

Índices

Unidad 7.

Almacenamiento y Estructuras de Archivos

Introducción

El ejemplo de la biblioteca

- Se desea buscar un libro en base a un autor, un tema o un título en específico
- En lugar de tener tres edificios con los libros ordenados de cada manera en cada uno de los edificios se cuenta con un catálogo de tarjetas
- Este catálogo utiliza diferentes llaves y un identificador como campo de referencia

Definiciones

- Un índice es una tabla que contiene una lista de temas (llaves) y el lugar donde pueden ser encontrados (campos de referencia)

Propiedades

- Son herramientas muy útiles para el procesamiento de archivos
- Un índice permite tener un orden en un archivo sin tener la necesidad de reorganizarlo
- Permite que las operaciones de escritura y lectura sean menos costosas
- Los índices simples son representados utilizando estructuras simples de arreglos que contienen las llaves y los campos de referencia

Llave primaria

- El elemento más importante en el manejo de índices es la llave primaria
- Una llave primaria es aquella que identifica de manera única a un registro en el archivo
- Se debe elegir información (un campo) del registro tal que sea única para ese registro en particular
- En caso de no existir, es posible combinar campos para obtener la llave primaria

Ejemplo

- Se tiene una colección de archivos con información sobre discos de música y se desea mantener un orden de la colección a través de índices
- Cada registro del archivo tiene la siguiente información:
 - Número de identificador (ID)
 - Título
 - Compositor
 - Artista
 - Compañía

Eligiendo la llave primaria

- Una llave primaria que consiste en las iniciales de la compañía combinada con el ID es una buena combinación para crear una llave primaria
- Para evitar confusiones, es recomendable utilizar una forma canónica de la llave
- Las iniciales de la compañía en mayúsculas y el número de identificador

Información para crear el índice

Etiqueta	ID	Título	Compositor	Artista
LON	2312	Romeo y Julieta	Prokofiev	Maazel
RCA	2626	Cuarteto en C menor	Beethoven	Julliard
WAR	23699	Touchstone	Corea	Corea
ANG	3795	Sinfonía No. 9	Beethoven	Guilini
COL	38358	Nebraska	Springsteen	Springsteen
DG	18807	Sinfonía No. 9	Beethoven	Karajan
MER	75016	Coq d'Or Suite	Rimsky-Korsakov	Leindsdorf
COL	31809	Sinfonía No. 9	Dvorak	Bernstein
DG	139201	Concierto de violín	Beethoven	Ferras
FF	245	Good News	Sweet Money	Sweet Honey

Índice y registros en el archivo

Llave	Referencia
ANG3795	152
COL31809	338
COL38358	196
DG139201	382
DG18807	241
FF245	427
LON2312	17
MER75016	285
RCA2626	62
WAR23699	117

Dirección	Datos
17	LON 2312 Romeo y Julieta Prokofiev ...
62	RCA 2626 Cuarteto en C menor Beethoven ...
117	WAR 23699 Touchstone Corea ...
152	ANG 3795 Sinfonía No. 9 Beethoven ...
196	COL 38358 Nebraska Springsteen ...
241	DG 18807 Sinfonía No. 9 Beethoven ...
285	MER 75016 Coq d'Or Suite Rimsky-Korsakov ...
338	COL 31809 Sinfonía No. 9 Dvorak ...
382	DG 139201 Concierto de violín Beethoven ...
427	FF 245 Good News Sweet Honey ...

Construcción

- En la derecha esta el archivo con información acerca de la colección de registros
- Contiene información de longitud variable y sólo se muestran cuatro campos
- A la izquierda se muestra el índice, donde cada entrada de él contiene una llave correspondiente a cierto ID de la compañía
- Cada llave está asociada con un campo de referencia que proporciona la dirección del primer byte del registro correspondiente

Explicación

- ANG3795, corresponde al campo de referencia que contiene el número 152, lo que significa que el registro que contiene la información del archivo con el identificador ANG3795 puede ser encontrado comenzando en el byte 152 en el archivo de registros.

