



CONCEPTOS Y DISEÑO DE BASES DE DATOS

2

Conceptos básicos: tablas, campos y registros

Toda base de datos está formada por uno o varios bloques de información llamados TABLAS (inicialmente denominados FICHEROS o ARCHIVOS) que normalmente tendrán alguna característica en común.

Una TABLA o archivo de datos es un conjunto conexo de información del mismo tipo; por ejemplo, en una base de datos de una biblioteca, una tabla estará constituida por la información relativa a todos los libros de la misma, otra tabla contendrá información sobre los lectores, etc.

Cada tabla está formada por REGISTROS. Un registro es la unidad elemental de información de la tabla o fichero (en un archivo clásico no automatizado, un registro se corresponde con lo que suele llamarse ficha). En la tabla o fichero de libros, un registro estaría constituido por la información correspondiente a cada libro concreto, con su título, autor, área, editorial, etc.

Cada registro está formado por uno o más elementos llamados CAMPOS. Un campo es cada una de las informaciones que interesa almacenar en cada registro y es, por tanto, la unidad elemental de información del registro. En el ejemplo anterior, un campo sería el título del libro; otro campo su autor, etc.

Gracias a la aparición de los llamados programas de usuario es posible realizar la gestión de tablas de una base de datos; sin tener que realizar programas que procesen esos datos, facilitando todas las operaciones de creación, actualización, consulta y creación de informes con los datos recogidos.

La forma de almacenar los ficheros, registros y campos es distinta, desde el punto de vista técnico, dependiendo de las características de la información que se quiere tratar. Sea cual sea la estructura de la base de datos, siempre se pueden incluir ficheros de índices secundarios para facilitar una respuesta eficaz a determinados tipos de acceso.

Diseño de una base de datos relacional

Lo primero que hay que tener presente a la hora de diseñar una base de datos relacional es el propio concepto de modelo relacional, que organiza los datos en una base de datos como una colección de tablas teniendo presente inicialmente lo siguiente:

- Cada tabla tiene un nombre que la identifica únicamente.
- Cada tabla tiene una o más columnas nominadas, que están dispuestas en un orden específico de izquierda a derecha.
- Cada tabla tiene cero o más filas, conteniendo cada una un único valor en cada columna. Las filas están desordenadas.
- Todos los valores de una columna determinada tienen el mismo tipo de datos y éstos están extraídos de un conjunto de valores legales llamado el dominio de la columna.

A su vez, las tablas están relacionadas unas con otras por los datos que contienen. El modelo de datos relacional utiliza claves primarias y claves secundarias (externas o foráneas) para representar estas relaciones entre tablas. A la hora de definir las claves primarias y secundarias es necesario tener presente de entrada lo siguiente:

- Una clave primaria es una columna o combinación de columnas dentro de una tabla cuyo(s) valor(es) identifica(n) únicamente a cada fila de la tabla. Cada tabla tiene una única clave primaria.
- Una clave secundaria es una columna o combinación de columnas en una tabla cuyo(s) valor(es) es(son) un valor de clave primaria para alguna otra tabla. Una tabla puede contener más de una clave secundaria, enlazándola a una o más tablas.
- Una combinación clave primaria/clave secundaria crea una relación padre/hijo entre las tablas que las contienen.

Integridad

El término *integridad de datos* se refiere a la corrección y completitud de los datos en una base de datos. Cuando los contenidos de una base de datos se modifican con sentencias INSERT (insertar), DELETE (borrar) o UPDATE (actualizar), la integridad de los datos almacenados puede perderse de muchas maneras diferentes. Por ejemplo:

- Pueden añadirse datos no válidos a la base de datos, tales como un pedido que especifica un producto no existente.
- Pueden modificarse datos existentes tomando un valor incorrecto, como por ejemplo, si se reasigna un vendedor a una oficina no existente.

- Los cambios a la base de datos pueden perderse debido a un error del sistema o a un fallo en el suministro de potencia.
- Los cambios pueden ser aplicados parcialmente, como por ejemplo, si se añade un pedido de un producto sin ajustar la cantidad disponible para vender.

Una función importante de un DBMS (sistema gestor de la base de datos) relacional es preservar la integridad de los datos almacenados en la mayor medida posible.

Índices

Un índice es una estructura interna que el sistema puede usar para encontrar uno o más registros en una tabla de forma rápida. En efecto, un índice de base de datos es, conceptualmente, similar a un índice encontrado al final de cualquier libro de texto. De la misma forma que el lector de un libro acudiría a un índice para determinar en qué páginas se encuentra un determinado tema, un sistema de base de datos leerá un índice para determinar las posiciones de registros seleccionados por una consulta. En otras palabras, la presencia de un índice puede ayudar al sistema a procesar algunas consultas de un modo más eficiente.

Un índice de base de datos se crea para una columna o grupo de columnas. Es más fácil que el sistema utilice el índice para localizar un registro con un valor dado, a que explore toda la tabla en busca de ese valor. A un índice se le asigna un nombre, de acuerdo con las mismas reglas que se aplican para los nombres de tablas. El nombre debe comenzar con una letra que puede ir seguida de una o más letras, dígitos o caracteres de subrayado. Su longitud no puede exceder de un determinado número de caracteres. Suele seguirse una convención común de elegir "X" como la primera letra seguida del nombre de columna para formar el nombre del índice. Sin embargo, esta convención no es indispensable para el sistema. Una vez que se ha creado el índice, el sistema lo mantendrá automáticamente.

La mayoría de los usuarios del sistema puede que no perciban la existencia de índices de base de datos. Siempre que una sentencia de selección sea introducida por el usuario, el sistema usa un módulo interno especial, llamado *optimizador* que analiza la sentencia para determinar si el uso de un índice sería beneficioso. Si éste es el caso, el sistema consultará el catálogo para determinar si se encuentra disponible un índice. Si lo está, el sistema utilizará el índice. En caso contrario, el sistema realizará una operación de exploración en busca de los registros deseados.

El uso de un *optimizador* para determinar el camino de acceso más eficaz, significa que el sistema posee la importante característica de la independencia física de los datos. El administrador de la base de datos puede modificar las estructuras internas de datos sin producir cambios en los programas de aplicaciones escritos anteriormente. En concreto, los índices pueden ser creados y asignados sin que afecten a la validez de otras sentencias ya existentes o de futuras sentencias de selección.

Ejemplo de diseño de una base de datos relacional

Consideremos la base de datos EMPLEADOS que contiene información correspondiente a una sencilla aplicación de procesamiento de pedidos para una pequeña empresa de distribución. Consta de cinco tablas:

CLIENTES, que contiene una fila por cada uno de los clientes de la empresa. Sus campos son NUM_CLIE (número de cliente), EMPRESA, REP_CLIENTE (número de empleado del representante que atiende al cliente) y LIM_CREDITO (límite de crédito).

REPVENTAS, que contiene una fila por cada uno de los diez vendedores de la empresa. Sus campos son NUM_EMPL (número de empleado), NOMBRE, EDAD, OFICINA REP (número de oficina de representación del vendedor), TÍTULO, CONTRATO, DIRECTOR, CUOTA (ventas previstas) y VENTAS (ventas realizadas).

OFICINAS, que contiene una fila por cada una de las cinco oficinas en las que trabajan los vendedores. Sus campos son OFICINA (número de oficina), CIUDAD, REGIÓN, DIR (número de empleado del director), OBJETIVO (ventas anuales previstas) y VENTAS (ventas anuales realizadas).

PRODUCTOS, que contiene una fila por cada tipo de producto disponible para la venta. Sus campos son ID_FAB (identificador de fabricante), ID_PRODUCTO (identificador de producto), DESCRIPCIÓN, PRECIO y EXISTENCIA.

PEDIDOS, que contiene una fila por cada pedido ordenado por un cliente. Por simplicidad, se supone que cada pedido se refiere a un solo producto. Sus campos son NUM_PEDIDO (número de pedido), FECHA_PEDIDO (fecha del pedido), CLIE (número de cliente), REP (número de empleado del representante), FAB (identificador de fabricante), PRODUCTO (identificador de producto), CANT (cantidad) e IMPORTE.

