

El modelo entidad relación

Maria-Amparo Vila

Departamento de Ciencias de la Computación e Inteligencia Artificial

E.T.S. de Ingeniería Informática

Universidad de Granada

Octubre 2000

1

Los Modelos de Datos Semánticos

1.1 Introducción

Como ya se ha puesto de manifiesto en el tema anterior, la pasada década ha visto un rápido crecimiento de los sistemas de bases de datos, no sólo con respecto al número de tales sistemas sino también con respecto a la amplitud de la información almacenada y a la complejidad de las aplicaciones desarrolladas. La complejidad de las aplicaciones ha creado la necesidad de disponer de conceptos de alto nivel, herramientas y técnicas para el diseño y desarrollo de bases de datos. El problema del diseño de una base de datos es esencialmente un proceso en dos etapas que supone:

1. Una fase de diseño conceptual donde se obtiene un primer esquema conceptual que es una abstracción de la parcela de mundo real a modelar.
2. El diseño de una estructura lógica que represente el esquema y que sea implementable en un sistema de gestión de bases de datos.

Las metodologías de diseño han estado fuertemente influenciadas por los avances en lenguajes de programación. Estos han desarrollado sofisticados mecanismos de abstracción que permiten, por ejemplo la especificación de tipos de datos y objetos independientemente de su implementación. Esta idea trasladada al contexto de bases de datos ha dado lugar al concepto de *modelo de dato*. Un modelo de dato puede definirse como un patron de acuerdo con el cual los datos se organizan lógicamente. Un modelo consta siempre de lo que se llama unidades lógicas de datos y de algún mecanismo que permita expresar relaciones entre ellas. El modelo de datos debe permitir capturar tanto el aspecto estático como

el aspecto dinámico de una aplicación. Las propiedades estáticas de un sistema son aquellas que se refieren a la abstracción de los elementos del mismo, sus propiedades (atributos) y las relaciones entre ellos. Estas propiedades se describen en un esquema utilizando alguna forma de lenguaje de descripción de datos. Los aspectos dinámicos del sistema se refieren a las operaciones que se realizan sobre los elementos del mismo y las propiedades requeridas por transacciones y consultas, es decir las propiedades dinámicas modelan la conducta del sistema y toman la forma de especificaciones para las operaciones a realizar en la base de datos.

En el contexto de la teoría de bases de datos han aparecido numerosos modelos de datos. La mayoría de ellos son modelos experimentales que no han sido realmente utilizados ni en diseño, ni en implementación de bases de datos, no obstante existen algunos muy famosos y conocidos:

- Entre los modelos que han servido de base para sistemas de gestión de bases de datos, es decir que han sido implementados se encuentran los primeros basados en grafos (jerárquicos y en red), el modelo relacional y el orientado a objetos. Si bien este último está aún lejos de estar definitivamente cerrado.
- Existe otro grupo de modelos que no han estado inicialmente orientados a la implementación sino a la primera fase del diseño. Obviamente en ellos el aspecto dinámico tiene menos importancia frente al intento de capturar una mayor parte de realidad. Estos son los modelos semánticos cuyas ideas han inspirado la mayor parte de sistemas orientados a objetos. Entre los modelos semánticos el más conocido es el modelo E/R debido inicialmente a Chen que posteriormente ha sido enriquecido hasta configurar lo que se ha dado en llamar el modelo entidad/relación extendido EER. Este modelo ha atraído el interés en el entorno de las aplicaciones comerciales y juega un papel fundamental en muchas herramientas CASE (Computer Aided Software Engineering)

1.1.1 Nota histórica

Como es bien conocido, los sistemas de bases de datos comerciales que estuvieron disponibles durante casi dos décadas fueron de tipo jerárquico o basado en red. A principio de los 70 aparecen dos líneas de investigación opuestas acerca de los modelos de bases de datos, concretamente la del modelo relacional y las de las representaciones semánticas. El desarrollo del modelo relacional es bien conocido, así como su gran importancia tanto desde el punto de vista teórico como práctico, por lo que no vamos a entrar en ello.

