

Veri Tabanı Sistemleri

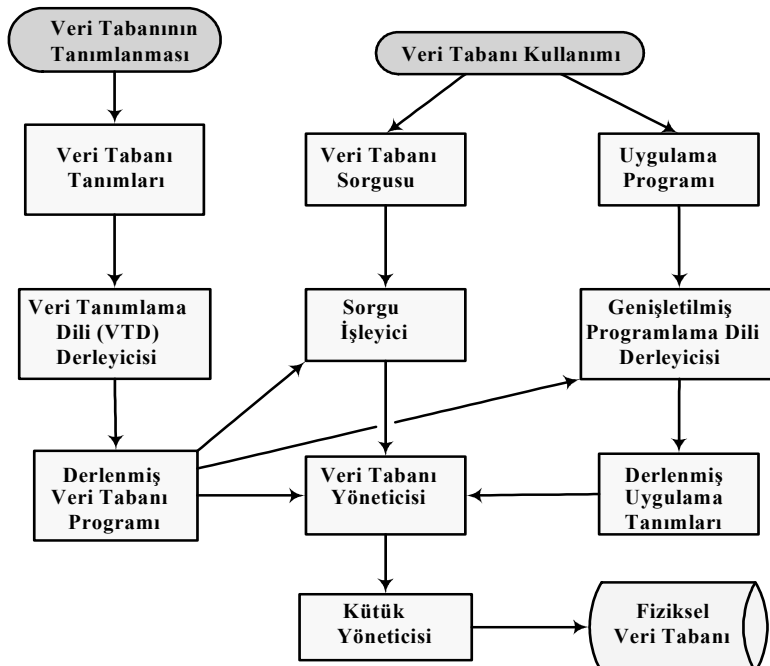
1.1. Giriş

- Veri tabanı sistemleri, bilgisayar ortamında büyük boyutlu düzenli veri kümelerinin yönetimi için yaygın biçimde kullanılan bir araçtır.
- Veri tabanını "bir ya da birkaç uygulamada kullanılmak için, gereksiz yinelemelerden arınmış olarak, düzenli biçimde bilgisayar belleklerinde saklanan birbirleriyle ilişki veriler topluluğu" olarak tanımlamak mümkündür.
- Veri tabanlarını daha iyi anlatabilmek için, birkaç satırlık bir tanım vermek yerine özelliklerinin sıralanması daha uygun bir yaklaşım olur.
 - Veri tabanı belirli bir kuruluşun birçok uygulamasında kullanılan, birbirleriyle ilişkili, işletimsel verilerinden oluşur.
 - Veri tabanındaki veriler gereksiz yinelemelerden arınmış olarak, düzenli bir biçimde bilgisayar belleklerinde saklanır
 - Veri tabanında saklanan veriler değişmeyen statik veriler değildir.
 - Veri tabanındaki veriler üzerinde merkezi bir denetim vardır (VTYS).
 - Veri Tabanı Yönetim Sistemi aracılığıyla, veri tabanının bilgisayar belleklerindeki fiziksel yapısı kullanıcılardan gizlenir.

1.2. Veri Tabanı Yönetim Sisteminin Temel Bileşenleri

- Veri tabanı sistemlerini tanımlamak, yaratmak, kullanmak, değiştirmek ve veri tabanı sistemleri ilgili her türlü işletimsel gereksinimleri karşılamak için kullanılan yazılım sistemine Veri Tabanı Yönetim Sistemi denir.
- VTYS'nin en temel bileşenleri aşağıda açıklanmaktadır.
 - a. Veri Modeli, Veri Tanımlama Dili (VTD) ve VTD Derleyicisi
Veri Sözlüğü
 - b. Sorgu Dili ve Sorgu İşleyici
 - c. Veri İşleme Dili ve Derleyiciler
 - d. Veri Tabanı Yöneticisi
Bellek Yöneticisi
Kütük Yöneticisi
Yastık (buffer) Yöneticisi
Hareket Yöneticisi

Bir Veri Tabanı Sisteminin İşlevsel Çizeneği



➤ Veri Tabanı Sorumlusu (*Data Base Administrator*) : veri tabanının tümünden sorumlu olan kişi ya da kişiler :

- mantıksal düzeyde yapılan düzenlemeler, oluşturulan yapılar ve her yapıda hangi verilerin yer aldığı,
- her verinin türü, uzunluğu, varsa varsayılan değeri ve diğer özellikleri,
- veriler arası ilişkiler ve her türlü kısıtlamalar,
- fiziksel veri yapıları ile ilgili kimi tercihler ve parametreler,
- kullanıcı tanımları ve kullanıcıların hangi veriler üzerinde hangi işlemleri yapmaya yetkili olduklarına ilişkin tanımlar.

1.3. Veri Tabanı Yaklaşımının Yararları

➤ Bir bilişim sistemi (*information system*) kapsamında kullanılan ve uygulamalar tarafından paylaşılan ortak verilerin düzenlenmesi, saklanması ve kullanılması için iki yaklaşımdan söz edilebilir.

- Geleneksel yaklaşım.
- Veri tabanı yaklaşımı.
- VT Yaklaşımının Yararları:
- Veri Tekrarının Azaltılması
- Veri Bütünlüğünün Sağlanması.
- Veri Paylaşımının Sağlanması.
- Kullanımın Yaygınlaştırılması.
- Veri Gizliliği ve Güvenliğinin Sağlanması.
- Standartların Uygulanabilmesi.

1.4. Veri Tabanı İle İlgili Diğer Kavramlar

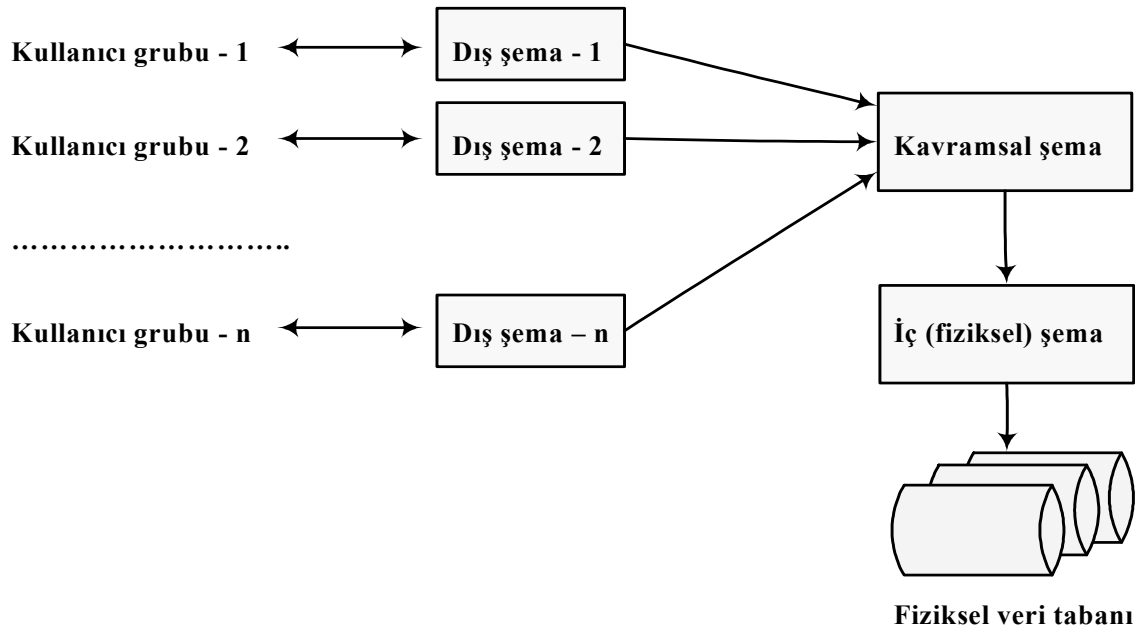
1.4.1. Kullanıcı Türleri ve Kullandıkları Olanaklar

- Veri tabanı sorumlusu
- Uygulama programcıları
- Sorgu dili kullanıcıları
- Uygulama programı kullanıcıları
- Rasgele kullanıcılar

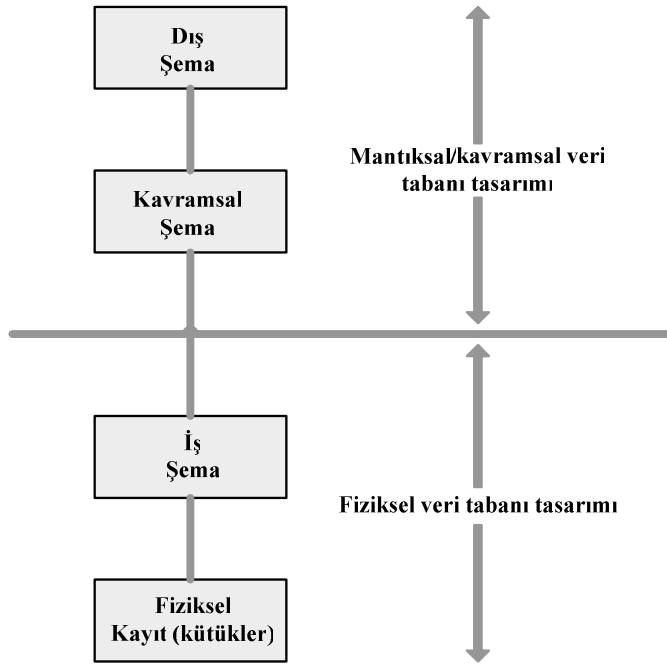
1.4.2. Soyutlama Düzeyleri ve Veri Bağımsızlığı

