

TEMA 53

DISEÑO LÓGICO DE DATOS. TRANSFORMACIÓN DEL MODELO CONCEPTUAL A MODELOS LÓGICOS. ANÁLISIS RELACIONAL DE DATOS. DOCUMENTACIÓN.

ÍNDICE

1. INTRODUCCIÓN
2. CONVERSIÓN A ESTRUCTURAS LÓGICAS DE REGISTRO
3. TERMINACIÓN DE LA ESPECIFICACIÓN DE LA BASE DE DATOS
4. CONVERSIÓN EN CONJUNTO DE FICHEROS
5. CONVERSIÓN EN ESTRUCTURA SGBD
6. BIBLIOGRAFÍA

1. INTRODUCCIÓN

Este capítulo describe el diseño de bases de datos. El diseño de bases de datos convierte el modelo de datos desarrollado en el diseño lógico en una definición de bases de datos que soportada por un software de base de datos. Este software utilizado para dar soporte a bases de datos se conoce como **sistema de gestión de base de datos (SGBD)**.

El diseño de bases de datos pasa por una serie de pasos:

- Conversión a tipos de registros.
- Diseño lógico.
- Diseño físico.

El primer paso es independiente del tipo de SGBD utilizado. Este paso convierte el modelo conceptual E-R o relacional en un conjunto de tipos de registro donde cada uno de ellos tiene varios campos. El conjunto de tipos de registro se conoce como **estructura lógica de registro (LRS)**. Esta conversión es necesaria porque el SGBD más comercial almacena los datos como un conjunto de registros. Los registros que tienen los mismos campos pertenecen al mismo tipo de registro. El primer paso también define cómo se utilizan los datos de esos registros. Estos requerimientos de acceso se utilizarán más tarde para elegir estructuras de SGBD.

El siguiente paso de diseño de base de datos convierte la LRS en una **definición de base de datos**. Este paso utiliza técnicas que dependen del SGBD. Estas técnicas dependientes del SGBD son necesarias porque SGBD diferentes soportan diferentes enlaces entre sus registros. Tales enlaces se usan para recuperar registros siguiendo esos enlaces de un registro a otro. El diseño de bases de datos depende de la estructura soportada por el SGBD y usa técnicas apropiadas para ellas.

El diseño dependiente del SGBD procede en dos partes. El primer paso es el **diseño lógico**. Define los tipos de registro del SGBD y los enlaces entre ellos. El siguiente paso es el diseño físico, que elige una organización física que soporte los métodos de acceso a la base de datos.

Este capítulo describe primero las estructuras lógicas de registro y cómo deducirlas. En segundo lugar describe el software del SGBD y los diseños lógico y físico.

2. CONVERSION A ESTRUCTURAS LOGICAS DE REGISTRO

El primer paso del diseño de bases de datos es *convertir el modelo conceptual en tipos de registros y especificar cómo se accede a esos registros*. Estos requerimientos de acceso se usarán más tarde para elegir las claves de fichero que faciliten el acceso a los datos. En este paso también se añaden datos cuantitativos como tamaño de elemento, número de registros y frecuencia de acceso. Este tipo de datos es necesario para calcular las necesidades de almacenamiento y volumen de transacciones que va a soportar el sistema informático.

La combinación de estructura lógica de registro, especificaciones de acceso y datos cuantitativos se conoce a veces como **especificación de la base de datos**. Se utiliza para elegir una estructura de registro que soporte el SGBD. Ahora se pasa a describir cómo se desarrolla la especificación de la base de datos, comenzando con la estructura lógica de registro.

¿QUÉ SON LAS ESTRUCTURAS LÓGICAS DE REGISTRO?

En la figura 53.1 se muestra una estructura lógica de registro. Esa estructura está compuesta por una serie de **tipos** de registro. Cada tipo se representa con una caja rectangular y tiene un nombre único. Los tres tipos de registro de la figura se llaman *proyectos*, *usa* y *elementos*. Cada tipo se compone de **campos**. El tipo de registro *proyectos* tiene tres campos: *num-proyecto*, *día-comienzo* y *presupuesto*. Para distinguir la estructura lógica del diagrama E-R, el nombre del tipo de registro aparece fuera de la caja; dentro de ella se sitúan los campos.

La estructura lógica de registro también contiene los **enlaces** entre tipos de registro. Estos enlaces se muestran con una flecha entre tipos e indican el sentido. Cada enlace se etiqueta con los campos que aparecen en ambos tipos de registro. Así, el enlace entre *proyectos* y *usa* se etiqueta con *num-proyecto*, porque este campo aparece en ambos tipos *proyectos* y *usa*.

