

Diseño de bases de datos relacionales

En general, el objetivo del diseño de una base de datos relacional es la generación de un conjunto de esquemas de relación que permita almacenar la información sin redundancias innecesarias, pero que también permita recuperarla fácilmente. Esto se consigue mediante el diseño de esquemas que se hallen en la forma normal adecuada. Para determinar si el esquema de una relación se halla en una de las formas normales deseables es necesario obtener información sobre la empresa real que se está modelando con la base de datos. Parte de esa información se halla en un diagrama E-R bien diseñado, pero puede ser necesaria información adicional sobre la empresa.

7.2 Dominios atómicos y la primera forma normal

El modelo E-R permite que los conjuntos de entidades y los de relaciones tengan atributos con algún tipo de subestructura. Concretamente, permite atributos multivalorados, y atributos compuestos (como el atributo dirección con los atributos componentes calle y ciudad). Cuando se crean tablas a partir de los diseños E-R que contienen ese tipo de atributos, se elimina esa subestructura. Para los atributos compuestos, se deja que cada atributo componente sea un atributo de pleno derecho. Para los atributos multivalorados, se crea una tupla por cada elemento del conjunto multivalorado. En el modelo relacional se formaliza esta idea de que los atributos no tienen subestructuras. Un dominio es atómico si se considera que los elementos de ese dominio son unidades indivisibles.

Se dice que el esquema de la relación **R está en la primera forma normal (1FN)** si los **dominios de todos los atributos de R son atómicos**. Los conjuntos de nombres son ejemplos de valores no atómicos. Por ejemplo, si el esquema de la relación empleado incluyera el atributo hijos, los elementos de cuyo dominio son conjuntos de nombres, el esquema no estaría en la primera forma normal. Los atributos compuestos, como el atributo dirección con los atributos componentes calle y ciudad, tienen también dominios no atómicos.

Algunos tipos de valores no atómicos pueden resultar útiles, aunque deben utilizarse con cuidado. Por ejemplo, los atributos con valores compuestos suelen resultar útiles, y los atributos con el valor determinado por un conjunto también resultan útiles en muchos casos, que es el motivo por el que el modelo E-R los soporta. En muchos dominios en los que las entidades tienen una estructura compleja, la imposición de la representación en la primera forma normal supone una carga innecesaria para el programador de las aplicaciones, que tiene que escribir código para convertir los datos a su forma atómica. También hay sobrecarga en tiempo de ejecución por la conversión de los datos entre su forma habitual y su forma atómica. Por tanto, el soporte de los valores no atómicos puede resultar muy útil en ese tipo de dominios.

7.3 Descomposición mediante dependencias funcionales

Esta metodología se basa en los conceptos de clave y de dependencia funcional.

7.3.1 Claves y dependencias funcionales

Las claves y, más en general, las dependencias funcionales son restricciones de la base de datos que exigen que las relaciones cumplan determinadas propiedades. Las relaciones que cumplen todas esas restricciones son legales.

Mientras que una clave es un conjunto de atributos que identifica de manera unívoca toda una tupla, una dependencia funcional permite expresar restricciones que identifican de manera unívoca el valor de determinados atributos.

Las dependencias funcionales permiten expresar las restricciones que no se pueden expresar con superclaves. Las dependencias funcionales se emplean de dos maneras:

1. Para probar las relaciones y ver si son legales de acuerdo con un conjunto dado de dependencias funcionales. Si una relación r es legal según el conjunto F de dependencias funcionales, se dice que r satisface F .
2. Para especificar las restricciones del conjunto de relaciones legales. Así, sólo habrá que preocuparse de las relaciones que satisfagan un conjunto dado de dependencias funcionales. Si uno desea restringirse a las relaciones del esquema R que satisfacen el conjunto F de dependencias funcionales, se dice que F se cumple en R .

Se dice que algunas dependencias funcionales son triviales porque las satisfacen todas las relaciones. Es importante darse cuenta de que una relación dada puede, en cualquier momento, satisfacer algunas dependencias funcionales cuyo cumplimiento no sea necesario en el esquema de la relación.

