

MP0484.

Bases de datos

UF1. Fundamentos conceptuales

1.4. Restricciones inherentes del modelo relacional

Índice

≡ Introducción y Objetivos	3
≡ Conceptos	4
≡ Restricciones Inherentes del Modelo Relacional	5
≡ Las 12 reglas de Codd	6
≡ Restricciones Semánticas o de Usuario	9
≡ Más restricciones semánticas	11
≡ Concepto del valor nulo	12

Introducción y Objetivos

Introducción

La implementación de bases de datos relacionales tiene una **serie de restricciones de construcción** que están relacionadas con el **modelo entidad relación ER** y son **inherentes** a la construcción de dicho modelo. En este capítulo no solo veremos dichas restricciones sino las reglas que propone Codd, éstas son un sistema de reglas (numeradas del 0 al 12) propuestas por Edgar F. Codd, del modelo relacional para las bases de datos, diseñado para definir qué requiere un sistema de administración de base de datos, y las **restricciones semánticas o de usuario**.

Objetivos

- Conocer cuáles son las **restricciones** relacionadas con la construcción de bases de datos relacionales.
- Conocer cuáles son las restricciones **inherentes** del modelo relacional.
- Conocer las **12 reglas de Codd**.
- Trabajar con las restricciones **semánticas o de usuario**.
- Concepto del **valor nulo**.

Conceptos

Las restricciones nos permiten **trabajar** en el **mundo real** en el caso de los datos y en el mundo **relacional** en caso de la representación de los datos de una forma específica.

Se pueden establecer **restricciones de integridad** que son unas condiciones específicas para una base de datos en particular. Existen las restricciones sobre los **valores** que puedan tomar los **atributos** y restricciones **estructurales** que se referirán a los atributos y a las interrelaciones.

Otras reglas que tienen que ver con cómo se construye el modelo relacional son las reglas **inherentes al modelo**, que son condiciones más generales, y propias de un modelo de datos, que se deben cumplir en toda base de datos que siga dicho modelo.

Restricciones Inherentes del Modelo Relacional

Estas restricciones marcan la **diferencia entre una tabla y una relación**, las relaciones se fundamentan en axiomas matemáticos y no se acercan a las estructuras de datos que suelen manejarse en los Sistemas de Gestión de Bases de Datos, una tabla presenta las **filas y las columnas** en un orden en el que se construyó, eso es un concepto que **no existe en el modelo relacional**. Por otro lado, una tabla podría contener filas repetidas. De todos modos, es muy común utilizar el término tabla para referirse a una relación.

Obligatoriedad de clave primaria

No existen tuplas repetidas. La relación se ha definido como un conjunto de tuplas, y en matemáticas los conjuntos por definición no incluyen elementos repetidos. Hay que decir, sin embargo, que muchos de los SGBD relacionales sí admiten duplicidad de tuplas.

Orden de Atributos

El **orden** de las tuplas y el de los atributos **no es relevante**.

Valor único atributo

Cada atributo de cada tupla solo puede tomar un único valor sobre el dominio sobre el que está definido.

No valor nulo

Ningún atributo que forme parte de la **clave primaria** de una relación puede tomar un valor nulo (regla de integridad de entidad).

Valores atómicos

Todos los **valores de los atributos son atómicos**. Otra forma de expresar esta propiedad es diciendo que todos los valores de los atributos simples son atómicos, sólo toman un valor en cada caso.

Las 12 reglas de Codd

Preocupado por los productos como los sistemas gestores de bases de datos relacionales (**RDBMS**), **Codd** publica las **12 reglas** que debe cumplir todo **DBMS** para ser considerado relacional.

