

DBA, sistemdeki haydut kullanıcıların etkisini en aza indirmek için veritabanı rollerini ve kaynak sınırlarını kullanabilir. (Bu konular hakkında daha fazla bilgi için Bölüm 16-10'd'ye bakın).

• *Erişim ayrıcalıkları atayın.* DBA, belirli veritabanlarına erişimleri için belirli kullanıcılara erişim ayrıcalıkları atar. Erişim hakları yalnızca okuma ile sınırlı olabilir veya yetkili erişim okuma, yazma ve silme ayrıcalıklarını içerebilir. İlişkisel veritabanlarında erişim ayrıcalıkları SQL GRANT ve REVOKE komutları aracılığıyla atanır.

## Not

GRANT ve REVOKE komutları SQL'de ayrıcalık ve kullanıcı belirtilerek uygulanır. Örneğin, MJORDAN kullanıcısına PRODUCT tablosunda güncelleme ayrıcalıkları vermek aşağıdaki komutla yapılır:

**MJORDAN'A ÜRÜN GÜNCELLEMESİ VERİN;**

PRODUCT tablosunda MJORDAN'dan güncelleme ayrıcalığını kaldırmak için aşağıdaki komutu kullanın:

**MJORDAN'DAN GELEN ÜRÜN GÜNCELLEMESİNİ IPTAL EDİN;**

• *Fiziksel erişimi kontrol edin.* Fiziksel güvenlik yetkisiz kullanıcıların DBMS kurulumuna ve tesislerine doğrudan erişimini engelleyebilir. Büyük veri tabanı kurulumları için yaygın fiziksel güvenlik, güvenli girişler, parola korumalı iş istasyonları, elektronik personel rozetleri, kapalı devre video, ses tanıma ve biyometrik teknolojiyi içerir.

- *Görünüm tanımla.* DBA, yetkili bir kullanıcı tarafından erişilebilen verilerin kapsamını korumak ve kontrol etmek için veri görünümünü tanımlamalıdır. DBMS, bir veya daha fazla tablodan oluşan görünümün tanımlanmasına izin veren araçlar sağlamalı ve kullanıcılara erişim hakları atamalıdır. İlişkisel veritabanlarında görünümünü tanımlamak için SQL CREATE VIEW komutu kullanılır. Oracle DBMS, DBA'nın farklı kullanıcılar için verilerin özelleştirilmiş görünümünü oluşturmasına olanak tanıyan Sanal Özel Veritabanı (VPD) sunar. Bu özellik sayesinde DBA, bir bordro veritabanını sorgulayan normal kullanıcıların yalnızca gerekli satır ve sütunları görmesini kısıtlayabilirken, departman yöneticileri yalnızca kendi departmanlarıyla ilgili satır ve sütunları görebilir.
- *DBMS erişim kontrolü.* Veritabanı erişimi, DBMS sorgu ve raporlama araçlarının kullanımına sınırlar getirilerek kontrol edilebilir. DBA, araçların doğru şekilde ve yalnızca yetkili personel tarafından kullanıldığından emin olmalıdır.
- *VTYS kullanım izleme.* DBA ayrıca veritabanındaki verilerin kullanımını da denetlemelidir. Birçok DBMS paketi, tüm kullanıcılar tarafından gerçekleştirilen veritabanı işlemlerinin kısa bir açıklamasını otomatik olarak kaydeden bir **denetim günlüğü**nün oluşturulmasına izin veren özellikler içerir. Bu tür denetim izleri DBA'nın erişim ihlallerini saptamasını sağlar. Denetim kayıtları tüm veritabanı erişimlerini ya da sadece başarısız olanları kaydedecek şekilde uyarlanabilir.

Bir veritabanının bütünlüğü, DBA'nın kontrolü dışındaki dış etkenler nedeniyle kaybolabilir. Örneğin, veritabanı bir patlama, yangın ya da deprem nedeniyle hasar görebilir ya da yok olabilir. Sebep ne olursa olsun, veritabanının bozulması ya da yok olması ihtimali, yedekleme ve kurtarma prosedürlerini her DBA için çok önemli hale getirir.

