

YAZILIM MÜHENDİSLİĞİNE GİRİŞ

HAFTA 9 VERİ TABANI SİSTEMLERİ

Dr. Öğr. Üyesi Ebru Aydoğan
eaydogan@mehmetakif.edu.tr

Ders İçeriği: Hafta 9

- Veri Tabanı Temelleri
- İlişkisel Model
- Nesneye Yönelik Veri Tabanı Sistemleri
- Veri Tabanının Bütünselliğini Korumak
- Geleneksel Dosya Yapıları
- Veri Madenciliği
- Veri Tabanı Teknolojisinin Sosyal Etkileri

Veri Tabanı Temelleri

- **Veri tabanı** terimi, veri seti içindeki ilişkiler aracılığı ile farklı perspektifte bilgiye erişebilme anlamında çok boyutlu bir veri setini ifade etmektedir.
- Büyük bir veri bütününe kullanıcıların kendilerine faydalı olabilecek bilgileri aramalarına olanak tanıyan ve ilgiyi bilgiyi çekip çıkarmalarına yardımcı olabilecek soyut araca çeviren bir sistemdir.

Veri Tabanı Temelleri

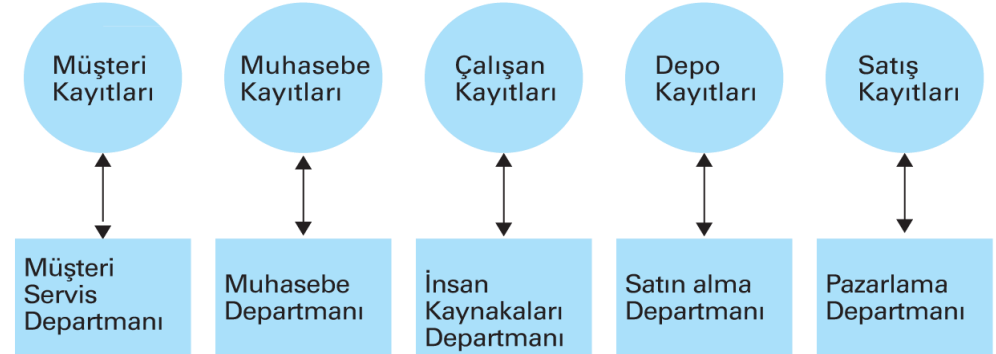
➤ Veri Tabanı Sistemlerinin Önemi

- Tarihsel olarak, hesaplama araçları bilgi yönetimi alanında daha geniş kullanım şekilleri buldukça her bir uygulama kendisine ait veri seti ile birlikte farklı bir sistem olarak kodlanmıştır.
- Veri tabanı sistemleri dünya çapına yayılmış bilgisayar ağ sistemi, yani kısaca www diye adlandırılan sistem, üzerinde barındırılan çok sayıdaki popüler internet sitelerinin altyapısını oluşturmaktadır.

