

# Estruturas de Indexação de Dados

Cristina Dutra de Aguiar Ciferri

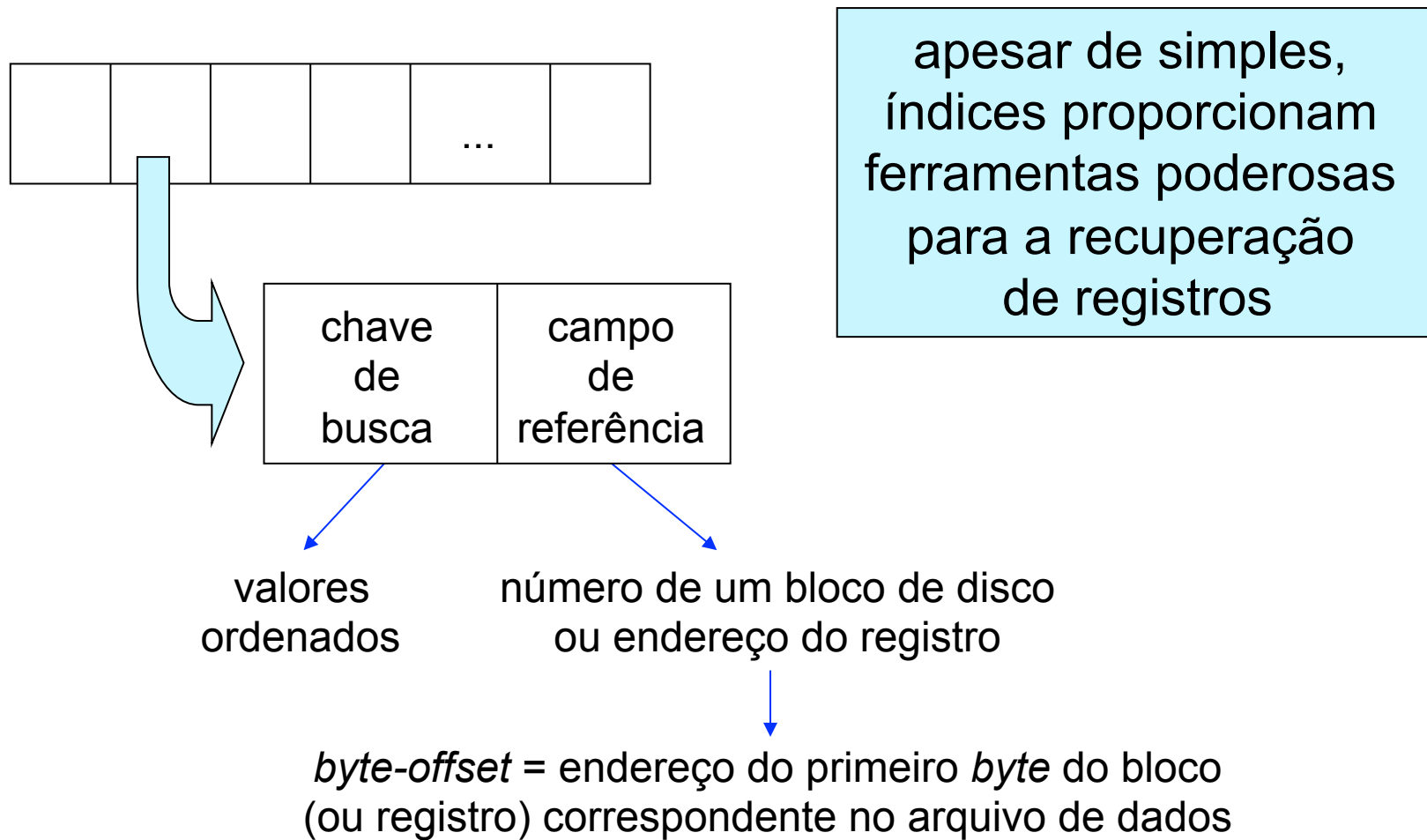
# Índice

- Estrutura de acesso auxiliar usada para melhorar o desempenho na recuperação de registros
- Pesquisa
  - restringida a um subconjunto dos registros, em contrapartida à análise do conjunto completo
  - realizada em resposta a certas condições

# Índice

- Observações
  - 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
  - vários índices podem ser definidos para um mesmo arquivo

# Índice Simples ou Linear



# Índice Simples ou Linear

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

*índice*

arquivo auxiliar em disco

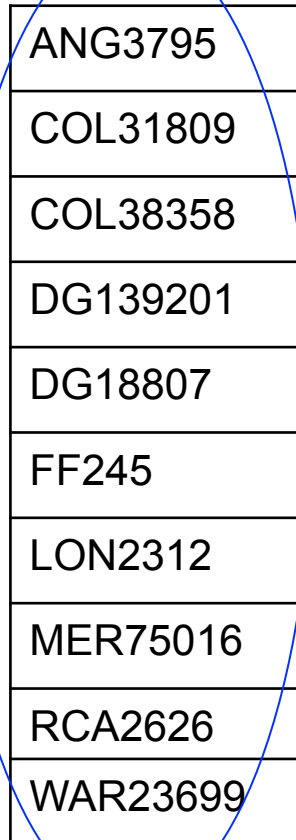
32	LON   2312   Romeo and Juliet   Prokofiev ...
77	RCA   2626   Quartet in C Sharp Minor ...
132	WAR   23699   Touchstone   Corea ...
167	ANG   3795   Symphony No. 9   Beethoven ...
211	COL   38358   Nebraska   Springsteen ...
256	DG   18807   Symphony No. 9   Beethoven ...
300	MER   75016   Coq d'or 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  
primária

*arquivo de dados*

arquivo armazenado em disco

# Índice Simples ou Linear



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

*índice*  
valores ordenados

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*  
geralmente registros desordenados

# Índice Simples ou Linear

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

*índice*

campos e

registros de tamanho fixo

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*

campos e

registros de tamanho fixo ou variável

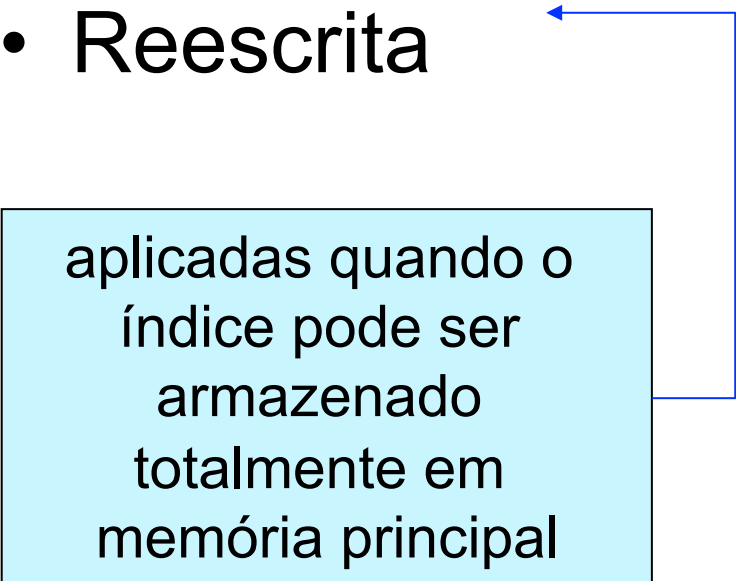
# Considerações

- Índice
  - possui tamanho muito menor do que o tamanho do arquivo de dados
  - possibilita a busca binária de seus registros
  - pode ser replicado em memória principal caso seja pequeno o suficiente
- A existência de índices não afeta a localização física dos registros dos arquivos de dados



# Operações em um Índice

- Pesquisa
- Criação
- Inserção
- Remoção
- Atualização
- Destruição
- Carregamento
- Reescrita



aplicadas quando o índice pode ser armazenado totalmente em memória principal

# Pesquisa

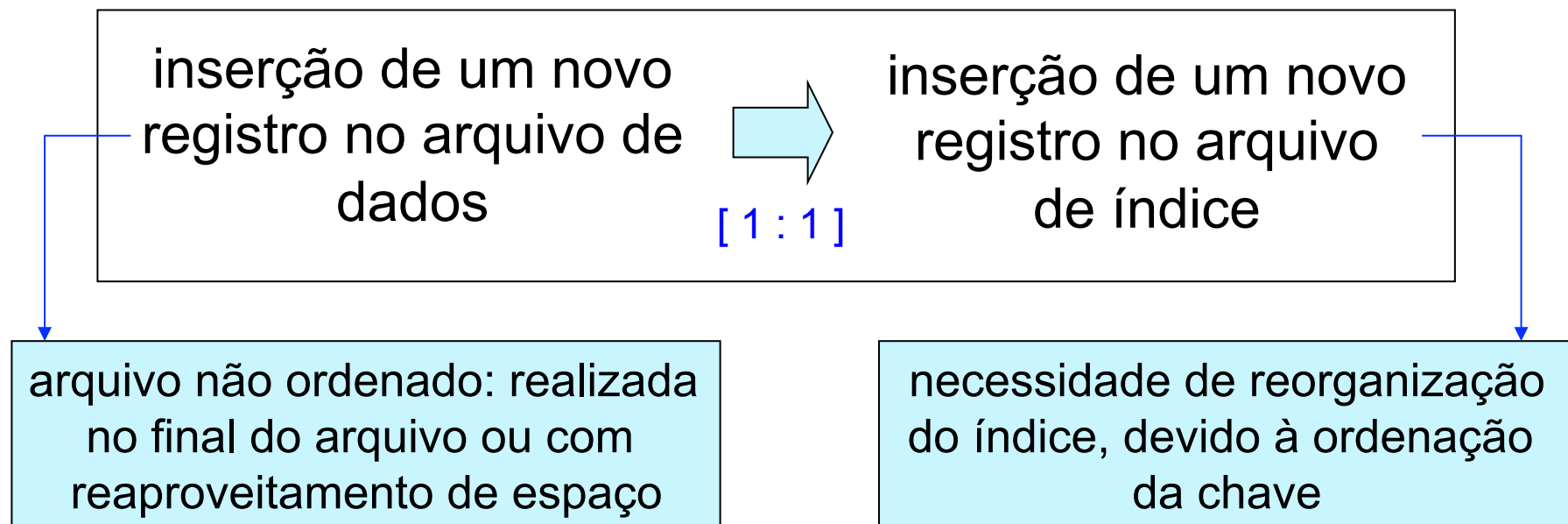
- Baseada na chave de busca
  - encontra a posição da chave no índice
  - obtém o *byte-offset* no registro correspondente à posição encontrada
  - move para o registro no arquivo de dados
  - recupera o registro solicitado do arquivo de dados

