

Profundización en Conceptos Avanzados del Modelo Relacional

Tiempo estimado: 10 minutos

Objetivo

Al final de esta lectura, usted:

1. Entenderá las dependencias funcionales
2. Aprenderá sobre las dependencias multivaluadas
3. Se enfocará en las claves candidatas
4. Practicará el concepto en un escenario práctico

En el mundo de la gestión de bases de datos, es muy importante comprender algunos conceptos avanzados para hacer que las bases de datos funcionen bien. Tres de los más importantes son las dependencias funcionales, las dependencias multivaluadas y las claves candidatas. Estos son como los bloques de construcción para organizar y acelerar las bases de datos.

1. Dependencias Funcionales (FDs)

Las FDs son bloques de construcción fundamentales para garantizar la integridad y consistencia de los datos en bases de datos relacionales. Representan un tipo específico de relación entre atributos en una relación, donde el valor de un atributo (determinante) determina de manera única el valor de otro atributo (dependiente). En términos más simples, si conoces el valor del determinante, siempre puedes identificar el valor exacto del atributo dependiente.

- **Por ejemplo**, si conoces el número de identificación de un empleado, puedes buscarlo en la tabla y averiguar su nombre y apellido. En este caso, el ID del empleado “determina” el nombre y apellido. Esta es una dependencia funcional. Piensa en ello como un tipo especial de conexión entre piezas de información. El atributo determinante actúa como una llave que desbloquea la información relacionada.

Propiedades de las FDs:

- **Notación:** Las FDs se escriben típicamente como $X \rightarrow Y$, donde X es el determinante y Y es el atributo dependiente.
- **Propiedades:**
 - Reflexividad: $X \rightarrow X$ (cada atributo se determina a sí mismo).
 - Transitividad: Si $X \rightarrow Y$ y $Y \rightarrow Z$, entonces $X \rightarrow Z$ (las dependencias pueden encadenarse).
 - Cerradura: El conjunto mínimo de FDs que implica todas las demás FDs en una relación.
- **Puntos clave**
 - Las FDs son esenciales para mantener la precisión de los datos y minimizar la redundancia.
 - Juegan un papel crucial en la normalización de bases de datos, asegurando la organización eficiente de los datos.
 - Las FDs ayudan a eliminar repeticiones innecesarias y a asegurar la corrección de los datos.

2. Dependencias Multivaluadas (MVDs)

Las MVDs son más complejas que las FD. Con una MVD, un atributo, el determinante, determina un conjunto de valores posibles para otro atributo, el dependiente. En otras palabras, conocer el valor del determinante reduce los posibles valores que el atributo dependiente puede tener.

Así es como funciona:

- Si conoces el ID de un empleado (digamos ID 123), no puedes simplemente buscarlo y encontrar su único proyecto asignado.
- Con MVD, conocer el ID 123 te dice que habrá múltiples filas en la tabla, cada una listando un proyecto diferente en el que trabaja el empleado 123.
- Así, el ID del empleado “multi-determina” el proyecto porque no te da una única respuesta, sino que te informa que hay potencialmente varias asignaciones de proyectos para ese empleado en filas separadas. Esto es diferente de una dependencia regular, donde una pieza de información conduce a una o dos específicas.

Propiedades de los MVDs:

- **Notación:** Los MVDs se escriben como $X \twoheadrightarrow \{Y_1, Y_2, \dots, Y_n\}$, donde X es el determinante y Y_1, Y_2, \dots, Y_n son los posibles valores para el atributo dependiente.
- **Propiedades:**
 - Los MVDs tienen propiedades similares a los FDs, como la reflexividad y la transitividad. Sin embargo, carecen de la propiedad de cierre.
 - Identificar MVDs ayuda a entender las relaciones complejas entre atributos y asegura la consistencia de los datos dentro de esas relaciones. Las violaciones de los MVDs pueden llevar a entradas de datos inválidas.
- **Puntos clave:**
 - Los MVDs son esenciales para organizar relaciones complejas entre conjuntos de atributos.
 - Ayudan a evitar confusiones y asegurar la correcta organización de los datos.
 - Entender los MVDs es crucial para mantener la integridad de los datos y optimizar el rendimiento de la base de datos.

3. Claves Candidatas

Las claves candidatas identifican de manera única cada fila en una relación. Representan un conjunto mínimo de atributos que garantizan colectivamente que no existan duplicados en la tabla. En otras palabras, si conoces los valores de la clave candidata, puedes señalar una fila específica y solo esa fila. Las claves son únicas dentro de esa tabla.

Imagina volver a la tabla básica con solo el ID del empleado, el nombre y el apellido.

- Un ID de empleado es una clave candidata fuerte por sí sola. ¿Por qué? Porque cada empleado tiene un ID único, conocer el ID te dice instantáneamente qué registro pertenece a ese empleado en particular.
- Aquí está el giro: ¡Esta tabla podría tener otra clave candidata! Piensa en esto: ¿qué pasaría si dos empleados tienen el mismo nombre y apellido (menos común, pero posible)? En ese caso, la combinación de nombre y apellido no sería suficiente para distinguir entre ellos.
- Pero el ID del empleado, por sí solo, siempre será único y señalará al empleado exacto. Por eso, un solo atributo (ID de empleado) puede ser una clave candidata en este escenario.