Los datos son los siguientes:

REPVENTAS

NUM_EMPL	NOMBRE	EDAD	OFICINA REP	TÍTULO	CONTRATO	DIRECTOR	CUOTA	VENTAS
105	Bill Adams	37	13	Rep Ventas	12 - FEB - 88	104	350.000,00	367.911,00
109	Mary Jones	31	11	Rep Ventas	12 - OCT - 99	101	300.00,00	392.725,00
102	Sue Smith	48	21	Rep Ventas	10 - DIC - 86	108	350.000,00	474.050,00
106	Sam Clark	52	11	VP Ventas	14 - JUN - 88	NULL	275.000,00	299.912,00
104	Bob Smith	33	12	Dir Ventas	19 - MAY - 87	106	200.000,00	142.594,00
101	Dan Roberts	45	12	Rep Ventas	20 - OCT - 86	104	300.000,00	305.673,00
110	Tom Snyder	41	NULL	Rep Ventas	13 - ENE - 90	101	NULL	75.985,00
108	Larry Fitch	62	21	Dir Ventas	12 - OCT - 89	106	350.000,00	361.865,00
103	Paul Cruz	29	12	Rep Ventas	01 - MAR - 87	104	275.000,00	286.775,00
107	Nancy Angelli	49	22	Rep Ventas	14 - NOV - 88	108	300.000,00	186.042,0

PRODUCTOS

ID_FAB	ID_PRODUCTO	DESCRIPCIÓN	PRECIO	EXISTENCIAS
REI	2A445C	V Stago Trinquete	79,00	210
ACI	4100Y	Extractor	2.750,00	25
QSA	XK47	Reductor	355,00	38
BIC	41672	Plate	180,00	0
IMM	779C	Riostra 2-Tm	1.875,00	9
ACI	41003	Articulo Tipo3	107,00	207
ACI	41004	Articulo Tipo4	117,00	139
BIC	41003	Manivela	652,00	3
IMM	887P	Perno Riostra	250,00	24
QSA	XK48	Reductor	134,00	203
REI	2A444L	Bisagra Izqda.	4.500,00	12
FEA	112	Cubierta	148,00	115
IMM	887H	Soporte Riostra	54,00	223
BIC	41089	Retn	225,00	78
ACI	41001	Articulo Tipo1	55,00	277
IMM	775C	Riostra 1-Tm	1.425,00	5
ACI	4100Z	Montador	2.500,00	28
QSA	XK484	Reductor	117,00	37
ACI	41002	Articulo Tipo2	76,00	167
REI	2A444R	Bisagra Dcha.	4.500,00	12
IMM	773C	Riostra 1/2-Tm	975,00	28
ACI	4100X	Ajustador	25,00	37
FEA	114	Bancada Motor	243,00	15
IMM	887X	Retenedor Riostra	475,00	32
REI	2A444G	Pasador Bisagra	350,00	14

CLIENTES

NUM_CLIE	EMPRESA	REP_CLIE	LIMITE_CREDITO
2111	JCP Inc.	103	50.000,00
2102	Firts Corp.	101	65.000,00
2103	Acme Mfg.	105	50.000,00
2123	Carter & Sons	102	40.000,00
2107	Ace International	110	35.000,00
2115	Smithson Corp.	101	20.000,00
2101	Jones Mfg.	106	65.000,00
2112	Zetacorp	108	50.000,00
2121	QMA Assoc.	103	45.000,00
2114	Orion Corp	102	20.000,00
2124	Peter Brothers	107	40.000,00
2108	Holm & Landis	109	55.000,00
2117	J.P. Sinclair	106	35.000,00
2122	Three-Way Lines	105	30.000,00
2120	Rico Enterprises	102	50.000,00
2106	Fred Lewis Corp.	102	65.000,00
2119	Solomon Inc.	109	25.000,00
2118	Midwest Systems	108	60.000,00
2113	Ian & Schmidt	104	20.000,00
2109	Chen Associates	103	25.000,00
2105	AAA Investments	101	45.000,00

PEDIDOS

NUM_PEDIDO	FECHA_PEDIDO	CLIE	REP	FAB	PRODUCTO	CANT	IMPORTE
112961	17-DIC-89	2117	106	REI	2A44L	7	31.500,00
113012	11-ENE-90	2111	105	ACI	41003	35	3.745,00
112989	03-ENE-90	2101	106	FEA	114	6	1.458,00
113051	10-FEB-90	2118	108	QSA	K47	4	1.420,00
112968	12-OCT-89	2102	101	ACI	41004	34	3.978,00
110036	30-ENE-90	2107	110	ACI	4100Z	9	22.500,00
113045	02-FEB-90	2112	108	REI	2A44R	10	45.000,00
112963	17-DIC-89	2103	105	ACI	41004	28	3.276,00
113013	14-ENE-90	2118	108	BIC	41003	1	652,00
113058	23-FEB-90	2108	109	FEA	112	10	1.480,00
112997	08-ENE-90	2124	107	BIC	41003	1	652,00
112983	27-DIC-89	2103	105	ACI	41004	6	702,00
113024	20-ENE-90	2114	108	QSA	XK47	20	7.100,00
113062	24-FEB-90	2124	107	FEA	114	10	2.430,00
112979	12-OCT-89	2114	102	ACI	4100Z	6	15.000,00
113027	22-ENE-90	2103	105	ACI	41002	54	4.104,00
113007	08-ENE-90	2112	108	IMM	773C	3	2.925,00
113069	02-MAR-90	2109	107	IMM	775C	22	31.350,00
113034	29-ENE-90	2107	110	REI	2A45C	8	632,00
112992	04-NOV-89	2118	108	ACI	41002	10	760,00
112975	12-OCT-89	2111	103	REI	2A44G	6	2.100,00
113055	15-FEB-90	2108	101	ACI	4100X	6	150,00
113048	10-FEB-90	2120	102	IMM	779C	2	3.750,00
112993	04-ENE-89	2106	102	REI	2A45C	24	1.896,00
113065	27-FEB-90	2106	102	QSA	XK47	6	2.130,00
113003	25-ENE-90	2108	109	IMM	779C	3	5.625,00
113049	10-FEB-90	2118	108	QSA	XK47	2	776,00
112987	31-DIC-89	2103	105	ACI	4100Y	11	27.500,00
113057	18-FEB-90	2111	103	ACI	4100X	24	600,00
113042	02-FEB-90	2113	101	REI	2A44R	5	22.500,00

OFICINAS

OFICINA	CIUDAD	REGIÓN	DIR	OBJETIVO	VENTAS
22	Denver	Oeste	108	300.000,00	186.042,00
11	New York	Este	106	575.000,00	692.637,00
12	Chicago	Este	104	800.000,00	735.042,00
13	Atlanta	Este	105	350.000,00	367.911,00
21	Los Angeles	Oeste	108	725.000,00	835.915,00

En una base de datos relacional bien diseñada cada tabla tiene una columna o combinación de columnas cuyos valores identifican únicamente cada fila en la tabla. Esta columna (o columnas) se denomina *clave primaria* de la tabla. Observemos en primer lugar la tabla OFICINAS. A primera vista, tanto la columna OFICINA como la columna CIUDAD podrían servir como clave primaria para la tabla, pero si la empresa se amplía y abre dos oficinas de ventas en la misma ciudad, la columna CIUDAD ya no podría servir como clave primaria. En la práctica, “números de ID”, tales como el número de oficina (OFICINA en la tabla OFICINAS), el número de empleado (NUM_EMP en la tabla REPVENTAS) y los números de clientes (NUM_CLIE en la tabla CLIENTES) se eligen con frecuencia como claves primarias. En el caso de la tabla PEDIDOS no hay elección, ya que lo único que identifica únicamente un pedido es el número de pedido (NUM_PEDIDO).

La tabla PRODUCTOS es un ejemplo de una tabla en donde la clave primaria debe ser una *combinación* de columnas. La columna ID_FAB identifica al fabricante de cada producto en la tabla y la columna ID_PRODUCT especifica el número de producto del fabricante. La columna ID_PRODUCT podría ser una buena clave primaria, pero no hay nada que impida que dos fabricantes diferentes utilicen el mismo número para sus productos. Por tanto, debe utilizarse una combinación de las columnas ID_FAB e ID_PRODUCT como clave primaria de la tabla PRODUCTOS. Cada producto de la tabla se garantiza que tiene una combinación única de valores en estas dos columnas.

En cuanto a las claves secundarias, tendremos en cuenta que una columna de una tabla cuyo valor coincide con la clave primaria de alguna otra tabla se denomina una *clave secundaria, externa o foránea*. En nuestra base de datos, la columna OFICINA REP es una clave foránea para la tabla OFICINAS. Aunque es una columna en la tabla REPVENTAS, los valores que esta columna contiene son números de oficina. Coinciden con valores en la columna OFICINA, que es la clave primaria para la tabla OFICINAS. Juntas, una clave primaria y una clave foránea crean una relación padre/hijo entre las tablas que las contienen, del mismo modo que las relaciones padre/hijo de una base de datos jerárquica. Por la misma razón, la columna DIR de la tabla OFICINAS es una clave foránea para la tabla REPVENTAS, ya que los valores que esta columna contiene son números de empleado que coinciden con valores de la columna NUM_EMPL que es clave primaria en la tabla REPVENTAS. En idéntica situación está la columna REP_CLIE de la tabla CLIENTES, que también es una clave foránea para la tabla REPVENTAS, ya que los valores que esta columna contiene son números de empleado que coinciden con valores de la columna NUM_EMPL que es clave primaria en la tabla REPVENTAS. Incluso la columna DIRECTOR de la tabla REPVENTAS puede considerarse como una clave foránea de la propia tabla REPVENTAS, ya que los valores que esta columna contiene son números de empleado que coinciden con valores de la columna NUM_EMPL que es clave primaria en la tabla REPVENTAS.