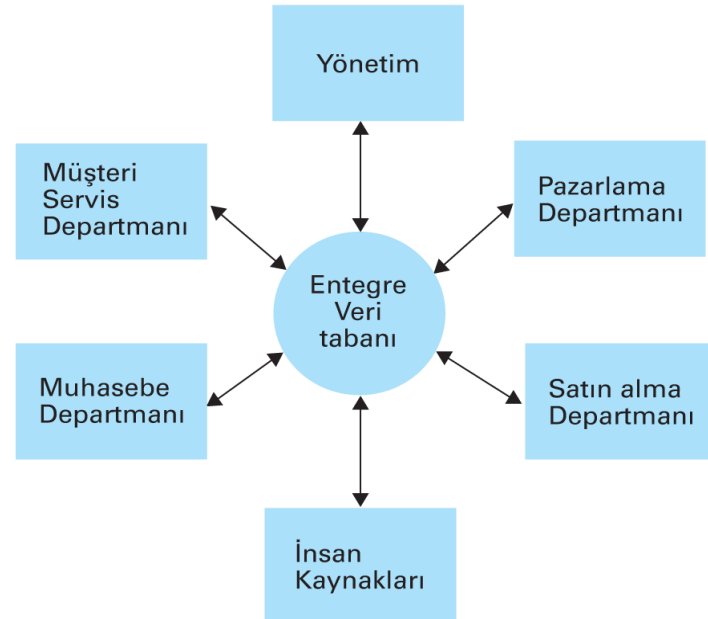
Veri Tabanı Temelleri

- Bir dosya sistemi ile veri tabanı sistemine ait düzenleme

a. Dosya-tabanlı bir bilgi sistemi



b. Veri tabanı sistemine dayalı bir bilgi sistemi



Veri Tabanı Temelleri

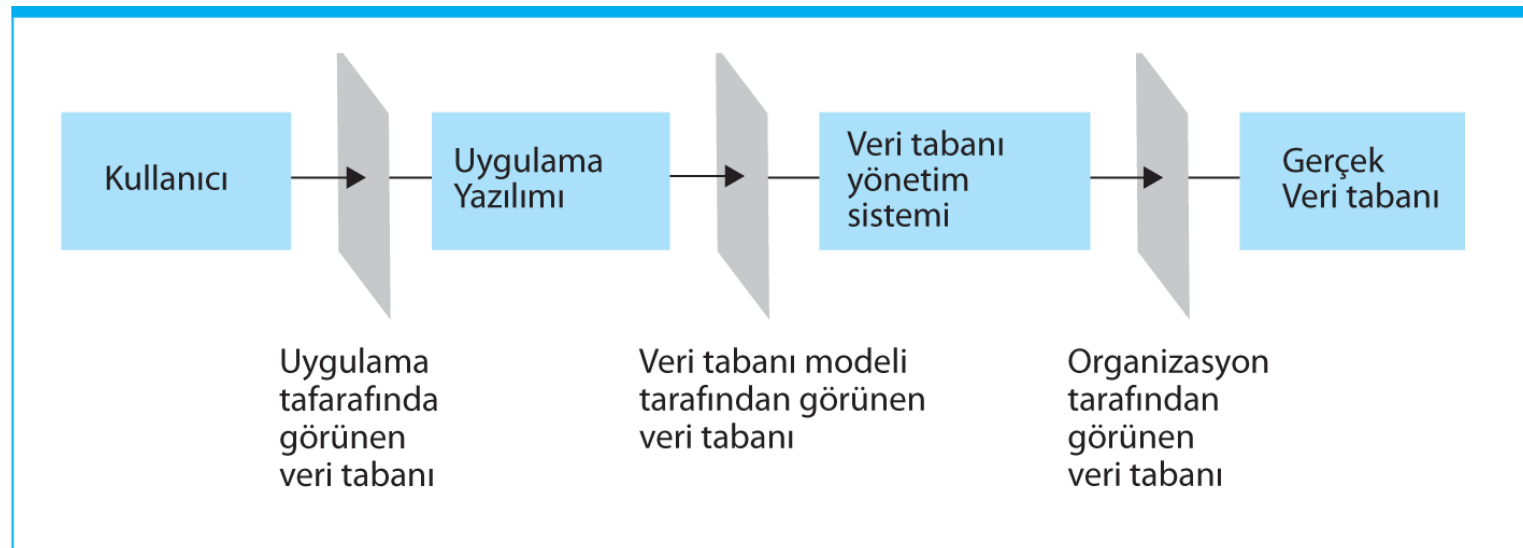
➤ Şemaların Önemi

- Farklı kullanıcılara veri tabanındaki bilgilere değişik düzeylerde erişim izni verebilme veri tabanlarında sıklıkla şema veya alt şemaların kullanımına bağlıdır.
- Şema, veri tabanı yazılımı tarafından kullanılan toplam veri tabanı yapısının şematik olarak ifade edilmesidir.
- Alt şema ise veri tabanının herhangi bir kullanıcının kullanım ihtiyaçlarına göre düzenlenmiş bir kısmının şematik gösterimidir.

Veri Tabanı Temelleri

➤ Veri Tabanı Yönetim Sistemleri (VTYS)

- Tipik bir veri tabanı uygulaması çok katmanlı bir yazılımdan oluşur.
- Biz bunu iki ana katman ile gruplayabiliriz:
 - Uygulama katmanı
 - Bir veri tabanı yönetim katmanı
- Bir veri tabanı implementasyonunun kavramsal katmanları



Veri Tabanı Temelleri

➤ Veri Tabanı Modelleri

- VTYS kullanıcıya tablolar, yani verilerin içeride satır ve sütunlar halinde saklandığı izlenimine, göre kodlama yapmasına olanak sağlayan araçları ve kullanıcının kodlarını VTYS içerisinde verilerin gerçekte saklanmış haline dönüştüren araçları barındırırlar.
- Veri tabanın bu şekildeki görünümüne veri tabanı modeli adı verilir.

İlişkisel Model

- İlişkisel model verinin, adına **ilişkiler** denilen dikdörtgen tablolar halinde kaydedildiğini tasvir eder.
- İlişkideki her bir satır bir **kayıt** (tuple) olarak adlandırılır.
- İlişkideki sütunlar ise genellikle **öznitelik** (attribute) olarak adlandırılır.

İlişkisel Model

➤ İlişkisel Tasarım Konuları

- İlişkisel veri tabanı tasarılmanın en can alıcı noktası veri tabanının oluşturan ilişkileri tasarlamaktır.
- Örnek: Çalışanların bilgisini içeren bir ilişki

Empl Id (Çalışan_Numarası)	Name (İsim)	Address (Adres)	SSN (Sosyal_Güvenlik_Numarası)
25X15	Joe E. Baker	33 Nowhere St.	111223333
34Y70	Cheryl H. Clark	563 Downtown Ave.	999009999
23Y34	G. Jerry Smith	1555 Circle Dr.	111005555
•	•	•	•
•	•	•	•
•	•	•	•

İlişkisel Model

➤ İlişkisel Tasarım Konuları

- İlişkisel tasarımda tek bir ilişkide birden fazla kavram bulundurulmamalı:
 - Gereksiz tutulan veriye yol açabilir.
 - Bir satırı silmek, gerekli ancak ilgisiz bilgileri de silebilir.
- İlişkisel tasarımı iyileştirmek için:
 - Ayırıştırma: Bir ilişkinin sütunlarını iki veya daha fazla ilişkiye bölmek, ilişkileri korumak için gerekli olan sütunları çoğaltmak
 - Kayıpsız ayırıştırma: Herhangi bir bilgi kaybetmeyen "doğru" ayırıştırma

İlişkisel Model

➤ Üç ilişki içeren bir
Çalışan
(Employee) veri
tabanı

EMPLOYEE ilişkisi

Empl Id	Name	Address	SSN
25X15	Joe E. Baker	33 Nowhere St.	111223333
34Y70	Cheryl H. Clark	563 Downtown Ave.	999009999
23Y34	G. Jerry Smith	1555 Circle Dr.	111005555

