

Şekil 1.5'te gösterildi, ancak tablolar, verileri farklı şekillerde görüntüleyebilmeniz ve basit soruları yanıtlayabilmeniz için kolayca düzenlenebilir şekilde yapılandırılmıştır.

Şekil 1.6'da gösterilen geliştirilmiş yapı ile, standart bir veri işleme dili kullanarak aşağıdakileri gerçekleştirebilirsiniz:

- Çalışanları soyadına göre alfabetik olarak listeleyin:

```
SELECT * FROM EMPLOYEE ORDER BY EMPLOYEE_LNAME;
```

- Temel Veritabanı İşleme sertifikasına sahip çalışan sayısını belirleyin:

```
SELECT COUNT(*)
FROM SKILL JOIN CERTIFIED ON SKILL.SKILL_ID = CERTIFIED.SKILL_ID
WHERE SKILL_NAME = 'Basic Database Manipulation';
```

Bu komutlar hakkında daha fazla bilgiyi Bölüm 7, Yapılandırılmış Sorguya Giriş'te bulabilirsiniz

Her beceri adı yalnızca bir kez saklandığından, adların hecelenemeyeceğini veya farklı çalışanlar için farklı şekilde kısaltılmıştır. Ayrıca, bir çalışanın ek sertifikasyonu dördüncü veya beşinci bir beceri ile tabloların yapısında değişiklik yapılmasını gerektirmez.

Bir işlem veritabanı tasarlamak, doğru ve tutarlı veri ile operasyonel hız üzerine odaklanır. Bir veri ambarı veritabanı tasarlamak ise tarihsel ve toplanmış verilerin kullanılmasına odaklanır. Merkezi, tek kullanıcı ortamında kullanılacak bir veritabanı tasarlamak, dağıtılmış, çok kullanıcı bir veritabanı tasarımından farklı bir yaklaşım gerektirir. Bu kitap, işlemlere dayalı, merkezi, tek kullanıcı ve çok kullanıcı veritabanlarının tasarımına odaklanmaktadır. Bölüm 12 ve 13, ayrıca dağıtılmış ve veri ambarı veritabanı tasarımcılarının karşılaştığı kritik sorunları incelemektedir.

Entegre edilmiş bilgilerin uygun veri depolarını tasarlamak, çoğu veritabanında bulunan iki boyutlu tablo yapılarıyla yapılan bir çözümleme sürecidir. Entegre edilen veriler, her bir parçası kendi tablosunda saklanacak şekilde doğru bir şekilde çözümlenmelidir. Ayrıca, bu tablolar arasındaki ilişkiler dikkatlice göz önünde bulundurulmalı ve uygulanmalıdır, böylece verinin entegre görünümü daha sonra son kullanıcı için bilgi olarak yeniden oluşturulabilir. İyi tasarlanmış bir veritabanı, veri yönetimini kolaylaştırır ve doğru ve değerli bilgiler üretir. Kötü tasarlanmış bir veritabanı, izlenmesi zor hatalar için bir üreme alanına dönüşebilir ve bu da kötü kararlar alınmasına yol açabilir, kötü kararlar ise bir organizasyonun başarısızlığına yol açabilir. Veritabanı tasarımı, şansa bırakılacak kadar önemli değildir. Bu yüzden üniversite öğrencileri veritabanı tasarımını öğrenir, her türden ve boyuttan organizasyonlar personelini veritabanı tasarımı seminerlerine gönderir ve veritabanı tasarımı danışmanları genellikle iyi bir yaşam kazanır.