La historia de la investigación en modelos los semánticos, es algo diferente. Los modelos semánticos fueron desarrollados inicialmente para servir de herramientas de diseño. La idea básica es que un esquema podía escribirse primero usando un modelo semántico y luego trasladarlo a un modelo tradicional para su implementación final. Por ello, el énfasis inicial de los mismos estuvo en una modelización lo más cuidadosa posible de todas las relaciones entre datos que habitualmente se presentan en una aplicación típica de bases de datos. Consecuentemente estos modelos son siempre más complejos que el modelo relacional y además se basan en una visión de los datos mucho más estructurada. Actualmente el campo de los modelos semánticos está evolucionando rápidamente, ya no se conciben sólo como herramienta de diseño sino que se empiezan a ver como modelos básicos para sistemas de gestión de bases de datos o al menos como los modelos en los que se desarrollen las interfaces de usuario , dejando los modelos implementables clásicos a un nivel más interno.

El primer modelo de datos semántico se publicó en 1974. El área de trabajo creció rápidamente durante la década siguiente gracias al trabajo de prominentes investigadores tales como: Chen, Hammer y McLeod, Kent, y Smith entre otros. El resultado central de esta investigación ha sido de mecanismos de representación muy potentes sobre todo desde el punto de vista estructural. Ultimamente los objetivos de investigación han cambiado, se trata ahora de utilizar los modelos semánticos de una forma más directa, es por ello que los últimos trabajos acerca de estos modelos estudian problemas referentes a manipulación de datos y no a descripción, aparecen entonces lenguajes de consulta para modelos semánticos, o estudios de los aspectos dinámicos de los mismos tales como mantenimiento de integridad frente actualizaciones. En este tema nos vamos a centrar en los aspectos estructurales de los modelos dado que los aspectos dinámicos y de consulta se encuentran en la actualidad menos elaborados y además son muy dependientes de cada caso.

El objetivo del presente tema es el estudio de los modelos de datos semánticos más conocidos. Comenzaremos con un repaso al modelo EER como el más utilizado y emblemático dentro de este grupo. Continuaremos con un breve análisis de alternativas al modelo EER que aparecieron en la misma época como son el modelo RM/T y el modelo de datos funcional para terminar con el estudio genérico de modelos semánticos más avanzados.

1.2 El modelo Entidad/Relación Extendido

1.2.1 Elementos Básicos

En el modelo EER la información se representa por medio de tres conceptos primitivos

1. *Entidades*, que representan los items u objetos a ser modelados.
2. *Atributos* que representan las propiedades de dichos items.
3. *Relaciones* que representan las asociaciones entre items.

Cada uno de estos conceptos se describe en detalle en las siguientes secciones.

1.2.2 Entidades y atributos

Desde un punto de vista filosófico el concepto de entidad es complejo de definir, en el contexto de bases de datos se define entidad como algo acerca de lo cual se puede almacenar información descriptiva que es capaz de existencia independiente y que puede ser unívocamente identificado. Una entidad puede ser un objeto tal como una casa, un empleado o un coche, o un suceso o actividad tal como un partido de futbol, un viaje etc...

La descripción de una entidad se hace siempre por medio de sus atributos, por ejemplo una casa puede describirse por los atributos de dirección, estilo, color, tamaño etc... Si un atributo contiene a su vez información descriptiva puede ser a su vez caracterizado como una entidad, este sería el caso de un coche, uno de cuyos atributos es el modelo, el cual, si ello se considera oportuno, podría verse como una entidad que estaría relacionada con coche.

El nombre de una entidad junto con sus atributos define un *tipo de entidad* del cual puede haber diferentes *instancias*. La distinción entre tipos de entidades y sus instancias es equivalente a la existente entre tipos de datos y sus instancias en lenguajes de programación. Un atributo o conjunto de atributos que identifican unívocamente cada instancia de un tipo de entidad se denomina *Llave candidata*. Normalmente se elige una *Llave primaria* entre las llaves candidatas del tipo de entidad para identificar unívocamente cada instancia cuando se transforme en esquema de acuerdo con otros modelos de datos.

