

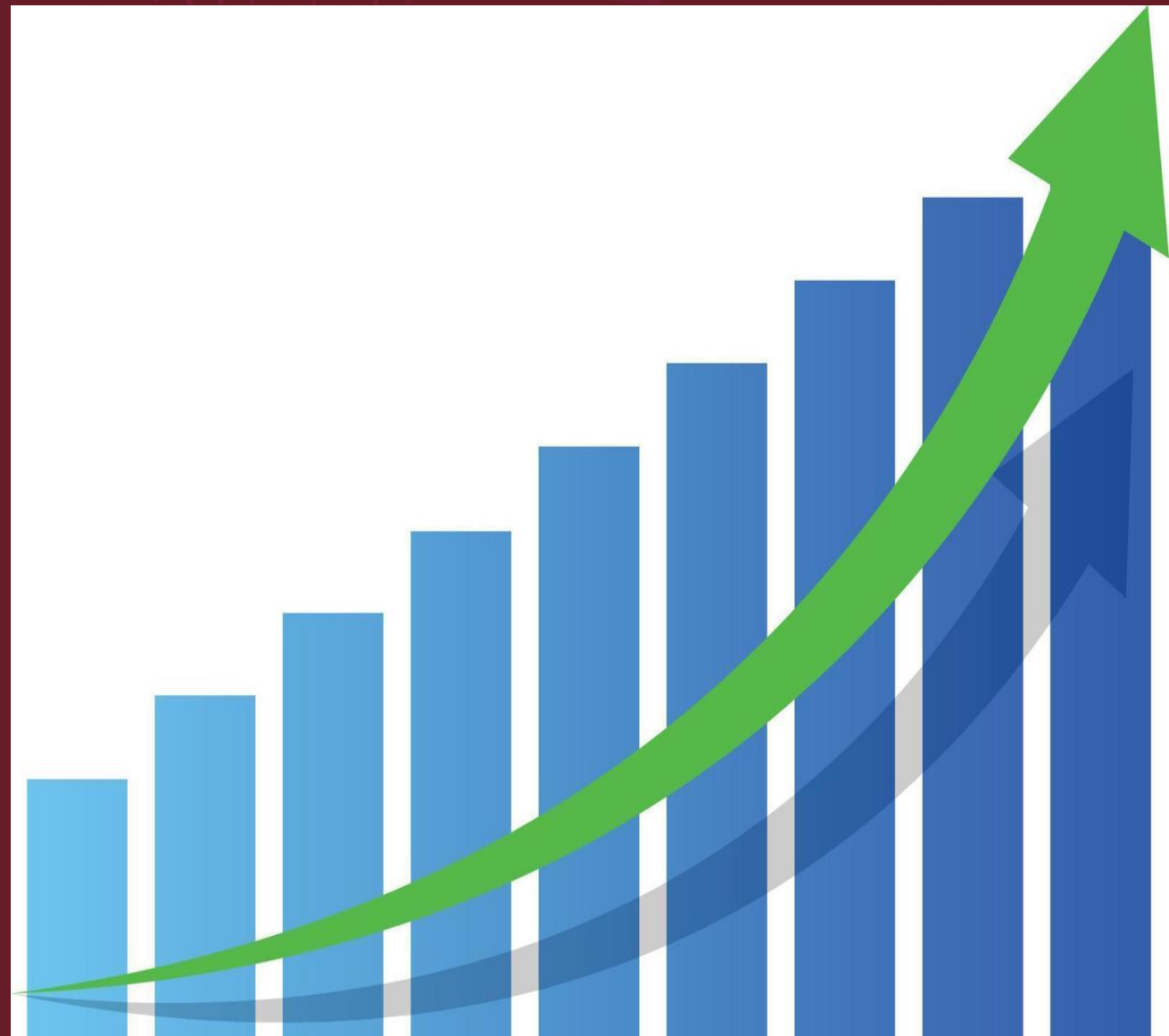


Indexación en bases de datos

Trabajo hecho por: Renato Oscar Corrales Peña

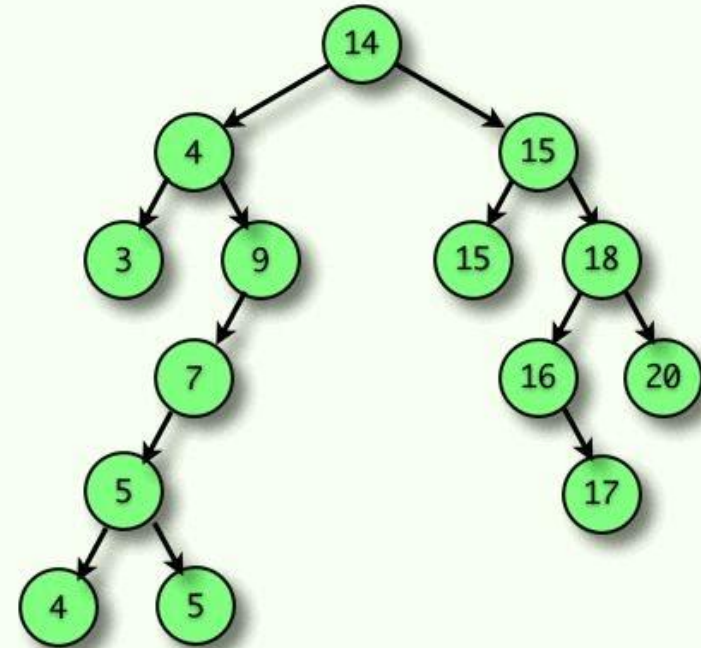
¿Por qué es importante la indexación en las bases de datos?

- Porque la indexación es un componente que nos ayuda a mejorar el rendimiento y la velocidad de las consultas



¿De qué trata?

- Consiste en crear estructuras de datos especiales (índices) para almacenar los conjuntos de datos para que estén mejor optimizados.



En resumen

- La indexación es una forma de optimizar el rendimiento de una base de datos al minimizar la cantidad de accesos al disco necesarios cuando se procesa una consulta. Es una técnica de estructura de datos que se utiliza para localizar y acceder rápidamente a los datos en una base de datos.

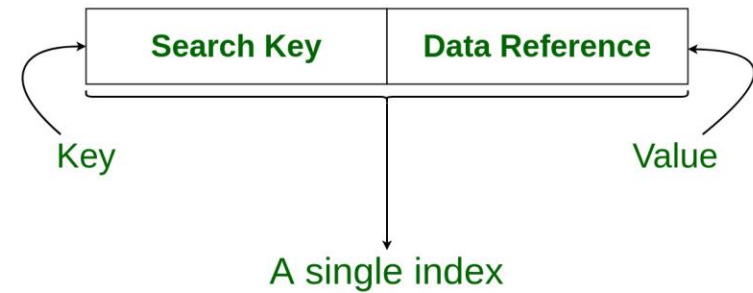
¿Que es un DBMS?

- Un DBMS (database management system) es un conjunto de aplicativos y programas que se encargan de administrar la información que se incluye en una base de datos. Los DBMS permiten hacer consultas, modificaciones y añadir campos y registros. También actúan como un intermediario entre el usuario, la base de datos y la información que contiene.

¿Cómo indexar una base de datos con un DBMS?

- Para crear los índices en una base de datos se utilizan unas pocas columnas de la misma. La primera columna de la tabla de base de datos contendrá la clave de búsqueda, que es una copia de la clave principal o candidata. Estos valores pueden estar ordenados para un acceso rápido a los datos correspondientes. La segunda columna contiene punteros, que son referencias a la ubicación en el disco donde se puede encontrar el valor de la clave específica.

Structure of an Index in Database



Atributos

- **Tipos de acceso** : se refiere al tipo de acceso, como búsqueda basada en valores, acceso por rango, etc.
- **Tiempo de acceso** : se refiere al tiempo necesario para encontrar un elemento de datos en particular o un conjunto de elementos.
- **Tiempo de inserción** : se refiere al tiempo necesario para encontrar el espacio adecuado e insertar un nuevo dato.
- **Tiempo de eliminación** : tiempo necesario para encontrar un elemento y eliminarlo, así como para actualizar la estructura del índice.
- **Space Overhead** : Se refiere al espacio adicional requerido por el índice.

