

## Capítulo 8

# Bases de datos

## INTRODUCCIÓN

La información que se almacena en cualquier organización constituye una base de datos que se considera de gran valor. Los individuos, las escuelas, los negocios y demás usuarios nos apoyamos cotidianamente en la información para llevar a cabo nuestras actividades. Cuando retiramos dinero de un cajero automático, cuando compramos alguna mercancía en un sitio *Web* o cuando nos inscribimos en un curso en línea, interactuamos directamente con una base de datos.

## ¿QUÉ ES UNA BASE DE DATOS?

Una base de datos es un conjunto de información de cualquier tipo: un directorio telefónico, un catálogo de fichas bibliográficas, un archivo de inventario, los registros de calificaciones escolares de un estudiante, etc. Cualquier conjunto de información puede convertirse en una base de datos computarizada.

Una **base de datos computarizada** es un conjunto de información almacenada en forma organizada en una computadora, y un **sistema de administración de base de datos** (DBMS, siglas en inglés de *database management system*) es una herramienta de *software* para organizar el almacenamiento y la recuperación de esa información.

Las bases de datos computarizadas ofrecen diferentes ventajas en relación con las bases de datos que se registran en papel y con lápiz:

- Facilitan el almacenamiento de grandes cantidades de información.
- Facilitan la recuperación rápida y flexible de información.
- Facilitan la organización y reorganización de la información.
- Facilitan la impresión y distribución de información por medios diversos.
- Evitan la redundancia de datos.

En la sección siguiente explicaremos cada una de las ventajas anteriores que justifican el uso de bases de datos.

## CONVENIENCIA DE UTILIZAR UNA DBMS

Algunas de las razones que justifican la utilización de bases de datos son las siguientes:

- Las bases de datos facilitan el almacenamiento de grandes volúmenes de información; por ejemplo, si tienes una colección de 20 o 30 CD de música, podrías escribir la información sobre ellos en una libreta. Pero si tienes 500 o 1 000 CD, te sería tan difícil manejar la libreta como la colección de discos compactos. Con una base de datos computarizada puedes almacenar todo tu catálogo de música en un solo disco. Conforme aumenta el volumen de información, mayor es el beneficio de utilizar una DBMS.
- Las bases de datos facilitan la recuperación rápida y flexible de información. Si eres estudiante en alguna universidad, cada vez que efectúes el pago de la colegiatura hay que buscar tu expediente para actualizar tu saldo. Si la universidad tiene registrados a más de 1 000 alumnos, es obvia la ventaja de la base de datos computarizada. Esta ventaja es mayor aún si la búsqueda no concuerda con la organización del archivo. Por ejemplo, supongamos que tienes un número de teléfono en una hoja de papel y deseas buscar en el directorio telefónico el nombre y la dirección de la persona a quien corresponde ese número. Si la organización del directorio fuera por números de menor a mayor, la búsqueda no sería difícil, pero si no fuera así, recuperar la información resultaría sumamente complejo.
- Las bases de datos facilitan la organización y reorganización de la información. Los sistemas de archivo en papel nos obligan a acomodar la información en un orden determinado. ¿Debemos organizar la información de los alumnos por nombre, por carrera o por matrícula? De esta decisión depende que después podamos recuperar la información con facilidad. Si la decisión fue ordenar a los alumnos por matrícula, ¿qué pasará cuando busquemos a un alumno por su nombre? Con una base de datos computarizada es posible cambiar al instante de un esquema de organización a otro que tenga la secuencia que deseamos.

- Las bases de datos facilitan la impresión y distribución de información en varias formas. Supongamos que deseamos enviar cartas a cientos de amigos para invitarlos a nuestra fiesta de graduación; tendremos que incluir indicaciones para llegar al lugar solamente a los invitados que no viven en la ciudad, pero no para los que sí viven en ella. Una base de datos, en conjunto con un procesador de textos, puede imprimir formatos de cartas personalizadas, incluso con indicaciones adicionales para quienes las necesiten e imprimir sobres o etiquetas en un momento; además, reducimos la posibilidad de cometer errores.
- Las bases de datos evitan la redundancia de información. Cuando una empresa no tiene la información en una base de datos corre el riesgo de tener confusión en sus datos debido a la redundancia de éstos; por ejemplo, si el departamento de nóminas de una empresa tiene un archivo con todos los datos de los empleados, como su nombre, teléfono, dirección, sueldo, etc., cuando un empleado se muda de domicilio informa el cambio de dirección al departamento de recursos humanos, que a su vez mantiene otra base de datos con los datos del empleado, ¿cómo saber cuál de los dos departamentos tiene los datos correctos?

Las bases de datos que funcionan en una red con muchos usuarios exigen precauciones especiales y seguridad. El acceso a estos datos deberá protegerse mediante claves de acceso, así como usar procedimientos que eliminen los problemas asociados con actualizaciones concurrentes. Una **actualización concurrente** se lleva a cabo cuando dos o más usuarios intentan modificar un registro en el mismo momento desde distintas computadoras; por ejemplo, cuando un empleado de una aerolínea vende un asiento de avión, lo hace porque observa que está disponible, pero ¿qué pasaría si al mismo tiempo otro empleado en otra ciudad hiciera exactamente lo mismo con otro cliente? Este problema puede evitarse cuando se utiliza un *software* de base de datos que bloquee los registros y no permita que dos usuarios los actualicen simultáneamente.

Cuando las bases de datos tenían que programarse en función de las necesidades de cada empresa, era muy difícil justificar su costo; en la actualidad existen muchos paquetes en el mercado que reducen costos y esfuerzo, lo cual hace que las bases de datos sean una aplicación práctica para la mayoría de los usuarios de computadora.

## TERMINOLOGÍA BÁSICA

Las bases de datos computacionales evolucionaron a partir de sistemas de archivos manuales. Los sistemas de archivos manuales y la mayor parte de las bases de datos computarizadas contienen datos organizados en archivos estructurados. Un archivo estructurado usa un formato uniforme para almacenar los datos de cada persona o cosa que figura en el archivo. Veamos uno de estos archivos y observemos los elementos que lo componen.

Matrícula	Nombre	Carrera	Semestre	Edad	Sexo
271111	Pedro Hinojosa	LIN	8	21	M
272222	Juan Reyes	LAE	7	20	M
274343	Fabiola López	CP	9	22	F
278822	Patricia Gómez	LIN	6	19	F
276262	Gonzalo Pérez	LAE	9	21	M
277227	Julia Ruiz	CP	8	19	F
271122	Rosa Álvarez	LIN	7	21	F

Diagrama de etiquetado de la tabla:

- Tabla:** Señala a toda la estructura de la tabla.
- Carácter:** Señala a un solo carácter dentro de una celda.
- Campo:** Señala a una celda completa.
- Dato:** Señala al contenido específico dentro de una celda.
- Registro:** Señala a una fila completa de la tabla.