## 16-7 Veritabanı Yönetim Araçları

Kurumlardaki veri yönetimi faaliyetlerinin olağanüstü büyümesi, daha iyi yönetim standartlarına, süreçlerine ve araçlarına ihtiyaç duyulmasına neden oldu. Yıllar içinde, yalnızca veri yönetim araçlarına adanmış yeni bir endüstri ortaya çıktı. Bu araçlar, veri yönetiminin tüm spektrumunu kapsamaktadır.

### denetim günlüğü

Tüm kullanıcılar tarafından gerçekleştirilen veritabanı işlemlerinin kısa bir açıklamasını otomatik olarak kaydeden bir veritabanı yönetim sisteminin güvenlik özelliği.

Seçimden başlangıca, dağıtım, geçiş ve günlük işlemlere kadar veri yönetimi görevleri. Örneğin, aşağıdakiler için gelişmiş veri yönetimi araçları bulabilirsiniz:

- Veritabanı izleme
- Veritabanı yük testi
- Veritabanı performans ayarı
- SQL kodu optimizasyonu
- Veritabanı darboğazlarının belirlenmesi ve giderilmesi
- Veritabanı modelleme ve tasarımı
- Veritabanı veri çıkarma, dönüştürme ve yükleme

Tüm bu veri yönetim araçlarının ortak bir noktası vardır. Hepsi veri tabanının meta verilerini veya veri sözlüğünü genişletir. Bir DBA aracı olarak veri sözlüğünün önemi abartılamaz. Bu bölümde, bir veri yönetim aracı olarak veri sözlüğünün yanı sıra DBA'nın veritabanı analizi ve tasarımını desteklemek için bilgisayar destekli sistem mühendisliği (CASE) araçlarını kullanımı incelenmektedir.

## 16-7a Veri Sözlüğü

Bölüm 1'de *veri sözlüğü* "veri özelliklerinin ve ilişkilerinin tanımını saklayan bir VTYS bileşeni" olarak tanımlanmıştı. Bu tür "veri hakkındaki verilere" *meta veri* denildiğini hatırlayabilirsiniz. VTYS veri sözlüğü, VTYS'ye kendi kendini tanımlayan karakteristiğini sağlar. Aslında veri sözlüğü şirketin tüm veri setinin bir röntgenini andırır ve veri yönetiminde çok önemli bir unsurdur.

İki ana veri sözlüğü türü mevcuttur: *entegre* ve *bağımsız*. Entegre bir veri sözlüğü DBMS ile birlikte verilir. Örneğin, tüm ilişkisel DBMS'ler, RDBMS tarafından sık sık erişilen ve güncellenen yerleşik bir veri sözlüğü veya sistem kataloğu içerir. Diğer DBMS'ler, özellikle de eski tipler, yerleşik bir veri sözlüğüne sahip değildir; bunun yerine, DBA üçüncü taraf *bağımsız* sistemleri kullanabilir.

Veri sözlükleri *aktif* veya *pasif* olarak da sınıflandırılabilir. **Aktif** bir **veri sözlüğü**, erişim bilgilerini güncel tutmak için her veritabanı erişiminde DBMS tarafından otomatik olarak güncellenir. **Pasif** bir **veri sözlüğü** otomatik olarak güncellenmez ve genellikle bir toplu işlemin çalıştırılmasını gerektirir. Veri sözlüğü erişim bilgileri normalde DBMS tarafından sorgu optimizasyonu için kullanılır.