Estas reglas en la práctica las cumplen pocos sistemas relacionales ya que algunas de estas reglas son difíciles de implementar, así que un sistema podrá considerarse más relacional cuanto más siga estas reglas. Según se indica en Wikipedia, las reglas son:

- 1 El sistema de administración y gestión de la base de datos debe ser relacional. Eso significa que el sistema debe utilizar las reglas relacionales de forma exclusiva para manejar la base de datos.
- 2 La regla de la información, toda la información en la base de datos es representada unidireccionalmente, por valores en posiciones de las **columnas** dentro de **filas de tablas**. Toda la información en una base de datos relacional se representa explícitamente en el **nivel Lógico** exactamente de una manera: con valores en tablas. Los **metadatos** (diccionario, catálogo) se representan y se manipulan **exactamente igual** que los datos de usuario.
- 3 La regla del acceso garantizado, todos los datos deben ser accesibles sin ambigüedad. Esta regla es esencialmente una nueva exposición del requisito fundamental para las llaves primarias. Dice que cada valor escalar individual en la base de datos debe ser lógicamente accesible especificando el nombre de la tabla, la columna que lo contiene y la llave primaria. Para ello hay que indicar en qué tabla está, cuál es la columna y cuál es la fila (mediante la clave primaria).
- 4 Tratamiento sistemático de valores nulos, el sistema de gestión de base de datos debe permitir que haya campos nulos. Debe tener una representación de la "información que falta y de la información inaplicable" que es sistemática, distinto de todos los valores regulares. Hay problemas en el soporte de los valores nulos en operaciones relacionales, especialmente en las operaciones lógicas, para lo cual se usa una lógica de tres valores, con tres (no dos) valores de verdad: **Verdadero, Falso y null**. Se crean tablas de verdad para las operaciones lógicas:
 - null AND null = null
 - Verdadero AND null = null
 - Falso AND null = Falso
 - Verdadero OR null = Verdadero, etc.
- 5 Catálogo dinámico en línea basado en el modelo relacional, el sistema debe soportar un catálogo en línea, el catálogo relacional debe ser accesible a los usuarios autorizados. Es decir, los usuarios autorizados deben poder tener acceso a la estructura de la base de datos (catálogo).
- 6 Regla comprensiva del lenguaje de los datos, el sistema debe soportar por lo menos un lenguaje relacional que:

- Tenga una **sintaxis lineal**, y tendrá que soportar:
 - Definición de datos.
 - Definición de vistas.
 - Manipulación de datos (interactiva y por programa).
 - Restricciones de integridad.
 - Restricciones de transacciones (begin, commit, rollback).
- Puede ser utilizado de **manera interactiva**.
- Soporte **operaciones de definición de datos**, operaciones de **manipulación de datos** (tanto la actualización como la recuperación), **seguridad e integridad** y **operaciones** de administración de **transacciones**.

7

Actualización de vistas, todas las vistas que son teóricamente actualizables deberán ser **actualizables por el sistema**. El problema es determinar cuáles son las vistas teóricamente actualizables, ya que no está muy claro. Cada sistema puede hacer unas suposiciones particulares sobre las vistas que son actualizables. La **capacidad de manejar una relación base o derivada** como un solo operando se aplica no sólo a la recuperación de los datos (consultas), sino también a la **inserción, actualización y borrado de datos**.

8

Inserción, actualización y borrado de alto nivel, permitiendo el sistema realizar manipulación de datos de alto nivel, es decir, sobre conjuntos de tuplas. Esto significa que los datos no solo se pueden recuperar de una base de datos relacional de filas múltiples y/o de tablas múltiples, sino también pueden realizarse inserciones, actualización y borrados sobre varias tuplas y/o tablas al mismo tiempo (no sólo sobre registros individuales).

9

Independencia física de los datos, los programas de aplicación y actividades del terminal permanecen **inalterados a nivel lógico** cuando quiera que se realicen cambios en las representaciones de almacenamiento o métodos de acceso.

10

Independencia lógica de los datos, los cambios al nivel lógico (tablas, columnas, filas, etc.) no deben requerir un cambio a una solicitud basada en la estructura. La independencia de datos lógica es más difícil de lograr que la independencia física de datos. Ejemplos de cambios que preservan la información:

- Añadir un atributo a una tabla base.
- Sustituir dos tablas base por la unión de las mismas. Usando vistas de la unión se pueden recrear las tablas anteriores.

11

Independencia de la integridad, las limitaciones de la integridad se deben especificar por **separado de los programas** de la aplicación y se **almacenan en la base de datos**. Debe ser posible cambiar esas limitaciones sin afectar innecesariamente las aplicaciones existentes.