### Carácter, dato, campo, registro y archivo

Un **carácter** es una letra, un dígito, un símbolo o un espacio en blanco.

Un **dato** es una partida de información individual que no tiene importancia en sí misma; por ejemplo, el dato "21" por sí solo no tiene mucho significado, pero lo adquiere cuando decimos que es la edad de una alumna. Un dato está formado por uno o más caracteres.

Un **campo** contiene la unidad más pequeña de información importante, así como un nombre único que describe su contenido. En nuestro ejemplo el nombre de una persona, su matrícula o su carrera representan campos. Si la base de datos está en forma de tabla, entonces podemos decir que un campo es una columna de la tabla. Un campo puede ser de longitud fija o variable. Un **campo de longitud variable** es como un acordeón que se expande para acomodar los datos que introducimos hasta llegar a un número máximo de caracteres. Un **campo de longitud fija** contiene un número predeterminado de caracteres (*bytes*). Los datos que introducimos en un campo de longitud fija no pueden exceder la longitud asignada al campo.

Un **registro** es toda la información de un elemento que se almacena en un archivo o tabla de la base de datos; por ejemplo, el conjunto de datos que incluye la matrícula, el nombre, la carrera, el semestre, la edad y el sexo de un estudiante forman el registro de ese estudiante. Si la base de datos está en forma de tabla, entonces podemos decir que un registro es un renglón de la tabla. Un **archivo** es un conjunto de todos los registros del mismo tipo en una base de datos.

## CONCEPTOS PARA EL DISEÑO DE LA BASE DE DATOS

La clave para contar con una base de datos efectiva es su diseño inicial. En una base de datos bien diseñada los datos pueden manejarse con flexibilidad para producir información oportuna, significativa y exacta. Un mal diseño puede ocasionar la pérdida de registros e inexactitud en los datos.

El objetivo de un buen diseño de base de datos es almacenar la información de manera que nos permita acceder a ella y darle mantenimiento con facilidad, pero que a la vez sea lo suficientemente concisa para que ocupe el menor espacio posible en el disco. En esta sección veremos las bases de datos desde la perspectiva del diseñador de base de datos y explicaremos cómo crear una estructura eficaz para una base de datos relacional.

### Definición de campos

La estructura de la base de datos es la disposición de los campos, tablas y relaciones de una base de datos. El primer paso al estructurar una base de datos relacional es determinar los datos recopilados y almacenados; para esto, un diseñador de base de datos deberá empezar por consultar a los usuarios y estudiar el sistema actual de archivos para obtener una lista de los datos disponibles, así como cualquier dato adicional necesario para producir una salida en la pantalla o reportes impresos.

Para diseñar la estructura de la base de datos de una librería, tendríamos que reconocer datos como título del libro, nombre del autor, fecha de edición, ISBN, cantidad de ejemplares en almacén, editorial, precio de venta sugerido, precio de descuento y foto de la portada del libro.

Después de que el diseñador de la base de datos determina los datos que tienen que almacenarse, el siguiente paso es organizar los datos por campos. Por lo general, es fácil descomponer los datos en campos utilizando el sentido común y considerando cómo desearía la gente acceder a esa información.

Los campos en una base de datos separan el tipo de información que contiene; por ejemplo, cada registro en el archivo de alumnos tiene matrícula, nombre, carrera, semestre, y hay un campo en cada uno de los registros. Todos los registros de alumnos siguen la misma definición en la secuencia de sus campos; por ejemplo, todos los alumnos deberán tener un número en su campo matrícula y ninguno puede tener un dato alfabético en el lugar de la matrícula.

Los diferentes productos de DBMS ofrecen diversos tipos de campos; los que se emplean más comúnmente son los de texto, los de fecha y los numéricos, aunque en algunas situaciones también se necesitan otros campos especiales.

### Tipos de campos

#### Campos de texto

Un campo de texto almacena una cadena o serie de caracteres alfanuméricos, el cual puede contener en texto el nombre de una persona, de una compañía, una dirección o cualquier otra información importante, así como números, pero se tratan como si fuesen una serie de dígitos y no como el número al que representan, es decir, no podrán utilizarse para efectuar cálculos.

## Campos numéricos y de tipo de moneda

Los **campos numéricos** almacenan números. En casi todos los programas es posible utilizar un formato para mostrar números. El número real del campo que se almacena en el disco no contiene ningún formato, pero cuando el programa muestra en el monitor o en papel dicho número, puede agregar un separador entre los miles y los millones, desplegar o no la precisión a la derecha del punto decimal e incluir otros caracteres especiales, como el signo de pesos. En caso de que el campo almacene números grandes o números que utilicen el punto decimal flotante puede utilizarse un campo tipo numérico, pero con la posibilidad de utilizar más bytes de almacenamiento.

Un campo tipo moneda es un campo decimal, con un formato establecido que representa dinero; este campo muestra sus valores con separadores de coma, dos lugares decimales de precisión para los centavos y un signo de pesos.

## Campos de fecha y hora

Son **campos especializados**. Las fechas y horas se almacenan internamente como un número, pero se muestran como una fecha o una hora. Cuando introducimos un campo de este tipo, el DBMS acepta la entrada del número como fecha u hora, pero convierte el dato en un número antes de almacenarlo en la base de datos; así, los datos ocupan menos espacio en el disco y pueden emplearse para efectuar cálculos. Además, los DBMS permiten hacer una verificación automática de errores de fechas y horas (por ejemplo, cuando introducimos una fecha, el programa verifica el dato para asegurarse de que se trate de una fecha válida).

## Campos de memorando (memo)

Son campos especiales que pueden contener información de longitud variable. En una ficha bibliográfica, por ejemplo, se emplearía un campo tipo memo para la descripción del libro.

## Campos de contador

Estos campos contienen un valor numérico único que el DBMS asigna para cada registro. Cuando introducimos el primer registro en una tabla vacía de la base de datos, se le asigna el número 1 en su campo tipo contador y al segundo registro se le asigna el número 2, y así sucesivamente.

## Campos lógicos

A estos datos también se los llama booleanos y se utilizan para datos con información de **falso/verdadero** o **sí/no** que utilizan un espacio mínimo de almacenamiento; por ejemplo, un diseñador de base de datos podría definir un campo lógico llamado *EnOferta*, el cual podría contener una *S* si un libro se encuentra en oferta o una *N* si el libro no se encuentra en oferta. Algunos archivos y sistemas de administración de base de datos incluyen datos adicionales como los BLOB (*binary large object*) o hiperligas.