1.2.3 Relaciones

Se denomina relación (en el contexto de este modelo) a una asociación entre dos o más tipos de entidades, por ejemplo la relación CIUDADANO DE entre las

entidades PERSONA y PAIS. La *funcionalidad* de una relación puede ser uno-a-uno (1:1), uno-a-muchos (1:N), o muchos-a-muchos (N:M). Ejemplos de estos tipos de relaciones son las de DIRECCION entre DEPARTAMENTO y EMPLEADO, TRABAJA-EN entre estos mismo tipos de entidades y VENTAS entre TIENDAS y PRODUCTOS repectivamente.

Algunas relaciones pueden tener propiedades que son, a su vez, atributos de las relaciones, por ejemplo en una empresa, la relación ASIGNADO entre los tipos de entidades EMPLEADO y PROYECTO puede tener como atributo la FECHA en que dicha asignación tuvo lugar.

Tambien es importante considerar que, con respecto a la forma en que los tipos de entidades pertenecen a una relación, existen dos clases de pertenencia:

- *Obligatoria*: aparece cuando se impone que cada instancia del tipo de entidad considerado debe participar en la relación. Ejemplo de esto sería el caso de la relación ASIGNADO entre los tipos de entidades EMPLEADO y PUESTO DE TRABAJO que puede considerarse obligatoria para el tipo EMPLEADO.
- *Opcional*: aparece cuando no se da el caso anterior.

Obviamente tanto la funcionalidad como la clase de pertenencia de una relación dependen de la semaántica del problema considerado y una misma relación puede tener características diferentes en distintos contextos. Dichas características aparecerán como reglas de integridad en el diseño final de la base de datos.

1.2.4 Representación esquemática de los modelos ER extendidos

Los modelos EER utilizan representaciones gráficas para describir la estructura de los datos. En estos diagramas los rectangulos representan los tipos de entidades, los atributos se representan por elipses y los rombos representan relaciones. La conexión tipos de entidades-relaciones se representa mediante líneas rectas. La funcionalidad de las relaciones se puede representar de distintas formas, mediante flechas dirigidas a los tipos de entidades que participen en la forma "uno" o mediante la explicitación 1:1, 1:N, o N:M.

En la figura anterior aparece la representación de un diagrama referido a cuestiones hospitalarias, donde se reflejan todos los elementos antes mencionados.

1.2.5 Relaciones más complejas

Las relaciones binarias permiten, en general, representar casi todas las posibles conexiones entre entidades. No obstante, existen algunas clases de relaciones que

