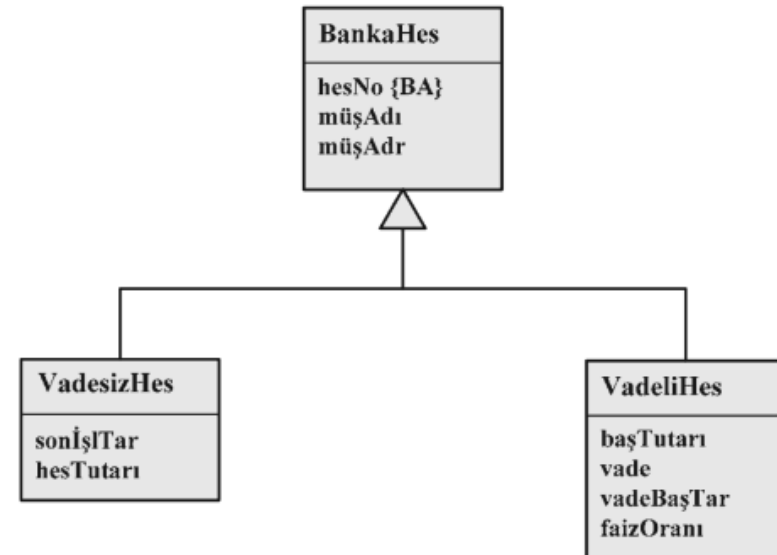
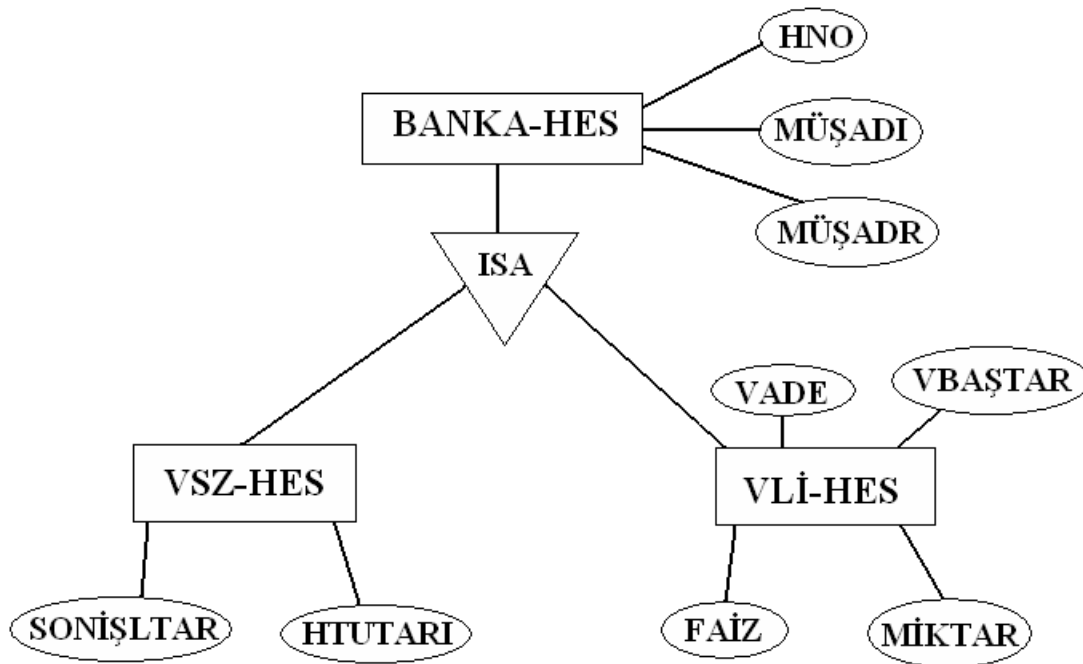


GENELLEME (Generalization)

- Varlık kümeleri arasında özel ilişkiler bulunabilir, bunların en önemlisi “ait olma” ilişkisidir.
- **Alt düzey varlık kümeleri genellenerek bir üst düzey varlık kümesi oluşturulabilir.**

Genelleme Örneği

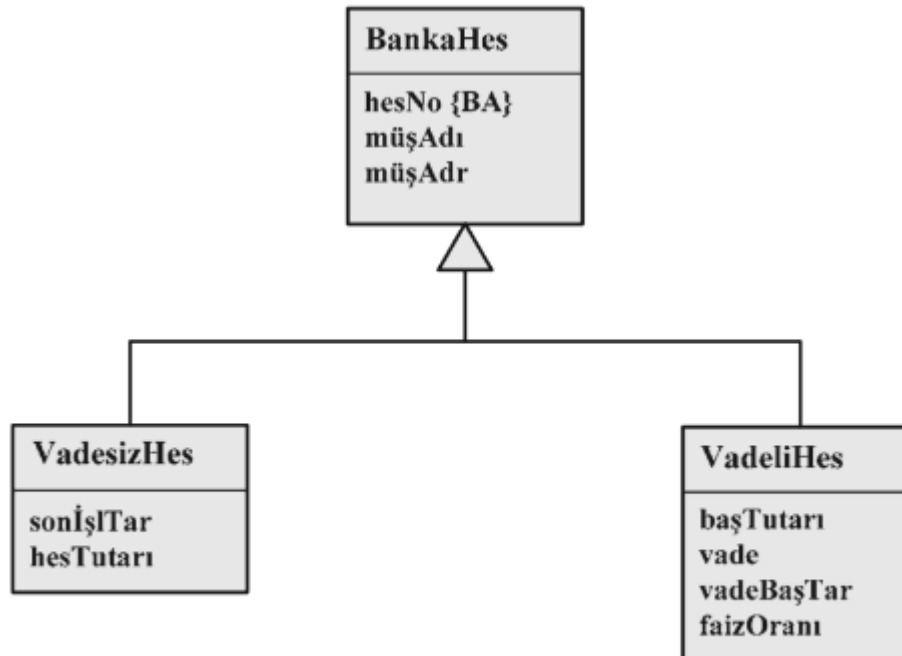
- VSZ-HES (vadesiz hesap) ile VLİ-HES (vadeli hesap) kümeleri genellenerek BANKA-HES (banka hesap) varlık kümesi oluşturulabilir.



Çizim 2.16. Genelleme Örneği

Genelleme Örneği

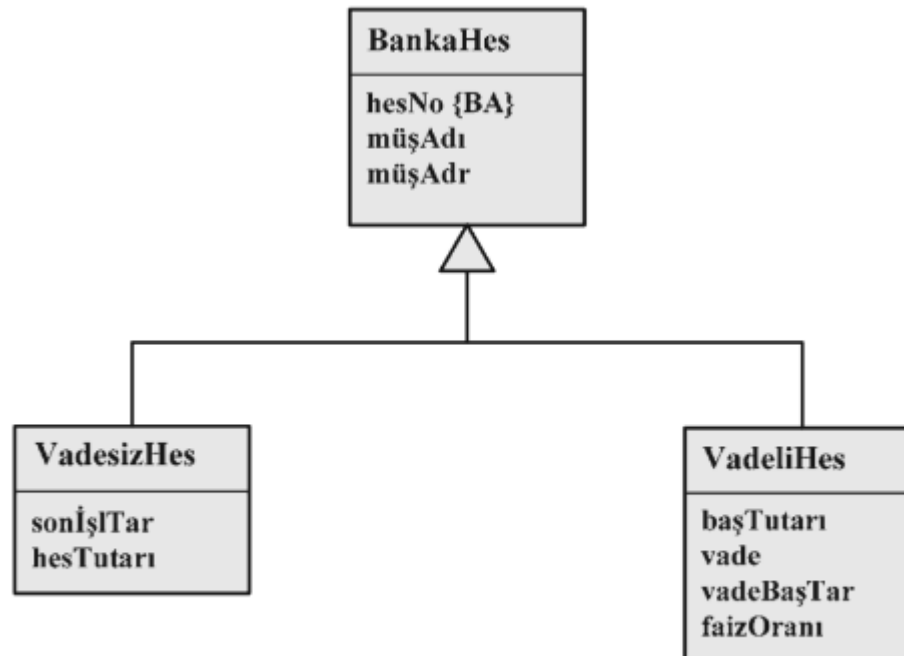
- VSZ-HES ile BANKA-HES varlık kümeleri arasındaki ilişki özel ilişkidir (ait olma ilişkisi).
- Her VSZ-HES bir BANKA-HES'dir biçiminde ifade edilebilecek bu ilişkiye **ISA (is-a)** ilişkisi adı verilir.



Çizim 2.16. Genelleme Örneği

Genelleme Örneği

- ISA ilişkisi üst düzey bir varlık türü ile bunun alt türleri arasında kurulan özel bir ilişkidir.
- ISA ilişkisinde üst düzey varlığın nitelikleri kalıtım yoluyla alt düzey varlık türlerine geçer.



Çizim 2.16. Genelleme Örneği

Genelleme ya da Özelleştirme

ISA bağıntısı bakış açısına göre bir genelleme ya da bir özelleştirme olarak görülebilir. Yukarıdan-aşağıya yaklaşıma göre *ISA* bağıntısı bir özelleştirmedir. Aşağıdan-yukarıya yaklaşıma göre ise *ISA* bağıntısı bir genellemedir. Genelleme/özelleştirme ile sınıf/altsınıf (ya da tür/alttür) bağıntıları oluşturulur; sınıf (ya da tür) sıradüzeni kurulur. Özelleştirme ile bir üstsınıfın bir ya da birçok altsınıfı oluşturulabilir. Genelleme ile de bir ya da birden çok altsınıfın ortak bir üstsınıfı oluşturulur. Gerek özelleştirmede, gerekse genellemede üstsınıfın nitelikleri kalıtım yoluyla altsınıflara geçer.

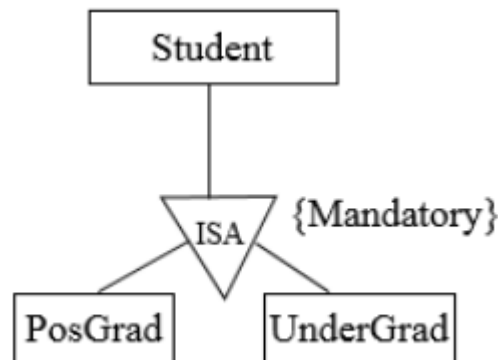
Genelleme/Özelleştirme Kısıtlamaları

Her genelleme/özelleştirme ile ilgili iki kısıtlama belirtilir. Bu kısıtlamalar şunlardır:

1. Katılım kısıtlaması: zorunlu (*mandatory*) ya da seçimli (*optional*) olabilir.

Zorunlu katılım, üstsınıfın her ögesinin altsınıfların en az birinde yer almasının zorunlu olduğunu gösterir.

