BDI Tema 8. Normalización del Modelo Relacional

Martín González Dios

7 de noviembre de 2024

1. Introducción

- **Normalización**: técnica para producir un conjunto de relaciones con una serie de propiedades a partir de unos requisitos de datos. Esta técnica es de abajo a arriba. Se basa en aplicar al conjunto de relaciones una serie de pruebas ordenadas.
- **Dependencia funcional**: son relaciones entre valores de conjuntos de atributos. Las dependencias funcionales entre atributos sirven como una forma de representar la semántica que se incorpora a la base de datos.
- **Formas Normales**: una serie de condiciones o pruebas formales que deben cumplirse sucesivamente en el proceso de normalización de una base de datos. Alcanzar niveles más altos de Formas Normales asegura una mejor organización y eficiencia de la base de datos. Una base de datos en Tercera Forma Normal o en Forma Normal de Boyce-Codd es considerada razonablemente correcta.

2. Buenos Diseños Relacionales

Los buenos diseños relaciones tienen que tener las siguientes características:

- Atributos mínimos
- Mínima redundancia
- Agrupación temática máxima en las tablas

Las dependencias funcionales ayudan a definir relaciones clave entre atributos. Por ejemplo, la dependencia nombre_dept $\rightarrow$ presupuesto indica que cada departamento tiene un único presupuesto.
Usar **dependencias funcionales** permite reconocer cuando un esquema contiene datos redundantes y cuándo es necesario dividirlo para mejorar la organización de la base de datos.

La **descomposición** consiste en dividir un esquema grande en varios esquemas más pequeños para reducir redundancias.
Descomposición sin pérdidas: es ideal porque permite recuperar toda la información original sin perder integridad ni crear datos incorrectos.
Descomposición con pérdidas: es problemática, ya que puede dar lugar a resultados incorrectos o pérdida de información, como ocurre en el caso de empleados con nombres iguales.

Para que la descomposición sea válida es necesaria una combinación sin pérdidas y una conservación de las dependencias.

3. Primera Forma Normal

Primera Forma Normal (1FN): relación en la que la intersección de toda fila y columna contiene un valor atómico. (todos los atributos son atómicos). Esta primera forma normal es obligatoria en el MR.
Relación universal: una única tabla con todos los atributos de interés de la BD de un MR.

4. Descomposición mediante Dependencias Funcionales

Definiciones importantes:

- **Dependencia funcional**: describe la relación existente entre atributos de una relación. Si A y B son conjuntos de atributos de la relación R, B será funcionalmente dependiente de A (A→B) si cada valor de A está asociado con exactamente un valor de B. La parte izquierda de una dependcia funcional se denomina **determinante**.
- **Determinante**: grupo de atributos del lado izquierdo de una Dependencia Funcional.
- **Dependencia Funcional Completa**: si A y B son conjuntos de atributos de la relación R, B depende funcionalmente de manera completa de A si B depende funcionalmente de A pero no de ningún subconjunto de A.
- **Dependencia parcial**: dependencia funcional en la cual existe algún atributo que puede eliminarse del determinante y la dependencia continua verificándose.
- **Dependencia transitiva**: si A, B y C son conjuntos de atributos de la relación R, C depende transitivamente de A a través de B si C depende funcionalmente de B y B depende funcionalmente de A.

Ejemplos de dependencias funcionales serían el dni de una persona y su nombre (si en dos tuplas de una tabla apareciera el mismo dni, debería aparecer el mismo nombre), la matrícula de un coche y su modelo (si en dos tuplas de una tabla se repitiera el valor de la matrícula, debería repetirse el modelo del coche).

- La **Primera Forma Normal** prohíbe los atributos compuestos y/o multivalorados.
- La **Segunda Forma Normal** impide la existencia de dependencias funcionales parciales (dependencias en las cuales el lado izquierdo, α, sea una parte de una clave candidata pero no la clave candidata completa). Es decir, todo atributo que no sea de una clave candidata depende funcionalmente de forma completa de cualquier clave candidata.
- La **Tercera Forma Normal** impide la existencia de dependencias funcionales transitivas (dependencias en las cuales el lado izquierdo, α, no forme parte de una clave candidata).
- La **Forma Normal de Boyce-Codd** obliga a que en todas las dependencias funcionales no triviales, el lado izquierdo, α, sea una clave candidata de la relación.

Un ejemplar de relación que satisfaga todas las relaciones del mundo real de este tipo se llama ejemplar legal de relación; un **ejemplar legal** de una base de datos es aquel en el que todos los ejemplares de relación son ejemplares legales.

Cuando un ejemplar **satisface** una dependencia funcional $\alpha \rightarrow \beta$, significa que, en ese ejemplar, si dos tuplas tienen los mismos valores en los atributos de α, también deben tener los mismos valores en los atributos de β.

Decimos que una dependencia funcional se **cumple** en un esquema de relación si todos los ejemplares legales de esa relación satisfacen dicha dependencia. Esto implica que la dependencia no solo es cierta para un conjunto específico de datos, sino que es una restricción intrínseca del esquema, por lo que todos los datos futuros también deberían cumplirla.

