

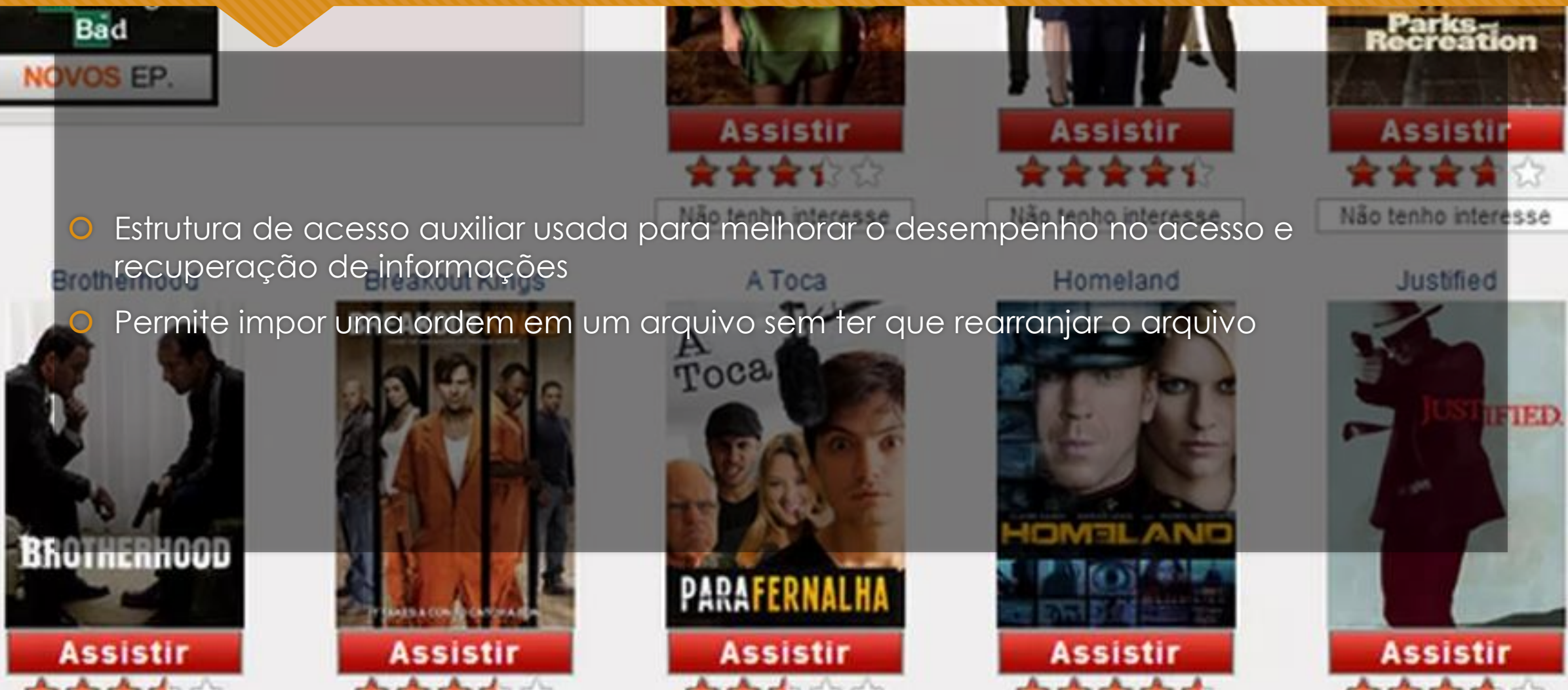
# Estruturas de Indexação de Dados



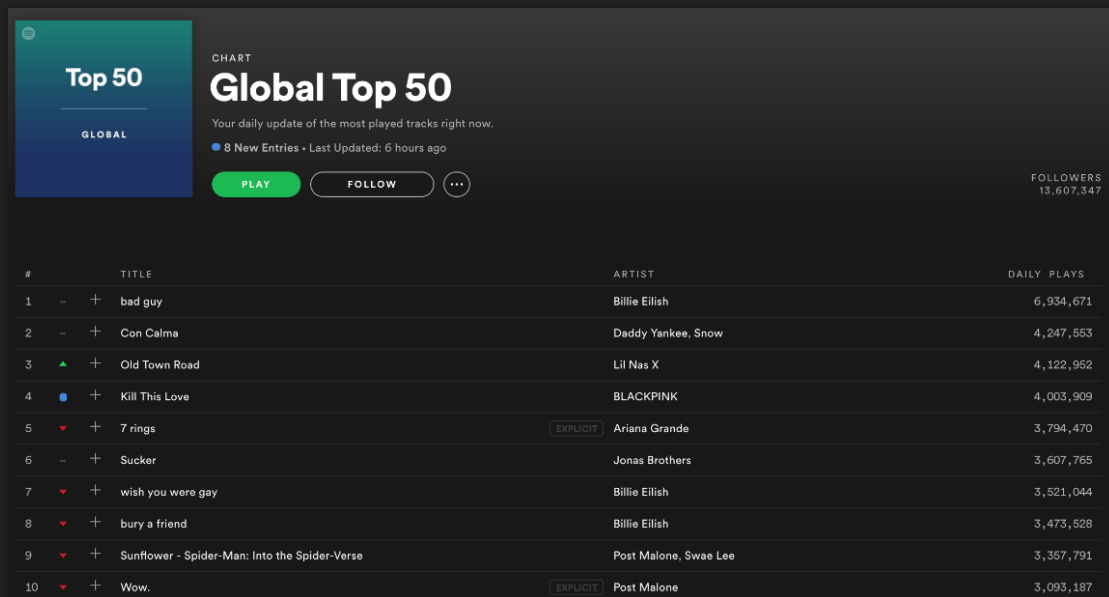
Prof. André Backes | @progdescomplicada

# Índice

- Estrutura de acesso auxiliar usada para melhorar o desempenho no acesso e recuperação de informações
- Permite impor uma ordem em um arquivo sem ter que rearranjar o arquivo



# Índice



The screenshot shows the Spotify 'Global Top 50' chart. At the top, there's a header with 'Top 50' and 'GLOBAL'. Below it, a 'CHART' section titled 'Global Top 50' includes the text 'Your daily update of the most played tracks right now.' and '8 New Entries • Last Updated: 6 hours ago'. There are 'PLAY', 'FOLLOW', and a menu icon button. On the right, it says 'FOLLOWERS 13,607,347'. The main table lists the top 10 tracks with columns for rank, title, artist, and daily plays.

