


Nesneye Yönelik Programlama

HAFTA 12

VERİ TABANI İŞLEMLERİ

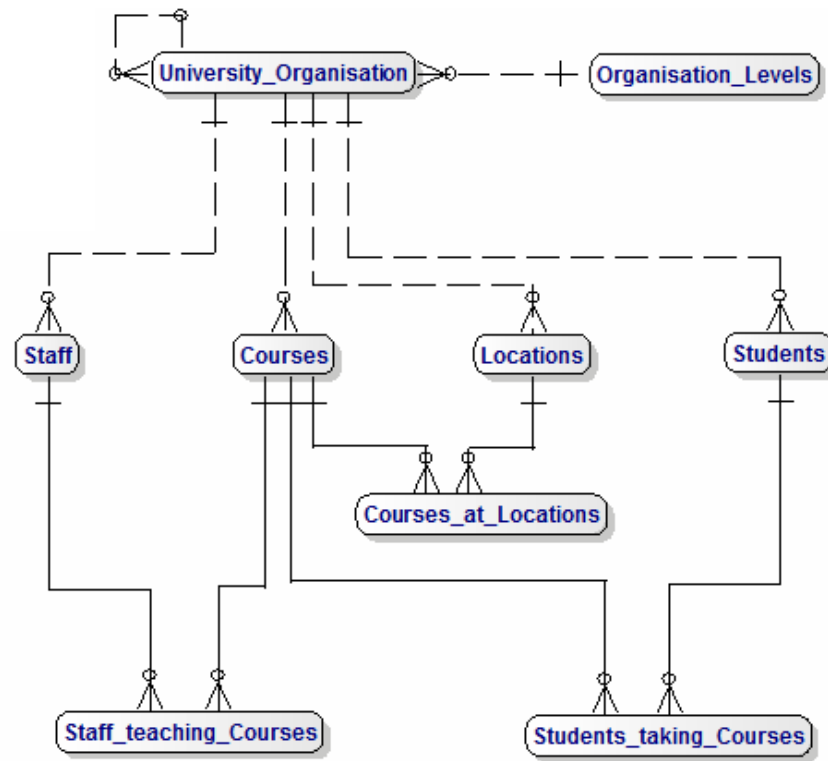
ER DİYAGRAMLARI VE NORMALİZASYON

Veritabanı Kavramı

- Veri  Bilgi
- Veritabanı Nedir?
- Veri tabanı:
 - zararlı veya gereksiz fazlalık olmadan birlikte depolanan ve birbiriyle ilişkili veriler
 - Kullanıcıya özgü uygulamalara hizmet eder
 - Verilerin programlardan bağımsız

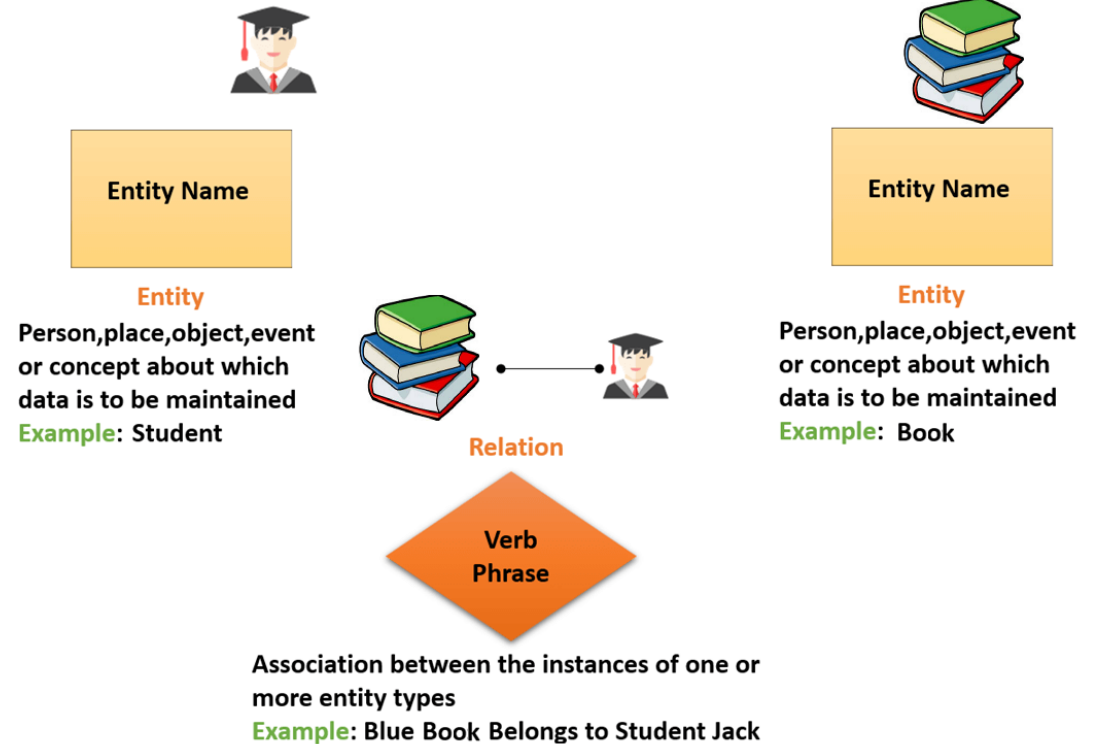
Veri Tabanı

- Gerçek dünya problemi
- Verinin Saklanması
- Veriler arasındaki bağlantılar
- İlişkisel Veri Modeli

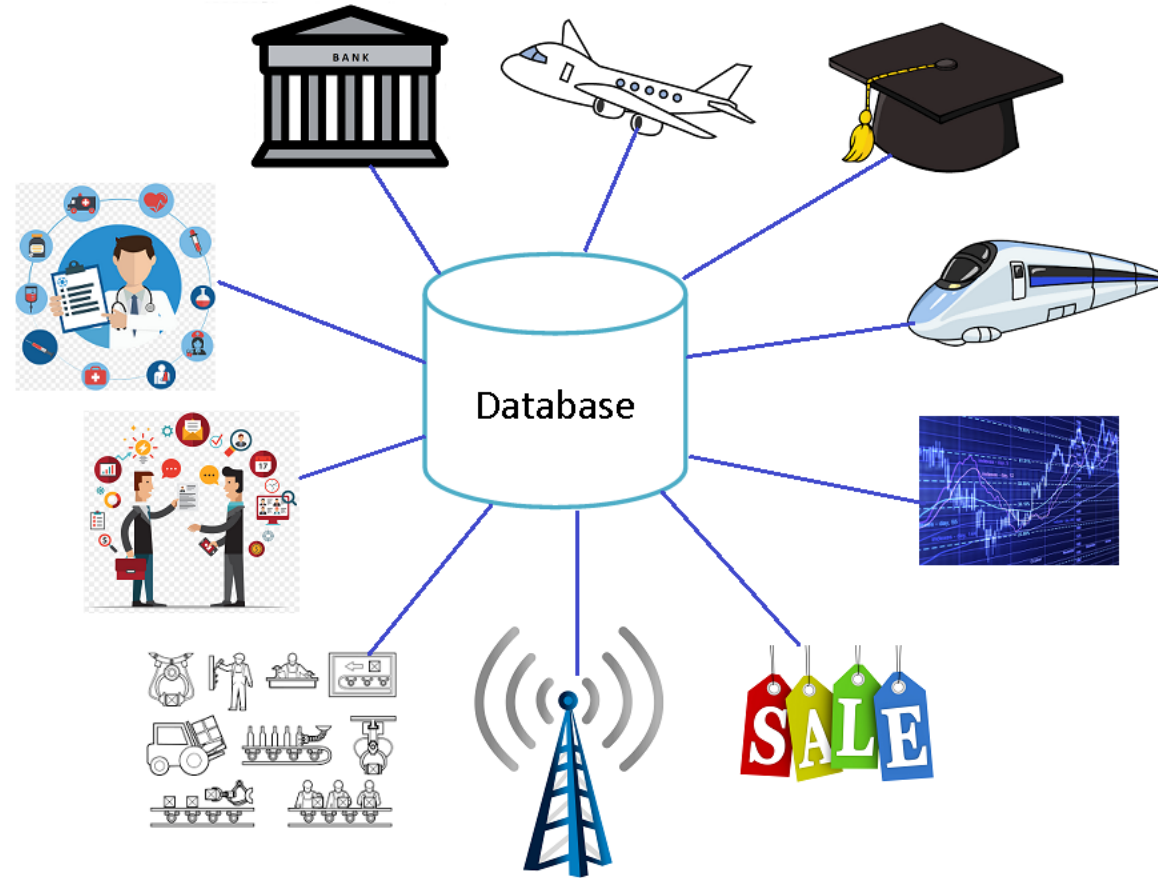


Örnek Veritabanı

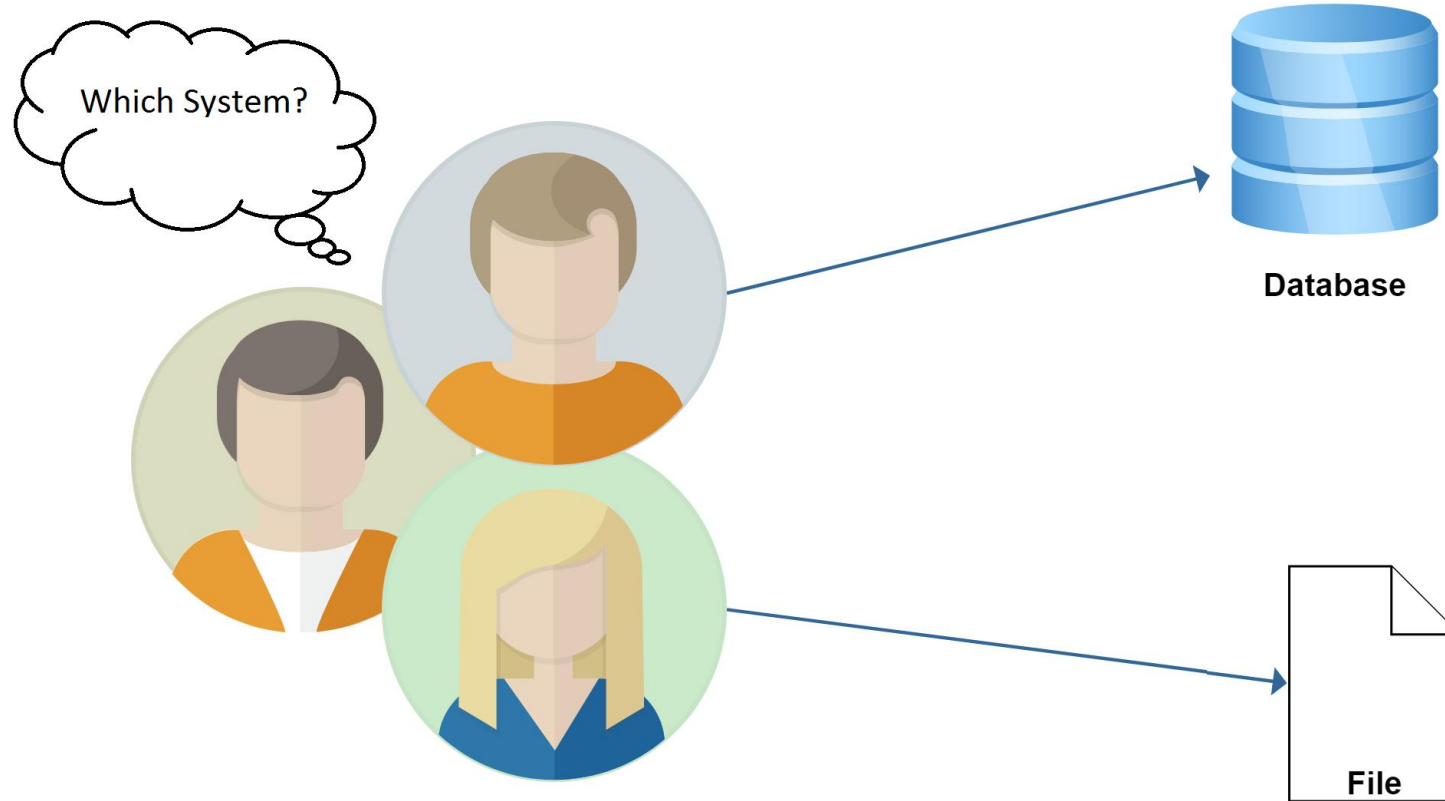
- Varlık olarak; öğrenciler, fakülteler, dersler, kitaplar ve sınıflar... Vb
- İlişki olarak ise; öğrencilerin derslere kayıt olmaları, fakülte dersleri ve odaların kullanımı.. Vb ilişkiler



