Arquivos Indexados por Chaves Secundárias

Vanessa Braganholo

Arquivos Indexados

- Até agora, as alternativas que vimos funcionam apenas para indexar arquivos por chaves primárias
- Isso otimiza a busca de um registro pelo valor da chave

- Em várias aplicações, é necessário buscar registros por atributos não chave

Arquivos Indexados

- **Exemplo**: Recupere todos os registros de clientes que residem no estado de SP

CodCli	Nome	Cidade	Estado
01	MURILO	RIO DE JANEIRO	RJ
03	MARIA	RIO CLARO	SP
05	PEDRO	RIO DE JANEIRO	RJ
07	CAROLINA	SAO PAULO	SP
10	JOAO	RIO DE JANEIRO	RJ
15	CARLOS	BELO HORIZONTE	MG

- Neste cenário, arquivos indexados pela chave primária não resolvem o problema
- A consulta tem que ser respondida através de uma busca sequencial no arquivo

Como evitar a busca sequencial?

Solução:
Indexar o arquivo pelas chaves secundárias

CodCli	Nome	Cidade	Estado
01	MURILO	RIO DE JANEIRO	RJ
03	MARIA	RIO CLARO	SP
05	PEDRO	RIO DE JANEIRO	RJ
07	CAROLINA	SAO PAULO	SP
10	JOAO	RIO DE JANEIRO	RJ
15	CARLOS	BELO HORIZONTE	MG

- Pode-se usar um índice sobre o atributo Estado, outro sobre Cidade, e outro sobre Nome, se necessário
- Consultas podem ser respondidas por interseção ou união de conjuntos de listas

Solução:
Indexar o arquivo pelas chaves secundárias

CodCli	Nome	Cidade	Estado
01	MURILO	RIO DE JANEIRO	RJ
03	MARIA	RIO CLARO	SP
05	PEDRO	RIO DE JANEIRO	RJ
		SAO PAULO	

Arquivo Invertido

- Pode-se [illegible] sobre Cidade, [illegible] necessário
- Consultas podem ser respondidas por interseção ou união de conjuntos de listas

Como funciona este índice?

- Arquivo invertido:
  - A cada valor de chave secundária diferente que aparece no arquivo, está associado um conjunto de endereços que possuem aquele valor de chave

- De onde vem o nome Arquivo Invertido?
  - Em um arquivo comum, ao fornecer um endereço, tem-se a lista dos valores dos atributos do registro armazenado naquele endereço
  - Em um arquivo invertido, ao fornecer um valor de atributo, tem-se a lista de endereços dos registros que possuem aquele valor de atributo
  - Funciona como se fosse uma inversão da "função" arquivo

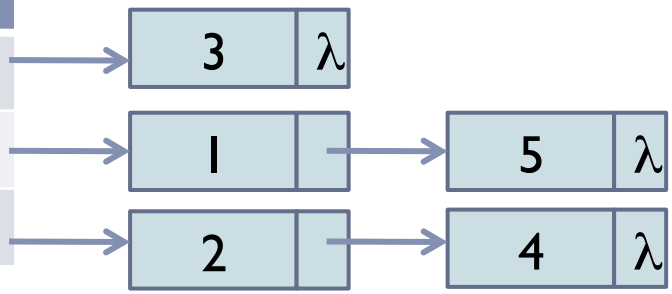
Exemplo: Arquivo Invertido para o Atributo Estado

Arquivo Original

	CodCli	Nome	Cidade	Estado
1	10	JOAO	RIO DE JANEIRO	RJ
2	02	MARIA	RIO CLARO	SP
3	15	CARLOS	BELO HORIZONTE	MG
4	04	CAROLINA	SAO PAULO	SP
5	01	MURILO	ANGRA DOS REIS	RJ

Arquivo Invertido para Estado

	Estado	Q
1	MG	1
2	RJ	2
3	SP	2



Implementação de Arquivos Invertidos

Fonte de consulta: Ferraz, I. Programação com Arquivos, ed. Manole. Capítulo 7.