7.3.2 Forma normal de Boyce Codd

Una de las formas normales más deseables que se pueden obtener es la forma normal de Boyce-Codd (FNBC). Elimina todas las redundancias que se pueden descubrir a partir de las dependencias funcionales aunque, puede que queden otros tipos de redundancia. Los diseños de bases de datos están en la FNBC si cada miembro del conjunto de esquemas de relación que constituye el diseño se halla en la FNBC. Cuando se descomponen esquemas que no se hallan en la FNBC, puede que uno o varios de los esquemas resultantes no se hallen en la FNBC. En ese caso, hacen falta más descomposiciones, cuyo resultado final es un conjunto de esquemas FNBC.

7.3.3 FNBC y la conservación de las dependencias

Se han visto varias maneras de expresar las restricciones de consistencia de las bases de datos: restricciones de clave primaria, dependencias funcionales, restricciones check, asertos y disparadores. La comprobación de estas restricciones cada vez que se actualiza la base de datos puede ser costosa y, por tanto, resulta útil diseñar la base de datos de manera que las restricciones se puedan comprobar de manera eficiente. En concreto, si la comprobación de las dependencias funcionales puede realizarse considerando sólo una relación, el coste de comprobación de esa restricción será bajo. Se verá que la descomposición en la FNBC puede impedir la comprobación eficiente de determinadas dependencias funcionales.

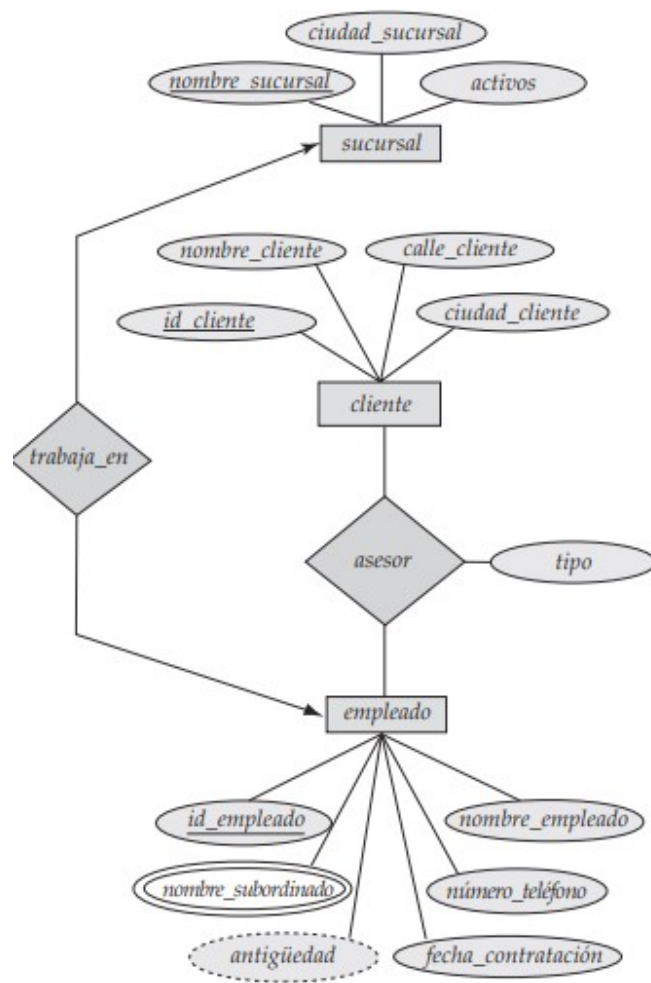
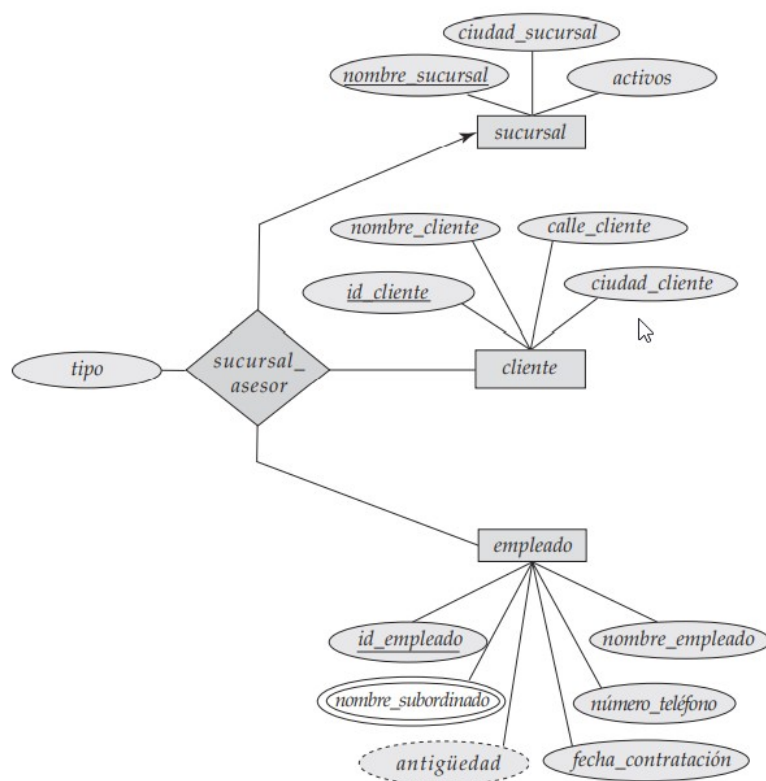


