SAKARYA ÜNİVERSİTESİ

SİSTEM ANALİZİ VE TASARIMI

Veri Tabanı Tasarımı

Kavramsal Veri Modelleme

- Kurumsal verilerin temsili
- Amaç, veriler arasındaki anlam ve karşılıklı ilişkilere dair kuralları göstermektir
- Varlık-İlişki (E-R) diyagramları genellikle verilerin nasıl organize edildiğini göstermek için kullanılır
- Kavramsal veri modellemenin temel amacı doğru E-R diyagramları oluşturmaktır
- Bilgi toplamak için mülakat, anket ve JAD gibi yöntemler kullanılır
- Süreç akışı, karar mantığı ve veri modelleme açıklamaları arasında tutarlılık sağlanmalıdır

Kavramsal Veri Modelleme Süreci

- İlk adım, değiştirilen sistem için bir veri modeli geliştirmektir
- Ardından, yeni sistemin tüm gereksinimlerini içeren yeni bir kavramsal veri modeli oluşturulur
- Tasarım aşamasında, kavramsal veri modeli fiziksel bir tasarıma dönüştürülür
- Proje havuzu, SDLC sırasında gerçekleştirilen tüm tasarım ve veri modelleme adımlarını birbirine bağlar

Çıktılar ve Sonuçlar

- Birincil çıktı varlık-ilişki diyagramıdır
- Kavramsal veri modelleme sırasında 4 adede kadar E-R diyagramı üretilebilir ve analiz edilebilir
  - Sadece proje uygulamasında ihtiyaç duyulan verileri kapsar
  - Değiştirilen sistem için E-R diyagramı
  - Yeni uygulamanın verilerinin çıkarıldığı tüm veritabanı için bir E-R diyagramı
  - Değiştirilmekte olan uygulama sistemine ait verilerin alındığı tüm veritabanı için bir E-R diyagramı

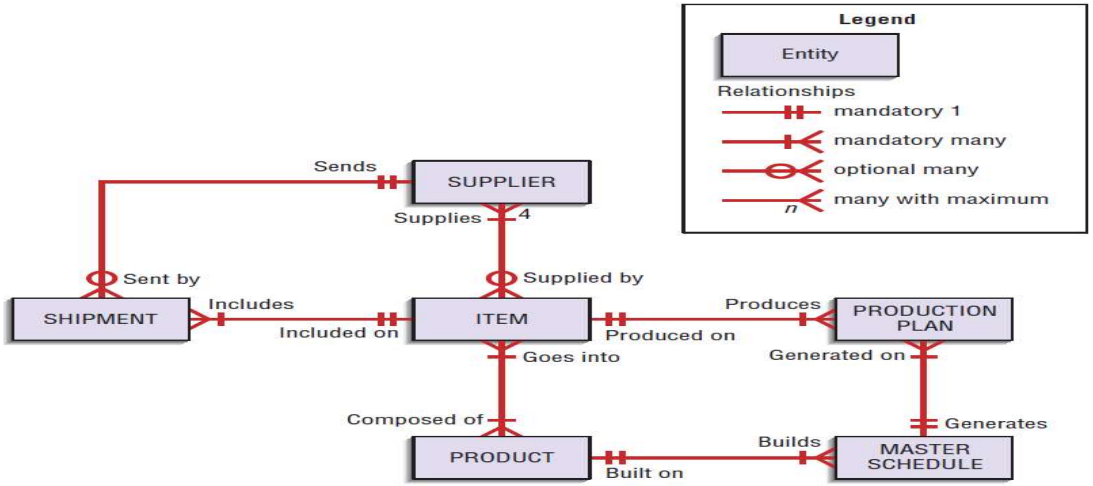


FIGURE 7-3
Sample conceptual data model diagrams (A) Standard E-R notation.

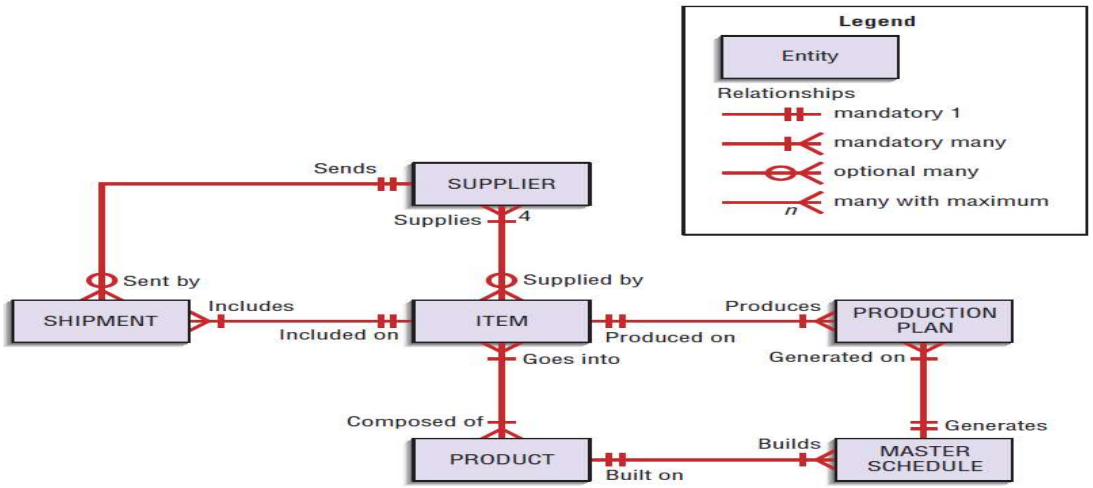
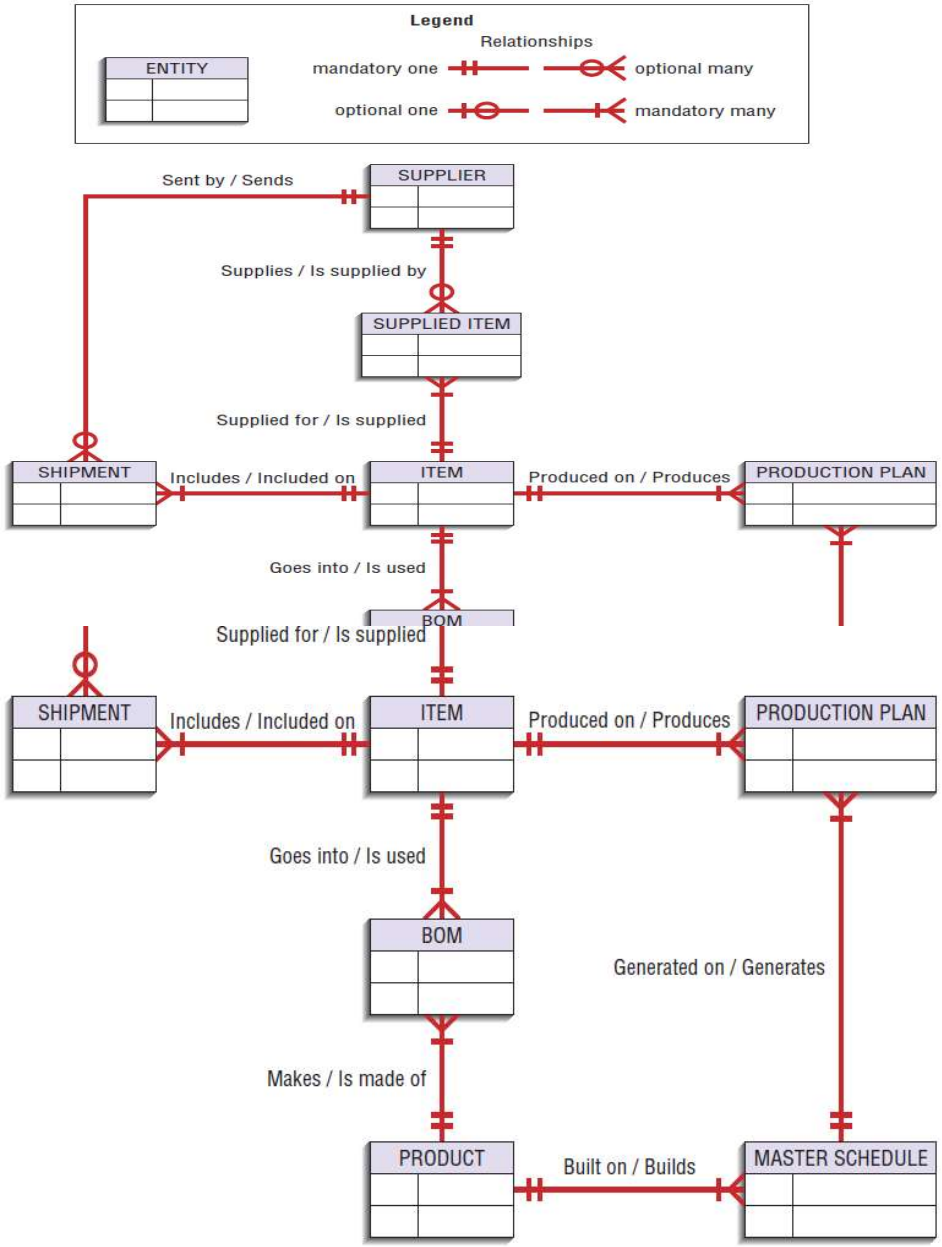


FIGURE 7-3
Sample conceptual data model diagrams (A) Standard E-R notation.