Seçimli katılım ise, üstsınıfın her ögesinin altsınıflarda yer almasının seçimli olduğunu; üstsınıfın kimi öğelerinin hiçbir altsınıfta yer almayabileceğini gösterir.



Genelleme/Özelleştirme Kısıtlamaları

2. Ayrıklık kısıtlaması: OR ya da AND olabilir.

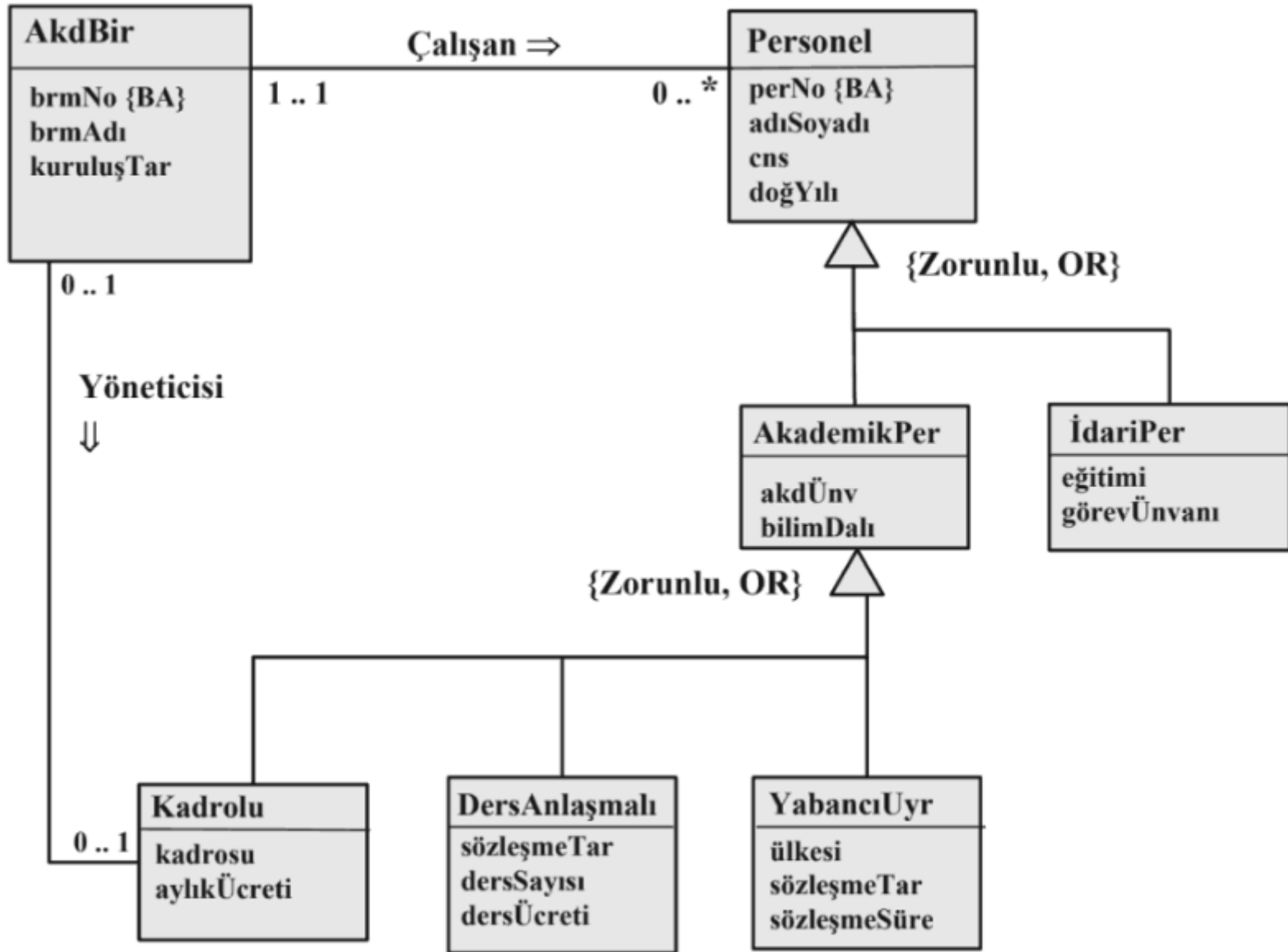
OR, üstsınıfın her ögesinin en çok bir altsınıfta yer alabileceğini, bir başka deyişle altsınıfların ayırık (*disjoin*) olacağını gösterir.

AND ise, üstsınıfın her ögesinin birden çok altsınıfta yer alabileceğini, bir başka deyişle altsınıfların örtüşen (*non disjoin*) altsınıflar olduğunu gösterir.

Buna göre genelleme/özelleştirme için kullanılabilecek kısıtlamaların alabileceği 4 farklı birleşim aşağıdakilerdir:

- a. {seçimli, OR}
- b. {seçimli, AND}
- c. {zorunlu, OR}
- d. {zorunlu, AND}

Genelleme/Özelleştirme Kısıtlamaları



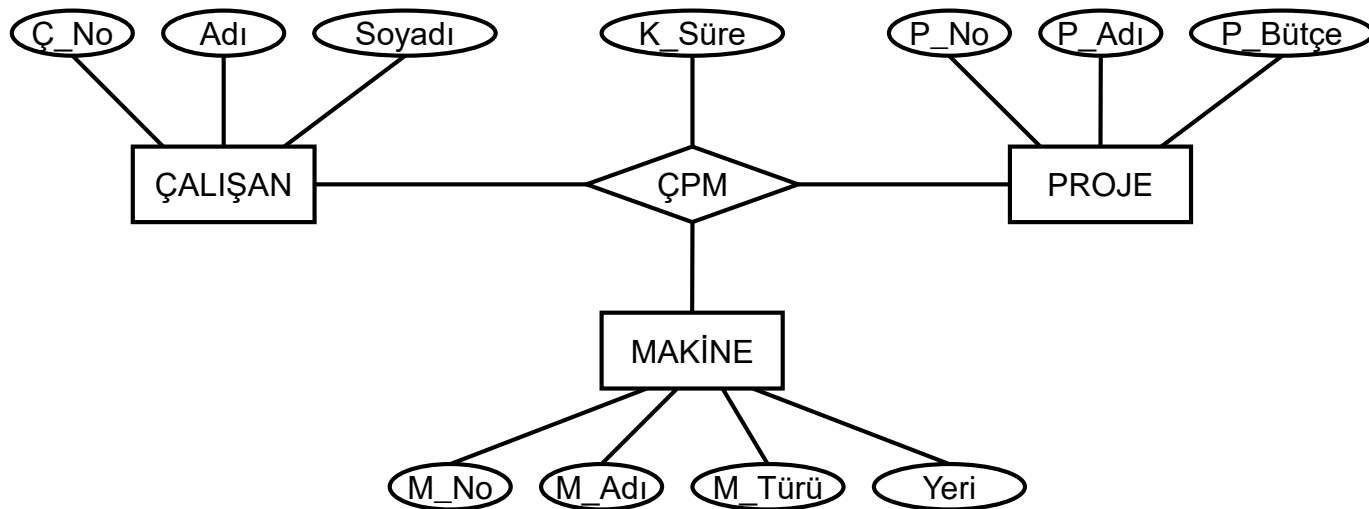
Çizim 2.17. Personel Varlık Kümesi İle İlgili Özelleştirme Örnekleri

KÜMELEME (Aggregation)

- Varlık-ilişki modelinde veri modellemeyi kolaylaştıran kavramlardan biri de kümeleme kavramıdır.
- Bilindiği gibi ilişki kümeleri iki ya da daha çok sayıdaki varlık kümesi arasında kurulur, **ikiden çok sayıda varlık kümesi arasında kurulan ilişkiler**, çoğunlukla, hem semantik açıdan hem de işletimsel açıdan uygun çözümler oluşturmazlar.

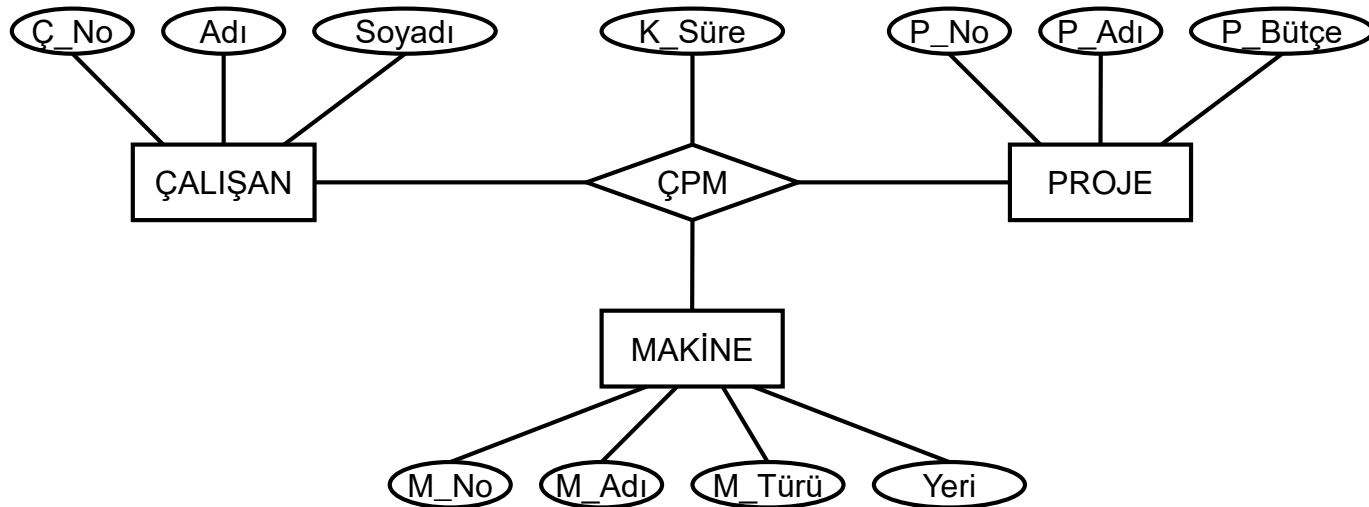
Kümeleme Örneği

- ÇALIŞAN, PROJE ve MAKİNE varlık kümeleri arasındaki üçlü ÇPM ilişkisini düşünelim.