#		TITLE	ARTIST	DAILY PLAYS
1	- +	bad guy	Billie Eilish	6,934,671
2	- +	Con Calma	Daddy Yankee, Snow	4,247,853
3	▲ +	Old Town Road	Lil Nas X	4,122,952
4	● +	Kill This Love	BLACKPINK	4,003,909
5	▼ +	7 rings	<span>EXPLICIT</span> Ariana Grande	3,794,470
6	- +	Sucker	Jonas Brothers	3,607,765
7	▼ +	wish you were gay	Billie Eilish	3,521,044
8	▼ +	bury a friend	Billie Eilish	3,473,528
9	▼ +	Sunflower - Spider-Man: Into the Spider-Verse	Post Malone, Swae Lee	3,357,791
10	▼ +	Wow.	<span>EXPLICIT</span> Post Malone	3,093,187

CAPÍTULO 10 297

## Tabela *Hash* 297

Definição 298

Aplicações 299

Criando o TAD tabela *Hash* 300

Definindo o tipo tabela *Hash* 300

Criando e destruindo uma tabela *Hash* 301

Calculando a posição da chave: função de *Hashing* 304

Método da divisão 304

Método da multiplicação 305

Método da sobra 306

Tratando uma *string* como chave 307

Inserção e busca sem colisão 308

*Hashing* universal 310

*Hashing* perfeito e imperfeito 311

Tratamento de colisões 311

Endereçamento aberto 312

Encadeamento separado 319

# Índice

- É definido com base em um único campo do arquivo, chamado de campo de indexação
- Assim, a pesquisa é restringida a um subconjunto dos registros, em contrapartida à análise do conjunto completo
- Os valores dos índices são ordenados para possibilitar busca binária

# Índice

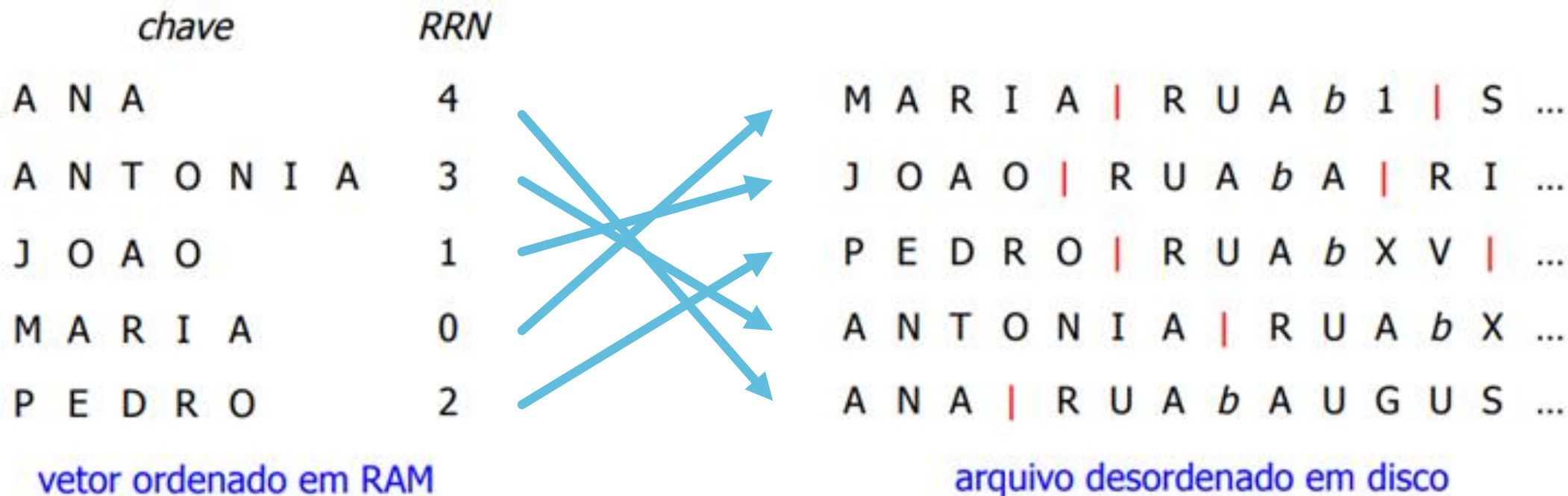
<i>chave</i>	<i>RRN</i>		
M A R I A	0	→	M A R I A   R U A b 1   S ...
J O A O	1	→	J O A O   R U A b A   R I ...
P E D R O	2	→	P E D R O   R U A b X V   ...
A N T O N I A	3	→	A N T O N I A   R U A b X ...
A N A	4	→	A N A   R U A b A U G U S ...

vetor em RAM

arquivo desordenado em disco



# Índice



# Índice

- Existe uma variedade de índices, cada qual com uma estrutura de dados particular
- Qualquer campo em um arquivo pode ser usado para criar um índice
  - Valor armazenado em um array de inteiros
  - Campo nome de uma struct
  - etc
- Vários índices podem ser definidos para um mesmo arquivo