## Campos BLOB

Un *binary large object* (BLOB por sus siglas en inglés) es un conjunto de datos binarios almacenados en un solo campo de la base de datos. Un BLOB puede tener cualquier tipo de datos que por lo general almacenaríamos como archivo (por ejemplo, una pista de música MP3). Por ejemplo, una tienda de música en la Web podría almacenar una pequeña muestra de cada canción de un CD en un campo BLOB llamado *MuestraDeLaMúsica*; también se puede incluir un archivo con un gráfico o video, además de objetos OLE, como gráficos u hojas de trabajo creados con una hoja de cálculo o procesador de textos.

## Campos para ligas de hipertexto

El dato liga de hipertexto almacena los URL utilizados para ligar datos directamente desde una base de datos hasta una página Web; por ejemplo, los datos almacenados en un campo hiperliga de una librería podrían proporcionar una liga con el sitio Web del autor de un libro.

La siguiente figura resume cómo se utilizan hoy día los tipos de datos.

Tipo de dato	Descripción	Ejemplo de un campo	Ejemplo de un dato
Real	Números que incluyen punto decimal	<i>PrecioDeDescuento</i>	19.99
Entero	Números enteros	<i>DistanciaKms</i>	500
Fecha	Día, mes y año	<i>FechaDeNacimiento</i>	30/09/1985
Texto	Letras o números que no se utilizan para cálculos	<i>Nombre</i> <i>CódigoPostal</i>	Gonzalo 27200
Lógicos	Datos que pueden tener 1 de 2 valores	<i>EnExistencia</i>	S
Memo	Campo para texto de longitud variable	<i>DescripciónLibroUsado</i>	Portada maltratada y no se encuentra rayado
BLOB	Datos binarios	<i>MuestraDeMúsica</i>	[un archivo MP3]
Hiperliga	Un URL	<i>LigaAlAutor</i>	www.garciamarquez.com

### Campo calculado

Cuando un cliente busca un libro en un sitio Web, encuentra tres partidas de información relacionadas con el precio. La primera es el precio sugerido por la editorial, la segunda es el precio de descuento que ofrece el sitio Web y la tercera es la cantidad de dinero que el cliente ahorra al adquirir el libro por medio de ese sitio Web. Sin embargo, el registro de Libros solamente contiene dos partidas de información relacionadas con el precio: el precio sugerido por la editorial y el precio de descuento. La tercera partida de información, la cantidad que el cliente ahorra, es un campo calculado.

Un **campo calculado** es un cálculo que realiza un DBMS durante el proceso y que temporalmente se almacena en una ubicación de la memoria. Una base de datos debidamente diseñada utiliza campos calculados siempre que es posible porque no requieren espacio de almacenamiento en el disco. La siguiente fórmula muestra cómo un campo calculado genera la cantidad ahorrada con la compra si se adquiere el libro al precio de descuento del sitio.

Fórmula para el cálculo:

$$\text{CantidadAhorrada} = \text{PrecioDeVenta} - \text{PrecioDeDescuento}$$

El campo CantidadAhorrada no debe almacenar ningún dato, ya que el cálculo se efectúa cada vez que se necesita mostrar el resultado.

### Llave primaria

Aunque dos personas podrían tener el mismo nombre o dos cheques podrían tener la misma cantidad, una computadora debe tener la posibilidad de diferenciar los registros. Una **llave primaria** es un campo que contiene datos únicos que hacen que ese registro sea diferente a los demás. Los diseñadores normalmente designan campos como *NúmeroDeCuenta* o *NúmeroDeCliente* como la llave primaria que distingue a un registro de otro.

## NORMALIZACIÓN

El proceso **normalización** ayuda a los diseñadores de bases de datos a crear una estructura en la base de datos que ahorre espacio de almacenamiento y que incremente la eficacia en el proceso. El objetivo de la normalización es reducir al mínimo la redundancia de datos, es decir, la cantidad de datos que se repitan o dupliquen en la base de datos. Una de las principales tareas del diseñador consiste en normalizar la base de datos y decidir la mejor manera de agrupar los datos en las tablas.

El primer paso al agrupar los campos en tablas es obtener una idea del panorama completo de los datos. Con frecuencia los grupos corresponden a los artículos físicos o entidades a los que la base de datos dará seguimiento; por ejemplo, los datos de una librería se agrupan en varias tablas: *Libros*, *Autores*, *Clientes*, *Órdenes* y *DetallesDeLaOrden*. Algunos de estos agrupamientos son obvios, pero otros podrían parecer confusos; por ejemplo, quizá nos preguntemos cuál es la razón para utilizar dos tablas, *Órdenes* y *DetallesDeLaOrden*, para almacenar los datos sobre una orden. ¿Por qué no basta con una tabla? Para responder a la pregunta primero veamos los datos relevantes de cada orden en la siguiente figura.

### LibrosWeb Orden de compra

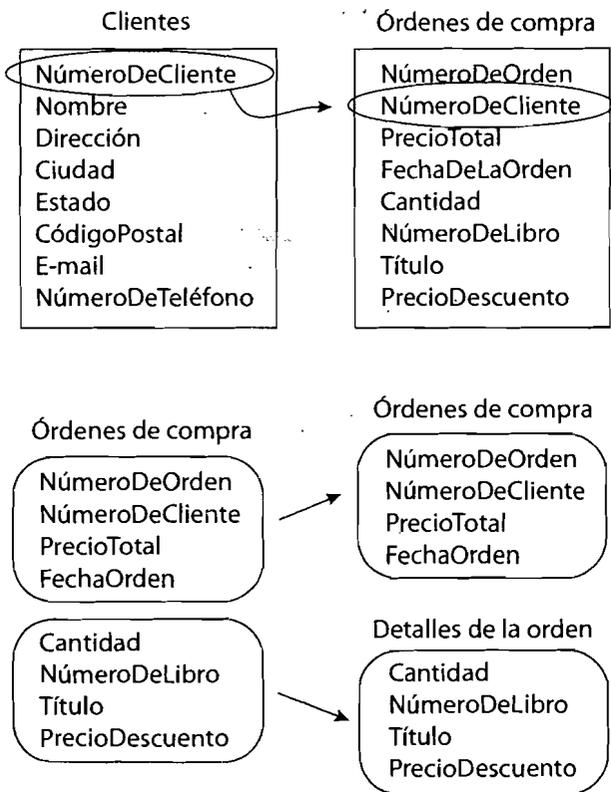
Número de orden	500231	Nombre	Gonzalo Villanueva
Fecha	10/10/2005	Dirección	Juárez 101 Ote.
Número de cliente	155543	Ciudad	Torreón
		Estado	Coahuila
		Código Postal	27650

Cantidad	Número de libro	Título	Precio
1	9789681	Análisis Financiero	\$150.00
1	9788448	Ingeniería de Software	\$180.00
1	9789684	Programación en C++	\$300.00

<b>Total</b>	<b>\$630.00</b>
--------------	-----------------



Si la información del cliente y la información de la orden se agruparan en la misma tabla, cada vez que Gonzalo Villanueva colocara una orden, su nombre, dirección de embarque, dirección de cobranza, número de teléfono y dirección de e-mail deberían introducirse y almacenarse; esta redundancia no sólo exige espacio de almacenamiento extra, sino que puede llevarnos a almacenar datos inconsistentes o inexactos. La solución es crear tablas separadas para *Órdenes* y *Cientes* que se relacionen e incluyan un campo **NúmeroDeCliente** en ambas tablas, como se aprecia en la figura que presentamos a la izquierda.

