

# DISEÑO DE BASES DE DATOS

# 13

## OBJETIVOS DE APRENDIZAJE

Una vez que haya dominado el material de este capítulo, podrá:

1. Entender los conceptos de las bases de datos.
2. Usar la normalización para almacenar eficazmente los datos en una base de datos.
3. Usar bases de datos para presentar datos.
4. Entender el concepto de almacenes de datos.
5. Comprender la utilidad de publicar bases de datos en la Web.

Algunos consideran que el almacenamiento de datos es el corazón de un sistema de información. Primero, los datos deben estar disponibles cuando el usuario desee utilizarlos. Segundo, los datos deben ser exactos y consistentes (deben tener integridad). Además de estos requerimientos, los objetivos del diseño de base de datos incluyen el almacenamiento eficaz de los datos así como su eficiente actualización y recuperación. Finalmente, es necesario que la recuperación de información tenga un propósito. La información obtenida de los datos almacenados debe estar en una forma que sirva para administrar, planear, controlar o tomar decisiones en una organización.

En un sistema basado en computadora hay dos enfoques para el almacenamiento de datos. El primero es almacenar los datos en archivos individuales, cada uno para una aplicación específica. El segundo enfoque implica la construcción de una base de datos. Una base de datos es un almacén de datos definido formalmente y controlado centralmente, con el propósito de usarse en muchas aplicaciones diferentes.

Los archivos convencionales seguirán siendo una forma práctica de almacenar datos para algunas aplicaciones (pero no para todas). Un archivo se puede diseñar y construir con bastante rapidez, y cualquier asunto acerca de la disponibilidad y seguridad de los datos se minimiza. Cuando los diseños de archivos se planean con cuidado, se puede incluir toda la información necesaria y se reducirá el riesgo de omitir involuntariamente datos.

El uso de archivos individuales tiene muchas consecuencias. A menudo los archivos se diseñan tomando en cuenta únicamente las necesidades inmediatas. Cuando se requiere consultar el sistema para obtener una combinación de algunos de los atributos, estos últimos podrían encontrarse en archivos separados o quizá ni siquiera existan. Con frecuencia, el rediseño de los archivos implica que también los programas que tienen acceso a ellos se deben rescribir, lo cual se traduce en tiempo de programación costoso para el desarrollo y mantenimiento de archivos y programas.

Un sistema que usa archivos convencionales implica que los datos almacenados serán redundantes. Además, la actualización de archivos requiere más tiempo. Finalmente, la integridad de los datos representa un problema, debido a que un cambio en un archivo también requerirá modificar los mismos datos en otros archivos.

## BASES DE DATOS

Las bases de datos no son tan sólo una colección de archivos. Más bien, una base de datos es una fuente central de datos destinados a compartirse entre muchos usuarios para una diversidad de aplicaciones. El corazón de una base de datos lo constituye el sistema de administración de base de datos (DBMS, *database management system*), el cual permite la creación, modificación y actualización de la base de datos, la recuperación de datos y la generación de informes y pantallas. La persona encargada de garantizar que la base de datos cumpla sus objetivos se conoce como administrador de base de datos.

Entre los objetivos de efectividad de la base de datos están los siguientes:

1. Asegurar que los datos se puedan compartir entre los usuarios para una diversidad de aplicaciones.
2. Mantener datos que sean exactos y consistentes.
3. Asegurar que todos los datos requeridos por las aplicaciones actuales y futuras se podrán acceder con facilidad.
4. Permitir a la base de datos evolucionar conforme aumenten las necesidades de los usuarios.
5. Permitir a los usuarios construir su vista personal de los datos sin preocuparse por la forma en que los datos se encuentren almacenados físicamente.

La anterior lista de objetivos nos proporciona un recordatorio de las ventajas y desventajas del enfoque de base de datos. Primero, la compartición de los datos significa que éstos deben almacenarse una sola vez. Como consecuencia, esto ayuda a lograr la integridad de los datos, debido a que los cambios en los datos se realizan con mayor facilidad y confiabilidad si éstos aparecen sólo una vez en lugar de en muchos archivos diferentes.

Cuando un usuario necesita datos específicos, una base de datos bien diseñada anticiparía dicha necesidad (o quizás ya se habrían usado en otra aplicación). Por lo tanto, es más probable que los datos estén disponibles en una base de datos que en un sistema de archivos convencional. Una base de datos bien diseñada también puede ser más flexible que los archivos separados; es decir, una base de datos puede evolucionar conforme cambien las necesidades de los usuarios y las aplicaciones.

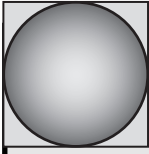
Finalmente, el enfoque de base de datos tiene la ventaja de permitir a los usuarios obtener su propia vista de los datos. Los usuarios no tienen que preocuparse por la estructura real de la base de datos o su almacenamiento físico.

Muchos usuarios extraen partes de la base de datos central desde mainframes y las descargan en sus PCs o en sus dispositivos portátiles. Después estas bases de datos más pequeñas se usan para generar informes o responder consultas específicas para el usuario final.

Las bases de datos relacionales para PCs se han perfeccionado de manera importante durante los últimos años. Un cambio tecnológico importante ha sido el diseño de software de base de datos que toma ventaja de la GUI. Con la llegada de programas tal como Microsoft Access, los usuarios pueden arrastrar y colocar campos entre dos o más tablas. Desarrollar bases de datos relacionales con estas herramientas es relativamente fácil.

## CONCEPTOS DE DATOS

Antes de considerar el uso de archivos o del enfoque de la base de datos, es importante entender cómo se representan los datos. En esta sección se tratan las definiciones críticas, incluyendo la abstracción de datos del mundo real para el almacenamiento de datos en tablas y relaciones de la base de datos.



## ENGANCHE SU CARRETA DE LIMPIEZA A UNA ESTRELLA

La compañía Marc Schnieder Janitorial Supply le ha pedido ayuda para limpiar su almacenamiento de datos. En cuanto usted empieza a plantear a Marc Schnieder preguntas detalladas sobre su base de datos, su cara se descompone. “En realidad nosotros no tenemos una base de datos como usted la describe”, dice Marc con cierta angustia. “Siempre he querido limpiar nuestros registros, pero no he podido encontrar a una persona capaz que se encargue de ello.”

Después de hablar con el señor Schnieder, usted se dirige a la pequeña oficina de Stan Lessink, el jefe de programadores. Stan lo pone al tanto del desarrollo histórico del sistema de información actual. “La compañía Marc Schnieder Janitorial Supply es una historia de la pobreza a la riqueza”, comenta Stan. “El primer trabajo del señor Schnieder fue como empleado de limpieza en un boliche. Él ahorró suficiente dinero para comprar algunos productos y empezó a venderlos a otros boliches. Pronto decidió extender el negocio de productos de limpieza. Él se dio cuenta de que conforme su negocio crecía, tenía más líneas de productos y tipos de clientes. Los vendedores de la compañía se asignan a

diferentes líneas de productos principales (tiendas, oficinas, etc.); algunos son de ventas internas, y algunos se especializan en equipo pesado, como pulidoras y enceradoras. Los registros se almacenan en archivos separados.”

“El problema es que no tenemos ninguna manera de comparar las ganancias de cada división. Nos gustaría establecer programas de incentivos para vendedores y equilibrar mejor la asignación de los vendedores a cada línea de producto”, usted recuerda lo que dijo el señor Schnieder.

Sin embargo, Stan agrega: “Cada división tiene su propio sistema de incentivos. Las comisiones varían. No veo cómo podemos tener un sistema común. Además, yo puedo conseguir nuestros informes rápidamente porque nuestros archivos están de la manera que los necesitamos. Nunca hemos emitido un pago con retraso”.

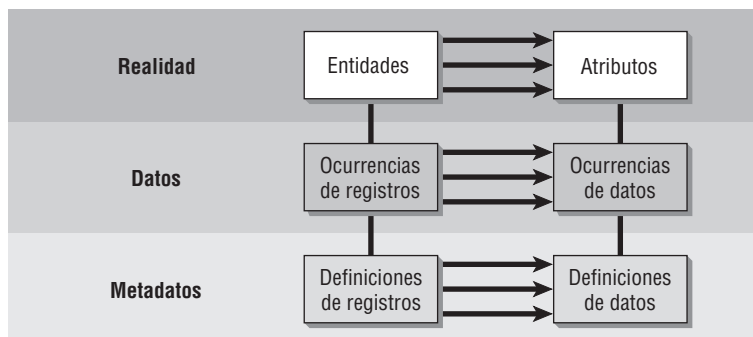
Describa de qué manera podría analizar las necesidades de almacenamiento de datos de la compañía Marc Schnieder Janitorial Supply. ¿Desecharía el sistema viejo o simplemente lo puliría un poco? Discuta las implicaciones de su decisión.

### REALIDAD, DATOS Y METADATOS

Al mundo real se le llamará realidad. En la realidad, los datos recopilados de personas, lugares o eventos se almacenarán eventualmente en un archivo o una base de datos. Para entender la forma y estructura de los datos, se necesita información sobre los datos mismos. A la información que describe los datos se le llama metadatos.

En la figura 13.1 se describe la relación entre realidad, datos y metadatos. Dentro del reino de la realidad hay entidades y atributos; dentro del reino de los datos reales hay ocurrencias de registros y ocurrencias de datos, y dentro del reino de los metadatos hay definiciones de registros y definiciones de datos. Los significados de estos términos se analizan en las siguientes subsecciones.

**Entidades** Una entidad es cualquier objeto o evento sobre el cual alguien escoge recolectar datos. Una entidad podría ser una persona, lugar o cosa (por ejemplo, un vendedor, una ciudad o un producto). Cualquier entidad también puede ser un evento o unidad de tiempo tal como la avería de una máquina, una venta o un mes o año. Además de las entidades que se explicaron en el capítulo 2 hay una entidad menor adicional llamada subtipo de entidad. Su símbolo es un rectángulo más pequeño dentro del rectángulo de la entidad.



**FIGURA 13.1**

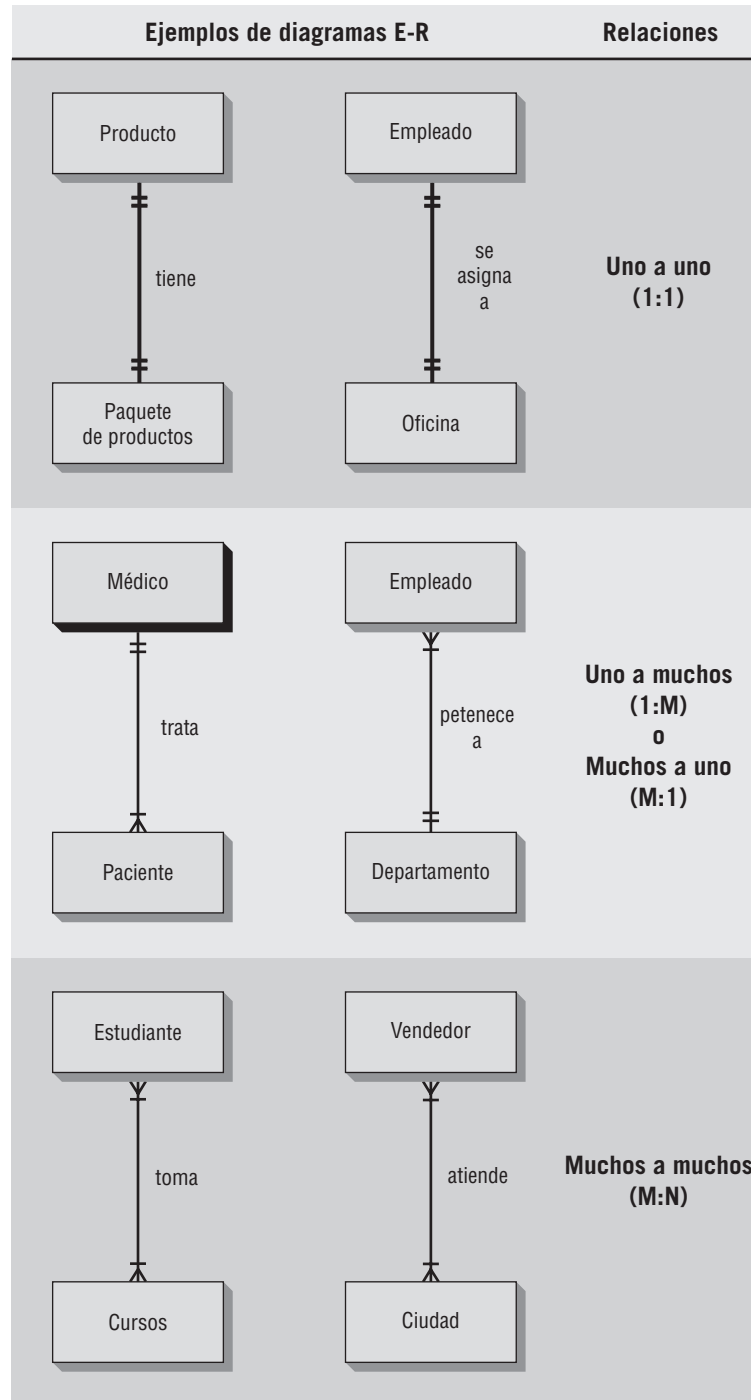
Realidad, datos y metadatos.

Un subtipo de entidad es una relación especial uno a uno que representa los atributos adicionales (campos) de otra entidad que podría no estar presente en cada registro de la primera entidad. Los subtipos de entidades eliminan la posibilidad de que una entidad pueda tener campos nulos almacenados en las tablas de la base de datos.

Un ejemplo es la entidad principal de un cliente. Los clientes preferidos podrían tener campos especiales que contengan información de descuentos especiales, y esta información estaría en un subtipo de entidad. Otro ejemplo son los estudiantes que tienen periodos de prácticas profesionales. El ARCHIVO MAESTRO DE ESTUDIANTES no debe contener información sobre los periodos de prácticas profesionales para cada estudiante, debido a que quizás sólo un número pequeño de estudiantes tiene dicho periodos.

**FIGURA 13.2**

Los diagramas entidad-relación (E-R) pueden mostrar relaciones de uno a uno, uno a muchos, muchos a uno o muchos a muchos.



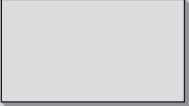
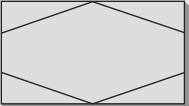
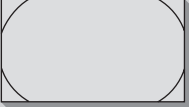
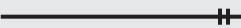

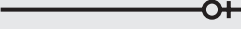

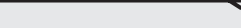
**Relaciones** Estas son asociaciones entre las entidades (a veces se conocen como asociaciones de datos). La figura 13.2 es un diagrama entidad-relación (E-R) que muestra varios tipos de relaciones.

El primer tipo de relación es una relación uno a uno (designada como 1:1). El diagrama muestra que sólo hay un PAQUETE DE PRODUCTOS para cada PRODUCTO. La segunda relación uno a uno muestra que cada EMPLEADO tiene una sola OFICINA. Observe que todas estas entidades se pueden describir aún más (el precio de un producto no sería una entidad, ni una extensión telefónica).

Otro tipo de relación es una relación uno a muchos (1:M) o muchos a uno. Como se muestra en la figura, a un MÉDICO, en un centro de salud, se le asignan muchos PACIENTES, pero a un PACIENTE se le asigna un solo MÉDICO. Otro ejemplo muestra que un EMPLEADO es un miembro de un solo DEPARTAMENTO, pero cada DEPARTAMENTO tiene muchos EMPLEADOS.

Finalmente, una relación muchos a muchos (designada como M:N) describe la posibilidad de que las entidades podrían tener muchas asociaciones en cualquier dirección. Por ejemplo, un ESTUDIANTE puede tener muchos CURSOS, y al mismo tiempo en un CURSO podría haber muchos ESTUDIANTES inscritos. El segundo ejemplo muestra que un VENDEDOR puede visitar muchas CIUDADES y una CIUDAD puede ser el área de ventas para muchos VENDEDORES.

En la figura 13.3 se dan los símbolos estándar para la notación de tipo pata de cuervo, la explicación oficial de los símbolos y su significado real. Observe que el símbolo para una entidad es un rectángulo. Una entidad se define como una clase de persona, lugar o cosa. Un rectángulo con un diamante dentro simboliza una entidad asociativa, la cual se usa para unir dos entidades. Un rectángulo con un óvalo dentro representa una entidad atributiva, la cual se usa para los grupos repetitivos.

Símbolo	Explicación oficial	Significado real
	Entidad	Una clase de personas, lugares o cosas
	Entidad asociativa	Se usa para unir dos entidades
	Entidad atributiva	Se usa para grupos repetitivos
	Relación a 1	Exactamente uno
	Relación a muchos	Uno o más
	Relación a 0 o 1	Sólo cero o uno
	Relación a 0 o más	Puede ser cero, uno o más
	Relación a más de 1	Mayor que uno

**FIGURA 13.3**  
Los símbolos entidad-relación y sus significados.

Las otras notaciones necesarias para dibujar los diagramas E-R son las conexiones, de las cuales hay cinco tipos diferentes. En la parte inferior de la figura se explica el significado de la notación. Cuando una línea recta conecta a dos entidades planas y el extremo de la línea se marca con dos marcas cortas (||), existe una relación uno a uno. Lo siguiente que observará es una unión tipo pata de cuervo con una marca corta (|); cuando esta notación vincula entidades, indica una relación uno a uno o uno a muchos (a uno o más).

Las entidades vinculadas con una línea recta más una marca corta (|) y un cero (el cual se parece más a un círculo, O) describen una relación uno a cero o uno a uno (sólo cero o uno). Un cuarto tipo de vínculo para relacionar las entidades se dibuja con una línea recta marcada en el extremo con un cero (O) seguido por una conexión tipo pata de cuervo. Este tipo muestra una relación cero a cero, cero a uno o cero a muchos. Finalmente, una línea recta que vincula las entidades con una conexión tipo pata de cuervo en el extremo describe una relación de más de uno.

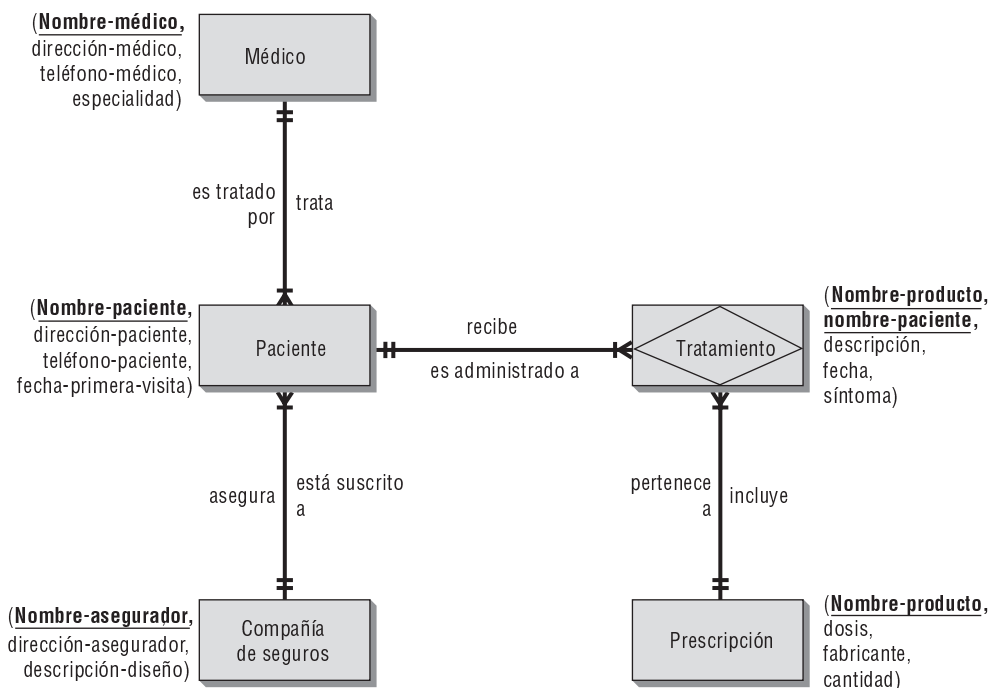
Una entidad podría tener una relación que la conecte a sí misma. Este tipo de relación se llama relación recursiva; la implicación es que debe haber una forma de vincular un registro de un archivo a otro registro del mismo archivo. Un ejemplo de una relación recursiva se puede encontrar en las simulaciones de HyperCase que se mencionan en los capítulos. Una tarea podría tener una tarea precedente (es decir, una tarea que se debe completar antes de empezar la actual). En esta situación, un registro (la tarea actual) apunta a otro registro (la tarea precedente) en el mismo archivo.

Las relaciones se pueden escribir con palabras en la parte superior o al lado de cada línea de conexión. En realidad, usted ve la relación en una dirección, aunque puede escribir las relaciones en ambos lados de la línea, donde cada una representa el enfoque de una de las dos entidades. (Véase el capítulo 2 para más detalles sobre la ilustración de diagramas E-R.)

**Ejemplo de entidad-relación** En la figura 13.4 se presenta un diagrama entidad-relación que contiene muchas entidades, muchos tipos diferentes de relaciones y varios atributos. En este diagrama E-R nos enfocamos en un sistema de facturación, y en particular con la parte

**FIGURA 13.4**

El diagrama entidad-relación para el tratamiento de un paciente. Los atributos se pueden listar al lado de las entidades. En cada caso, la clave se subraya.



de la prescripción del sistema. (Por simplicidad, asumimos que las visitas al consultorio se manejan de forma diferente y están fuera del alcance de este sistema.)

Las entidades son PRESCRIPCIÓN, MÉDICO, PACIENTE y COMPAÑÍA DE SEGUROS. La entidad de TRATAMIENTO no es importante para el sistema de facturación, pero es parte del diagrama E-R porque se usa para establecer una conexión entre la PRESCRIPCIÓN y el PACIENTE. Por lo tanto lo dibujamos como una entidad asociativa en la figura.

Aquí, un MÉDICO trata muchos PACIENTES (1:M), quienes se suscriben por separado a una COMPAÑÍA DE SEGUROS individual. Por supuesto, el PACIENTE es sólo uno de los muchos pacientes que se suscriben a dicha COMPAÑÍA DE SEGUROS particular (M:1).