Uso de Listas Encadeadas

Arquivo de Índice (Estado)

Estado	PT	Q
MG	6	1
RJ	1	3
SP	2	2

Arquivo de Dados

	CodCli	Nome	Cidade	Estado	Próximo Estado
1	01	MURILO	RIO DE JANEIRO	RJ	3
2	03	MARIA	RIO CLARO	SP	4
3	05	PEDRO	RIO DE JANEIRO	RJ	5
4	07	CAROLINA	SAO PAULO	SP	-1
5	10	JOAO	RIO DE JANEIRO	RJ	-1
6	15	CARLOS	BELO HORIZONTE	MG	-1

Uso de Listas Encadeadas

Arquivo de Índice (Estado)

Estado	PT	Q
MG	6	1
RJ	1	3
SP	2	2

O arquivo de índice tem o endereço do primeiro registro que possui aquele valor.
O arquivo de índice é **ordenado** pelo valor do atributo que está indexando.

Arquivo de Dados

	CodCli	Nome	Cidade	Estado	Próximo Estado
1	01	MURILO	RIO DE JANEIRO	RJ	3
2	03	MARIA	RIO CLARO	SP	4
3	05	PEDRO	RIO DE JANEIRO	RJ	5
4	07	CAROLINA	SAO PAULO	SP	-1
5	10	JOAO	RIO DE JANEIRO	RJ	-1
6	15	CARLOS	BELO HORIZONTE	MG	-1

Pode-se ter várias listas encadeadas

- Uso de várias listas, uma para cada atributo que se deseja indexar
- Isso implica que serão usados vários arquivos de índice, um para cada atributo

Arquivos Multilista

Arquivo de Índice (Estado)

Estado	PT	Q
MG	6	1
RJ	1	3
SP	2	2

Arquivo de Índice (Cidade)

Cidade	PT	Q
BELO HORIZONTE	6	1
RIO CLARO	2	1
RIO DE JANEIRO	1	3
SAO PAULO	4	1

Arquivo de Dados

	CodCli	Nome	Cidade	Estado	Próx Estado	Próx Cidade
1	01	MURILO	RIO DE JANEIRO	RJ	3	3
2	03	MARIA	RIO CLARO	SP	4	-1
3	05	PEDRO	RIO DE JANEIRO	RJ	5	5
4	07	CAROLINA	SAO PAULO	SP	-1	-1
5	10	JOAO	RIO DE JANEIRO	RJ	-1	-1
6	15	CARLOS	BELO HORIZONTE	MG	-1	-1

Método de Indexação

- Suponha que já existe um arquivo de dados, com vários registros
- Suponha que desejamos criar dois índices para este arquivo

- Como poderíamos implementar a criação destes índices?
  - Exige criação dos dois arquivos de índice
  - Exige adição de duas colunas no arquivo de dados

Algoritmo de Lefkowitz

- Algoritmo em 6 passos que cria os índices e o arquivo de dados modificado
- Notação:
  - ED: Endereço de disco. Numeração dos registros a partir do início do arquivo
  - CP: Chave primária
  - CS: Chave secundária
  - Q: Quantidade de registros que possuem uma determinada chave secundária
  - PT: Ponteiro para registro que possui uma determinada chave secundária
  - AT: Atributo não chave
  - PROX: Ponteiro para o próximo registro que possui uma determinada chave secundária

Algoritmo de Lefkowitz - Passo 1

- Arquivo de dados original, **ordenado pela chave primária**, é chamado A1
- Criar um novo arquivo A2
- Copiar para este arquivo a Chave Primária (CP) e todas as Chaves Secundárias (CSs) para as quais se deseja construir um índice
- Inserir também uma coluna ED (Endereço de Disco) para cada registro

- Estrutura Resultante do Arquivo A2

ED	CP	CS_1	CS_2		CS_N

Exemplo

- Arquivo Original
- Indexar por Cidade e Estado

Arquivo de Dados A1

CodCli	Nome	Cidade	Estado
01	MURILO	RIO DE JANEIRO	RJ
03	MARIA	RIO CLARO	SP
05	PEDRO	RIO DE JANEIRO	RJ
07	CAROLINA	SAO PAULO	SP
10	JOAO	RIO DE JANEIRO	RJ
15	CARLOS	BELO HORIZONTE	MG

