

# capítulo 14

## Conceptos básicos de funcional Dependencias y normalización para bases de datos relacionales

### En los

del modelo relacional y los capítulos 5 al 8, presentamos varios aspectos con eso. Cada esquema de relación consta de una serie de atributos, y la relación El esquema de la base de datos consta de varios esquemas de relación. Hasta ahora, hemos asumido que los atributos se agrupan para formar un esquema de relación utilizando el sentido común de el diseñador de la base de datos o mapeando un diseño de esquema de base de datos desde un modelo de datos como el modelo de datos ER o ER mejorado (EER). Estos modelos hacen el diseñador identifica los tipos de entidad y los tipos de relación y sus respectivos atributos butes, lo que conduce a una agrupación natural y lógica de los atributos en relaciones cuando se siguen los procedimientos de mapeo discutidos en el Capítulo 9. Sin embargo, nos Todavía se necesita alguna forma formal de analizar por qué una agrupación de atributos en una relación El esquema de instalación puede ser mejor que otro. Al discutir el diseño de bases de datos en Capítulos 3, 4 y 9, no desarrollamos ninguna medida de idoneidad o bondad para medir la calidad del diseño, más allá de la intuición del diseñador. En esto En este capítulo discutimos algunas de las teorías que se han desarrollado con el objetivo de evaluar esquemas relacionales para la calidad del diseño, es decir, medir formalmente por qué un conjunto de agrupaciones de atributos en esquemas de relación es mejor que otro.

Hay dos niveles en los que podemos discutir la bondad de los esquemas de relación. los primero es la **lógica** (o **conceptual**) **nivel** -cómo los usuarios a interpretar los esquemas de relación y el significado de sus atributos. Tener buenos esquemas de relaciones a este nivel permite a los usuarios comprender claramente el significado de los datos en las relaciones, y

de ahí para formular correctamente sus consultas. El segundo es la **implementación** (o **nivel de almacenamiento físico**): cómo se almacenan y actualizan las tuplas en una relación base.

459

---

## Página 2

460

Capítulo 14 Conceptos básicos de las dependencias funcionales y la normalización para bases de datos relacionales

Este nivel se aplica solo a los esquemas de relaciones base, que se almacenarán físicamente como archivos, mientras que en el nivel lógico estamos interesados en esquemas tanto de relación de base como de vistas (relaciones virtuales). La teoría del diseño de bases de datos relacionales desarrollada en este capítulo se aplica principalmente a las relaciones de base, aunque algunos criterios de atención también se aplica a las vistas, como se muestra en la Sección 14.1.

Como ocurre con muchos problemas de diseño, el diseño de la base de datos se puede realizar utilizando dos enfoques: de abajo hacia arriba o de arriba hacia abajo. Una **metodología de diseño de abajo hacia arriba** (también llamada **diseño por síntesis**) considera las relaciones básicas entre atributos individuales como el punto de partida y los utiliza para construir esquemas de relación. Este enfoque no es muy popular en la práctica porque adolece del problema de tener que recoger un gran número de relaciones binarias entre atributos como punto de partida. Para practicar situaciones típicas, es casi imposible capturar las relaciones binarias entre todos los pares de atributos. En contraste, una **metodología de diseño de arriba hacia abajo** (también llamada **diseño por análisis**) comienza con una serie de agrupaciones de atributos en relaciones que existen juntos naturalmente, por ejemplo, en una factura, un formulario o un informe. Las relaciones son luego analizadas individual y colectivamente, lo que lleva a una mayor descomposición hasta que todos se cumplen las propiedades deseables. La teoría descrita en este capítulo es aplicable principalmente al enfoque de diseño de arriba hacia abajo, y como tal es más apropiado cuando formando el diseño de bases de datos por análisis y descomposición de conjuntos de atributos que aparecen juntos en archivos, informes y formularios en situaciones de la vida real.

El diseño de bases de datos relacionales finalmente produce un conjunto de relaciones. Los objetivos implícitos de la actividad de diseño son la preservación de la información y la redundancia mínima.

La información es muy difícil de cuantificar; por lo tanto, consideramos la preservación de la información en términos de mantener todos los conceptos, incluidos los tipos de atributos, tipos de entidades y tipos de relación, así como relaciones de generalización / especialización, que son descritos utilizando un modelo como el modelo EER. Por lo tanto, el diseño relacional debe preservar todos estos conceptos, que se capturaron originalmente en el concepto de diseño después del mapeo de diseño conceptual a lógico. Minimizar la redundancia implica minimizar el almacenamiento redundante de la misma información y reducir la necesidad de múltiples actualizaciones para mantener la coherencia en múltiples copias de la misma información en respuesta a eventos del mundo real que requieren una actualización.

Comenzamos este capítulo discutiendo informalmente algunos criterios para las buenas y malas relaciones. Esquemas de instalación en la Sección 14.1. En la Sección 14.2, definimos el concepto de dependencia funcional, una restricción formal entre atributos que es la principal herramienta para formalmente medir la idoneidad de las agrupaciones de atributos en esquemas de relación. En la Sección 14.3, discutimos las formas normales y el proceso de normalización utilizando funciones de dependencias funcionales. Las formas normales sucesivas se definen para satisfacer un conjunto de deseables

restricciones expresadas mediante claves primarias y dependencias funcionales. Lo normal-  
El procedimiento de ización consiste en aplicar una serie de pruebas a las relaciones para cumplir estas  
requisitos cada vez más estrictos y descomponer las relaciones cuando sea necesario. En  
En la sección 14.4, discutimos definiciones más generales de formas normales que pueden ser directamente

<sup>1</sup> Una excepción en la que se utiliza este enfoque en la práctica se basa en un modelo llamado modelo relacional binario.  
modelo. Un ejemplo es la metodología NIAM (Verheijen y VanBekum, 1982).

---

## Página 3

### 14.1 Pautas de diseño informal para esquemas de relaciones 461

aplicados a cualquier diseño dado y no requieren análisis paso a paso y normalización  
ción. Las secciones 14.5 a 14.7 discuten otras formas normales hasta la quinta forma normal.  
En la Sección 14.6 presentamos la dependencia multivalor (MVD), seguida de la  
Unir dependencia (JD) en la Sección 14.7. La sección 14.8 resume el capítulo.

El capítulo 15 continúa el desarrollo de la teoría relacionada con el diseño de bienes  
esquemas relacionales. Discutimos las propiedades deseables de la descomposición relacional:  
propiedad de unión no aditiva y propiedad de conservación de dependencia funcional. UN  
algoritmo general que prueba si una descomposición tiene o no el no aditivo  
propiedad de unión (o sin pérdidas) (también se presenta el algoritmo 15.3). Luego discutimos la propiedad  
entes de dependencias funcionales y el concepto de una cobertura mínima de dependencias  
cies. Consideramos el enfoque de abajo hacia arriba para el diseño de bases de datos que consiste en un conjunto de  
algoritmos para diseñar relaciones en una forma normal deseada. Estos algoritmos asumen  
como entrada un conjunto dado de dependencias funcionales y lograr un diseño relacional en un  
apuntar a la forma normal mientras se adhiere a las propiedades deseables anteriores. En el capítulo 15  
también definimos tipos adicionales de dependencias que mejoran aún más la evaluación  
de la bondad de los esquemas de relación.

Si el Capítulo 15 no está cubierto en un curso, recomendamos una introducción rápida al  
propiedades deseables de descomposición de la Sección 15.2, y la importancia de la  
propiedad de unión no aditiva durante la descomposición.

## 14.1 Pautas de diseño informal para esquemas de relaciones

Antes de discutir la teoría formal del diseño de bases de datos relacionales, discutimos cuatro  
directrices informales que pueden utilizarse como medidas para determinar la calidad de la relación  
diseño de esquema:

- Asegurarse de que la semántica de los atributos esté clara en el esquema
- Reducir la información redundante en tuplas
- Reducir los valores NULL en tuplas
- No permitir la posibilidad de generar tuplas falsas

Estas medidas no siempre son independientes entre sí, como veremos.

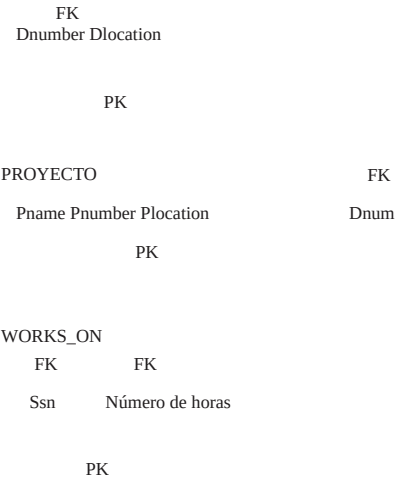
14.1.1 Impartir una semántica clara a los atributos en las relaciones

Siempre que agrupamos atributos para formar un esquema de relación, asumimos que el atributo pero los que pertenecen a una relación tienen cierto significado del mundo real y una interpretación asociada con ellos. La **semántica** de una relación se refiere a su significado, el resultado de la interpretación de valores de atributo en una tupla. En el capítulo 5, discutí cómo una relación puede interpretarse como un conjunto de hechos. Si el conceptual El diseño descrito en los Capítulos 3 y 4 se realiza con cuidado y el procedimiento de mapeo en el Capítulo 9 se sigue sistemáticamente, el diseño del esquema relacional debe tener un significado claro.

En general, es más fácil explicar la semántica de la relación, o en otras palabras, lo que significa y representa exactamente una relación: mejor es el diseño del esquema de relación estarán. Para ilustrar esto, considere la Figura 14.1, una versión simplificada de la EMPRESA esquema de base de datos relacional en la Figura 5.5 y la Figura 14.2, que presenta un ejemplo de estados de relación poblados de este esquema. El significado de la relación EMPLEADO El esquema es simple: cada tupla representa un empleado, con valores para el empleado nombre ( Ename ), número de seguro social ( Ssn ), fecha de nacimiento ( Bdate ) y dirección ( Dirección ) y el número del departamento para el que trabaja el empleado ( Dnumber ). El atributo Dnumber es una clave externa que representa una relación implícita entre EMPLEADO y DEPARTAMENTO . La semántica del DEPARTAMENTO y PROYECTO Los esquemas también son sencillos: cada tupla DEPARTAMENTO representa un departamento entidad, y cada tupla de PROYECTO representa una entidad de proyecto. El atributo Dmgr\_ssn de DEPARTAMENTO relaciona un departamento con el empleado que es su gerente, mientras que El número de PROYECTO relaciona un proyecto con su departamento de control; ambos son clave externa atributos. La facilidad con la que se puede explicar el significado de los atributos de una relación. es una medida informal de qué tan bien está diseñada la relación.

**Figura 14.1**  
Una EMPRESA relacional simplificada  
esquema de base de datos.





**Figura 14.2**  
Estado de la base de datos de muestra para el esquema de la base de datos relacional en la Figura 14.1.

EMPLEADO				
Ename	Ssn	Bdate	Habla a	Dnumber
Smith, John B.	123456789	1965-01-09	731 Fondren, Houston, TX	5
Wong, Franklin T.	333445555	1955-12-08	638 Voss, Houston, TX	5
Zelaya, Alicia J.	999887777	1968-07-19	3321 Castle, Spring, TX	4
Wallace, Jennifer S.	987654321	20/06/1941	291Berry, Bellaire, TX	4
Narayan, Ramesh K.	666884444	1962-09-15	975 Fire Oak, Humble, TX	5
Inglés, Joyce A.	453453453	1972-07-31	5631 Rice, Houston, TX	5
Jabbar, Ahmad V.	987987987	1969-03-29	980 Dallas, Houston, TX	4
Borg, James E.	888665555	10/11/1937	450 Stone, Houston, Estados Unidos	1

DEPARTAMENTO			DEPT_LOCATIONS	
Dname	Dnumber	Dmgr_ssn	Dnumber	Dlocation
Investigación	5	333445555	1	Houston
Administración	4	987654321	4	Stafford
Sede	1	888665555	5	Bellaire
			5	La tierra del azúcar
			5	Houston

WORKS_ON Ssn	Número de horas		PROYECTO Pname	Ubicación del número		Dnum
123456789	1	32,5	ProductoX	1	Bellaire	5
123456789	2	7,5	ProductoY	2	La tierra del azúcar	5
666884444	3	40,0	ProductoZ	3	Houston	5
453453453	1	20,0	Informatización	10	Stafford	4
453453453	2	20,0	Reorganización	20	Houston	1
333445555	2	10,0	Nuevosbeneficios	30	Stafford	4
333445555	3	10,0				
333445555	10	10,0				
333445555	20	10,0				
999887777	30	30,0				
999887777	10	10,0				
987987987	10	35,0				
987987987	30	5,0				
987654321	30	20,0				
987654321	20	15,0				
888665555	20	Nulo				

La semántica de los otros dos esquemas de relación en la figura 14.1 es un poco más complejo. Cada tupla en DEPT\_LOCATIONS da un número de departamento ( Dnumber ) y una de las ubicaciones del departamento ( Dlocation ). Cada tupla en WORKS\_ON le da a un empleado el número de Seguro Social ( Ssn ), el número de proyecto de uno de los proyectos en los que trabaja el empleado ( Pnumber ), y el número de horas por semana que el empleado trabaja en ese proyecto ( Horas ). Sin embargo, ambos esquemas tienen un interpretación bien definida e inequívoca. El esquema DEPT\_LOCATIONS representa residente un atributo multivalor de DEPARTMENT , mientras que WORKS\_ON representa una relación M: N entre EMPLEADO y PROYECTO . Por tanto, toda la relación Los esquemas de la figura 14.1 pueden considerarse fáciles de explicar y, por lo tanto, buenos desde el punto de vista de tener una semántica clara. Por tanto, podemos formular lo siguiente directriz de diseño informal.

Directriz 1. Diseñar un esquema de relación para que sea fácil de explicar su significado. Hacer no combinar atributos de múltiples tipos de entidades y tipos de relaciones en un solo relación gle. Intuitivamente, si un esquema de relación corresponde a un tipo de entidad o una tipo de relación, es sencillo explicar su significado. De lo contrario, si la relación ción corresponde a una mezcla de múltiples entidades y relaciones, ambi- se producirán errores y la relación no se puede explicar fácilmente.

Ejemplos de incumplimiento de la Directriz 1. Los esquemas de relación de las Figuras 14.3 (a)

y 14.3 (b) también tienen una semántica clara. (El lector debe ignorar las líneas debajo del relaciones por ahora; se utilizan para ilustrar la notación de dependencia funcional, cussed en la Sección 14.2.) Una tupla en el esquema de relación EMP\_DEPT en la Figura 14.3 (a) representa a un solo empleado pero incluye, junto con el Dnumber (el identificador para el departamento para el que trabaja), información adicional, es decir, el nombre ( Dname ) del departamento para el que trabaja el empleado y el Social Número de seguridad ( Dmgr\_ssn ) del director del departamento. Para la relación EMP\_PROJ En la figura 14.3 (b), cada tupla relaciona a un empleado con un proyecto, pero también incluye

**Figura 14.3**  
Dos esquemas de relación  
sufriendo de actualización  
anomalías.  
(a) EMP\_DEPT y  
(b) EMP\_PROJ.

(un)						
EMP_DEPT						
Ename	Ssn	Bdate	Habla a	Dnumber	Dname	Dmgr_ssn
(segundo)						
EMP_PROJ						
Ssn	Pnumber	Horas	Ename	Pname	Ubicación	
FD1						
FD2						
FD3						

el nombre del empleado ( Ename ), el nombre del proyecto ( Pname ) y la ubicación del proyecto ( Ubicación ). Aunque lógicamente no hay nada malo en estas dos relaciones, violan Directriz 1 mediante la combinación de atributos de distintas entidades del mundo real: mezclas EMP\_DEPT atributos de empleados y departamentos, y EMP\_PROJ mezcla atributos de empleados y proyectos y la relación WORKS\_ON . Por lo tanto, les va mal contra la medida anterior de calidad de diseño. Pueden usarse como vistas, pero causan problemas cuando se utilizan como relaciones base, como se analiza en la siguiente sección.