Estructura del índice

- La estructura del objeto índice es muy sencilla, es una lista de pares de campos, un campo para la llave de búsqueda y un campo para el offset
- Hay una entrada en el índice para cada registro en el archivo
- Un índice se implementa con arreglos de longitud fija que contienen las diferentes llaves y el offset
- Se tienen llaves únicas (no se permiten duplicados) o llaves no únicas (se permiten duplicados)

Propiedades

- Mientras un índice se encuentra ordenado de acuerdo a la llave, el archivo no lo está, por consecuencia, a pesar de que el identificador ANG3795 es la primer entrada en el índice, no necesariamente la primer entrada del archivo es la información correspondiente a ese identificador
- De hecho en ocasiones el archivo es creado secuencialmente, lo que significa que los registros quedan ordenados de acuerdo al orden en el que se van introduciendo

Propiedades

- La creación de un archivo de manera secuencial es una manera más sencilla de realizar la operación de almacenamiento
- Mantener en memoria el índice permite encontrar registros de manera más rápida que si se tuviera un archivo ordenado
- Una vez que el offset ha sido encontrado, una búsqueda simple es todo lo que se requiere para recuperar la información

Operaciones con índices

Operaciones

- Las operaciones utilizadas para el manejo de información en archivos con índices son:
 - Creación de los archivos vacíos de índice y de registros
 - Carga del índice en la memoria antes de utilizarlo
 - Reescribir el archivo de índices de la memoria después de utilizarlo
 - Añadir registros al archivo de datos
 - Eliminar registros del archivo de datos
 - Actualizar registros en el archivo de datos
 - Actualizar el índice para reflejar los cambios en el archivo de datos

Creación de archivos

- Dos archivos deben ser creados:
 - Un archivo de datos que contenga los registros
 - Un archivo de índices que contenga las llaves de búsqueda primaria
- Ambos archivos, son creados inicialmente como archivos vacíos

Cargando el índice en memoria

- Se necesita cargar el índice en una estructura que permita su fácil manejo, por ejemplo puede manejarse como una lista o un arreglo que contenga los siguientes datos miembro:
 - Llave primaria
 - *Offset*

Re escribir el índice

- El índice en memoria necesita ser escrito nuevamente en el archivo de índices después de ser utilizado o creado
- Es importante considerar lo que sucede si esta escritura no es correcta o es incompleta:
 - El índice no estará actualizado por lo que su información no corresponderá a lo contenido en el archivo

Uso de banderas

- Un mecanismo que permite que el programa sepa cuando el índice se no se encuentra actualizado es utilizando una bandera
- Esta bandera puede ser creada tan pronto el archivo de índices es leído y eliminada cuando el índice es re escrito en el archivo
- Esta bandera puede incluirse en el archivo de índices y ser el último elemento en actualizar

Volver a crear el índice

- Si un programa al momento de cargar el índice, detecta que no está actualizado, se debe tener acceso a un procedimiento que reconstruya el índice a partir de la información contenida en el archivo
- Esto debe realizarse de manera automática antes de cualquier intento de usar el índice

Agregando registros al archivo

- Añadir un nuevo registro al archivo de datos requiere que también se añada una entrada al índice
- Dado que el índice es mantenido ordenado según la llave, la inserción de una nueva entrada probablemente requiera alguna reorganización del índice
- La ventaja que se tiene es que la información del índice está en memoria, por lo que se puede ordenar sin tener la necesidad de acceder al archivo

Eliminando registros

- Cuando se elimina algún registro del archivo de datos, también debe eliminarse la correspondiente entrada en el índice
- Dado que el índice se encuentra en memoria durante la ejecución del programa, eliminar la entrada del índice y reordenar el resto de las llaves no será una operación muy costosa
- Alternativamente, simplemente se puede marcar la entrada del archivo de índice como eliminada

