Bilgisayar Teknolojileri Bölümü
Bilgisayar Programcılığı Programı

Öğr. Gör. Cansu AYVAZ GÜVEN

VERITABANI-I

Varlık-İlişki Diyagramlarının İlişki Şemalarına Dönüştürülmesi

E-R Çizelgelerini İVTYS'ye Dönüştürme

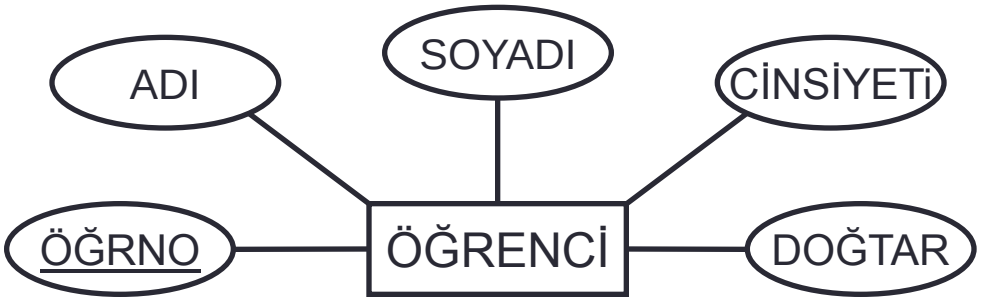
- Varlık-ilişki modeli kullanılarak veri modelleme yapıldıktan sonra, eğer veri tabanını gerçekleştirmek için ilişkisel bir VTYS kullanılacaksa, oluşturulan varlık-ilişki çizelgesinin tablolara (ilişki şemalarına) dönüştürülmesi gerekir.

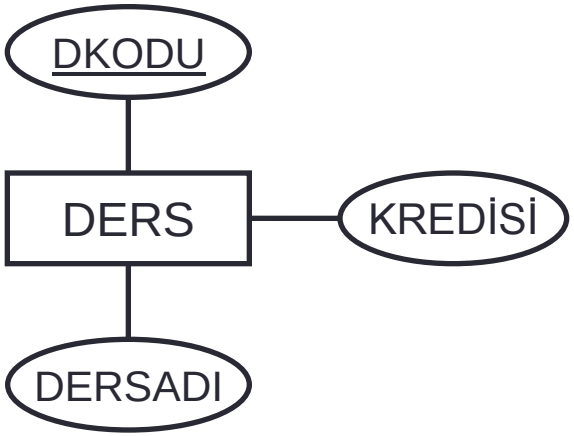
- Bu amaçla varlık-ilişki modelindeki her farklı kavramın (varlık kümesi, nitelik, ilişki kümesi, ilişki kümesinin tanımlayıcı niteliği, güçlü ve zayıf varlık kümeleri, var olma bağımlılığı, genelleme, kümeleme, ..vb) ilişkisel modelde nasıl gösterileceğinin bilinmesi gerekir.

Varlık Kümelerinin Dönüştürülmesi

- Varlık-ilişki modelindeki her varlık kümesi için ilişkisel modelde bir tablo oluşturulur.
- Tablonun alanları olarak varlık kümesinin nitelikleri kullanılır.

