

UNIVERSIDAD DE CASTILLA-LA MANCHA

ESCUELA SUPERIOR DE INFORMÁTICA

**EL MODELO DE DATOS
JERÁRQUICO**

Profesor: D. Francisco Ruiz

Alumna: M. Ángeles Moraga de la Rubia

ASIGNATURA: BASE DE DATOS

----- 000 0 000 -----

Ciudad Real, 26 de abril de 2001

ÍNDICE

	<u>Página</u>
1.- Introducción	2
2.- Conceptos básicos	2
3.- Características de la estructura jerárquica	3
4.- Esquema y ocurrencia de árbol	6
5.- Definición del modelo jerárquico	6
6.- Problemas del modelo jerárquico	6
7.- Transformación de un esquema e/r en un esquema jerárquico	9
8.- La función de manipulación de datos en los modelos jerárquicos	11
9.- El lenguaje de manipulación jerárquico	12
10.- Ejemplo de lenguaje para el modelo jerárquico	12
11.- Bibliografía	15

1.- INTRODUCCIÓN

Entre los primeros modelos de datos que surgieron en los SGBD comerciales se encuentran las estructuras en árbol, propias de los productos jerárquicos.

Los árboles, como instrumentos para la representación de estructuras de datos, presentan problemas por su poca flexibilidad, lo que da origen a una falta de adaptación a muchas organizaciones reales.

No se ha llegado a una formalización matemática del modelo y de sus lenguajes, como ha ocurrido en el caso del relacional; ni tampoco se ha intentado su estandarización, a pesar de lo cual los productos jerárquicos (más bien debíamos decir el IMS y el DL/I de IBM como máximos exponentes de estos sistemas) consiguieron altas cuotas de mercado, aunque la actual difusión de la tecnología relacional los han llevado a convertirse en sistemas superados, lo cual no quiere decir que no persistan todavía importantes aplicaciones soportadas en estos productos que están trabajando, por su eficiente respuesta, a satisfacción de sus usuarios, en tanto que las aplicaciones desarrolladas sobre ellos se mantengan sin apenas cambios.

IMS es un producto de software de IBM. Fue uno de los primeros sistemas de bases de datos comerciales -- la primera versión de sistema (AIMS/360 Versión 1") apareció en 1968--

Los productos basados en este tipo de modelos han perdido las altas cuotas de mercado de las que disfrutaban hace una década y se consideran sistemas muy superados por la tecnología relacional, sin embargo aún persisten muchas aplicaciones basadas en este modelo.

2.- CONCEPTOS BÁSICOS

La implementación del modelo Jerárquico en los productos se lleva a cabo en base a punteros; estructura física que varía según los productos, e incluso un mismo producto proporciona distintas organizaciones físicas a fin de que el usuario pueda conseguir una mayor eficiencia en el diseño físico de la base de datos.

El producto comercial de tipo Jerárquico más extendido y el único que ha llegado hasta nuestros días es el IMS de IBM con su lenguaje de datos DL/I². Otro sistema Jerárquico, el System 2000 también tuvo una alta aceptación comercial y fue adquirido posteriormente por el Instituto SAS.

El modelo jerárquico fue desarrollado para permitir la representación de aquellas situaciones de la vida real en las que predominan las relaciones de tipo 1 : N.

Es un modelo muy rígido en el que las diferentes entidades de las que está compuesta una determinada situación, se organizan en niveles múltiples de acuerdo a una estricta relación PADRE/HIJO, de manera que un padre puede tener más de un hijo, todos ellos localizados en el mismo nivel, y un hijo únicamente puede tener un padre situado en el nivel inmediatamente superior al suyo.

Esta estricta relación PADRE/HIJO implica que no puedan establecerse relaciones entre segmentos dentro de un mismo nivel.

La representación gráfica de un modelo jerárquico se realiza mediante la estructura de ARBOL INVERTIDO, en la que el nivel superior está ocupado por una única entidad, bajo la cual se distribuyen el resto de las entidades en niveles que se van ramificando. Los diferentes niveles quedan unidos por medio de las relaciones,

Las entidades se denominan en el caso particular del modelo jerárquico SEGMENTOS, mientras que los atributos reciben el nombre de CAMPOS.

Los segmentos, se organizan en niveles de manera que en un mismo nivel estén todos aquellos segmentos que dependen de un segmento de nivel inmediatamente superior.

3.- CARACTERÍSTICAS DE LA ESTRUCTURA JERÁRQUICA

Los segmentos, en función de su situación en el árbol y de sus características, pueden denominarse como:

1) **SEGMENTO PADRE:** Es aquél que tiene descendientes, todos ellos localizados en el mismo nivel.

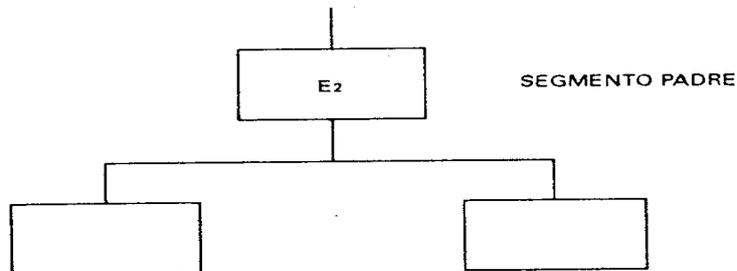


Fig. 1

2) **SEGMENTO HIJO:** Es aquél que depende de un segmento de nivel superior. Todos los hijos de un mismo padre están en el mismo nivel del árbol.