# Índice Linear ou Simples

- Basicamente, mantém uma estrutura em memória primária (vetor) com os índices para os registros em memória secundária (arquivo sequencial)
- Chaves são ordenadas

Chave	Índice
Chave 1	10
Chave 2	1302
Chave 3	71



# Índice Linear ou Simples

- Funcionamento
  - Leitura completa do arquivo de dados
  - Para cada registro do vetor em RAM obtém o RRN (*Relative Record Number*)
  - Identifica o *byte offset* do registro em disco ( $\text{byte offset} = \text{RRN} * \text{tamRegistro}$ )

M A R I A | R U A b 1 | S ...  
J O A O | R U A b A | R I ...  
P E D R O | R U A b X V | ...  
A N T O N I A | R U A b X ...  
A N A | R U A b A U G U S ...

arquivo desordenado em disco

<i>chave</i>	<i>RRN</i>
M A R I A	0
J O A O	1
P E D R O	2
A N T O N I A	3
A N A	4

vetor em RAM

# Índice Linear ou Simples

- Funcionamento
  - Ordenação do vetor de chaves em RAM usando um método de ordenação tradicional
  - Busca agora pode ser feita pela chave usando busca binária
  - Maior eficiência, não é necessário ordenar o arquivo, apenas o índice

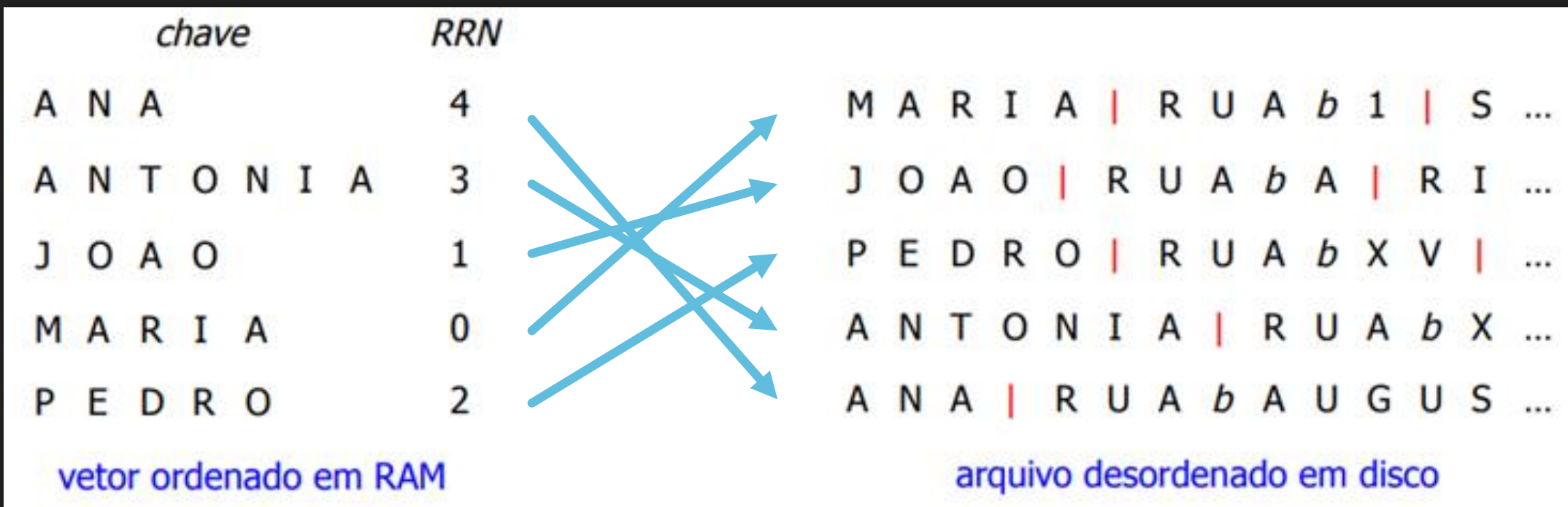
<i>chave</i>	<i>RRN</i>
M A R I A	0
J O A O	1
P E D R O	2
A N T O N I A	3
A N A	4

vetor desordenado em RAM

<i>chave</i>	<i>RRN</i>
A N A	4
A N T O N I A	3
J O A O	1
M A R I A	0
P E D R O	2

vetor ordenado em RAM

# Índice Linear ou Simples



# Índice Linear ou Simples

- Possui tamanho muito menor do que o arquivo de dados
- Adequado quando cabe em memória primária
  - Pode ser replicado em memória principal caso seja pequeno o suficiente
  - Caso contrário, pode ser mantido em um arquivo secundário

# Índice Linear ou Simples

- Armazenamento em memória secundária
  - Pode necessitar de vários acessos a disco, por causa da busca binária
  - Pode ter manutenção cara, devido à adição e remoção de registros
  - Requer o uso de outras organizações mais apropriadas, como árvore-B