JOB ilişkisi

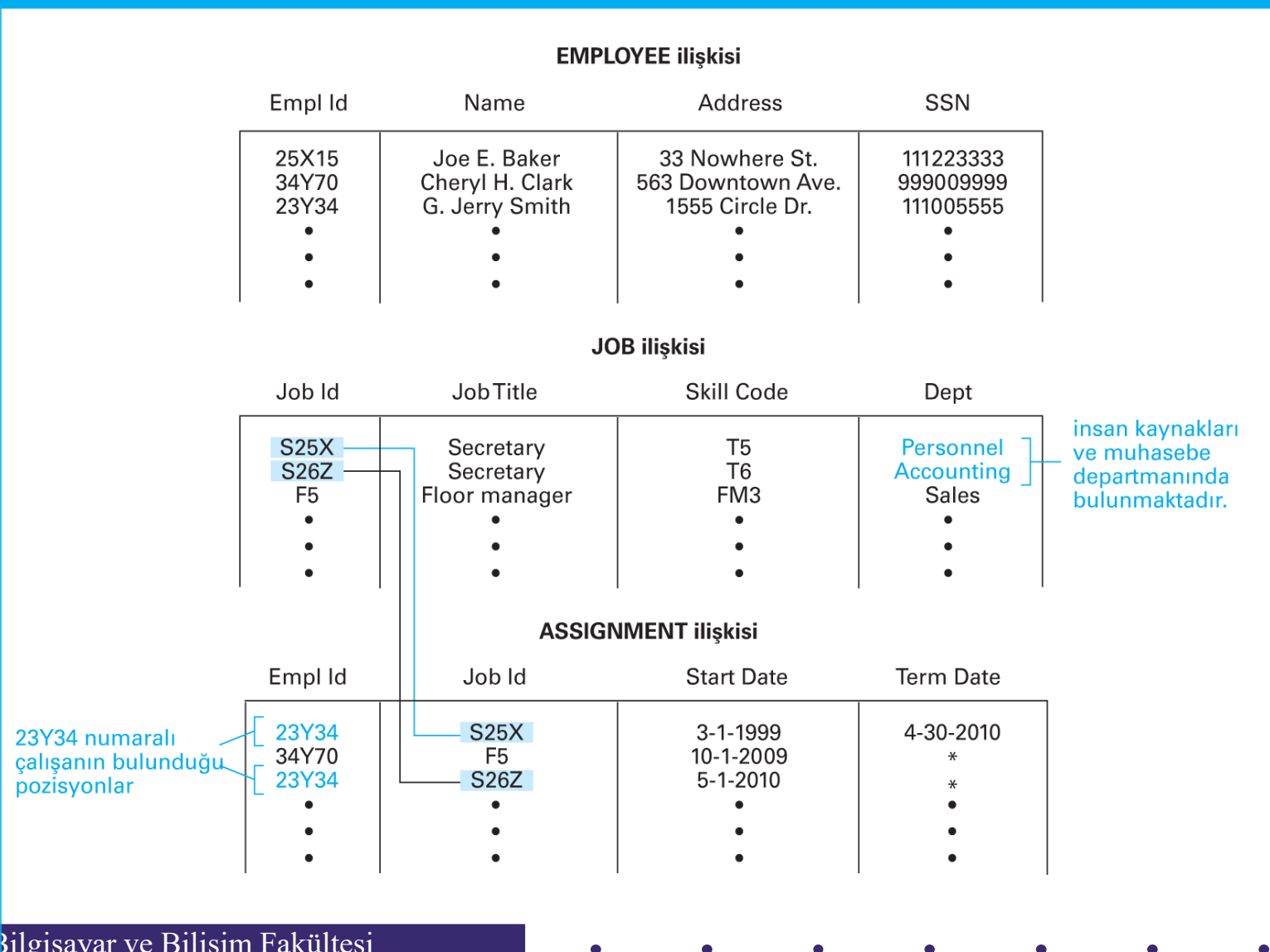
Job Id	Job Title	Skill Code	Dept
S25X	Secretary	T5	Personnel
S26Z	Secretary	T6	Accounting
F5	Floor manager	FM3	Sales
•	•	•	•
•	•	•	•
•	•	•	•

ASSIGNMENT ilişkisi

Empl Id	Job Id	Start Date	Term Date
23Y34	S25X	3-1-1999	4-30-2010
34Y70	F5	10-1-2009	*
23Y34	S26Z	5-1-2010	*
•	•	•	•
•	•	•	•
•	•	•	•

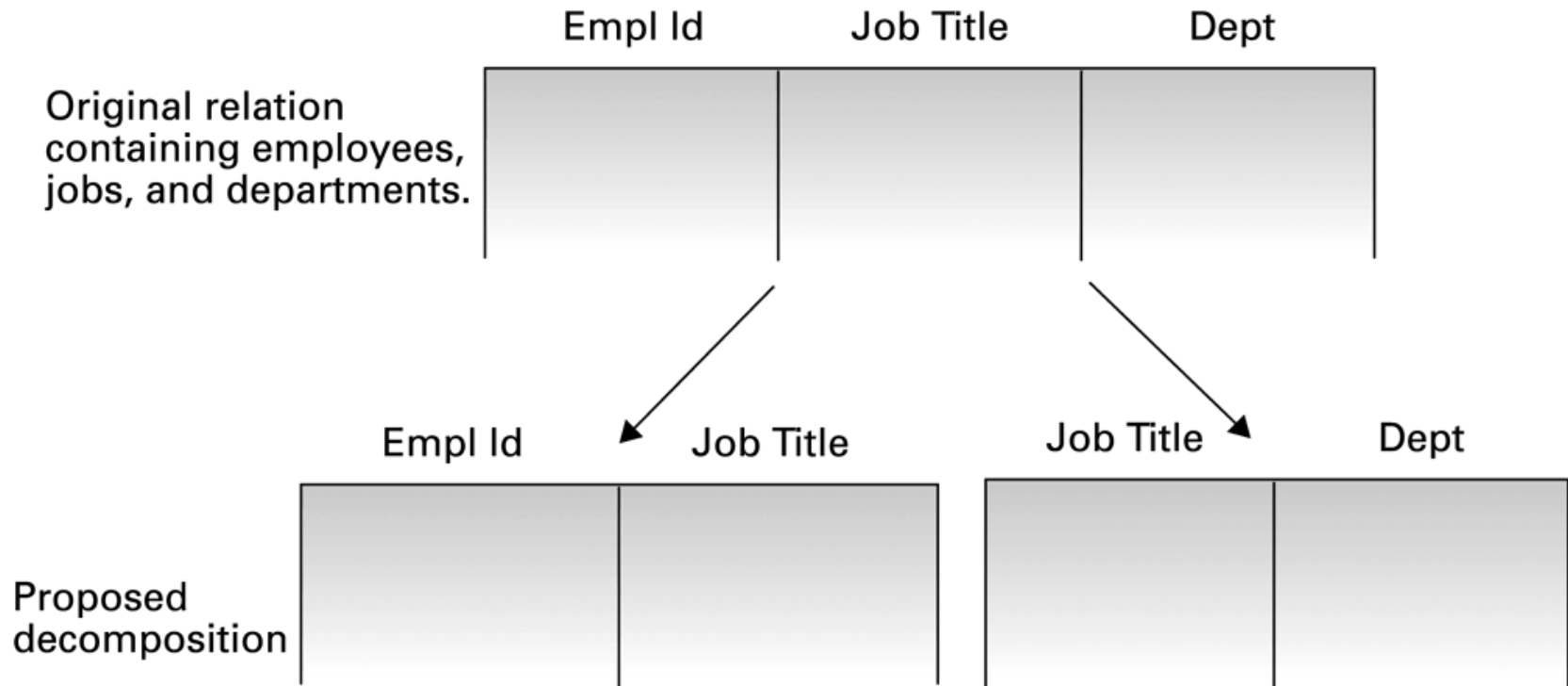
İlişkisel Model

➤ 23Y34 numaralı çalışanın çalışmış olduğu departmanları bulma



İlişkisel Model

➤ Bir ilişki ve önerilen ayrıştırma



İlişkisel Model

➤ İlişkisel Operasyonlar

- İlişkilerden oluşan veri tabanında bilginin nasıl çıkarıldığına bakalım.
- İlişkiler üzerinde gerçekleştirme ihtimalimiz olan işlemleri inceleyelim.

