Una experiencia en diseño lógico ("logical design") de bases de datos

JOSE M.ª GIRÓ. Servicio de Informática, C.E.P.S.A.
RAFAEL ANDREU. I.E.S.E. y Facultad de Informática, U.P.B.

1. INTRODUCCION

El objetivo de este artículo es discutir y compartir una experiencia de diseño de una base de datos para una aplicación concreta. Lo que nos proponemos en realidad es poner un ejemplo real de utilización del concepto de base de datos, del que con frecuencia uno tiene ocasión de leer acerca de sus excelencias pero sin (o con muy pocas) referencias a problemas concretos.

Por otro lado, creemos que puede resultar ilustrativo exponer qué tipo de consideraciones juegan o pueden jugar un papel importante a la hora de diseñar una estructura "lógica" para la base de datos en cuestión, condicionadas –como están con frecuencia en la práctica– por circunstancias tan comunes como el hecho de disponer de un sistema de gestión concreto, o querer orientar el diseño de manera que ciertas consultas a la B.D. resulten lo más eficientes posible, o el deseo de aprovecharse de cualquier característica "física" que el sistema de gestión disponible presente y que signifique una ventaja digamos "no desaprovechable" desde el punto de vista de la aplicación que nos interese.

Como quedará claro más adelante, la decisión de emplear el concepto de base de datos para la aplicación que discutimos se fundamentó en los criterios más o menos "tradicionales" o "de libro". Por una parte, el hecho de que con frecuencia, en el entorno de la aplicación, se sintiera la necesidad de información que teóricamente podía elaborarse a partir de ficheros clásicos existentes pero cuyo tiempo de acceso –incluido el de escribir los programas necesarios– convertía su obtención en poco práctica. Por otra, la circunstancia de que existiera un sistema de gestión de B.D. disponible para el equipo empleado en el tratamiento de aquellos ficheros. Desde este punto de vista, y para el usuario final, la tecnología de B.D. significaba únicamente, aunque parezca prosaico, la posibilidad de tener acceso fácil a unos ficheros ya existentes que se podían de esta forma interrelacionar de acuerdo con ciertas características de su contenido. Esto permitía obtener informes no rutinarios, normalmente difíciles de prever, pero que podían ser cruciales para ayudar a tomar determinado tipo de decisiones. Curiosamente, fue en el campo del departamento de personal de una gran empresa donde estas necesidades se manifestaron con especial relevancia.

2. EL PROBLEMA DEL "DISEÑO LOGICO" DE LA B.D.

En el caso que nos ocupa, el problema de elegir un *modelo de datos* (léase jerárquico, en red o relacional) adecuado para la aplicación concreta no lo fue entre otras causas, porque se dio la circunstancia de que razones de disponibilidad limitaron las alternativas a sólo una: el modelo en red (CODASYL). El problema de elección de un sistema de gestión desapareció por las mismas razones. Así, nos vimos limitados desde el primer momento en estas decisiones que, como quien dice, el entorno tomó por nosotros.

Teóricamente, la elección de un modelo determina en gran parte el problema de diseño del esquema para la base de datos en cada caso. Este problema, denominado con frecuencia "diseño lógico" ("logical design") de la B.D. ha sido objeto de especial atención en los últimos años. Sin embargo, no puede decirse que exista una metodología bien definida para abordarlo. A menudo, en la práctica, el proceso de diseño tiene más de arte que de técnica y en él juega, consecuentemente, un papel importante la experiencia previa del diseñador. A pesar de que se han propuesto algunos esquemas conceptuales que ayudan a visualizar el problema (por ejemplo, la propuesta de Chen, 1976) de estructurar el esquema en un modelo "neutral" –su "entity relationship model"– a partir del cual pueden deducirse más o menos automáticamente esquemas "equivalentes" en los tres modelos fundamentales, (ver Chen 1977), o el mismo proceso de normalización de relaciones, útil cuando se emplea el modelo relacional (Date 1975), e incluso algunas propuestas más recientes que pretenden especificar el esquema inicial en el contexto de lo que podríamos llamar un "modelo del mundo", haciendo intervenir características del lenguaje natural y teniendo en cuenta consideraciones de tipo semántico en el mismo (ver Fry y Teorey, 1978 para un resumen), lo cierto es que con frecuencia, y sobre todo cuando el diseñador tiene relativamente poca experiencia en la utilización de bases de datos, el proceso de diseño se desarrolla en términos de sucesivos refinamientos, que lo hacen iterativo por naturaleza y lo transforman, de hecho, en una experiencia diríamos incluso educacional para el diseñador (en el sentido de descubrir, a cada paso, nuevos aspectos del problema que no habían aparecido con claridad anteriormente).