# Criação

- Cria o índice juntamente com a criação do arquivo de dados
  - apenas registro de cabeçalho
- Cria o índice baseado em um arquivo de dados já existente
  - registro de cabeçalho
  - demais registros (chave de busca + campo de referência), obtidos a partir de uma varredura no arquivo de dados

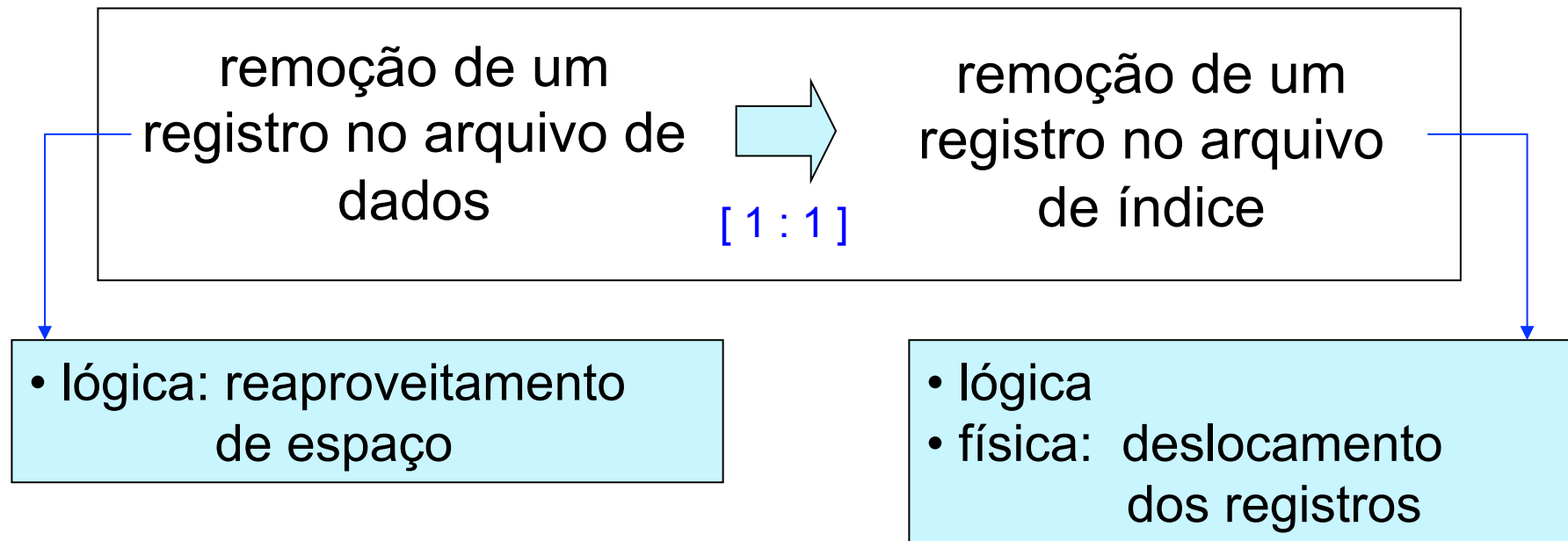
# Inserção

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



# Remoção

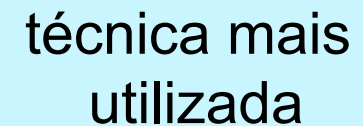
- Remove registros no índice devido às remoções no arquivo de dados



# Atualização

- Modifica registros no índice devido às modificações no arquivo de dados
- Tratamento
  - remoção seguida de inserção
  - ou*
  - campo chave: reordenação do índice
  - campo não chave: ajuste do campo de referência se o registro mudar fisicamente no arquivo de dados

técnica mais utilizada




# Carregamento

- Carrega o arquivo de índice na memória principal antes de usá-lo
- Passos
  - 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 vetor

# 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

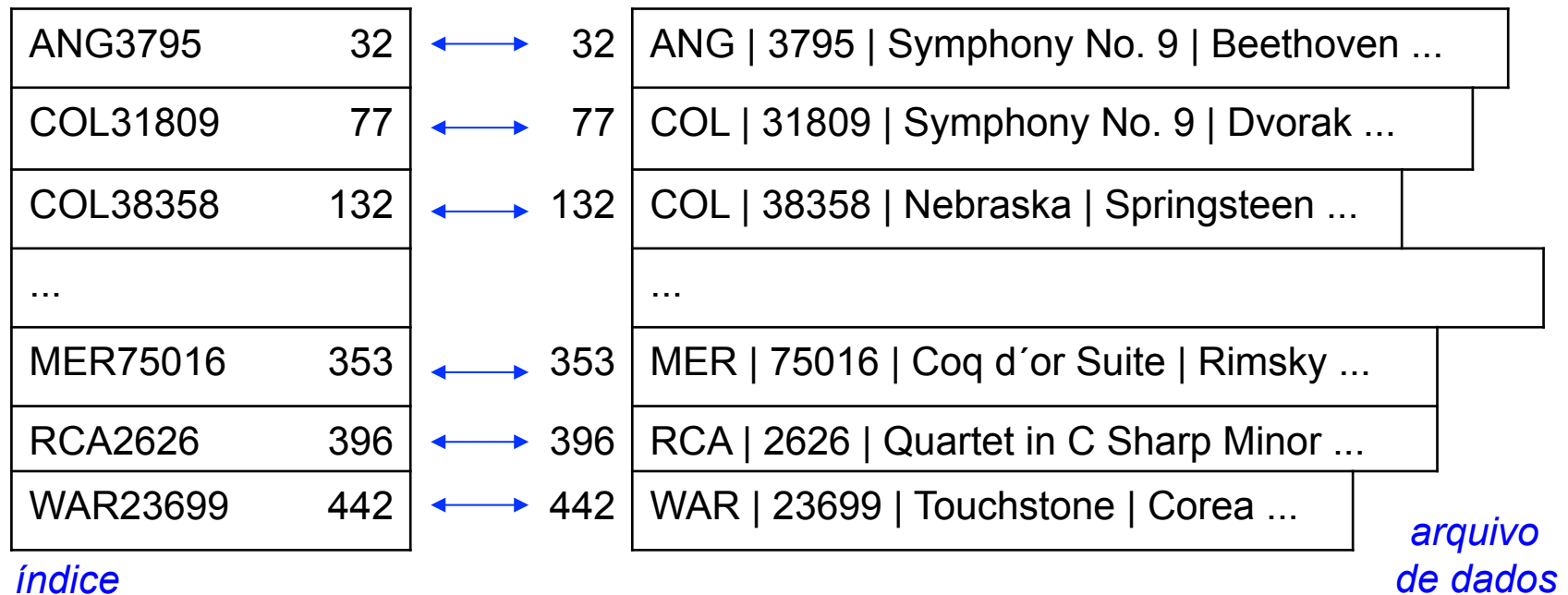


inconsistência nos índices,  
devido à queda de energia,  
travamento do programa de  
atualização, etc.