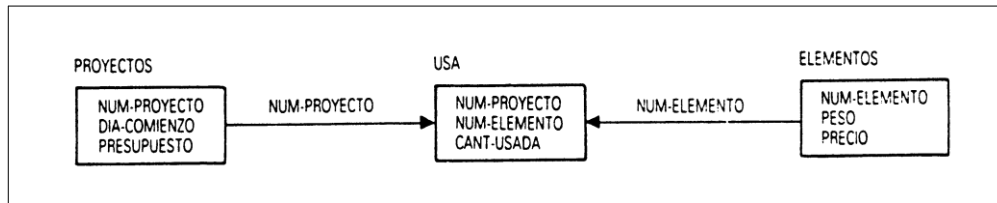


Figura 53.1. Estructura lógica de registro

Los enlaces tienen, además, otra propiedad. Si se mira la dirección del enlace y su etiqueta, se encuentra que en el enlace parte de un tipo de registros que contiene un único registro para un valor dado del campo etiqueta de enlace. Así, en la figura, el enlace etiquetado *num-proyecto* se origina en el tipo de registro *proyectos*, en el que aparecerá un único registro *proyectos* para un valor dado de *num-proyecto*. El enlace termina en un tipo de registro que puede tener muchos registros *usa* para ese valor de la etiqueta. De esta forma, en la figura puede haber muchos registros con un valor dado de *num-proyecto*, porque un proyecto puede usar muchos elementos.

Hay una razón semántica para tales enlaces. Los enlaces definen registros poseedores de otros registros. En la figura, los registros *proyectos* poseen registros *usa*. El campo común *num-proyecto* identifica los registros particulares poseídos. Así, un registro *proyectos* con un valor dado de *num-proyecto* poseerá todos los registros *usa* con ese mismo valor de *num-proyecto*. Esta es la razón del enlace. Supóngase que se desea encontrar todos los elementos usados en un proyecto. Primero se encontraría el registro de *proyectos* con el *num-proyecto* requerido. Entonces se usarían los enlaces para encontrar todos los registros *usa* poseídos por ese registro *proyectos*. Estos registros *usa* contienen el *num-elemento* de los elementos utilizados por un proyecto dado.

MÉTODOS PARA PRODUCIR UNA ESTRUCTURA LÓGICA DE REGISTRO

Ahora se describe qué es una *estructura lógica de registro (LRS)*; se empezará diciendo cómo derivarla del modelo conceptual. Se pueden utilizar dos métodos. Uno comienza con un modelo relacional de datos. El modelo relacional se puede entonces convertir en estructura lógica de registro. El otro método empieza con un modelo E-R y lo convierte directamente en estructura lógica de registro.

Conversión del modelo relacional en estructura lógica de registro

El modelo relacional de los datos del sistema se convierte en estructura lógica de registro en dos pasos. En el primer paso se crean los tipos de registro y en el segundo se añaden los enlaces entre ellos. En el primer paso *cada relación se convierte en un tipo de registro y cada atributo de relación en un campo de registro*.

Como ejemplo, considérense las relaciones:

Elementos (*num-elemento*, *peso*, *precio*)

Usa (*num-proyecto*, *num-elemento*, *cant-usada*)

Proyectos (*num-proyecto*, *día-comienzo*, *presupuesto*)

En el primer paso, cada una de estas relaciones forma un tipo lógico de registro.

Los enlaces entre ellos se definen en el paso siguiente. Para hacerlo, se necesita utilizar las claves ajenas. Un conjunto de atributos, **A**, es una **clave ajena** en una relación, **R**, si:

1. **A** no es una clave de relación de **R**; pero,
2. **A** es una clave de relación en alguna otra relación, **R2**.

Obsérvese que la relación *usa* contiene los atributos *num-elemento* y *num-proyecto*. *Num-elemento* es clave de relación en la relación *elementos*, pero no es clave de relación en *usa*. De igual forma, *num-proyecto* es una clave ajena de *usa*, porque es clave en la relación *proyectos* pero no lo es en la relación *usa*.

El siguiente paso es crear los enlaces entre los registros de la estructura lógica de registros. Hay un enlace desde la relación R1 a la R2 si la clave de relación de R1 es una clave ajena en R2. Así, como *num-elemento* es clave de relación en *elementos* y clave ajena en *usa*, hay un enlace de *elementos* a *usa*. De esta forma, las tres relaciones se convertirán en la estructura lógica de registro, mostrada en la figura anterior.

Conversión de un modelo entidad-relación en estructura lógica de registro

Algunas metodologías de diseño no producen un modelo relacional, se quedan en el modelo E-R. En estos casos es posible derivar la estructura lógica de registro directamente del diagrama E-R.

Lo más fácil es hacer de cada conjunto del diagrama E-R un tipo registro (record). Una conversión de este tipo se muestra en la figura 53.2. En él los conjuntos *proyectos*, *para* y *elementos* se convierten cada uno en un tipo de registro lógico. El conjunto de entidad *pedidos* y el conjunto de relación *hace* se combinan en un tipo de registro lógico. Se verá que tales combinaciones son siempre posibles en relaciones 1:N. En una relación 1:N, las entidades de un conjunto entidad aparecen en una sola relación. En la figura, cada pedido del conjunto *pedidos* aparece en una sola relación *hace*. *Hace* y *pedidos* tendrán la misma clave de relación, por tanto también se podrían combinar.

Entonces se añaden los enlaces. Los enlaces empiezan en los tipos de registro lógico que representan entidades y terminan en registros lógicos que representan relaciones. Hay un enlace para *elementos*, que representa un conjunto de entidad, para *para*, que representa un conjunto de relación. La misma regla se aplica para conjuntos que han sido combinados. Hay un enlace para *proyectos*, que representa un conjunto de entidad, para *pedidos*, que contiene la relación *hace*.