Figura 7.6 Los conjuntos de relaciones *asesor* y *trabaja_en*.



En el diseño FNBC escogido no hay ningún esquema que incluya todos los atributos que aparecen en esta dependencia funcional. Como el diseño hace computacionalmente difícil que se haga cumplir esta dependencia funcional, se dice que el diseño **no conserva las dependencias**. Como la conservación de las dependencias suele considerarse deseable, se considera otra forma normal, más débil que la FNBC, que permita que se conserven las dependencias. Esa forma normal se denomina tercera forma normal.

7.3.4 Tercera forma normal

La FNBC exige que todas las dependencias no triviales sean de la forma $\alpha \rightarrow \beta$, donde α es una superclave. La tercera forma normal (3NF) relaja ligeramente esta restricción al permitir dependencias funcionales no triviales cuya parte izquierda no sea una superclave. Antes de definir la 3NF hay que recordar que las claves candidatas son superclaves mínimas—es decir, superclaves de las que ningún subconjunto propio sea también superclave. Las dos primeras alternativas son iguales que las dos alternativas de la definición de la FNBC. La tercera alternativa de la definición de la 3NF parece bastante poco intuitiva, y no resulta evidente el motivo de su utilidad. Representa, en cierto sentido, una relajación mínima de las condiciones de la FNBC que ayuda a garantizar que cada esquema tenga una descomposición que conserve las dependencias en la 3NF. Se ha visto el equilibrio que hay que guardar entre la FNBC y la 3NF cuando no hay ningún diseño FNBC que conserve las dependencias.

7.3.5 Formas normales superiores

Puede que en algunos casos el empleo de las dependencias funcionales para la descomposición de los esquemas no sea suficiente para evitar la repetición innecesaria de información. Debido a que las formas normales basadas en las dependencias funcionales no son suficientes para tratar con situaciones como ésta, se han definido otras dependencias y formas normales.

7.4 Teoría de las dependencias funcionales

Resulta útil poder razonar de manera sistemática sobre las dependencias funcionales como parte del proceso de comprobación de los esquemas para la FNBC o la 3NF.

7.4.1 Cierre de los conjuntos de dependencias funcionales

No es suficiente considerar el conjunto dado de dependencias funcionales. También hay que considerar todas las dependencias funcionales que se cumplen. Se verá que, dado un conjunto F de dependencias funcionales, se puede probar que también se cumple alguna otra dependencia funcional. Se dice que F “implica lógicamente” esas dependencias funcionales. De manera más formal, dado un esquema relacional R , una dependencia funcional f de R está implicada lógicamente por un conjunto de dependencias funcionales F de R si cada ejemplar de la relación $r(R)$ que satisface F satisface también f .

7.4.2 Cierre de los conjuntos de atributos

Se dice que un atributo B está determinado funcionalmente por α si $\alpha \rightarrow B$. Para comprobar si un conjunto α es superclave hay que diseñar un algoritmo para el cálculo del conjunto de atributos determinados funcionalmente por α . Una manera de hacerlo es calcular F^+ , tomar todas las dependencias funcionales con α como término de la izquierda y tomar la unión de los términos de la derecha de todas esas dependencias. Sin embargo, hacer esto puede resultar costoso, ya que F^+ puede ser de gran tamaño. Un algoritmo eficiente para el cálculo del conjunto de atributos determinados funcionalmente por α no sólo resulta útil para comprobar si α es superclave, sino también para otras tareas, como se verá más adelante en este apartado.