- Fiziksel veri bağımsızlığı
- Mantıksal veri bağımsızlığı

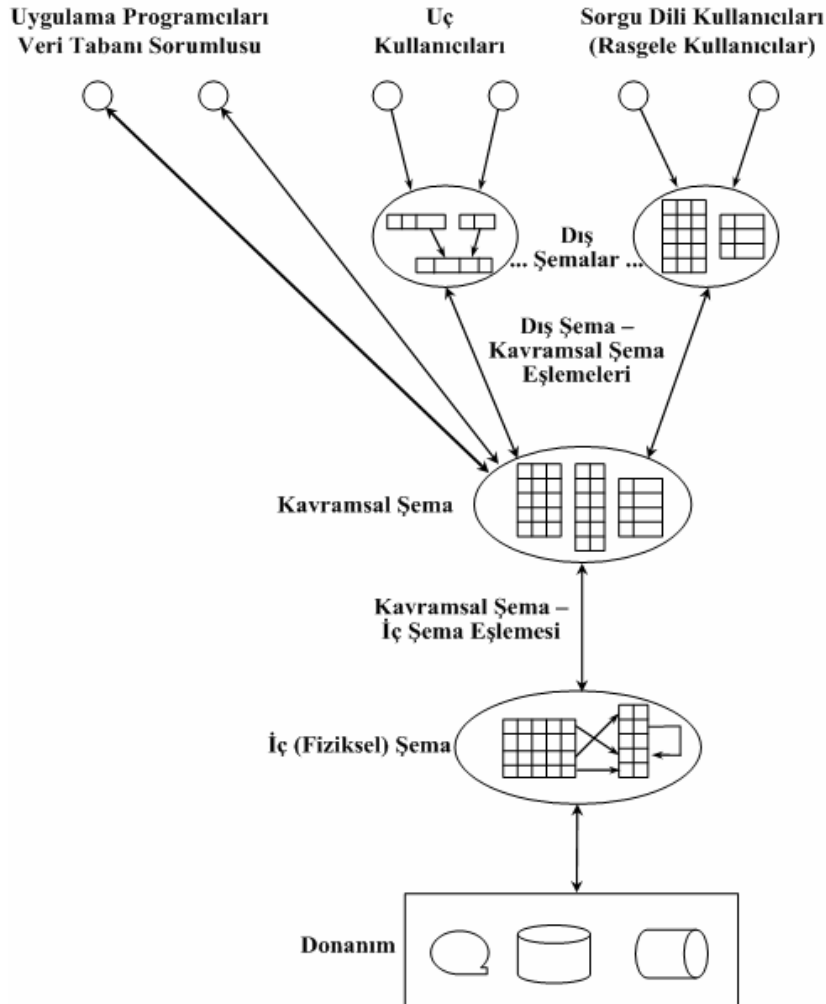
Veri Tabanı Sistemi Soyutlama Düzeyleri



Veri Modelleme ve ANSI-SPARC Mimarisi



Veri Tabanı Yönetim Sistemi İçin Örnek Bir Mimari



Bölüm2: Varlık-Bağıntı Modeli

2.1. Veri Modelleri ve Varlık-Bağıntı Modeli

- Verileri mantıksal düzeyde düzenlemek için kullanılan kavramlar, yapılar ve işlemler bütününe veri modeli denir. Her Veri Tabanı Yönetim Sistemi (VTYS) belirli bir veri modelini kullanır.
- Veri modeli kullanılarak veri tabanının kavramsal ve dış şemaları oluşturulur.
- Geçmişte ve günümüzde yaygın kullanılan veri modellerini 4 grupta toplamak mümkündür. Bu 4 temel veri modeli yaklaşımı şunlardır:
 - Sıradüzensel veri modeli (*Hierarchical data model*)
 - Ağ veri modeli (*Network data model*)
 - İlişkisel veri modeli (*Relational data model*)
 - Nesneye-yönelik veri modeli (*Object-oriented data model*)
- Varlık-bağıntı modeli, ya da kısaca E-R modeli (*Entity-Relationship model*) 1976 yılında P.P. Chen tarafından geliştirilen bir modeldir. Varlık-bağıntı modeli çok bilinen, VTYS'den bağımsız veri çözümleme ve modellemede çok kullanılan araçlardan biri, belki de birincisidir.
- Tasarım adımının güçlüğü : tasarımcı, programcı ve kullanıcıların verilere farklı bakış açılarıdır.
- Kuruluşun çalışmaları ve gereksinimleri konusunda ortak bir anlayışa varmak için farklı kişiler arasındaki iletişimi sağlayacak bir modelleme aracına ihtiyaç vardır.
- ER modelleme VT tasarımı için kullanılan bir yukarıdan aşağıya bir yaklaşımdır.
- Modelin temel kavramları varlıklar, bağıntılar, nitelikler ve kısıtlamalardır. Ayrıntıya inildiğinde bunlara yenileri eklenir.
- ER çizimlerini için, Chen gösterimleri yerine, daha yaygın olan UML gösterimleri (notasyonu) kullanılacaktır.

2.2. Varlık-Bağıntı Modelindeki Başlıca Kavram ve Yapılar

2.2.1. Varlık, Varlık Kümesi ve Nitelikler

- Varolan ve benzerlerinden ayırddilebilen her şeye, her nesneye bir varlık (*entity*) denir.
- Aynı türden benzer varlıkların oluşturduğu kümeye ise varlık kümesi (*entity set*) adı verilir
- Bir varlık kümesindeki varlıkların özelliklerini göstermek ve varlıkları birbirinden ayırtmak için nitelikler (*attributes*) kullanılır. Başka bir deyişle her varlığa belirli sayıda nitelik değeri eşlenir.
- Her niteliğin bir değer alanı (*domain*) vardır. Değer alanı ilgili niteliğin olurlu değerlerinin tümünü içeren bir kümedir.

Nitelikler

- Varlık ya da bağıntı türünün bir özelliği.
- Nitelik alanı : bir ya da birkaç niteliğin geçerli değerlerinin kümesi.
- Alanların alt alanları olabilir : örneğin adres alanı kent, mahalle, cadde, posta kodu ..vb. alanları içerebilir.
- Nitelikler yalın (atomik), ya da birleşik nitelikler olabilir.
- Kullanıcı görüşüne göre bir nitelik yalın ya da birleşik bir nitelik olarak modellenebilir.
- Tek değerli nitelik : her varlık/bağıntı için tek bir değer alabilen nitelik.
- Çok değerli nitelik : bir varlık/bağıntı için birden çok değer alabilen nitelik.
- Türetilmiş nitelik : değeri diğer niteliklerden (bir ya da birkaç) türetilen nitelikler.
- Bu niteliklerin aynı varlık/bağıntı türünün nitelikleri olması gerekmez.

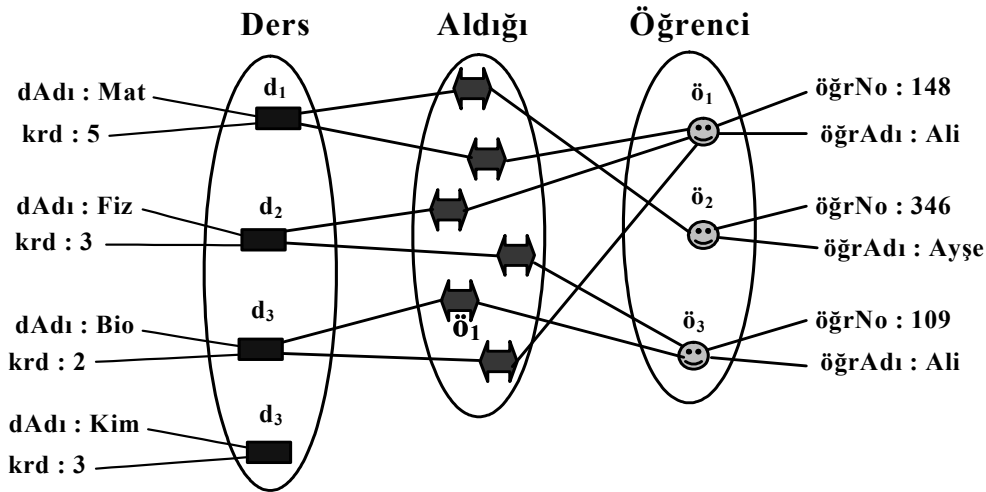
2.2.2. Bağntı ve Bağntı Kümesi

- Bağntı iki ya da daha çok sayıda varlığın birleşmesi, bir araya gelmesi, aralarında ilişki kurulmasıdır.
- Aynı tür, benzer bağntıların kümesine bağntı kümesi (*relationship set*) adı verilir. Matematiksel olarak $E_1, E_2, E_3, \dots, E_n$ varlık kümeleri arasındaki bir R bağntısı aşağıdaki gibi tanımlarır:

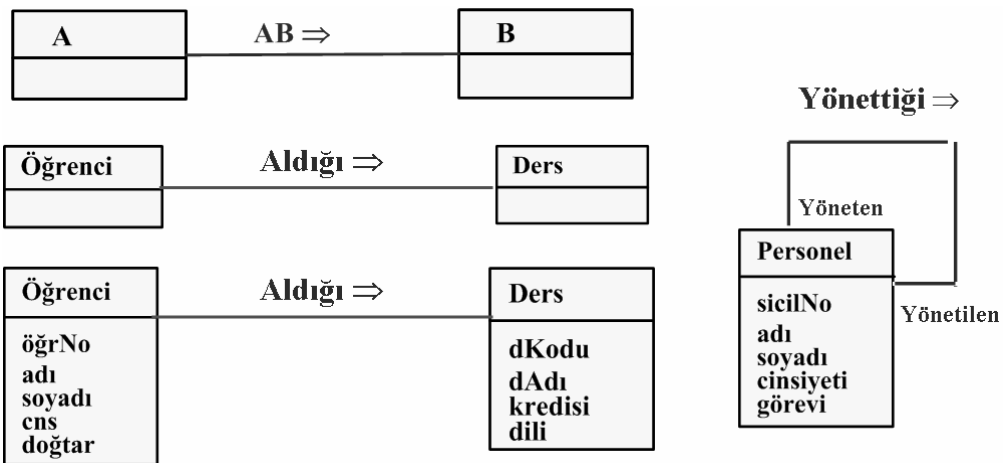
$$R = \{ (e_1, e_2, e_3, \dots, e_n) : e_1 \in E_1, e_2 \in E_2, e_3 \in E_3, \dots, e_n \in E_n \}$$

- Bağntı kümeleri ikili, üçlü, dördü, .. vb. olabilir. Ancak karşılaşılan ve kullanılan bağntı kümelerinin büyük çoğunluğu ikilidir. Üçlü, dördü, ..vb. bağntı kümesi yerine, istenirse birkaç ikili bağntı kümesi kullanılabilir.
- Aralarında bağntı kurulan varlıklardan her birinin bağntıdaki işlevine varlığın rolü denir. Roller dolaylı biçimde anlaşıldığı için çoğunlukla açıkça belirtilmez. Oysa kişi1 ve kişi2 arasında kurulan evlilik bağntısında; personel1 ve personel2 arasında kurulan yönetici bağntısında rollerin belirtilmesine gerek vardır.