Lo mismo que una combinación de columnas puede servir como clave primaria de una tabla, una clave foránea puede ser también una combinación de columnas. De hecho, la clave foránea será *siempre* una clave compuesta (multicolumna) cuando referencia a una tabla con una clave primaria compuesta. Obviamente el número de columnas y los tipos de datos de las columnas en la clave foránea y en la clave primaria deben ser idénticos unos a otros.

Una tabla puede contener más de una clave foránea si está relacionada con más de una tabla adicional. En nuestra base de datos podemos definir tres claves foráneas en la tabla PEDIDOS. Serían las siguientes:

- La columna CLIE es una clave foránea para la tabla CLIENTES, que relaciona cada pedido con el cliente que lo remitió.

- La columna REP es una clave foránea para la tabla REPVENTAS que relaciona cada pedido con el vendedor que lo tomó.
- Las columnas FAB y PRODUCTO juntas son una clave foránea compuesta para la tabla PRODUCTOS, que relacionan cada pedido con el producto solicitado.

Una vez discutido el diseño de la base de datos, podemos representarlo como se indica en la Figura 2-1.

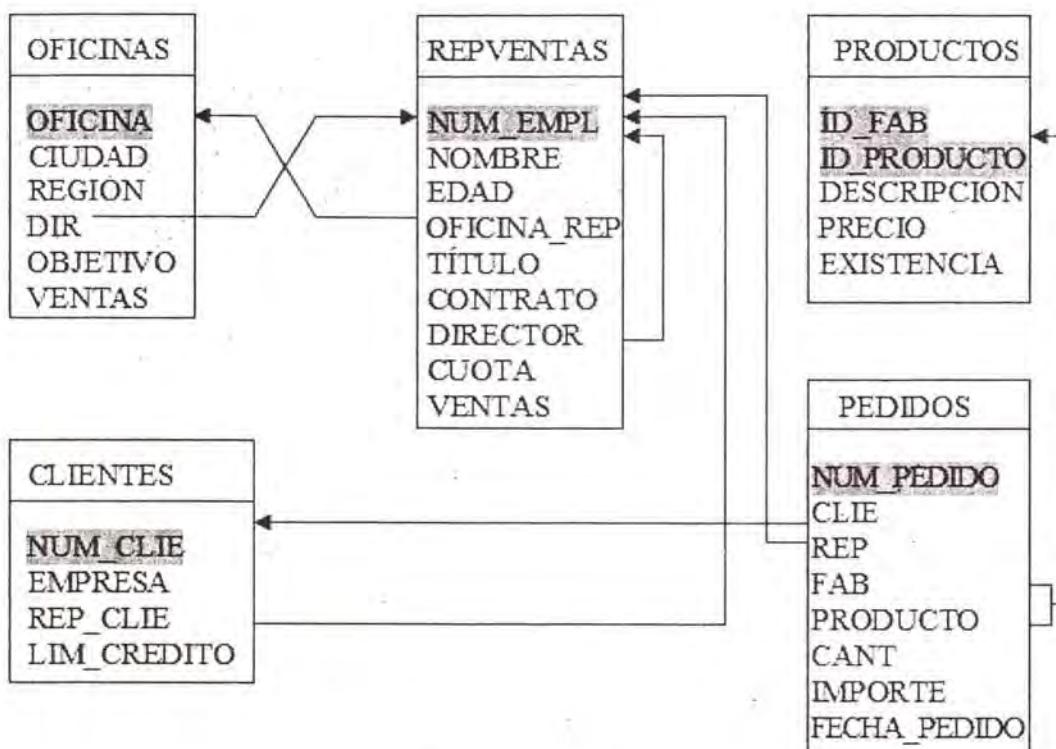


Figura 2-1

Objetos típicos en bases de datos Access

Office Access 2007 es un sistema de administración de bases de datos (DBMS) que puede ajustarse al modelo relacional. Una base de datos Access almacena sus tablas (en el modelo relacional cada base de datos puede contener varias tablas) en un solo archivo, junto con otros objetos, como formularios, informes, macros y módulos. Las bases de datos creadas con formato de Access 2007 tienen la extensión de nombre de archivo *.accdb* y las bases de datos creadas con formatos de versiones anteriores de Access tienen la extensión de nombre de archivo *.mdb*. Access 2007 se puede utilizar para crear archivos con formatos de versiones anteriores (por ejemplo, Access 2000 y Access 2002-2003).

Con Access es sencillo realizar las siguientes tareas:

- Agregar más datos a una base de datos, por ejemplo, un elemento nuevo en un inventario.
- Modificar datos existentes en la base de datos, por ejemplo, cambiar la ubicación de un elemento.
- Eliminar información, por ejemplo, si se ha vendido o retirado un artículo.
- Organizar y ver los datos de distintas formas.
- Compartir los datos con otros usuarios mediante informes, mensajes de correo electrónico, una Intranet o Internet.

Tablas

Una tabla de una base de datos es similar en apariencia a una hoja de cálculo, en cuanto a que los datos se almacenan en filas y columnas. Como consecuencia, normalmente es bastante fácil importar una hoja de cálculo en una tabla de una base de datos. La principal diferencia entre almacenar los datos en una hoja de cálculo y hacerlo en una base de datos es la forma de organizarse los datos. Para lograr la máxima flexibilidad para una base de datos, la información tiene que estar organizada en tablas (Figura 2-2), para que no haya redundancias. Por ejemplo, si se almacena información sobre empleados, cada empleado se insertará una sola vez en una tabla que se configurará para contener únicamente datos de los empleados. Los datos sobre productos se almacenarán en su propia tabla, y los datos sobre sucursales también tendrán su tabla aparte. Este proceso se conoce como *normalización*.

Cada fila de una tabla se denomina registro. En los registros es donde se almacena cada información individual. Cada registro consta de campos (al menos uno). Los campos corresponden a las columnas de la tabla. Por ejemplo, puede trabajar con una tabla denominada "EMPLEADOS", en la que cada registro (fila) contiene información sobre un empleado distinto y cada campo (columna) contiene un tipo de información diferente, como el nombre, los apellidos, la dirección, o similares. Los campos se deben configurar con un determinado tipo de datos, ya sea texto, fecha, hora, numérico, o cualquier otro tipo.

Otra forma de describir registros y campos es imaginando un catálogo de fichas tradicional de una biblioteca. Cada ficha del armario corresponde a un *registro* de la base de datos. Cada información contenida en una ficha (autor, título, etc.) corresponde a un *campo* de la base de datos.



Figura 2-2

Formularios

Los formularios se conocen a veces como "pantallas de entrada de datos". Son las interfaces que se utilizan para trabajar con los datos y, a menudo, contienen botones de comando que ejecutan diversos comandos. Se puede crear una base de datos sin usar formularios, editando los datos de las hojas de las tablas. No obstante, casi todos los usuarios de bases de datos prefieren usar formularios para ver, escribir y editar datos en las tablas.

Los formularios proporcionan un formato fácil de utilizar para trabajar con los datos (Figura 2-3). Además, se les puede agregar elementos funcionales, como botones de comando. Puede programar los botones para determinar qué datos aparecen en el formulario, abrir otros formularios o informes, o realizar otras tareas diversas. Por ejemplo, podría crear un formulario denominado "Formulario de cliente" para trabajar con datos de clientes. El formulario de cliente podría tener un botón para abrir un formulario de pedido en el que se pudiese escribir un pedido nuevo del cliente.

Los formularios también permiten controlar la manera en que otros usuarios interactúan con los datos de la base de datos. Por ejemplo, puede crear un formulario que muestre únicamente ciertos campos y que permita la ejecución de determinadas operaciones solamente. Así, se favorece la protección de los datos y se facilita la entrada correcta de datos.