Puntos clave:

- **Unicidad:** Cada combinación de valores en la clave candidata debe identificar de manera única una fila distinta.
- **Minimalidad:** Ningún subconjunto propio de la clave candidata debería poder identificar de manera única una fila. Esto asegura que la clave candidata sea el conjunto más pequeño posible de atributos necesarios para la identificación única.
- **Rendimiento:** Tener claves candidatas bien definidas mejora significativamente el rendimiento de las consultas. Las consultas que buscan filas específicas pueden utilizar la indexación en la clave candidata para una recuperación de datos más rápida. También ayuda a mantener la integridad de los datos al prevenir entradas duplicadas.
- **Múltiples claves:** Una relación puede tener múltiples claves candidatas, lo que significa que diferentes conjuntos mínimos de atributos pueden identificar de manera única cada fila.

Aquí está lo importante: Una tabla puede tener múltiples claves candidatas. Siempre que un conjunto de atributos garantice encontrar un registro específico de empleado y ningún otro, califica como una clave candidata. La clave es que identifica de manera única una fila en la tabla.

Escenario práctico

Consideremos un escenario hipotético donde tenemos una base de datos para rastrear los proyectos de los empleados. Cada empleado puede trabajar en múltiples proyectos, y cada proyecto puede involucrar a múltiples empleados. Crearemos una tabla para representar este escenario:

EmployeeID	ProjectID	EmployeeName	ProjectName	Department
1	101	Alice	Proyecto X	RRHH
1	102	Alice	Proyecto Y	Finanzas
2	101	Bob	Proyecto X	RRHH
3	101	Charlie	Proyecto X	TI
3	102	Charlie	Proyecto Y	Finanzas

En esta tabla:

- **EmployeeID**: Identificador único para cada empleado.
- **ProjectID**: Identificador único para cada proyecto.
- **EmployeeName**: Nombre del empleado.
- **ProjectName**: Nombre del proyecto.
- **Department**: Departamento al que pertenece el proyecto.

Dependencias Funcionales (DFs)

EmployeeID -> EmployeeName

- Conocer el EmployeeID determina de manera única el EmployeeName.
- Por ejemplo, si buscas el EmployeeID 1 en la tabla, identifica de manera única el EmployeeName como “Alice”.

EmployeeID -> EmployeeName
1 -> Alice

ProjectID -> ProjectName

- Conocer el ProjectID determina de manera única el ProjectName.
- En la tabla, el ProjectID 101 corresponde a “Project X” y el ProjectID 102 corresponde a “Project Y”, demostrando una relación uno a uno.

ProjectID ↔ ProjectName
101 ↔ Project X
102 ↔ Project Y

Dependencias de Múltiples Valores (MVDs)

{EmployeeID} ->> {ProjectID}

- Conocer el EmployeeID nos dice todos los proyectos en los que están trabajando los empleados, de manera independiente entre sí.

Revisemos el ejemplo del EmployeeID 1 (Alice) de la tabla:

- El EmployeeID 1 está asociado con el ProjectID 101 (Proyecto X) en el departamento de Recursos Humanos.
- El EmployeeID 1 también está asociado con el ProjectID 102 (Proyecto Y) en el departamento de Finanzas.

EmployeeID – {asociado con} -> ProjectID (departamento)
1————> 101 (Recursos Humanos)
1————> 102 (Finanzas)

Conocer solo el EmployeeID 1 no te dice en qué proyecto específico trabaja Alice. Indica que potencialmente hay varias asignaciones de proyectos para ella, reflejadas en filas separadas en la tabla. Esto destaca la relación de muchos a muchos entre EmployeeID y ProjectID.

Claves Candidatas

Hay dos claves candidatas (CKs):

- **EmployeeID**: Como se discutió anteriormente, cada empleado tiene un identificador único (EmployeeID) que señala su registro específico. Puedes identificar a cualquier empleado únicamente basado en su EmployeeID, lo que lo convierte en una clave candidata.

EmployeeID -> EmployeeName
1 -> Alice

- **Combinación de EmployeeID y ProjectID:** Esto puede parecer sorprendente, pero considera un escenario donde cada empleado tiene un nombre distinto dentro de un proyecto (sin duplicados dentro de los proyectos). En ese caso, la combinación de EmployeeName y ProjectID identificaría de manera única cada registro de empleado.

EmployeeID y ProjectID determinan el Departamento
EmployeeID -> ProjectID ==> Departamento
1 -> 101 => RRHH

Por ejemplo, si Alice fue asignada al Proyecto X y Charlie al Proyecto Y (suponiendo nombres únicos dentro de los proyectos), entonces la combinación de “John” y “Proyecto X” o “Mary” y “Proyecto Y” identificaría de manera única sus respectivas filas.

Resumen

Aspecto	Dependencias Funcionales	Dependencias de Múltiples Valores (MVDs)	Claves Candidatas
Definición	Describen relaciones entre atributos	Amplían el concepto a grupos de atributos	Conjuntos de atributos que identifican de manera única las filas
Esencia	Valores de ciertos atributos determinados por otros	Dependencias entre conjuntos de atributos	Identifican de manera única cada fila en una tabla
Ejemplo	Conocer un atributo permite encontrar otro	Describen cómo los conjuntos de atributos se determinan entre sí	Combinación de atributos que identifican de manera única cada registro
Propósito	Asegurar la precisión de los datos y minimizar la redundancia	Organizar los datos de manera efectiva, evitando confusiones	Hacer cumplir las restricciones de integridad de entidad
Uso	Esencial para la normalización de bases de datos	Crucial para mantener la integridad de los datos	Establecer relaciones entre tablas

Autor(es)

[Akansha Yadav](#)