İlişkisel Model

➤ İlişkisel Operasyonlar (SELECT)

- Bir ilişkide bazı kayıtları (satırları) seçmemiz gerektiğinde kullanılır.

EMPLOYEE ilişkisi	Empl Id	Name	Address	SSN
	25X15	Joe E. Baker	33 Nowhere St.	111223333
	34Y70	Cheryl H. Clark	563 Downtown Ave.	999009999
	23Y34	G. Jerry Smith	1555 Circle Dr.	111005555
	•	•	•	•
	•	•	•	•
	•	•	•	•

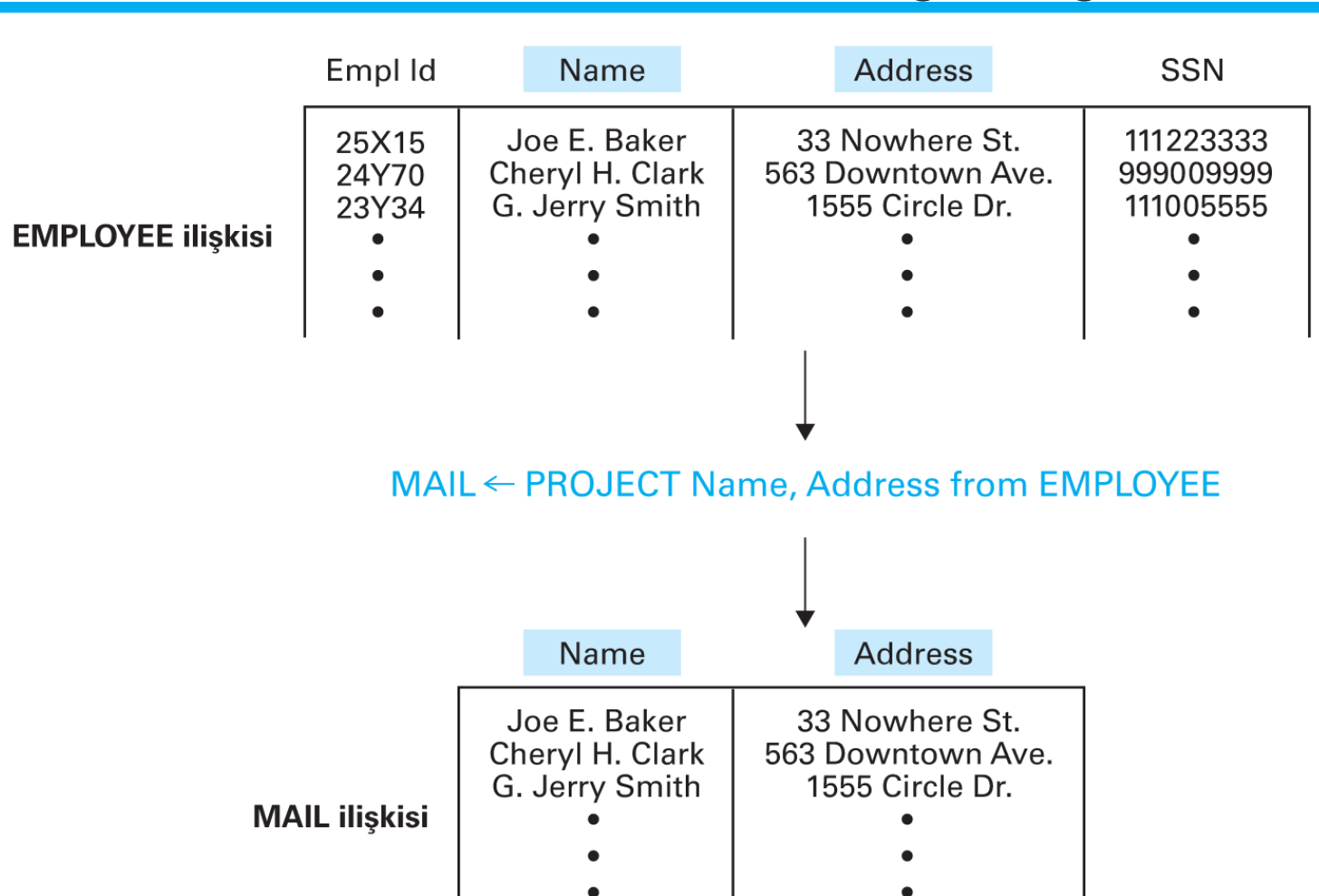
NEW ← SELECT from EMPLOYEE where EmplId = '34Y70'

NEW isimli ilişki	Empl Id	Name	Address	SSN
	34Y70	Cheryl H. Clark	563 Downtown Ave.	999009999

İlişkisel Model

➤ İlişkisel Operasyonlar (PROJECT)