14.1.2 Información redundante en tuplas y anomalías de actualización

Un objetivo del diseño de esquemas es minimizar el espacio de almacenamiento utilizado por el ciones (y por lo tanto los archivos correspondientes). Agrupar atributos en esquemas de relaciones mas tiene un efecto significativo en el espacio de almacenamiento. Por ejemplo, compare el espacio utilizado por las dos relaciones base EMPLEADO y DEPARTAMENTO en la Figura 14.2 con que para una relación base EMP\_DEPT en la Figura 14.4, que es el resultado de aplicar el Operación NATURAL JOIN a EMPLEADO y DEPARTAMENTO . En EMP\_DEPT , el atributo

pero los valores correspondientes a un departamento en particular ( Dnumber , Dname , Dmgr\_ssn ) son repetido para cada empleado que trabaja para ese departamento. En contraste, cada departamento la información aparece sólo una vez en la relación DEPARTAMENTO en la Figura 14.2. Sólo el número de departamento ( Dnumber ) se repite en la relación EMPLOYEE para cada empleado que trabaja en ese departamento como clave externa. Comentarios similares aplicar a la relación EMP\_PROJ (ver Figura 14.4), que aumenta el WORKS\_ON relación con atributos adicionales de EMPLEADO y PROYECTO .

El almacenamiento de uniones naturales de relaciones de base conduce a un problema adicional denominado **actualizar anomalías** . Estos se pueden clasificar en anomalías de inserción, anomalías de deletación alteraciones y anomalías de modificación. 2

Anomalías de inserción. Las anomalías de inserción se pueden diferenciar en dos tipos, ilustrado por los siguientes ejemplos basados en la relación EMP\_DEPT :

- Para insertar una nueva tupla de empleados en EMP\_DEPT , debemos incluir el valores de atributo para el departamento para el que trabaja el empleado, o NULL s (si el empleado aún no trabaja para un departamento). Por ejemplo, para insertar una nueva tupla para un empleado que trabaja en el departamento número 5, debemos ingrese todos los valores de atributo del departamento 5 correctamente para que coherente con los valores correspondientes para el departamento 5 en otras tuplas en EMP\_DEPT . En el diseño de la Figura 14.2, no tenemos que preocuparnos por esto problema de consistencia porque ingresamos solo el número de departamento en el tupla de empleados; todos los demás valores de atributo del departamento 5 se registran únicamente una vez en la base de datos, como una única tupla en la relación DEPARTAMENTO .
- Es difícil insertar un nuevo departamento que todavía no tiene empleados en el Relación EMP\_DEPT . La única forma de hacer esto es colocar valores NULL en el

2 Estas anomalías fueron identificadas por Codd (1972a) para justificar la necesidad de normalización de las relaciones, como discutiremos en la Sección 15.3.

Redundancia						
EMP_DEPT						
Ename	Ssn	Bdate	Habla a	Dnumber	Dname	Dmgr_ssn
Smith, John B.	123456789	1965-01-09	731 Fondren, Houston, TX	5	Investigación	333445555
Wong, Franklin T.	333445555	1955-12-08	638 Voss, Houston, TX	5	Investigación	333445555
Zelaya, Alicia J.	999887777	1968-07-19	3321 Castle, Spring, TX	4	Administración	987654321
Wallace, Jennifer S.	987654321	20/06/1941	291 Berry, Bellaire, TX	4	Administración	987654321
Narayan, Ramesh K.	666884444	1962-09-15	975 FireOak, Humble, TX	5	Investigación	333445555
Inglés, Joyce A.	453453453	1972-07-31	5631 Rice, Houston, TX	5	Investigación	333445555
Jabbar, Ahmad V.	987987987	1969-03-29	980 Dallas, Houston, TX	4	Administración	987654321
Borg, James E.	888665555	1937-11-10	450 Stone, Houston, Estados Unidos	1	Sede	888665555



			Redundancia	Redundancia	
EMP_PROJ					
Ssn	Pnumber	Horas	Ename	Pname	Ubicación
123456789	1	32,5	Smith, John B.	ProductoX	Bellaire
123456789	2	7.5	Smith, John B.	ProductoY	La tierra del azúcar
666884444	3	40,0	Narayan, Ramesh K.	ProductoZ	Houston
453453453	1	20,0	Inglés, Joyce A.	ProductoX	Bellaire
453453453	2	20,0	Inglés, Joyce A.	ProductoY	La tierra del azúcar
333445555	2	10.0	Wong, Franklin T.	ProductoY	La tierra del azúcar
333445555	3	10.0	Wong, Franklin T.	ProductoZ	Houston
333445555	10	10.0	Wong, Franklin T.	Informatización	Stafford
333445555	20	10.0	Wong, Franklin T.	Reorganización	Houston
999887777	30	30,0	Zelaya, Alicia J.	Nuevosbeneficios	Stafford
999887777	10	10.0	Zelaya, Alicia J.	Informatización	Stafford
987987987	10	35,0	Jabbar, Ahmad V.	Informatización	Stafford
987987987	30	5,0	Jabbar, Ahmad V.	Nuevosbeneficios	Stafford
987654321	30	20,0	Wallace, Jennifer S.	Nuevosbeneficios	Stafford
987654321	20	15.0	Wallace, Jennifer S.	Reorganización	Houston
888665555	20	Nulo	Borg, James E.	Reorganización	Houston

**Figura 14.4**

Estados de muestra para EMP\_DEPT y EMP\_PROJ resultantes de aplicar NATURAL JOIN a las relaciones en la Figura 14.2. Estos pueden almacenarse como relaciones base por motivos de rendimiento.

atributos para el empleado. Esto viola la integridad de la entidad para EMP\_DEPT porque su clave principal Ssn no puede ser nula. Además, cuando el primer empleado está asignado a ese departamento, no necesitamos esta tupla con Más valores NULL . Este problema no ocurre en el diseño de la figura. 14.2 porque un departamento se ingresa en la relación DEPARTAMENTO si o ningún empleado trabaja para él, y siempre que un empleado sea asignado a ese departamento, se inserta una tupla correspondiente en EMPLEADO .

Anomalías de supresión. El problema de las anomalías por delección está relacionado con el segundo Situación de anomalía de inserción que acabamos de comentar. Si borramos de EMP\_DEPT un empleado tupla que representa al último empleado que trabaja para un departamento en particular la información relativa a ese departamento se pierde inadvertidamente del base de datos. Este problema no ocurre en la base de datos de la Figura 14.2 porque Las tuplas DEPARTAMENTO se almacenan por separado.

Anomalías de modificación. En EMP\_DEPT, si cambiamos el valor de uno de los atributos de un departamento en particular, digámos, el gerente del departamento 5, debemos actualizar las tuplas de todos los empleados que trabajan en ese departamento; de lo contrario, el la base de datos se volverá inconsistente. Si no actualizamos algunas tuplas, el mismo departamento Se mostrará que el gerente tiene dos valores diferentes para el gerente en diferentes empleados. tuplas, lo cual estaría mal. <sup>3</sup>

Es fácil ver que estas tres anomalías son indeseables y causan dificultades para mantener la coherencia de los datos y requerir actualizaciones innecesarias que pueden evitado; por tanto, podemos enunciar la siguiente directriz de la siguiente manera.

Directriz 2. Diseñe los esquemas de relación base de modo que no haya inserción, eliminación o las anomalías de modificación están presentes en las relaciones. Si hay alguna anomalía, <sup>4</sup> anótelos claramente y asegúrese de que los programas que actualizan la base de datos funcionar correctamente.

La segunda directriz es coherente con y, en cierto modo, una reafirmación de la primera guía. También podemos ver la necesidad de un enfoque más formal para evaluar si un diseño cumple con estas pautas. Las secciones 14.2 a 14.4 proporcionan estos conceptos formales necesarios. Es importante tener en cuenta que estas pautas pueden, en algunos casos, los tiempos deben violarse para mejorar el rendimiento de determinadas consultas. Si EMP\_DEPT se utiliza como una relación almacenada (también conocida como vista materializada) en Además de las relaciones básicas de EMPLEADO y DEPARTAMENTO, las anomalías en EMP\_DEPT debe anotarse y contabilizarse (por ejemplo, mediante el uso de activadores o procedimientos almacenados que realizarían actualizaciones automáticas). De esta manera, siempre que el Se actualiza la relación base, no terminamos con inconsistencias. En general, es aconsejable utilizar relaciones base libres de anomalías y especificar vistas que incluyan se una para colocar juntos los atributos a los que se hace referencia con frecuencia en consultas importantes.

### 14.1.3 Valores NULL en tuplas

En algunos diseños de esquemas podemos agrupar muchos atributos juntos en una relación "gruesa" ción. Si muchos de los atributos no se aplican a todas las tuplas de la relación, terminamos con muchos NULL en esas tuplas. Esto puede desperdiciar espacio en el nivel de almacenamiento y puede también conducen a problemas para comprender el significado de los atributos y con

<sup>3</sup> Esto no es tan grave como los otros problemas, porque todas las tuplas pueden actualizarse mediante una única consulta SQL.

<sup>4</sup> Otras consideraciones de aplicación pueden dictar y hacer inevitables ciertas anomalías. Por ejemplo, el La relación EMP\_DEPT puede corresponder a una consulta o un informe que se requiere con frecuencia.

especificando operaciones JOIN en el nivel lógico. <sup>5</sup> Otro problema con NULL s es cómo para contabilizarlos cuando se apliquen operaciones agregadas como COUNT o SUM .

Las operaciones SELECT y JOIN implican comparaciones; si hay valores NULL presentes, el

los resultados pueden volverse impredecibles. <sup>6</sup> Además, los valores NULL pueden tener múltiples interpretaciones, como las siguientes:

- El atributo no se aplica a esta tupla. Por ejemplo, `Visa_status` puede no aplicar a estudiantes estadounidenses.
- Se desconoce el valor del atributo para esta tupla. Por ejemplo, `Date_of_birth` puede ser desconocido para un empleado.
- El valor es conocido pero ausente; es decir, aún no se ha registrado. Por ejemplo, el `Home_Phone_Number` de un empleado puede existir, pero no estar disponible y grabado todavía.

Tener la misma representación para todos los NULL compromete los diferentes significados ellos quizás tengan. Por tanto, planteamos otra pauta.

Directriz 3. En la medida de lo posible, evite colocar atributos en una relación base cuyo los valores pueden ser NULL con frecuencia. Si los NULL son inevitables, asegúrese de que se apliquen solo en casos excepcionales y no se aplica a la mayoría de las tuplas en la relación.

Usar el espacio de manera eficiente y evitar uniones con valores NULL son los dos principales criterios que determinan si incluir las columnas que pueden tener NULL en un relación o tener una relación separada para esas columnas (con la clave apropiada columnas). Por ejemplo, si solo el 15% de los empleados tiene oficinas individuales, hay poca justificación para incluir un atributo `Office_number` en la relación EMPLEADO; más bien, se puede crear una relación EMP\_OFFICES (`Essn`, `Office_number`) para incluir tuplas solo para los empleados con oficinas individuales.

#### 14.1.4 Generación de tuplas espurias

Considere los dos esquemas de relación EMP\_LOCS y EMP\_PROJ1 en la Figura 14.5 (a), que se puede utilizar en lugar de la única relación EMP\_PROJ en la Figura 14.3 (b). Una tupla en EMP\_LOCS significa que el empleado cuyo nombre es `Ename` trabaja en al menos un proyecto ubicado en `Plocation`. Una tupla en EMP\_PROJ1 se refiere al hecho de que el empleado cuyo número de Seguro Social es `Ssn` trabaja las Horas indicadas por semana en el proyecto cuyo nombre, número y ubicación son `Pname`, `Pnumber` y `Plocation`. La figura 14.5 (b) muestra los estados de relación de EMP\_LOCS y EMP\_PROJ1 correspondientes a la relación EMP\_PROJ en la Figura 14.4, que se obtienen aplicando la aprobación piado PROYECTO ( $\pi$ ) operaciones a EMP\_PROJ.

<sup>5</sup> Esto se debe a que las combinaciones internas y externas producen resultados diferentes cuando los NULL están involucrados en las combinaciones. Los usuarios Por lo tanto, debe ser consciente de los diferentes significados de los distintos tipos de combinaciones. Aunque esto es razonable para usuarios sofisticados, puede ser difícil para otros.

<sup>6</sup> En la Sección 5.5.1 presentamos comparaciones que involucran valores NULOS donde el resultado (en tres valores lógica) es VERDADERO, FALSO y DESCONOCIDO.

(un)

EMP\_LOCS

Ename Ubicación

PK

EMP\_PROJ1

Ssn Pnumber Horas Pname Plocation

PK

(segundo)

EMP\_LOCS

EMP\_PROJ1

Ename	Ubicación	Ssn	Pnumber	Horas	Pname	Ubicación
Smith, John B.	Bellaire	123456789	1	32,5	ProductoX	Bellaire
Smith, John B.	La tierra del azúcar	123456789	2	7,5	ProductoY	La tierra del azúcar
Narayan, Ramesh K.	Houston	666884444	3	40,0	ProductoZ	Houston
Inglés, Joyce A.	Bellaire	453453453	1	20,0	ProductoX	Bellaire
Inglés, Joyce A.	La tierra del azúcar	453453453	2	20,0	ProductoY	La tierra del azúcar
Wong, Franklin T.	La tierra del azúcar	333445555	2	10,0	ProductoY	La tierra del azúcar
Wong, Franklin T.	Houston	333445555	3	10,0	ProductoZ	Houston
Wong, Franklin T.	Stafford	333445555	10	10,0	Informatización	Stafford
Zelaya, Alicia J.	Stafford	333445555	20	10,0	Reorganización	Houston
Jabbar, Ahmad V.	Stafford	999887777	30	30,0	Nuevosbeneficios	Stafford
Wallace, Jennifer S.	Stafford	999887777	10	10,0	Informatización	Stafford
Wallace, Jennifer S.	Houston	987987987	10	35,0	Informatización	Stafford
Borg, James E.	Houston	987987987	30	5,0	Nuevosbeneficios	Stafford
		987654321	30	20,0	Nuevosbeneficios	Stafford
		987654321	20	15,0	Reorganización	Houston
		888665555	20	NULO	Reorganización	Houston

Figura 14.5

Diseño particularmente pobre para la relación EMP\_PROJ en Figura 14.3 (b). (a) Los dos esquemas de relación EMP\_LOCS y EMP\_PROJ1. (b) El resultado de proyectar el extensión de EMP\_PROJ de la Figura 14.4 a la relaciones EMP\_LOCS y EMP\_PROJ1.

Supongamos que usamos EMP\_PROJ1 y EMP\_LOCS como relaciones base en lugar de EMP\_PROJ. Esto produce un diseño de esquema particularmente malo porque no podemos recuperar la información que estaba originalmente en EMP\_PROJ de EMP\_PROJ1 y EMP\_LOCS. Si intentamos una operación NATURAL JOIN en EMP\_PROJ1 y EMP\_LOCS, el resultado produce muchas más tuplas que el conjunto original de tuplas en EMP\_PROJ. En la figura 14.6, el resultado de aplicar la combinación solo a las tuplas para Se muestra empleado con Ssn = "123456789" (para reducir el tamaño de la relación resultante ción). Las tuplas adicionales que no estaban en EMP\_PROJ se denominan **tuplas falsas**, porque representan información falsa que no es válida. El espurio las tuplas están marcadas con asteriscos (\*) en la Figura 14.6. Se deja al lector completar el resultado de la operación NATURAL JOIN en EMP\_PROJ1 y EMP\_LOCS tablas en su totalidad y para marcar las tuplas falsas en este resultado.

Ssn	Pnumber	Horas	Pname	Ubicación	Ename
123456789	1	32,5	ProductoX	Bellaire	Smith, John B.
* 123456789	1	32,5	ProductoX	Bellaire	Inglés, Joyce A.
123456789	2	7.5	ProductoY	La tierra del azúcar	Smith, John B.
* 123456789	2	7.5	ProductoY	La tierra del azúcar	Inglés, Joyce A.
* 123456789	2	7.5	ProductoY	La tierra del azúcar	Wong, Franklin T.
666884444	3	40,0	ProductoZ	Houston	Narayan, Ramesh K.
* 666884444	3	40,0	ProductoZ	Houston	Wong, Franklin T.
* 453453453	1	20,0	ProductoX	Bellaire	Smith, John B.
453453453	1	20,0	ProductoX	Bellaire	Inglés, Joyce A.
* 453453453	2	20,0	ProductoY	La tierra del azúcar	Smith, John B.
453453453	2	20,0	ProductoY	La tierra del azúcar	Inglés, Joyce A.
* 453453453	2	20,0	ProductoY	La tierra del azúcar	Wong, Franklin T.
* 333445555	2	10.0	ProductoY	La tierra del azúcar	Smith, John B.
* 333445555	2	10.0	ProductoY	La tierra del azúcar	Inglés, Joyce A.
333445555	2	10.0	ProductoY	La tierra del azúcar	Wong, Franklin T.
* 333445555	3	10.0	ProductoZ	Houston	Narayan, Ramesh K.
333445555	3	10.0	ProductoZ	Houston	Wong, Franklin T.
333445555	10	10.0	Informatización	Stafford	Wong, Franklin T.
* 333445555	20	10.0	Reorganización	Houston	Narayan, Ramesh K.
333445555	20	10.0	Reorganización	Houston	Wong, Franklin T.

