Índices de Bases de Datos.

Hash: La técnica de Hashig (o asociación) nos permite evitar este acceso adicional. Disponemos de una función (función de hash) que tiene como dominio las posibles llaves de búsqueda y como recorrido las posibles posiciones en memoria secundaria donde se encuentran los datos. A pesar de que el conjunto de las posibles llaves puede ser muy grande, el conjunto de llaves efectivamente almacenadas en la BD es mucho más pequeño.

Si h es la función de hash, se busca que la probabilidad de que h(k1) = h(k2) para dos elementos k1 6= k2 almacenados en la BD sea pequeña, puede ocurrir que h(k1) = h(k2). A esto se llama colisión.

Todos los valores que colisionan se almacenan en un mismo bloque (como una lista ligada). Cada elemento en el bloque apunta a la dirección física representada por la llave de búsqueda. Una buena función de hash tiene pocas colisiones

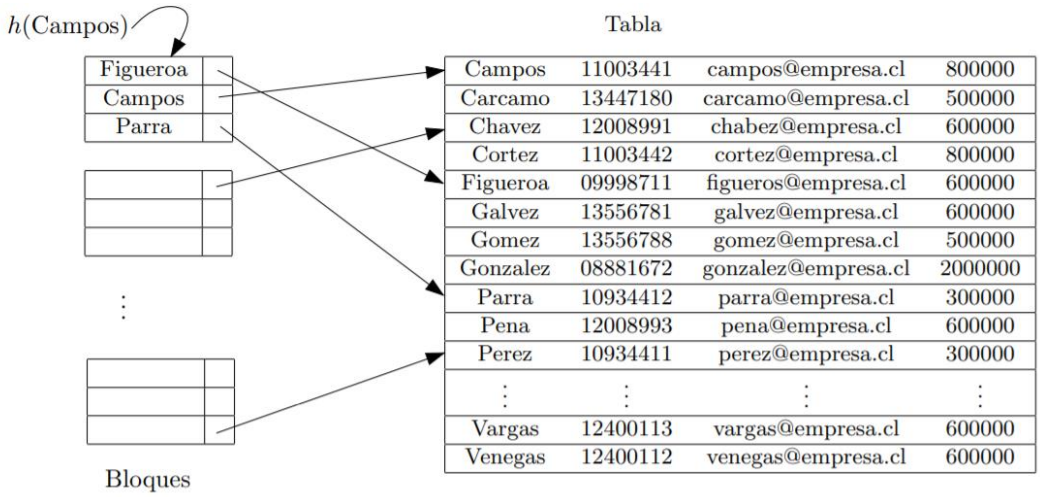
Para encontrar los datos asociados a una llave de búsqueda se debe:

• Aplicar h a la llave para obtener la dirección de un bloque.

• Buscar secuencialmente dentro del bloque el puntero asociado a esa llave.

• Seguir el puntero para acceder a los datos. Es bastante eficiente si h distribuye uniformemente las llaves en los bloques (minimiza colisión).

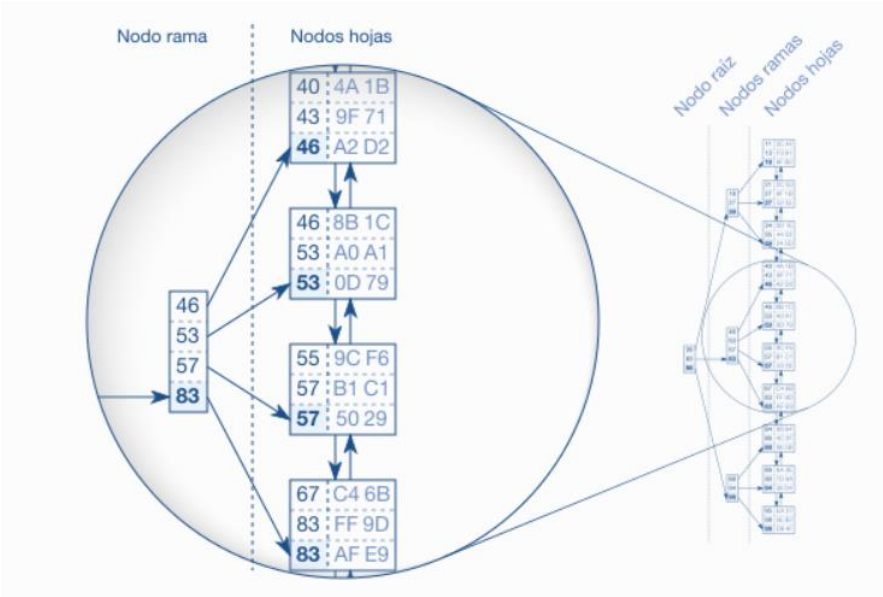
Algunas desventajas que tiene este tipo de índice son que la función h debe ser construida antes de comenzar a ocupar la BD y no puede cambiar, la cantidad de bloques utilizables se debe especificar previamente y Si la BD crece mucho, la probabilidad de colisión aumenta por lo tanto baja la eficiencia.



Bitmap: se utiliza cuando la cardinalidad de los datos es baja. 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.

B-Tree: Los nodos hojas del índice están almacenados en un orden aleatorio, es decir su posición en el disco no corresponde a la posición lógica según el orden del índice.

Un B-tree se ocupa porque una base de datos necesita una segunda estructura para encontrar rápidamente los datos dentro de las hojas mezcladas.



Referencias:

Perez Rojas, J. (s. f.). *Indexación y Hashing*. Bases de Datos. Recuperado 19 de octubre de 2020, de http://dns.uls.cl/~ej/daa_08/teo_daa_2008/hashing/indices.pdf

Oracle Database - Índice de mapa de bits | oracle Tutorial. (s. f.). RIP Tutorial. Recuperado 19 de octubre de 2020, de https://riptutorial.com/es/oracle/example/30658/indice-de-mapa-de-bits

El árbol de búsqueda equilibrado (B-Tree) en las bases de datos SQL. (s. f.). Use the index, Luke! Recuperado 19 de octubre de 2020, de https://use-the-index-luke.com/es/sql/i%CC%81ndice-anatomi%CC%81a/b-tree