Incluso después de separar los datos del cliente de los datos de la orden, la estructura de la base de datos puede mejorarse. La tabla *Órdenes* en la figura permite a los clientes ordenar solamente un libro porque los campos para el número de artículo y *PrecioDeDescuento* ocurren solamente una vez. Obviamente, la base de datos necesita manejar órdenes para más de un libro.

Por lo tanto, es razonable proporcionar varios campos para pedir varios libros en una orden, los cuales podrían llamarse **NúmeroDeArtículo1**, **NúmeroDeArtículo2**, **NúmeroDeArtículo3** etc. Entonces, ¿cuántos campos deberá proporcionar el diseñador para ese propósito? Si el diseñador proporciona campos para almacenar 10 artículos, la base de datos no podrá manejar órdenes de más de 10 libros; además, si el cliente ordena menos de 10 libros, el espacio se desperdiciará pues tendrá campos vacíos en cada registro.

Debemos reconocer que existe una relación uno-a-muchos entre una orden y los artículos ordenados. Esta clave nos indica que el diseñador de la base de datos deberá separar los datos en dos tablas: *Órdenes* y *DetallesDeLaOrden*. Estas dos tablas se relacionan por medio del campo *NúmeroDeOrden*. La figura presentada en la página anterior nos muestra cómo se normaliza la tabla en dos tablas para almacenar los datos más eficazmente.

## ORGANIZACIÓN DE LOS REGISTROS

Los registros pueden organizarse en diferentes secuencias, dependiendo de cómo desea utilizarlos la gente; por ejemplo, un cliente que visita el sitio Web para adquirir libros frecuentemente verá la información en la tabla de Libros de acuerdo con el *Nombre* del libro o del *Autor*. El gerente de inventarios desea que los datos se ordenen por *CantidadEnExistencia* para que sea fácil ver qué libro deberá reordenarse. En contraste, el gerente de mercadería está más interesado en la *FechaDeProducción* para asignar a los libros nuevos un precio más agresivo, de manera que no hay una sola forma de organizar los datos que se adapte a las necesidades de todos; lo bueno es que las tablas pueden ordenarse o indexarse de muchas maneras.

### Ordenamiento de una tabla

El orden de clasificación en una tabla es la forma en que se almacenan los registros en el disco. Las tablas clasificadas normalmente permiten hacer consultas y actualizaciones más rápidas porque tienen la ventaja de contar con algoritmos inteligentes que encuentran rápidamente los registros. En una tabla ordenada los registros nuevos se insertan en forma organizada. Si no se especifica el orden de clasificación, los registros nuevos se agregan al final del archivo, lo que da como resultado un archivo que no tiene un orden en particular. Las consultas y actualizaciones dentro de una base de datos sin orden son lentas porque el único algoritmo para buscar en una tabla desordenada es la búsqueda secuencial registro por registro.

### Llave de ordenamiento

La mayor parte de los DBMS utilizan una **llave de ordenamiento** para determinar el orden en que se almacenan los registros; dicha llave consiste en uno o más campos que se usan para especificar dónde se insertarán los nuevos registros en una tabla. Una tabla sólo puede tener una llave de ordenamiento, pero puede modificarse; sin embargo, hacerlo puede llevar mucho tiempo, porque el proceso reordena físicamente los registros en el disco. El diseñador de la base de datos normalmente especifica la llave de ordenamiento de una tabla en una base de datos en el momento de crear la estructura de la base de datos.

### Indexamiento

El índice de una base de datos es muy similar al índice de un libro: contiene una lista de palabras clave e indicadores que remiten a las páginas donde pueden encontrarse esas palabras. Un índice de base de datos contiene una lista de llaves, y cada llave proporciona un indicador que remite al registro que contiene el resto de los campos relacionados con esa llave. La figura siguiente muestra el funcionamiento de un índice.

Llave índice	#
05/03/2000	3
02/02/2002	4
01/02/2004	1

#	NúmeroDeLibro	Título	Autor	Fecha de edición	Existencias	Precio de venta
1	968381376 - 3	¿Provenimos del Mono?	Phill E. Johnson	01/02/2004	40	\$75.00
2	84-666-1595-4	¡Todos a la Calle!	Michael Moore	02/02/2002	30	\$199.00
3	968-13-3939-8	¿Y qué es de la Vida?	Laura Gonzalez Ling	05/02/2000	70	\$99.00
4	84-08-05088-5	¿Arde Nueva York?	Lapierre/Collins	06/08/2003	60	\$269.00

A diferencia de la llave de ordenamiento, un índice nada tiene que ver con la secuencia física de los registros en el disco. Un índice simplemente apunta al registro donde pueden encontrarse los datos. La ventaja de un índice sobre el ordenamiento es que una tabla puede tener múltiples índices, pero sólo un ordenamiento; por ejemplo, la tabla *Libros* podría estar indexada por *TítuloDelLibro* para facilitar las búsquedas de los títulos de libros. La misma tabla pudo haberse indexado por *NombreDelAutor* para facilitar las búsquedas por nombres de autores.

Las tablas de las bases de datos deberán indexarse por cualquier campo o campos que se utilizan comúnmente como campos de búsqueda. El diseñador de la base de datos por lo general crea los índices en el momento en que diseña la estructura de la base de datos, aunque pueden crearse después, en función de las necesidades que vayan surgiendo.

## DISEÑO DE LA INTERFAZ

La manera en que se muestran en pantalla los registros, las consultas y los reportes depende de la interfaz del usuario. Un sistema operativo normalmente proporciona ciertas convenciones para la interfaz del usuario, como estilos de cuadros de diálogo y botones, pero además hay que tomar decisiones adicionales de diseño para la interfaz del usuario de la base de datos.

Diseñar la interfaz del usuario de la base de datos puede constituir todo un desafío. Si la base de datos de una compañía incluye muchas tablas que utilizan diferentes personas, un diseñador profesional de interfaz de usuario debe crearla y darle mantenimiento. Las bases de datos más grandes incluso pueden requerir a todo un grupo de diseñadores de interfaz.

La interfaz de las bases de datos más pequeñas, como las que utilizan negocios pequeños o individuos, las diseña comúnmente un diseñador de bases de datos. Algunos DBMS incluyen herramientas para crear interfaces, otros exigen herramientas separadas para esta tarea.