Organización de archivos secuenciales o archivo de índice ordenado

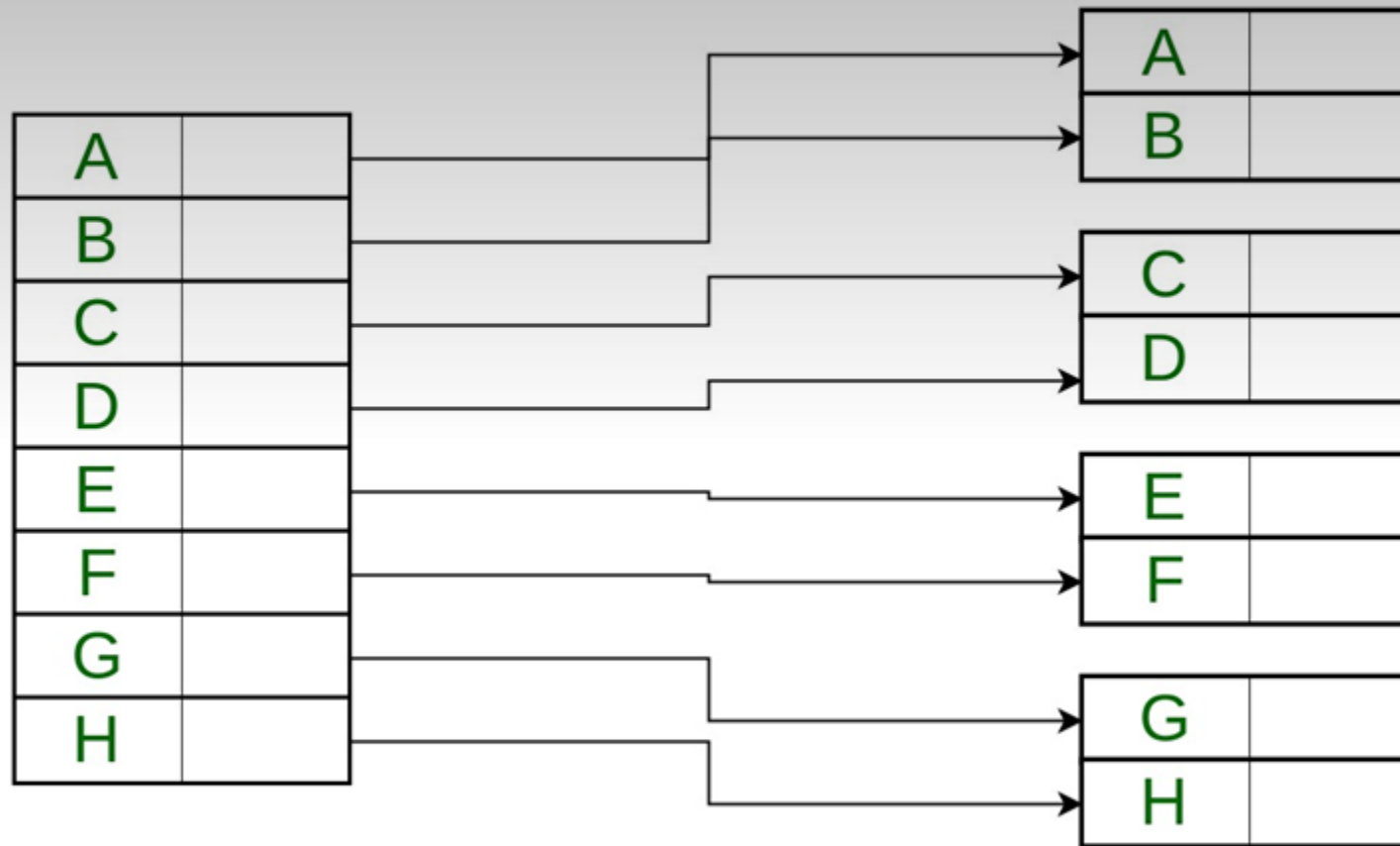


- En este mecanismo de organización, los índices se crean a partir de una ordenación de los valores. Este método, que es rápido y tradicional, permite que los archivos ordenados o secuenciales almacenen datos en formatos densos o dispersos.

