

# Tipos de Índices

Cristina Dutra de Aguiar Ciferri

# Tipos de Índice

- Ordenados em um único nível
  - primário
  - agrupamento (cluster)
  - secundário
- Estruturas de dados de árvores
  - índices multiníveis
  - árvores-B, árvores-B+



implementam  
dinamicamente índices  
multiníveis

# Índice Primário

- Características

- ordenado

- definido com base em um **arquivo de dados ordenado** pela **chave primária**

- possui um único nível

- **esparso**

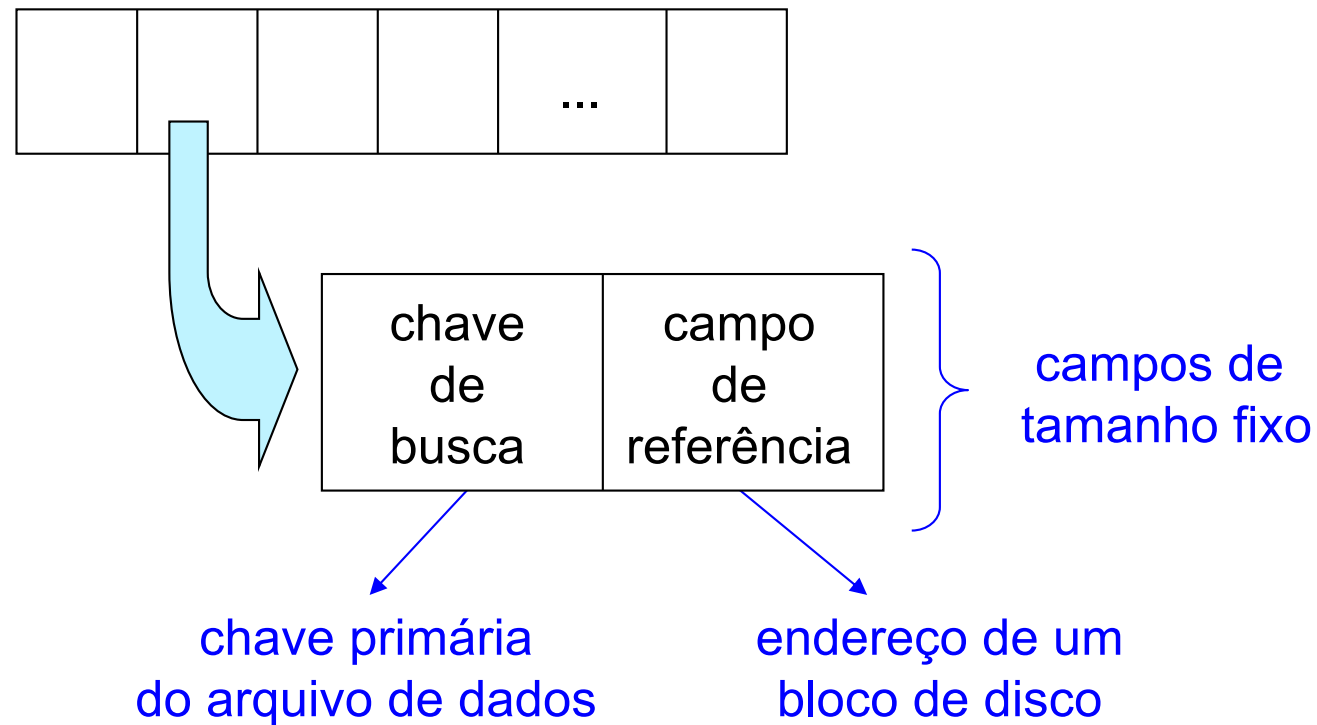
- total de entradas no índice = número de blocos do arquivo de dados



diminui o total de blocos e melhora o desempenho na pesquisa

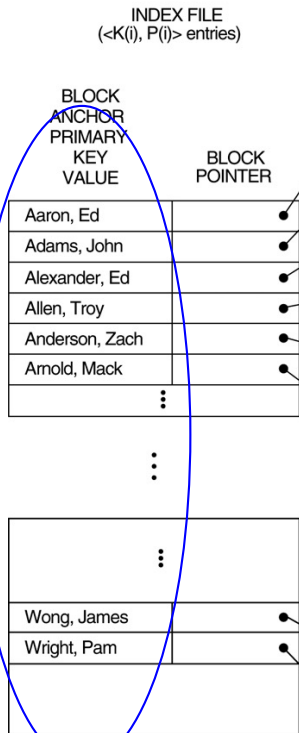
# Índice Primário

- Estrutura do registro (entrada)



