Bases de Datos I
14/10/2020
Resumen Cap 14 14/10
IS-501
José Inestroza

Capítulo 14 Fundamentals

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

Cada esquema de relación consta de varios atributos, y el esquema de base de datos relacional consta de varios esquemas de relación. Se necesita alguna forma formal de analizar por qué una agrupación de atributos en un esquema de relación puede ser mejor que otra, 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. El primero es el nivel lógico (o conceptual): cómo los usuarios interpretan los esquemas de relación y el significado de sus atributos. Tener buenos esquemas de relaciones en este nivel permite a los usuarios comprender claramente el significado de los datos en las relaciones y, por lo tanto, formular sus consultas correctamente. El segundo es el nivel de implementación (o almacenamiento físico): cómo se almacenan y actualizan las tuplas en una relación base.

Este nivel se aplica sólo a los esquemas de las relaciones base, que se almacenarán físicamente como archivos, mientras que en el nivel lógico nos interesan los esquemas tanto de las relaciones base como de las vistas (relaciones virtuales). 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 ascendente (también llamada diseño por síntesis) considera las relaciones básicas entre atributos individuales como punto de partida y las utiliza para construir esquemas de relaciones. Para situaciones prácticas, es casi imposible capturar las relaciones binarias entre todos esos 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 juntas de forma natural, El diseño de bases de datos relacionales finalmente produce un conjunto de relaciones. Los objetivos implícitos de la actividad de diseño son la conservación 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 relaciones, así como las relaciones de generalización / especialización, que se describen utilizando un modelo como el modelo EER. Por lo tanto, el diseño relacional debe preservar todos estos conceptos, que se capturaron originalmente en el diseño conceptual después del mapeo del diseño conceptual al lógico. Minimizar la redundancia implica minimizar el almacenamiento redundante de la misma información y reducir la necesidad de múltiples actualizaciones para mantener la coherencia entre múltiples copias de la misma información en respuesta a eventos del mundo real que requieren realizar una actualización.
Las formas normales sucesivas se definen para cumplir con un conjunto de restricciones deseables expresadas mediante claves primarias y dependencias funcionales. El procedimiento de normalización consiste en aplicar una serie de pruebas a las relaciones para cumplir con estos requisitos cada vez más estrictos y descomponer las relaciones cuando sea necesario.

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 pautas informales que pueden usarse como medidas para determinar la calidad del diseño de esquemas de relaciones:

- 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 los atributos que pertenecen a una relación tienen cierto significado del mundo real y una interpretación adecuada asociada con ellos. En general, cuanto más fácil sea explicar la semántica de la relación, o en otras palabras, qué significa y representa exactamente una relación, mejor será el diseño del esquema de relación. 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.

Directriz 1. Diseñar un esquema de relación para que sea fácil de explicar su significado. No combine atributos de varios tipos de entidades y tipos de relaciones en una sola relación. Intuitivamente, si un esquema de relación corresponde a un tipo de entidad o un tipo de relación, es sencillo explicar su significado. De lo contrario, si la relación corresponde a una mezcla de múltiples entidades y relaciones, se producirán ambigüedades semánticas y la relación no podrá explicarse fácilmente.