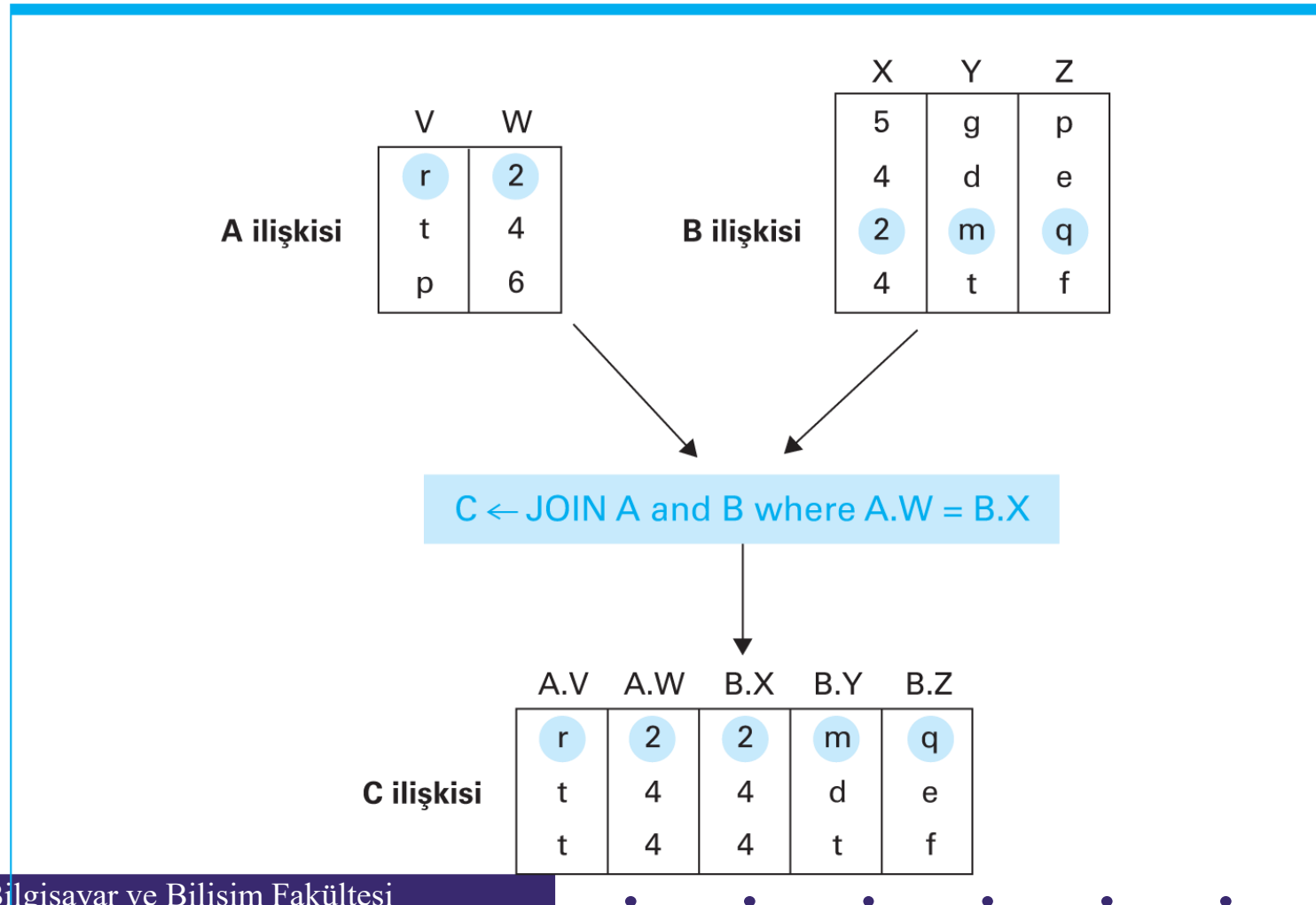
- Bir ilişkideki bazı sütunları seçmemiz gerektiğinde kullanılır.



İlişkisel Model

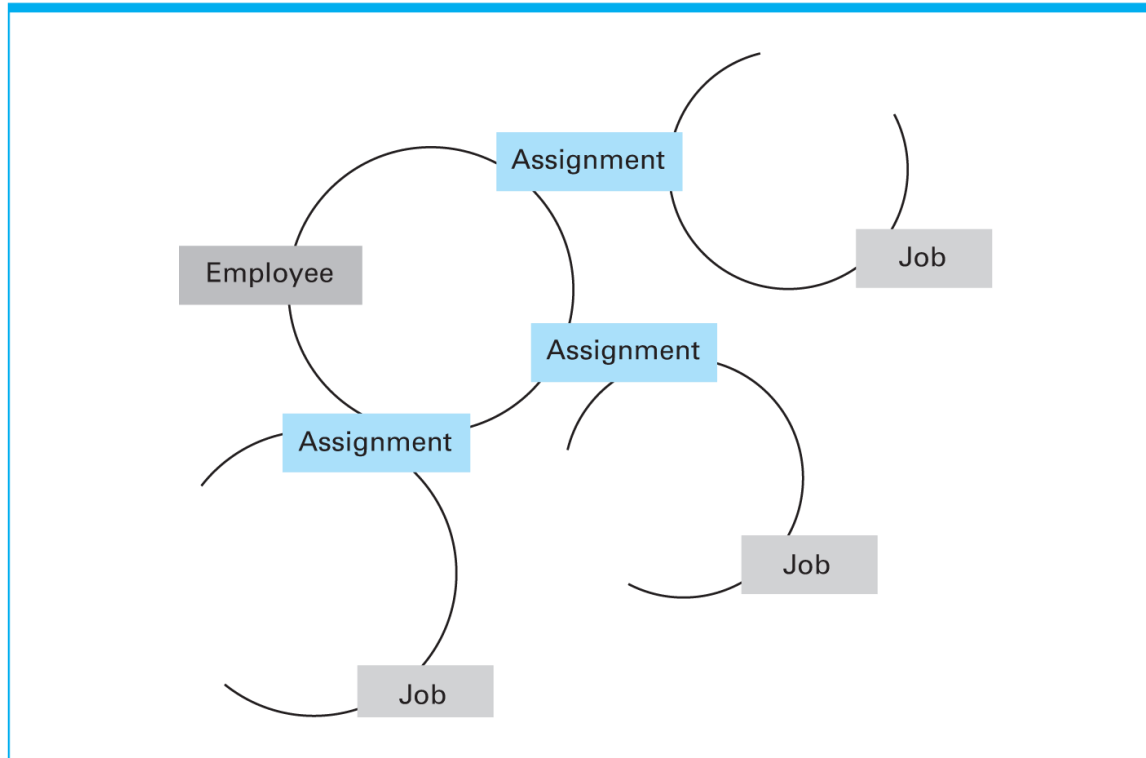
➤ İlişkisel Operasyonlar (JOIN)

- İki ilişkinin birleştirilmesi için kullanılır.



Nesneye Yönelik Veri Tabanı Sistemleri

- Diğer bir veri tabanı modeli nesneye yönelik paradigmaya dayanır.
- Bu yaklaşım nesneye yönelik veri tabanı modelinin ortaya çıkmasına sebep olur ki burada nesneler birbirine bağlanarak ilişkilendirilir.
- Nesneye yönelik bir veri tabanında nesneler arasındaki bağlantılar



Nesneye Yönelik Veri Tabanı Sistemleri

➤ Avantajları

- Nesneye yönelik tasarım, niteliklerin uygulama ayrıntılarınının gizlenmesine olanak tanır
 - Örnek: “name” özelliğinin farklı formatları vardır, name özelliğini bir nesne olarak uygulamak daha esnektir
- Egzotik veri türlerini işleyebilir
 - Örnek: bir multimedya veri ögesi genellikle çeşitli niteliklerden (ses, video, grafikler, açıklamalar) oluşur; nesneye yönelik tasarım konsepti bunları tek bir veri nesnesinde kapsayabilir
- Akıllı varlıkları saklayabilir
 - Zeka nesnenin yöntemlerinin içindedir yani, eğer veri tabanı akıllıysa, VTYS daha basit olabilir

