SCC 503 - ALGORITMOS E ESTRUTURAS DE DADOS II

Arquivos - Índices

Prof.: Leonardo Tórtoro Pereira
leonardop@usp.br

*Material baseado em aulas dos professores: Elaine Parros Machado de Souza, Gustavo Batista, Robson Cordeiro, Moacir Ponti Jr., Maria Cristina Oliveira e Cristina Ciferri.

Índice

➔ Mecanismo para localizar informação via chave por meio de mapeamento

- Chave → localização da informação
- ex: índice de um livro, catálogo de biblioteca...
- Agiliza o trabalho de busca

Índice

➔ No caso de arquivos:

- Permite localizar registros rapidamente
- Evita ter que reorganizar o arquivo de dados conforme este for modificado

➔ Não é necessário manter arquivo ordenado…

Índice

- Arquivo de Índice (*index file*):
  - Impõe "ordem" a um arquivo de dados sem precisar rearranjar o arquivo em si
  - Permite acesso a registros via chave sem precisar varrer o arquivo de dados

Índice

- Arquivo de Índice (index file):
  - Permite várias visões diferentes de um mesmo arquivo de dados
    - Acesso por índices diferentes
  - Permite acesso rápido a arquivos com registros de tamanho variável

Índice Primário

➔ Exemplo Prático (Arquivo de CDs 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

Índice Primário

ANG3795	167
COL31809	353
COL38358	211
DG139201	396
DG18807	256
FF245	442
LON2312	32
MER75016	300
RCA2626	77
WAR23699	132

Arquivo de Índice

Arquivo auxiliar em disco

32	LON | 2312 | Romeo and Juliet | Prokofiev ...
77	RCA | 2626 | Quartet in C Sharp Minor ...
132	WAR | 23699 | Touchstone | Corea ...
167	ANG | 3795 | Symphony No. 9 | Beethoven ...
211	COL | 38358 | Nebraska | Springsteen ...
256	DG | 18807 | Symphony No. 9 | Beethoven ...
300	MER | 75016 | Coq d´or Suite | Rimsky ...
353	COL | 31809 | Symphony No. 9 | Dvorak ...
396	DG | 139201 | Violin Concerto | Beethoven ...
442	FF | 245 | Good News | Sweet Honey In The ...

Chave

Arquivo de Dados

Arquivo armazenado em disco

Arquivo de Índice

➔ Cada par (chave , localização) é um registro

- ◆ Implementação eficiente usa registros de tamanho fixo
  - Campos de tamanho fixo => chave primária e localização (RRN ou byte offset)
  - Pode eventualmente conter outros campos
    - Ex.: tamanho do registro no arquivo de dados

Arquivo de Índice

- ➔ Em geral, mantido ordenado
  - ◆ Permite busca binária (BB) se forem registros de tamanho fixo
- ➔ Menor e mais simples que o arquivo de dados original
  - ◆ Muitas vezes cabe todo em memória primária
    - ● Busca e manutenção mais eficientes!

Arquivo de Índice

➔ O Arquivo de Dados, em contraste…

- Cada registro no arquivo de dados possui um correspondente no arquivo de índice
- Em geral, muito maior que o arquivo de índices
- Em geral, possui registros de tamanho variável
- Em geral, "organizado" segundo a ordem de entrada dos registros
  - *Entry sequenced file*

Arquivo de Índice

➔ Exemplo: índice “moderado”

- Arquivo de dados com 10^6 registros de ~1KB em média
  - Deve ser indexado até byte offset ~10^6 × 1000
    - = 1 Bilhão
- 4 bytes são suficientes para representar esse offset

Arquivo de Índice

➔ Exemplo: índice “moderado”

- Arquivo de índice com registros de 24 bytes
  - 4 bytes para o byte offset + 20 bytes para a chave
    - CPF, por exemplo, requer apenas 11 bytes na maior representação
- Com 10^6 registros, arquivo de índice ocupa 24 × 10^6
  - ≅ 24MB
    - Cabe na RAM!

Operações Básicas - Índices em Memória Primária

- ➔ Criação dos arquivos de índice (*index file*) e de dados (*data file*);
- ➔ Carregar índice para memória (em geral em vetor);
- ➔ Busca;
- ➔ Inserção de registro;
- ➔ Atualização de registro;
- ➔ Eliminação de registro.

