UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

BASES DE DATOS

ING. FERNANDO ARREOLA FRANCO

GRUPO: 01

ARCE HERNÁNDEZ CHRISTIAN ALEXIS

NO. DE CUENTA: 314159993

SEMESTRE 2021-1

INDICES.

HASH.

Todas las tablas optimizadas para memoria deben tener como mínimo un índice porque son los índices los que conectan las filas. En una tabla optimizada para memoria, todos los índices también son optimizados para memoria. Los índices de hash son uno de los tipos de índice posibles en una tabla optimizada para memoria.

Los índices de hash constan de una matriz de punteros, y cada elemento de la matriz se llama "cubo de hash".

- Cada depósito tiene 8 bytes, que se usan para almacenar la dirección de memoria de una lista de vínculos de entradas de índice.
- Cada entrada es un valor correspondiente a una clave de índice, además de la dirección de su fila correspondiente en la tabla subyacente optimizada para memoria.
- Cada entrada apunta a la siguiente entrada en una lista de vínculos de entradas, todas ellas encadenadas al depósito actual.

El número de cubos debe especificarse en el momento de definir los índices:

- Cuanto menor sea la proporción de depósitos con respecto a las filas de la tabla o valores distintos, más larga será la lista de vínculos de depósito promedio.
- Las listas de vínculos cortas se ejecutan más rápidamente que las listas de vínculos largas.
- El número máximo de cubos en los índices de hash es de 1 073 741 824.

BITMAP.

El índice de mapa de bits se utiliza cuando la cardinalidad de los datos es baja.

Aquí, el género tiene valor con baja cardinalidad. Los valores pueden ser masculinos, femeninos y otros.

Por lo tanto, si creamos un árbol binario para estos 3 valores mientras lo buscamos, tendremos un recorrido innecesario.

En las estructuras de mapa de bits, se crea una matriz bidimensional con una columna para cada fila de la tabla que se está indexando. Cada columna representa un valor distinto dentro del índice de mapa de bits. Esta matriz bidimensional representa cada valor dentro del índice multiplicado por el número de filas en la tabla.

En el momento de la recuperación de la fila, Oracle descomprime el mapa de bits en los buffers de datos de RAM para que pueda escanearse rápidamente en busca de valores coincidentes. Estos valores coincidentes se entregan a Oracle en forma de una lista de ID de fila, y estos valores de ID de fila pueden acceder directamente a la información requerida.

B-TREE.

El índice de árbol B almacena los datos como formato de árbol binario. El índice es un objeto de esquema que almacena algún tipo de entrada para cada valor para la columna indexada. Por lo tanto, siempre que se realice una búsqueda en esas columnas, se verifica en el índice la ubicación exacta de ese registro para acceder rápidamente. Algunos puntos sobre la indexación:

- Para buscar una entrada en el índice, se utiliza algún tipo de algoritmo de búsqueda binario.
- Cuando la cardinalidad de los datos es alta, el índice b-tree es perfecto para usar.
- El índice hace que el DML sea lento, ya que, para cada registro, debe haber una entrada en el índice para la columna indexada.
- Por lo tanto, si no es necesario, evitar crear índices.

Referencias:

Guía de diseño y de arquitectura de índices de SQL Server. Recuperado de: https://docs.microsoft.com/es-es/sql/relational-databases/sql-server-index-design-guide?view=sql-server-ver15#hash_index

Índice de mapa de bits. Recuperado de: https://riptutorial.com/es/oracle/example/30658/indice-de-mapa-de-bits

índice de árbol b. Recuperado de: https://riptutorial.com/es/oracle/example/30657/indice-de-arbol-b