Algoritmos e Estruturas de Dados II

Prof. Ricardo J. G. B. Campello

Indexação de Arquivos III:

Busca com Múltiplas Chaves & Listas Invertidas

Adaptado e Estendido dos Originais de:

Leandro C. Cintra
Maria Cristina F. de Oliveira

Arquivo de Índice (Revisão)

- Exemplo Prático (Arquivo de Músicas)
  - Registros de tamanho variável com:
    - **ID Number:** Número de identificação
    - **Title**: Título
    - **Composer**: Compositor(es)
    - **Artist**: Artista(s)
    - **Label**: Rótulo (código da gravadora)
  - Chave primária:
    - Combinação de **Label** e **ID Number**

Arquivo de Índice (Revisão)

Record address	Label	ID number	Title	Composer(s)	Artist(s)
17	LON	2312	Romeo and Juliet	Prokofiev	Maazel
62	RCA	2626	Quartet in C Sharp Minor	Beethoven	Julliard
117	WAR	23699	Touchstone	Corea	Corea
152	ANG	3795	Symphony No. 9	Beethoven	Giulini
196	COL	38358	Nebraska	Springsteen	Springsteen
241	DG	18807	Symphony No. 9	Beethoven	Karajan
285	MER	75016	Coq d'Or Suite	Rimsky-Korsakov	Leinsdorf
338	COL	31809	Symphony No. 9	Dvorak	Bernstein
382	DG	139201	Violin Concerto	Beethoven	Ferras
427	FF	245	Good News	Sweet Honey in the Rock	Sweet Honey in the Rock

Figure 7.2 Contents of sample recording file.

Arquivo de Índice (Revisão)

Index

Key	*Reference field*
ANG3795	152
COL31809	338
COL38358	196
DG139201	382
DG18807	241
FF245	427
LON2312	17
MER75016	285
RCA2626	62
WAR23699	117

Recording file

Address of record	*Actual data record*
17	LON | 2312 | Romeo and Juliet | Prokofiev | ...
62	RCA | 2626| Quartet in C Sharp Minor | Beethoven | ...
117	WAR | 23699 | Touchstone | Corea | ...
152	ANG | 3795 | Symphony No. 9 | Beethoven | ...
196	COL | 38358 | Nebraska | Springsteen | ...
241	DG | 18807 | Symphony No. 9 | Beethoven | ...
285	MER | 75016 | Coq d'Or Suite | Rimsky-Korsakov | ...
338	COL | 31809 | Symphony No. 9 | Dvorak | ...
382	DG | 139201 | Violin Concerto | Beethoven | ...
427	FF | 245 | Good News |Sweet Honey in the Rock | ...

Figure 7.3 Index of the sample recording file.

Indexação Secundária (Revisão)

- Exemplo Prático (Arquivo de Músicas):

Composer index

Secondary key	*Primary key*
BEETHOVEN	ANG3795
BEETHOVEN	DG139201
BEETHOVEN	DG18807
BEETHOVEN	RCA2626
COREA	WAR23699
DVORAK	COL31809
PROKOFIEV	LON2312
RIMSKY-KORSAKOV	MER75016
SPRINGSTEEN	COL38358
SWEET HONEY IN THE R	FF245

Title index

Secondary key	*Primary key*
COQ D'OR SUITE	MER75016
GOOD NEWS	FF245
NEBRASKA	COL38358
QUARTET IN C SHARP M	RCA2626
ROMEO AND JULIET	LON2312
SYMPHONY NO. 9	ANG3795
SYMPHONY NO. 9	COL31809
SYMPHONY NO. 9	DG18807
TOUCHSTONE	WAR23699
VIOLIN CONCERTO	DG139201

5

Busca com Múltiplas Chaves

- Uma das aplicações mais importantes das chaves secundárias é localizar conjuntos de registros do arquivo de dados usando uma ou mais chaves
- Pode-se fazer uma busca (consulta) em vários índices e combinar (AND, OR, NOT) os resultados individuais
- Exemplo: Encontre todos os registros tal que
  - salário > R$3000 **OR** tempo_serviço > 10 anos

6

Busca com Múltiplas Chaves

- Exemplo: Encontre todos os registros tal que
  - composer = "BEETHOVEN" **AND** title = "SYMPHONY NO. 9"

ANG3795
DG139201
DG18807
RCA2626

AND

ANG3795
COL31809
DG18807

  - Co-processamento seqüencial dos arquivos (Cap. Seguinte...) !
    - beneficia-se da ordenação local pelas chaves primárias !