Para completar los registros del MÉDICO, el médico necesita guardar la información acerca de los tratamientos que tiene un PACIENTE. Muchos PACIENTES experimentan muchos TRATAMIENTOS, lo que se convierte en una relación muchos a muchos (M:N). El TRATAMIENTO se representa como una entidad asociativa porque no es importante en nuestro sistema de facturación por sí mismo. Los TRATAMIENTOS pueden incluir la toma de PRESCRIPCIONES, y por ello también es una relación M:N, debido a que muchos tratamientos podrían requerir combinaciones de fármacos y muchos medicamentos podrían funcionar para muchos tratamientos.

Después algunos detalles se completan para los atributos. Los atributos se listan al lado de cada una de las entidades, y la clave se subraya. Por ejemplo, la entidad PRESCRIPCIÓN tiene un NOMBRE-PRODUCTO, DOSIFICACIÓN, FABRICANTE y CANTIDAD. En teoría, sería benéfico diseñar una base de datos de esta forma, usando diagramas entidad-relación y después completando los detalles acerca de los atributos. Este enfoque de arriba abajo es provechoso, pero a veces es muy difícil lograr.

**Atributos** Un atributo es una característica de una entidad. Puede haber muchos atributos para cada entidad. Por ejemplo, un paciente (entidad) puede tener muchos atributos, tal como apellido, nombre, calle, ciudad, estado, etc. La fecha de última visita del paciente así como los detalles de la prescripción también son atributos. Cuando se construyó el diccionario de datos en el capítulo 8, el componente más pequeño descrito se llamó elemento de datos. Cuando los archivos y bases de datos se discuten, estos elementos de datos generalmente se conocen como datos. De hecho, estos datos son las unidades más pequeñas en un archivo o base de datos. El término *datos* también se usa de forma indistinta con la palabra *atributo*.

Los datos pueden tener valores. Estos valores pueden ser de longitud fija o variable; pueden ser caracteres alfabéticos, numéricos, especiales o alfanuméricos. En la figura 13.5 se pueden encontrar ejemplos de datos y sus valores.

A veces un dato también se conoce como campo. Sin embargo, un campo representa algo físico, no lógico. Por lo tanto, muchos datos se pueden empaquetar en un campo; el campo se puede leer y convertir en varios datos. Un ejemplo común de esto es almacenar la fecha en un solo campo como MM/DD/AAAA. Para ordenar el archivo de acuerdo la fecha, se extraen por separado tres datos del campo y se ordenan primero por AAAA, luego por MM y finalmente por DD.

**Registros** Un registro es una colección de datos que tiene algo en común con la entidad descrita. La figura 13.6 es una ilustración de un registro con muchos datos relacionados. El registro mostrado es para un pedido hecho con una compañía de ventas por correo. El #PEDIDO, APELLIDO, INICIAL, CALLE, CIUDAD, ESTADO y TARJETA DE CRÉDITO son atributos. La mayoría de los registros son de longitud fija, de modo que no es necesario determinar la longitud todo el tiempo.

Bajo ciertas circunstancias (por ejemplo, cuando el espacio es importante), se usan registros de longitud variable. Un registro de longitud variable se usa como alternativa para reservar una gran cantidad de espacio para el registro más grande posible, tal como el número

**FIGURA 13.5**

Los valores típicos asignados a datos podrían ser números, caracteres alfabéticos, caracteres especiales y combinaciones de los tres.

Entidad	Datos	Valor
Vendedor	Número del vendedor	87254
	Nombre del vendedor	Kaytell
	Nombre de la compañía	Music Unlimited
	Dirección	45 Arpeum Circle
	Ventas	\$20,765
Paquete	Ancho	2
	Alto	16
	Longitud	16
	Peso	3
	Dirección de envío	765 Dulcinea Drive
	Dirección de devolución	P.O. Box 341, Spring Valley, MN
Pedido	Producto(s)	B521
	Descripción(es)	"My Fair Lady" disco compacto
	Cantidad pedida	1
	Apellido del cliente	Kiley
	Inicial	R.
	Calle	765 Dulcinea Drive
	Ciudad	La Mancha
	Estado	CA
	Código postal	93407
	Número de la tarjeta de crédito	65-8798-87
	Fecha en la que se hizo el pedido	05/01/2003
	Cantidad	\$6.99
	Estado	Nueva orden de pedido

máximo de visitas que un paciente ha hecho a un médico. Cada visita podría contener muchos datos que serían parte del registro completo del paciente (o carpeta de archivo en un sistema manual). Más adelante en este capítulo, se discute la normalización de una relación. La normalización es un proceso que elimina los grupos repetitivos encontrados en los registros de longitud variable.

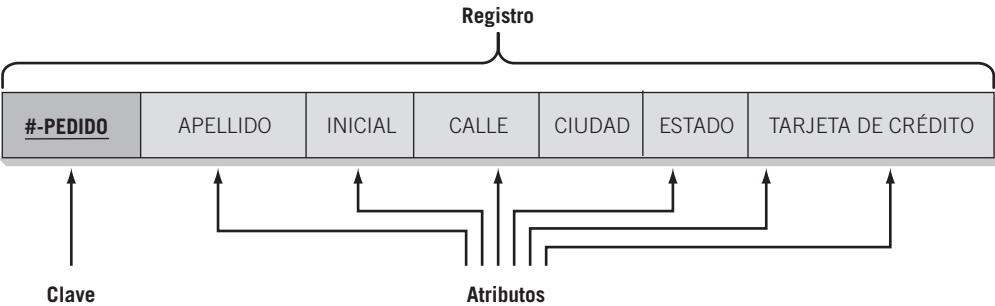
**Claves** Una clave es uno de los datos en un registro que se usa para identificar al registro. Cuando una clave identifica de forma única un registro, se llama clave primaria. Por ejemplo, #-PEDIDO puede ser una clave primaria porque a cada pedido del cliente se asigna un solo número. De esta forma, la clave primaria identifica la entidad real (pedido del cliente).

Si una clave no identifica de forma única un registro, se le llama clave secundaria. Las claves secundarias se pueden usar para seleccionar un grupo de registros que pertenecen a un conjunto (por ejemplo, pedidos que vienen del estado de Virginia).

Cuando no es posible identificar de forma única un registro usando uno de los datos encontrados en un registro, se puede construir una clave seleccionando dos o más datos y

**FIGURA 13.6**

Un registro tiene una clave primaria y podría tener muchos atributos.





Datos	Valor	
Número del vendedor	N	5
Nombre del vendedor	A	20
Nombre de la compañía	A	26
Dirección	A	36
Ventas	N	9.2
Ancho	N	2
Alto	N	2
Longitud	N	2
Peso	N	2
Dirección de envío	A	36
Dirección de devolución	A	36
Producto(s)	A	4
Descripción(es)	A	30
Cantidad pedida	N	2
Apellido del cliente	A	24
Inicial	A	1
Calle	A	28
Ciudad	A	12
Estado	A	2
Código postal	N	9
Número de la tarjeta de crédito	N	10
Fecha en la que se hizo el pedido	D	8
Cantidad	\$	7.2
Estado	A	22

Campos

N Numérico

A Alfanumérico o texto

D Fecha MM/DD/AAAA

\$ Dinero

M Memorándum

7.2 significa que el campo ocupa 7 dígitos, dos de los cuales están a la derecha del decimal.

Se podrían especificar formatos especiales para los campos.

MM/DD/AAAA

**FIGURA 13.7**

Los metadatos incluyen una descripción de cómo se deben ver los valores de cada dato.

combinándolos. Esta clave se llama clave concatenada. Cuando un dato se usa como clave en un registro, se subraya la descripción. Por lo tanto, en el registro PEDIDO (#-PEDIDO, APELLIDO, INICIAL, CALLE, CIUDAD, ESTADO, TARJETA DE CRÉDITO), la clave es #-PEDIDO. Si un atributo es una clave en otro archivo, se debe subrayar con una línea punteada.

**Metadatos** Los metadatos son datos que definen a los datos en el archivo o base de datos. Los metadatos describen el nombre dado y la longitud asignada a cada dato. Los metadatos también describen la longitud y composición de cada uno de los registros.

La figura 13.7 es un ejemplo de metadatos para una base de datos perteneciente a algún software genérico. La longitud de cada dato se indica según una convención, donde 7.2 significa que se reservan siete espacios para el número y que dos de esos dígitos están a la derecha del punto decimal. La letra N significa “numérico” y la A representa “alfanumérico”. La D representa la “fecha” y está automáticamente en el formato MM/DD/AAAA. Algunos programas, tal como Microsoft Access, usan español común para los metadatos, de manera que se usan palabras tales como *texto*, *dinero* y *número*. Microsoft Access proporciona, de forma predeterminada, 50 caracteres como la longitud del campo para los nombres, lo cual está bien al trabajar con sistemas pequeños. Sin embargo, si trabaja con una base de datos grande para un banco o una compañía de servicios públicos, por ejemplo, no deberá destinar tanto espacio para dicho campo. De lo contrario, la base de datos crecería bastante y tendría demasiado espacio desperdiciado. En estos casos es cuando puede usar metadatos para anticipar la solución a posibles problemas futuros y diseñar una base de datos más eficaz.

## ORGANIZACIÓN DE ARCHIVOS

Un archivo contiene grupos de registros que proporcionan información para la operación, diseño, administración y toma de decisiones en una organización. Los tipos de archivos usados se describen primero, seguidos de una descripción de las muchas formas en que se pueden organizar los archivos convencionales.

**Tipos de archivo** Los archivos se pueden usar para almacenar datos por un periodo indefinido, o se pueden usar para almacenar datos temporalmente para un propósito específico. Los archivos maestros y de tabla se usan para almacenar datos por un periodo largo. Los archivos temporales normalmente se llaman archivos de transacción, archivos de trabajo o archivos de reporte.

**Archivos maestros.** Los archivos maestros contienen registros para un grupo de entidades. Con frecuencia los atributos se podrían actualizar, pero los registros en sí son relativamente permanentes. Estos archivos son propensos a tener registros grandes que contienen toda la información sobre una entidad de datos. Cada registro normalmente contiene una clave primaria y varias claves secundarias. Los archivos maestros se encuentran como tablas en una base de datos o como archivos indexados o del tipo indexado-secuencial.

Aunque el analista es libre de distribuir en cualquier orden los elementos de datos en un archivo maestro, una distribución estándar es poner primero el campo de clave primaria, seguido por los elementos descriptivos y finalmente por elementos que reflejan el negocio y cambian frecuentemente con las actividades del negocio. Este procedimiento permite a los analistas, o a otras personas que tienen acceso a estos archivos, identificar fácilmente los registros cuando un archivo se lista con una rutina de impresión.

La información descriptiva son datos que no cambian con los eventos del negocio, tal como una descripción del artículo, nombre del cliente, dirección o departamento del empleado. Estos elementos normalmente se cambian por programas de mantenimiento que usan métodos de acceso directo. Normalmente, estos elementos contienen claves alternas o índices, y los datos están en un formato de despliegue.

Los elementos de información del negocio son aquellos que cambian periódicamente con los eventos del negocio, tales como sueldo bruto acumulado en el año, promedio de calificaciones, saldo de la cuenta del cliente y la fecha de la última compra del cliente. Estos elementos son modificados por programas de actualización que, por eficiencia, normalmente leen tanto los archivos como los registros que coinciden de manera secuencial. Los elementos del registro sólo son modificados por los programas cuando los datos tienen algún error, usando métodos aleatorios de actualización. Con frecuencia los campos de tipo monetario (dólares o pesos) están en un formato de datos comprimido llamado decimal condensado para ahorrar espacio en los archivos y acelerar el tiempo de ejecución del programa.

Si el archivo maestro se almacena usando métodos convencionales, se reserva un área de expansión al final de cada registro. Esta área proporciona espacio para agregar nuevos campos al registro conforme cambien las necesidades del negocio. Si el archivo es parte de una estructura de la base de datos, no se requiere el área de expansión. Entre los ejemplos de un archivo maestro se incluyen los registros de pacientes, registros de clientes, un archivo de personal y un archivo de inventario de partes.

**Archivos de tabla.** Un archivo de tabla contiene datos usados para calcular más datos o medidas de desempeño. Un ejemplo es una tabla de tasas de correos usada para determinar los gastos de envío de un paquete. Otro ejemplo es una tasa de impuestos. Normalmente los archivos de tabla se leen únicamente por un programa.

**Archivos de transacción.** Un archivo de transacción se usa para hacer cambios que actualizan el archivo maestro y producen informes. Suponga que el archivo maestro de un suscriptor de periódico necesita ser actualizado; el archivo de transacción contendría el número del suscriptor y un código de transacción tal como E para extender la suscripción, C para cancelar la suscripción o A para cambiar la dirección. Después sólo se necesita introducir la información relevante para la actualización; es decir, la longitud de renovación si es E o la dirección si es A. Si la suscripción fue cancelada no se necesitaría ninguna información adicional. El resto de la información ya existe en el archivo maestro. Como resultado, nor-

	# DEL PEDIDO	APELLIDO	I	CALLE	CIUDAD	ES	TARJETA DE CRÉDITO
1	10784	MacRae	G	2314 Curly Circle	Lincoln	NE	45-4654-76
2	10796	Jones	S	34 Dream Lane	Oklahoma City	OK	45-9876-74
3	11821	Preston	R	1008 Madison Ave.	River City	IA	34-7642-64
4	11845	Channing	C	454 Harmonia St.	New York	NY	34-0876-87
5	11872	Kiley	R	765 Dulcinea Drive	La Mancha	CA	65-8798-87
6	11976	Verdon	G	7564 K Street	Chicago	IL	67-8453-18
7	11998	Rivera	C	4342 West Street	Chicago	IL	12-2312-54
8	12765	Orbach	J	1345 Michigan Ave.	Chicago	IL	23-4545-65
9	12769	Steele	T	3498 Burton Lane	Finnian	NJ	65-7687-09
10	12965	Crawford	M	1986 Barnum Cir.	London	NH	23-0098-23
11	13432	Cullum	J	354 River Road	Shenandoah	VT	45-8734-33
12	13542	Mostel	Z	65 Fiddler Street	Anatevka	ND	34-6723-98

**FIGURA 13.8**

Un archivo secuencial ordenado por #-PEDIDO.

malmente los archivos de transacción se mantienen a una longitud mínima. Los archivos de transacción podrían contener varios tipos diferentes de registros, tales como los tres usados para actualizar el maestro de suscriptores a un periódico, con un código en el archivo de transacción que indica el tipo de transacción.

**Archivos de trabajo.** Algunas veces un programa se puede ejecutar con mayor eficacia si se usa un archivo de trabajo. Un ejemplo común de un archivo de trabajo es cuando se reordena un archivo para acceder a los registros con mayor rapidez para cierto tipo de procesos.

**Archivos de reporte.** Cuando se necesita imprimir un informe y no hay ninguna impresora disponible (por ejemplo, cuando la impresora está ocupada imprimiendo otros trabajos), se usa un archivo de reporte. Enviar la salida a un archivo en lugar de a una impresora se denomina *spooling*. Después, cuando el dispositivo está listo, el documento se puede imprimir. Los archivos de reporte son muy útiles, debido a que los usuarios pueden tomar los archivos de otros sistemas de cómputo y enviarlos a dispositivos especializados tales como graficadores, impresoras láser, unidades de microficha e incluso máquinas de composición tipográfica computarizadas.

**Organización secuencial** Cuando los registros están físicamente en orden en un archivo, se dice que éste es un archivo secuencial. Cuando un archivo secuencial se actualiza, es necesario pasar por el archivo entero. Debido a que los registros no se pueden insertar en medio del archivo, normalmente se copia un archivo secuencial completo durante el proceso de actualización.

La figura 13.8 ilustra un archivo de pedidos actuales para una compañía de ventas por correo que vende CDs de música. El archivo contiene 12 registros y se almacena en forma secuencial según el #-PEDIDO. Si queremos buscar el pedido 13432, tendríamos que empezar desde el principio y leer todo el archivo hasta llegar a dicho pedido.

Los archivos maestros secuenciales se usan cuando el hardware lo requiere (recuerde que una cinta magnética es un dispositivo secuencial) o cuando el acceso normal requiere que la mayoría de los registros se acceda. Es decir, cuando necesitamos leer o actualizar sólo unos registros, es ineficaz usar una estructura secuencial, pero cuando se necesita leer o modificar muchos registros, la organización secuencial tendría sentido. Normalmente la organización secuencial se usa para todos los tipos de archivos excepto los archivos maestros.

**Listas enlazadas** Cuando los archivos se almacenan en dispositivos de acceso directo tal como un disco, las opciones se extienden. Los registros se pueden ordenar lógicamente, en lugar de físicamente, usando listas enlazadas. Las listas enlazadas se logran usando un conjunto de indicadores para dirigirlo al próximo registro lógico ubicado en cualquier parte del archivo.

La figura 13.9 muestra el archivo que ordena los CDs de música con un atributo adicional usado para almacenar el indicador. Debido a que el archivo ya está almacenado en orden secuencial de acuerdo con el #-PEDIDO, el indicador se usa para apuntar a los registros en orden lógico (alfabético) por el APELLIDO. Este ejemplo muestra una ventaja obvia de usar las listas enlazadas: los archivos se pueden ordenar de forma lógica de muchas maneras diferentes usando una variedad de indicadores.

**FIGURA 13.9**

Una lista enlazada usa indicadores para designar el orden lógico de los registros.

		ORDEN						
		INICIO = 4						
	# DEL PEDIDO	APELLIDO	I	CALLE	CIUDAD	ES	TARJETA DE CRÉDITO	INDICADOR
1	10784	MacRae	G	2314 Curly Circle	Lincoln	NE	45-4654-76	6
2	10796	Jones	S	34 Dream Lane	Oklahoma City	OK	45-9876-74	4
3	11821	Preston	R	1008 Madison Ave.	River City	IA	34-7642-64	9
4	11845	Channing	C	454 Harmonia St.	New York	NY	34-0876-87	1
5	11872	Kiley	R	765 Dulcinea Drive	La Mancha	CA	65-8798-87	5
6	11976	Verdon	G	7564 K Street	Chicago	IL	67-8453-18	END
7	11998	Rivera	C	4342 West Street	Chicago	IL	12-2312-54	10
8	12765	Orbach	J	1345 Michigan Ave.	Chicago	IL	23-4545-65	8
9	12769	Steele	T	3498 Burton Lane	Finnian	NJ	65-7687-09	11
10	12965	Crawford	M	1986 Barnum Cir.	London	NH	23-0098-23	2
11	13432	Cullum	J	354 River Road	Shenandoah	VT	45-8734-33	3
12	13542	Mostel	Z	65 Fiddler Street	Anatevka	ND	34-6723-98	7

**Organización de un archivo *hash*** Los dispositivos de acceso directo también permiten acceso a un registro dado yendo directamente a su dirección. Debido a que no es factible reservar una dirección física para cada registro posible, se usa un método llamado *hashing* (reordenamiento). *Hashing* es el proceso de calcular una dirección a partir de la clave del registro.

Suponga que en una organización había 500 empleados y quisimos usar el número de seguros social como una clave. Sería ineficaz reservar 999,999,999 direcciones, una para cada número de seguros social. Por lo tanto, podríamos tomar el número de seguros social y usarlo para derivar la dirección del registro.

Hay muchas técnicas de *hashing*. Una común es dividir el número original entre un número primo que se aproxime a las ubicaciones de almacenamiento y después usar el residuo como la dirección, como sigue: empiece con el número de seguros social 053-46-8942. Después divida entre 509, para obtener 105,047. Observe que 105,047 multiplicado por 509 no es igual al número original; en cambio es igual a 53,468,923. La diferencia entre el número original, 53,468,942, y el dividendo, 53,468,923, es el residuo, y es igual a 19. De tal manera, la ubicación de almacenamiento del registro para un empleado cuyo número de seguros social es 053-46-8942 sería 19.

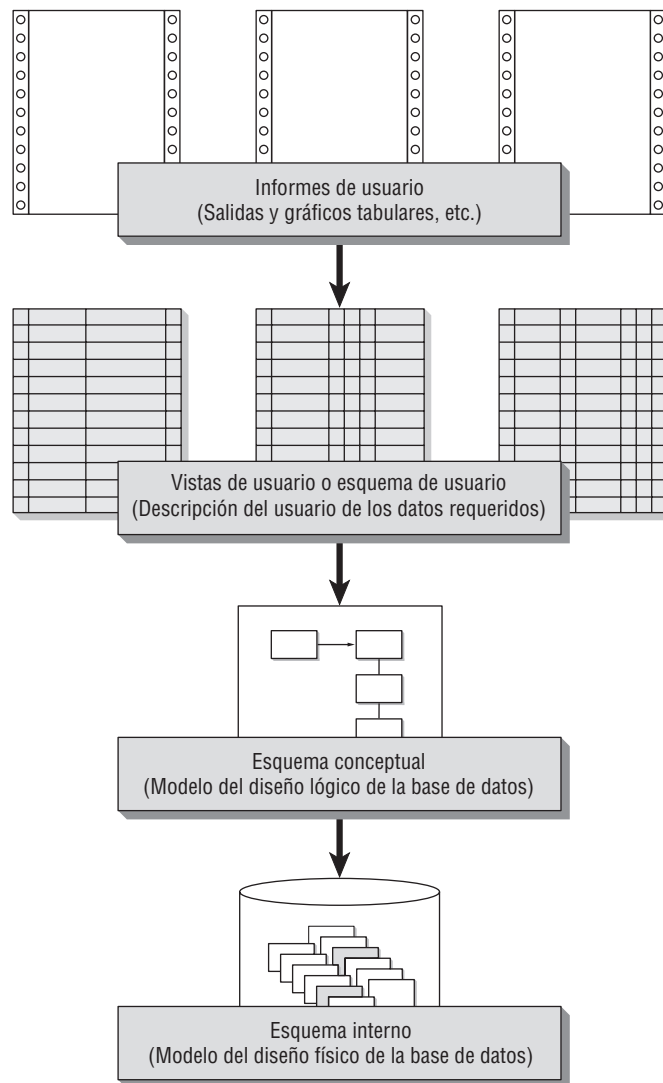
Sin embargo, surge un problema cuando una persona con un número de seguro social diferente (por decir, 472-38-4086) tiene el mismo residuo. Cuando esto ocurre, el registro de la segunda persona se debe poner en un área de reserva especial.

## BASES DE DATOS RELACIONALES

Las bases de datos se pueden organizar de varias formas. Aquí consideraremos el enfoque más común, la base de datos relacional.