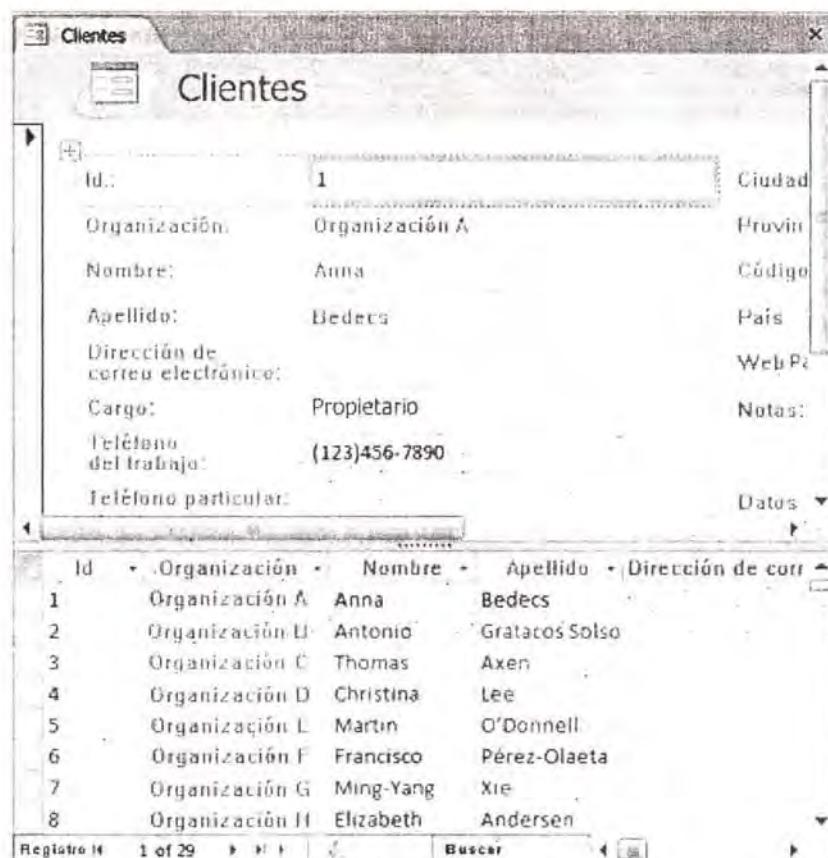


Figura 2-3

Informes

Los informes (Figura 2-4) sirven para resumir y presentar los datos de las tablas. Normalmente, un informe responde a una pregunta específica, como "¿Cuánto dinero se ha facturado por cliente este año?" o "¿En qué ciudades están nuestros clientes?". Cada informe se puede diseñar para presentar la información de la mejor manera posible. Un informe se puede ejecutar en cualquier momento y siempre reflejará los datos actualizados de la base de datos. Los informes suelen tener un formato que permite imprimirllos, pero también se pueden consultar en la pantalla, exportar a otro programa o enviar por correo electrónico. Después de elegir el origen de los registros, normalmente es más sencillo crear el informe utilizando un asistente para informes. El Asistente para informes es una característica de Access que le guía por una serie de preguntas y, a continuación, genera un informe tomando como base las respuestas proporcionadas.

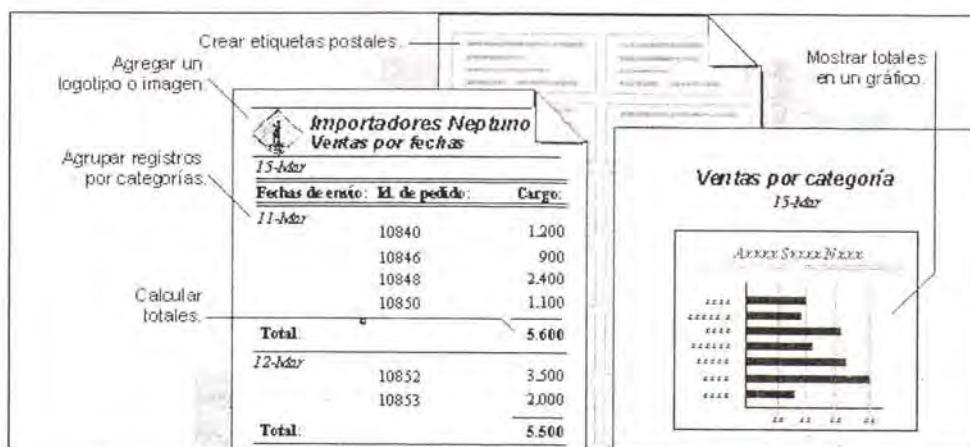


Figura 2-4

Consultas

Las consultas (Figura 2-5) son los objetos que verdaderamente hacen el trabajo en una base de datos. Pueden realizar numerosas funciones diferentes. Su función más común es recuperar datos específicos de las tablas. Los datos que desea ver suelen estar distribuidos por varias tablas y, gracias a las consultas, puede verlos en una sola hoja de datos. Además, puesto que normalmente no desea ver todos los registros a la vez, las consultas le permiten agregar criterios para "filtrar" los datos hasta obtener solo los registros que deseé. Las consultas a menudo sirven de origen de registros para formularios e informes.

Algunas consultas son "actualizables", lo que significa que es posible editar los datos de las tablas base mediante la hoja de datos de la consulta. Si trabaja con una consulta actualizable, recuerde que los cambios se producen también en las tablas, no solo en la hoja de datos de la consulta.

Hay dos tipos básicos de consultas: las de selección y las de acción. Una consulta de selección simplemente recupera los datos y hace que estén disponibles para su uso. Los resultados de la consulta pueden verse en la pantalla, imprimirse o copiarse al portapapeles, o se pueden utilizar como origen de registros para un formulario o un informe. Una consulta de acción, como su nombre indica, realiza una tarea con los datos. Las consultas de acción pueden servir para crear tablas nuevas, agregar datos a tablas existentes, actualizar datos o eliminar datos.

Campo:	Elemento	Ubicación	Fecha de adquisición
Tabla:	Activos	Activos	Activos
Actualizar a:		"Almacén 3"	
Criterios:			>#1/5/2005#
O:			

Figura 2-5

Macros

Las macros en Access (Figura 2-6) se pueden considerar como un lenguaje de programación simplificado, que se puede utilizar para aumentar la funcionalidad de la base de datos. Por ejemplo, puede adjuntar una macro a un botón de comando en un formulario, de modo que la macro se ejecute cuando se haga clic en el botón. Las macros contienen acciones que realizan tareas, como abrir un informe, ejecutar una consulta o cerrar la base de datos. Casi todas las operaciones de bases de datos que normalmente se realizan manualmente se pueden automatizar mediante macros, ahorrando así mucho tiempo.

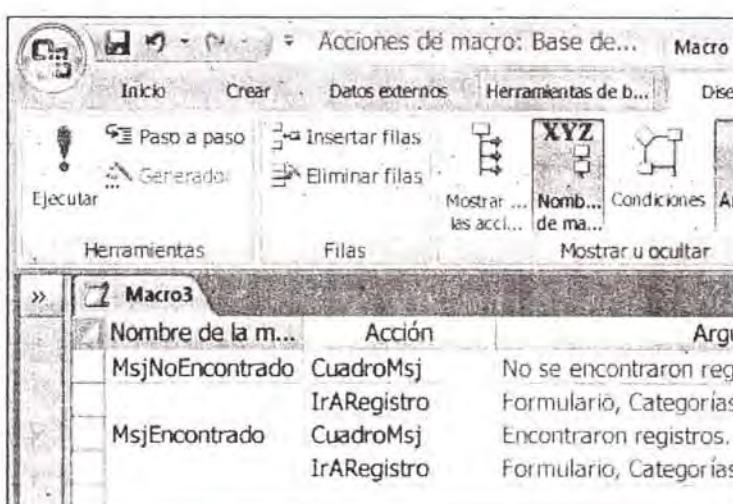


Figura 2-6

Módulos

Los módulos, como las macros, son objetos que sirven para aumentar la funcionalidad de la base de datos. Mientras que las macros en Access se crean seleccionando acciones de una lista, los módulos se escriben en el lenguaje de programación de Visual Basic para Aplicaciones (VBA).

Un módulo es una colección de declaraciones, instrucciones y procedimientos que se almacenan conjuntamente como una unidad (Figura 2-7). Un módulo puede ser de clase o estándar.

Los módulos de clase se adjuntan a formularios o informes, y normalmente contienen procedimientos específicos del formulario o el informe al que se adjuntan.

Los módulos estándar contienen procedimientos generales que no están asociados a ningún otro objeto. Los módulos estándar se enumeran en la opción *Módulos* en el panel de exploración, pero los módulos de clase no.