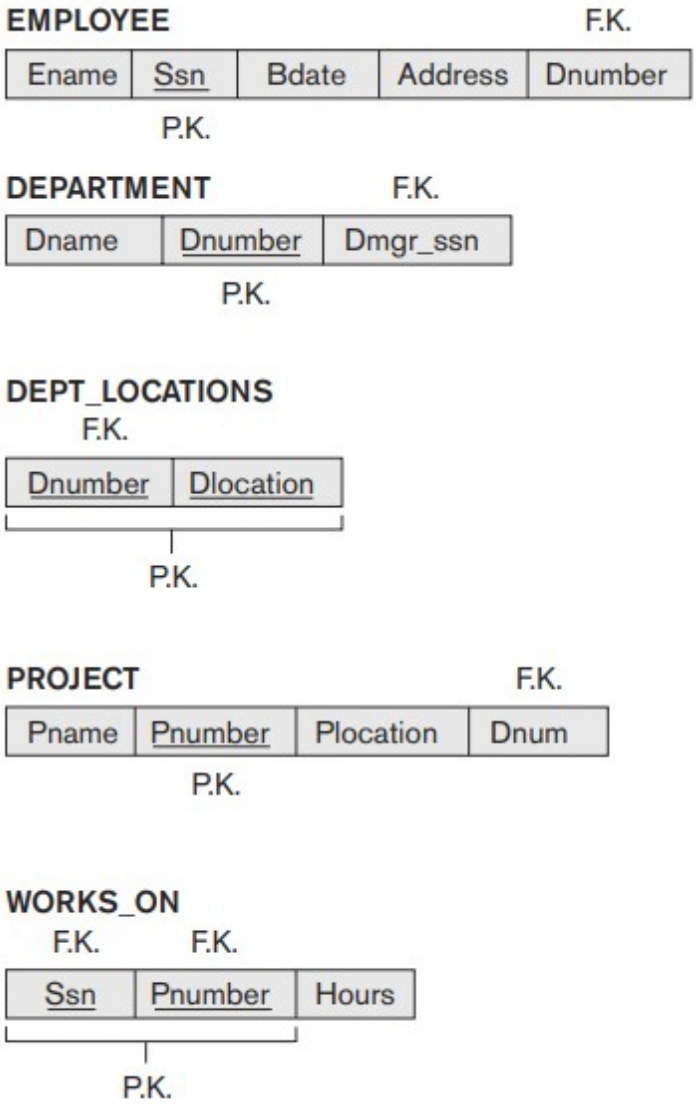
14.1.2 Información redundante en tuplas y anomalías de actualización

Uno de los objetivos del diseño de esquemas es minimizar el espacio de almacenamiento utilizado por las relaciones base (y, por tanto, los archivos correspondientes). La agrupación de atributos en esquemas de relación tiene un efecto significativo en el espacio de almacenamiento. El almacenamiento de combinaciones naturales de relaciones base genera un problema adicional denominado anomalías de actualización. Estos pueden clasificarse en anomalías de inserción, anomalías de eliminación y anomalías de modificación. Anomalías de inserción. **Las anomalías de inserción** se pueden diferenciar en dos tipos, ilustrados por los siguientes ejemplos basados en la relación EMP_DEPT:

- Para insertar una nueva tupla de empleado en EMP_DEPT, debemos incluir los valores de atributo para el departamento para el que trabaja el empleado o NULL (si el empleado aún no trabaja para un departamento).
- Es difícil insertar un nuevo departamento que todavía no tiene empleados en la relación EMP_DEPT. La única forma de hacer esto es colocar valores NULL en los atributos de empleado. Esto viola la integridad de la entidad para EMP_DEPT porque su clave principal Ssn no puede ser nula. Además, cuando se asigna el primer empleado a ese departamento, ya no necesitamos esta tupla con valores NULL.

Anomalías de supresión. El problema de las anomalías de eliminación está relacionado con la segunda situación de anomalía de inserción que se acaba de discutir. Si eliminamos de EMP_DEPT una tupla de empleados que representa al último empleado que trabaja para un departamento en particular, la información relativa a ese departamento se pierde inadvertidamente de la base de datos.

Figure 14.1
A simplified COMPANY relational database schema.



Anomalías de modificación. En EMP_DEPT, si cambiamos el valor de uno de los atributos de un departamento en particular, digamos, el gerente del departamento 5, debemos actualizar las tuplas de todos los empleados que trabajan en ese departamento; de lo contrario, la base de datos se volverá inconsistente.

Directriz 2. Diseñar los esquemas de relación base de modo que no existan anomalías de inserción, eliminación o modificación en las relaciones. Si se presentan anomalías, anótelas claramente y asegúrese de que los programas que actualizan la base de datos funcionarán correctamente. Es importante tener en cuenta que, en ocasiones, es posible que sea necesario infringir estas directrices para mejorar el rendimiento de determinadas consultas.