Kullanım Alanları

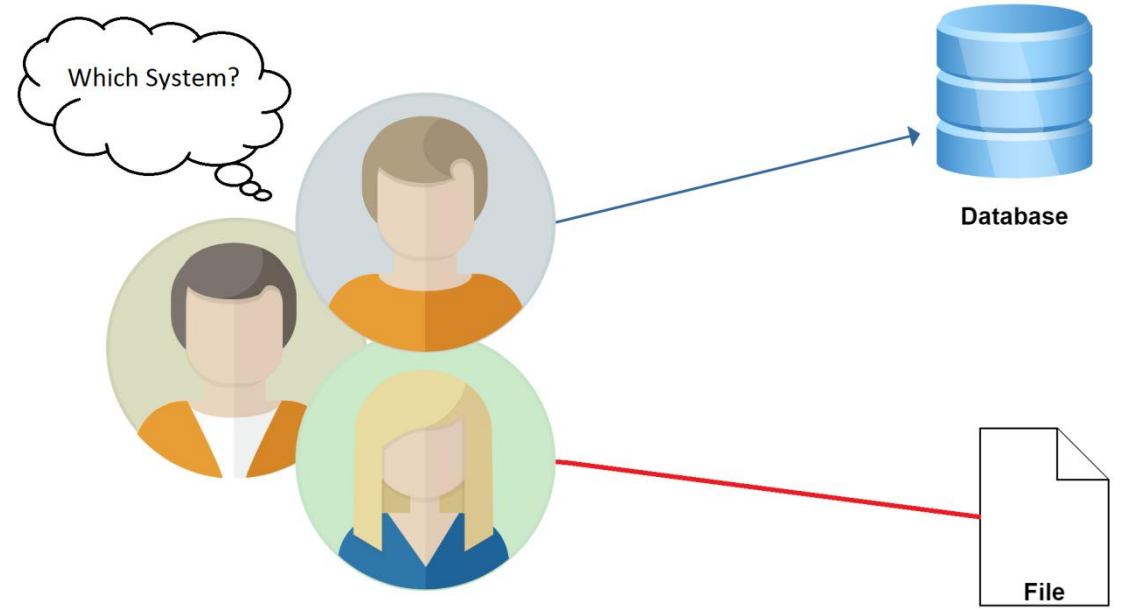


Senaryo: 500 GB veriye sahip bir işletme



Dosya ile Saklama

- Verinin parçalanması
- Her tür soru için farklı arama algoritmalarının tasarımı
- Tutarsız değişiklikler
- Veri kısıtlamaları



Dosya Sisteminin Dezavantajları

- Aşırı programlama Çabası
- Veri Tutarsızlığı
- Sınırlı veri paylaşımı
- Kötü veri kontrolü
- Yetersiz veri işleme yetenekleri
- Veri Artıklığı (veya çoğaltma)
- Atomiklik sorunları
- Güvenlik sorunları
- Bütünlük sorunları
- Program Veri Bağımlılığı
- Veri izolasyonu
- Verilere erişme zorluğu
- Eşzamanlı erişim problemleri


DBMS Avantajları

- Veri fazlalığının kontrol edilmesi
- Yetkisiz erişimi kısıtlama
- Yedekleme ve Kurtarma sağlanması
- Çok Kullanıcı Arabirim Sağlanması
- Bütünlük kısıtlamalarını desteklenmesi
- Verimli veri erişimi
- Veri paylaşımı iyileştirilmesi
- Gelişmiş güvenlik
- İyileştirilmiş veri tutarlılığı
- Verilerin Programdan Bağımsızlığı
- İyileştirilmiş veri kalitesi
- Program nesneleri ve veri yapıları için kalıcı depolama sağlanması
- Veriler arasındaki karmaşık ilişkileri temsil edilmesi
- Kurallar kullanarak çıkarsama ve işlem yapılmasına olanak sağlaması
- Tüm kullanıcılar için güncel bilgilendirme imkanı
- Esneklik
- Artan eşzamanlılık
- Çelişen talepler dengesi

DBMS Zorlukları/Dezavantajları

- Yedekleme ve Kurtarmanın Karmaşıklığı
- Artan kurulum ve yönetim maliyeti
- Ek donanım maliyeti
- Yeni ve uzmanlaşmış insan gücünün gerekliliği
- Artan karmaşıklık
- Merkezileştirme ile ilgili sorunlar
- Büyük DBMS boyutu

Veri Modeli Tasarlarken?

- Problem Ne
- Problemin yazınsal ve grafiksel gösterimi
- Genelden  Detaya
- Detayların şekilsel desteklenmesi



Veri Modeli Tasarlarken?

- Problemin yazınsal ve grafiksel gösterimi
- Veri yapılarının belirlenmesi (şemalar)
- Problemin içinde tanımlı kural yapıları
- Veri dönüşümleri için uygulanacak yöntem
- Nihai Tasarım: Deneme Yanılma



Problem Belirleme Adımları

- Firmanın kuralları
 - Yasal prosedürler
 - Standartlar
- İşin kuralları
 - İşin yapılmasında takip edilen iş planları
 - Hasta –ilaç: Hasta kaydı açılması
 - İşin müşteri/çalışan/diğer işler ile ilişkisi
 - Müşteri-Fatura
- İsimlendirme kuralları
 - Firmanın kullandığı özel terimler

Veri Modeli Yapı Taşları

○ Varlıklar

○ Kişi

○ Yer

○ Eşya

○ Olay

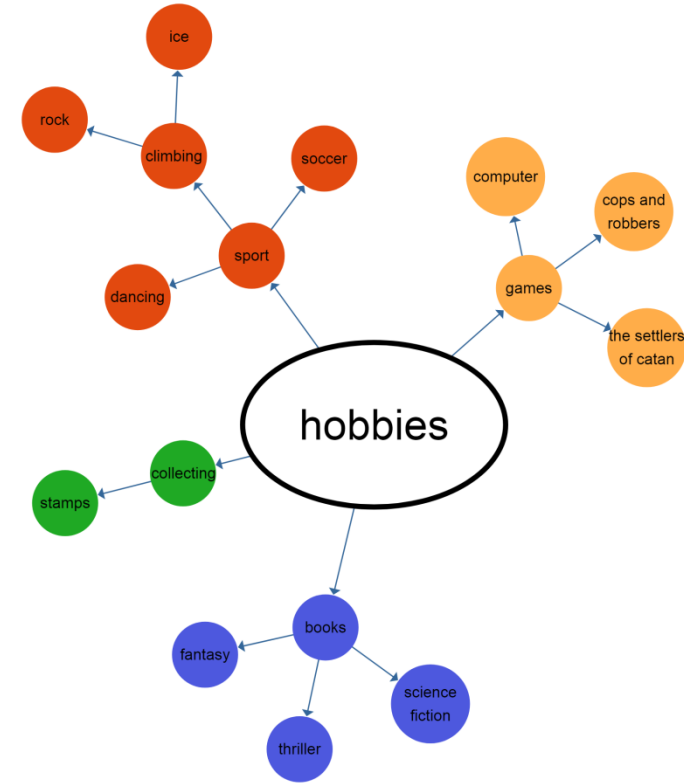
○ Nitelikler

○ Kişi :ad,soyad

○ Yer:Konum

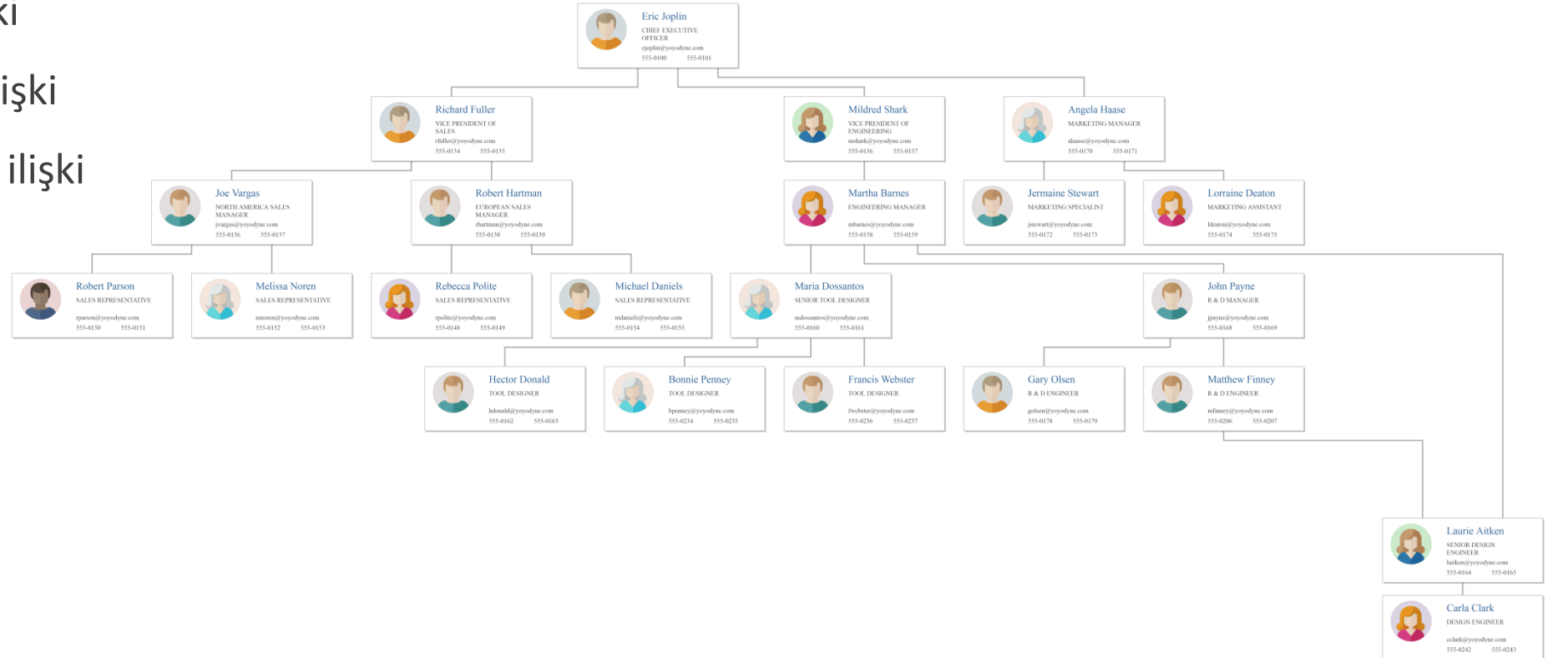
○ İlişkiler

○ Kısıtlamalar



İlişki Türleri

- Bire bir (1:1) ilişki
- Bire çok (1:M) ilişki
- Çokla Çok (N:M) ilişki

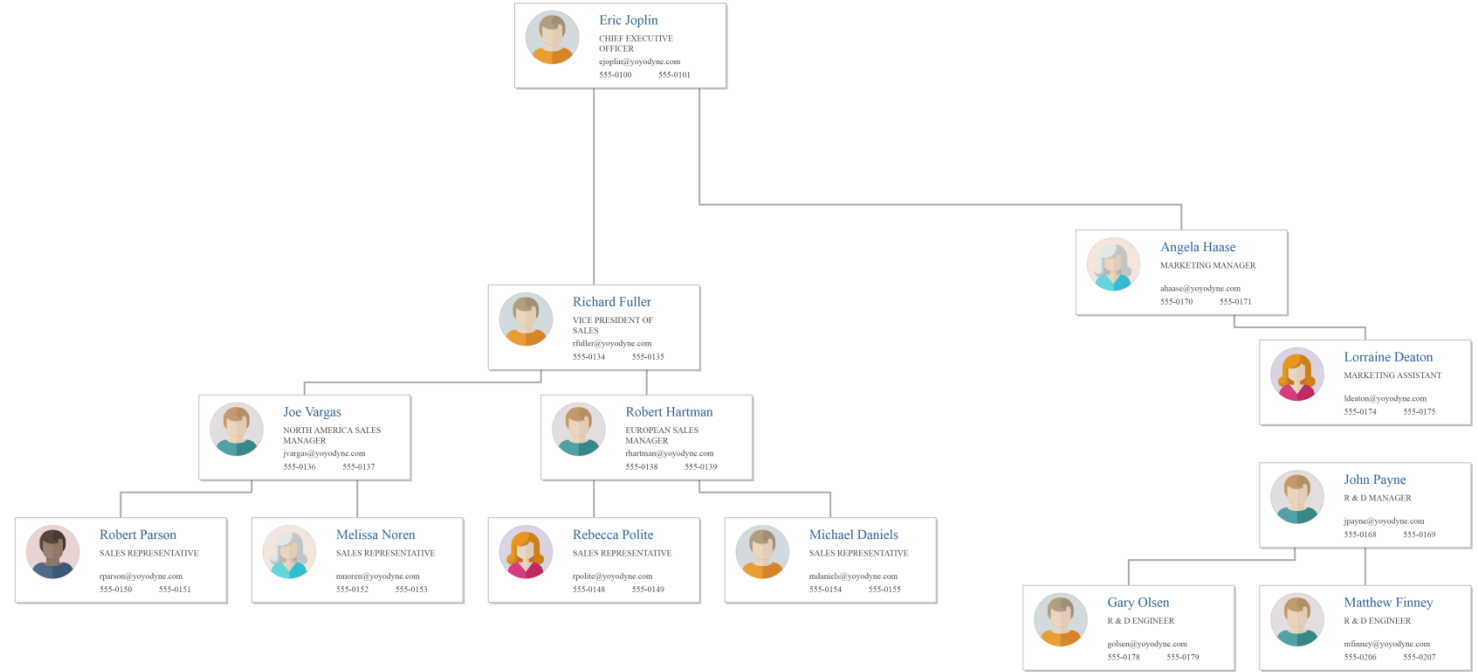


Veri Modeli Türleri

- Dosya Sistemleri
- Hiyerarşik Model
- Ağ Modeli
- İlişkisel(Relational) Model
- Varlık-İlişki(Entity Relationship) Modeli
- Nesne Tabanlı (Object Oriented) Model