Las **dependencias funcionales triviales** son aquellas que son automáticamente ciertas para todos los posibles ejemplares de una relación, sin importar los datos específicos. Estas dependencias cumplen la condición $\beta \subseteq \alpha$. Un ejemplo de dependencia funcional trivial es $A \rightarrow A$: siempre se cumple que si dos tuplas tienen el mismo valor para AA, entonces tendrán el mismo valor en A. Otra forma trivial es $AB \rightarrow A$, ya que cualquier conjunto que contenga A en su parte izquierda siempre garantiza que el valor de A será el mismo en ambas tuplas si coinciden en todos los atributos de la izquierda.

El **cierre** del conjunto F es el conjunto de todas las dependencias funcionales que pueden inferirse dado el conjunto F. F+ (el cierre del conjunto F) contiene todas las dependencias funcionales de F.

5. Teoría de las Dependencias Funcionales

5.1. Dependencia funcional implicada lógicamente

Una dependencia funcional f de R **está implicada lógicamente por un conjunto de dependencias funcionales F si**, cada vez que F se cumple en una relación, también se cumple f. En otras palabras, dado un esquema relacional r(R) y un conjunto de dependencias funcionales F, decimos que f está implicada lógicamente por F si cualquier relación que satisface todas las dependencias en F también satisface f.

Por **ejemplo**, en un esquema con atributos A, B, C, G, H, I y el conjunto de dependencias:
$A \to B$
$A \to C$
$CG \to H$
$CG \to I$
$B \to H$

Podemos demostrar que $A \to H$ está implicada lógicamente. Si dos tuplas tienen el mismo valor en A, y dado que $A \to B$ y $B \to H$, entonces ambas tuplas deben tener el mismo valor en H. Así, podemos deducir que $A \to H$ también se cumple.

5.2. Cierre de un conjunto de dependencias funcionales

El **cierre** de un conjunto de dependencias funcionales F, denotado como F^+, es el conjunto de todas las dependencias funcionales que **están implicadas lógicamente por F**. Este cierre contiene todas las dependencias que se derivan de las originales en F.

Se dice que un atributo B está **determinado funcionalmente** por α si $\alpha \to B$.

5.3. Axiomas de Armstrong y reglas adicionales

Axiomas de Armstrong:

- **Regla de Relfexividad**: si α es un conjunto de atributos y $\beta \subseteq \alpha$, entonces se cumple que $\alpha \to \beta$.
- **Regla de la aumentatividad**: si se cumple que $\alpha \to \beta$ y γ es un conjunto de atributos, entonces se cumple que $\gamma\alpha \to \gamma\beta$.
- **Regla de la transitividad**: si se cumple que $\alpha \to \beta$ y que $\beta \to \gamma$, entonces se cumple que $\alpha \to \gamma$.

Reglas adicionales que se derivan de los axiomas de Armstrong:

- **Regla de la unión**: Si se cumple que $\alpha \to \beta$ y que $\alpha \to \gamma$, entonces se cumple que $\alpha \to \beta\gamma$.
- **Regla de la descomposición**: si se cumple que $\alpha \to \beta\gamma$, entonces se cumple que $\alpha \to \beta$ y que $\alpha \to \gamma$.
- **Regla de la pseudotransitividad**: si se cumple que $\alpha \to \beta$ y que $\gamma\beta \to \delta$, entonces se cumple que $\alpha\gamma \to \delta$.

5.4. Recubrimiento canónico

El recubrimiento canónico es un **conjunto simplificado de dependencias funcionales** de un esquema de relación en una base de datos. La finalidad de este conjunto es garantizar que las dependencias funcionales del esquema se cumplan sin tener que evaluar dependencias innecesarias, lo cual facilita la verificación de restricciones en las actualizaciones de la base de datos.
Para que sea efectivo debe cumplir las siguientes propiedades:

- **Mismo cierre**: debe tener el mismo cierre que el conjunto original de dependencias funcionales F. Esto significa que cualquier dependencia funcional que sea cierta en el conjunto original F también debe ser cierta en el recubrimiento canónico F_c, y viceversa.
- **Minimización de atributos**: no debe contener atributos innecesarios (o "**raros**") en el lado izquierdo o derecho de las dependencias funcionales.
- **Unicidad en el lado izquierdo**: no debe haber dos dependencias funcionales en el recubrimiento canónico que tengan el mismo conjunto de atributos en el lado izquierdo.

Un **atributo raro** es aquel que se puede eliminar de una dependencia funcional sin cambiar el cierre del conjunto de dependencias. En otras palabras, si un atributo es raro en una dependencia funcional, su eliminación no afecta las dependencias que el sistema de bases de datos debe garantizar.