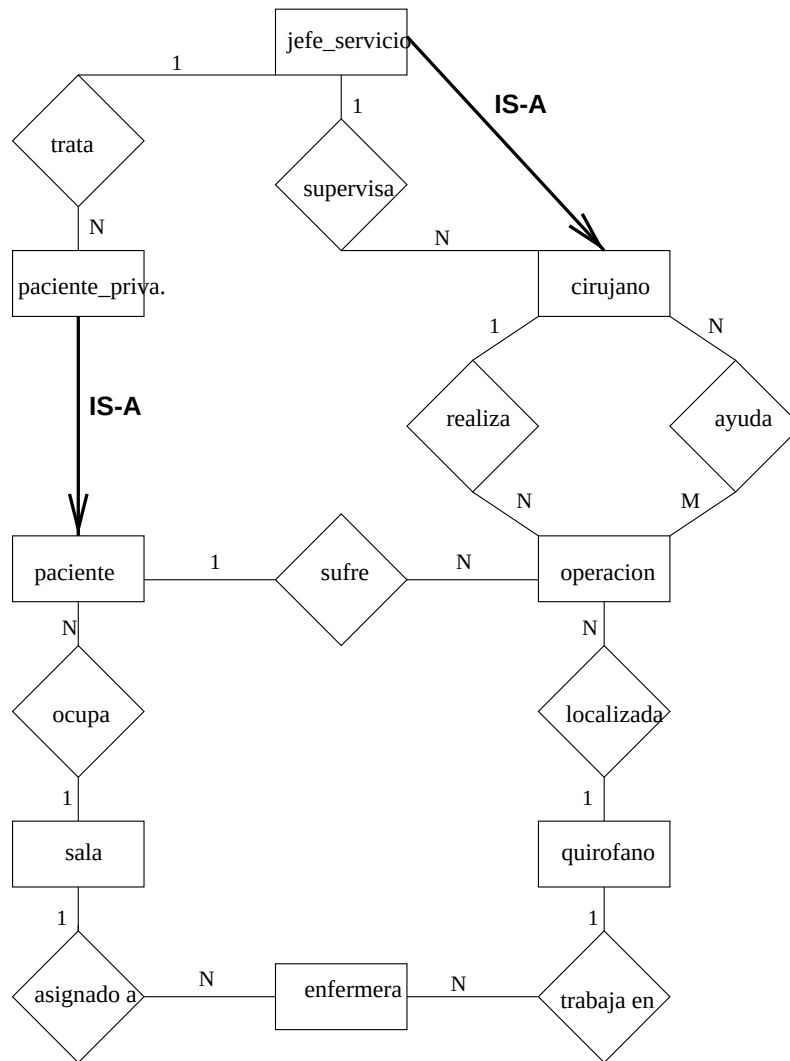


Figure 1.1: Ejemplo de esquema entidad/relación

conviene considerar con un cierto cuidado:

Relaciones involutivas

Las relaciones involutivas son aquellas que se plantean entre diferentes instancias del mismo tipo de entidad. La funcionalidad de estas relaciones puede ser, como en el caso general 1:1, 1:N y N:M. Ejemplos de estas relaciones son:

- En el caso 1:1, la relación MATRIMONIO definida en el tipo de entidad PERSONA
- En el caso 1:N la relación SUPERVISA definida en el tipo de entidad EMPLEADOS.
- En el caso de N:M la relación ES-PARTE-DE definida en el tipo de entidad PIEZAS

Las relaciones involutivas que no son 1:1 conducen frecuentemente a problemas de recuperación recursiva. En el caso de nuestros ejemplos anteriores, estos problemas aparecen cuando se quieren recuperar todos los empleados que directa o indirectamente son supervisados por otro, o cuando se quieren encontrar todas las piezas de las cuales forma parte una determinada.

Relaciones ternarias

En algunos casos las relaciones pueden involucrar a más de dos tipos de entidades. Aparecen así, relaciones ternarias, cuaternarias etc... Desde un punto de vista teórico podríamos considerar en general relaciones n-arias. No obstante, resulta poco realista analizar el caso de relaciones que no sean ternarias ya que estas aparecen muy poco en casos prácticos. De hecho muchas de las relaciones modeladas como ternarias en algunos esquemas se representan mucho más fácilmente por medio de relaciones binarias o como una nueva entidad.

Por ejemplo, si se suponen las VENTAS de determinados PRODUCTOS a distintos CENTROS por parte de diversos PROVEEDORES, nos encontramos con una relación ternaria N:M:P es decir muchos-a-muchos-a-muchos que además incluiría atributos como fecha y cantidad vendida. Ahora bien, si suponemos VENTAS como una entidad conectada mediante relaciones binarias uno-a-muchos con los otros tipos de entidad obtenemos un esquema equivalente donde la relación ternaria se ha eliminado.

En general se recomienda utilizar lo menos posible relaciones no binarias ya que, alguna posible restricción sobre la forma de interconectarse los tipos de entidades implicadas conduciría a problemas de redundancia muy fuerte a la hora