1-5 Dosya Sistemi Veri İşlemenin Evrimi:

Bir veritabanının ne olduğunu, ne işe yaradığını ve nasıl doğru şekilde kullanılacağını anlamak, bir veritabanının ne olmadığına bakarak netleştirilebilir. Dosya sistemi veri işleminin evrimine dair kısa bir açıklama, veritabanlarının aşmaya çalıştığı veri erişim sınırlamalarını anlamada faydalı olabilir. Bu sınırlamaları anlamak, veritabanı tasarımcıları ve geliştiricileri için önemlidir çünkü veritabanı teknolojileri bu sorunları sihirli bir şekilde ortadan kaldırmaz veritabanı teknolojileri, bu sorunlardan kaçınan çözümler oluşturmayı sadece daha kolay hale getirir. Önceki sistemlerin tuzaklarından kaçınan veritabanı tasarımları oluşturmak, tasarımcının bu sorunları ve bunlardan nasıl kaçınılacağına dair bilgi sahibi olmasını gerektirir; aksi takdirde, veritabanı teknolojileri, yerine geçtikleri teknolojilerden ve tekniklerden daha iyi (ve potansiyel olarak daha kötü!) olmayacaktır.

data processing (DP) specialist

The person responsible for developing and managing a computerized file processing system.

field

A character or group of characters (alphabetic or numeric) that has a specific meaning. A field is used to define and store data.

record

A logically connected set of one or more fields that describes a person, place, or thing.

file

A collection of related records. For example, a file might contain data about the students currently enrolled at Gigantic University.

1-5a Manuel Dosya Sistemleri

Tarihsel olarak, veriler, verilerin beklenen kullanımını kolaylaştırmak için organize edilmiş kağıt ve kalem manuel sistemlerinde tutulmuştur. Tipik olarak bu, dosya klasörleri ve dosya dolaplarından oluşan bir sistem aracılığıyla gerçekleştirilirdi. Ancak kuruluşlar büyüdükçe ve raporlama gereksinimleri daha karmaşık hale geldikçe, verileri manuel bir dosya sisteminde tutmak daha zor hale geldi. Bu nedenle şirketler yardım için bilgisayar teknolojisine başvurdu

1-5b Bilgisayarlı Dosya Sistemleri

Manuel dosya sistemlerinden rapor üretmek yavaş ve zahmetliydi. Aslında, bazı işletme yöneticileri, iyi tasarlanmış bir manuel sistem kullanıldığında bile, her çeyrekte haftalarca süren yoğun çabalara yol açan devlet tarafından dayatılan raporlama gereklilikleriyle karşı karşıya kaldı. Bu nedenle, verileri takip edecek ve gerekli raporları üretecek bilgisayar tabanlı bir sistem oluşturmak için bir veri işleme (DP) uzmanı işe alındı.

Başlangıçta, dosya sistemindeki bilgisayar dosyaları manuel dosyalara benziyordu. Küçük bir sigorta şirketi için müşteri verisi dosyasının basit bir örneği Şekil 1.7'de gösterilmiştir. (Daha sonra, Şekil 1.7'de gösterilen dosya yapısının, erken dosya sistemlerinde genellikle bulunan bir yapı olmasına rağmen, bir veritabanı için tatmin edici olmadığını keşfedeceksiniz.)

Bilgisayar dosyalarının tanımlanması, özel bir kelime dağarcığı gerektirir. Her disiplin, uygulayıcılarının net bir şekilde iletişim kurabilmesi için kendi terminolojisini geliştirir. Tablo 1.2'de gösterilen temel dosya kelime dağarcığı, sonraki tartışmaları daha kolay anlamanızı sağlayacaktır.