# Índice Primário

ordenado pela chave primária do arquivo de dados



ordenado pela chave primária

DATA FILE

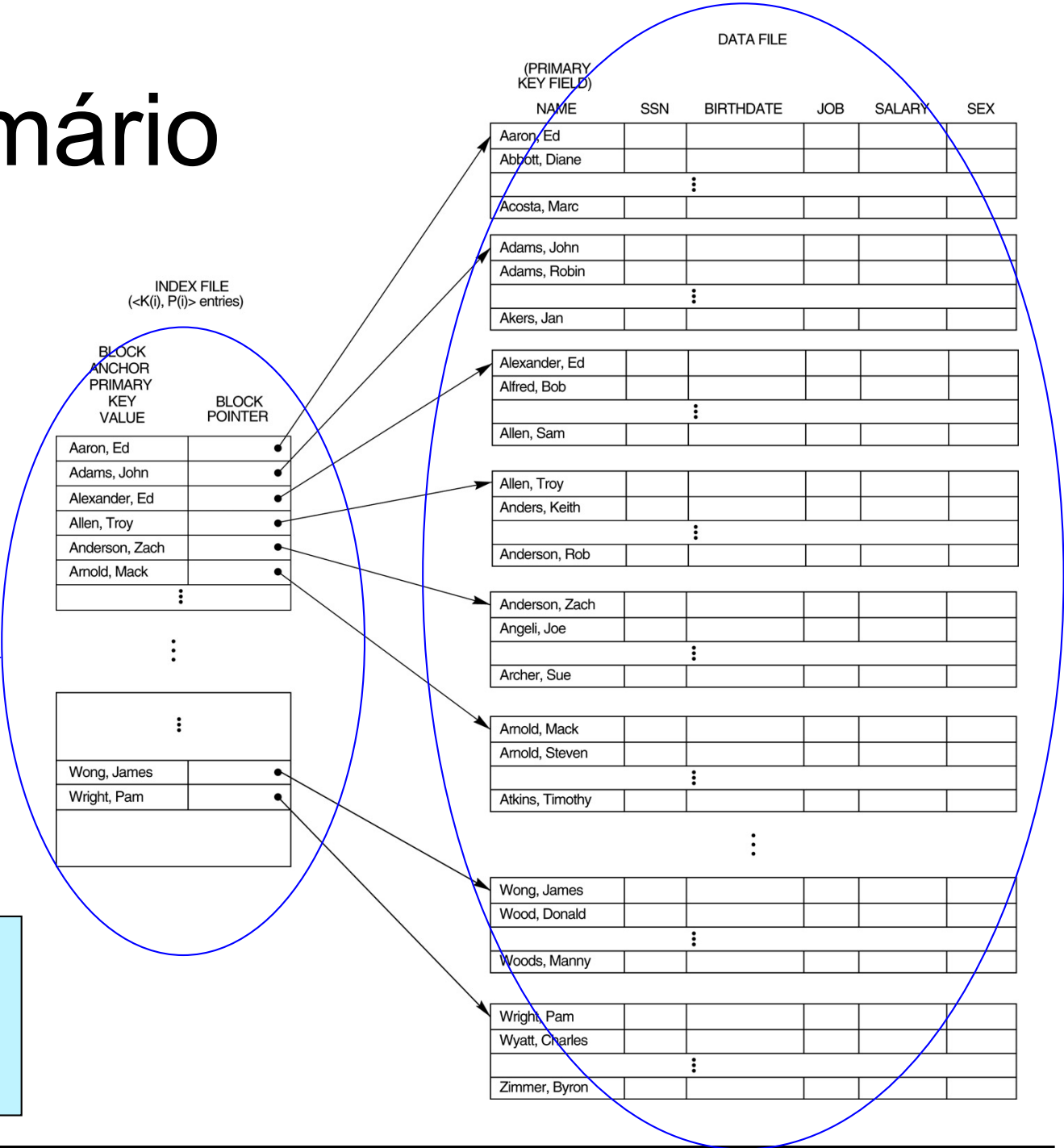
(PRIMARY KEY FIELD) NAME	SSN	BIRTHDATE	JOB	SALARY	SEX
Aaron, Ed					
Abbott, Diane					
⋮					
Acosta, Marc					
⋮					
Adams, John					
Adams, Robin					
⋮					
Akers, Jan					
⋮					
Alexander, Ed					
Alfred, Bob					
⋮					
Allen, Sam					
⋮					
Allen, Troy					
Anders, Keith					
⋮					
Anderson, Rob					
⋮					
Anderson, Zach					
Angeli, Joe					
⋮					
Archer, Sue					
⋮					
Arnold, Mack					
Arnold, Steven					
⋮					
Atkins, Timothy					
⋮					
Wong, James					
Wood, Donald					
⋮					
Woods, Manny					
⋮					
Wright, Pam					
Wyatt, Charles					
⋮					
Zimmer, Byron					

