

LAS 12 REGLAS DE CODD QUE DETERMINAN LA FIDELIDAD DE UN SISTEMA RELACIONAL AL MODELO RELACIONAL

En la década de los 80 comenzaron a aparecer numerosos SGBD que se anunciaban como "relacionales". Sin embargo estos sistemas carecían de muchas características que se consideran importantes en un sistema relacional, perdiendo muchas ventajas del modelo relacional. Como ejemplo extremo de esto "sistemas relacionales" eran simplemente sistemas que utilizaban tablas para almacenar la información, no disponiendo de elementos como claves primarias, etc.

En 1984 Codd publicó 12 reglas que un verdadero sistema relacional debería de cumplir. En la práctica algunas de ellas son difíciles de realizar.

Un sistema podrá considerarse "más relacional" cuanto más siga estas reglas.

REGLA 0

Para que un sistema se denomine sistema de gestión de bases de datos relacionales, este sistema debe usar (exclusivamente) sus capacidades relacionales para gestionar la base de datos.

REGLA 1: REGLA DE LA INFORMACIÓN

Toda la información en una base de datos relacional se representa explícitamente en el nivel lógico exactamente de una manera: con valores en tablas.

- Por tanto los metadatos (diccionario, catálogo) se representan exactamente igual que los datos de usuario.
- Y puede usarse el mismo lenguaje (ej. SQL) para acceder a los datos y a los metadatos (regla 4)
- Un valor posible es el valor nulo, con sus dos interpretaciones:
 - Valor desconocido (ej. dirección desconocida)
 - Valor no aplicable (ej. empleado soltero no tiene esposa).

REGLA 2: REGLA DEL ACCESO GARANTIZADO

Para todos y cada uno de los datos (valores atómicos) de una BDR se garantiza que son accesibles a nivel lógico utilizando una combinación de nombre de tabla, valor de clave primaria y nombre de columna.

- Cualquier dato almacenado en una BDR tiene que poder ser direccionado unívocamente. Para ello hay que indicar en qué tabla está, cuál es la columna y cuál es la fila (mediante la clave primaria).
- Por tanto se necesita el concepto de clave primaria, que no es soportado en muchas implementaciones. En estos casos, para lograr un efecto similar se puede hacer lo siguiente:

- Hacer que los atributos clave primaria no puedan ser nulos (NOT NULL).
- Crear un índice único sobre la clave primaria.
- No eliminar nunca el índice.

REGLA 3: TRATAMIENTO SISTEMÁTICO DE VALORES NULOS

Los valores nulos (que son distintos de la cadena vacía, blancos, 0, ...) se soportan en los SGBD totalmente relacionales para representar información desconocida o no aplicable de manera sistemática, independientemente del tipo de datos.

- Se reconoce la necesidad de la existencia de valores nulos, para un tratamiento sistemático de los mismos.
- Hay problemas para soportar los valores nulos en las operaciones relacionales, especialmente en las operaciones lógicas.

Lógica trivaluada. Es una posible solución. Existen tres (no dos) valores de verdad: Verdadero, Falso y Desconocido (null). Se crean tablas de verdad para las operaciones lógicas:

- null Y null = null
- Verdadero Y null = null
- Falso Y null = Falso
- Verdadero O null = Verdadero
- etc.

Un inconveniente es que de cara al usuario el manejo de los lenguajes relacionales se complica pues es más difícil de entender.

REGLA 4: CATÁLOGO DINÁMICO EN LÍNEA BASADO EN EL MODELO RELACIONAL

La descripción de la base de datos se representa a nivel lógico de la misma manera que los datos normales, de modo que los usuarios autorizados pueden aplicar el mismo lenguaje relacional a su consulta, igual que lo aplican a los datos normales.

- Es una consecuencia de la regla 1 que se destaca por su importancia. Los metadatos se almacenan usando el modelo relacional, con todas las consecuencias.

REGLA 5: REGLA DEL SUBLENGUAJE DE DATOS COMPLETO

Un sistema relacional debe soportar varios lenguajes y varios modos de uso de terminal (ej: rellenar formularios, etc.). Sin embargo, debe existir al menos un lenguaje cuyas sentencias sean expresables, mediante una sintaxis bien definida, como cadenas de caracteres y que sea completo, soportando:

- Definición de datos
- Definición de vistas

- Manipulación de datos (interactiva y por programa)
- Limitantes de integridad
- Limitantes de transacción (iniciar, realizar, deshacer) (Begin, commit, rollback).
- Además de poder tener interfaces más amigables para hacer consultas, etc. siempre debe de haber una manera de hacerlo todo de manera textual, que es tanto como decir que pueda ser incorporada en un programa tradicional.
- Un lenguaje que cumple esto en gran medida es SQL.