\* \* \*

**Figura 14.6**

Resultado de aplicar NATURAL JOIN a las tuplas en EMP\_PROJ1 y EMP\_LOCS de la Figura 14.5 solo para el empleado con Ssn = "123456789". Espurio generado las tuplas están marcadas con asteriscos.

Descomponer EMP\_PROJ en EMP\_LOCS y EMP\_PROJ1 no es deseable porque cuando los UNEMOS nuevamente usando NATURAL JOIN, no obtenemos el original correcto información. Esto se debe a que, en este caso, la ubicación es el atributo que relaciona EMP\_LOCS y EMP\_PROJ1, y Plocation no es una clave primaria ni una clave externa en EMP\_LOCS o EMP\_PROJ1. Ahora declaramos informalmente otra directriz de diseño.

Directriz 4. Diseñar esquemas de relación para que puedan unirse con igualdad. condiciones sobre atributos que están relacionados adecuadamente (clave primaria, clave externa) pares de una manera que garantice que no se generen tuplas falsas. Evita las relaciones que contienen atributos coincidentes que no son (clave externa, clave primaria) combinables porque la unión de tales atributos puede producir tuplas falsas.

Evidentemente, esta directriz informal debe establecerse de manera más formal. En la sección 15.2 discutimos una condición formal llamada propiedad de unión no aditiva (o sin pérdida) que garantiza que determinadas combinaciones no producen tuplas falsas.

#### 14.1.5 Resumen y discusión de las pautas de diseño

En las Secciones 14.1.1 a 14.1.4, discutimos de manera informal situaciones que conducen a esquemas de relación problemáticos y propusimos pautas informales para una buena relación de diseño. Los problemas que señalamos, que se pueden detectar sin más herramientas de análisis tradicionales, son las siguientes:

- Anomalías que hacen que se realice trabajo redundante durante la inserción y modificación de una relación, y que puede causar pérdida accidental de información durante una eliminación de una relación
- Desperdicio de espacio de almacenamiento debido a NULL y la dificultad de realizar las selecciones, operaciones de agregación y uniones debido a valores NULL
- Generación de datos no válidos y espurios durante combinaciones en relaciones base con atributos coincidentes que pueden no representar una adecuada (clave externa, primaria clave) relación

En el resto de este capítulo presentamos conceptos formales y teorías que pueden usarse para definir la bondad y la maldad de los esquemas de relaciones individuales con mayor precisión. Primero discutimos la dependencia funcional como herramienta de análisis. Luego especificamos el tres formas normales y la forma normal de Boyce-Codd (BCNF) para esquemas de relación como los estándares de calidad establecidos y aceptados en el diseño relacional. La estrategia para lograr un buen diseño es descomponer adecuadamente una relación mal diseñada para lograr formas normales más altas. También presentamos brevemente formas normales adicionales que tratan con dependencias adicionales. En el Capítulo 15, discutimos las propiedades de descomposición en detalle y proporcionar una variedad de algoritmos relacionados con dependencias funcionales, bondad de descomposición y el diseño de relaciones de abajo hacia arriba utilizando las dependencias funcionales como punto de partida.

## 14.2 Dependencias funcionales

Hasta ahora nos hemos ocupado de las medidas informales del diseño de bases de datos. Ahora introducimos una herramienta formal para el análisis de esquemas relacionales que nos permita detectar y describir algunos de los problemas mencionados anteriormente en términos precisos. El más soltero concepto importante en la teoría del diseño de esquemas relacionales es el de una dependencia funcional. En esta sección definimos formalmente el concepto, y en la Sección 14.3 vemos cómo se puede utilizar para definir formas normales de esquemas de relación.

### 14.2.1 Definición de dependencia funcional

Una dependencia funcional es una restricción entre dos conjuntos de atributos del base de datos. Suponga que nuestro esquema de base de datos relacional tiene  $n$  atributos  $A_1, A_2, \dots, A_n$ ; pensemos en toda la base de datos como si estuviera descrita por un solo **universal**

esquema de relación  $R = \{A_1, A_2, \dots, A_n\}$ .<sup>7</sup> No damos a entender que realmente almacenaremos la base de datos como una única tabla universal; usamos este concepto solo para desarrollar la teoría formal de las dependencias de datos.<sup>8</sup>

**Definición.** Una **dependencia funcional**, denotada por  $X \rightarrow Y$ , entre dos conjuntos de atributos  $X$  e  $Y$  que son subconjuntos de  $R$  especifica una restricción sobre la posible tuplas que pueden formar un estado de relación  $r$  de  $R$ . La restricción es que, para dos tuplas  $t_1$  y  $t_2$  en  $r$  que tienen  $t_1[X] = t_2[X]$ , también deben tener  $t_1[Y] = t_2[Y]$ .

Esto significa que los valores del componente  $Y$  de una tupla en  $r$  dependen o están determinados extraído por, los valores del componente  $X$ ; alternatively, los valores del componente  $X$  de una tupla determina de forma única (o **funcional**) los valores del componente  $Y$ . Nosotros también dicen que hay una dependencia funcional de  $X$  a  $Y$ , o que  $Y$  es **funcionalmente dependiente** de  $X$ . La abreviatura de dependencia funcional es **FD** o **fd**. El conjunto de los atributos  $X$  se denominan el **lado izquierdo** del FD, y  $Y$  se llama el **lado derecho**.

Por tanto,  $X$  determina funcionalmente  $Y$  en un esquema de relación  $R$  si, y sólo si, siempre que dos tuplas de  $r(R)$  coinciden en su valor  $X$ , necesariamente deben estar de acuerdo en su Valor  $Y$ . Tenga en cuenta lo siguiente:

- Si una restricción en  $R$  establece que no puede haber más de una tupla con un dado el valor  $X$  en cualquier instancia de relación  $r(R)$ , es decir,  $X$  es una **clave candidata** de  $R$ : esto implica que  $X \rightarrow Y$  para cualquier subconjunto de atributos  $Y$  de  $R$  (porque la restricción clave implica que no hay dos tuplas en cualquier estado legal  $r(R)$  mismo valor de  $X$ ). Si  $X$  es una clave candidata de  $R$ , entonces  $X \rightarrow R$ .
- Si  $X \rightarrow Y$  en  $R$ , esto no dice si  $Y \rightarrow X$  en  $R$ .

Una dependencia funcional es una propiedad de la **semántica** o **significado de la atributos**. Los diseñadores de bases de datos utilizarán su comprensión de la semántica de la atributos de  $R$ , es decir, cómo se relacionan entre sí, para especificar la función dependencias que deben mantenerse en todos los estados de relación (extensiones)  $r$  de  $R$ . Relación Las extensiones  $r(R)$  que satisfacen las restricciones de dependencia funcional se denominan **legales, estados de relación** (o **extensiones legales**) de  $R$ . Por lo tanto, el uso principal de la dependencia funcional dencias es describir más a fondo un esquema de relación  $R$  especificando restricciones en su Atributos que deben cumplirse en todo momento. Ciertos FD se pueden especificar sin consultar a una relación específica, sino como una propiedad de esos atributos dados sus significado entendido. Por ejemplo,  $\{State, Driver\_license\_number\} \rightarrow Ssn$  debe normalmente es válido para cualquier adulto en los Estados Unidos y, por lo tanto, debe mantenerse siempre que estos atributos aparecen en una relación.<sup>9</sup> También es posible que determinadas

<sup>7</sup> Este concepto de relación universal es importante cuando discutimos los algoritmos para bases de datos relacionales. diseño en el Capítulo 15.

<sup>8</sup> Esta suposición implica que cada atributo de la base de datos debe tener un nombre distinto. En el capítulo 5 prefijamos los nombres de los atributos por los nombres de las relaciones para lograr unicidad siempre que los atributos sean distintos las relaciones tenían el mismo nombre.

<sup>9</sup> Tenga en cuenta que existen bases de datos, como las de agencias de tarjetas de crédito o departamentos de policía, Es posible que la dependencia funcional no se mantenga debido a registros fraudulentos resultantes del mismo número de licencia utilizado por dos o más personas diferentes.

las dependencias pueden dejar de existir en el mundo real si cambia la relación. por ejemplo, el FD  $Zip\_code \rightarrow Area\_code$  solía existir como una relación entre códigos y códigos de números de teléfono en los Estados Unidos, pero con la proliferación de los códigos de área telefónicos ya no es cierto.

Considere el esquema de relación EMP\_PROJ en la figura 14.3 (b); de la semántica de los atributos y la relación, sabemos que las siguientes dependencias funcionales debe contener:

- a.  $Ssn \rightarrow Ename$
- segundo.  $Pnumber \rightarrow \{ Pname, Plocation \}$
- C.  $\{ Ssn, Pnumber \} \rightarrow Horas$

Estas dependencias funcionales especifican que (a) el valor del Social de un empleado El número de seguridad (  $Ssn$  ) determina de forma única el nombre del empleado (  $Ename$  ), (b) el El valor del número de un proyecto (  $Pnumber$  ) determina de forma única el nombre del proyecto. (  $Pname$  ) y la ubicación (  $Plocation$  ), y (c) una combinación de  $Ssn$  y  $Pnumber$  valores determina de forma única el número de horas que el empleado trabaja actualmente en el proyecto por semana (  $Horas$  ). Alternativamente, decimos que  $Ename$  se determina funcionalmente minados por (o funcionalmente dependientes de)  $Ssn$  , o dado un valor de  $Ssn$ , sabemos que valor de  $Ename$ , y así sucesivamente.

Una dependencia funcional es una propiedad del esquema de relación R, no de un estado de relación jurídica r de R. Por lo tanto, un FD no puede inferirse automáticamente de un dada extensión de relación r pero debe ser definido explícitamente por alguien que sepa la semántica de los atributos de R. Por ejemplo, la figura 14.7 muestra un particular estado del esquema de relación TEACH . Aunque a primera vista podemos pensar que  $Texto \rightarrow$  Por supuesto , no podemos confirmar esto a menos que sepamos que es cierto para todos los estados legales de TEACH . Sin embargo, es suficiente demostrar un solo contraexamen ple para refutar una dependencia funcional. Por ejemplo, porque 'Smith' enseña tanto 'Estructuras de datos' y 'Sistemas de bases de datos', podemos concluir que Teacher no determinar funcionalmente el Curso .

Dada una relación poblada, no podemos determinar qué DF son válidos y cuáles no. a menos que sepamos el significado y las relaciones entre los atributos. Todo lo que podamos decir es que una determinada FD puede existir si se mantiene en esa extensión en particular. No podemos garantizar su existencia hasta que entendamos el significado del atributo correspondiente butes. Sin embargo, podemos afirmar enfáticamente que un cierto DF no se cumple si hay

ENSEÑAR		
Profesor	Curso	Texto
Herrero	Estructuras de datos	Bartram
Herrero	Gestión de datos	Martín
sala	Compiladores	Hoffman
marrón	Estructuras de datos	Horowitz

**Figura 14.7**  
Un estado de relación de TEACH con un posible dependencia funcional  $TEXTO \rightarrow CURSO$ . Sin embargo,  $PROFESOR \rightarrow CURSO$ ,  $TEXTO \rightarrow PROFESOR$  y  $CURSO \rightarrow TEXTO$  están descartados.



**Figura 14.8**  
Una relación R (A, B, C, D)  
con su extensión.

UN	segundo	C	re
a1	b1	c1	d1
a1	b2	c2	d2
a2	b2	c2	d3
a3	b3	c4	d3

tuplas que muestran la violación de tal FD. Vea la relación de ejemplo ilustrativa en Figura 14.8. Aquí, los siguientes FD pueden ser válidos porque las cuatro tuplas en el actual extensión no tiene violación de estas restricciones:  $B \rightarrow C$ ;  $C \rightarrow B$ ;  $\{A, B\} \rightarrow C$ ;  $\{A, B\} \rightarrow D$ ; y  $\{C, D\} \rightarrow B$ . Sin embargo, lo siguiente no es válido porque ya tenemos violaciones de ellos en la extensión dada:  $A \rightarrow B$  (las tuplas 1 y 2 violan esta restricción);  $B \rightarrow A$  (las tuplas 2 y 3 violan esta restricción);  $D \rightarrow C$  (las tuplas 3 y 4 la violan).

La figura 14.3 presenta una **notación esquemática** para mostrar FD: Cada FD es se muestra como una línea horizontal. Los atributos del lado izquierdo del FD están conectados por líneas verticales a la línea que representa el FD, mientras que el atributo del lado derecho Los butes están conectados por líneas con flechas que apuntan hacia los atributos.

Denotamos por F el conjunto de dependencias funcionales que se especifican en la relación esquema R. Normalmente, el diseñador de esquemas especifica las dependencias funcionales que son semánticamente obvios; Por lo general, sin embargo, numerosas otras dependencias funcionales En todas las instancias de relaciones legales, las cias se mantienen entre conjuntos de atributos que pueden derivarse de y satisfacer las dependencias en F. Esas otras dependencias pueden inferirse o deducido de los FD en F. Diferimos los detalles de las reglas y propiedades de inferencia de dependencias funcionales al Capítulo 15.

### 14.3 Formas normales basadas en claves primarias

Habiendo introducido dependencias funcionales, ahora estamos listos para usarlas según las especificaciones averigüe cómo utilizarlos para desarrollar una metodología formal para probar y mejorar esquemas de relación. Suponemos que se da un conjunto de dependencias funcionales para cada relación, y que cada relación tiene una clave primaria designada; esta información comprende combinado con las pruebas (condiciones) para las formas normales impulsa el proceso de normalización para el diseño de esquemas relacionales. La mayoría de los proyectos prácticos de diseño relacional toman uno de los siguientes dos enfoques:

- Realizar un diseño de esquema conceptual utilizando un modelo conceptual como ER o EER y mapear el diseño conceptual en un conjunto de relaciones.
- Diseñar las relaciones basadas en el conocimiento externo derivado de un

implementación de archivos o formularios o informes.  
Siguiendo cualquiera de estos enfoques, es útil evaluar las relaciones para bondad y descomponerlos aún más según sea necesario para lograr formas normales más altas utilizando la teoría de la normalización presentada en este capítulo y el siguiente. Nos enfocamos en

esta sección sobre las tres primeras formas normales de esquemas de relación y la intuición detrás de ellos, y discutimos cómo se desarrollaron históricamente. Mas general definiciones de estas formas normales, que tienen en cuenta todas las claves candidatas de un relación en lugar de solo la clave principal, se difieren a la Sección 14.4.

Comenzamos discutiendo informalmente las formas normales y la motivación detrás de sus desarrollo, así como revisar algunas definiciones del Capítulo 3 que son necesarias aquí. Luego discutimos la primera forma normal (1NF) en la Sección 14.3.4, y presentamos las definiciones de segunda forma normal (2NF) y tercera forma normal (3NF), que se basan en claves primarias, en las Secciones 14.3.5 y 14.3.6, respectivamente.

### 14.3.1 Normalización de relaciones

El proceso de normalización, propuesto por primera vez por Codd (1972a), toma una relación esquema mediante una serie de pruebas para certificar si satisface una determinada **forma normal**. El proceso, que procede de arriba hacia abajo evaluando cada relación contra los criterios para las formas normales y descomponer las relaciones según sea necesario, puede por tanto, ser considerado como diseño relacional por análisis. Inicialmente, Codd propuso tres formas normales, a las que llamó primera, segunda y tercera forma normal. Un mas fuerte La definición de 3NF, llamada forma normal de Boyce-Codd (BCNF), se propuso más tarde por Boyce y Codd. Todas estas formas normales se basan en una única herramienta analítica: la dependencias funcionales entre los atributos de una relación. Más tarde, una cuarta normal forma (4NF) y una quinta forma normal (5NF) se propusieron, con base en los conceptos de dependencias multivalor y dependencias de unión, respectivamente; estos se describen brevemente tratado en las Secciones 14.6 y 14.7.

**La normalización de datos** puede considerarse un proceso de análisis de la relación dada. esquemas basados en sus FD y claves primarias para lograr las propiedades deseables de (1) minimizar la redundancia y (2) minimizar la inserción, eliminación y actualización anomalías discutidas en la Sección 14.1.2. Puede considerarse como un "filtrado" o "purificación" para hacer que el diseño tenga una calidad sucesivamente mejor. Un insatisfactorio esquema de relación que no cumple la condición de una forma normal, la **normal prueba de forma:** se descompone en esquemas de relación más pequeños que contienen un subconjunto de atributos y cumplen con la prueba que de otro modo no se cumplía en la relación original. Así, el procedimiento de normalización proporciona a los diseñadores de bases de datos lo siguiente:

- Un marco formal para analizar los esquemas de relación en función de sus claves y sobre las dependencias funcionales entre sus atributos

- Una serie de pruebas de forma normal que se pueden realizar en relación individual para que la base de datos relacional se pueda **normalizar** a cualquier la licenciatura

**Definición.** La **forma normal** de una relación se refiere a la forma normal más alta condición que cumple, y por lo tanto indica el grado en que ha sido normalizado.