Indice denso

- Para cada valor de clave de búsqueda en el archivo de datos, hay un registro de índice.
- Este registro contiene la clave de búsqueda y también una referencia al primer registro de datos con ese valor de clave de búsqueda.

Dense Index



Data File

Index Record

For every search value in a Data File,

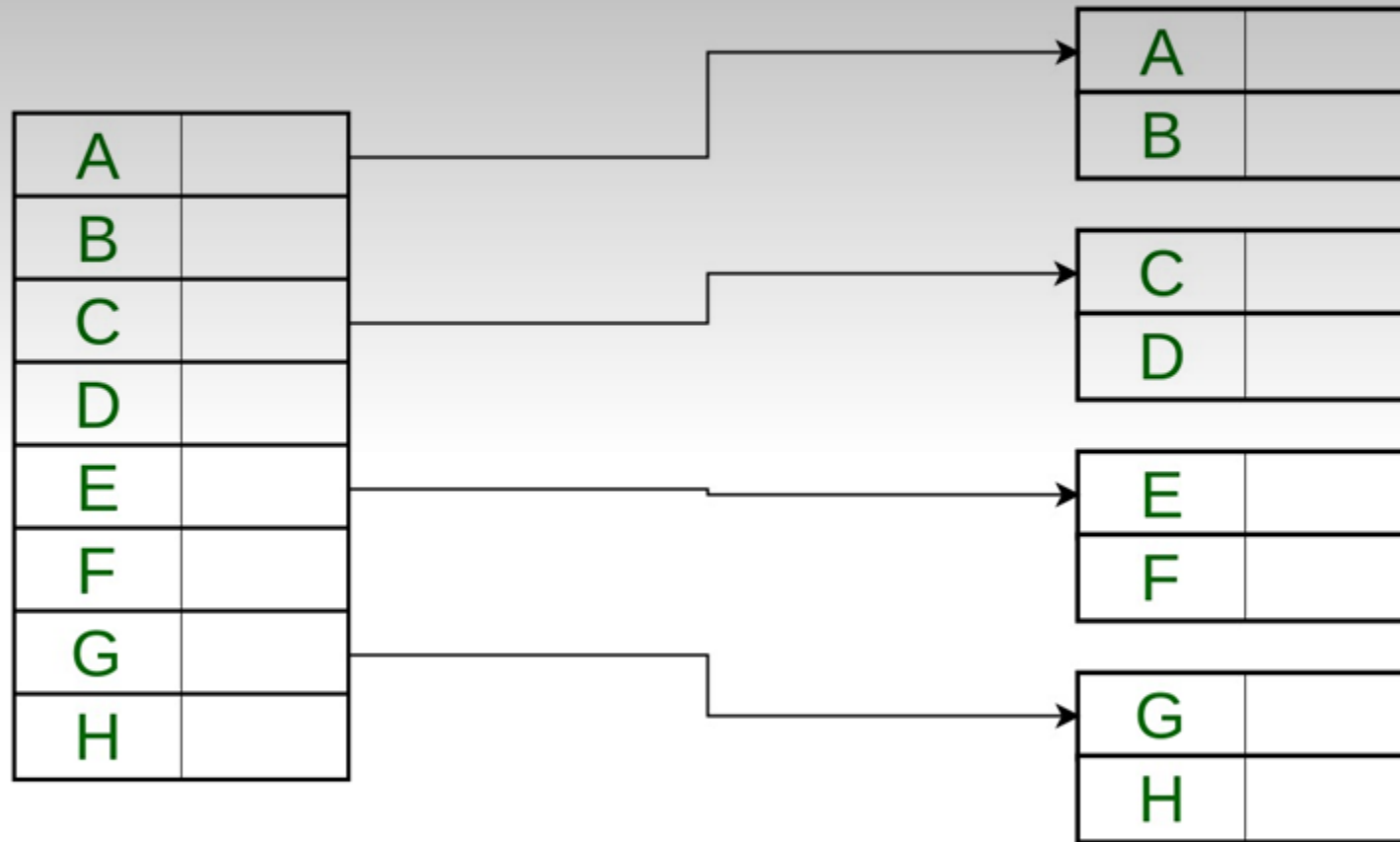
There is an Index Record.