Varlık ve Bağntı Kümeleri Örneği



Varlık-bağntı çizeneğinde varlık kümesi, nitelik ve bağntı kümesinin gösterilmesi (UML notasyonu):



2.2.3. Bağntı Kümeleri ile İlgili Sınırlamalar ve Bağntı Türleri

- Bağntı kümeleri ile ilgili olarak bir dizi sınırlama tanımlanabilir. Bu sınırlamaların en önemlileri, aralarında bağntı kurulan varlık kümeleri arasındaki eşlemelerle ilgili sayısal sınırlamalardır.
- A ve B varlık kümeleri arasında tanımlanan (A ve B varlık kümeleri aynı da olabilir), A'dan B'ye AB bağntı kümesi, eşleme sınırlamaları açısından aşağıdaki 4 türden birinde olabilir.
 - Bire-bir (*one-to-one*) (1 : 1). Bire-bir bağntıda, her a ile en çok bir b; her b ile de en çok bir a arasında bağntı kurulabilir ($a \in A, b \in B$).
 - Bire-birçok (*one-to-many*) (1 : n). Bire-birçok bağntıda her a ile sıfır, bir ya da birçok b; her b ile de en çok bir a arasında bağntı kurulabilir.
 - Birçoğa-bir (*many-to-one*) (n : 1). Birçoğa-bir bağntıda her a ile en çok bir b; her b ile de sıfır, bir ya da birçok a arasında bağntı kurulabilir.
 - Birçoğa-birçok (*many-to-many*) (n : m). Birçoğa-birçok bağntıda her a ile sıfır, bir ya da birçok b; her b ile de sıfır, bir ya da birçok a arasında bağntı kurulabilir.
 - Varlık kümeleri ile ilgili diğer bir sınırlama da varolma bağntılılığıdır (*existence dependency*). Eğer A ve B varlık kümeleri arasında bire-bir, ya da A'dan B'ye bire-birçok bir R bağntısı varsa ve de bir b'nin varolması bu b ile bir a arasında r bağntısının kurulmuş olmasına bağlı ise, ya da başka bir deyişle r bağntısı ile bir a'ya bağlı olmayan b'ler varolamıyorsa, b a'ya varolma bağntılıdır denir.

Bu durumda:

- a) üstün (*dominant*) ya da birincil varlık,
- b) bağntılı (*subordinate*) ya da ikincil varlık

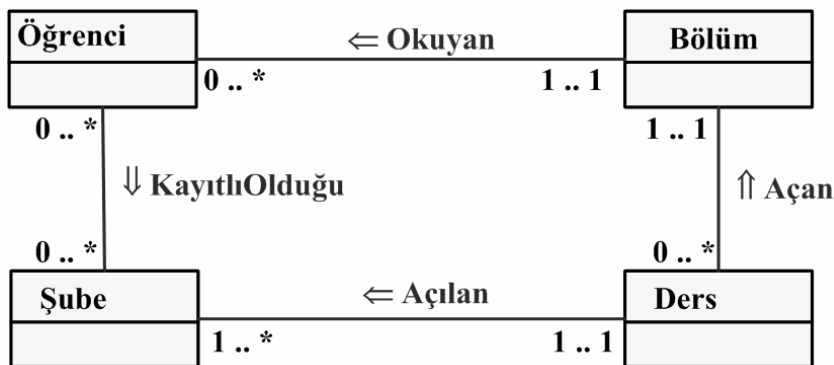
olarak nitelenir.

- UML notasyonunda bağntıların çoğulculuk (*multiplicity*) sınırlaması ve varolma bağntılılığı, bağntının her iki tarafında aşağıdaki gibi gösterilir:

0..1	Sıfır ya da bir
1..1	Tam olarak bir
0..*	Sıfır, bir ya da birçok
1..*	Bir ya da birçok
n..m	En az n, en çok m

- Çoğulculuğun bileşenleri : en büyük değer (*cardinality*) ve katılım'dır:
katılım göstergesi \rightarrow n..m \leftarrow en büyük değer

➤ Örnek:



Bağıntı Örnekleri

A	AB \Rightarrow	B
	0..1	0..1

Bire-bir bağıntı;
varolma bağımlılığı yok

A	AB \Rightarrow	B
	1..1	0..1

Bire-bir bağıntı;
B A'ya varolma bağımlı

A	AB \Rightarrow	B
	0..1	0..*

Bire-birçok bağıntı;
varolma bağımlılığı yok

A	AB \Rightarrow	B
	1..1	0..*

Bire-bir bağıntı;
B A'ya varolma bağımlı

A	AB \Rightarrow	B
	1..1	1..*

Bire-birçok bağıntı;
B A'ya varolma bağımlı
Her A'ya bağlı en az bir B bulunmalı

A	AB \Rightarrow	B
	0..*	0..*

Birçoğa-birçok bağıntı;

A	AB \Rightarrow	B
	0..*	1..*

Bire-birçok bağıntı;
Her A'ya bağlı en az bir B bulunmalı

A	AB \Rightarrow	B
	0..50	1..6

Birçoğa-birçok bağıntı bağıntı;

Bağıntı nitelikleri

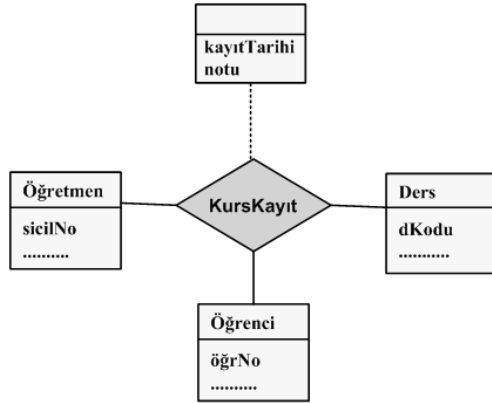
- Birçoğa-birçok bağıntıların tanımlayıcı nitelikleri de bulunabilir. Eğer birçoğa-birçok bir bağıntıda, aralarında bağıntı kurulan varlıkların değil de, bağıntının bir özelliğini gösteren bir nitelik varsa, bu bir bağıntı niteliğidir.
- Bağıntı niteliklerinin Varklık-bağıntı çizeneğinde gösterilmesi:

Öğrenci	Aldığı \Rightarrow	Ders
öğrNo adı soyadı cns doğtar	0..*	0..*
		dKodu dAdı kredisi dili
	yarıyılı zorSeç notu	

- Bağntı niteliklerini birer varlık kümesi olarak düşünmek ve bağntıyı ikili bir bağntı yerine üçlü, dörü, .. vb. bir bağntı olarak görmek de mümkündür.

Üçlü, Dörü, ... Bağntılar

- İkili bağntılar anlaşılması ve yönetimi kolay bağntılardır. Kullanılan bağntıların büyük çoğunluğu da ikili bağntılardır. Ancak zaman zaman üç ya da daha çok varlık kümesi arasında bağntılar da tanımlamak gerekebilir.
- Örnek:



- Üçlü, dörü, ... bağntıların sakıncaları

2.2.4. Anahtarlar ve Güçlü/Zayıf Varlık Kümeleri

- Süper anahtar (*superkey*). Değerleri ile bir kümedeki varlıkları/bağntıları ayırtmayı sağlayan nitelik ya da nitelik grubuna bu varlık/bağntı kümesinin süper anahtarı denir. (*Superkey* deyiimi *Super set of key* deyiiminin kısaltmasıdır)
- Anahtar adayı (*candidate key*) ya da kısaca anahtar. Eğer bir varlık/bağntı kümesinin bir süper anahtarının (K) hiçbir öz altkümesi bu varlık/bağntı kümesinin süper anahtarı değilse, K bu varlık/bağntı kümesinin anahtar adayı ya da kısaca anahtarıdır.
- Süper anahtar biricik belirleme ya da ayırtma özelliğini sağlar. Anahtar adayı ise biricik belirlemeye ek olarak en küçük olma (gerekenden fazla nitelik içermeme) özelliğini de sağlar. Anahtar dendiğinde anlaşılması gereken anahtar adaydır.
- Sadece “Anahtar” dendiğinde anlaşılması gereken anahtar adaydır.
- Eğer bir varlık kümesinin niteliklerinden en az bir anahtar oluşturulabiliyorsa, bu varlık kümesine güçlü (*strong*) varlık kümesi denir. Buna karşılık eğer bir varlık kümesinin niteliklerinin tümü alınsa bile bir anahtar oluşturulmuyorsa bu varlık kümesine zayıf (*weak*) varlık kümesi denir.
- Zayıf bir varlık kümesinde, niteliklerin değerleri ile varlıkları birbirinden ayırtmak mümkün değildir. Zayıf bir varlık kümesinin anlamlı olabilmesi için:
 - Bu varlık kümesi ile güçlü bir varlık kümesi arasında bire-bir ya da (güçlünden-zayıfa) bire-birçok bir bağntı bulunması,
 - Zayıf varlıklar için bu bağntının varolma bağımlılığı oluşturması,
 - Zayıf varlık kümesinin nitelikleri arasında, aynı güçlü varlığa bağlı zayıf varlıkları birbirinden ayırtmayı sağlayan bir nitelik grubunun (*discriminator*) bulunması gerekir.
- Zayıf bir varlığın anahtarı, bağlı olduğu üstün varlığın anahtarına ayırıcı nitelikler eklenerek elde edilir.

Bağıntı Kümelerinin Anahtarları

➤ Bire-bir bağıntı kümesinin anahtarı

A	AB ⇒	B
	0..1	0..1

➤ Bire-birçok bağıntı kümesinin anahtarı

A	AB ⇒	B
	0..1	0..*

➤ Birçoğa-bir bağıntı kümesinin anahtarı

A	AB ⇒	B
	0..*	0..1

➤ Birçoğa-birçok bağıntı kümesinin anahtarı

A	AB ⇒	B
	0..*	0..*

➤ Birçoğa-birçok bağıntıların ayırıcı niteliklerinin anahtarda yer alıp almaması

➤ Üçlü, dörtlü, ... bağıntı kümelerinin anahtarı

Modellemedeki Seçenekler

➤ Varlık-bağıntı modelinin sunduğu 3 temel kavram varlık kümesi, nitelik ve bağıntı kümesi kavramlarıdır.

➤ Gerçek dünyayı soyutlayarak semantik modelleme yapmak için kullanılabilen bu üç kavram birbirinden kesin çizgilerle ayrılmamaktadır.

➤ Varlık kümesi ile niteliğin, ya da varlık kümesi ile bağıntı kümesinin birbirinin yerine kullanılması; aynı konuda değişik düzenlemeler yapılması olasıdır.

➤ Birkaç somut örnek:

• “Telefon numarası” kişinin bir niteliği mi, ayrı bir varlık kümesi mi?

• “Ad” kişinin bir niteliği mi, ayrı bir varlık kümesi mi?

• “Banka hesabı” banka şubesi ve müşteri varlık kümeleri arasında bir bağıntı mı, ayrı bir varlık kümesi mi?