Cada conjunto entidad dependiente se convierte también en un tipo de registro lógico. Se añade un enlace desde el conjunto entidad a sus conjuntos de entidad dependientes. La etiqueta del enlace será el identificador del conjunto entidad.

Para convertir subconjuntos en estructuras lógicas de registro, se utilizan técnicas adicionales.

El siguiente paso es añadir los enlaces a la estructura lógica de registros. De nuevo se usa el método de la figura. Los enlaces terminan siempre en aquellos registros lógicos que representan conjuntos de relación o contienen conjuntos de relación. El tipo de registro lógico *líneas-factura* contiene los conjuntos de relación *sobre*, *en* y *para*. Estos tres enlaces terminan el *líneas-factura* para esos conjuntos de relación. Además, hay un cuarto enlace desde el tipo de registro lógico, *facturas*, para mostrar la dependencia de *líneas-factura* en *factura*.

La conversión de las relaciones es fácil, porque algunos de los conjuntos del modelo E-R ya se habían combinado.

Los enlaces se dibujan identificando primero la clave ajena y después dibujando los enlaces en la forma descrita anteriormente. Así, por ejemplo, *num-proyecto* es una clave ajena de la relación *requerimientos-del-proyecto*, pero una clave de la relación *proyectos*. Por ello se añade un enlace etiquetado por *num-proyecto* desde el tipo de registro lógico *proyectos* al tipo de registro lógico *requerimientos-proyecto*.

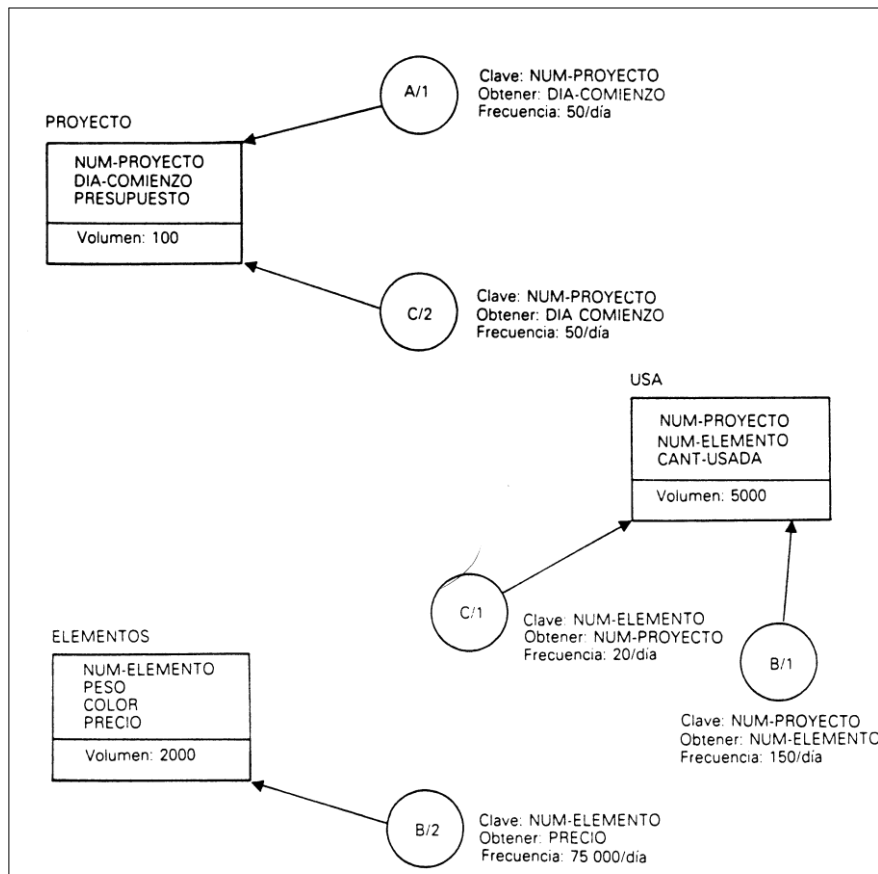


Figura 53.2. Rutas de acceso al nivel de estructura lógica

3. TERMINACIÓN DE LA ESPECIFICACIÓN DE LA BASE DE DATOS

La estructura lógica de registro es una parte de la especificación de base de datos. Sirve para definir la estructura de la base de datos. Para el diseño de la base de datos se necesita información adicional, en particular:

Datos cuantitativos que informan del volumen de información a almacenar.

Requerimientos de acceso, que informan de cómo se usa la base de datos. Estos requerimientos se utilizan para elegir las claves de ficheros durante el diseño de la base de datos.

La información sobre datos cuantitativos y requerimientos de acceso se obtienen durante el análisis de sistema. Los datos cuantitativos consisten en:

- El tamaño de los elementos de datos, y
- El número de ocurrencias de tipos de registro.

El tamaño de los elementos de datos se da normalmente en el diccionario de datos. Los volúmenes de registro se pueden añadir a la estructura lógica de registro según el método mostrado en la figura 53.2. El número de registro se añade simplemente en el registro lógico. Así, la figura muestra que hay 100 registros de *proyectos*, 2000 de *elementos* y 5000 de *usa*.