**Vistas lógicas y físicas de datos** Una base de datos, a diferencia de un archivo, es diseñada para ser compartida por muchos usuarios. Está claro que todos los usuarios ven los datos de formas diferentes. Nos referiremos a la forma en que un usuario visualiza y describe los datos como una vista de usuario. Sin embargo, el problema es que diferentes usuarios tienen vistas de usuario distintas. El analista de sistemas debe examinar estas vistas y debe desarrollar un modelo lógico global de la base de datos. Finalmente, dicho modelo lógico se debe transformar en el diseño físico correspondiente de la base de datos. El diseño físico describe la forma como se almacenan y relacionan los datos, así como también la forma en que se acceden.

En la literatura de base de datos, las vistas se denominan esquema. La figura 13.10 muestra cómo el informe de usuario y la vista de usuario (esquema de usuario) se relaciona al modelo lógico (esquema conceptual) y al diseño físico (esquema interno).



**FIGURA 13.10**

El diseño de bases de datos incluye el sintetizar los informes de los usuarios, vistas de usuarios y los diseños lógicos al igual que los físicos.

Hay tres tipos principales de bases de datos estructuradas de forma lógica: jerárquica, red y relacional. Los primeros dos tipos se pueden encontrar en sistemas heredados (antiguos). Hoy en día, un analista típicamente diseñaría una base de datos relacional.

**Estructuras relacionales de datos** Una estructura relacional de datos consiste en una o más tablas bidimensionales, las cuales se denominan relaciones. Las filas de la tabla representan registros y las columnas contienen atributos.

En la figura 13.11 se denomina estructura relacional a la base de datos que ordena los CDs de música descrita anteriormente en este capítulo. Aquí, se necesitan tres tablas para (1) describir los artículos y llevar un registro del precio actual de CDs (PRECIO DEL ARTÍCULO), (2) describir los detalles del pedido (PEDIDO) y (3) identificar el estado del pedido (ESTADO DEL ARTÍCULO).

Para determinar el precio de un artículo, necesitamos saber el número del artículo para poder encontrarlo en la relación PRECIO DEL ARTÍCULO. Para actualizar el número de tarjeta de crédito de G. MacRae, podemos investigar la relación de PEDIDO para MacRae y corregirla una sola vez, aunque haya pedido muchos CDs. Sin embargo, para averiguar el estado de una parte de un pedido debemos saber el #-ARTÍCULO y el #-PEDIDO y después debemos localizar esa información en la relación ESTADO DEL ARTÍCULO.

Normalmente, mantener las tablas en una estructura relacional es bastante simple en comparación con mantenerlas en una estructura jerárquica o de red. Una de las prin-

**FIGURA 13.11**

En una estructura de datos relacionales, los datos se almacenan en muchas tablas.

PRECIO DEL ARTÍCULO		
#-ARTÍCULO	TÍTULO	PRECIO
B235	Guys and Dolls	8.99
B521	My Fair Lady	6.99
B894	42nd Street	10.99
B992	A Chorus Line	10.99

PEDIDO						
#-PEDIDO	APELLIDO	I	CALLE	CIUDAD	ES	CUENTA DE COBRO
10784	MacRae	G	2314 Curiy Circle	Lincoln	NE	45-4654-76
10796	Jones	S	34 Dream Lane	Oklahoma City	OK	44-9876-74
11821	Preston	R	1008 Madison Ave.	River City	IA	34-7642-64
11845	Channing	C	454 Harmonia St.	New Y ork	NY	34-0876-87
11872	Kiley	R	765 Dulcinea Drive	La Mancha	CA	65-8798-87

ESTADO DEL ARTÍCULO		
#-ARTÍCULO	#-PEDIDO	ESTADO
B235	10784	Enviado 5/12
B235	19796	Enviado 5/14
B235	11872	En proceso
B521	11821	En proceso
B894	11845	Nueva orden de pedido
B894	11872	Enviado 5/12
B992	10784	Enviado 5/12

cipales ventajas de las estructuras relacionales es que las consultas *ad hoc* se manejan eficazmente.

Cuando se discuten las estructuras relacionales en la literatura de base de datos, con frecuencia se usa una terminología diferente. Un archivo se conoce como tabla o relación, un registro normalmente se denomina tupla y el conjunto de valores de un atributo se llama dominio.

Para que las estructuras relacionales sean útiles y manejables, primero se deben normalizar las tablas relacionales. La normalización se detalla en la sección siguiente.

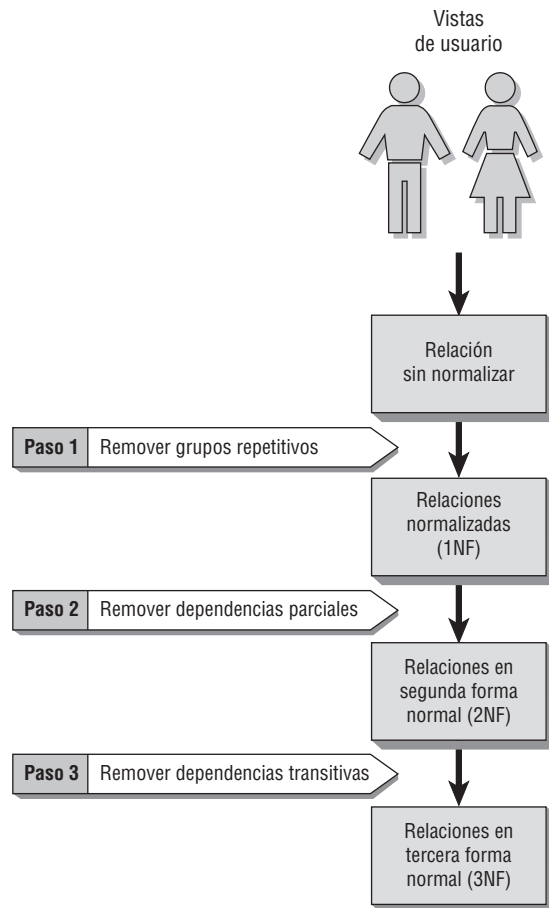
## NORMALIZACIÓN

La normalización es la transformación de las vistas de usuario complejas y del almacén de datos a un juego de estructuras de datos más pequeñas y estables. Además de ser más simples y estables, las estructuras de datos normalizadas son más fáciles de mantener que otras estructuras de datos.

### LOS TRES PASOS DE LA NORMALIZACIÓN

Ya sea que empiece con una vista de usuario o un almacén de datos desarrollado para un diccionario de datos (véase el capítulo 8), el analista normaliza una estructura de datos en tres pasos, como se muestra en la figura 13.12. Cada paso involucra un procedimiento importante, el cual simplifica la estructura de datos.

La relación derivada de la vista de usuario o del almacén de datos probablemente no estará normalizada. El primer paso del proceso incluye quitar todos los grupos repetitivos e identificar la clave primaria. Para ello, la relación se debe dividir en dos o más rela-



**FIGURA 13.12**

La normalización de una relación se realiza en tres pasos importantes.

ciones. A estas alturas, las relaciones ya podrían ser de la tercera forma normal, pero probablemente se necesitarán más pasos para transformar las relaciones a la tercera forma normal.

El segundo paso asegura que todos los atributos sin clave son totalmente dependientes de la clave primaria. Todas las dependencias parciales se remueven y se ponen en otra relación.

El tercer paso remueve cualesquier dependencias transitivas. Una dependencia transitiva es aquella en la que los atributos sin clave son dependientes de otros atributos sin clave.

## EJEMPLO DE NORMALIZACIÓN

La figura 13.13 es una vista de usuario para la Al S. Well Hydraulic Equipment Company. El informe muestra el (1) NÚMERO-VENDEDOR, (2) NOMBRE-VENDEDOR y (3) ÁREA-VENTAS. El cuerpo del informe muestra el (4) NÚMERO-CLIENTE y (5) NOMBRE-CLIENTE. Luego está (6) NÚMERO-ALMACÉN que servirá al cliente, seguido por (7) UBICACIÓN-ALMACÉN, la cual es la ciudad en donde se localiza la compañía. La última información contenida en la vista de usuario es la (8) CANTIDAD-VENTAS. Las filas (una para cada cliente) en la vista de usuario muestran que los artículos 4 a 8 forman un grupo repetitivo.

Si el analista usara un enfoque de flujo de datos/diccionario de datos, la misma información en la vista de usuario aparecería en una estructura de datos. La figura 13.14 muestra cómo aparecería la estructura de datos en el paso del diccionario de datos del análisis. El grupo repetitivo también se indica en la estructura de datos con un asterisco (\*) y con sangría.

**FIGURA 13.13**

Un informe de usuario para la  
Al S. Well Hydraulic Equipment  
Company.

**Al S. Well  
Hydraulic Equipment Company  
Spring Valley, Minnesota**

# del Vendedor: 3462  
Nombre: Waters W  
Área de ventas: Oeste

NÚMERO DEL CLIENTE	NOMBRE DEL CLIENTE	NÚMERO DEL ALMACÉN	UBICACIÓN DEL ALMACÉN	VENTAS
18765	Delta Service	4	Fargo	13,540
18830	M. Levy and Sons	3	Bismarck	10,600

Antes de proceder, observe las asociaciones de datos de los elementos de datos de la figura 13.15. Este tipo de ilustración se denomina diagrama de burbuja o diagrama de modelo de datos. Cada entidad se encierra en una elipse, y las flechas se usan para mostrar las relaciones. Aunque es posible dibujar estas relaciones con un diagrama E-R, a veces es más fácil usar el diagrama de burbuja más simple para modelar los datos.

En este ejemplo, sólo hay un NÚMERO-VENDEDOR asignado a cada NOMBRE-VENDEDOR, y esa persona cubrirá sólo un ÁREA-VENTAS, pero cada ÁREA-VENTAS se podría asignar a muchos vendedores: por ello, la notación de flecha doble del ÁREA-VENTAS al NÚMERO-VENDEDOR. Para cada NÚMERO-VENDEDOR, podría haber muchos NÚMEROS-CLIENTES.

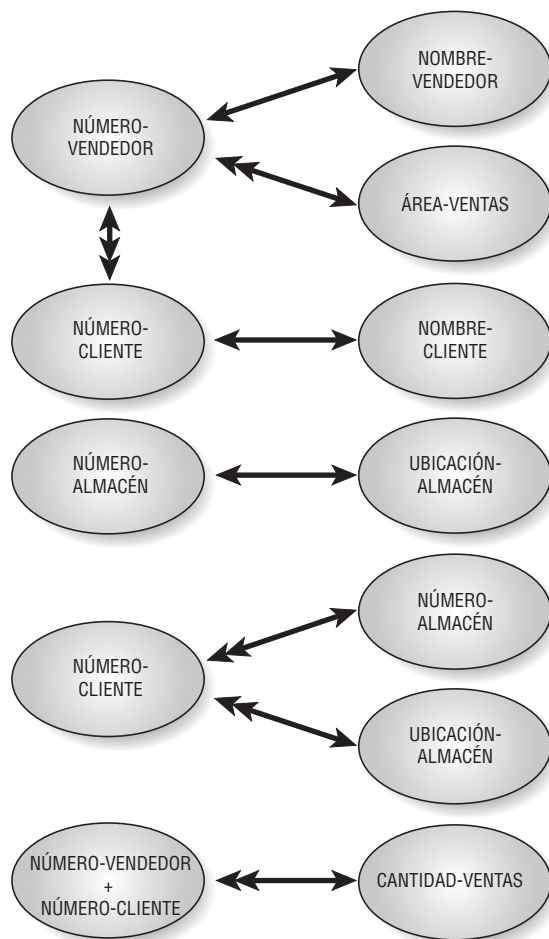
Además, habría una correspondencia de uno a uno entre NÚMERO-CLIENTE y NOMBRE-CLIENTE; lo mismo se aplica para NÚMERO-ALMACÉN y UBICACIÓN-ALMACÉN. El NÚMERO-CLIENTE tendrá sólo un NÚMERO-ALMACÉN y UBICACIÓN-ALMACÉN, pero cada NÚMERO-ALMACÉN o UBICACIÓN-ALMACÉN podrían servir

**FIGURA 13.14**

El analista podría encontrar  
una estructura de datos (de un  
diccionario de datos) útil en  
el desarrollo de una base de  
datos.

NÚMERO-VENDEDOR  
NOMBRE-VENDEDOR  
ÁREA-VENTAS  
NÚMERO-CLIENTE\* (1-)  
NOMBRE-CLIENTE  
NÚMERO-ALMACÉN  
UBICACIÓN-ALMACÉN  
CANTIDAD-VENTAS





**FIGURA 13.15**

Dibujar diagramas de modelo de datos para las asociaciones de datos algunas veces ayuda a analistas a apreciar la complejidad del almacén de datos.

a muchos NÚMERO-CLIENTE. Finalmente, para determinar la CANTIDAD-VENTAS para las visitas de un vendedor a una compañía particular, es necesario saber el NÚMERO-VENDEDOR y el NÚMERO-CLIENTE.

El objetivo principal del proceso de la normalización es simplificar todos los datos complejos que se encuentran a menudo en las vistas de usuario. Por ejemplo, si el analista tomara la vista de usuario descrita arriba y hubiera intentado extender una tabla relacional de ella, la tabla se vería como la figura 13.16. Debido a que esta relación se basa en nuestra vista de usuario inicial, la denominaríamos INFORME-VENTAS.

El INFORME-VENTAS es una relación sin normalizar, debido a que tiene grupos repetitivos. También es importante observar que un solo atributo tal como NÚMERO-VENDEDOR no puede servir como la clave. La razón queda clara cuando alguien examina las relaciones entre el NÚMERO-VENDEDOR y los otros atributos en la figura

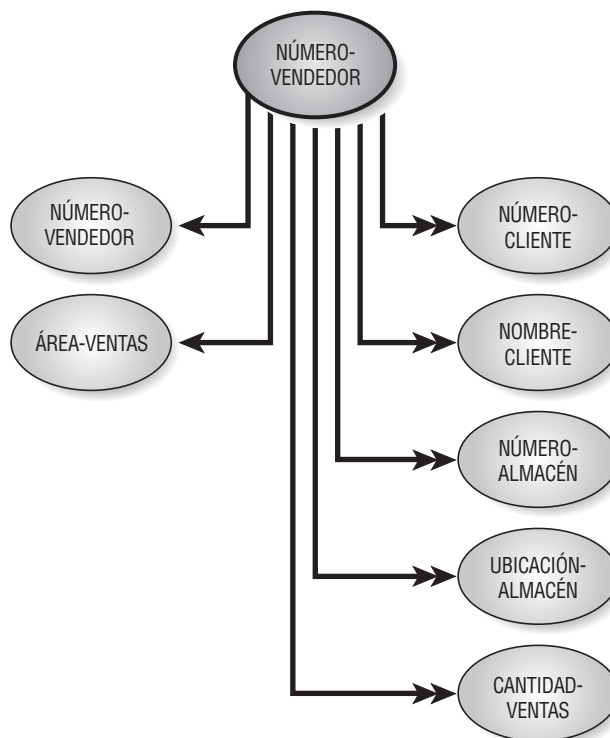
NÚMERO VENDEDOR	NOMBRE VENDEDOR	ÁREA VENTAS	NÚMERO CLIENTE	NOMBRE CLIENTE	NÚMERO ALMACÉN	UBICACIÓN ALMACÉN	CANTIDAD VENTAS
3462	Waters	West	18765	Delta Systems	4	Fargo	13540
			18830	A. Levy and Sons	3	Bismarck	10600
			19242	Ranier Company	3	Bismarck	9700
3593	Dryne	East	18841	R. W. Flood Inc.	2	Superior	11560
			18899	Seward Systems	2	Superior	2590
			19565	Stodola's Inc.	1	Plymouth	8800
etc.							

**FIGURA 13.16**

Si los datos fueran listados en una tabla sin normalizar, podría haber grupos repetidos.

**FIGURA 13.17**

Un diagrama de modelo de datos muestra que en una relación sin normalizar, el NÚMERO-VENDEDOR tiene una relación uno a muchos con algunos atributos.



13.17. Aunque hay una correspondencia de uno a uno entre el NÚMERO-VENDEDOR y dos atributos (NOMBRE-VENDEDOR y ÁREA-VENTAS), hay una relación uno a muchos entre el NÚMERO-VENDEDOR y los otros cinco atributos (NÚMERO-CLIENTE, NOMBRE-CLIENTE, NÚMERO-ALMACÉN, UBICACIÓN-ALMACÉN y CANTIDAD-VENTAS).

El INFORME-VENTAS se puede expresar en la siguiente notación de abreviatura:

INFORME	(NÚMERO-VENDEDOR,
DE VENTAS	NOMBRE-VENDEDOR, ÁREA-VENTAS,
	(NÚMERO-CLIENTE,
	NOMBRE-CLIENTE,
	NÚMERO-ALMACÉN,
	UBICACIÓN-ALMACÉN,
	CANTIDAD-VENTAS))

donde el juego interno de paréntesis representa el grupo repetitivo.

**Primera forma normal (1NF)** El primer paso para normalizar una relación es remover los grupos repetitivos. En nuestro ejemplo, la relación sin normalizar INFORME-VENTAS se separará en dos relaciones separadas. Estas nuevas relaciones se nombrarán VENDEDOR y VENDEDOR-CLIENTE.

La figura 13.18 muestra cómo se normaliza la relación original sin normalizar INFORME-VENTAS al separar la relación en dos nuevas relaciones. Observe que la relación VENDEDOR contiene la clave primaria NÚMERO-VENDEDOR y todos los atributos que no eran repetitivos (NOMBRE-VENDEDOR y ÁREA-VENTAS).

La segunda relación, CLIENTE-VENDEDOR, contiene la clave primaria de la relación VENDEDOR (la clave primaria de VENDEDOR es NÚMERO-VENDEDOR), así como también todos los atributos que eran parte del grupo repetitivo (NÚMERO-CLIENTE, NOMBRE-CLIENTE, NÚMERO-ALMACÉN, UBICACIÓN-ALMACÉN y

INFORME DE LAS VENTAS

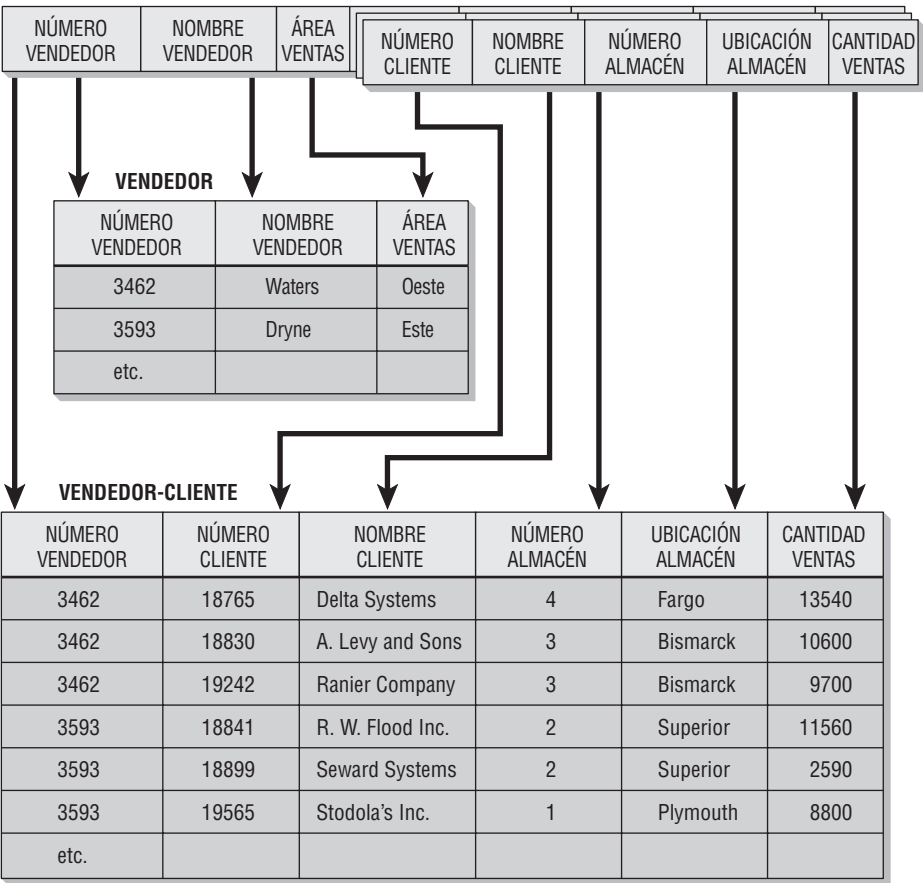


FIGURA 13.18

La relación sin normalizar original INFORME-VENTAS se divide en dos relaciones, VENDEDOR (3NF) y CLIENTE-VENDEDOR (1NF).

CANTIDAD-VENTAS). Sin embargo, debe saber que el NÚMERO-VENDEDOR no significa automáticamente lo que usted sabrá del NOMBRE-CLIENTE, CANTIDAD-VENTAS, UBICACIÓN-ALMACÉN, etc. En esta relación, alguien debe usar una clave concatenada (NÚMERO-VENDEDOR y NÚMERO-CLIENTE) para acceder al resto de la información. Es posible escribir las relaciones en notación abreviada como sigue:

VENDEDOR (NÚMERO-VENDEDOR, NOMBRE-VENDEDOR, ÁREA-VENTAS)

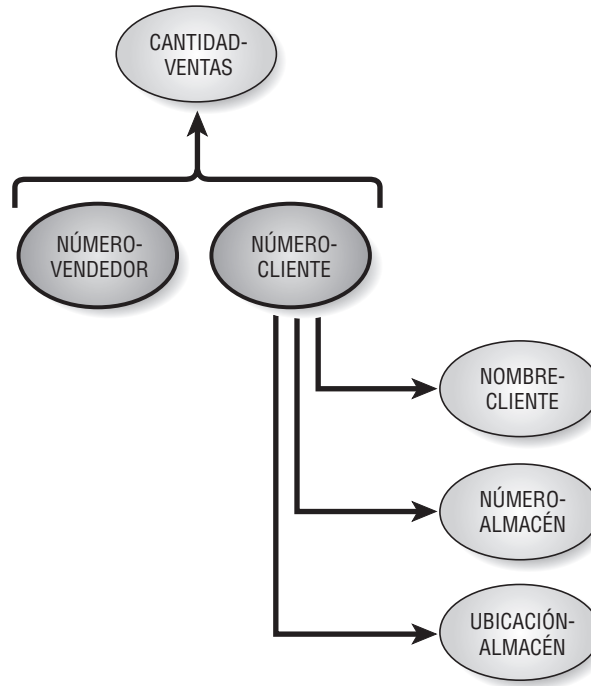
y

CLIENTE-VENDEDOR (NÚMERO-VENDEDOR,  
NÚMERO-CLIENTE,  
NOMBRE-CLIENTE,  
NÚMERO-ALMACÉN,  
UBICACIÓN-ALMACÉN,  
CANTIDAD-VENTAS)

La relación CLIENTE-VENDEDOR es una relación de primera forma normal, pero no está en su forma ideal. Los problemas surgen porque algunos de los atributos no son funcionalmente dependientes de la clave primaria (es decir, NÚMERO-VENDEDOR, NÚMERO-CLIENTE). En otras palabras, algunos de los atributos sin clave sólo son dependientes del NÚMERO DEL CLIENTE y no de la clave concatenada. El diagrama de modelo de datos de la figura 13.19 muestra que CANTIDAD-VENTAS es dependiente de NÚ-

**FIGURA 13.19**

Un diagrama de modelo de datos muestra que tres atributos son dependientes del NÚMERO-CLIENTE, de manera que la relación aún no se ha normalizado. El NÚMERO-VENDEDOR y el NÚMERO-CLIENTE se requieren para buscar la CANTIDAD-VENTAS.