- El objetivo de las bases de datos no es sólo almacenar los datos, sino también sus relaciones y evitar que estas restricciones se codifiquen en los programas. Por tanto en una base de datos relacional se deben poder definir restricciones de integridad.
- Cada vez se van ampliando más los tipos de restricciones de integridad que se pueden utilizar en los Sistemas de Gestión de Bases de Datos Relacionales, aunque hasta hace poco eran muy escasos.
- Como parte de las restricciones inherentes al modelo relacional (forman parte de su definición) están:
 - **Integridad de Entidad**: toda tabla debe tener una clave primaria.
 - **Integridad de Dominio**: toda columna de una tabla contendrá valores exclusivamente de un determinado dominio (conjunto de valores válidos)

- **Integridad Referencial:** toda clave foránea no nula debe existir en la relación donde es clave primaria.

12

Independencia de la distribución, la distribución de las porciones de la base de datos a las varias localizaciones debe ser invisible a los usuarios de la base de datos. Los usos existentes deben continuar funcionando con éxito:

- Cuando una versión distribuida del SGBD se introdujo por primera vez.
- Cuando se distribuyen los datos existentes se redistribuyen en todo el sistema.
- Las mismas órdenes y programas se ejecutan igual en una base de datos centralizada que en una distribuida.
- Las bases de datos son fácilmente distribuibles.
- Esta regla es responsable de tres tipos de transparencia de distribución:
 - **Transparencia de Localización:** el usuario tiene la impresión de que trabaja con una base de datos local. (Regla de Independencia Física).
 - **Transparencia de Fragmentación:** el usuario no se da cuenta de que la relación con que trabaja está fragmentada. (Regla de Independencia Lógica).
 - **Transparencia de Replicación:** el usuario no se da cuenta de que pueden existir copias (réplicas) de una misma relación en diferentes lugares.

13

No subversión, si el sistema proporciona una interfaz de bajo nivel de registro, aparte de una interfaz relacional, aquella interfaz de bajo nivel **no debe poder usarse para saltarse las reglas del sistema**, por ejemplo: sin pasar por seguridad relacional o limitación de integridad. Esto es debido a que existen sistemas anteriormente no relacionales que añadieron una interfaz relacional, pero con la interfaz nativa existe la posibilidad de trabajar no relationalmente.

Restricciones Semánticas o de Usuario

Restricción de Clave Primaria (PRIMARY KEY)

Permite declarar un **atributo o conjunto de atributos** como la **clave primaria de una relación**.

La regla de integridad de unicidad de la clave primaria establece que si el conjunto de atributos CP es la clave primaria de una relación R, entonces la extensión de R no puede tener en ningún momento dos tuplas con la misma combinación de valores para los atributos de CP.

Restricción de Unicidad (UNIQUE)

Permite que una **clave alternativa o secundaria** pueda tomar **valores únicos** para las tuplas de una relación (como si de una clave primaria se tratara). Se entiende que la clave primaria siempre tiene esta restricción. La regla de integridad de entidad de la clave primaria establece que, si el conjunto de atributos CP es la clave primaria de una relación R, la extensión de R no puede tener ninguna tupla con algún valor nulo para alguno de los atributos de CP.

Restricción De Obligatoriedad (NOT NULL)

Permite declarar si **uno o varios atributos** de una relación debe tomar siempre un valor.

Restricción de Integridad Referencial o de Clave Foránea (FOREING KEY)

Se utiliza para que mediante **claves foráneas podamos enlazar relaciones** de una base de datos. La integridad referencial nos indica que si una relación tiene una clave foránea que referencia a otra relación, cada valor de la clave foránea o ajena tiene que ser igual a un valor de la clave principal de la relación a la que referencia, o bien, ser completamente nulo. Los atributos que son clave foránea en una relación no necesitan tener los mismos nombres que los atributos de la clave primaria con la cual ellos se corresponden.

En relación a la **clave foránea**, el diseñador de la base de datos deberá poder especificar qué **operaciones han de rechazarse y cuáles han de aceptarse**, y en este caso, qué operaciones de compensación hay que realizar para **mantener la integridad de la base de datos**. Para ello el diseñador debe plantearse tres preguntas por cada clave foránea:

1

¿Puede aceptar nulos esa clave foránea?

