

Las doce reglas de Codd

Introducción

En este trabajo damos un breve resumen de la biográfica de Edgar Frank Codd, decimos quien es y nos centramos en las 12 reglas que propuso para Diseño de Bases de Datos Relacionales y como están implementadas en las diferentes gestores de Bases de datos.

Edgar Frank Codd

Nacido en Inglaterra el 23 de agosto de 192, Edgar es considera el padre de las bases de datos relacional.

En 1969 Edgar Codd inventó el modelo relacional, el modelo de bases de datos más usado hoy en día y para muchas personas, el único que conocen. Desde el sistema R de IBM a Oracle han pasado 30 años y aún es el modelo dominante. Inicialmente el apoyo de IBM a los sistemas de bases de datos tradicionales (de redes) era mayoritario, poderoso y agresivo. Sólo años más tarde, en 1978, durante una reunión técnica de alto nivel el modelo relacional llamó la atención del presidente de IBM, Frank Cary. Más tarde IBM anunció SQL/DS, su primer producto relacional comercial en 1981, seguido de DB2 en 1983. Sin embargo esta tardanza en adoptar el modelo relacional significó perder un mercado que tomaron otros. El trabajo inicial de Codd fue publicado en Communications of the ACM en 1970. Su trabajo sobre normalización de bases de datos fue publicado como un informe técnico de IBM en 1971. Ocho años más tarde, en ACM Transactions of Database Systems, publicó varias extensiones al modelo relacional. En 1985 postuló una lista de 13 reglas que debía cumplir un producto de bases de datos para ser llamado relacional.

12 reglas de Codd

Codd se percató de que existían bases de datos en el mercado las cuales decían ser relacionales, pero lo único que hacían era guardar la información en las tablas, sin estar estas tablas literalmente normalizadas; entonces éste publicó 12 reglas que un verdadero sistema relacional debería tener aunque en la práctica algunas de ellas son difíciles de realizar. Un sistema podrá considerarse "más relacional" cuanto más siga estas reglas.

– **Regla 0:** el sistema debe ser relacional, base de datos y administrador de sistema. Ese sistema debe utilizar sus facilidades relacionales (exclusivamente) para manejar la base de datos.

– **Regla 1:** 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.

– **Regla 2:** 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 direccionable especificando el nombre de la tabla, la columna que lo contiene y la llave primaria.

– **Regla 3:** 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.

- **Regla 4:** 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 deben poder tener acceso a la estructura de la base de datos (catálogo).
- **Regla 5:** la regla comprensiva del sublenguaje de los datos, el sistema debe soportar por lo menos un lenguaje relacional que:
 - Tenga una sintaxis lineal.
 - Puede ser utilizado de manera interactiva.
 - Soporte operaciones de definición de datos, operaciones de manipulación de datos (actualización así como la recuperación), seguridad e integridad y operaciones de administración de transacciones.
- **Regla 6:** regla de actualización, todas las vistas que son teóricamente actualizables deben ser actualizables por el sistema.
- **Regla 7:** alto nivel de inserción, actualización, y cancelación, el sistema debe soportar suministrar datos en el mismo tiempo que se inserte, actualiza o esté borrando. Esto significa que los datos se pueden recuperar de una base de datos relacional en los sistemas construidos de datos de filas múltiples y/o de tablas múltiples.
- **Regla 8:** independencia física de los 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.
- **Regla 9:** 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.
- **Regla 10:** 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.
- **Regla 11:** 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.
- **Regla 12:** la regla de la no subversión, si el sistema proporciona una interfaz de bajo nivel de registro, a parte de una interfaz relacional, que esa interfaz de bajo nivel no se pueda utilizar para subvertir el 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 relacionamente.

Implementación de las Reglas de Codd en los Gestores de Bases de Datos

ORACLE

ORACLE dice cumplir las 12 reglas de Codd y se considera la base de datos relacional por excelencia, aquí muestro un extracto de cómo maneja ORACLE las 12 reglas.

Regla 1 – La	Toda la información en una base de datos relacional se representa explícitamente
--------------	--

Diseño de Base de Datos I

Prof. Lic. Gabriel F. Arenas

Información	en el nivel lógico y exactamente de una manera – por los valores de las tablas.
Regla 2 – Acceso Garantizado	Todos y cada dato (valor atómico) en una base de datos relacional se garantiza que sea lógicamente accesibles recurriendo a una combinación de nombre de tabla, valor de clave principal, y el nombre de la columna.
Regla 3 – Tratamiento sistemático de valores nulos	Los valores nulos (distintos de la cadena de caracteres vacía de caracteres en blanco y distintos de los números cero u otro) son compatibles con completamente relacional DBMS para representar la información que falta y la información aplicable de manera sistemática.
Regla 4 – Catálogo dinámico en línea basado en el Modelo Relacional	La descripción de la base de datos está representado en el nivel lógico de la misma manera que los datos ordinarios, por lo que los usuarios autorizados pueden aplicar el mismo lenguaje relacional para su interrogatorio que se aplican a los datos normales.
Regla 5 – Sublenguaje de datos completa	<p>Un sistema relacional puede soportar varios lenguajes y varios modos de uso de terminal (por ejemplo, el modo de relleno en los espacios en blanco). Sin embargo, debe haber al menos una lengua cuyas declaraciones son expresables, por una sintaxis bien definida, como cadenas de caracteres, que es integral en el apoyo de todos los siguientes elementos:</p> <ul style="list-style-type: none"> - Definición de datos - Ver definición - Manipulación de datos (interactivo y por el programa) - Restricciones de integridad - Autorización - Límites de transacción (begin, commit y rollback)
Regla 6 – Actualización de Vistas	Todas las vistas que son teóricamente actualizables también son actualizables por el sistema.
Regla 7 – Insert de Alto Nivel, Update y Delete	La capacidad de manejar una relación base o una relación derivada como un solo operando se aplica no sólo a la recuperación de los datos, sino también para la inserción, actualización y supresión de los datos.
Regla 8 – Independencia de Datos Físico	Los programas de aplicación y actividades del terminal permanecen lógicamente irreprochable cuando se realiza algún cambio en cualquiera de las representaciones de almacenamiento o métodos de acceso.
Regla 9 – Independencia lógica de datos	Los programas de aplicación y actividades del terminal permanecen lógicamente irreprochable cuando la información de preservación de cambios de toda naturaleza que discapacidad de permiso teóricamente se hacen a las tablas base.
Regla 10 – Independencia de Integridad	Las restricciones de integridad específicos para una base de datos relacional en particular deben ser definibles en los datos relacionales sub-lenguaje y almacenables en el catálogo, no en los programas de aplicación.
Regla 11 – Independencia Distribución	Un DBMS relacional tiene dependencia de distribución.
Regla 12 – No	Si un sistema relacional tiene un (solo registro a la vez) de bajo nivel de lenguaje,