Actualizando registros

- Al momento de actualizar un campo de un registro se tiene dos posibilidades:
 - La actualización cambia el valor del campo que involucra o contiene la llave primaria
 - Este tipo de actualización puede traer como consecuencia la organización tanto del archivo de índices como del archivo de datos
 - La actualización no afecta al campo que involucra o contiene la llave primaria
 - Este tipo de actualización no requiere una reorganización del archivo de índices pero involucra una reorganización del archivo de datos

Modificación del tamaño

- Si el tamaño del registro no ha sido modificado por la actualización, este puede ser escrito directamente en el espacio que ocupaba anteriormente
- Pero si el tamaño del registro se incrementó durante la actualización, se debe encontrar un nuevo espacio para almacenarlo

Optimización

- Una manera de optimizar las operaciones es evitar escribir el registro que se encuentra en memoria en el archivo de índices cuando éste no ha sido alterado
- Colocar la bandera en falso cuando el registro es inicialmente cargado en memoria y se le asignarle verdadero cuando el índice haya sido modificado
- Se verifica el valor de esta bandera y se escribe el índice en el archivo solo cuando es necesario

Indexado con varias llaves
(uso de llaves secundarias)

Llaves secundarias

- En ocasiones es complicado realizar búsquedas utilizando la llave primaria ya que puede ser difícil de recordar
- También es deseable encontrar información combinando valores de los campos
- Para esto se utilizan las llaves secundarias
- La estructura de una llave secundaria incluye el valor de un cierto campo y el valor de la llave primaria del registro al que corresponde ese campo

Ejemplo de llaves secundarias

Llave secundaria	Llave primaria
BEETHOVEN	ANG3795
BEETHOVEN	DG139201
BEETHOVEN	DG18807
BEETHOVEN	RCA2626
COREA	WAR23699
DVORAK	COL31809
PROKOFIEV	LON2312
RIMSKY	MER75016
SPRINGSTEEN	COL38358
SWEET HONEY	FF245

Uso de llaves secundarias

- Siguiendo con el ejemplo de los archivos musicales:
 - Difícilmente se busca un disco con la referencia DG18807, lo que se desea es buscar basándose en el autor, título, canción, etc.
 - Dado este tipo de datos, se busca en el catálogo para encontrar la llave primaria

Manejo de llaves secundarias

- Después de consultar el índice del compositor para encontrar el identificador, se debe consultar otro índice adicional, el de la llave primaria para encontrar el offset que se necesita para acceder a la información deseada

Operaciones con el manejo de llaves secundarias

Agregando un registros

- Añadir un registro al archivo significa añadir una entrada al índice secundario
- De manera similar con los índices primarios, el costo de realizar estas operaciones decrece bastante si los índices secundarios son leídos en memoria y modificados ahí
- Una diferencia importante entre los índices primarios y secundarios, es que los índices secundarios pueden contener llaves duplicadas

Eliminando un registro

- Eliminar un registro usualmente implica remover todas las referencias hacia ese registro en el sistema de archivos
- Eliminar un registro del archivo de datos significa remover no solamente la entrada correspondiente en el índice primario sino también todas las entradas en los índices secundarios que referencian a esta entrada en el índice primario

Actualizando de registros

- El índice primario sirve como un tipo de buffer de protección, aislando los índices secundarios de los cambios en el archivo de datos
- En caso que el archivo de índices secundario contuviera referencias directas hacia el offset en el archivo de datos, entonces las actualizaciones al archivo de datos que resulten en modificar un registro en el archivo también requerirán actualizar los índices secundarios

Actualizando registros

- Se tienen tres situaciones posibles:
 - Actualizar cambios en la llave secundaria: si la llave secundaria se altera, se debe organizar el archivo de índices secundarios de tal manera que permanezca ordenado
 - Actualizar cambios en la llave primaria: este tipo de cambios tienen un gran impacto en el índice de llaves primarias pero a menudo requiere que se actualicen solo los campos de referencia afectados en los índices secundarios. Esto involucra la búsqueda de índices secundarios y la reescritura de los campos alterados
 - Actualización confinada a otros campos: Todas las actualizaciones que no afecten a los campos de las llaves primaria o secundaria no afectan al índice secundario