FIGURE 7-3
Sample conceptual data model diagrams (B) Visio E-R notation.



Çıktılar ve Sonuçlar (devam)

- İkinci çıktı, depoda veya proje sözlüğünde saklanacak veri nesneleri hakkında bir dizi girdidir
  - DFD'de yer alan veri unsurları veri modelinde yer almalıdır ve bunun tersi de geçerlidir
  - Bir süreç modelindeki her veri deposu, veri modelinde temsil edilen iş nesneleriyle ilişkili olmalıdır

Kavramsal Veri Modellemesi için Bilgi Toplama

- İki Perspektif:
  - Yukarıdan aşağıya
    - Veri modeli, işin yakından anlaşılmasıyla elde edilir
  - Aşağıdan yukarıya
    - Veri modeli, spesifikasyonlar ve iş belgeleri incelenerek türetilir

Varlık-İlişki Modellemesine Giriş

- Notasyon üç ana yapı kullanır
  - Veri varlıkları
  - İlişkiler
  - Nitelikler
- Varlık-İlişki (E-R) Diyagramı
  - Bir kuruluş veya işletme için varlıkların, ilişkilerin ve veri öğelerinin ayrıntılı, mantıksal ve grafiksel gösterimi

Varlık-İlişki (E-R) Modellemesi
Anahtar Terimler

- Varlık
  - Kuruluşun hakkında veri tutmak istediği kullanıcı ortamındaki bir kişi, yer, nesne, olay veya kavram
  - E-R diyagramlarında bir dikdörtgen ile temsil edilir
- Varlık Türü
  - Ortak özellikleri veya karakteristikleri paylaşan varlıklar topluluğu
- Varlık Örneği
  - Bir varlık türünün tek oluşumu

Varlık-İlişki (E-R) Modellemesi (devam)
Anahtar Terimler

- Öznitelik
  - Bir kuruluşun ilgisini çeken bir varlığın adlandırılmış bir özelliği veya karakteristiği
- Aday Anahtarlar ve Tanımlayıcılar
  - Her varlık türü, bir örneği aynı türün diğer örneklerinden ayıran bir özniteliğe veya özniteliklerin kümesine sahip olmalıdır
  - Aday anahtarı
    - Bir varlık türünün her bir örneğini benzersiz bir şekilde tanımlayan öznitelik (veya öznitelik kombinasyonu)

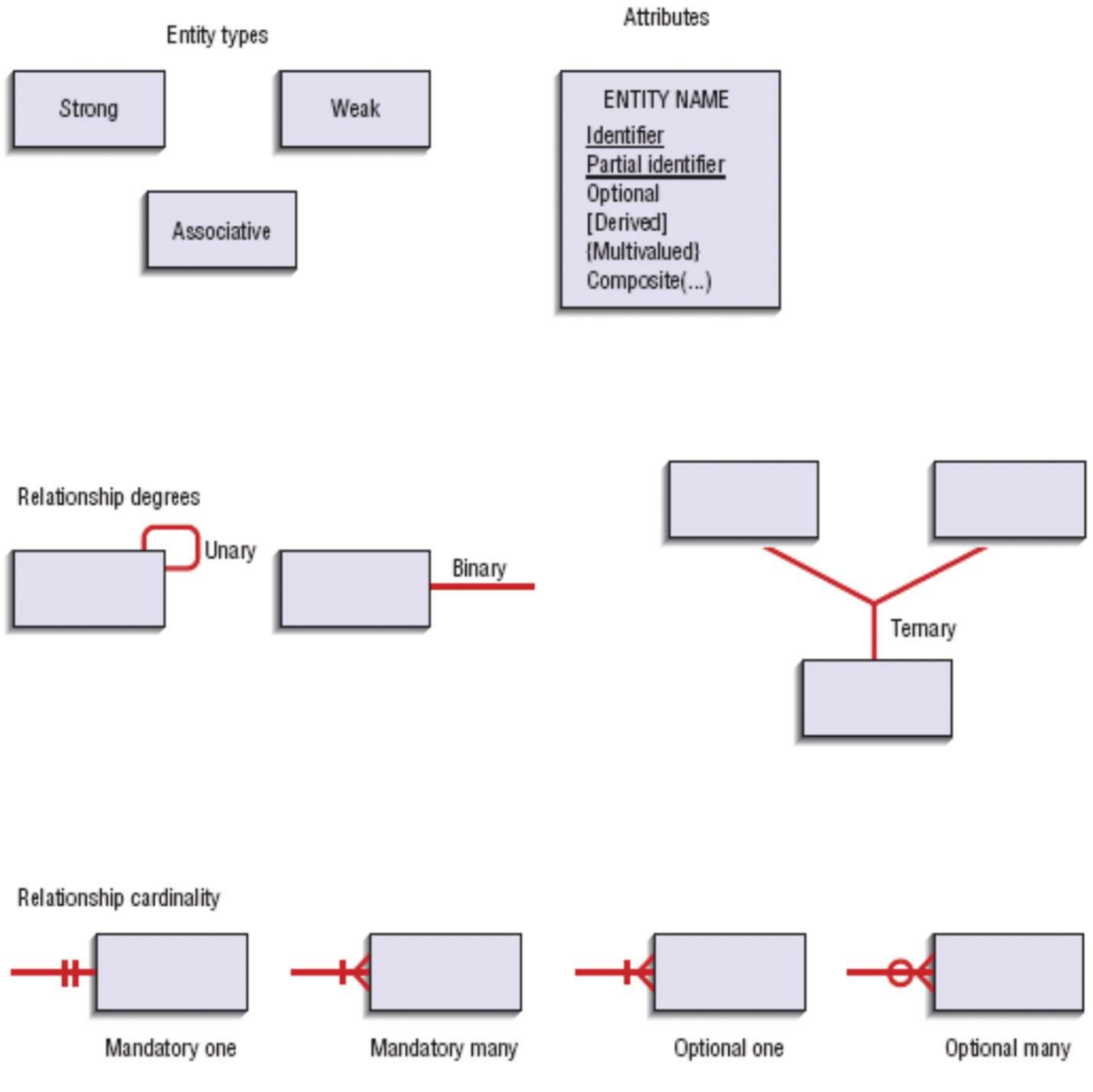


FIGURE 7-5
Entity-Relationship Diagram Notations: Basic Symbols, Relationship Degree, and Relationship Cardinality

Varlık-İlişki (E-R) Modellemesi (devam)

Anahtar Terimler

- Tanımlayıcı
  - Bir varlık türü için benzersiz tanımlayıcı özellik olarak seçilen bir aday anahtar
  - Bir tanımlayıcı için seçim kuralları
    1. Değeri değişmeyecek bir aday anahtar seçin
    2. Asla null olmayacak bir aday anahtar seçin
    3. Akıllı tuşları kullanmaktan kaçının
    4. Büyük bileşik anahtarlar yerine tek değerli vekil anahtarlar kullanmayı düşünün

Varlık-İlişki (E-R) Modellemesi (devam)
Anahtar Terimler

- Çok Değerli Öznitelik
  - Her varlık örneği için birden fazla değer alabilen bir öznitelik
  - E-R diyagramında iki şekilde temsil edilir:
    - çift çizgili elips
    - zayıf varlık

Varlık-İlişki (E-R) Modellemesi (devam)

Anahtar Terimler

- İlişki
  - Kuruluşun ilgi alanına giren bir veya daha fazla varlık türünün örnekleri arasındaki bir ilişki
  - İlişkilendirme, bir olayın meydana geldiğini veya varlık türleri arasında doğal bir bağ olduğunu gösterir
  - İlişkiler her zaman fiil cümleleri ile etiketlenir