Örnek



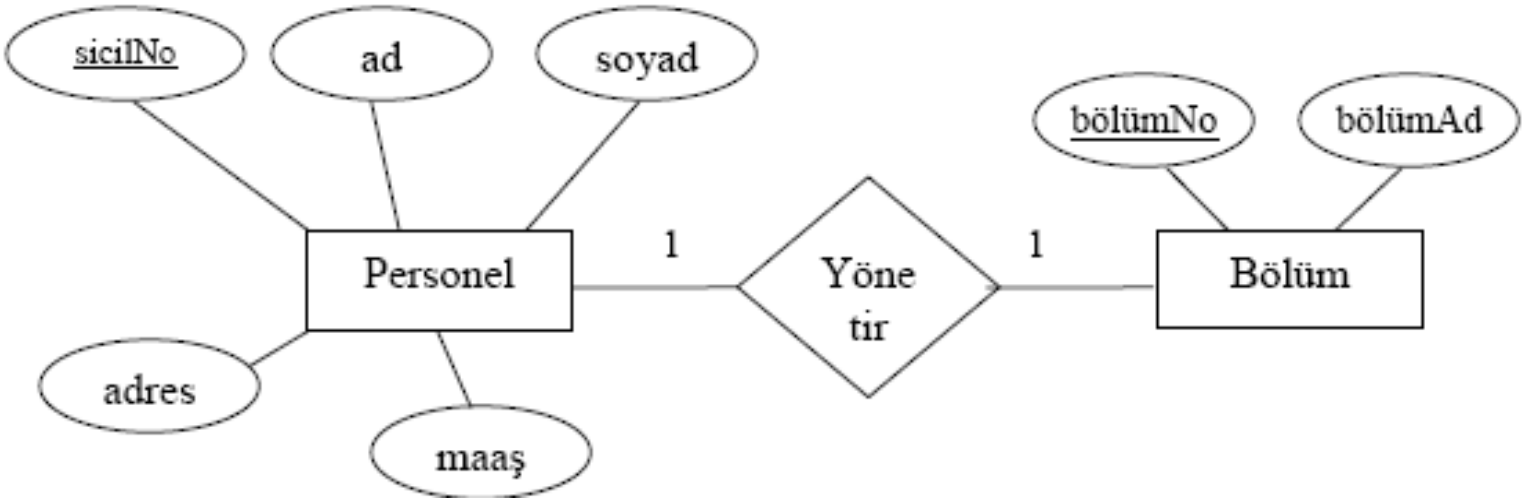


- ÖĞRENCİ ve DERS varlık kümelerine karşılık, ilişkisel modelde aşağıdaki tablolar oluşturulur.

  - ÖĞRENCİ (ÖĞRNO, ADI, SOYADI, CİNSİYETİ, DOĞTAR)
  - DERS (DKODU, DERSADI, KREDİSİ)

Bire-bir ilişkilerin tabloya dönüştürülmesi

- Birebir ilişkiyi oluşturan varlık kümeleri tablolara dönüştürülür. Nitelikleri tabloların alanlarına dönüşür. Uygun olan varlık kümesinin anahtar alanı diğer varlık kümesine yabancı anahtar olarak eklenir. Bire-bir ilişkide belirtilen tanımlayıcı nitelikler, yabancı anahtar eklenen tabloya alan olarak eklenir.



1. Yol

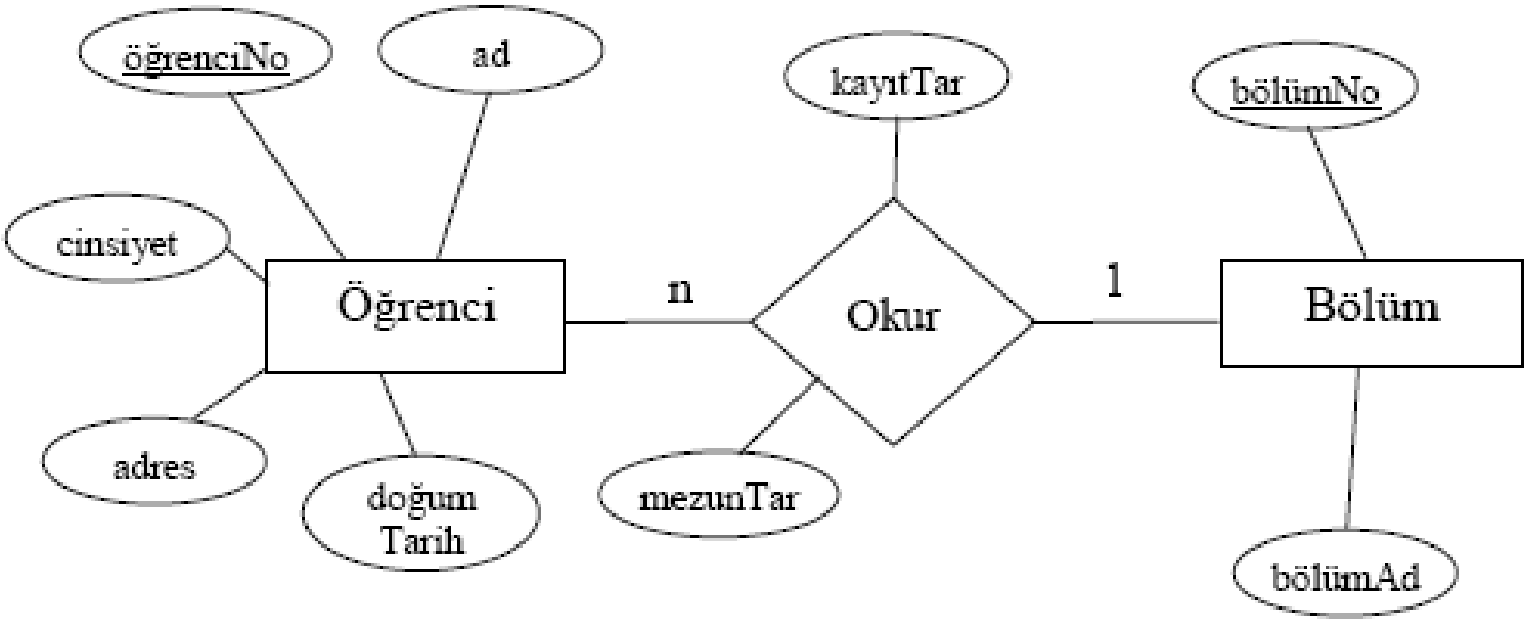
- Bölüm (bölümNo, bölümAd, sicilNo)
- Personel (sicilNo, ad, soyad, adres, maaş)