Especificación de los requerimientos de acceso

Los requerimientos de acceso se toman inicialmente de las especificaciones de procedimientos de usuario, que incluye sentencias sobre cómo acceden los usuarios a los datos. Estas sentencias pertenecen a los requerimientos de acceso. Durante el diseño de base de datos, se delinean las rutas de acceso a los tipos de registro lógico para cada requerimiento. Estas rutas muestran cómo se usa el dato y describen:

Los tipos de registro accedidos por cada requerimiento de acceso.
 La secuencia en la que se accede a esos tipos de registro.
 Las claves de acceso utilizadas para seleccionar los tipos de registro.
 Los elementos recuperados de cada registro, y
 El número de registros a los que se accede.

Hay muchas formas de dibujar rutas de acceso. En la figura se ilustra un método. Muestra tres requerimientos de acceso delineados para tres tipos de registro lógico. Los tres requerimientos son:

- A. encontrar el *día-comienzo* para un proyecto dado.
- B. encontrar el *precio* de los elementos usados en un proyecto con un *num-proyecto* dado.
- C. encontrar el *día-comienzo* de proyectos que usan un *num-elemento* dado.

Hay una ruta de acceso para cada requerimiento y cada ruta de acceso puede estar formada por uno o más pasos de acceso. Cada paso de acceso tiene una etiqueta que se compone de un mnemónico que describe el requerimiento y de un número secuencial. Las claves de acceso son los nombres de los campos cuyos valores se conocen en el momento del paso de acceso. El paso de acceso también incluye una descripción de la actividad del paso.

En la figura, el requerimiento A se especifica por un paso de acceso etiquetado A/1. Este paso se utiliza para acceder a los registros de *proyecto*. El acceso usa el valor de *num-proyecto* como clave de acceso y recupera el valor de *día-comienzo*. Se hace una media de 50 veces al día y recupera un registro.

El requerimiento de acceso B consta de dos pasos de acceso, uno para el tipo de registro *usa* y el otro para el tipo de registro *elementos*. El paso de acceso 1 (etiquetado B/1) encuentra los elementos usados por el proyecto. Se hace una media de 150 veces al día y en cada acceso recupera una media de 50 registros, porque cada proyecto utiliza una media de 50 elementos. El 50 se calcula dividiendo el número de registros *usa* por el número de proyectos. El paso de acceso 2 (etiquetado B/2) encuentra el *precio* de esos elementos. Obsérvese que este acceso tiene una frecuencia de 7500 por día. La frecuencia se calcula multiplicando las 150 veces que se ejecuta el paso de acceso B/1 por los 50 registros de elementos que se recuperan en cada ejecución del paso de acceso B/1.

El último requerimiento de acceso de la figura es el C. El primer paso (etiquetado C/1) encuentra los *num-proyecto* de proyectos que usan un *num-elemento* dado. El paso 1 del paso de acceso 3 se ejecuta 20 veces al día y recupera una media de 2,5 registros (ya que cada elemento se usa en 2,5 proyectos diferentes). El paso de acceso 2 (C/2) encuentra el *día-comienzo* de estos proyectos.

La definición de requerimientos de acceso completa la especificación de la base de datos, que se utiliza ahora para crear un diseño de base de datos. La estructura de registro lógico se convierte en un **diseño lógico de base de datos**. Las rutas de acceso se utilizan para seleccionar una estructura física apropiada. Normalmente, para satisfacer los requerimientos de acceso se puede elegir varias alternativas físicas. Parte del diseño es seleccionar a partir de estas estructuras. Para hacer esas elecciones se utiliza un proceso iterativo que estima los tratamientos de cada iteración para comparar las alternativas.

Las clases de técnicas de diseño que se usen dependerán del software que se utilice para la implementación de la base de datos. La conversión más simple es ir hacia un **conjunto de ficheros**. Alternativamente, la base de datos se puede implementar en un **SGBD**. Primero se describirá la conversión en un conjunto de ficheros y después en SGBD. También se hará un breve estudio de técnicas de diseño utilizadas para convertir una estructura lógica de registro en una estructura de base de datos.

4. CONVERSIÓN EN CONJUNTO DE FICHEROS

Un fichero está compuesto por un número de registros del mismo tipo. Cada uno de esos registros tiene una serie de campos. Como ejemplo, la figura 53.3 es un fichero llamado *personas*. Cada registro de este fichero tiene tres campos: *nombre*, *día-nacimiento* y *lugar-nacimiento*.

Los programas deben poder recuperar registros de ese fichero. Una forma de hacerlo es leer secuencialmente el fichero de principio a fin. Esto se llama **acceso secuencial**. Sin embargo,

generalmente es necesario recuperar un solo registro. Por ejemplo, se quiere encontrar el registro de José para ver en qué día ha nacido. Para ello se empezaría a leer por el primer registro hasta encontrar el registro de José. Esto puede consumir mucho tiempo si el fichero es grande, porque habría que leer muchos registros hasta encontrar el buscado. Se necesita la capacidad de acceder directamente a un registro del fichero, en vez de leer todo el fichero hasta encontrar el registro requerido.