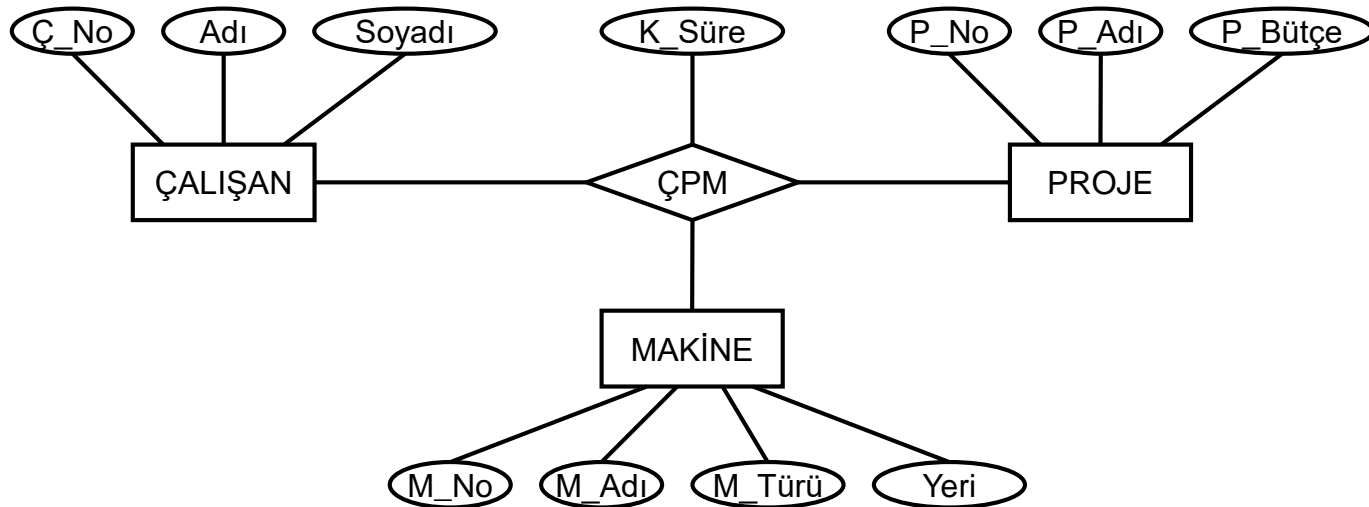
Kümeleme Örneği

- ÇPM ilişki kümesindeki $[ç,p,m]$ üçlüsü çalışan $ç$ 'nin p projesinde çalıştığını ve bu proje için m makinesini kullandığını gösterecektir.
- Bu ilişki hangi çalışanın hangi projelerde çalıştığını ve hangi çalışanın hangi proje için hangi üretim makinesini ne kadar süre kullandığını gösterecektir.



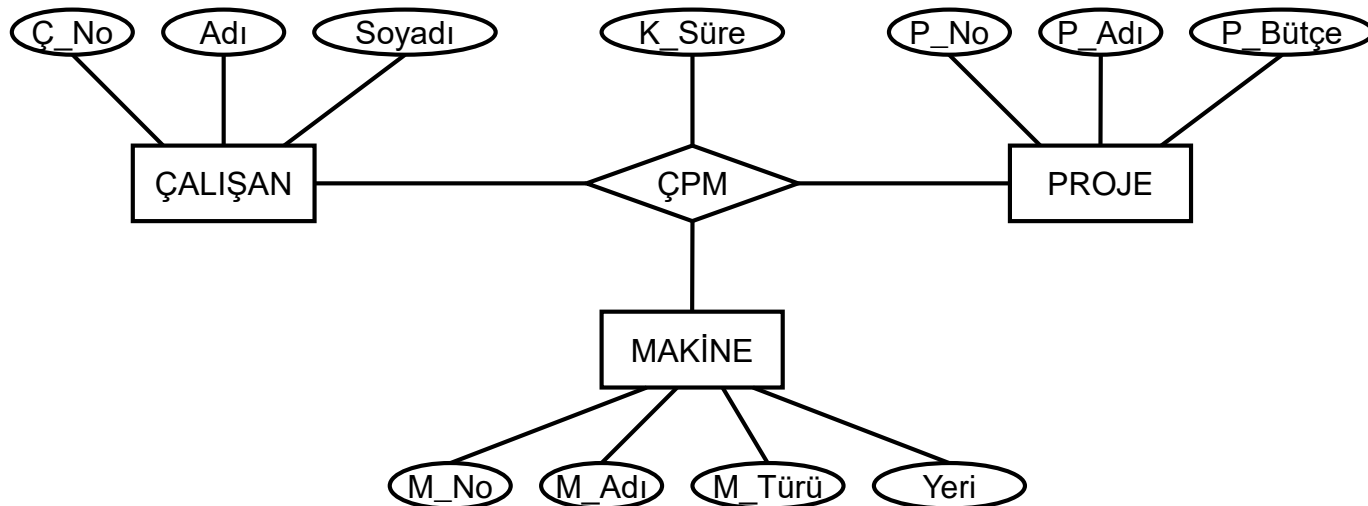
Kümeleme Örneği

- Bu ilişkinin uygun bir düzenleme olmadığı görülmektedir. Nedenleri:
 - Çalışanlar ile projeler arasındaki görevlendirme bilgisi makine kullanımından bağımsız yapılamamaktadır.
 - Bir çalışanın bir projedeki görevlendirme bilgisi, kullanılan makine sayısı kadar tekrarlanmaktadır.



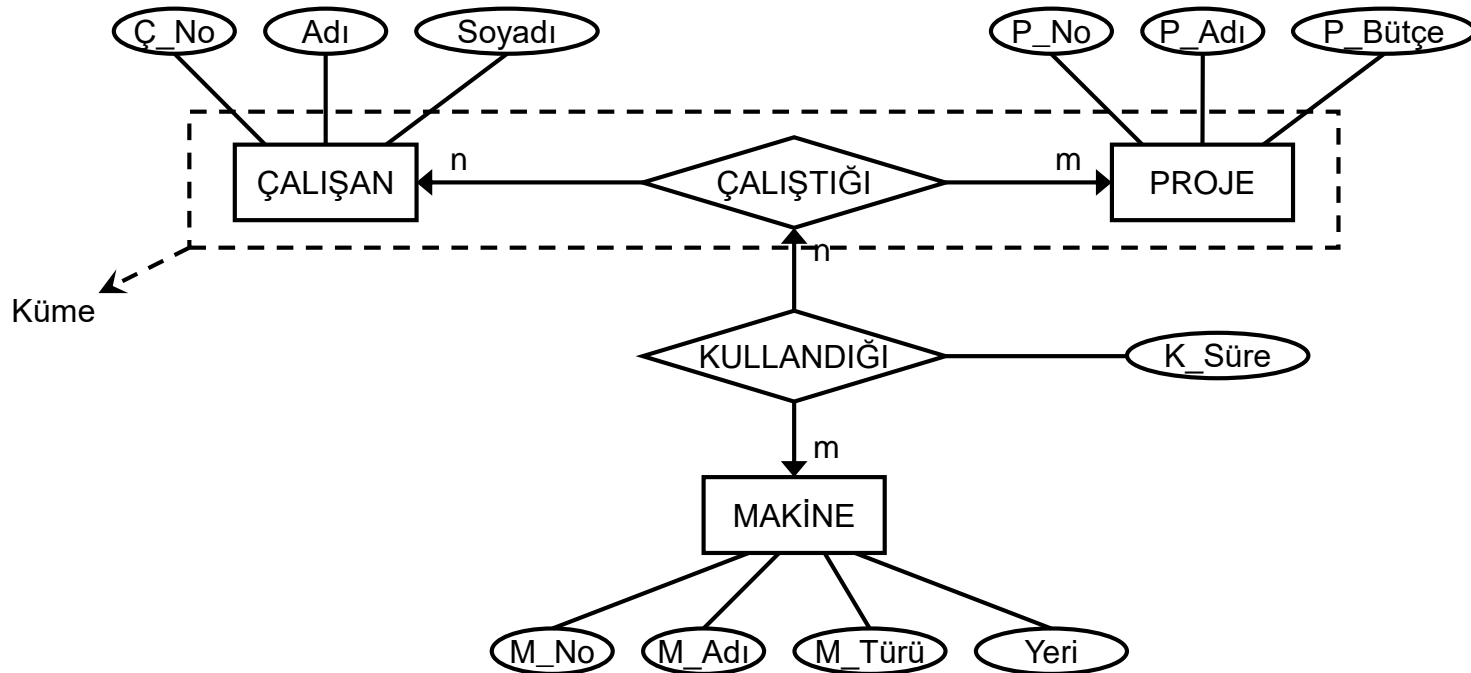
Kümeleme Örneği

- Bu üçlü ilişki hem gereksiz tekrarlar içerir, hem de bir dizi aykırılıklara yol açabilir.
- Bu nedenle bu üçlü ilişki yerine iki tane ikili ilişki kullanmak daha uygun bir çözümdür.