resultado

```
ANG|3795|Symphony No. 9|Beethoven|Giulini
DG|18807|Symphony No. 9|Beethoven|Karajan
```

7

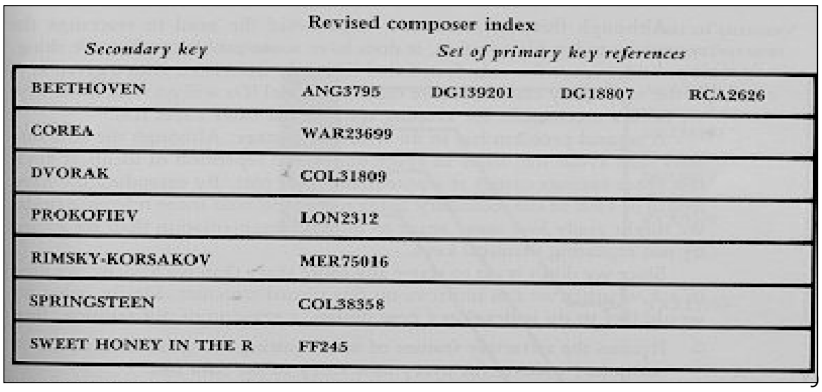
Índices Secundários Melhorados

- Dois problemas nas estruturas de índices vistas até agora:
  - repetição de chaves secundárias
    - arquivos de índices secundários maiores que o necessário
  - necessidade de rearranjar os índices mesmo quando um novo registro que tenha um valor de chave secundária já existente no arquivo seja inserido
    - P. ex. se uma nova gravação da sinfonia no. 9 de Beethoven for inserida no nosso arquivo de música

8

Índices Secundários Melhorados

- **Solução 1:** Associar um conjunto de chaves primárias (tamanho fixo) a cada chave secundária



Revised composer index

Secondary key	*Set of primary key references*			
BEETHOVEN	ANG3795	DG139201	DG18807	RCA2626
COREA	WAR23699			
DVORAK	COL31809			
PROKOFIEV	LON2312			
RIMSKY-KORSAKOV	MER75016			
SPRINGSTEEN	COL38358			
SWEET HONEY IN THE R	FF245			

Índices Secundários Melhorados

- **Solução 1:**
  - Elimina entradas com chaves secundárias duplicadas; e
  - Não é necessário reordenar o índice a cada inserção de registro com chave secundária existente,
  - Porém ...
  - É limitado a um número fixo de repetições da chave; e
  - Quanto maior esse número, maior a fragmentação interna do arquivo de índice !
    - talvez não compense a eliminação das chaves duplicadas

10

Índices Secundários Melhorados

- **Solução 2:** Listas invertidas
  - Associar cada chave secundária a uma lista encadeada (denominada invertida) dessas chaves primárias
  - Substitui-se a referência à chave primária nos registros do arquivo de índice secundário por uma referência ao RRN do primeiro registro com essa chave na lista invertida
  - Listas invertidas são mantidas em um arquivo seqüencial separado, organizado segundo a entrada dos registros
    - *entry sequenced file*

11

Listas Invertidas (visão conceitual)

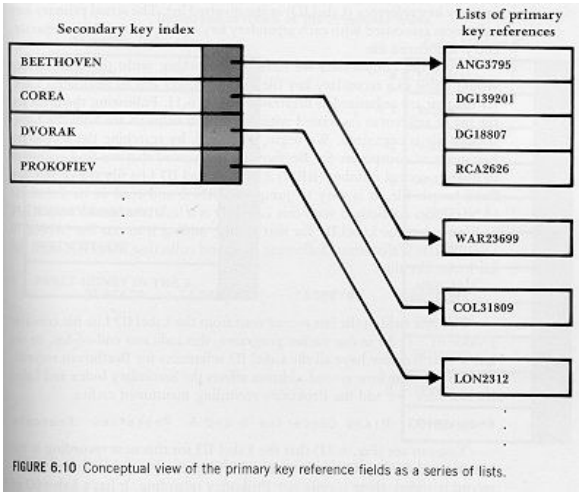
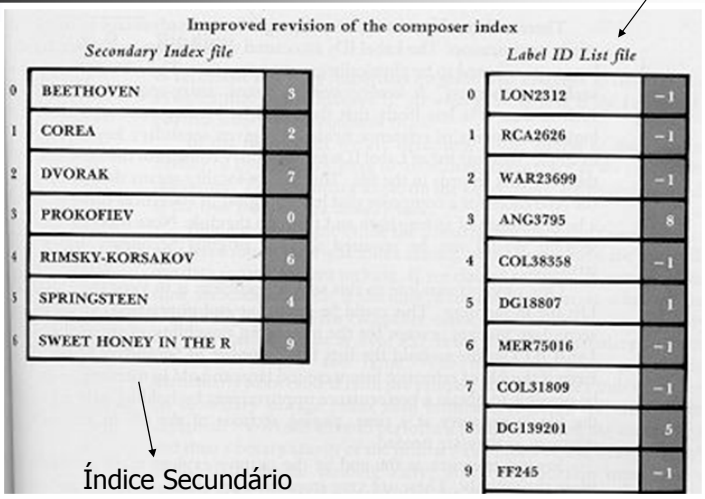


