak kullanılırlar.

Örnek Veritabanı Tasarımı 2
Basit Bir Örnek

Örnek Veritabanı Tasarımı -2

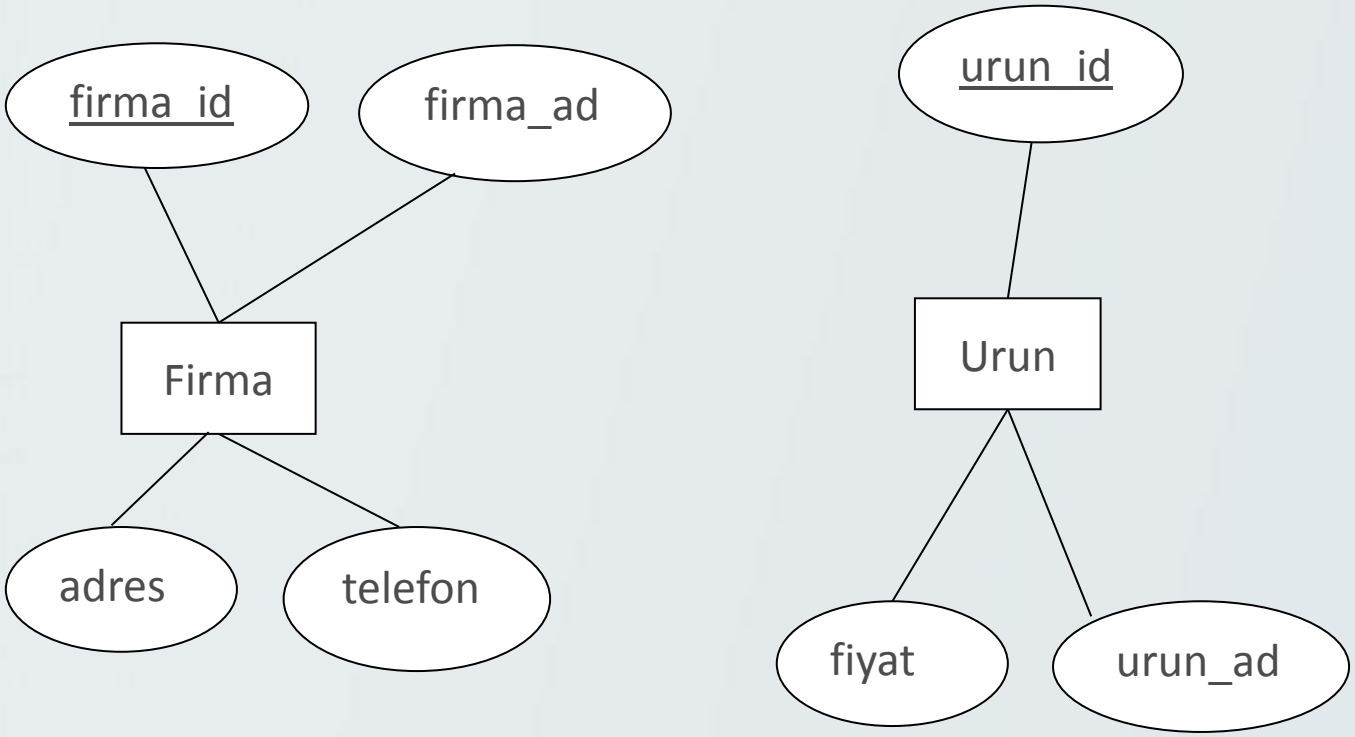
- Hazırlayacağımız veritabanı bir şirketin
  - aldığı ve sattığı ürün bilgilerini
  - ürünleri hangi firmalardan ne zaman ve ne kadar alındığını
  - hangi firmalara ne zaman ve kadar sattığı bilgilerini tutacaktır.

- İlk iş kavramsal tasarımı yapmak.