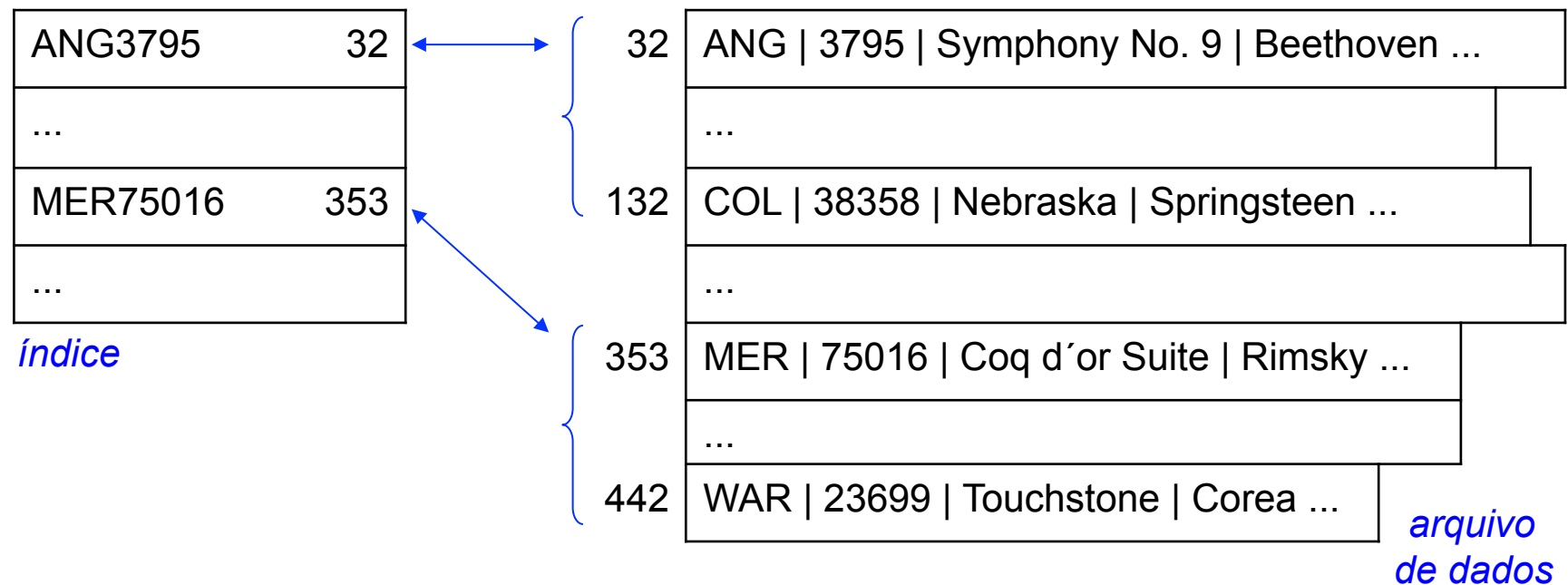
# Índice Denso

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



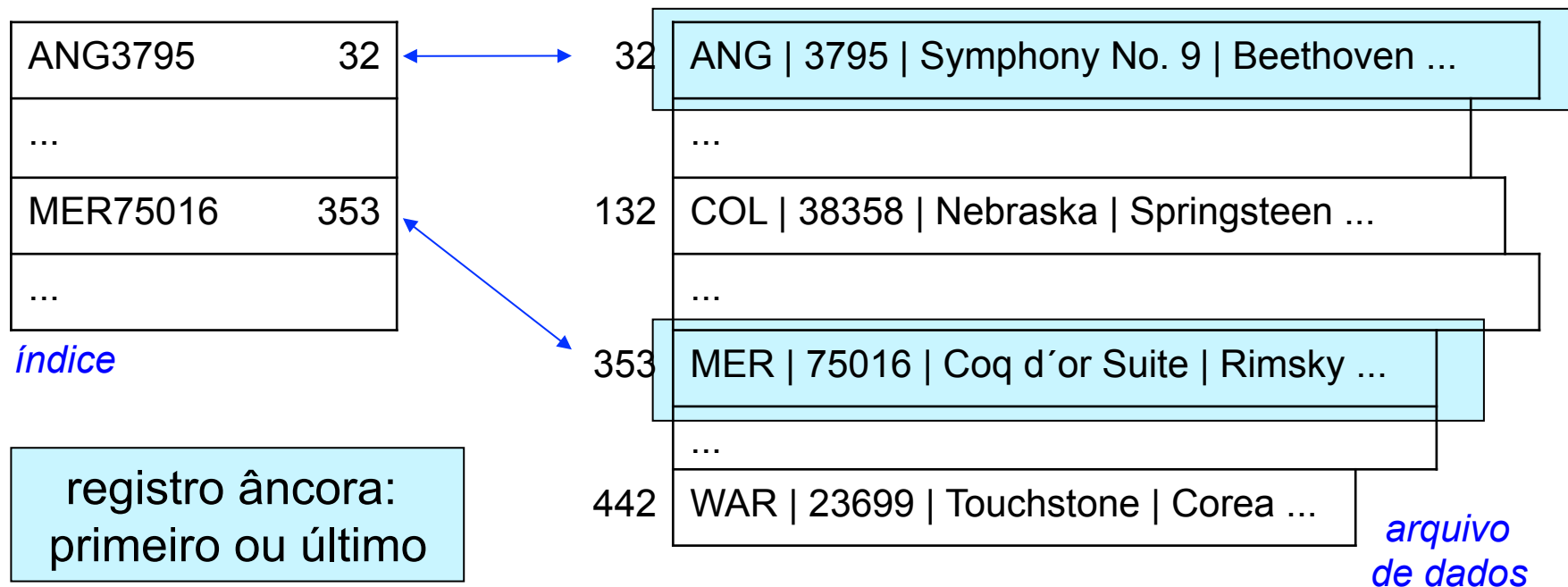
# Índice Esperso

- Possui uma entrada no índice para cada bloco do arquivo de dados



# Índice Esperso

- Possui uma entrada no índice para cada bloco do arquivo de dados

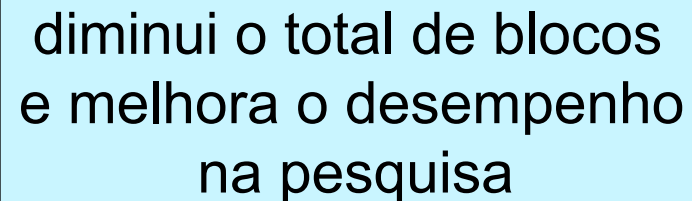


# Tipos de Índice Ordenados

Índice	Arquivo de Dados	Campo Indexado	Registros Recuperados
primário	ordenado	chave primária	0 ou 1
cluster	ordenado	atributo não chave	0 ou vários
secundário	desordenado	chave primária	0 ou 1
		atributo não chave	0 ou vários

# Índice Primário

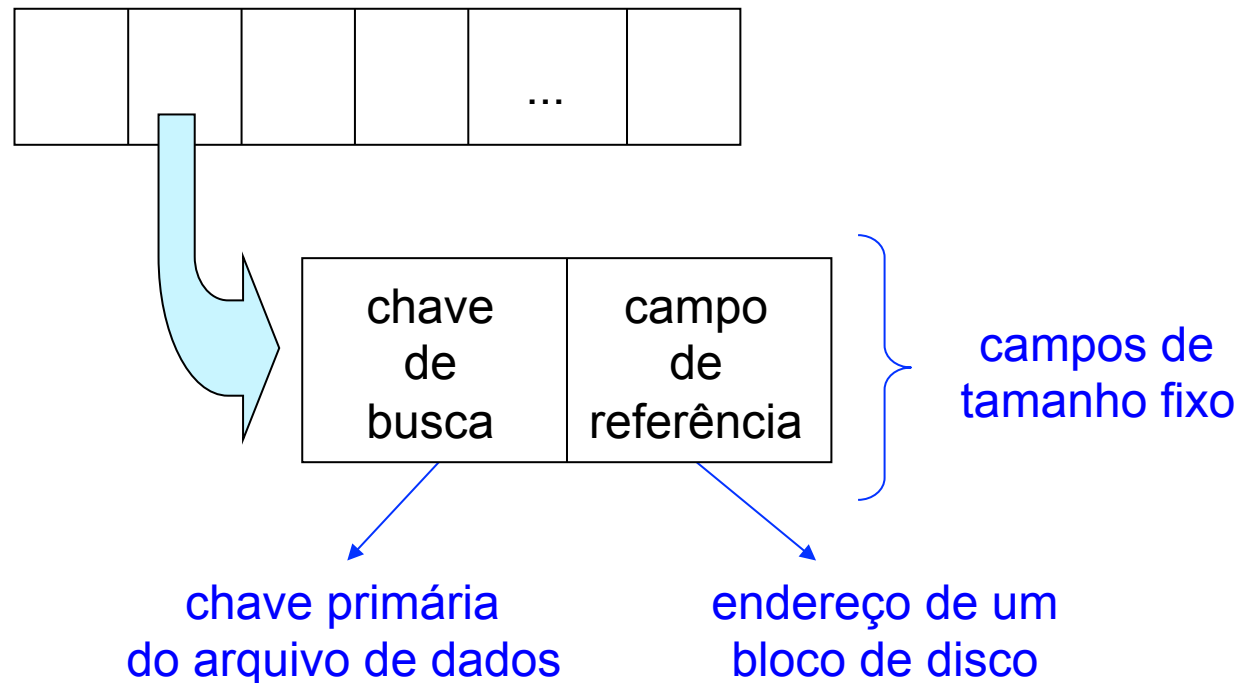
- Características
  - ordenado
  - definido com base em um arquivo de dados ordenado pela chave primária
  - possui um único nível
  - esperso
    - total de entradas no índice = número de blocos do arquivo de dados