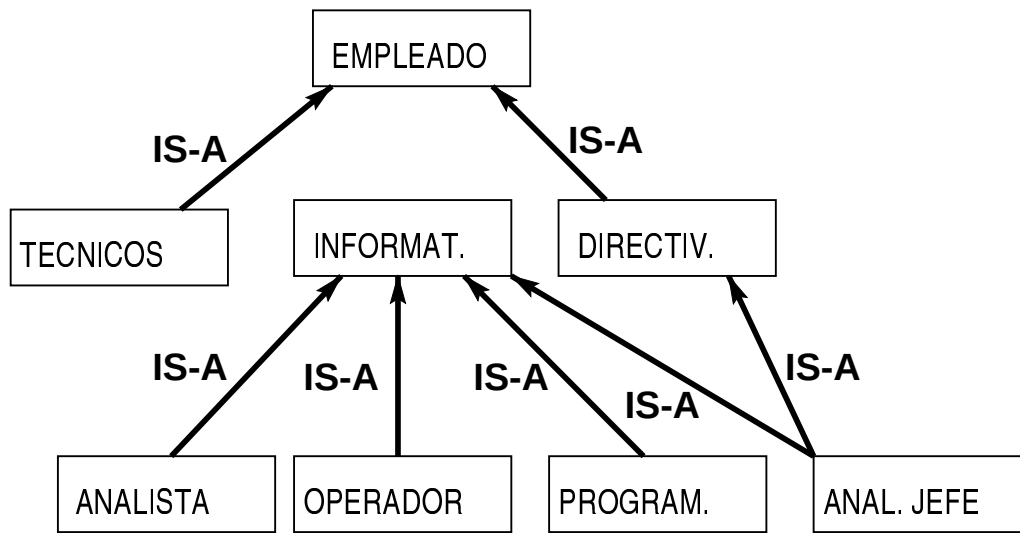


Figure 1.2: Ejemplo de jerarquía de tipos

de la implementación del esquema. Un caso de este tipo es el de la existencia de dependencias multivaluadas entre tres tipos de entidades y la implementación de la correspondiente relación ternaria en el modelo relacional.

1.2.6 Subtipos

Es obvio que toda entidad pertenece al menos a un tipo de entidad, pero una entidad puede ser de varios tipos a un mismo tiempo. Por ejemplo, si algunos empleados son programadores y todos los programadores son empleados podríamos decir que PROGRAMADOR es un *subtipo* del *supertipo* EMPLEADO. Todas las propiedades de los empleados se aplican de manera automática a los programadores pero lo contrario no se cumple. Por ejemplo los programadores podrían tener una propiedad tal como "lenguaje de programación comunmente utilizado" que no se aplica a todos los empleados. El mismo caso aparece con las relaciones, el tipo programador participa de las relaciones del tipo empleado pero lo contrario no es cierto.

Siguiendo con el ejemplo, adviértase además que algunos programadores podrían ser programadores de aplicaciones y otros programadores de sistemas y por tanto que PROGRAMADOR-DE-SISTEMAS y PROGRAMADOR-DE-APLICACIONES son a su vez subtipos del tipo PROGRAMADOR. En otras palabras un subtipo de entidad sigue siendo un tipo de entidad y por tanto tener sus propios subtipos. Un tipo de entidad y sus subtipos inmediatos y los subtipos de estos etc..., constituyen lo que se denomina una *jerarquía de tipos* del tipo de entidad en cuestión. En la figura 2.2 puede verse una jerarquía de tipos algo más amplia

Las jerarquías de tipos se conocen con varios nombres distintos entre ellos

- *Jerarquías de especialización*
- *Jerarquías de generalización*

- *Jerarquías ES-UN*

La existencia de jerarquías de tipos es una de las herramientas más importantes del modelo EER y es una de las extensiones que se le hizo al modelo ER inicial. Dichas estructuras permiten una modelización fácil y elegante de la relación de inclusión entre entidades e introducen el importante mecanismo de la herencia.

1.2.7 Transformación del modelo EER en el modelo relacional

Como hemos comentado al principio del capítulo, los esquemas de base de datos diseñados de acuerdo con el modelo EER son independientes de cualquier sistema de gestión de bases de datos. Normalmente estos sistemas poseen formalismos de representación de la información que no son tan flexibles y generales como los del modelo que nos ocupa. No obstante, la traslación de un esquema de base de datos según el EER a un esquema implementable es razonablemente directa. Obviamente, esta traslación dependerá del modelo de datos que el sistema a considerar soporte, y la riqueza semántica del esquema será tanto mayor cuanto este último modelo lo permita. En muchos casos se hace necesario transformar algún elemento del esquema con objeto de que pueda ser representado. Por ejemplo: en el caso de sistemas que se ajusten al modelo jerárquico es imposible representar directamente relaciones muchos-a-muchos y en general todos los modelos basados en grafos soportan mal las relaciones involutivas.

En nuestro caso nos vamos a centrar en la transformación con respecto al modelo relacional, el cual, a pesar de su simplicidad es capaz de capturar la mayoría de la información del modelo EER.