NÚMERO-VENDEDOR y de NÚMERO-CLIENTE, pero los otros tres atributos sólo son dependientes de NÚMERO-CLIENTE.

**Segunda forma normal (2NF)** En la segunda forma normal, todos los atributos serán funcionalmente dependientes de la clave primaria. Por lo tanto, el próximo paso es quitar todos los atributos parcialmente dependientes y ponerlos en otra relación. La figura 13.20 muestra cómo la relación VENDEDOR-CLIENTE es dividida en dos nuevas relaciones: VENTAS y ALMACÉN-CLIENTE. Estas relaciones también se pueden expresar como sigue:

VENTAS    (NÚMERO-VENDEDOR, NÚMERO-CLIENTE,  
CANTIDAD-VENTAS)

y

CLIENTE-ALMACÉN    (NÚMERO-CLIENTE,  
NOMBRE-CLIENTE,  
NÚMERO-ALMACÉN,  
UBICACIÓN-ALMACÉN)

La relación ALMACÉN-CLIENTE está en la segunda forma normal. Ésta todavía se puede simplificar más porque en la relación hay dependencias adicionales. Algunos de los atributos sin clave son dependientes no sólo de la clave primaria, sino también de un atributo sin clave. Esta dependencia se denomina dependencia transitiva.

La figura 13.21 muestra las dependencias en la relación ALMACÉN-CLIENTE. Para que la relación sea de la segunda forma normal, todos los atributos deben depender de la clave primaria NÚMERO-CLIENTE, como se muestra en el diagrama. Sin embargo, UBICACIÓN-ALMACÉN evidentemente también es dependiente de NÚMERO-ALMACÉN. Para simplificar esta relación, se requiere otro paso.

**VENDEDOR-CLIENTE**

NÚMERO VENDEDOR	NÚMERO CLIENTE	NOMBRE CLIENTE	NÚMERO ALMACÉN	UBICACIÓN ALMACÉN	CANTIDAD VENTAS
-----------------	----------------	----------------	----------------	-------------------	-----------------

**CLIENTE-ALMACÉN**

NÚMERO CLIENTE	NOMBRE CLIENTE	NÚMERO ALMACÉN	UBICACIÓN ALMACÉN
18765	Delta Systems	4	Fargo
18830	A. Levy and Sons	3	Bismarck
19242	Ranier Company	3	Bismarck
18841	R. W. Flood Inc.	2	Superior
18899	Seward Systems	2	Superior
19565	Stodola's Inc.	1	Plymouth
etc.			

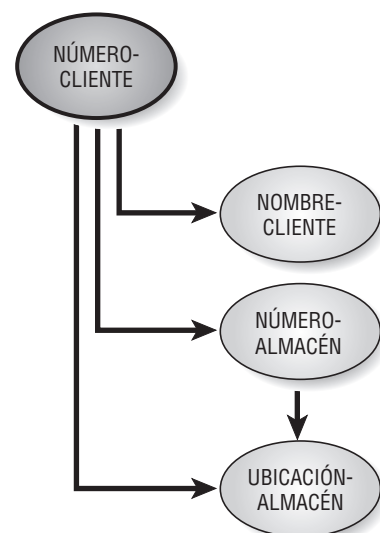
**VENTAS**

NÚMERO VENDEDOR	NÚMERO CLIENTE	CANTIDAD VENTAS
3462	18765	13540
3462	18830	10600
3462	19242	9700
3593	18841	11560
3593	18899	2590
3593	19565	8800
etc.		

**FIGURA 13.20**

La relación VENDEDOR-CLIENTE se divide en una relación llamada CLIENTE-ALMACÉN (2NF) y en una relación llamada VENTAS (1NF).

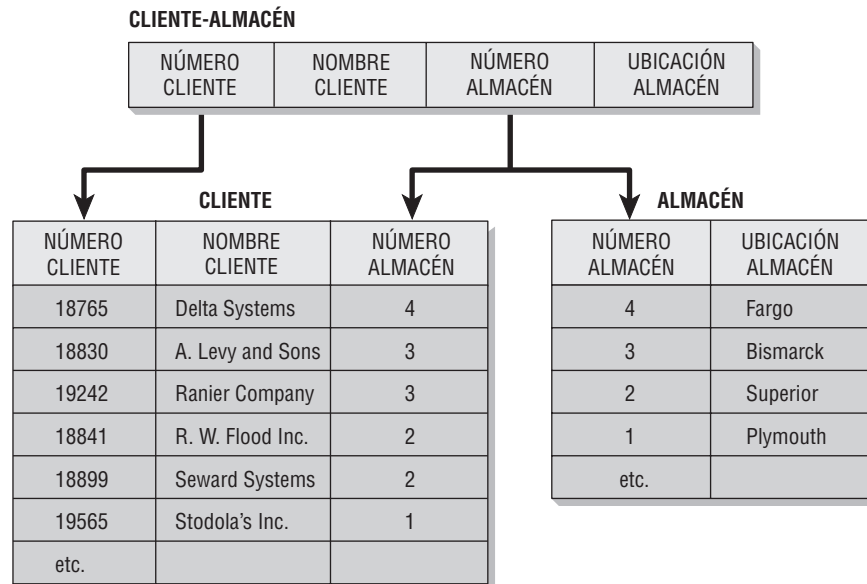
**Tercera forma normal (3NF)** Una relación normalizada está en tercera forma normal si todos los atributos sin clave son funcionalmente dependientes por completo de la clave primaria y si no hay dependencias transitivas (sin claves). De forma similar a los pasos anteriores, es posible dividir la relación ALMACÉN-CLIENTE en dos relaciones, como se muestra en la figura 13.22.

**FIGURA 13.21**

Un diagrama de modelo de datos muestra que entre el NÚMERO-ALMACÉN y la UBICACIÓN-ALMACÉN existe una dependencia transitiva.

**FIGURA 13.22**

La relación CLIENTE-ALMACÉN se divide en dos relaciones llamadas CLIENTE (1NF) y ALMACÉN (1NF).



Las dos nuevas relaciones se llaman CLIENTE y ALMACÉN y se pueden escribir como sigue:

CLIENTE    (NÚMERO-CLIENTE, NOMBRE-CLIENTE,  
                  NÚMERO-ALMACÉN)

y

ALMACÉN    (NÚMERO-ALMACÉN,  
                  UBICACIÓN-ALMACÉN)

La clave primaria para la relación CLIENTE es NÚMERO-CLIENTE, y la clave primaria para la relación ALMACÉN es NÚMERO-ALMACÉN.

Además de estas claves primarias, podemos identificar el NÚMERO-ALMACÉN para ser una clave externa en la relación CLIENTE. Una clave extranjera es cualquier atributo que no tiene clave en una relación pero es una clave primaria en otra. Designamos el NÚMERO-ALMACÉN como una clave externa en la notación anterior y en las figuras subrayándolo con una línea punteada: \_ \_ \_ \_ \_.

Finalmente, la relación sin normalizar original INFORME-VENTAS se ha transformado en la cuarta relación de tercera forma normal (3NF). Al repasar las relaciones que se muestran en la figura 13.23, uno puede ver que la relación sencilla INFORME-VENTAS se transformó en las siguientes cuatro relaciones:

VENDEDOR    (NÚMERO-VENDEDOR, NOMBRE-VENDEDOR,  
                  ÁREA-VENTAS)

VENTAS        (NÚMERO-VENDEDOR, NÚMERO-CLIENTE,  
                  CANTIDAD-VENTAS)

CLIENTE      (NÚMERO-CLIENTE, NOMBRE-CLIENTE,  
                  NÚMERO-ALMACÉN)

y

ALMACÉN      (NÚMERO-ALMACÉN,  
                  UBICACIÓN-ALMACÉN)

La tercera forma normal es adecuada para la mayoría de los problemas de diseño de bases de datos. La simplificación lograda al transformar una relación sin normalizar en un juego

**VENDEDOR**

NÚMERO VENDEDOR	NOMBRE VENDEDOR	ÁREA VENTAS
3462	Waters	Oeste
3593	Dryne	Este
etc.		

**VENTAS**

NÚMERO VENDEDOR	NÚMERO CLIENTE	CANTIDAD VENTAS
3462	18765	13540
3462	18830	10600
3462	19242	9700
3593	18841	11560
3593	18899	2590
3593	19565	8800
etc.		

**FIGURA 13.23**

La base de datos completa consiste de cuatro relaciones 1NF llamadas VENDEDOR, VENTAS, CLIENTE y ALMACÉN.

**CLIENTE**

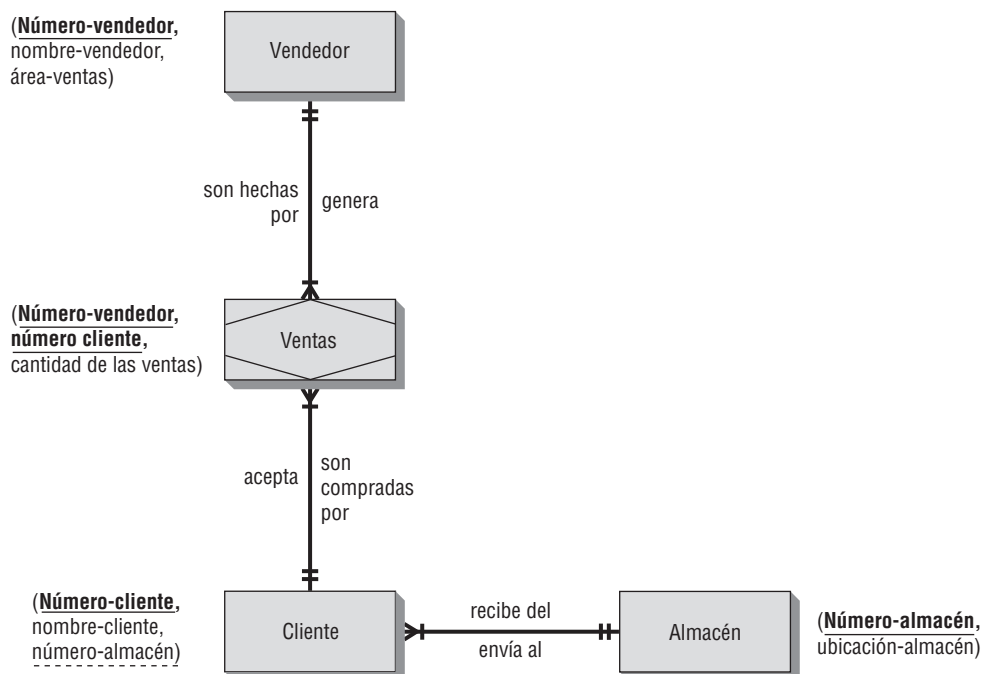
NÚMERO CLIENTE	NOMBRE CLIENTE	NÚMERO ALMACÉN
18765	Delta Systems	4
18830	A. Levy and Sons	3
19242	Ranier Company	3
18841	R. W. Flood Inc.	2
18899	Seward Systems	2
19565	Stodola's Inc.	1
etc.		

**ALMACÉN**

NÚMERO ALMACÉN	UBICACIÓN ALMACÉN
4	Fargo
3	Bismarck
2	Superior
1	Plymouth
etc.	

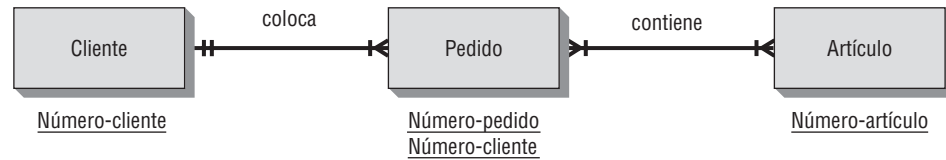
de relaciones 3NF es un gran beneficio cuando llega el momento de insertar, eliminar y actualizar la información en la base de datos.

En la figura 13.24 se muestra un diagrama entidad-relación para la base de datos. Un VENDEDOR sirve a muchos CLIENTES, quien genera VENTAS y recibe sus artículos de un ALMACÉN (el ALMACÉN más cercano a su ubicación). Tome el tiempo para observar cómo se relacionan las entidades y los atributos a la base de datos.

**FIGURA 13.24**

Un diagrama entidad-relación para la base de datos de Al S. Well Hydraulic Company.

Un diagrama entidad-relación para los pedidos del cliente.



## USO DEL DIAGRAMA ENTIDAD-RELACIÓN PARA DETERMINAR LAS CLAVES DEL REGISTRO

El diagrama entidad-relación se podría usar para determinar las claves necesarias para una relación de un registro o de una base de datos. El primer paso es construir el diagrama entidad-relación y etiquetar una clave (principal) única para cada entidad de datos. La figura 13.25 muestra un diagrama entidad-relación para un sistema de pedido de cliente. Hay tres entidades de datos: CLIENTE, con una clave primaria de NÚMERO-CLIENTE; PEDIDO, con una clave primaria de NÚMERO-PEDIDO, y ARTÍCULO, con NÚMERO-ARTÍCULO como la clave primaria. Un CLIENTE podría hacer muchos pedidos, pero cada PEDIDO podría hacerse por un solo CLIENTE, de modo que la relación es uno a muchos. Cada PEDIDO podría contener muchos ARTÍCULOS, y cada ARTÍCULO podría estar contenido en muchos PEDIDOS, de manera que la relación del ARTÍCULO-PEDIDO es de muchos a muchos.

Sin embargo, una clave externa es un campo de datos en un archivo dado que es la clave primaria de un archivo maestro diferente. Por ejemplo, un NÚMERO-DEPARTAMENTO que indica la especialidad de un estudiante podría existir en la tabla MAESTRO DE ESTUDIANTES. El NÚMERO-DEPARTAMENTO también podría ser la única clave para la tabla MAESTRO DE DEPARTAMENTOS.

## RELACIÓN UNO A MUCHOS

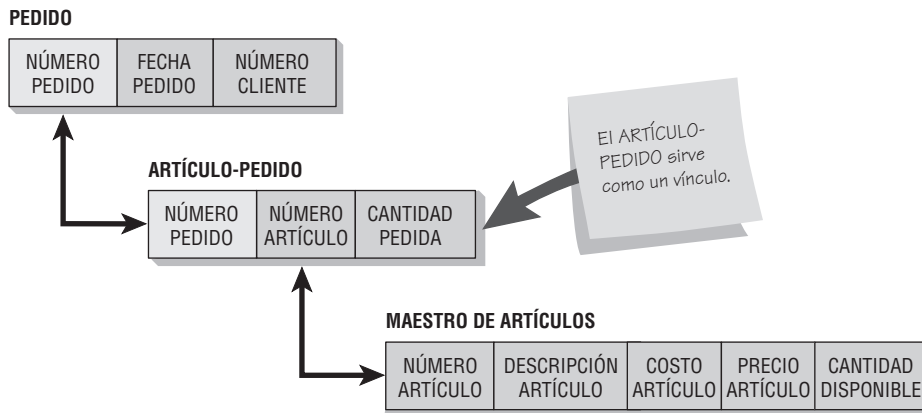
Una tabla de base de datos no puede contener un grupo repetitivo o tabla, pero podría tener un archivo tradicional indexado de forma secuencial. El archivo en el extremo *muchos* podría tener claves externas almacenadas en una tabla dentro del archivo en el extremo *uno*. Por ejemplo, el MAESTRO DE CLIENTES podría diseñarse para contener una tabla de números de pedidos sobresalientes. La desventaja de usar dicha tabla del registro es que todas las entradas de la tabla se podrían completar con números de pedidos, y después el analista se enfrenta a la decisión de tener que extender la tabla (lo cual es costoso y requiere tiempo) o de perder algunos datos de la tabla.

## RELACIÓN MUCHOS A MUCHOS

Quando la relación es de muchos a muchos, se necesitan tres tablas: una para cada entidad de datos y otra para la relación. Las entidades PEDIDO y ARTÍCULO de nuestro ejemplo tienen una relación muchos a muchos. La clave primaria de cada entidad de datos se almacena como una clave externa de la tabla relacional. Esta última podría contener simplemente las claves primarias para cada entidad de datos o podría contener datos adicionales, tales como la calificación recibida de un curso o la cantidad de un artículo pedido. Consulte el diseño de tabla ilustrado en la figura 13.26. La tabla ARTÍCULO DEL PEDIDO contiene información acerca de qué artículos contiene el pedido, y proporciona un vínculo entre la tabla PEDIDO y la tabla MAESTRO DE ARTÍCULOS.

La tabla de relación se debe indexar en cada una de las claves externas —una para cada una de las tablas de la relación— y podría tener una clave primaria consistente en una combinación de las dos claves externas. Para encontrar muchos registros en una segunda tabla dada la primera, lea directamente la tabla relacional para la clave deseada. Localice el registro de coincidencia en la segunda tabla *muchos*. Continúe buscando en la tabla relacional hasta que ya no se encuentre la clave deseada. Por ejemplo, para encontrar registros en el MAESTRO DE ARTÍCULOS para un registro específico en la tabla PEDIDO, lea directamente la tabla ARTÍCULO DEL PEDIDO usando NÚMERO-PEDIDO como el índice. Los registros están en secuencia de forma lógica basados en los datos del índice, de modo





**FIGURA 13.26**

Cuando la relación es muchos a muchos, se necesitan tres archivos.

que se agrupan todos los registros para el mismo NÚMERO-PEDIDO. Para cada registro de ARTÍCULO DEL PEDIDO que hace coincidir el NÚMERO-PEDIDO deseado, lea directamente la tabla MAESTRO DE ARTÍCULOS usando NÚMERO-ARTÍCULO como un índice.

La lógica es la misma para una situación inversa, tal como encontrar todos los pedidos para un artículo de una nueva orden de pedido que se ha recibido. Use el NÚMERO-ARTÍCULO deseado para leer directamente la tabla ARTÍCULO-PEDIDO. El índice de ARTÍCULO DEL PEDIDO se establece para el NÚMERO-ARTÍCULO. Para todos los registros de coincidencia de ARTÍCULO DEL PEDIDO, use el NÚMERO-PEDIDO para leer directamente la tabla PEDIDO. Finalmente, lea directamente la tabla MAESTRO DE CLIENTES para obtener el NOMBRE-CLIENTE y la DIRECCIÓN usando el NÚMERO-CLIENTE en la tabla PEDIDO.

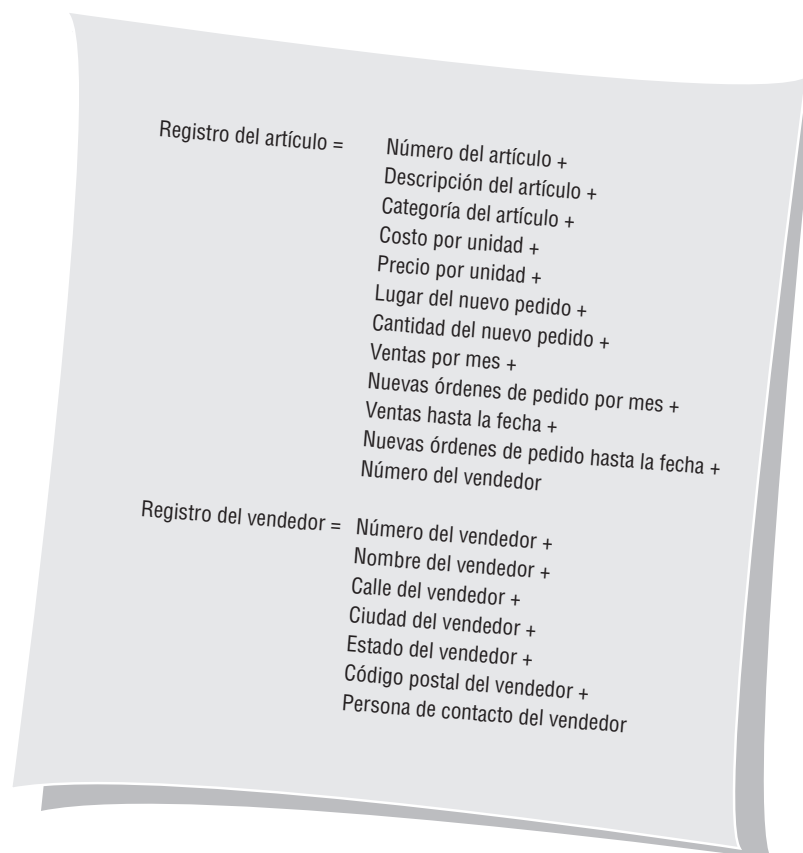
## LINEAMIENTOS PARA EL DISEÑO DE RELACIÓN ARCHIVO MAESTRO/BASE DE DATOS

Al diseñar relaciones de archivos maestros o bases de datos se deben tomar en cuenta los siguientes lineamientos:

1. Cada entidad de datos separada debe crear una tabla maestra de base de datos. No combine dos entidades distintas en un archivo. Por ejemplo, los artículos se compran de los vendedores. La tabla MAESTRO DE ARTÍCULOS sólo debe contener información del artículo y la tabla MAESTRO DE VENDEDORES sólo debe contener información del vendedor. La figura 13.27 ilustra el diccionario de datos para las tablas MAESTRO DE ARTÍCULOS y MAESTRO DE VENDEDORES.
2. Un campo de datos específico sólo debe existir en una tabla maestra. Por ejemplo, el NOMBRE DEL CLIENTE sólo debe existir en la tabla MAESTRO DE CLIENTES, no en la tabla PEDIDO o en cualquier otra tabla maestra. Las excepciones a este lineamiento son la clave o los campos de índice, los cuales podrían estar en tantas tablas como sea necesario. Si un informe o pantalla necesita información de muchas tablas, los índices deben proporcionar la vinculación para obtener los registros necesarios.
3. Cada tabla maestra o relación de la base de datos debe tener programas para Crear, Leer, Actualizar y Eliminar (abreviado CLAE) los registros. En teoría, un solo programa debe agregar nuevos registros y un solo programa debe eliminar registros específicos. Sin embargo, muchos programas podrían ser responsables de cambiar campos de datos en el curso de actividades normales del negocio. Por ejemplo, un archivo MAESTRO DE CLIENTES podría tener un campo SALDO ACTUAL que se incrementa por el TOTAL PEDIDO en el programa de procesamiento del pedido y disminuye por una CANTIDAD A PAGAR o una CANTIDAD DEVUELTA de dos programas adicionales.