# Índice Linear ou Simples

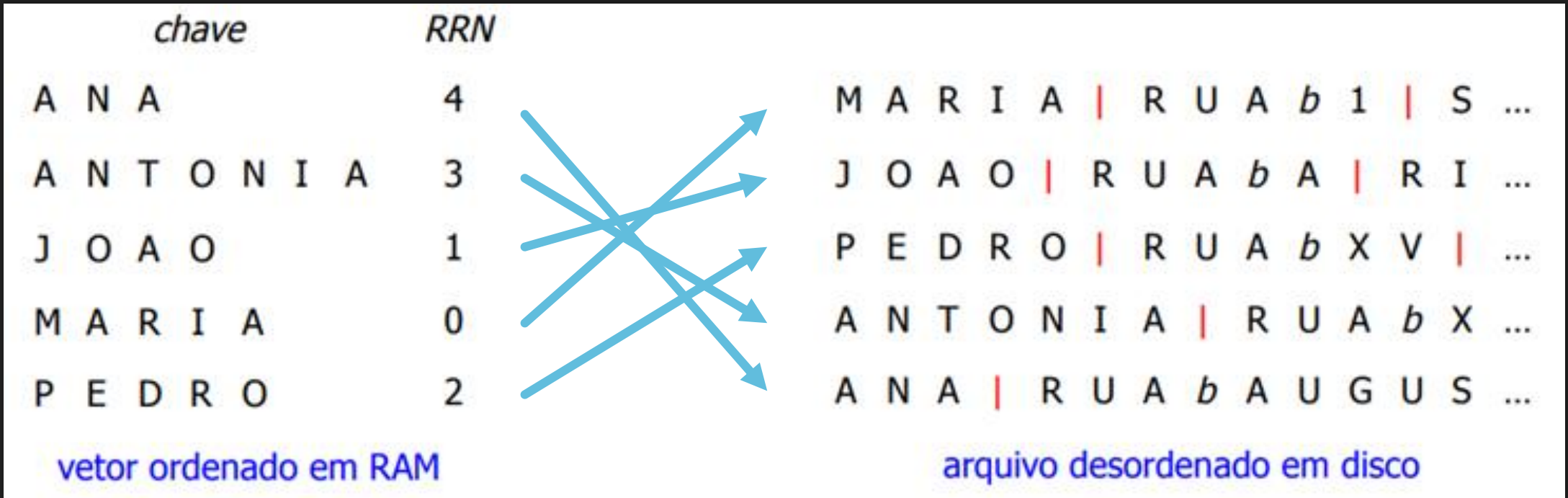
- Operações
  - Pesquisa
  - Criação
  - Inserção
  - Remoção
  - Alteração / Atualização
  - Destruição
- Operações aplicadas quando o índice pode ser armazenado totalmente em memória principal
  - Carregamento
  - Reescrita no disco



# Pesquisa

- Baseada na chave de busca
  - Encontra a posição da chave no índice
  - Obtém o endereço (RRN / Byte Offset) do registro correspondente à chave encontrada
  - Encontra o registro no arquivo de dados, posicionando para leitura no endereço
  - Lê os dados do registro no arquivo de dados

# Pesquisa



# Criação

- Índice Vazio
  - Cria o índice vazio, juntamente com a criação do arquivo de dados
  - Inclui apenas o registro de cabeçalho, indicando a chave e o índice

Chave	Índice

# Criação

- Índice Completo
  - Construção do índice completo a partir de arquivo de dados já existente
  - Registro cabeçalho + registros de dados (chave + RRN ou Byte offset), a partir de leitura sequencial do arquivo de dados
  - Demais registros (chave + índice) são obtidos a partir de uma varredura no arquivo de dados

Chave	Índice
Chave 1	10
Chave 2	1302
Chave 3	71

# Inserção

- Adiciona novos registros no índice devido às inserções no arquivo de dados

inserção de um novo  
registro no arquivo de  
dados



inserção de um novo  
registro no arquivo de  
índice

```
M A R I A | R U A b 1 | S ...
J O A O | R U A b A | R I ...
P E D R O | R U A b X V | ...
A N T O N I A | R U A b X ...
A N A | R U A b A U G U S ...
```

arquivo desordenado em disco

<i>chave</i>	<i>RRN</i>
A N A	4
A N T O N I A	3
J O A O	1
M A R I A	0
P E D R O	2

vetor ordenado em RAM

# Inserção

- Inserção em arquivo não ordenado
  - Realizada no final do arquivo ou com reaproveitamento de espaço
- Necessidade de reorganização do índice, devido à ordenação da chave
  - Deslocamento dos registros de índice
  - Alteração dos valores dos campos de referência no índice

```
M A R I A | R U A b 1 | S ...
J O A O | R U A b A | R I ...
P E D R O | R U A b X V | ...
A N T O N I A | R U A b X ...
A N A | R U A b A U G U S ...
```

arquivo desordenado em disco

<i>chave</i>	<i>RRN</i>
A N A	4
A N T O N I A	3
J O A O	1
M A R I A	0
P E D R O	2

vetor ordenado em RAM



# Remoção

- Remove registros no índice devido às remoções no arquivo de dados

remoção de um registro  
no arquivo de dados



remoção de um registro  
no arquivo de índice

M A R I A | R U A b 1 | S ...  
J O A O | R U A b A | R I ...  
P E D R O | R U A b X V | ...  
~~A N T O N I A | R U A b X ...~~  
A N A | R U A b A U G U S ...

arquivo desordenado em disco

chave	RRN
A N A	4
<del>A N T O N I A</del>	<del>3</del>
J O A O	1
M A R I A	0
P E D R O	2

vetor ordenado em RAM

# Remoção

- Remoção
  - Lógica: reaproveitamento de espaço
  - Física: deslocamento dos registros
- Preferência: lógica, ao invés de física
  - Registros marcados como removidos
  - Necessidade de reorganização periódica com recriação do índice

M A R I A | R U A b 1 | S ...  
J O A O | R U A b A | R I ...  
P E D R O | R U A b X V | ...  
~~A N T O N I A | R U A b X~~ ...  
A N A | R U A b A U G U S ...

arquivo desordenado em disco

chave	RRN
A N A	4
<del>A N T O N I A</del>	<del>3</del>
J O A O	1
M A R I A	0
P E D R O	2

vetor ordenado em RAM

# Alteração / Atualização

- Modifica registros no índice devido às modificações no arquivo de dados
- A alteração
  - Pode ser apenas no arquivo de dados
  - Pode necessitar de um ajuste no índice