Veri sözlüğünün ana işlevi, veritabanı ile etkileşime giren tüm nesnelerin tanımını saklamaktır. Entegre veri sözlükleri, meta verilerini DBMS tarafından yönetilen verilerle sınırlama eğilimindedir. Bağımsız veri sözlüğü sistemleri genellikle daha esnektir ve DBA'nın bilgisayarda olsun ya da olmasın kuruluşun tüm verilerini tanımlamasına ve yönetmesine olanak tanır. Veri sözlüğünün formatı ne olursa olsun, veri tabanı tasarımcılarına ve son kullanıcılara çok daha gelişmiş bir iletişim yeteneği sağlar. Ayrıca veri sözlüğü, DBA'nın veri çakışmalarını çözmesine yardımcı olan bir araçtır.

Veri sözlüğünde saklanan bilgiler için standart bir format olmamasına rağmen, birkaç özellik ortaktır. Örneğin, veri sözlüğü tipik olarak aşağıdakilerin açıklamalarını depolar:

- *Tüm veritabanılarının tüm tablolarında tanımlanan veri öğeleri.* Veri sözlüğü özellikle öğe adlarını, veri türlerini, görüntüleme biçimini, dahili depolama biçimini ve doğrulama kurallarını depolar. Veri sözlüğü, bir öğenin nerede kullanıldığını, kimin kullandığını vb. açıklar.
- *Tüm veritabanılarında tanımlanan tablolar.* Örneğin, veri sözlüğünde tablo oluşturucunun adı, oluşturulma tarihi, erişim yetkileri ve sütun sayısı saklanabilir.
- *Her veritabanı tablosu için tanımlanmış dizinler.* Her bir dizin için DBMS en azından dizin adını, kullanılan öznitelikleri, konumu, belirli dizin özelliklerini ve oluşturma tarihini saklar.

### aktif veri sözlüğü

Veritabanı yönetim sistemi tarafından her seferinde otomatik olarak güncellenen bir veri sözlüğü. veritabanına erişilir, böylece bilgiler güncel tutulur.

**pasif veri sözlüğü** Veri erişim istatistiklerini güncellemek için bir son kullanıcı tarafından başlatılan bir komut gerektiren bir DBMS veri sözlüğü.

- *Tanımlanmış veritabanları.* Bu bilgiler, her bir veritabanının kimin oluşturduğunu, veri tabanının ne zaman oluşturulduğunu, veritabanının nerede bulunduğunu ve DBA'nın adını içerir.
- *Veritabanının son kullanıcıları ve yöneticileri.* Bu bilgiler veritabanının kullanıcılarını tanımlar.
- *Veritabanına erişen programlar.* Bu bilgiler ekran formatlarını, rapor matlarını, uygulama programlarını ve SQL sorgularını içerir.
- *Tüm veritabanlarının tüm kullanıcıları için erişim yetkileri.* Bu bilgiler, kimin hangi nesneleri manipüle edebileceğini ve ne tür işlemlerin gerçekleştirilebileceğini tanımlar.
- *Veri elemanları arasındaki ilişkiler.* Bu bilgiler, hangi unsurların dahil olduğunu, ilişkilerin zorunlu veya isteğe bağlı olup olmadığını ve bağlantı ve kardinalite gereksinimlerini içerir.

Veri sözlüğü VTYS'nin dışındaki verileri de içerecek şekilde düzenlenebilirse, daha genel kurumsal kaynak yönetimi için özellikle esnek bir araç haline gelir. Böylesine kapsamlı bir veri sözlüğü, köklerinin veritabanı verilerinde olup olmadığına bakılmaksızın, kuruluşun tüm bilgilerinin kullanımını ve tahsisini yönetmeyi mümkün kılar. Bu nedenle bazı yöneticiler veri **sözlüğünü** bilgi kaynakları yönetiminin kilit bir unsuru olarak görürler, bu nedenle veri **sözlüğü bilgi kaynakları sözlüğü** olarak da tanımlanabilir.