En las secciones siguientes analizamos un problema de diseño concreto, con el que uno de los autores se ha enfrentado recientemente, precisamente desde este punto de vista. Se intenta dar una descripción fehaciente de lo que fue su experiencia personal en el diseño de la

B.D. que se describe a continuación. El objetivo es, en cierto sentido, doble. Por un lado, se intenta documentar, sin embudos, el proceso, a veces penoso, por el que un diseñador sin demasiada experiencia tiene que pasar para lograr un diseño que le convenza mínimamente; esperamos que esto contribuya a hacer perder el miedo a la tecnología que tantos profesionales sienten después de haber leído acerca de bases de datos pero que siguen sin ver muy claro cómo todo esto, que parece tan "académico", "se come". Por otro lado, creemos que la transcripción siguiente ilustra qué tipo de consideraciones ajenas al modelo de datos en sí, juegan un papel importante –o lo han jugado en este caso– y determinan parcialmente el diseño final.

3. PROBLEMAS PRACTICOS EN EL DISEÑO DE BASES DE DATOS

Al estudiar, en la literatura especializada, la *conveniencia* de utilización de sistemas de bases de datos y la metodología de diseño, se encuentran dos grandes tendencias. Por una parte autores como Lyon (1976), prestan especial atención a bases integradas concebidas para contener la mayor parte de la información necesaria para la gestión de la empresa. En otros casos (Martin 1977, p. 19), se considera poco menos que imposible el diseño y utilización de sistemas muy generales. Kroenke (1977) considera que la utilización de bases de datos "pequeñas", como hacen algunas empresas, es una política que puede dar buenos resultados.

El objetivo a corto plazo que perseguíamos, al estudiar el caso que a continuación presentamos, era dotar a algunas áreas de actividad de la Empresa de un sistema de información completamentaria de las aplicaciones ya desarrolladas y que, en su caso, pudiera llegar a convertirse en soporte de estas mismas aplicaciones y otras futuras.

Vamos a resumir pues, las fases de diseño seguidas en el desarrollo de la base de datos "orientada a la aplicación" que más adelante podrá convertirse en un sistema de información más ambicioso.

No vamos a prestar atención a fases previas como:

Análisis preliminar,

Identificación de objetivos,

Especificación de alternativas,

Evaluación de alternativas,

que son generalmente recomendadas (Kroenke, Martin), porque nuestro objetivo, en este trabajo, es dicutir los problemas de diseño de la B.D. que se fueron planteando. Por otra parte, y según hemos comentado anteriormente, un auténtico análisis y evaluación de alternativas no podía hacerse en aspectos tales como el tipo de software a emplear, por desear ceñirse al distribuido por UNIVAC en su serie 1100, que siguiendo el modelo DBTG de CODASYL admite estructuras arborescentes y en red simple.

4. BASE DE DATOS PARA LA GESTION DE PERSONAL

4.1. Introducción

Como complemento a la aplicación de Nómina se desarrollaron, en su día, una serie de programas destinados a facilitar, al departamento correspondiente, la gestión del personal de la Compañía. Una característica común de este tipo de aplicaciones es la aparición de constantes modificaciones que se solicitan por ser incompleta o inadecuada la información que proporcionan debido, fundamentalmente, a cambios producidos en el entorno de la aplicación: nuevos pactos laborales, nuevas políticas de personal, nueva legislación, cambios de métodos y sistemas de incentivos, modificaciones de tipos de horarios, etc. A la vista de esta realidad y comprendiendo que son aportaciones fundamentales de los sistemas de bases de datos la posibilidad de consulta a la información sin necesidad de desarrollo de programas específicos; la independencia de programas y datos con lo que la información almacenada puede ir complementándose sin modificar aplicaciones; la elaboración de informes nuevos a coste reducido utilizando opciones "report" de lenguajes "query"; etc., decidimos diseñar una base de datos que soportara los programas de gestión de personal y constituyera un auténtico sistema de información para esta gestión.