Las formas normales, cuando se consideran aisladas de otros factores, no garantizan un buen diseño de base de datos. Por lo general, no es suficiente comprobar por separado que cada

El esquema de relación en la base de datos está, digamos, en BCNF o 3NF. Más bien, el proceso de normalización por descomposición también debe confirmar la existencia de propiedades que los esquemas relacionales, tomados en conjunto, deben poseer. Estos harían incluir dos propiedades:

- La **propiedad de combinación no aditiva o combinación sin pérdida**, que garantiza que El problema de generación de tuplas espúreas discutido en la Sección 14.1.4 no ocurren con respecto a los esquemas de relación creados después de la descomposición
- La **propiedad de conservación de la dependencia**, que garantiza que cada función La dependencia está representada en alguna relación individual resultante de descomposición

La propiedad de unión no aditiva es extremadamente crítica y **debe lograrse en cualquier costo**, mientras que la propiedad de preservación de la dependencia, aunque deseable, es algo veces sacrificados, como discutimos en la Sección 15.2.2. Aplazamos la discusión de los conceptos y técnicas mal que garantizan las dos propiedades anteriores al Capítulo 15.

### 14.3.2 Uso práctico de formas normales

La mayoría de los proyectos de diseño prácticos en entornos comerciales y gubernamentales adquieren diseños existentes de bases de datos de diseños anteriores, de diseños en modelos heredados, o de archivos existentes. Ciertamente están interesados en asegurarse de que los diseños sean buenos calidad y sostenible durante largos períodos de tiempo. Los diseños existentes son evaluados por aplicando las pruebas para formas normales, y la normalización se lleva a cabo en la práctica que los diseños resultantes sean de alta calidad y cumplan con las propiedades deseables indicadas previamente. Aunque se han definido varias formas normales superiores, como la 4NF y 5NF que discutimos en las Secciones 14.6 y 14.7, la utilidad práctica de estas formas se vuelven cuestionables. La razón es que las restricciones sobre las que están basados son raros y difíciles de entender para los diseñadores de bases de datos y los usuarios detectar. Los diseñadores y usuarios deben conocerlos o descubrirlos como parte del negocio. Por lo tanto, el diseño de bases de datos tal como se practica en la industria hoy en día paga atención a la normalización solo hasta 3NF, BCNF o como máximo 4NF.

Otro punto que vale la pena señalar es que los diseñadores de bases de datos no necesitan normalizar forma normal más alta posible. Las relaciones pueden dejarse en un estado de normalización más bajo,

como 2NF, por razones de rendimiento, como las que se analizan al final de la sección 14.1.2. Hacerlo incurre en las sanciones correspondientes de hacer frente a las anomalías.

**Definición.** La **desnormalización** es el proceso de almacenar la unión de normas superiores relaciones de forma mal como una relación de base, que está en una forma normal inferior.

### 14.3.3 Definiciones de claves y atributos que participan en claves

Antes de continuar, veamos nuevamente las definiciones de claves de una relación. esquema del Capítulo 3.

**Definición.** Una **superclave** de un esquema de relación  $R = \{A_1, A_2, \dots, A_n\}$  es un conjunto de atributos pero  $S \subseteq R$  con la propiedad de que no hay dos tuplas  $t_1$  y  $t_2$  en cualquier relación legal el estado  $r$  de  $R$  tendrá  $t_1[S] = t_2[S]$ . Una **clave**  $K$  es una superclave con la propiedad adicional que la eliminación de cualquier atributo de  $K$  hará que  $K$  ya no sea una superclave.

La diferencia entre una clave y una superclave es que una clave debe ser mínima; es decir, si tenemos una clave  $K = \{A_1, A_2, \dots, A_k\}$  de  $R$ , entonces  $K - \{A_i\}$  no es una clave de  $R$  para cualquier  $A_i$ ,  $1 \leq i \leq k$ . En la Figura 14.1,  $\{Ssn\}$  es una clave para EMPLEADO, mientras que  $\{Ssn\}$ ,  $\{Ssn, Ename\}$ ,  $\{Ssn, Ename, Bdate\}$  y cualquier conjunto de atributos que incluya  $Ssn$  son superclaves.

Si un esquema de relación tiene más de una clave, cada una se denomina **clave candidata**. Uno de las claves candidatas se designan arbitrariamente para ser la **clave principal**, y las otras se llaman claves secundarias. En una base de datos relacional práctica, cada esquema de relación debe tener una clave primaria. Si no se conoce ninguna clave candidata para una relación, toda la relación se puede tratar como una superclave predeterminada. En la Figura 14.1,  $\{Ssn\}$  es el único candidato clave para EMPLEADO, por lo que también es la clave principal.

**Definición.** Un atributo del esquema de relación  $R$  se denomina **atributo principal** de  $R$  si es un miembro de alguna clave candidata de  $R$ . Un atributo se llama **nonprime** si no es un atributo principal, es decir, si no es miembro de ninguna clave candidata.

En la Figura 14.1, tanto  $Ssn$  como  $Pnumber$  son atributos principales de WORKS\_ON, mientras que otros atributos de WORKS\_ON no son primarios.

Ahora presentamos las tres primeras formas normales: 1NF, 2NF y 3NF. Estos fueron propuestos por Codd (1972a) como una secuencia para lograr el estado deseable de las relaciones 3NF progresando a través de los estados intermedios de 1NF y 2NF si es necesario. Como nosotros Verá, 2NF y 3NF atacan independientemente diferentes tipos de problemas que surgen de dependencias funcionales problemáticas entre atributos. Sin embargo, para la historia razones cal, se acostumbra seguirlas en esa secuencia; por lo tanto, por definición un La relación 3NF ya satisface 2NF.

### 14.3.4 Primera forma normal

La **primera forma normal (1NF)** ahora se considera parte de la definición formal de un relación en el modelo relacional básico (plano); históricamente, se definió para no permitir

atributos multivalor, atributos compuestos y sus combinaciones. Se afirma que el dominio de un atributo debe incluir solo valores atómicos (simples, indivisibles) y que el valor de cualquier atributo en una tupla debe ser un valor único del dominio de ese atributo. Por tanto, 1NF no permite tener un conjunto de valores, una tupla de valores o una combinación de ambos como valor de atributo para una sola tupla. En otras palabras, 1NF no permite relaciones dentro de relaciones o relaciones como valores de atributo dentro de tuplas. los Sólo los valores de atributos permitidos por 1NF son individuales **atómicos (o indivisibles ) valores** .

Considere el esquema de relación DEPARTAMENTO que se muestra en la figura 14.1, cuyo principal La clave es Dnumber , y supongamos que la ampliamos incluyendo el atributo Dlocations como se muestra en la Figura 14.9 (a). Suponemos que cada departamento puede tener varios ubicaciones. El esquema DEPARTAMENTO y un estado de relación de muestra se muestran en la figura. ure 14,9. Como podemos ver, esto no está en 1NF porque Dlocations no es un atributo atómico. bute, como lo ilustra la primera tupla en la Figura 14.9 (b). Hay dos formas en que podemos mira el atributo Dlocations :

- El dominio de Dlocations contiene valores atómicos, pero algunas tuplas pueden tener un conjunto de estos valores. En este caso , Dlocations no depende funcionalmente de la clave principal Dnumber .



- El dominio de Dlocations contiene conjuntos de valores y, por tanto, no es atómico. En este caso, Dnumber  $\rightarrow$  Dlocations porque cada conjunto se considera un miembro del dominio de atributo.<sup>10</sup>

En cualquier caso, la relación DEPARTAMENTO de la figura 14.9 no está en 1NF; de hecho, lo hace ni siquiera calificamos como una relación de acuerdo con nuestra definición de relación en la Sección 3.1. Hay tres técnicas principales para lograr la primera forma normal para tal relación:

1. Elimine el atributo Dlocations que infringe 1NF y colóquelo en un lugar relación DEPT\_LOCATIONS junto con la clave principal Dnumber of DEPARTAMENTO. La clave principal de esta relación recién formada es la combinación { Dnumber, Dlocation }, como se muestra en la Figura 14.2. Una tupla distinta en DEPT\_LOCATIONS existe para cada ubicación de un departamento. Esta descomposición plantea la relación no 1NF en dos relaciones 1NF.

<sup>10</sup> En este caso podemos considerar el dominio de Dlocations como el **conjunto de potencias** del conjunto de ubicaciones; es decir, el dominio está formado por todos los posibles subconjuntos del conjunto de ubicaciones únicas.

2. Expandir la clave para que haya una tupla separada en el original.  
Relación DEPARTAMENTO para cada ubicación de un DEPARTAMENTO, como se muestra en la Figura 14.9 (c). En este caso, la clave principal se convierte en la combinación { Dnumber, Dlocation }. Esta solución tiene la desventaja de introducir redundancia en la relación y, por lo tanto, rara vez se adopta.
3. Si se conoce un número máximo de valores para el atributo, por ejemplo, si se sabe que como máximo pueden existir tres ubicaciones para un departamento: reemplace el atributo Dlocations por tres atributos atómicos: Dlocation1, Dlocation2 y Dlocation3. Esta solución tiene la desventaja de introducir valores NULL si la mayoría de los departamentos tienen menos de tres ubicaciones. Además introduce semántica espuria sobre el orden entre los valores de ubicación; ese el pedido no está previsto originalmente. Consultar este atributo se vuelve más difícil; Por ejemplo, considere cómo escribiría la consulta: Enumere los departamentos que tienen 'Bellaire' como una de sus ubicaciones en este diseño. Para todos Por estas razones, es mejor evitar esta alternativa.

De las tres soluciones anteriores, la primera se considera generalmente la mejor porque no adolece de redundancia y es completamente general; no coloca máximo límite inum en el número de valores. De hecho, si elegimos la segunda solución, se descompondrá aún más durante los siguientes pasos de normalización en el

primera solución.

La primera forma normal tampoco permite atributos multivalor que se componen posite. Se denominan **relaciones anidadas** porque cada tupla puede tener una relación dentro de ella. La figura 14.10 muestra cómo podría aparecer la relación EMP\_PROJ si el anidamiento es permitido. Cada tupla representa una entidad empleado, y una relación PROJS (Pnumber, Horas ) dentro de cada tupla representa los proyectos del empleado y las horas por semana ese empleado trabaja en cada proyecto. El esquema de esta relación EMP\_PROJ puede ser representado de la siguiente manera:

```
EMP_PROJ (Ssn, Ename, {PROJS (Pnumber, Hours)})
```

El conjunto de llaves { } identifica el atributo PROJS como multivalor, y enumeramos el atributos del componente que forman PROJS entre paréntesis (). Curiosamente, reciente tendencias para admitir objetos complejos (consulte el capítulo 12) y datos XML (consulte el capítulo ter 13) intentar permitir y formalizar relaciones anidadas dentro de la base de datos relacional sistemas, que fueron rechazados desde el principio por 1NF.

Observe que Ssn es la clave principal de la relación EMP\_PROJ en las Figuras 14.10 (a) y (b), mientras que Pnumber es la clave **parcial** de la relación anidada; es decir, dentro de cada tupla, la relación anidada debe tener valores únicos de Pnumber . Para normalizar esto en 1NF, eliminamos los atributos de relación anidados en una nueva relación y propagamos ingrese la clave principal; la clave primaria de la nueva relación combinará el clave parcial con la clave primaria de la relación original. Descomposición y pri- La propagación de la clave María produce los esquemas EMP\_PROJ1 y EMP\_PROJ2 , como se muestra en Figura 14.10 (c).

Este procedimiento se puede aplicar de forma recursiva a una relación con anidamiento de varios niveles para **desanidar** la relación en un conjunto de relaciones 1NF. Esto es útil para convertir un

(un)			
EMP_PROJ		Proyectos	
Ssn	Ename	Número de horas	
(segundo)			
EMP_PROJ			
Ssn	Ename	Pnumber	Horas
123456789	Smith, John B.	1	32,5
		2	7.5
666884444	Narayan, Ramesh K.	3	40,0
453453453	Inglés, Joyce A.	1	20,0
		2	20,0
333445555	Wong, Franklin T.	2	10.0

			3	10.0
			10	10.0
			20	10.0
	999887777	Zelaya, Alicia J.	30	30,0
			10	10.0
Figura 14.10				
Normalizando anidado	987987987	Jabbar, Ahmad V.	10	35,0
relaciones en 1NF.			30	5,0
(a) Esquema del	987654321	Wallace, Jennifer S.	30	20,0
Relación EMP_PROJ con			20	15.0
un atributo de relación anidado	888665555	Borg, James E.	20	NULO
PROYECTOS. (b) Muestra				
extensión de la				
Relación EMP_PROJ				
mostrando relaciones anidadas	(C)			
dentro de cada tupla.	EMP_PROJ1			
(c) Descomposición de	Ssn	Ename		
EMP_PROJ en relaciones				
EMP_PROJ1 y	EMP_PROJ2			
EMP_PROJ2 por	Ssn	Pnumber	Horas	
propagando el primario				
llave.				

esquema de relación no normalizado con muchos niveles de anidamiento en relaciones 1NF. Como un ejemplo, considere lo siguiente:

CANDIDATO (Ssn, Nombre, {JOB\_HIST (Empresa, Posición\_más alta, {SAL\_HIST (Año, Max\_sal)}}))

Lo anterior describe datos sobre candidatos que solicitan puestos de trabajo con su historial de trabajo. historia como una relación anidada dentro de la cual el historial salarial se almacena como un anidado más profundo

relación. La primera normalización utilizando claves parciales internas Compañía y Año, respectivamente, resulta en las siguientes relaciones 1NF:

CANDIDATE\_1 (Ssn, Nombre)  
CANDIDATE\_JOB\_HIST (Ssn, Company, Highest\_position)  
CANDIDATE\_SAL\_HIST (Ssn, Company, Year, Max-sal)

La existencia de más de un atributo multivalor en una relación debe manejarse dled cuidadosamente. Como ejemplo, considere la siguiente relación no 1NF:

PERSONA (Ss #, {Car\_lic #}, {Phone #})

Esta relación representa el hecho de que una persona tiene varios coches y varios teléfonos.



Si se sigue la estrategia 2 anterior, da como resultado una relación de todas las claves:

PERSON\_IN\_1NF (Ss #, Car\_lic #, Phone #)

Para evitar introducir una relación extraña entre Car\_lic # y Phone #, todos posibles combinaciones de valores se representan para cada Ss #, dando lugar a redundancy. Esto conduce a los problemas que normalmente se descubren en una etapa posterior de normalización y que se manejan mediante dependencias multivalor y 4NF, que discutiremos en la Sección 14.6. La forma correcta de lidiar con los dos atributos multivalor pero en PERSONA mostrado anteriormente es descomponerlo en dos relaciones separadas, usando la estrategia 1 discutida anteriormente: P1 (Ss #, Car\_lic #) y P2 (Ss #, Phone #) .

Una nota sobre las relaciones que involucran atributos que van más allá de lo numérico y datos de cadena de caracteres. Se está volviendo común en las bases de datos actuales incorporar imágenes, documentos, videoclips, clips de audio, etc. Cuando estos se almacenan en un relación, todo el objeto o archivo se trata como un valor atómico, que se almacena como un Tipo de datos BLOB (objeto grande binario) o CLOB (objeto grande de caracteres) usando SQL. A efectos prácticos, el objeto se trata como un atributo atómico de un solo valor. y por lo tanto mantiene el estatus 1NF de la relación.