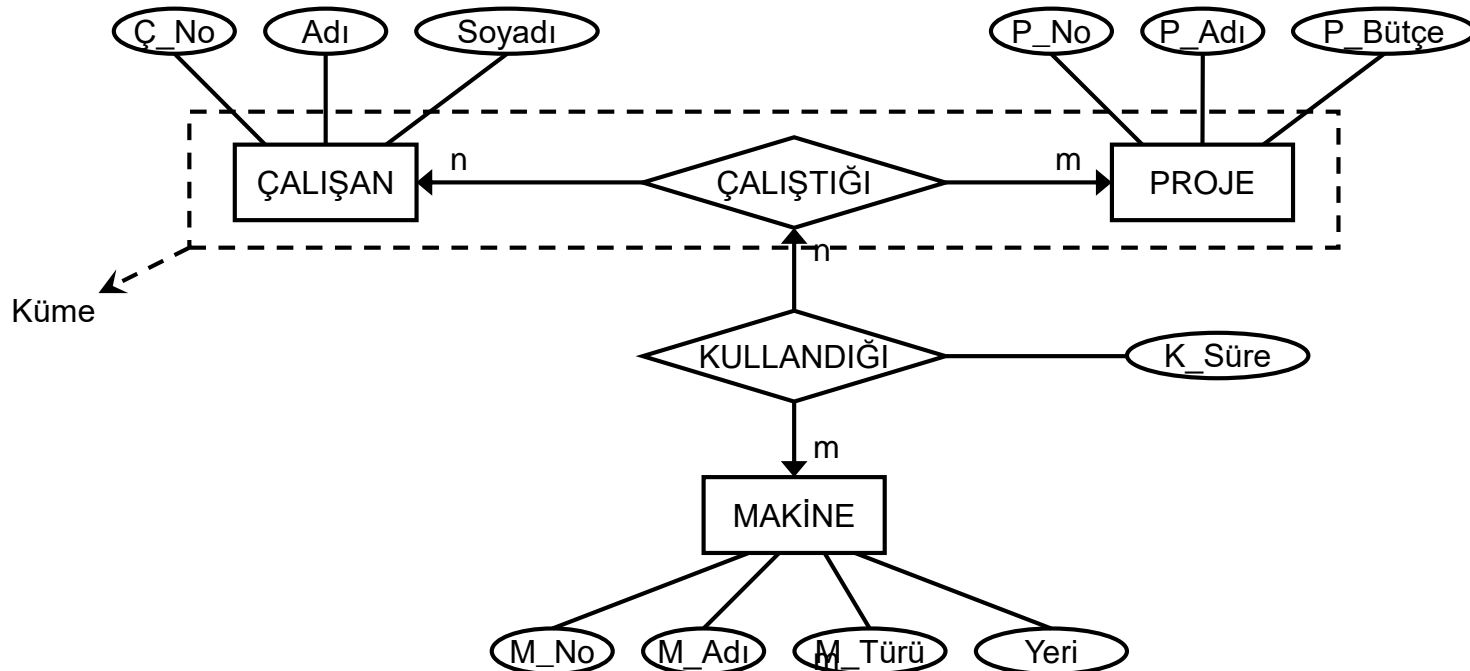
Kümeleme Örneği

- Daha önce yapılan tanıma göre ikili ilişki iki varlık kümesi arasında kurulur. Kümeleme kavramı varlık kümeleri ve aralarındaki ilişkinin kümelenmesini ve bu kümenin bir varlık kümesi gibi düşünülmesini sağlar.

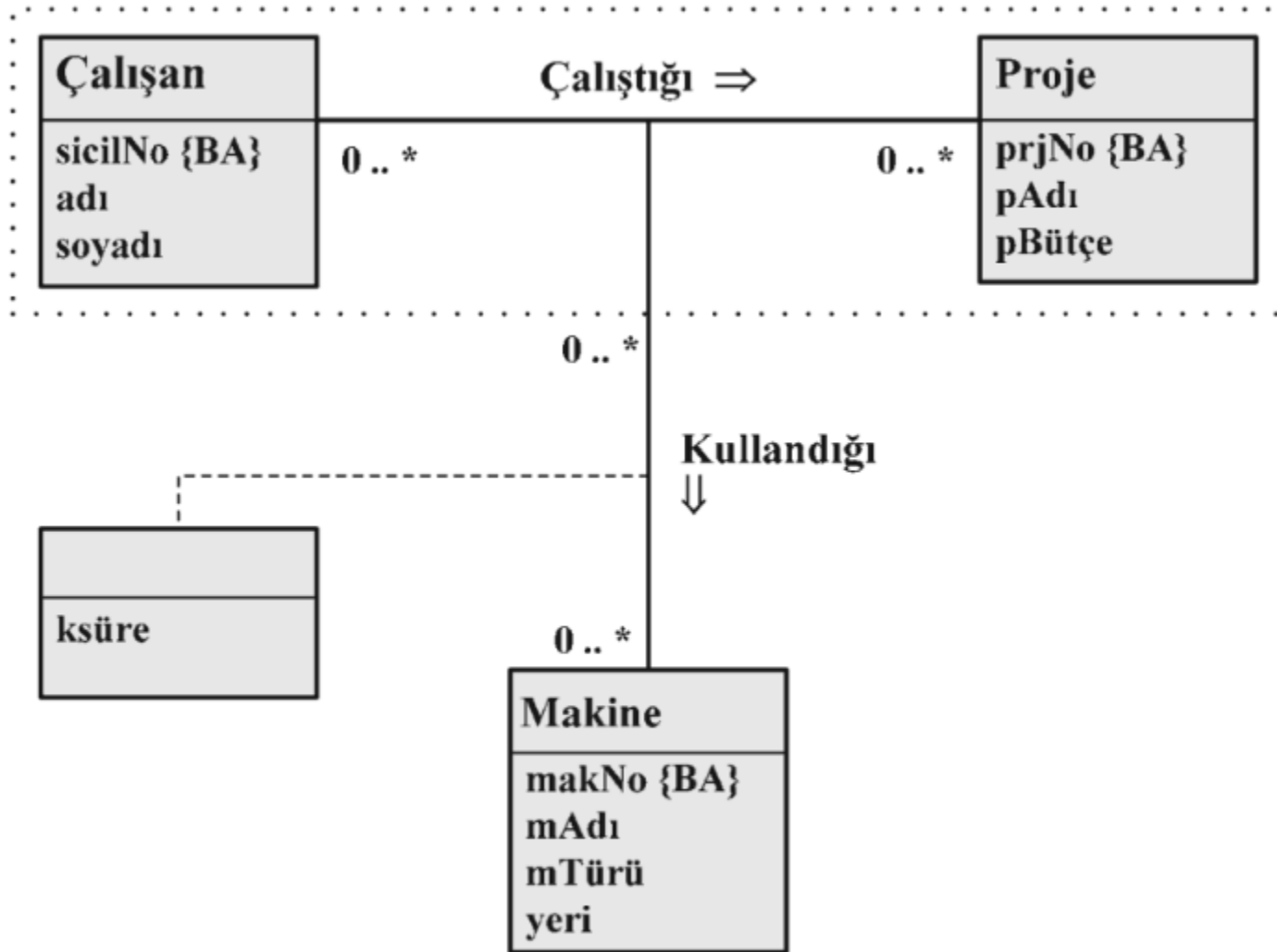


Kümeleme Örneği

- Böylece ÇALIŞAN ve PROJE varlık kümeleri ile aralarındaki ÇALIŞTIĞI ilişkisi kümelenir; bu küme ile MAKİNE varlık kümesi arasında KULLANDIĞI ilişkisi kurulur.



Kümeleme Örneği (UML Notasyonu)



Çizim 2.20. Kümeleme Örneği (2. Anlamda)

Kümeleme Örneği

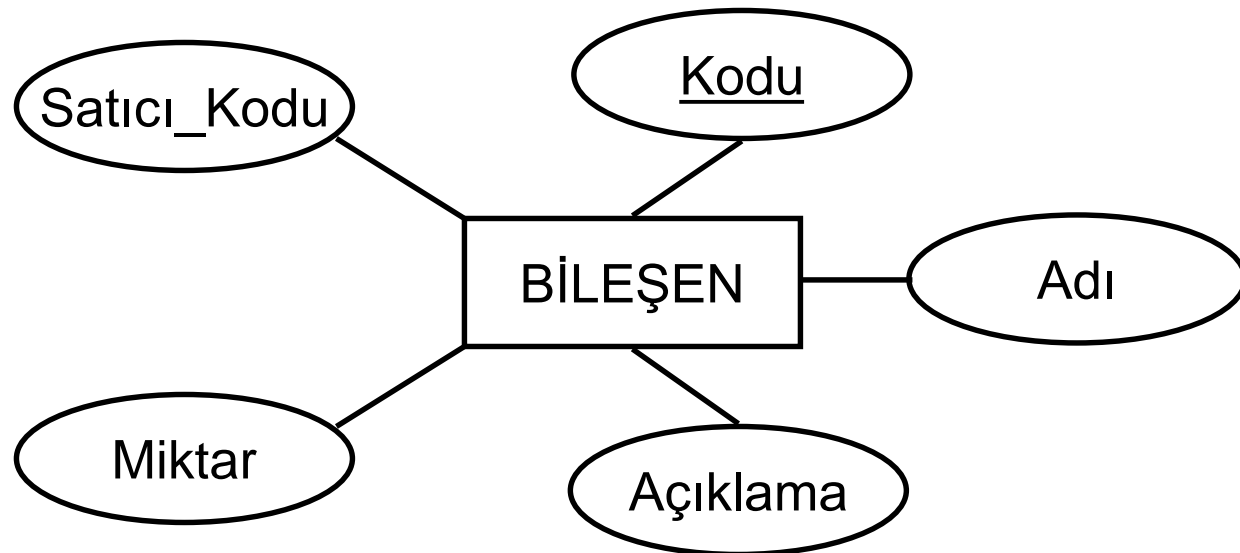
- Kümeleme kavramı, ikili bir ilişkinin iki varlık kümesi arasında olduğu gibi bir varlık kümesi ile bir ilişki kümesi, ya da iki ilişki kümesi arasında kurulmasını olanaklı kılar.
- Not: Varlık-ilişki çizeneklerinde, çizimin karmaşıklaşmaması için, genellikle kümelemeyi gösteren kesikli dikdörtgenlere yer verilmez.

Varlık-İlişki Modeli Örneği

- Bir elektronik firması ürettiği ürünler, kullandığı bileşenler ve her ürün için hangi bileşenlerin hangi miktarlarda kullanıldığını gösteren bir Ürün-Bileşen Veri Tabanı oluşturmak istiyor:
 1. Veri tabanında her bileşenin kodu, adı, tanıtıcı açıklaması, stok miktarı ile hangi satıcıdan ne zaman satın alındığı verilerinin bulunması isteniyor.
 2. Her satıcının numarası, adı, adresi ve telefon numarasının veri tabanında yer alması isteniyor.
 3. Her bileşen tek bir satıcıdan satın alınıyor; ancak aynı satıcıdan birden çok bileşen alınabiliyor.
 4. Her ürünün kodu, adı, birim fiyatı ve stok miktarı gerekiyor.
 5. Bu ürünün üretilmesi için hangi bileşenlerden kaçar adet gerekli olduğu önem taşıyor.
 6. Her ürün bileşenlere varolma bağımlıdır.