Veri Tabanının Bütünselliğini Korumak

- Kişisel kullanımlar için oluşturulmuş mevcut olan ucuz veri tabanı yönetim sistemleri göreceli olarak basit sistemlerdir.
- Genelde tek bir amaçları vardır kullanıcıyı veri tabanının bilgisayarda nasıl oluşturulduğu konusundaki teknik detaylardan arındırmak.
- Veri tabanında gerçekleştirilen bir işlem, hep birlikte gerçekleşmesi gereken bir dizi işlemdir
 - Örnek: banka hesapları arasında para transferi

Veri Tabanının Bütünselliğini Korumak

➤ Commit/Rollback Protokolü

- İşlem günlüğü (transaction log), işlemin gerçekleşmesine izin verilmeden önce oluşturulan, her işlemin etkinliklerinin kalıcı kaydıdır.
- Bir işlemdeki tüm adımların log dosyasında kaydedilmesi anına **commit (hazır) noktası** adı verilir.
- Log dosyası geriye doğru kullanılır. Yani o ana kadar yapılan işlem adımlarının geri alınması olayına **rollback (geri al)** adı verilir.

Veri Tabanının Bütünselliğini Korumak

➤ Kilitleme

- Eşzamanlı erişimde yanlış özet problemi ve kayıp güncelleme problemi gibi anormal durumlara karşı koruma sağlamak için bu düzenleyiciler bir **kilitleme protokolü (locking protocol)** aracılığı ile çalışırlar.
- Yani, veri tabanı içerisinde bileşenlerin hangi işlemler tarafından kullanıldığı kayıt altına alınır.
- İki türlü kilitleme mevcuttur:
 - **Paylaşımlı kilitleme:** veri okunurken kullanılır.
 - **Özel kilitleme:** veri değiştirilirken kullanılır.
- VTYS'de oluşan kilitlenmeyi çözmek için kullanılan yaygın bir yöntem:
 - Yarala-bekle protokolü: daha genç olan işlem yaşı işlem tarafından zorla alınır.

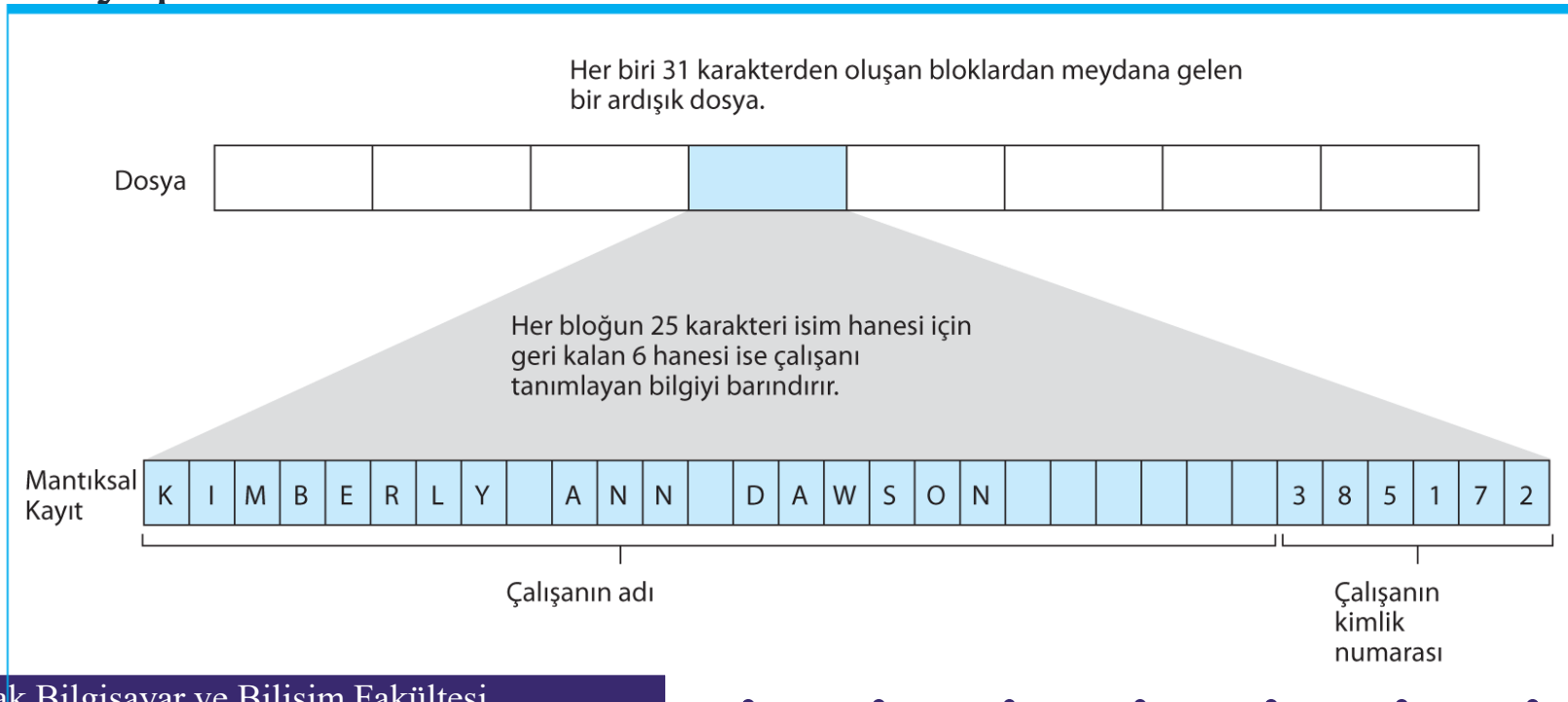
Geleneksel Dosya Yapıları

- **Dosya yapıları**, veri depolamanın ve bilgi edinme sistemlerinin tarihsel başlangıcını oluştururlar ki bu yapılar zamanla günümüz modern veri tabanı teknolojilerine evrilmiştir.
- Dosya yapıları için geliştirilmiş olan indeksleme ve hash'leme günümüzdeki veri tabanları için önemli araçlardır.