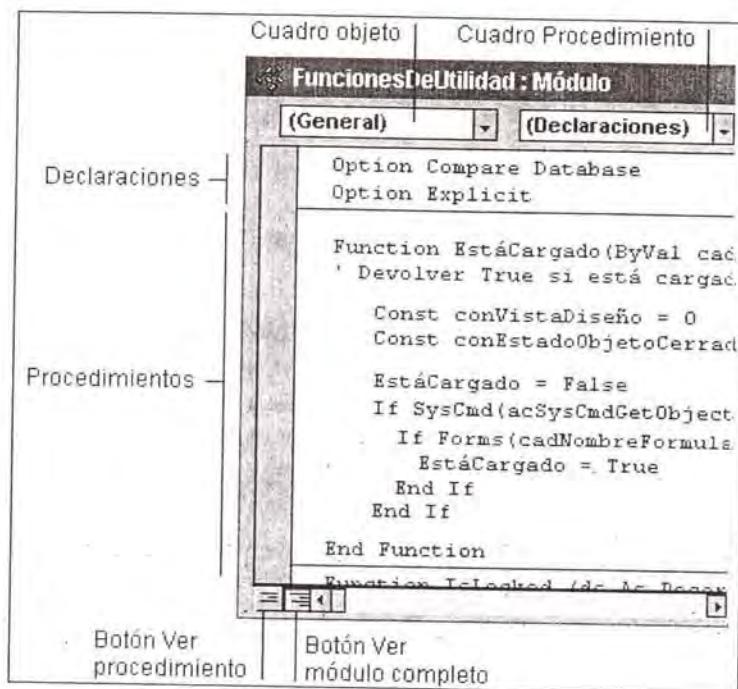


Figura 2-7

El proceso de diseño de las bases de datos Access

El proceso de diseño de una base de datos se guía por varios principios. El primero de ellos es que se debe evitar la información duplicada o, lo que es lo mismo, los datos redundantes, porque malgastan el espacio y aumentan la probabilidad de que se produzcan errores e incoherencias. El segundo principio es que es importante que la información sea correcta y completa. Si la base de datos contiene información incorrecta, los informes que recogen información de la base de datos contendrán también información incorrecta y, por tanto, las decisiones que tome a partir de esos informes estarán mal fundamentadas.

Un buen diseño de base de datos es, por tanto, aquél que divide la información en tablas basadas en temas para reducir los datos redundantes, proporciona a Access la información necesaria para reunir la información de las tablas cuando así se precise, ayuda a garantizar la exactitud e integridad de la información y satisface las necesidades de procesamiento de los datos y de generación de informes. Los sucesivos pasos de un buen proceso de diseño podrían ser los siguientes:

Determinar la finalidad de la base de datos: Es conveniente plasmar en papel el propósito de la base de datos, cómo piensa utilizarse y quién va a utilizarla. La idea es desarrollar una declaración de intenciones bien definida que sirva de referencia durante todo el proceso de diseño. Esta declaración de intenciones permitirá centrarse en los objetivos a la hora de tomar decisiones.

Buscar y organizar la información necesaria: Es necesario reunir todos los tipos de información que se desea registrar en la base de datos, como los nombres de productos o los números de pedidos.

Dividir la información en tablas: Conviene dividir los elementos de información en entidades o temas principales, como Productos o Pedidos. Cada tema pasará a ser una tabla.

Convertir los elementos de información en columnas: Habrá que decidir qué información se desea almacenar en cada tabla. Cada elemento se convertirá en un campo y se mostrará como una columna en la tabla. Por ejemplo, una tabla Empleados podría incluir campos como Apellido y Fecha de contratación.

Especificar claves principales o primarias: Elija la clave principal de cada tabla. La clave principal es una columna que se utiliza para identificar inequívocamente cada fila, como Id. de producto o Id. de pedido.

Definir relaciones entre las tablas: Examine cada tabla y decida cómo se relacionan los datos de una tabla con las demás tablas. Agregue campos a las tablas o cree nuevas tablas para clarificar las relaciones según sea necesario.

Ajustar el diseño: Es importante analizar el diseño para detectar errores. Conviene crear las tablas y agregar algunos registros con datos de ejemplo. También conviene comprobar si es posible obtener los resultados previstos de las tablas. Realizar los ajustes necesarios en el diseño.

Normalizar la base de datos: Es imprescindible aplicar reglas de normalización de los datos (formas normales) para comprobar si las tablas están estructuradas correctamente y realizar los ajustes necesarios en las tablas.

Claves principales en Access

Cada tabla debe incluir una columna o conjunto de columnas que identifiquen inequívocamente cada fila almacenada en la tabla. Ésta suele ser un número de identificación exclusivo, como un número de identificador de empleado o un número de serie. En la terminología de bases de datos, esta información recibe el nombre de clave principal de la tabla. Access utiliza los campos de clave principal para asociar rápidamente datos de varias tablas y reunir automáticamente esos datos. Si ya tiene un identificador exclusivo para una tabla, como un número de producto que identifica inequívocamente cada producto del catálogo, puede utilizar ese identificador como clave principal de la tabla, pero sólo si los valores de esa columna son siempre diferentes para cada registro. No puede tener valores duplicados en una clave principal. Por ejemplo, no utilice los nombres de las personas como clave principal, ya que los nombres no son exclusivos. Es muy fácil que dos personas tengan el mismo nombre en la misma tabla.

Una clave principal siempre debe tener un valor. Si el valor de una columna puede quedar sin asignar o vacío (porque no se conoce) en algún momento, no puede utilizarlo como componente de una clave principal. Se debe elegir siempre una clave principal cuyo valor no cambie. En una base de datos con varias tablas, la clave principal de una tabla se puede utilizar como referencia en las demás tablas. Si la clave principal cambia, el cambio debe aplicarse también a todos los lugares donde se haga referencia a la clave. Usar una clave principal que no cambie reduce la posibilidad de que se pierda su sincronización con las otras tablas en las que se hace referencia a ella. A menudo, se utiliza como clave principal un número único arbitrario. Por ejemplo, puede asignar a cada pedido un número de pedido distinto. La única finalidad de este número de pedido es identificar el pedido. Una vez asignado, nunca cambia.

Si piensa que no hay ninguna columna o conjunto de columnas que pueda constituir una buena clave principal, considere la posibilidad de utilizar una columna que tenga el tipo de datos Autonumérico. Cuando se utiliza el tipo de datos Autonumérico, Access asigna automáticamente un valor. Los identificadores de este tipo son perfectos para usarlos como claves principales, ya que no cambian. Una clave principal que contiene datos sobre una fila, como un número de teléfono o el nombre de un cliente, es más probable que cambie.

En algunos casos, es conveniente utilizar dos o más campos juntos como clave principal de una tabla. Por ejemplo, una tabla Detalles de pedidos que contenga artículos de línea de pedidos tendría dos columnas en su clave principal: Id. de pedido e Id. de producto. Cuando una clave principal está formada por más de una columna se denomina *clave compuesta*.

Para la base de datos de ventas de productos, puede crear una columna autonumérica para cada una de las tablas que funcione como clave principal: IdProducto para la tabla Productos, IdPedido para la tabla Pedidos, IdCliente para la tabla Clientes e IdProveedores para la tabla Proveedores.

Relaciones en Access

Access es un sistema de administración de bases de datos relacional. En una base de datos relacional, la información se divide en tablas distintas en función del tema. A continuación, se utilizan relaciones entre las tablas para reunir la información según se precise, ya que en la práctica, lo lógico es que los formularios contengan información de varias tablas. Por ejemplo, el formulario de la Figura 2-8 contiene información de las tablas Clientes (1), Empleados (2), Pedidos (3), Productos (4) y Detalles de pedidos (5) de la base de datos de Pedidos de productos. Las relaciones entre las tablas permiten manejar simultáneamente la información de varias de ellas.

Id. de producto	Nombre de producto	Precio unit.	Cantidad	Descuento	Precio total
1	Spices	12,00 \$	2	25%	18,0
2	Chartreuse verte	18,00 \$	21	25%	283,5
3	Rossle Sauerkraut	45,60 \$	15	25%	513,0
				0%	
					Subtotal: 814,50 \$
					Carga: 29,46 \$
					Total: 843,96 \$

Figura 2-8

Relación uno a varios

Supongamos que al considerar las tablas Proveedores y Productos de la base de datos de Pedidos de productos, un proveedor puede suministrar cualquier número de productos y, por consiguiente, para cada proveedor representado en la tabla Proveedores, puede haber muchos productos representados en la tabla Productos. Sin embargo, no ocurre que para cada producto haya muchos proveedores (para cada producto habrá un sólo proveedor). La relación entre la tabla Proveedores y la tabla Productos es, en este caso, una relación de uno a varios. Para representar una relación de uno a varios en el diseño de la base de datos, se toma la clave principal del lado "uno" de la relación y se agrega como columna o columnas adicionales a la tabla en el lado "varios" de la relación. En este caso, por ejemplo, se agregaría la columna Id. de proveedor de la tabla Proveedores a la tabla Productos. Access utilizaría entonces el número de identificador de proveedor de la tabla Productos para localizar el proveedor correcto de cada producto (Figura 2-9).