2. Yol

- Bölüm (bölümNo, bölümAd)
- Personel (sicilNo, ad, soyad, adres, maaş, bölümNo)

Bire - çok(1- n) ilişkilerin tablolara dönüştürülmesi

- İlişkiyi oluşturan varlık kümeleri tablolara dönüştürülür. İlişkinin n tarafındaki tabloya 1 tarafındaki tablonun anahtar alanı yabancı anahtar olarak eklenir. İlişkide belirtilen tanımlayıcı nitelikler n tarafına alan olarak eklenirler.

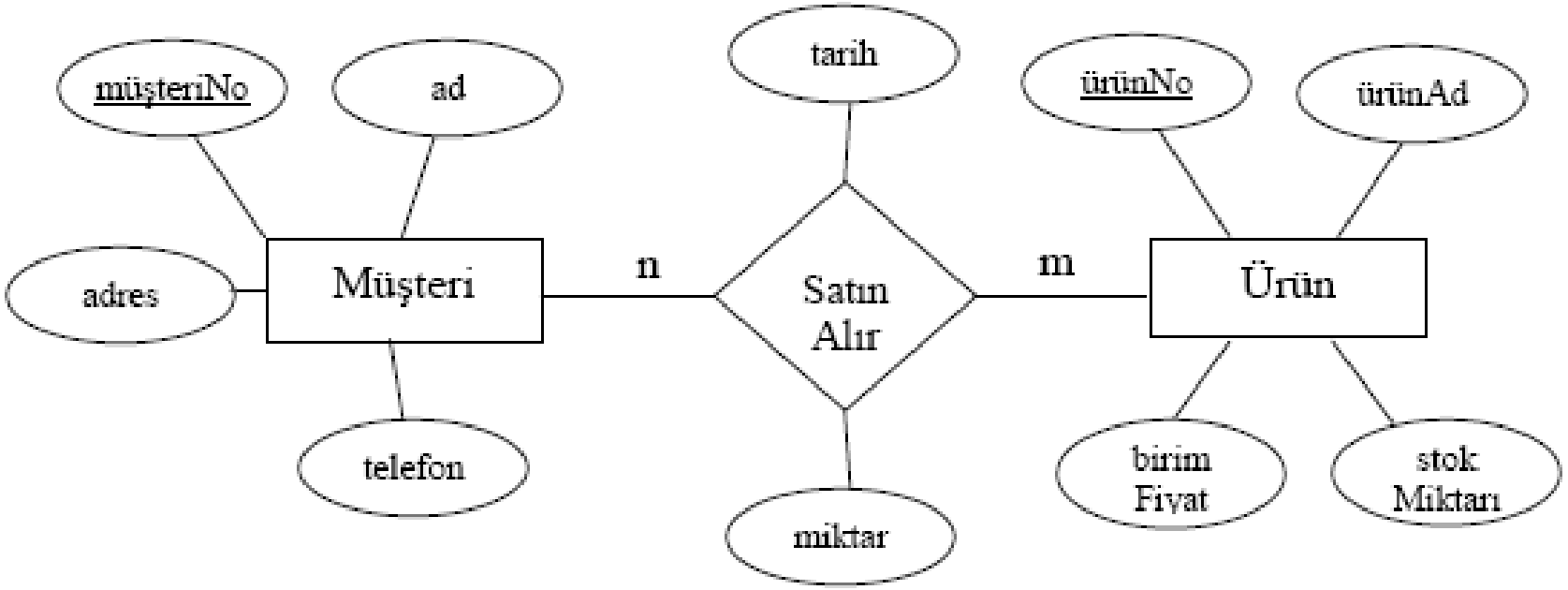


- Öğrenci (öğrenciNo, ad, cinsiyet, adres, doğumTarih, bölümNo, kayıtTar, mezunTar)
- Bölüm(bölümNo, bölümAd)