# Índice Primário

menor número de blocos

- menos entradas
- registros menores

pesquisa binária mais eficiente no índice



# Arquivo de Dados

- Número de registros ( $r_d$ ) = 30.000
- Tamanho do bloco ( $B$ ) = 1.024 *bytes*
- Tamanho dos registros ( $R_d$ ) = 100 *bytes*
- Fator de bloco de disco ( $bfr_d$ ) =  $\lfloor B/R_d \rfloor = 10$   
– número de registros por bloco
- Número de blocos ( $b_d$ ) =  $\lceil r_d/bfr_d \rceil = 3.000$

# Arquivo de Índice

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



# Acessos a Disco

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

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

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

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

# Inserção e Remoção

- Inserção
  - deslocamento dos registros nos arquivos de dados e de índice para ordenação
  - alteração dos valores dos campos de referência no índice (registros âncoras)
- Remoção
  - lógica, ao invés de física
    - registros marcados como removidos

necessidade de reorganização periódica  
com recriação do índice

# Índice de Agrupamento

- Características

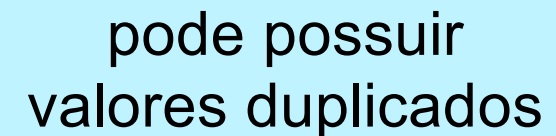
- ordenado

- definido com base em um **arquivo de dados ordenado** por um **atributo não chave** (atributo de agrupamento)

- possui um único nível

- **esparso**

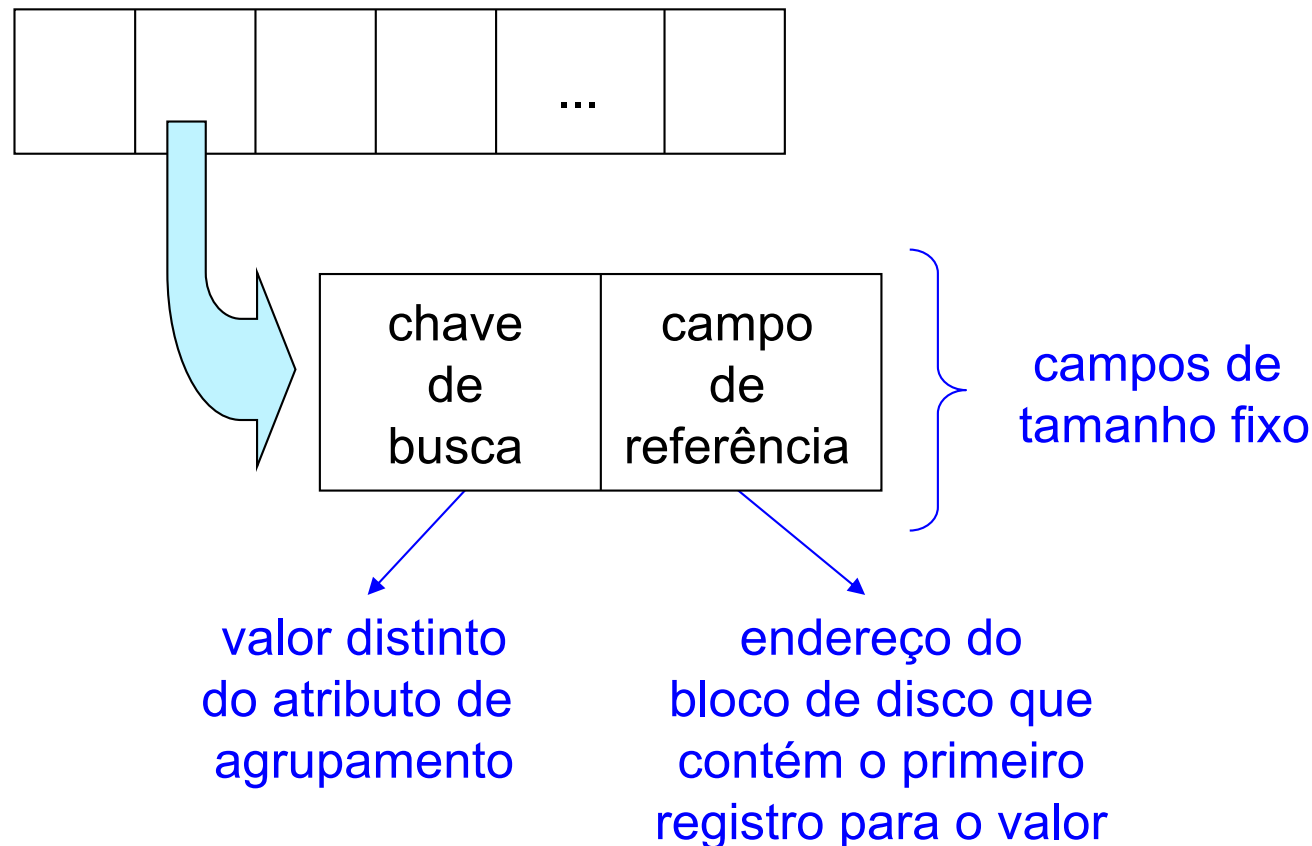
- total de entradas no índice = número de valores distintos do atributo de agrupamento