Hence the name **Dense Index**.

Índice disperso

- El registro de índice aparece solo para algunos elementos en el archivo de datos. Cada elemento apunta a un bloque como se muestra.
- Para localizar un registro, buscamos el registro de índice con el mayor valor de clave de búsqueda menor o igual que el valor de clave de búsqueda que estamos buscando.
- Comenzamos en ese registro al que apunta el registro índice y avanzamos junto con los punteros en el archivo (es decir, secuencialmente) hasta que encontramos el registro deseado.

Sparse Index



Data File

Index Record

For very few
search value
in a Data File,

There is an
Index Record.

Hence the name
Sparse Index.

Organización de archivos hash

- los índices se basan en que los valores se distribuyen uniformemente en un rango de cubos. Los cubos a los que se asigna un valor están determinados por una función llamada función hash.

- Existen principalmente tres métodos de indexación:
- Indexación agrupada
- Indexación secundaria o no agrupada
- Indexación multinivel



Indexación agrupada

- Cuando se almacenan más de dos registros en el mismo archivo, este tipo de almacenamiento se conoce como indexación en clúster. Al usar la indexación de clústeres, podemos reducir el costo de la búsqueda, ya que se almacenan múltiples registros relacionados con lo mismo en un solo lugar. Esto también facilita la unión frecuente de más de dos tablas (registros).

- El índice de agrupamiento se define en un archivo de datos ordenados. El archivo de datos se ordena en un campo no clave. En algunos casos, el índice se crea en columnas de clave no primaria que pueden no ser únicas para cada registro. En tales casos, para identificar los registros más rápido, agruparemos dos o más columnas para obtener los valores únicos y crear un índice a partir de ellos.

- Este método se conoce como el índice de agrupamiento. Básicamente, los registros con características similares se agrupan y se crean índices para estos grupos. Por ejemplo, los estudiantes que estudian en cada semestre se agrupan. Es decir, se agrupan los estudiantes del 1er semestre, los estudiantes del 2do semestre, los estudiantes del 3er semestre, etc.

Indexacion primaria

- Este es un tipo de indexación agrupada en la que los datos se ordenan según la clave de búsqueda y la clave principal de la tabla de la base de datos se usa para crear el índice. Es un formato predeterminado de indexación donde induce la organización secuencial de archivos. Como las claves primarias son únicas y se almacenan ordenadas, el rendimiento de la operación de búsqueda es bastante eficiente.

Indexacion secundaria

- Un índice no agrupado simplemente nos dice dónde se encuentran los datos, es decir, nos brinda una lista de punteros virtuales o referencias a la ubicación donde se almacenan realmente los datos. Los datos no se almacenan físicamente en el orden del índice. En cambio, los datos están presentes en los Nodes hoja.

