

Tarea 6. Reglas CODD

Lemus González, Javier Issac
javierissaclemusgonzalez@gmail.com
 Universidad Nacional Autónoma de México

el recorrido registro a registro, éste no puede utilizarse para incumplir las reglas relacionales.

I. REGLAS CODD

El propósito de las reglas Codd, es establecer estándares para el diseño y la gestión de bases de datos relacionales. Estas reglas fueron propuestas por Edgar F. Codd, un científico informático conocido por su trabajo pionero en el campo de las bases de datos relacionales.

El objetivo principal de las reglas Codd es garantizar la integridad, la consistencia y la eficiencia de las bases de datos relacionales. Estas reglas proporcionan pautas para organizar los datos de manera que se minimicen las redundancias y se maximice la capacidad de realizar consultas y manipulaciones de datos de manera eficiente.

1. **Información.** Toda la información de la base de datos debe estar representada explícitamente en el esquema lógico. Es decir, todos los datos están en las tablas.

2. **Acceso garantizado.** Todo dato es accesible sabiendo el valor de su clave y el nombre de la columna o atributo que contiene el dato.

3. **Tratamiento sistemático de los valores nulos.** El DBMS debe permitir el tratamiento adecuado de estos valores.

4. **Catálogo en línea basado en el modelo relacional.** Los metadatos deben de ser accesibles usando un esquema relacional.

5. **Sublenguaje de datos completo.** Al menos debe de existir un lenguaje que permita el manejo completo de la base de datos. Este lenguaje, por lo tanto, debe permitir realizar cualquier operación.

6. **Actualización de vistas.** El DBMS debe encargarse de que las vistas muestren la última información.

7. **Inserciones, modificaciones y eliminaciones de dato nivel.** Cualquier operación de modificación debe actuar sobre conjuntos de filas, nunca deben actuar registro a registro.

8. **Independencia física.** Los datos deben de ser accesibles desde la lógica de la base de datos aún cuando se modifique el almacenamiento.

9. **Independencia lógica.** Los programas no deben verse afectados por cambios en las tablas.

10. **Independencia de integridad.** Las reglas de integridad deben almacenarse en la base de datos (en el diccionario de datos), no en los programas de aplicación.

11. **Independencia de la distribución.** El sublenguaje de datos debe permitir que sus instrucciones funcionen igualmente en una base de datos distribuida que en una que no lo es.

12. **No subversión.** Si el DBMS posee un lenguaje que permite

II. INDEPENDENCIA FÍSICA Y LÓGICA

La independencia física se refiere a la capacidad de modificar el almacenamiento físico de los datos sin afectar la forma en que los usuarios o las aplicaciones interactúan con esos datos. En otras palabras, los cambios en la forma en que se almacenan los datos no deben requerir cambios en el acceso o en las aplicaciones que utilizan esos datos. Esto implica que los usuarios y las aplicaciones no deben verse afectados por cambios en la ubicación física de los datos, la forma en que se almacenan o cómo se accede a ellos.

Por ejemplo, si se decide cambiar la base de datos de un sistema de almacenamiento en disco a un sistema de almacenamiento en memoria, o si se cambia de un sistema de archivos a una base de datos distribuida, los usuarios y las aplicaciones deben poder continuar operando sin necesidad de modificaciones significativas. Esta independencia permite una mayor flexibilidad y mantenibilidad del sistema, ya que los cambios en el almacenamiento pueden realizarse sin afectar a las aplicaciones que utilizan los datos.

La independencia lógica se refiere a la capacidad de modificar la estructura lógica de la base de datos sin afectar las aplicaciones que la utilizan. Esto implica que los cambios en el esquema de la base de datos, como la adición o eliminación de tablas, la modificación de las relaciones entre tablas o la inclusión de nuevos atributos, no deben requerir cambios en las aplicaciones que acceden a esos datos.

Por ejemplo, si se agrega una nueva tabla a la base de datos para almacenar información adicional, las aplicaciones existentes deben poder continuar funcionando sin necesidad de modificaciones siempre y cuando no necesiten acceder directamente a esa nueva tabla. Esto proporciona flexibilidad en el diseño y la evolución de la base de datos, ya que se pueden realizar cambios en la estructura lógica para adaptarse a nuevos requisitos sin afectar a las aplicaciones existentes.

REFERENCIAS

- [1] J. Sánchez, "Principios sobre bases de datos relacionales," Informe, Creative Commons, 2004, vol. 11, pp. 20.
- [2] J. Quiroz, "El modelo relacional de bases de datos," Boletín de Política Informática, vol. 6, pp. 53-61, 2003.