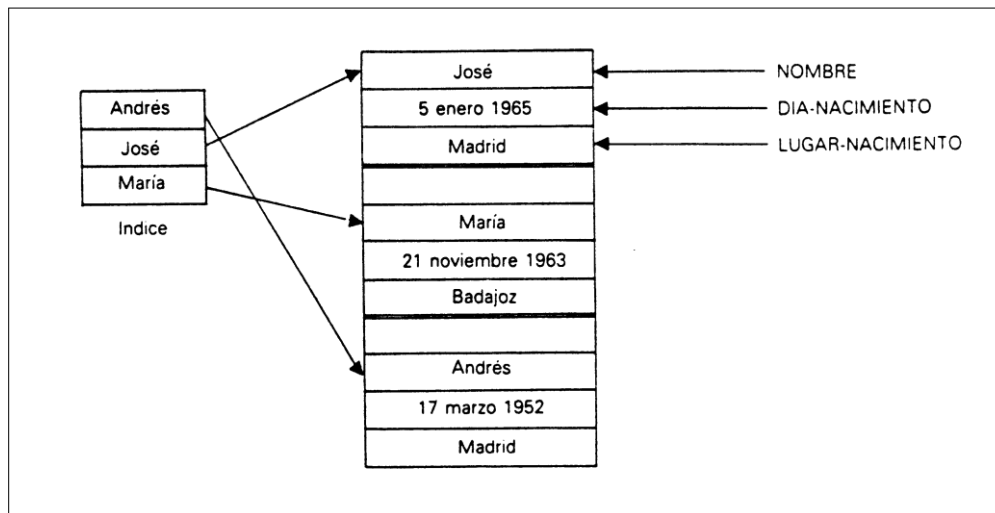


Figura 53.3. Fichero personas

El acceso directo utiliza **índices**. Un índice utiliza un campo del fichero como clave y hay una entrada en el índice por cada registro del fichero. Esta entrada contiene el valor de la clave de registro y un enlace del índice al fichero. Así, el fichero *personas* de la figura tiene un índice cuya clave es *nombre*. Tiene tres entradas con valores "José", "María" y "Andrés". Cada entrada tiene un enlace al registro con ese valor clave y los programas pueden acceder a registros a través del índice. Un programa especifica el valor de la clave y el software del fichero utilizará el índice para acceder directamente al registro del fichero que tenga esa clave.

Conversión de una Estructura Lógica de Registro en un Conjunto de Ficheros

El primer paso convierte cada registro lógico en un fichero. Así, la estructura lógica de registro de la figura 53.2 se convertiría en tres ficheros: *proyectos*, *elementos* y *usa*. Los campos del registro lógico serían campos del fichero. El fichero *proyectos* tendría tres campos: *num-proyecto*, *día-comienzo* y *presupuesto*.

Las claves en los pasos de acceso se utilizan para elegir claves para los índices de fichero. El fichero *proyectos* tiene dos pasos, A/1 y C/2. La clave en cada paso es *num-proyecto*. *Num-proyecto* es ahora la clave de un índice para el fichero *proyectos*. Todos los accesos al fichero pueden ahora ser directos.

De forma parecida se elige *num-elementos* como clave de un índice del fichero *elementos*. El fichero *usa* tiene dos pasos de acceso. Sin embargo, cada uno de ellos tiene una clave diferente. El paso de acceso C/1 tiene la clave *num-elemento* y el B/1 tiene como clave *num-proyecto*. Si todos los accesos son directos, entonces el fichero *usa* debe tener dos índices. Un índice utiliza la clave *num-elemento* y el otro la clave *num-proyecto*.

Así, la figura 53.4 ilustra la estructura del fichero que se obtiene a partir de la estructura lógica de registro de la figura. Está compuesta por tres ficheros: *proyectos*, *usa* y *elementos*. Los ficheros *proyectos* y *elementos* tienen un índice y el fichero *usa* tiene dos. Además, en la figura 53.2. se muestran los campos de los registros de cada fichero. Así, los registros del fichero *elementos* tienen cuatro campos: *num-elemento*, *peso*, *color* y *precio*. Obsérvese que los enlaces entre tipos de registros en la estructura lógica de registro no se llevan a la estructura de ficheros, porque el software de ficheros no soporta enlaces entre registros. Sin embargo, el gestor de base de datos puede soportarlos.