# Alteração / Atualização

- Alteração no arquivo de dados
  - Modifica dados que não fazem parte da chave
  - Modifica o valor da chave
  - Pode alterar o endereço do registro
    - Caso mudar/aumentar o tamanho
    - Solução: remoção seguida de inserção, técnica mais utilizada

```
M A R I A | R U A b 1 | S ...
J O A O | R U A b A | R I ...
P E D R O | R U A b X V | ...
A N T O N I A | R U A b X ...
A N A | R U A b A U G U S ...
```

arquivo desordenado em disco

<i>chave</i>	<i>RRN</i>
A N A	4
A N T O N I A	3
J O A O	1
M A R I A	0
P E D R O	2

vetor ordenado em RAM

# Alteração / Atualização

- Modifica o índice devido às modificações no arquivo de dados
  - Mudou o valor do campo chave: reordenação do índice
  - Mudou o registro de lugar: ajuste do campo de referência

```
M A R I A | R U A b 1 | S ...
J O A O | R U A b A | R I ...
P E D R O | R U A b X V | ...
A N T O N I A | R U A b X ...
A N A | R U A b A U G U S ...
```

arquivo desordenado em disco

<i>chave</i>	<i>RRN</i>
A N A	4
A N T O N I A	3
J O A O	1
M A R I A	0
P E D R O	2

vetor ordenado em RAM

# Carregamento

- Carrega o arquivo de índice na memória principal antes de usá-lo
  - Utilizado apenas quando o índice pode ser armazenado totalmente em memória principal
- Passo a passo
  - Aponta para o primeiro registro do arquivo de índice em disco
  - Varre o arquivo de índices sequencialmente
  - Cria o índice em memória principal, em geral implementado como um array



# Reescrita

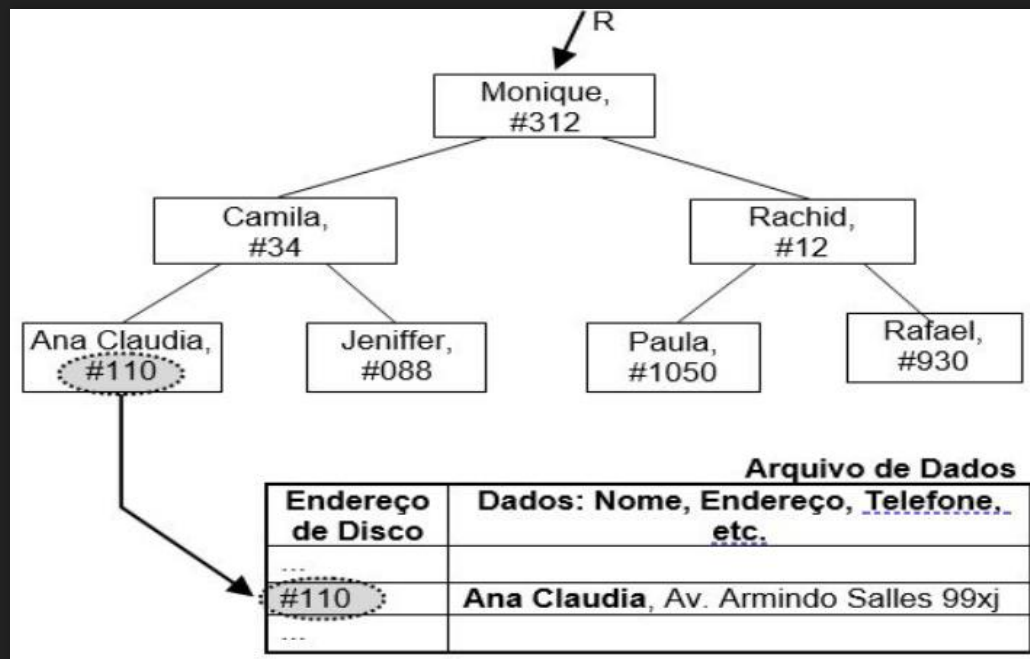
- Atualiza o arquivo de índice em disco com base no arquivo de índice em memória principal, quando necessário
- Informação adicional
  - status no registro de cabeçalho: verdadeiro/falso
  - Inconsistência nos índices, devido à queda de energia, travamento do programa de atualização, etc

# Índices não-lineares

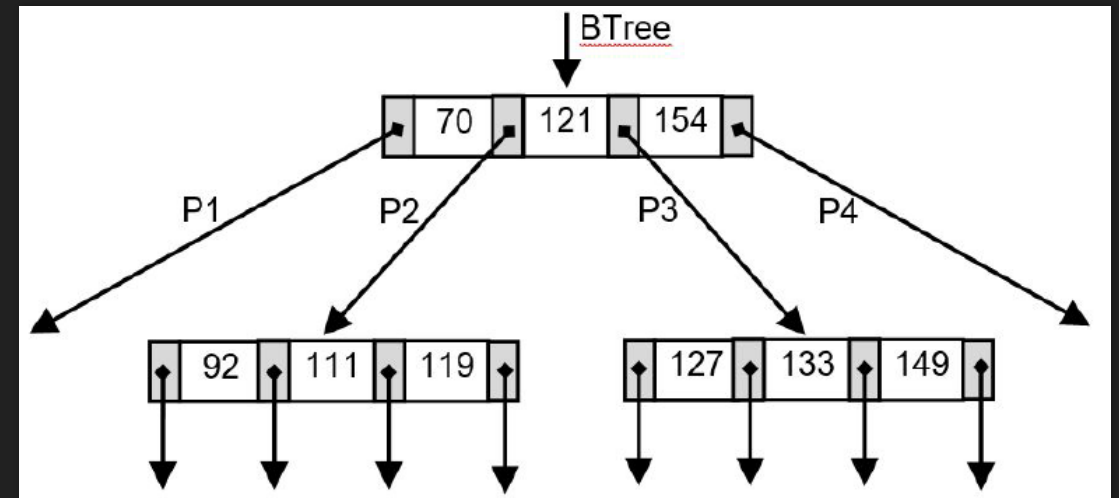
- Utiliza uma estrutura hierárquica não-linear para indexar os objetos modelados
  - Árvore Binária de Busca Balanceada
  - Árvore B