**FIGURA 13.27**

Archivos maestros del artículo mejorado y del vendedor.



### RESTRICCIONES DE INTEGRIDAD

Las restricciones de integridad son reglas que controlan el cambio y eliminación de registros, y ayuda a mantener los datos en la base de datos exacta. En una base de datos se aplican tres tipos de restricciones de integridad:

1. Integridad de identidad.
2. Integridad referencial.
3. Integridad de dominio.

Las restricciones de integridad son reglas que controlan la composición de reglas principales. La clave primaria no puede tener un valor nulo, y si la clave primaria es una clave compuesta, ninguno de los campos de componente en la clave puede contener un valor nulo. Algunas bases de datos le permiten definir una restricción única o una clave única. Esta última identifica un solo registro, el cual no es una clave primaria. La diferencia entre una clave única y una clave primaria, es que la primera podría tener un valor nulo.

La integridad referencial controla la naturaleza de los registros en una relación uno a muchos. La tabla que se conecta en el extremo uno de la relación se llama padre. La tabla que se conecta en el extremo muchos de la relación se llama tabla hija. La integridad referencial significa que todas las claves externas de la tabla muchos (la tabla hija) deben tener un registro de coincidencia en la tabla padre. Por lo tanto, no puede agregar un registro en la tabla (muchos) hija sin un registro de coincidencia en la tabla padre.

Una segunda implicación es que no puede cambiar una clave primaria que tiene registros de coincidencia de la tabla hija. Si pudiera cambiar el registro padre, el resultado sería un registro hijo que tendría un registro padre diferente o un registro huérfano, o un registro hijo sin un registro padre. Los ejemplos son un registro de CALIFICACIONES para un estudiante que no estaría en la tabla MAESTRO DE ESTUDIANTES y un registro de PEDIDO

para un NÚMERO DEL CLIENTE que no existió. La última implicación de la integridad referencial es que no puede eliminar un registro padre que tenga registros hijos. Esto también se aplicaría para los registros huérfanos mencionados anteriormente.

La integridad referencial se implementa de dos formas diferentes. Una forma es tener una base de datos restringida, en la cual el sistema pueda actualizar o eliminar un registro padre sólo si no hay registros hijos de coincidencia. Una base de datos en cascada eliminará o actualizará todos los registros hijos al eliminar o cambiar un registro padre (el padre activa los cambios).

Una relación restringida es buena cuando se cambian los registros. ¡No querría eliminar un registro del cliente ni tampoco todas las facturas a pagar! El enfoque en cascada es mejor cuando se cambian los registros. Si se cambia la clave primaria de un registro del estudiante, también se deben cambiar las claves externas de todos los registros del curso para dicho estudiante (el NÚMERO DEL ESTUDIANTE en el MAESTRO DE CURSOS).

Las reglas de integridad de dominio se usan para validar los datos, tales como la tabla, límite, rango y otras marcas de validación. Éstos se explican con mayor detalle en el capítulo 15. Normalmente las reglas de integridad de dominio se almacenan en la estructura de base de datos de una o dos formas. Las restricciones de verificación se definen en el nivel de la tabla y pueden consultar uno o más campos en la tabla. Un ejemplo es que la FECHA DE LA COMPRA siempre es menor o igual a la fecha actual. Las reglas se definen al nivel de la base de datos como objetos separados y se pueden usar con varios campos. Un ejemplo es un valor que es mayor a cero, usado para validar varios elementos.

---

## USO DE LA BASE DE DATOS

Hay varios pasos que deben seguir un orden secuencial para asegurar que la base de datos será útil para presentar los datos.

### PASOS EN LA RECUPERACIÓN Y PRESENTACIÓN DE DATOS

Hay ocho pasos en la recuperación y presentación de datos:

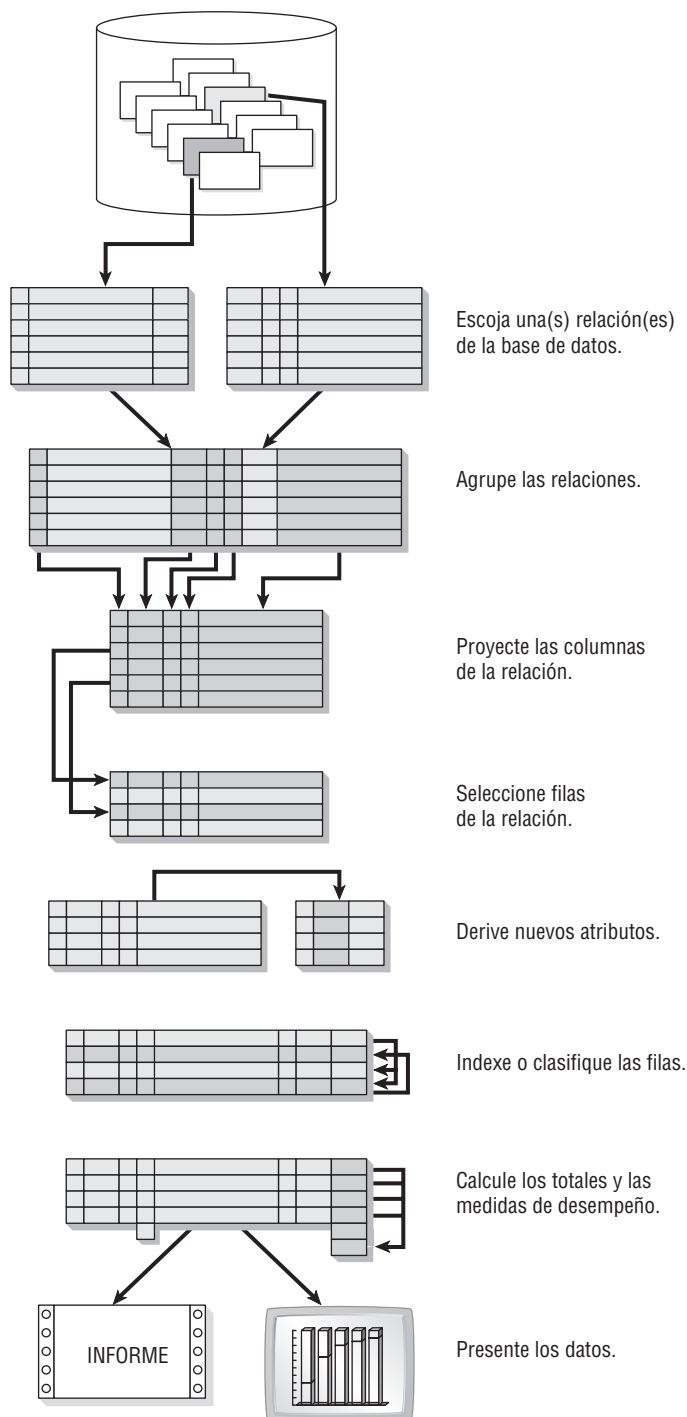
1. Escoja una relación de la base de datos.
2. Una dos relaciones.
3. Proyecte las columnas de la relación.
4. Seleccione filas de la relación.
5. Derive nuevos atributos.
6. Indexe o clasifique las filas.
7. Calcule los totales y medidas de desempeño.
8. Presente los datos.

El primer y último pasos son obligatorios, pero los seis pasos intermedios son opcionales, dependiendo de cómo se usen los datos. La figura 13.28 es una guía visual para los pasos que se describen en las siguientes subsecciones.

**Escoja una relación de la base de datos** El primer y obvio paso es escoger una relación de la base de datos. Una buena forma de realizar este paso es llevar un directorio de las vistas de usuario como auxiliar para su memoria. Aun cuando el usuario quiere una consulta *ad hoc*, es útil tener disponibles vistas similares.

**FIGURA 13.28**

Los datos se recuperan y presentan en ocho pasos distintos.



**Agrupe dos relaciones** La acción de unir se piensa como tomar dos relaciones y agruparlas para hacer una relación más grande. Para que dos relaciones se unan, deben tener un atributo en común. Por ejemplo, tome dos relaciones de nuestro ejercicio:

CLIENTE (NÚMERO-CLIENTE, NOMBRE-CLIENTE,  
NÚMERO-ALMACÉN)

y

ALMACÉN (NÚMERO-ALMACÉN,  
UBICACIÓN-ALMACÉN)

**CLIENTE**

NÚMERO CLIENTE	NOMBRE CLIENTE	NÚMERO ALMACÉN
18765	Delta Systems	4
18830	A. Levy and Sons	3
19242	Ranier Company	3
18841	R. W. Flood Inc.	2
18899	Seward Systems	2
19565	Stodola's Inc.	1
etc.		

**ALMACÉN**

NÚMERO ALMACÉN	UBICACIÓN ALMACÉN
4	Fargo
3	Bismarck
2	Superior
1	Plymouth
etc.	

Unión

**CLIENTE-ALMACÉN-UBICACIÓN**

NÚMERO CLIENTE	NOMBRE CLIENTE	NÚMERO ALMACÉN	UBICACIÓN ALMACÉN
18765	Delta Systems	4	Fargo
18830	A. Levy and Sons	3	Bismarck
19242	Ranier Company	3	Bismarck
18841	R. W. Flood Inc.	2	Superior
18899	Seward Systems	2	Superior
19565	Stodola's Inc.	1	Plymouth
etc.			

**FIGURA 13.29**

La operación de unir toma dos relaciones y las agrupa para formar una sola relación.

Suponga que unimos estas relaciones al NÚMERO-ALMACÉN para obtener una nueva relación, el CLIENTE-ALMACÉN-UBICACIÓN. En la figura 13.29 se ilustra la unión de estas relaciones. También observe que la nueva relación no es 3NF.

La acción de unir también podría ir un paso más allá; es decir, podría combinar los archivos para las filas que tienen un atributo que se encuentra en una cierta condición. La figura 13.30 muestra un ejemplo en el cual dos relaciones, VENTAS y CUOTA, se unen para satisfacer la condición que se ha encontrado un vendedor o ha excedido las cuotas predeterminadas.

La acción de unir es importante porque puede tomar muchas relaciones 3NF y combinarlas para hacer una relación más útil. Junto con las acciones de abajo, unir es una acción poderosa.

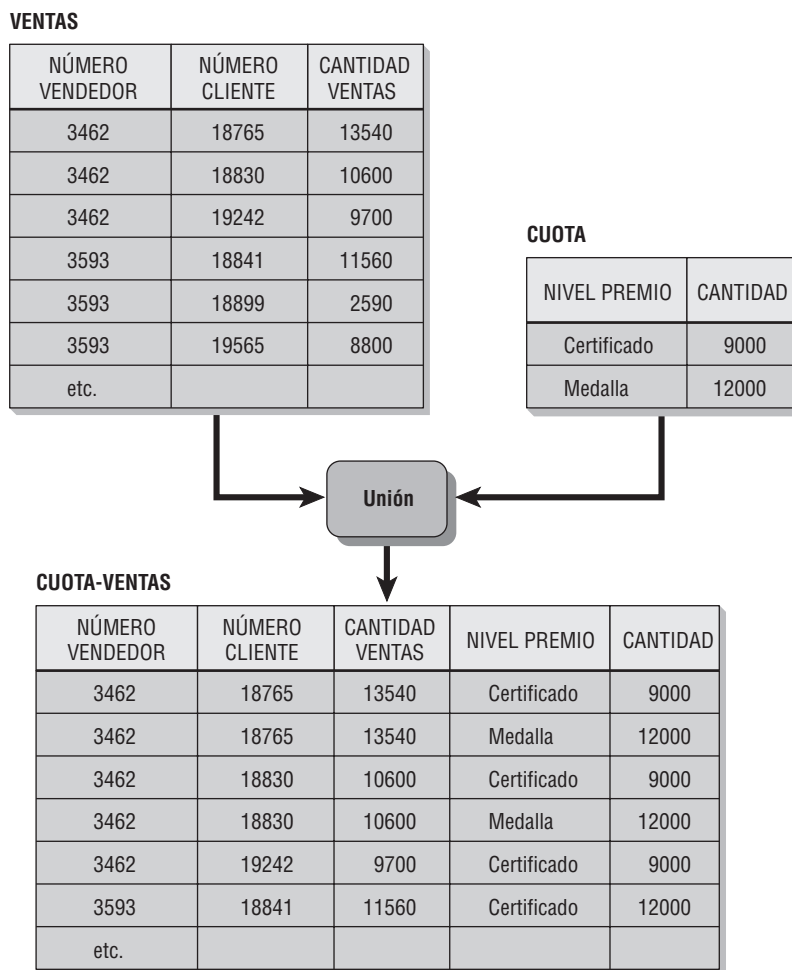
**Proyecte las columnas de la relación** Proyección es el proceso de construir una relación más pequeña escogiendo únicamente atributos relevantes de una relación existente. En otras palabras, proyección es la extracción de ciertas columnas de una tabla relacional.

En la figura 13.31 se presenta un ejemplo de proyección. La relación

CLIENTE-ALMACÉN-UBICACIÓN (NÚMERO-CLIENTE,  
NOMBRE-CLIENTE,  
NÚMERO-ALMACÉN),  
UBICACIÓN-ALMACÉN

**FIGURA 13.30**

Las relaciones se pueden unir bajo ciertas condiciones.



se proyecta en NÚMERO-CLIENTE y UBICACIÓN-ALMACÉN, y durante el proceso de la proyección, se remueven los registros duplicados.

**Selección de filas de la relación** La acción denominada selección es parecida a la proyección, pero en lugar de extraer las columnas, extrae las filas. La selección crea una relación (más pequeña) nueva mediante la extracción de registros que contienen un atributo que coincide con una cierta condición.

La figura 13.32 muestra cómo funciona la acción de selección. La selección se desempeña en la relación PERSONAL para extraer sólo a los empleados asalariados. Aquí no es necesario remover los registros duplicados, como se hizo en el ejemplo anterior de la proyección.

La selección también se podría desempeñar para un juego más complejo de condiciones, tal como seleccionar a todos los empleados que están asalariados y que ganan más de \$40,000 al año, o seleccionar empleados a quienes se les paga por hora y que ganan más de \$15.00 por hora. La selección es una acción importante para las consultas *ad hoc*.

**Derive nuevos atributos** El quinto paso involucra la manipulación de los datos existentes además de algunos parámetros adicionales (si son necesarios) para derivar los nuevos datos. Se crean nuevas columnas para la relación resultante. En la figura 13.33 se puede encontrar un ejemplo de la derivación de nuevos atributos. Aquí, se determinan dos atributos nuevos: (1) CIRCUNFERENCIA (multiplicar la suma de ancho y alto por 2 y agregarlo a la longitud) y (2) PESO-ENVÍO (el cual depende de la circunferencia).

**CLIENTE-ALMACÉN-UBICACIÓN**

NÚMERO CLIENTE	NOMBRE CLIENTE	NÚMERO ALMACÉN	UBICACIÓN ALMACÉN
18765	Delta Systems	4	Fargo
18830	A. Levy and Sons	3	Bismarck
19242	Ranier Company	3	Bismarck
18841	R. W. Flood Inc.	2	Superior
18899	Seward Systems	2	Superior
19565	Stodola's Inc.	1	Plymouth
etc.			

Proyección

**CLIENTE-UBICACIÓN**

NÚMERO CLIENTE	UBICACIÓN ALMACÉN
18765	Fargo
18830	Bismarck
19242	Bismarck
18841	Superior
18899	Superior
19565	Plymouth
etc.	

**FIGURA 13.31**

La proyección crea una relación pequeña al seleccionar sólo atributos relevantes (columnas) de la relación.

**Indexación o clasificación de las filas** Las personas necesitan que los datos se organicen en un cierto orden de modo que puedan localizar los artículos con mayor facilidad en una lista o un grupo y que puedan calcular subtotales más fácil. Están disponibles dos opciones para ordenar datos: indexación y clasificación.

La indexación es el orden lógico de las filas en una relación de acuerdo con alguna clave. Como se discutió en la sección anterior, el indicador lógico ocupa espacio, y listar la relación usando un índice es más lento que si la relación estuviera en el orden físico apropiado. Sin embargo, el índice ocupa mucho menos espacio que un archivo duplicado.

La clasificación es el orden lógico de una relación. El resultado de una clasificación física es un archivo secuencial como se discutió en el capítulo anterior. La figura 13.34 ilustra la indexación y clasificación de la relación PERSONAL por el nombre del empleado en orden alfabético.

**Cálculo de los totales y de las medidas de desempeño** Una vez que se define el subconjunto de datos apropiado y que las filas de la relación se ordenan de la forma requerida, se pueden calcular los totales y medidas de desempeño. La figura 13.35 muestra cómo se realiza el cálculo.

**Presentación de los datos al usuario** El último paso en la recuperación de datos es la presentación. La presentación de los datos abstraídos de la base de datos puede tomar muchas formas. Algunas veces los datos se presentarán en forma tabular, algunas veces en gráficos y otras como una respuesta de una sola palabra en una pantalla. El diseño de salida, como se trató en el capítulo 11, proporciona una apariencia más detallada en cuanto a los objetivos de la presentación, formularios y métodos.

**FIGURA 13.32**

La selección extrae los registros pertinentes (filas) de la relación.

**PERSONAL**

NÚMERO	NOMBRE EMPLEADO	DEPARTAMENTO	S/H	SUELDO BRUTO
72845	Waters	Ventas por fuera	S	48960
72888	Dryne	Ventas por fuera	S	37200
73712	Fawcett	Distribución	H	23500
80345	Well, Jr.	Marketing	S	65000
84672	Piper	Mantenimiento	H	20560
89760	Acquia	Contabilidad	H	18755
etc.				

Selección

**EMPLEADOS-ASALARIADOS**

NÚMERO	NOMBRE EMPLEADO	DEPARTAMENTO	S/H	SUELDO BRUTO
72845	Waters	Ventas por fuera	S	48960
72888	Dryne	Ventas por fuera	S	37200
80345	Well, Jr.	Marketing	S	65000
etc.				

## DESNORMALIZACIÓN

Una de las razones principales para la normalización es organizar los datos para reducir los datos redundantes. Si no se le pide almacenar los mismos datos una y otra vez, puede ahorrar mucho espacio. Dicha organización permite al analista reducir la cantidad necesaria de almacenamiento, algo muy importante cuando el almacenamiento era caro.

En la última sección aprendimos que para usar los datos normalizados teníamos que progresar en una serie de pasos que involucra unir, clasificar y resumir. Cuando acelera la consulta de base de datos (es decir, hacer una pregunta y requerir una respuesta rápida) es crítico, podría ser importante almacenar los datos de otras formas.

La desnormalización es el proceso de tomar el modelo de datos lógicos y transformarlo en un modelo físico que es eficaz para las tareas más comunes. Estas tareas pueden incluir generación de informes, pero también pueden significar consultas más eficaces. Las consultas complejas tales como el proceso analítico en línea (OLAP), así como también la minería de datos y los procesos de descubrimiento de datos del conocimiento (KDD), también pueden usar las bases de datos denormalizadas.

La desnormalización se puede lograr de varias formas diferentes. La figura 13.36 describe algunos de estos enfoques. Primero, podemos tomar una relación muchos a muchos, tal como el de VENDEDOR y CLIENTE, la cual comparte las entidades asociativas de VENTAS. Al combinar los atributos de VENDEDOR y VENTAS podemos evitar uno de los procesos de la unión. Esto podría producir una cantidad considerable de duplicidad de datos, pero hace las consultas sobre los modelos de las ventas más eficaz.

Otra razón para la desnormalización es evitar la referencia repetida para una tabla de búsqueda. Podría ser más eficaz repetir la misma información —por ejemplo, la ciudad, el es-



#### INFORMACIÓN-PAQUETE

NÚMERO PAQUETE	ANCHO	ALTO	TAMAÑO	PESO
A3456	4	3	26	4
A3457	12	12	20	10
A3458	10	20	34	20
A3459	15	15	22	18
A3460	10	10	40	40
A3461	10	20	34	22
A3462	5	10	15	30
A3463	8	14	44	35

$$\text{CIRCUNFERENCIA} = 2 (\text{ANCHO} + \text{ALTO}) + \text{TAMAÑO}$$

Derivación

SI CIRCUNFERENCIA > 84 Y PESO < 25  
ENTONCES PESO DE ENVÍO = 25  
SI-NO PESO DE ENVÍO = PESO

#### INFORMACIÓN-ENVÍO-PAQUETE

NÚMERO PAQUETE	ANCHO	ALTO	TAMAÑO	PESO	CIRCUNFERENCIA	PESO ENVÍO
A3456	4	3	26	4	50	4
A3457	12	12	20	10	68	10
A3458	10	20	34	20	94	25
A3459	15	15	22	18	82	18
A3460	10	10	40	40	80	40
A3461	10	20	34	22	90	25
A3462	5	10	15	30	45	30
A3463	8	14	44	35	88	35

**FIGURA 13.33**

La derivación crea nuevos atributos (columnas) en la relación al manipular los datos contenidos en atributos existentes.

tado y el código postal— aun cuando esta información normalmente se puede almacenar sólo como un código postal. Por lo tanto, en el ejemplo de las ventas, se podrían combinar CLIENTE y ALMACÉN.

Finalmente, veremos relaciones uno a uno porque es muy probable que se combinen por razones prácticas. Si entendemos que muchas de las consultas con respecto a los pedidos también se interesan en cómo fue enviado el pedido, tendría sentido para combinar, o denormalizarse. Por lo tanto, en el ejemplo, algunos de los detalles pueden aparecer en DETALLES-PEDIDO y DETALLES-ENVÍO cuando vemos la desnormalización.

## ALMACENES DE DATOS

Los almacenes de datos difieren de las bases de datos tradicionales. El propósito de un almacén de datos es organizar la información para consultas rápidas y eficaces. De hecho, almacenan datos denormalizados, pero van un paso más adelante. Dichos almacenes organizan los datos en torno a los temas. En la mayoría de los casos, un almacén de datos es más que una base de datos procesada de manera que los datos se representen de formas uniformes. Por consiguiente, los datos almacenados en los almacenes de datos provienen de diferentes fuentes, normalmente de bases de datos que se establecen para diferentes propósitos.