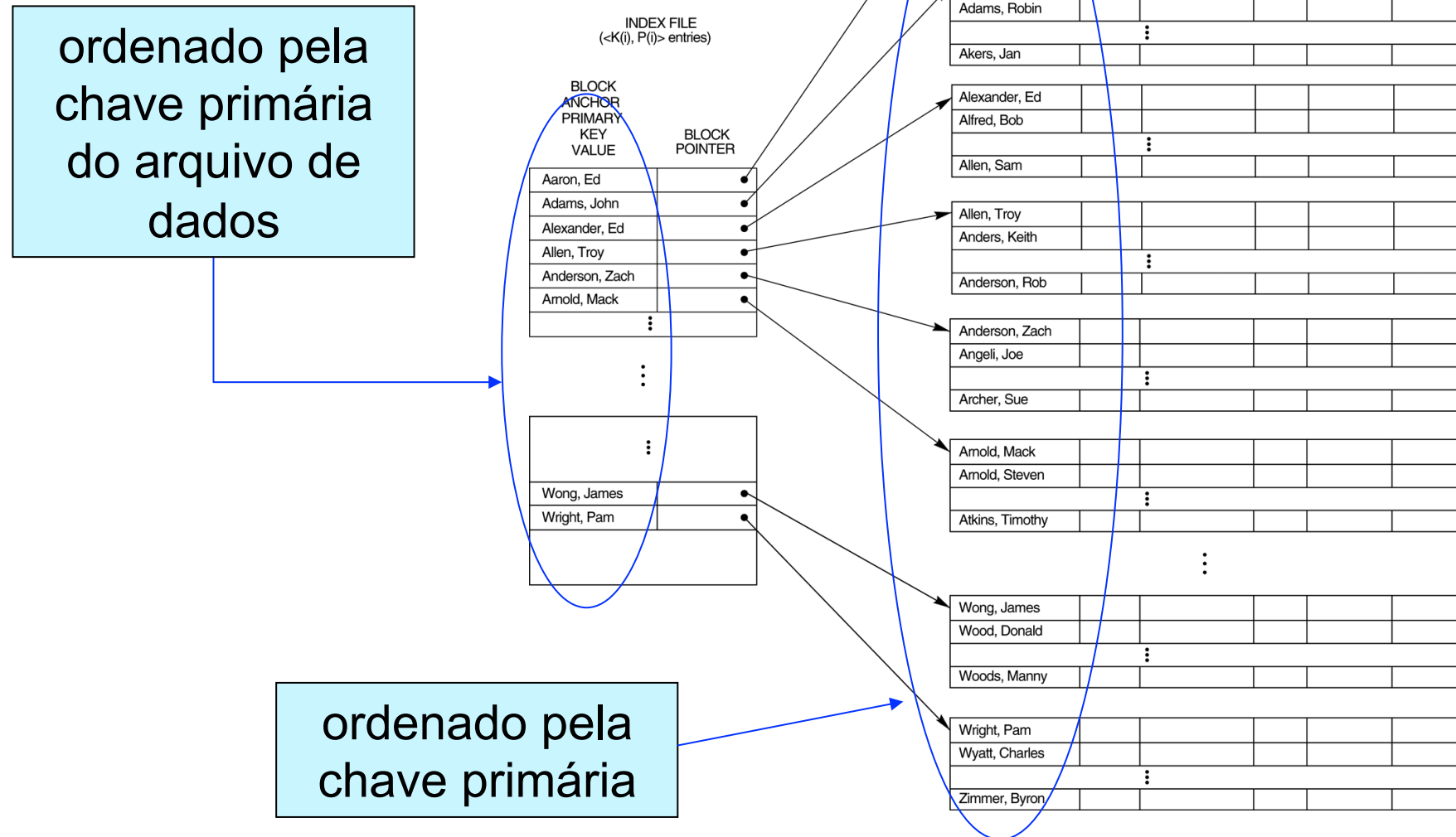
diminui o total de blocos  
e melhora o desempenho  
na pesquisa

# Índice Primário

- Estrutura do registro (entrada)



# Índice Primário

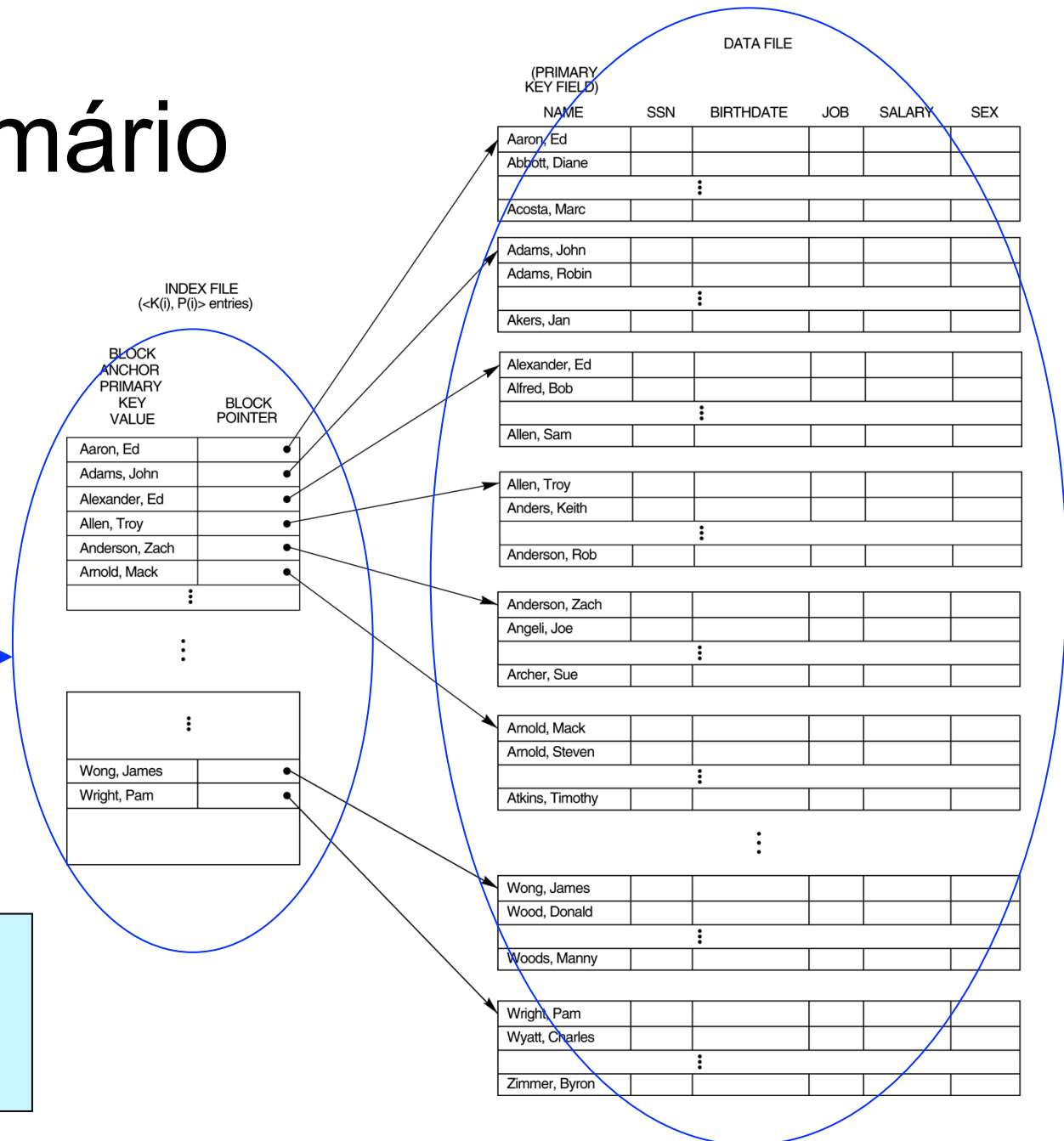


# Índice Primário

menor número de blocos

- menos entradas
- registros menores

pesquisa binária  
mais eficiente no índice





# Arquivo de Dados

- Número de registros ( $r$ ) = 30.000
- Tamanho do bloco ( $B$ ) = 1.024 *bytes*
- Tamanho dos registros ( $R$ ) = 100 *bytes*
- Fator de bloco de disco ( $bfr$ ) =  $\lfloor B/R \rfloor = 10$
- Número de blocos ( $b$ ) =  $\lceil r/bfr \rceil = 3.000$

# Arquivo de Índice

- Número de registros ( $r$ ) = 3.000
  - número de blocos do arquivo de dados
- Tamanho do bloco ( $B$ ) = 1.024 *bytes*
- Tamanho dos registros ( $R$ ) = 15 *bytes*
  - chave = 9 *bytes*
  - endereço = 6 *bytes*
- Fator de bloco de disco ( $bfr$ ) =  $\lfloor B/R \rfloor = 68$
- Número de blocos ( $b$ ) =  $\lceil r/bfr \rceil = 45$

# Acessos a Disco

- Sem o uso do índice
  - busca binária no arquivo de dados

$$\lceil \log_2 3000 \rceil = 12$$

- Com o uso do índice
  - busca binária no arquivo de índice +
  - leitura do registro no arquivo de dados

$$\lceil \log_2 45 \rceil + 1 = 6 + 1 = 7$$

# Inserção

- Problemas
  - deslocamento dos registros nos arquivos de dados e de índice para ordenação
  - alteração dos valores dos campos de referência no índice (registros âncoras)
- Soluções
  - arquivo *overflow* desordenado
  - lista ligada de registros *overflow* para cada bloco no arquivo de dados

