



---

# TEORÍA DE BASES DE DATOS

## TAREA

---

Estudiante: Pérez Gutiérrez Sandra Susana

Grupo: 1

19 de octubre de 2020

sandraconnors271198@gmail.com



Semestre 2021-1

Tarea de Bases de datos.



## Índice

1. **12 reglas de Codd** 4
2. **Indice Hash, BitMap y B-Tree** 5



# 1. 12 reglas de Codd

Estas reglas tienen el objetivo de establecer reglas en el diseño de bases de datos relacionales para homogeneizar su planeación y construcción.

Regla 0: Regla de fundación. Cualquier sistema que se proclame como relacional, debe ser capaz de gestionar sus bases de datos enteramente mediante sus capacidades relacionales.

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 en el nivel Lógico exactamente de una manera: con valores en tablas.

Regla 2: Regla del acceso garantizado. Todos los datos deben ser accesibles sin ambigüedad.

Regla 3: Regla del 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 sea sistemática y distinta 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, que da acceso a la estructura de la base de datos y que debe ser accesible a los usuarios autorizados.

Regla 5: 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.
- Tenga soporte de operaciones de definición de datos, operaciones de manipulación de datos (actualización así como la recuperación), de control de la seguridad e integridad y operaciones de administración de transacciones.

Regla 6: Regla de actualización de vistas. Todas las vistas que son teóricamente actualizables deben poder ser actualizadas por el sistema.

Regla 7: Alto nivel de inserción, actualización y borrado. El sistema debe permitir la manipulación de alto nivel en los datos, es decir, sobre conjuntos de tuplas. Esto significa que los datos no solo se pueden recuperar de una base de datos relacional a partir de filas múltiples y/o de tablas múltiples, sino que también pueden realizarse inserciones, actualización y borrados sobre varias tuplas y/o tablas al mismo tiempo y no solo sobre registros individuales.

Regla 8: Independencia física de los datos. Los programas de aplicación y actividades del terminal permanecen inalterados a nivel lógico aunque realicen cambios en las representaciones de almacenamiento o métodos de acceso.



Regla 9: Independencia lógicas de los datos. Los programas de aplicación y actividades del terminal permanecen inalterados a nivel lógico aunque se realicen cambios a las tablas base que preserven la información. La independencia de datos lógica es más difícil de lograr que la independencia física de datos.

Regla 10: Independencia de la integridad. Las restricciones de integridad se deben especificar por separado de los programas de aplicación y almacenarse en la base de datos. Debe ser posible cambiar esas restricciones sin afectar innecesariamente a las aplicaciones existentes.

Regla 11: Independencia de la distribución. La distribución de porciones de base de datos en distintas localizaciones debe ser invisible a los usuarios de la base de datos.

Regla 12: La regla de la no subversión. Si el sistema proporciona una interfaz de bajo nivel de registro, aparte de una interfaz relacional, esa interfaz de bajo nivel no debe permitir su utilización para subvertir el sistema.

## 2. Indice Hash, BitMap y B-Tree

Indice Hash: es una estructura de datos que asocia llaves o claves con valores. La operación principal que soporta de manera eficiente es la búsqueda: permite el acceso a los elementos (teléfono y dirección, por ejemplo) almacenados a partir de una clave generada (usando el nombre o número de cuenta, por ejemplo). Funciona transformando la clave con una función hash en un hash, un número que identifica la posición (casilla o cubeta) donde la tabla hash localiza el valor deseado.

Indice BitMap: se logra almacenando una lista de rowids de las filas de la tabla con el valor clave. Oracle almacena cada valor clave en forma repetida para cada fila. En un índice bitmap, en vez de una lista de rowids, Oracle crea un mapa de bits para cada valor clave del índice. Cada bit del mapa corresponde a un rowid posible. Si el bit está en 1, significa que el rowid contiene dicho valor clave. Una función interna de Oracle convierte la posición del bit en el rowid correspondiente, de modo tal que los índices bitmap ofrecen la misma funcionalidad que los índices B-tree, a pesar de la diferente representación interna. Si la cantidad de valores diferentes del índice es chica, entonces el índice bitmap será muy eficiente en cuanto al uso de espacio físico.

Indice B-Tree: es el tipo de índice más común en una base de datos Oracle. Es el índice default, es decir que si uno crea un índice sin especificar el tipo, Oracle lo creará de tipo B-Tree. La estructura de un índice B-Tree tiene la forma de un árbol invertido. Las estructuras superiores se llaman ramas y la estructura inferior constituyen las hojas. Generalmente los índices B-Tree tienen uno o más niveles de ramas. Cada rama de un nivel contiene información sobre otras ramas del nivel inmediatamente inferior.