Operações Básicas - Índices em Memória Primária

➔ Busca
- Localização da chave no índice é rápida
- Em geral índice é mantido ordenado
  - Pode-se usar Busca Binária
- Se índice não estiver ordenado
  - Busca é sequencial

Operações Básicas - Índices em Memória Primária

➔ Busca

- Localização e recuperação do registro de dados
  - O(1) acessos externos
  - Qualquer consulta baseada na chave será O(1)
  - Base também para operações de remoção e atualização baseadas em chave

Operações Básicas - Índices em Memória Primária

➔ Inserção

- Novo registro é inserido no arquivo de dados
  - No final do arquivo ou
  - Seguindo uma política do tipo first-fit ou worst-fit

Operações Básicas - Índices em Memória Primária

➔ Inserção

- Um registro (entrada) correspondente é inserido no arquivo de índice
  - Campos: chave e indicador do início do novo registro no arquivo de dados (byte offset ou RRN)
  - Índice em vetor ordenado
    - Inserção demanda deslocamentos
    - Rápido em RAM!

Operações Básicas - Índices em Memória Primária

➔ Remoção

- Registro é removido do arquivo de dados segundo alguma política de marcação de registros removidos (remoção lógica)
- O registro correspondente é removido do índice
  - Deslocamentos (remoção física)
  - Ou marcação da entrada correspondente no vetor (lógica)

Operações Básicas - Índices em Memória Primária

➔ Atualização

- Registro é alterado no arquivo de dados
- Se atualização não muda o valor da chave:
  - Se tamanho do registro não aumenta, nada muda no índice
  - Caso contrário, altera-se apenas o byte offset no índice (por que?)

Operações Básicas - Índices em Memória Primária

➔ Atualização
- ◆ Se atualização muda o valor da chave
  - Altera-se o registro no vetor de índices em RAM
    - Chave e, se necessário, byte offset
  - Reordena-se o vetor de índices
  - No índice atualização pode ser tratada como remoção + inserção

Operações Básicas - Índices em Memória Primária

➔ Ao final de uma sessão de operações:

- Se o índice foi alterado em RAM => atualizar o correspondente arquivo de índice no disco
- Política de atualização definida de acordo com as características da aplicação

Operações Básicas - Índices em Memória Primária

➔ Mas... é imperativo que programas que acessam os arquivos se protejam contra índices desatualizados

- Queda de energia
- Crashs do sistema (software ou hardware)
- ...

➔ Estratégia de prevenção?

Prevenção de Índices Desatualizados

- ➔ É fundamental implementar um mecanismo que permita saber se o índice está atualizado em relação ao arquivo de dados
- ➔ Uma possível estratégia:
  - ◆ Flag de status
    - ● "Setado" no arquivo índice mantido em disco assim que a sua cópia na memória é alterada

Prevenção de Índices Desatualizados

➔ Flag de status

- Flag pode ser mantido no registro cabeçalho do arquivo índice
  - Atualizado sempre que o índice é reescrito no disco
- Se um programa detecta que o índice está desatualizado:
  - Uma função é ativada para reconstruir o índice a partir do arquivo de dados

Arquivos de Índices Grandes

➔ Se o índice não cabe na memória primária

- Acesso e manutenção feitos em memória secundária
- Nada muda para o arquivo de dados

Arquivos de Índices Grandes

- ➔ Busca
  - Busca sequencial => O(n) acessos
  - Busca Binária => O(log n) acessos
    - Pode demandar um acesso para cada registro verificado

Arquivos de Índices Grandes

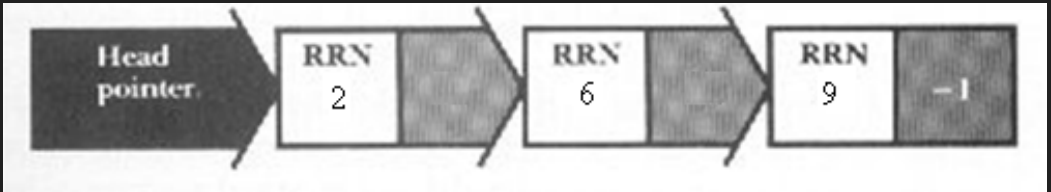
➔ Remoção
- ◆ Alternativa 1:
  - Deslocar todos os registros subsequentes no arquivo de índice para preencher espaço do registro removido
    - Otimiza espaço, mas a um custo computacional altíssimo...