## CARACTERÍSTICAS DE UNA BUENA INTERFAZ

Una interfaz diseñada adecuadamente debe ser clara, intuitiva y eficaz; para lograrlo, el diseñador debe considerar los siguientes principios:

- Acomodar los campos en un orden lógico empezando por la esquina superior izquierda de la pantalla. Los primeros campos deberán ser los que se utilizan con más frecuencia o los que aparecen primero en la secuencia de entrada de datos.
- Proporcionar claves visuales en las áreas de entrada. Un cuadro de edición, una línea o un área sombreada pueden delinear las áreas para entrada de datos.
- Hacer que las áreas de entrada aparezcan en una posición congruente con sus etiquetas. Por convención, las etiquetas se colocan a la izquierda de las áreas de entrada o sobre ellas.
- Proporcionar una función que permita desplazarse por los campos en forma rápida y ordenada. Por convención, la tecla *tab* desempeña esta función.
- Si no caben en una sola pantalla todos los campos, debe usar un *scroll* o crear una segunda pantalla.
- Proporcionar botones u otros controles fáciles de usar para avanzar de un registro a otro.
- Proporcionar instrucciones en pantalla que permitan introducir correctamente los datos. Las bases de datos en las páginas Web pueden beneficiarse de ligas y páginas de ayuda.

## DISEÑO DE PLANTILLAS PARA REPORTES

### Generador de reportes

Un **generador de reportes** es una herramienta de software que ofrece la posibilidad de crear plantillas de reportes para una base de datos. Aunque los usuarios finales también pueden utilizar los generadores de reportes, por lo general los diseñadores de la base de datos los emplean.

Una plantilla para reportes contiene el resumen o las especificaciones generales para generar un reporte, lo que incluye elementos como título del reporte, los campos que deben incluirse, los campos para subtotalizar o totalizar y las especificaciones para el formato del reporte. Sin embargo, la plantilla no contiene los datos de la base de datos. Los datos se incluyen en la plantilla al generar el reporte.

Cuando se produce el reporte, se basa en los datos reales que contienen las tablas de la base de datos. Como ejemplo podríamos crear una plantilla de reporte llamada *Libros por Autor*, la cual especifica lo siguiente:

- El título del reporte es *Lista de Libros por Autor*.
- El reporte contiene datos de la tabla Libros, ordenada en cuatro columnas con los datos: *Autor*, *Título*, *Cantidad en existencia* y *Precio de descuento*.
- El reporte está agrupado por *Autor*.

Con las especificaciones anteriores debemos generar un reporte similar al que se muestra en la siguiente figura.

<i>Lista de Libros por Autor</i>			
<i>Autor</i>	<i>Título</i>	<i>Cantidad en Existencia</i>	<i>Precio</i>
<i>Carlos Fuentes</i>			
	El naranjo	30	\$89.00
	En esto creo	15	\$169.00
	La quinta compañía	5	\$199.00
	La región más transparente	25	\$129.00
<i>Gabriel García Márquez</i>			
	El amor en tiempos del cólera	50	\$215.00
	Cómo se cuenta un cuento	15	\$109.00
	Crónica de una muerte anunciada	11	\$189.00
<i>Gabriel García Márquez</i>			
	Cien años de soledad	30	\$229.00
<i>Jorge Luis Borges</i>			
	Ficciones	15	\$89.00
<i>Julio Cortázar</i>			
	Las armas secretas	21	\$79.00

Los reportes creados por el generador de reportes pueden mostrarse en la pantalla, imprimir, guardar como archivos o colocar en una página Web. Existe software que proporciona herramientas para mostrar los datos como gráficos o sonidos.

El diseñador de la base de datos puede crear plantillas para reportes que presenten efectivamente la información de acuerdo con los siguientes lineamientos:

- **Proporcionar solamente la información requerida.** Demasiada información hará difícil identificar lo que realmente es esencial.
- **Presentar la información en un formato utilizable.** Por ejemplo, si se necesitan subtotales para tomar una decisión, entonces habrá que incluirlos. La gente que utiliza los reportes no debe efectuar cálculos manuales adicionales.
- **Presentar información oportuna.** Los reportes tienen que llegar a tiempo para que puedan utilizarse y tomar decisiones efectivas. Algunas decisiones exigen información periódica (por ejemplo, los reportes de ventas mensuales); otras exigen información "continua", como los precios actuales de las acciones, la cual sería mejor colocarla en un desplegado continuo.
- **Presentar la información en un formato claro y sin ambigüedades** e incluir títulos, números de página, fechas, etiquetas y encabezados de columna necesarios.
- **Presentar la información en el formato más apropiado para el usuario.** En muchos casos un reporte tradicional organizado por filas y columnas es el más apropiado; en otros los gráficos pueden ser más efectivos.

## Captura de datos

Una vez que el diseño de la estructura de la base de datos está completo, hay que cargar la base de datos con un conjunto inicial de datos; por ejemplo, antes de que la base de datos de libros se ponga en línea, ésta deberá alimentarse con los datos de todos los libros existentes en el inventario.

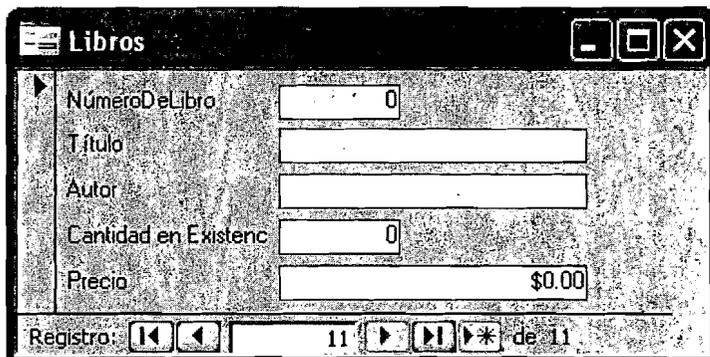
Los datos pueden cargarse en la base de datos manualmente mediante herramientas de entrada de datos genéricas proporcionadas por el DBMS o mediante un módulo de entrada de datos creado por el diseñador de la base de datos. Introducir los datos manualmente puede llevarse mucho tiempo y pueden cometerse errores al oprimir teclas falsas.

Si se cuenta con los datos en formato electrónico en otro tipo de base de datos o en archivos planos, normalmente es posible transferir los datos por medio de una rutina de conversión o de rutinas de importación y exportación de datos. Una rutina de conversión convierte los datos de su formato actual en un formato que pueda incorporarse en forma automática en la nueva base de datos; este proceso lleva algo de tiempo y exige conocimientos sobre los formatos de la base de datos. Sin embargo, si la base de datos es grande, es mucho más rápido convertir los datos que digitarlos manualmente. Convertir los datos puede generar menos errores.

