



Reglas de Codd

Luis Fernando Franco Arellano
Soto Huerta Gustavo Isaac

August 30, 2024



Introducción

- ▶ Las 12 reglas de Codd fueron formuladas por Edgar F. Codd, un pionero en bases de datos relacionales.
- ▶ Estas reglas establecen los principios fundamentales que debe cumplir un sistema de bases de datos relacionales.



Regla 1: La Regla de la Información

Regla: Todos los datos se organizan en tablas lógicas (como hojas de cálculo), donde cada fila representa un registro y cada columna representa un tipo de información

Ejemplo:

- ▶ **Correcto:** Imagínate que llevas un inventario de productos en una tienda. Cada producto tiene su propio registro en una tabla, con su nombre, precio y cantidad en stock.
- ▶ **Incorrecto:** Guardar la información en notas dispersas o en diferentes formatos, como archivos de texto o en tu memoria, en lugar de un formato estructurado.



Regla 2: La Regla del Acceso Garantizado

Regla: Para encontrar cualquier dato, solo necesitas saber el nombre de la tabla, la clave principal (Primary Key) y el nombre de la columna donde esta almacenado el dato que buscas.

Ejemplo:

- ▶ **Correcto:** En una biblioteca, cada libro tiene un código único (como un ISBN) que te permite encontrarlo fácilmente entre miles de libros.
- ▶ **Incorrecto:** Imagina una biblioteca donde varios libros diferentes tienen el mismo código o ninguno en absoluto, lo que hace difícil encontrarlos.



Regla 3: El Tratamiento Sistemático de Valores Nulos

Regla: Los valores nulos pueden tener significado y el sistema debe tener la capacidad para manejarlos, reconociéndolos y tratándolos diferente de otros datos como vacíos o 0. La lógica aplicada debe ser correcta, y los valores nulos son independientes del dato de la columna, siendo esenciales para representar la ausencia de información de manera precisa.

Ejemplo:

- ▶ **Correcto:** En un formulario de inscripción, si alguien no tiene un segundo nombre, dejas ese campo vacío para que sea claro que no aplica.
- ▶ **Incorrecto:** Poner un valor arbitrario como "N/A" o "Desconocido", que podría confundirse con una entrada real.



Regla 4: Catálogo Activo y Dinámico Basado en el Modelo Relacional

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

Ejemplo:

- ▶ **Correcto:** En una aplicación de gestión de empleados, puedes acceder a la estructura y a los datos de las tablas que contienen la información sobre los empleados, departamentos, etc., desde un solo lugar.
- ▶ **Incorrecto:** Tener toda la información de la estructura de la base de datos en documentos separados que no están conectados directamente a la base de datos, lo que dificulta el acceso y la gestión.



Regla 5: Regla comprensiva del sub-lenguaje de los datos completos

Regla: El lenguaje que utilizas para trabajar con las bases de datos debe permitirte hacer cualquier cosa que necesites, como agregar nuevos datos, cambiar los existentes o hacer consultas complejas.

Ejemplo:

- ▶ **Correcto:** Piensa en una hoja de cálculo con una lista completa de empleados. Si quieres ver solo a los empleados de un departamento específico, puedes filtrar la hoja sin cambiar los datos originales.
- ▶ **Incorrecto:** Imagina que cada vez que quieres ver a los empleados de un departamento específico, tienes que borrar temporalmente a los otros empleados y luego volver a ingresarlos. Esto sería ineficiente y propenso a errores.



Regla 6: Actualización de Vistas

Regla: Todas las vistas que son teóricamente actualizables deben poder ser actualizadas por el sistema.

Ejemplo:

- ▶ **Correcto:** Imagina que tienes una lista en tu aplicación que muestra solo los empleados activos de tu empresa. Si decides cambiar el estado de un empleado a "inactivo" desde esta lista, la información debería actualizarse automáticamente en la base de datos principal.
- ▶ **Incorrecto:** Si, al intentar cambiar el estado de un empleado en la lista de empleados activos, el cambio no se refleja en la base de datos principal, lo que te obliga a hacer la actualización manualmente en otro lugar.