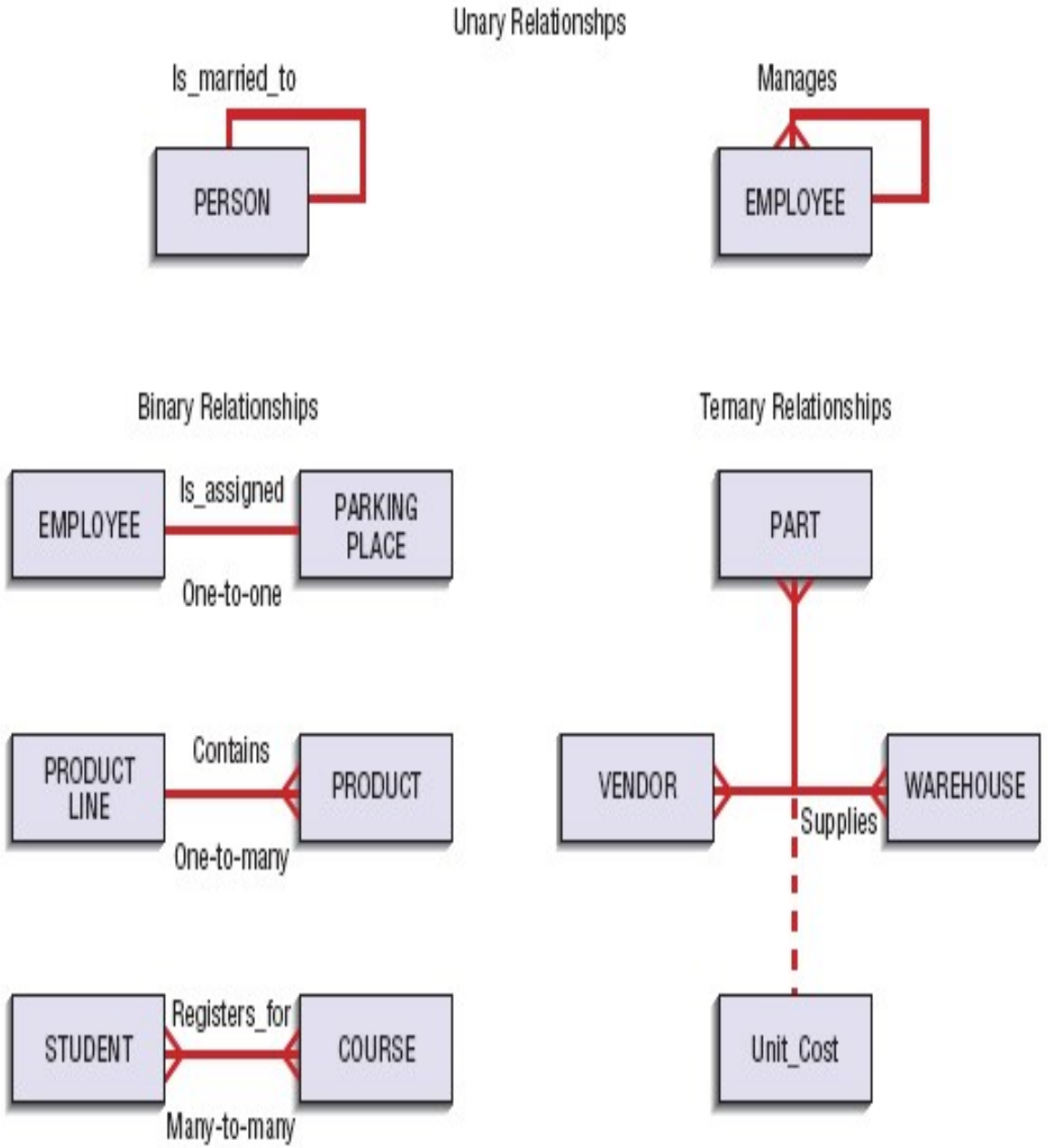
Kavramsal Veri Modelleme ve E-R Diyagramı

- Hedef
  - Verilerin anlamının mümkün olduğunca çoğunu yakalayın
- Sonuç
  - Bakımı daha kolay olan daha iyi bir tasarım

İlişki Derecesi

- Derece
  - Bir ilişkiye katılan varlık türü sayısı
- Üç Vaka:
  - Tekli
    - Bir varlık türünün örnekleri arasındaki ilişki
  - İkili
    - İki varlık türünün örnekleri arasındaki bir ilişki
  - Üçlü
    - Üç varlık türünün örnekleri arasında eşzamanlı bir ilişki
    - Üç ikili ilişki ile aynı değildir

FIGURE 7-6
Examples of the Three Most Common Relationships in E-R Diagrams: Unary, Binary, and Ternary



Kardinalite

- A varlığının her bir örneği ile ilişkilendirilebilecek B varlığı örneklerinin sayısı
- Minimum Kardinalite
  - A varlığının her bir örneğiyle ilişkilendirilebilecek minimum B varlığı örneği sayısı
- Maksimum Kardinalite
  - A varlığının her bir örneğiyle ilişkilendirilebilecek maksimum B varlığı örneği sayısı

İlişkisel Varlık

- Bir veya daha fazla varlık türünün örneklerini ilişkilendiren ve bu varlık örnekleri arasındaki ilişkiye özgü nitelikler içeren bir varlık türü

FIGURE 7-7
Example of an Associative Entity

PVF Web Mağazası: Kavramsal Veri Modelleme

- İnternet uygulamaları için kavramsal veri modellemesi, diğer uygulama türleri için izlenen süreçten farklı değildir
- Pine Valley Mobilya Web Mağazası
  - Tanımlanmış dört varlık türü
    - Müşteri
    - Envanter
    - Sipariş
    - Alışveriş sepeti

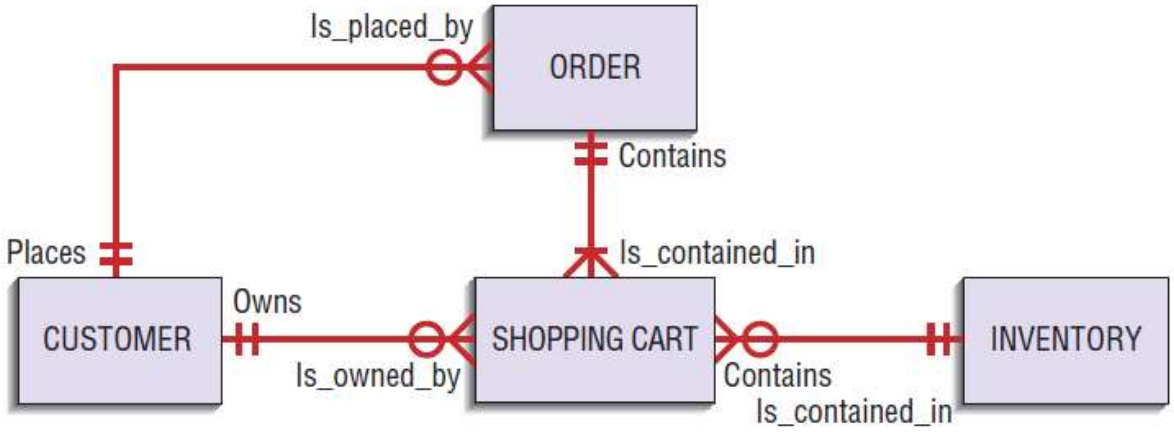


FIGURE 7-13
Entity-relationship diagram for the WebStore system.