Veri sözlüğünde saklanan meta veriler genellikle veritabanı kullanımını izlemek ve veritabanı kullanıcılarına erişim hakları atamak için temel oluşturur. Veri sözlüğünde saklanan bilgiler genellikle ilişkisel tablo formatına dayanır, böylece DBA'nın SQL komutlarıyla veritabanını sorgulamasına olanak tanır. Örneğin, SQL komutları belirli bir tablonun kullanıcıları veya belirli bir kullanıcının erişim hakları hakkında bilgi almak için kullanılabilir. Aşağıdaki bölümde, IBM DB2 sistem kataloğu tabloları, bir veri sözlüğünün bilgi üretmek için nasıl kullanıldığına ilişkin çeşitli örneklerin temelini oluşturmaktadır:

- SYSTABLES her tablo veya görünüm için bir satır saklar.
- SYSCOLUMNS her tablo veya görünümün her sütunu için bir satır saklar.
- SYSTABAUTH, bir veritabanındaki bir tablo veya görünüm için bir kullanıcıya verilen her yetki için bir satır saklar.

## Veri Sözlüğü Kullanım Örnekleri

### Örnek 1

Geçerli veritabanında JONESVI kullanıcısı tarafından oluşturulan tüm tabloların adlarını ve oluşturulma tarihlerini listeleyin.

```
SELECT NAME, CTIME FROM
SYSTABLES
WHERE CREATOR= 'JONESVI';
```

### Örnek 2

Geçerli veritabanında JONESVI tarafından oluşturulan tüm tablolar için sütunların adlarını

```
listeleyin. SELECT ADI
SYSCOLUMNS'DAN
WHERE TBcreator= 'JONESVI';
```

### Örnek 3

JONESVI kullanıcısının DELETE yetkisine sahip olduğu tüm tabloların adlarını listeleyin.

```
SELECT TTNAME
SYSTABAUTH'DAN
WHERE GRANTEE= 'JONESVI' AND DELETEAUTH= 'Y';
```

### bilgi kaynağı sözlüğü

Veri sözlüğü için başka bir isim.

## Örnek 4

INVENTORY tablosu üzerinde bir tür yetkiye sahip olan tüm kullanıcıların adlarını listeleyin.

```
SELECT DISTINCT GRANTEE
SYSTABAUTH'DAN
WHERE TTNAME= 'INVENTORY';
```

## Örnek 5

Veritabanındaki herhangi bir tablo için veritabanı yapısını değiştirebilen tüm kullanıcıların kullanıcı ve tablo adlarını listeleyin.

```
SELECT GRANTEE, TTNAME
FROM SYSTABAUTH
WHERE ALTERAUTH = 'Y' ORDER
BY GRANTEE, TTNAME;
```

Önceki örneklerde de görebileceğiniz gibi, veri sözlüğü, veri erişim ayrıcalıklarının atanmasını kontrol ederek veritabanı güvenliğini izlemek için bir araç olabilir. Önceki örneklerde veritabanı tabloları ve kullanıcıları hedef alınmış olsa da, veritabanına erişen uygulama programları hakkındaki bilgiler de veri sözlüğünden alınabilir.

DBA, veri analizini ve tasarımını desteklemek için veri sözlüğünü kullanabilir. Örneğin, DBA belirli bir uygulamada kullanılacak tüm veri öğelerini listeleyen bir rapor; belirli bir programa erişen tüm kullanıcıların bir listesi; veri fazlalıklarını, yinlemeleri ve eşanlamlıların ve eşanlamlıların kullanımını kontrol eden bir rapor; ve veri kullanıcılarını, veri erişimini ve veri yapısını tanımlayan bir dizi başka rapor oluşturabilir. Veri sözlüğü aynı zamanda uygulama programcılarının veritabanındaki veri öğeleri için adlandırma standartlarını karşıladığından ve veri doğrulama kurallarının doğru olduğundan emin olmak için de kullanılabilir. Böylece, veri sözlüğü çok çeşitli veri yönetimi faaliyetlerini desteklemek ve bilgi sistemlerinin tasarımını ve uygulanmasını kolaylaştırmak için kullanılabilir. Entegre veri sözlükleri, bilgisayar destekli sistem mühendisliği araçlarının kullanımı için de gereklidir.