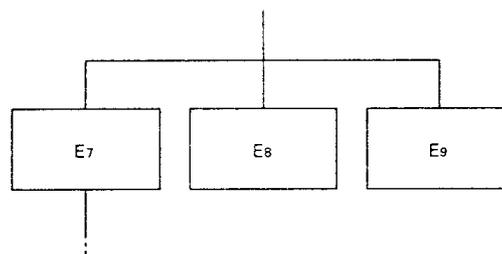


Fig. 2

3) **SEGMENTO RAÍZ:** El segmento raíz de una base de datos jerárquica es el padre que no tiene padre. La raíz siempre es única y ocupa el nivel superior del árbol.

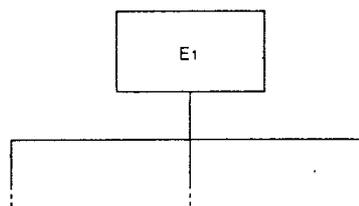


Fig. 3

Un ejemplo sería:

Sea una determinada empresa de ámbito nacional con delegaciones por todo el país; esta empresa tiene centralizadas todas las compras de material de sus delegaciones en la oficina central, para lo cual dispone de una base de datos jerárquica que le permite almacenar los datos de todos sus proveedores.

La base de datos de proveedores, denominada PROVEBAS, presenta cinco segmentos. El segmento raíz en el que se almacenan los datos que son comunes a todos los proveedores, como pueden ser: Nombre de la empresa, Director de la empresa, N.I.F., entre otros. Este segmento se denomina DATGEN.

En el segundo nivel del árbol hay tres segmentos dependientes del segmento raíz. El primero de ellos contiene las direcciones de las sucursales de la empresa proveedora, indicando: Calle, Número, ciudad, Tipo de

dirección, entre otros. El nombre de este segmento es DIRPRO.

El segundo segmento que ocupa este nivel es el que contiene los datos de todos los productos suministrados por cada una de las empresas proveedoras, Este segmento se denota como PRODUC. El último segmento del segundo nivel es el que permite guardar las diferentes notas informativas que sobre un proveedor van remitiendo las delegaciones a la oficina central, este segmento se denomina NOTINF.

El tercer nivel del árbol está ocupado por un solo segmento que es el que permite almacenar las zonas de distribución de cada uno de los productos suministrados por los diferentes proveedores, este segmento se reconoce como AREDIS y depende del segmento PRODUC.

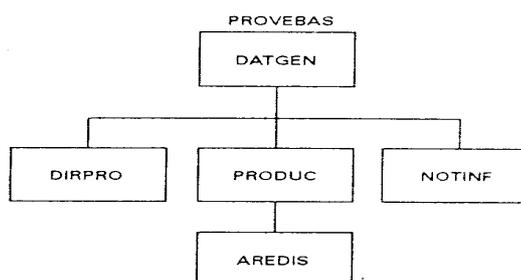


Fig. 4

Una OCURRENCIA de un segmento de una base de datos jerárquica es el conjunto de valores particulares que toman todos los campos que lo componen en un momento determinado.

Un REGISTRO de la base de datos es el conjunto formado por una ocurrencia del segmento raíz y todas las ocurrencias del resto de los segmentos de la base de datos que dependen jerárquicamente de dicha ocurrencia raíz.

La relación PADRE/HIJO en la que se apoyan las bases de datos jerárquicas, determina que el camino de acceso a los datos sea ÚNICO; este camino, denominado CAMINO SECUENCIA JERÁRQUICA, comienza siempre en una ocurrencia del segmento raíz y recorre la base de datos de arriba a abajo, de izquierda a derecha y por último de adelante a atrás.

El esquema es una estructura arborescente compuesta de nodos, que representan las entidades, enlazados por arcos, que representan las asociaciones o interrelaciones entre dichas entidades.

La estructura del modelo de datos jerárquico es un caso particular de la del modelo en red, con fuertes restricciones adicionales derivadas de que las asociaciones del modelo jerárquico deben formar un árbol ordenado, es decir, un árbol en el que el orden de los nodos es importante. Una estructura jerárquica, tiene las siguientes características:

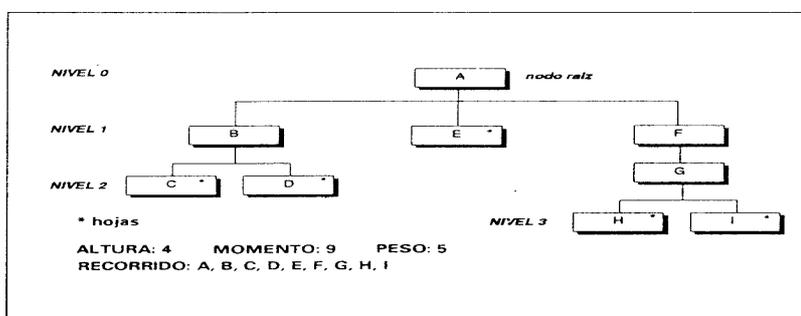


Fig. 5

- El árbol se organiza en un conjunto de niveles.
- El nodo raíz, el más alto de la jerarquía, se corresponde con el nivel 0.
- Los arcos representan las asociaciones jerárquicas entre dos entidades y no tienen nombre, ya que no es necesario porque entre dos conjuntos de datos sólo puede haber una interrelación.