Arquivos de Índices Grandes

➔ Remoção
- ◆ Alternativa 2:
  - Colocar um marcador e encadear o registro removido em uma lista de registros de índice disponíveis
    - Análogo ao que é feito para o arquivo de dados

Arquivos de Índices Grandes

- Alternativa 2:
  - Inserção deverá respeitar ordem da chave para permitir BB ...
  - Pode não valer a pena manter e percorrer a lista de disponíveis com baixa possibilidade de sucesso ...



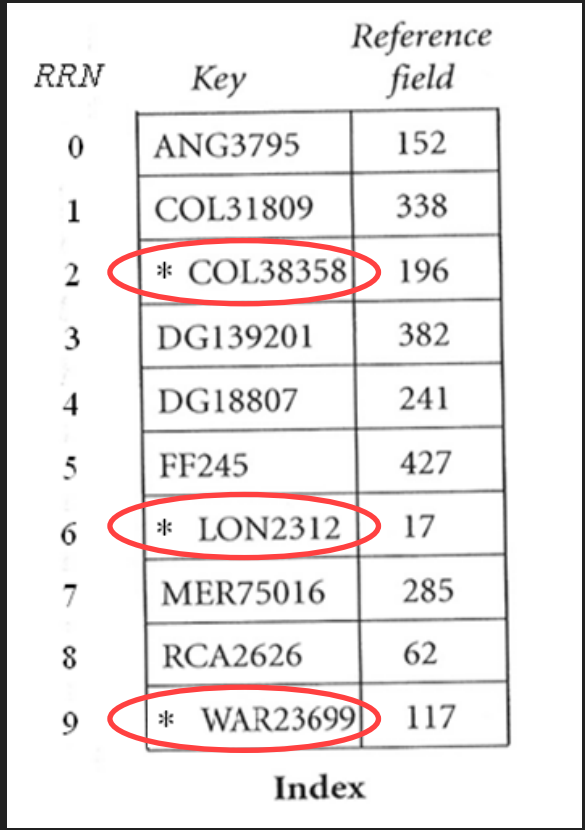
head.first_avail = 2

RRN	Key	Reference field
0	ANG3795	152
1	COL31809	338
2	*|6|COL38358	196
3	DG139201	382
4	DG18807	241
5	FF245	427
6	*|9|LON2312	17
7	MER75016	285
8	RCA2626	62
9	*|-1|WAR23699	117

Index

Arquivos de Índices Grandes

- ➔ Remoção
  - ◆ Alternativa 3:
    - Apenas marcar os registros como disponíveis (sem lista)



RRN	Key	Reference field
0	ANG3795	152
1	COL31809	338
2	* COL38358	196
3	DG139201	382
4	DG18807	241
5	FF245	427
6	* LON2312	17
7	MER75016	285
8	RCA2626	62
9	* WAR23699	117

Index

Arquivos de Índices Grandes

➔ Inserção (alternativa 3 de remoção)

- Para BB: Chave inserida deve respeitar ordem do índice
  - Busca-se pela localização onde a chave deveria ser inserida (BB)
  - Se localização corresponde a um slot disponível, tudo resolvido
  - Caso contrário, é necessário deslocar todos os registros de índice subsequentes até o próximo slot vago ou EOF

Arquivos de Índices Grandes

- ➔ Atualização
  - Se atualização muda o valor da chave:
    - Trata-se como uma remoção + inserção de registro de índice
  - Se atualização não muda o valor da chave:
    - Se tamanho do registro não aumenta, nada muda no índice
    - Caso contrário, muda-se apenas o byte offset no índice

Arquivos de Índices Grandes

- ➔ Na prática.... não é aconselhável usar índices simples se o índice não cabe em memória primária:
  - A busca binária pode exigir vários acessos a disco;
  - A necessidade de deslocar registros nas inserções e remoções
    - Manutenção do índice excessivamente cara.