7.4.3 Recubrimiento canónico

Supóngase que se tiene un conjunto F de dependencias funcionales de un esquema de relación. Siempre que un usuario lleve a cabo una actualización de la relación, el sistema de bases de datos debe asegurarse de que la actualización no viole ninguna dependencia funcional, es decir, que se satisfagan todas las dependencias funcionales de F en el nuevo estado de la base de datos. El sistema debe retroceder la actualización si viola alguna dependencia funcional del conjunto F . Se puede reducir el esfuerzo dedicado a la comprobación de las violaciones comprobando un conjunto simplificado de dependencias funcionales que tenga el mismo cierre que el conjunto dado. Cualquier base de datos que satisfaga el conjunto simplificado de dependencias funcionales satisfará también el conjunto original y viceversa, ya que los dos conjuntos tienen el mismo cierre.

El recubrimiento canónico F_c de F es un conjunto de dependencias tal que F implica lógicamente todas las dependencias de F_c y F_c implica lógicamente todas las dependencias de F . Dado un conjunto F de dependencias funcionales, puede suceder que toda una dependencia funcional del conjunto sea rara, en el sentido de que eliminarla no modifique el cierre de F . Se puede demostrar que el recubrimiento canónico F_c de F no contiene esa dependencia funcional rara. Supóngase que, por el contrario, esa dependencia rara estuviera en F_c . Los atributos del lado derecho de la dependencia serían raros, lo que no es posible por la definición de recubrimiento canónico.

7.4.4 Descomposición sin pérdida

Sean R un esquema de relación y F un conjunto de dependencias funcionales de R . Supóngase que R_1 y R_2 forman una descomposición de R . Sea $r(R)$ una relación con el esquema R . Se dice que la descomposición es una descomposición sin pérdidas si, para todos los ejemplares legales de la base de datos (es decir, los ejemplares de la base de datos que satisfacen las dependencias funcionales especificadas y otras restricciones). En otros términos, si se proyecta r sobre R_1 y R_2 y se calcula la reunión natural del resultado de la proyección, se vuelve a obtener exactamente r . Las descomposiciones que no son sin pérdidas se denominan descomposiciones con pérdidas. Los términos descomposición de reunión sin pérdidas y descomposición de reunión con pérdidas se utilizan a veces en lugar de descomposición sin pérdidas y descomposición con pérdidas. Se pueden utilizar las dependencias funcionales para probar si ciertas descomposiciones son sin pérdidas. Para el caso general de la descomposición simultánea de un esquema en varios, la comprobación de la descomposición sin pérdidas es más complicada. Mientras que la comprobación de la descomposición binaria es, claramente, una condición suficiente para la descomposición sin pérdidas, sólo se trata de una condición necesaria si todas las restricciones son dependencias funcionales.

7.4.5 Conservación de las dependencias

Resulta más sencillo caracterizar la conservación de las dependencias empleando la teoría de las dependencias funcionales. Sea F un conjunto de dependencias funcionales del esquema R y R_1, R_2, \dots, R_n una descomposición de R . La restricción de F a R_i es el conjunto F_i de todas las dependencias funcionales de F que sólo incluyen atributos de R_i . Dado que todas las dependencias funcionales de cada restricción implican a los atributos de un único esquema de relación, es posible comprobar el cumplimiento de esas dependencias verificando sólo una relación.

7.5 Algoritmos de descomposición

Los esquemas de bases de datos del mundo real son mucho mayores a los vistos, Por este motivo, hacen falta algoritmos para la generación de diseños que se hallen en la

forma normal adecuada. En este apartado se presentan algoritmos para la FNBC y para la 3NF.

7.5.1 Descomposición en la FNBC

Se puede emplear la definición de la FNBC para comprobar directamente si una relación se halla en esa forma normal. Sin embargo, el cálculo de F^+ puede resultar una tarea tediosa. En primer lugar se van a describir pruebas simplificadas para verificar si una relación dada se halla en la FNBC. En caso de que no lo esté, se puede descomponer para crear relaciones que sí estén en la FNBC. Más avanzado este apartado se describirá un algoritmo para crear descomposiciones sin pérdidas de las relaciones, de modo que esas descomposiciones se hallen en la FNBC.