### 14.3.5 Segunda forma normal

**La segunda forma normal (2NF)** se basa en el concepto de dependencia funcional completa.

Una dependencia funcional  $X \rightarrow Y$  es una **dependencia funcional completa** si se elimina cualquier el atributo A de X significa que la dependencia ya no se mantiene; es decir, para cualquier atributo  $A \in X$ ,  $(X - \{A\})$  no determina funcionalmente Y. Un funcional

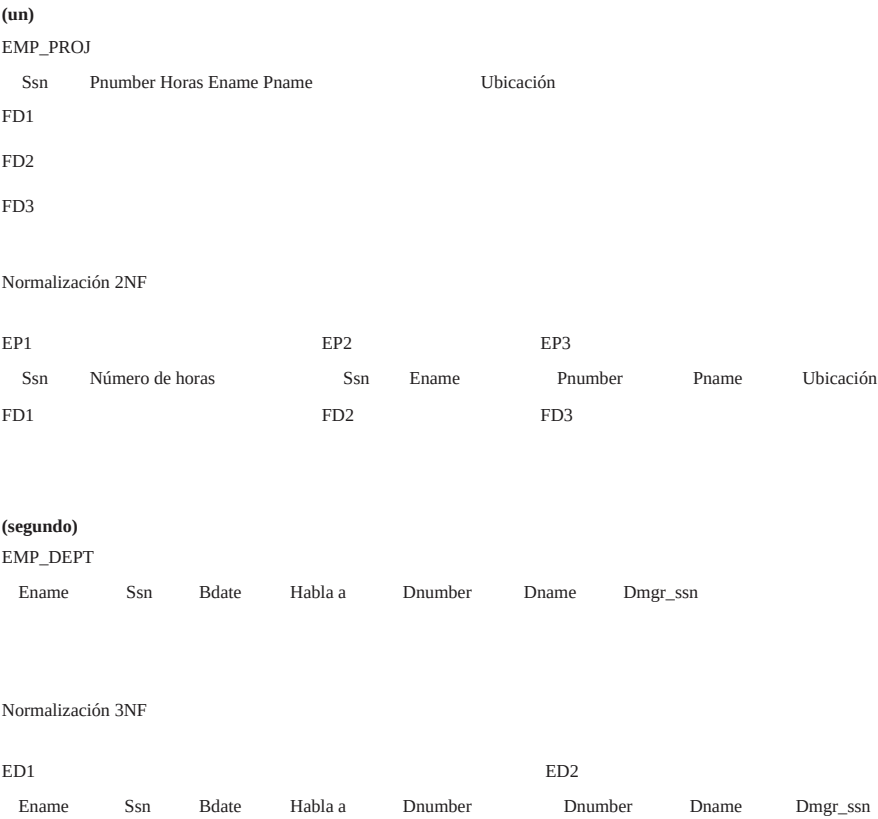
La dependencia  $X \rightarrow Y$  es una **dependencia parcial** si se puede eliminar algún atributo  $A \in X$  de X y la dependencia todavía se mantiene; es decir, para algunos  $A \in X$ ,  $(X - \{A\}) \rightarrow Y$ . En Figura 14.3 (b),  $\{Ssn, Pnumber\} \rightarrow Hours$  es una dependencia completa (ni  $Ssn \rightarrow Hours$  ni  $Pnumber \rightarrow Hours$  retenidas). Sin embargo, la dependencia  $\{Ssn, Pnumber\} \rightarrow Ename$  es parcial porque  $Ssn \rightarrow Ename$  se mantiene.

**Definición.** Un esquema de relación R está en 2NF si cada atributo no principal A en R es totalmente funcionalmente dependiente de la clave primaria de R.

La prueba para 2NF implica probar las dependencias funcionales cuyo lado izquierdo los atributos son parte de la clave principal. Si la clave principal contiene un solo atributo, no es necesario aplicar la prueba en absoluto. La relación EMP\_PROJ en la Figura 14.3 (b) está en

1NF pero no en 2NF. El atributo nonprime Ename viola 2NF debido a FD2, al igual que los atributos no preferenciales Pname y Plocation debido a FD3. Cada una de las funciones dependencias nacionales FD2 y FD3 viola 2NF porque Ename puede ser funcionalmente determinado solo por Ssn, y tanto Pname como Plocation se pueden determinar funcionalmente minado solo por Pnumber. Los atributos Ssn y Pnumber son parte de la clave principal  $\{Ssn, Pnumber\}$  de EMP\_PROJ, violando así la prueba 2NF.

Si un esquema de relación no está en 2NF, se puede normalizar en segundo lugar o normalizar en 2NF en una serie de relaciones 2NF en las que los atributos no principales están asociados solo con la parte de la clave primaria de la que dependen totalmente funcionalmente. Por lo tanto, las dependencias funcionales FD1 , FD2 y FD3 en la Figura 14.3 (b) conducen a la descomposición de EMP\_PROJ en los tres esquemas de relación EP1 , EP2 y EP3 que se muestra en la Figura 14.11 (a), cada uno de los cuales está en 2NF.



**Figura 14.11**  
Normalizando en 2NF y 3NF. (a) Normalizar EMP\_PROJ en Relaciones 2NF. (b) Normalización de EMP\_DEPT en relaciones 3NF.

**La tercera forma normal (3NF)** se basa en el concepto de dependencia transitiva. Una función de dependencia funcional  $X \rightarrow Y$  en un esquema de relación  $R$  es una **dependencia transitiva** si hay existe un conjunto de atributos  $Z$  en  $R$  que no es una clave candidata ni un subconjunto de ninguna clave de  $R$ ,<sup>11</sup> y tanto  $X \rightarrow Z$  como  $Z \rightarrow Y$  se mantienen. La dependencia  $Ssn \rightarrow Dmgr\_ssn$  es transitiva a  $Dnumber$  en  $EMP\_DEPT$  en la Figura 14.3 (a), porque ambas dependencias  $Ssn \rightarrow Dnumber$  y  $Dnumber \rightarrow Dmgr\_ssn$  hold y  $Dnumber$  no es una clave en sí misma ni una subconjunto de la clave de  $EMP\_DEPT$ . Intuitivamente, podemos ver que la dependencia de  $Dmgr\_ssn$  on  $Dnumber$  no es deseable en  $EMP\_DEPT$  ya que  $Dnumber$  no es una clave de  $EMP\_DEPT$ .

**Definición.** Según la definición original de Codd, un esquema de relación  $R$  está en **3NF** si satisface 2NF y ningún atributo no principal de  $R$  es transitivamente dependiente en la clave principal.

El esquema de relación  $EMP\_DEPT$  en la figura 14.3 (a) está en 2NF, ya que no hay dependencia parcial existencias en una clave. Sin embargo,  $EMP\_DEPT$  no está en 3NF debido a la transitiva dependencia de  $Dmgr\_ssn$  (y también  $Dname$ ) en  $Ssn$  a través de  $Dnumber$ . Podemos normalizar  $EMP\_DEPT$  descomponiéndolo en los dos esquemas de relación 3NF  $ED1$  y  $ED2$  mostrado en la Figura 14.11 (b). Intuitivamente, vemos que  $ED1$  y  $ED2$  representan datos sobre empleados y departamentos, los cuales son entidades en su derecho propio. Una operación NATURAL JOIN en  $ED1$  y  $ED2$  recuperará el original relación  $EMP\_DEPT$  sin generar tuplas falsas.

Intuitivamente, podemos ver que cualquier dependencia funcional en la que el lado izquierdo está parte (un subconjunto adecuado) de la clave primaria, o cualquier dependencia funcional en la que el el lado izquierdo es un atributo no clave, es un FD problemático. Normalización 2NF y 3NF elimine estos DF problemáticos descomponiendo la relación original en nuevas relaciones. En términos del proceso de normalización, no es necesario eliminar la dependencia parcial cías antes de las dependencias transitivas, pero históricamente, 3NF se ha definido con el suposición de que una relación se prueba para 2NF primero antes de probarla para 3NF. Además, la definición general de 3NF que presentamos en la Sección 14.4.2 cubre automáticamente el condición de que la relación también satisfaga 2NF. La tabla 14.1 resume informalmente la tres formas normales basadas en claves primarias, las pruebas utilizadas en cada caso, y las remedio espontáneo o normalización realizada para lograr la forma normal.

## 14.4 Definiciones generales de segundo y terceras formas normales

En general, queremos diseñar nuestros esquemas de relación para que no tengan ni parciales ni dependencias transitivas porque este tipo de dependencias provocan la actualización anomalías discutidas en la Sección 14.1.2. Los pasos para la normalización en relación 3NF las que hemos discutido hasta ahora no permiten dependencias parciales y transitivas en

<sup>11</sup> Ésta es la definición general de dependencia transitiva. Porque nos preocupamos solo por la claves en esta sección, permitimos dependencias transitivas donde  $X$  es la clave primaria pero  $Z$  puede ser (un subconjunto de) una clave candidata.

**Tabla 14.1** Resumen de formas normales basadas en claves primarias y normalización correspondiente

Forma normal	Prueba	Remedio (normalización)
Primero (1NF)	La relación no debe tener valores múltiples atributos o relaciones anidadas.	Forme nuevas relaciones para cada multivalor atributo o relación anidada.
Segundo (2NF)	Para relaciones donde la clave principal contiene múltiples atributos, sin clave atributo debe ser funcionalmente depende de una parte de la clave principal.	Descomponer y establecer una nueva relación para cada clave parcial con su dependiente atributo (s). Asegúrate de mantener una relación con la clave primaria original y cualquier atributos que son completamente funcionales dependiente de ella.
Tercero (3NF)	La relación no debe tener una clave atributo funcionalmente determinado por otro atributo no clave (o por un conjunto de atributos no clave). Es decir, debería no habrá dependencia transitiva de una no clave atributo en la clave principal.	Descomponer y establecer una relación que incluye los atributos no clave que determinan funcionalmente otros no clave atributo (s).

la clave principal. El procedimiento de normalización descrito hasta ahora es útil para el análisis en situaciones prácticas para una base de datos determinada donde las claves primarias ya han sido definido. Estas definiciones, sin embargo, no toman otras claves candidatas de una relación, si cualquiera, en cuenta. En esta sección damos las definiciones más generales de 2NF y 3NF que tienen en cuenta todas las claves candidatas de una relación. Tenga en cuenta que esto no afectan la definición de 1NF ya que es independiente de las claves y dependencias funcionales. Como definición general de **atributo principal**, un atributo que forma parte de cualquier La clave candidata se considerará principal. Dependencias funcionales parciales y completas y las dependencias transitivas ahora se considerarán con respecto a todas las claves candidatas de una relación.

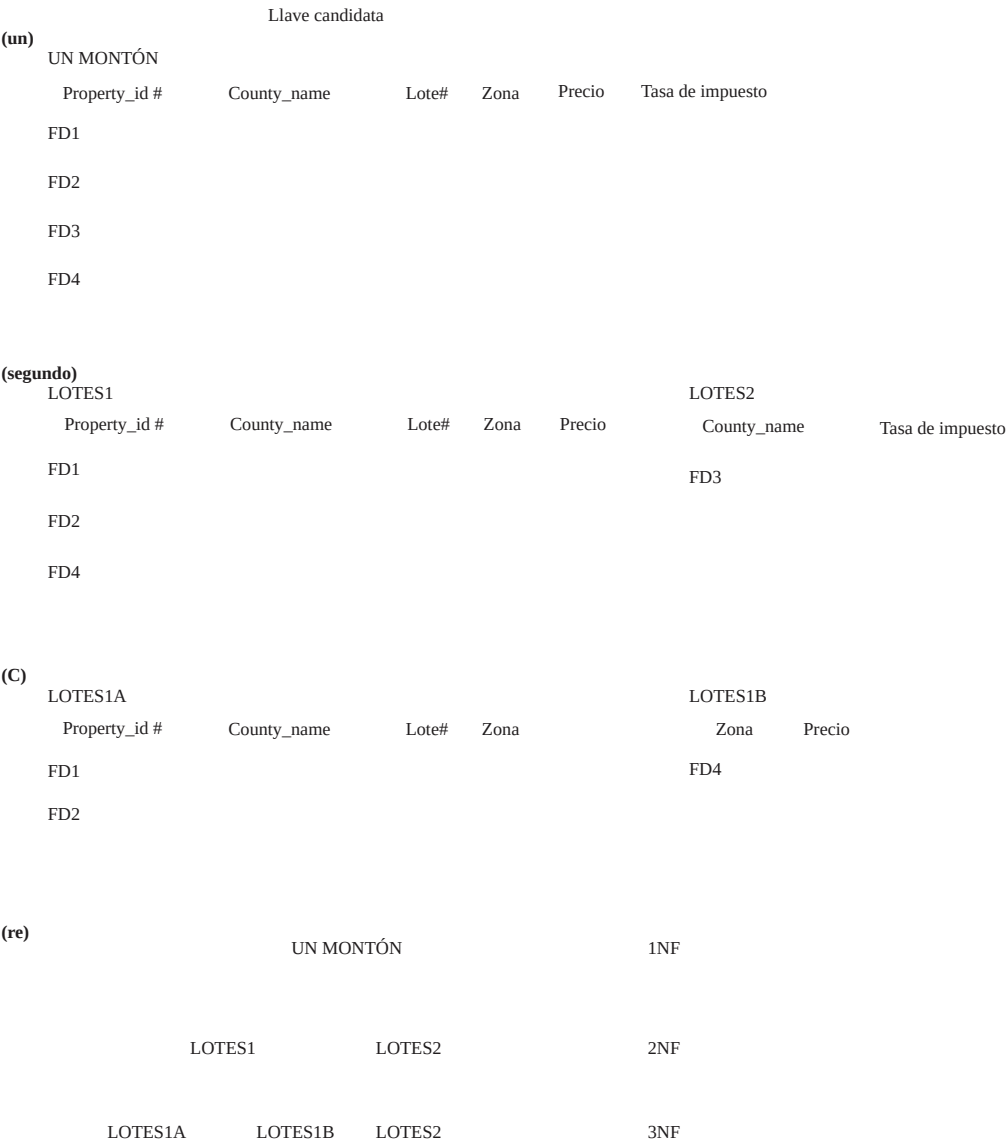
#### 14.4.1 Definición general de la segunda forma normal

**Definición.** Un esquema de relación R está en **segunda forma normal (2NF)** si cada atributo no primo A en R no es parcialmente dependiente de cualquier tecla de R. <sup>12</sup>

La prueba para 2NF implica probar las dependencias funcionales cuyo lado izquierdo los atributos son parte de la clave principal. Si la clave principal contiene un solo atributo, no es necesario aplicar la prueba en absoluto. Considere el esquema de relación LOTS que se muestra en Figura 14.12 (a), que describe parcelas de tierra para la venta en varios condados de un estado. Suponga que hay dos claves candidatas: Property\_id # y { County\_name , Lote # }; es decir, los números de lote son únicos solo dentro de cada condado, pero Property\_id # los números son únicos en los condados de todo el estado.

<sup>12</sup> Esta definición se puede reformular de la siguiente manera: Un esquema de relación R está en 2NF si cada atributo no principal A en R es completamente funcionalmente dependiente de cada tecla de R.

**Figura 14.12**  
Normalización en 2NF y 3NF. (a) La relación LOTS con sus dependencias funcionales FD1 a FD4. (b) Descomposición en las relaciones 2NF LOTS1 y LOTS2. (c) Descomponer LOTS1 en las relaciones 3NF LOTS1A y LOTS1B. (d) progresivo normalización de LOTES en un diseño 3NF.



Basado en las dos claves candidatas  $\text{Property\_id \#}$  y  $\{ \text{County\_name}, \text{Lot \#} \}$ , la función se mantienen las dependencias regionales FD1 y FD2 de la figura 14.12 (a). Elegimos  $\text{Property\_id \#}$  como clave principal, por lo que se subraya en la Figura 14.12 (a), pero no hay ninguna consideración especial. Se le dará a esta clave sobre la otra clave candidata. Supongamos que el siguiente tener dos dependencias funcionales adicionales en LOTES :

FD3:  $\text{County\_name} \rightarrow \text{Tax\_rate}$

FD4:  $\text{Área} \rightarrow \text{Precio}$

En palabras, la dependencia FD3 dice que la tasa impositiva es fija para un condado determinado (no no varían lote por lote dentro del mismo condado), mientras que FD4 dice que el precio de un lote está determinada por su área independientemente del condado en el que se encuentre. (Suponga que esta es la precio del lote a efectos fiscales.)

El esquema de relación LOTS viola la definición general de 2NF porque  $\text{Tax\_rate}$  es parcialmente dependiente de la clave candidata  $\{ \text{County\_name}, \text{Lot \#} \}$ , debido a FD3 . Para normalizar LOTS en 2NF, lo descomponemos en las dos relaciones LOTS1 y LOTS2 , mostrado en la Figura 14.12 (b). Construimos LOTES1 eliminando el atributo  $\text{Tasa\_impuesto}$  que viola 2NF de LOTS y lo coloca con  $\text{County\_name}$  (el lado izquierdo de FD3 que causa la dependencia parcial) en otra relación LOTS2 . Ambos LOTES1 y LOTS2 están en 2NF. Tenga en cuenta que FD4 no viola 2NF y se transfiere a LOTES 1 .