Figure 1.7 Contents of the Customer File

Database name: Ch01_Text								
C_NAME	C_PHONE	C_ADDRESS	C_ZIP	A_NAME	A_PHONE	TP	AMT	REN
Alfred A. Ramas	615-844-2573	218 Fork Rd., Babs, TN	36123	Leah F. Hahn	615-882-1244	T1	100.00	05-Apr-2022
Leona K. Dunne	713-894-1238	Box 12A, Fox, KY	25246	Alex B. Alby	713-228-1249	T1	250.00	16-Jun-2022
Kathy W. Smith	615-894-2285	125 Oak Ln., Babs, TN	36123	Leah F. Hahn	615-882-2144	S2	150.00	29-Jan-2023
Paul F. Olowski	615-894-2180	217 Lee Ln., Babs, TN	36123	Leah F. Hahn	615-882-1244	S1	300.00	14-Oct-2022
Myron Orlando	615-222-1672	Box 111, New, TN	36155	Alex B. Alby	713-228-1249	T1	100.00	28-Dec-2022
Amy B. O'Brian	713-442-3381	387 Troll Dr., Fox, KY	25246	John T. Okon	615-123-5589	T2	850.00	22-Sep-2022
James G. Brown	615-297-1228	21 Tye Rd., Nash, TN	37118	Leah F. Hahn	615-882-1244	S1	120.00	25-Mar-2023
George Williams	615-290-2558	155 Maple, Nash, TN	37119	John T. Okon	615-123-5589	S1	250.00	17-Jul-2022
Anne G. Farriss	713-382-7185	2119 Elm, Crew, KY	25432	Alex B. Alby	713-228-1249	T2	100.00	03-Dec-2022
Olette K. Smith	615-297-3809	2782 Main, Nash, TN	37118	John T. Okon	615-123-5589	S2	500.00	14-Mar-2023

C_NAME = Customer name
C_PHONE = Customer phone
C_ADDRESS = Customer address
C_ZIP = Customer zip code

A_NAME = Agent name
A_PHONE = Agent phone
TP = Insurance type
AMT = Insurance policy amount, in thousands of \$
REN = Insurance renewal date

Table 1.2 Basic File Terminology

Term	Definition
Data	Ham veriler, orneğin bir telefon numarası, doğum tarihi, müşteri adı ve yılbaşından bugüne kadar (YTD) satış değeri gibi bilgileri ifade eder. Veri, mantıklı bir şekilde organize edilmedikçe pek bir anlam taşımaz.
Field	Belirli bir anlamı olan bir karakter veya karakter grubu (alfabetik veya sayısal). Bir alan, verileri tanımlamak ve saklamak için kullanılır.
Record	Mantıksal olarak bağlantılı bir veya daha fazla alandan oluşan ve bir kişi, yer veya şeyi tanımlayan bir veri kumesidir. Örneğin, bir müşteri için oluşturulan bir kaydın alanları, müşterinin adı, adresi, telefon numarası, doğum tarihi, kredi limiti ve ödenmemiş bakiyesi gibi bilgileri içerebilir.
File	İlişkili kayıtların bir koleksiyonudur. Örneğin, bir dosya, Gigantic Üniversitesi'ne şu anda kayıtlı öğrencilerle ilgili verileri içerebilir.

Tablo 1.2'deki doğru dosya terminolojisini kullanarak, Şekil 1.7'de gösterilen dosya bileşenlerini tanımlayabilirsiniz. CUSTOMER dosyası 10 kayıttan oluşmaktadır. Her bir kayıt, dokuz alandan oluşur: C_NAME, C_PHONE, C_ADDRESS, C_ZIP, A_NAME, A_PHONE, TP, AMT ve REN. 10 kayıt, adlandırılmış bir dosyada saklanır. Şekil 1.7'deki dosya, sigorta şirketinin müşteri verilerini içerdiğinden, dosya adı CUSTOMER'dır.

İş kullanıcıları bilgisayarlı dosyadan veri istediğinde, DP uzmanının verileri dosya(lar)dan almak, kullanıcının istediği şekilde değiştirmek ve basılı bir rapor olarak sunmak için programlar oluşturması gerekiyordu. Örneğin, sigorta şirketindeki Satış departmanı, günlük satış çabalarının izlenmesine yardımcı olan SALES adlı bir dosya oluşturdu. Ardından, DP uzmanı Personel departmanı bordro işlemlerini ve diğer personel işlevlerini otomatikleştirmek için Şekil 1.8'de gösterilen AGENT dosyasını oluşturdu. AGENT dosyasındaki veriler, diğer görevlerin yanı sıra Personel departmanının çek yazmasına, ödenen vergileri takip etmesine ve sigorta kapsamını özetlemesine olanak sağladı.