Id. de proveedor	Nombre de la compañía
1	Tropic Liquids

Id. de producto	Nombre de producto	Proveedor
1	Chai	1
2	Chang	1
3	Sirope de anís	1

Figura 2-9

La columna Id. de proveedor de la tabla Productos se denomina *clave externa*. Una clave externa es la clave principal de otra tabla. La columna Id. de proveedor de la tabla Productos es una clave externa porque también es la clave principal en la tabla Proveedores.

Relación varios a varios

Considere la relación entre la tabla Productos y la tabla Pedidos de modo que un solo pedido puede incluir varios productos. Por otro lado, supongamos que un único producto puede aparecer en muchos pedidos. Por tanto, para cada registro de la tabla Pedidos puede haber varios registros en la tabla Productos. Y para cada registro de la tabla Productos puede haber varios registros en la tabla Pedidos. Este tipo de relación se denomina relación de varios a varios porque para un producto puede haber varios pedidos, y para un pedido puede haber varios productos. Tenga en cuenta que para detectar las relaciones de varios a varios entre las tablas, es importante que considere ambas partes de la relación.

Los temas de las dos tablas (pedidos y productos) tienen una relación de varios a varios. Esto presenta un problema. Para comprender el problema, imagine qué sucedería si intenta crear la relación entre las dos tablas agregando el campo Id. de producto a la tabla Pedidos. Para que haya más de un producto por pedido, necesita más de un registro en la tabla Pedidos para cada pedido y, en ese caso, tendría que repetir la información de pedido para cada fila relacionada con un único pedido, lo que daría lugar a un diseño ineficaz que podría producir datos inexactos. El mismo problema aparece si coloca el campo Id. de pedido en la tabla Productos: tendría varios registros en la tabla Productos para cada producto. ¿Cómo se soluciona este problema? La solución a este problema consiste en crear una tercera tabla que descomponga la relación de varios a varios en dos relaciones de uno a varios. Insertaría la clave principal de cada una de las dos tablas en la tercera tabla y, por consiguiente, la tercera tabla registraría todas las apariciones o instancias de la relación (Figura 2-10).

PEDIDOS		PRODUCTOS	
IdPedido	IdPersonaliza	IdProducto	NombreProducto
10248	WILMK	11	Queso Cabrales
10311	DUMON	42	Singaporean Hokkien Fried Mee
		69	Gudbrandsdalsost
		72	Mozzarella di Giovanni

DETALLES PEDIDO				
IdPedido	IdProdu...	PrecioUn...	Cantid...	
10248	11	21.00	12	
10248	42	14.00	10	
10248	72	34.80	5	
10311	42	14.00	6	
10311	69	28.80	7	

Figura 2-10

Relación uno a uno

Supongamos que necesitamos registrar información complementaria sobre productos que apenas se va a necesitar. Como no se necesita la información con frecuencia, y como almacenar la información en la tabla Productos crearía un espacio vacío para todos los productos que no necesitan esa información, se coloca en una tabla distinta. Al igual que en la tabla Productos, se utiliza el identificador de producto como clave principal. La relación entre esta tabla complementaria y la tabla Productos es una relación de uno a uno. Para cada registro de la tabla Productos hay un único registro coincidente en la tabla complementaria. Cuando se identifique esta relación, ambas tablas deben compartir un campo común.

Cuando necesite crear una relación de uno a uno en la base de datos, considere si puede incluir la información de las dos tablas en una tabla. Si no desea hacer eso por algún motivo (quizás porque se crearía una gran cantidad de espacio vacío), puede representar esa relación en su diseño guiándose por las pautas siguientes:

- Si las dos tablas tienen el mismo tema, probablemente podrá definir la relación utilizando la misma clave principal en ambas tablas.
- Si las dos tablas tienen temas diferentes con claves principales distintas, elija una de las tablas (cualquiera de ellas) e inserte su clave principal en la otra tabla como clave externa.

Determinar las relaciones entre las tablas le ayudará a asegurarse de que tiene las tablas y columnas correctas. Cuando existe una relación de uno a uno o de uno a varios, las tablas implicadas deben compartir una o varias columnas comunes. Cuando la relación es de varios a varios, se necesita una tercera tabla para representar la relación.

Normalización

En un buen diseño es necesario aplicar las reglas de normalización de datos. Estas reglas sirven para comprobar si las tablas están estructuradas correctamente. El proceso de aplicar las reglas al diseño de la base de datos se denomina normalizar la base de datos o, simplemente, normalización.

La normalización es más útil una vez representados todos los elementos de información y después de haber definido un diseño preliminar. La idea es asegurarse de que se han dividido los elementos de información en las tablas adecuadas. Lo que la normalización no puede hacer es garantizar que se dispone de los elementos de datos correctos para empezar a trabajar.

Las reglas se aplican consecutivamente en cada paso para garantizar que el diseño adopta lo que se conoce como *forma normal*. Hay cinco formas normales ampliamente aceptadas: de la primera forma normal a la quinta forma normal, aunque las tres primeras son las que suelen necesarias para la mayoría de los diseños de base de datos.

Primera forma normal

La primera forma normal establece que en cada intersección de fila y columna de la tabla existe un valor y nunca una lista de valores. Por ejemplo, no puede haber un campo denominado Precio en el que se incluya más de un precio. Si se considera cada intersección de filas y columnas como una celda, cada celda sólo puede contener un valor.

Segunda forma normal

La segunda forma normal exige que cada columna que no sea clave, dependa por completo de toda la clave principal y no sólo de parte de la clave. Esta regla se aplica cuando existe una clave principal formada por varias columnas. Suponga, por ejemplo, que existe una tabla con las siguientes columnas, de las cuales Id. de pedido e Id. de producto forman la clave principal: Id. de pedido (clave principal), Id. de producto (clave principal) y Nombre de producto.

Este diseño infringe los requisitos de la segunda forma normal, porque Nombre de producto depende de Id. de producto, pero no de Id. de pedido, por lo que no depende de toda la clave principal. Se debe quitar Nombre de producto de la tabla, ya que pertenece a una tabla diferente (a la tabla Productos).

Tercera forma normal

La tercera forma normal exige no sólo que cada columna que no sea clave dependa de toda la clave principal, sino también que las columnas que no sean clave sean independientes unas de otras. O dicho de otra forma: cada columna que no sea clave debe depender de la clave principal y nada más que de la clave principal. Por ejemplo, considere una tabla con las siguientes columnas: IdProducto (clave principal), Nombre, PVP y Descuento. Suponga que la columna Descuento depende del precio de venta al público (PVP) sugerido. Esta tabla infringe los requisitos de la tercera forma normal porque una columna que no es clave, la columna Descuento, depende de otra columna que no es clave, la columna PVP. La independencia de las columnas implica que debe poder cambiar cualquier columna que no sea clave sin que ninguna otra columna resulte afectada. Si cambia un valor en el campo PVP, la columna Descuento cambiaría en consecuencia e infringiría esa regla. En este caso, la columna Descuento debe moverse a otra tabla cuya clave sea PVP.

Ejercicio 2-1. Se trata de diseñar una base de datos en un contexto educativo, cuyas tablas se refieren a los cursos que van a ser impartidos en distintas clases de un centro educativo, por varios profesores pertenecientes a diversos departamentos y en los que se matricularán distintos estudiantes. En el centro educativo también existe personal no docente. Consideraremos como tablas de la base de datos CURSO, DEPARTAMENTO, CLASE, ESTUDIANTE, MATRÍCULA, CLAUSTRO y PERSONAL con las siguientes características:

Tabla CURSO: define las características de los cursos que van a ser impartidos y sus campos van a ser el número del curso (CNO), el nombre del curso (CNOMBRE), la descripción del curso (CDESCP), los créditos que vale el curso (CRED), el valor de la tarifa del laboratorio (CTARIFA) y el nombre identificativo del departamento que da el curso (CDEPT). Puede tomarse CNO como la clave primaria, dado que CNO toma valores distintos para todas las filas y no tiene valores faltantes.