Resultado do Passo 1

Arquivo de Dados A1

CodCli	Nome	Cidade	Estado
01	MURILO	RIO DE JANEIRO	RJ
03	MARIA	RIO CLARO	SP
05	PEDRO	RIO DE JANEIRO	RJ
07	CAROLINA	SAO PAULO	SP
10	JOAO	RIO DE JANEIRO	RJ
15	CARLOS	BELO HORIZONTE	MG

Arquivo A2

ED	CodCli	Cidade	Estado
1	01	RIO DE JANEIRO	RJ
2	03	RIO CLARO	SP
3	05	RIO DE JANEIRO	RJ
4	07	SAO PAULO	SP
5	10	RIO DE JANEIRO	RJ
6	15	BELO HORIZONTE	MG

Algoritmo de Lefkowitz - Passo 2

- Decompor o arquivo A2 em vários arquivos A3, um para cada atributo que será indexado
- Cada arquivo A3 terá a seguinte estrutura:

Resultado do Passo 2

Arquivo A2

ED	CodCli	Cidade	Estado
1	01	RIO DE JANEIRO	RJ
2	03	RIO CLARO	SP
3	05	RIO DE JANEIRO	RJ
4	07	SAO PAULO	SP
5	10	RIO DE JANEIRO	RJ
6	15	BELO HORIZONTE	MG

Arquivo A3-Estado

ED	CodCli	Estado
1	01	RJ
2	03	SP
3	05	RJ
4	07	SP
5	10	RJ
6	15	MG

Arquivo A3-Cidade

ED	CodCli	Cidade
1	01	RIO DE JANEIRO
2	03	RIO CLARO
3	05	RIO DE JANEIRO
4	07	SAO PAULO
5	10	RIO DE JANEIRO
6	15	BELO HORIZONTE

Algoritmo de Lefkowitz - Passo 3

- Ordenar os arquivos A3 por chave secundária, gerando arquivos A4

Resultado do Passo 3

Arquivo A3-Estado

ED	CodCli	Estado
1	01	RJ
2	03	SP
3	05	RJ
4	07	SP
5	10	RJ
6	15	MG

Arquivo A3-Cidade

ED	CodCli	Cidade
1	01	RIO DE JANEIRO
2	03	RIO CLARO
3	05	RIO DE JANEIRO
4	07	SAO PAULO
5	10	RIO DE JANEIRO
6	15	BELO HORIZONTE

Arquivo A4-Estado

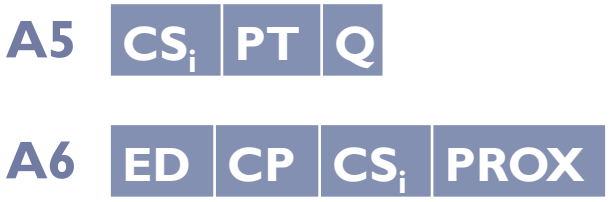
ED	CodCli	Estado
6	15	MG
1	01	RJ
3	05	RJ
5	10	RJ
2	03	SP
4	07	SP

Arquivo A4-Cidade

ED	CodCli	Cidade
6	15	BELO HORIZONTE
2	03	RIO CLARO
1	01	RIO DE JANEIRO
3	05	RIO DE JANEIRO
5	10	RIO DE JANEIRO
4	07	SAO PAULO

Algoritmo de Lefkowitz - Passo 4

- Cada arquivo A4 é processado para adicionar a quantidade de registros que possuem a Chave Secundária CS_i, e o endereço do primeiro registro que possui CS_i, gerando vários arquivos A5 (um para cada arquivo A4)
- Cada arquivo A4 é processado para adicionar o endereço do próximo registro que contém a Chave Secundária CS_i, gerando vários arquivos A6 (um para cada arquivo A4)



Resultado do Passo 4

Arquivo A4-Estado

ED	CodCli	Estado
6	15	MG
1	01	RJ
3	05	RJ
5	10	RJ
2	03	SP
4	07	SP

Arquivo A5-Estado

Estado	PT	Q
MG	6	1
RJ	1	3
SP	2	2

Arquivo A6-Estado

ED	CodCli	Estado	PROX
6	15	MG	-1
1	01	RJ	3
3	05	RJ	5
5	10	RJ	-1
2	03	SP	4
4	07	SP	-1