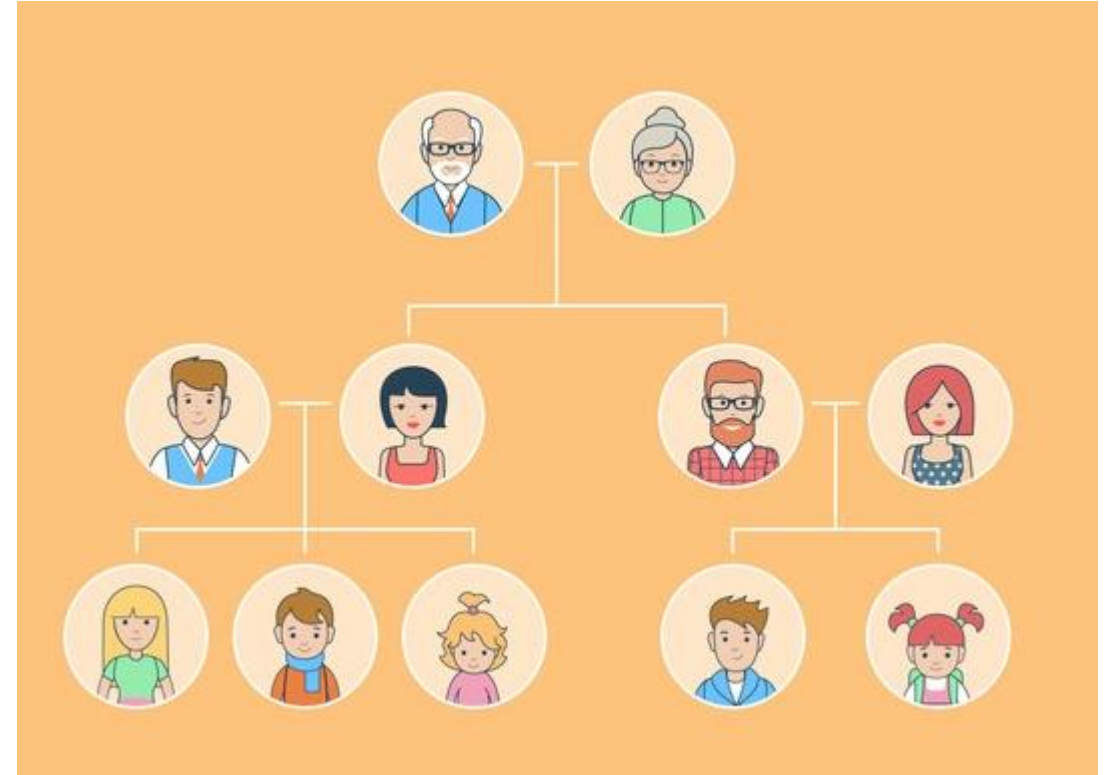
Hiyerarşik Model

- Ağaç yapısında:
 - Ebeveyn- çocuk ilişkisi
- Avantajları:
 - Basit
 - 1:M ilişki
 - Veri bağımsız
 - Veri Entegrasi kolay
- Dezavantajları:
 - Uygulamak zor
 - Esnek değil



Ağ Modeli

- Ağaç modelinde birden fazla root olması
- Avantajları:
 - Basit
 - 1:M ve N:M ilişkiler
- Dezavantajları:
 - Karmaşık
 - Kullanıcı dostu değil
- Bugün kullanılan Şema,DML,DDL gibi kavramların Tanımı Ağ Modeli sayesinde.



İlişkisel Model

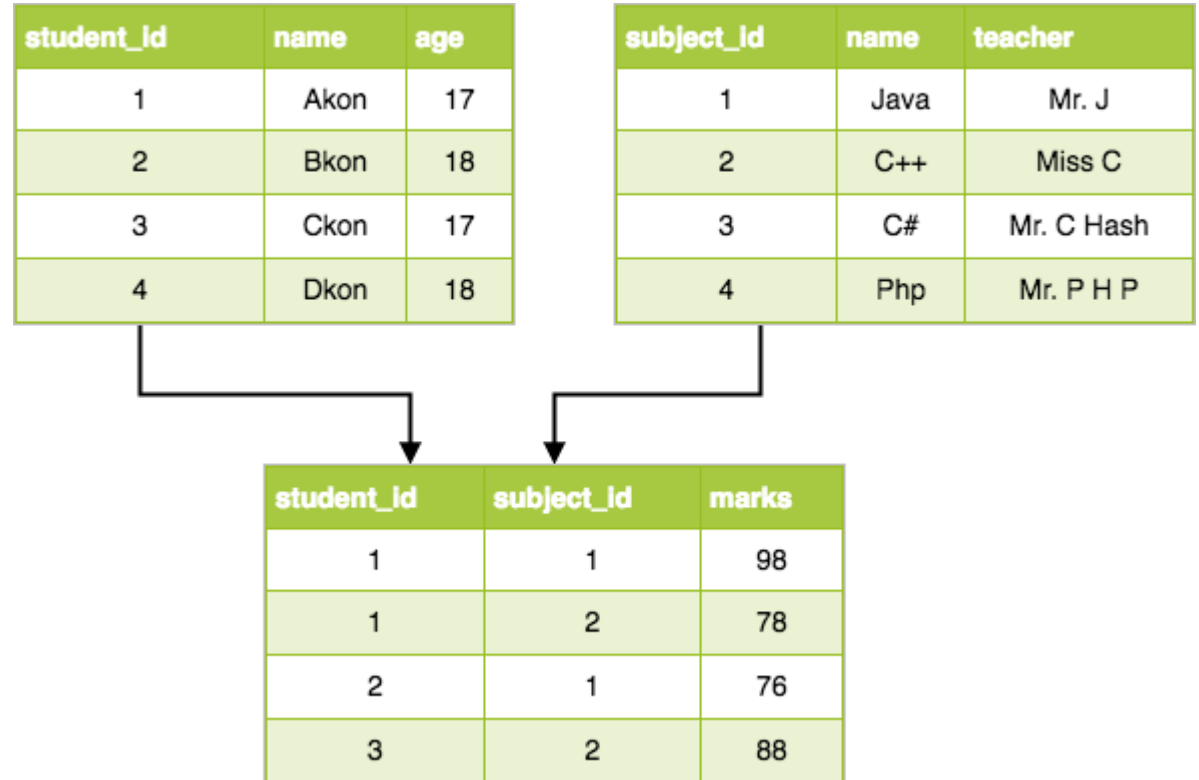
- Codd tarafından, 1970
- İlişkisel veri tabanındaki tüm bilgiler tablolardaki değerlerle tamamen tek ve tek bir şekilde temsil edilir.
- Her bir verinin (atomik değer), tablo adı, anahtar değeri veya sütun ismi gibi farklı parametreler kullanılarak erişilebilir olmalıdır.
- Eksik bilgiden kaynaklı oluşabilecek NULL değerler desteklenir.
- Veri tabanındaki açıklamalar; mantıksal seviyede, sıradan verilerle aynı şekilde gösterilir. Yani yetkili kullanıcılar, aynı ilişkisel dili kullanarak hem sıradan verilere hemde açıklamalara ulaşabilirler.
- İlişkisel bir sistem birkaç dili veya çeşitli metodolojileri destekleyebilir. Ancak veri tanımlama, görüntüleme, kısıtlama, yetkilendirme gibi çeşitli özelliklerin tanımlanabileceği en az bir dil olmalıdır.

İlişkisel Model

- Tüm görünümüler güncellenebilir olmalıdır.
- Yüksek seviyede; ekleme, silme, güncelleme işlemlerinin yapılabilir olmalıdır.
- Depolamada veya erişim yöntemlerinde bir değişiklik yapıldığında uygulama programlarında bir değişim yapılmasına gerek duyulmamalıdır.
- Tablo sütunlarında herhangi bir değişim yapıldığında uygulama programlarında bir bozulmanın olmaması gerekir
- Kısıtlamalar veritabanı tarafından tanımlanabilmelidir.
- Verinin hangi diskte saklandığı yada fiziksel olarak nerede bulunduğı ile ilgili işlemlerin DBMS tarafından yapılması gerekir.
- DBMS düşük seviye bir dil tarafından destekleniyorsa, yüksek seviyeli dil tarafında yazılmış kuralları uygulamak zorundadır.

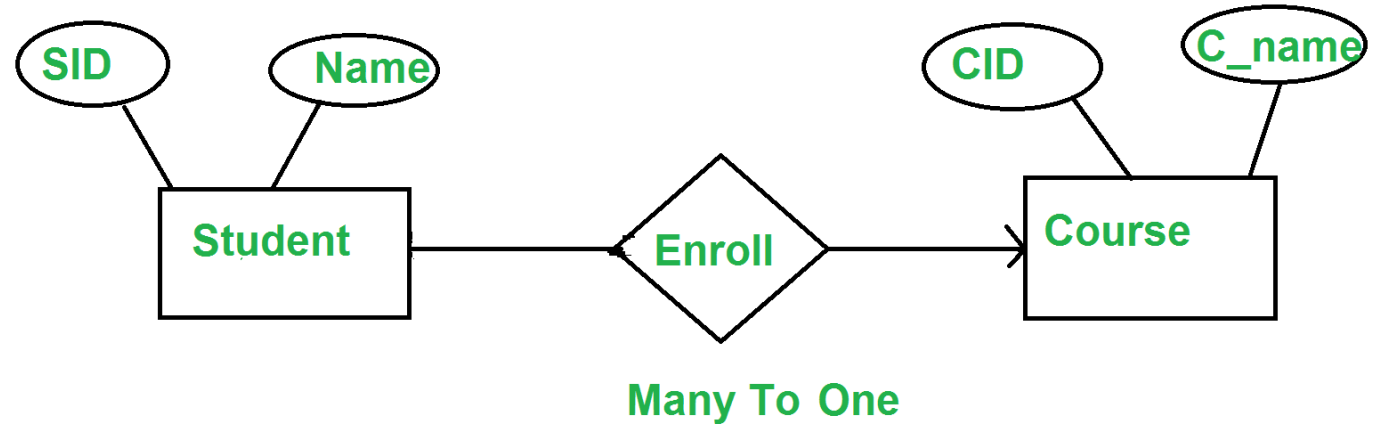
İlişkisel Model

- ilişkisel model en çok kullanılan veritabanı



ER Model

- Veritabanına geçiş sürecinde kolaylık sağlar
- Grafiksel gösterimi sayesinde anlaşılır, basit
- Varlık
 - Çalışan
 - Öğrenci
- Özellik
 - Çalışana yada öğrenciye ait isim
- İlişkiler
 - Çalışanın birimi
 - Öğrencinin aldığı ders