A modo de resumen, para no cansar al lector, consideraremos los siguientes datos básicos:

a) *Datos de identificación:*
- Número de orden
- Documento Nacional de Identidad
- Apellidos y nombre
- Número de la Seguridad Social.

b) *Datos personales:*
- domicilio
- estado civil
- nombre esposa
- hijos
- fecha y lugar de nacimiento
- profesionales.

c) *Datos laborales:*
- categoría en la Empresa
- centro de trabajo
- departamento
- puesto de trabajo
- fecha de ingreso
- fecha de ascenso.

d) *Datos económicos:*
- retribuciones y sus conceptos
- descuentos
- préstamos y sus condiciones.

e) *Incidencias laborales:*
- horas trabajadas
- retrasos
- permisos
- enfermedad
- vacaciones
- horas extraordinarias.

La información que se pretenderá que el sistema proporcione puede clasificarse en: informes periódicos, informes ocasionales y consultas rápidas.

El diseño de la red de información de la base de datos deberá hacerse teniendo en cuenta principalmente las consultas rápidas, pues será en estos casos cuando más importancia tenga el tiempo de respuesta, característica del sistema dependiente, en buena parte, de la adecuación de la red al tipo de consulta.

Por otra parte, la información necesaria tanto en las consultas rápidas como en los informes ocasionales, podrá, por su propia naturaleza, estimarse únicamente a priori, por lo que cabe esperar modificar el esquema una vez se haya probado.

Entre las consultas previsibles podrían citarse, a modo de ejemplo, las siguientes:

- relaciones de individuos por departamentos, cumpliendo determinadas condiciones económicas, laborales, familiares, etc.

- relaciones por condiciones laborales, económicas, familiares y sus combinaciones.

– totalizaciones de conceptos económicos, laborales y comparaciones entre colectivos.

4.2. Diseño de la red de información

A la vista de las ideas expuestas en el apartado precedente vamos a ir dando forma, en sucesivas aproximaciones, al esquema de la red de información soportada por la base de datos. El objetivo de esta sección, como hemos dicho anteriormente, es mostrar al lector sin experiencia cómo se puede ir elaborando un esquema y las consideraciones básicas a tener en cuenta. Esta es probablemente la parte menos tratada en la bibliografía y, paradójicamente, la más necesaria cuando se pretende diseñar la primera base de datos.

Una estructura natural de la información citada podría ser:

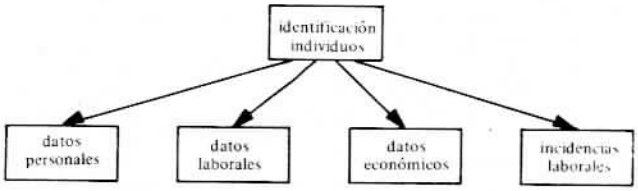


FIGURA 1

Los rectángulos indican, como es habitual, *registros de información*, y las flechas representan ocurrencias conjuntas o relacionadas de registros, constituyendo los *conjuntos (sets)* en la terminología de base de datos CODASYL. Cada ocurrencia de un *set* tendrá un *registro propietario (owner)* y uno o más *registros miembros*, constituyendo la ocurrencia conjunta citada. Las flechas indican pues esta relación de dependencia o paternidad.

Una limitación natural de este tipo de estructuras para poder ser introducidas "sin problemas" en una base de datos es que un registro miembro debe tener un único "propietario" en cada set. Es decir, sólo son válidas las relaciones "uno a varios" (de un padre a varios hijos). Debe hacerse notar que, de precisarse relaciones de varios a varios, se pueden introducir mediante la utilización del *conjunto producto*.