REGLA 6: REGLA DE ACTUALIZACIÓN DE VISTAS

Todas las vistas que son teóricamente actualizables se pueden actualizar 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.

REGLA 7: INSERCIÓN, ACTUALIZACIÓN Y BORRADO DE ALTO NIVEL

La capacidad de manejar una relación base o derivada como un solo operando se aplica no sólo a la recuperación de los datos (consultas), si no también a la inserción, actualización y borrado de datos.

- Esto es, el lenguaje de manejo de datos también debe ser de alto nivel (de conjuntos). Algunas bases de datos inicialmente sólo podían modificar las tuplas de la base de datos de una en una (un registro de cada vez).

REGLA 8: INDEPENDENCIA FÍSICA DE DATOS

Los programas de aplicación y actividades del terminal permanecen inalterados a nivel lógico cuandoquiera que se realicen cambios en las representaciones de almacenamiento o métodos de acceso.

- El modelo relacional es un modelo lógico de datos, y oculta las características de su representación física.

REGLA 9: INDEPENDENCIA LÓGICA DE DATOS

Los programas de aplicación y actividades del terminal permanecen inalterados a nivel lógico cuandoquiera que se realicen cambios a las tablas base que preserven la información.

- Cuando se modifica el esquema lógico preservando información (no valdría p.ej. eliminar un atributo) no es necesario modificar nada en niveles superiores.
- 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 puedo recrear las tablas anteriores...

REGLA 10: INDEPENDENCIA DE INTEGRIDAD

Los limitantes de integridad específicos para una determinada base de datos relacional deben poder ser definidos en el sublenguaje de datos relacional, y almacenables en el catálogo, no en los programas de aplicación.

- El objetivo de las bases de datos no es sólo almacenar los datos, si no también sus relaciones y evitar que estas (limitantes) se codifiquen en los programas. Por tanto en una BDR se deben poder definir limitantes de integridad.
- Cada vez se van ampliando más los tipos de limitantes de integridad que se pueden utilizar en los SGBDR, aunque hasta hace poco eran muy escasos.
- Como parte de los limitantes inherentes al modelo relacional (forman parte de su definición) están:
 - Una BDR tiene **integridad de entidad**. Es decir, toda tabla debe tener una clave primaria.
 - Una BDR tiene **integridad referencial**. Es decir, toda clave externa no nula debe existir en la relación donde es primaria.

REGLA 11: INDEPENDENCIA DE DISTRIBUCIÓN

Una BDR tiene independencia de distribución.

- Las mismas órdenes y programas se ejecutan igual en una BD centralizada que en una distribuida.
- Las BDR son fácilmente distribuibles:
 - Se parten las tablas en fragmentos que se distribuyen.
 - Cuando se necesitan las tablas completas se recombinan usando operaciones relacionales con los fragmentos.
 - Sin embargo se complica más la gestión interna de la integridad, etc.
- Esta regla es responsable de tres tipos de **transparencia de distribución**:
 - Transparencia de localización. El usuario tiene la impresión de que trabaja con una BD local. (Aspecto de la regla de independencia física)
 - Transparencia de fragmentación. El usuario no se da cuenta de que la relación con que trabaja está fragmentada. (Aspecto de la regla de independencia lógica de datos).

- Transparencia de replicación. El usuario no se da cuenta de que pueden existir copias (réplicas) de una misma relación en diferentes lugares.

REGLA 12: REGLA DE LA NO SUBVERSIÓN

Si un sistema relacional tiene un lenguaje de bajo nivel (un registro de cada vez), ese bajo nivel no puede ser usado para saltarse (subvertir) las reglas de integridad y los limitantes expresados en los lenguajes relacionales de más alto nivel (una relación (conjunto de registros) de cada vez).

- Algunos problemas no se pueden solucionar directamente con el lenguaje de alto nivel.
- Normalmente se usa SQL inmerso en un lenguaje anfitrión para solucionar estos problemas. Se utiliza el concepto de cursor para tratar individualmente las tuplas de una relación. En cualquier caso no debe ser posible saltarse los limitantes de integridad impuestos al tratar las tuplas a ese nivel.