# Índices não-lineares

Árvore Binária de Busca Balanceada



Árvore B



# Propriedade de índices

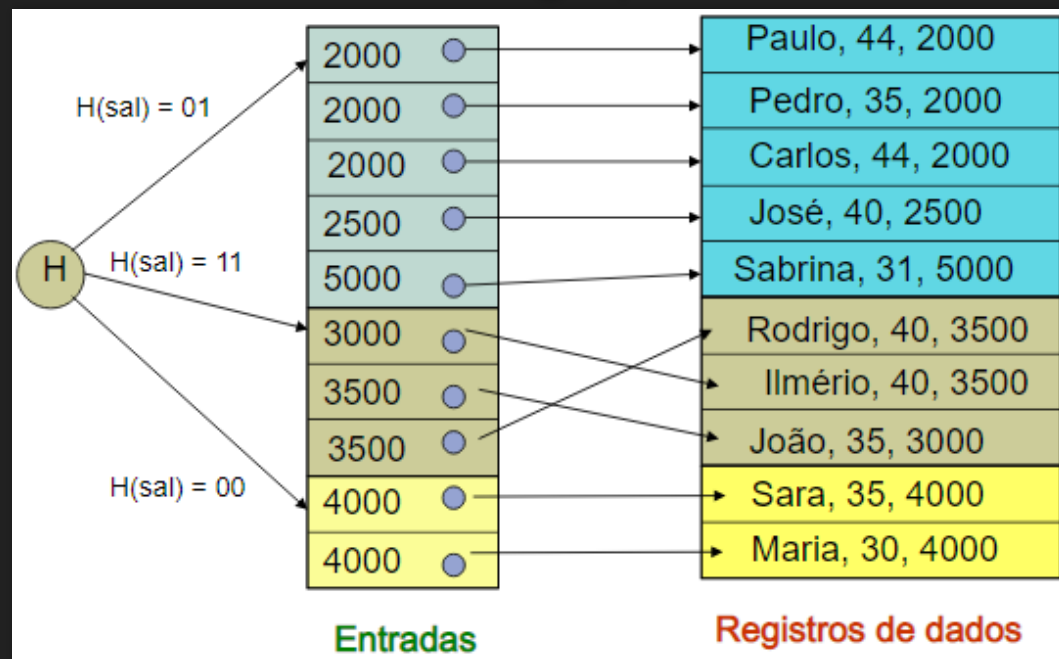
- Agrupado x Não Agrupado
- Denso x Esparso
- Primário x Secundário

# Propriedade de índices

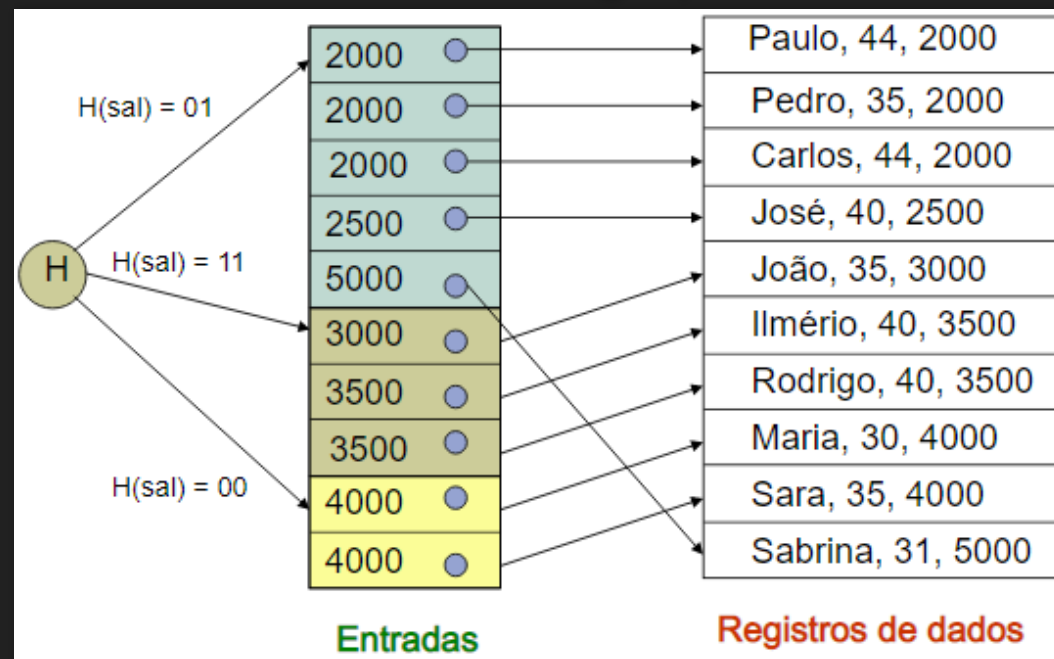
- Agrupado
  - A ordem dos registros é compatível com a ordem das entradas no arquivo de índice
  - Devem estar em ordem sequencial ou classificada
  - Não deve haver um valor chave, o que significa que pode ter valores repetidos
- Não agrupado
  - Não mantém a ordem original do arquivo de dados
  - O índice contém o ponteiro para os dados