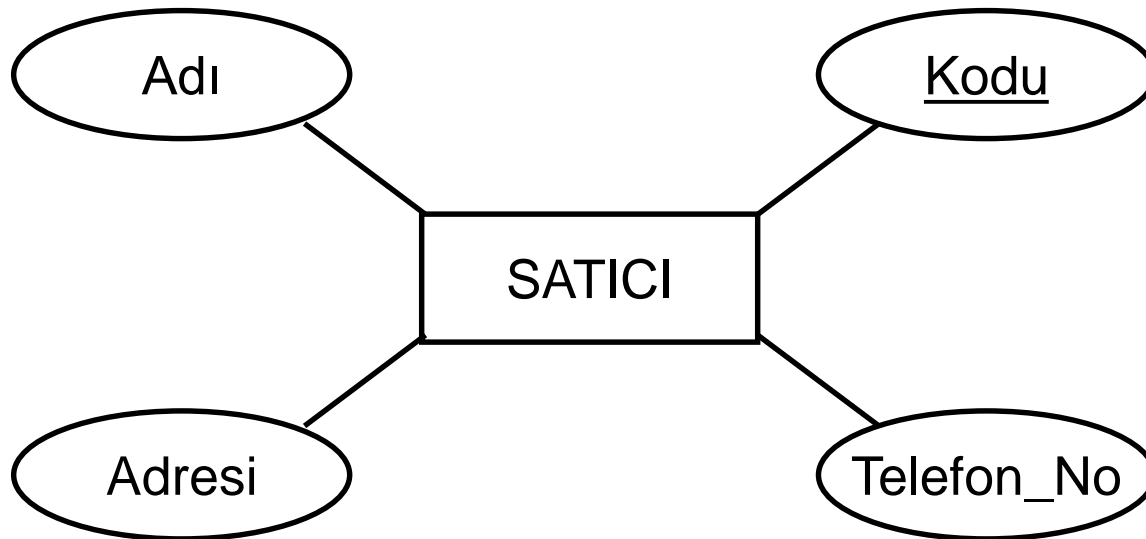
Çözüm 1

- Veri tabanında her bileşenin kodu, adı, tanıtıcı açıklaması, stok miktarı ile hangi satıcıdan satın alındığı verilerinin bulunması isteniyor.



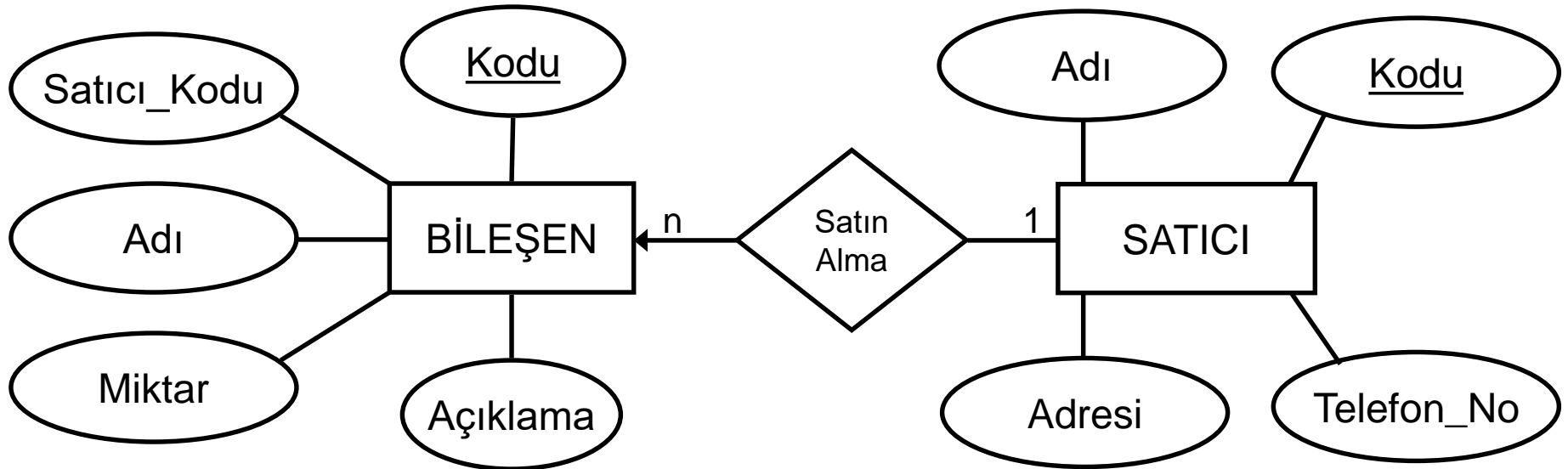
Çözüm 2

- Her satıcının numarası, adı, adresi ve telefon numarasının veri tabanında yer alması isteniyor.



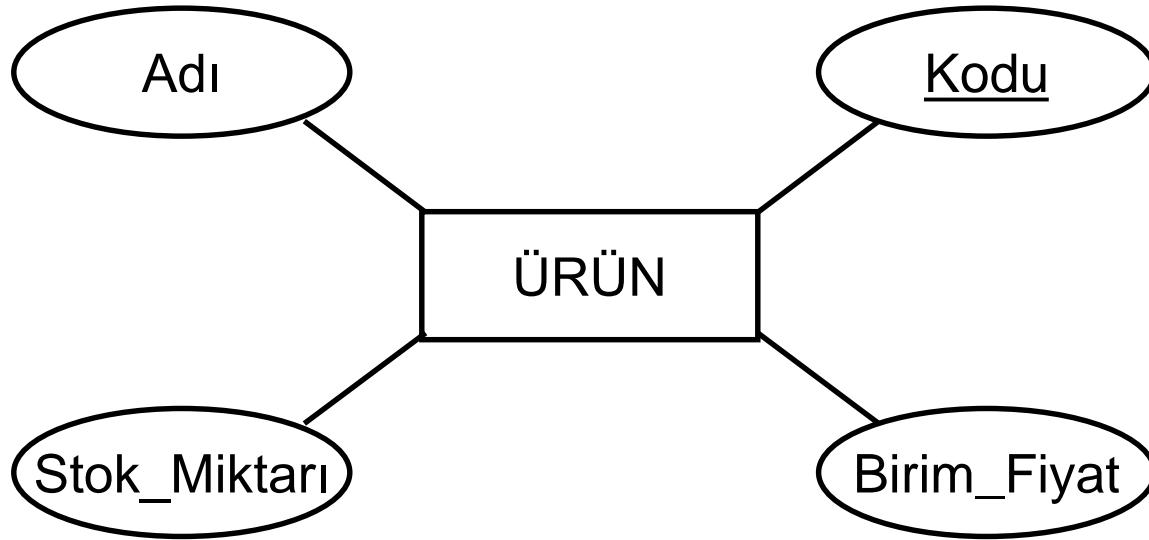
Çözüm 3

- Her bileşen tek bir satıcıdan satın alınıyor; ancak aynı satıcıdan birden çok bileşen alınabiliyor.



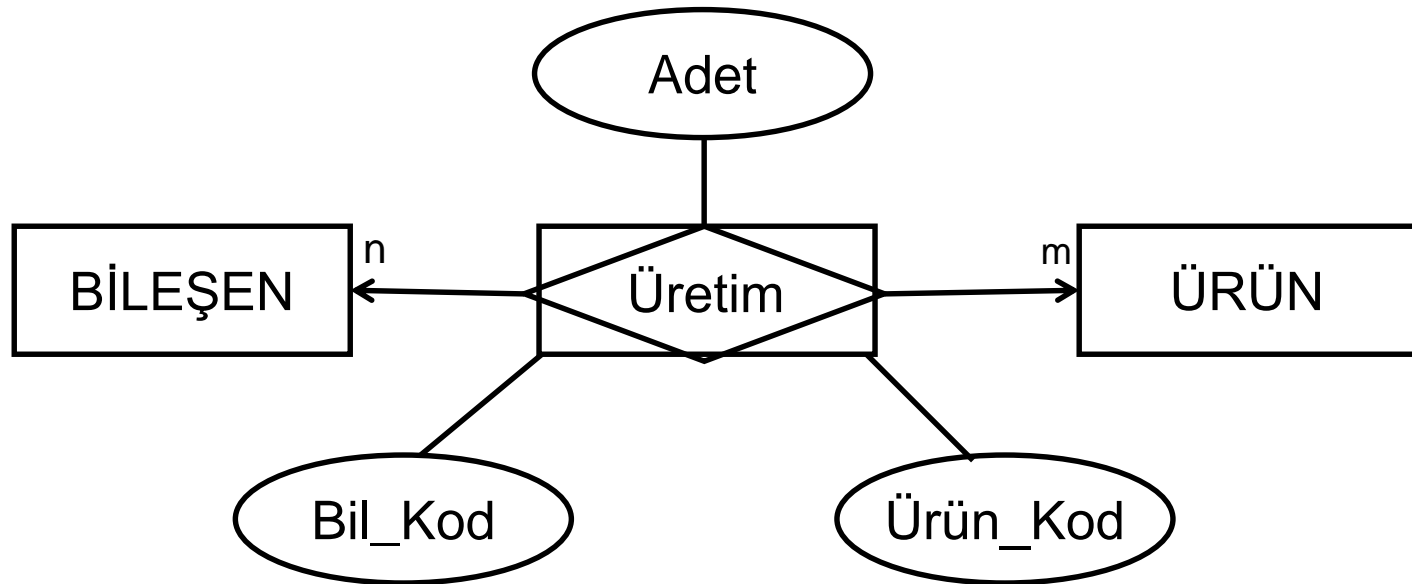
Çözüm 4

- Her ürünün kodu, adı, birim fiyatı ve stok miktarı gerekiyor.