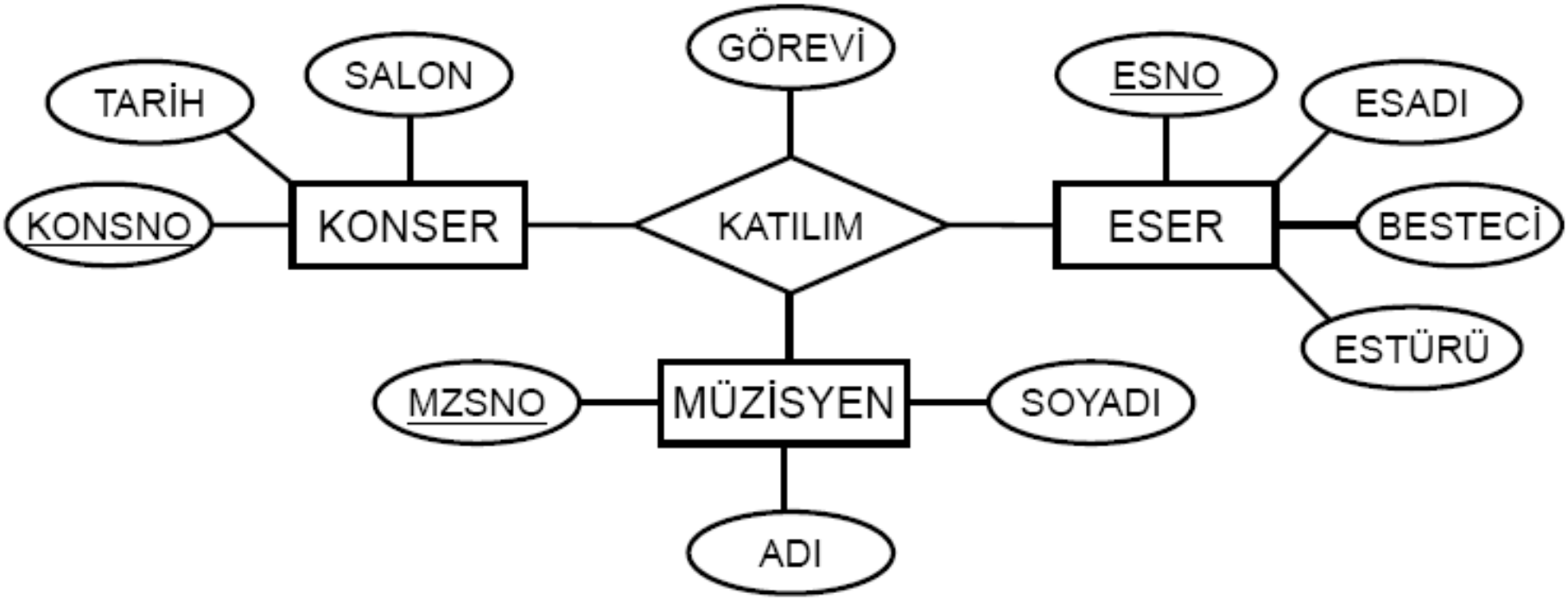
Çoğa - çok(n-m) ilişkilerin tablolara dönüştürülmesi

- Eğer bir ikili ilişki kümesinin türü çoktan-çoğa (ilişki hiçbir yönde işlevsel değil) ise ilişkisel modelde bu ilişki kümesi için ayrı bir şema oluşturulur.

- İlişki kümesine karşılık gelen ilişki tabloda, her iki varlık kümesinin anahtarları ile, varsa ilişki tanımlayıcı niteliklerine yer verilir.



- Müşteri (müşteriNo, ad, adres, telefon)
- Ürün (ürünNo, ürünAd, birimFiyat, stokMiktarı)
- SatınAlır (müşteriNo, ürünNo, tarih, miktar)