En İyi Alternatif Tasarım Stratejisinin Seçilmesi

- İki temel adım:
  1. Kapsamlı bir alternatif tasarım stratejileri seti oluşturun
  2. İstenen bilgi sistemi ile sonuçlanma olasılığı en yüksek olan tasarım stratejisini seçin
- Süreç:
  1. Gereksinimleri farklı yetenek kümelerine ayırın
  2. Farklı yetenek setlerini sunmak için kullanılabilecek farklı potansiyel uygulama ortamlarını sıralayın
  3. Farklı uygulama ortamları için çeşitli kabiliyet setlerini tedarik etmek veya edinmek için farklı yollar önermek

En İyi Alternatif Tasarım Stratejisinin Seçilmesi (devam)

- Teslim Edilecekler
  1. Yedek bilgi sisteminin oluşturulması için en az üç önemli ölçüde farklı sistem tasarım stratejisi
  2. En çok arzu edilen bilgi sistemine yol açma olasılığı en yüksek olan tasarım stratejisi

Alternatif Tasarım Stratejileri Oluşturma

- En iyisi üç alternatif oluşturmaktır:
  - Düşük Uç
    - Mevcut sistemden minimum düzeyde farklı bir sistemle kullanıcıların talep ettiği tüm gerekli işlevleri sağlar
  - Üst Düzey
    - Söz konusu sorunu çözer ve kullanıcıların istediği birçok ekstra özelliği sağlar
  - Orta seviye
    - Üst düzey alternatifin özellikleri ile alt düzey alternatifin tutumluluğunun uzlaşması

Alternatif Tasarımlara Sınır Çizmek

- Minimum Gereksinimler
  - Zorunlu özelliklere karşı istenen özellikler
  - Özelliklerin biçimleri
    - Veri
    - Çıktılar
    - Analizler
    - Erişilebilirlik, yanıt süresi ve geri dönüş süresine ilişkin kullanıcı beklentileri

Alternatif Tasarımlara Sınır Çizmek (devam)

- Sistem Geliştirme Üzerindeki Kısıtlamalar:
  - Zaman
  - Finansal
  - Mevcut sistemin değiştirilemeyecek unsurları
  - Yasal
  - Sorunun dinamikleri

Hoosier Burger'in Yeni Envanter Kontrol Sistemi

- Mevcut sistemin değiştirilmesi
- Şekil 7-15 sistem gereksinimlerini ve kısıtlamalarını sıralar

FIGURE 7-15
Ranked System Requirements and Constraints for Hoosier Burger's Inventory System

SYSTEM REQUIREMENTS (in descending priority)	SYSTEM CONSTRAINTS (in descending order)
1. Must be able to easily enter shipments into system as soon as they are received.	1. System development can cost no more than $50,000.
2. System must automatically determine whether and when a new order should be placed.	2. New hardware can cost no more than $50,000.
3. Management should be able to determine at any time approximately what inventory levels are for any given item in stock.	3. The new system must be operational in no more than six months from the start of the contract.
	4. Training needs must be minimal (i.e., the new system must be easy to use).

Hoosier Burger'in Yeni Envanter Kontrol Sistemi (devam)

- Şekil 7-16 mevcut sistemin adımlarını göstermektedir
- Alternatifler önerilirken, gereklilikler ve kısıtlamalar dikkate alınmalıdır

FIGURE 7-16
The Steps in Hoosier Burger's Inventory Control System

1. Meet delivery trucks before opening restaurant.
2. Unload and store deliveries.
3. Log invoices and file in accordion file.
4. Manually add amounts received to stock logs.
5. After closing, print inventory report.
6. Count physical inventory amounts.
7. Compare inventory reports totals to physical count totals.
8. Compare physical count totals to minimum order quantities; if the amount is less, make order; if not, do nothing.
9. Pay bills that are due and record them as paid.

Hoosier Burger'in Yeni Envanter Kontrol Sistemi (devam)

CRITERIA	ALTERNATIVE A	ALTERNATIVE B	ALTERNATIVE C
Requirements			
1. Easy real-time entry of new shipment data	Yes	Yes	Yes
2. Automatic reorder decisions	For some items	For all items	For all items
3. Real-time data on inventory levels	Not available	Available for some items only	Fully available
Constraints			
1. Cost to develop	$25,000	$50,000	$65,000
2. Cost of hardware	$25,000	$50,000	$50,000
3. Time to operation	Three months	Six months	Nine months
4. Ease of training	One week of training	Two weeks of training	One week of training

FIGURE 7-18
Description of Three Alternative Systems That Could Be Developed for Hoosier Burger's Inventory System

- Şekil 7-18 3 alternatifi listelemektedir:
  - Alternatif A düşük kaliteli bir tekliftir
  - Alternatif C üst düzey bir tekliftir
  - Alternatif B orta menzilli bir tekliftir

Hoosier Burger'in Yeni Envanter Kontrol Sistemi (devam)

- En Olası Alternatifin Seçilmesi
  - Üç alternatifi karşılaştırmak için ağırlıklı yaklaşım kullanılabilir
  - Şekil 7-19 Hoosier Burger için ağırlıklı bir yaklaşımı göstermektedir
  - Tablonun sol tarafı karar kriterlerini içerir
    - Sabitler ve gereksinimler
    - Ağırlıklar analiz ekibi, kullanıcılar ve yöneticilerle tartışılarak belirlenir
  - Her gereksinim ve kısıtlama sıralanır
    - 1, alternatifin taleple iyi eşleşmediğini veya kısıtlamayı ihlal ettiğini gösterir
    - 5 alternatifin gereklilikleri karşıladığını veya aştığını ya da kısıtlamaya açıkça uyduğunu gösterir

Criteria	Weight	Alternative A		Alternative B		Alternative C	
		Rating	Score	Rating	Score	Rating	Score
Requirements							
Real-time data entry	18	5	90	5	90	5	90
Auto reorder	18	3	54	5	90	5	90
Real-time data query	14	1	14	3	42	5	70
	50		158		222		250
Constraints							
Development costs	20	5	100	4	80	3	60
Hardware costs	15	5	75	4	60	4	60
Time to operation	10	5	50	4	40	3	30
Ease of training	5	5	25	3	15	5	25
	50		250		195		175
Total	100		408		417		425

FIGURE 7-19
Weighted approach for comparing the three alternative systems for Hoosier Burger's inventory system.

Hoosier Burger'in Yeni Envanter Kontrol Sistemi (devam)

- En Olası Alternatifin Seçilmesi
  - Kullanılan ağırlıklara göre, C alternatifi en iyi seçenek olarak görünmektedir

Veritabanı Tasarım Süreci

- Mantıksal Tasarım
  - Kavramsal veri modeline dayalı olarak
  - Dört temel adım:
    1. Normalleştirme ilkelerini kullanarak uygulama için bilinen her kullanıcı arayüzü için bir mantıksal veri modeli geliştirin
    2. Tüm kullanıcı arayüzlerinden gelen normalleştirilmiş veri gereksinimlerinin tek bir konsolide mantıksal veritabanı modelinde birleştirilmesi
    3. Uygulama için kavramsal E-R veri modelinin normalleştirilmiş veri gereksinimlerine dönüştürülmesi
    4. Birleştirilmiş mantıksal veritabanı tasarımını çevrilmiş E-R modeli ile karşılaştırın ve uygulama için nihai bir mantıksal veritabanı modeli üretin

Veritabanı Tasarım Süreci (devam)

- Fiziksel Tasarım
  - Mantıksal veritabanı tasarımının sonuçlarına göre
  - Önemli kararlar:
    1. Mantıksal veritabanı modelinden her bir öznitelik için depolama formatının seçilmesi
    2. Mantıksal veritabanı modelindeki öznitelikleri fiziksel kayıtlar halinde gruplama
    3. Kayıtların hızlı bir şekilde saklanabilmesi, geri getirilebilmesi ve güncellenebilmesi için ilgili kayıtların ikincil bellekte (sabit diskler ve manyetik bantlar) düzenlenmesi
    4. Erişimi daha verimli hale getirmek için veri depolamaya yönelik ortam ve yapıların seçilmesi

Veritabanı Tasarım Süreci (devam)

- Birincil Anahtar
  - Değeri bir ilişkinin tüm oluşumlarında benzersiz olan bir öznitelik

Çıktılar ve Sonuçlar

- Mantıksal veritabanı tasarımı her veri öğesini, sistem girdisini veya çıktısını hesaba katmalıdır
- Normalleştirilmiş ilişkiler birincil çıktıdır
- Fiziksel veritabanı tasarımı, ilişkilerin dosyalara dönüştürülmesiyle sonuçlanır

İlişkisel Veritabanı Modeli

- İlgili tablolar veya ilişkiler kümesi olarak temsil edilen veriler
- İlişki
  - Adlandırılmış, iki boyutlu bir veri tablosu. Her ilişki bir dizi adlandırılmış sütundan ve rastgele sayıda adlandırılmamış satırdan oluşur
  - Mülkler
    - Hücrelerdeki girişler basittir
    - Sütunlardaki girişler aynı değerler kümesindendir
    - Her satır benzersizdir
    - Sütunların sırası, ilişkinin anlamını veya kullanımını değiştirmeden değiştirilebilir
    - Satırlar herhangi bir sırada değiştirilebilir veya saklanabilir

İlişkisel Veritabanı Modeli (devam)

- İyi Yapılandırılmış İlişki
  - Minimum miktarda fazlalık içeren ve kullanıcıların satırları hata veya tutarsızlık olmadan eklemesine, değiştirmesine ve silmesine olanak tanıyan bir ilişki

FIGURE 9-5
EMPLOYEE1 relation with sample data.