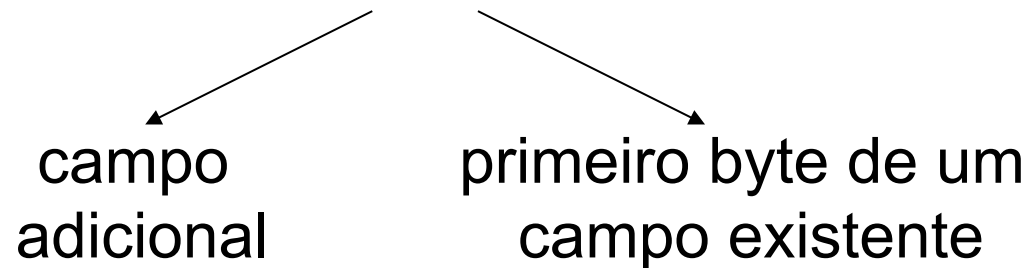
reorganização  
periódica com  
recriação  
do índice

# Remoção

- Características

- lógica, ao invés de física

- registros marcados como removidos



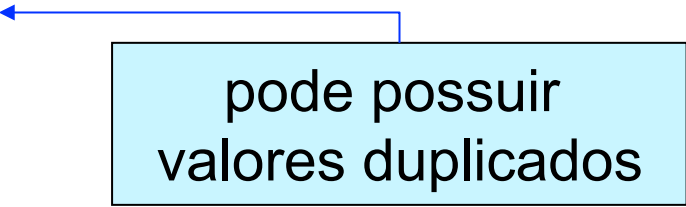
- reorganização periódica

- recriação do índice

# Índice de Agrupamento (Cluster)

- Características

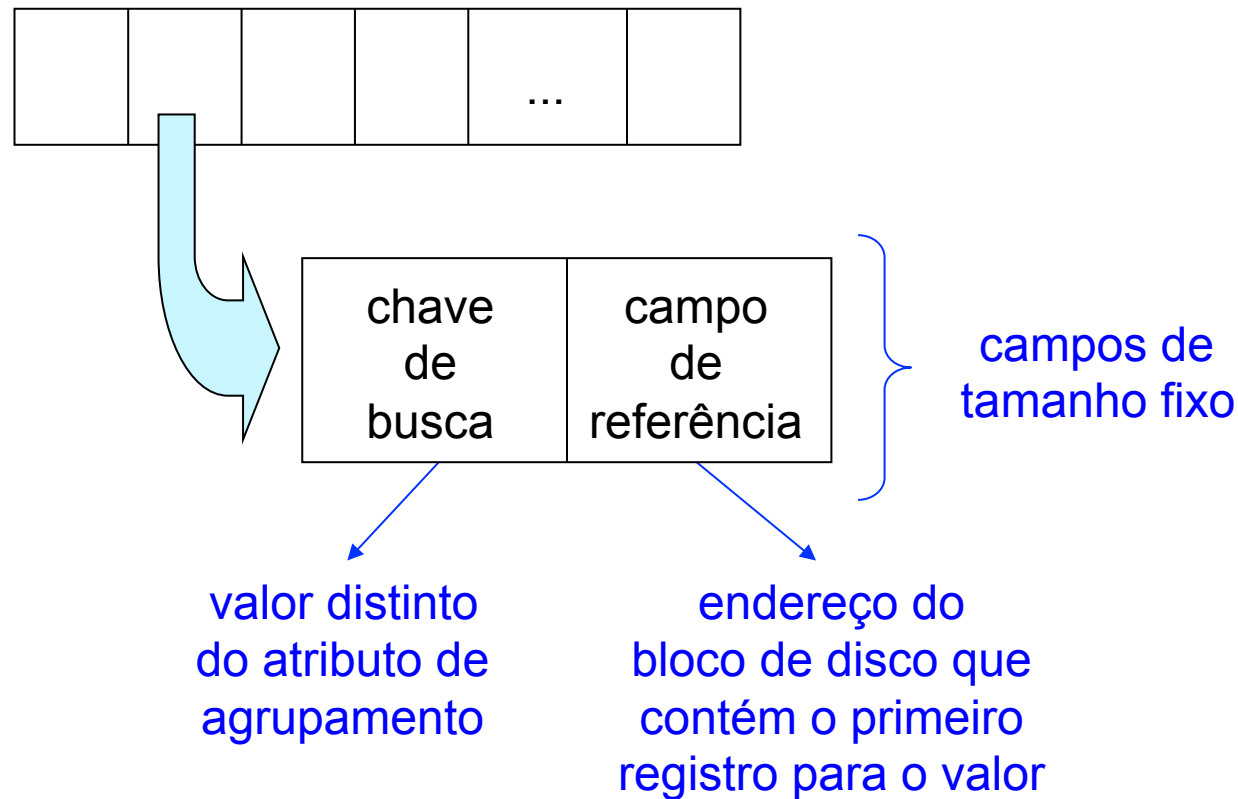
- ordenado
- definido com base em um arquivo de dados ordenado por um atributo não chave (atributo de agrupamento)
- possui um único nível
- esperso
  - total de entradas no índice = número de valores distintos do atributo de agrupamento



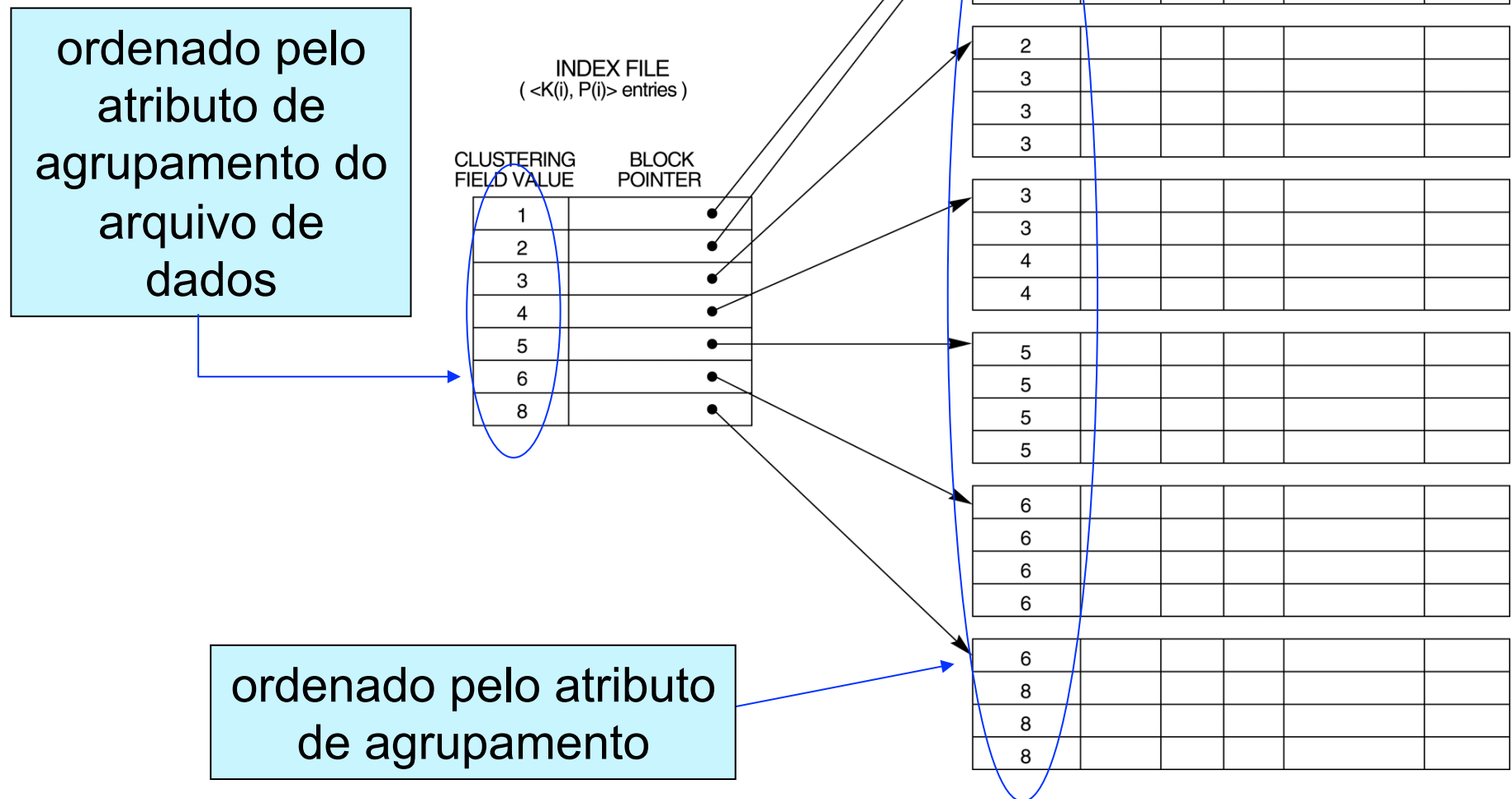
pode possuir  
valores duplicados

# Índice de Agrupamento

- Estrutura do registro (entrada)



