GBC053 - Gerenciamento de Bancos de Dados

Aula 9
Indexação
Listas Invertidas

Humberto Razente
humberto.razente@ufu.br

Listas invertidas

- A estrutura de índices secundários vista até agora tem dois problemas:
  - a cada inserção, o índice tem que ser reordenado, mesmo que seja uma chave já inserida no índice
  - se há chaves secundárias duplicadas, a chave é repetida para cada entrada
    - desperdício de espaço
    - e quanto maior um índice, menor a probabilidade dele caber em memória principal

Listas invertidas: tentativa 1

- Primeira tentativa para resolver o problema:
  - associar um vetor de referências para cada chave secundária
  - por exemplo, usando uma estrutura de registro que permita associar uma quantidade de identificadores com uma única chave secundária
    - no exemplo no próximo slide, cada chave secundária pode conter até 4 referências

Listas invertidas: tentativa 1

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			

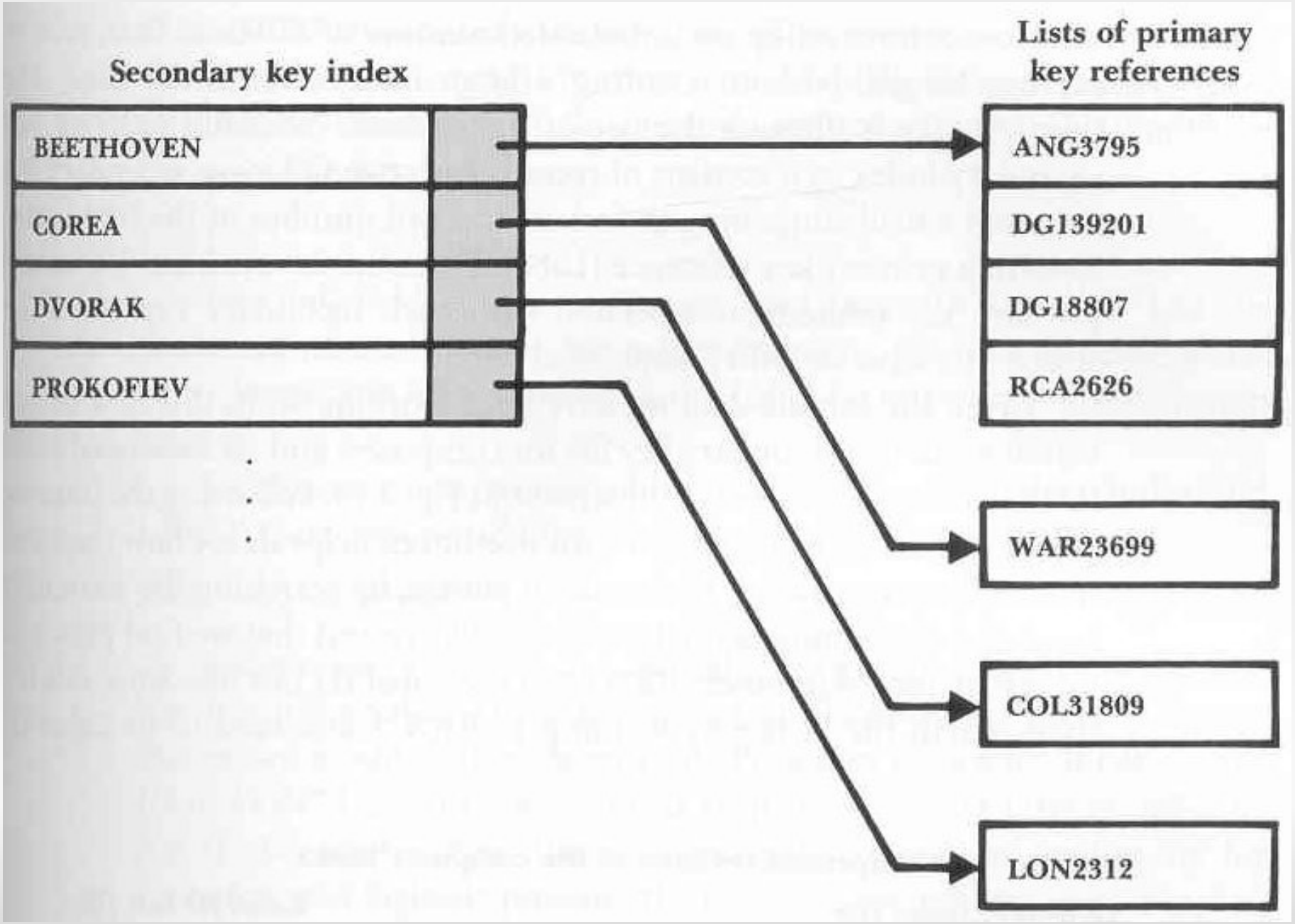
Listas invertidas: tentativa 1

- Problemas:
  - vetor de tamanho fixo:
    - limitação
    - desperdício de espaço (fragmentação interna)