- Mientras que un nodo de nivel superior (padre) puede tener un número ilimitado de nodos de nivel inferior (hijos), al nodo de nivel inferior sólo le puede corresponder un único nodo de nivel superior. en otras palabras, un progenitor o padre puede tener varios descendientes o hijos, pero un hijo sólo tiene un padre.
- Todo nodo, a excepción del nodo raíz, ha de tener obligatoriamente un padre.
- Se llaman *hojas* los nodos que no tienen descendientes.
- Se llama *altura* al número de niveles de la estructura jerárquica.
- Se denomina *momento* al número de nodos.
- El número de hojas del árbol se llama *peso*.
- Sólo están permitidas las interrelaciones 1:1 ó 1:N
- Cada nodo no terminal y sus descendientes forman un subárbol, de forma que un árbol es una estructura recursiva.

El árbol se suele recorrer en preorden; es decir, raíz, subárbol izquierdo y subárbol derecho.

Entre las restricciones propias de este modelo se pueden resaltar:

- A) Cada árbol debe tener un único segmento raíz.
- B) No puede definirse más de una relación entre dos segmentos dentro de un árbol.
- C) No se permiten las relaciones reflexivas de un segmento consigo mismo.
- D) No se permiten las relaciones N:M.
- E) No se permite que exista un hijo con más de un padre.
- F) Para cualquier acceso a la información almacenada, es obligatorio el acceso por la raíz del árbol, excepto en el caso de utilizar un índice secundario.
- G) El árbol debe recorrer siempre de acuerdo a un orden prefijado: el camino jerárquico.
- H) La estructura del árbol, una vez creada, no se puede modificar.

Las estructuras jerárquicas se clasifican también como:

- *Lineales*: es un caso particular y simple en el que cada tipo de registro padre sólo puede tener un tipo de registro hijo, donde se muestra la interrelación entre DEPARTAMENTO y EMPLEADO.

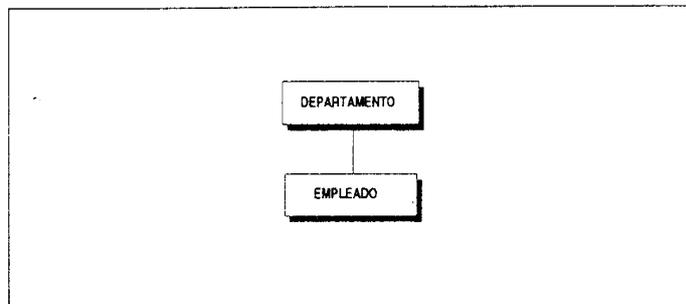


Fig. 6

- *Arborescente propiamente dicha*: un tipo de registro padre puede tener varios tipos de registro descendientes.

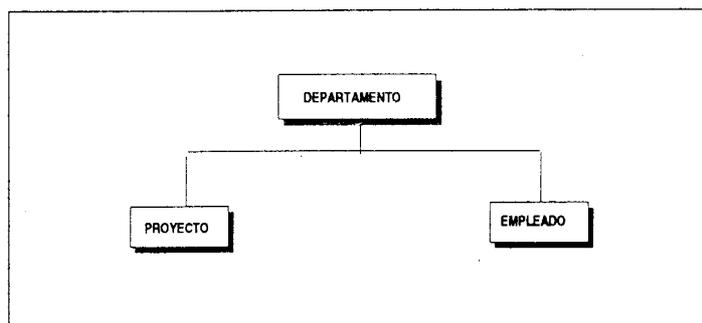


Fig. 7

4.- ESQUEMA Y OCURRENCIA DE ÁRBOL

Un esquema jerárquico consiste en una descripción de un determinado universo del discurso mediante un árbol en el que los nodos representan los tipos de registro (entidades), y los arcos, los tipos de interrelaciones jerárquicas existentes entre los mismos. Una ocurrencia de dicho esquema será también un árbol, pero en él los nodos representan las ocurrencias de los registros, y los arcos, las interrelaciones jerárquicas entre dichas ocurrencias.

Una base de datos jerárquica está formada por una colección o *bosque* de árboles disjuntos.