pode possuir  
valores duplicados

# Índice de Agrupamento

- Estrutura do registro (entrada)







# Inserção e Remoção

- Inserção
  - deslocamento dos registros nos arquivos de dados e de índice para ordenação
  - alteração dos valores dos campos de referência no índice (registros âncoras)
- Remoção
  - lógica, ao invés de física
    - registros marcados como removidos

necessidade de reorganização periódica  
com recriação do índice

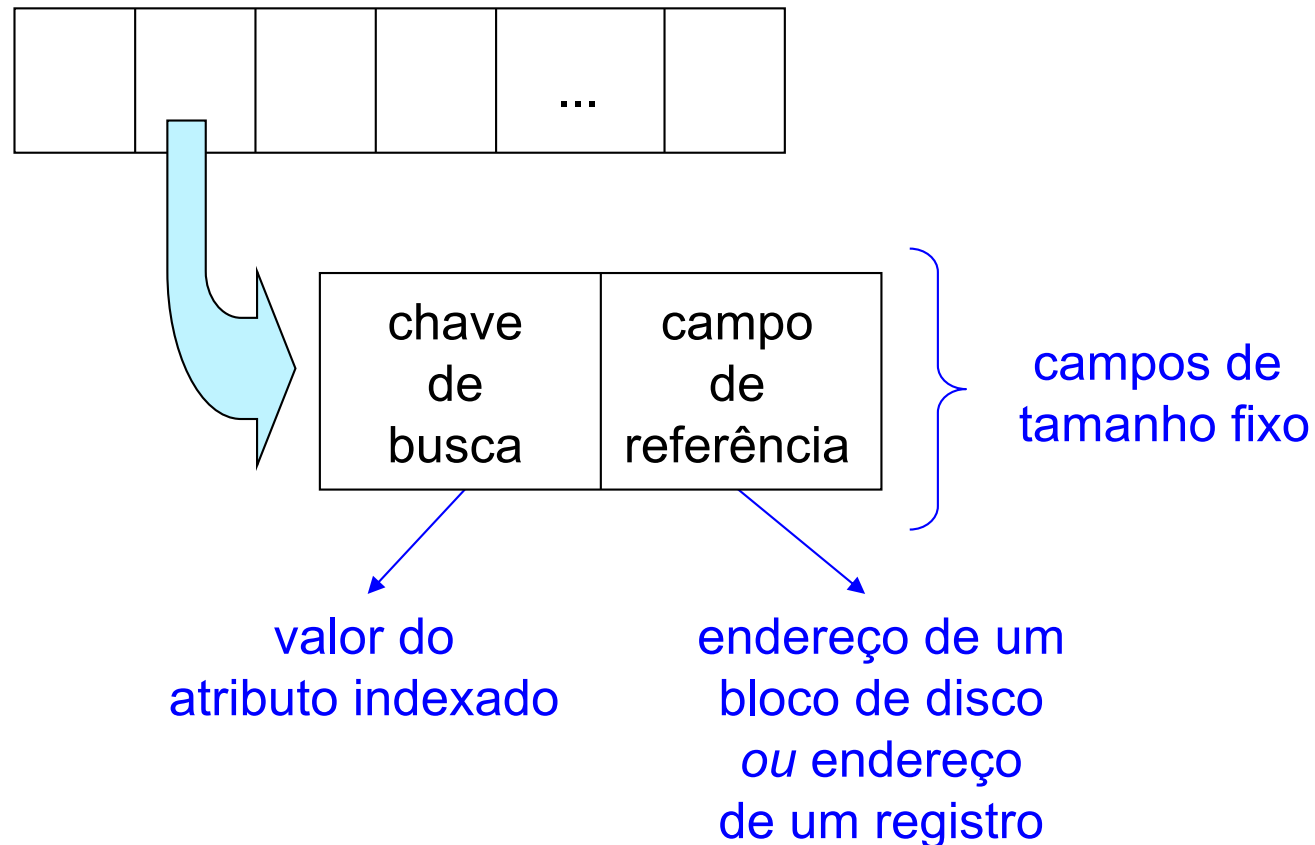
# Índice Secundário

- 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)