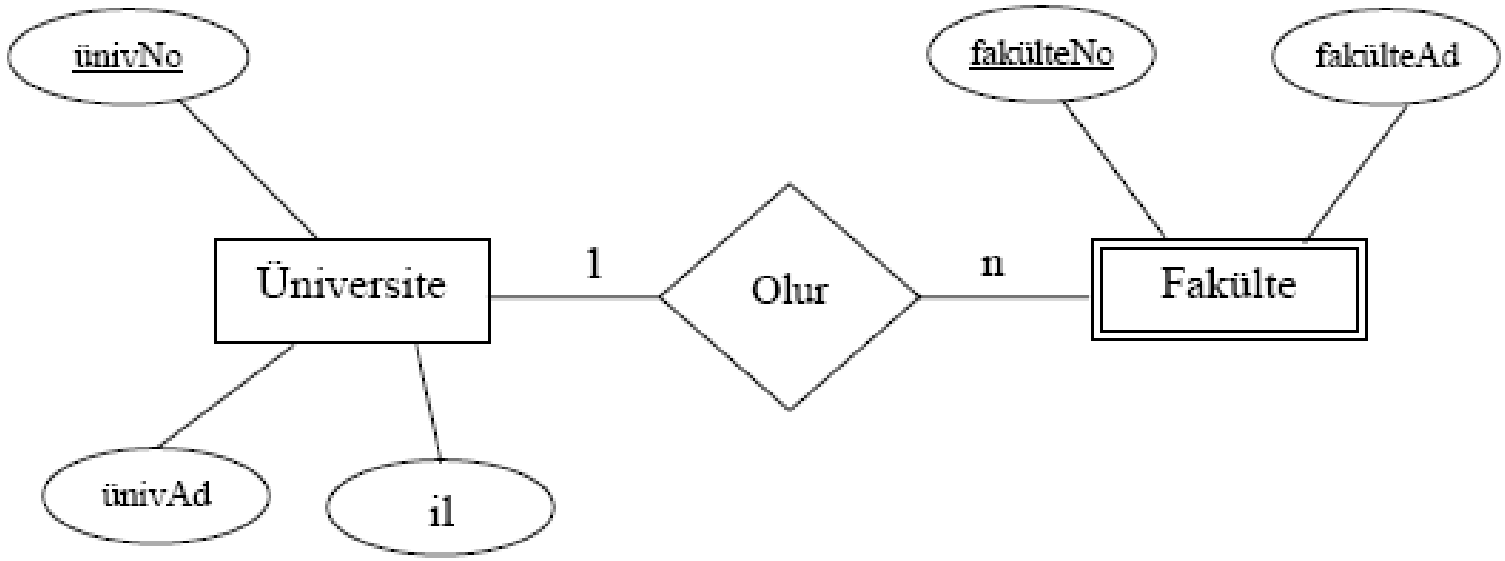


- KONSER (KONSNO, TARİH, SALON)
- ESER (ESNO, ESADI, ESTÜRÜ, BESTECİ)
- MÜZİSYEN (MZSNO, ADI, SOYADI)
- KATILIM (KONSNO, ESNO, MZSNO, GÖREVİ)

Zayıf varlık kümelerinin tablolara dönüştürülmesi

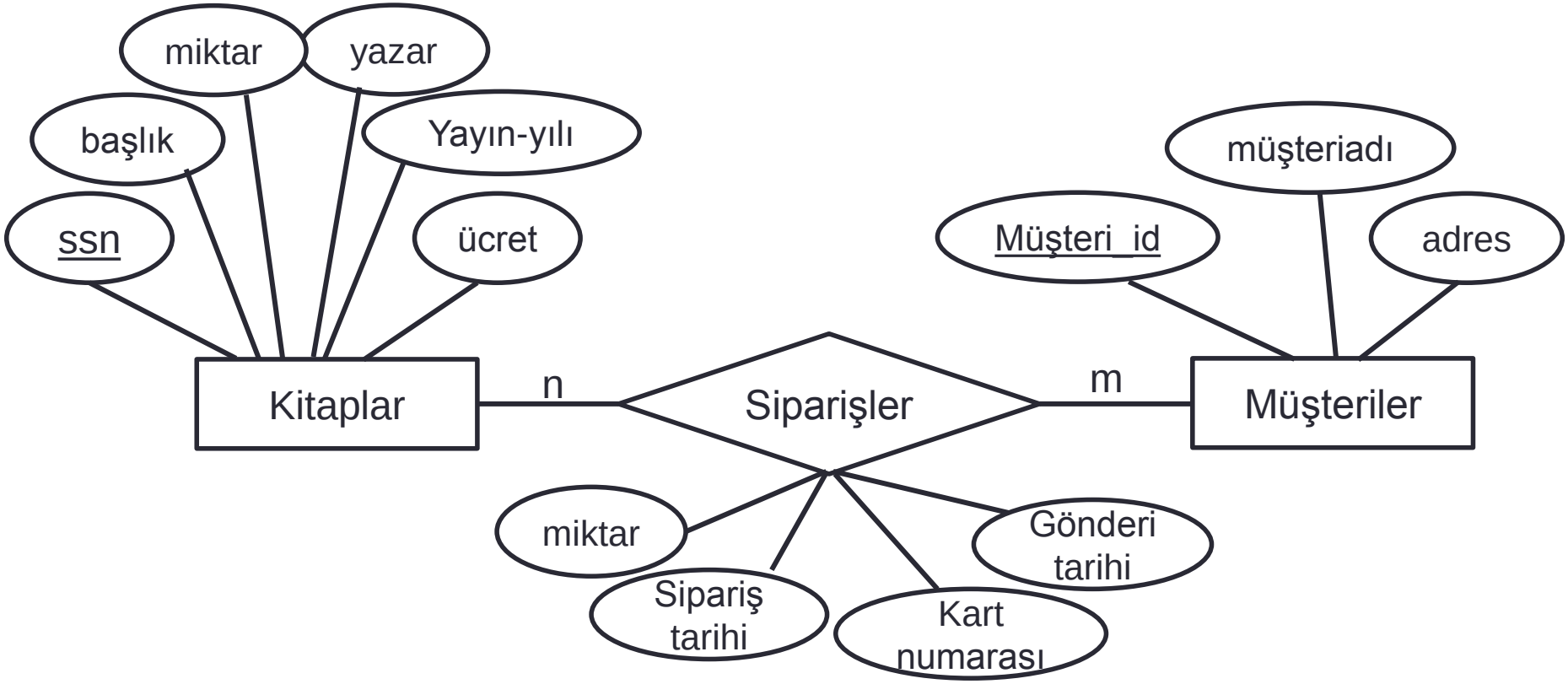
- Zayıf varlık kümeleri tablolara dönüştürülür. Ancak zayıf varlık kümesinin anahtar alanı kendi anahtar alanı ile birlikte ilişki kurduğu güçlü varlık kümesinin yabancı anahtarının birleşimidir.