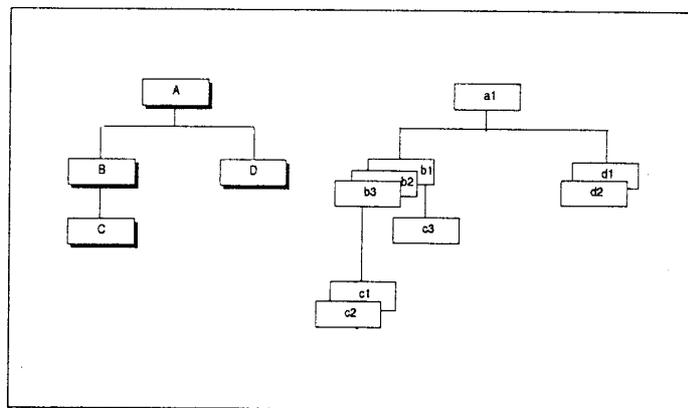


Fig. 8

5.- DEFINICIÓN DEL MODELO JERÁRQUICO

Podemos definir el modelo jerárquico como:

- Un conjunto de tipos de entidad E_1, E_2, \dots, E_n .
- Un conjunto de interrelaciones o asociaciones no nominadas R_{ij} que conectan los tipos de entidad E_i y E_j .
- Un conjunto de restricciones inherentes que provienen de la estructura jerárquica.

6.- PROBLEMAS DEL MODELO JERÁRQUICO

El modelo de datos jerárquico presenta importantes inconvenientes, que provienen principalmente de su rigidez, la cual deriva de la falta de capacidad de las organizaciones jerárquicas para representar sin redundancias ciertas estructuras muy difundidas en la realidad, como son las interrelaciones reflexivas y N:M. En la figura 9 se muestran algunos ejemplos de estructuras no admitidas en el modelo jerárquico puro.

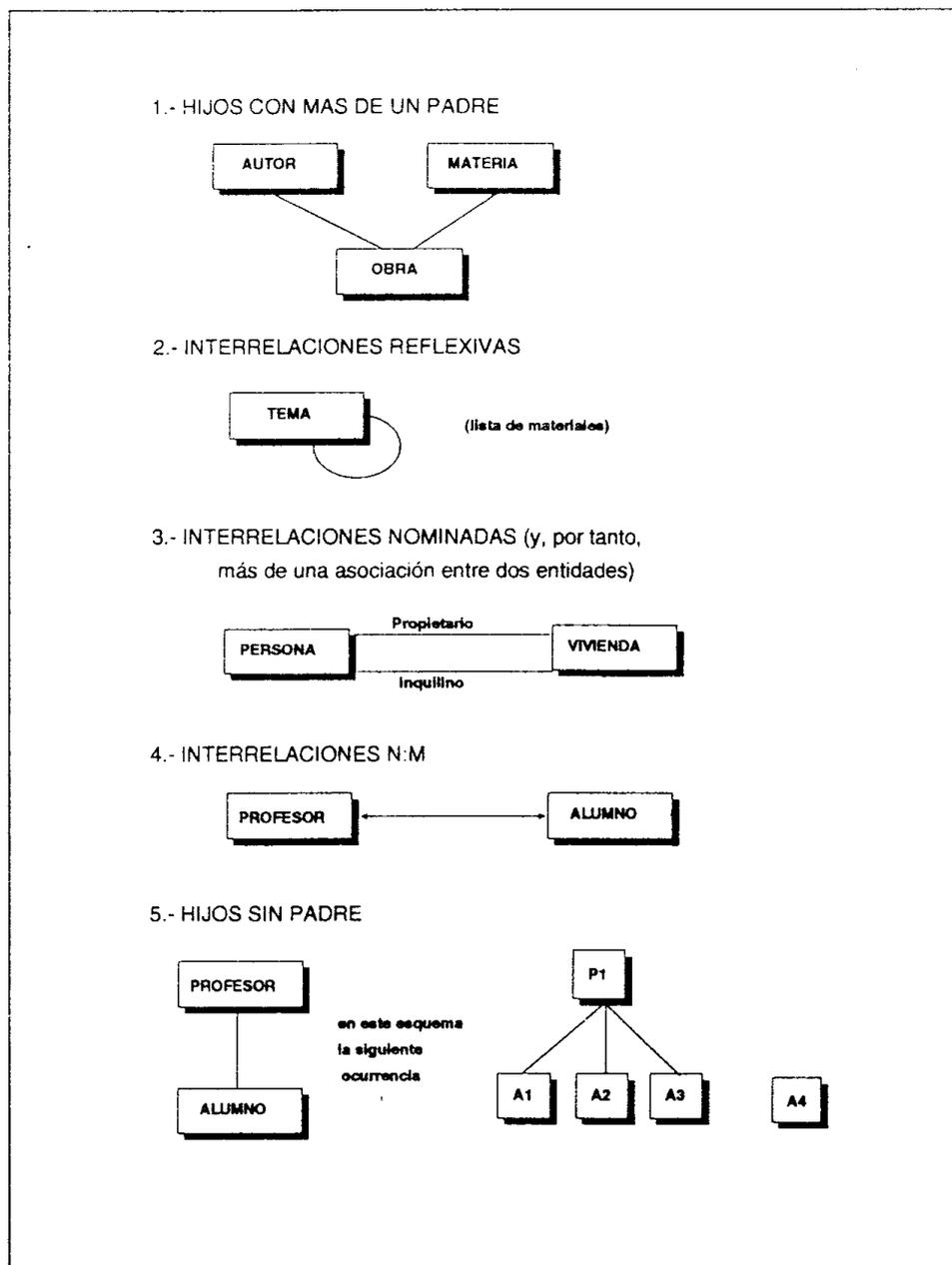
La poca flexibilidad de este modelo puede obligar a la introducción de redundancias cuando es preciso instrumentar, mediante el modelo jerárquico, situaciones del mundo real que no responden a una jerarquía; así ocurre con el esquema no jerárquico de la figura 10 cuando se quiere adaptar al modelo jerárquico, donde es preciso crear dos árboles e introducir redundancias, ya que los registros D, H e I aparecen en ambas jerarquías. Se trata de una pobre solución de diseño.