Arquivos de Índices Grandes

➔ Desempenho das operações só pode ser melhorado com abordagens de indexação mais sofisticadas:

- Hashing Externo
  - Máximo desempenho para acesso direto
- Árvores-B (B-trees)
  - Bom compromisso entre desempenho, manutenção e possibilidade de acesso sequencial ordenado por chaves

Índice Secundário

➔ O que fazer quando a chave primária não é o alvo da consulta?

- Ex: CPF é uma chave muito usual, mas ... e o código do arquivo de músicas usado como exemplo na aula anterior?
  - Como saber que se deve procurar por ANG3795 quando se deseja a ficha musical de “Symphony No. 9 ”, de Beethoven ???

Índice Secundário

➔ Quais os dados da música de código ANG3795 ?

- Geralmente usado internamente por um programa, mas raramente pelo usuário de modo direto

➔ Quais os dados da Symphony No. 9, de Beethoven ?

- Consulta típica de um usuário

Índice Secundário

- Em diversas aplicações, a busca por registros não se faz por chave primária, mas por chaves secundárias
  - Você procura um livro na biblioteca por título/autor ou por ISBN?
- Como localizar o registro, de maneira eficiente, se o índice primário é construído em função da chave primária?

Índice Secundário

➔ Solução:

- Definir índices que relacionam chaves secundárias ao arquivo de dados
  - Índices Secundários

Índice Secundário

➔ Tipos de Índices Secundários

- Fracamente ligado (*Loose Binding*)
  - Relaciona uma chave secundária à chave primária
- Fortemente ligado (*Tight Binding*)
  - Relaciona uma chave secundária diretamente ao registro no arquivo de dados

Índice Secundário Fracamente Acoplado

➔ Exemplo: Arquivo de Música

- Dados o arquivo de dados e índice primário definidos na aula anterior...
  - Criar um índice secundário com Compositor como chave de indexação
  - Buscar a Symphony No 9 de Beethoven

Índice Secundário Fracamente Acoplado

ANG3795	167
COL31809	353
COL38358	211
DG139201	396
DG18807	256
FF245	442
LON2312	32
MER75016	300
RCA2626	77
WAR23699	132

Arquivo de Índice

32	LON | 2312 | Romeo and Juliet | Prokofiev ...
77	RCA | 2626 | Quartet in C Sharp Minor ...
132	WAR | 23699 | Touchstone | Corea ...
167	ANG | 3795 | Symphony No. 9 | Beethoven ...
211	COL | 38358 | Nebraska | Springsteen ...
256	DG | 18807 | Symphony No. 9 | Beethoven ...
300	MER | 75016 | Coq d´or Suite | Rimsky ...
353	COL | 31809 | Symphony No. 9 | Dvorak ...
396	DG | 139201 | Violin Concerto | Beethoven ...
442	FF | 245 | Good News | Sweet Honey In The ...

Arquivo de Dados

Chaves Repetidas

Índice Secundário Fracamente Acoplado

Beethoven	ANG3795
Beethoven	DG139201
Beethoven	DG18807
Beethoven	RCA2626
Corea	WAR23699
Dvorak	COL31809
Prokofiev	LON2312
Rimsky	MER75016
Springsteen	COL38358
Sweet Honey In The	FF245

Arquivo de Índice Secundário

ANG3795	167
COL31809	353
COL38358	211
DG139201	396
DG18807	256
FF245	442
LON2312	32
MER75016	300
RCA2626	77
WAR23699	132

Arquivo de Índice Primário

32	LON | 2312 | Rom...
77	RCA | 2626 | Quar...
132	WAR | 23699 | To...
167	ANG | 3795 | Sym...
211	COL | 38358 | Neb...
256	DG | 18807 | Symp...
300	MER | 75016 | Coq...
353	COL | 31809 | Sym...
396	DG | 139201 | Viol...
442	FF | 245 | Good N...