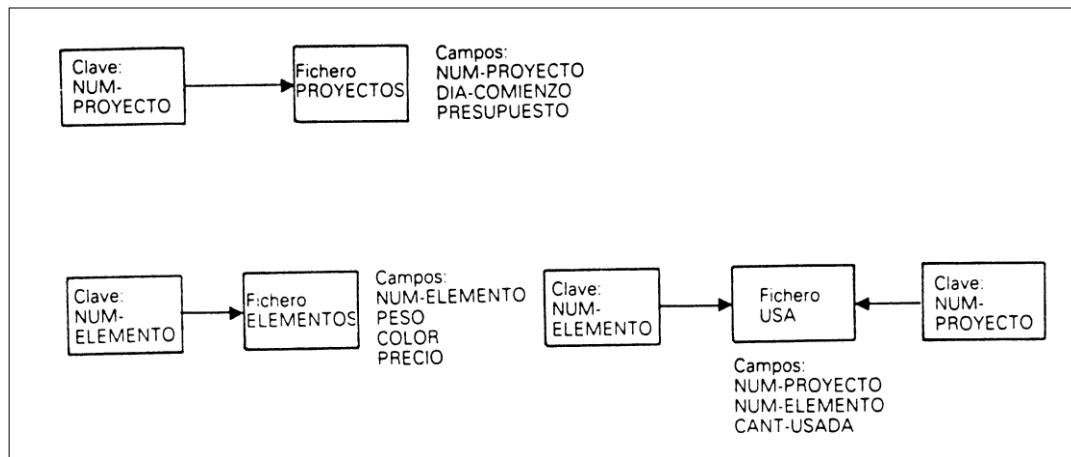


Figura 53.4. Estructura de fichero para la estructura lógica de la figura 53.2.

Sistemas de gestión de base de datos

Al igual que los ficheros, los sistemas de bases de datos almacenan los datos como tipos de registro. A diferencia de aquéllos, un sistema de gestión de bases de datos (**SGBD**) también almacena los enlaces entre registros. Todo SGBD soporta un tipo diferente de estructura de enlace entre tipos de registro. Así, virtualmente todo SGBD soporta una única estructura. Sin embargo, hay algunas características comunes en algunos SGBD. Estas características comunes surgen de la estandarización de los SGBD. Ahora hay varios SGBD que soportan el modelo relacional. Se conocen como **SGBD relacionales**. Hay también varios SGBD que soportan estructuras en red. Se propuso un estándar de estructuras en red, y siguiéndolo han surgido varios SGBD. Tales estructuras se conocen como **SGBD en red**. Finalmente, hay varios SGBD que soportan las estructuras jerárquicas conocidas como **SGBD jerárquicas**.

Se ilustrarán las estructuras soportadas por cada uno de estos tipos de SGBD. Esta ilustración se muestra en la figura 53.5, que presenta las estructuras de registro de SGBD para un sistema donde:

- Se asignan proyectos a departamentos.
- Los proyectos se componen de un número cualquiera de trabajos, y
- Cada proyecto utiliza diferentes tipos de elementos.

Sistemas de gestión de bases de datos relacionales

En un sistema de gestión de base de datos relacional, cada una de esas relaciones se define utilizando el lenguaje de definición del sistema. El sistema de gestión proporciona los comandos que se usan para almacenar o recuperar datos.

Sistema de gestión de bases de datos en red

El SGBD que soporta las estructuras en red almacena los datos como tipos de registro. Además, se puede establecer entre esos tipos de registro la relación padre-hijo. Tal relación se ilustra en la figura, donde cada registro de *departamentos* poseerá un número cualquiera de registros *proyectos*. Un registro de *departamentos* poseerá los registros de *proyectos* de los proyectos asignados a ese departamento. Cada registro *proyectos* también posee los registros *trabajos* de los trabajos que conforman el proyecto. Cada registro de *proyectos* posee, además, aquellos registros *usa* que representan el uso de elementos en un proyecto. Los enlaces padre-hijo se llaman tipos de registro en la terminología de redes.

En un modelo en red, cada tipo de registro puede ser padre de cualquier otro. Además, puede tener un número cualquiera de padres. Sin embargo, éste no es el caso para sistemas de gestión de bases de datos jerárquicas.

Sistemas de gestión de bases de datos jerárquicas

El modelo de datos jerárquico difiere del modelo en red en que cada tipo de registro sólo puede tener un padre. No se puede tener un tipo de registro como *usa*, que tiene dos padres en el modelo en red. El diseñador tiene que decidir si *usa* se modela como hijo del tipo de registro *elementos* o como hijo del tipo de registro *proyectos*. La figura representa el caso en que el tipo de registro *usa* es hijo de *proyectos*. En ella se combinan los campos del registro *elemento* con el registro *usa* para formar el registro *elementos-usados*, que es hijo del registro *proyecto*. Los registros de *elementos-usados* almacenan los detalles de cada uso de elemento en un proyecto. La desventaja de este acercamiento es que *color* y *peso* de un elemento se almacenan más de una vez. De hecho, están duplicados para cada proyecto que usa un elemento.

Es posible una representación jerárquica alternativa. Por ejemplo, se podría hacer de los registros *usados* un hijo de los registros *elementos* y los registros *proyectos* un hijo de los registros *elementos*. En este caso, los datos de *proyecto* estarían duplicados para cada elemento que se usase en él. Alternativamente, se podría almacenar cada registro *usa* dos veces, una como hijo de registro *proyectos* y otra como hijo del registro *usa*. Los diseñadores deben elegir entre estas alternativas.