Subversión	ese bajo nivel no puede ser utilizado para subvertir o pasar por alto las reglas de integridad y las limitaciones expresadas en el lenguaje relacional de alto nivel (varios registros a la vez).
------------	---

SQL SERVER

SQL SERVER al igual que ORACLE tiene una forma de manejar las reglas que mostramos a continuación.

Regla 1 – La Información	La información es guardada en tablas, hay variables también, pero que son específicos sesión.
Regla 2 – Acceso Garantizado	Se busca Llave primaria (que es único para una tabla), y una vez que se identifica la fila, correspondiente valor de columna se puede derivar.
Regla 3 – Tratamiento sistemático de valores nulos	<p>Idealmente, cualquier operación de trabajo con NULL debe conducir a NULL. Aunque hubo algunos problemas en el pasado debido a que algunas opciones SQLServer están en su lugar, que decide cómo se maneja NULL (en lugar de manejar de la manera "definitiva").- ANSI_NULLS: Si OFF, entonces NULL = NULL es True para la comparación. Si ON (por defecto), NULL = NULL devuelve DESCONOCIDO (situación ideal).</p> <p>- CONCAT_NULL_YIELDS_NULL: Si ON, NULL se tratan de manera ideal. por ejemplo, NULL + <numValue> = NULL. Si está desactivado, los valores NULL son tratados de una forma no estándar tal que NULL + <Valor> = <Valor>. (Hecho para la compatibilidad con versiones anteriores de acuerdo BOL, que lo dice todo)</p>
Regla 4 – Catálogo dinámico en línea basado en el Modelo Relacional	INFORMATION_SCHEMA en SQL Server es un tanto la aplicación de la misma regla. Hay otros puntos de vista en SYS que pueden proporcionar información más específica DB.
Regla 5 – Sublenguaje de datos completa	TSQL cumple los mismos requisitos. Es importante aquí que TSQL no es de procedimiento como en el caso de Oracle. Simplemente proporciona un interfaz para extraer algo de DBEngine SQLServer.
Regla 6 – Actualización de Vistas	SQL Server no es muy bueno en el manejo de vistas actualizables, que han comenzado a llegar un poco tarde recientemente. Puede usar desencadenadores INSTEAD OF, pero se ven patch'y. El manejo concreto vistas cuando son agregados es un poco complicado.
Regla 7 – Insert de Alto Nivel, Update y Delete	INSERT, DELETE y UPDATE cubre esto. Más reciente MERGE también trabaja en el mismo principio de manejo de fila set. Aunque BCP no moverse y trabaja en un tanto en el nivel inferior, pero eso es un propósito definido y no por defecto.
Regla 8 – Independencia de Datos Físico	Interfaz TSQL es transparente para el usuario final, aunque el sistema podría estar haciendo uso de índices, particiones, grupos de archivos, etc Además, entrada OLEDB, base de datos (relacional) y motor de almacenamiento se maneja bien esta regla.
Regla 9 – Independencia lógica de datos	Esto no es un objetivo muy sencillo y requiere algo de las mejores prácticas que se siguió en la parte de desarrollo. Si llama por nombres de columna en concreto en vez de SELECT *, no hay mucho en ella. Además, las opiniones se pueden crear en la parte superior de las tablas divididas y que pueden ser completamente ocultos a usuario final.
Regla 10 – Independencia de Integridad	NULL, PK y FK limitaciones se gestionan bien dentro de la DBEngine. Los disparadores pueden estar también se añaden en la parte superior de estas limitaciones para proporcionar los siguientes pasos.



Diseño de Base de Datos I

Prof. Lic. Gabriel F. Arenas

Regla 11 – Independencia Distribución	Las transacciones distribuidas son sólo para eso. Federated servidores es otro ejemplo de aplicación de esta norma.
Regla 12 – No Subversión	Aunque si estamos dentro del contexto TSQL, esta regla está bastante conservada. Pero hay formas de evitar con ello el uso de BCP y desactivando las limitaciones / disparadores, su importancia para dev / DBA ser consciente de este déficit al utilizar estas aplicaciones. Cumple, pero puede romper también.

Conclusiones

En este trabajo incluimos a los 2 más grandes gestores de bases de datos SQL SERVER y ORACLE los cuales utilizan las 12 reglas de Codd pero hace cambios en ellas según lo que ellos creen conveniente y lo que consideran óptimo.