**FIGURA 13.34**

La operación ordena los pedidos de los registros (filas) en la relación de manera que los registros se puedan desplegar en orden y se puedan agrupar por los subtotales. Aquí la relación PERSONAL se ordena alfabéticamente de acuerdo con el NOMBRE DEL EMPLEADO.

**PERSONAL**

NÚMERO	NOMBRE EMPLEADO	DEPARTAMENTO	S/H	SUELDO BRUTO
72845	Waters	Ventas por fuera	S	48960
72888	Dryne	Ventas por fuera	S	37200
73712	Fawcett	Distribución	H	23500
80345	Well, Jr.	Marketing	S	65000
84672	Piper	Mantenimiento	H	20560
89760	Acquia	Contabilidad	H	18755

Orden

**POR-NOMBRE-EMPLEADOS**

NÚMERO	NOMBRE EMPLEADO	DEPARTAMENTO	S/H	SUELDO BRUTO
89760	Acquia	Contabilidad	H	18755
72888	Dryne	Ventas por fuera	S	37200
73712	Fawcett	Distribución	H	23500
84672	Piper	Mantenimiento	H	20560
72845	Waters	Ventas por fuera	S	48960
80345	Well, Jr.	Marketing	S	65000

El concepto de almacén de datos es único. Las diferencias entre los almacenes de datos y las bases de datos tradicionales incluyen lo siguiente:

1. En un almacén de datos, los datos se organizan en torno a los temas principales en lugar de transacciones individuales.
2. En un almacén de datos los datos normalmente se almacenan como datos resumidos en lugar de detallados, los datos básicos se encuentran en una base de datos de transacciones.
3. En un almacén de datos los datos cubren un periodo más largo que los datos en una base de datos tradicional orientada a transacciones porque las consultas normalmente involucran toma de decisiones a largo plazo en lugar de los detalles de transacción diarios.
4. La mayoría de los almacenes de datos se organizan para consultas más rápidas, mientras que las bases de datos más tradicionales se normalizan y estructuran de tal manera que proporcionen almacenamiento eficaz de información.
5. Los almacenes de datos normalmente se optimizan para responder consultas complejas, conocidas como OLAP, de gerentes y analistas, en lugar de consultas hechas de forma simple y repetida.
6. Los almacenes de datos permiten acceso fácil mediante software de minería de datos (denominado software) que busca modelos y puede identificar relaciones no imaginadas por los tomadores de decisiones humanos.
7. Los almacenes de datos no incluyen una sola sino muchas bases de datos que se han procesado para que los datos del almacén se definan uniformemente. Estas bases de datos se denominan datos limpios.
8. Los almacenes de datos normalmente incluyen datos de fuentes externas (tales como un informe de industria, los reportes de la empresa para el Gobierno o incluso la información acerca de los productos de competidores), así como también generaron datos para uso interno.

## PESO-ENVÍO

NÚMERO PAQUETE	ANCHO	ALTO	TAMAÑO	PESO	CIRCUNFERENCIA	PESO ENVÍO
A3456	4	3	26	4	50	4
A3457	12	12	20	10	68	10
A3458	10	20	34	20	94	25
A3459	15	15	22	18	82	18
A3460	10	10	40	40	80	40
A3461	10	20	34	22	90	25
A3462	5	10	15	30	45	30
A3463	8	14	44	35	88	35

Cálculo

### Algunos ejemplos:

Circunferencia total y peso del envío

597 187

Número de paquetes que pesaron menos de 25 libras pero que fueron enviados a una tasa de 25 libras

2

Número total de paquetes enviados

8

Porcentaje de paquetes que pesaron menos de 25 libras pero que fueron enviados a una tasa de 25 libras

25

**FIGURA 13.35**

El cálculo proporciona los subtotales, totales y otras medidas de desempeño.

Construir un almacén de datos es una gran tarea. El analista necesita recopilar los datos de una variedad de fuentes y convertirlos a una forma común. Por ejemplo, una base de datos podría almacenar información acerca del género como “Masculino” y “Femenino”, otra la podría almacenar como “M” y “F” y una tercera la podría almacenar como “1” y “0”. El analista necesita establecer un estándar y convertir todos los datos al mismo formato.

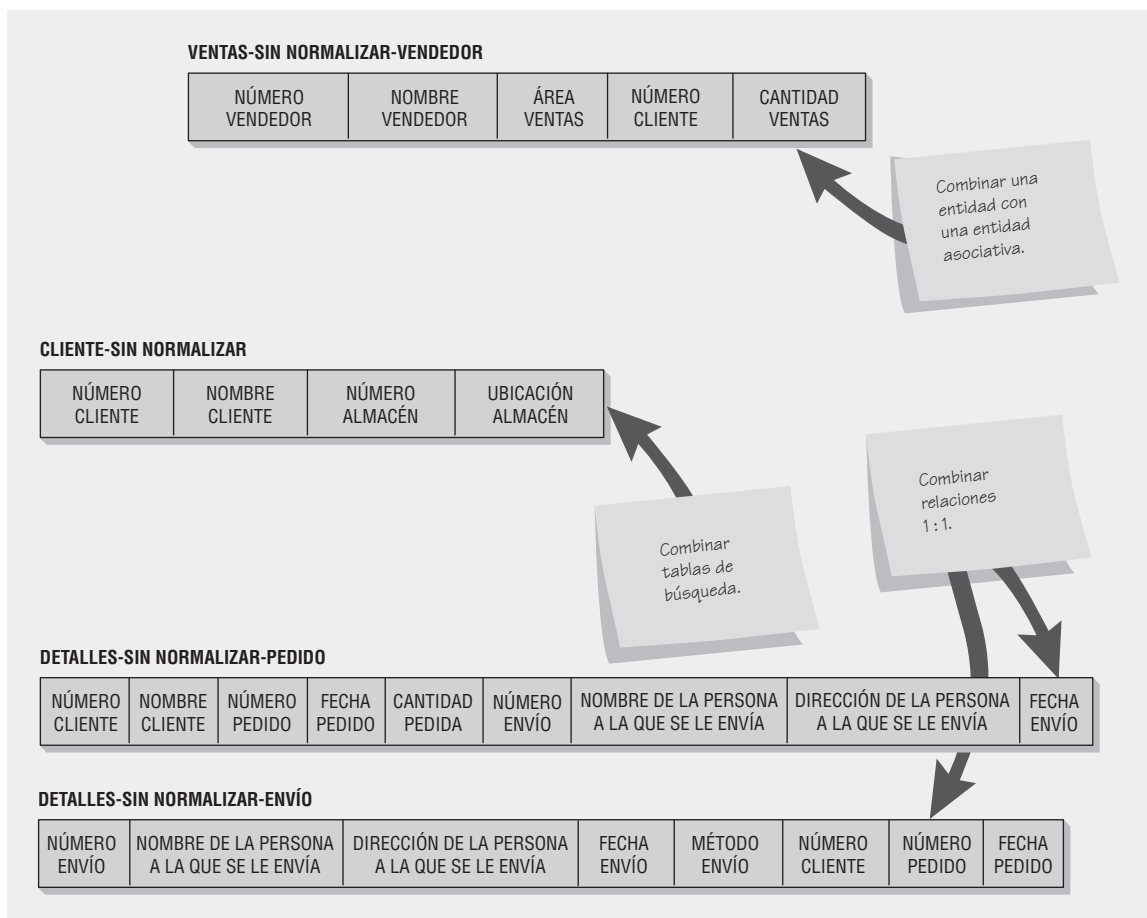
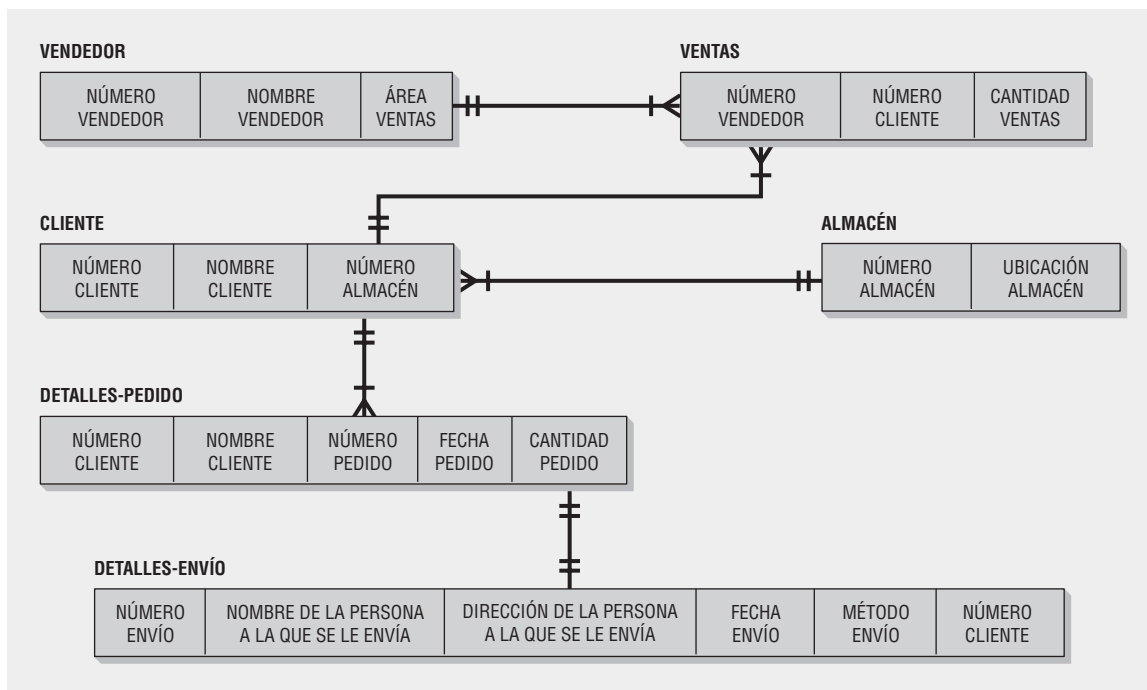
Una vez que los datos estén limpios, el analista debe decidir cómo resumirlos. Una vez resumidos, se pierden los detalles, de manera que un analista tiene que predecir el tipo de consultas que se podrían hacer.

Después, el analista necesita diseñar el almacén de datos con una organización lógica, y quizás incluso una agrupación física, de los datos por tema. Esto requiere mucho análisis y diseño. El analista necesita conocer bastante sobre el negocio.

Los típicos almacenes de datos tienden a medir desde 50 gigabytes hasta decenas de terabytes. Debido a que son grandes, también son caros. La mayoría de los almacenes de datos cuestan millones de dólares.

## PROCESAMIENTO ANALÍTICO EN LÍNEA

Introducido por primera vez en 1993 por E. F. Codd, el procesamiento analítico en línea (OLAP) se diseñó para responder a las consultas complejas de los tomadores de decisiones. Codd concluyó que un tomador de decisiones debía ver los datos de diferentes formas. Por lo tanto, la base de datos en sí tenía que ser multidimensional. Muchas personas describen a OLAP como el cubo de Rubik de datos. Puede ver los datos desde todos los diferentes lados y también los puede manipular torciéndolos o rotándolos para que tengan sentido.



**FIGURA 13.36**

Tres ejemplos de desnormalización para hacer más eficiente el acceso.

Este enfoque de OLAP validó el concepto de almacenes de datos. Ahora tenía sentido que los datos se organizaran de formas que permitieran consultas eficaces. Por supuesto, OLAP abarca el procesamiento de datos mediante la manipulación, resumen y cálculo, para que se abarque más de un almacén de datos.

## MINERÍA DE DATOS

La minería de datos puede identificar modelos que un humano no puede. Ya sea que el tomador de decisiones no pueda visualizar un patrón, o quizás no se le ocurre preguntar si ese patrón existe. Los algoritmos de minería de datos buscan patrones en almacenes de datos siguiendo ciertos algoritmos.

La minería de datos se conoce con otro nombre, descubrimiento de datos del conocimiento (KDD o *Knowledge Data Discovery*). Algunos piensan que KDD difiere de la minería de datos porque el propósito de KDD es ayudar a los tomadores de decisiones a encontrar los modelos en lugar de pasar el control a un algoritmo para que los encuentre. Las herramientas disponibles para ayudar en la toma de decisiones se denominan *siftware*; incluyen análisis estadístico, árboles de decisión, redes neurales, agentes inteligentes, lógica difusa y visualización de datos.

Los tipos de patrones que los tomadores de decisiones intentan identificar incluyen asociaciones, secuencias, agrupamientos y tendencias. Las asociaciones son modelos que ocurren al mismo tiempo. Por ejemplo, una persona que compra cereal normalmente compra leche para acompañar el cereal. Por otro lado, las sucesiones son modelos de acciones que tienen lugar durante un periodo. Por ejemplo, si una familia compra una casa este año, probablemente comprarán muebles (un refrigerador o lavadora y secadora) el próximo año. El agrupamiento es el modelo que se desarrolla entre un grupo de personas. Por ejemplo, clientes que tienen un código postal particular podrían tender a comprar un automóvil particular. Finalmente, las tendencias son modelos que se observan durante un periodo. Por ejemplo, los clientes podrían cambiar de comprar bienes genéricos a productos de marca.

En el capítulo 14 se trata con mayor detalle la minería de datos, junto con las consultas en general.

---

## PUBLICACIÓN DE BASES DE DATOS PARA WEB

Todas las bases de datos que se basan en la Web son para compartir información. Por ejemplo, puede establecer un sistema de administración de ventas y de inventario que permita a los empleados remotamente vista, edición y actualización de pedidos. Puede desarrollar las aplicaciones de administración de tarea para la Web que permita a los empleados asignar tareas o proyectos de búsqueda.

También es posible establecer una tienda en línea que permite a clientes navegar en una base de datos de productos y hacer compras. Incluso es posible personalizar aplicaciones Web que usen páginas dinámicas con el contenido personalizado basado en intereses que el cliente introduce.

El ejemplo presentado en la figura 13.37 ilustra cómo establecer un directorio de empleados en línea. Cualquier persona que visite el sitio Web puede buscar un empleado para localizar una extensión telefónica correcta. El ejemplo muestra una pantalla dividida con el código fuente en la parte superior y una representación visual de la página Web en la parte inferior de la pantalla. El ejemplo se generó usando Macromedia Dreamweaver UltraDev 4.

Macromedia Dreamweaver UltraDev es un paquete de software de avanzado para el desarrollo de sitios de comercio electrónico. UltraDev se enfoca en la publicación de bases de datos para Web y apoya el lenguaje de marcado extensible (XML).

UltraDev se orienta visualmente en lugar de orientarse hacia la edición de código fuente. Por lo tanto, el diseñador puede dibujar celdas de tabla directamente en una página, arrastrar celdas de otras ubicaciones o agruparlas para crear una tabla anidada. UltraDev

**FIGURA 13.37**

Pantalla de diseño de Macromedia Dreamweaver UltraDev 4 usada para publicar en Web una base de datos de directorio del empleados.



también permite al diseñador trabajar con datos vivos, páginas y lógica que se puede seguir usando mientras se trabaja con los datos del servidor. UltraDev permite a diseñadores crear rápidamente las aplicaciones Web manejadas con bases de datos para las múltiples plataformas de servidores.

La figura 13.38 muestra cómo establecer un formulario que permite a un usuario insertar un registro en una base de datos. Por supuesto, también necesita establecer las restricciones de seguridad para que los usuarios primero introduzcan una contraseña, y se apruebe, antes de que puedan modificar la base de datos. UltraDev se puede usar para crear aplicaciones de páginas activas de servidor (ASP), aplicaciones ColdFusion (CF) o aplicaciones de Java-Server Pages (JSP).

XML, un estándar no patentado, proporciona el mecanismo para tomar los datos básicos y traducirlos a un lenguaje universal que se puede leer por cualquiera con las herramientas de traducción adecuadas. Muchas compañías están usando XML para mejorar el intercambio de datos electrónicos y para mejorar aún más la experiencia de Web para clientes. XML hace posible que los datos se entreguen por Internet, compartan y presenten de la mejor forma posible.

Muchos paquetes de software están disponible para permitir la publicación de bases de datos en la Web mediante el uso de XML. Un ejemplo para las computadoras personales es FileMaker Pro, por FileMaker, Inc. es fácil integrar XML con FileMaker Pro mediante los formularios y hojas de estilo. FileMaker Pro recibe datos de una persona que usa un formulario HTML estándar y envía una respuesta en XML que incluye los datos y detalles acerca

## ALMACENAMIENTO DE MINERALES PARA LA SALUD, DATOS PARA LA MINERÍA

Uno de los empleados de Marathon Vitamin Shops, Esther See, se acerca al propietario, Bill Berry, para hacerle una observación. “Me he percatado de que nuestros clientes tienen diferentes hábitos. Algunos vienen de manera regular y otros son menos predecibles”, dice Esther. “Cuando veo a un cliente habitual, me enorgullezco de saber lo que el cliente comprará e incluso tal vez le recomiendo otras vitaminas que podrían gustarle. Creo que de esta manera genero más ventas. También, el cliente se siente mejor.”

Esther continúa: “Sin embargo, quisiera poder ayudar a algunos clientes que vienen con menos frecuencia”.

“Ésa es una actitud muy positiva, Esther, y también es de mucha ayuda para nuestra tienda”, responde Bill. “Sé que nos podemos beneficiar en otros sentidos si comprendemos los patrones de conducta de los clientes. Por ejemplo, podemos asegurarnos de contar con un artículo específico.”

Esther asiente y agrega: “No sólo me refiero al tipo de vitamina. Algunos clientes prefieren una marca en vez de otra. No sé si esto dependa de sus ingresos o del interés que tengan en determinadas actividades de esparcimiento. Por ejemplo, los deportes”.

“Entiendo, señorita See, ¿pero tiene alguna idea?”

“Sí, señor Berry”, responde ella. “Deberíamos organizar los datos que tenemos sobre nuestros clientes utilizando un almacén de datos. Podemos combinar nuestros datos con datos de otras fuentes. A continuación podemos buscar patrones en nuestros datos. Quizá podamos identificar patrones existentes y predecir nuevas tendencias.”

Piense de qué manera podría organizar un almacén de datos para Marathon Vitamin Shops. ¿Qué otras bases de datos le gustaría combinar en el almacén de datos? ¿Qué tipo de patrones debe buscar Bill Berry? Identifique estos patrones por tipo (asociaciones, secuencias, agrupamientos o tendencias).

de dónde se puede encontrar una hoja de estilo para los datos. Proporcionar una hoja de estilo diferente o un programa de JavaScript puede mostrar una vista diferente de los mismos datos a cada usuario. De esta forma, los usuarios que prefieren un estilo de presentación a otro reciben lo que realmente quieren.



**FIGURA 13.38**

Formulario INSERT RECORD (INTRODUCIR REGISTRO) de Macromedia Dreamweaver UltraDev 4 usado para permitir a usuarios modificar remotamente los registros de la base de datos.

## RESUMEN

Con frecuencia, la forma de almacenar datos es una decisión importante en el diseño de un sistema de información. Hay dos enfoques para almacenar datos. El primero es almacenarlos en archivos individuales, un archivo para cada aplicación. El segundo enfoque es desarrollar una base de datos que se pueda compartir por muchos usuarios para una variedad de aplicaciones conforme sea necesario. Ha habido mejoras impresionantes en el diseño de software de bases de datos para tomar ventaja de las posibilidades que ofrecen las interfaces gráficas.

El enfoque de archivo convencional a veces podría ser más eficaz, debido a que el archivo puede ser específico de una aplicación. Por otro lado, el enfoque de base de datos podría ser más apropiado porque los mismo datos necesitan ser introducidos, almacenados y actualizados una sola vez.

Una comprensión del almacenamiento de datos requiere entender tres dominios: realidad, datos y metadatos. Una entidad es cualquier objeto o evento para el cual estamos deseosos recopilar y almacenar datos. Los atributos son las características reales de estas entidades. Los datos pueden tener valores y se pueden organizar en los registros que se pueden acceder mediante una clave. Los metadatos describen los datos y pueden contener restricciones sobre el valor de los datos (como numérico).

Los ejemplos de archivos convencionales incluyen archivos maestros, archivos de tabla, archivos de transacción, archivos de trabajo y archivos de informe. Pueden tener una organización secuencial, listas enlazadas u organización de archivos *hash*. Las bases de datos se construyen típicamente con una estructura relacional. Sin embargo, los sistemas heredados pueden tener estructuras jerárquicas o de red.

La normalización es el proceso que toma vistas de usuario y las transforma en estructuras menos complejas llamadas relaciones normalizadas. Hay tres pasos en el proceso de normalización. Primero, se remueven todos los grupos repetitivos. Segundo, se eliminan todas las dependencias parciales. Finalmente, se remueven las dependencias transitivas. Después que se completen estos tres pasos, el resultado es la creación de varias relaciones en tercera forma normal (3NF).

El diagrama entidad-relación se podría usar para determinar las claves necesarias para un registro o una relación de base de datos. Los tres lineamientos a seguir al diseñar tablas maestras o relaciones de base de datos son que (1) cada entidad de datos separada debe crear una tabla maestra (no combine dos entidades distintas en una tabla); (2) un campo de datos específicos debe existir únicamente en una tabla maestra, y (3) cada tabla maestra o relación de base de datos debe tener programas para Crear, Leer, Actualizar y Eliminar.

El proceso de recuperación de datos podría incluir hasta ocho pasos: (1) se escogen una relación o relaciones y (2) se unen; (3) la proyección y (4) la selección se realizan en la relación para extraer las filas y columnas relevantes; (5) se podrían derivar nuevos atributos; (6) las filas se clasifican o indexan; (7) se calculan los totales y medidas de desempeño y, finalmente, (8) los resultados se presentan al usuario.

La desnormalización es un proceso que toma el modelo de datos lógico y lo transforma en un modelo físico que es eficaz para las tareas que son más necesarias. Los almacenes de datos difieren de las bases de datos tradicionales de muchas formas; una es que estas últimas almacenan datos desnormalizados, los cuales se organizan por temas. Los almacenes de datos permiten fácil acceso mediante software de minería de datos, denominado *siftware*, que busca los modelos e identifica las relaciones que los tomadores de decisiones humanos no imaginan.