## 16-7 b Kasa Araçları

## bilgisayar destekli sistem mühendisliği (CASE)

Sistem Geliştirme Yaşam Döngüsünün bir kısmını veya tamamını otomatikleştirmek için kullanılan araçlar.

## ön uç CASE aracı

SDLC'nin planlama, analiz ve tasarım aşamaları için destek sağlayan bilgisayar destekli bir yazılım aracı.

## arka uç CASE aracı

SDLC'nin kodlama ve uygulama aşamaları için destek sağlayan bilgisayar destekli bir yazılım aracı.

**CASE, bilgisayar destekli sistem mühendisliğinin** kısaltmasıdır. Bir CASE aracı, Sistem Geliştirme Yaşam Döngüsü (SDLC) için otomatik bir çerçeve sağlar. CASE yapılandırılmış metodolojiler ve güçlü grafik arayüzler kullanır. Birçok sıkıcı sistem tasarım ve uygulama faaliyetini otomatikleştirdikleri için CASE araçları bilgi sistemleri geliştirmede giderek daha önemli bir rol oynamaktadır.

CASE araçları genellikle SDLC için sağladıkları desteğin kapsamına göre sınıflandırılır. Örneğin, **ön uç CASE araçları** planlama, analiz ve tasarım aşamaları için destek sağlarken; **arka uç CASE araçları** kodlama ve uygulama aşamaları için destek sağlar. CASE araçları ile ilişkili faydalar şunları içerir:

- Geliştirme süresi ve maliyetlerinde azalma
- SDLC'nin Otomasyonu
- Sistem geliştirme metodolojilerinin standartlaştırılması
- CASE araçları ile geliştirilen uygulama sistemlerinin daha kolay bakımı

CASE araçlarının en önemli bileşenlerinden biri, sistem tasarımcısı tarafından oluşturulan tüm nesnelerin kaydını tutan kapsamlı bir veri sözlüğüdür. Örneğin, CASE veri sözlüğü veri akış diyagramlarını, yapı şemalarını, tüm harici ve dahili varlıkların açıklamalarını, veri depolarını, veri öğelerini, rapor formatlarını ve ekran formatlarını saklar. CASE veri sözlüğü ayrıca sistem bileşenleri arasındaki ilişkileri de tanımlar.

Bazı CASE araçları, DBMS ile çalışan ve CASE aracının DBMS'yi kullanarak veri sözlüğü bilgilerini depolamasına izin veren arayüzler sağlar. Bu tür bir etkileşim, sistem geliştirme ve veritabanı geliştirme arasındaki karşılıklı bağımlılığı gösterir ve tamamen entegre bir geliştirme ortamı yaratmaya yardımcı olur.

Bir CASE geliştirme ortamında, veritabanı ve uygulama tasarımcıları, veritabanı şemasının, veri öğelerinin, uygulama süreçlerinin, ekranların, raporların ve geliştirme ile ilgili diğer verilerin açıklamasını saklamak için CASE aracını kullanır. CASE aracı, tüm sistem geliştirme bilgilerini DBA'nın tutarlılık ve doğruluk açısından kontrol edebileceği ortak bir havuzda bütünleştirir.

Ek bir fayda olarak, CASE ortamı DBA, uygulama tasarımcıları ve son kullanıcılar arasındaki iletişimin kapsamını ve kalitesini artırma eğilimindedir. DBA, CASE aracını uygulamanın veri şemasının tanımını, adlandırma kurallarına uyulup uyulmadığını, veri öğelerinin tekrarlanıp tekrarlanmadığını, veri öğeleri için doğrulama kurallarını ve bir dizi başka gelişimsel ve yönetsel değişkeni kontrol etmek için kullanabilir. CASE aracı çakışmaları, kural ihlallerini ve tutarsızlıkları bulduğunda, düzeltmelerin yapılmasını kolaylaştırır. Daha da iyisi, CASE aracı bir düzeltme yapabilir ve ardından etkilerini uygulama ortamı boyunca kademelendirebilir, bu da DBA ve uygulama tasarımcısının işini basitleştirir.