7.5.1.2 Algoritmo de descomposición de la FNBC

Ahora se puede exponer un método general para descomponer los esquemas de relación de manera que satisfagan la FNBC. El algoritmo utiliza las dependencias que demuestran la violación de la FNBC para llevar a cabo la descomposición. La descomposición que genera este algoritmo no sólo está en la FNBC, sino que también es una descomposición sin pérdidas.

```
resultado := {R};  
hecho := falso;  
calcular  $F^+$ ;  
while (not hecho) do  
  if (hay algún esquema  $R_i$  de resultado que no se halle en la FNBC)  
    then begin  
      sea  $\alpha \rightarrow \beta$  una dependencia funcional no trivial que se cumple  
      en  $R_i$  tal que  $\alpha \rightarrow R_i$  no se halla en  $F^+$ , y  $\alpha \cap \beta = \emptyset$ ;  
      resultado := (resultado -  $R_i$ )  $\cup$  ( $R_i - \beta$ )  $\cup$  ( $\alpha, \beta$ );  
    end  
  else hecho := cierto;
```

Figura 7.12 Algoritmo de descomposición en la FNBC.

7.5.2 Descomposición en la 3FN

El algoritmo garantiza la conservación de las dependencias mediante la creación explícita de un esquema para cada dependencia del recubrimiento canónico. Asegura que la descomposición sea sin pérdidas al garantizar que, como mínimo, un esquema contenga una clave candidata del esquema que se está descomponiendo. Este algoritmo también se denomina algoritmo de síntesis de la 3NF, ya que toma un conjunto de dependencias y añade los esquemas de uno en uno, en lugar de descomponer el esquema inicial de manera repetida. El resultado no queda definido de manera única, ya que cada conjunto de dependencias funcionales puede tener más de un recubrimiento canónico y, además, en algunos casos, el resultado del algoritmo depende del orden en que considere las dependencias.

7.5.3 Comparación de la FNBC y la 3FN

De las dos formas normales para los esquemas de las bases de datos relacionales, la 3FN y la FNBC, la 3FN es más conveniente, ya que se sabe que siempre es posible obtener un diseño en la 3FN sin sacrificar la ausencia de pérdidas ni la conservación de las dependencias. Sin embargo, la 3FN presenta inconvenientes: puede que haya que

emplear valores nulos para representar algunas de las relaciones significativas posibles entre los datos y existe el problema de la repetición de la información. Los objetivos del diseño de bases de datos con dependencias funcionales son: 1. FNBC 2. Ausencia de pérdidas 3. Conservación de las dependencias Como no siempre resulta posible satisfacer las tres, puede que nos veamos obligados a escoger entre la FNBC y la conservación de las dependencias con la 3FN. Merece la pena destacar que el SQL no ofrece una manera de especificar las dependencias funcionales, salvo para el caso especial de la declaración de las superclaves mediante las restricciones primary key o unique.

Aunque puede que la comprobación de las dependencias funcionales implique una reunión si la descomposición no conserva las dependencias, se puede reducir su coste empleando vistas materializadas, que se pueden utilizar en la mayor parte de los sistemas de bases de datos. Por tanto, en caso de que no se pueda obtener una descomposición en la FNBC que conserve las dependencias, suele resultar preferible optar por la 3NF y emplear técnicas como las vistas materializadas para reducir el coste de la comprobación de las dependencias funcionales.

7.6.2 Cuarta forma normal

Un esquema de relación R está en la cuarta forma normal (4FN) con respecto a un conjunto D de dependencias funcionales y multivaloradas si, para todas las dependencias multivaloradas de D^+ de la forma $\alpha \twoheadrightarrow \beta$, donde $\alpha \subseteq R$ y $\beta \subseteq R$, se cumple, como mínimo, una de las condiciones siguientes. El diseño de una base de datos está en la 4FN si cada componente del conjunto de esquemas de relación que constituye el diseño se halla en la 4FN. Téngase en cuenta que la definición de la 4FN sólo se diferencia de la definición de la FNBC en el empleo de las dependencias multivaloradas en lugar de las dependencias funcionales. Todos los esquemas en la 4FN están en la FNBC.