Si se adoptara la red de información (esquema) de la fig. 1, aparecerían los siguientes problemas básicos:

– La descripción (textos) de la mayor parte de los campos tendría que duplicarse para cada *Individuo*. Por ejemplo *Departamentos, Puestos de trabajo*, etc., con lo que el uso de esta base de datos implicaría una repetición intolerable de información. Evidentemente si se recurriera a su codificación y uso de las tablas correspondientes, se perderían muchas de las ventajas del concepto de B.D. En definitiva se estaría hablando de la distribución convencional de la información en un fichero secuencial indexado, por ejemplo, y las correspondientes tablas.

– No sería posible el acceso inmediato a los *Individuos* de un determinado *Departamento, Puesto de Trabajo*, etc., al no existir ninguna preclasificación. Es decir, no aparece en la red la dependencia de pertenencia de *Individuos* a un *Departamento* determinado, mediante el correspondiente *set*, con propietario el *Departamento* y miembros los *Individuos* que pertenezcan a él, por ejemplo.

Parece pues evidente que para sacar provecho a estas dependencias de pertenencia hay que reestructurar la información de otra forma, menos evidente a primera vista.

El próximo diseño se elaboró procurando empezar la red por los registros de mayor nivel de paternidad (bisabuelos antes de abuelos, éstos antes que padres, etc.). De esta forma, una vez localizado un propietario se conocerán automáticamente sus miembros como pertenecientes al correspondiente set.

Un próximo esquema podría ser pues el de la fig. 2.

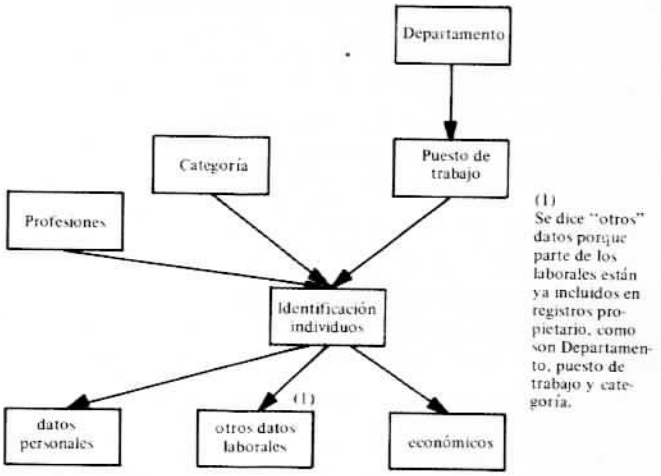


FIGURA 2

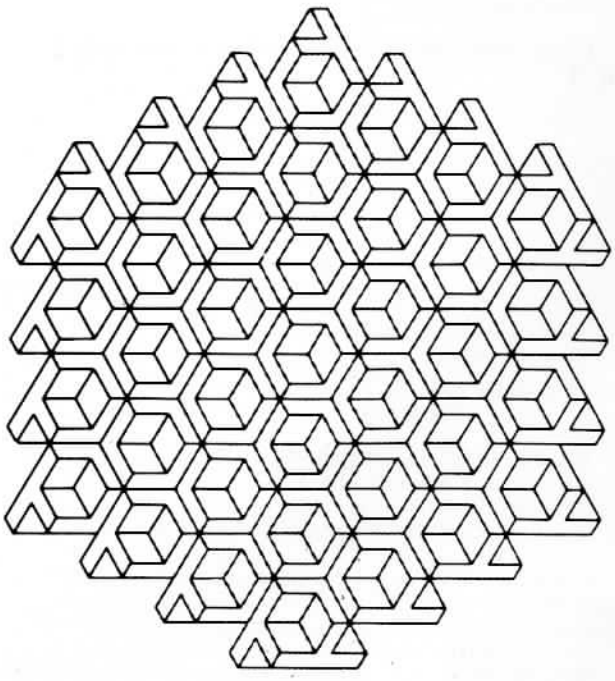
El esquema de la fig. 2 presenta los siguientes inconvenientes o limitaciones:

– No aparece reflejada la relación definida en la Empresa entre *Categorías y Puestos de Trabajo* al haberse establecido para estos últimos qué *Categorías* les corresponden.

– Para acceder a los empleados de un *Departamento* hay que hacerlo vía *Puesto de Trabajo*. Dado que se estima frecuente este tipo de consultas, debería considerarse la posibilidad de definir el correspondiente set *Departamento/Empleado*.