#### 14.4.2 Definición general de la tercera forma normal

**Definición.** Un esquema de relación  $R$  está en **tercera forma normal (3NF)** si, siempre que un dependencia funcional no trivial  $X \rightarrow A$  se mantiene en  $R$ , ya sea (a)  $X$  es una superclave de  $R$ , o (b)  $A$  es un atributo principal de  $R$ .<sup>13</sup>

De acuerdo con esta definición, LOTS2 (Figura 14.12 (b)) está en 3NF. Sin embargo, FD4 en LOTS1 viola 3NF porque el área no es una superclave y el precio no es un atributo principal en LOTES1 . Para normalizar LOTS1 en 3NF, lo descomponemos en el esquema de relación mas LOTS1A y LOTS1B que se muestran en la Figura 14.12 (c). Construimos LOTS1A por eliminando el atributo Precio que viola 3NF de LOTES1 y colocándolo con Área (el lado izquierdo de FD4 que causa la dependencia transitiva) en otra relación ción LOTS1B . Tanto LOTS1A como LOTS1B están en 3NF.

Vale la pena señalar dos puntos sobre este ejemplo y la definición general de 3NF:

- LOTS1 infringe 3NF porque el precio depende transitivamente de cada uno de los claves candidatas de LOTS1 a través del atributo no principal Área .
- Esta definición general se puede aplicar directamente para probar si un esquema de relación está en 3NF; no tiene que pasar primero por 2NF. En otras palabras, si una relación pasa la prueba general 3NF, luego pasa automáticamente la prueba 2NF.

<sup>13</sup> Tenga en cuenta que en base a los fd inferidos (que se discuten en la Sección 15.1), el fd  $Y \rightarrow X$  también se cumple siempre que  $X \rightarrow Y$  sea verdadero. Por lo tanto, una forma un poco mejor de decir esta afirmación es que  $\{AX\}$  es un primo atributo de  $R$ .

Si aplicamos la definición 3NF anterior a LOTES con las dependencias FD1 a FD4, encontramos que tanto FD3 como FD4 violan 3NF según la definición general anterior porque LHS County\_name en FD3 no es una superclave. Por lo tanto, podríamos descomponer LOTS en LOTS1A, LOTS1B y LOTS2 directamente. Por tanto, el transitivo y las dependencias parciales que violen 3NF se pueden eliminar en cualquier orden.

#### 14.4.3 Interpretación de la definición general de la tercera forma normal

Un esquema de relación R viola la definición general de 3NF si una dependencia funcional  $X \rightarrow A$  se cumple en R que cumple cualquiera de las dos condiciones, a saber (a) y (b). La primera condición "detecta" dos tipos de dependencias problemáticas:

- Un atributo nonprime determina otro atributo nonprime. Aquí tenemos una dependencia transitiva que viola 3NF.
- Un subconjunto adecuado de una clave de R determina funcionalmente un atributo no principal. Aquí tenemos una dependencia parcial que viola 2NF.

Por tanto, la condición (a) por sí sola aborda las dependencias problemáticas que fueron causas para la segunda y tercera normalización como comentamos.

Por lo tanto, podemos establecer una **definición alternativa general de 3NF de la** siguiente manera:

Definición alternativa. Un esquema de relación R está en 3NF si cada atributo no principal de R cumple las dos condiciones siguientes:

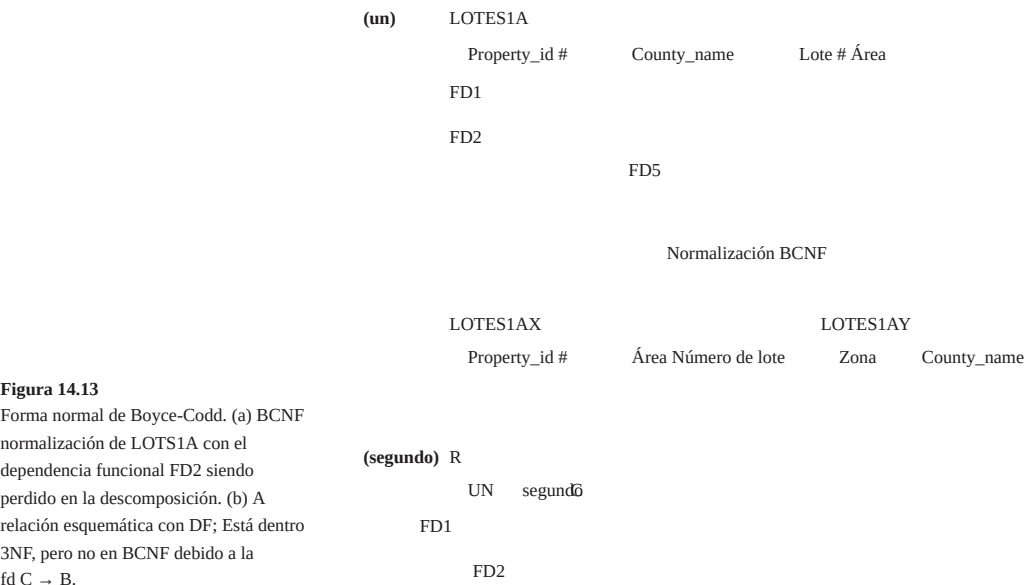
- Es completamente funcionalmente dependiente de cada tecla de R.
- No depende de forma transitoria de cada tecla de R.

Sin embargo, tenga en cuenta la cláusula (b) en la definición general de 3NF. Permite ciertas funciones dependencias tradicionales para deslizarse o escapar en el sentido de que están bien con el 3NF definición y por lo tanto no están "capturados" por la definición de 3NF a pesar de ser potencialmente problemático. La forma normal de Boyce-Codd "atrapa" estas dependencias en que no las permite. A continuación, discutimos esa forma normal.

### 14.5 Forma normal de Boyce-Codd

La **forma normal de Boyce-Codd (BCNF)** se propuso como una forma más simple de 3NF, pero se encontró que era más estricto que 3NF. Es decir, toda relación en BCNF también está en 3NF; sin embargo, una relación en 3NF no está necesariamente en BCNF. Señalamos en el último subsección que aunque 3NF permite dependencias funcionales que se ajustan a la cláusula (b) en la definición de 3NF, BCNF los rechaza y por lo tanto es una definición más estricta de una forma normal.

Intuitivamente, podemos ver la necesidad de una forma normal más fuerte que 3NF volviendo a el esquema de relación LOTS en la Figura 14.12 (a) con sus cuatro dependencias funcionales FD1 a través de FD4. Suponga que tenemos miles de lotes en la relación pero los lotes son de solo dos condados: DeKalb y Fulton. Suponga también que los tamaños de lote en DeKalb El condado tiene solo 0.5, 0.6, 0.7, 0.8, 0.9 y 1.0 acres, mientras que los tamaños de lote en el condado de Fulton



**Figura 14.13**  
Forma normal de Boyce-Codd. (a) BCNF normalización de LOTS1A con el dependencia funcional FD2 siendo perdido en la descomposición. (b) A relación esquemática con DF; Está dentro 3NF, pero no en BCNF debido a la fd  $C \rightarrow B$ .

están restringidas a 1.1, 1.2,..., 1.9 y 2.0 acres. En tal situación tendríamos la dependencia funcional adicional  $FD5 : Area \rightarrow County\_name$ . Si sumamos esto al otro dependencias, el esquema de relación LOTS1A todavía está en 3NF porque este fd se ajusta a cláusula (b) en la definición general de 3NF, siendo *County\_name* un atributo principal.

El área de un lote que determina el condado, según lo especificado por *FD5*, se puede representar por 16 tuplas en una relación separada *R* (*Area*, *County\_name*), ya que solo hay 16 posiciones *Area* valores (véase la figura 14.13). Esta representación reduce la redundancia de repitiendo la misma información en miles de tuplas LOTS1A. BCNF es un forma normal más fuerte que no permitiría LOTS1A y sugeriría la necesidad de descomposición planteándolo.

**Definición.** Un esquema de relación *R* está en **BCNF** si siempre que un funcional no trivial dependencia  $X \rightarrow A$  se mantiene en *R*, entonces *X* es una superclave de *R*.

La definición formal de BCNF difiere de la definición de 3NF en esa cláusula (b) de 3NF, que permite que fd tengan el RHS como atributo principal, está ausente de BCNF. Eso hace que BCNF sea una forma normal más fuerte en comparación con 3NF. En nuestro examen Por ejemplo, *FD5* viola BCNF en LOTS1A porque *Area* no es una superclave de LOTS1A. Nosotros puede descomponer LOTS1A en dos relaciones BCNF LOTS1AX y LOTS1AY, mostrado en la Figura 14.13 (a). Esta descomposición pierde la dependencia funcional *FD2* porque sus atributos ya no coexisten en la misma relación después de la descomposición.

En la práctica, la mayoría de los esquemas de relación que están en 3NF también están en BCNF. Solo si hay



y siendo A un atributo principal, R estará en 3NF pero no en BCNF. El esquema de relación R que se muestra en la Figura 14.13 (b) ilustra el caso general de tal relación. Tal un fd conduce a una posible redundancia de datos, como ilustramos anteriormente en el caso de FD5 : Área → County\_name .in LOTS1A relación. Idealmente, diseño de bases de datos relacionales debe esforzarse por lograr BCNF o 3NF para cada esquema de relación. Alcanzando lo normal El estado de ización de solo 1NF o 2NF no se considera adecuado, ya que ambos se desarrollaron históricamente han sido formas normales intermedias como trampolines hacia 3NF y BCNF.

14.5.1 Descomposición de relaciones no en BCNF

Como otro ejemplo, considere la Figura 14.14, que muestra una relación ENSEÑAR con el siguientes dependencias:

FD1: {Estudiante, curso} → Instructor  
FD2: 14 Instructor → Curso

Tenga en cuenta que { Student , Course } es una clave candidata para esta relación y que la Las deficiencias que se muestran siguen el patrón de la Figura 14.13 (b), con Estudiante como A, Curso como B, e Instructor como C. Por lo tanto, esta relación está en 3NF pero no en BCNF. Descomposición de este esquema de relación en dos esquemas no es sencillo porque puede ser descompuesto en uno de los tres siguientes pares posibles:

- 1. R1 ( estudiante , instructor) y R2 ( estudiante , curso )
- 2. R1 ( curso , instructor ) y R2 ( curso , estudiante )
- 3. R1 ( instructor , curso ) y R2 ( instructor , estudiante )

Las tres descomposiciones pierden la dependencia funcional FD1. La pregunta entonces se convierte en: ¿Cuál de los tres anteriores es una descomposición deseable? Como señalamos antes (Sección 14.3.1), nos esforzamos por cumplir dos propiedades de descomposición durante

ENSEÑAR		
Estudiante	Curso	Instructor
Narayan	Base de datos	marca
Herrero	Base de datos	Navathe
Herrero	Sistemas operativos	Ammar
Herrero	Teoría	Schulman
Wallace	Base de datos	marca
Wallace	Sistemas operativos	Ahamad
Wong	Base de datos	Omiecinski

Figura 14.14  
Una relación ENSEÑAR que está en 3NF pero no BCNF.

<sup>14</sup> Esta dependencia significa que cada instructor imparte un curso es una limitación para esta aplicación.

## Página 32

490

Capítulo 14 Conceptos básicos de las dependencias funcionales y la normalización para bases de datos relacionales

el proceso de normalización: la propiedad de unión no aditiva y la dependencia funcional propiedad de preservación dency. No somos capaces de satisfacer la dependencia funcional preservación para cualquiera de las descomposiciones BCNF anteriores como se ve arriba; pero debemos cumplir con la propiedad de unión no aditiva. Una prueba simple es útil para probar el binario descomposición de una relación en dos relaciones:

**NJB (Prueba de unión no aditiva para descomposiciones binarias).** Una descomposición  $D = \{R_1, R_2\}$  de  $R$  tiene la propiedad de unión sin pérdida (no aditiva) con respecto a un conjunto de dependencias funcionales  $F$  sobre  $R$  si y solo si

- El FD  $((R_1 \cap R_2) \rightarrow (R_1 - R_2))$  está en  $F^{+15}$ , o
- El FD  $((R_1 \cap R_2) \rightarrow (R_2 - R_1))$  está en  $F^{+}$

Si aplicamos esta prueba a las tres descomposiciones anteriores, encontramos que solo la tercera la descomposición cumple con la prueba. En la tercera descomposición, el  $R_1 \cap R_2$  para lo anterior la prueba es Instructor y  $R_1 - R_2$  es Curso. Porque  $\text{Instructor} \rightarrow \text{Curso}$ , la prueba NJB se satisface y la descomposición no es aditiva. (Se deja como ejercicio para el lector para mostrar que las dos primeras descomposiciones no cumplen con la prueba NJB). la descomposición adecuada de TEACH en relaciones BCNF es:

TEACH1 ( instructor , curso ) y TEACH2 ( instructor , estudiante )

Nos aseguramos de cumplir con esta propiedad, porque la descomposición no aditiva es imprescindible durante la normalización. Debe verificar que esta propiedad se cumple con respecto a nuestros ejemplos de normalización sucesiva informal en las Secciones 14.3 y 14.4 y también por la descomposición de LOTS1A en dos relaciones BCNF LOTS1AX y LOTS1AY .

En general, una relación  $R$  que no está en BCNF se puede descomponer para cumplir con los activa la propiedad de unión mediante el siguiente procedimiento. <sup>16</sup> Descompone  $R$  sucesivamente en un conjunto de relaciones que están en BCNF:

Sea  $R$  la relación no en BCNF, sea  $X \subseteq R$ , y sea  $X \rightarrow A$  la FD que provoca una infracción de BCNF.  $R$  se puede descomponer en dos relaciones:

$R - A$   
 $XA$

Si es  $R - A$  o  $XA$ . no está en BCNF, repita el proceso.

El lector debe verificar que si aplicamos el procedimiento anterior a LOTS1A,

obtenga las relaciones LOTS1AX y LOTS1AY como antes. Del mismo modo, la aplicación de este procedimiento La duración de TEACH da como resultado las relaciones TEACH1 y TEACH2

<sup>15</sup> La notación  $F^+$  se refiere a la cobertura del conjunto de dependencias funcionales e incluye todas las fd implícitas por  $F$ . Se analiza en detalle en la Sección 15.1. Aquí, es suficiente asegurarse de que uno de los dos fd's en realidad es válido para la descomposición no aditiva en  $R_1$  y  $R_2$  para pasar esta prueba.

<sup>16</sup> Tenga en cuenta que este procedimiento se basa en el algoritmo 15.5 del capítulo 15 para producir esquemas BCNF por descomposición de un esquema universal.

#### 14.6 Dependencia multivalor y cuarta forma normal 491

Tenga en cuenta que si designamos ( Estudiante, Instructor ) como clave primaria de la relación ENSEÑAR , el instructor de  $FD \rightarrow$  El curso provoca una dependencia parcial (no completamente funcional) de Curso por parte de esta clave. Este DF puede eliminarse como parte de una segunda normalización. (o mediante una aplicación directa del procedimiento anterior para lograr BCNF) produciendo exactamente las mismas dos relaciones en el resultado. Este es un ejemplo de un caso en el que puede alcanzar el mismo diseño BCNF final a través de rutas alternativas de normalización.

### 14.6 Dependencia multivalor y cuarta forma normal

Considere la relación EMP que se muestra en la figura 14.15 (a). Una tupla en esta relación EMP representa el hecho de que un empleado cuyo nombre es  $Ename$  trabaja en el proyecto cuyo nombre es  $Pname$  y tiene un dependiente cuyo nombre es  $Dname$  . Un empleado puede trabajar en varios proyectos y puede tener varios dependientes, y el empleado los proyectos y los dependientes son independientes entre sí. <sup>17</sup> Para mantener la relación Estado coherente y para evitar cualquier relación espuria entre los dos independientes atributos de  $dent$ , debemos tener una tupla separada para representar cada combinación de un dependiente del empleado y proyecto de un empleado. En el estado de relación que se muestra en Figura 14.15 (a), el empleado de  $Ename$  Smith trabaja en dos proyectos 'X' e 'Y' y tiene dos dependientes 'John' y 'Anna', y por lo tanto hay cuatro tuplas para representar estos hechos juntos. La relación EMP es una relación de **todas las claves** (con clave compuesto por todos los atributos) y, por lo tanto, no tiene fd y, como tal, califica para ser un Relación BCNF. Podemos ver que hay una redundancia obvia en la relación EMP: la información dependiente se repite para cada proyecto y el proyecto la información se repite para cada dependiente.