Geleneksel Dosya Yapıları

➤ Ardışık Dosyalar

- **Ardışık bir dosya;** en başından en sonunda kadar seri şekilde erişilen, sanki bilginin uzun bir satırdan oluştuğu izlenimi veren bir dosyadır.
- Metin dosyası olarak uygulanan basit bir çalışan dosyasının yapısı



Geleneksel Dosya Yapıları

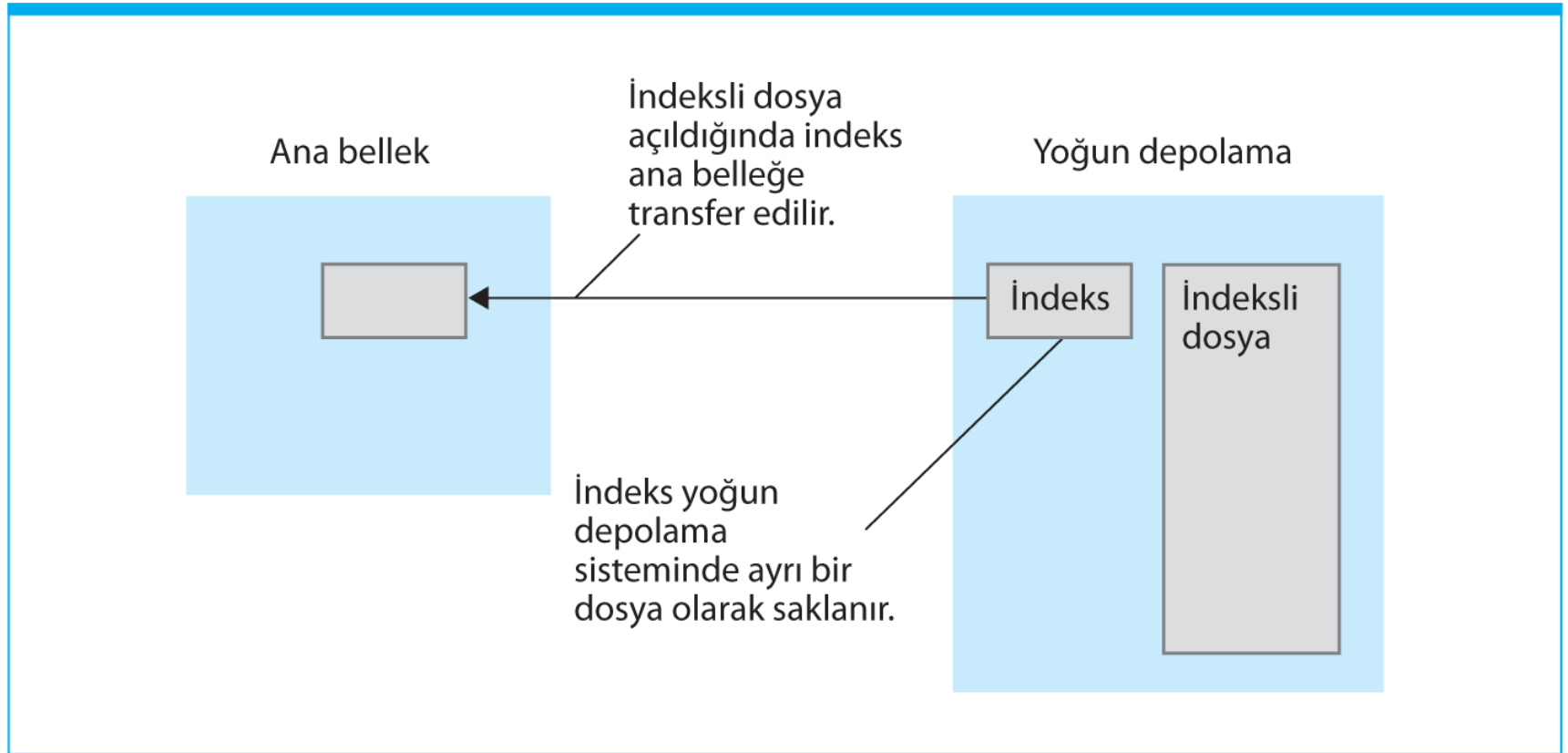
➤ İndekslenmiş Dosyalar

- Ardışık dosyalar, verinin dosya içeriğinin saklandığı sıraya göre işlenmesinin gerektiği durumlar için ideal bir depolamadır.
- Popüler bir çözüm bir kitapta herhangi bir konunun nerede olduğunu bulmaya yarayan indeksleme gibi bir indeksleme sistemini kullanmaktır. Bu tür bir dosya sistemi **indekslenmiş dosya** olarak isimlendirilir.