• “Yönetici” personel varlık kümesinin bir niteliği mi, personel varlık kümesi içinde bir bağıntı mı?

2.3. Genelleme ve Kümeleme

Genelleme / Özelleştirme

➤ Varlık kümeleri arasında özel bağıntılar da bulunabilir. Bunların en önemlisi ait olma bağıntısıdır. Alt düzey varlık kümeleri genellenerek bir üst düzey varlık kümesi oluşturulabilir (ISA, is-a bağıntısı; bire-bir bağıntısının özel bir durumu)

➤ ISA bağıntısı üst düzey bir varlık türü ile bunun alt türleri arasında kurulan özel bir bağıntıdır. ISA bağıntısında üst düzey varlığın nitelikleri kalıtım yoluyla alt düzey varlık türlerine geçer.

➤ Üstsınıf ve altsınıf kavramları

➤ Üstsınıfın bazı altsınıfları örtüşebilir.

➤ Tür sıradüzeni

➤ Nitelik kalıtımı

➤ Bir altsınıfın birden çok üstsınıfı olabilir : çoklu kalıtım.

➤ Özelleştirme süreci : yukarıdan-aşağıya yaklaşım.

➤ Genelleme süreci : aşağıdan-yukarıya yaklaşım.

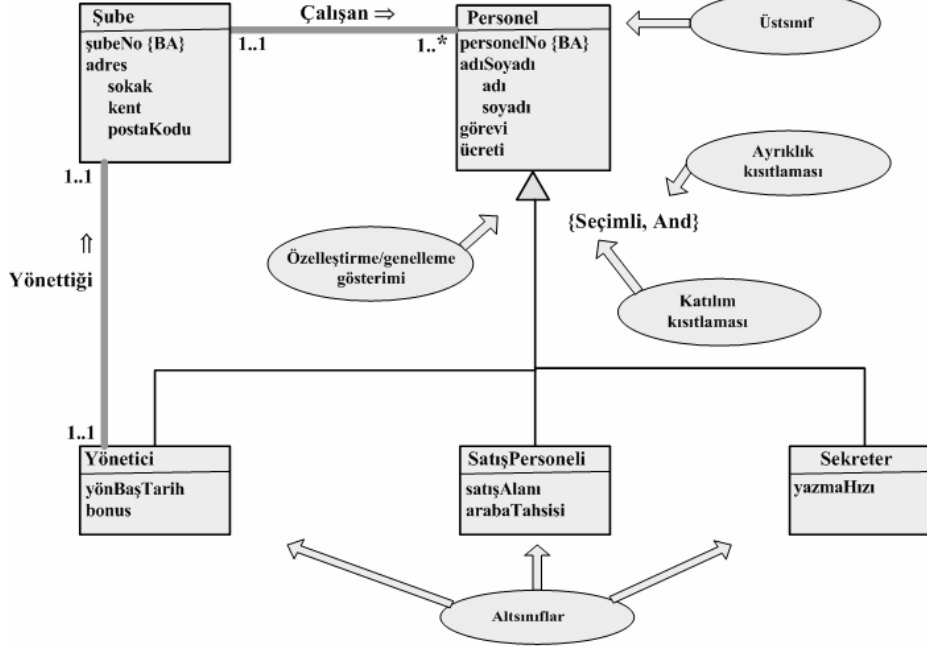
Genelleme/Özelleştirme ile İlgili Kısıtlamalar

➤ Katılım kısıtlaması : zorunlu ya da seçimli (*mandatory or optional*).

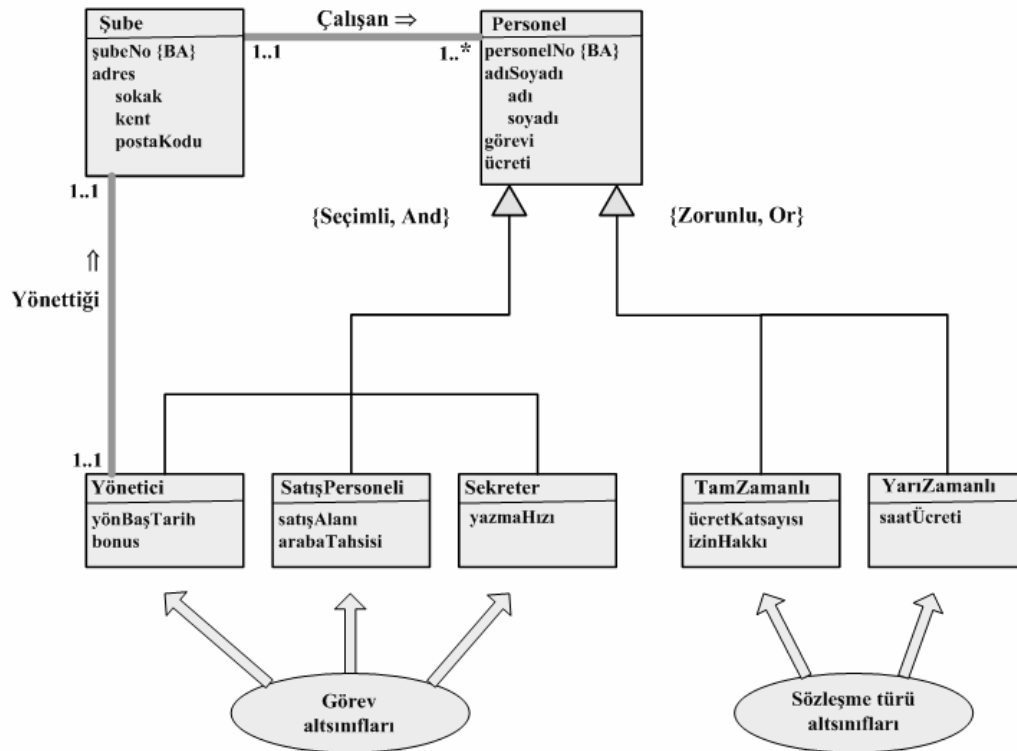
1. Zorunlu katılım : üstsınıfın her ögesi mutlaka bir atsınıfta yer almalıdır.

2. Seçimli katılım : üstsınıfın bir ögesi hiçbir altsınıfta yer almayabilir.
 - Ayrık olma/örtüşme kısıtlaması : (*disjoin/nondisjoin*)
1. Ayrık altsınıflar : üstsınıfın bir ögesi ancak bir altsınıfta yer alabilir (OR).
2. Örtüşen altsınıflar : üstsınıfın bir ögesi birden çok altsınıfta yer alabilir (AND).

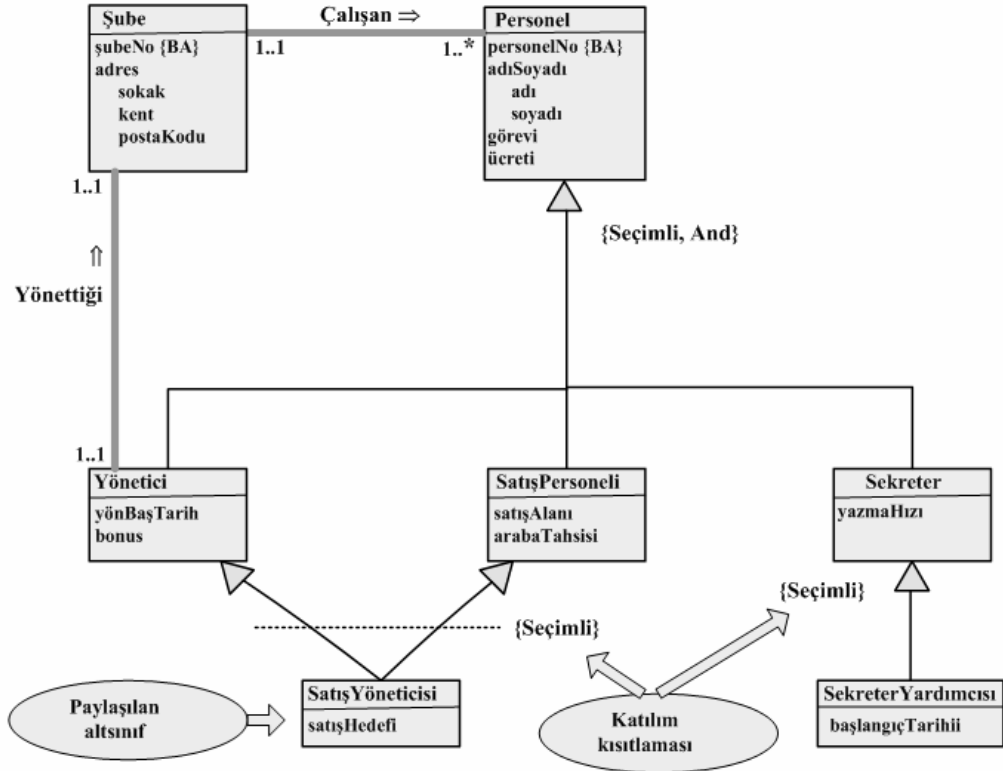
Personel Varlık Türünün Görev Türüne Göre Özelleştirilmesi (1)



Personel Varlık Türünün Görev Türüne Göre Özelleştirilmesi (2)

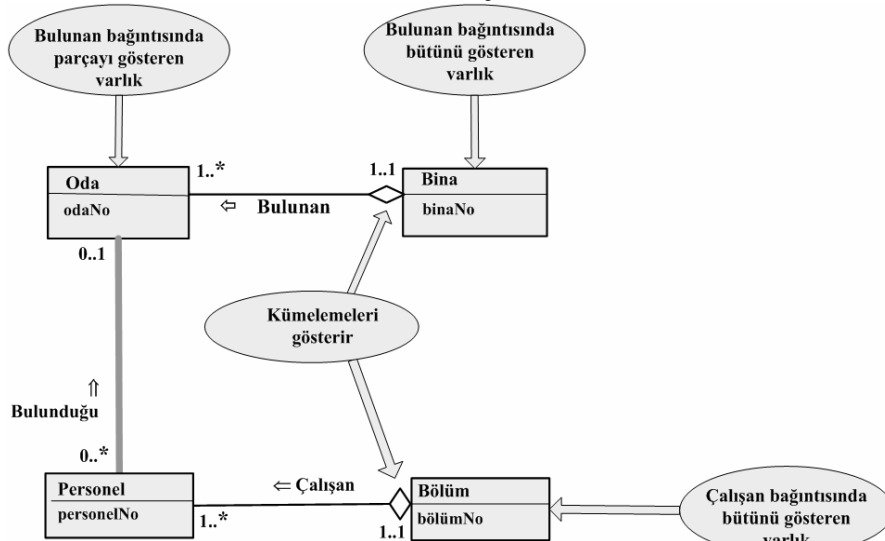


Personel Varlık Türünün Görev Türüne Göre Özelleştirilmesi (3)