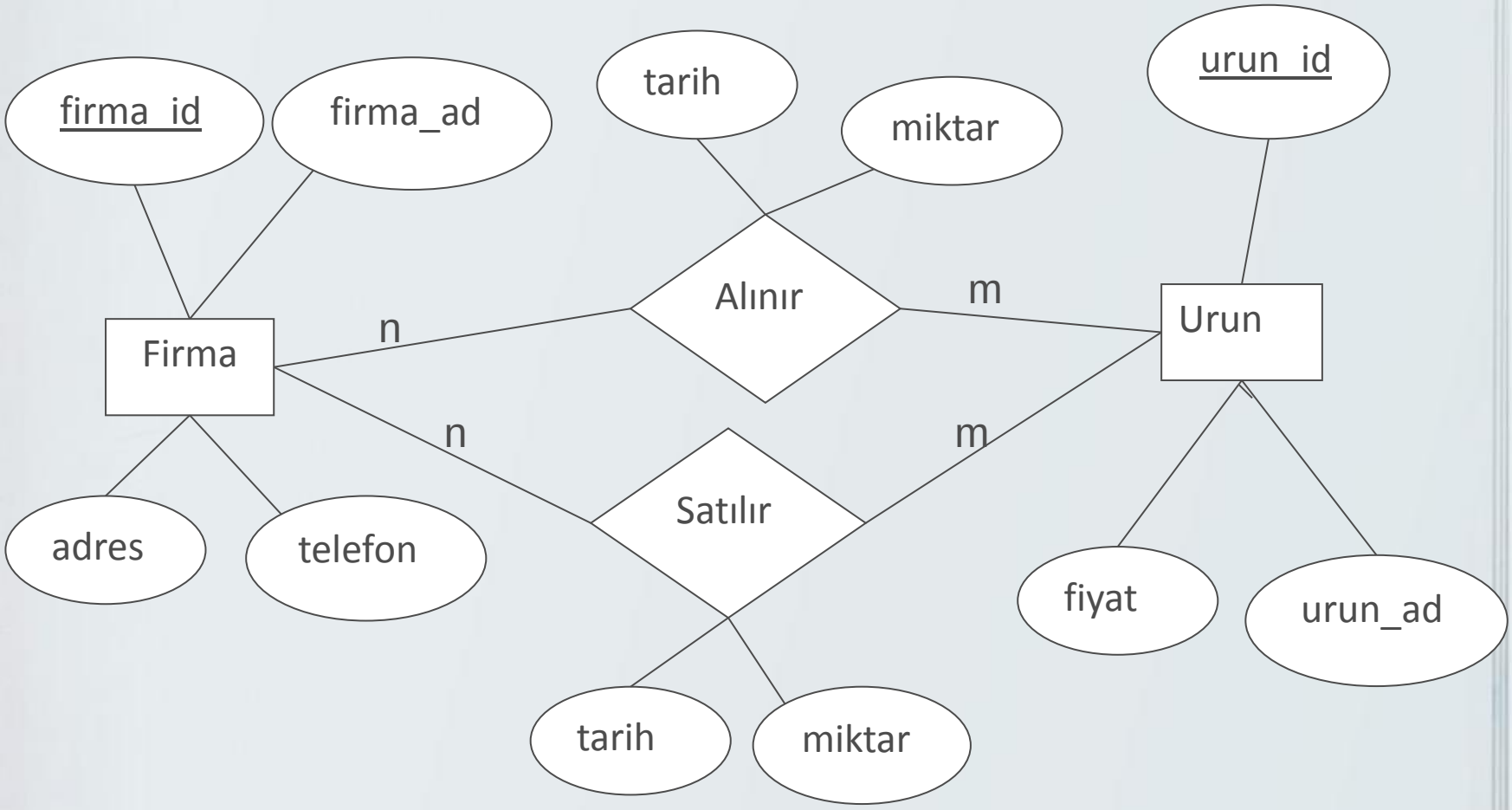
Örnek Veritabanı Tasarımı -2

- Varlıklarımız
  - Ürünler,Firmalar
- Niteliklerimiz
  - Urunler
    - urun_id, urunAd, fiyat
  - Firmalar
    - firma_id, firmaAd, telefon, adres