# Propriedade de índices

Índice agrupado



Índice não agrupado





# Propriedade de índices

- Índice agrupado
- Desvantagem
  - Grande *overhead* para mover registros a fim de preservar a ordem depois de inserções e remoções
- Vantagem
  - Seleções do tipo = ou <> são altamente otimizadas caso os registros sejam ordenados de acordo com a chave do índice

# Propriedade de índices

## ○ Índice Denso

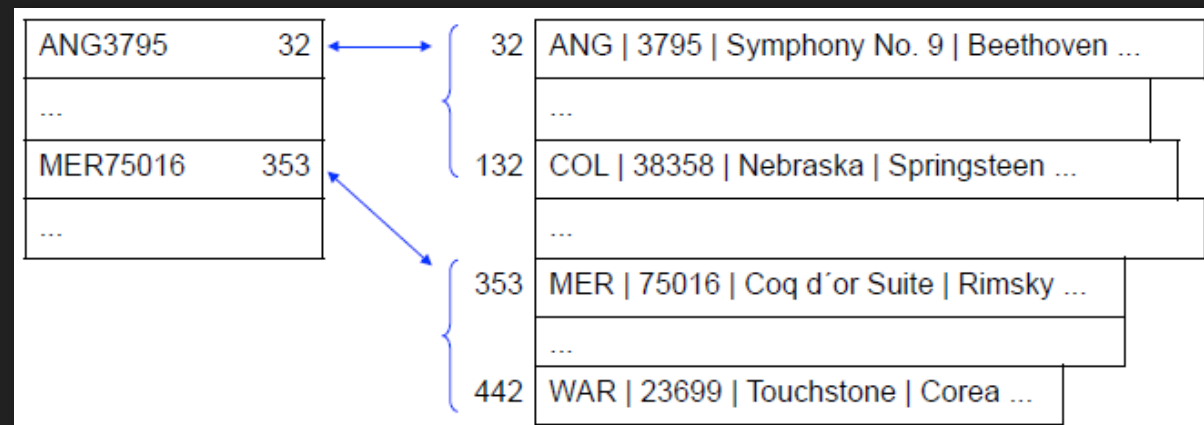
- Possui uma entrada no índice para cada valor de chave (i.e., cada registro) no arquivo de dados
- Cada registro no arquivo de dados implica em um registro no índice

ANG3795	32	↔	32	ANG   3795   Symphony No. 9   Beethoven ...
COL31809	77	↔	77	COL   31809   Symphony No. 9   Dvorak ...
COL38358	132	↔	132	COL   38358   Nebraska   Springsteen ...
...				...
MER75016	353	↔	353	MER   75016   Coq d'or Suite   Rimsky ...
RCA2626	396	↔	396	RCA   2626   Quartet in C Sharp Minor ...
WAR23699	442	↔	442	WAR   23699   Touchstone   Corea ...

# Propriedade de índices

## ○ Índice Esperso

- Possui uma entrada no índice para cada bloco do arquivo de dados
- Deve ser sempre agrupado
- Indexa um registro âncora, que pode ser o primeiro ou o último
- O índice contém um número menor de entradas em comparação com o arquivo de dados



# Propriedade de índices

- Índice primário
  - Construído sobre a chave primária
  - Definido sobre um campo sem repetição
- Índice secundário
  - Construído sobre uma chave secundária
  - Definido sobre um campo com repetição

# Índice secundário

- Fracamente ligado (*Loosely Binding*)
  - Relaciona uma chave secundária à chave primária

Beethoven	ANG3795	ANG3795	167	32	LON   2312   Rom ...
Beethoven	DG139201	COL31809	353	77	RCA   2626   Quar ...
Beethoven	DG18807	COL38358	211	132	WAR   23699   Tou ...
Beethoven	RCA2626	DG139201	396	167	ANG   3795   Sym ...
Corea	WAR23699	DG18807	256	211	COL   38358   Nebr ...
Dvorak	COL31809	FF245	442	256	DG   18807   Sym ...
Prokofiev	LON2312	LON2312	32	300	MER   75016   Coq ...
Rimsky	MER75016	MER75016	300	353	COL   31809   Sym ...
Springsteen	COL38358	RCA2626	77	396	DG   139201   Violin ...
Sweet Honey In The	FF245	WAR23699	132	442	FF   245   Good ...

arquivo de índice secundário

arquivo de índice primário

arquivo de dados

# Índice secundário fracamente ligado

- Operação de Busca
  - Pesquisar o índice de chave secundária para encontrar a chave primária relacionada
    - Pode ser mais de uma chave primária
  - 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 desejado

# Índice secundário fracamente ligado

- Operação de Inserção
  - Inserir o novo registro no arquivo de dados
  - Inserir a chave e o endereço no índice primário, reordenando, caso necessário
  - Inserir a entrada correspondente em cada índice secundário (podem existir vários)
  - Chaves duplicadas devem ser mantidas agrupadas e ordenadas

# Índice secundário fracamente ligado

- Operação de Remoção
  - Remover o registro do arquivo de dados
  - Remover a entrada correspondente no arquivo de índice primário, reordenando, caso necessário
  - Remover a entrada correspondente de cada índice secundário (podem existir vários)



# Índice secundário fracamente ligado

- Opções de remoção
  - *delete all references*
  - *delete some references*

	<i>delete all references</i>	<i>delete some references</i>
Vantagens	<ul style="list-style-type: none"><li>- sem queda de desempenho na busca por registros removidos</li><li>- índices permanecem do tamanho necessário</li></ul>	<ul style="list-style-type: none"><li>- sem necessidade de reorganização a cada remoção</li><li>- economia de tempo nas remoções</li></ul>
Desvantagens	<ul style="list-style-type: none"><li>- necessidade de reorganização a cada remoção</li><li>- processo altamente custoso, devido à ordenação</li></ul>	<ul style="list-style-type: none"><li>- com queda de desempenho na busca na busca por registros removidos</li><li>- crescimento do tamanho dos índices e necessidade de reorganização periódica</li></ul>