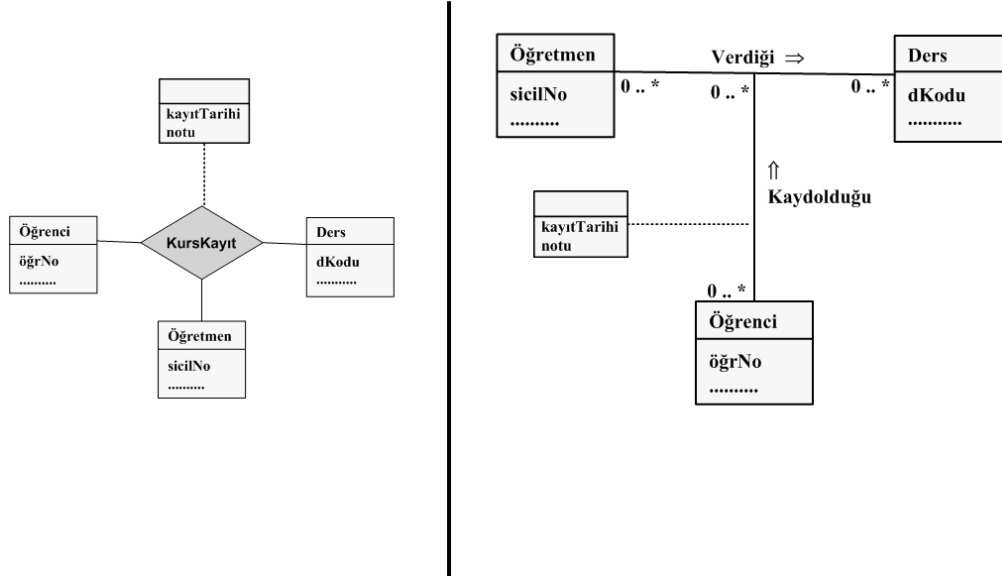
Kümeleme (*Aggregation*)

- Varlık-bağıntı modelinde veri modellemeyi kolaylaştıran kavramlardan biridir.
 - Biri bütün olan, diğeri ise bütünü oluşturan bir parça olan iki varlık türü arasındaki “bulunur/sahiptir” (*has a*) ya da “parçasıdır” (*is-part-of*) bağıntısını gösteririr.
 - Örneğin Bina-Oda arasındaki “Bulunan” bağıntısı (her binada bir ya da birçok oda bulunur); Bölüm-Personel arasındaki “Çalışan” bağıntısı (her bölümde çalışan sıfır, bir, ya da birçok personel vardır; personel bölümü oluşturan öğelerden biridir) kümeleme örnekleridir.
 - Kümelemede bir parça birden çok bütünün içinde yer alabilir. Örneğin bir personel (değişik zamanlarda) birden çok bölümde görev yaptıysa, birden çok bölümle bağıntılı olabilir (birden çok bütünde yer alabilir).
 - Kümeleme bağıntılarının ER çizeneklerinde gösterilmesi (Çizim 3.9).
 - Kümelemenin Chen terminolojisindeki anlamı



Kümeleme (Aggregation) : İkinci Anlamı

- İki varlık kümesi ile aralarındaki birçoğa-birçok bağıntıyı kümeleyerek bir bütün olarak düşünmek mümkündür. Bu durumda bu kümeyle varlık kümeleri ya da diğer kümeler arasında bağıntılar tanımlanabilir.
 - Bu yaklaşım, ikili bağıntıların iki varlık kümesi yanında, bir varlık kümesi ile bir küme (aggregation) ya da iki küme (aggregations) arasında kurulmasına da olanak sağlar.
 - Bunun en önemli uygulaması bir üçlü bağıntı yerine iki ikili bağıntı tanımlanabilmesidir.



Bölüm3: İlişkisel Model

- Temel düşünce: veri tabanındaki tüm veri ve bağıntıları insanların en çok kullandıkları, alışık oldukları bir yapı olan çizelgeler biçiminde göstermek.

3.1. Temel İlişki Yapısı

- İlişki (relation) kavramı özde matematiksel bir kavramdır:
 D_1, D_2, \dots, D_k alanlar (domains) yığını düşünelim. Her alan bir değerler kümesidir. Bu alanların Karteziyen çarpımı $D_1 \times D_2 \times \dots \times D_k$ ile gösterilir ve 1. elemanı D_1 , 2. elemanı D_2 , ..., k. elemanı ise D_k alanından alınan $\langle d_1, d_2, \dots, d_k \rangle$ k-çoklular kümesidir. Bu k alan üzerinde tanımlanan her r ilişkisi, bu alanların Karteziyen çarpımının bir altkümesidir: $r \subseteq D_1 \times D_2 \times \dots \times D_k$
 - İlişki iki boyutlu bir çizelge olarak görülebilir. Çizelgenin her kolonuna bir nitelik (attribute) atanır. Çizelgenin her satırı ise bir çokluyu (1. elemanı 1. niteliğin, 2. elemanı 2. niteliğin, .., k. elemanı ise k. niteliğin değer alanından alınan ve k değerden oluşan bir çoklu) gösterir. Çizelge, niteliklerin değer alanlarının Karteziyen çarpımının bir altkümesidir.
- Her ilişkinin biricik bir adı vardır. İlişki şeması, ilişkinin adı ile niteliklerini gösteren $R(A_1, A_2, \dots, A_k)$ biçimindeki bir tanımdır. R ilişki şemasına göre gerçekleşen her r ilişkisi (ya da ilişki olgusu), tanıma uygun k-çoklular kümesi ya da bir çizelgedir.
 - Örnek 3.1. $D_1 = \{a, b, c\}$ $D_2 = \{1, 2, 3\}$ $D_3 = \{x, y\}$ olsun.
Bu üç alanın Karteziyen çarpımı $3 \times 3 \times 2 = 18$ adet 3-çokludan oluşan bir kümedir.
 $D_1 \times D_2 \times D_3 = \{ \langle a, 1, x \rangle, \langle a, 1, y \rangle, \langle a, 2, x \rangle, \dots, \langle c, 3, y \rangle \}$
Eğer A, B, C niteliklerinin değer alanı sırasıyla D_1 , D_2 ve D_3 ise, bu üç değer alanı üzerinde tanımlı, şeması $R(A, B, C)$ olan bir r ilişkisi örneğin aşağıdaki gibi olabilir.

$r = \{ \langle a, 2, x \rangle, \langle a, 2, y \rangle, \langle a, 3, y \rangle, \langle b, 1, x \rangle, \langle c, 1, y \rangle, \langle c, 3, x \rangle \}$
 r nin çizelge görünümü:

A	B	C
a	2	x
a	2	y
a	3	y
b	1	x
c	1	y
c	3	x

3.2. İlişkilerin Özellikleri

İlişki bir küme (çoklular/satırlar kümesi) olduğuna göre aşağıdaki özelliklere sahiptir.

- Satırların (çokluların) sırası önemsizdir.
- Tüm satırların birbirinden farklı olması gerekir
- Satırlar birbirinden farklı olduğuna göre ilişkinin en az bir anahtarı (anahtar adayı) vardır. En kötü olasılıkla, tüm niteliklerin birleşimi anahtar oluşturur.
- İlişkinin her kolonu bir niteliğe karşı geldiğine göre, kolonların sırası da önemsizdir.
- Bir kolonda yer alan tüm verilerin türü aynıdır.
- Başka bir deyişle çizelgenin nitelikleri yalın ve tek değerli niteliklerdir

İlişki anahtarları

- Süper Anahtar
- Anahtar adayı, İlişki Anahtarı, Anahtar, Birincil Anahtar

Şema ve Olgu (Örnek)

- İlişkinin adı ile ilişki niteliklerinin adlarından oluşan tanım deyimine ilişki şeması denir: $R(N_1, N_2, \dots, N_k)$
- İlişki şemasında genellikle, ilgili niteliklerin altı çizilerek, ilişki anahtarı gösterilir.
- Belirli bir kuruluşa/konuya ilişkin verilerin mantıksal düzeyde düzenlenmesini gösteren ve kendi aralarında anlamlı bir bütün oluşturan ilişki şemalarının bütününe, ilişkisel veri tabanı şeması adı verilir.
- İlişki şeması örnekleri
 - Öğrenci (öğrNo, adı, soyadı, cns, doğTar, öğrBno)
 - Bölüm (bno, badı, fakülte)
 - Ders (dKodu, dAdı, krd, açBno)
 - ÖğrDers (öğrNo, dKodu, dönemi, notu)

- İlişki şemasına, başka bir deyişle ilişkinin tanımına uygun olarak gerçekleşen ilişkinin belirli bir andaki durumuna ise ilişki olgusu ya da ilişki örneği (*instance*) adı verilir

Şeması Verilen İlişkilerin Birer Örneği

Öğrenci

öğrNo	adı	soyadı	cns	doğTar	öğrBno
3	Ahmet	Korkmaz	Erk	17/05/1974	56
17	Kemal	Güçlü	Erk	07/04/1976	55
26	Ayşe	Çetin	Kad	29/09/1975	56

Bölüm

bno	bAdı	fakülte
55	İşletme	İkt.ve İd.Bil.
56	Bilgisayar Müh.	Mühendislik

ÖğrDers

öğrNo	dKodu	dönemi	notu
3	BİL200	9394Bah	F4
3	BİL354	9495Bah	B2
3	İYB332	9495Güz	A2
3	BİL200	9495Bah	B1
17	İYB332	9495Güz	C
17	İYB105	9495Bah	F4
26	BİL200	9394Bah	F2
26	BİL200	9495Bah	A2
26	İYB332	9495Güz	B1

Ders

dKodu	dAdı	krd	açBno
BİL200	Mantısal Tas.	3	56
BİL354	Veri Tab. Sis.	3	56
İYB105	Muhasebe	2	55
İYB332	Gen. İşletme	3	55

3.3. Varlık-Bağıntı Çizeneklerinin İlişki Şemalarına Dönüştürülmesi

3.3.1. Varlık Kümelerinin Dönüştürülmesi

- Varlık-bağıntı modelindeki her varlık kümesi için ilişkisel modelde bir ilişki şeması oluşturulur. İlişkinin nitelikleri olarak da varlık kümesinin nitelikleri kullanılır.