FIGURE 6.10 Conceptual view of the primary key reference fields as a series of lists.

12

Lista Invertida



13

Lista Invertida

- **Vantagens:**
  - Índice secundário só precisa ser alterado quando:
    - inserido um registro com chave secundária ainda não existente
    - removido registro cabeça de lista invertida (talvez o único...)
    - alterada uma chave (primária ou secundária) já existente
  - Quando necessário, rearranjar o índice é mais simples:
    - contém menos registros; e
    - não existe duplicidade de chaves secundárias
  - Pode ser feito com as técnicas de manutenção de arquivos de índice ordenados discutidas anteriormente

14

Lista Invertida

- **Vantagens:**
  - Em muitos casos, as operações de remoção, inserção ou alteração de registros no arquivo de dados implicam apenas em alterar o arquivo de listas invertidas
  - Arquivo de listas invertidas nunca precisa ser ordenado, pois é *entry sequenced*
    - única preocupação é encadear cada lista de forma ordenada segundo a chave primária
  - Logo, é trivial reutilizar o espaço liberado por registros eliminados do arquivo de listas invertidas

15

Lista Invertida

- **Problema**
  - Registros associados a cada valor de chave secundária, encadeados em uma mesma lista invertida, não estão adjacentes no arquivo lógico e no disco:
    - podem ser necessários vários *seeks* para recuperar uma lista
  - O ideal seria manter o índice e as listas em RAM
  - Quando não é possível, é recomendável pensar em estruturas de indexação mais sofisticadas

16

Binding

- Nos índices primários a associação (*binding*) entre a chave primária e a localização do registro a que ela se refere ocorre no momento em que o registro é criado e introduzido no arquivo de índices
  - Fornece acesso direto rápido a um registro, dada a sua chave
- Já as chaves secundárias são associadas às localizações apenas no momento em que são de fato usadas (*late binding*)
  - Dada a chave secundária busca-se pela(s) primária(s) e, só então, associa-se a primeira ao endereço de um ou mais registros
  - Isso implica um acesso mais lento
  - Mas também implica manutenção mais eficiente e confiável (localizada)
- Em arquivos estáticos (e.g. CD-ROM), no entanto, pode ser mais interessante associar diretamente cada índice secundário à(s) localização(ões) dos registros correspondentes (*early binding*)
  - Não existe manutenção...

17

Exercícios

- Insira vários novos registros no arquivo de músicas utilizado como exemplo em aula e mostre, a cada inserção, como fica o arquivo de índice secundário com chave "compositor" e o arquivo correspondente de listas invertidas
  - Insira alguns registros com chaves secundárias ainda não existentes e outros com chaves já existentes

18

Exercícios

- Repita o exercício anterior, mas desta vez atualizando alguns registros já existentes, ao invés de inserir novos
  - Em alguns casos mude a chave primária de um ou mais registros, em outros a secundária, e por fim mude apenas alguns campos não indexados
- Faça a remoção de alguns registros do arquivo de dados resultante dos exercícios anteriores e repita esses exercícios assumindo que as inserções de novos registros deverão ser feitas tal que as entradas correspondentes no índice secundário e nas listas invertidas reutilizem os espaços liberados pelas remoções

19

Outros Exercícios

- Capítulo 7 (Folk & Zoellick, 1987)
- Lista de Exercícios (CoTeia)
  - **Nota.** A lista faz referências à 2ª edição do livro de Folk & Zoellic. Nesse caso, o capítulo de indexação é o Capítulo 6
    - FOLK, M. & ZOELLICK, B., *File Structures*, 2nd Edition, Addison-Wesley, 1992.

20

Bibliografia

- **M. J. Folk and B. Zoellick, *File Structures: A Conceptual Toolkit*, Addison Wesley, 1987.**

21