En general, se recomienda utilizar relaciones base libre de anomalías y especificar vistas que incluyan las combinaciones para colocar juntos los atributos a los que se hace referencia con frecuencia en consultas importantes.

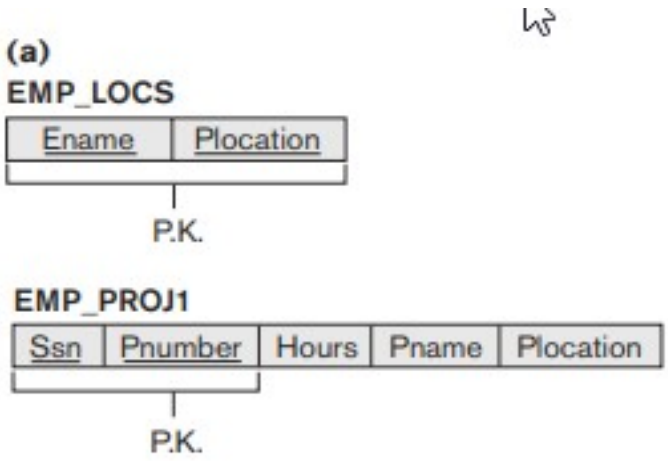
14.1.3 Valores NULL en tuplas

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

de los atributos y para especificar operaciones JOIN en el nivel lógico. Otro problema con los NULL es cómo contabilizarlos cuando se agregan operaciones como COUNT o SUM se aplican. Las operaciones SELECT y JOIN implican comparaciones; si hay valores NULL presentes, los resultados pueden volverse impredecibles. Además, los NULL pueden tener múltiples interpretaciones, como las siguientes:
- El atributo no se aplica a esta tupla. Por ejemplo, Visa_status puede no aplicarse a estudiantes de EE. UU.
- Se desconoce el valor del atributo para esta tupla. Por ejemplo, la fecha de nacimiento puede ser desconocida para un empleado.
- El valor es conocido pero ausente; es decir, aún no se ha registrado. Por ejemplo, es posible que exista el número de teléfono de la casa de un empleado, pero es posible que aún no esté disponible ni registrado.

Tener la misma representación para todos los NULL compromete los diferentes significados que pueden tener. Por tanto, planteamos otra pauta.

Directriz 3. En la medida de lo posible, evite colocar atributos en una relación base cuyos valores puedan ser NULL con frecuencia. Si los NULL son inevitables, asegúrese de que se apliquen solo en casos excepcionales y no se apliquen 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 una relación o tener una relación separada para esas columnas (con las columnas clave apropiadas).



14.1.4 Generación de tuplas espurias

Considere los dos esquemas de relación EMP_LOCS y EMP_PROJ1
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 por semana determinadas en el proyecto cuyo nombre, número y ubicación son Pname, Pnumber y Plocation. Las tuplas adicionales que no estaban en EMP_PROJ se denominan tuplas falsas porque representan información falsa que no es válida.

Directriz 4. Diseñar esquemas de relaciones para que puedan unirse con condiciones de igualdad en atributos que estén apropiadamente relacionados (clave primaria, clave externa) pares de manera que garantice que no se generen tuplas falsas. Evite las relaciones que contengan atributos coincidentes que no sean combinaciones (clave externa, clave primaria) porque la unión de dichos atributos puede producir tuplas falsas.

14.1.5 Resumen y discusión de las pautas de diseño

Discutimos informalmente situaciones que conducen a esquemas de relación problemáticos y propusimos pautas informales para un buen diseño relacional. Los problemas que señalamos, que se pueden detectar sin herramientas de análisis adicionales, son los siguientes:

- Anomalías que causan que se realice trabajo redundante durante la inserción y modificación de una relación, y que pueden causar la pérdida accidental de información durante la eliminación de una relación.
- Desperdicio de espacio de almacenamiento debido a NULL y la dificultad de realizar 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 relación adecuada (clave externa, clave primaria).

14.2 Dependencias funcionales

La Herramienta formal para el análisis de esquemas relacionales que nos permite detectar y describir algunos de los problemas antes mencionados en términos precisos. El concepto más importante en la teoría del diseño de esquemas relacionales es el de dependencia funcional.