Öğrenci	Ders
öğrNo {BA} adı soyadı cns doğTar	dKodu {BA} dAdı kredisi dili

Öğrenci (öğrNo, adı, soyadı, cns, doğTar)
Ders (dkodu, dAdı, kredisi, dili)

- Zayıf bir varlık kümesine karşı gelen ilişki şeması oluşturulurken, ilişki nitelikleri olarak varlık kümesinin niteliklerine ek olarak, bu varlık kümesinin varolma bağımlı olduğu güçlü varlık kümesinin anahtarında yer alan nitelikler de kullanılır.

Öğrenci	Okuyan	Lise
öğrNo adı soyadı cns doğTar	0..*	1..1
		liseNo {BA} liseAdı türü bulKent

Lise (liseNo, liseAdı, türü, bulKent)
Öğrenci (liseNo, öğrNo, adı, soyadı, cns, doğTar)

3.3.2. Bağlantı Kümelerinin Dönüştürülmesi

İkili Bire-Bir Bağlantı Kümelerinin Dönüştürülmesi

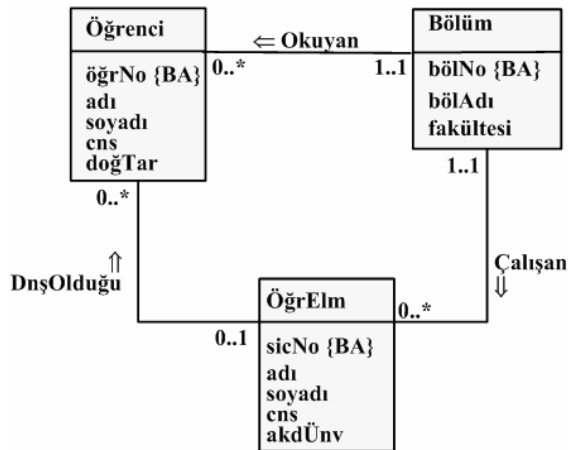
Personel	Yöneten	Bölüm
perNo {BA} adı soyadı cns doğTar	0..1	0..1
		bölNo {BA} bölAdı bulKat

Bölüm (bölNo, bölAdı, bulKat, yönPerNo)
Personel (perNo, adı, soyadı, cns, doğTar)
Bölüm (bölNo, bölAdı, bulKat)
Personel (perNo, adı, soyadı, cns, doğTar, yönBölNo)

Personel	Yöneten	Bölüm
perNo {BA} adı soyadı cns doğTar	1..1	0..1
		bölNo {BA} bölAdı bulKat

Bölüm (bölNo, bölAdı, bulKat, yönPerNo)
Personel (perNo, adı, soyadı, cns, doğTar)

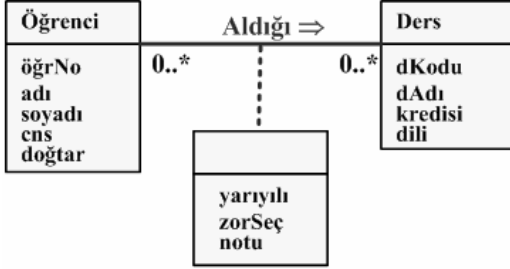
İkili Bire-Birçok Bağlantı Kümelerinin Dönüştürülmesi



Bölüm (bölNo, bölAdı, fakültesi)
ÖğrElm (sicNo, adı, soyadı, cns, akdÜnv, bölNo)
Öğrenci (öğrNo, adı, soyadı, cns, doğTar, bölNo, danSicNo)

İkili Birçoğa-Birçok Bağntı Kümelerinin Dönüştürülmesi

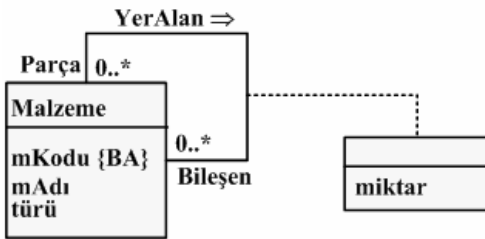
➤ Birçoğa-birçok her ikili bağntı kümesi için ilişkisel modelde ayrı bir ilişki oluşturulur.



Öğrenci (öğrNo, adı, soyadı, cns, doğtar)

Ders (dKodu, dAdı, kredisi, dili)

Aldığı (öğrNo, dKodu, yarıyılı, zorSeç, notu)

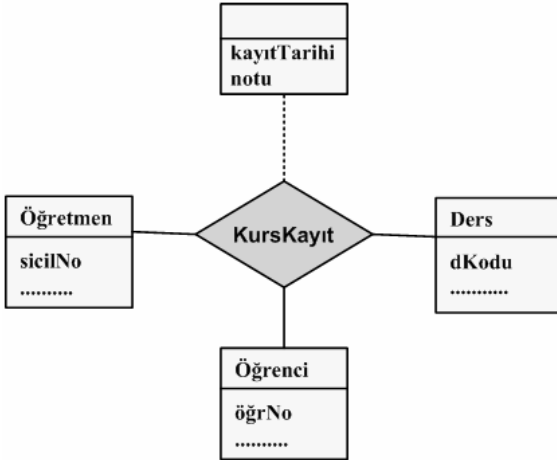


Malzeme (mKodu, mAdı, türü)

YerAlan (parKodu, bilKodu, miktar)

İkiden Büyük Dereceli Bağntı Kümelerinin Dönüştürülmesi

➤ Üçlü, dördlü, .. her ikili bağntı kümesi için ilişkisel modelde ayrı bir ilişki oluşturulur.



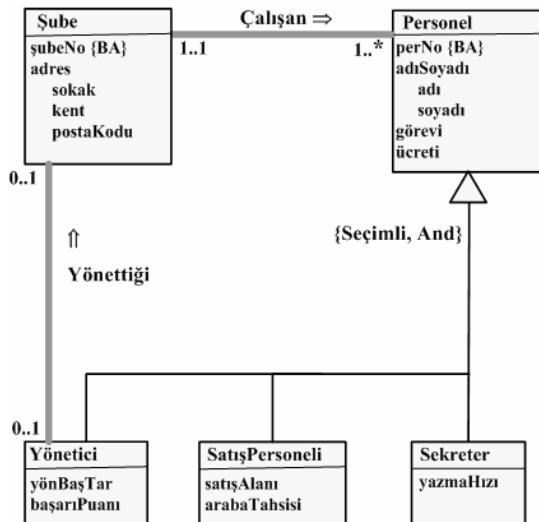
Öğretmen (sicilNo,)

Ders (dKodu,)

Öğrenci (öğrNo,)

KursKayıt (sicilNo, dKodu, öğrNo, kayıtTarihi, notu)

Genellemelerin Dönüştürülmesi



Şube (şubeNo, aSokak, aKent, aPKodu, yönPerNo)

Personel (perNo, adı, soyadı, görevi, ücreti, şubeNo)

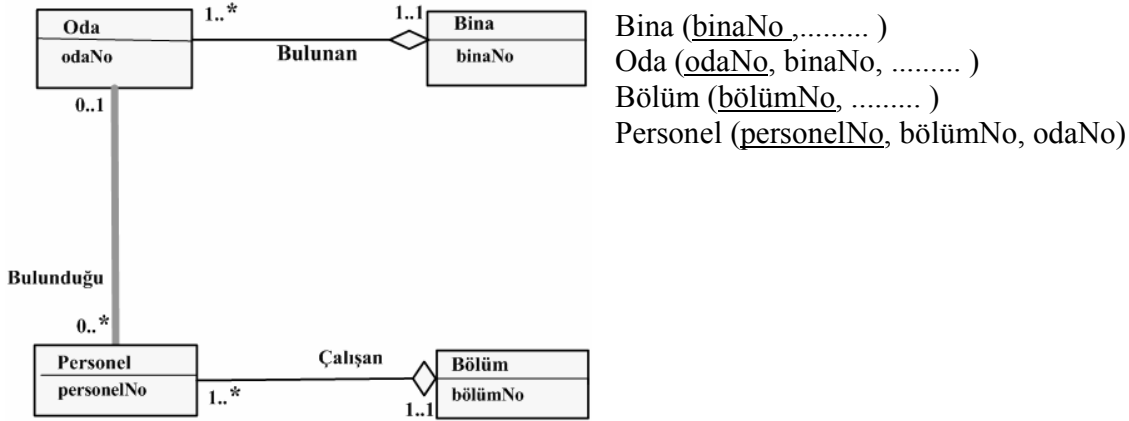
Yönetici (perNo, yönBaşTar, başarıPuanı)

SatışPersoneli (perNo, satışAlanı, arabaTahsisi)

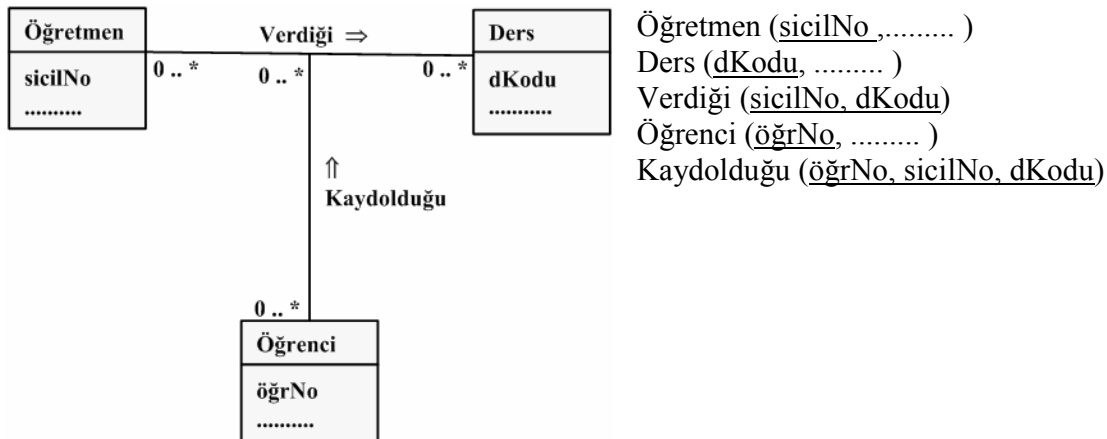
Sekreter (perNo, yazmaHızı)

Kümelemelerin Dönüştürülmesi

- Biri bütün olan, diğeri ise bütünü oluşturan bir parça olan iki varlık türü arasındaki “bulunur/sahiptir” (*has a*) ya da “parçasıdır” (*is-part-of*) bağıntısını gösteren kümelemenin ilişkisel modelde özel bir gösterimi yoktur.



- İki varlık kümesi ile aralarındaki birçoğa-birçok bağıntıyı bir küme olarak düşünerek bu kümeyle varlık kümeleri ya da diğer kümeler arasında kurulan bağıntıların ilişkisel modelde gösterilmesi.