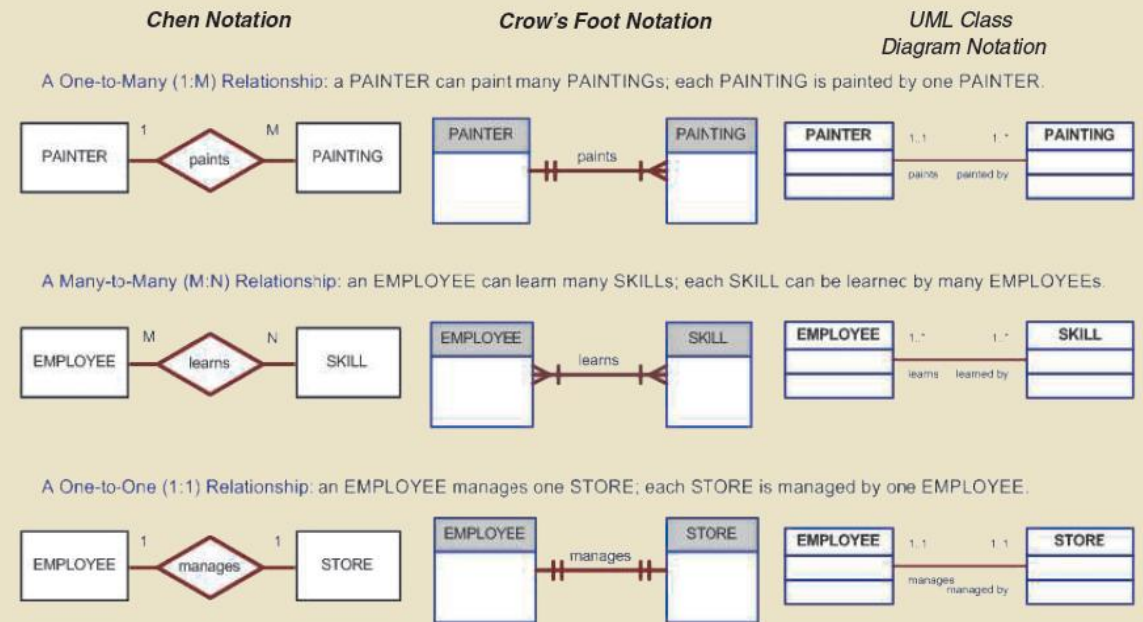


ER Model

3 Farklı gösterimi

- Chen
- Crow's Foot
- UML

FIGURE 2.3 THE ER MODEL NOTATIONS



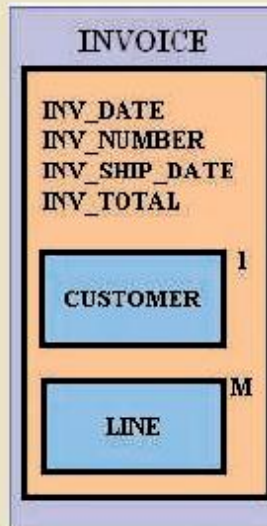
Nesneye Dayalı Model

- Er Modelindeki varlıklar, bu modelde Nesne olarak düşünülebilir
- Özellikler ise, nesnenin nitelikleridir.
- Benzer özellikteki nesneler gruplanır.
- Sınıflar hiyerarşik olarak düzenlenir.
- UML diyagramları kullanılarak gösterimler gerçekleştirilir.

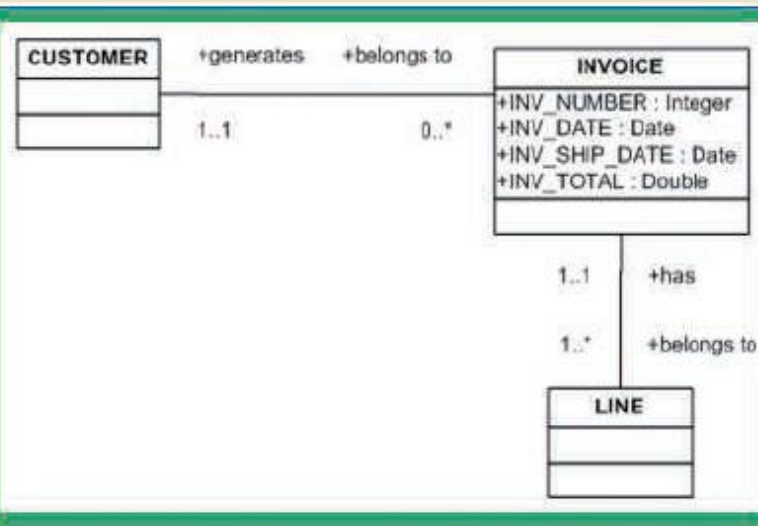
Nesneye Dayalı Model

FIGURE 2.4 A COMPARISON OF OO, UML AND ER MODELS

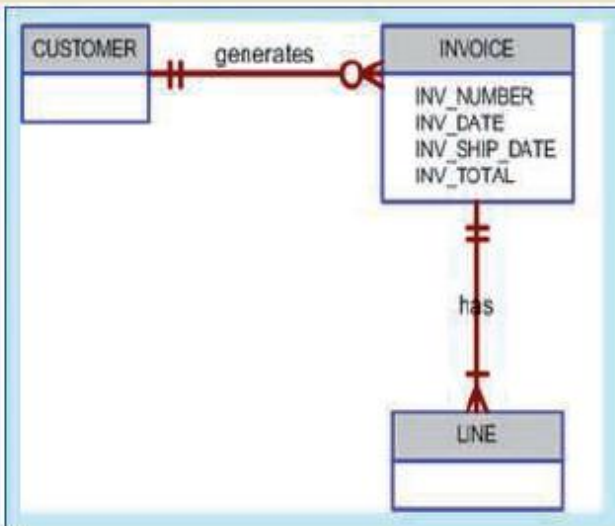
Object Representation



UML Class Diagram



ER Model



Entity Relationship (ER) Diyagramları

- Varlık (Entity)
 - Kişi
 - Olay
 - Yer
- Özellik (Attribute)
 - Kişiler – Ad, Soyad
 - Olay – Gerçekleşme koşulları
 - Yer – Konum
- İlişkiler (Relationship)
 - Kişi – Arkadaşlık ilişkileri
 - Olay – Zaman sıralaması
 - Yer – İki bölge arasındaki zaman farkı, Km farkı

Varlık(Entity)

- Ayırt edilebilen nesne
- Entity kümesi:
 - Aynı özellikleri veya niteliklere sahip
- Zayıf varlık
 - Başka bir varlığa bağımlı
 - Kısmen veya tamamen başka bir varlıktan türetilmiş anahtar
 - Çalışan – Bağlı olduğu üst
- Güçlü Varlık

Nitelikler (Attributes)

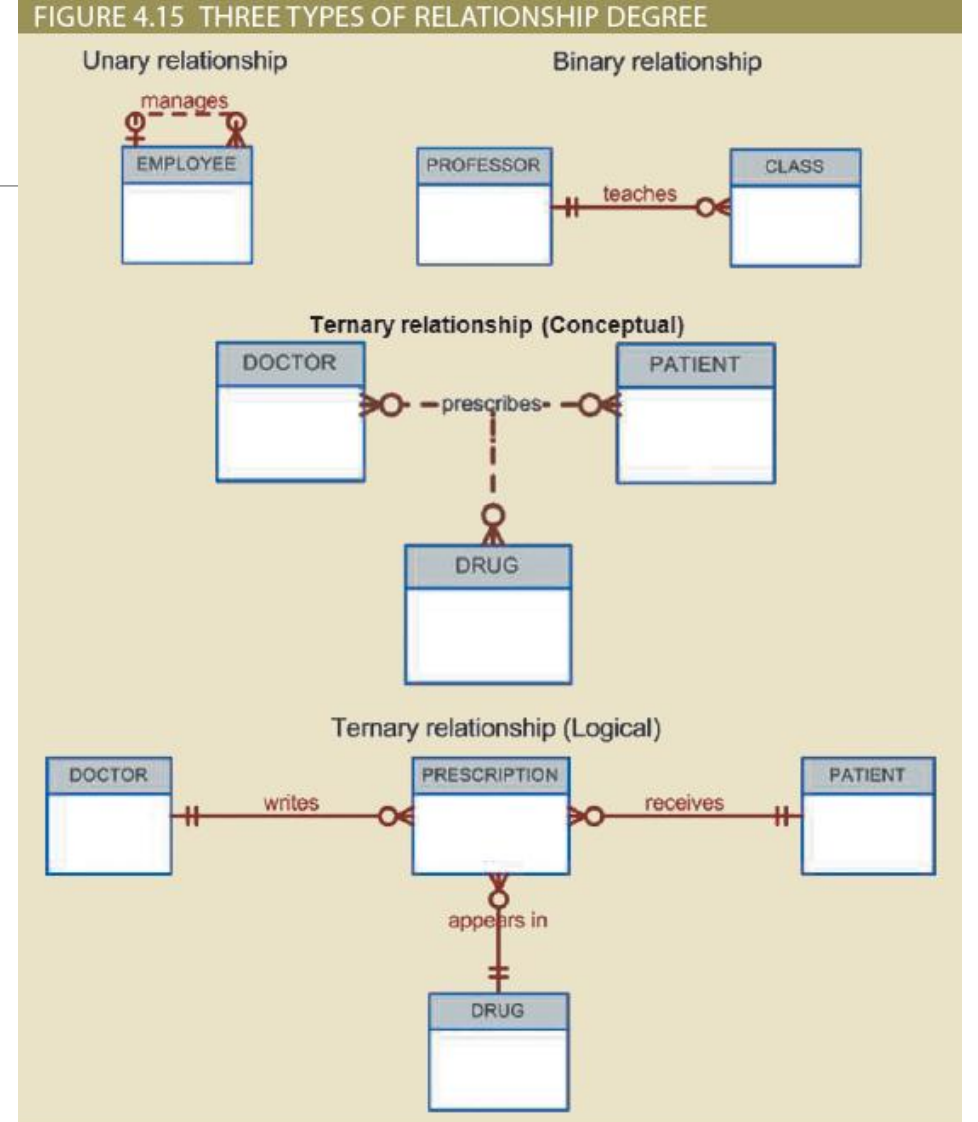
- Her varlık için temelde bulunan özellikler.
- Farklı değerler – Anahtar değer
- Türleri
 - Basit Nitelikler (Simple Attribute): Bölünemeyen özellikler
 - Maaş, Tc Kimlik
 - Bileşik Nitelikler (Composite Attributes): Bölünebilecek özellikler
 - Adres, Tarih
 - Tek Değerli Nitelikler (Single Valued Attributes): Tek bir değere sahip olunabilecek nitelikler
 - Yaş, Tc Kimlik
 - Çok Değerli Nitelikler (Multi Valued Attributes): Birden fazla değer alabilen nitelikler
 - E-Mail, Telefon
 - Boş Nitelikler (Null Attributes): Bir nitelik aynı varlık sınıfındaki diğer bireyler için boş olabilir
 - Sertifika bilgisi, Yabancı dil bilgisi
 - Türetilmiş Nitelikler (Derived attribute): Bir nitelik başka bir nitelikten türetilebilir
 - Net maaş bilgisi, brüt maaş ve vergi oranları gibi bilgilerden üretilir

İlişkiler (Relationships)

- İlişki Türü
 - Bire bir (1:1) ilişki
 - Bire çok (1:M) ilişki
 - Çokla Çok (N:M) ilişki
- İlişki Kuvveti
- Zayıf ilişkiler
 - Ders ve Dersin yapıldığı fiziksel sınıf arasındaki ilişki
- Güçlü İlişkiler
 - Ders ve Dersin Şubesi arasındaki ilişki

İlişkiler (Relationships)

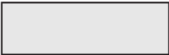
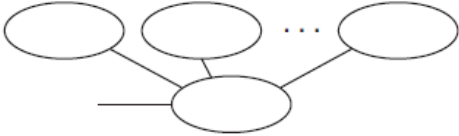
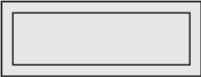

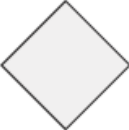
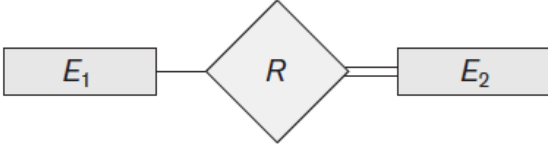

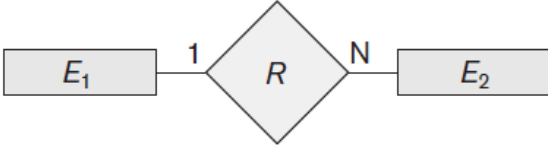

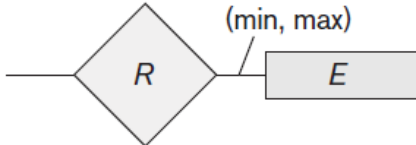

- İlişki Dereceleri
 - Unary İlişkiler
 - Çalışan – Yönetici
 - İkili İlişkiler
 - Ders – Öğretim Üyesi
 - Üçlü veya daha fazla ilişkiler
 - Doktor – Bir veya daha fazla Reçete
 - Hasta – Bir veya daha fazla Reçete
 - İlaç – Bir veya daha fazla Reçete



Kısıtlamalar/Sınırlamalar (Constraints)

- İlişki kısıtlamaları (Mapping Cardinalities)
 - Bölüm – Bölüm Yöneticisi
 - Çalışan – Bölüm
- Katılımlı Kısıtlamalar (Participation Cardinalities)
 - Bir varlık, ilgili olduğu tabloya eklendiği zaman ortaya çıkan kısıtlamalar
 - Banka: Müşteri – Kredi
 - Total Participation
 - Her Krediye karşılık bir müşteri
 - Partial Participation
 - Her Müşterinin bir kredisi yok

ER Diyagramı Sembolleri

Symbol	Meaning		
	Entity		Composite Attribute
	Weak Entity		Derived Attribute
	Relationship		Total Participation of E_2 in R
	Identifying Relationship		Cardinality Ratio 1 : N for $E_1 : E_2$ in R
	Attribute		Structural Constraint (min, max) on Participation of E in R
	Key Attribute		

Örnek: Firma Modellemesi

Proje bazlı bir firmada aşağıdaki özellikleri saklayıp, farklı niteliklerde raporlama işlemlerinin yapılabilceği bir veritabanı geliştirilmesi istenmektedir.