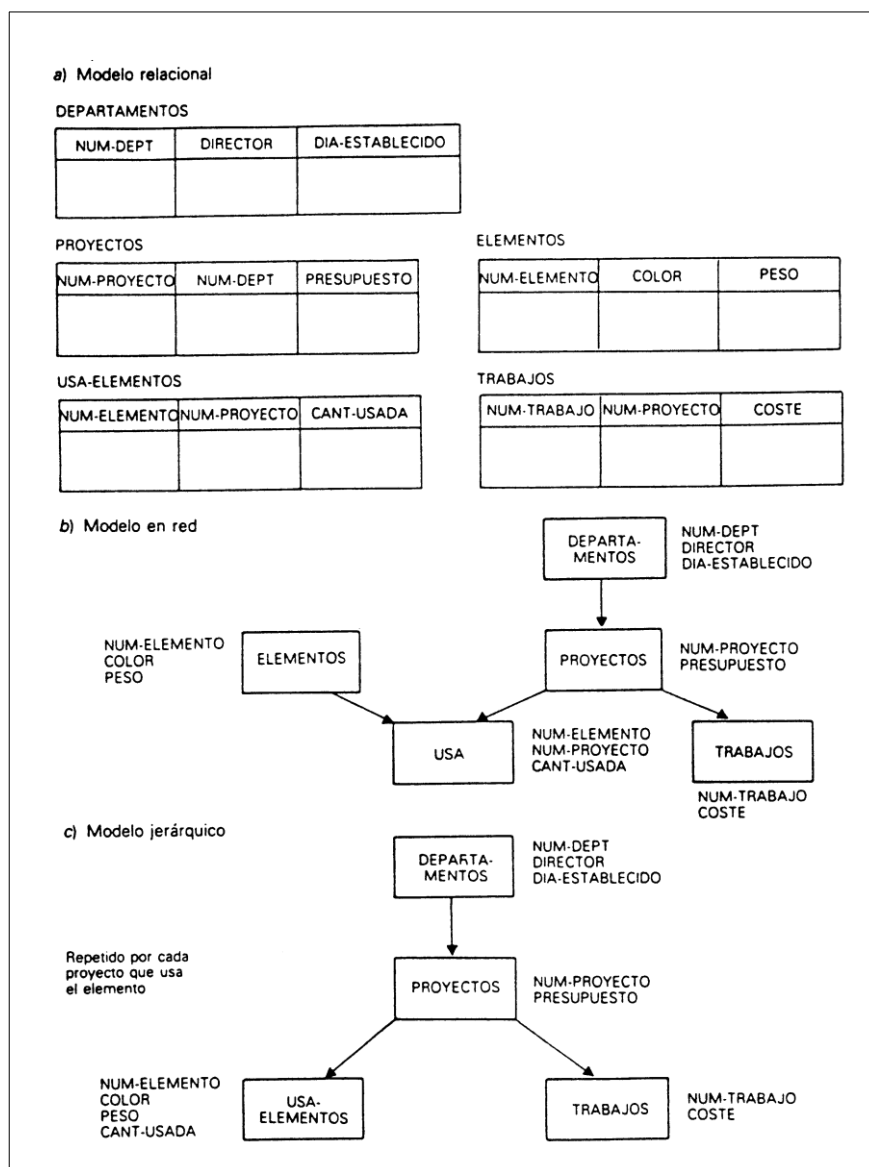


Figura 53.5. Modelos de datos

5. CONVERSIÓN EN ESTRUCTURA SGBD

Los pasos dependientes de SGBD del diseño de base de datos convierten la estructura de registro en una estructura soportada por un SGBD. Este proceso tiene dos pasos:

- Diseño lógico para convertir la estructura lógica de registro en un modelo de datos soportado por el sistema de gestión de base de datos, y
- Diseño físico para elegir la estructura física de la base de datos. (Se ve en otro tema)

DISEÑO LÓGICO

El diseño lógico convierte la estructura lógica de registro en una estructura soportada por un SGBD. El método de conversión depende del tipo de modelo de datos soportado por el SGBD. La conversión más simple es cuando el SGBD soporta el modelo relacional. En este caso no se necesita construir una estructura de registro lógico. Al contrario, la descripción relacional de los datos y cada relación en el modelo de usuario pasa a relación SGBD. Las conversiones en estructuras en red o jerárquica son más elaboradas.

Conversión en sistema de gestión de bases de datos en red

La estructura de registro lógico se puede convertir, de forma relativamente sencilla, en un modelo en red. Cada registro lógico se convierte en un tipo de registro y cada enlace de registro lógico pasa a tipo de registro. Un ejemplo de este tipo de conversión se ve en la figura 53.6. En la figura hay una estructura lógica de registro a convertir. Esta estructura se convierte en la estructura en red mostrada en la figura.