EMPLOYEE1

Emp_ID	Name	Dept	Salary
100	Margaret Simpson	Marketing	42,000
140	Allen Beeton	Accounting	39,000
110	Chris Lucero	Info Systems	41,500
190	Lorenzo Davis	Finance	38,000
150	Susan Martin	Marketing	38,500

Normalleştirme

- Karmaşık veri yapılarını basit, kararlı veri yapılarına dönüştürme süreci
- Fazlalığı ortadan kaldırır (bkz. Şekil 9-6)

EMPLOYEE2

Emp_ID	Name	Dept	Salary	Course	Date_Completed
100	Margaret Simpson	Marketing	42,000	SPSS	6/19/2015
100	Margaret Simpson	Marketing	42,000	Surveys	10/7/2015
140	Alan Beeton	Accounting	39,000	Tax Acc	12/8/2015
110	Chris Lucero	Info Systems	41,500	SPSS	1/12/2015
110	Chris Lucero	Info Systems	41,500	C++	4/22/2015
190	Lorenzo Davis	Finance	38,000	Investments	5/7/2015
150	Susan Martin	Marketing	38,500	SPSS	6/19/2015
150	Susan Martin	Marketing	38,500	TQM	8/12/2015

FIGURE 9-6
Relation with redundancy.

Normalleştirme (devam)

- İkinci Normal Form (2NF)
  - Her bir birincil olmayan anahtar özniteliği tüm anahtar tarafından tanımlanır (*tam işlevsel bağımlılık olarak* adlandırılır)
- Üçüncü Normal Form (3NF)
  - Birincil olmayan anahtar nitelikler birbirlerine bağlı değildir (*geçişli bağımlılıklar olarak* adlandırılır)
- Normalleştirmenin sonucu, birincil anahtar olmayan her özniteliğin tüm birincil anahtara bağlı olmasıdır.

İşlevsel Bağımlılıklar ve Birincil Anahtarlar

- Fonksiyonel Bağımlılık
  - İki nitelik arasındaki belirli bir ilişki. Belirli bir ilişki için, A'nın her geçerli değeri için A'nın değeri B'nin değerini benzersiz bir şekilde belirliyorsa, B niteliği A niteliğine işlevsel olarak bağımlıdır.
  - Bir ilişkideki örnekler (veya örnek veriler) işlevsel bir bağımlılığın varlığını kanıtlamaz
  - Sorun alanı bilgisi, işlevsel bağımlılığı tanımlamak için en güvenilir yöntemdir

İşlevsel Bağımlılıklar ve Birincil Anahtarlar (devam)

- İkinci Normal Form (2NF)
  - Bir ilişki, aşağıdaki koşullardan herhangi biri geçerli ise ikinci normal formdadır (2NF):
    1. Birincil anahtar yalnızca bir öznitelikten oluşur
    2. İlişkide birincil olmayan anahtar öznitelikleri yoktur
    3. Birincil olmayan her bir anahtar öznitelik işlevsel olarak birincil anahtar özniteliklerin tamamına bağlıdır

Fonksiyonel Bağımlılıklar ve Birincil Anahtarlar (devam)

- İkinci normal forma (2NF) dönüştürme
  - Bir ilişkiyi 2NF'ye dönüştürmek için, diğer nitelikleri belirleyen ve belirleyici olarak adlandırılan nitelikleri kullanarak ilişkiyi yeni ilişkilere ayrıştırın
  - Belirleyiciler yeni ilişkinin birincil anahtarı haline gelir

Fonksiyonel Bağımlılıklar ve Birincil Anahtarlar (devam)

- Üçüncü Normal Form (3NF)
  - Bir ilişki ikinci normal formdaysa (2NF) ve iki (veya daha fazla) birincil olmayan anahtar nitelik arasında işlevsel (geçişli) bağımlılık yoksa üçüncü normal formdadır (3NF)
  - Bir ilişkiyi 2NF'ye dönüştürmek için, iki belirleyiciyi kullanarak ilişkiyi iki ilişkiye ayırın

FIGURE 9-9
Removing Transitive Dependencies
(A) Relation with Transitive Dependency
(B) Relations in 3NF

A

SALES

Customer_ID	Customer_Name	Salesperson	Region
8023	Anderson	Smith	South
9167	Bancroft	Hicks	West
7924	Hobbs	Smith	South
6837	Tucker	Hernandez	East
8596	Eckersley	Hicks	West
7018	Arnold	Faulb	North

B

SALES1

Customer_ID	Customer_Name	Salesperson
8023	Anderson	Smith
9167	Bancroft	Hicks
7924	Hobbs	Smith
6837	Tucker	Hernandez
8596	Eckersley	Hicks
7018	Arnold	Faulb

SPERSON

Salesperson	Region
Smith	South
Hicks	West
Hernandez	East
Faulb	North

Fonksiyonel Bağımlılıklar ve Birincil Anahtarlar (devam)

- Yabancı Anahtar
  - Bir ilişkide birincil olmayan anahtar özniteliği olarak ve başka bir ilişkide birincil anahtar özniteliği (veya birincil anahtarın bir parçası) olarak görünen bir öznitelik
- Referans Bütünlüğü
  - Bir ilişkideki bir özniteliğin değerinin (veya varlığının) başka bir ilişkideki aynı özniteliğin değerine (veya varlığına) bağlı olduğunu belirten bir bütünlük kısıtı

E-R Diyagramlarını İlişkilere Dönüştürme

- Kavramsal veri modelini bir dizi normalleştirilmiş ilişkiye dönüştürmek yararlıdır
- Adımlar:
  1. Kuruluşları temsil edin
  2. İlişkileri temsil edin
  3. İlişkileri normalleştirin
  4. İlişkileri birleştirin

E-R Diyagramlarını İlişkilere Dönüştürme (devam)

1. Tüzel Kişileri Temsil Edin
   - Her düzenli varlık bir ilişkiye dönüştürülür
   - Varlık türünün tanımlayıcısı, ilgili ilişkinin birincil anahtarı olur
   - Birincil anahtar aşağıdaki iki koşulu karşılamalıdır
     a. Anahtarın değeri ilişkideki her satırı benzersiz bir şekilde tanımlamalıdır
     b. Anahtar yedekli olmamalıdır

A

FIGURE 9-10
Transforming an entity type to a relation
(A) E-R diagram
(B) Relation.

CUSTOMER

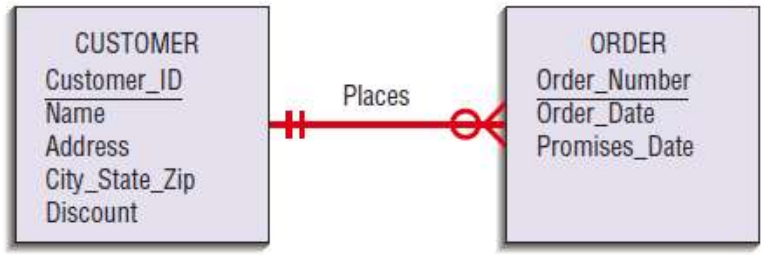
Customer_ID	Name	Address	City_State_Zip	Discount
1273	Contemporary Designs	123 Oak St.	Austin, TX 28384	5%
6390	Casual Corner	18 Hoosier Dr.	Bloomington, IN 45821	3%

B

E-R Diyagramlarını İlişkilere Dönüştürme (devam)

2. İlişkileri Temsil Edin
   - İkili 1:N İlişkiler
     - İlişkinin bir tarafındaki varlığın birincil anahtar niteliğini (veya niteliklerini) sağ taraftaki ilişkiye yabancı anahtar olarak ekleyin
     - Tek taraf çok tarafa *göç eder*

FIGURE 9-11
Representing a 1:*N* relationship.
(A) E-R diagram, (B) relations.