Se puede calcular un índice de redundancia mediante

$$Ir \ni \frac{N1. \text{ nodos extra}}{N1. \text{ total de nodos}} \times 100$$

En nuestro caso, el índice de redundancia se calcularía como

$$Ir \ni \underline{3} \times 100 \ni 20\%$$



La redundancia que hemos calculado es una redundancia lógica que, en general, no coincide con la redundancia física. Esto es debido a que al almacenar los datos en el dispositivo físico no se suelen repetir los registros completos, sino sólo los identificadores, o bien se introducen punteros.

Fig. 9

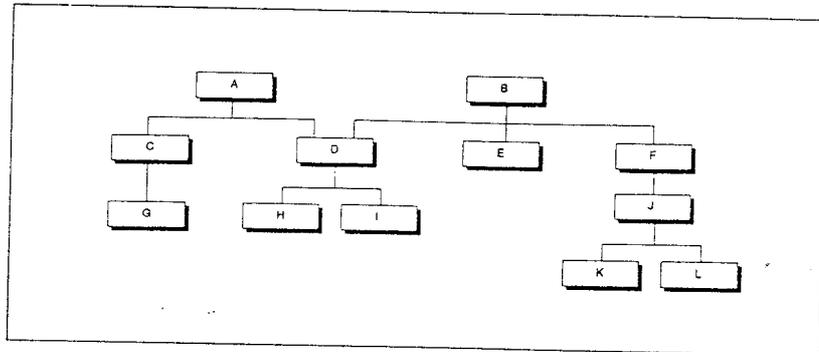


Fig. 10

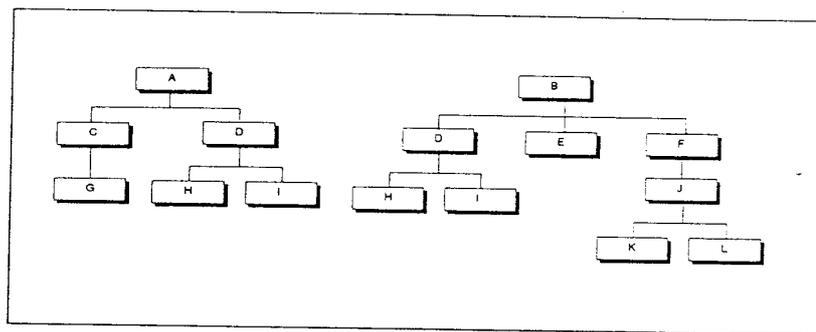


Fig. 11

En general, los productos ofrecen algún tipo de solución a estos problemas.

Otra limitación importante del modelo jerárquico es no estar preparado para representar interrelaciones N:M, como la existente entre profesores y alumnos. Se podría pensar en emplear la estructura que aparece en la Figura 12, duplicando las ocurrencias del registro ALUMNO.

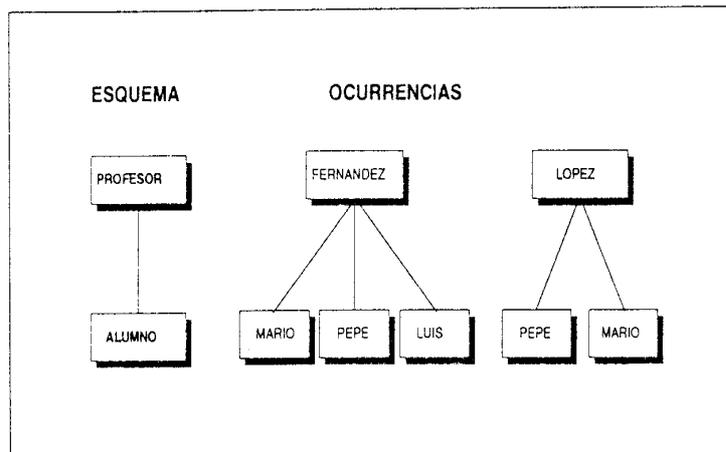


Fig. 12

Además del grave problema que provocan estas redundancias no controladas por el sistema, existe otro importante inconveniente en este tipo de solución como es la no conservación de las simetrías naturales existentes en el mundo real.

Las actualizaciones en las bases de datos jerárquicas pueden también originar problemas, debido a las restricciones inherentes al modelo:

- Toda alta, a no ser que corresponda a un nodo raíz, debe tener un padre. Por ejemplo, sería imposible insertar en la base de datos de la Figura 12 un alumno que aún no tuviera asignado profesor.