– A efectos de cálculo de nómina es conveniente tener a los individuos agrupados por niveles salariales, pues al existir las ramas administrativa y técnica, dos categorías distintas pueden tener el mismo nivel.

– La relación *Profesión/Individuo* no es "una a varios" a menos que se defina una profesión principal y otras complementarias para quienes tengan varias titulaciones. A continuación analizaremos las posibilidades de resolución de los problemas que esto plantea.



– La relación *Departamento/Puesto de trabajo* es de uno a varios si se consideran en un mismo puesto los que estando en departamentos distintos tienen la misma denominación.

Para resolver el problema planteado por la relación varios/varios entre *Profesión* y *Empleado* (un empleado puede tener varias profesiones y una misma profesión puede ser la de varios empleados) se estudiaron las dos soluciones que a continuación se exponen:

1. Utilizar los elementos del conjunto producto *Empleados-Profesiones* para representar las profesiones de cada individuo.

2. Definir los registros de *Profesión* como miembros de los registros *Empleados*.

La solución 1, frecuentemente sugerida en la literatura, daría lugar a un esquema como el de la fig. 3, que debería ser incluido en el esquema general de la fig. 2.

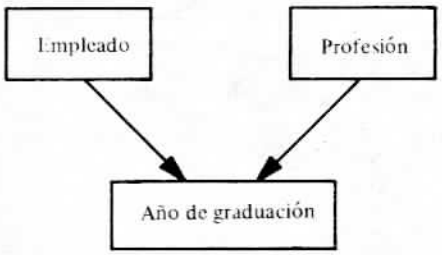


FIGURA 3

Como es sabido, esta solución es interesante cuando para cada elemento del conjunto producto desea guardarse información de interés, como sería el año de graduación en la fig. 3. En nuestro caso sólo deseamos, en realidad, definir las combinaciones existentes, por lo que esta solución resulta poco interesante, especialmente si se tiene en cuenta que cuando deseemos conocer las profesiones de un cierto empleado tendremos que localizar cada uno de los elementos del conjunto producto (años de graduación en la fig. 3) y a partir de ellos, la profesión a que hacen referencia (el padre o propietario de este registro del conjunto producto).

Más adelante haremos alguna consideración análoga al discutir la conveniencia de introducir información duplicada para evitar consultas indirectas como la que acabamos de explicar.

La opción 2 daría lugar a un esquema como el de la fig. 4.

FIGURA 4

Hemos dicho anteriormente que esta relación, no es uno a varios. A pesar de ello, esta estructura puede perfectamente mantenerse en el esquema si se cargan y consultan los datos de la base de forma adecuada.

Cada *Empleado* estará de "alguna forma" encadenado a sus *Profesiones*, pero éstas estarán repetidas tantas veces como correspondan a empleados diferentes.

De esta manera hay una abundante duplicidad de información, pero se facilita su consulta. Estas duplicidades, además no dificultan el uso de la base, al no tener necesidad de definir registros miembros de los de profesiones.

Habrá observado el lector que este esquema originará un almacenamiento de información análoga a un registro de empleado de longitud variable, en el que existiera un campo profesión para cada una de las que un empleado posea. Este tipo de registros, siempre engorrosos en sistemas convencionales, pueden ser pues emulados fácilmente con una base de datos.

En nuestro caso decidimos adoptar la solución de la fig. 4 porque *consideramos que la rapidez de consulta compensaría la pérdida de espacio en memoria lenta como consecuencia de las duplicidades introducidas.*

En cuanto al problema presentado por la relación varios/varios entre *Departamentos* y *Puestos de Trabajo*, decidimos considerar puestos distintos los pertenecientes a departamentos diferentes, dado que, en muchos casos, la coincidencia de nombres no significa igualdad de contenido, con lo que la relación se convierte de hecho en una de uno *(Departamento)* a varios *(Puesto de Trabajo)*.

De las consideraciones precedentes resulta pues el esquema de la fig. 5.