El Lenguaje de Marcado Extensible (XML) es un lenguaje estándar no patentado que sirve como un mecanismo para tomar los datos básicos y traducirlos a un lenguaje universal que se puede leer por cualquiera con las herramientas de traducción adecuadas. Principalmente se usa para el intercambio de datos de negocio.





“La gente de Sistemas de Administración me ha dicho cosas muy buenas de su equipo. También se ha ganado algunos elogios de la gente de Capacitación, que difícilmente los hace. Ya sabe, es difícil complacer a Tom Ketcham en estos días. Incluso él les ve algunas posibilidades. Creo que usted nos puede hacer jalar parejo... a menos que cada quien jale por su lado otra vez. Sólo estoy bromeando. Le dije que pensara si somos una familia, un zoológico o una zona de guerra. Ahora es el momento de que empiece a diseñar sistemas adecuados para nosotros. Usted lleva suficiente tiempo aquí para haberse formado sus propias opiniones. Espero que sean favorables. Creo que nuestra famosa hospitalidad sureña debió influir en usted, ¿no es verdad? Estuve tan ocupado tratando de persuadirlo de que vale la pena el esfuerzo que haga con nosotros, que casi olvido decirle: Tom y Snowden están de acuerdo en considerar la posibilidad de adoptar algún tipo de base de datos. ¿Podría preparar algo durante las dos semanas siguientes? Tom se encuentra en una conferencia en Minneapolis, pero a su regreso usted debe tener algunas ideas sobre la base de datos para que Snowden y él las analicen. Concéntrese en ello”.

## PREGUNTAS DE HYPERCASE

1. Suponga que los miembros de su equipo han utilizado el Informe de Características de Clientes de la Unidad de Capacitación para diseñar una tabla de base de datos con el propósito de almacenar la información importante de este informe, con el resultado siguiente:  
Nombre de la tabla: TABLA DE CLIENTES

NOMBRE DE COLUMNA	DESCRIPCIÓN
ID CLIENTE (clave primaria)	Nemónico hecho por los usuarios, como HSPEST para Hospital Estatal
NOMBRE DEL CLIENTE	El nombre real y completo del cliente
DIRECCIÓN	La dirección del cliente
CONTACTO	El nombre de la persona a contactar
NÚMERO TELEFÓNICO	El número telefónico de la persona a contactar
CLASE	El tipo de institución (Hospital de Veteranos, clínica, otro)
TAMAÑO DEL PERSONAL	Tamaño del personal del cliente (número)
NIVEL DE CAPACITACIÓN	Nivel mínimo de experiencia del personal (como se define en la clase)
CANT-EQUIPO	El número de máquinas que tiene el cliente
TIPO-EQUIPO	El tipo de máquinas (por ejemplo, de rayos X, MRI, CAT)
MODELO-AÑO DEL EQUIPO	El modelo y año de cada máquina

2. Aplique normalización a la tabla que desarrolló su equipo para eliminar los grupos que se repitan. Muestre sus resultados.
3. Elimine las dependencias transitivas de su tabla y muestre la tabla de base de datos que obtuvo.

## PALABRAS Y FRASES CLAVE

administrador de base de datos  
almacén de datos  
almacenamiento de datos  
archivo convencional

archivo de informe  
archivo de tabla  
archivo de trabajo  
archivo de transacciones

archivo maestro  
 atributo  
 base de datos  
 caracteres especiales  
 CLAE - Crear, Leer, Actualizar  
 y Eliminar (CURD)  
 clave  
 clave concatenada  
 clave primaria  
 clave secundaria  
 datos  
 datos limpios  
 dependencias parciales  
 dependencias transitivas  
 desnormalización  
 diagrama de burbuja  
 diagrama de modelo de datos  
 diagrama entidad-relación (E-R)  
 entidad  
 estructura de datos de red  
 estructura de datos jerárquica  
 estructura de datos relacional  
 grupo repetitivo  
 integridad de dominio  
 integridad referencial  
 lenguaje de marcado extensible  
 (XML)  
 lista enlazada  
 minería de datos  
 normalización  
 organización de un archivo hash

pasos en la recuperación de información  
 calcular  
 derivar  
 escoger  
 indexar o clasificar  
 presentar  
 proyectar  
 seleccionar  
 unir  
 patrones  
 agrupamiento  
 asociaciones  
 secuencias  
 tendencias  
 primera forma normal (1NF)  
 procesamiento analítico en línea (OLAP)  
 realidad, datos y metadatos  
 recuperación  
 registro  
 relación  
 relación sin normalizar  
 restricción de integridad de entidad  
 segunda forma normal (2NF)  
 software  
 sistema de administración de base  
 de datos (DBMS)  
 subtipo de entidad  
 tercera forma normal (3NF)  
 vista física  
 vista lógica

## PREGUNTAS DE REPASO

1. ¿Cuáles son las ventajas de organizar el almacenamiento de datos como archivos separados?
2. ¿Cuáles son las ventajas de organizar el almacenamiento de datos usando un enfoque de base de datos?
3. ¿Cuáles son las medidas de efectividad del diseño de la base de datos?
4. Mencione algunos ejemplos de entidades y sus atributos.
5. Defina el término *metadatos*. ¿Cuál es el propósito de los metadatos?
6. Mencione los tipos de archivos convencionales comúnmente usados. ¿De éstos cuáles son archivos temporales?
7. ¿Qué es una lista enlazada?
8. ¿Qué sucede con frecuencia cuando se usa la organización de un archivo *hash*?
9. Mencione los tres tipos principales de organización de base de datos.
10. Defina el término *normalización*.
11. ¿Qué se remueve cuando una relación se convierte a la primera forma normal?
12. ¿Qué se remueve cuando una relación se convierte de 1NF a 2NF?
13. ¿Qué se remueve cuando una relación se convierte de 2NF a 3NF?
14. Mencione las tres restricciones de una entidad. En una frase, describa el significado de cada restricción.
15. Mencione los ocho pasos para recuperar, preclasificar y presentar los datos.
16. ¿Qué hace una función de unión? ¿Qué es la proyección? ¿Qué es la selección?

17. Establezca las diferencias entre clasificar e indexar.
18. Mencione dos formas de almacenar una relación muchos a muchos y las diferencias entre los dos métodos.
19. Defina la desnormalización.
20. Explique las diferencias entre las bases de datos tradicionales y los almacenes de datos.
21. Defina lo que hace el software cuando se usa en la minería de datos.
22. Explique cómo funciona XML para facilitar el intercambio de datos de negocio.

## PROBLEMAS

1. Dado el siguiente archivo de arrendatarios:

Número del registro	Apellido	Número de departamento	Renta	Caducidad del arrendamiento
41	Warkentin	102	550	4/30
42	Buffington	204	600	4/30
43	Schuldt	103	550	4/30
44	Tang	209	600	5/31
45	Cho	203	550	5/31
46	Yoo	203	550	6/30
47	Pyle	101	500	6/30

- a. Desarrolle una lista enlazada por el número de departamento en orden ascendente.
  - b. Desarrolle una lista enlazada según el apellido en orden ascendente.
2. El siguiente ejemplo muestra un reporte de calificación para dos estudiantes en la Universidad del sur de Nueva Jersey:

<p style="text-align: center;"><i>Informe de calificación USNJ</i> <i>Semestre primavera 2000</i></p> <p>Nombre: I. M. Smarte <span style="float: right;">Especialidad: MIS</span>  Estudiante: 053-6929-24 <span style="float: right;">Estado: Estud. de últ. grado</span></p>				
Número del curso	Título del curso	Profesor	Departamento del profesor	Calificación
MIS 403	Análisis de sistemas	Diggs, T.	MIS	A
MIS 411	Bases conceptuales	Barre, G.	MIS	A
MIS 420	Factores humanos en IS	Barre, G.	MIS	B
CIS 412	Diseño de bases de datos	Menzel, I.	CIS	A
DESC 353	Modelos de administración	Murney, J.	MIS	A

<p style="text-align: center;"><i>Informe de calificación USNJ</i> <i>Semestre primavera 2000</i></p> <p>Nombre: E.Z. Grayed <span style="float: right;">Especialidad: MIS</span>  Estudiante: 472-6124-59 <span style="float: right;">Estado: Estud. de últ. grado</span></p>				
Número del curso	Título del curso	Profesor	Departamento del profesor	Calificación
MIS 403	Análisis de sistemas	Diggs, T.	MIS	B
MIS 411	Bases conceptuales	Barre, G.	MIS	A

3. Dibuje un diagrama de modelo de datos con asociaciones para la vista de usuario en el problema 2.
4. Convierta la vista de usuario del problema 3 a una relación 3NF. Muestre cada paso en el proceso.

5. Dibuje un diagrama entidad-relación para la siguiente situación: muchos estudiantes practican muchos deportes diferentes. Una persona, llamada entrenador principal, asume el papel de entrenamiento en todos estos deportes. Cada una de las entidades tiene un número y un nombre. (Haga cualesquier suposiciones necesarias para completar un diagrama razonable. Mencione sus suposiciones.)
6. El diagrama entidad-relación que dibujó en el problema 5 representa las entidades de datos que se necesitan implementar en un sistema para llevar un registro de los estudiantes y los equipos deportivos que practican. Mencione las tablas que el sistema necesita implementar, junto con las claves principal, secundaria y externa que se requieren para vincular las tablas.
7. Dibuje un diagrama entidad-relación para la siguiente situación: una panadería comercial hace muchos productos diferentes. Estos productos incluyen panes, postres, pasteles especiales y muchos otros tipos de repostería. Los ingredientes tales como la harina, especias y leche se compran a vendedores. A veces un ingrediente se compra de un solo vendedor y otras veces un ingrediente se compra de muchos vendedores. La panadería tiene clientes comerciales, tales como escuelas y restaurantes que regularmente hacen pedidos de pasteles. Cada pastel tiene un especialista que supervisa su proceso de producción e inspecciona el producto terminado.
8. Mencione las tablas y claves que se necesitan para implementar el sistema de la panadería comercial.
9. Dibuje un diagrama E-R para el sistema de pedidos de la figura 13.26.
10. Dibuje un diagrama de flujo de datos para hacer un pedido. Base su diagrama de flujo de datos en el modelo E-R.

## PROYECTOS DE GRUPO

1. Gregg Baker pide por Web boletos para dos conciertos. Sus pedidos se procesan, se les asigna un asiento único en el auditorio y los boletos se envían por correo por separado. Uno de los juegos de boletos se pierde en el correo. Cuando él llama al número de servicio, no recuerda la fecha o los números de asiento, pero la agencia del boleto pudo localizar sus boletos rápidamente porque la agencia desnormalizó la relación. Describa el sistema de pedido de boleto mencionando los elementos de datos que se llevan en el formulario de pedido y el formulario de envío. ¿Qué información tuvo que dar Gregg a la agencia del boleto para recuperar la información?

## BIBLIOGRAFÍA SELECCIONADA

- Avison, D. E., *Information Systems Development: A Database Approach*, 2a. ed., Londres: Blackwell Scientific, 1992.
- Avison, D. E. y G. Fitzgerald, *Information Systems Development: Methodologies, Techniques, and Tools*, 2a ed., Nueva York: McGraw-Hill, 1995.
- Codd, E. F., "Twelve Rules for On-Line Analytic Processing", *Computerworld*, 13 de abril de 1995.
- Dietel, H. M., P. J. Dietel y T. R. Nieto, *E-Business and e-Commerce: How to Program*, Upper Saddle River, NJ: Prentice Hall, 2001.
- Everest, G. C., *Database Management: Objectives, System Functions, and Administration*, Nueva York: McGraw-Hill, 1985.
- Gane, C. y T. Sarson, *Structured Systems Analysis: Tools and Techniques*, Englewood Cliffs, NJ: Prentice Hall, 1979.
- Gray, P., "Data Warehousing: Three Major Applications and Their Significance", en K. E. Kendall, ed., *Emerging Information Technologies, Improving Decision, Cooperation, and Infrastructure*, Thousand Oaks, CA: Sage Publications, 1999.
- McFadden, F. y J. A. Hoffer, *Modern Database Management*, 4a. ed., Redwood City, CA: Benjamin-Cummings, 1994.
- Sanders, G. L. *Data Modeling*, Nueva York: Boyd and Fraser, 1995.
- "XML", disponible en: <www.xml.com>. Último acceso, 9 de febrero de 2001.



ALLEN SCHMIDT, JULIE E. KENDALL Y KENNETH E. KENDALL

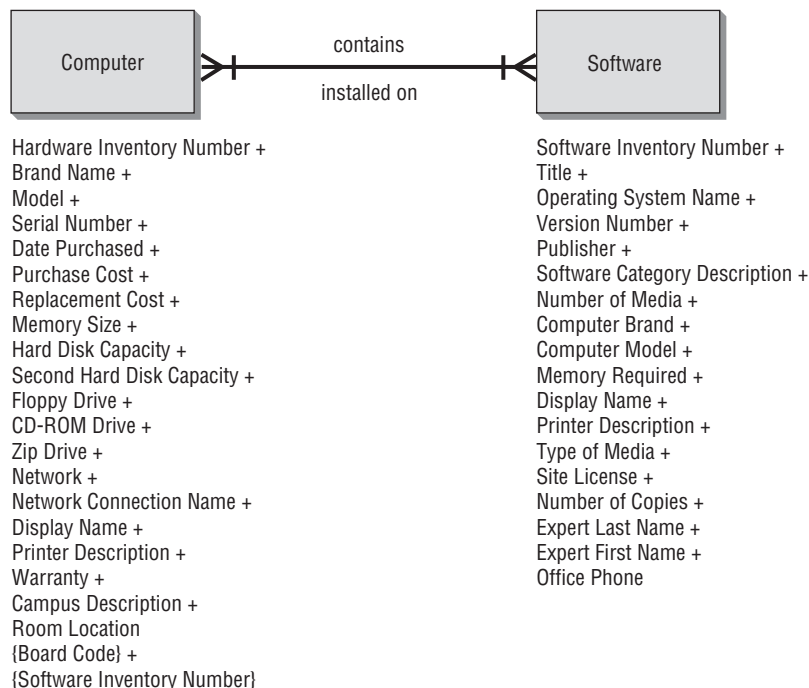
## FUNDAMENTOS DE DATOS

# 13

Después de haber terminado las numerosas entrevistas, prototipos, diagramas de flujo de datos y entradas del diccionario de datos, Anna y Chip comienzan a trabajar en el modelo entidad-relación. “Yo me encargaré de crear las relaciones de las tablas de Microsoft Access”, promete Anna. Chip se ofrece a elaborar un diagrama de entidad-relación. “Comparemos la precisión y consistencia de los dos diagramas cuando terminemos”, sugiere Anna, y así lo hacen.

La figura E13.1 muestra el diagrama de entidad-relación para el sistema de inventario de computadoras. Visible Analyst denomina entidad a cada uno de los rectángulos. Cada entidad representa un archivo de información almacenado en el sistema, y corresponde a un almacén de datos en el diagrama de flujo de datos. Cada uno de los rectángulos con un diamante representa una relación entre las entidades de datos. Un rectángulo con un óvalo representa una entidad que no puede existir sin la entidad a la que se conecta, como una tabla de códigos.

“Ya elaboré el diagrama de entidad-relación, empezando por las partes más simples del sistema”, Chip le dice a Anna. “Las primeras entidades de datos creadas son SOFTWARE y COMPUTER. La relación consiste en que el software está instalado en la computadora. Luego determiné la cardinalidad de la relación. Puesto que un paquete de software podría instalarse en muchas computadoras, esta relación es uno-a-muchos. Cada computadora también



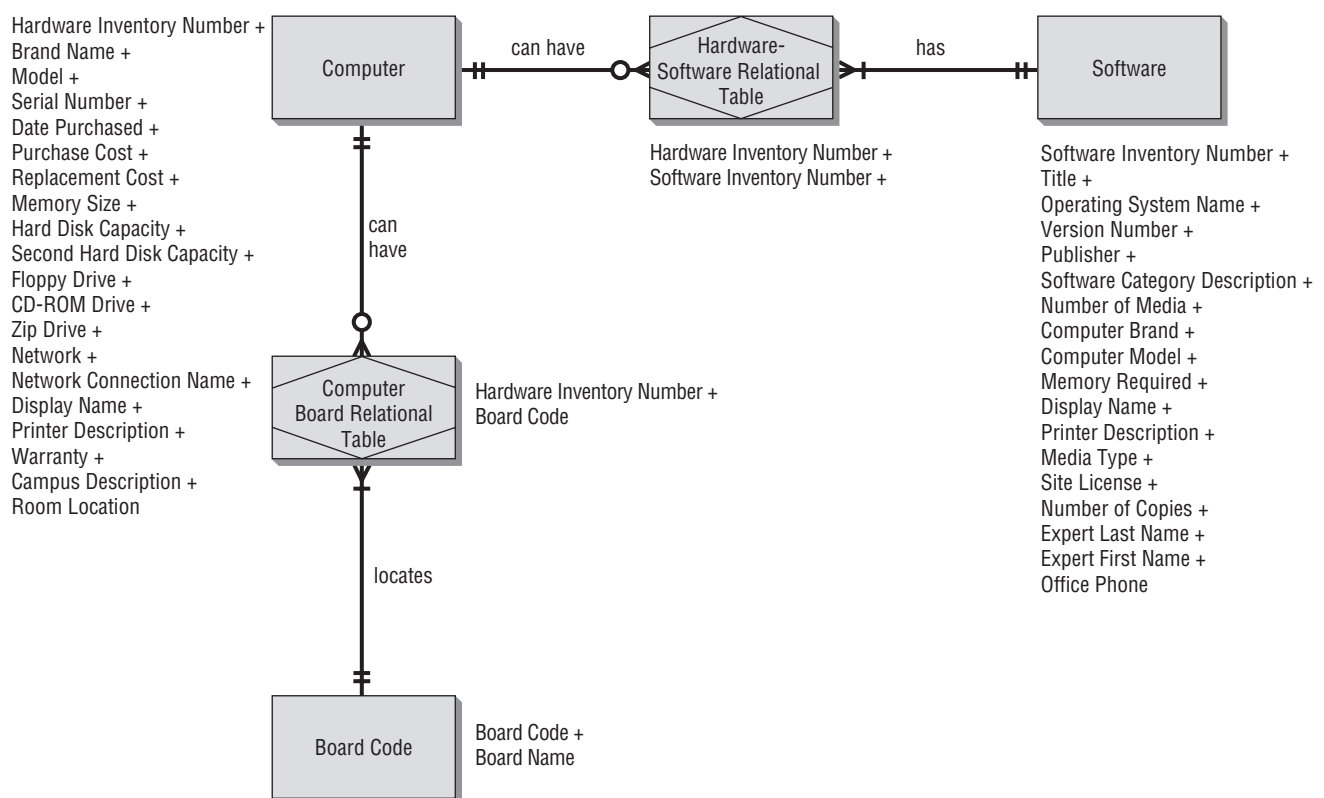
**FIGURA E13.1**

Diagrama de entidad-relación sin normalizar para el sistema de cómputo.

## 13

puede tener muchos paquetes de software diferentes instalados, así que también presenta una relación uno-a-muchos. Dado que hay una relación uno-a-muchos para cada una de las entidades de datos, la relación entre ellos es muchos-a-muchos.” El diagrama de entidad-relación sin normalizar se muestra en la figura E13.1, junto con las entradas del diccionario de datos para las tablas.

Chip continúa: “Esta primera vista aún no está normalizada. Observa que BOARD CODE y SOFTWARE INVENTORY NUMBER son elementos que se repiten en la entidad HARDWARE. Tendré que crear varias entidades para cada uno de ellos”. Un poco más tarde Chip revisa su trabajo con Anna. BOARD CODE se ha vuelto una entidad separada, enlazada por una entidad relacional, y SOFTWARE INVENTORY NUMBER se ha eliminado y colocado en una entidad relacional. Consulte el diagrama de entidad-relación ilustrado en la figura E13.2. “Esto pone los datos en la primera forma normal”, afirma Chip. “Asimismo, tampoco hay elementos que dependan sólo de una parte de la clave, así que los datos también están en la segunda forma normal. Sin embargo, hay elementos que no son parte de la entidad que se representa en el diagrama, y tendrán que ser eliminados. Por ejemplo, observa DISPLAY NAME y PRINTER DESCRIPTION. Estos elementos no son parte de la computadora pero se conectan a ella. Deben tener su propia entidad. Esto hace más fácil el cambio de la escritura de, digamos, una impresora. En lugar de tener que cambiar la escritura de la impresora en muchos de los registros de COMPUTER, sólo tendría que cambiarse una vez.”



**FIGURA E13.2**

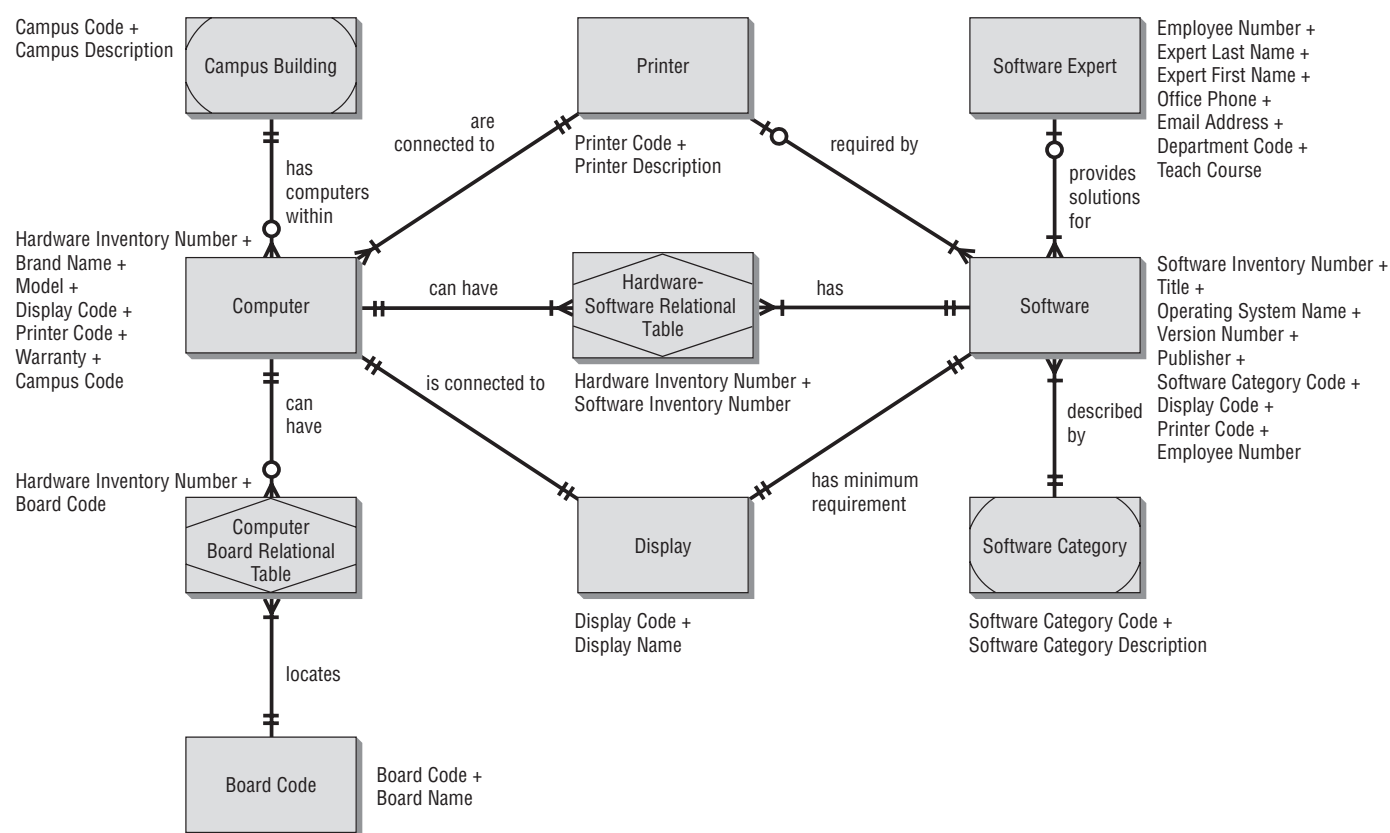
El diagrama de entidad-relación del sistema de cómputo en la primera forma normal.