Listas invertidas: solução melhorada

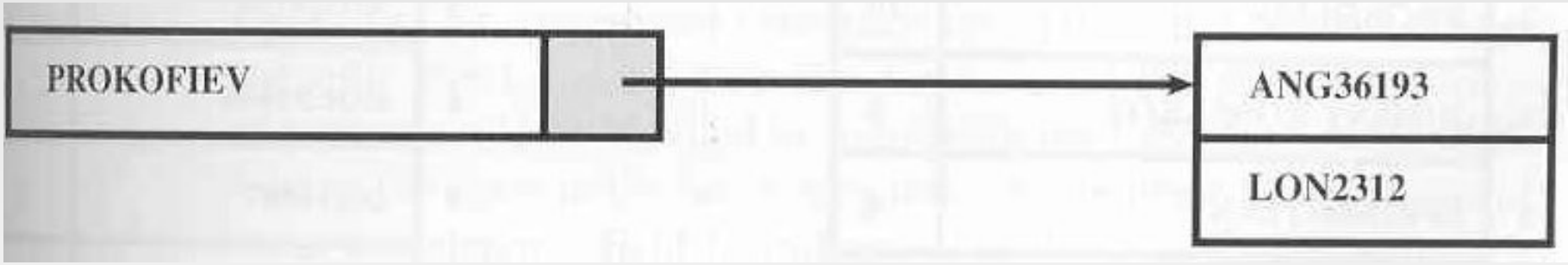
- Lidar de fato com uma lista de referências a chaves **primárias**
- O índice secundário armazena um número de chaves para cada chave secundária
  - como **listas** de chaves

Listas invertidas: solução melhorada



Listas invertidas: solução melhorada

- E a inclusão de um novo registro com chave ”Prokofiev” apenas insere a chave na lista:



Listas invertidas: solução melhorada

- Lista invertida:
  - definição: arquivo no qual uma chave secundária leva a um conjunto de uma ou mais chaves primárias

Listas invertidas: solução melhorada

- Desafio: como criar um número de listas diferentes, cada uma com tamanho variável, sem ter que criar um grande número de pequenos arquivos?
  - idéia simples: uso de listas ligadas
  - uso de um índice secundário com um atributo contendo o número relativo da primeira ocorrência da chave no arquivo da lista invertida

Listas invertidas: solução melhorada

Improved revision of the composer index

Secondary Index file

0	BEETHOVEN	3
1	COREA	2
2	DVORAK	7
3	PROKOFIEV	10
4	RIMSKY-KORSAKOV	6
5	SPRINGSTEEN	4
6	SWEET HONEY IN THE R	9

Label ID List file

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

Figure 7.13 Secondary key index referencing linked lists of primary key references.

Listas invertidas: solução melhorada

- Vantagens:
  - somente é necessário rearranjar o arquivo de índices secundários quando
    - um valor de chave que ainda não estava no índice é adicionado
    - um valor de chave que já estava é alterado
    - na remoção, pode-se apenas atribuir -1 no índice secundário
  - a tarefa de rearranjar o arquivo de índices secundários é mais rápida uma vez que contém menos elementos

Listas invertidas: solução melhorada

- Vantagens:
  - o arquivo de listas invertidas não precisa ser ordenado
  - é possível implementar uma politica de reuso de espaços disponíveis no o arquivo de listas invertidas, como visto anteriormente

Listas invertidas: solução melhorada

- Desvantagens:
  - As chaves primárias relacionadas com alguma chave secundária não estão mais garantidas de serem agrupadas fisicamente juntas
  - Se há uma longa lista de referências a chaves primárias, pode haver uma grande quantidade de buscas (***seeks***) no disco.
    - uma maneira de evitar isso é manter o arquivo que contém a lista invertida em memória
    - pode não ser factível, se há vários índices secundários

Leitura complementar

- Capítulo ”**Indexing**” do livro
  - Folk et al. ”File Structures: An Object-Oriented Approach with C++”, Editora Pearson, 3ª edição, 1998