Las reglas de Codd

•En 1990 Codd se preocupó por los SGBD que decían ser relacionales y no lo eran

•Había SGBD que utilizaban tablas, pero no tablas relacionales

•Incumplían las normas de las relaciones definidas en su modelo

•Por ello enunció 12 reglas que los SGBD relacionales debían de cumplir

•En la práctica es difícil cumplir las 12

•Pero, un SGBD es más relacional cuantas más reglas cumpla

Regla 1: Información

•Todos los datos deben estar almacenados en las tablas

•Esas tablas deben de cumplir las premisas del modelo relacional

•No puede haber información a la que accedemos por otra vía

Regla 2: Acceso garantizado

•Cualquier dato es accesible sabiendo la clave de su fila y el nombre de su columna o atributo

•Por ejemplo el “Sánchez” es un dato al que podremos acceder conociendo la clave de la persona en concreto y usando el atributo “Primer apellido”

•Si a un dato no podemos acceder de esta forma, no estamos usando un modelo relacional

Regla 3: Tratamiento sistemático de los valores nulos

•Esos valores pueden dar significado a la columna que los contiene (una persona sin teléfono, tendrá valor nulo en el teléfono)

•El SGBD tiene que tener la capacidad de manejar valores nulos•El SGBD reconocerá este valor como un valor distinto de cualquier otro

•El SGBD sabrá aplicarle la lógica apropiada

•Es un valor independiente del tipo de datos de la columna

Regla 4: Catálogo en línea relacional

•El catálogo en línea es el diccionario de datos

•El diccionario de datos se debe de poder consultar usando las mismas técnicas que para los datos

•Los metadatos, por tanto, se organizan también en tablas relacionales

•Si SELECT es la instrucción que consulta datos, también será la que consulta los metadatos

Regla 5: Sublenguaje de datos completo

•Al menos tiene que existir un lenguaje capaz de hacer todas las funciones del SGBD

•No puede haber funciones fuera de ese lenguaje

•Puede haber otros lenguajes en el SGBD para hacer ciertas tareas

•Pero esas tareas también se deben poder hacer con el “lenguaje completo”

Regla 6: Vistas actualizadas

•Las vistas tienen que mostrar información actualizada

•No puede haber diferencia entre los datos de las vistas y los datos de las tablas base

Regla 7: Inserciones, modificaciones y eliminaciones de alto nivel

•La idea es que el lenguaje que maneja la BD sea muy humano•Eso implica que las operaciones DML trabajen con conjuntos de filas a la vez

•Para modificar, eliminar o añadir datos no hará falta programar de la forma en la que lo hacen los lenguajes de tercera generación como C o Java

Regla 8: Independencia física

•Cambios en la física de la BD no afecta a las aplicaciones ni a los esquemas lógicos

•El acceso a las tablas (elemento lógico) no cambia porque la física de la base de datos cambie

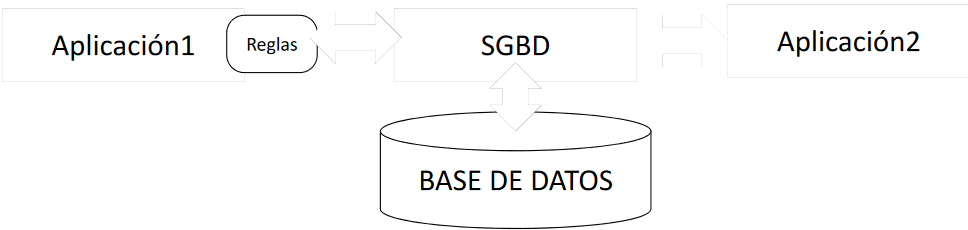
Regla 9: Independencia lógica

•Cambios en el esquema lógico (tablas) de la BD no afectan al resto de esquemas

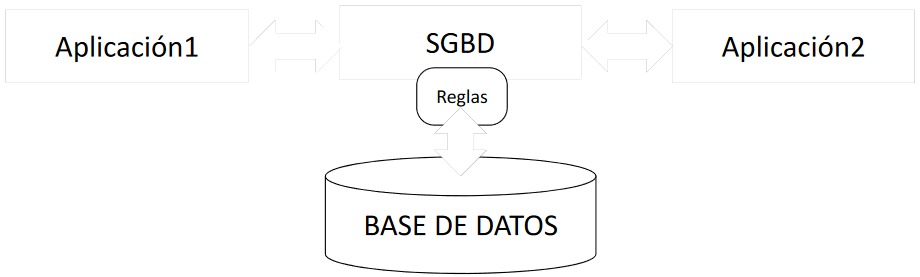
•Si cambiamos nombres de tabla, o de columna o modificamos información de las filas, las aplicaciones (esquema externo) no se ven afectadas

•Es más difícil de conseguir

Regla 10: Independencia de integridad

•Las reglas de integridad (restricciones) deben de ser gestionadas y almacenadas por el SGBD

•Las reglas de integridad (restricciones) deben de ser gestionadas y almacenadas por el SGBD



Regla 11: Independencia de distribución

•Que la base de datos se almacene o gestione de forma distribuida en varios servidores, no afecta al uso de la misma ni a la programación de las aplicaciones de usuario

•El esquema lógico es el mismo independientemente de si la BD es distribuida o no

Regla 12: No subversión

•La base de datos no permitirá que exista un lenguaje o forma de acceso, que permita saltarse las reglas anteriores

\*Regla 0

•Un SGBD relacional debe gestionar sus BD de forma completa usando el modelo relaciona

Fuente:

Centro don Bosco. (s. f.). Las 12 reglas de Codd. Recuperado 20 de octubre de 2020, de https://jorgesanchez.net/presentaciones/bases-de-datos/modelo-relacional/reglas-codd.pdf

Bitmap:

Los índices BitMap se utilizan cuando las columnas tienen baja cardinalidad y serán frecuentemente utilizadas en consultas

https://www.geeksforgeeks.org/bitmap-indexing-in-dbms/

HashTable:

Se utilizan cuando se harán frecuentes ediciones a los elementos de las tablas.

Fuente: conclusiones de lo que lei.

Los índices de hash se usan principalmente para búsquedas de puntos y no para exámenes de intervalos.

Es preferible un índice de hash sobre un índice no agrupado cuando las consultas usan predicados de igualdad y la cláusula WHERE se asigna a todas las columnas de clave de índice.

https://docs.microsoft.com/es-es/sql/relational-databases/in-memory-oltp/indexes-for-memory-optimized-tables?view=sql-server-ver15

https://docs.microsoft.com/es-es/sql/relational-databases/indexes/indexes?view=sql-server-ver15