Algunos DBMS ofrecen rutinas de importación y exportación que convierten automáticamente los datos de un tipo de formato de archivo en otro. La rutina de importación trae los datos a la base de datos; por ejemplo, si los datos se encontraban almacenados en un archivo de hoja de cálculo, puede utilizarse la rutina para importar de Microsoft Access los datos desde la hoja de cálculo hasta la base de datos en Access. A la inversa, la rutina para exportar copia los datos desde un paquete de software, como sería una hoja de cálculo, y los deja listos para que puedan utilizarse con la base de datos. Por lo regular utilizamos una de las dos funciones, exportar o importar, pero no ambas.

## MANEJO DE LOS DATOS

### Agregar datos al archivo



El DBMS ofrece posibilidades para que el usuario pueda **agregar** datos al archivo. Muchos paquetes proporcionan una pantalla amigable que permite al usuario agregar registros. Durante esta operación se muestran al usuario los campos en blanco que debe llenar.

### Modificar o borrar

Estos programas también permiten **modificar** o **borrar** los datos. Cuando el usuario requiere la información de cierto registro, el programa le muestra el registro solicitado y le da la oportunidad de efectuar modificaciones o de **eliminarlo**.

### Generar reportes

El usuario puede consultar el sistema si desea buscar algún dato específico. Las **consultas** permiten que el usuario acceda a los datos sin modificarlos; además, el DBMS debe permitir al usuario **generar reportes** impresos que contengan los resultados de sus consultas.

### Consulta o filtro

Una **consulta** o **filtro** es la acción que se realiza para obtener de la base de datos un subconjunto de datos que cumple con ciertas características o restricciones. En nuestro ejemplo podemos efectuar una consulta o filtro para obtener la información de los libros escritos por Carlos Fuentes.

NúmeroDeLibro	Título	Autor	Cantidad en Existencia	Precio
5	El Naranja	Carlos Fuentes	30	\$89.00
6	En esto creo	Carlos Fuentes	15	\$169.00
7	Inquita Compañía	Carlos Fuentes	5	\$199.00
8	La Región Mas Transparente	Carlos Fuentes	25	\$129.00

También podemos efectuar una consulta de las órdenes de compra cuyo *PrecioTotal* sea mayor que \$150.00 y cuya *FechaDeLaOrden* sea posterior a 01/10/2005.

NúmeroDeOrden	NúmeroDeCliente	PrecioTotal	FechaDeLaOrden
58992	2734	\$170.00	01/11/2005
59325	2734	\$250.00	12/11/2005

Por lo general utilizamos las consultas para especificar sólo la información con la que deseamos trabajar, para organizar nuestra información o para responder con rapidez a una pregunta; esto nos permite ver, modificar o analizar datos de diferentes maneras.

## DISEÑO DE LA BASE DE DATOS

### Definición de la estructura de los registros en un archivo

Un diseñador de base de datos define los campos de una base de datos. Esta tarea es similar a diseñar un formulario con campos en blanco, como una solicitud de crédito o un formato de ingreso a una escuela. Cada uno de estos formularios se denomina tipo de registro. Se necesitan diferentes tipos de registro para almacenar los datos de las diferentes entidades; por ejemplo, una librería utiliza un tipo de registro para almacenar la información de los clientes, otro para almacenar la información de los libros y otro para las órdenes de compra. Además se utilizan otros registros para almacenar datos adicionales, como se aprecia en la figura de la derecha.

#### Cientes

NúmeroDeCliente  
Nombre  
Dirección  
Ciudad  
Estado  
CódigoPostal  
E-mail  
NúmeroDeTeléfono

#### Libros

NúmeroDelArtículo  
TituloDelLibro  
Autor  
FechaDeEdición  
ISBN  
CantidadEnExistencia  
Editorial  
PrecioDeVenta  
PrecioDeDescuento  
FotoDeLaPortada

#### Reseñas

NúmeroDelArtículo  
ReseñaDelLibro

#### Órdenes de compra

NúmeroDeOrden  
NúmeroDeCliente  
PrecioTotal  
FechaDeLaOrden

#### Detalles de la orden

NúmeroDeOrden  
NúmeroDeArtículo  
Cantidad  
PrecioDeDescuento

### Relaciones y cardinalidad

Un archivo que contiene solamente un tipo de registro se llama archivo plano. Los archivos planos pueden utilizarse para almacenar datos simples como los nombres y direcciones de un grupo de individuos. En contraste, una base de datos puede contener diversos tipos de registro; por ejemplo, la librería necesita tener un archivo con la información de los libros, otro con la información de los clientes y otro más con la información de las órdenes de compra que efectúan los clientes, por lo que se necesita una base de datos en lugar de un archivo plano.

Una de las características clave de una base de datos es la posibilidad que brinda de mantener relaciones, de manera que los datos de distintos tipos de registros puedan consolidarse en una unidad para consultar los datos y para generar reportes.

En el lenguaje de las bases de datos, una relación es una asociación entre los datos almacenados en distintos tipos de registro; por ejemplo, el registro Clientes de la librería se relaciona con el tipo de registro Órdenes de Compra, porque los clientes efectúan órdenes de compra.

#### Clientes

NúmeroDeCliente	Nombre	Dirección	Ciudad	Estado	CódigoPostal	E-mail	NúmeroDeTeléfono
2734	Jesús Bitar Espada	Zuloaga 260	Torreón	Coahuila	27400	jbitar@hotmail.com	(871)-729-29-20
2745	Alberto Gaitán	Rosas 330	Torreón	Coahuila	27200	agaitan@hotmail.com	(871)-725-25-34
2756	José Huberto García	San Isidro 1235	Torreón	Coahuila	27590	jhung@yahoo.com	(871)-7-21-09-87

#### Órdenes de compra

NúmeroDeOrden	NúmeroDeCliente	PrecioTotal	FechaDeLaOrden
52935	2734	\$940.00	04/08/2005
53112	2734	\$720.00	08/10/2005
58992	2734	\$170.00	01/11/2005
59325	2734	\$250.00	12/11/2005
59322	2745	\$188.00	02/05/2004
51888	2745	\$270.00	02/07/2005

## Cardinalidad

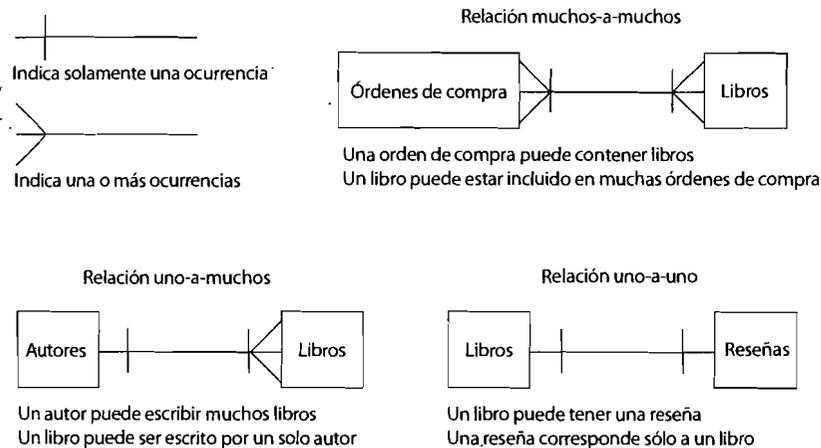
Un aspecto importante de la relación entre los tipos de registros es la **cardinalidad**, la cual es la cantidad de asociaciones que existen entre dos tipos de registros; por ejemplo, un cliente de la librería podría efectuar más de un pedido. Sin embargo, lo contrario no se cumple pues dos clientes no pueden efectuar un pedido en particular. Cuando un solo registro se relaciona con muchos registros se llama **relación uno-a-muchos**.