# Índice de Agrupamento





# Índice de Agrupamento

menor número de  
blocos

- menos entradas
- registros menores

pesquisa binária  
mais eficiente no  
índice



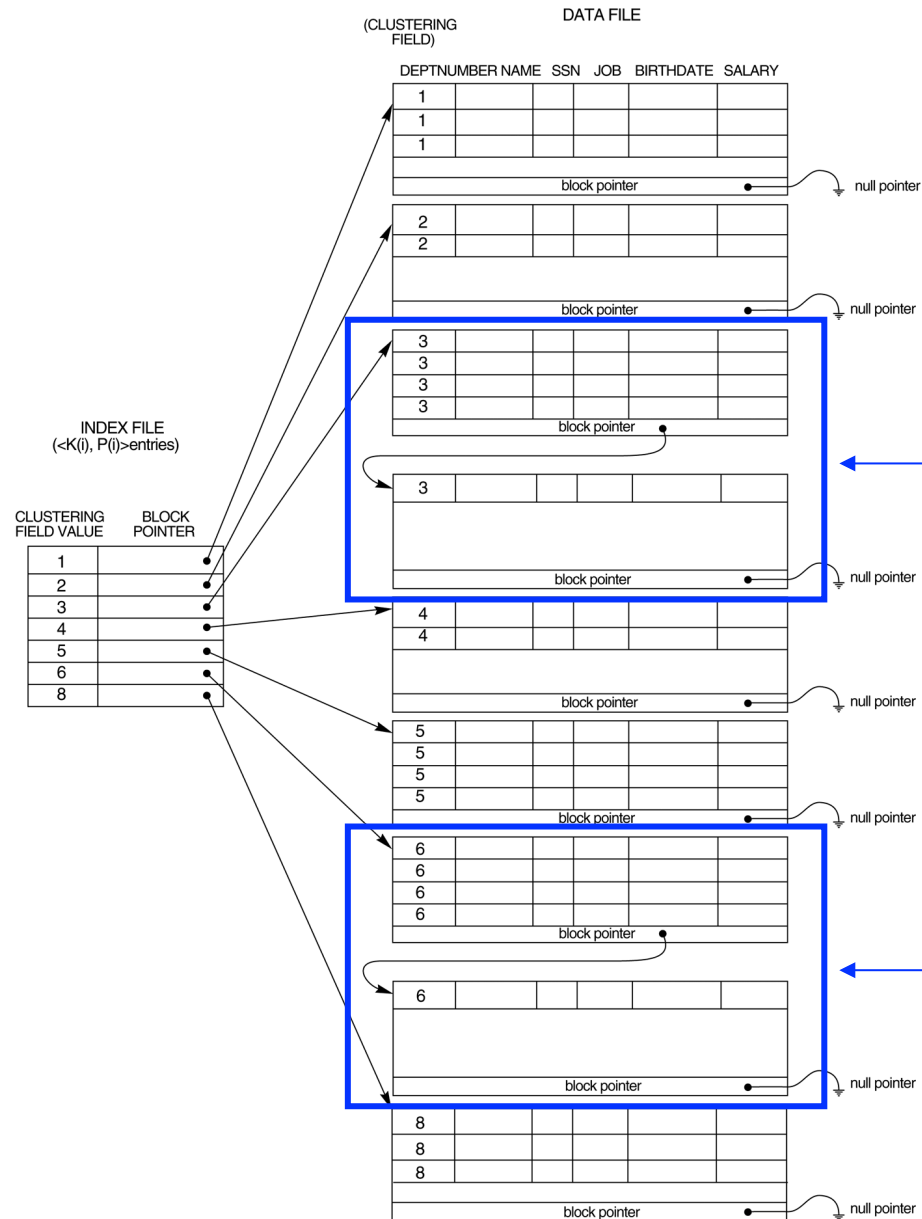
CLUSTERING FIELD VALUE	BLOCK POINTER
1	
2	
3	
4	
5	
6	
8	

(CLUSTERING FIELD)		DEPTNUMBER	NAME	SSN	JOB	BIRTHDATE	SALARY
1							
1							
1							
2							
2							
3							
3							
3							
3							
5							
5							
5							
5							
6							
6							
6							
6							
8							
8							
8							
8							

# Inserção e Remoção

- Problemas
  - deslocamento dos registros nos arquivos de dados e de índice para ordenação
  - alteração dos valores dos campos de referência no índice
- Solução
  - reservar um bloco no arquivo de dados para cada valor distinto do atributo de agrupamento
  - adicionar um campo de encadeamento nesses blocos

# Índice de Agrupamento



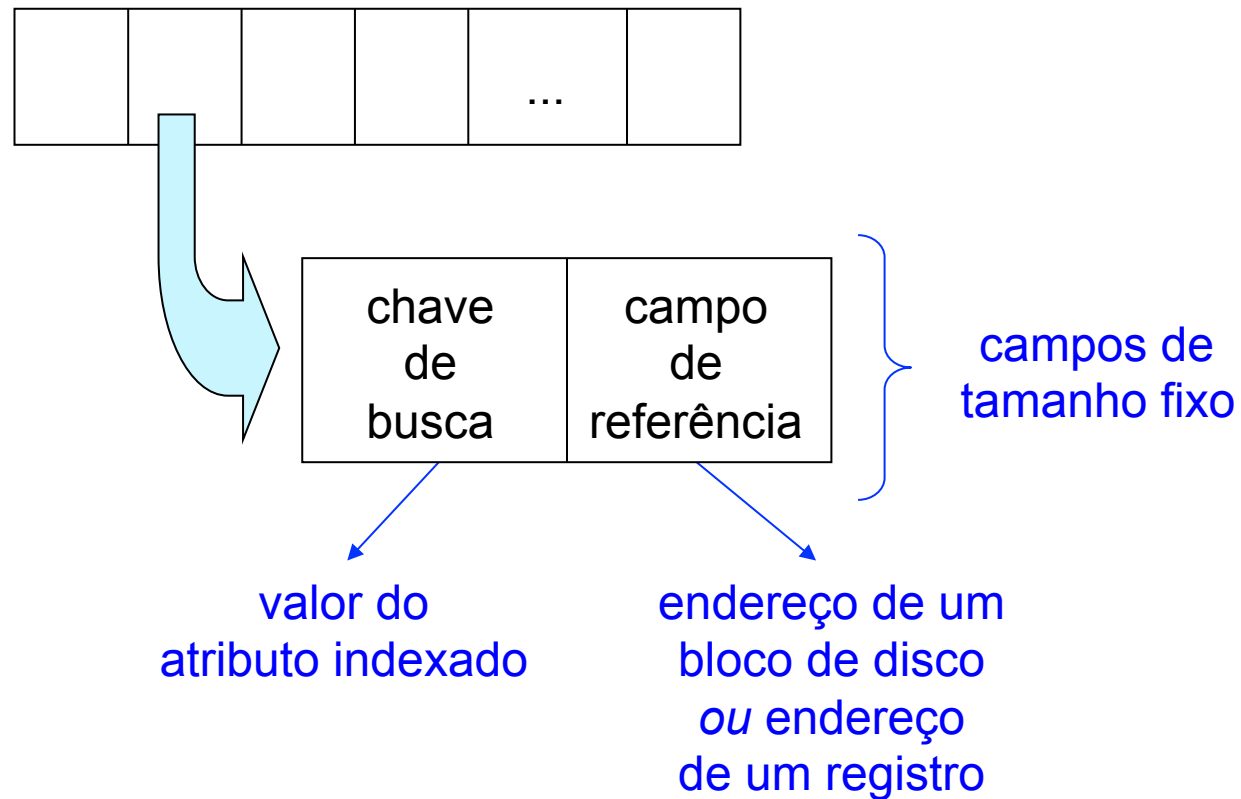
blocos adicionais  
encadeados

# Índice Secundário

- Características
  - ordenado
  - definido sobre um atributo não ordenado do arquivo de dados
  - possui um único nível
- Arquivo de dados
  - em geral, desordenado
  - porém, pode estar ordenado por outro atributo que não o indexado com índice secundário

# Índice Secundário

- Estrutura do registro (entrada)

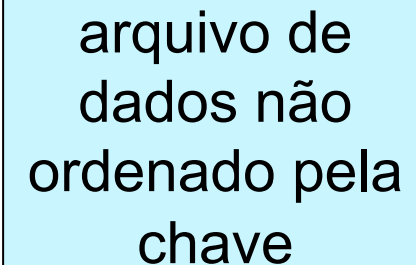


# Índice Secundário

- Vantagens
  - propicia uma ordenação lógica do arquivo de dados
  - facilita as operações de inserção e remoção em arquivos de dados desordenados
- Pode ser definido sobre atributo
  - chave (UNIQUE)
  - não chave