A

CUSTOMER

Customer_ID	Name	Address	City_State_ZIP	Discount
1273	Contemporary Designs	123 Oak St.	Austin, TX 28384	5%
6390	Casual Corner	18 Hoosier Dr.	Bloomington, IN 45821	3%

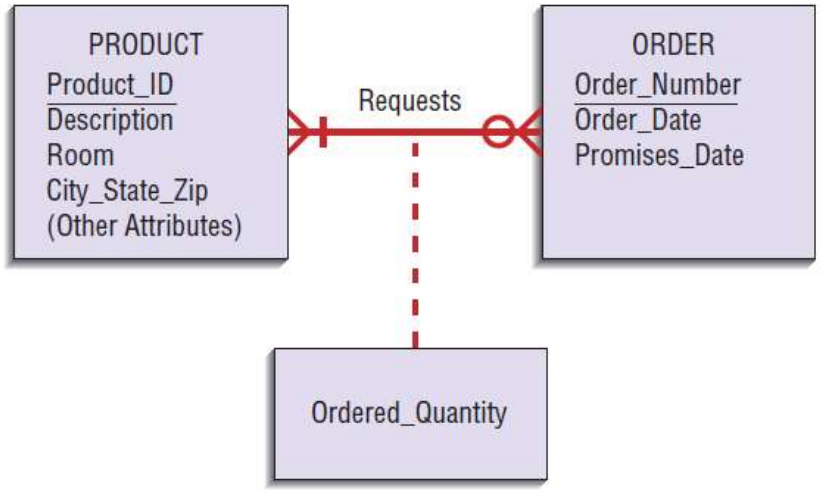
ORDER

Order_Number	Order_Date	Promised_Date	Customer_ID
57194	3/15/15	3/28/15	6390
63725	3/17/15	4/01/15	1273
80149	3/14/15	3/24/15	6390

B

E-R Diyagramlarını İlişkilere Dönüştürme (devam)

2. Temsil İlişkileri (devam)
   - İkili veya Tekli 1:1
     - Üç olası seçenek:
       a. A'nın birincil anahtarını B'nin yabancı anahtarı olarak ekleme
       b. B'nin birincil anahtarını A'nın yabancı anahtarı olarak ekleme
       c. Her ikisi de
   - İkili ve daha yüksek *M:N* ilişkileri
     - Başka bir ilişki oluşturun ve tüm ilişkilerin birincil anahtarlarını yeni ilişkinin birincil anahtarı olarak ekleyin



A

B

ORDER

Order_Number	Order_Date	Promised_Date
61384	2/17/2012	3/01/2012
62009	2/13/2012	2/27/2012
62807	2/15/2012	3/01/2012

ORDER LINE

Order_Number	Product_ID	Quantity_ Ordered
61384	M128	2
61384	A261	1

PRODUCT

Product_ID	Description	(Other Attributes)
M128	Bookcase	—
A261	Wall unit	—
R149	Cabinet	—

FIGURE 9-12
Representing an *M:N* relationship (A) E-R diagram (B) Relations.

E-R Diyagramlarını İlişkilere Dönüştürme (devam)

- Tekil *1:N* İlişkiler
  - Tek bir varlık türünün örnekleri arasındaki ilişki
  - Özyinelemeli yabancı anahtar kullanma
    - Bir ilişkide, aynı ilişkinin birincil anahtar değerlerine referans veren bir yabancı anahtar
- Tek *M:N* İlişkileri
  - Ayrı bir ilişki oluşturun
  - Yeni ilişkinin birincil anahtarı, her ikisi de değerlerini aynı birincil anahtardan alan iki özniteliğin bileşimidir

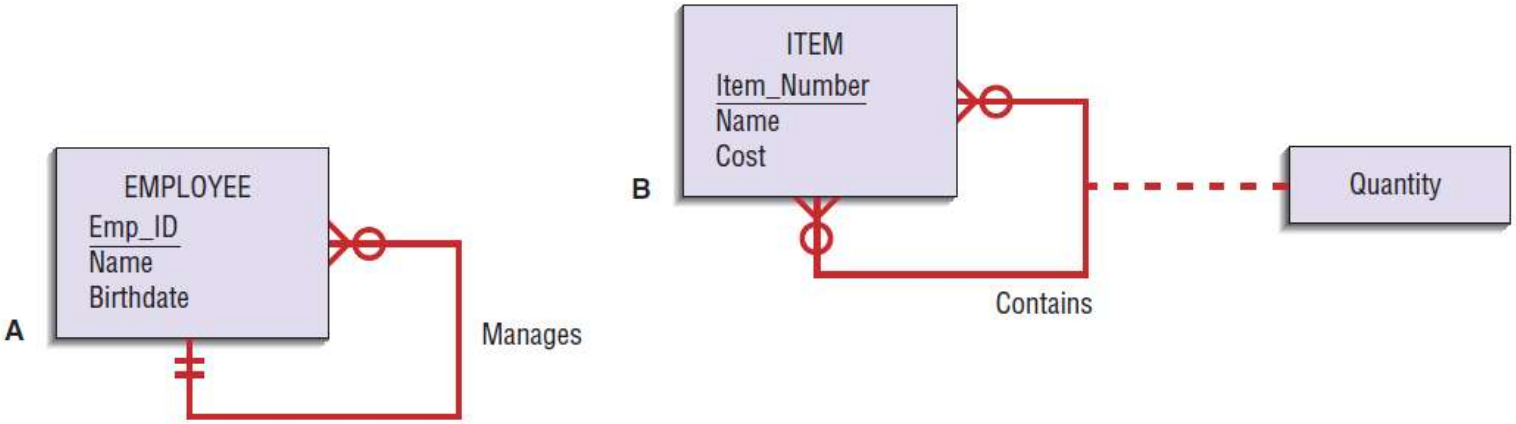


FIGURE 9-13
Two unary relations (A) EMPLOYEE with manages relationship (1:*N*) (B) Bill-of-materials structure (*M*:*N*).

Table 9.1 E-R to Relational Transformation

E-R Structure	Relational Representation
Regular entity	Create a relation with primary key and nonkey attributes.
Weak entity	Create a relation with a composite primary key (which includes the primary key of the entity on which this weak entity depends) and nonkey attributes.
Binary or unary 1:1 relationship	Place the primary key of either entity in the relation for the other entity or do this for both entities.
Binary 1:*N* relationship	Place the primary key of the entity on the one side of the relationship as a foreign key in the relation for the entity on the many side.
Binary or unary *M*:*N* relationship or associative entity	Create a relation with a composite primary key using the primary keys of the related entities, plus any non-key attributes of the relationship or associative entity.
Binary or unary *M*:*N* relationship or associative entity with additional key(s)	Create a relation with a composite primary key using the primary keys of the related entities and additional primary key attributes associated with the relationship or associative entity, plus any nonkey attributes of the relationship or associative entity.
Binary or unary *M*:*N* relationship or associative entity with its own key	Create a relation with the primary key associated with the relationship or associative entity, plus any nonkey attributes of the relationship or associative entity and the primary keys of the related entities (as nonkey attributes).

E-R Diyagramlarını İlişkilere Dönüştürme (devam)

3. İlişkileri Birleştirme (Entegrasyonu Görüntüle)
   - Amaç gereksiz ilişkileri ortadan kaldırmaktır
   - Entegrasyon Sorunlarını Görüntüle
     - Eşanlamlılar
       - Aynı nitelik için kullanılan iki farklı isim
       - Birleştirirken, kullanıcıların tek ve standart bir isim üzerinde anlaşmasını sağlayın
     - Eşsesli Kelimeler
       - İki veya daha fazla farklı öznitelik için kullanılan tek bir öznitelik adı
       - Yeni bir ad oluşturularak çözüldü
     - Anahtar olmayanlar arasındaki bağımlılıklar
       - Görünüm entegrasyonunun bir sonucu olarak bağımlılıklar oluşturulabilir
       - Çözümlemek için yeni ilişkinin normalleştirilmesi gerekir

Fiziksel Dosya ve Veritabanı Tasarımı

- Aşağıdaki bilgiler gereklidir
  - Hacim tahminleri de dahil olmak üzere normalleştirilmiş ilişkiler
  - Her bir niteliğin tanımları
  - Verilerin nerede ve ne zaman kullanıldığına, girildiğine, alındığına, silindiğine ve güncellendiğine ilişkin açıklamalar (sıklıklar dahil)
  - Yanıt süresi ve veri bütünlüğüne ilişkin beklentiler veya gereksinimler
  - Dosyaları ve veritabanını uygulamak için kullanılan teknolojilerin açıklamaları

Tasarım Alanları

- Saha
  - Sistem yazılımı tarafından tanınan en küçük adlandırılmış uygulama verisi birimi
  - Her ilişkideki her bir öznitelik bir veya daha fazla alan olarak temsil edilecektir
- Veri türlerini seçme
  - Veri Tipi
    - Kurumsal verileri temsil etmek için sistem yazılımı tarafından tanınan bir kodlama şeması
  - Dört hedef:
    - Depolama alanını en aza indirin
    - Alan için tüm olası değerleri temsil eder
    - Saha için veri bütünlüğünü iyileştirin
    - Sahada istenen tüm veri manipülasyonlarını destekleyin
  - Hesaplanan alanlar
    - Diğer veritabanı alanlarından türetilebilen bir alan