Çevrimiçi İçerik

Her bölümde kullanılan veritabanları www.cengage.com adresinde mevcuttur. Kitap boyunca, Çevrimiçi İçerik kutuları, bölüm içeriğiyle ilgili web sitesindeki materyalleri vurgulamaktadır.

Figure 1.8 Contents of the Agent File

A_NAME	A_PHONE	A_ADDRESS	ZIP	HIRED	YTD_PAY	YTD_FIT	YTD_FICA	YTD_SLS	DEP
Alex B. Alby	713-228-1249	123 Toll, Nash, TN	37119	01-Nov-2004	26566.24	6641.56	2125.30	132737.75	3
Leah F. Hahn	615-882-1244	334 Main, Fox, KY	25246	23-May-1990	32213.78	8053.44	2577.10	138967.35	0
John T. Okon	615-123-5589	452 Elm, New, TN	36155	15-Jun-2009	23198.29	5799.57	1855.86	127093.45	2

A_NAME = Agent name
A_PHONE = Agent phone
A_ADDRESS = Agent address
ZIP = Agent zip code
HIRED = Agent date of hire

YTD_PAY = Year-to-date pay YTD_FIT = Year-to-date federal income tax paid
YTD_FICA = Year-to-date Social Security taxes paid
YTD_SLS = Year-to-date sales
DEP = Number of dependents

Daha fazla bilgisayarlaştırılmış dosya geliştirildikçe, bu tür dosya sistemindeki sorunlar belirgin hale geldi. Sorunlar, ilişkili ve çoğu zaman örtüşen veriler içeren birçok veri dosyasının olması ve tüm dosyalar arasında veriyi tutarlı bir şekilde kontrol etme veya yönetme aracının olmaması etrafında yoğunlaşmıştır. Sistemdeki her dosya, veriyi depolamak, almak ve değiştirmek için kendi uygulama programını kullanıyordu.

Bilgisayar dosyalarının şirket verilerini depolamak için ortaya çıkışı önemli bir gelişmeydi; bu, bilgisayar teknolojilerinin kullanımında bir dönüm noktası oluşturdu ve bir işletmenin veri işleme yeteneğinde önemli bir adım ileriye doğru bir ilerlemeyi temsil etti. Daha önce, kullanıcılar tüm iş verilerine doğrudan, elle erişim sağlıyordu. Ancak, bu veriyi ihtiyaç duydukları bilgilere dönüştürmek için araçları yoktu. Bilgisayarlaştırılmış dosya sistemlerinin oluşturulması, onlara şirket verilerini manipüle etmek ve yeni bilgiler oluşturmak için geliştirilmiş araçlar sağladı. Bununla birlikte, bu, son kullanıcılar ile verileri arasındaki bir uçurumun ortaya çıkmasına da neden oldu. Son kullanıcılar ile veriler arasındaki bu uçurumu kapatma isteği, birçok bilgisayar teknolojisinin, sistem tasarımlarının ve birçok teknolojinin ve tekniğin kullanım (ve kötü kullanım) biçimlerinin gelişimini etkiledi. Ancak, bu tür gelişmeler, DP uzmanları ile son kullanıcıların veriyi görme biçimleri arasında bir ayrım yarattı.

- DP uzmanının perspektifinden bakıldığında, dosya sistemindeki bilgisayar dosyaları, manuel dosyalara benzer şekilde oluşturulmuştu. Veri yönetim programları, dosyaya veri eklemek, güncellemek ve silmek için yaratılmıştı.
- Son kullanıcıların perspektifinden bakıldığında, sistemler kullanıcıları verilerden ayırıyordu. Kullanıcıların rekabetçi ortamı, daha kısa sürede daha fazla karar almalarını zorunlu kıldıkça, kullanıcılar, verilerden yeni bilgiler oluşturmak için düşündükleri bir yöntemi hayata geçirme ile DP uzmanının o bilgiyi oluşturacak programları gerçekten yazması arasındaki gecikmeden dolayı hayal kırıklığına uğramışlardı.