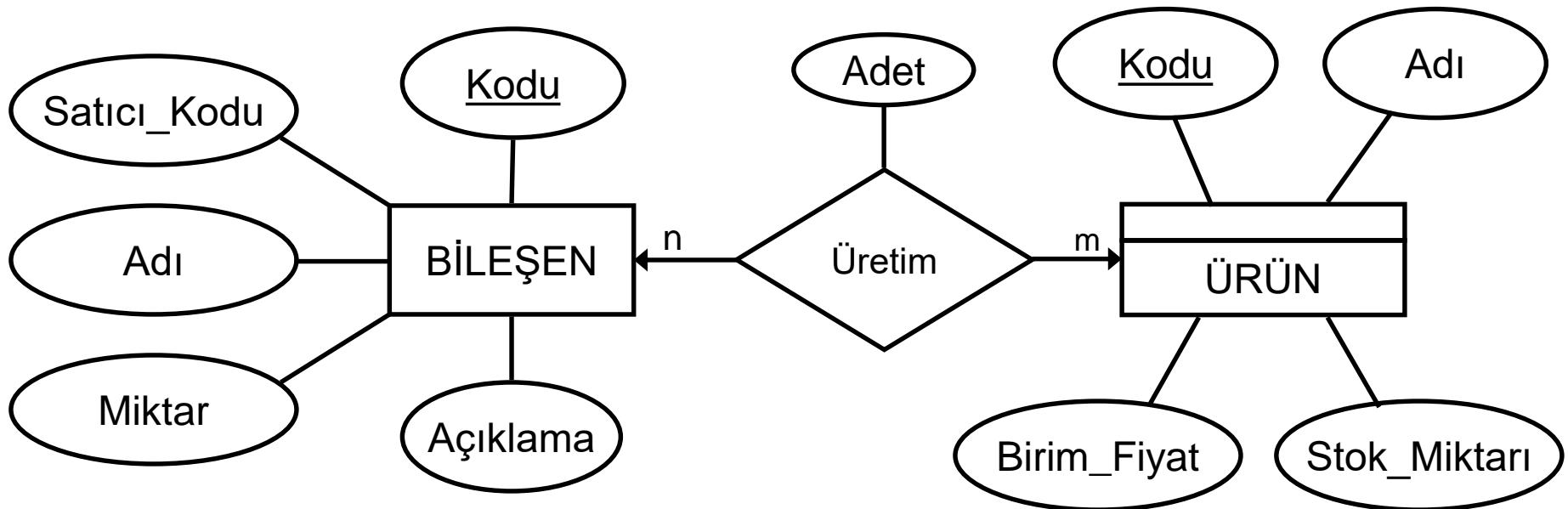
Çözüm 5

- Bu ürünün üretilmesi için hangi bileşenlerden kaçar adet gerekli olduğu önem taşıyor.

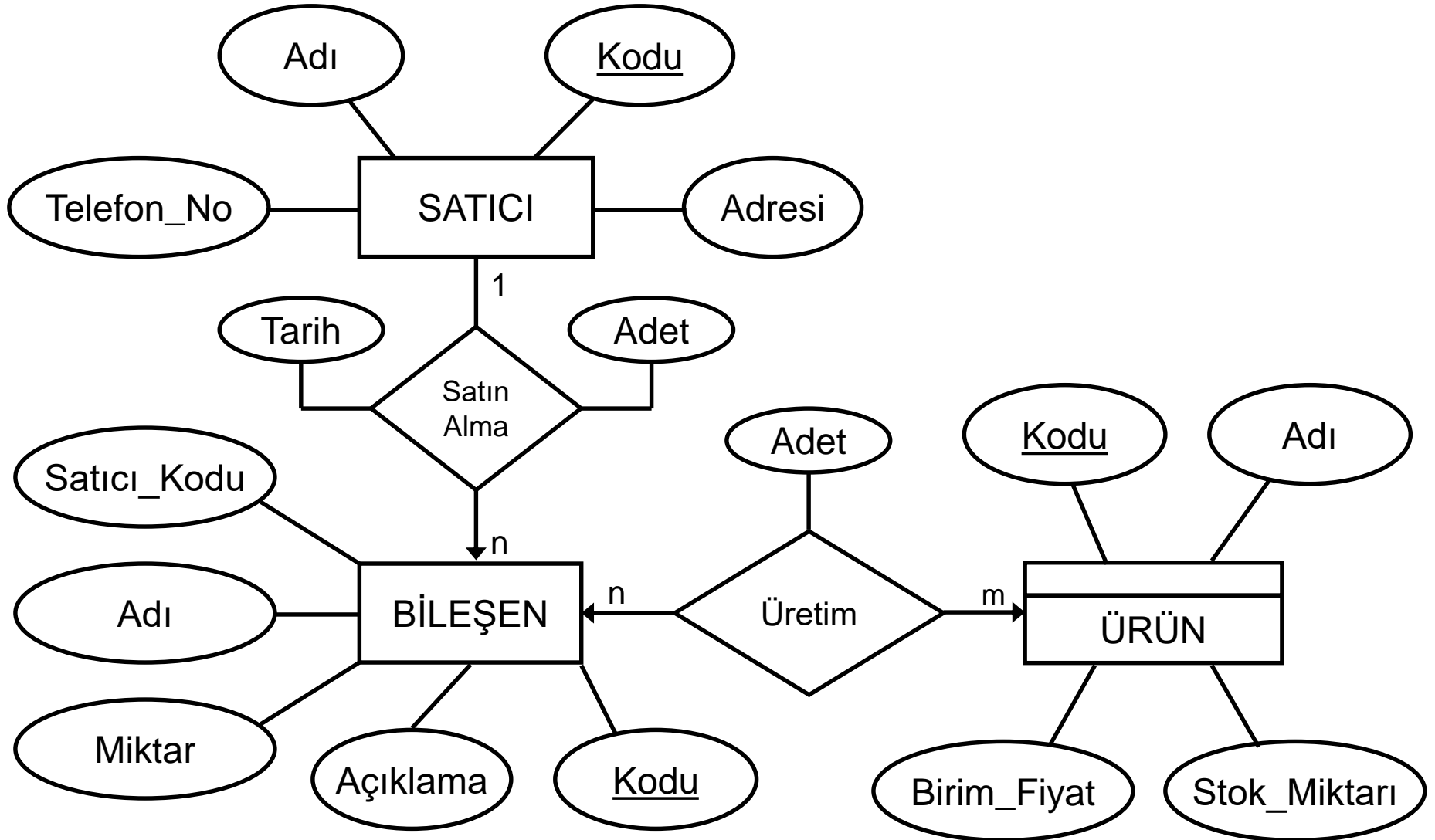


Çözüm 6

- Her ürün bileşenlere varolma bağımlıdır.
 - Bu durumda BİLEŞEN üstün (dominant) varlık, ÜRÜN ise bağımlı (subordinate) varlıktır.