Tipik bir CASE aracı beş bileşen sağlar:

- Veri akışı, ER diyagramları, sınıf diyagramları ve nesne diyagramları gibi yapılandırılmış diyagramlar üretmek için tasarlanmış grafikler
- Bilgi sisteminin girdi ve çıktı formatlarını (örneğin, son kullanıcı arayüzü) üretmek için ekran resimleri ve rapor oluşturmalar
- Sistem tasarım verilerinin depolanması ve çapraz referanslandırılması için entegre bir havuz; bu havuz kapsamlı bir veri sözlüğü içerir
- Sistem tutarlılığı, sözdizimi ve bütünlüğü üzerinde tam otomatik bir kontrol sağlamak için bir analiz segmenti
- Bir program dokümantasyonu oluşturucu

Şekil 16.7'de Microsoft Visio Professional'ın bir ER diyagramı oluşturmak için nasıl kullanılabileceği gösterilmektedir.

Çoğu CASE aracı, farklı soyutlama seviyelerinde görüntülenebilen tamamen belgelenmiş ER diyagramları üretir. Örneğin, Computer Associates tarafından geliştirilen erwin Data Modeler ayrıntılı ilişkisel tasarımlar üretebilir. Kullanıcı her bir varlık için öznitelikleri ve birincil anahtarları belirler ve ilişkileri tanımlar. Mevcut nesil veri modelleme araçları, varlıklar arasında belirtilen ilişkilere dayalı yabancı anahtarlar atar. Birincil anahtarlardaki değişiklikler sistem genelinde her zaman otomatik olarak güncellenir. Tablo 16.5, mevcut birçok CASE Veri Modelleme aracı tedarikçisinden bazılarını listelemektedir.

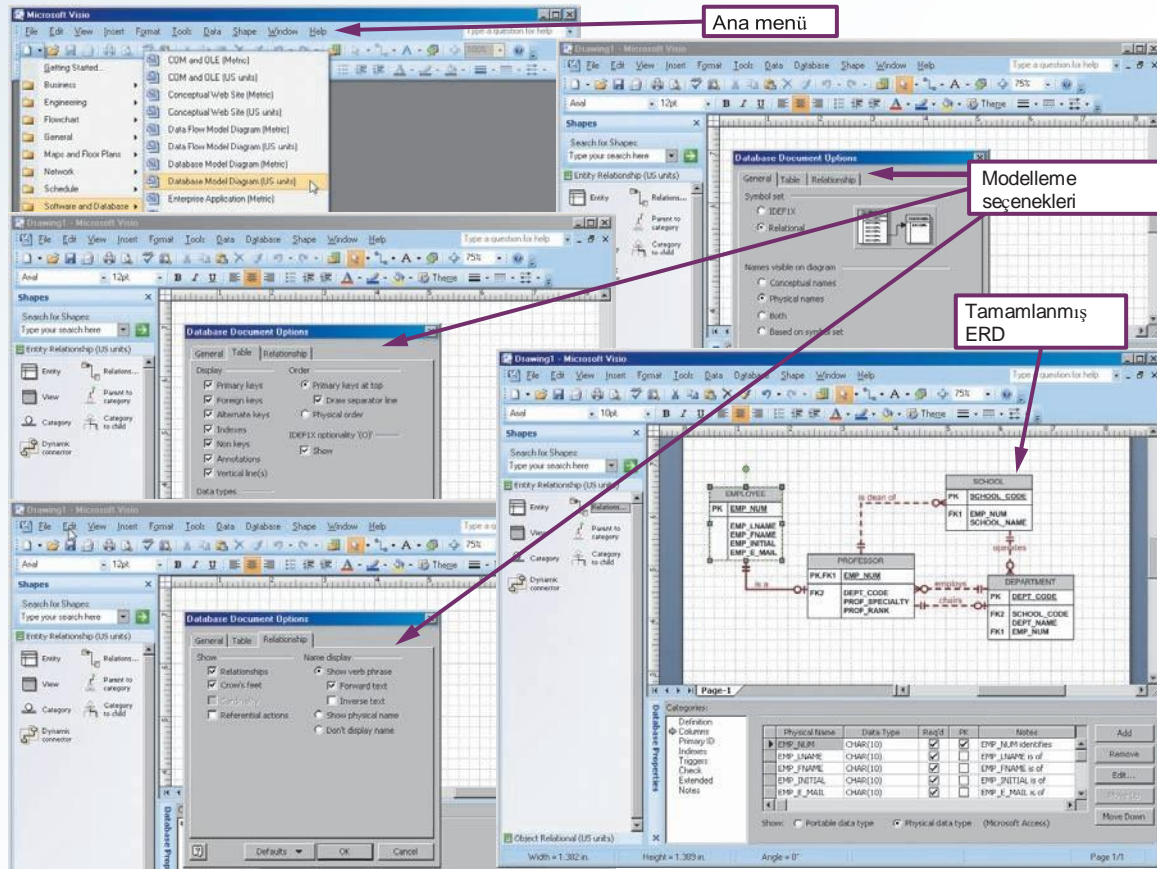
Oracle gibi büyük ilişkisel DBMS satıcıları artık kendi DBMS yazılımlarının yanı sıra diğer satıcılar tarafından sağlanan RDBMS'ler için de tam entegre CASE araçları sağlamaktadır. Örneğin,