Combinando llaves secundarias

Combinando llaves secundarias

- Una de las aplicaciones más importantes de llaves secundarias involucra el uso de dos o más de ellas en combinación para obtener subconjuntos especiales de registros del archivo de datos

Ejemplo

- Si se utiliza otro índice de llaves en el ejemplo de los archivos de música, es posible:
 - Encontrar la canción con la etiqueta COL38358 (llave primaria de acceso)
 - Encontrar todas las canciones de Beethoven (llave secundaria de autor)
 - Encontrar todas las canciones llamadas (“Concierto de Violín”) (llave secundaria de título)

Combinando llaves secundarias

- Lo que es más interesante, sin embargo, es que también se puede responder a las peticiones que combinan los resultados de una búsqueda con autor o con título
- Encontrar todo lo relacionado con la 9na sinfonía de Beethoven, sin el uso de índices secundarios, este tipo de petición necesitaría realizar una búsqueda secuencial a través del archivo entero. Dado un archivo que contiene cientos o miles de registros, esto es un proceso muy costoso

Combinando listas

- Con ayuda de un índice secundario, responder a esta petición es sencillo.
- Esta operación puede ser vista como una operación booleana AND, especificando la intersección de dos subconjuntos del archivo de datos

Ejemplo:

- Se comienza la respuesta a esta petición buscando en el índice de compositor por la lista de ID que estén identificados con Beethoven, esto regresaría:
 - ANG3795
 - DG139201
 - DG18807
 - RCA2626

Ejemplo, cont...

- Después comienza una búsqueda en el índice que contiene los títulos asociados con los registros que tienen SINFONIA No 9 como llave:
 - ANG3795
 - COL31809
 - DG18807

Ejemplo, cont...

- Ahora se realiza una operación booleana AND, que combine las listas de tal manera que solo los miembros que aparecen en ambas listas sean colocadas en la lista de salida

Compositores	Título	Coincidencia
ANG3795	ANG3795	ANG3795
DG139201	COL31809	
DG18807	DG18807	DG18807
RCA2626		

- Finalmente, una vez que se tiene la lista de llaves primarias que ocurren en ambas listas, se puede acceder al índice de llaves primarias para buscar las direcciones de los registros en el archivo de datos y obtener los siguientes resultados

ANG	3795	Sinfonía No. 9	Beethoven	Guilini
DG	18807	Sinfonía No. 9	Beethoven	Barajan

Combinación de llaves

- Este es el tipo de operaciones que hace los sistemas de archivos indexados útiles en una manera que exceden las capacidades de los sistemas manuales
- Trabajar mediante índices secundarios proporciona una múltiple vista de los registros ya que se puede buscar información ya sea por cualquier tipo que esté en un campo

Manejo de llaves secundarias

Índice secundario

- Índices secundarios basados en autor y en título creados a partir del ejemplo que se ha venido manejando son:

Secundaria	Primaria	Secundaria	Primaria
BEETHOVEN	ANG3795	COQ D'OR SU	MER75016
BEETHOVEN	DG139201	GOOD NEWS	FF245
BEETHOVEN	DG18807	NEBRASKA	COL38358
BEETHOVEN	RCA2626	QUARTET IN	RCA2626
COREA	WAR23699	ROMEO AND J	LON2312
DVORAK	COL31809	SYMPHONY NO	ANG3795