Tabla DEPARTAMENTO: tiene un registro por cada departamento académico al que pertenecen los profesores que imparten cursos. Sus campos van a ser el nombre identificativo de cada departamento (DEPT), el edificio al que pertenecen (DEDIF), el número de despacho en que están ubicados (DDESPACHO) y el número de identificación en el cuerpo docente del director del departamento (DCHFNO). El campo DEPT tomará valores distintos para todas las filas y no tiene valores faltantes. Además, según la política del centro educativo, cada curso debe estar organizado por un departamento académico, lo que nos indica que debemos tomar el campo DEPT como clave primaria de la tabla DEPARTAMENTO. Para aumentar la integridad de la base de datos, especificamos el campo CDEPT de la tabla CURSO como clave secundaria que referencia la *tabla padre* DEPARTAMENTO (se restringen los valores CDEPT de la tabla CURSO a aquéllos que se encuentran en la columna DEPT de la tabla DEPARTAMENTO). La tabla CURSO es una *tabla dependiente o tabla hijo* de la tabla padre DEPARTAMENTO.

Tabla CLASE: tiene un registro para cada clase que ofrezca un curso. Los campos que contiene son el número de curso (CNO), el número de sección (SEC), el número de identificación en el grupo de docentes del profesor (CINSTRFNO), el día en que se imparte (CDIA), la hora (CHORA), el edificio (CEDIF) y la sala en la que se encuentra la clase (CDESPACHO). En esta tabla consideramos el par de campos (CNO, SEC) como clave primaria. Además, sabemos que debido a la política del centro educativo, a cada clase le corresponde algún curso, por lo que cada valor del campo CNO de la tabla CLASE debe coincidir con algún valor CNO existente en la tabla CURSO. De esta forma, CNO en la tabla CLASE es clave secundaria que referencia CURSO, con lo que CURSO es *tabla padre* de CLASE, además de *depedir* (*ser tabla hijo*) de DEPARTAMENTO. Puesto que CLASE es dependiente de CURSO, que por su parte es dependiente de DEPARTAMENTO, decimos que CLASE es *tabla descendiente* de DEPARTAMENTO.

Tabla ESTUDIANTE: tiene un registro para cada estudiante matriculado en el centro educativo y sus campos son el número de estudiante (SNO), su nombre (SNOMBRE), su dirección (SDOMI), su teléfono (STLFNO), su fecha de nacimiento (SFNCIM), su número de identificación de centro educativo (SIQ), su número identificativo de departamento (SADVFNO) y el nombre de su departamento (SESP). La clave primaria para la tabla ESTUDIANTE es el campo SNO (número de estudiante).

Tabla MATRICULA: relativa a los datos recogidos cuando un estudiante se matricula en un curso y cuyos campos son el número de curso (CNO), el número de sección (SEC), el número de estudiante (SNO) y fecha (FECHA_MAT) y hora de matriculación del estudiante (HORA_MAT). La clave primaria de la tabla MATRÍCULA es una composición de tres columnas (CNO, SEC, SNO). Para asegurar la integridad referencial, especificaríamos dos claves secundarias en esta tabla. La clave secundaria compuesta (CNO, SEC) referenciaría la tabla ESTUDIANTE. De esta forma, MATRÍCULA es dependiente de dos tablas: CLASE y ESTUDIANTE. Supongamos, también, que todos los estudiantes deben especializarse en una asignatura correspondiente a algún departamento académico existente. Esto significa que el campo SESP (especialidad del estudiante) es clave secundaria que referencia la tabla DEPARTAMENTO. Por lo tanto, DEPARTAMENTO es padre de otras dos tablas: CURSO y ESTUDIANTE.

Tabla CLAUSTRO: tiene un registro para cada miembro del cuerpo docente del centro educativo y cuyos campos son el número de identificación del profesor (FNO), su nombre FNOMBRE, su dirección (FDOMI), su fecha de contratación (FFCANTI), el número de ayudantes o subordinados que dependen de él (FNUMDEP), su sueldo (FSUELDO) y el nombre identificativo del departamento al que se encuentre adscrito (FDEP). El campo FNO es la clave primaria de la tabla CLAUSTRO. Las claves secundarias pueden ser definidas de forma que se establezca una relación cíclica. Supongamos que es norma del centro que a cada miembro del personal docente se le asigne algún departamento académico. Entonces, la columna FDEPT, en la tabla CLAUSTRO, se especificaría como clave secundaria, referenciando DEPARTAMENTO (DEPARTAMENTO es la tabla padre y CLAUSTRO la tabla dependiente). Supongamos, también, que el director de cada departamento es miembro del personal docente. Entonces, el valor DCHFNO, en la tabla DEPARTAMENTO, se especifica como clave secundaria que referencia CLAUSTRO. (Aquí, CLAUSTRO es la tabla padre y DEPARTAMENTO es la tabla dependiente).

Tabla PERSONAL: tiene un registro para cada empleado que no pertenezca al cuerpo docente del centro y sus campos son: el nombre del empleado (ENOMBRE), su título (CARGO), su sueldo (ESUELDO) y el nombre identificativo del departamento del que dependa (DEPT). Estos campos no tienen relación con ninguna otra tabla.

Una vez definidas todas las relaciones entre las tablas de la base de datos podemos realizar ya un diagrama completo que refleje su diseño final (Figura 2-11).

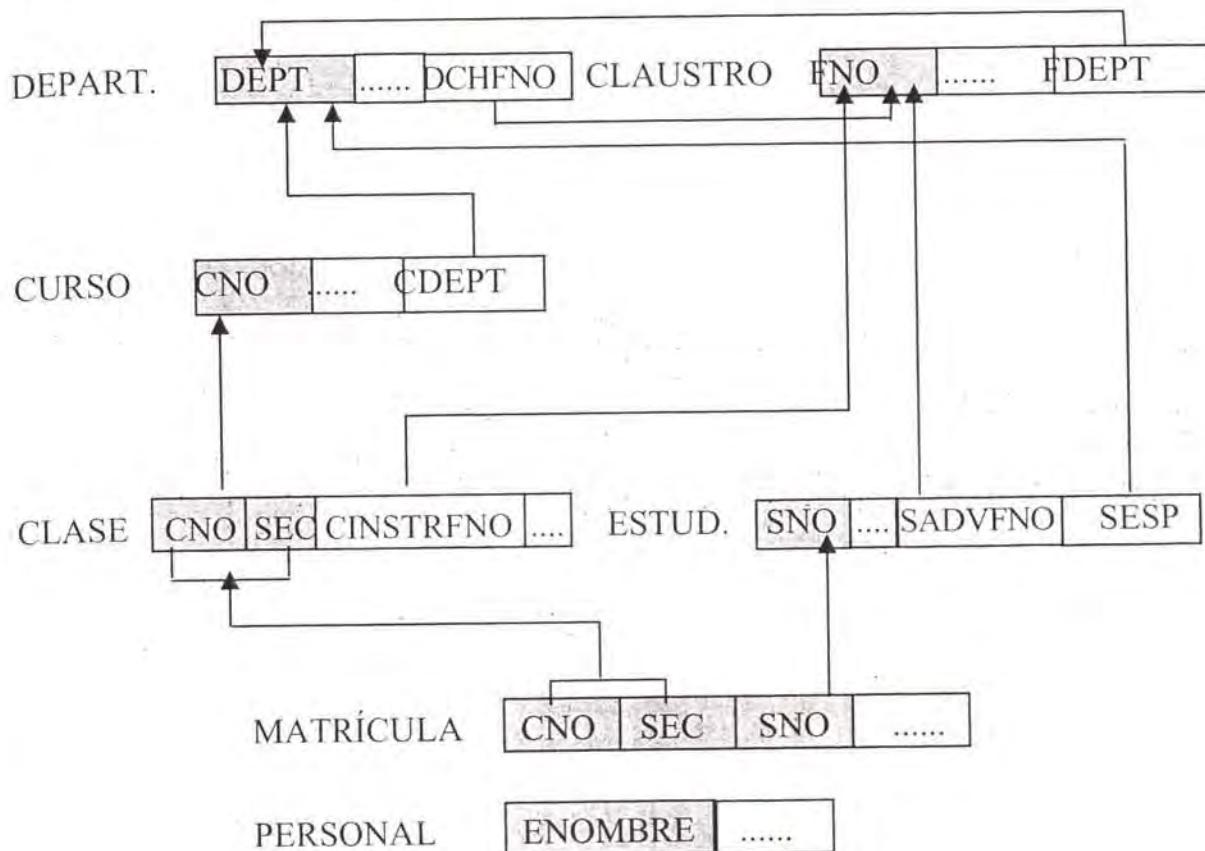


Figura 2-11

A continuación se presentan los datos de las tablas:

CURSO

CNO	CNOMBRE	CDESCP	CRED	CTARIFA	CDEPT
C11	INTROD. A LAS CC.	PARA NOVATOS	3	100.00	CIS
C22	ESTRUCT. DE DATOS	MUY ÚTIL	3	50.00	CIS
C33	MATEMÁTICAS DISCRETAS	ABSOLUTAMENTE NECESARIO	3	.00	CIS
C44	CIRCUITOS DIGITALES	AH HAB!	3	.00	CIS
C55	ARQUITECT.COMPUTADORES	MÁQ. VON NEUMANN	3	100.00	CIS
C66	BASES DE DATOS RELACIONALES	IMPRESINDIBLE	3	500.00	CIS
P11	EMPIRISMO	VERLO PARA CREERLO	3	100.00	PHIL
P22	RACIONALISMO	PARA USARLOS CIS	3	50.00	PHIL
P33	EXISTENCIALISMO	PARA USARLOS CIS	3	200.00	PHIL
P44	SOLIPSISMO	PARA MÍ MISMO	6	0.00	PHIL
T11	ESCOLASTICISMO	PARA BEATOS	3	150.00	THEO
T12	FUNDAMENTALISMO	PARA DESCUIDADOS	3	90.00	THEO
T33	HEDONISMO	PARA SANOS	3	.00	THEO
T44	COMUNISMO	PARA ÁVAROS	6	200.00	THEO

Capítulo 2**DEPARTAMENTO**

DEPT	DEDIF	DDESPACHO	DCHFNO
THEO	HU	200	10
CIS	SC	300	80
D.G.	SC	100	-
PHIL	HU	100	60

CLASE

CNO	SEC	CINSTRFNO	CDIA	CHORA	CEDIF	CDEPACHO
C11	01	08	LU	08:00-09:00 AM	SC	305
C11	02	08	MA	08:00-09:00 AM	SC	306
C33	01	80	MI	09:00-10:00 AM	SC	305
C55	01	85	JU	11:00-12:00 AM	HU	306
P11	01	06	JU	09:00-10:00 AM	HU	102
P33	01	06	VI	11:00-12:00 AM	HU	201
T11	01	10	LU	10:00-11:00 AM	HU	101
T11	02	65	LU	10:00-11:00 AM	HU	102
T33	01	65	MI	11:00-12:00 AM	HU	101

MATRICULA

CNO	SEC	SNO	FECHA_MAT	HORA_MAT
C11	01	325	1987-01-04	09.41.30
C11	01	800	1987-12-15	11.49.00
C11	02	100	1987-12-17	09.32.00
C11	02	150	1987-12-17	09.32.30
P33	01	100	1987-12-23	11.30.00
P33	01	800	1987-12-23	11.23.00
T11	01	100	1987-12-23	11.21.00
T11	01	150	1987-12-15	11.35.30
T11	01	800	1987-12-15	14.00.00

ESTUDIANTE

SNO	SNOMBRE	SDOMI	STLFNO	SFNACIM	SIQ	SADVFN0	SESP
325	CURLEY DUBAY	CONNECTICUT	203-123-4567	780517	122	10	THEO
150	LARRY DUBAY	CONNECTICUT	203-123-4567	780517	121	80	CIS
100	MOE DUBAY	CONNECTICUT	203-123-4567	780517	120	10	THEO
800	ROCKY BALBOA	PENNSYLVANIA	112-112-1122	461004	99	60	PHIL

CLAUSTRO

FNO	FNOMBRE	FDOMI	FFECANTI	FNUMDP	FSUELDO	FDEPT
06	KATHY PEPE	CALLE DE LA PIEDRA, 7	1979-01-15	2	35.000,00	PHIL
10	JESSIE MARTIN	DR. DEL ESTE, 4	1969-09-01	1	45.000,00	THEO
08	JOSE COHN	APTDO. CORREOS 1138	1979-07-09	2	35.000,00	CIS
85	AL HARTLEY	CALLE DE LA PLATA	1979-09-05	7	45.000,00	CS
60	JULIA MARTIN	DR. ESTE, 4	1969-09-01	1	45.000,00	PHIL
65	LISA BOBAK	CAMINO DE LA RISA, 77	1981-09-06	-	36.000,00	THEO
80	BARB HLAVATY	CALLE DEL SUR, 489	1982-01-16	3	35.000,00	CIS

PERSONAL

ENOMBRE	CARGO	ESUELDO	DEPT
LUCAS	EVANG1	53	THEO
MARCOS	EVANG2	52	THEO
MATEO	EVANG3	51	THEO
DICK NIX	LADRÓN	25001	PHIL
HANK KISS	BUFÓN	25000	PHIL
JUAN	EVANG4	54	THEO
EUCLIDES	AYTE. LAB.	1000	MATH
ARQUIMEDES	AYTE. LAB.	200	ENG
DA VINCI	AYTE. LAB.	500	-

Adicionalmente tendremos en cuenta para la integridad referencial lo siguiente:

Inserción de la clave primaria: el sistema verificará que el valor de la clave primaria es único. Si no lo es, la inserción será rechazada.

Inserción de la clave secundaria: el sistema verificará que cualquier clave secundaria tenga el mismo valor que alguna clave primaria existente en la tabla referenciada. Si esto no se cumple, la inserción será rechazada.

Actualización de la clave primaria: el sistema rechazará la actualización si existe un registro dependiente que refiere la clave primaria.

Actualización de la clave secundaria: el sistema verificará que el nuevo valor de la clave secundaria coincida con alguna clave primaria existente en la tabla referenciada. Si esto no ocurre, la actualización se rechazará.

Borrado de la clave primaria: se puede dirigir la respuesta del sistema a la eliminación de un registro que tenga una clave primaria igual a alguna clave secundaria correspondiente. Las opciones disponibles son: rechazar cualquier operación de borrado (RESTRICT), sustituir automáticamente los valores nulos por los valores correspondientes de la clave secundaria (SET NULL) y borrar automáticamente cualquier registro dependiente que tenga una clave secundaria igual a la primaria (CASCADE).

Borrado de una clave secundaria: no existen restricciones (a menos que la clave secundaria sea parte de la primaria en la que se apliquen otras restricciones).

El diseñador de la base de datos debe prestar una cuidadosa atención a estas reglas de procesamiento. Existen restricciones e implicaciones adicionales para las tablas descendentes, especialmente aquellas afectadas por relaciones cíclicas. El diseño completo de las relaciones existentes entre las tablas (Figura 2-11) y las restricciones de integridad son el corazón de una base de datos.

Ejercicio 2-2. Supongamos que hay que realizar una base de datos relativa a un aula con las siguientes características: Número del alumno, Dirección del alumno, Calificaciones, Número de aula, Nombre del Tutor, Dirección del Tutor y Calificación final. Definir un conjunto razonable de tablas para la base de datos que constituyan un diseño totalmente normalizado.

Dado que el campo Calificaciones podría tomar varios valores, uno por cada una de las asignaturas, podríamos obtener la *primera forma normal* mediante las tablas siguientes:

<i>Número del alumno</i>	<i>Número del alumno</i>
<i>Nombre del alumno</i>	<i>Asignatura</i>
<i>Dirección del alumno</i>	<i>Número de evaluación</i>
<i>Número de aula</i>	<i>Calificación</i>
<i>Nombre del Tutor</i>	<i>Calificación global</i>
<i>Dirección del Tutor</i>	
<i>Calificación final</i>	

Hemos situado los datos en tablas separadas, de manera que los datos de cada tabla sean de un tipo similar, y dando a cada tabla una *clave primaria* (en negrita).

Añadiendo un poco más vemos que el campo Calificación global depende del Número del alumno y del Número de evaluación, pero no de Asignatura, con lo que podríamos obtener la *segunda forma normal* con la siguiente estructura de tablas:

<i>Número del alumno</i>	<i>Número del alumno</i>	<i>Número del alumno</i>
<i>Nombre del alumno</i>	<i>Asignatura</i>	<i>Número de evaluación</i>
<i>Dirección del alumno</i>	<i>Número de evaluación</i>	<i>Calificación global</i>
<i>Número de aula</i>	<i>Calificación</i>	
<i>Nombre del Tutor</i>		
<i>Dirección del Tutor</i>		
<i>Calificación final</i>		

Para conseguir la *tercera forma normal* sólo nos falta que todos los campos dependan sólo de la clave. La siguiente estructura de tablas podría ser la solución:

<i>Número del alumno</i>	<i>Número de aula</i>	<i>Número del alumno</i>	<i>Número del alumno</i>
<i>Nombre del alumno</i>	<i>Nombre del Tutor</i>	<i>Asignatura</i>	<i>Número de evaluación</i>
<i>Dirección del alumno</i>	<i>Dirección del Tutor</i>	<i>Número de evaluación</i>	<i>Calificación global</i>
<i>Número de aula</i>		<i>Calificación</i>	
<i>Calificación final</i>			