Veri Bütünlüğünün Kontrolü

- Varsayılan Değer
  - Bir alan için açık bir değer girilmediği sürece o alanın alacağı değer
- Giriş Maskesi
  - Bir alanın her bir konumu için genişliği ve olası değerleri kısıtlayan bir kod kalıbı
- Menzil Kontrolü
  - Alana girilebilecek değer aralığını sınırlar
- Referans Bütünlüğü
  - Bir ilişkideki bir özniteliğin değerinin (veya varlığının) başka bir ilişkideki aynı özniteliğin değerine (veya varlığına) bağlı olduğunu belirten bütünlük kısıtı
- Boş Değer
  - Alan değerinin eksik olduğunu veya başka bir şekilde bilinmediğini gösteren, 0, boş veya başka bir değerden farklı özel bir alan değeri

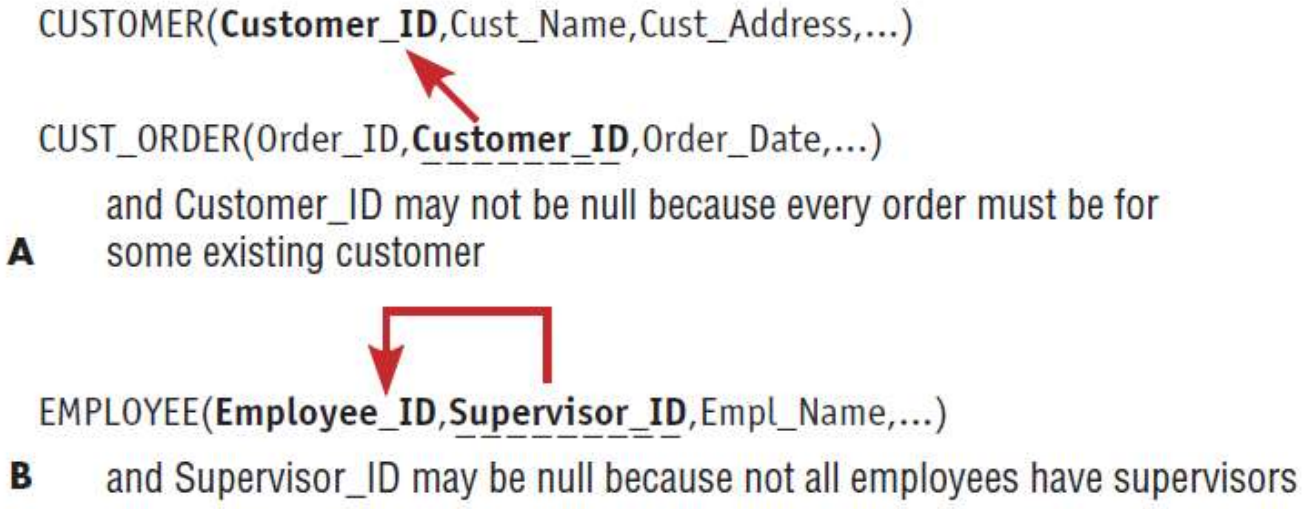


FIGURE 9-17
Examples of referential integrity field controls. (A) Referential integrity between relations, and (B) referential integrity within a relation.

Fiziksel Tabloların Tasarlanması

- İlişkisel Veritabanı İlişkili Tablolar Kümesidir
- Fiziksel Tablo
  - Tablonun her satırındaki alanları belirten adlandırılmış bir satır ve sütun kümesi
- Tasarım Hedefleri
  - İkincil depolamanın (disk alanı) verimli kullanımı
    - Diskler tek bir makine işleminde okunabilecek birimlere ayrılır
    - Alan, bir tablo satırının fiziksel uzunluğu depolama birimiyle neredeyse eşit olarak bölündüğünde en verimli şekilde kullanılır

Fiziksel Tabloların Tasarlanması (devam)

- Tasarım Hedefleri (devam)
  - Verimli veri işleme
    - Veriler ikincil bellekte yan yana depolandığında en verimli şekilde işlenir
- Denormalizasyon
  - Satır ve alanların kullanım yakınlığına dayalı olarak normalleştirilmiş ilişkileri fiziksel tablolara bölme veya birleştirme işlemi
  - Belirli işlemleri diğerlerinin zararına optimize eder
  - Normalleştirmenin kullanılabileceği üç yaygın durum:
    1. Bire bir ilişkisi olan iki varlık
    2. Anahtar olmayan özniteliklere sahip çoktan çoğa bir ilişki
    3. Referans verileri

FIGURE 9-18
Examples of denormalization
(A) Denormalization by columns
(B) Denormalization by rows.

A

Normalized Product Relation

PRODUCT(Product_ID, Description, Drawing_Number, Weight, Color, Unit_Cost, Burden_Rate, Price, Product_Manager)

Denormalized Functional Area Product Relations for Tables

Engineering: E_PRODUCT(Product_ID, Description, Drawing_Number, Weight, Color)
Accounting: A_PRODUCT(Product_ID, Unit_Cost, Burden_Rate)
Marketing: M_PRODUCT(Product_ID, Description, Color, Price, Product_Manager)

Normalized Customer Table

CUSTOMER

Customer_ID	Name	Region	Annual_Sales
1256	Rogers	Atlantic	10,000
1323	Temple	Pacific	20,000
1455	Gates	South	15,000
1626	Hope	Pacific	22,000
2433	Bates	South	14,000
2566	Bailey	Atlantic	12,000

Denormalized Regional Customer Tables

A_CUSTOMER

Customer_ID	Name	Region	Annual_Sales
1256	Rogers	Atlantic	10,000
2566	Bailey	Atlantic	12,000

P_CUSTOMER

Customer_ID	Name	Region	Annual_Sales

B

Denormalized Regional Customer Tables

A_CUSTOMER

Customer_ID	Name	Region	Annual_Sales
1256	Rogers	Atlantic	10,000
2566	Bailey	Atlantic	12,000

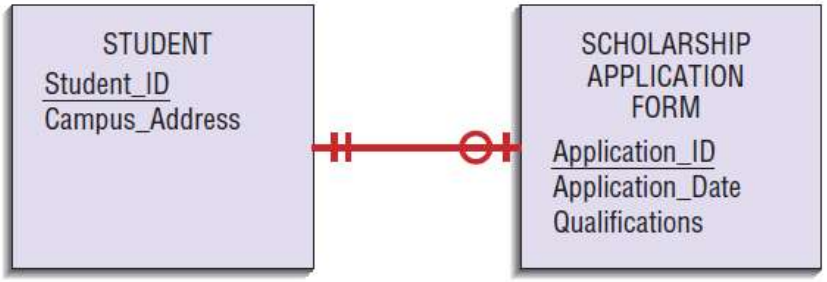
P_CUSTOMER

Customer_ID	Name	Region	Annual_Sales
1323	Temple	Pacific	20,000
1626	Hope	Pacific	22,000

S_CUSTOMER

Customer_ID	Name	Region	Annual_Sales
1455	Gates	South	15,000
2433	Bates	South	14,000

FIGURE 9-19
Possible denormalization situations (A) two entities with a one-to-one relationship (B) a many-to-many relationship with nonkey attributes (C) reference data.



Normalized relations:

STUDENT(Student_ID, Campus_Address, Application_ID)

APPLICATION(Application_ID, Application_Date, Qualifications, Student_ID)

Denormalized relation:

STUDENT(Student_ID, Campus_Address, Application_Date, Qualifications) and Application_Date and Qualifications may be null

(*Note:* We assume Application_ID is not necessary when all fields are stored in one record, but this field can be included if it is required application data.)