# Índice secundário fracamente ligado

- Operação de Alteração / Atualização
- Alteração do valor da chave secundária
  - Reordenação do índice secundário
- Alteração do valor da chave primária
  - Reordenação do índice primário
  - Atualização dos índices secundários
  - Reordenação dos índices secundários se houver repetição da chave secundária
- Alteração dos demais campos
  - Não afeta nenhum dos índices

# Índice secundário

- Índice secundário fortemente ligado (*Tight Binding*)
  - Relaciona uma chave secundária diretamente ao registro

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

*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*

# Índice secundário - comparação

	Fracamente ligado	Fortemente ligado
Vantagens	<ul style="list-style-type: none"><li>- diminui custo de remoções na abordagem <i>delete some references</i></li><li>- modificação no arquivo de dados afeta apenas o índice primário</li><li>- menor complexidade de codificação</li></ul>	<ul style="list-style-type: none"><li>- acesso direto</li><li>- índice primário ► arquivo de dados</li><li>- índice secundário ► arquivo de dados</li><li>- melhor desempenho na busca</li></ul>
Desvantagens	<ul style="list-style-type: none"><li>- acesso indireto</li><li>- índice secundário ► índice primário ► arquivo de dados</li><li>- queda do desempenho na busca</li></ul>	<ul style="list-style-type: none"><li>- alto custo para modificações</li><li>- modificação no arquivo de dados afeta todos os índices secundários</li><li>- maior complexidade de codificação</li></ul>

# Repetição de chaves secundárias

- Problemas
  - Necessidade de armazenar a mesma chave secundária várias vezes
  - Necessidade de reordenar os índices sempre que um novo registro é inserido no arquivo
    - Mesmo que esse registro tenha um valor de chave secundária já existente no arquivo

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

*arquivo de índice secundário*

# Repetição de chaves secundárias

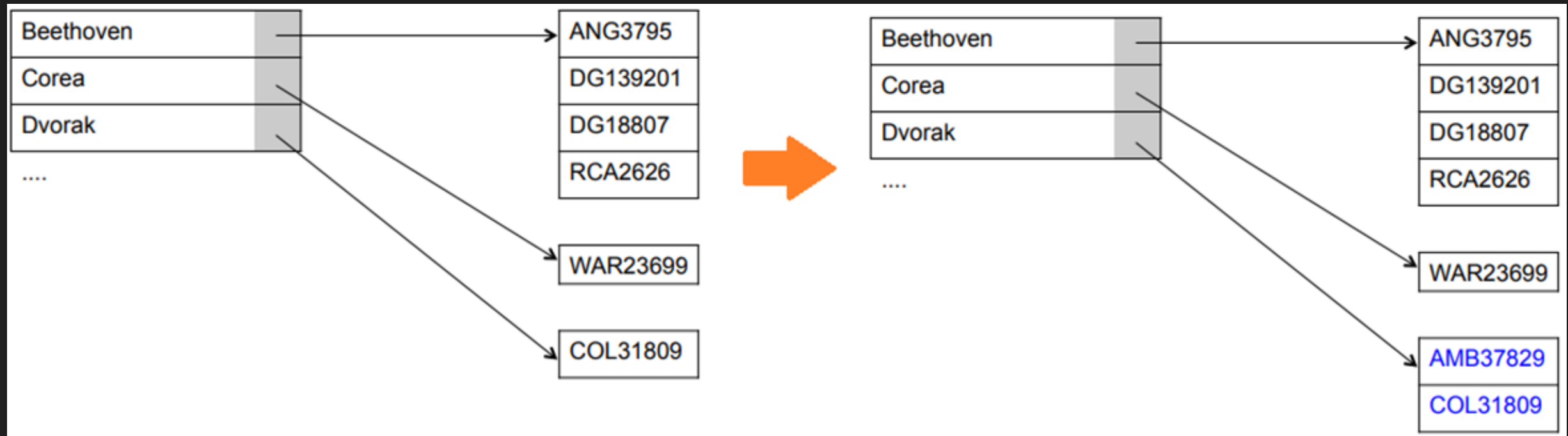
- Solução 1: Vetores de Tamanho Fixo
  - Associa um vetor de tamanho fixo a cada chave secundária
- Vantagem
  - Não é necessário reordenar o índice secundário a cada inserção de chave secundária repetida
- Desvantagens
  - Limitado a um número fixo de repetições
  - Grande ocorrência de fragmentação interna no índice
  - Pode não compensar a eliminação da duplicação de chaves

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

# Repetição de chaves secundárias

- Solução 2: Listas Invertidas
  - Associa uma lista encadeada das chaves primárias a cada chave secundária
  - Inserção de um novo registro equivale a inserção de um novo nó na lista

# Listas invertidas





# Listas invertidas: vantagens

- Índice secundário
  - Alterado quando insere-se um registro com chave inexistente, ou quando altera-se chave já existente
- Remoção, inserção ou alteração de registros já existentes
  - Alteração apenas no arquivo da lista invertida
  - Modificação do campo de referência do índice se necessário
- Ordenação do arquivo de índice secundário
  - Mais rápida: menos registros e registros menores
- Registros de tamanho fixo
  - Facilita a adoção de um mecanismo para reaproveitamento de espaço

# Listas invertidas: desvantagens

- Chaves primárias associadas a uma certa chave secundária não estão adjacentes fisicamente no disco
- Pode ser necessário realizar vários seeks para recuperar a lista
- O ideal é manter o índice e a lista na memória primária