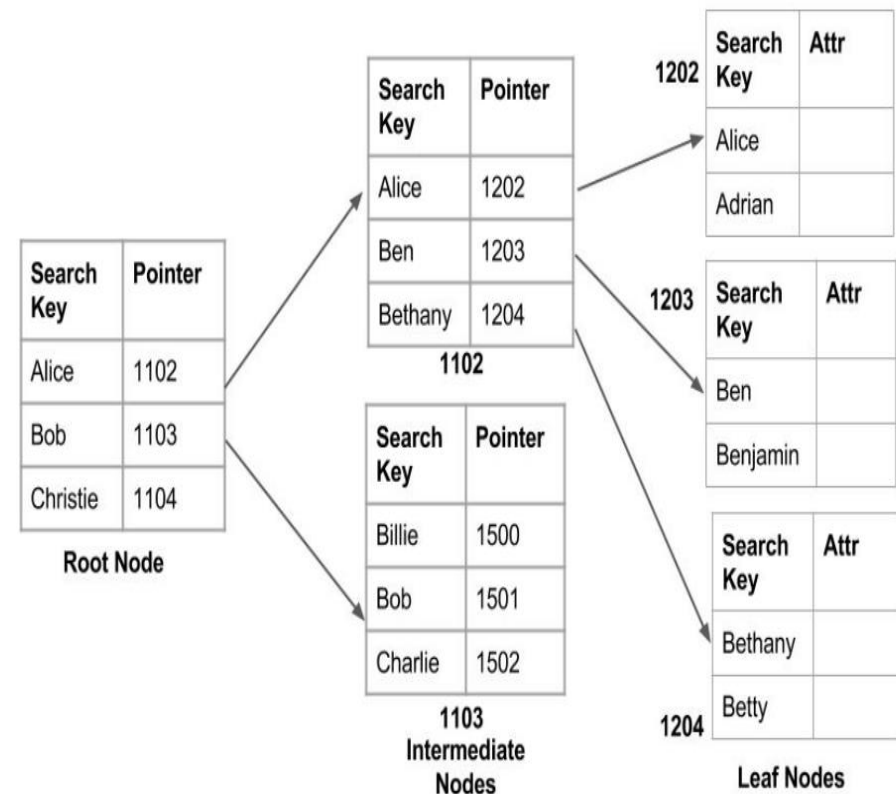
Ejemplo

- La página de contenido de un libro. Cada entrada nos da el número de página o la ubicación de la información almacenada. Los datos reales aquí (información en cada página del libro) no están organizados, pero tenemos una referencia ordenada (página de contenido) de dónde se encuentran realmente los puntos de datos. Solo podemos tener un ordenamiento denso en el índice no agrupado, ya que el ordenamiento disperso no es posible porque los datos no están organizados físicamente en consecuencia.

¿Mejor que la indexación agrupada?

- El índice no agrupado requiere más tiempo en comparación con el índice agrupado porque se realiza una cierta cantidad de trabajo adicional para extraer los datos siguiendo más el puntero. En el caso de un índice agrupado, los datos están directamente presentes delante del índice.