1-5c Dosya Sistemi Redux: Modern Son Kullanıcı Üretkenlik Aracı

Kullanıcıların veriye doğrudan, elle erişim sağlama isteği, kişisel bilgisayarların iş kullanımı için benimsenmesini hızlandırdı. Dosya sistemi evrimiyle doğrudan ilgili olmasa da, kişisel verimlilik araçlarının yaygın kullanımı, eski dosya sistemleriyle aynı sorunları ortaya çıkarabilir.

İş dünyasındaki kullanıcılar, Microsoft Excel gibi kişisel bilgisayar elektronik tablo programlarını, bir dizi satır ve sütuna veri girmek ve çok çeşitli işlevler kullanarak verileri manipüle etmek için yaygın olarak kullanmaktadır. Hesap tablosu uygulamalarının popülerliği, kullanıcıların verileri anlama ve daha iyi kararlar alma becerilerini büyük ölçüde geliştiren karmaşık veri analizi yapmalarını sağlamıştır. Ne yazık ki, eski bir atasözünde olduğu gibi “Sahip olduğunuz tek alet çekiçse, her sorun çivi gibi görünür”, kullanıcılar elektronik tablolarla çalışma konusunda o kadar ustalaşmıştır ki, elektronik tabloların uygun olmadığı görevleri tamamlamak için bunları kullanma eğilimindedirler.

Elektronik tabloların yaygın kötü kullanımlarından biri, veritabanının yerine kullanılmalarıdır. İlginç bir şekilde, son kullanıcılar genellikle doğrudan erişebildikleri sınırlı verileri alıp, geleneksel manuel veri depolama sistemlerine benzer bir elektronik tablo formatına yerleştirirler ki bu, erken DP uzmanlarının bilgisayarlaştırılmış veri dosyaları oluştururken yaptıkları şeydir. Elektronik tablo kullanan çok sayıda kullanıcı olduğu için, her biri verinin ayrı bir kopyasını oluşturduğunda, ortaya çıkan "dosya sistemi" aynı sorunlarla karşılaşır; bu sorunlar, erken DP uzmanları tarafından oluşturulan dosya sistemlerinde de vardı ve bir sonraki bölümde ele alınacaktır.

1-6 Dosya Sistemi Veri İşleme Problemleri

Dosya sistemi yönteminin veri organize etme ve yönetme şekli, manuel sisteme kıyasla kesin bir iyileşme sağlamıştı ve dosya sistemi, veri yönetimi için yirmi yılı aşkın bir süre boyunca — bilgisayar dönemi için oldukça uzun bir süre — faydalı bir amaç taşıdı. Bununla birlikte, bu yaklaşımda pek çok sorun ve sınırlama ortaya çıktı. Dosya sistemi yönteminin eleştirisi iki ana amaca hizmet eder:

- Dosya sistemi yönteminin eksikliklerini anlamak, modern veritabanlarının gelişimini anlamamızı sağlar.
- Bu tür sorunları anlamamak, veritabanı ortamında bu sorunların tekrarlanmasına yol açabilir, oysa veritabanı teknolojisi bunlardan kaçınmayı kolaylaştırır.

Dosya sistemleriyle ilişkili aşağıdaki sorunlar, verilerden oluşturulabilecek bilgi türlerini ve bilginin doğruluğunu ciddi şekilde zorluyor:

- **Uzun geliştirme süreleri.** Dosya sistemi yaklaşımının ilk ve en belirgin sorunu, en basit veri alma işleminin bile kapsamlı programlamalar gerektirmesidir. Eski dosya sistemlerinde, programcılar ne yapılması gerektiğini ve nasıl yapılacağını belirtmek zorundaydılar. Gelecek bölümlerde öğreneceğiniz gibi, modern veritabanları, kullanıcının ne yapılması gerektiğini belirlemesine olanak tanıyan ve nasıl yapılacağını belirtmeye gerek bırakmayan prosedürsüz bir veri manipülasyon dili kullanır.
- **Hızlı cevap almakta zorluk.** En basit raporları oluşturmak için bile program yazma ihtiyacı, ad hoc sorguları imkansız hale getirir. Olgun dosya sistemleriyle çalışan yoğun DP uzmanları, genellikle yeni raporlar için birçok talep alırlardı. Çoğu zaman, raporun "gelecek hafta" veya hatta "gelecek ay" hazır olacağını söylemek zorunda kalırlardı. Eğer bilgiyi hemen ihtiyacınız varsa, gelecek hafta ya da gelecek ay alacağınız bilgi, ihtiyaçlarınızı karşılamaz.
- **Karmaşık sistem yönetimi.** Sistemdeki dosya sayısı arttıkça, sistem yönetimi daha da zorlaşır. Az sayıda dosyaya sahip basit bir dosya sistemi bile, çeşitli dosya yönetim programlarının oluşturulmasını ve bakımını gerektirir. Her dosyanın, kullanıcının kayıt eklemesine, değiştirmesine ve silmesine olanak tanıyan kendi dosya yönetim programlarına sahip olması gerekir.

Dosya içeriğini listelemek ve raporlar oluşturmak için de dosya yönetim programları gereklidir. Ad hoc sorguların mümkün olmaması nedeniyle, dosya raporlama programlarının sayısı hızla artabilir. Sorun, her departmanın kendi dosyalarını oluşturarak verilerini "sahiplenmesi" gerçeğiyle daha da karmaşık hale gelir.

Güvenlik eksikliği ve sınırlı veri paylaşımı. Dosya sistemi veri deposunun bir diğer kusuru, güvenlik eksikliği ve sınırlı veri paylaşımıdır. Veri paylaşımı ve güvenlik yakından ilişkilidir. Verilerin coğrafi olarak dağılmış birden fazla kullanıcı arasında paylaşılması, birçok güvenlik riskini beraberinde getirir. Elektronik tablo verileri açısından bakıldığında, birçok elektronik tablo programı temel güvenlik seçenekleri sunsa da, bunlar her zaman kullanılmaz ve kullanıldığında bile kullanıcılar arasında güçlü bir veri paylaşımı için yeterli değildir. Veri yönetimi ve raporlama programları oluşturma açısından güvenlik ve veri paylaşımı özellikleri programlaması zor olduğundan, genellikle dosya sistemi ortamında göz ardı edilir. Bu özellikler arasında etkili parola koruması, dosyaların veya sistemin belirli bölümlerinin kilitlenmesi ve veri gizliliğini korumaya yönelik diğer önlemler bulunur. Sistem ve veri güvenliğini artırma girişiminde bulunulduğunda bile, güvenlik önlemleri genellikle sınırlı kapsamda ve etkinlikte kalır.

Kapsamlı programlama. Mevcut bir dosya yapısında değişiklik yapmak, dosya sistemi ortamında zor olabilir. Örneğin, orijinal CUSTOMER dosyasındaki tek bir alanı değiştirmek için bir programın aşağıdaki işlemleri gerçekleştirmesi gerekir:

1. orijinal dosyadan bir kaydı okur,
2. orijinal veriyi, yeni yapının depolama gereksinimlerine uyacak şekilde dönüştürür,
3. dönüştürülmüş veriyi yeni dosya yapısına yazar ve
4. orijinal dosyadaki her kayıt için önceki adımları tekrarlar.

Aslında, dosya yapısında yapılan en küçük bir değişiklik bile, o dosyadaki verileri kullanan tüm programlarda değişiklik yapılmasını gerektirir. Bu değişiklikler hata (bug) oluşturma olasılığı taşır ve bu hataları bulmak için ek zaman harcanarak hata ayıklama süreci uygulanır. Bu sınırlamalar ise yapısal ve veri bağımlılığı sorunlarına yol açar.