## 13

Anna está de acuerdo, y comenta: “En realidad, ésta es una buena evaluación de la situación. Facilitará mucho la implementación de las tablas de Microsoft Access”.

Chip continúa trabajando en el diagrama de entidad-relación. Unas horas más tarde exclama: “Creo que ya está. ¿Quieres darle un vistazo a la versión final?” Esta versión se muestra en la figura E13.3. Todas las entidades y relaciones se han descrito en el depósito. La figura E13.4 muestra la primera pantalla del depósito para la entidad COMPUTER. Observe que el área **Composition** contiene los elementos finales para dar seguimiento a las computadoras y que las claves se indican mediante la notación [Pk] para la clave principal y [Akn] para una clave alterna, donde n representa cualquier número único. Las claves externas se encuentran en la parte más baja de la lista y se indican con la notación [Fk]. La segunda pantalla contiene información sobre las relaciones que la entidad COMPUTER tiene con otras entidades, incluyendo la cardinal. Esta pantalla también contiene las ubicaciones de la entidad.

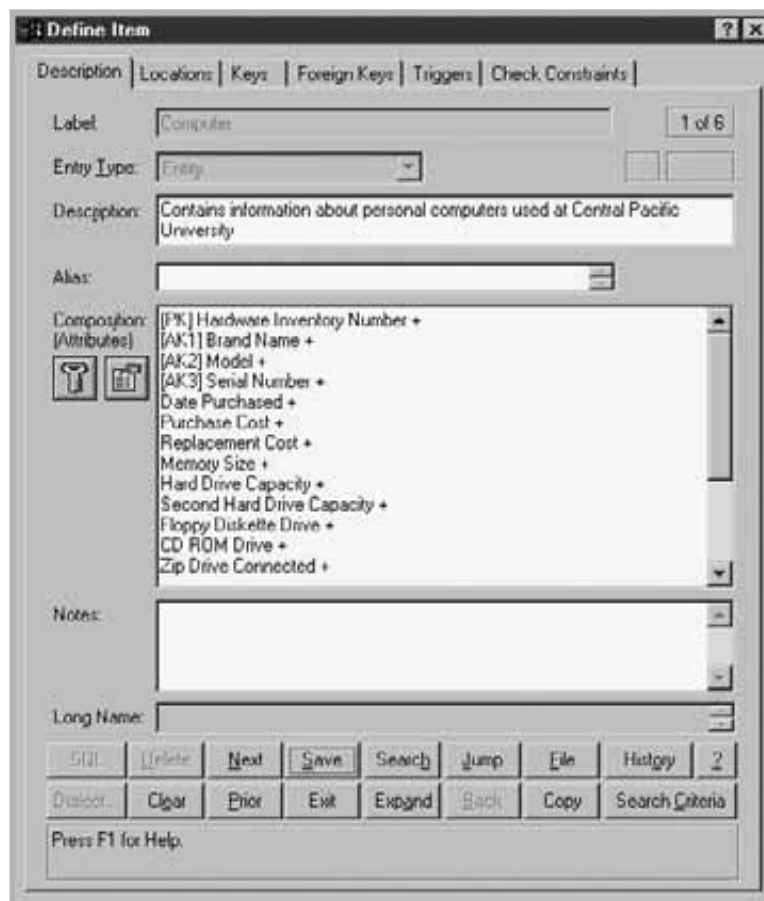
La relación PROVIDES SOLUTIONS FOR, que enlaza a SOFTWARE EXPERT y SOFTWARE, tiene una definición así como las entidades que están enlazadas en ambos extremos de la relación. Se incluye la cardinal para cada extremo de la relación.



**FIGURA E13.3**

Diagrama de relación-entidad final para el sistema de cómputo.

## 13

**FIGURA E13.4**

Pantalla de descripción de entidades de COMPUTERS, la primera pantalla del depósito.

Anna repasa la última versión y exclama: “¡Se ve muy bien! Fue buena decisión haber movido PRINTER y DISPLAY a sus propias entidades. Veo que CAMPUS BUILDING se ha movido a su propia entidad. Buena idea, puesto que el edificio no es parte de la computadora. Asimismo, SOFTWARE EXPERT definitivamente no es parte de la entidad SOFTWARE. ¿Qué hay de SOFTWARE CATEGORY?”

“Pasé SOFTWARE CATEGORY a su propia entidad para ahorrar espacio en los archivos maestros cuando se construyan”, responde Chip. “En realidad es una tabla de códigos, y podemos almacenar un código pequeño en lugar de una descripción larga. ¿Por qué no revisaste dos veces las diversas claves del diagrama? Cada entidad relacionada, en el extremo muchos, debe tener una clave externa que coincida con la clave principal de la entidad en el extremo uno.”

Anna examina el diagrama durante unos momentos y sugiere: “A mí me parece bien, ¿pero por qué no ejecutas algunos de los informes de Visible Analyst para el diagrama de entidad-relación?”

“¡Esa es una excelente idea!”, exclama Chip. El primer paso es ejecutar el verificador de sintaxis (Syntax Check) de Visible Analyst para el diagrama de entidad-relación final. Esta opción busca entidades y relaciones que no se hayan nombrado, y no produce ningún error cuando Chip la ejecuta. A continuación, Chip ejecuta la opción de análisis de normaliza-



## 13

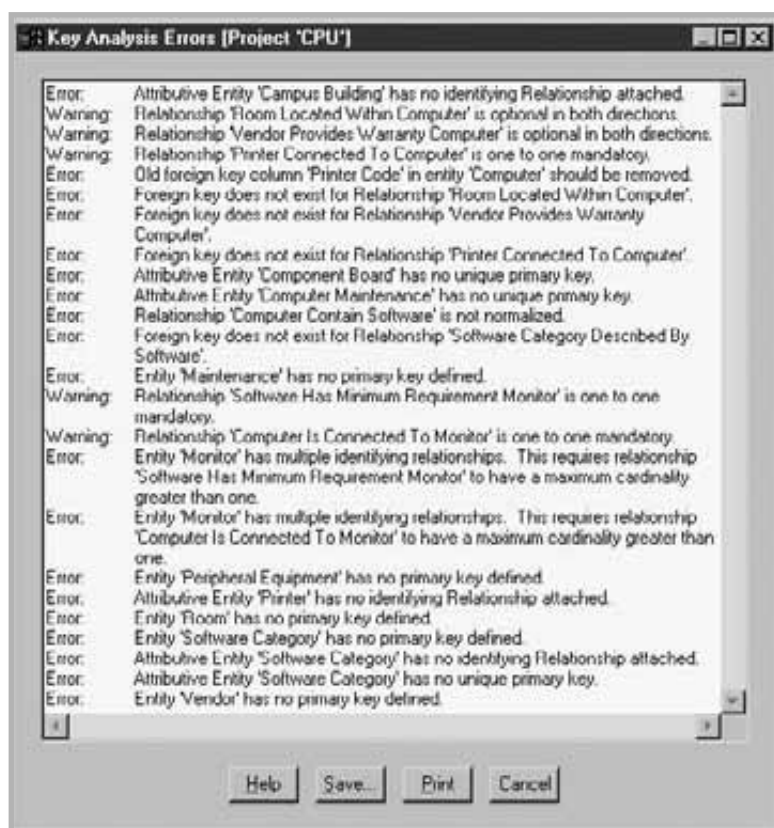
ción (Normalization Analysis) que despliega una serie de mensajes de error. Estos mensajes informan a Chip que algunas de las relaciones tienen una cardinalidad uno-a-uno, que es aceptable. Los otros mensajes de advierten a Chip que SOFTWARE CATEGORY y CAMPUS BUILDING no tienen relaciones que los identifiquen. En consecuencia, las claves del registro no se han definido en el depósito.

El siguiente análisis que Chip lleva a cabo es el informe de **errores de análisis de claves (Key Analysis Errors)**. Esta opción realiza un análisis de sintaxis y normalización, y reporta los problemas con las claves principales y las externas. En la figura E13.5 se ilustra el informe del análisis, que muestra que SOFTWARE CATEGORY no tiene ninguna clave principal y que PRINTER no tiene una relación que la identifique. El último informe de análisis que ejecuta Chip es el informe **Model Balancing**, el cual muestra algunos de los errores descubiertos por los informes anteriores, pero, además, muestra todos los elementos que se encuentran en el área **Composition** de las entidades y que no son utilizados por algún proceso del diagrama de flujo de datos.

“Ahora que ya realizamos el análisis, observa cómo funciona esta característica”, comenta Chip. “Voy a ejecutar la **sincronización de claves (Key Synchronization)**.”

“Pensé que ya habíamos acabado los informes”, contesta Anna.

“La **sincronización de claves** es más que un informe”, responde Chip. “Definirá todas las claves externas del proyecto.”



**FIGURA E13.5**

Pantalla de salida de KEY ANALYSIS ERRORS.

## 13

“¡Esto tengo que verlo!”, exclama Anna, al tiempo que coloca una silla junto a Chip.

“Primero tenemos que dibujar el diagrama de entidad-relación y definir las claves principales, lo cual ya hicimos”, explica Chip. “Échale un vistazo a esta entrada del depósito.” Chip abre el diagrama de entidad-relación y hace doble clic en la entidad COMPUTERS para abrir su entrada en el depósito. Se han definido la clave principal (la notación [pk] antes del elemento HARDWARE INVENTORY NUMBER en el área **Composition**) y varias claves alternas ([Ak1], [Ak2] y [Ak3]). Chip hace clic en **Repository** y en **Key Synchronization**. Esto produce un informe para los analistas donde se indican los cambios que se realizaron a las tablas así como los errores que aún persisten.

“Veamos nuevamente esta entrada del depósito”, murmura Chip. “Mira, **Key Synchronization** ha agregado las claves externas necesarias.” Estas claves se indican con la notación [FK] delante de las nuevas clave. La entidad COMPUTERS modificada se muestra en la figura E13.6.

Después de examinar el diagrama, así como las entradas del depósito y los informes de análisis, Anna y Chip están satisfechos porque las relaciones entre los datos se realizaron



**FIGURA E13.6**

Una entrada del depósito, COMPUTERS.

## 13

con precisión. En seguida, deciden cómo diseñar los archivos o la base de datos a partir de los diagramas.

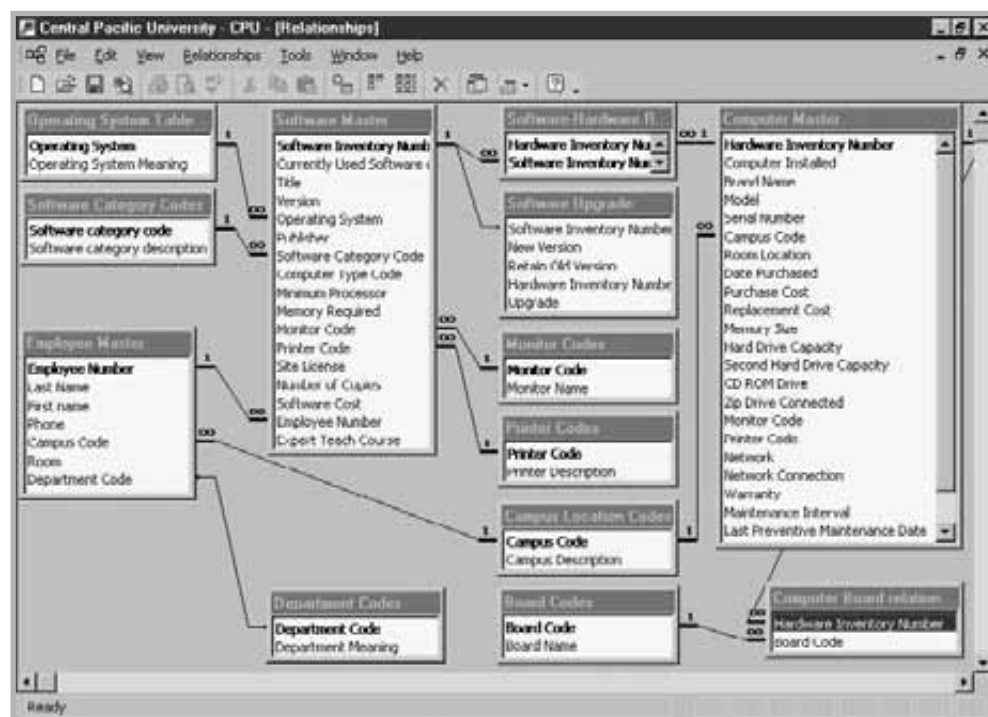
Anna dice: “El sistema debe implementarse con Microsoft Access, porque una estructura de base de datos dará cabida fácilmente a las diversas relaciones”.

Primero se analiza la relación HARDWARE/SOFTWARE. Como hay una relación muchos-a-muchos entre estas dos entidades de datos, puede implementarse con tres tablas de base de datos:

1. Una tabla HARDWARE MASTER.
2. Una tabla SOFTWARE MASTER.
3. Una tabla HARDWARE/SOFTWARE RELATIONSHIP que contendría los campos de clave para las tablas maestras HARDWARE y SOFTWARE para todo el software instalado en todas las máquinas.

“Creo que ahora me toca trabajar en las relaciones”, dice Anna al tiempo que toma una copia del diagrama de entidad-relación. “Modificaré las tablas de Microsoft Access a partir de las sesiones de creación de prototipos.”

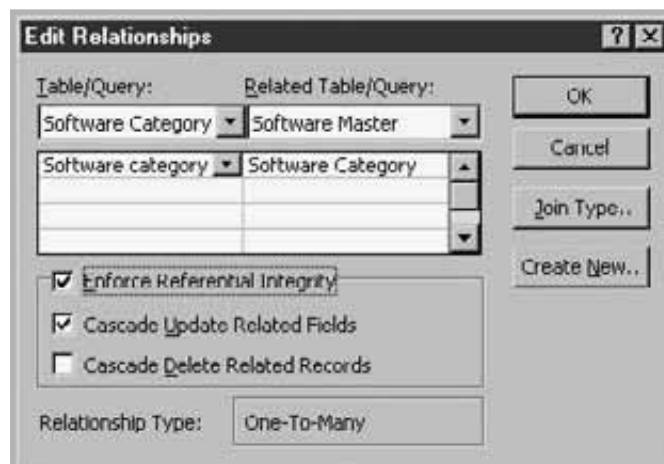
Anna empieza por establecer las claves principales para cada una de las tablas. Cuando termina las tablas, procede a crear las relaciones entre ellas. En la figura E13.7 se muestra el diagrama de relaciones de Microsoft Access. Los rectángulos del diagrama representan las tablas de la base de datos y corresponden a los diversos tipos de entidades encontrados en el diagrama de entidad-relación. Observe que la cardinalidad se representa mediante “1” y el símbolo de infinito. Los campos de las claves principales ocupan el primer campo de cada



**FIGURA E13.7**

Diagrama RELATIONSHIPS de Microsoft Access.

## 13

**FIGURA E13.8**

Ejemplo que muestra el establecimiento de la integridad referencial para la relación entre SOFTWARE CATEGORY CODES y SOFTWARE MASTER.

rectángulo; también se muestran en negritas. Si las claves externas son visibles en el rectángulo de la tabla, se muestran conectadas al otro extremo de la línea de la relación. Las claves se arrastran de una tabla a otra para establecer una relación, y aparece un cuadro de diálogo para determinar las propiedades de la relación. Por ejemplo, la propiedad **Enforce Referential Integrity** significa que usted no puede crear un registro en la tabla del extremo muchos sin crearlo primero en el extremo uno (que contiene la clave principal). La figura E13.8 ilustra el establecimiento de la integridad referencial para la relación entre SOFTWARE CATEGORY CODE y SOFTWARE MASTER. Observe que la opción **Cascade Update Related Fields** aparece seleccionada. Si usted cambia un valor de SOFTWARE CATEGORY CODE, el mismo código se actualizará en SOFTWARE MASTER. No obstante, **Cascade Delete Related Records** no está seleccionada. Usted no quiere que el sistema elimine un SOFTWARE CATEGORY CODE y que al mismo tiempo elimine todos los registros de SOFTWARE MASTER relacionados.

“Me gustaría producir algo de documentación para el sistema”, comenta Anna. “Sería útil cuando necesitaríamos modificar tanto el diseño como los objetos y el código de Microsoft Access.” En la característica **Repository Reports** hay diversas matrices que sería conveniente producir. La primera es **Entities versus Data Stores Matrix**. Esta matriz muestra las entidades que se encuentran en todos los diagramas de entidad-relación y los almacenes de datos que contienen elementos similares. También es útil para transformar el diagrama de entidad-relación en diagrama de flujo de datos.

La siguiente matriz que se produce es **Composition Matrix**. Ésta proporciona una cuadrícula de referencias cruzadas de elementos y las entidades que los contienen. Esta matriz es útil para determinar qué entidades necesitan modificarse (o qué tablas de Microsoft Access) si cambia el tamaño del elemento o cualquier otra de sus características. Una última matriz que es útil para evaluar cambios a todo el sistema es la **Diagram Location Matrix**. En la figura E13.9 se encuentra un ejemplo de esta matriz, que muestra las entidades y los diagramas en los que se localizan.

## 13




Date: 1/19/2003  
Time: 11:58 AM

Computer System—Unnormalized			
Computer System—First Normal Form			
Computer System			
Board Codes	X	X	
Campus Building	X		
Component Board			
Computer Board Relational Table	X	X	
Computer Maintenance			
Computers	X	X	X
Display	X		
Hardware-Software Relational Table	X	X	
Peripheral Equipment			
Printer	X		
Rooms			
Software	X	X	X
Software Category	X		
Software Expert	X		
Software-Hardware Relation			
Vendor			

**FIGURA E13.9**

Matriz de localización de diagramas.






**EJERCICIOS**

-  E-1. Use Visible Analyst para ver los diagramas de entidad relación del sistema de cómputo sin normalizar y en su primera forma normal.
-  E-2. Use Visible Analyst para ver el diagrama de entidad-relación para el sistema de cómputo.
-  E-3. Agregue la entidad VENDOR al diagrama. El vendedor garantiza las computadoras, y la relación entre VENDOR y COMPUTER es que un VENDOR puede garantizar muchas COMPUTER(s).



Los ejercicios precedidos por un icono Web indican que en el sitio Web del libro hay material de valor agregado. Los estudiantes pueden descargar una base de datos de Microsoft Access que pueden utilizar para completar los ejercicios.

## 13

-  E-4. Agregue la entidad MAINTENANCE al diagrama. A las computadoras se les hacen reparaciones de mantenimiento, y la relación entre el MAINTENANCE y COMPUTER(s) es que una COMPUTER puede tener muchos registros de MAINTENANCE.
-  E-5. Describa la entidad SOFTWARE CATEGORY en el depósito. Incluya los elementos que se encuentran en el diagrama de entidad-relación bajo SOFTWARE CATEGORY en el área **Composition**.
-  E-6. Describa la entidad MAINTENANCE en el depósito. Los elementos son los siguientes:
  - a. MAINTENANCE ORDER NUMBER.
  - b. HARDWARE INVENTORY NUMBER.
  - c. MAINTENANCE DATE.
  - d. TYPE OF MAINTENANCE.
  - e. COST OF MAINTENANCE.
  - f. MAINTENANCE COVERED BY WARRANTY.
-  E-7. Describa la entidad del VENDOR. Los elementos son los siguientes:
  - a. VENDOR NUMBER.
  - b. VENDOR NAME.
  - c. STREET.
  - d. CITY.
  - e. STATE.
  - f. ZIP CODE.
  - g. TELEPHONE NUMBER.
  - h. DATE LAST ORDER SENT.
  - i. TOTAL AMOUNT PURCHASED FROM VENDOR.
  - j. TOTAL NUMBER OF ORDERS SENT TO VENDOR.
-  E-8. Produzca los siguientes informes usando Visible Analyst:
  - a. Abra el diagrama de entidad-relación para el sistema de cómputo; a continuación aplique el verificador de sintaxis (**Syntax Check**) al diagrama (**Diagram/Analyze/Syntax Check**).
  - b. Ejecute el informe de análisis de normalización (**Normalization Analysis**) para el diagrama de entidad-relación del sistema de cómputo (**Diagram/Analyze/Normalization**).
  - c. El informe de análisis de claves (**Key Analysis Report**).
  - d. El informe de sincronización de claves (**Key Synchronization Report**). ¿Qué ha cambiado en el área **Composition** para cada una de las entidades?
  - e. El informe de equilibrio del modelo (**Model Balancing Report**).
  - f. La matriz **Entities versus Data Stores Matrix**.
  - g. La matriz **Composition Matrix**.
  - h. La matriz **Diagram Location Matrix**.
- E-9. Explique en un párrafo la relación entre una clave externa y una clave principal, y por qué es necesario tenerlas en entidades separadas cuando hay una relación entre las entidades.