Oracle'ın CASE araçları IBM'in DB2'si ve Microsoft'un SQL Server'ı ile birlikte kullanılarak tamamen belgelenmiş veritabanı tasarımları üretilmektedir. Hatta bazı satıcılar ilişkisel olmayan DBMS'leri alıp şemalarını geliştirmekte ve eşdeğer ilişkisel tasarımları otomatik olarak üretmektedir.

CASE araçlarının veritabanı tasarımcılarının ve uygulama programcılarının verimliliğini artırdığına şüphe yoktur. Ancak, CASE aracı ne kadar sofistike olursa olsun, kullanıcılarının kavramsal tasarım konusunda bilgili olması gerekir. Veritabanı acemilerinin elinde, CASE araçları etkileyici görünen ancak kötü tasarımlar üretir.



Şekil 16.7 Bir Vaka Aracı Örneği: Microsoft Visio Professional



Tablo 16.5 Vaka Veri Modelleme Araçları

Şirket	Ürün	Web sitesi
Erwin A.Ş.	erwin Veri Modelleyici	<a href="http://www.erwin.com">www.erwin.com</a>
Idera Inc.	ER/Studio Veri Mimarı	<a href="http://www.idera.com">www.idera.com</a>
Microsoft	Visio	<a href="http://office.microsoft.com/tr-us/visio">office.microsoft.com/tr-us/visio</a>
Oracle	SQL Geliştirici Veri Modelleyici	<a href="http://www.oracle.com/tools/downloads/sql-data-modeler-downloads.html">www.oracle.com/tools/downloads/sql-data-modeler-downloads.html</a>
IBM	Rasyonel Yazılım Mimarı Tasarımcısı	<a href="http://www.ibm.com/products/rational-software-architect-designer">www.ibm.com/products/rational-software-architect-designer</a>
SAP	Güç Tasarımcısı	<a href="http://www.sap.com/products/powerdesigner-data-modeling-tools.html">www.sap.com/products/powerdesigner-data-modeling-tools.html</a>
Görünür Sistemler	Görünür Analist	<a href="http://www.visible-systems.com">www.visible-systems.com</a>