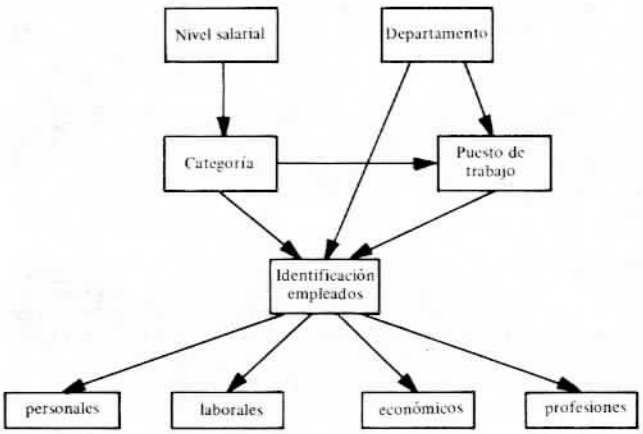


FIGURA 5

4.3. Refinamiento de la red de información

Hasta este momento se ha intentado definir la red considerando principalmente las dependencias lógicas existentes entre la información que se desea introducir en la base de datos. De todas maneras, este esquema presenta todavía serias limitaciones que serán más evidentes cuanto más experiencia se tenga en este tipo de tareas y mejor se conozcan los objetivos que se desean cubrir con la base en diseño.

Hemos dicho anteriormente que una de las utilizaciones previstas de la base de datos es como soporte de consultas rápidas para que constituya un sistema de información eficaz.

El lenguaje tipo **"query"**, *propio del software de base de datos que empleamos, está diseñado para soportar las preguntas cortas deseadas.* Para ello, en el sistema que utilizamos, hay que elegir para cada consulta un "camino" único entre los previamente definidos en el esquema, que debe tener la raíz en alguno de los registros y terminar en el último de los deseados, enlazando a ambos mediante una sucesión de *sets* (arcos de los esquemas) y *registros* (rectángulos en los esquemas). Un *set* puede ser recorrido, en un camino, en sentido inverso al de su definición, pero aparecen dificultades (o imposibilidad según los casos) para utilizar un mismo *registro* dos veces en un mismo camino.

La utilización de los *sets* en sentido inverso al indicado en el esquema en un camino de consulta, introducirá tiempos de respuesta muy elevados si cada uno de los *registros* miembros del *set* no está encadenado directamente al propietario. La introducción de este encadenamiento complementario significa, como es obvio, una ocupación adicional de espacio, proporcional al número de registros miembros.

Veamos, a continuación, los problemas que el esquema de la fig. 5 presenta para su utilización, como consecuencia de lo que acabamos de comentar. Finalizaremos esta sección proponiendo un nuevo esquema que permite un uso más eficiente de la base gracias, en parte, a simples modificaciones y, en parte, a la introducción de nuevas duplicidades de información.

Los registros de datos personales, laborales y económicos serán, en general, únicos para cada empleado; es decir, su relación con los registros empleado es más bien de uno a uno que de uno a varios. Por ello la información que contienen puede perfectamente ser incluida en los registros de empleado sin que ello exija repetir información. El mantenerlos según se indica en la fig. 5 podría justificarse por la ubicación física de la información en los discos. Debe recordarse que la definición de áreas y la asignación de registros a ellas (diseño físico) es una de las tareas importantes en la definición de una base y que no trataremos en este trabajo. Sin embargo, independientemente de las ventajas que puedan obtenerse en la ubicación física y correspondiente acceso, el esquema de la fig. 5, en cuanto a los registros que nos ocupan, presenta la dificultad de acceso simultáneo; es decir, si se define un camino que llegue hasta los datos económicos no podrán obtenerse por él los datos laborales y personales simultáneamente. Para evitar este inconveniente de obtener la información de los empleados "por partes" es pues aconsejable incluir todos sus datos en el registro empleado, recordando lo dicho anteriormente acerca de la dependencia uno a uno.