Como lo ilustra la relación EMP, algunas relaciones tienen restricciones que no pueden ser especificadas como dependencias funcionales y, por lo tanto, no infringen la BCNF. A Para abordar esta situación, se propuso el concepto de dependencia multivalor (MVD) y en base a esta dependencia se definió la cuarta forma normal. Un mas formal La discusión de los MVD y sus propiedades se pospone al Capítulo 15. Las diferencias son una consecuencia de la primera forma normal (1NF) (consulte la Sección 14.3.4), que no permite

baja un atributo en una tupla para tener un conjunto de valores. Si hay más de un atributo multivalor está presente, la segunda opción de normalizar la relación (ver Sección 14.3.4) intro- genera una dependencia multivalor. Informalmente, siempre que dos relaciones independientes 1: N- barcos A: B y A: C se mezclan en la misma relación, R (A, B, C), puede surgir un MVD. <sup>18</sup>

14.6.1 Definición formal de dependencia multivalor

**Definición.** Una dependencia de valores múltiples  $X \twoheadrightarrow Y$  especificada en el esquema de relación R, donde X e Y son ambos subconjuntos de R, especifica la siguiente restricción en cualquier

<sup>17</sup> En un diagrama ER, cada uno se representaría como un atributo multivalor o como un tipo de entidad débil (ver el Capítulo 7).

<sup>18</sup> Este MVD se denota como  $A \twoheadrightarrow B \mid C$ .

(un)	EMP				(C)	SUMINISTRO		
	Ename	Pname	Dname	Nombre de		Part_name	Proj_name	
	Herrero	X	Juan	Herrero		Tornillo	ProjX	
	Herrero	Y	Anna	Herrero		Nuez	ProyY	
	Herrero	X	Anna	Adamsky		Tornillo	ProyY	
	Herrero	Y	Juan	Walton		Nuez	ProjZ	
				Adamsky		Uña	ProjX	
				Adamsky		Tornillo	ProjX	
				Herrero		Tornillo	ProyY	
(segundo)	EMP_PROJECTS		EMP_DEPENDENTS					
	Ename	Pname	Ename	Dname				
	Herrero	X	Herrero	Juan				
	Herrero	Y	Herrero	Anna				
(re)	R 1		R 2		R 3			
	Nombre de	Part_name	Nombre de	Proj_name	Part_name	Proj_name		
	Herrero	Tornillo	Herrero	ProjX	Tornillo	ProjX		
	Herrero	Nuez	Herrero	ProyY	Nuez	ProyY		
	Adamsky	Tornillo	Adamsky	ProyY	Tornillo	ProyY		
	Walton	Nuez	Walton	ProjZ	Nuez	ProjZ		
	Adamsky	Uña	Adamsky	ProjX	Uña	ProjX		

Figura 14.15  
Cuarta y quinta formas normales.

(b) Descomponer la relación EMP en dos relaciones 4NF EMP\_PROJECTS y EMP\_DEPENDENTS.

(c) La relación SUMINISTRO sin MVD está en 4NF pero no en 5NF si tiene el JD (R<sub>1</sub>, R<sub>2</sub>, R<sub>3</sub>).

(d) Descomponer la relación OFERTA en las relaciones 5NF R<sub>1</sub>, R<sub>2</sub>, R<sub>3</sub>.

estado de relación r de R: Si dos tuplas t<sub>1</sub> y t<sub>2</sub> existen en r tales que t<sub>1</sub>[X] = t<sub>2</sub>[X], entonces dos tuplas t<sub>3</sub> y t<sub>4</sub> también deberían existir en r con las siguientes propiedades, <sup>19</sup> donde usamos Z para denotar (R - (X ∪ Y)):

$$\blacksquare t_3[X] = t_4[X] = t_1[X] = t_2[X]$$

$$\blacksquare t_3[Y] = t_1[Y] \text{ y } t_4[Y] = t_2[Y]$$

$$\blacksquare t_3[Z] = t_2[Z] \text{ y } t_4[Z] = t_1[Z]$$

<sup>19</sup> Las tuplas t<sub>1</sub>, t<sub>2</sub>, t<sub>3</sub> y t<sub>4</sub> no son necesariamente distintas.

<sup>20</sup> Z es la abreviatura de los atributos en R después de que los atributos en (X ∪ Y) se eliminan de R.

Siempre que  $X \twoheadrightarrow Y$  se cumple, decimos que X **multidetermina** Y. Debido a la simetría intente en la definición, siempre que  $X \twoheadrightarrow Y$  se mantenga en R, también lo hace  $X \twoheadrightarrow Z$ . Por lo tanto,  $X \twoheadrightarrow Y$  implica  $X \twoheadrightarrow Z$  y, por lo tanto, a veces se escribe como  $X \twoheadrightarrow Y \mid Z$ .

Un MVD  $X \twoheadrightarrow Y$  en R se denomina **MVD trivial** si (a) Y es un subconjunto de X, o (b)  $X \cup Y = R$ .

Por ejemplo, la relación EMP\_PROJECTS en la Figura 14.15 (b) tiene la trivial

MVD Ename  $\twoheadrightarrow$  Pname y la relación EMP\_DEPENDENTS tiene el MVD trivial

Ename  $\twoheadrightarrow$  Dname. Un MVD que no satisface ni (a) ni (b) se denomina **no trivial**

**MVD**. Un MVD trivial se mantendrá en cualquier estado de relación r de R; se llama trivial porque no especifica ninguna restricción significativa o significativa sobre R.

Si tenemos un MVD no trivial en una relación, es posible que tengamos que repetir valores redundantly en las tuplas. En la relación EMP de la Figura 14.15 (a), los valores 'X' e 'Y' de

Pname se repiten con cada valor de Dname (o, por simetría, los valores 'John'

y 'Anna' de Dname se repiten con cada valor de Pname). Esta redundancia es

claramente indeseable. Sin embargo, el esquema EMP está en BCNF porque no funciona

las dependencias se mantienen en EMP. Por lo tanto, necesitamos definir una cuarta forma normal que

es más fuerte que BCNF y no permite esquemas de relación como EMP. Darse cuenta de

Las relaciones que contienen MVD no triviales tienden a ser **relaciones clave**, es decir, su

La clave son todos sus atributos tomados en conjunto. Además, es raro que tales claves

relaciones con una ocurrencia combinatoria de valores repetidos se diseñarían en

práctica. Sin embargo, el reconocimiento de los MVD como una posible dependencia problemática es

esencial en el diseño relacional.

Presentamos ahora la definición de **cuarta forma normal (4NF)**, que se viola

cuando una relación tiene dependencias multivalor indeseables y, por tanto, se puede utilizar para identificar y descomponer tales relaciones.

**Definición.** Un esquema de relación  $R$  está en 4 **NF** con respecto a un conjunto de dependencias  $F$  (que incluye dependencias funcionales y dependencias multivalor) si, para cada dependencia multivalor no trivial  $X \twoheadrightarrow Y$  en  $F$ , <sup>21</sup>  $X$  es una superclave para  $R$ .

Podemos enunciar los siguientes puntos:

- Una relación de todas las claves siempre está en BCNF ya que no tiene FD.
- Una relación de todas las claves como la relación EMP en la Figura 14.15 (a), que no tiene FD pero tiene la MVD  $Ename \twoheadrightarrow Pname \mid Dname$ , no está en 4NF.
- Una relación que no está en 4NF debido a un MVD no trivial debe descomponerse para convertirlo en un conjunto de relaciones en 4NF.
- La descomposición elimina la redundancia causada por el MVD.

El proceso de normalización de una relación que involucra a los MVD no triviales que no están en 4NF consiste en descomponerlo para que cada MVD esté representado por una relación separada donde se convierte en un MVD trivial. Considere la relación EMP en la figura 14.15 (a). EMP es no en 4NF porque en los MVD no triviales  $Ename \twoheadrightarrow Pname$  y  $Ename \twoheadrightarrow Dname$ ,

<sup>21</sup>  $F^+$  se refiere a la cobertura de las dependencias funcionales  $F$ , o todas las dependencias que están implícitas en  $F$ . Esto es definido en la Sección 15.1.

y  $Ename$  no es una superclave de EMP. Descomponemos EMP en EMP\_PROJECTS y EMP\_DEPENDENTS, que se muestra en la Figura 14.15 (b). Ambos EMP\_PROJECTS y EMP\_DEPENDENTS están en 4NF, porque MVDs  $Ename \twoheadrightarrow Pname$  in EMP\_PROJECTS y  $Ename \twoheadrightarrow Dname$  en EMP\_DEPENDENTS son MVD triviales. No otros MVD no triviales se mantienen en EMP\_PROJECTS o EMP\_DEPENDENTS. No Los DF también se mantienen en estos esquemas de relación.

## 14.7 Unir dependencias y quinta forma normal

En nuestra discusión hasta ahora, hemos señalado la dependencia funcional problemática y mostró cómo fueron eliminadas por un proceso de descomposición binaria repetida posición durante el proceso de normalización para lograr 1NF, 2NF, 3NF y BCNF. Estas descomposiciones binarias deben obedecer a la propiedad NJB para la que introducimos una prueba en la Sección 14.5 mientras se analiza la descomposición para lograr BCNF. Lograr ing 4NF normalmente implica la eliminación de MVD mediante descomposiciones binarias repetidas como bien. Sin embargo, en algunos casos puede que no haya una descomposición de unión no aditiva de  $R$  en dos esquemas de relación, pero puede haber una descomposición de combinación no aditiva en más de dos esquemas de relación. Además, puede no haber dependencia funcional

en R que viola cualquier forma normal hasta BCNF, y puede que no haya nada trivial MVD presente en R ya sea que viole 4NF. Luego recurrimos a otra dependencia

llamada dependencia de unión y, si está presente, lleve a cabo una descomposición de múltiples vías en la quinta forma normal (5NF). Es importante señalar que tal dependencia es una restricción semántica peculiar que es difícil de detectar en la práctica; por lo tanto, normalmente, en la práctica, la conversión en 5NF rara vez se realiza.

**Definición.** Una **dependencia de unión** ( **JD** ), denotada por  $JD(R_1, R_2, \dots, R_n)$ , especificada en el esquema de relación R, especifica una restricción en los estados r de R. La restricción establece que cada estado legal r de R debe tener una descomposición de unión no aditiva en  $R_1, R_2, \dots, R_n$ . Por lo tanto, para cada r tenemos

$$* (\pi_{R_1}(r), \pi_{R_2}(r), \dots, \pi_{R_n}(r)) = r$$

Observe que un MVD es un caso especial de un JD donde  $n = 2$ . Es decir, un JD denotado como  $JD(R_1, R_2)$  implica un MVD  $(R_1 \cap R_2) \rightarrow \rightarrow (R_1 - R_2)$  (o, por simetría,  $(R_1 \cap R_2) \rightarrow \rightarrow (R_2 - R_1)$ ). Una dependencia de unión  $JD(R_1, R_2, \dots, R_n)$ , especificada en la relación el esquema R, es un **JD trivial** si uno de los esquemas de relación  $R_i$  en  $JD(R_1, R_2, \dots, R_n)$  es igual a R. Tal dependencia se llama trivial porque tiene la propiedad de unión no aditiva para cualquier estado de relación r de R y, por lo tanto, no especifica ninguna restricción sobre R. Podemos ahora definir la quinta forma normal, que también se llama forma normal de unión de proyecto.

**Definición.** Un esquema de relación R está en la **quinta forma normal (5NF)** (o **proyecto-unión forma normal (PJNF)**) con respecto a un conjunto F de funcional, multivalor y

Unir dependencias si, para cada dependencia de unión no trivial,  $JD(R_1, R_2, \dots, R_n)$  en  $F^+$  (es decir, implícito por F), <sup>22</sup> cada  $R_i$  es una superclave de R.

<sup>22</sup> Nuevamente,  $F^+$  se refiere a la cobertura de las dependencias funcionales F, o todas las dependencias que están implícitas en F. Esto se define en la Sección 15.1.

Para ver un ejemplo de un JD, considere una vez más la relación de todas las teclas SUMINISTRO en la Figura 14.15 (c). Suponga que siempre se cumple la siguiente restricción adicional: Siempre que un proveedor s suministra la parte p, y un proyecto j utiliza la parte p, y los suministros del proveedor en al menos una parte al proyecto j, entonces el proveedor s también suministrará la parte p al proyecto j. Esta restricción se puede reformular de otras formas y especifica una dependencia de unión  $JD(R_1, R_2, R_3)$  entre las tres proyecciones  $R_1(Sname, Part\_name)$ ,  $R_2(Sname, Proj\_name)$  y  $R_3(Part\_name, Proj\_name)$  de SUPPLY. Si esta restricción se mantiene, la tuplas debajo de la línea discontinua en la Figura 14.15 (c) deben existir en cualquier estado legal de la SUPPLY relación que también contiene las tuplas por encima de la línea discontinua. Figura 14.15 (d) muestra cómo la relación SUPPLY con la dependencia de unión se descompone en tres relaciones  $R_1, R_2$  y  $R_3$  que están cada una en 5NF. Observe que aplicar una combinación natural a dos de estas relaciones producen tuplas falsas, pero aplicando una combinación natural a los tres juntos no lo hace. El lector debe verificar esto en la relación de muestra en Figura 14.15 (c) y sus proyecciones en la Figura 14.15 (d). Esto se debe a que solo el JD

existe, pero no se especifican. MYD. Observe también que el JD  $(R_1, R_2, R_3)$  se especifica en todos los estados de relación jurídica, no solo el que se muestra en la Figura 14.15 (c).

Descubrir JDs en bases de datos prácticas con cientos de atributos es casi imposible. Solo se puede hacer con un alto grado de intuición sobre los datos de la pieza. del diseñador. Por lo tanto, la práctica actual de diseño de bases de datos presta poca atención. a ellos. Un resultado de Date y Fagin (1992) se relaciona con las condiciones detectadas usando fd's solo e ignora los JD por completo. Dice: "Si un esquema de relación está en 3NF y cada una de sus claves consta de un solo atributo, también está en 5NF".

## 14.8 Resumen

En este capítulo discutimos varias trampas en el diseño de bases de datos relacionales usando intuiciones argumentos interesantes. Identificamos informalmente algunas de las medidas para indicar si un esquema de relación es bueno o malo, y proporcionamos pautas informales para un buen diseño. Estas pautas se basan en realizar un cuidadoso diseño conceptual en el modelo ER y EER, siguiendo el procedimiento de mapeo en el Capítulo 9 para mapear entrelazos y relaciones en relaciones. Aplicación adecuada de estas pautas y La falta de redundancia evitará la inserción / eliminación / actualización de anomalías y generación de datos falsos. Recomendamos limitar los valores NULL, que causan problemas lemas durante las operaciones SELECT, JOIN y agregación. Luego presentamos algunos conceptos formales que nos permiten hacer un diseño relacional de arriba hacia abajo por analizar las relaciones individualmente. Definimos este proceso de diseño por análisis y descomposición introduciendo el proceso de normalización.

Definimos el concepto de dependencia funcional, que es la herramienta básica para analizar analizando esquemas relacionales, y discutimos algunas de sus propiedades. Funcional las dependencias especifican restricciones semánticas entre los atributos de una relación esquema. A continuación describimos el proceso de normalización para lograr buenos diseños. probando relaciones para tipos indeseables de dependencias funcionales problemáticas. Proporcionamos un tratamiento de normalización sucesiva basado en un primary Mary Key en cada relación, y luego relajamos este requisito y proporcionamos más

definiciones generales de segunda forma normal (2NF) y tercera forma normal (3NF) que tener en cuenta todas las claves candidatas de una relación. Presentamos ejemplos para ilustrar tratar cómo, utilizando la definición general de 3NF, se puede analizar una relación dada y descompuesto para finalmente producir un conjunto de relaciones en 3NF.

Presentamos la forma normal de Boyce-Codd (BCNF) y discutimos cómo es una forma más fuerte forma de 3NF. También ilustramos cómo la descomposición de una relación no BCNF debe hacerse considerando el requisito de descomposición no aditiva. Nosotros pre envió una prueba para la propiedad de unión no aditiva de las descomposiciones binarias y también dio un algoritmo general para convertir cualquier relación que no esté en BCNF en un conjunto de BCNF



relaciones. Motivamos la necesidad de una restricción adicional más allá de lo funcional dependencias basadas en la mezcla de atributos multivalor independientes en un solo relación. Introdujimos la dependencia multivalor (MVD) para abordar tales condiciones y definió la cuarta forma normal basada en MVD. Finalmente, presentamos la quinta forma normal, que se basa en la dependencia de unión y que identifica una restricción peculiar que hace que una relación se descomponga en varias componentes para que siempre produzcan la relación original después de una combinación. En la práctica, la mayoría los diseños comerciales han seguido las formas normales hasta BCNF. La necesidad de descomponerse en 5NF rara vez surge en la práctica, y las dependencias de unión son difíciles de detectar para la mayoría de situaciones prácticas, lo que hace que 5NF tenga más valor teórico.