○Çalışan

- Çalışanlara ait ad, soyad, isim, doğum tarihi, eş durumu, maaş gibi farklı bilgiler saklanmalıdır.
- Çalışanlar proje bazlı çalışmaktadır.
- Bir çalışan maaşlı yada saatlik çalışabilir.
- Bir çalışan başka bir çalışana bağlı olarak çalışabilir.
- Her çalışanın bağlı olduğu bir bölüm bulunmaktadır.
- Çalışanlar farklı rollerde bulunabilir: saatlik çalışan, maaşlı çalışan, bölüm şefi.. vb

○Bölüm

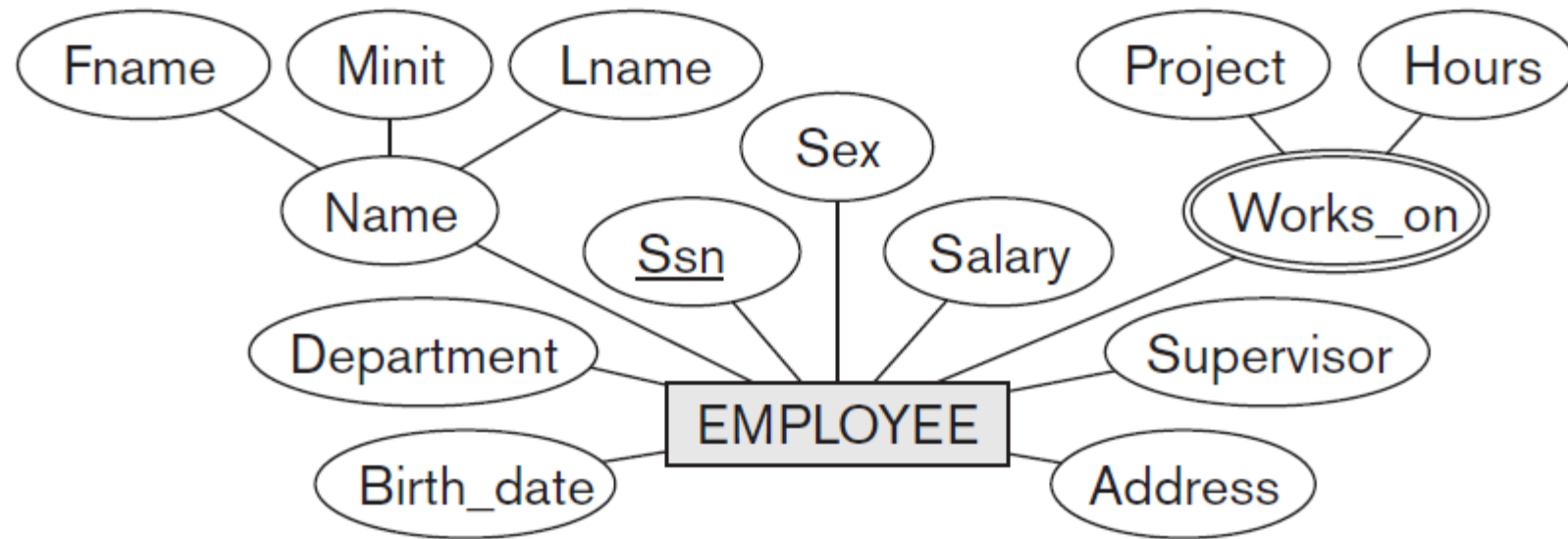
- Her bölümün bir birinden ayrı görevleri vardır.
- Bir bölümde birden fazla çalışan olabilir.
- Her bölümün kontrol ettiği farklı projeler vardır.
- Her bölümün bir tane şefi vardır.

○Proje

- Her proje farklı konularda geliştirilmektedir.
- Projede çalışan personel sayısı birden fazla olabilir.
- Proje birden fazla bölüm tarafından kontrol edilebilir.