14.2.1 Definición de dependencia funcional

Una dependencia funcional es una restricción entre dos conjuntos de atributos de la base de datos. Suponga que nuestro esquema de base de datos relacional tiene n atributos A1, A2,…, An; pensemos en toda la base de datos como descrita por un único esquema de relación universal R = {A1, A2,…, An} .7 No implicamos que realmente almacenaremos la base de datos como una única tabla universal; utilizamos este concepto solo para desarrollar la teoría formal de las dependencias de datos. Definición. Una dependencia funcional, denotada por X → Y, entre dos conjuntos de atributos X e Y que son subconjuntos de R especifica una restricción sobre las posibles tuplas que pueden formar un estado de relación r de R. La restricción es que, para dos tuplas cualesquiera t1 y t2 en r que tienen t1 [X] = t2 [X], también deben tener t1 [Y] = t2 [Y]. Esto significa que los valores del componente Y de una tupla en r dependen de, o están determinados por, los valores del componente X; alternativamente, los valores del componente X de una tupla determinan de forma única (o funcional) los valores del componente Y. También decimos 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 f.d. El conjunto de atributos X se llama el lado izquierdo del FD e Y se llama el lado derecho. Por tanto, X determina funcionalmente Y en un esquema de relación R si, y solo si, siempre que dos tuplas de r (R) concuerden en su valor X, necesariamente deben coincidir 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 valor X dado en cualquier instancia de relación r (R), es decir, X es una clave candidata de R, esto implica que X → Y para cualquier subconjunto de los atributos Y de R (porque la restricción clave implica que no hay dos tuplas en ningún estado legal r (R) que tengan el mismo valor de X). Si X es una clave candidata de R, entonces X → R.
- Si X → Y en R, esto no dice si Y → X en R.

Una dependencia funcional es una propiedad de la semántica o significado de los atributos. Los diseñadores de la base de datos usarán su comprensión de la semántica de los atributos de R, es decir, cómo se relacionan entre sí, para especificar las dependencias funcionales que deben mantenerse en todos los estados de relación

(extensiones) r de R. Extensiones de relación r (R) que satisfacen las restricciones de dependencia funcional se denominan estados de relación legal (o extensiones legales) de R. Por lo tanto, el uso principal de las dependencias funcionales es describir más a fondo un esquema de relación R especificando restricciones sobre sus atributos que deben cumplirse en todo momento.

Una dependencia funcional es una propiedad del esquema de relación R, no de un estado de relación legal particular r de R. Por lo tanto, una FD no puede inferirse automáticamente de una extensión de relación dada r, sino que debe ser definida explícitamente por alguien que conozca la semántica de la relación.

14.3 Formas normales basadas en claves primarias

Suponemos que se da un conjunto de dependencias funcionales para cada relación, y que cada relación tiene una clave primaria designada; esta información combinada 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 adoptan uno de los dos enfoques siguientes:

- 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 en base al conocimiento externo derivado de una implementación existente de archivos o formularios o informes.

14.3.1 Normalización de relaciones

El proceso de normalización, propuesto por primera vez por Codd (1972a), toma un esquema de relación a través de una serie de pruebas para certificar si satisface una determinada forma normal.
El proceso, que procede de forma descendente evaluando cada relación con los criterios de las formas normales y descomponiendo las relaciones según sea necesario, puede considerarse así como un diseño relacional por análisis. Inicialmente, Codd propuso tres formas normales, a las que llamó primera, segunda y tercera forma normal. Más tarde, Boyce y Codd propusieron una definición más estricta de 3NF, llamada forma normal de Boyce-Codd (BCNF). Todas estas formas normales se basan en una única herramienta analítica: las dependencias funcionales entre los atributos de una relación.
Posteriormente, se propuso una cuarta forma normal (4NF) y una quinta forma normal (5NF), basadas en los conceptos de dependencias multivalor y dependencias de unión, respectivamente.