Por ejemplo, (tomando como referencia las relaciones COLEGIO, ALUMNO) ¿tiene sentido la existencia de un ALUMNO cuyo COLEGIO se desconoce?

Evidentemente, **no**. En algunos casos esta respuesta podría ser distinta, por ejemplo, en una base de datos con las relaciones EMPLEADOS y DEPARTAMENTOS, podría existir un empleado no asignado de momento a un departamento.

2

¿Qué deberá suceder si se intenta eliminar una tupla referenciada por una clave foránea?

Por ejemplo, si se intenta eliminar un COLEGIO del cual existe algún ALUMNO. En general, para estos casos existen por lo menos tres posibilidades:

Restringir

La operación de eliminación **se restringe sólo al caso en el que no existe** alguna tupla con clave foránea que la referencia, rechazándose en caso contrario. En nuestro ejemplo, un proveedor podrá ser borrado, si y sólo si, por ahora, no suministra artículos.

Propagar en cascada

La operación de borrado se **propaga en cascada** eliminando también todas las tuplas cuya clave foránea la **referencien**. En nuestro ejemplo, se eliminaría el proveedor y todas las tuplas de artículos suministrados por él.

Anular

Se **asignan nulos** en la clave foránea de todas las **tuplas** que la **referencien** y se elimina la tupla referenciada. En nuestro ejemplo, no tiene mucho sentido, pero consistiría en asignar nulos al NIF-PROV de todas las tuplas de artículos pertenecientes al proveedor que queremos borrar, y posteriormente borrar al proveedor.

3

¿Qué deberá suceder si hay un intento de **modificar la clave primaria** de una tupla referenciada por una clave foránea?

Por ejemplo, si se intenta modificar la clave de un COLEGIO del cual existe algún ALUMNO.

Se actúa con las mismas tres posibilidades que en el caso anterior:

- Restringir.
- Propagar en cascada.
- Anular.

Más restricciones semánticas

Más restricciones semánticas

Restricción de Valor por Defecto (DEFAULT)

Permite que cuando se **inserte** una **tupla** o registro en una tabla, para aquellos atributos para los cuales no se indique un valor exacto se les **asigne un valor por defecto**.

Aserciones (ASSERTION)

Esta restricción generaliza a la anterior, lo forman las **aserciones** en las que la **condición** se establece **sobre elementos de distintas relaciones** (por ello debe tener un nombre que la identifique).

Restricción de Verificación o Chequeo (CHECK)

En algunos casos puede ocurrir que sea necesario **especificar una condición** que deben cumplir los **valores** de determinados atributos de una relación de la BD, aparte de las restricciones vistas anteriormente.

Disparadores (TRIGGERS)

A veces puede interesar especificar una **acción distinta del rechazo** cuando no se cumple una determinada restricción semántica. Se recurre al uso de **disparadores** o **triggers** que nos permiten además de indicar una condición, especificar la acción que queremos que se lleve a cabo si la condición se hace verdadera. Los disparadores pueden interpretarse como reglas del tipo evento-condición-acción (ECA) que pueden interpretarse como reglas que especifican que cuando se produce un evento, si se cumple una condición, entonces se realiza una determinada acción.

Concepto del valor nulo

No es exclusivo del modelo relacional, pero en este contexto es dónde se ha abordado su estudio de manera más sistemática y donde se están realizando más investigaciones a fin de formalizar su tratamiento.

Es como una señal utilizada para representar información desconocida, inaplicable, inexistente, no válida, no proporcionada, indefinida, etc.

Motivos de necesidad en las B.D.

- Crear tuplas con ciertos atributos desconocidos en ese momento. P.ej. Nueva Alumno sin teléfono.
- Añadir un nuevo atributo a una relación existente (al añadirse, no tiene ningún valor para las tuplas de la relación). P.ej. Añadir Estado Civil.
- Atributos inaplicables a ciertas tuplas. P.ej. la profesión para un alumno de primaria.

El tratamiento de valores nulos exige definir unas operaciones específicas para el caso de que alguno de los operandos tome valores nulos, y a introducir operadores especiales.



PROEDUCA