5.5. Descomposición sin pérdidas

Una descomposición sin pérdidas es aquella en la que **podemos reconstruir la relación original** a partir de sus partes sin perder datos. Si dividimos la relación original R en dos relaciones R1 y R2, la descomposición es sin pérdidas si: $\pi_{R1}(r) \bowtie \pi_{R2}(r) = r$
Esto significa que si proyectamos (o seleccionamos) los atributos correspondientes de R1 y R2 de la relación r, y luego hacemos una reunión natural de estas proyecciones, obtenemos exactamente el conjunto de tuplas originales de r.
Para asegurar que la descomposición sea sin pérdidas, al menos una de las siguientes condiciones debe cumplirse para las dependencias funcionales en R:

- $R1 \cap R2 \rightarrow R1$
- $R1 \cap R2 \rightarrow R2$

Una **descomposición con pérdidas** ocurre cuando, al reconstruir la relación original, obtenemos un conjunto de datos más amplio o incompleto respecto al original. En una descomposición con pérdidas, no es posible reconstruir con exactitud la relación original a partir de sus partes.

5.6. Conservación de las dependencias

En el contexto de dependencias funcionales, decimos que una descomposición conserva las dependencias cuando **todas las dependencias funcionales del esquema original se pueden verificar en las subrelaciones resultantes de la descomposición**, sin necesidad de recomponer los datos. (En cualquiera de las relaciones resultantes, no se tiene que cumpliar la dependencia en todas las relaciones)

El **cierre de un conjunto de atributos**, digamos α, **bajo un conjunto de dependencias funcionales F**, representa todos los atributos que pueden ser determinados a partir de los atributos de α usando las dependencias de F. Este cierre se denota como α^+.

Para calcular el cierre de un conjunto de atributos α bajo F, seguimos estos pasos:

1. **Inicialización**: Comienza el cierre con el conjunto inicial de atributos, es decir, $\alpha^+ = \alpha$.
2. **Aplicación de las dependencias**: Recorremos las dependencias funcionales en F. Si una dependencia funcional $X \to Y$ cumple que $X \subseteq \alpha^+$, añadimos Y al cierre α^+ porque los atributos de Y pueden ser determinados a partir de α usando la dependencia $X \to Y$.
3. **Repetición**: Repetimos el paso anterior hasta que no haya cambios en α^+; es decir, cuando ya no se puedan añadir más atributos al cierre.

Al finalizar, α^+ contendrá todos los atributos que pueden ser inferidos o determinados a partir de α usando F.

6. Algoritmo de Bernstein

Permite descomponer relaciones para eliminar redundancias y cumplir las formas normales, garantizando que los datos se almacenen sin anomalias. Aquí te explico brevemente cada paso:

1. **Análisis semántico**: El primer paso es entender los atributos de la relación que estamos estudiando, creando la **relación universal** U, que contiene todos los atributos relevantes. Además, se debe identificar el conjunto de **dependencias funcionales** F, que define cómo los valores de ciertos atributos dependen de otros en la relación.

2. **Primera Forma Normal (1FN)**: Aquí se eliminan atributos **compuestos o multivalorados**. Esto implica que cada atributo contenga un solo valor por fila, asegurando que todos los datos son atómicos.

3. **Recubrimiento Canónico del conjunto** F: Este paso simplifica el conjunto de dependencias funcionales.

   - **Descomposición**: Aplicando las reglas de Armstrong, descomponemos dependencias funcionales complejas en dependencias más simples.
   - **Eliminación de dependencias redundantes**: Quitamos aquellas que son **implicadas lógicamente** por otras, reduciendo redundancia.
   - **Unión**: Reagrupamos dependencias funcionales cuando sea posible, combinando aquellas que tienen el mismo atributo en la parte izquierda de la dependencia.

4. **Cálculo de claves candidatas y verificación de BCNF**: Se calculan las **claves candidatas** de la relación (conjuntos mínimos de atributos que determinan todos los demás). Esto permite verificar si la relación cumple la **Forma Normal de Boyce-Codd (BCNF)**. Si no la cumple, se procede con los siguientes pasos.

5. **Descomposición por grupos de atributos equivalentes**: Identificamos grupos de atributos que dependen unos de otros y los dividimos en nuevas relaciones independientes, de manera que cada uno mantenga sus dependencias funcionales y minimice la redundancia.

6. **Descomposición por eliminación de dependencias funcionales**: Finalmente, eliminamos las dependencias funcionales que violen las condiciones de la BCNF, asegurando que cada relación esté en la forma más alta posible y se eviten dependencias no deseadas entre atributos.

Este algoritmo lleva a las relaciones a una estructura normalizada en la que las dependencias funcionales están mejor organizadas, eliminando redundancias y facilitando la integridad de los datos.

7. Otras formas normales

- **Dependencia multivalorada**: dependencia entre conjuntos de atributos A, B y C con valores independientes de modo que para cada valor de A hay un conjunto de valores de B y un conjunto de valores de C.
- **Relación en Cuarta Forma Normal (4FN)**: relación que está en FNBC y no contiene dependencias multivaluadas.
- **Relación en Quinta Forma Normal (5FN)**: relación que no presenta dependencias de combinación (propiedad de la descomposición de relaciones que garantiza que no se generan tuplas incorrectas al volver a combinar relaciones de una descomposición mediante una operación de reunión natural).