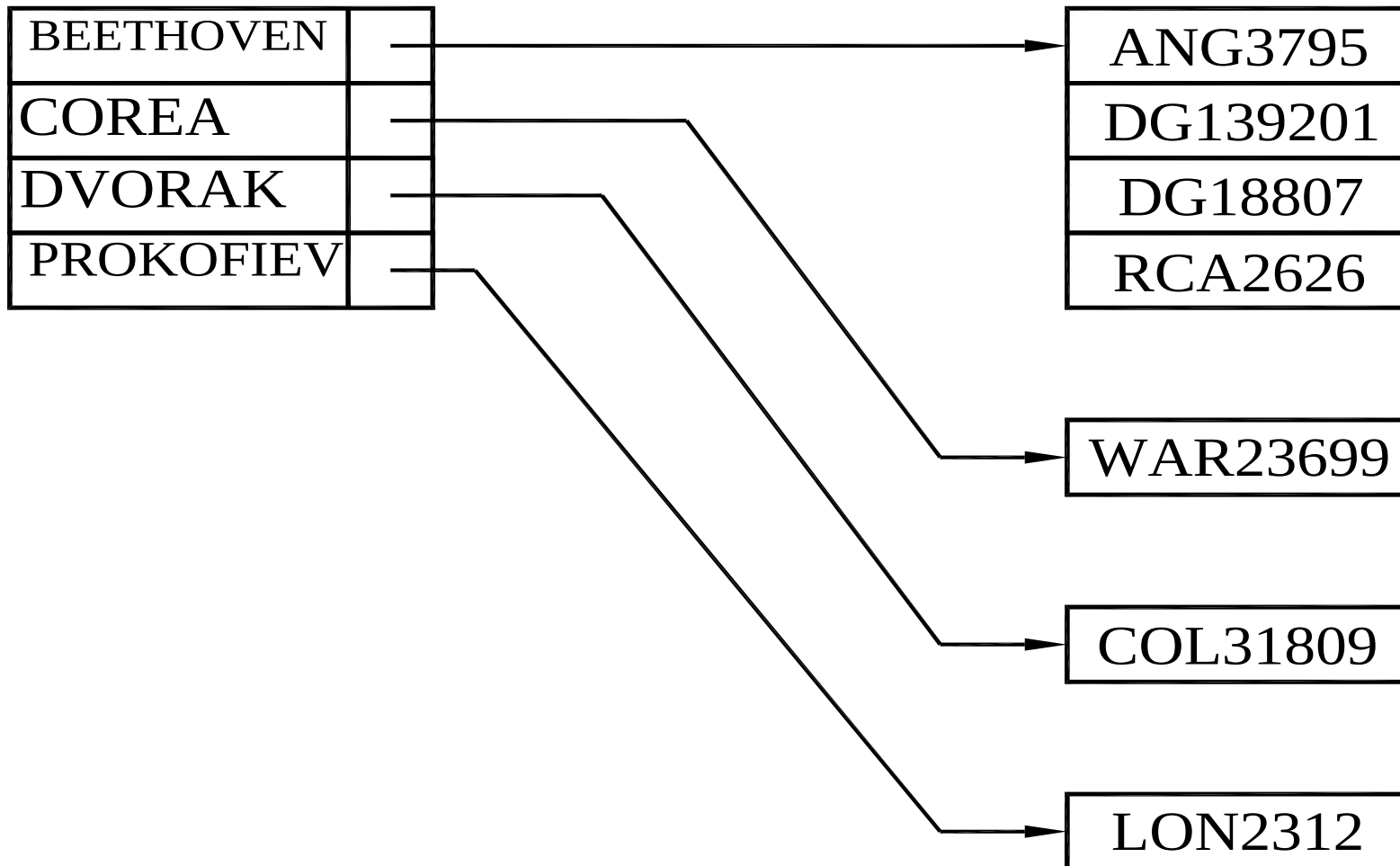
Listas invertidas

- Es poco eficiente escribir en el archivo todas las llaves secundarias y la llave primaria con la que están relacionadas
- Una posibilidad es manejar un arreglo de tamaño fijo con las llaves primarias que correspondan a una llave secundaria

BEETHOVEN	ANG3795	DG139201	DG18807	RCA2626
-----------	---------	----------	---------	---------

- Este manejo se conoce como listas invertidas, ya que una llave secundaria hace referencia a una primaria