Arquivo de Dados

Índice Secundário Fracamente Acoplado

➔ Diferença importante entre os índices dos tipos primário e secundário:

- Índice secundário:
  - Permite múltiplos registros com chaves iguais
  - Chaves duplicadas devem ser mantidas agrupadas e ordenadas internamente ao grupo de acordo com a chave primária

Índice Secundário Fracamente Acoplado

➔ Diferença importante entre os índices dos tipos primário e secundário:

- ◆ Índice secundário:
  - ● Permite consultas eficientes envolvendo combinações de chaves secundárias
    - ○ Múltiplos índices

Operações Básicas

➔ Busca

- Pesquisar o índice de chave secundária para encontrar a chave primária relacionada
- Usar a chave primária para pesquisar o índice de chave primária para encontrar o byte offset (ou RRN) do registro no arquivo de dados
- Recuperar o registro no arquivo de dados

Operações Básicas

➔ Inserção:

- Inserir novo registro no arquivo de dados
- Inserir a entrada correspondente no índice primário
- Inserir a entrada correspondente em cada índice secundário
  - Entradas duplicadas devem ser mantidas agrupadas e ordenadas
  - Pode ser muito custoso se os arquivos de índices não couberem em RAM

Operações Básicas

- ➔ Atualização (3 situações):
  - ◆ Situação 1:
    - Alteração de chave secundária
    - Índice secundário para esta chave precisa ser reordenado

Operações Básicas

➔ Situação 2:

- Alteração de chave primária
- Índice primário precisa ser reordenado
- Índices secundários precisam ser varridos e as entradas contendo a chave primária alterada devem ser atualizadas
  - Se houver chaves secundárias duplicadas, pode ser necessário reordená-las localmente pela chave primária

Operações Básicas

➔ Situação 3:

- Alteração de outros campos apenas
- Não afeta nenhum dos índices
- Se necessário: atualizar o byte offset do índice primário

Operações Básicas

- Remoção (2 situações):
  - Abordagem *delete all references*
    - Remover o registro do arquivo de dados
    - Remover a entrada correspondente do arquivo de índice primário

Operações Básicas

- Abordagem *delete all references*
  - Remover as entradas correspondentes de cada arquivo de índice secundário
  - Buscar o registro, gerenciar espaços vagos e reordenar registros em múltiplos arquivos de índices pode ser custoso se não couberem em RAM

Operações Básicas

- Remoção (2 situações):
  - Abordagem *Delete Some References*
    - Remover o registro no arquivo de dados
    - Remover a entrada correspondente no arquivo de índice primário
    - As entradas correspondentes nos índices secundários são mantidas

Operações Básicas

➔ Abordagem *Delete Some References*

- Busca no arquivo de índice secundário, por uma chave primária que não existe mais, indicará que o registro foi removido
  - Nesse momento, é possível eliminar o registro do índice secundário
- Custo computacional extra associado?
  - Busca por chave inexistente no índice primário

Operações Básicas

	delete all references	*delete some references*
vantagens	➔ sem queda de desempenho na busca por registros removidos ➔ índices permanecem do tamanho necessário	➔ mais simples - sem necessidade de reorganização a cada remoção ◆ economia de tempo nas remoções
desvantagens	➔ necessidade de reorganização a cada remoção ◆ processo altamente custoso, devido à ordenação	➔ com queda de desempenho na busca na busca por registros removidos ➔ crescimento do tamanho dos índices secundários e necessidade de reorganização periódica

Busca em Múltiplos Índices Secundários

- ➔ Múltiplos índices permitem manter diferentes visões dos registros de um mesmo arquivo de dados
  - ◆ Consultas por chaves secundárias diferentes
    - Localizar conjuntos de registros do arquivo de dados usando uma ou mais chaves

Busca em Múltiplos Índices Secundários

➔ Possível fazer uma busca (consulta) em vários índices e combinar (AND, OR, NOT) os resultados individuais

- ◆ Combina chaves relacionadas
  - ● Consultas que combinam visões particulares

Ex: encontre todos os registros tal que:

composer = "BEETHOVEN" **AND** title = "SYMPHONY NO. 9“

Busca em Múltiplos Índices Secundários

➔ Exemplo: Arquivo de Músicas – Múltiplos Índices

Índice por compositor

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

Índice por título

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

Busca em Múltiplos Índices Secundários

➔ Exemplo: composer = "BEETHOVEN" AND title = "SYMPHONY NO. 9“

ANG3795
DG139201
DG18807
RCA2626

ANG3795
COL31809
DG18807

AND

➔ Resultado

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

Busca em Múltiplos Índices Secundários

- ➔ Processamento co-sequencial dos arquivos
  - ◆ Beneficia-se da ordenação local pelas chaves primárias!