3.4. Diğer Veri Modelleri

- İlişkisel veri modeli dışında en çok bilinen modeller geçmişte kullanılmış olan sıradüzensel ve ağ modelleri ile günümüzde kullanılan nesneye-yönelik modeldir.
- Nesneye- yönelik model ayrı bir bölümde ele alınacaktır. Bu bölümde ise, günümüzde artık kullanılsa bile, geçmişte çok kullanılmış olan ve veri tabanı kavramlarının gelişmesinde çok önemli rol oynamış olan ağ veri modeli ile sıradüzensel veri modeli özet olarak tanıtılacaktır.

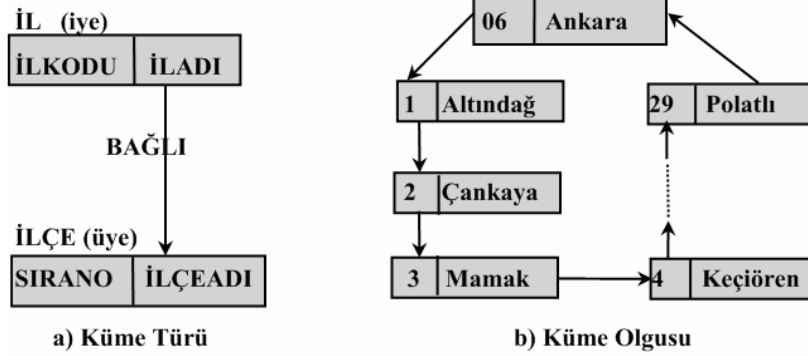
3.4.1 Ağ Veri Modeli

- Ağ veri modelinde varlık kümelerini göstermek için tutanak türleri (*record types*) oluşturulur. Her tutanak türü bir veri öğeleri (*data items*) yığıdır.
- İlişkisel modeldeki niteliklerden farklı olarak, veri öğeleri çok değerli karmaşık öğeler olabilir.
- Tutanak türleri varlık kümelerini göstermek için kullanılabileceği gibi, bağıntı kümelerini

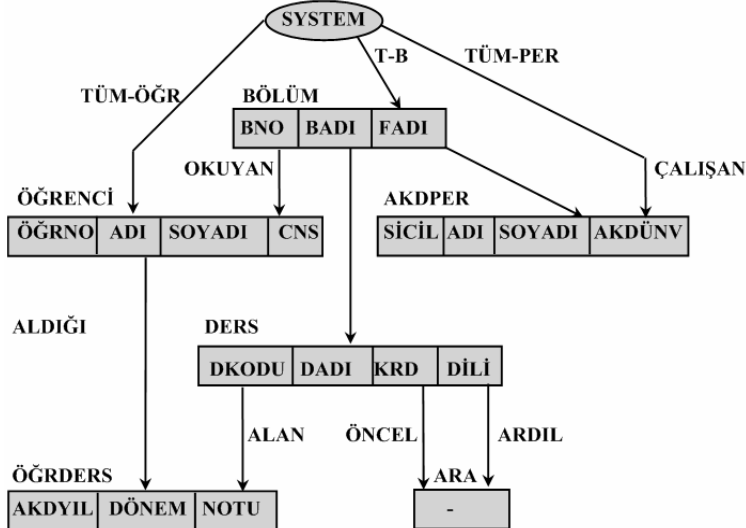
göstermek için de kullanılabilir. Ancak tanımlayıcı niteliği bulunmayan bağıntı kümelerini göstermek için ağ veri modelinde bağ (*link*) adı verilen ayrı bir yapı vardır.

- Genel ağ veri modelinde bağlar ile her türlü (bire-bir, bire-birçok, birçoğa-birçok) bağıntı gerçekleştirilebilir. Aralarında bağ kurulan iki tutanak türü farklı tutanak türleri olabileceği gibi aynı tutanak türü de olabilir.
- Tutanak türleri arasındaki bağları gösteren ve “veri yapısı çizeneği” olarak adlandırılan çizenecek veri tabanının kavramsal ya da dış (kullanıcı düzeyi) şemasını gösterir.
- Uygulamada genel ağ veri modeli pek kullanılmamış, yerine kısıtlı ağ veri modeli kullanılmıştır. Kısıtlı ağ modelinin en bilinen örneği CODASYL DBTG (CONference on DATA SYSTEMS Languages – Data Base Task Group) modelidir.
- Kısıtlı ağ modelinde (CODASYL’de) bağlar ile ilgili aşağıdaki iki kısıtlama konulmuştur.
 1. Aralarında bağ kurulan iki tutanak türü farklı tutanak türleri olmalıdır.
 2. Bir bağ ile gerçekleştirilen bağıntı birçoğa-birçok olamaz; bir yönde işlevsel olmak zorundadır.
- CODASYL modelinde “bağ” yerine “küme türü” ya da kısaca “küme” terimi kullanılır. *System* adlı özel bir tutanak türü vardır. System tutanak türü ile kullanıcının tanımladığı tutanak türleri arasında tanımlanan kümeler “tekil kümeler” denir.

CODASYL Küme Türü ve Küme Olgusu Örnekleri



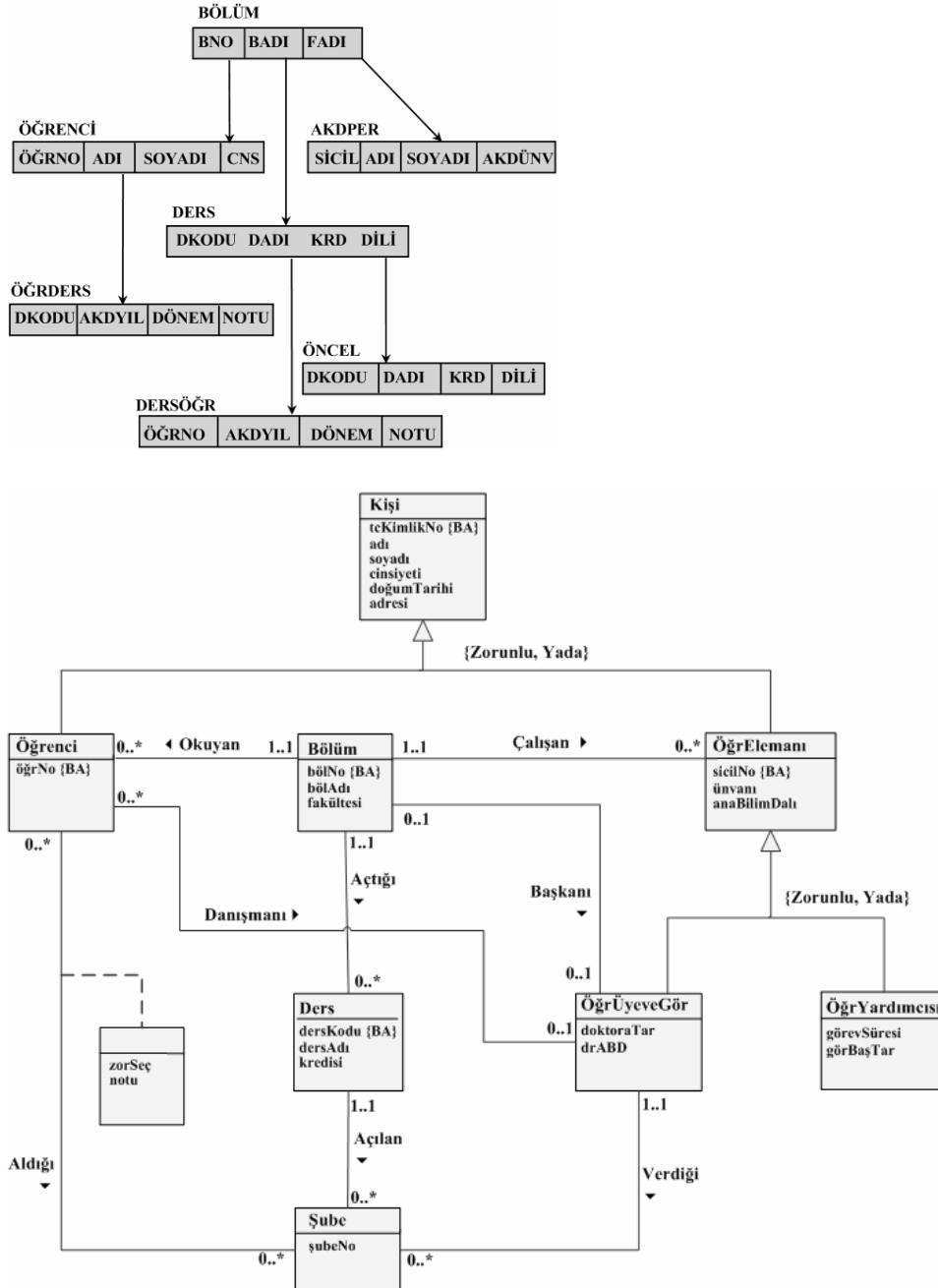
CODASYL Ağ Veri Modelinde Örnek Bir Veri Tabanı Şeması



3.4.2. Sıradüzensel Veri Modeli

- Sıradüzensel Veri Modeli (*Hierarchical Data Model*) ağ veri modelinin özel bir durumu olarak görülebilir.
- Kısıtlı ağ veri modelinde, tutanak türleri ve bağlardan oluşan veri yapısının bir ağaç yapısı oluşturması zorunluluğu getirilirse sıradüzensel veri modeli elde edilmiş olur.
- Bu modelde kullanıcı düzeyindeki her şema (altşema, dış şema ya da kullanıcı görünümü) belirli sayıda sıradüzen tanım ağacı'ndan oluşur.
- Tüm bağlar ana tutanak türünden oğul tutanak türüne bire-birçok'tur.
- Sıradüzensel veri modelinin gösterim gücü bundan önce gördüğümüz varlık-bağıntı, ilişkisel ve ağ modellerinden daha düşüktür. Ağ yapıları, birçoğa-birçok bağıntılar ve döngülü yapılar gibi karmaşık yapı ve bağıntıları sıradüzensel modelde göstermek zordur.
- Modelin yetersizliğini gidermek için, kullanıcı düzeyinde veri tekrarlanır. Aynı verilere bir ağacın farklı düğümlerinde ya da farklı ağaçlarda yer verilebilir. Ancak dış şema düzeyindeki bu veri tekrarları kavramsal ve fiziksel şema düzeylerinde sürdürülmez.