Se podría resolver el problema creando un padre ficticio para este registro, pero con ello se complicaría la labor del usuario de la base de datos, que tendría, cuando deseara conectar el registro a su padre real, que eliminarlo de la base de datos y volver a introducirlo ligado a su verdadero padre. Además, el sistema no podría distinguir entre padre ficticio y real y cuando contara el número de profesores de la base de datos nos daría uno más.

- La baja de un registro implica que desaparezca todo el subárbol que tiene dicho registro como nodo raíz, con lo que pueden desaparecer datos importantes que convendría conservar en la base de datos.

Los SGBD basados en el modelo jerárquico suelen facilitar instrumentos que los dotan de una mayor flexibilidad para representar estructuras no estrictamente jerárquicas.

Por lo que respecta a las restricciones de usuario, el modelo jerárquico no ofrece ninguna posibilidad de definir las.

En cuanto a las ventajas proporcionadas por los SGBD de tipo jerárquico, cabe citar su sencillez de comprensión y la mayor facilidad de instrumentación en los soportes físicos, aunque esto depende en gran medida de los productos.

7.- TRANSFORMACIÓN DE UN ESQUEMA E/R EN UN ESQUEMA JERÁRQUICO.

Ya se han señalado los inconvenientes que presenta el modelado del mundo real según esquemas jerárquicos, y también hemos indicado una técnica de diseño jerárquico que consiste en introducir redundancias.

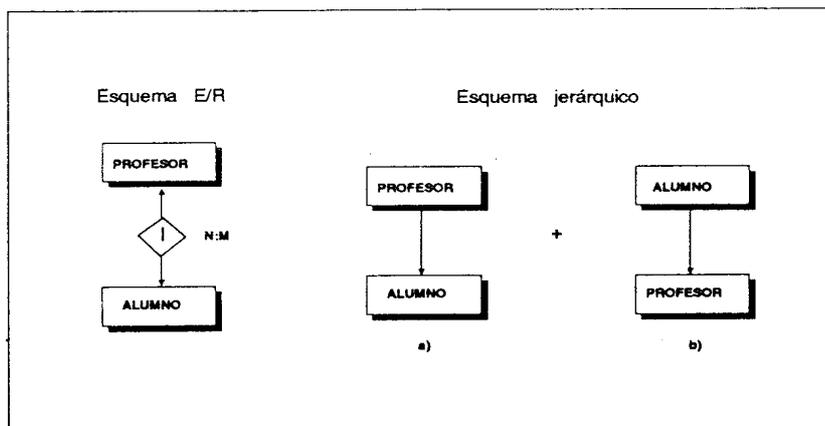


Fig. 13

Se podría evitar la pérdida de simetrías introduciendo mucha mayor redundancia, como se muestra en la Figura 13, donde se presenta la transformación de un esquema E/R con dos entidades y una interrelación N:M es un esquema jerárquico en el que existen dos árboles, de modo que se conservan las simetrías naturales, ya que los algoritmos para dos preguntas simétricas, como son recuperar los alumnos de un profesor y recuperar los

profesores de un alumno, serían también simétricos.

Soluciones de este tipo no son en absoluto eficientes.

A partir del modelo E/R, vamos a analizar la forma de transformar algunos tipos, de interrelaciones al modelo jerárquico.

A) Interrelaciones 1:N con cardinalidad mínima 1 en la entidad padre.

En este caso no existe ningún problema y el esquema jerárquico resultante será prácticamente el mismo que en el ME/R.

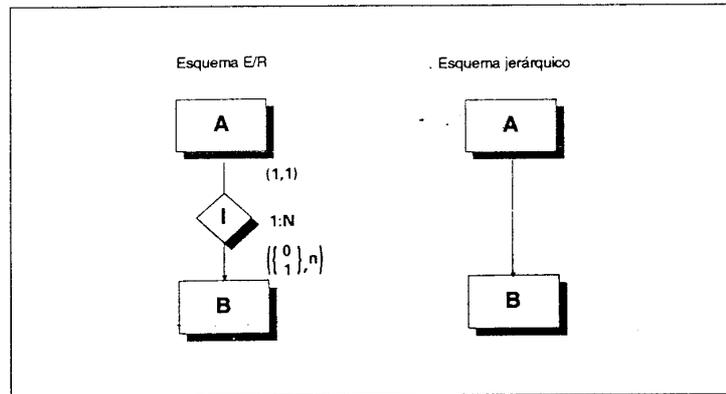


Fig. 14

B) Interrelaciones 1:N con cardinalidad mínima 0 en el registro propietario.

El problema es que podrían existir hijos sin padre, por lo que o se crea un padre ficticio para estos casos o se crean dos estructuras arborescentes.