Índice Secundário Fortemente Acoplado

- ➔ Índice fortemente Acoplado
  - ◆ Relaciona chave secundária diretamente ao registro no arquivo de dados

Índice Secundário Fortemente Acoplado

Beethoven	167
Beethoven	353
Beethoven	211
Beethoven	396
Corea	256
Dvorak	442
Prokofiev	32
Rimsky	300
Springsteen	77
Sweet Honey In The	132

Arquivo de Índice Secundário

32	LON | 2312 | Romeo and Juliet | Prokofiev ...
77	RCA | 2626 | Quartet in C Sharp Minor ...
132	WAR | 23699 | Touchstone | Corea ...
167	ANG | 3795 | Symphony No. 9 | Beethoven ...
211	COL | 38358 | Nebraska | Springsteen ...
256	DG | 18807 | Symphony No. 9 | Beethoven ...
300	MER | 75016 | Coq d´or Suite | Rimsky ...
353	COL | 31809 | Symphony No. 9 | Dvorak ...
396	DG | 139201 | Violin Concerto | Beethoven ...
442	FF | 245 | Good News | Sweet Honey In The ...

Arquivo de Dados

Operações Básicas

- ➔ Busca
  - ◆ Pesquisar o índice de chave secundária para encontrar o byte offset (ou RRN) do registro no arquivo de dados

Operações Básicas

➔ Inserção

- Inserir o registro no arquivo de dados
- Inserir a entrada correspondente em cada arquivo de índice secundário
- Chaves duplicadas devem ser mantidas agrupadas e ordenadas

Operações Básicas

- Remoção
  - Remover o registro no arquivo de dados
    - *Delete all references*: remover a entrada correspondente em cada arquivo de índice secundário

Operações Básicas

- Atualização
  - Situação 1: alteração de chave secundária
    - Índice secundário para esta chave precisa ser reordenado
  - Situação 2: alteração de outros campos apenas
    - Não afeta nenhum dos índices
  - Se necessário: atualizar o byte offset no índice secundário

Tipos de Índice Secundário- Resumo

	fracamente acoplado	fortemente acoplado
vantagens	diminui custo de remoções na abordagem *delete some references*: modificação no arquivo de dados afeta apenas o índice primário	Acesso direto: índice primário -> arquivo de dados índice secundário -> arquivo de dados
	menor complexidade de codificação	melhor desempenho na busca

Tipos de Índice Secundário- Resumo

	fracamente acoplado	fortemente acoplado
desvantagens	Acesso indireto: Índice secundário -> Índice primário -> arquivo de dados	Alto custo para modificações: Modificações no arquivo de dados afeta todos os índices secundários
	Queda no desempenho na busca	Maior complexidade de codificação

Índices Secundários Melhorados

➔ Problemas nas estruturas de índices secundário vistas até agora:

- Repetição de chaves secundárias
  - Arquivos de índices secundários maiores que o necessário

Índices Secundários Melhorados

➔ Problemas nas estruturas de índices secundário vistas até agora:

- Necessidade de rearranjar os índices mesmo quando o novo registro (inserção) tem um valor de chave secundária já existente
  - Ex: se uma nova gravação da Sinfonia no. 9 de Beethoven for inserida no arquivo de música

Índices Secundários Melhorados

➔ Solução 1: Vetor de Tamanho Fixo

- Para índices fracamente acoplados
  - Permitir a associação de múltiplas chaves primárias (tamanho fixo) a cada chave secundária

Índices Secundários Melhorados

Beethoven	ANG3795	DG139201	DG18807	RCA2626
Corea	WAR23699			
Dvorak	COL31809			
Prokofiev	LON2312			
Rimsky	MER75016			
Springsteen	COL38358			
Sweet Honey In The	FF245			

Índices Secundários Melhorados

➔ Solução 1: Vetor de Tamanho Fixo

- Elimina entradas com chaves secundárias duplicadas
- Não é necessário reordenar o índice a cada inserção de registro com chave secundária já existente