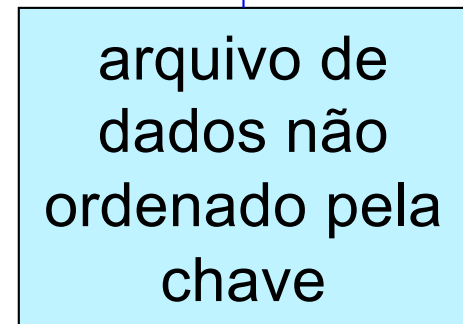


# Índice Secundário

- 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** (sem valores repetidos)
  - **não chave** (com valores repetidos)

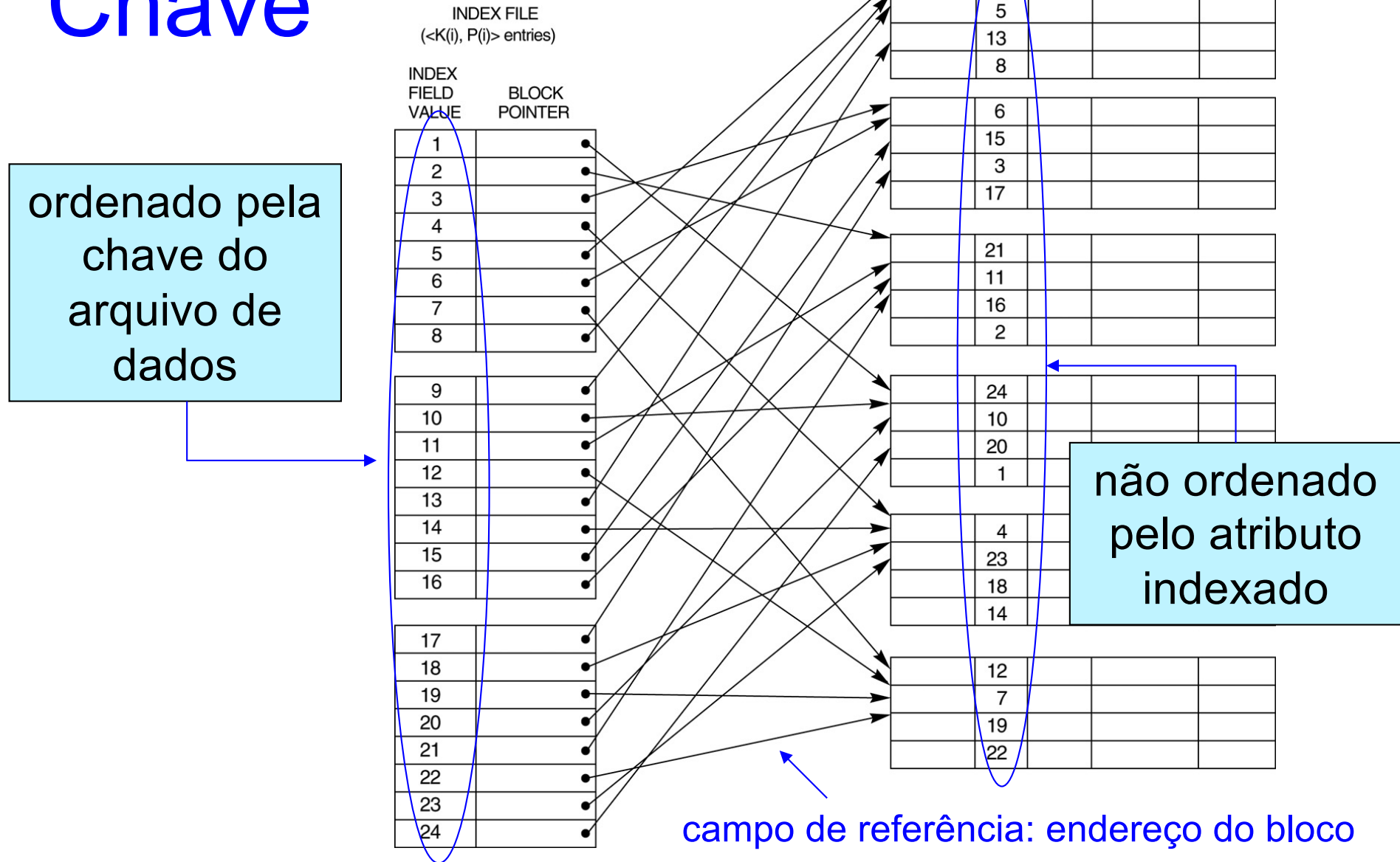
# Índice Secundário: Chave

- 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



# Arquivo de Dados

- Número de registros ( $r_d$ ) = 30.000
- Tamanho do bloco ( $B$ ) = 1.024 *bytes*
- Tamanho dos registros ( $R_d$ ) = 100 *bytes*