La normalización de datos puede considerarse un proceso de análisis de los esquemas de relación dados en función de sus FD y claves primarias para lograr las propiedades deseables de (1) minimizar la redundancia y (2) minimizar las anomalías de inserción, eliminación y actualización, el procedimiento de normalización proporciona a los diseñadores de bases de datos lo siguiente:

- Un marco formal para analizar esquemas de relaciones basados en sus claves y en las dependencias funcionales entre sus atributos.
- Una serie de pruebas de forma normal que se pueden realizar en esquemas de relaciones individuales para que la base de datos relacional se pueda normalizar al grado deseado.

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

normalización a través de la descomposición también debe confirmar la existencia de propiedades adicionales que los esquemas relacionales, tomados en conjunto, deberían poseer. Estos incluirían dos propiedades:

- La propiedad de unión no aditiva o unión sin pérdida, que garantiza que el problema de generación de tuplas espúreas no ocurre con respecto a los esquemas de relación creados después de la descomposición.
- La propiedad de conservación de la dependencia, que asegura que cada dependencia funcional esté representada en alguna relación individual resultante después de la descomposición

La propiedad de unión no aditiva es extremadamente crítica y debe lograrse a cualquier costo, mientras que la propiedad de conservación de la dependencia, aunque deseable, a veces se sacrifica.

14.3.2 Uso práctico de formas normales

El diseño de bases de datos, tal como se practica en la industria actual, presta especial 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 a la forma normal más alta posible. Las relaciones pueden dejarse en un estado de normalización más bajo, como 2NF, por razones de rendimiento.

La desnormalización es el proceso de almacenar la unión de relaciones de forma normal superior como una relación base, que se encuentra en una forma normal inferior.

14.3.3 Definiciones de claves y atributos que participan en claves

Una superclave de un esquema de relación R = {A1, A2,…, An} es un conjunto de atributos S ⊆ R con la propiedad de que no hay dos tuplas t1 y t2 en ningún estado de relación legal r de R t1 [S] = t2 [S]. **Una clave** K es una superclave con la propiedad adicional de que la eliminación de cualquier atributo de K hará que K deje de ser una superclave.

La diferencia entre una clave y una superclave es que una clave debe ser mínima; es decir, si tenemos una clave K = {A1, A2,…, Ak} de R, entonces K - {Ai} no es una clave de R para cualquier Ai, $1 \leq i \leq k$.
Si un esquema de relación tiene más de una clave, cada una se denomina clave candidata. Una de las claves candidatas se designa arbitrariamente como clave principal y las otras se denominan claves secundarias. **Un atributo** del esquema de relación R se llama atributo principal de R si es miembro de alguna clave candidata de R. 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.

14.3.4 Primera forma normal

La primera forma normal (1NF) ahora se considera parte de la definición formal de una relación en el modelo relacional básico (plano); históricamente, se definió para no permitir atributos multivalor, atributos compuestos y sus combinaciones. Establece 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 lo 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 únicos valores de atributo permitidos por 1NF son valores atómicos únicos (o indivisibles). La primera forma normal también rechaza atributos multivalor que son compuestos en sí mismos. Se denominan relaciones anidadas porque cada tupla puede tener una relación dentro de ella. Este procedimiento se puede aplicar de forma recursiva a una relación con anidamiento de múltiples niveles para desanidar la relación en un conjunto de relaciones 1NF. Esto es útil para convertir un esquema de relación no normalizado con muchos niveles de anidamiento en relaciones 1NF.

14.3.5 Segunda forma normal

La segunda forma normal (2NF) se basa en el concepto de dependencia funcional completa. Una dependencia funcional X → Y es una dependencia funcional completa si la eliminación de cualquier atributo A de X significa que la dependencia ya no se mantiene; es decir, para cualquier atributo A ε X, (X - {A}) no determina funcionalmente Y. Una dependencia funcional X → Y es una dependencia parcial si algún atributo A ε X puede eliminarse de X y la dependencia aún se mantiene; es decir, para algunos A ε X, (X - {A}) → Y.

Un esquema de relación R está en 2NF si cada atributo no principal A en R es completamente funcionalmente dependiente de la clave primaria de R. La prueba para 2NF implica probar las dependencias funcionales cuyos atributos del lado izquierdo son parte de la clave primaria. Si la clave principal contiene un solo atributo, no es necesario aplicar la prueba.