Índices Secundários Melhorados

➔ Solução 1: Vetor de Tamanho Fixo

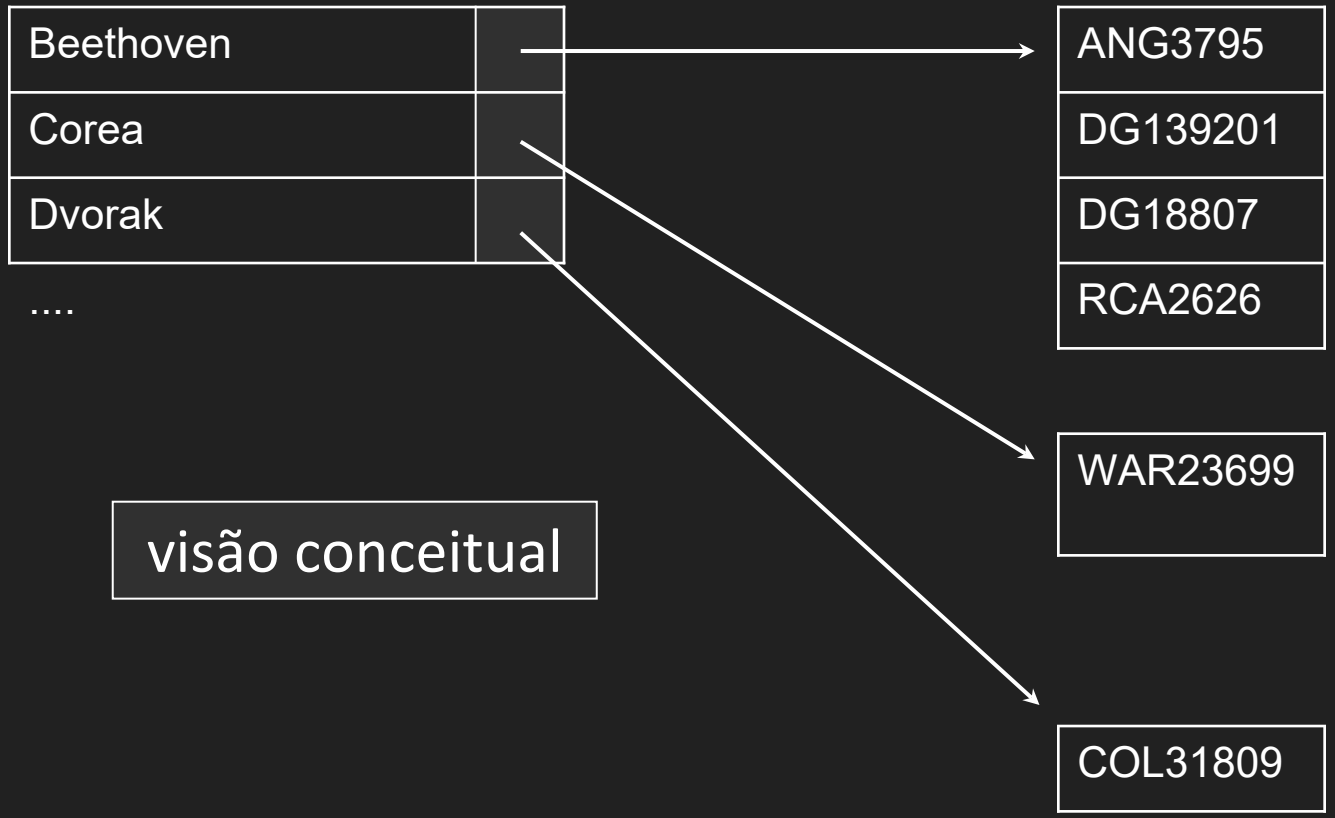
- Porém:
  - É limitado a um número fixo de repetições da chave
  - Quanto maior esse número, maior a fragmentação interna do arquivo de índice !
    - Talvez não compense a eliminação das chaves duplicadas

Índices Secundários Melhorados

➔ Solução 2: Listas invertidas

- Para índices fracamente acoplados
  - Cada chave secundária é associada a uma lista encadeada de chaves primárias (ordenadas)

Índices Secundários Melhorados



Índices Secundários Melhorados

➔ Exemplo de **inserção** de novo registro para Dvorak

Índices Secundários Melhorados

➔ Solução 2: Listas invertidas

- Implementação:
  - Para cada valor de chave secundária, cria-se no arquivo de índice secundário apenas uma entrada com uma referência ao RRN do primeiro registro da lista encadeada que armazena as chaves primárias correspondentes