- Örneğin, aşağıdaki şekildeki üniversite-fakülte 1-n ilişkisinde, üniversite varlık kümesi güçlü, fakülte varlık kümesi ise zayıf varlık kümesidir. Üniversite ve fakülte varlık kümeleri, ayrı ayrı üniversite ve fakülte varlık kümelerine dönüştürülür. Ancak, fakülte varlık kümesinin anahtar alanı, fakülteNo ile birlikte üniversiteNo alanlarından oluşturulur.



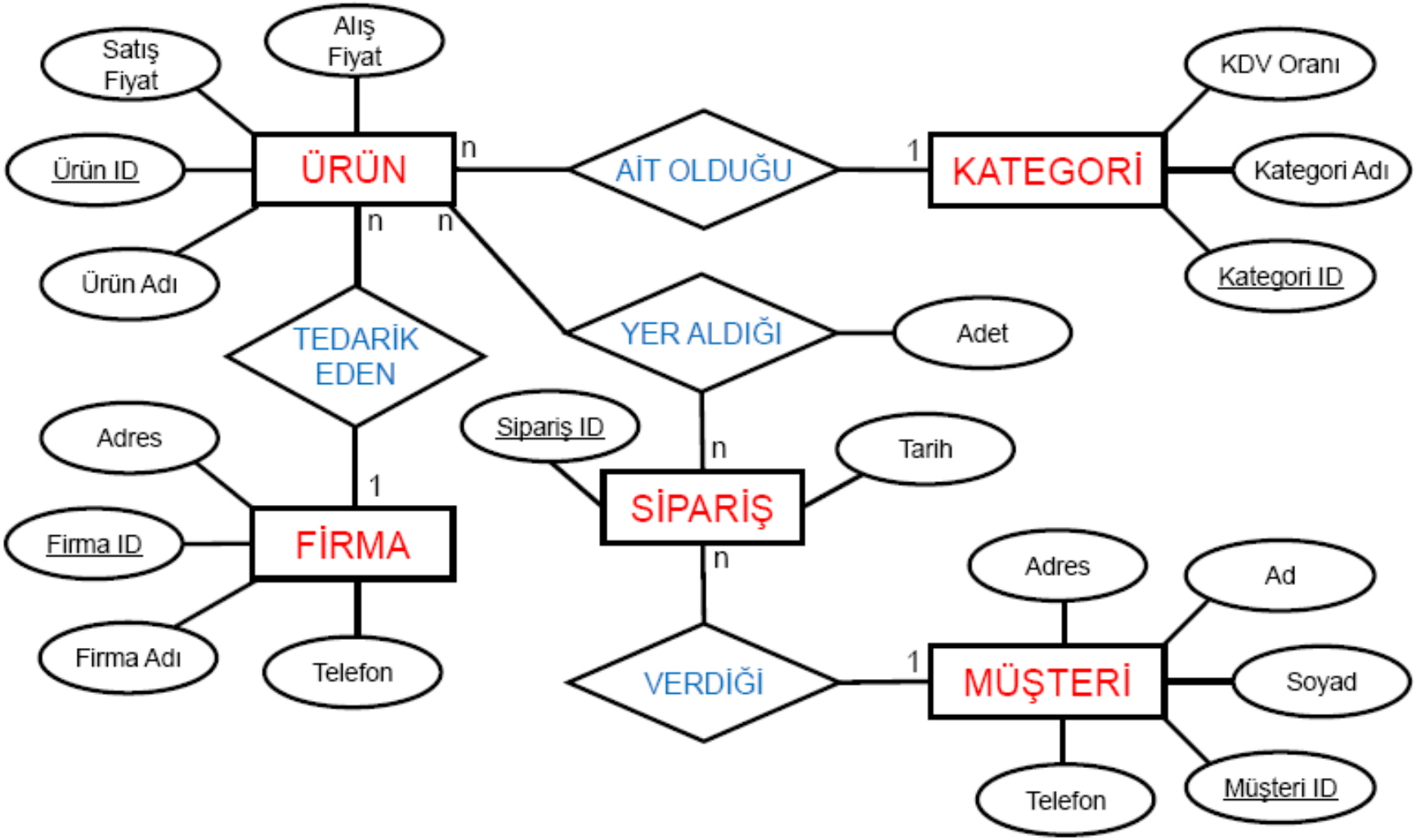
- Üniversite (ünivNo, ünivad, il)
- Fakülte (fakülteNo, ünivNo, fakülteAd)