A

Normalized relations:

VENDOR(Vendor_ID, Address, Contact_Name)

ITEM(Item_ID, Description)

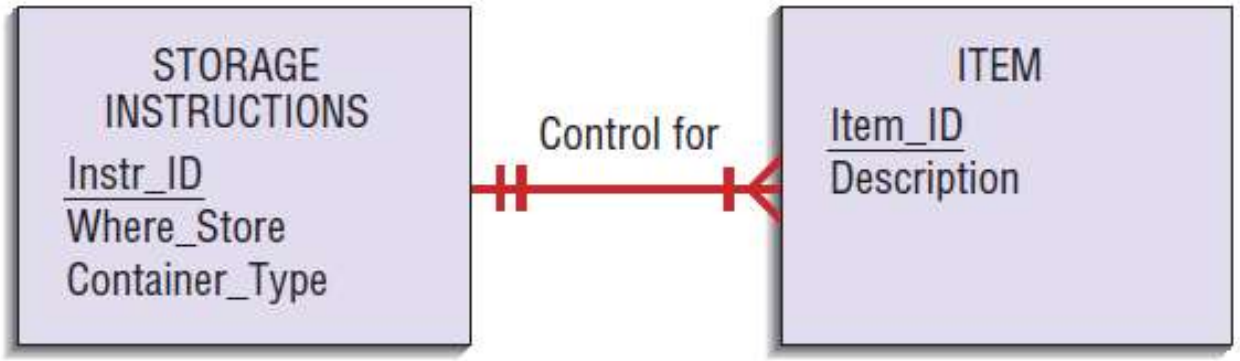
PRICE QUOTE(Vendor_ID, Item_ID, Price)

Denormalized relations:

VENDOR(Vendor_ID, Address, Contact_Name)

ITEM-QUOTE(Vendor_ID, Item_ID, Description, Price)

B



Normalized relations:

STORAGE(Instr_ID, Where_Store, Container_Type)

ITEM(Item_ID, Description, Instr_ID)

Denormalized relation:

ITEM(Item_ID, Description, Where_Store, Container_Type)

C

Fiziksel Tabloların Tasarlanması (devam)

- Tablo Satırlarını Düzenleme
  - Fiziksel Dosya
    - İkincil belleğin bitişik bir bölümünde saklanan adlandırılmış bir tablo satırları kümesi
  - Kullanılan veritabanı yönetim yazılımına bağlı olarak her tablo fiziksel bir dosya olabilir veya tüm veritabanı tek bir dosya olabilir

Fiziksel Tabloların Tasarlanması (devam)

- Dosya Organizasyonu
  - Bir dosyanın kayıtlarını fiziksel olarak düzenlemek için bir teknik
  - Dosya organizasyonunu seçmek için hedefler:
    1. Hızlı veri alma
    2. İşlemleri işlemek için yüksek verim
    3. Depolama alanının verimli kullanımı
    4. Arızalara veya veri kaybına karşı koruma
    5. Yeniden yapılanma ihtiyacının en aza indirilmesi
    6. Büyümeye uyum sağlama
    7. Yetkisiz kullanıma karşı güvenlik

Fiziksel Tabloların Tasarlanması (devam)

- Dosya Düzenleme Türleri
  - Sıralı
    - Dosyadaki satırlar birincil anahtar değerine göre sırayla saklanır
    - Kayıtların güncellenmesi ve eklenmesi dosyanın yeniden yazılmasını gerektirebilir
    - Kayıtların silinmesi alanın boşa harcanmasına neden olur
  - Endeksli
    - Satırlar sıralı veya sırasız olarak saklanır ve yazılımın tek tek satırları bulmasını sağlayan bir dizin oluşturulur
    - Dizin
      - Bir dosyada bazı koşulları sağlayan satırların konumunu belirlemek için kullanılan bir tablo
    - İkincil İndeks
      - Birden fazla satırın aynı değer kombinasyonuna sahip olabileceği bir alan kombinasyonuna dayalı dizin

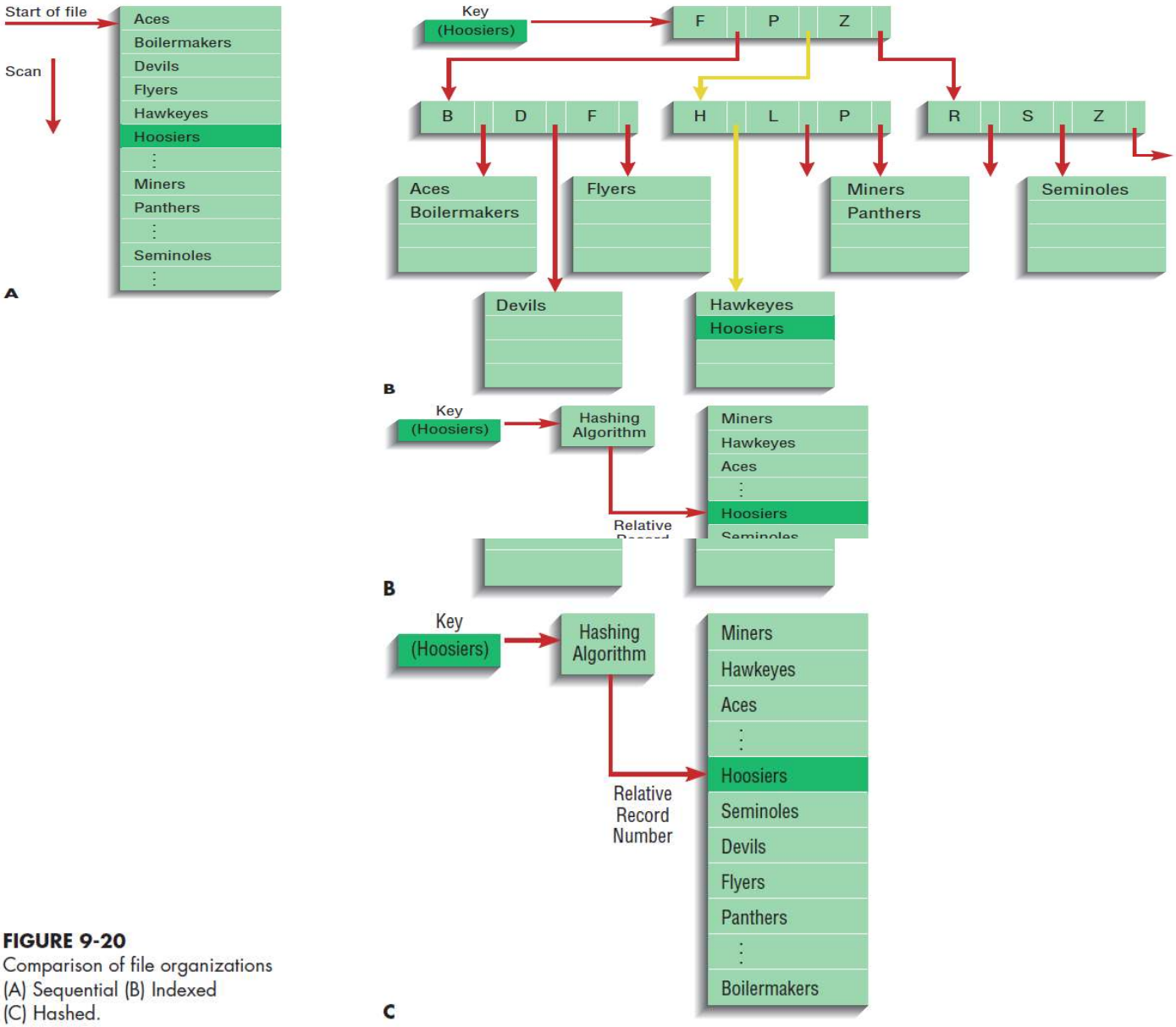


FIGURE 9-20
Comparison of file organizations
(A) Sequential (B) Indexed
(C) Hashed.

Fiziksel Tabloların Tasarlanması (devam)

- İndeks seçimi için yönergeler
  1. Her dosyanın birincil anahtarı için benzersiz bir dizin belirtin
  2. Yabancı anahtarlar için bir dizin belirtin
  3. Verileri almak amacıyla niteleme, sıralama ve gruplama komutlarında başvurulan anahtar olmayan alanlar için bir dizin belirtin
- Karma Dosya Organizasyonu
  - Her satır için adres bir algoritma kullanılarak belirlenir

Dosyalar için Kontroller Tasarlama

- Yedekleme Teknikleri
  - Dosyaların periyodik olarak yedeklenmesi
  - İşlem günlüğü veya denetim izi
  - Değişiklik günlüğü
- Veri Güvenliği Teknikleri
  - Kodlama veya şifreleme
  - Kullanıcı hesabı yönetimi
  - Kullanıcıların verilerle doğrudan çalışmasının yasaklanması; kullanıcıların dosyaları yalnızca doğrulama kontrollerinden sonra güncelleyen bir kopya ile çalışması

PVF WebStore: Veritabanı Tasarımı

- Tasarım süreci diğer uygulamalardan farklı değildir