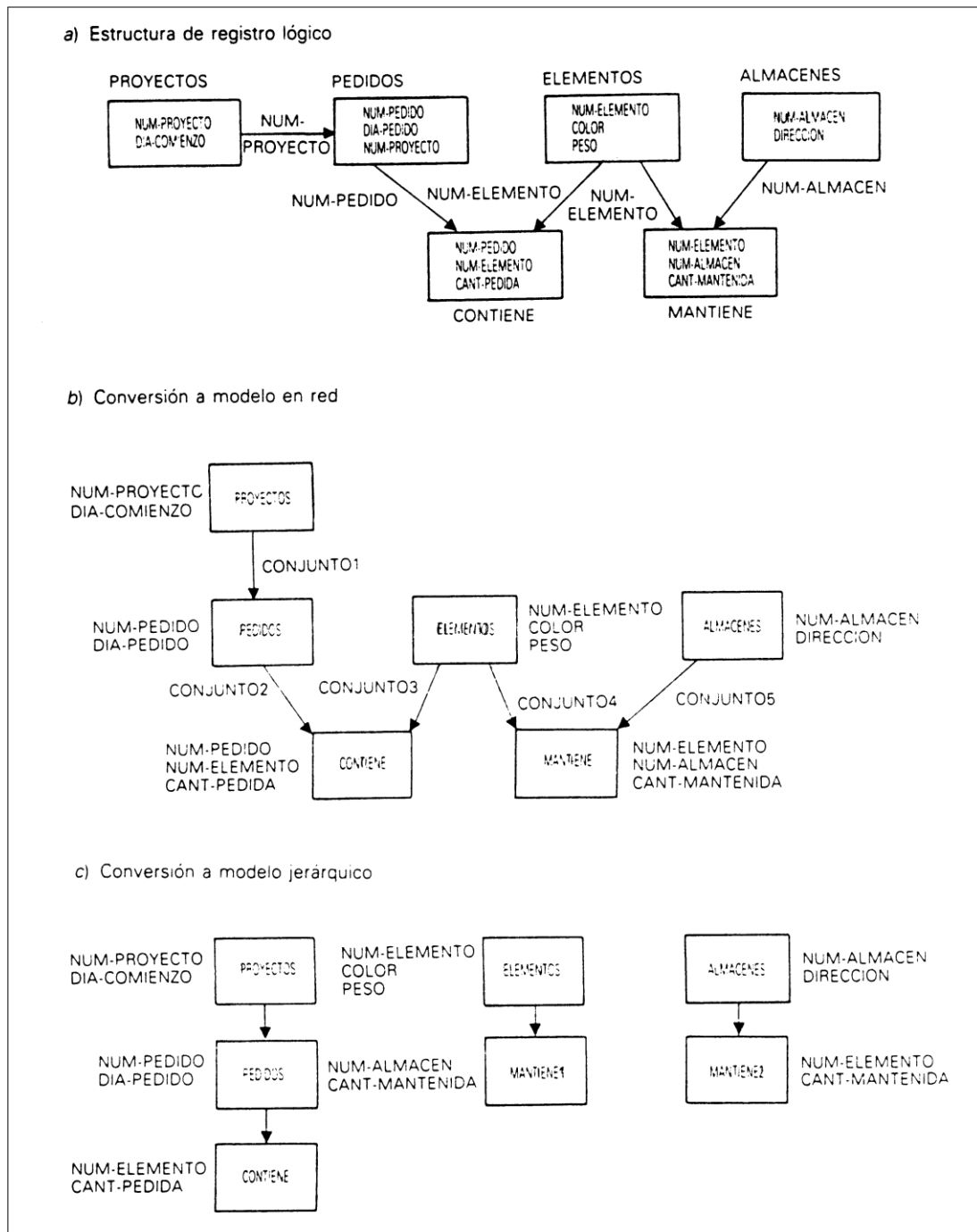
Cada registro lógico pasa a ser un tipo de registro en la estructura en red, y todos los enlaces en la estructura lógica de registro forman enlaces en la representación en red.

Conversión en sistema de gestión de bases de datos jerárquicas

La conversión en modelo jerárquico impone una restricción adicional. Cada tipo de registro en un modelo jerárquico puede tener como mucho un padre. Este no es el caso en el diseño de modelo en red. Por ejemplo, los registros *contiene* y *mantiene* de la figura, tienen dos padres en la estructura de registro lógico. Después de la conversión en modelo en red, también tienen dos padres. Sin embargo, en sistemas jerárquicos no se permiten dos padres para un tipo de registro. El diseñador debe elegir un padre de entre varios padres posibles. Alternativamente, un registro LRS puede aparecer más de una vez en la estructura lógica jerárquica. Aparecerá una vez para cada padre en la estructura lógica de registro.

La figura ilustra una posible conversión en un modelo de datos jerárquico. En ella, el tipo de registro *pedidos* se elige como padre del registro *contiene*. Es más fácil en este diseño acceder al contenido de cada pedido que encontrar los pedidos que contiene un elemento dado. El registro lógico *mantiene*, sin embargo, aparece dos veces en el diseño, una vez como *mantiene1* y otra como *mantiene2*. El primero tiene como padre *elementos* y el segundo *almacenes*. El padre lógico *almacenes* se utiliza para permitir acceder fácilmente el contenido de cada almacén. El padre lógico *elementos* se utiliza para permitir un acceso fácil al emplazamiento de los elementos. Sin embargo, obsérvese que ahora hay información, *cant-mantenida*, duplicada.

Para eliminar tales duplicaciones, muchos sistemas jerárquicos ofrecen hoy día enlaces físicos entre bases de datos físicas separadas.



6. BIBLIOGRAFÍA

Pressman, Roger S.
Ingeniería del Software
 Mc Graw-Hill, 1993

López, Antonio
Metodologías de Desarrollo
 Ra-ma, 1990