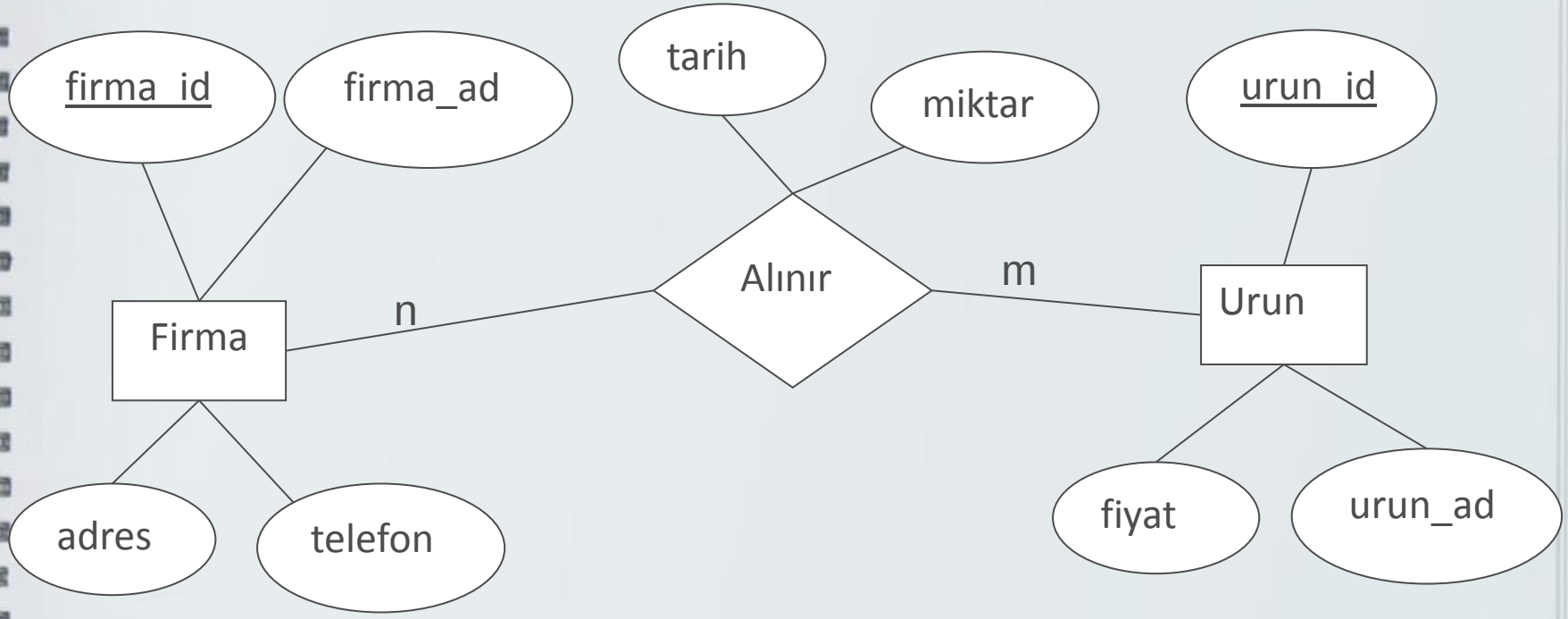
Varlıklar ve Nitelikleri



Kavramsal Tasarım

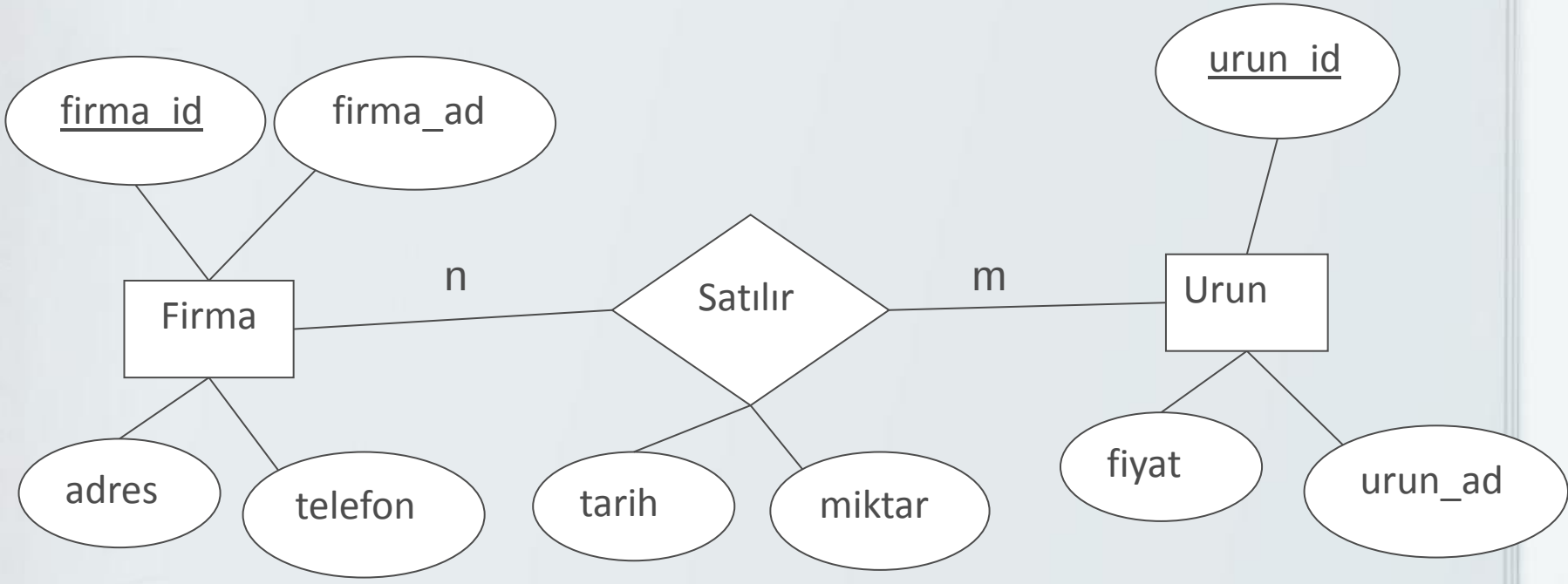


Kavramsal Tasarımın Tabloya Dönüşümü



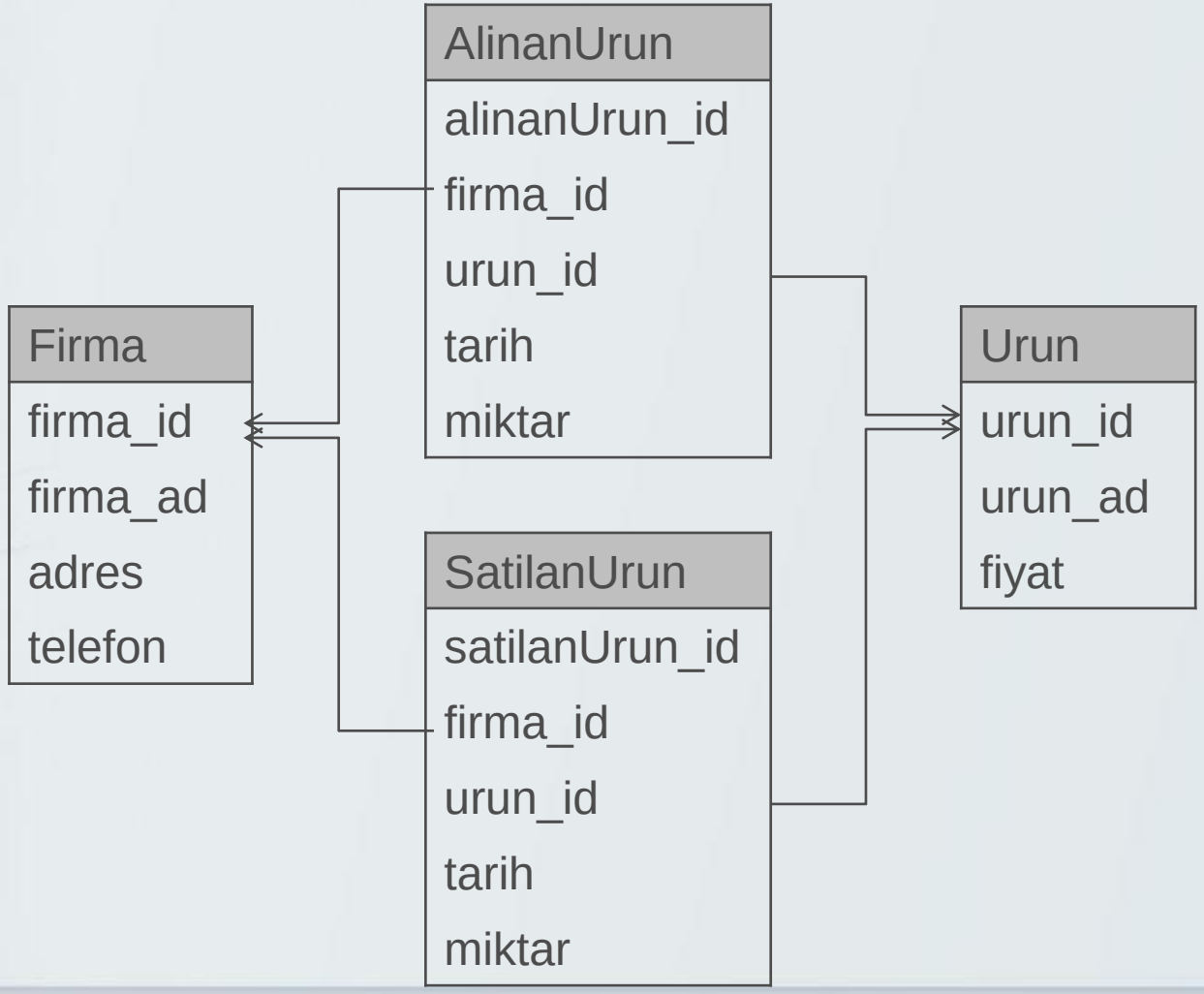
- Firma (**firma_id**, firma_ad, adres, telefon)
- Urun (**urun_id**, urun_ad, fiyat)
- AlinanUrun (tarih, miktar, firma_id, urun_id, **alinanUrun_id)**

Kavramsal Tasarımın Tabloya Dönüşümü



- Firma (**firma_id**, firma_ad, adres, telefon)
- Urun (**urun_id**, urun_ad, fiyat)
- SatilanUrun (tarih, miktar, firma_id**,** urun_id**, satilanUrun_id**)

Kavramsal Tasarımın Tabloya Dönüşümü

- Firma (**firma_id**, firma_ad, adres, telefon)
- Urun (**urun_id**, urun_ad, fiyat)
- AlinanUrun (tarih, miktar, firma_id, urun_id, **alinanUrun_id)**
- SatilanUrun (tarih, miktar, firma_id**,** urun_id**, satilanUrun_id**)

Tablolar arası ilişkiler