## Relación de muchos-a-muchos y relación de uno-a-uno

En contraste, una **relación muchos-a-muchos** significa que un tipo particular de registro puede relacionarse con muchos registros de otro tipo y viceversa; por ejemplo, una orden de compra podría relacionarse con varios libros y un libro podría haberse incluido en muchas órdenes de compra.

Una **relación de uno-a-uno** se da cuando un registro de cierto tipo se relaciona únicamente con un registro de otro tipo. Este tipo de relación es infrecuente en el mundo de las bases de datos; algunas veces se utiliza para conservar espacio en disco cuando no se almacenará una partida de información por cada registro en la base de datos; por ejemplo, si el gerente de la librería desea incluir algunas veces una reseña del libro en la base de datos, pero sólo para los libros que tuvieron éxito y que están clasificados como *best sellers*. Si incluimos un campo para la reseña en el tipo de registro *Libros*, este campo permanecería vacío en la mayor parte de los registros. Los registros vacíos ocupan espacio en el disco, por lo que no es deseable tener campos que tengan probabilidades de permanecer en blanco. Si se crea otro tipo de registro llamado *Reseñas*, permitirá almacenar los datos en forma eficaz, ya que sólo los *best sellers* contarán con su correspondiente registro de *Reseña*. La relación entre las tablas *Libros* y *Reseña* será de uno-a-uno, pues un libro se relaciona con una reseña, y una reseña pertenece sólo a un libro.

La relación entre los distintos tipos de registros puede ilustrarse gráficamente en un diagrama de entidad-relación, al que a veces se le llama "diagrama E-R". En la figura siguiente se aprecia cómo se representan estas relaciones en un diagrama E-R.



## BASE DE DATOS RELACIONAL

### Tupla y atributo

Una base de datos relacional almacena los datos en un conjunto de tablas relacionadas; cada una es una secuencia o lista de registros. Todos los registros en la tabla son del mismo tipo. Cada fila de la tabla es equivalente a un registro y se le denomina **tupla**. Cada columna de la tabla es equivalente a un campo, que por lo general se llama **atributo**.

En una base de datos relacional, las relaciones se especifican mediante el uso de datos comunes almacenados en los campos de los registros de diferentes tablas. Este método para establecer relaciones permite que las tablas sean esencialmente independientes, pero es posible consolidarlas para las tareas que se requieran. Se pueden agregar, modificar o borrar relaciones de acuerdo con las necesidades que se tengan. En la figura siguiente se aprecia una relación entre la tabla *Clientes* y

la tabla *Órdenes de Compra* mediante el campo *NúmeroDeCliente*. Además se muestra otra relación entre la tabla *Órdenes de Compra* y la tabla *Detalles de la Orden* mediante el campo *NúmeroDeOrden*. También existe otra relación entre la tabla *Detalles de la Orden* y la tabla *Libros* mediante el campo *NúmeroDeArtículo* (o *NúmeroDeLibro*). Estas relaciones nos permitirían generar reportes que incluyeran datos de varias tablas; por ejemplo, podría generarse un reporte con el nombre del cliente, el número de orden de compra y los títulos de los libros que incluye esa orden de compra.

#### Cientes

NúmeroDeClien	Nombre	Dirección	Ciudad	Estado	CódigoPostal	E-mail	NúmeroDeTeléf
2734	Jesús Bitar Esp	Zuloaga 260	Torreón	Coahuila	27400	jbitar@hotmail.i	(871)-729-29-20
2745	Alberto Gaitán	Rosas 330	Torreón	Coahuila	27200	agaitan@hotmail	(871)-725-25-34
2756	José Huberto G	San Isidro 1235	Torreón	Coahuila	27590	jhumq@yahoo.i	(871)-7-21-09-8

#### Órdenes de compra

NúmeroDeOrden	NúmeroDeCliente	PrecioTotal	FechaDeLaOrden
51888	2745	\$304.00	02/07/2005
52935	2734	\$617.00	04/08/2005
53112	2734	\$109.00	08/10/2005
58992	2734	\$170.00	01/11/2005
59322	2745	\$188.00	02/05/2004
59325	2734	\$250.00	12/11/2005

#### Detalles de la orden

Clave	NúmeroDeOrde	NúmeroDeArtic	Cantidad	PrecioDeDescu
1	51888	1	1	\$215.00
2	51888	5	1	\$89.00
3	52935	2	1	\$229.00
4	52935	7	1	\$199.00
5	52935	4	1	\$189.00
6	53112	3	1	\$109.00

#### Libros

NúmeroDeLibro	Título	Autor	Cantidad en Existencia	Precio
1	Amor en los Tiempos del Cólera	Gabriel García Márquez	50	\$215.00
2	Cien Años de Soledad	Gabriel García Márquez	30	\$229.00
3	Como se Cuenta un Cuento	Gabriel García Márquez	15	\$109.00
4	Crónica de una Muerte Anunciada	Gabriel García Márquez	11	\$189.00
5	El Naranja	Carlos Fuentes	30	\$89.00
6	En esto creo	Carlos Fuentes	15	\$169.00
7	Inquietá Compañía	Carlos Fuentes	5	\$199.00
8	La Región Mas Transparente	Carlos Fuentes	25	\$129.00
9	Las Armas Secretas	Julio Cortazar	21	\$79.00
10	Ficciones	Jorge Luis Borges	15	\$89.00

## BASE DE DATOS ORIENTADA A OBJETOS

Una base de datos orientada a objetos almacena los datos como objetos que pueden agruparse en clases, las cuales pueden definirse por sus atributos y métodos. En el capítulo 4 se aborda la terminología de la orientación a objetos, pero en el contexto de las bases de datos orientadas a objetos, una clase define a un grupo de objetos ya que especifica los atributos y métodos que estos objetos comparten.

Los atributos de un objeto son equivalentes a los campos de una base de datos relacional. Un método es cualquier comportamiento que un objeto asume. Las especificaciones del modelo de la base de datos orientada a objetos las proporcionan organizaciones que crean estándares como el grupo *Object Data Management Group*.