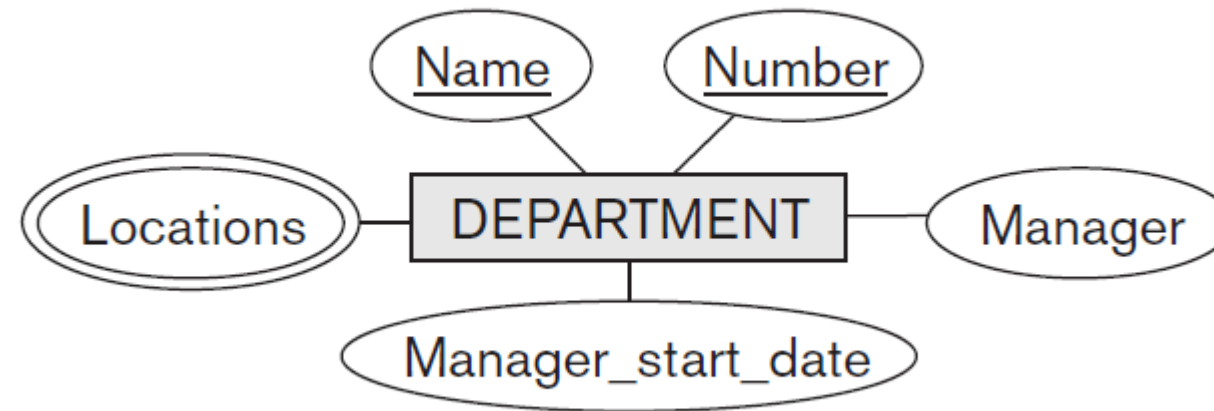
Çalışan

Çalışan



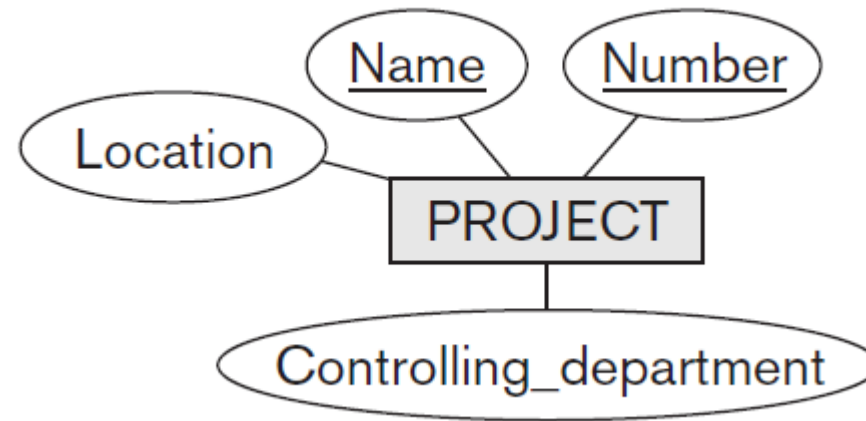
Bölüm

Bölüm



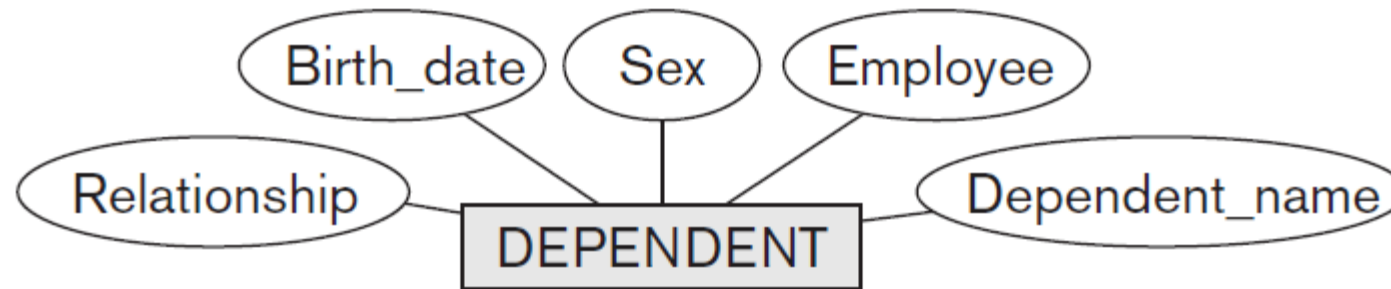
Proje

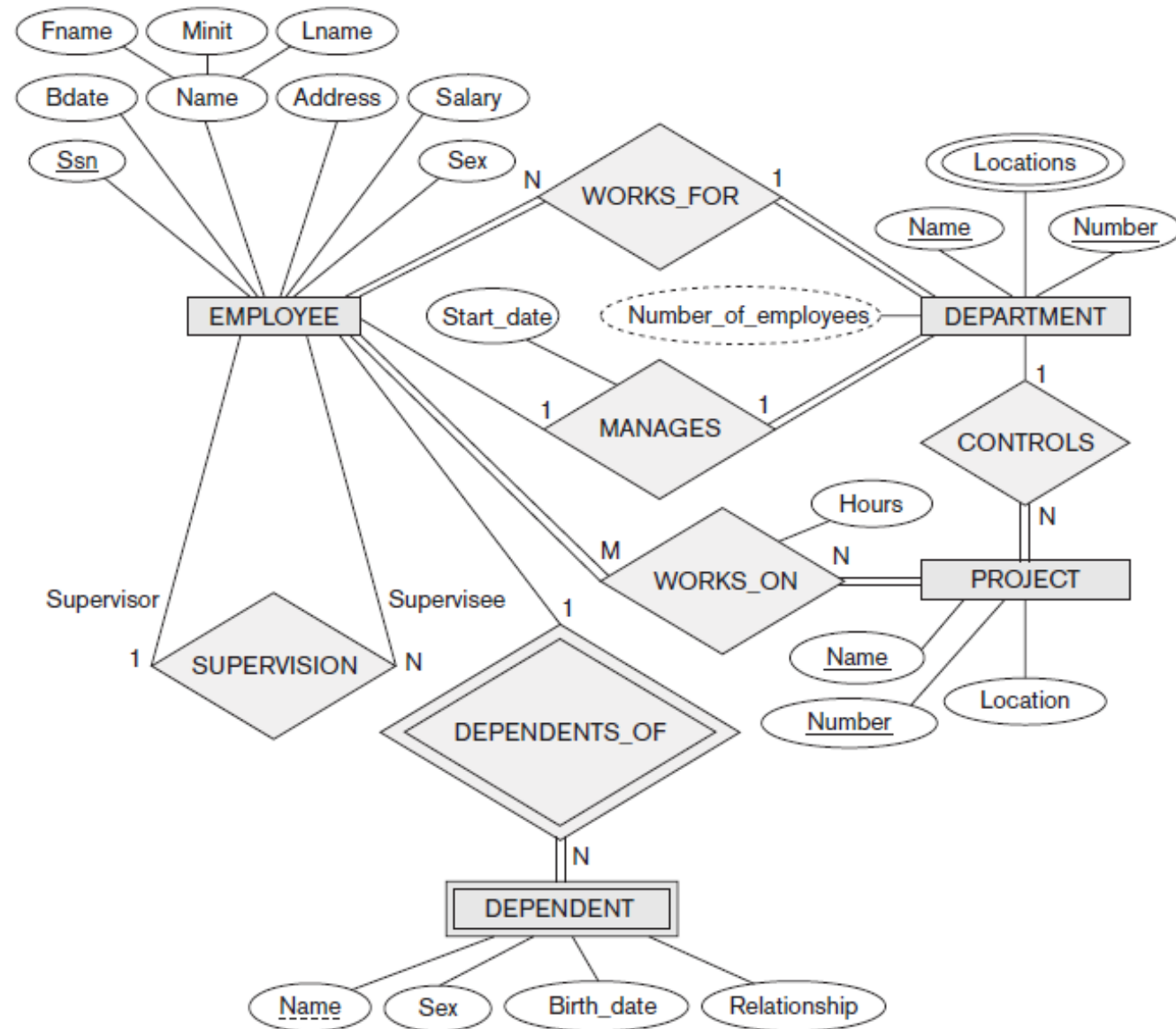
Proje

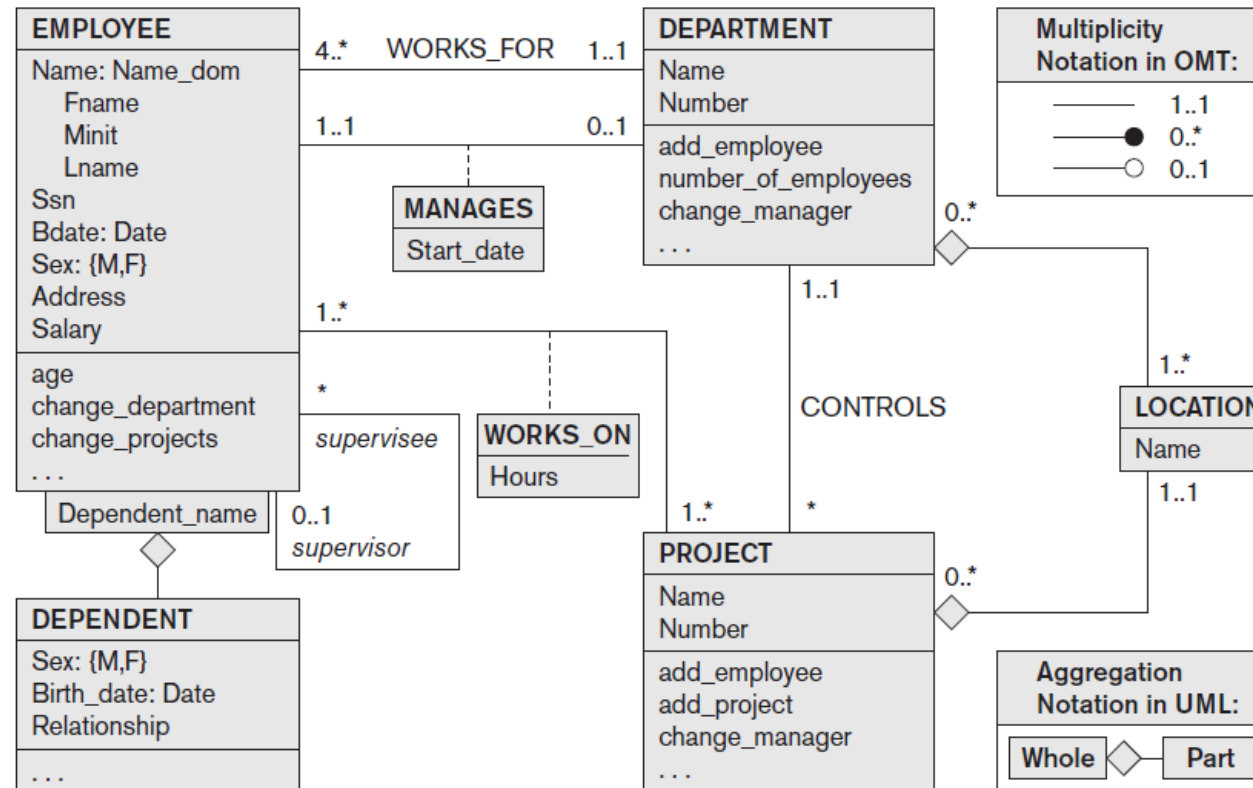


Dependent

Dependent





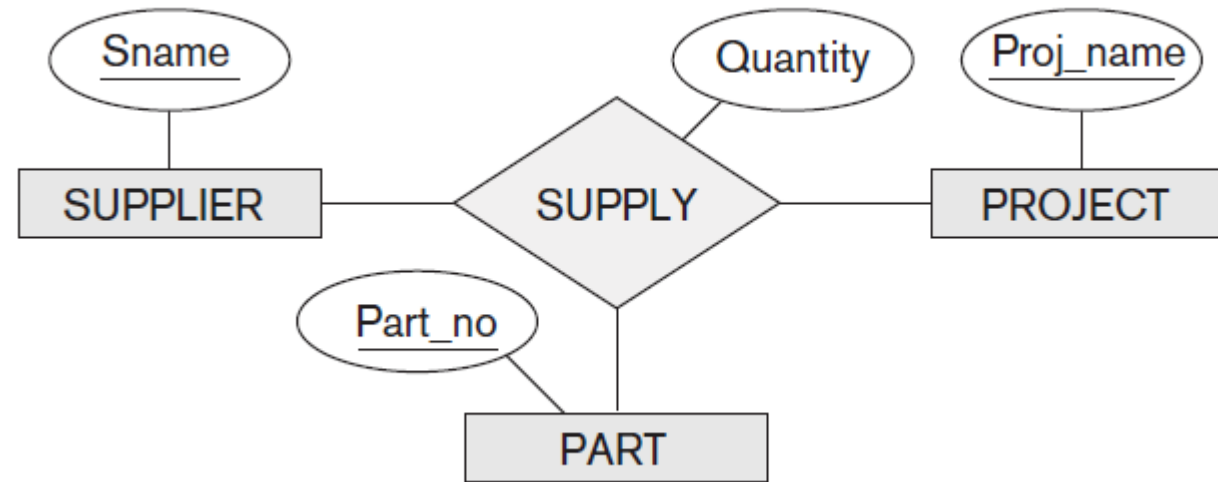


Ürün tedarik edilirse?

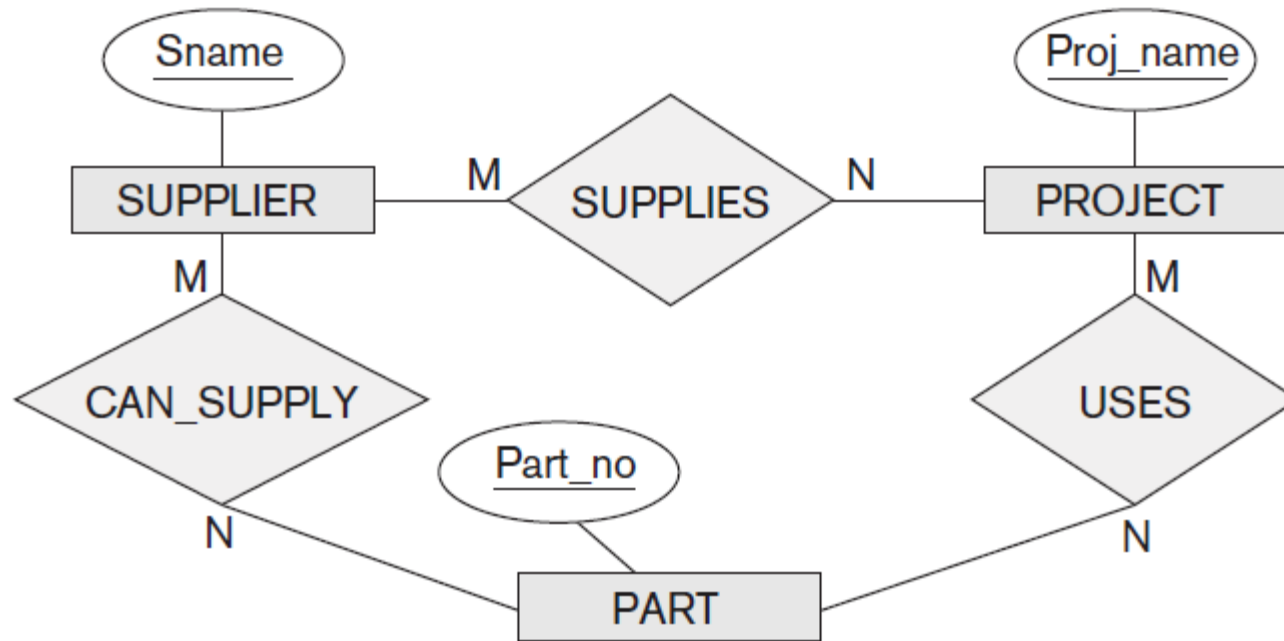
- Tedarikçi
- Tedarik Edilen Ürün
- Tedarik edilen proje