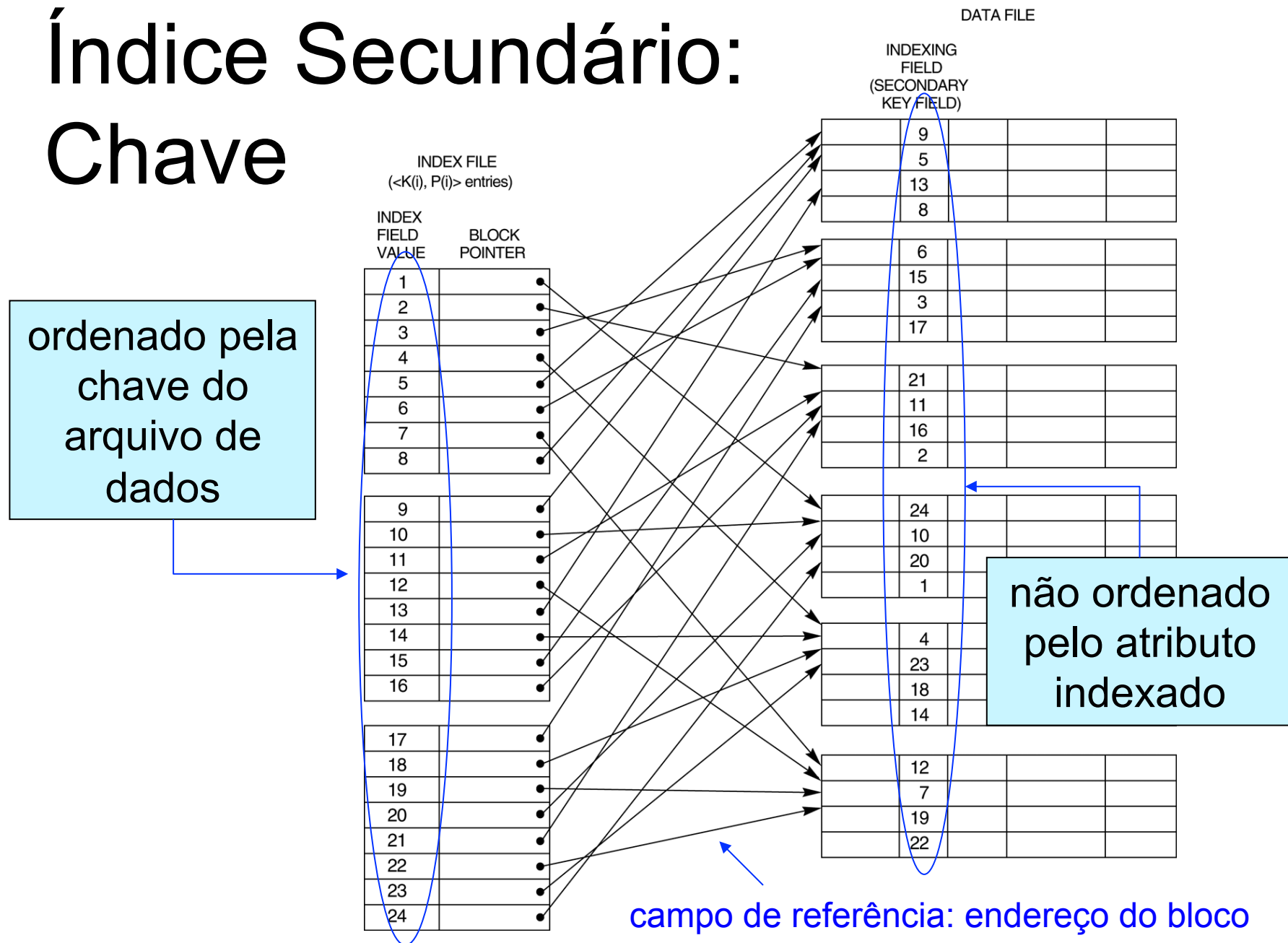
# Índice Secundário: Chave

- Denso
  - possui uma entrada para cada registro no arquivo de dados
  - não pode usar registros âncoras
  - chave de busca
    - valores distintos do atributo indexado



arquivo de  
dados não  
ordenado pela  
chave

# Índice Secundário: Chave





# Chave Primária

- Seqüência no arquivo de dados
  - ordenada: índice primário
  - desordenada: índice secundário
- Índice secundário
  - possui maior densidade do que o primário
    - maior número de entradas
    - maior espaço alocado em disco
  - apresenta pior desempenho na busca binária

# Desempenho da Pesquisa

Tipo de Índice	Arquivo de Índice	Arquivo de Dados	Melhora no Desempenho
primário <i>chave primária</i>	busca binária $O(\log_2 b)$	busca binária $O(\log_2 b)$	discreta
secundário <i>chave primária</i>	busca binária $O(\log_2 b)$	busca linear $O(b)$	significativa

- Índice secundário
  - deve ser utilizado para pesquisas freqüentes

# Arquivo de Dados

- Número de registros ( $r$ ) = 30.000
- Tamanho do bloco ( $B$ ) = 1.024 *bytes*
- Tamanho dos registros ( $R$ ) = 100 *bytes*
- Fator de bloco de disco ( $bfr$ ) =  $\lfloor B/R \rfloor = 10$
- Número de blocos ( $b$ ) =  $\lceil r/bfr \rceil = 3.000$

# Arquivo de Índice

- Número de registros ( $r$ ) = 30.000
  - número de registros do arquivo de dados
- Tamanho do bloco ( $B$ ) = 1.024 *bytes*
- Tamanho dos registros ( $R$ ) = 15 *bytes*
  - chave = 9 *bytes*
  - endereço = 6 *bytes*
- Fator de bloco de disco ( $bfr$ ) =  $\lfloor B/R \rfloor = 68$
- Número de blocos ( $b$ ) =  $\lceil r/bfr \rceil = 442$

# Acessos a Disco

- Sem o uso do índice
  - busca linear (custo médio)

$$b/2 = 3.000/2 = 1500$$

- Com o uso do índice
  - busca binária no arquivo de índice +
  - leitura do registro no arquivo de dados

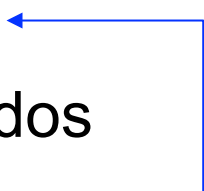
$$\lceil \log_2 442 \rceil + 1 = 9 + 1 = 10$$

# Índice Secundário: Não Chave

- Atributo não chave
    - pode possuir valores duplicados no arquivo de dados
  - Opções de implementação
    - incluir uma entrada no índice para cada registro do arquivo de dados
      - diversas entradas podem possuir o mesmo valor de chave de busca
- ⇒ índice denso

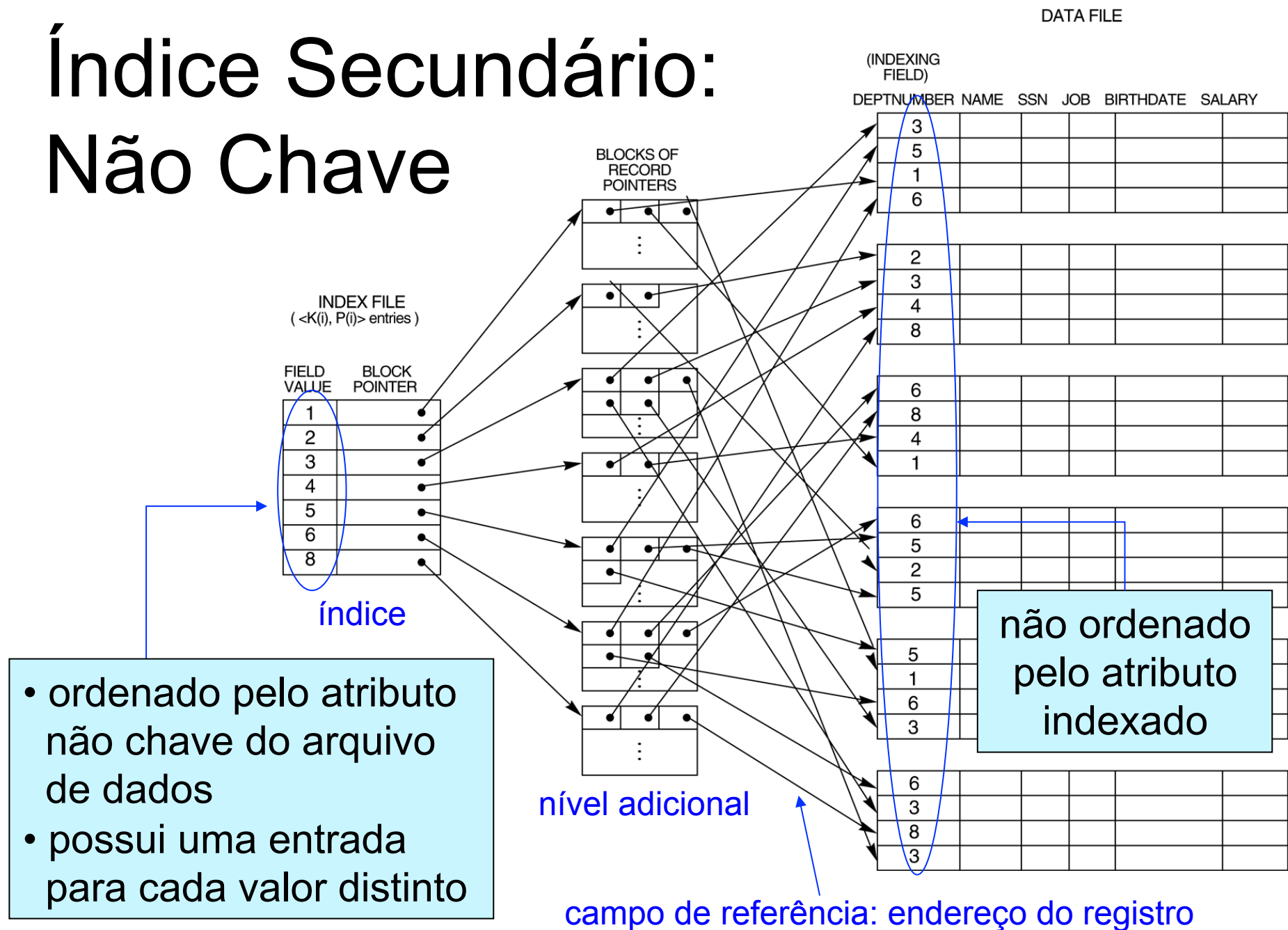
# Índice Secundário: Não Chave

- Opções de implementação
  - criar registros de tamanho variável
    - chave de busca: valores não repetidos
    - campo de referência: um ou mais endereços dos registros do arquivo de dados que satisfazem à chave de busca
  - utilizar um nível adicional de indireção
    - endereços dos registros do arquivo de dados que satisfazem à chave de busca são armazenados no nível adicional