1-6a Yapısal ve Veri Bağımlılığı

Bir dosya sistemi yapısal bağımlılık sergiler; bu da bir dosyaya erişimin dosyanın yapısına bağlı olduğu anlamına gelir. Örneğin, Şekil 1.7'de gösterilen CUSTOMER dosyasına bir müşteri doğum tarihi alanı eklemek, önceki bölümde açıklanan dört adımı gerektirecektir. Bu değişiklik göz önüne alındığında, önceki programların hiçbirisi yeni CUSTOMER dosya yapısıyla çalışmayacaktır. Bu nedenle, tüm dosya sistemi programları yeni dosya yapısına uyacak şekilde değiştirilmelidir. Kısacası, dosya sistemi uygulama programları dosya yapısındaki değişikliklerden etkilendikleri için yapısal bağımlılık sergilerler. Tersine, uygulamanın verilere erişim yeteneğini etkilemeden dosya yapısını değiştirebildiğinizde yapısal bağımsızlık söz konusudur.

Veri özelliklerinde yapılan değişiklikler, örneğin bir alanın tam sayıdan ondalıklı sayıya değiştirilmesi, dosyaya erişen tüm programlarda değişiklik yapılmasını gerektirir. Çünkü dosyanın veri depolama özelliklerinden herhangi biri değiştiğinde (örneğin, veri türünü değiştirmek), tüm veri erişim programlarının değişime uğraması gerekir. Bu nedenle, dosya sistemi veri bağımlılığı sergiler. Tersine, veri bağımsızlığı, veri depolama özelliklerini değiştirebilmeniz ve bunun programın veriye erişim yeteneğini etkilememesini sağlamanız anlamına gelir.

Veri bağımlılığının pratik önemi, mantıksal veri formatı (veriyi insanın nasıl gördüğü) ile fiziksel veri formatı (bilgisayarın verilerle nasıl çalışması gerektiği) arasındaki farktır. Bir dosya sisteminin dosyasına erişen herhangi bir program, bilgisayara sadece ne yapması gerektiğini söylemekle kalmaz, aynı zamanda nasıl yapması gerektiğini de belirtmek zorundadır. Sonuç olarak, her program, belirli bir dosya türünün açılmasını, kayıt tanımını ve alan tanımlarını belirten satırlara sahip olmalıdır. Veri bağımlılığı, dosya sistemini bir programcı ve veritabanı yöneticisi açısından son derece zahmetli hale getirir.

structural dependence
Veritabanı şemasındaki bir değişikliğin veri erişimini etkilemesi ve bu nedenle tüm erişim programlarında değişiklik yapılmasını gerektiren bir veri özelliğidir.

structural independence
Veritabanı şemasındaki değişikliklerin veri erişimini etkilemediği bir veri özelliği.

data type
Kullanılabilecek veya saklanabilecek değer türlerini tanımlar. Ayrıca, programlama dillerinde ve veritabanı sistemlerinde bu tür verilere uygulanabilecek işlemleri belirlemek için kullanılır.

.data dependence
Veri temsili ve manipülasyonunun fiziksel veri depolama özelliklerine bağlı olduğu bir veri durumu.

data independence
Veri erişiminin fiziksel veri depolama özelliklerindeki değişikliklerden etkilenmediği bir durum.

logical data format
Bir kişinin bir sorun alanı bağlamında verileri görme biçimi.

physical data format
Bir bilgisayarın verileri "görme" (saklama) şekli.

islands of information

Eski dosya sistemi ortamında, farklı departmanlar tarafından oluşturulan ve yönetilen, bağımsız, sıklıkla çoğaltılmış ve tutarsız verilerin havuzları vardı.

data redundancy

Veri yinelenmesi, aynı verinin gereksiz bir şekilde farklı yerlerde depolandığı durumdur.

data integrity

Veritabanı bütünlüğü, bir ilişkisel veritabanında verilerin, tüm varlık ve referans bütünlüğü kısıtlamalarına uyduğu bir durumu ifade eder.

