

@author: aar.velasquez@gmail.com

@date: 2020/10/19

Capítulo 14: Conceptos básicos de Dependencias funcional y normalización para base de datos relacionales.

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 1 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 pares de atributos.

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.

14.1 Pautas de diseño informal para esquemas de relaciones

Impartir una semántica clara a los atributos en las relaciones

La semántica de una relación se refiere a su significado. ing resultante de la interpretación de valores de atributo en una tupla.

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.

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 deleción alteraciones y anomalías de modificación. 2

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 .
```

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.

Si no actualizamos algunas tuplas, el mismo departamento Se mostrará que el ment tiene dos valores diferentes para el gerente en diferentes empleados. tuplas, lo cual estaría mal.

podemos enunciar la siguiente directriz de la siguiente manera:

```
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, 4
anótelos claramente y asegúrese de que los programas que actualizan la base
de datos
funcionar correctamente.
```

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.

los valores NULL pueden tener múltiples interpretaciones ciones, 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
Por ejemplo, el Home_Phone_Number de un empleado puede existir, pero no
estar disponible y grabado todavía.
```

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.

Generación de tuplas espurias

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.

Resumen y discusión de las pautas 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 sy la dificultad de realizar la selección ciones, 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

14.2 Dependencias funcionales

Una dependencia funcional es una restricción entre dos conjuntos de atributos del base de datos.

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 $t_1[Y] = t_2[Y]$.

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 el
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 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.

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 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.

Normalización de relaciones

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 cation" para hacer que el diseño tenga una calidad
sucesivamente mejor.

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.

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.

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

Uso práctico de formas normales

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.

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.

Definiciones de claves y atributos que participan en claves

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.

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.

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.

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.
- no permite relaciones dentro de relaciones o relaciones como valores de atributo dentro de tuplas.
- Sólo los valores de atributos permitidos por 1NF son individuales atómicos (o indivisibles) valores.

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 $Pnumber \rightarrow Horas$ retenidas). Sin embargo, la dependencia $\{Ssn, Pnumber\} \rightarrow Ename$ es

parcial porque $Ssn \rightarrow Ename$ se mantiene.

- 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 tercera forma normal (3NF)

se basa en el concepto de dependencia transitiva.

Una función dependencia $X \rightarrow Y$ 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 ninguna clave de R, 11 y tanto $X \rightarrow Z$ como $Z \rightarrow Y$ se mantienen.

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.

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

1NF

prueba

La relación no debe tener valores múltiples
atributos o relaciones anidadas.

Solución

Forme nuevas relaciones para cada multivalor
atributo o relación anidada.

2NF

prueba

Para relaciones donde la clave principal
contiene múltiples atributos, sin clave
atributo debe ser funcionalmente
depende de una parte de la clave principal.

Solución

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.

3NF

Prueba 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 haber dependencia transitiva de una no clave atributo en la clave principal.

Solución

Descomponer y establecer una relación que incluye los atributos no clave que determinan funcionalmente otros no clave atributo (s).

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 . 12

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.

Definición general de la tercera forma normal

Definición. Un esquema de relación R está en tercera forma normal (3NF) si, siempre que una 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 . 13

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.

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.