- Fator de bloco de disco ( $bfr_d$ ) =  $\lfloor B/R_d \rfloor = 10$
- Número de blocos ( $b_d$ ) =  $\lceil r_d/bfr_d \rceil = 3.000$

# Arquivo de Índice

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

# Acessos a Disco

- Sem o uso do índice
  - busca linear no arquivo de dados (custo médio)

$$b_d/2 = 3.000/2 = 1500$$

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

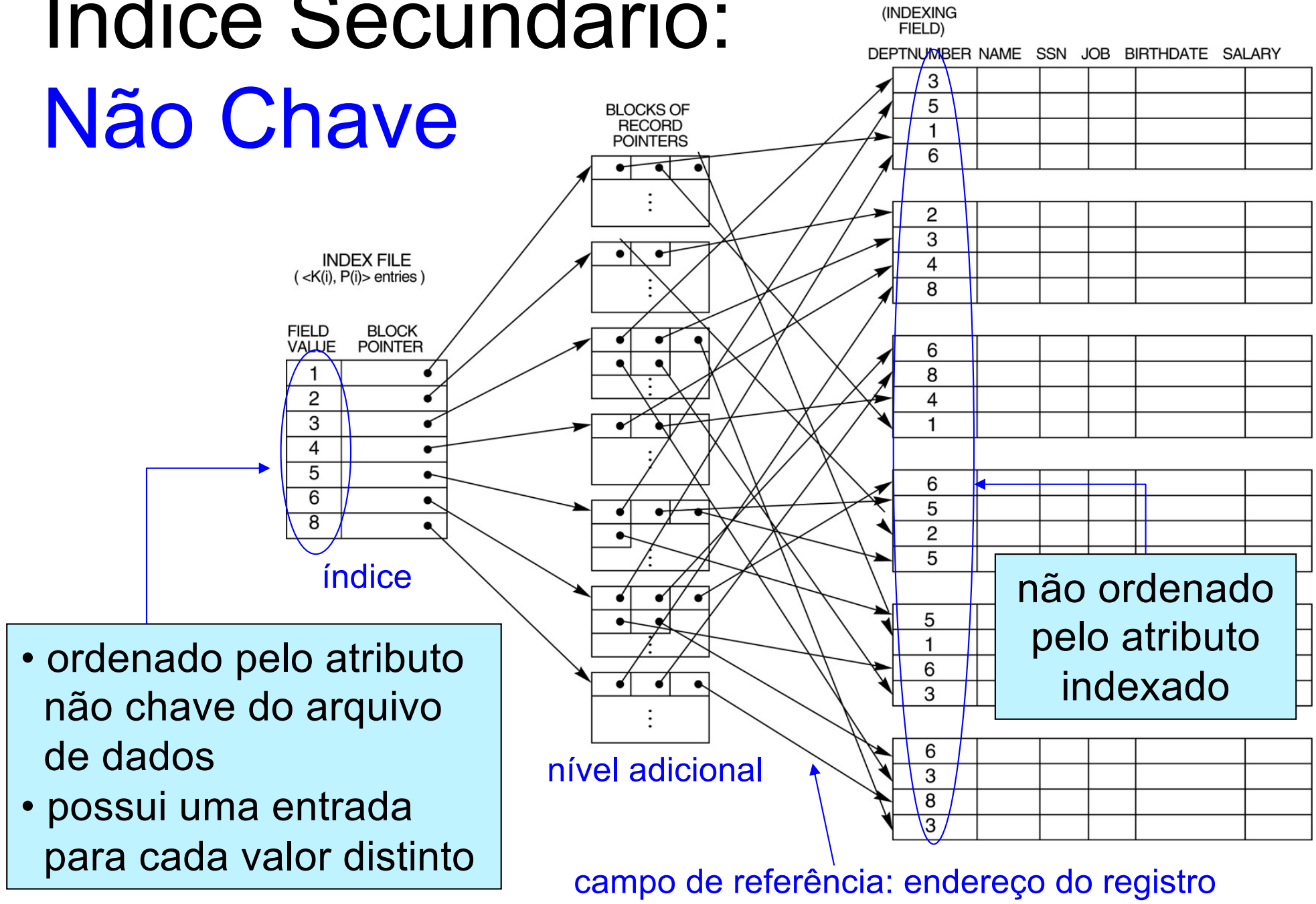
$$\lceil \log_2 b_i \rceil + 1 = \lceil \log_2 442 \rceil + 1 = 9 + 1 = 10$$