1-6b Veri Yedekliliği

Dosya sisteminin yapısı, birden fazla kaynaktan veri birleştirmeyi zorlaştırır ve güvenlik eksiklikleri, dosya sistemini güvenlik ihlallerine karşı savunmasız hale getirir. Organizasyon yapısı, aynı temel verilerin farklı yerlerde depolanmasını teşvik eder. (Veritabanı uzmanları, böyle dağılmış veri konumları için "bilgi adaları" terimini kullanır.) Verilerin dağılması, verilerin saklanması elektronik tabloların kullanılmasından daha da kötüleşir. Bir dosya sisteminde, tüm Satış departmanı, DP uzmanı tarafından oluşturulan veri yönetimi ve raporlama programları aracılığıyla SALES veri dosyasına erişebilir. Elektronik tablo kullanıldığında, Satış departmanındaki her birey kendi satış verisi kopyasını oluşturabilir. Farklı yerlerde depolanan verilerin tutarlı bir şekilde güncellenmemesi olasılığı olduğundan, bilgi adaları genellikle aynı verinin farklı versiyonlarını içerir. Örneğin, Şekil 1.7 ve 1.8'de, ajan adları ve telefon numaraları hem CUSTOMER hem de AGENT dosyalarında bulunur. Ajan adları ve telefon numaralarının sadece bir doğru kopyasına ihtiyacınız vardır. Bunların birden fazla yerde bulunması, veri yedekliliğine neden olur. Veri yedekliliği, aynı verilerin gereksiz bir şekilde farklı yerlerde depolanması durumudur.

Kontrolsüz veri yedekliliği, aşağıdaki durumların ortaya çıkmasına zemin hazırlar:

- **Kötü veri güvenliği.** Verinin birden fazla kopyasının olması, bir kopyanın yetkisiz erişime açık olma olasılığını artırır. Bölüm 16, Veritabanı Yönetimi ve Güvenliği, veriyi güvence altına alma ile ilgili sorunlar ve teknikler hakkında daha fazla bilgi sunmaktadır.
- **Veri tutarsızlığı.** Veri tutarsızlığı, aynı verinin farklı yerlerde farklı ve çelişkili sürümlerinin bulunduğu durumlarda ortaya çıkar. Örneğin, bir ajanının telefon numarasını AGENT dosyasına değiştirirseniz, ancak bu değişikliği CUSTOMER dosyasına yapmayı unutursanız, dosyalar aynı ajan için farklı veriler içerir. Raporlar, hangi veri sürümünün kullanıldığına bağlı olarak tutarsız sonuçlar verecektir.
- **Veri girişi hataları.** Veri girişi hatalarının oluşma olasılığı, karmaşık girişlerin (örneğin, 10 haneli telefon numaraları) birkaç farklı dosyada yapılması veya bir veya daha fazla dosyada sıkça tekrarlanması durumunda daha yüksektir. Aslında, Şekil 1.7'de gösterilen CUSTOMER dosyasında böyle bir giriş hatası bulunmaktadır: CUSTOMER dosyasındaki üçüncü kayıttaki ajan telefon numarasındaki rakamlar yer değiştirmiştir (615-882-2144 yerine 615-882-1244).
- **Veri bütünlüğü problemleri.** Mevcut olmayan bir satış ajanının adı ve telefon numarası, CUSTOMER dosyasına girilebilir, ancak müşteriler, sigorta ajansının var olmayan bir ajanın adını ve telefon numarasını verdiğinde pek etkilenmezler. Personel yöneticisi, var olmayan bir ajanın bonus ve faydaları biriktirmesine izin vermeli midir? Aslında, yanlış yazılmış bir isim veya yanlış bir telefon numarası gibi bir veri girişi hatası da aynı tür veri bütünlüğü problemlerine yol açar.

Note

Veri tutarsızlığını gösteren veriler, aynı zamanda veri bütünlüğü eksik olan veriler olarak da adlandırılır. Veri bütünlüğü, veritabanındaki verilerin gerçek dünya olayları ve koşullarıyla tutarlı olduğu durum olarak tanımlanır. Başka bir deyişle, veri bütünlüğü şu anlama gelir:

- Veri doğru ve tutarlıdır—veri tutarsızlıkları yoktur.
- Veri doğrulanabilir—veri her zaman tutarlı sonuçlar verir.