Sıradüzensel Veri Modelinde Örnek Bir Veri Tabanı Şeması



Ek:
Üniversite
Öğrenci İşleri
Veri Tabanı
Varlık-Bağıntı
Çizeneği

Üniversite Öğrenci İşleri Veri Tabanı İlişkisel Şeması

Kişi (tcKimlikNo, adı, soyadı, cinsiyeti, doğumTarihi, adresi)

Bölüm(bölNo, bölümAdı, fakültesi)

ÖğrElemanı(tcKimlikNo, sicilNo, ünvanı, anaBilimDalı, bölümNo)

Öğrenci(tcKimlikNo, öğrNo, bölümNo, danSicNo)

ÖğryeveGör(sicilNo, doktoraTar, drABD)

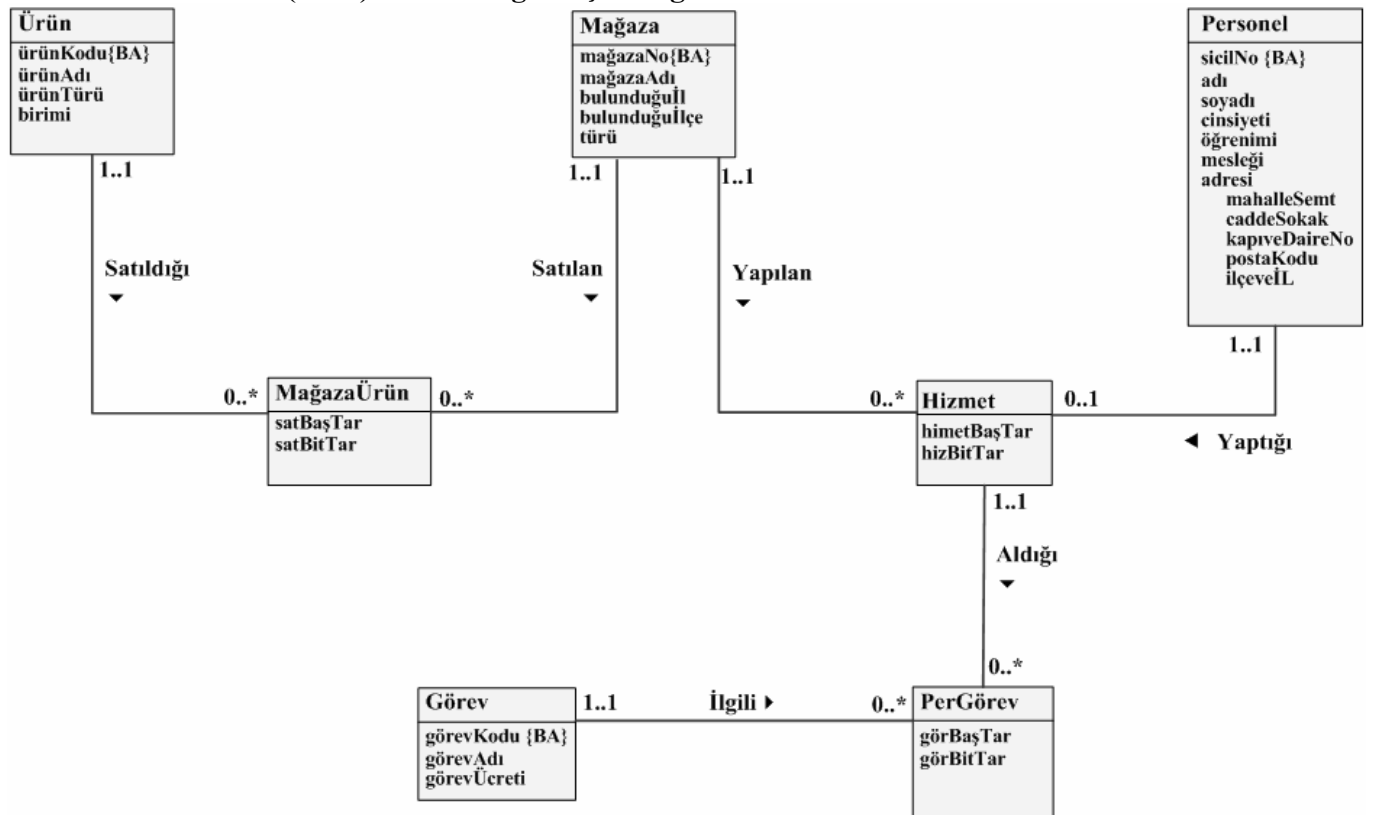
ÖğrYardımcısı(sicilNo, görevSüresi, görBaşTar)

Ders(dersKodu, dersAdı, kredisi, açanBölNo)

Şube(dersKodu, şubeNo, sicilNo)

DersKayıt(öğrNo, dersKodu, şubeNo, zorSeç, notu)

Market Veri Tabanı (S.2.2) Varlık-Bağıntı Çizeneği



Market Veri Tabanı İlişkisel Şeması

Ürün (ürünKodu, ürünAdı, ürünTürü, birimi)

Mağaza (mağazaNo, mağazaAdı, bulunduğull, bulunduğullçe, türü)

Personel (sicilNo, adı, soyadı, cinsiyeti, öğrenimi, mesleği, mahalleSemt, caddeSokak, kapıveDaireNo, postaKodu, ilçevelL)

Görev (görKodu, görAdı, görÜcreti)

MağazaÜrün (mağazaNo, ürünKodu, satBaşTar, satBitTar)

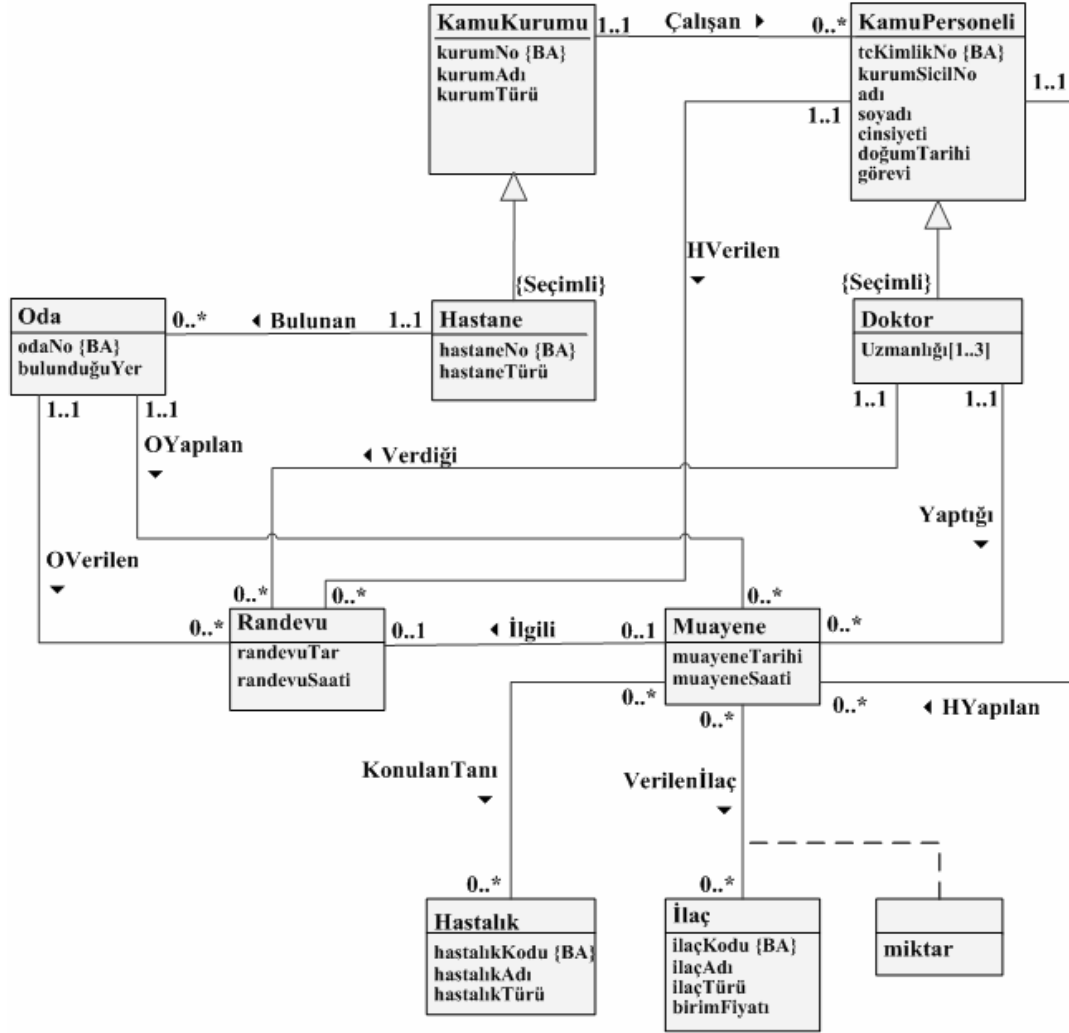
Hizmet (sicilNo, mağazaNo, hizBaşTar, hizBitTar)

HizGörev (sicilNo, mağazaNo, hizBaşTar, görKodu, görBaştar, görBitTar)

Hizmet (sicilNo, hizSıraNo, mağazaNo, hizBaşTar, hizBitTar)

HizGörev (sicilNo, hizSıraNo, görSıraNo, görKodu, görBaştar, görBitTar)

Kamu Çalışanları Sağlık Veri Tabanı (S.2.3)Varlık-Bağıntı Çizeneği



Kamu Çalışanları Sağlık Veri Tabanı İlişkisel Şeması

KamuKurumu (kurumNo, kurumAdı, kurumTürü)

Hastane (kurumNo, hastaneNo, hastaneTürü)

Oda (hastaneNo, odaNo, bulunduđuYer)

KamuPersoneli (tcKimlikNo, kurumSicilNo, adı, soyadı, cinsiyeti, doğumTarihi, görevi, kurumNo)

Doktor (tcKimlikNo)

Uzmanlık (tcKimlikNo, uzmanlığı)

Randevu (hastaTcKimlikNo, drTcKimlikNo, hastaneNo, odaNo, randevuTar, randevuSaati, randevuNo)

Muayene (hastaTcKimlikNo, drTcKimlikNo, hastaneNo, odaNo, muayeneTar, muayeneSaati, muayeneNo, randevuNo)

Hastalık (hastalıkKodu, hastalıkAdı, hastalıkTürü)

İlaç (ilaçKodu, ilaçAdı, ilaçTürü, birimFiyatı)

Verilenİlaç (ilaçKodu, muayeneNo, miktar)

KonulanTanı (muayeneNo, hastalıkKodu)