El capítulo 15 presenta algoritmos de síntesis y descomposición para diseño de base de datos basado en dependencias funcionales. Relacionado con la descomposición, discutir los conceptos de unión no aditiva (o sin pérdida) y preservación de la dependencia, que se aplican mediante algunos de estos algoritmos. Otros temas en el Capítulo 15 incluyen un tratamiento más detallado de las dependencias funcionales y multivalor, y otras tipos de dependencias.

## Preguntas de revisión

- 14.1. Discuta la semántica de atributos como una medida informal de bondad para una relación esquema de
- 14.2. Analice las anomalías de inserción, eliminación y modificación. ¿Por qué están considerado malo? Ilustre con ejemplos.
- 14.3. ¿Por qué deben evitarse los valores NULL en una relación tanto como sea posible? Discutir el problema de las tuplas falsas y cómo podemos prevenirlo.
- 14.4. Enuncie las pautas informales para el diseño de esquemas de relación que discutimos. Ilustre cómo la violación de estas pautas puede ser perjudicial.
- 14.5. ¿Qué es una dependencia funcional? ¿Cuáles son las posibles fuentes de información que define las dependencias funcionales que se mantienen entre los atributos de un esquema de relación?
- 14.6. ¿Por qué no podemos inferir automáticamente una dependencia funcional de una estado de relación ular?

- 14.7. ¿A qué se refiere el término relación no normalizada? ¿Cómo surgieron las formas normales desarrollarse históricamente desde la primera forma normal hasta la forma normal de Boyce-Codd?
- 14.8. Defina la primera, segunda y tercera formas normales cuando solo se considerado. ¿Cómo se aplican las definiciones generales de 2NF y 3NF, que consideran todas las claves de una relación, ¿difieren de aquellas que consideran solo claves primarias?

- 14.9. ¿Qué dependencias indeseables se evitan cuando una relación está en 2NF?
- 14.10. ¿Qué dependencias indeseables se evitan cuando una relación está en 3NF?
- 14.11. ¿De qué manera las definiciones generalizadas de 2NF y 3NF extienden la definición niciones más allá de las claves primarias?
- 14.12. Defina la forma normal de Boyce-Codd. ¿En qué se diferencia de 3NF? Porque es considerado una forma más fuerte de 3NF?
- 14.13. ¿Qué es la dependencia multivalor? ¿Cuándo surge?
- 14.14. ¿Una relación con dos o más columnas siempre tiene un MVD? Mostrar con un ejemplo.
- 14.15. Defina cuarta forma normal. ¿Cuándo se viola? ¿Cuándo suele aplicarse?
- 14.16. Defina la dependencia de unión y la quinta forma normal.
- 14.17. ¿Por qué 5NF también se llama forma normal de unión de proyecto (PJNF)?
- 14.18. ¿Por qué los diseños prácticos de bases de datos suelen tener como objetivo BCNF y no formas normales superiores?

## Ejercicios

- 14.19. Supongamos que tenemos los siguientes requisitos para una base de datos universitaria que se utiliza para realizar un seguimiento de las transcripciones de los estudiantes:
- La universidad realiza un seguimiento del nombre de cada estudiante ( Sname ), número de estudiante ( Snum ), número de seguro social ( Ssn ), dirección actual ( Sc\_addr ) y teléfono ( Sc\_phone ), dirección permanente ( Sp\_addr ) y teléfono ( Sp\_phone ), fecha de nacimiento ( Bdate ), sexo ( Sexo ), clase ( Clase ) ('freshman', 'sophomore',..., 'graduado'), departamento principal ( Major\_code ), departamento menor ( Minor\_code ) (si lo hay) y programa de grado ( Prog ) ('ba', 'bs',..., 'ph.d.'). Tanto el Ssn como el número de estudiante tienen valores únicos para cada estudiante.
  - Cada departamento se describe con un nombre ( Dname ), código de departamento ( Dcode ), número de oficina ( Doffice ), teléfono de la oficina ( Dphone ) y universidad ( Dcollege ). Tanto el nombre como el código tienen valores únicos para cada departamento.
  - Cada curso tiene un nombre de curso ( Cname ), descripción ( Cdesc ), curso número ( Cnum ), número de horas semestrales ( crédito ), nivel ( nivel ) y departamento de ofrendas ( Cdept ). El número de curso es único para cada curso.

curso ( Sec\_course ) y número de sección ( Sec\_num ). El número de sección distingue diferentes secciones del mismo curso que se imparten durante el mismo semestre / año; sus valores son 1, 2, 3,..., hasta el número total de secciones enseñadas durante cada semestre.

mi. Un registro de calificaciones se refiere a un estudiante ( Ssn ), una sección en particular y un grado ( grado ).

Diseñe un esquema de base de datos relacional para esta aplicación de base de datos. Primera show, primera emision todas las dependencias funcionales que deben tener entre los atributos. Luego diseñar esquemas de relación para la base de datos que están en 3NF o BCNF. Especificaciones-identificar los atributos clave de cada relación. Tenga en cuenta los requisitos no especificados, y hacer suposiciones apropiadas para completar la especificación.

14.20. ¿Qué anomalías de actualización ocurren en las relaciones EMP\_PROJ y EMP\_DEPT de Figuras 14.3 y 14.4?

14.21. ¿En qué forma normal se encuentra el esquema de relación LOTS de la figura 14.12 (a) con respecto a las interpretaciones restrictivas de la forma normal que toman sólo el clave principal en cuenta? ¿Sería en la misma forma normal si la generación ¿Se utilizaron definiciones generales de forma normal?

14.22. Demuestre que cualquier esquema de relación con dos atributos está en BCNF.

14.23. ¿Por qué ocurren tuplas falsas en el resultado de unir EMP\_PROJ1 y Relaciones EMP\_LOCS en la Figura 14.5 (el resultado se muestra en la Figura 14.6)?

14.24. Considere la relación universal  $R = \{A, B, C, D, E, F, G, H, I, J\}$  y el conjunto de dependencias funcionales  $F = \{ \{A, B\} \rightarrow \{C\}, \{A\} \rightarrow \{D, E\}, \{B\} \rightarrow \{F\}, \{F\} \rightarrow \{G, H\}, \{D\} \rightarrow \{I, J\} \}$ . ¿Cuál es la clave de R? Descomponer R en 2NF y luego las relaciones 3NF.

14.25. Repita el ejercicio 14.24 para el siguiente conjunto diferente de dependencias funcionales.  $G = \{ \{A, B\} \rightarrow \{C\}, \{B, D\} \rightarrow \{E, F\}, \{A, D\} \rightarrow \{G, H\}, \{A\} \rightarrow \{I\}, \{H\} \rightarrow \{J\} \}$ .

14.26. Considere la siguiente relación:

UN	segundo	C	TUPLE #
10	b1	c1	1
10	b2	c2	2
11	b4	c1	3
12	b3	c4	4
13	b1	c1	5
14	b3	c4	6

a. Dada la extensión anterior (estado), ¿cuál de las siguientes dependencias pueden tener en la relación anterior? Si la dependencia no puede sostenerse, explique por qué especificando las tuplas que causan la infracción.

yo.  $A \rightarrow B$ , ii.  $B \rightarrow C$ , iii.  $C \rightarrow B$ , iv.  $B \rightarrow A$ , v.  $C \rightarrow A$

segundo. ¿La relación anterior tiene una clave candidata potencial? Si lo hace, que es eso? Si no es así, ¿por qué no?

14.27. Considere una relación R (A, B, C, D, E) con las siguientes dependencias:

$$AB \rightarrow C, CD \rightarrow E, DE \rightarrow B$$

¿AB es una clave candidata de esta relación? Si no es así, ¿es ABD? Explica tu respuesta.

14.28. Considere la relación R, que tiene atributos que contienen horarios de cursos y secciones en una universidad; R = { Course\_no , Sec\_no , Offering\_dept , Credit\_hours , Course\_level , Instructor\_ssn , Semester , Year , Days\_hours , Room\_no , No\_of\_students }. Suponga que las siguientes dependencias funcionales se mantienen en R:

$$\begin{aligned} \{ \text{Course\_no} \} &\rightarrow \{ \text{Offer\_dept}, \text{Credit\_hours}, \text{Course\_level} \} \\ \{ \text{Course\_no}, \text{Sec\_no}, \text{Semester}, \text{Year} \} &\rightarrow \{ \text{Days\_hours}, \text{Room\_no}, \\ &\quad \text{No\_of\_students}, \text{Instructor\_ssn} \} \\ \{ \text{Room\_no}, \text{Days\_hours}, \text{Semester}, \text{Year} \} &\rightarrow \{ \text{Instructor\_ssn}, \text{Course\_no}, \\ &\quad \text{Sec\_no} \} \end{aligned}$$

Intente determinar qué conjuntos de atributos forman las claves de R. ¿Cómo normalizar esta relación?

14.29. Considere las siguientes relaciones para los datos de una aplicación de procesamiento de pedidos: base en ABC, Inc.

ORDER (O #, Odate, Cust #, Total\_amount)  
ORDER\_ITEM (O #, I #, Qty\_ordered, Total\_price, Discount%)

Suponga que cada artículo tiene un descuento diferente. El precio\_total se refiere a uno artículo, Odate es la fecha en la que se realizó el pedido, y Total\_amount es el monto del pedido. Si aplicamos una combinación natural en las relaciones ORDER\_ITEM y ORDER en esta base de datos, ¿qué significa la relación resultante esquema RES parece? ¿Cuál será su clave? Muestre los FD en este resultado relación. ¿Está RES en 2NF? ¿Está en 3NF? ¿Por qué o por qué no? (Indique las suposiciones, si haces alguno.)

14.30. Considere la siguiente relación:

CAR\_SALE (N.º de automóvil, Fecha\_vendido, N.º de vendedor, % de comisión, Descuento\_amt)

Suponga que varios vendedores pueden vender un automóvil y, por lo tanto, { Car # , El vendedor # } es la clave principal. Las dependencias adicionales son

$$\begin{aligned} \text{Date\_sold} &\rightarrow \text{Discount\_amt} \text{ y} \\ \text{Vendedor \#} &\rightarrow \text{Comisión\%} \end{aligned}$$

Según la clave principal dada, ¿esta relación es 1NF, 2NF o 3NF? Por qué o por que no ¿Cómo lo normalizarías sucesivamente por completo?

14.31. Considere la siguiente relación para los libros publicados:

LIBRO (Título del libro, Nombre del autor, Tipo de libro, Precio de lista, Autor\_affil, Editor)

Author\_affil se refiere a la afiliación del autor. Suponga la siguiente dependencia existencial:

Book\_title  $\rightarrow$  Publisher , Book\_type  
 Tipo\_libro  $\rightarrow$  Precio\_lista  
 Author\_name  $\rightarrow$  Author\_affil

a. ¿En qué forma normal está la relación? Explique su respuesta.

segundo. Aplique la normalización hasta que no pueda descomponer más las relaciones. Indique las razones detrás de cada descomposición.

14.32. Este ejercicio le pide que convierta los extractos comerciales en dependencias.

Considere la relación DISK\_DRIVE ( Serial\_number , Manufacturer , Model , Batch , Capacidad , Minorista ). Cada tupla en la relación DISK\_DRIVE contiene información sobre una unidad de disco con una única Núm\_de\_serie , hecho por un fabricante, con un número de modelo particular, lanzado en un lote determinado, que tiene un cierto almacenamiento capacidad de edad y lo vende un minorista determinado. Por ejemplo, la tupla Disk\_drive ('1978619', 'WesternDigital', 'A2235X', '765234', 500, 'CompUSA') especifica que WesternDigital hizo una unidad de disco con el número de serie 1978619 y el modelo número A2235X, lanzado en el lote 765234; tiene 500 GB y lo vende CompUSA.

Escriba cada una de las siguientes dependencias como un FD:

a. El fabricante y el número de serie identifican de forma exclusiva la unidad.

segundo. Un número de modelo está registrado por un fabricante y, por lo tanto, no puede ser utilizado por otro fabricante.

c. Todas las unidades de disco de un lote en particular son del mismo modelo.

re. Todas las unidades de disco de un determinado modelo de un fabricante en particular tienen exactamente la misma capacidad.

14.33. Considere la siguiente relación:

R ( N.º de médico , N.º de paciente , Fecha , Diagnóstico , Código de tratamiento , Cargo )

En la relación anterior, una tupla describe una visita de un paciente a un médico junto con código de tratamiento y carga diaria. Suponga que el diagnóstico está determinado (únicamente) para cada paciente por un médico. Suponga que cada código de tratamiento Tiene un cargo fijo (independientemente del paciente). ¿Está esta relación en 2NF? Justifique su respuesta y descomponer si es necesario. Luego discuta si una mayor normalización es necesaria la conexión a 3NF y, de ser así, hágalo.

14.34. Considere la siguiente relación:

CAR\_SALE ( Car\_id , Option\_type , Option\_listprice , Sale\_date , Option\_discountedprice )

Esta relación se refiere a las opciones instaladas en automóviles (por ejemplo, control de crucero) que fueron vendidos en un concesionario, y la lista y los precios con descuento de las opciones.

Si CarID  $\rightarrow$  Sale\_date y Option\_type  $\rightarrow$  Option\_listprice y CarID , Option\_type  $\rightarrow$  Option\_discountedprice , argumenta usando la definición generalizada de 3NF

que esta relación no está en 3NF. Luego argumente a partir de su conocimiento de 2NF, por qué ni siquiera está en 2NF.

14.35. Considere la relación:

LIBRO ( Nombre\_libro , Autor , Edición , Año )

con los datos:

Nombre del libro	Autor	Edición	Copyright_Year
DB_fundamentals	Navathe	4	2004
DB_fundamentals	Elmasri	4	2004
DB_fundamentals	Elmasri	5	2007
DB_fundamentals	Navathe	5	2007

a. Basado en una comprensión de sentido común de los datos anteriores, ¿cuáles son los posibles claves candidatas de esta relación?

segundo. Justifique que esta relación tiene el MVD  $\{ \text{Libro} \} \twoheadrightarrow \{ \text{Autor} \} \mid \{ \text{Edición}, \text{año} \}$ .

c. ¿Cuál sería la descomposición de esta relación en base a lo anterior?  
MVD? Evaluar cada relación resultante para la forma normal más alta posee.

14.36. Considere la siguiente relación:

VIAJE ( Trip\_id , Start\_date , Cities\_visited , Cards\_used )

Esta relación se refiere a los viajes de negocios realizados por los vendedores de la empresa. Suponer el VIAJE tiene una única fecha de inicio, pero involucra a muchas ciudades y vendedores puede usar varias tarjetas de crédito en el viaje. Haga una población de maquetas de la mesa.

a. Discuta qué FD y / o MVD existen en esta relación.

segundo. Muestre cómo hará para normalizar la relación.

## Ejercicios de laboratorio

Nota: Los siguientes ejercicios usan el sistema DBD (Data Base Designer) que es descrito en el manual de laboratorio.

El esquema relacional R y el conjunto de dependencias funcionales F deben codificarse como liza. Como ejemplo, R y F para este problema se codifican como:

R = [a, segundo, c, re, e, f, g, h, i, j]  
 F = [[a, b], [c]],  
     [[a], [d, e]],  
     [[b], [f]],  
     [[f], [g, h]],  
     [[d], [i, j]]]

Dado que DBD está implementado en Prolog, el uso de términos en mayúsculas está reservado para variables en el lenguaje y, por lo tanto, las constantes en minúsculas se utilizan para codificar el atributo. Para obtener más detalles sobre el uso del sistema DBD, consulte el laboratorio manual.

14.37. Usando el sistema DBD, verifique sus respuestas a los siguientes ejercicios:

a. 14.24 (solo 3NF)

segundo. 14.25

c. 14.27

re. 14.28

## Bibliografía seleccionada

Las dependencias funcionales fueron introducidas originalmente por Codd (1970). El original Las definiciones de primera, segunda y tercera forma normal también se definieron en Codd (1972a), donde se puede encontrar una discusión sobre anomalías de actualización. Boyce-Codd normal La forma mal se definió en Codd (1974). La definición alternativa de tercera normal La forma se da en Ullman (1988), al igual que la definición de BCNF que damos aquí. Ullman (1988), Maier (1983) y Atzeni y De Antonellis (1993) contienen muchos de los teoremas y demostraciones relativas a las dependencias funcionales. Fecha y Fagin (1992) dan algunos resultados simples y prácticos relacionados con formas normales superiores.

En el Capítulo 15 se ofrecen referencias adicionales a la teoría del diseño relacional.