Regla 7: Alto nivel de inserción, actualización y cancelación

Regla: Se centra en simplificar la forma en que interactuamos con la información almacenada. En lugar de tener que programar cada operación individualmente como lo haríamos en lenguajes como C o Java, el objetivo es utilizar un lenguaje que sea más fácil de entender para los humanos. Esto significa que podemos manipular conjuntos de datos a la vez, lo que facilita la inserción, actualización y eliminación de información sin necesidad de escribir código complicado.

Ejemplo:

- ▶ **Correcto:** Piensa en una aplicación de recursos humanos donde puedes actualizar el salario de todos los empleados de un departamento con un solo comando, en lugar de tener que actualizar cada salario individualmente.



Regla 8: Independencia Física de Datos

Regla: La forma en que se almacenan los datos no debería importarte al interactuar con ellos. Pueden estar guardados de diferentes maneras, pero eso no afecta como los usas.

Ejemplo:

- ▶ **Correcto:** Imagina que reorganizas los muebles en tu casa, pero los visitantes aún pueden encontrar las habitaciones usando las mismas puertas y pasillos que siempre han usado, sin preocuparse por cómo están dispuestos los muebles.
- ▶ **Incorrecto:** Si cada vez que reorganizas los muebles en tu casa, tienes que modificar las puertas y pasillos para que los visitantes puedan llegar a las habitaciones, lo que sería inconveniente.



Regla 9: Independencia Lógica de Datos

Regla: Si cambias la forma en que ves los datos (por ejemplo, reorganizas las tablas), no debería afectar a las aplicaciones que usan esos datos.

Ejemplo:

- ▶ **Correcto:** Supón que añades una nueva columna con información adicional en una hoja de cálculo de clientes. Las fórmulas y gráficos que utilizan los datos existentes siguen funcionando sin necesidad de ser modificados.
- ▶ **Incorrecto:** Si al añadir una nueva columna en la hoja de cálculo, todas las fórmulas y gráficos que utilizaban los datos existentes se rompen, requiriendo que los ajustes manualmente.



Regla 10: Independencia de la integridad

Regla: La regla que garantiza que los datos sean correctos y completos deben estar separadas de las aplicaciones que lo utilizan.

Ejemplo:

- ▶ **Correcto:** Imagina que tienes un formulario en una aplicación de registro de usuarios que valida automáticamente la dirección de correo electrónico para asegurarse de que tiene el formato correcto antes de guardarla en la base de datos.
- ▶ **Incorrecto:** Permitir que los usuarios ingresen cualquier texto en el campo de correo electrónico sin verificar si es un formato válido, lo que podría llevar a errores y datos inconsistentes.



Regla 11: Independencia de la distribución

Regla: Si los datos están en diferentes lugares físicos, el sistema debe poder manejar eso sin problemas, como si estuvieran todos en un solo lugar.

Ejemplo:

- ▶ **Correcto:** Usar una interfaz gráfica en una aplicación de inventario que permita a los usuarios buscar productos, actualizar existencias y generar reportes sin necesidad de conocimientos técnicos avanzados.
- ▶ **Incorrecto:** Requerir que los usuarios editen archivos de configuración o utilicen scripts complejos para acceder y gestionar la información de inventario.



Regla 12: Regla de la no subversión

Regla: Aunque puedas acceder a la base de datos de formas más avanzadas, no deberías poder evitar las reglas y restricciones que has establecido para garantizar la integridad de los datos.

Ejemplo:

- ▶ **Correcto:** En una aplicación de gestión de proyectos, añadir un nuevo campo para rastrear una métrica adicional (como el tiempo estimado) sin afectar el resto del sistema ni requerir grandes cambios.
- ▶ **Incorrecto:** Si cada vez que necesitas añadir un nuevo tipo de información, debes rediseñar completamente la base de datos y actualizar todas las aplicaciones que la utilizan.



Conclusión

- ▶ Las reglas de Codd son fundamentales para la integridad y consistencia de las bases de datos relacionales.
- ▶ Cumplir con estas reglas asegura que el sistema de bases de datos sea confiable y eficiente.