



BASES DE DATOS.

GRUPO: 01.

TAREA 7: "REGLAS DE CODD".

ALUMNO: AGUILAR BUSTILLOS ALAN RODRIGO.

PROFESOR: ING. FERNANDO ARREOLA FRANCO.

SEMESTRE 2024-1

El diseño de las bases de datos tradicionalmente se sujeta a una serie de reglas que permiten su diseño y construcción de forma eficiente. Estas reglas son conocidas como “*Las 12 Reglas de las Bases de Datos*”. Estas reglas fueron propuestas por el mismo creador del lenguaje SQL (*Structure Query Language*), el Dr. Edgar Frank Codd. Resulta conveniente rápidamente repasar estas reglas, de modo que para nuestra aplicación, apliquemos estas recomendaciones. En futuros proyectos, usted podrá diseñar eficientemente tipos de bases de datos.

LAS DOCE REGLAS DE CODD

- **Regla 0:** El sistema debe ser relacional, tanto la base de datos y administrador de sistema; es decir, un sistema de base de datos relacional debe utilizar sus facilidades relacionales (exclusivamente) para manejar la base de datos. Todo en una base de datos está guardado en un sistema relacional y cualquier elemento (usuario, tabla, índice, etc.) se guarda dentro de la misma base de datos.
- **Regla 1. 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 como valores en tablas. **No hay información que no esté en tablas.**
- **Regla 2. Del acceso garantizado.** Todos los datos deben ser accesibles sin ambigüedad. Cada valor individual en la base de datos debe ser direccionable especificando el nombre de la tabla, la columna que lo contiene y la llave primaria. Eso significa que todo dato puede ser ubicado teniendo el nombre de la tabla, el nombre del campo y el registro del que se trate.
- **Regla 3. Tratamiento 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», distinto de todos los valores regulares. Esto es debatible por la complejidad que acarrea en la operación del día a día, sin embargo todo motor de base de datos debe tener la posibilidad de utilizar valores nulos, aunque en la práctica es decisión del que modela la base de datos el utilizar esta posibilidad o no (En lo particular, yo recomiendo no usar valores nulos a menos que sea estrictamente necesario u obligatorio).

- **Regla 4. Catálogo basado en el modelo relacional.** El sistema debe soportar un catálogo en línea (la estructura misma de la base de datos). 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). Esto es lo que en varios motores de base de datos se denomina **esquema**. Como todo en la base de datos está definido dentro de la misma base de datos, incluso la estructura está en tablas (hay una tabla con los nombres de las tablas, una tabla con los nombres de los campos existentes, una tabla con los nombres de los índices, etc.).
- **Regla 5. Regla comprensiva del sublenguaje de los datos.** El sistema debe soportar por lo menos un lenguaje relacional que:
 - Tenga una sintaxis lineal.
 - Pueda 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.

Este punto es el punto de partida del [lenguaje SQL](#) tal y como se conoce actualmente.

- **Regla 6. Regla de actualización.** Todas las vistas que son teóricamente actualizables deben ser actualizables por el sistema, de manera transparente; es decir, que si en la base de datos se crea una vista de una tabla, se podría añadir un registro a la vista y eso significaría que se daría de alta el registro en la tabla original.
- **Regla 7.** Alto nivel de inserción, actualización y borrado, permitiendo el sistema realizar manipulación de datos de alto nivel, es decir, sobre conjuntos de registros. Esto significa que, además de leer muchos registros, se puede actualizar más de un registro a la vez, no sólo sobre registros individuales.
- **Regla 8. Independencia física de los datos.** Los clientes (aplicaciones, sistemas) permanecen inalterados a nivel lógico cuando se realicen cambios en las representaciones de almacenamiento o métodos de acceso. Ante cualquier cambio en la ubicación física de los datos, los queries escritos y probados no deben requerir modificaciones por dichos cambios en la ubicación física.

- **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, pero también debe ser posible que los queries que ya están escritos (si se modifica un tipo de dato, se añaden campos, se eliminan campos que no se requieren) no requieran cambios.
- **Regla 10. Independencia de la integridad.** Las reglas de integridad se deben especificar por separado de los programas o aplicaciones y se almacenan en la base de datos. Debe ser posible cambiar esas reglas 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 DBMS se introduce por primera vez
 - Cuando los datos existentes se redistribuyen en todo el sistema.

En términos reales, el usuario final o el desarrollador no debe preocuparse por la partición en la que estén los datos, el motor de base de datos debe saber dónde se encuentra cada dato y extraerlo cuando un query lo requiera.

- **Regla 12. Regla de la no subversión.** Si el sistema proporciona una interfaz de bajo nivel de registro, además de una interfaz relacional, que esa interfaz de bajo nivel no se pueda utilizar para subvertir el sistema (sin pasar por alto la seguridad relacional o limitación de integridad, por ejemplo).

REFERENCIAS BIBLIOGRAFICAS:

- A. Wagner. "Las Doce Reglas de las Bases de Datos". LinkedIn. Accedido el 12 de septiembre de 2023. [En línea]. Disponible: <https://es.linkedin.com/pulse/las-doce-reglas-de-bases-datos-ariel-alejandro-wagner>
- "Reglas de Codd de las bases de datos relacionales - DBA dixit". DBA dixit. Accedido el 12 de septiembre de 2023. [En línea]. Disponible: <http://dbadixit.com/reglas-codd-las-bases-datos-relacionales/>