# Índice Secundário: Não Chave

- Atributo não chave
  - pode possuir valores duplicados no arquivo de dados
- Técnica de implementação mais usada
  - utilizar um nível adicional de indireção
    - endereços dos registros do arquivo de dados que satisfazem à chave de busca são armazenados no nível adicional



# Índice Secundário: Não Chave



- ordenado pelo atributo não chave do arquivo de dados
- possui uma entrada para cada valor distinto

não ordenado pelo atributo indexado

nível adicional

campo de referência: endereço do registro

# Número de Índices

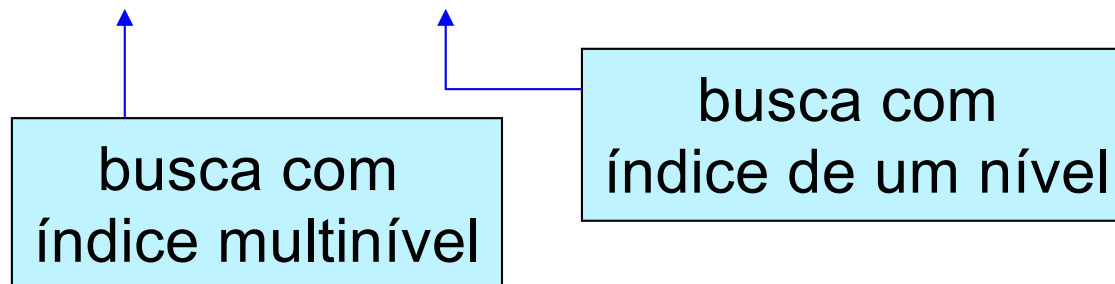
- Um arquivo pode possuir
  - no máximo um índice primário *ou* um índice de agrupamento
    - existe somente um campo ordenado
  - vários índices secundários
- Arquivo totalmente invertido
  - possui um índice secundário para cada um de seus campos

# Tipos de Campos e de Índices

- Campo de indexação: **chave**
  - ordenada (arquivo de dados): índice primário
  - desordenada (arquivo de dados): índice secundário
- Campo de indexação: **não chave**
  - ordenada (arquivo de dados): índice de agrupamento
  - desordenada (arquivo de dados): índice secundário

# Índice Multiníveis

- Objetivo
  - reduzir a parte do índice que a pesquisa seguirá
    - fo: *fan-out* (bfr)
    - 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
  - $f_{o_2} = f_{o_1}$