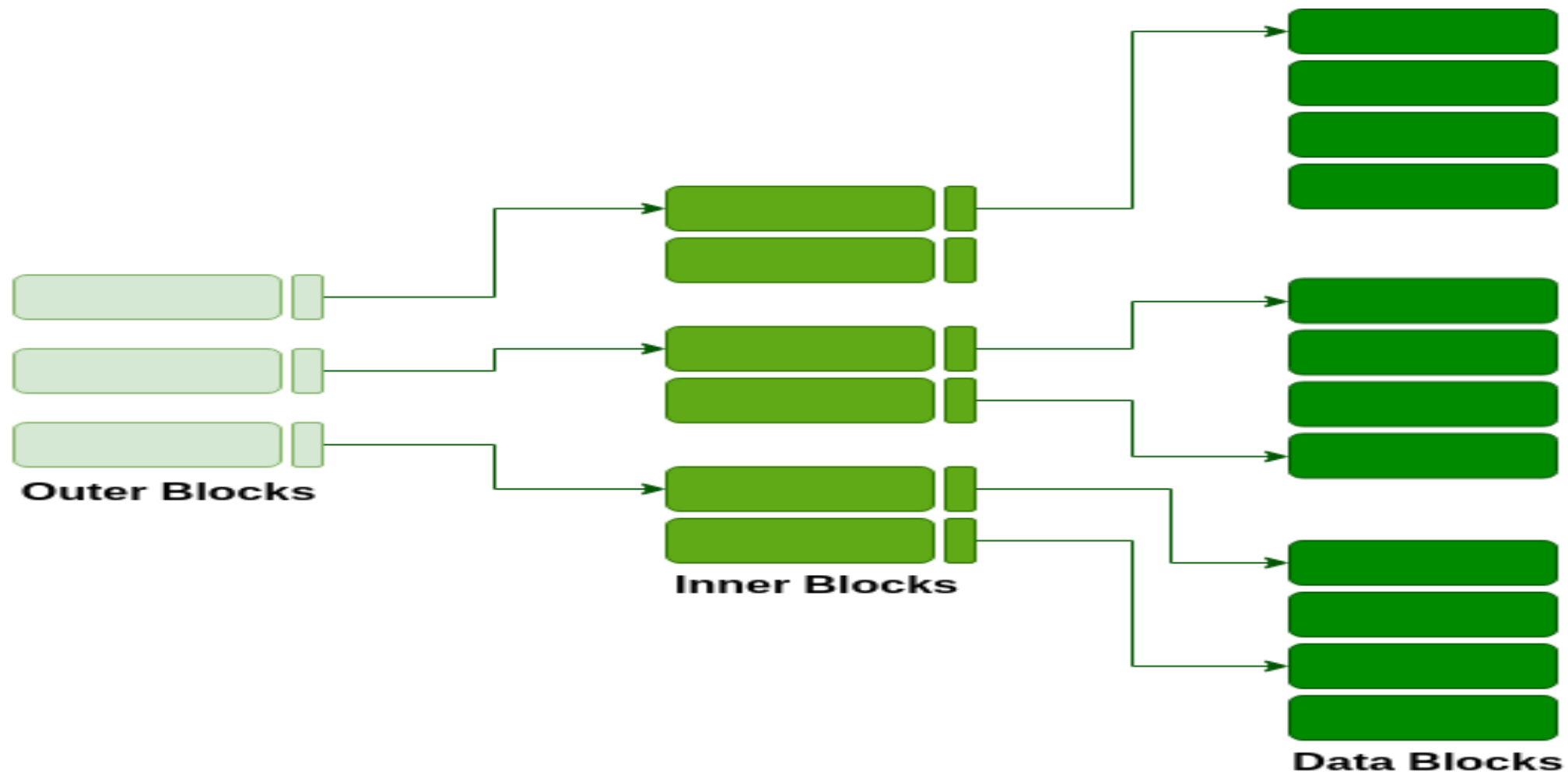
INDEX FILE							
SEMESTER	INDEX ADDRESS	Data Blocks in Memory					
1		→	100	Joseph	Alaiedon Township	20	200
2			101				
3							
4			110	Allen	Fraser Township	20	200
5			111				
			120	Chris	Clinton Township	21	200
			121				
			200	Patty	Troy	22	205
			201				
			210	Jack	Fraser Township	21	202
			211				
			300				