Uygulama



Kitaplar(ssn, baslık, miktar, yazar, yayın_yılı, ücret)
Müşteriler(MüşteriId, adı, adres)
Siparişler(ssn, müşteriId, miktar, sipariş_tarihi, kart_no, gönderi_tarihi

Uygulama


Satış Fiyat
Alış Fiyat
Ürün ID
ÜRÜN
Ürün Adı
n
n
n
AİT OLDUĞU
1
KATEGORİ
KDV Oranı
Kategori Adı
Kategori ID
TEDARİK EDEN
YER ALDIĞI
Adet
Adres
Firma ID
FİRMA
Firma Adı
Telefon
1
Sipariş ID
SİPARİŞ
Tarih
n
n
VERDİĞİ
1
Adres
Ad
MÜŞTERİ
Soyad
Telefon
Müşteri ID

Çözüm

- ÜRÜNLER(ÜrünID, ÜrünAdı, AlışFiyatı, SatışFiyatı, KategoriID, FirmaID)
- KATEGORİLER(KategoriID, KategoriAdı, KDVOranı)
- FİRMALAR(FirmaID, MüşteriID,Tarih)
- MÜŞTERİLER(MüşteriID, Adı, Soyadı, Adresi, Telefonu)
- SİPARİŞLER(SiparişID, MüşteriID, Tarih)
- SİPARİŞ_ÜRÜN(SiparişID, ÜrünID, Adet)

Tablo Oluşturma

CREATE TABLO <tablo adı>
(<Tablonun alanın adı> <alanın tipi> <alanın kısıtlaması>...)

CREATE TABLO komutu kullanılırken tablonun her alanı (sütunu) için veri tipi ve başka bazı parametreleri vardır. Bazı alanlar (sütunlar) için kısıtlamalarının gösterilmesi gereklidir. Bu kısıtlamalar aşağıdaki şekilde sayılabilir;

1. Not Null, alan için bir değerin mutlaka girilmesi gerektiğini gösterir. (Bu parametre daha çok birincil anahtarlar ve eşsiz alanlar için kullanılır.)

2. Default, alan için bir başlangıç değeri verilmesinde kullanılır. Bu değer sabit bir değer olabileceği gibi, Null değeri de olabilir.

3. İsteğe bağlı olarak constraint anahtar kelimesi kullanılarak bir veya birden fazla kısıt kullanılabilir. Bu kısıtlar şunlardır.

    a. PRIMARY KEY (Birincil anahtar)
    b. UNIQUE (her kaydın bir alan için farklı bir değer olması gerektiğini gösterir.
    c. REFERENCES (başka bir tablonun alanıyla ilişki kurmak için

Veri Tipleri - Karakter Tabanlı (Metinsel)

- **char:** Unicode olmayan sabit uzunlukta karakter verisi saklamak için kullanılır. Boyutu değişken olmakla birlikte en fazla 8000 karakter veri saklayabilir.

- **varchar:** Unicode olmayan değişken uzunlukta karakter verisi saklamak için kullanılır. Belirlenmiş veri kapasitesi 8000 karakter olmasına rağmen (max) parametresi ile bu değer 2^31 byte' a kadar genişletilebilir.

- **text:** Daha önceki sql versiyonlarını desteklemek amacıyla Sql Server 2012 içerisinde hala yer almaktadır. Bu veri tipi yerine varchar(max) kullanın.

Veri Tipleri - Karakter Tabanlı (Metinsel)

- **nchar:** Sabit uzunlukta 4000 karakter unicode karakter verisi saklamak için kullanılır. Boyutu değişken olmakla birlikte kısa olan değerler atanan uzunluğa tamamlanır.

- **nvarchar:** Değişken uzunlukta unicode karakter verisi saklamak için kullanılır. Belirlenmiş maksimum uzunluk 4000 karakter olmasına rağmen (max) parametresiyle 2^31 byte'a kadar veri depolanabilir.

- **ntext:** text veri tipi gibi, bu veri tipide sadece eski versiyonları desteklemek amacıyla mevcuttur. Text tipinden tek farkı unicode veri saklayabilir.

Veri Tipleri - Nümerik (Sayısal)

- **bit:** Bir byte uzunluğunda tamsayı veri tipidir. Boyutu aklınızı karıştırmasın. Tablodaki ilk bit, bir byte büyüklüğünde yer kaplar, sonraki yedi bit aynı byte'ı kullanabilir.

- **tinyint:** 1 byte büyüklüğünde 0 ile 255 arasında değer alabilen tamsayı veri tipidir.