técnica mais utilizada

# Índice Secundário: Não Chave



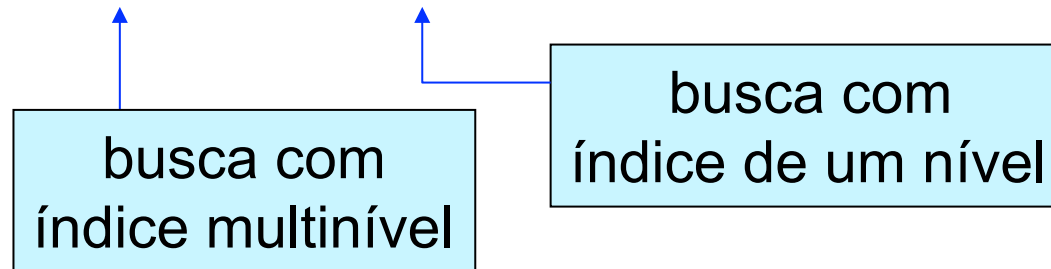


# Número de Índices

- Um arquivo pode possuir
  - no máximo um índice primário *ou* um índice de agrupamento
    - existe somente um campo ordenado
  - vários índices secundários
- Arquivo totalmente invertido
  - possui um índice secundário para cada um de seus campos

# Índice Multiníveis

- Objetivo
  - reduzir a parte do índice que a pesquisa seguirá
    - fo: *fan-out*
    - b: número de blocos
- Custo
  - $(\log_{fo} b) < (\log_2 b)$ , se  $fo > 2$



# Arquivos de Índice

- Primeiro nível
  - ordenado
  - armazena uma entrada para cada valor distinto de chave de busca

⇒ índice

- primário
- de agrupamento
- secundário

- Segundo nível
  - ordenado
  - armazena uma entrada para cada bloco do índice de primeiro nível
  - usa registros âncoras
  - estatísticas
    - $r_2 = \lceil r_1 / f_o \rceil$  registros
    - $f_{o2} = f_{o1}$

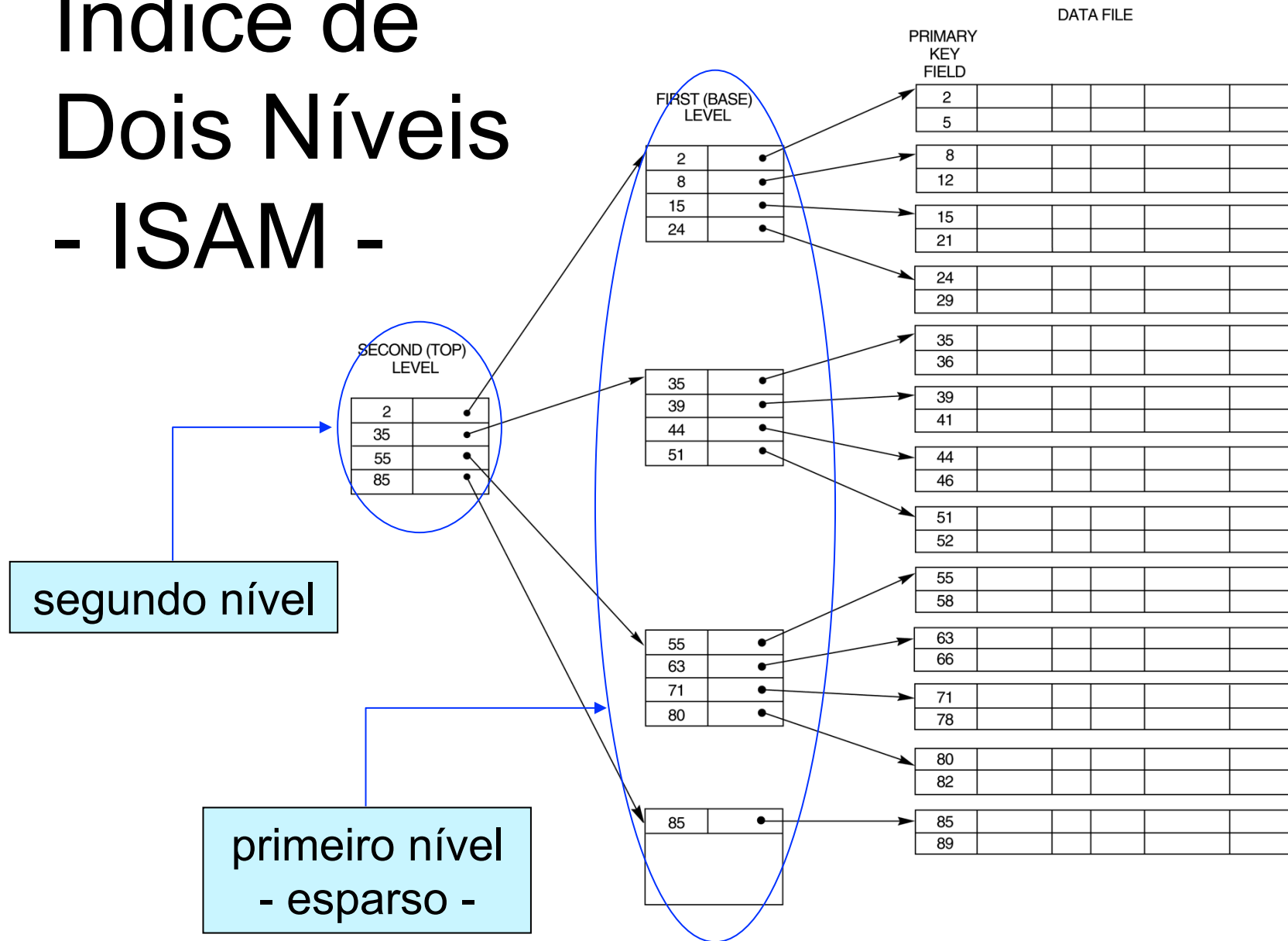
⇒ índice primário

# Arquivos de Índice

- Demais níveis (Terceiro, Quarto, ...)
  - possuem as mesmas características que o segundo nível
- Observação
  - um novo nível somente é necessário se o nível anterior necessita de mais do que um bloco de disco

$$\text{número de níveis} = \lceil (\log_{f_0}(r_1)) \rceil$$

# Índice de Dois Níveis - ISAM -



# Arquivo de Dados

- Número de registros ( $r$ ) = 30.000
- Tamanho do bloco ( $B$ ) = 1.024 *bytes*
- Tamanho dos registros ( $R$ ) = 100 *bytes*
- Fator de bloco de disco ( $bfr$ ) =  $\lfloor B/R \rfloor = 10$
- Número de blocos ( $b$ ) =  $\lceil r/bfr \rceil = 3.000$

# Arquivos de Índice

- Primeiro nível
  - número de registros ( $r$ ) = 30.000
    - número de registros do arquivo de dados
  - tamanho do bloco ( $B$ ) = 1.024 *bytes*
  - tamanho dos registros ( $R$ ) = 15 *bytes*
    - chave = 9 *bytes*
    - endereço = 6 *bytes*
  - fator de bloco de disco ( $bfr$ ) =  $fo = \lfloor B/R \rfloor = 68$
  - número de blocos ( $b_1$ ) =  $\lceil r/bfr \rceil = 442$

# Arquivos de Índice

- Número de níveis
  - $\lceil (\log_{f_0}(r_1)) \rceil = \lceil (\log_{68}(30.000)) \rceil = 3$
- Número de blocos
  - segundo nível
    - $b_2 = \lceil b_1/f_0 \rceil = \lceil 442/68 \rceil = 7$  blocos
  - terceiro nível
    - $b_3 = \lceil b_2/f_0 \rceil = \lceil 7/68 \rceil = 1$  bloco



# Acessos a Disco

- Sem o uso do índice
  - busca binária no arquivo de dados

$$\lceil \log_2 3000 \rceil = 12$$

- Com o uso de índice de um nível (denso)
  - busca binária no arquivo de índice +
  - leitura do registro no arquivo de dados

$$\lceil \log_2 442 \rceil + 1 = 9 + 1 = 10$$

# Acessos a Disco

- Com o uso do índice multinível
  - acesso a um bloco de cada nível +
  - leitura do registro no arquivo de dados

$$3 + 1 = 4$$

# Inserção e Remoção

- Problemas
  - deslocamento dos registros nos arquivos de dados e de índice para ordenação
  - alteração dos valores dos campos de referência nos arquivos de índice
- Soluções
  - reservar espaço extra livre
  - utilizar árvores B ou B<sup>+</sup>