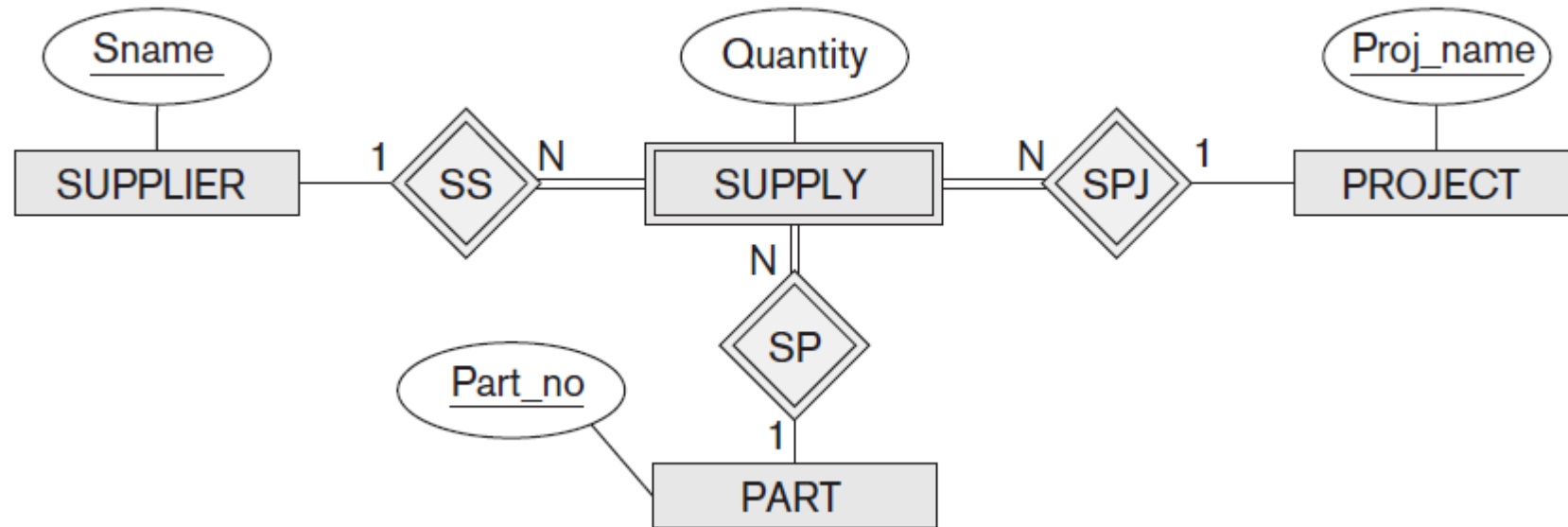
Tedarikçi

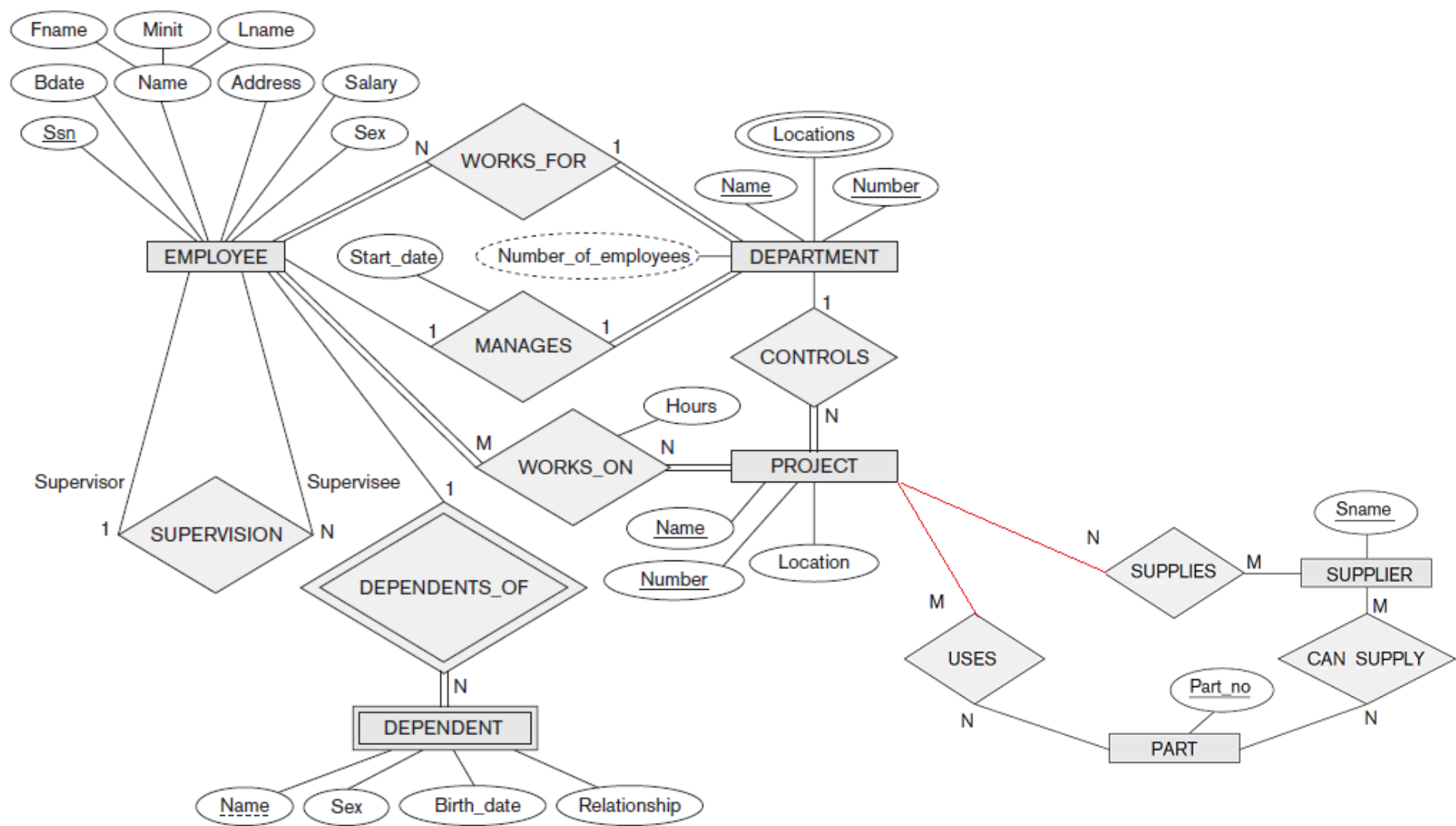


Tedarikçi



Tedarikçi

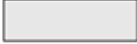








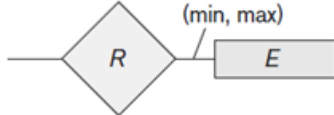





Örnek: Fakülte Otomasyonu

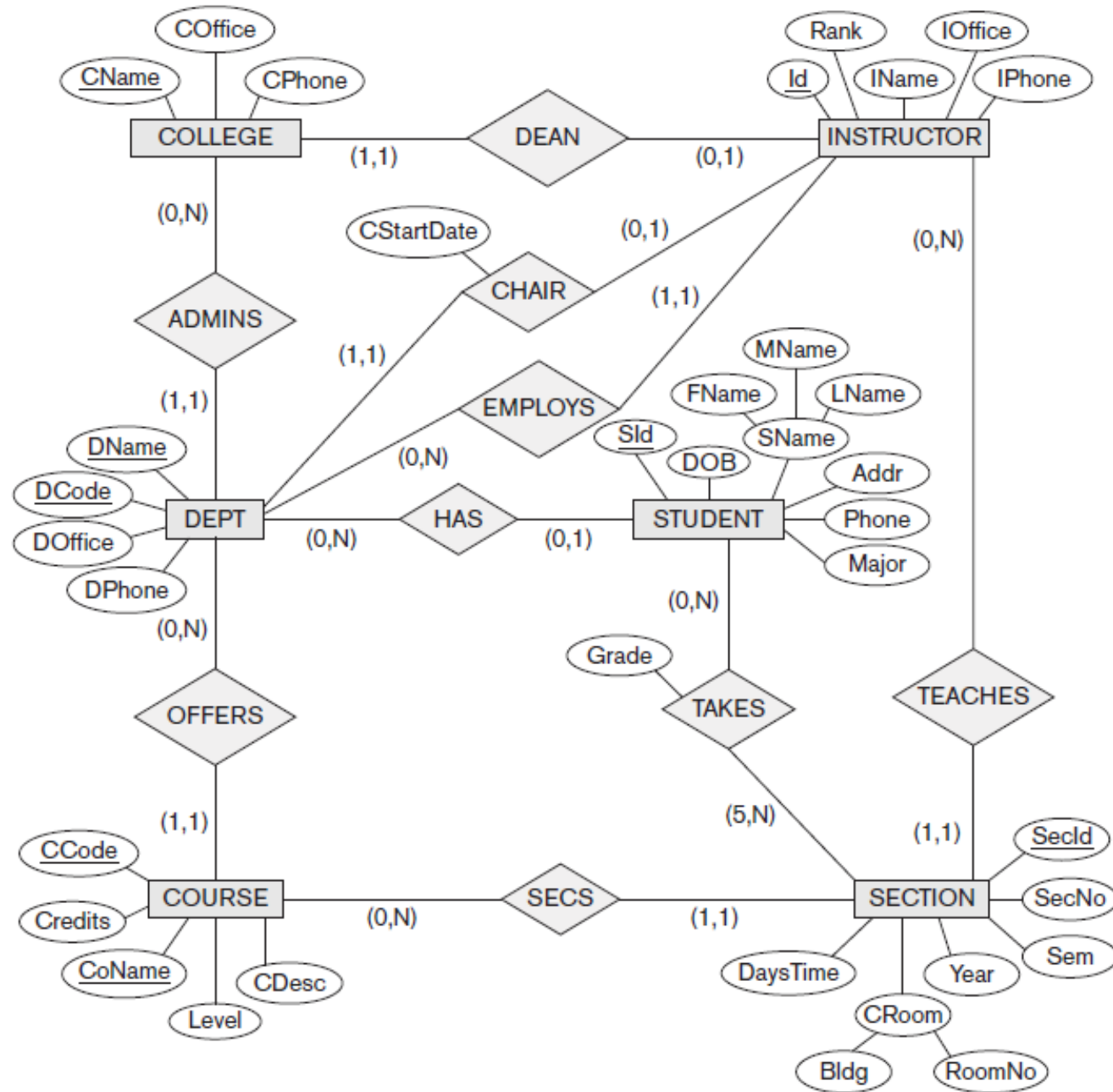


Örnek: Fakülte Otomasyonu

Symbol	Meaning	Symbol	Meaning
	Entity		Composite Attribute
	Weak Entity		Derived Attribute
	Relationship		Total Participation of E_2 in R
	Identifying Relationship		Cardinality Ratio 1: N for $E_1 : E_2$ in R
	Attribute		Structural Constraint (min, max) on Participation of E in R
	Key Attribute		



Örnek: I

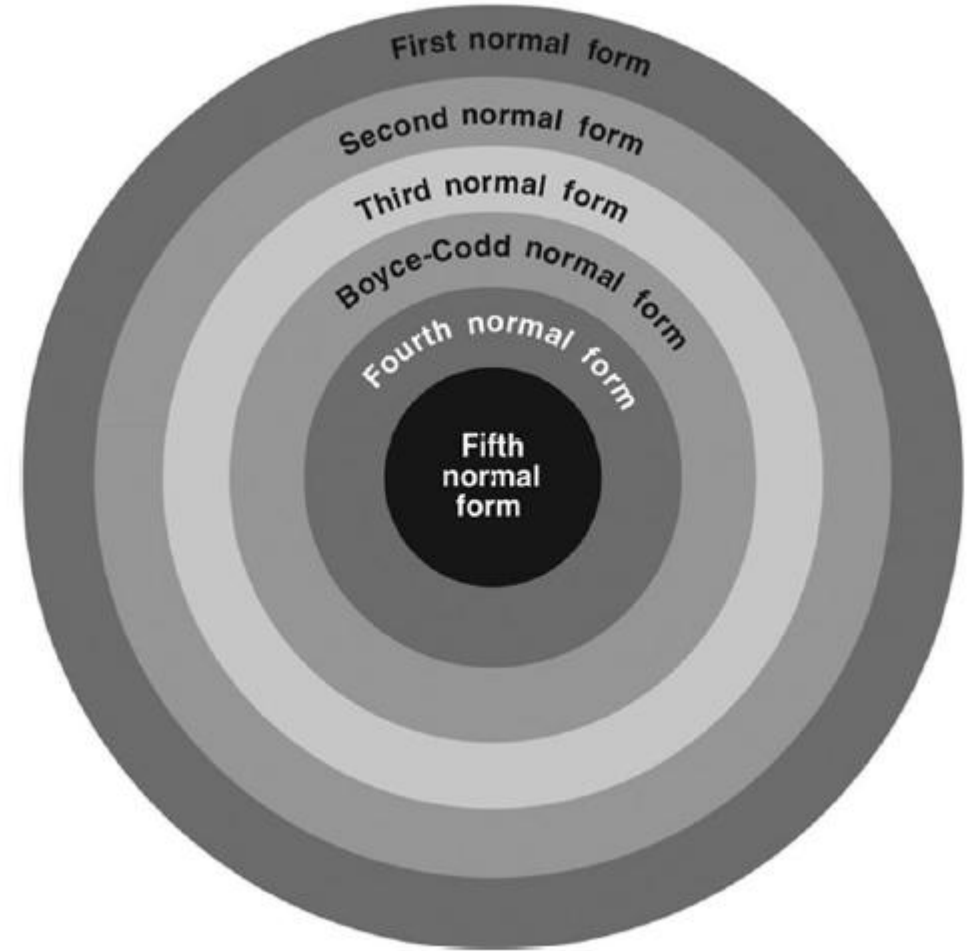


Normalizasyon

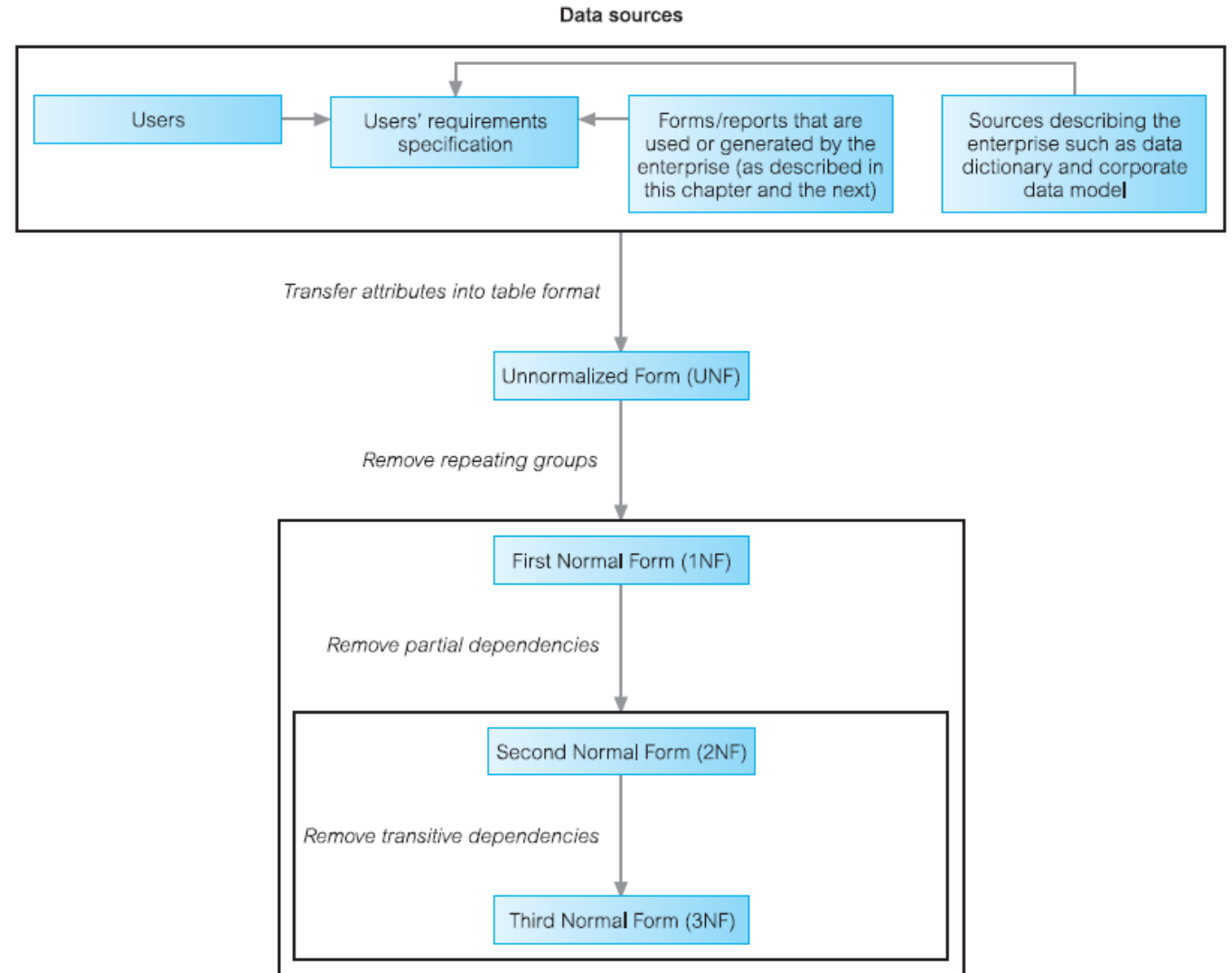
- Normalleşmenin amacı, bir işletmenin veri gereksinimlerini destekleyen uygun bir ilişki kümesi tanımlamaktır.
- ER model gibi bir veritabanı tasarlama tekniği
- Uygun bir ilişki kümesinin özellikleri aşağıdakileri içerir:
 - İşletmenin veri gereksinimlerini için gereken minimum nitelik sayısı
 - Mantıksal ilişkiye sahip nitelikler
 - İlgili ilişkiler için gerekli ara bağlantılar