Non clustered index

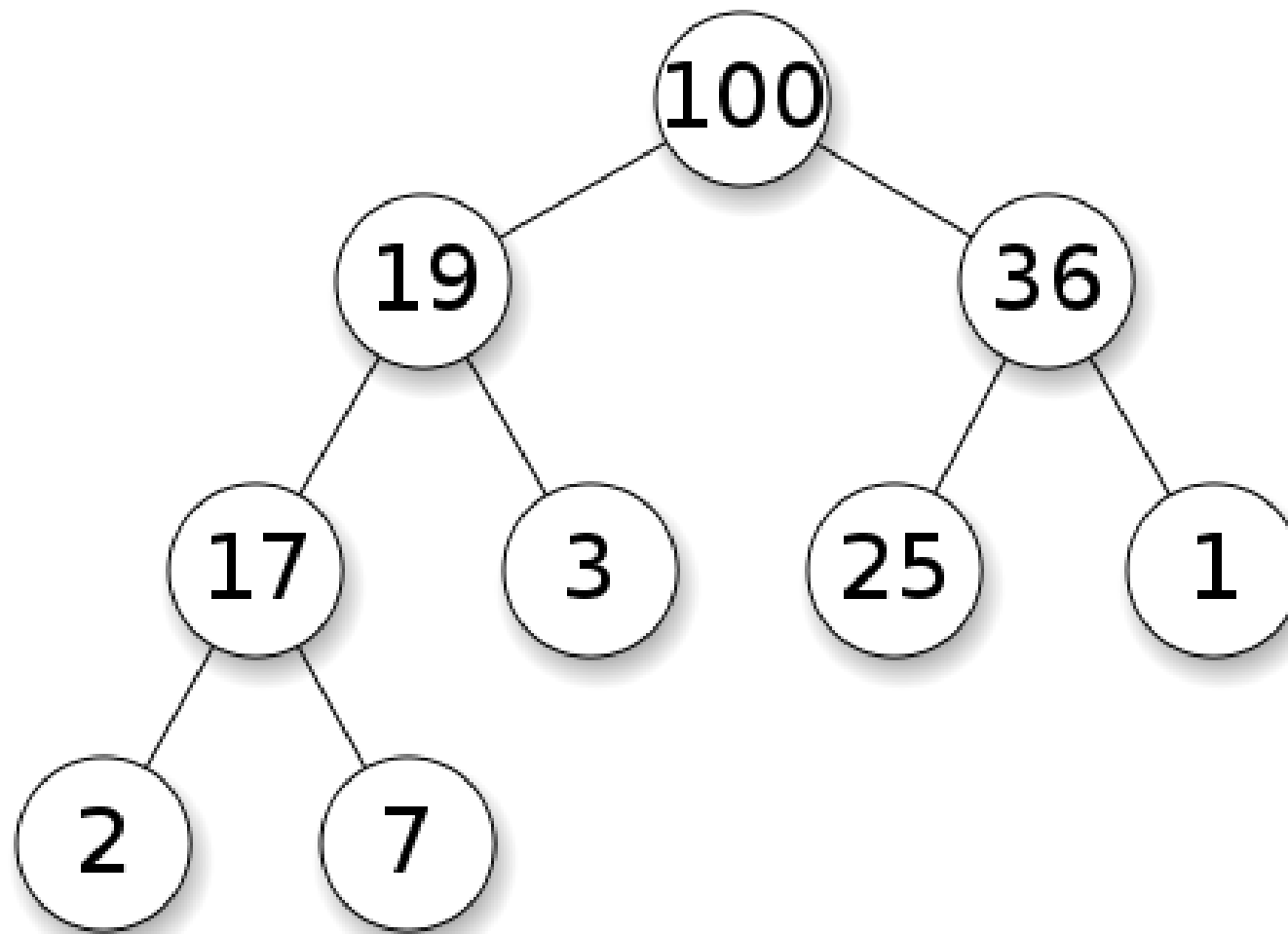
Indexacion multinivel

- Con el crecimiento del tamaño de la base de datos, los índices también crecen. Como el índice se almacena en la memoria principal, un índice de un solo nivel puede tener un tamaño demasiado grande para almacenarse con múltiples accesos al disco. La indexación multinivel segrega el bloque principal en varios bloques más pequeños para que el mismo pueda almacenarse en un solo bloque. Los bloques exteriores se dividen en bloques interiores que, a su vez, apuntan a los bloques de datos. Esto se puede almacenar fácilmente en la memoria principal con menos gastos generales.



Indice usando arboles binarios

- El índice B-tree es una estructura de datos ampliamente utilizada para la indexación, basada en árboles en DBMS. Es un formato multinivel de indexación basada en árboles en la técnica DBMS que ha equilibrado árboles de búsqueda binaria. Todos los nodos hoja del árbol B significan punteros de datos reales.
- Además, todos los nodos hoja están interconectados con una lista de enlaces, lo que permite que un árbol B admita acceso tanto aleatorio como secuencial.



En conclusión

- La indexación de bases de datos es esencial para garantizar un rendimiento óptimo y una experiencia de usuario ágil. Al aplicar índices adecuados y actualizar regularmente las estadísticas de la base de datos, puedes reducir el tiempo de respuesta de las consultas y mejorar la eficiencia del sistema de la base de datos en general. No subestimes la importancia de la indexación de bases de datos, ¡es un factor clave en el éxito y la eficiencia de tus aplicaciones y sistemas!