Resultado do Passo 4

Arquivo A4-Cidade

ED	CodCli	Cidade
6	15	BELO HORIZONTE
2	03	RIO CLARO
1	01	RIO DE JANEIRO
3	05	RIO DE JANEIRO
5	10	RIO DE JANEIRO
4	07	SAO PAULO

Arquivo A5-Cidade

Cidade	PT	Q
BELO HORIZONTE	6	1
RIO CLARO	2	1
RIO DE JANEIRO	1	3
SAO PAULO	4	1

Arquivo A6-Cidade

ED	CodCli	Cidade	PROX
6	15	BELO HORIZONTE	-1
2	03	RIO CLARO	-1
1	01	RIO DE JANEIRO	3
3	05	RIO DE JANEIRO	5
5	10	RIO DE JANEIRO	-1
4	07	SAO PAULO	-1

Algoritmo de Lefkowitz - Passo 5

- Ordenar os arquivos A6 por chave primária, gerando arquivos A7

Resultado do Passo 5

Arquivo A6-Estado

ED	CodCli	Estado	PROX
6	15	MG	-1
1	01	RJ	3
3	05	RJ	5
5	10	RJ	-1
2	03	SP	4
4	07	SP	-1

Arquivo A7-Estado

ED	CodCli	Estado	PROX
1	01	RJ	3
2	03	SP	4
3	05	RJ	5
4	07	SP	-1
5	10	RJ	-1
6	15	MG	-1

Resultado do Passo 5

Arquivo A6-Cidade

ED	CodCli	Cidade	PROX
6	15	BELO HORIZONTE	-1
2	03	RIO CLARO	-1
1	01	RIO DE JANEIRO	3
3	05	RIO DE JANEIRO	5
5	10	RIO DE JANEIRO	-1
4	07	SAO PAULO	-1

Arquivo A7-Cidade

ED	CodCli	Cidade	PROX
1	01	RIO DE JANEIRO	3
2	03	RIO CLARO	-1
3	05	RIO DE JANEIRO	5
4	07	SAO PAULO	-1
5	10	RIO DE JANEIRO	-1
6	15	BELO HORIZONTE	-1

Algoritmo de Lefkowitz – Passo 6

- Juntar o arquivo A1 com todos os arquivos A7, gerando um arquivo A8 (no arquivo A8, a coluna ED não deve estar presente)
- Os arquivos de índice são os arquivos A5

Resultado do Passo 6

Arquivo de Dados A1

CodCli	Nome	Cidade	Estado
01	MURILO	RIO DE JANEIRO	RJ
03	MARIA	RIO CLARO	SP
05	PEDRO	RIO DE JANEIRO	RJ
07	CAROLINA	SAO PAULO	SP
10	JOAO	RIO DE JANEIRO	RJ
15	CARLOS	BELO HORIZONTE	MG

Arquivo A7-Estado

ED	CodCli	Estado	PROX
1	01	RJ	3
2	03	SP	4
3	05	RJ	5
4	07	SP	-1
5	10	RJ	-1
6	15	MG	-1

Arquivo A7-Cidade

ED	CodCli	Cidade	PROX
1	01	RIO DE JANEIRO	3
2	03	RIO CLARO	-1
3	05	RIO DE JANEIRO	5
4	07	SAO PAULO	-1
5	10	RIO DE JANEIRO	-1
6	15	BELO HORIZONTE	-1

Arquivo A8

CodCli	Nome	Cidade	Estado	PROXEstado	PROXCidade
01	MURILO	RIO DE JANEIRO	RJ	3	3
03	MARIA	RIO CLARO	SP	4	-1
05	PEDRO	RIO DE JANEIRO	RJ	5	5
07	CAROLINA	SAO PAULO	SP	-1	-1
10	JOAO	RIO DE JANEIRO	RJ	-1	-1
15	CARLOS	BELO HORIZONTE	MG	-1	-1