7.6.3 Descomposición en la 4FN

La analogía entre la 4FN y la FNBC es aplicable al algoritmo para descomponer esquemas en la 4FN, Es idéntico al algoritmo de descomposición en la FNBC salvo en que emplea dependencias multivaloradas en lugar de funcionales y en que utiliza la restricción de D^+ .

7.7 Más formas normales

La cuarta forma normal no es, de ningún modo, la forma normal “definitiva”. Como ya se ha visto, las dependencias multivaloradas ayudan a comprender y a abordar algunas formas de repetición de la información que no pueden comprenderse en términos de las dependencias funcionales. Hay restricciones denominadas dependencias de reunión que generalizan las dependencias multivaloradas y llevan a otra forma normal denominada forma normal de reunión por proyección (FNRP) (la FNRP se denomina en algunos libros quinta forma normal). Hay una clase de restricciones todavía más generales, que lleva a una forma normal denominada forma normal de dominios y claves (FNDC). Un problema práctico del empleo de estas restricciones generalizadas es que no sólo es difícil razonar con ellas, sino que tampoco hay un conjunto de reglas de inferencia seguras y completas para razonar sobre las restricciones. Por tanto, FNRP y FNDC se utilizan muy rara vez. El Apéndice C ofrece más detalles sobre estas formas normales. Destaca por su ausencia en este estudio de las formas normales la segunda forma normal (2FN). No se ha estudiado porque sólo es de interés histórico.

7.8.1 El modelo ER y la normalización

Cuando se definen con cuidado los diagramas E-R, identificando correctamente todas las entidades, los esquemas de relación generados a partir de ellos no deben necesitar mucha más normalización. No obstante, puede haber dependencias funcionales entre los atributos de alguna entidad.

De manera parecida, puede que una relación que implique a más de dos entidades no se halle en la forma normal deseable. Como la mayor parte de las relaciones son binarias, estos casos resultan relativamente raros (de hecho, algunas variantes de los diagramas E-R hacen realmente difícil o imposible especificar relaciones no binarias). Las dependencias funcionales pueden ayudar a detectar un mal diseño E-R. Si las relaciones generadas no se hallan en la forma normal deseada, el problema puede solucionarse en el diagrama E-R. Es decir, la normalización puede llevarse a cabo formalmente como parte del modelado de los datos. De manera alternativa, la normalización puede dejarse a la intuición del diseñador durante el modelado E-R, y puede hacerse formalmente sobre las relaciones generadas a partir del modelo E-R. El lector atento se habrá dado cuenta de que para que se pudiera ilustrar la necesidad de las dependencias multivaloradas y de la cuarta forma normal hubo que comenzar con esquemas que no se obtuvieron a partir del diseño E-R. En realidad, el proceso de creación de diseños E-R tiende a generar diseños 4FN. Si se cumple alguna dependencia multivalorada y no la implica la dependencia funcional correspondiente, suele proceder de alguna de las fuentes siguientes:

- Una relación de varios a varios.
- Un atributo multivalorado de un conjunto de entidades.

En las relaciones de varios a varios cada conjunto de entidades relacionado tiene su propio esquema y hay un esquema adicional para el conjunto de relaciones. Para los atributos multivalorados se crea un esquema diferente que consta de ese atributo y de la clave primaria del conjunto de entidades.

7.8.2 Denominación de los atributos y de las relaciones

Una característica deseable del diseño de bases de datos es la asunción de un rol único, lo que significa que cada nombre de atributo tiene un significado único en toda la base de datos. Esto evita que se utilice el mismo atributo para indicar cosas diferentes en esquemas diferentes. Aunque, técnicamente, el orden de los nombres de los atributos en los esquemas no tiene ninguna importancia, es costumbre relacionar en primer lugar los atributos de la clave primaria. Esto facilita la lectura de los resultados predeterminados (como los generados por select *). En los esquemas de bases de datos de gran tamaño los conjuntos de relaciones (y los esquemas derivados) se suelen denominar mediante la concatenación de los nombres de los conjuntos de entidades a los que hacen referencia, quizás con guiones o caracteres de subrayado intercalados.