Normalizasyon

- İlk üç düzey ihlal edilirse;
 - Ekleme, Silme, Güncelleme yada sorgulama kısımlarında problem
- 3NF; 1NF ve 2NF'yi kapsar.
- 2NF ise 1NF'yi kapsar.



Normalizasyon



1. Normal Form(1NF)

- Her varlığa ait özellikler ayrı hücrelerde tutulur
- Her varlık ayrı ayrı sınıflanır
- Bir firmaya ait Kira kontraları:

The diagram illustrates a 1NF form for 'DreamHome Lease' contracts. It consists of four stacked forms, each with a light blue header bar containing the text 'DreamHome Lease'. The forms are arranged in a perspective view, showing the front and slightly offset back forms. The front form is divided into four sections by a grid. The top-left section contains 'Client Number' (CR76) and 'Full Name' (John Kay). The top-right section contains 'Property Number' (PG4) and 'Property Address' (6 Lawrence St, Glasgow). The bottom-left section contains 'Monthly Rent' (350), 'Rent Start' (01/07/12), and 'Rent Finish' (31/08/13). The bottom-right section contains 'Owner Number' (C040) and 'Full Name' (Tina Murphy). Each field is followed by a horizontal line for input.

DreamHome Lease	
Client Number <u>CR76</u> (Enter if known)	Property Number <u>PG4</u>
Full Name <u>John Kay</u> (Please print)	Property Address <u>6 Lawrence St, Glasgow</u>
Monthly Rent <u>350</u>	Owner Number <u>C040</u> (Enter if known)
Rent Start <u>01/07/12</u>	Full Name <u>Tina Murphy</u> (Please print)
Rent Finish <u>31/08/13</u>	

1. Normal Form(1NF)

ClientRental

clientNo	cName	propertyNo	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	John Kay	PG4	6 Lawrence St, Glasgow	1-Jul-12	31-Aug-13	350	CO40	Tina Murphy
		PG16	5 Novar Dr, Glasgow	1-Sep-13	1-Sep-14	50	CO93	Tony Shaw
CR56	Aline Stewart	PG4	6 Lawrence St, Glasgow	1-Sep-11	10-June-12	350	CO40	Tina Murphy
		PG36	2 Manor Rd, Glasgow	10-Oct-12	1-Dec-13	375	CO93	Tony Shaw
		PG16	5 Novar Dr, Glasgow	1-Nov-14	10-Aug-15	450	CO93	Tony Shaw

1. Normal Form(1NF)

Tekrar eden gruplar:propertyNo, pAddress, rentStart, rentFinish, rent, ownerNo, oName

clientNo	propertyNo	cName	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	PG4	John Kay	6 Lawrence St, Glasgow	1-Jul-12	31-Aug-13	350	CO40	Tina Murphy
CR76	PG16	John Kay	5 Novar Dr, Glasgow	1-Sep-13	1-Sep-14	450	CO93	Tony Shaw
CR56	PG4	Aline Stewart	6 Lawrence St, Glasgow	1-Sep-11	10-Jun-12	350	CO40	Tina Murphy
CR56	PG36	Aline Stewart	2 Manor Rd, Glasgow	10-Oct-12	1-Dec-13	375	CO93	Tony Shaw
CR56	PG16	Aline Stewart	5 Novar Dr, Glasgow	1-Nov-14	10-Aug-15	450	CO93	Tony Shaw

2. Normal Form(2NF)

- 1NF'yi sağlamalı
- Tabloda bir birincil anahtar olmalı ve anahtar olmayan sütunlar birincil anahtara bağımlı olmalı
- Her tablo ayrı bir varlığın özelliğini taşımalı

Client

clientNo	cName
CR76	John Kay
CR56	Aline Stewart

Rental

clientNo	propertyNo	rentStart	rentFinish
CR76	PG4	1-Jul-12	31-Aug-13
CR76	PG16	1-Sep-13	1-Sep-14
CR56	PG4	1-Sep-11	10-Jun-12
CR56	PG36	10-Oct-12	1-Dec-13
CR56	PG16	1-Nov-14	10-Aug-15

PropertyOwner

propertyNo	pAddress	rent	ownerNo	oName
PG4	6 Lawrence St, Glasgow	350	CO40	Tina Murphy
PG16	5 Novar Dr, Glasgow	450	CO93	Tony Shaw
PG36	2 Manor Rd, Glasgow	375	CO93	Tony Shaw

3. Normal Form(3NF)

- 2NF'yi sağlamalıdır.
- Tablolar arasındaki bağlantılar kurulmalıdır.

Client

fd2 clientNo ® cName (Primary key)

Rental

fd1 clientNo, propertyNo ® rentStart, rentFinish (Primary key)

fd5' clientNo, rentStart ® propertyNo, rentFinish (Candidate key)

fd6' propertyNo, rentStart ® clientNo, rentFinish (Candidate key)

PropertyOwner

fd3 propertyNo ® pAddress, rent, ownerNo, oName (Primary key)

fd4 ownerNo ® oName (Transitive dependency)

3. Normal Form(3NF)

- 2NF'yi sağlamalıdır.
- Tablolar arasındaki bağlantılar kurulmalıdır.

PropertyForRent (propertyNo, pAddress, rent, ownerNo)

Owner (ownerNo, oName)

PropertyForRent

propertyNo	pAddress	rent	ownerNo
PG4	6 Lawrence St, Glasgow	350	CO40
PG16	5 Novar Dr, Glasgow	450	CO93
PG36	2 Manor Rd, Glasgow	375	CO93

Owner

ownerNo	oName
CO40	Tina Murphy
CO93	Tony Shaw

Örnek: Sigorta Firması

ABS isimli bir sigorta acentesi; hayat sigortası, ev sigortası ve araç sigortası olmak üzere üç farklı alan için sigorta işlemi yapmaktadır.

Yapılan her sigorta işlemine ait bir poliçe bilgisi tutulmaktadır. Bu poliçede; poliçe numarası, sigorta türü (hayat/araç/ev), sigorta türüne göre sigorta yapılan kişi/nesneye ait bilgiler, sigorta için ödenen miktar (sigorta yaptıran kişinin ödediği tutar), sigorta tutarı (olumsuz bir durumla karşılaşıldığında kişiye ödenecek tutar), sigortanın başlangıç tarihi, sigortanın süresi, poliçeyi yaptıran kişi, poliçeyi yaptıran kişiye ait adres ve telefon bilgileri tutulmaktadır. Ayrıca poliçe için ödenen tutarın nakit ya da kredi kartı ile ödendiği bilgisi de sistem üzerinde tutulmak istenmektedir.

Sigorta poliçesi bir kişiye ait ise; kişinin adı, soyadı, doğum tarihi, cinsiyeti, daha önce ameliyat olup olmadığı, kronik bir hastalığının olup olmadığı, daha önce hastanede yatıp yatmadığı gibi bilgiler tutulmalıdır. Ayrıca kişi hastaneye yatarsa masrafların ne kadar ödeneceği, şehirlerarası hasta nakli gerektiğinde nakil masraflarının ödenip ödenmeyeceği, ölüm durumunda poliçe tutarının kime ödeneceği gibi bilgiler de poliçede yer almaktadır.

Sigorta poliçesi bir eve ait ise; evin adresi, metrekare bilgisi, eşyaların sigortaya dahil olup olmadığı, yangın, su baskını, deprem gibi nedenlerden hangilerine karşı sigortalandığı bilgisi sistemde saklanmalıdır.

Sigorta poliçesi bir araca ait ise; aracın plakası, motor numarası, markası, modeli, yılı, sürücü bilgisi (bir kişi/birden fazla kişi), kaza, yangın, çalınma gibi nedenlerin hangilerine karşı sigorlandığı gibi bilgiler saklanmalıdır.

Çözüm:

1NF:

Police numarası, sigorta türü (hayat/araç/ev), sigorta için ödenen miktar (sigorta yaptıran kişinin ödediği tutar), sigorta tutarı (olumsuz bir durumla karşılaşıldığında kişiye ödenecek tutar) , sigortanın başlangıç tarihi, sigortanın süresi, poliçeyi yaptıran kişi, poliçeyi yaptıran kişiye ait adres, telefon, ödeme bilgisi(nakit/kredi kartı), kişinin adı, soyadı, doğum tarihi, cinsiyeti, ameliyat, hastalık, hastane, ölüm, evin adresi, metrekaresi bilgisi, eşyalar, yangın, su baskını, deprem, aracın plakası, motor numarası, markası, modeli, yılı, sürücü bilgisi (bir kişi/birden fazla kişi), kaza, yangın, çalınma

Çözüm:

2.NF:

Police: numarası, sigorta türü (hayat/araç/ev), sigorta için ödenen miktar (sigorta yaptıran kişinin ödediği tutar), sigorta tutarı (olumsuz bir durumla karşılaşıldığında kişiye ödenecek tutar) , sigortanın başlangıç tarihi, sigortanın süresi, poliçeyi yaptıran kişi, poliçeyi yaptıran kişiye ait adres, telefon ödeme bilgisi(nakit/kredi kartı)

Sigorta türü: id, info (hayat/araç/ev)

Müşteri: id, ad, soyad, adres, telefon

Sigortalı_Kişi: id, kişinin adı, soyadı, doğum tarihi, cinsiyeti, ameliyat, hastalık, hastane, ölüm,

Sigortalı_Ev: id, evin adresi, metrekare bilgisi, eşyalar, yangın, su baskını, deprem,

Sigortalı_Araç: aracın plakası, motor numarası, markası, modeli, yılı, sürücü bilgisi (bir kişi/birden fazla kişi), kaza, yangın, çalınma

Çözüm:

3NF:

Police: numarası, sigorta_türü (hayat/araç/ev), sigorta için ödenen miktar (sigorta yaptıran kişinin ödediği tutar), sigorta tutarı (olumsuz bir durumla karşılaşıldığında kişiye ödenecek tutar), sigortanın başlangıç tarihi, sigortanın süresi, poliçeyi yaptıran kişi, poliçeyi yaptıran kişiye ait adres, telefon ödeme bilgisi(nakit/kredi kartı)

Police_Kişi: Police_No,sigortalı_kişi_id

Police_Ev: Police_No,sigortalı_ev_id

Police_Araç: Police_No,sigortalı_arac_id

Sigorta türü: id, info (hayat/araç/ev)

Müşteri: id, ad, soyad, adres, telefon

Sigortalı_Kişi: id, kişinin adı, soyadı, doğum tarihi, cinsiyeti, ameliyat, hastalık, hastane, ölüm,

Sigortalı_Ev: id, evin adresi, metrekare bilgisi, eşyalar, yangın, su baskını, deprem,

Sigortalı_Araç: aracın plakası, motor numarası, markası, modeli, yılı, sürücü bilgisi (bir kişi/birden fazla kişi), kaza, yangın, çalınma