Resultado Final

Arquivo A8 – Arquivo de Dados

CodCli	Nome	Cidade	Estado	PROXEstado	PROXCidade
01	MURILO	RIO DE JANEIRO	RJ	3	3
03	MARIA	RIO CLARO	SP	4	-1
05	PEDRO	RIO DE JANEIRO	RJ	5	5
07	CAROLINA	SAO PAULO	SP	-1	-1
10	JOAO	RIO DE JANEIRO	RJ	-1	-1
15	CARLOS	BELO HORIZONTE	MG	-1	-1

Arquivo A5-Estado

Estado	PT	Q
MG	6	1
RJ	1	3
SP	2	2

Arquivo A5-Cidade

Cidade	PT	Q
BELO HORIZONTE	6	1
RIO CLARO	2	1
RIO DE JANEIRO	1	3
SAO PAULO	4	1

Curiosidade

- Arquivos invertidos são muito utilizados em máquinas de busca (ex. Google) para saber quais documentos contêm uma determinada palavra

Não será cobrado na prova

Exemplo de Documentos a serem indexados

Doc1	A casa amarela é bonita
Doc2	O carro amarelo está amassado
Doc3	Os carros são velozes
Doc4	A febre amarela é uma doença

Passo 1

- Palavras que não possuem significado (artigos, preposições) são eliminadas (**remoção de *stop words***)
- Isso ajuda a diminuir o tamanho do índice

Doc1	casa amarela é bonita
Doc2	carro amarelo está amassado
Doc3	carros são velozes
Doc4	febre amarela é doença

Passo 2

- As palavras precisam passar por um processo de radicalização, para garantir que variações da mesma palavra sejam indexadas juntas
- Cada palavra é reduzida ao seu radical
- Se não fizermos isso, ao procurar pela palavra "carro", o usuário não encontraria o Doc3

Doc1	casa amarela é bonita
Doc2	carro amarelo está amassado
Doc3	carros são velozes
Doc4	febre amarela é doença

Depois da radicalização (exemplo simplificado)

Doc1	casa amarel ser bonit
Doc2	carro amarel estar amassad
Doc3	carro ser veloz
Doc4	febre amarel ser doença

Passo 3

- Construir o arquivo invertido

Doc1	casa amarel ser bonit
Doc2	carro amarel estar amassad
Doc3	carro ser veloz
Doc4	febre amarel ser doença

amarel	3	(Doc1) (Doc2) (Doc 4)
amassad	1	(Doc 2)
bonit	1	(Doc1)
casa	1	(Doc1)
carro	2	(Doc2) (Doc3)
doença	1	(Doc4)
estar	1	(Doc2)
febre	1	(Doc4)
ser	3	(Doc1) (Doc3) (Doc4)
veloz	1	(Doc3)

Consultas

- As consultas passam pelo mesmo processo de eliminação de stop words e radicalização
- Depois, basta ir direto ao índice

- Consultas com AND: intersecção das listas de resultado
- Consultas com OR: união das listas de resultado

Exemplo

- Consulta: "carros AND amarelos"
- Resultado: Doc2

amarel	3	(Doc1) (Doc2) (Doc 4)
amassad	1	(Doc 2)
bonit	1	(Doc1)
casa	1	(Doc1)
carro	2	(Doc2) (Doc3)
doença	1	(Doc4)
estar	1	(Doc2)
febre	1	(Doc4)
ser	3	(Doc1) (Doc3) (Doc4)
veloz	1	(Doc3)

Doc1	casa amarel ser bonit
Doc2	carro amarel estar amassad
Doc3	carro ser veloz
Doc4	febre amarel ser doença

Exercício

- Aplicar o algoritmo de Lefkowitz para indexar o arquivo abaixo por País

CodCli	Nome	País
01	MURILO	BRASIL
03	MARIA	EUA
05	PEDRO	MÉXICO
07	CAROLINA	BRASIL
10	JOAO	BRASIL
15	CARLOS	EUA
21	VANESSA	BRASIL
35	LEO	MÉXICO
42	BRUNA	CANADÁ

Implementar o Algoritmo de Lefkowitz

- Entrada: arquivo binário com a seguinte estrutura (COD, NOME, IDADE), onde COD é a chave primária, e IDADE é o atributo que será indexado
- Saída: arquivo de índice