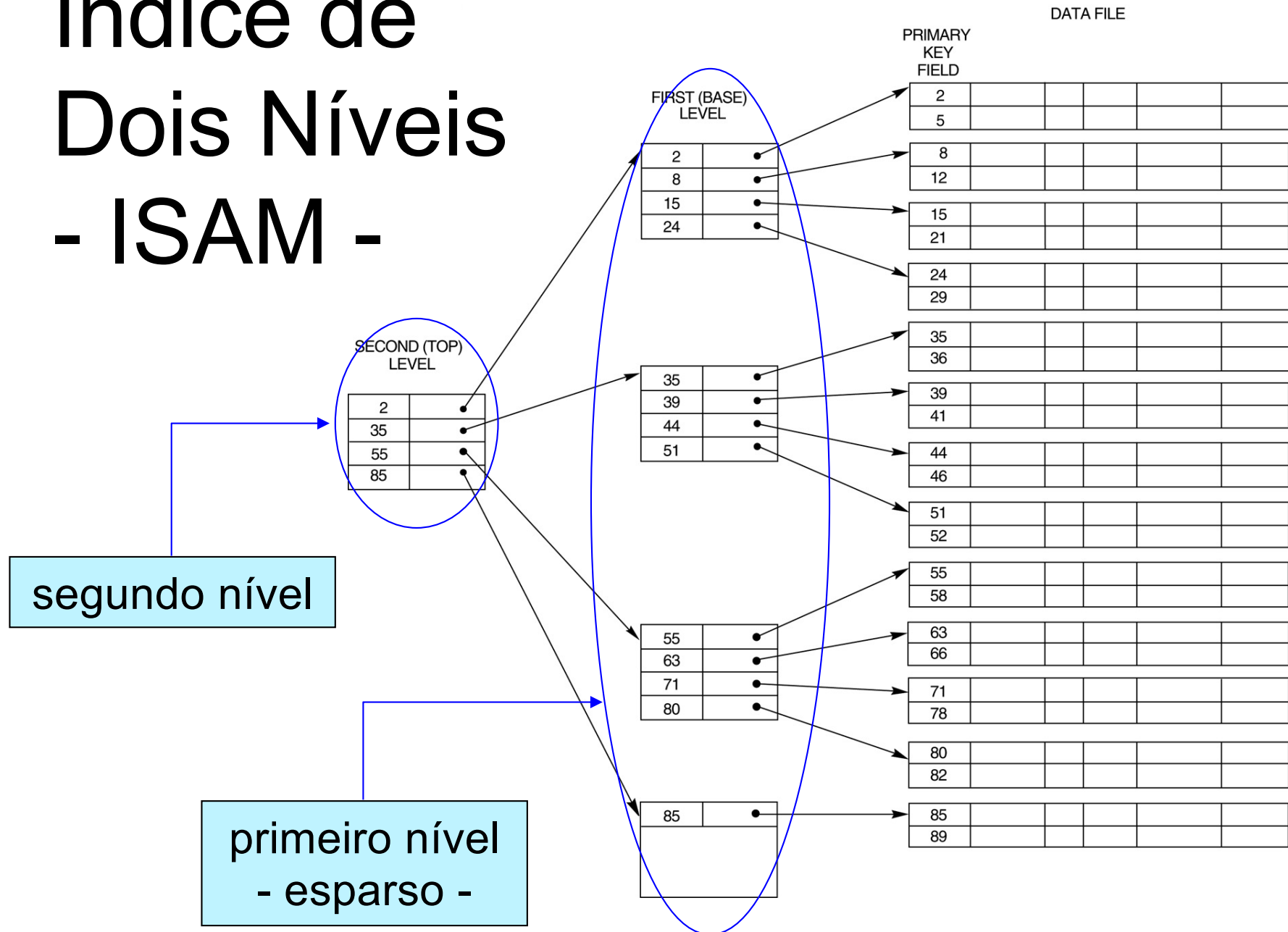
⇒ índice primário

# 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 ( $r_d$ ) = 30.000
- Tamanho do bloco ( $B$ ) = 1.024 *bytes*
- Tamanho dos registros ( $R_d$ ) = 100 *bytes*
- Fator de bloco de disco ( $bfr_d$ ) =  $\lfloor B/R_d \rfloor = 10$
- Número de blocos ( $b_d$ ) =  $\lceil r_d/bfr_d \rceil = 3.000$



# Arquivos de Índice

- Primeiro nível
  - número de registros ( $r_{i_1}$ ) = 30.000
    - número de registros do arquivo de dados
  - tamanho do bloco ( $B$ ) = 1.024 *bytes*
  - tamanho dos registros ( $R_{i_1}$ ) = 15 *bytes*
    - chave = 9 *bytes*
    - endereço = 6 *bytes*
  - fator de bloco de disco ( $bfr_{i_1}$ ) =  $fo_{i_1} = \lfloor B/R_{i_1} \rfloor = 68$
  - número de blocos ( $b_{i_1}$ ) =  $\lceil r/bfr \rceil = 442$

# Arquivos de Índice

- Número de níveis
  - $\lceil (\log_{f_0}(r_{i_1})) \rceil = \lceil (\log_{68}(30.000)) \rceil = 3$
- Número de blocos
  - segundo nível
    - $b_{i_2} = \lceil b_{i_1}/f_0 \rceil = \lceil 442/68 \rceil = 7$  blocos
  - terceiro nível
    - $b_{i_3} = \lceil b_{i_2}/f_0 \rceil = \lceil 7/68 \rceil = 1$  bloco

# Acessos a Disco

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

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

- 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
- Solução amplamente usada em SGBDs
  - utilizar **árvores B ou B<sup>+</sup>**
  - conteúdo coberto na disciplina de Algoritmos e Estruturas de Dados II