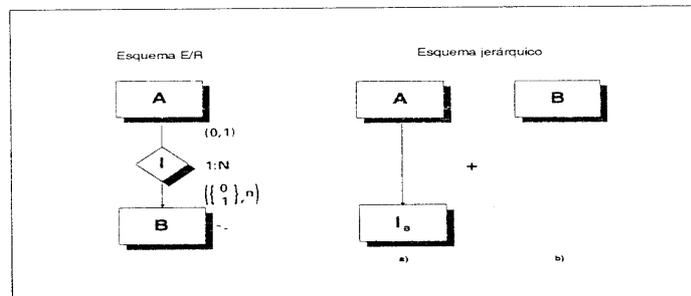


Fig. 15

La primera estructura arborescente tendrá como nodo padre el tipo de registro A y como nodo hijo los identificadores del tipo de registro B. De esta forma no se introducen redundancias, estando los atributos de la entidad B en la segunda arborescencia, en la cual sólo existiría un nodo raíz B sin descendientes.

C) Interrelaciones N:M

La solución es muy parecida, creándose también dos arborescencias.

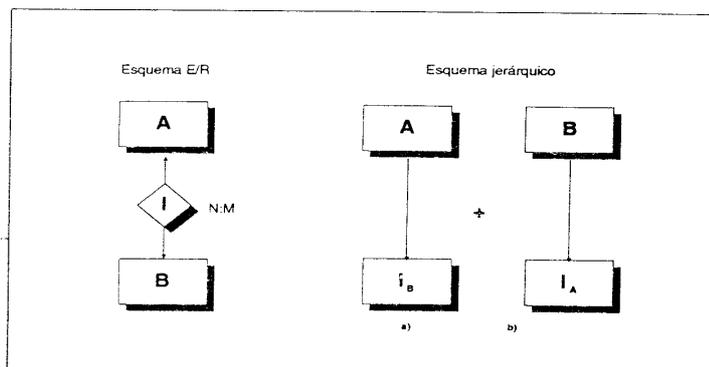


Fig. 16

La solución es independiente de las cardinalidades mínimas. Se podría suprimir, en la primera arborescencia o en la segunda, el registro hijo, pero no se conservaría la simetría.

D) Interrelaciones reflexivas

La jerarquía a) se utilizaría siempre que se desee obtener la explosión.

La aplicación de estas normas de diseño evita la introducción de redundancias, así como la pérdida de simetría, pero complica enormemente el esquema jerárquico resultante que estará constituido por más de un árbol, lo que no resulta fácilmente comprensible a los usuarios.

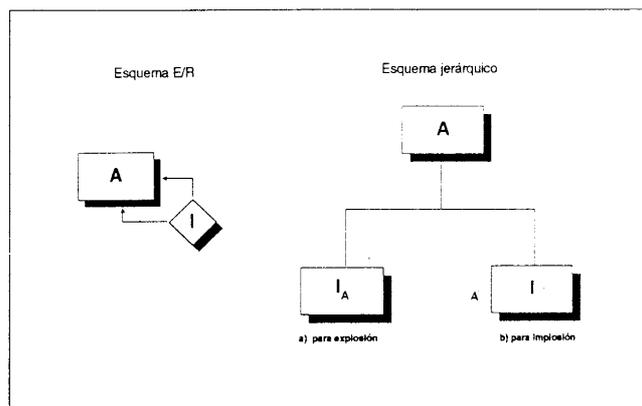


Fig. 17

8.- LA FUNCIÓN DE MANIPULACIÓN DE DATOS EN LOS MODELOS JERÁRQUICOS.

La manipulación de datos jerárquicos necesita *localizar* (seleccionar) primero los datos sobre los que va a trabajar para realizar a continuación la acción de recuperación o actualización sobre dichos datos.

A) Localización o selección.

La función de selección jerárquica es de tipo navegacional, es decir, trabaja registro a registro. Dada la sencillez del modelo, la función de selección es también muy sencilla, existiendo únicamente las siguientes formas básicas de búsqueda:

- Seleccionar un determinado registro que cumpla una cierta condición. en el lenguaje DL/I se realizará este tipo de selección mediante una sentencia (GET UNIQUE -GU-) que activará el primer registro que cumpla la condición especificada en el predicado que acompaña a la sentencia.
- Seleccionar el siguiente registro, que se encuentra perfectamente definido al existir un único camino jerárquico. También en este caso se puede especificar una condición que habrá de cumplir el registro para ser seleccionado. En DL/I se utiliza una sentencia (GET NEXT -GN-) que selecciona y, al mismo tiempo, recupera el siguiente registro en el preorden.
- Seleccionar el siguiente registro dentro de un padre. Esta sentencia (GET NEXT PARENT -GNP-) es análoga a la anterior, pero la selección termina cuando no haya más descendientes de ese padre.
- Seleccionar el registro padre de otro dado (que ha sido activado previamente) se conoce como normalización jerárquica ascendente, mientras que la selección de descendientes se llama normalización jerárquica descendente.

B) Acción

Una vez seleccionado un registro, se tendrá que realizar sobre él una acción, sea de recuperación o de

actualización.

La recuperación, que va asociada a la selección en el DL/I, consiste en llevar el registro marcado como activo en la selección realizada previamente al área de entrada/salida. Se utiliza la sentencia GET.