Çözümün Tamamı

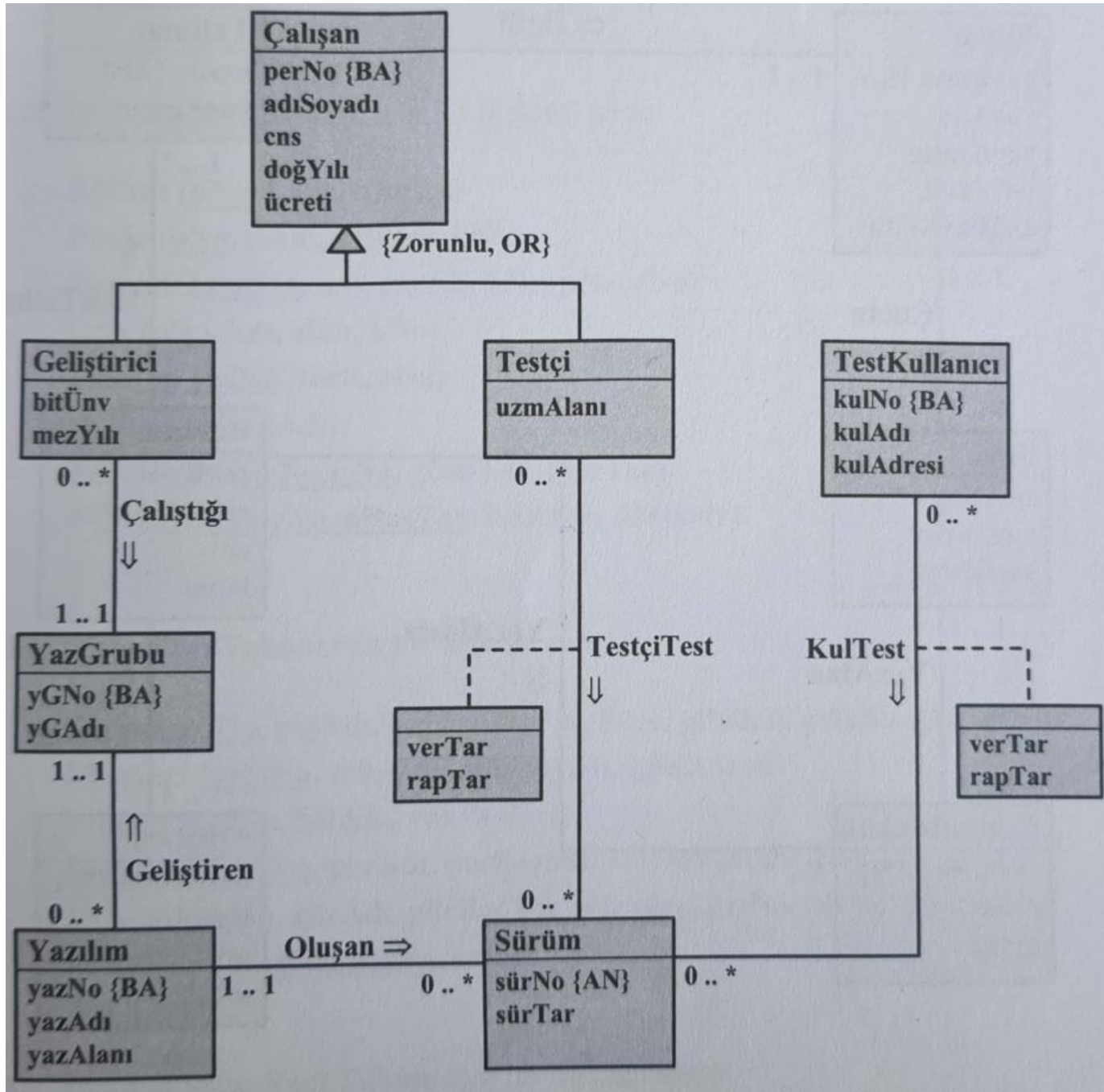


E-R Örneđi - Yazılım Veri Tabanı

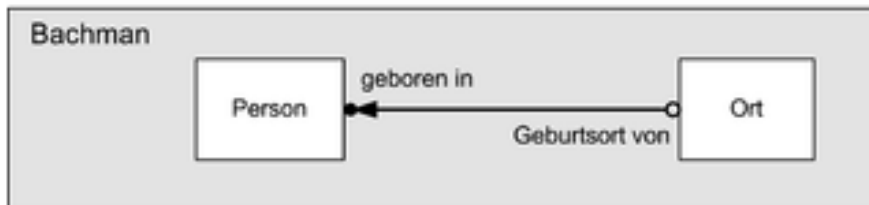
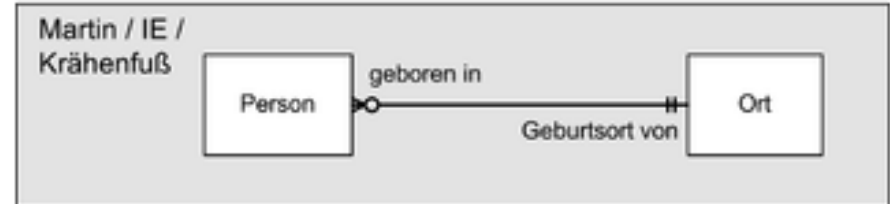
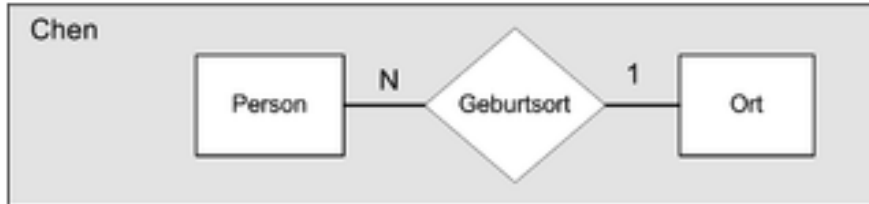
Paket yazılım üreten bir firma için, yazılımlar satışa sunulmadan önceki test sürümleri ve test çalışmaları ile ilgili verilerin yer alacağı bir **Yazılım Veri Tabanı** oluşturulmak isteniyor. Oluşturulacak veri tabanında yer alması istenilen veri ve bağıntılarla ilgili kısa açıklamalar aşağıda yer almaktadır.

- Firmada çalışan her teknik personelin numarası, adı ve soyadı, cinsiyeti, doğum yılı ve ücreti. Firmada çalışan her teknik personel ya geliştirici (bilgisayar mühendisi) ya da testçi (test görevlisi) olarak görev yapmaktadır. Veri tabanında her geliştiricinin bitirdiđi üniversite ve mezuniyet yılı; her test görevlisinin de uzmanlık alanı bilgisinin yer alması isteniyor.
- Geliştiriciler gruplar (yazılım grupları) halinde çalışmaktadırlar. Her yazılım grubunun bir numarası, bir de adı vardır. Her geliştirici bir ve yalnız bir yazılım grubunda çalışmaktadır. Testçiler ise bireysel olarak görev yapmaktadır.
- Geliştirilen her yazılımın numarası, adı ve alanı ile hangi yazılım grubu tarafından geliştirildiđi.
- Her yazılımın birçok test sürümü bulunuyor. Veri tabanında her test sürümün sıra numarası ile geliştirme tarihinin bulunması isteniyor.
- Yazılımların test çalışmaları testçilere ek olarak özel kullanıcılar (test kullanıcıları) tarafından yapılıyor. Her test kullanıcısının numarası, adı, ve adresi bulunuyor. Her yazılımın her sürümünü test etmek üzere belirli sayıda (0, 1 veya birçok) testçi ile belirli sayıda test kullanıcısı görevlendiriliyor. Her görevlendirme belirli bir tarihte yapılıyor. Görevlendirilen testçinin ve test kullanıcısının çalışmalarını tamamladıđında bir rapor hazırlayarak ilgili yazılım grubuna vermesi gerekiyor. Veri tabanında raporun veriliş tarihinin de bulunması isteniyor.

Yukarıdaki açıklamalar ışığında Yazılım Veri Tabanı'nın varlık-bağıntı çizeneđini oluşturunuz.



Alternativ Gösterimler



Toad Data Modeler

File Edit Objects View Notation Model Settings Tools Help

Physical Model Explorer

- Videorental
 - Workspaces
 - Entities
 - Relationships
 - Views
 - Functions
 - Procedures
 - Categories
 - Defaults
 - Domains
 - Note Lines
 - Notes
 - Check Constraint Rules
 - User Data Types
 - User Groups
 - Users
 - Sequences
 - Packages
 - Tablespaces

Physical Model Explorer

All Items Views

[1,1]

Customer

- Customer ID NN (PK)
- Name (IX1)
- Address

Order Record

- Customer ID NN (PFK)
- Film ID NN (PFK)
- Order Date

Customer Rating

- Title NN (PFK)
- Director NN (PFK)
- Rating

Film

- Film ID NN (PK)
- Title NN (AK1)
- Director NN (AK1)
- Production Company
- Genre ID NN (FK)

Borrowing

- Exemplar ID NN (PFK)
- Customer ID NN (FK)
- Start Date
- End Date
- Total Price
- VAT

Exemplar

- Film ID NN (FK)
- Exemplar ID NN (PK)

Relationships:

- Customer --- Borrowing (Makes)
- Customer --- Order Record (Places)
- Order Record --- Film (Is Required by)
- Customer Rating --- Film (Is Rated)
- Borrowing --- Exemplar (Is Related to)
- Film --- Exemplar (Has)

Display notations

- IE notation
- Customer
- Indexes
- Logical notation
- Data type

Project

Project	Videorental
Model	Videorental
Author	Rad
Firm	Que
Version	15.3
Date of Creation	13.2
Last Change	13.2

Name: All Items DB: Oracle 10g

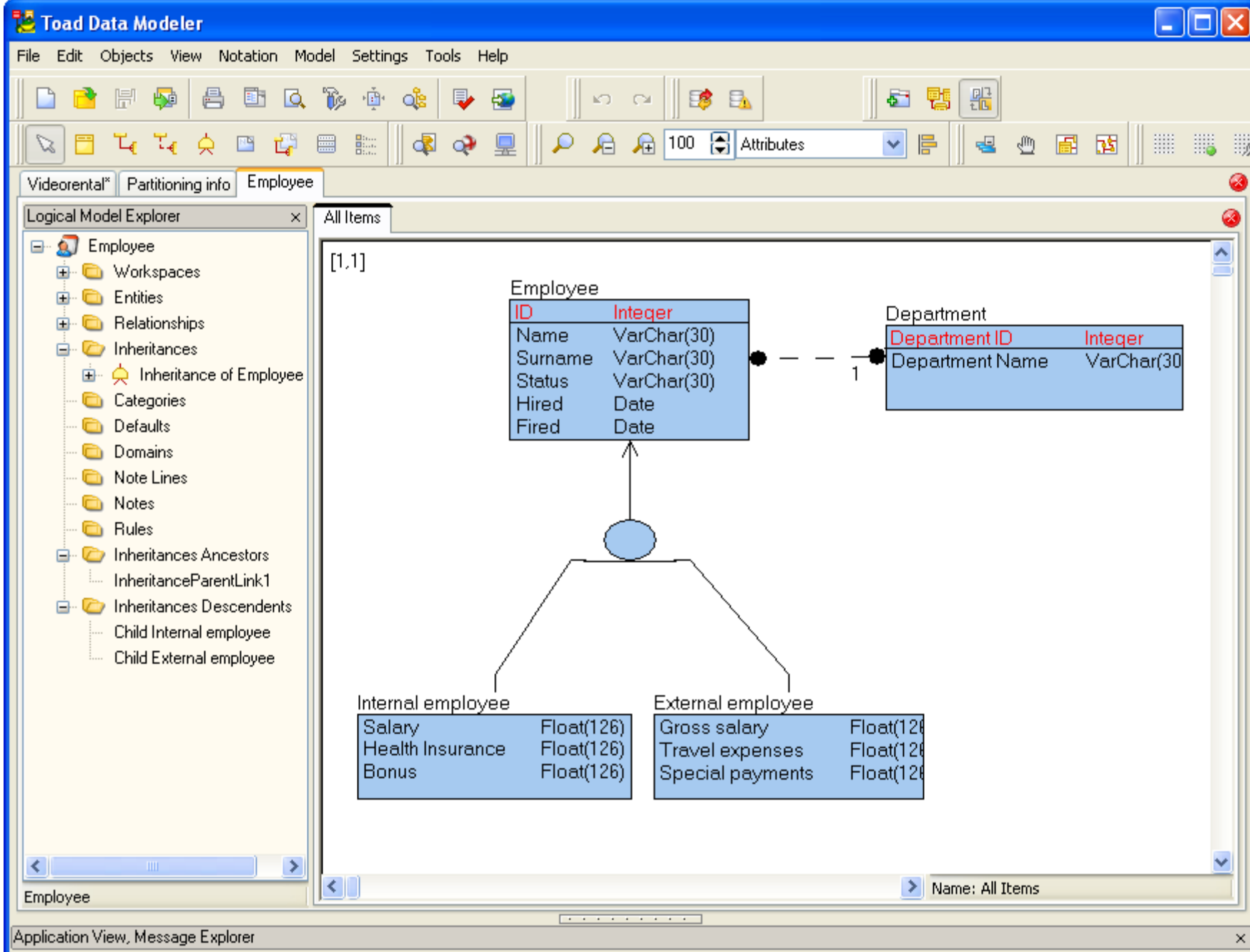
Application View, Message Explorer

Application View

- Videorental*
- Partitioning info

Message Explorer

Id	Date	Time	Message
3	11.3.2008	11:12:21	---
4	11.3.2008	11:12:21	Number of Errors: 0
5	11.3.2008	11:12:21	Number of Warnings: 0



İLİŞKİSEL VERİ MODELİ

- İlişkisel veri modelinin temelindeki düşüncüyü, bu modeli ilk kez geliştiren E. F. Codd'un bir makalesinde yazdığı şu cümleler çok güzel ifade etmektedir.
 - “Bilgisayar başındaki rastgele kullanıcı çoğunlukla tabloların yazılmasını ya da gösterilmesini ister. Tablodan daha yalın, daha evrensel, daha çok gereksinim duyulan ve daha kolay anlaşılan veri yapısı ne olabilir? Niçin bu tür kullanıcıların veri tabanındaki tüm veri ve ilişkileri tablolar biçiminde görmesine izin verilmesin?”

İlişki Kavramı

- İlişki (relation) kavramı bu modelde matematiksel bir kavramdır.
- Daha önce gördüğümüz Varlık-İlişki modelindeki ilişki ile karıştırılmamalıdır.
- Aslında İlişkisel Veri modelindeki ilişki, Varlık-İlişki modelindeki hem varlıkların hem de ilişkilerin yerini tutar.