7.8.3 Desnormalización para el rendimiento

A veces, los diseñadores de bases de datos escogen un esquema que tiene información redundante; es decir, que no está normalizado. Utilizan la redundancia para mejorar el rendimiento de aplicaciones concretas. La penalización sufrida por no emplear un esquema normalizado es el trabajo adicional (en términos de tiempos de codificación y de ejecución) de mantener consistentes los datos redundantes. Al igual que la desnormalización, el empleo de las vistas materializadas supone sobrecargas de espacio y de tiempo; sin embargo, presenta la ventaja de que conservar actualizadas las vistas es labor del sistema de bases de datos, no del programador de la aplicación.

7.8.4 Otros problemas del diseño

Hay algunos aspectos del diseño de bases de datos que la normalización no aborda y, por tanto, pueden llevar a un mal diseño de la base de datos. Los datos relativos al tiempo o a intervalos temporales presentan varios de esos problemas. A continuación se ofrecen

algunos ejemplos; evidentemente, conviene evitar esos diseños. Considérese una base de datos empresarial, en la que se desea almacenar los beneficios de varias compañías a lo largo de varios años.

Las representaciones como las de la relación año_empresa, con una columna para cada valor de cada atributo, se denominan de referencias cruzadas; se emplean mucho en las hojas de cálculo, en los informes y en las herramientas de análisis de datos. Aunque esas representaciones resultan útiles para mostrárselas a los usuarios, por las razones que se acaban de dar, no resultan deseables en el diseño de bases de datos. Se han propuesto extensiones del SQL para pasar los datos de la representación relacional normal a la de referencias cruzadas, para su visualización.

7.10 Resumen

- Se han mostrado algunas dificultades del diseño de bases de datos y el modo de diseñar de manera sistemática esquemas de bases de datos que eviten esas dificultades. Entre esas dificultades están la información repetida y la imposibilidad de representar cierta información.
- Se ha mostrado el desarrollo del diseño de bases de datos relacionales a partir de los diseños E-R, cuándo los esquemas se pueden combinar con seguridad y cuándo se deben descomponer. Todas las descomposiciones válidas deben ser sin pérdidas.
- Se han descrito las suposiciones de dominios atómicos y de primera forma normal.
- Se ha introducido el concepto de las dependencias funcionales y se ha utilizado para presentar dos formas normales, la forma normal de Boyce–Codd (FNBC) y la tercera forma normal (3NF).
- Si la descomposición conserva las dependencias, dada una actualización de la base de datos, todas las dependencias funcionales pueden verificarse a partir de las diferentes relaciones, sin necesidad de calcular la reunión de las relaciones de la descomposición.
- Se ha mostrado la manera de razonar con las dependencias funcionales. Se ha puesto un énfasis especial en señalar las dependencias que están implicadas lógicamente por conjuntos de dependencias. También se ha definido el concepto de recubrimiento canónico, que es un conjunto mínimo de dependencias funcionales equivalente a un conjunto dado de dependencias funcionales.
- Se ha descrito un algoritmo para la descomposición de las relaciones en la FNBC. Hay relaciones para las cuales no hay ninguna descomposición en la FNBC que conserve las dependencias.
- Se han utilizado los recubrimientos canónicos para descomponer las relaciones en la 3NF, que es una pequeña relajación de las condiciones de la FNBC. Las relaciones en la 3NF pueden tener alguna redundancia, pero siempre hay una descomposición en la 3NF que conserva las dependencias.
- Se ha presentado el concepto de dependencias multivaloradas, que especifican las restricciones que no pueden especificarse únicamente con las dependencias funcionales. Se ha definido la cuarta forma normal (4FN) con las dependencias multivaloradas. El Apartado C.1.1 del apéndice da detalles del razonamiento sobre las dependencias multivaloradas.
- Otras formas normales, como la FNRP y la FNDC, eliminan formas más sutiles de redundancia. Sin embargo, es difícil trabajar con ellas y se emplean rara vez. El Apéndice C ofrece detalles de estas formas normales.
- Al revisar los temas de este capítulo hay que tener en cuenta que el motivo de que se hayan podido definir enfoques rigurosos del diseño de bases de datos relacionales es que el modelo relacional de datos descansa sobre una base matemática sólida. Ésa es una de las principales ventajas del modelo relacional en comparación con los otros modelos de datos que se han estudiado.