Índices Secundários Melhorados

➔ Solução 2: Listas invertidas

- Implementação:
  - As listas encadeadas de chaves primárias ORDENADAS correspondentes às chaves secundárias são mantidas em um arquivo sequencial separado, organizado segundo a entrada dos registros
    - Entry sequenced file

Listas Invertidas

Arquivo de Índice Secundário

Beethoven	3
Corea	2
Dvorak	7
Prokofiev	0
Rimsky	6
Springsteen	4
Sweet Honey In The	9
…	

Arquivo de Listas Invertidas

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

Campo de chave secundária

Campo com **RRN** da primeira referência da chave primária na lista invertida

Campo de chave primária

Campo com **RRN** da próxima referência da chave primária na lista invertida, ou -1

Listas Invertidas

Arquivo de Índice Secundário

Beethoven	3
Corea	2
Dvorak	7
Prokofiev	0
Rimsky	6
Springsteen	4
Sweet Honey In The	9
…	

Arquivo de Listas Invertidas

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

Ex:
Chave: Beethoven
Lista de Chaves Primárias: ANG3795 -> DG139201 -> DG18807 -> RCA2626

Listas Invertidas

Arquivo de Índice Secundário

Beethoven	3
Corea	2
Dvorak	7
Prokofiev	0
Rimsky	6
Springsteen	4
Sweet Honey In The	9
…	

Arquivo de Listas Invertidas

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

Ex: Inserção de um novo registro relativo a Dvorak

Antes:
lista de Códigos: COL31809

Listas Invertidas

Arquivo de Índice Secundário

Beethoven	3
Corea	2
Dvorak	10
Prokofiev	0
Rimsky	6
Springsteen	4
Sweet Honey In The	9
…	

Arquivo de Listas Invertidas

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	AMB37829	7

Ex: Inserção de um novo registro relativo a Dvorak

Antes:
lista de Códigos: AMB37829 -> COL31809

Listas Invertidas

➔ Exemplo: inserção de novo registro (Dvorak)

- Inserir o novo registro no arquivo de dados
- Inserir a entrada correspondente no índice primário
  - AMB37829

Listas Invertidas

➔ Como a chave secundária (Dvorak) já existe no índice secundário

- Inserir nova chave primária (AMB37829) na lista invertida de Dvorak (“inserção ORDENADA na lista”)
- Inserção do registro no final do arquivo
- Atualização de RRNs no arquivo de índice secundário e lista invertida para manter ordenação

Listas Invertidas

➔ Vantagens:

- Índice secundário só precisa ser alterado quando:
  - For inserido um registro com chave secundária ainda não existente
  - For inserido/removido registro cabeça de lista invertida
  - For alterada uma chave (primária ou secundária) já existente

Listas Invertidas

➔ Vantagens:

- Quando necessário, rearranjar o índice é mais simples:
  - Contém menos registros
  - Não existe duplicidade de chaves secundárias

Listas Invertidas

➔ 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

Listas Invertidas

➔ Vantagens:

- Arquivo de listas invertidas nunca precisa ser ordenado, pois é *entry sequenced*
  - Unica preocupação é encadear cada lista de forma ordenada segundo a chave primária (ordenação lógica)
  - Logo, é trivial reutilizar o espaço liberado por registros eliminados do arquivo de listas invertidas (registros de tamanho fixo)

Listas Invertidas

➔ Desvantagens:

- Registros associados a um valor de chave secundária, encadeados em uma mesma lista de chaves primárias, não estão adjacentes no disco:
  - Podem ser necessários vários *seeks* para recuperar os registros em 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...

Índices Seletivos

- ➔ O índice secundário não precisa cobrir todo o arquivo de dados
  - ◆ Exemplos:
    - Índice de músicas do gênero rock
    - Índice de músicas lançadas depois de 1980
- ➔ Dependente da aplicação
  - ◆ Operações e uso dos dados

Referências

- M. J. Folk, B. Zoellick and G. Riccardi. File Structures: An object-oriented approach with C++, Addison Wesley, 1998.
- R. Elmasri, S. Navathe. Sistemas de Banco de Dados, Person, 6a Edição, 2010.