Temel İlişki Yapısı

- Matematiksel ilişki kavramı aşağıdaki gibi tanımlanır:
 - Her biri birer değerler kümesi olan D_1, D_2, \dots, D_k alanları (domains) göz önüne alınsın.
 - Bu alanların kartezyen çarpımı $D_1 \times D_2 \times \dots \times D_k$ ile gösterilir ve 1. elemanı D_1 alanından, 2. elemanı D_2 alanından, .. , k. elemanı ise D_k alanından alınan $[d_1, d_2, \dots, d_k]$ k-çoklular kümesidir.
 - Bu k alan üzerinde tanımlanan her r ilişkisi, bu alanların kartezyen çarpımının bir altkümesidir:

$$r \subseteq D_1 \times D_2 \times \dots \times D_k$$

Tablolar ile Gösterim

- Her ilişki iki boyutlu bir tablo olarak gösterilir.
- Tablonun her sütununa bir nitelik atanır.
- Tablonun her satırı ise bir kaydı gösterir.
 - 1. elemanı 1. niteliğin değer alanından, 2. elemanı 2. niteliğin değer alanından, .. , k. elemanı ise k. niteliğin değer alanından alınan ve k değerden oluşan bir çoklu değerler topluluğu kayıt olarak adlandırılır.

Tablolar ile Gösterim

- Tablo, niteliklerin değer alanlarının kartezyen çarpımının bir altkümesidir.
- Bu tanıma göre tablo matematiksel anlamda bir ilişkidir.
- İlişkisel model, temel yapı olarak, matematiksel bir yapı olan ilişki yapısını kullanır.
- Böylece modeli sağlam bir matematiksel temele oturtarak gerek tasarım kriterlerinde, gerekse kullanıcı dillerinde bu kuramdan yararlanır.

Tablolar ile Gösterim

- Matematiksel ilişki yapısı, özde hiçbir değişiklik yapmadan, kullanıcının alışık olduğu ve görsel bir yapı olan tablo yapısı olarak sunulur.
- Buna göre ilişkisel veri modeli açısından ilişki ve tablo terimleri eşanlamlıdır.
- Bilimsel kesimde daha çok ilişki terimi kullanılırken kullanıcılar arasında daha çok tablo terimi kullanılmaktadır.
- Her ilişkinin (tablonun) tekil bir adı vardır.

Tablolar ile Gösterim

- **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), yukarıdaki tanıma uygun olarak oluşmuş k-çoklular kümesi ya da bir tablodur.
- Veri tabanı ilişkisi ile matematiksel ilişki arasındaki başlıca fark, veri tabanı ilişkisinin kullanıldığı ortama ve zamana göre değişmesine karşın matematiksel ilişkinin evrensel olması, ortama ve zamana göre değişmemesidir.

Örnek

- $D_1 = \{a, b, c\}$ $D_2 = \{1, 2, 3\}$ $D_3 = \{x, y\}$
olsun. Bu üç alanın kartezyen çarpımı
 $3 \times 3 \times 2 = 18$ adet 3-çokludan oluşan bir
kümedir.

$$D_1 \times D_2 \times D_3 = \{ [a,1,x], [a,2,x], [a,3,x], [a,1,y], [a,2,y], [a,3,y], [b,1,x], [b,2,x], [b,3,x], [b,1,y], [b,2,y], [b,3,y], [c,1,x], [c,2,x], [c,3,x], [c,1,y], [c,2,y], [c,3,y] \}$$

Örnek

- 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.

$$r = \{[a,2,x], [a,2,y], [a,3,y], [b,1,x], [c,1,y], [c,3,x]\}$$

Örnek

- Örnek olan r ilişkisi 3 kolon, 6 satırlı bir tablo olarak görülebilir.
- İlişkinin nitelikleri tablonun kolonlarına karşı gelir ve kolon başlıklarına niteliklerin adları yazılır. İlişkinin çokluları (kayıtları) ise tablonun satırlarına karşı gelir.

Nitelik İsimleri ←	A	B	C
Değer Alanları (Kayıtlar)	a	2	x
	a	2	y
	a	3	y
	b	1	x
	c	1	y
	c	3	x

İlişkinin (tablonun) Özellikleri

1. Satırların (kayıtların) sırası önemsizdir.
2. Tüm satırların birbirinden farklı olması gerekir (ilişki bir satırlar kümesi olduğuna göre aynı satır ilişkide birden çok kez yer alamaz).
3. Tüm satırlar birbirinden farklı olduğuna göre ilişkinin en az bir anahtarı (anahtar adayı) vardır. En kötü olasılıkla, tüm niteliklerin birleşimi anahtarı oluşturur.
4. İlişkinin her sütunu bir niteliğe karşı geldiğine ve sütun başlığında niteliğin adı yer aldığına göre, sütunların sırası da önemsizdir.

İlişkinin (tablonun) Özellikleri

5. Bir sütünde yer alan tüm değerler, belirli bir değer alanından gelen değerler olup tümünün türü aynıdır.
6. Tabloda yer alan her değer bir sayı, bir ad, bir miktar, bir tutar, bir adres, bir tarih,.. vb gibi yalın değerler olabilir. Tablonun bir ögesi bir dizi, bir vektör, bir başka tablo ya da karmaşık bir değer olamaz. Başka bir deyişle tablonun nitelikleri yalın ve tek değerli niteliklerdir.

İlişki Anahtarları

- İlişki kuramına göre her ilişkinin mutlaka bir anahtarının bulunması gerekir.
- Bir ilişkinin niteliklerinin bir ya da birkaç altkümesinin değerleri tüm kayıtlarda birbirinden farklıdır.
- Bu özelliğe sahip nitelik altkümelerine ilişkinin süper anahtar'ı denir.
- Süper anahtarlar ilişkideki kayıtları (satırları) tekil belirleme özelliğine sahiptir.

İlişki Anahtarları

- Eğer bir süper anahtarda “artık nitelik” yoksa (süper anahtarı oluşturan nitelik grubunun hiçbir öz altkümesi ilişkinin bir süper anahtarı değilse) bu süper anahtara aday anahtar veya ilişki anahtarı, ya da kısaca anahtar denir.
- Uygulama açısından önemli olan süper anahtar değil ilişki anahtarıdır.

İlişki Anahtarları

- İlişki anahtarı hem ilişkideki kayıtları tekil belirleme özelliğine sahip olan hem de içinde “artık nitelik” bulunmayan bir nitelik grubudur.
- En kötü koşullarda ilişki anahtarı tüm niteliklerden oluşur.
- Her ilişkinin anahtarının ya da anahtarlarının bilinmesi son derece önemlidir.

Birincil Anahtar

- İlişkisel veri tabanında bir tablonun benzer değerler içermeyen (tekil : unique) bir sütunu ya da birkaç sütunu birlikte **birincil anahtar** (primary key - PK) olarak tanımlanabilir.
- Birincil anahtar bir aday anahtardır ve söz konusu varlığın kayıtlarını en iyi biçimde karakterize eden bir anahtardır.

Birincil Anahtar (Primary Key) ile Anahtar (Unique Key) Arasındaki Farklar

Primary Key	Unique Key
Birincil Anahtar, satırları benzersiz bir şekilde belirtmek için kullanılır.	Benzersiz Anahtar, ilgili nitelik/nitelikler için mükerrer girişleri önlemede kullanılır.
Bir tabloda yalnızca bir birincil anahtar vardır.	Bir tabloda birden fazla benzersiz anahtar olabilir.
NULL değerler kabul edilemez.	Benzersiz bir anahtar olması durumunda NULL bir değer kaydedebiliriz ancak yalnızca bir NULL değeri desteklenir.
clustered index oluşturur.	Non-clustered index oluşturur.

Dış Anahtar

- Bir dış anahtar (foreign key - FK) bir sütun veya çok sayıdaki sütunların birleşiminden oluşur.
- Dış anahtar aynı tablo ya da başka bir tablodaki bir birincil anahtar ile eşleştirilir.

PK			FK		PK	
No	Adı	BölümNo			BölümNo	BölümAdı
25	Burak	10	←		10	Personel
13	Begüm	10			20	Muhasebe
28	Dilay	20	←		30	Satış

İlişki Şeması

- İlişkinin adı ile ilişki niteliklerinin adlarından oluşan tanım deyimine **ilişki şeması** denir.
- İlişki şemasının özet gösterimi $R(N_1, N_2, \dots, N_k)$ biçimindeki tanımdır (ayrıntılı ilişki şemasında ise niteliklerin değer alanları, veri türleri, bütünlük kısıtlamaları gibi ileride görülecek tanımlar da yer alır).
- İlişki şemasında genellikle ilgili niteliklerin altı çizilerek ilişki anahtarı gösterilir.

İlişkisel Veri Tabanı Şeması

- Eğer ilişkinin birden çok anahtarı varsa bunlardan en çok kullanılanı, en anlamlısı seçilir ve ilişki şemasında bu anahtar 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 Örneği

- ÖĞRENCİ (ÖĞRNO, ADI, SOYADI, CNS, DOĞ-TAR, ÖĞR-BNO)
- BÖLÜM (BNO, BADI, FAKÜLTE)
- DERS (DKODU, DADI, KRD, AÇ-BNO)
- ÖĞRDERS (ÖĞRNO, DKODU, DÖNEMİ, NOTU)

ÖĞRENCİ					
OGRNO	ADI	SOYADI	CNS	DOĞ-TAR	ÖĞRBNO
3	Ahmet	Korkmaz	Erk	17.04.1978	56
17	Ali	Harun	Erk	10.08.1977	55
26	Ayşe	Ak	Kad	29.09.1975	56

BÖLÜM		
BNO	BADI	FAKÜLTE
55	İşletme	İkt.ve İd.Bil.
56	Bilgisayar Müh.	Mühendislik

DERS			
DKODU	DADI	KRD	AÇ-BNO
BİL200	Mantıksal Tas.	3	56
BİL354	Veri Tabanı	3	56
İYB105	Muhasebe	2	55
İYB332	Gen.İşletme	3	55

ÖĞRDERS			
ÖNO	DKODU	DÖNEMİ	NOTU
3	BİL200	9394Bah	50
3	BİL354	9495Bah	80
3	İYB332	9495Güz	60
3	BİL200	9495Bah	55
17	İYB332	9495Güz	90
17	İYB105	9495Bah	75
26	BİL200	9394Bah	50
26	BİL200	9495Bah	60
26	İYB332	9495Güz	55