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 primarios se asocian solo con la parte de la clave primaria de la que dependen totalmente funcionalmente.

14.3.6 Tercera forma normal

La tercera forma normal (3NF) se basa en el concepto de dependencia transitiva. Una dependencia funcional X → Y en un esquema de relación R es una dependencia transitiva si existe un conjunto de atributos Z en R que no es una clave candidata ni un subconjunto de cualquier clave de R, 11 y tanto X → Z como Z → Y sostener.

De acuerdo con la definición original de Codd, un esquema de relación R está en 3NF si satisface 2NF y ningún atributo no primario de R depende transitivamente de la clave primaria.

La normalización 2NF y 3NF eliminan estos FD problemáticos al descomponer la relación original en nuevas relaciones. En términos del proceso de normalización, no es necesario eliminar las dependencias parciales antes de las dependencias transitivas, pero históricamente, 3NF se ha definido con el supuesto de que una relación se prueba para 2NF primero antes de probarla para 3NF. Además,
La definición general de 3NF cubre automáticamente la condición de que la relación también satisfaga 2NF.

14.4 Definiciones generales de segunda y tercera formas normales

En general, queremos diseñar nuestros esquemas de relación para que no tengan dependencias parciales ni transitivas porque estos tipos de dependencias causan las anomalías de actualización,

Como definición general de atributo principal, un atributo que forma parte de cualquier clave candidata se considerará principal. Las 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

Un esquema de relación R está en la **segunda forma normal (2NF)** si cada atributo no principal A en R no depende parcialmente de ninguna clave de R. La prueba para 2NF implica probar las dependencias funcionales cuyos atributos del lado izquierdo son parte de la clave principal. Si la clave principal contiene un solo atributo, no es necesario aplicar la prueba.

Forma Normal	Prueba	Normalización
Primera forma normal (1FN)	La relación no debe tener atributos de valores múltiples ni relaciones anidadas.	Forme nuevas relaciones para cada atributo multivalor o relación anidada.
Segunda forma normal (2FN)	Para las relaciones en las que la clave principal contiene varios atributos, ningún atributo que no sea de clave debe ser funcionalmente dependiente de una parte de la clave principal.	Descomponga y configure una nueva relación para cada clave parcial con sus atributos dependientes. Asegúrese de mantener una relación con la clave primaria original y cualquier atributo que sea funcionalmente dependiente de ella.
Tercera forma normal (3FN)	La relación no debe tener un atributo no clave determinado funcionalmente por otro atributo no clave (o por un conjunto de atributos no clave). Es decir, no debería haber dependencia transitiva de un atributo que no sea de clave en la clave principal.	Descomponer y establecer una relación que incluya los atributos no clave que determinan funcionalmente otros atributos no clave.

14.4.1 Definición general de la segunda forma normal

Definición. Un esquema de relación R está en la segunda **forma normal (2NF)** si cada atributo no principal A en R no depende parcialmente de ninguna clave de R. La prueba para 2NF implica probar las dependencias funcionales cuyos atributos del lado izquierdo son parte de la clave principal. Si la clave principal contiene un solo atributo, no es necesario aplicar la prueba.

14.4.2 Definición general de la tercera forma normal

Definición. Un esquema de relación R está en la tercera forma normal (3NF) si, siempre que se cumple una dependencia funcional no trivial $X \rightarrow A$ en R, (a) X es una superclave de R, o (b) A es un atributo primo de R.13

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 se cumple una dependencia funcional X → A 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 normalmente 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 lo tanto, la condición (a) por sí sola aborda las dependencias problemáticas que fueron causas de la segunda y tercera normalización, como ya 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 que ciertas dependencias funcionales se escapen o se escapen en el sentido de que están de acuerdo con la definición de 3NF y, por lo tanto, no son "capturadas" por la definición de 3NF, aunque pueden ser potencialmente problemáticas. La forma normal de Boyce-Codd "captura" estas dependencias en el sentido de que no las permite.