En relación con los caminos de consulta, esta red presenta otras dificultades. Si se desea obtener los empleados de un *Departamento* determinado que tengan un cierto nivel salarial, es preciso definir un camino que "retroceda" por el set *Departamento-Empleado*. Recordemos que el sistema *"query"* exige la utilización de un único camino en cada consulta. Será pues preciso "encadenar al propietario" todos los registros en los sets "por encima" de empleados (*Nivel Salarial/Categoría, Categoría/Empleado*, etc.) si se desea poder realizar consultas eficientes con el sistema *"querry"*. Con el fin de agilizar las consultas rápidas puede ser interesante repetir información incluyendo, por ejemplo, los datos de *Departamento, Puesto de Trabajo* y *Categoría* en el registro empleado. Esta práctica puede parecer poco adecuada, pero en el caso que nos ocupa resultó de indiscutible utilidad. Por una parte proporcionó, como puede comprenderse, rapidez y facilidad de consulta; no significó además un volumen complementario de información importante, ni medida en porcentaje del total de la base (menos de un 2,5 %) ni un porcentaje de la disponibilidad de disco "on line" para las aplicaciones de gestión (menos de 1 °/oo); finalmente, la duplicidad de información no introduce graves riesgos de obsolescencia o actualización deficiente porque en la carga de la base se pueden, de forma muy natural, completar los registros de *Empleado* a partir de los de *Departamento, Puesto de Trabajo* y *Categoría* que se hayan definido. Téngase presente que para grabar un registro empleado es preciso tener "activos" sus padres por lo que resulta natural leer de ellos los datos que se desean duplicar.

El esquema que resulta de las consideraciones precedentes es el de la fig. 6.

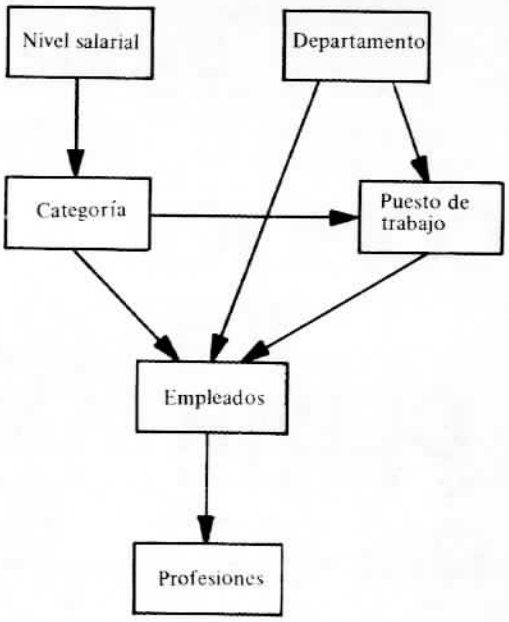


FIGURA 6

5. VERSION ACTUAL DE LA BASE. CONSIDERACIONES FINALES

La base de personal actualmente en explotación tiene algunas diferencias y complementos respecto a la descrita, debido a necesidades complementarias no citadas en este trabajo y a características especiales de utilización propias de la Compañía. Es importante señalar, sin embargo, que las modificaciones y complementos se han introducido apoyándonos en los conocimientos teóricos, siempre necesarios, de bases de datos y las ideas que en la exposición que precede hemos intentado presentar.

El problema del diseño lógico de la red es hoy por hoy, como decíamos al principio, más un arte que una técnica. Tenemos la esperanza que estas notas sean de utilidad para quienes nunca hayan creado un esquema propio. Cuanto hemos dicho se ajusta bastante a cómo unos profesionales, con los necesarios conocimientos de bases y sin experiencia, fueron descubriendo, de la mano de un experto, cómo abordar el diseño.

José M.ª Giró
Rafael Andreu

REFERENCIAS

KROENKE, David; "Database Processing" SRA, 1977.
LYON, John K; "The Database Administrator" John Wiley, 1976.
MARTIN, James; "Organización de las Bases de Datos" PHI, 1977.
CHEN, P.P.S.; "The entity-relationship model – Toward a unified view of data" TODS, vol. 1, n.º 1, AC, Marzo 1976.
CHEN, P.P.S.; "The entity-relationship model – A basis for the enterprise view of data". Proceedings of the AFIPS Conference, 46, AFIPS, Montvale, N.J., 1977, págs. 77-84.
DATE, C.J.; "An introduction to database systems", Addison-Wesley, 1975.
FRY, J.P. y TEOREY, T.J.: "Design and perfomance tools for improving database usability and responsiveness" Academic Press, 1978.