Geleneksel Dosya Yapıları

➤ İndekslenmiş Dosyalar

- İndekslenmiş bir dosyayı açma



Geleneksel Dosya Yapıları

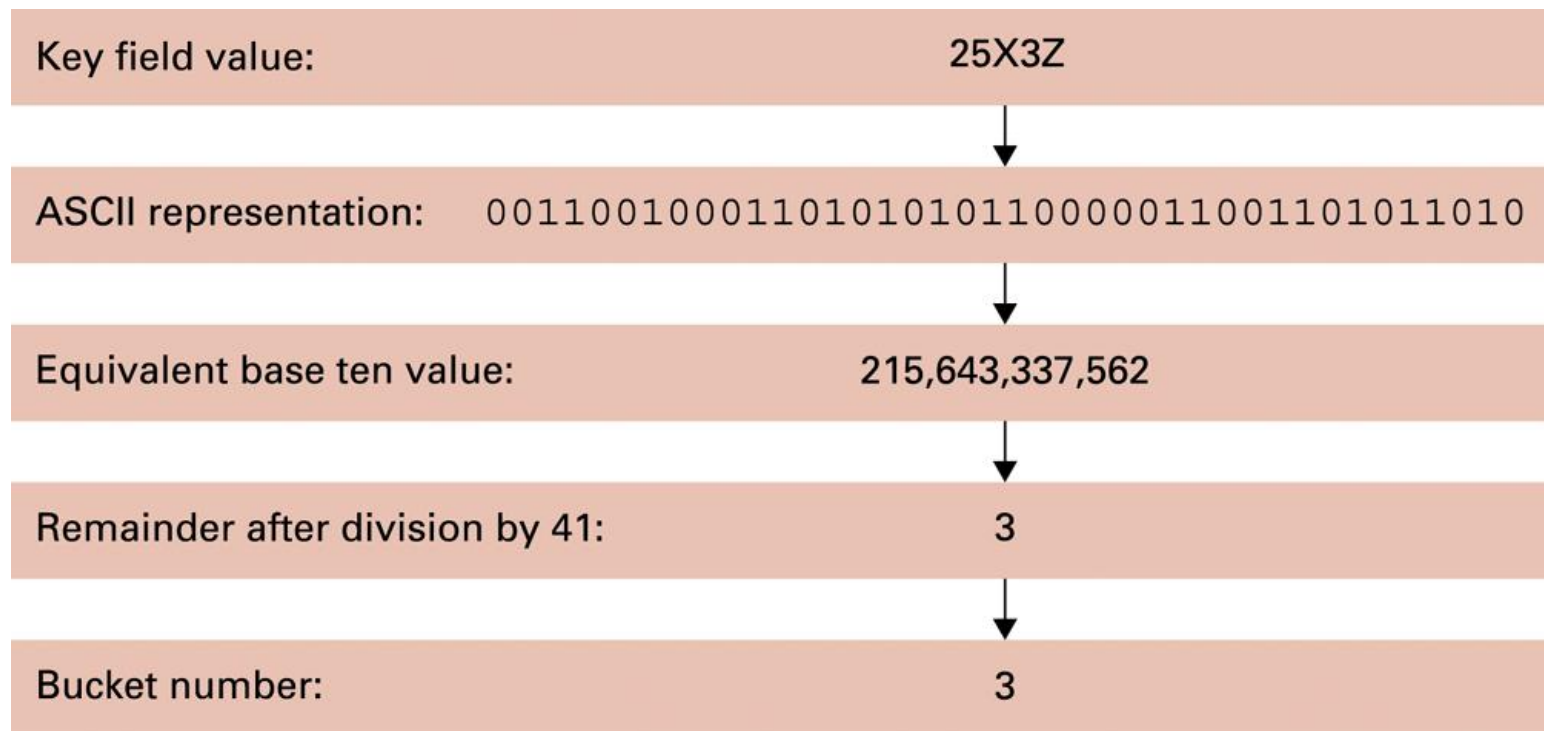
➤ Hash Dosyaları

- İndeksleme göreceli olarak bir veri depolama yapısındaki içeriğe hızlı bir erişim olanağı sunarken, bu işi indeksleme yönetimi pahasına göre gerçekleştirir.
- Hash'leme bu tür bir maliyet gerektirmeden benzer bir erişim olanağı sağlayan bir tekniktir.
 - Her kaydın bir anahtarı (key) vardır
 - Ana dosya paketlere (buckets) bölünmüştür
 - Hash fonksiyonu, her anahtar değer için bir paket numarası hesaplar
 - Her kayıt, anahtarının hash değerine karşılık gelen grupta saklanır

Geleneksel Dosya Yapıları

➤ Hash Dosyaları

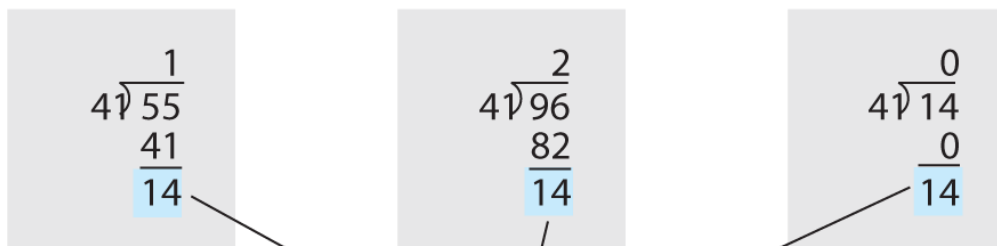
- 25X3Z anahtar kelimesini 41 kovadan birine hash'leme



Geleneksel Dosya Yapıları

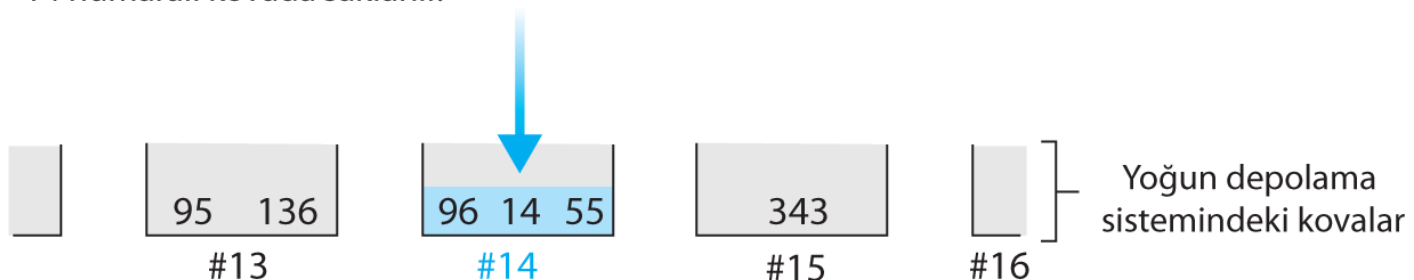
➤ Hash Dosyaları

■ Hash sisteminin esasları



Kalanlar

14, 55 ve 96 anahtar alan değerleri 41'e bölündüğünde her biri 14 kalanını verir. Bu nedenle kayıtların hepsi 14 numaralı kovada saklanır.



Veri Madenciliği

- Veri madenciliği, veriler içerisinde var olan örüntüleri keşfetmeye yarayan tekniklerden oluşan, veri tabanı teknolojisi ile yakından ilgili olan ve hızla genişleyen bir konudur.
- Veri madenciliği sık güncellemeler gerektiren “çevrim içi” operasyonel veri tabanlarından ziyade **veri ambarları** olarak adlandırılan statik veri koleksiyonları üzerinde uygulanmaktadır.
- Veri madenciliğinin en yaygın iki biçimi, sınıf tanımlama ve sınıf ayrıklaştırmadır.
- Veri madenciliğinin formları;
 - Kümeleme analizi
 - Birliktelik analizi
 - Aykırı değer analizi
 - Sıralı örüntü analizi

Veri Tabanı Teknolojisinin Sosyal Etkileri

- Veri tabanı teknolojisinin gelişmesi ile bir zamanlar gizemli kayıtlarda saklı olan bilgi erişilebilir hale gelmiştir.
- Teknoloji çok büyük miktarlarda verinin toplanmasını kolaylaştırdı, farklı veri setlerinin birleştirilmesi veya karşılaştırılması yoluyla normalde kapalı kalacak bilgilerin ortaya çıkmasına olanak sağladı.
- Dallanmalar/kırılımlar pozitif ve negatif olarak çok büyüktür.