Listas invertidas



Listas ligadas

- Otra opción es separar al archivo de índices secundario en dos archivos
- Uno contiene solo las llaves secundarias y una referencia a la primer llave primaria en el otro archivo
- El segundo archivo contiene una estructura similar a una lista ligada

Ejemplo de listas ligadas

Archivos de llave secundaria	
BEETHOVEN	3
COREA	2
DVORAK	7
PROKOFIEV	10
KRIMSKY	6
SPRINGSTEEN	4
SWEET HONEY	9

0	LON2312	-1
1	RCA2626	-1
2	WAR23699	-1
3	ANG3795	8
4	COL38358	-1
5	DG18807	1
6	MER75016	-1
7	COL31809	-1
8	DG139201	5
9	FF245	-1
10	ANG36193	0

Buscando información

- Se desean encontrar los registros que contengan la llave de *Beethoven* en compositor
- Se comienza buscando la llave secundaria para compositores, el registro que se encuentra apunta al RRN 3 en la lista de ID, dado que es un archivo de longitud fija, es fácil saltar al RRN 3 y leer en la lista de etiquetas (ANG3795)
- Asociado con este ID se encuentra una liga al RRN 8, se lee ese registro, se añade a la lista y se continúan siguiendo las ligas y recolectando etiquetas hasta encontrar una referencia que indique nulo (-1)

Ventajas

- Asociar el archivo de índices secundarios con un nuevo archivo que contenga listas ligadas de referencias proporciona algunas ventajas sobre otras estructuras:
 - La única vez que se necesita reorganizar el archivo de índices secundarios es cuando se agrega una nueva llave secundaria
 - Añadir o eliminar valores de llaves secundarias que ya se encuentran involucra un cambio únicamente en el archivo de etiquetas
 - El archivo de listas de etiquetas está ordenado, lo que significa que nunca necesita ser ordenado

Ataduras

Ataduras

- Una cuestión muy importante que surge durante el diseño de sistemas de archivos que utilizan índices es saber en que punto se encuentra la llave atada a la dirección física del registro que tiene asociado
- En algunos sistemas de archivos, la atadura de las llaves primarias a una dirección tiene lugar en el momento en que los archivos son creados, las llaves secundarias por otro lado, son atadas a una dirección en el momento en que son utilizadas

Ataduras

- Atar al momento de construir el archivo resulta en un acceso más rápido, una vez que se ha encontrado el índice del registro correcto, se tiene el *offset* del dato que se está buscando
- La desventaja de atar directamente al archivo es que las reorganizaciones de los archivos de datos resultarían en operaciones muy costosas

Índices muy grandes para
almacenarse en memoria

Introducción

- Los métodos que se estudiaron asumen que el índice es lo suficientemente pequeño para ser cargado en memoria completamente
- Si el índice es muy grande para este enfoque, entonces el acceso al índice y su mantenimiento debe ser realizado en un medio de almacenamiento secundario

Introducción

- Con índices simples como los que se han estado trabajando, acceder a ellos en un disco tiene las siguientes desventajas:
- La búsqueda binaria del índice requiere demasiadas búsquedas en lugar de tener lugar a velocidades de lectura de la memoria. La búsqueda binaria de un índice en un medio de almacenamiento secundario no es lo significativamente más rápida que la realizada con una búsqueda binaria en un archivo ordenado
- El re-acomodamiento de los índices debido a la adición o eliminación requiere ordenamiento de los registros en el medio de almacenamiento secundario, esto es literalmente millones de veces más costoso que realizar estas operaciones en memoria

Opciones

- Cuando se tenga un índice muy grande para ser manejado en memoria, lo que se debe considerar es lo siguiente:
 - Una organización basada en “*hashing*” si la velocidad de acceso es prioritaria
 - Un índice con una estructura de árbol multinivel, tal como un Árbol-B, si se necesita la flexibilidad de los accesos tanto por llave como secuencial