En cuanto a la actualización, es preciso distinguir entre:

- *Insertar* un conjunto de datos (INSERT -ISRT-)
- *Borrar* un conjunto de datos (DELETE -DLET-)
- *Reemplazar* -modificar- uno o varios campos de un registro (REPLACE -REPL-)

Debido a la naturaleza jerárquica de las conexiones entre registros, las inserciones y borrados de registros requieren consideraciones especiales:

- Cuando un nuevo registro se inserta en una base de datos jerárquica, excepto para la raíz, tiene que ser conectado a un nodo padre previamente seleccionado mediante alguna sentencia de selección. El nuevo registro se inserta como hijo del registro padre seleccionado.

- Cuando un registro se borra en una base de datos jerárquica, excepto si se trata de una hoja, se han de borrar todos los registros descendientes de él.

9.- EL LENGUAJE DE MANIPULACIÓN JERÁRQUICO.

Una instrucción de un lenguaje de manipulación constará:

- Un operador que indica el tipo de operación a realizar.
- Los datos sobre los que se lleva a cabo la operación.
- Una condición, que servirá para seleccionar el conjunto de datos sobre el que se desea trabajar, y que es una expresión de tipo lógico, es decir, constantes y variables unidas por operadores de comparación y del álgebra de Boole.

10.- EJEMPLO DE LENGUAJE PARA EL MODELO JERÁRQUICO.

1 Data Definition Language.

Declaración de un esquema jerárquico:

```
SCHEMA NAME = <nombre del esquema>  
<declaraciones de árboles>
```

Declaración de un árbol:

```
TREE <nombre> <lista de declaraciones de tipos de registros>
```

Declaración de tipo de registro:

```
RECORD <nombre> <información>
```

La información es de las siguientes clases:

a) Campos: <n1_nivel> <nombre> <tipo>

Ejemplo: 1 Cantidad integer

b) Posición:

```
ROOT | PARENT = <nombre>
```

c) Registro virtual (opcional):

```
VIRTUAL <nombre-registro> IN <nombre_árbol>
```

d) Punteros:

POINTER = [PARENT | <lista tipos de punteros>]

Ejemplo:

```
TREE Planes_de_Estudio
  RECORD Centro ROOT
    1 codigo char(3)
    1 nombre char(30)
  RECORD Estudio PARENT=Centro
    1 codigo char(2)
    1 nombre char(50)
    1 duracion integer
  RECORD Asignatura PARENT=Estudio
    1 clave char(3)
    1 nombre char(30)
    1 creditos integer
```

2 Data Manipulation Language.

2.1 Consultas

Se realizan con la orden GET. Se realizan dos tareas:

- 1) Localiza un registro en la BD y hace que el puntero de dirección indique a él.
- 2) Copia dicho registro a la plantilla del área de trabajo

Clases de Consultas:

- GET FIRST <tipo_registro> [WHERE <condición>]

Localiza el primer registro del tipo indicado. si hay cláusula where, se localiza el primero que cumple la condición. También se usa como GET UNIQUE.

- GET NEXT <tipo_registro> [WHERE <condición>]

Localiza el siguiente registro del tipo indicado. Si hay cláusula where, se localiza el siguiente que cumple la condición.

GET NEXT WITHIN PARENT <tipo_registro> [WHERE <condición>]

Localiza el siguiente registro del tipo indicado, dentro de un subárbol cuya raíz es el último registro localizado con GET FIRST o GET NEXT. Si hay cláusula where, se localiza el siguiente que cumple la condición.

Consultas con retención del registro:

El usuario que hace la consulta retiene el registro hasta que lo libera. El registro está bloqueado y no pueden acceder a él los demás usuarios. Las órdenes con retención son equivalentes a las anteriores:

GET HOLD { FIRST | NEXT | NEXT WITH }

2.2. Actualizaciones.

Las operaciones se realizan a nivel de registro. Los registros se almacenan desde las plantillas del área de trabajo.

- INSERT <tipo_registro> [WHERE <condición>]

El registro se inserta en la primera posición de la base de datos donde se pueden colocar registros de ese tipo. Si hay cláusula where el sistema busca un registro que satisfaga la condición, y el registro recién creado se inserta como su hermano más a la izquierda.

- REPLACE

Sustituye el contenido de un registro con el de la plantilla del área de trabajo. Dicho registro debe haber sido recuperado previamente con un GET HOLD ... para que el puntero de dirección señale hacia él.

- DELETE

Elimina un registro. Dicho registro debe haber sido recuperado previamente con un GET HOLD ... para que el puntero de dirección señale hacia él.

BIBLIOGRAFÍA

- De Miguel, A.; Piattini, M.; Fundamentos y Modelos de Bases de Datos (2ª edición). Ra-Ma, 1999.
- De Miguel, A.; Piattini, M.; Concepción y Diseño de Bases de Datos Relacionales (2ª edición). Ra-Ma, 1999.
- Lucas Gómez, A.; Diseño y Gestión de Sistemas de Bases de Datos. Paraninfo.
- Ullman, J.D.; Widom, J.; Introducción a los Sistemas de Bases de Datos. Prentice Hall, 1999.