- **smallint:** 2 byte büyüklüğünde -32768 ile 32767 arasında değer alabilen tamsayı veri tipidir.

- **int:** 4 byte büyüklüğünde yaklaşık -2 milyar ile +2 milyar arasında değer alabilen tamsayı veri tipidir.

Veri Tipleri - Nümerik (Sayısal)

- **bigint:** 8 byte yüyüklüğünde -2^63 ile +2^63 arasında değer alabilen tamsayı veri tipidir.

- **decimal ve numeric:** Decimal ve nümeric veri tipleri adları farklı olmasına rağmen kullanımları aynıdır. Boyutu değişken olmakla birlikte -10^38 ile +10^38 arasında ondalık ve tamsayı türünde verileri saklayabilir.

- **float:** Boyutu ve doğruluğu (ondalık kısım duyarlılığı) aldığı parametreye göre değişen kayan noktalı sayılar için kullanılır.

- **real:** Float veri tipinin parametre olarak 24 değerini aldığında ortaya çıkan halidir. -3.40E+38 ile 3.40E+38 arasında değer alır.

Veri Tipleri - Parasal

- **smallmoney:** 4 byte uzunluğunda yaklaşık -214 000 ile 214 000 arasında parasal değerleri tutmak için kullanılır. Ondalık kısım duyarlılığı 4 basamaktır.

- **money:** 8 byte uzunluğunda yaklaşık -922 milyar ile 922 milyar arasındaki parasal değerleri tutmak için kullanılır. Ondalık kısım duyarlılığı 4 basamaktır.

Veri Tipleri - Tarih-Zaman

- **date:** Tarihleri YYYY-MM-DD şeklinde saklayan 3 byte uzunluğunda veri tipidir. 0001-01-01 ile 9999-12-31 tarihleri arasındaki tüm değerleri tutabilir.

- **smalldatetime:** Tarih ve zaman verilerini YYYY-MM-DD hh:mm:ss şeklinde saklayan 4 byte uzunluğunda veri tipidir. 1900-01-01 00:00:00 ile 2079-06-06 23:59:59 arasındaki tarih ve zaman değerlerini saklar.

- **datetime:** YYYY-MM-DD hh:mm:ss:mmm şeklinde tarih ve zaman verilerini tutan 8 byte uzunluğunda veri tipidir. 1753-01-01 00:00:00.000 ile 9999-12-31 23:59:59.999 arası değerlerini saklar.

Veri Tipleri - Tarih-Zaman

- **datetime2:** datetime ile arasındaki fark salise hassasiyetinin daha yüksek olmasıdır. 7 basamağa kadar salise hassasiyeti kullanıcı tarafından belirlenebilir. Kapladığı alan salise hassasiyetine göre değişir. (6-8 byte)

- **time:** Sadece saat verilerini hh:mm:ss:nnnnnnn şeklinde saklayan veri tipidir. Dateteime2 gibi salise hassasiyeti maksimum 7 basamaktır ve kullanıcı tarafından değiştirilebilir. (3-5 byte)

- **datetimeoffset:** Ülkelere göre değişen zaman farkını tutmak için kullanılır

SQL İlişkileri Oluşturma

MNo	Ad	Adres
53666	Ali	ali@gmail
53688	Aslı	aslı@gmail
53650	Ahemt	ahmet@gmail

UrunNo	UrunAd	Fiyat	Stok
537	A	100	1000
538	B	50	2600
536	C	20	1200
532	D	30	3000

```
CREATE TABLE musteri
(mno int,
 ad CHAR(20),
 adres CHAR(100))
```

```
CREATE TABLE urun
(Uno int,
 Uad CHAR(20),
 fiyat money,
sMiktar int)
```

SQL’de Primary Key

```
CREATE TABLE musteri
(mno int ,
 ad CHAR(20),
 adres CHAR(100),
PRIMARY KEY (mno))
```

```
CREATE TABLE urun
(Uno int,
 Uad CHAR(20),
 fiyat money,
 sMiktar int,
PRIMARY KEY (Uno))
```

SQL'de Foreign Key

Musteri

MNo	Ad	Adres
53666	Ali	ali@gmail
53688	Aslı	aslı@gmail
53650	Ahemt	ahmet@gmail

Urun

MNo	Urun No	UrunAd	Fiyat	Stok
53666	537	A	100	1000
53688	538	B	50	2600
53650	536	C	20	1200
53666	532	D	30	3000

Musteri_Urun

MNo	UNo	Miktar
53666	537	100
53688	538	150
53650	536	50
53666	532	100

```
CREATE TABLE  Musterı_Urun
(Mno int, Uno int, Uad CHAR(20), fiyat money, sMiktar int,
PRIMARY KEY (Mno,Uno),
FOREIGN KEY (Mno) REFERENCES Musteri
FOREIGN KEY (Uno) REFERENCES Urun )
```