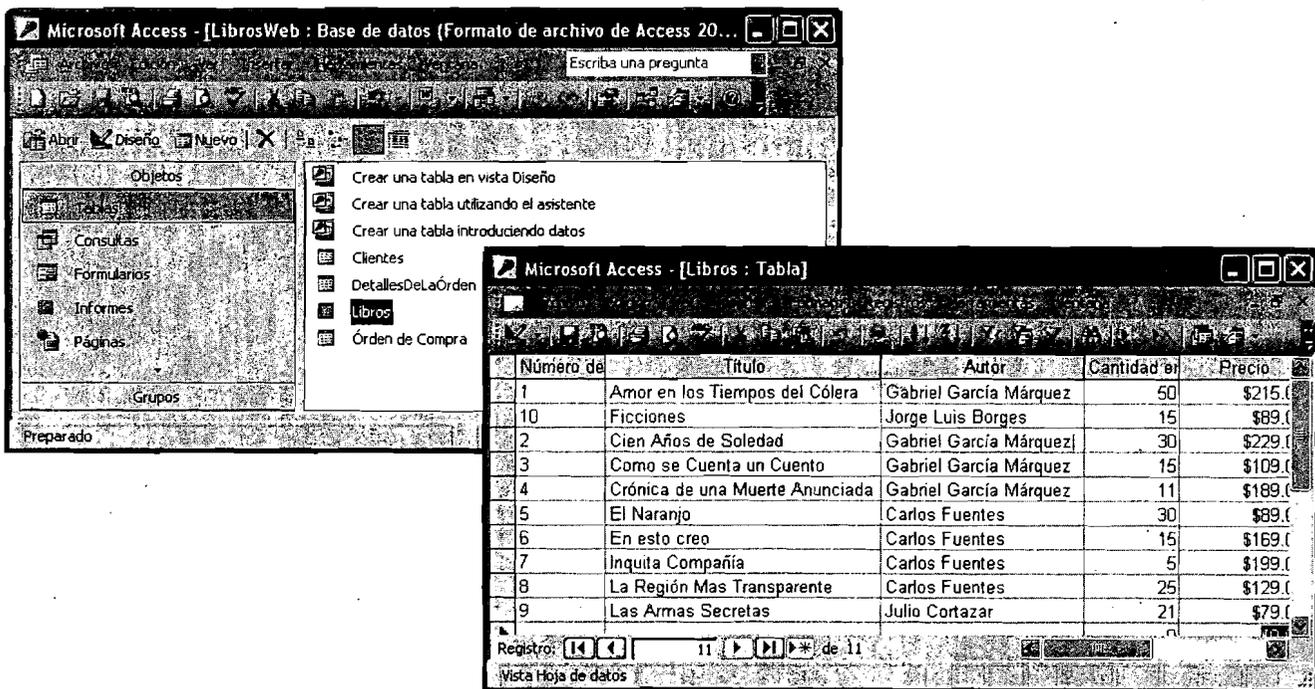
## SISTEMAS DE ADMINISTRACIÓN DE BASES DE DATOS

El término **sistema de administración de bases de datos** (*database management system*, DBMS, por sus siglas en inglés) se refiere al software diseñado para administrar los datos almacenados en una base de datos. Cada DBMS se especializa normalmente en uno de los distintos modelos que existen de bases de datos, pero algunas aplicaciones de base de datos ofrecen mayor versatilidad para ocuparse de distintos modelos y datos.

Por ejemplo, un sistema de manejo de bases de datos XML está optimizado para manejar datos que se encuentran en formato XML (en el siguiente capítulo se describe el formato XML). Un sistema de administración de bases de datos OODBMS (*object-oriented database management system*) es óptimo para el modelo de base de datos orientada a objetos, y nos permite almacenar y manejar clases de datos, atributos y métodos. Un sistema RDBMS (*relational database management system*) nos permite crear, actualizar y administrar una base de datos relacional. En la actualidad, el popular software RDBMS también brinda la posibilidad de manejar clases de objetos y datos XML, por lo que no es necesario adquirir un sistema OODBMS o XML.

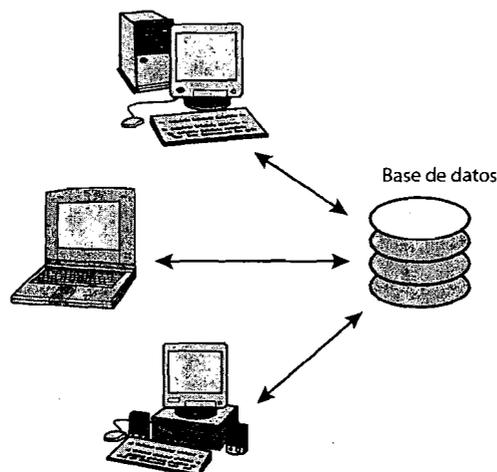
La mayor parte de los proyectos de bases de datos se implementan con un sistema de base de datos relacional; sin embargo, el paquete RDBMS que escojamos dependerá del alcance de nuestro proyecto, la cantidad de personas que accederán simultáneamente a la base de datos y el volumen esperado de registros, consultas y actualizaciones.

Existe software muy accesible como Microsoft Access, que está diseñado para uso personal y en pequeños negocios. Este software incluye todas las herramientas necesarias para manejar los datos de la base de datos, crear los formularios para la entrada de datos, consultar la base de datos y generar reportes. La figura siguiente muestra la pantalla principal de Access con la selección de la tabla *Libros*, la cual se aprecia en la segunda pantalla.



### Software del cliente de la base de datos

Es posible que algunos usuarios accedan al mismo tiempo a una base de datos de este tipo si se instala una aplicación como Microsoft Access en una red. El **software del cliente de la base de datos** permite que cualquier computadora remota o PC en la red acceda a los datos de la base de datos, como se aprecia en la figura de la página siguiente.



Cuando un DBMS de este tipo reside en el servidor de una red puede manejar simultáneamente muchas búsquedas de información; sin embargo, estos DBMS tienen capacidades limitadas para manejar los problemas que surgen cuando muchos usuarios tratan de actualizar el mismo registro en el mismo momento.

### Base de datos distribuida

Cuando el volumen de usuarios que efectúan actualizaciones simultáneas es muy grande, es necesario utilizar un software para servidor de bases de datos como Oracle, IBM DB2 Universal Database o Microsoft SQL Server. Este tipo de software para servidor de bases de datos se diseña para manejar miles de millones de registros y varios cientos de transacciones por segundo. Además, ofrece un rendimiento óptimo en ambientes cliente/servidor como LAN e Internet. Con este software también es posible manejar una **base de datos distribuida**, en la cual la base de datos se almacena en diferentes computadoras, en distintas redes y en distintas ubicaciones. Como se aprecia en la figura siguiente, el software del servidor de bases de datos reemplaza las funciones del cliente e interactúa directamente con la base de datos mientras los usuarios siguen comunicándose con el DBMS a través del software del cliente.