En lo que sigue supondremos al lector familiarizado con el modelo relacional y sus elementos más importantes, en caso contrario se recomienda la lectura de los capítulos correspondientes de (Date 1993, Korth 1994) como una buena introducción al tema.

Partimos pues del concepto de relación (en el modelo relacional) como colección de tuplas de valores de atributos modelizable en una tabla. Los datos del modelo EER pueden representarse por medio de dos clases de relaciones:

1. Un tipo de entidad se representa siempre por medio de una relación cuyo esquema coincide con el conjunto de atributos de dicho tipo y cuya llave primaria es la del tipo de entidad. Por ejemplo el tipo de entidad CLIENTE que aparece en la figura 2.1 se representaría como:

CLIENTE(NOMBRE,SEG-SOCIAL,CALLE,CIUDAD)

2. Una relación (en el sentido del modelo EER) entre tipos de entidades se representa mediante una relación (en el modelo relacional) cuyos atributos son las llaves primarias de los tipos de entidad involucrados. Por ejemplo, la relación CTACLI que aparece en la figura 2.1 se transformaría en:

CTACLI(NOMBRE,NUM-CUENTA)

La llave primaria de estas relaciones depende de la funcionalidad de las mismas. Como norma general, la llave primaria de cualquier tipo de entidad involucrado con funcionalidad de "muchos" en una relación forma parte de la llave primaria de la relación derivada, en el caso de que se trate de una relación uno-a-uno, se toma como llave primaria la de alguno de los tipos involucrados. Por ejemplo, en el caso anterior la relación es muchos-a-muchos (un cliente puede tener varias cuentas y una cuenta varios titulares) y la llave primaria está compuesta por los dos atributos de la relación. Es importante hacer notar que los atributos de una relación que surge de esta segunda son llaves exteriores con respecto a las llaves primarias de las relaciones que representan los tipos de entidad involucrados. Por ejemplo, el atributo NOMBRE en la relación CTACLI es obviamente llave exterior con respecto al atributo NOMBRE de la relación CLIENTE.

Problemas de representación del modelo relacional

Como puede verse el modelo relacional permite la representación de la mayor parte de los elementos del modelo EER, no obstante dicho modelo plantea algunos problemas que pasamos a analizar:

1. Si bien es posible representar las relaciones uno-a-muchos y muchos-a-muchos sin cambiar el esquema previo, la imposición de la condición de primera forma normal (atributos atómicos) hace que esta representación haya de hacerse en forma extensiva, con la consiguiente introducción de una cierta redundancia y de falta de naturalidad en la representación.
2. La existencia de relaciones involutivas no parece plantear ningún problema, y así es a nivel de representación. Pero a nivel de manipulación nos encontramos que los mecanismos de consulta facilitados por el modelo relacional no nos permiten un proceso de consulta recursiva. Este proceso aparece de forma natural en modelos con una estructura más elaborada.
3. No hay, en principio, posibilidad de representar directamente una jerarquía de tipos. Esto solamente puede hacerse mediante el uso sistemático del concepto de llave exterior y en cualquier caso, la herencia de propiedades se desarrollaría sólo a nivel de consulta. Esto sí es un grave inconveniente

ya que supone la utilización sistemática de la integridad referencial que no está muy elaborada a nivel de implementación (casi ningún sistema la incorpora).

Estos problemas en muchos casos pueden no ser importantes, pero en otros donde la estructuración es esencial, puede ser muy graves. El modelo de datos orientado a objetos nos permitirá resolver estos inconvenientes de forma fácil y elegante.

Bibliography

- [1] Date C.J.(1998) *Introduccion a los Sistemas de Bases de Datos. Vol. 1* 5. edic. Addison Wesley
- [2] Hughes J.G.(1991) *Object-Oriented Databases* Prentice Hall Int.
- [3] Khoshafian S., Abnous R. (1990) *Object Orientation: Concepts, Languages, Databases, User Interfaces* J. Wiley
- [4] Korth H.F., Silberschatz A. (1999) *Fundamentos de Bases de Datos* 3. edic. McGraw-Hill
- [5] Ullman J.D. (1988) *Database and Knowledge-Base Systems V.1 y 2* Computer Science Press.