

Otimização e Processamento de Consultas

- Objetivo
 - produzir uma estratégia de consulta para recuperar o resultado da mesma
 - plano para: executar a consulta
 - acessar os dados
 - armazenar resultados intermediários
 - Deve ser utilizado algum índice? Qual índice deve ser escolhido?

Estimativa de Custo

- Estratégia a ser escolhida depende
 - do tamanho de cada relação
 - da distribuição de valores dentro de colunas
 - custo
- Objetivo
 - estimar o tamanho do resultado (número de tuplas a serem retornadas) e o custo da consulta

Componentes da Função Custo

- Custo de acesso à memória secundária
 - custo para buscar, ler e escrever blocos de dados que residem em disco
- Custo de armazenamento
 - custo para armazenar quaisquer arquivos intermediários gerados pela estratégia de execução da consulta

Componentes da Função Custo

- Custo de computação
 - custo para a realização de operações em memória principal (i.e., buffers)
- Custo do uso da memória
 - custo relacionado ao número de buffers de memória principal necessários durante a execução da consulta

Componentes da Função Custo

- Custo de comunicação
 - custo de transmitir uma consulta e os seus resultados do site do banco de dados até o site ou terminar na qual a consulta foi originada

Funções Custo para Seleção

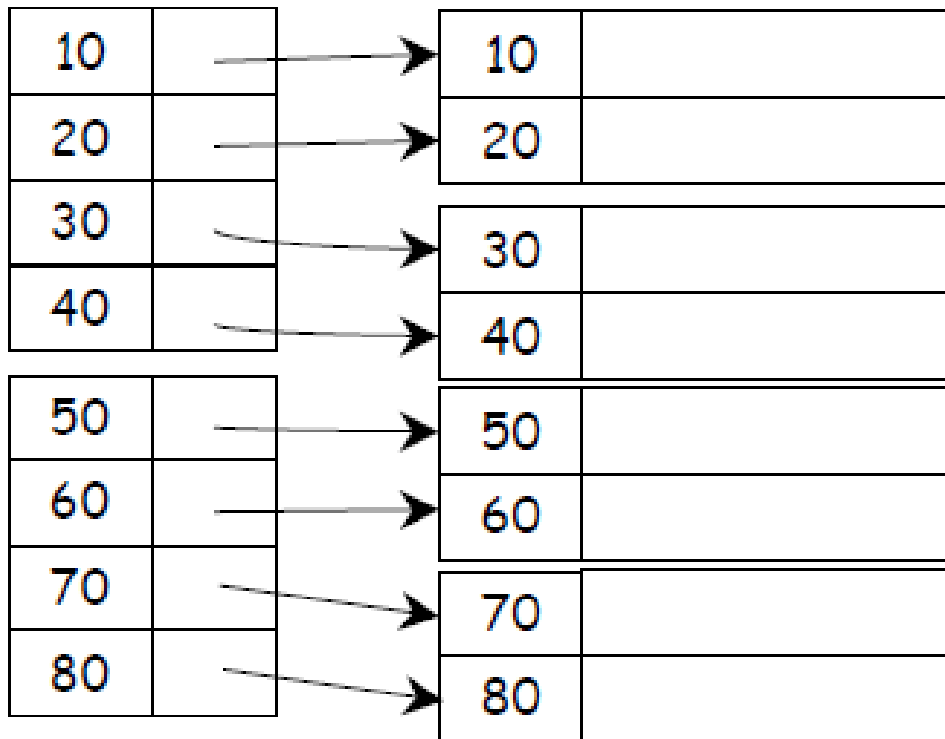
- Métodos para seleção simples
 - varredura de arquivo (i.e., file scan)
 - linear
 - binária
 - varredura de índice (i.e., index scan)
 - Primário
 - Agrupamento
 - Secundário

Estruturas de Índices

- Índice primário - baseado na chave de ordenação;
- Índice de agrupamento (clustering) - baseado no campo de ordenação não-chave de um arquivo;
- Índice secundário - baseado em qualquer campo não ordenado de um arquivo;
- Índices multiníveis;
- Árvores B e B+;
- Tabelas Hash;

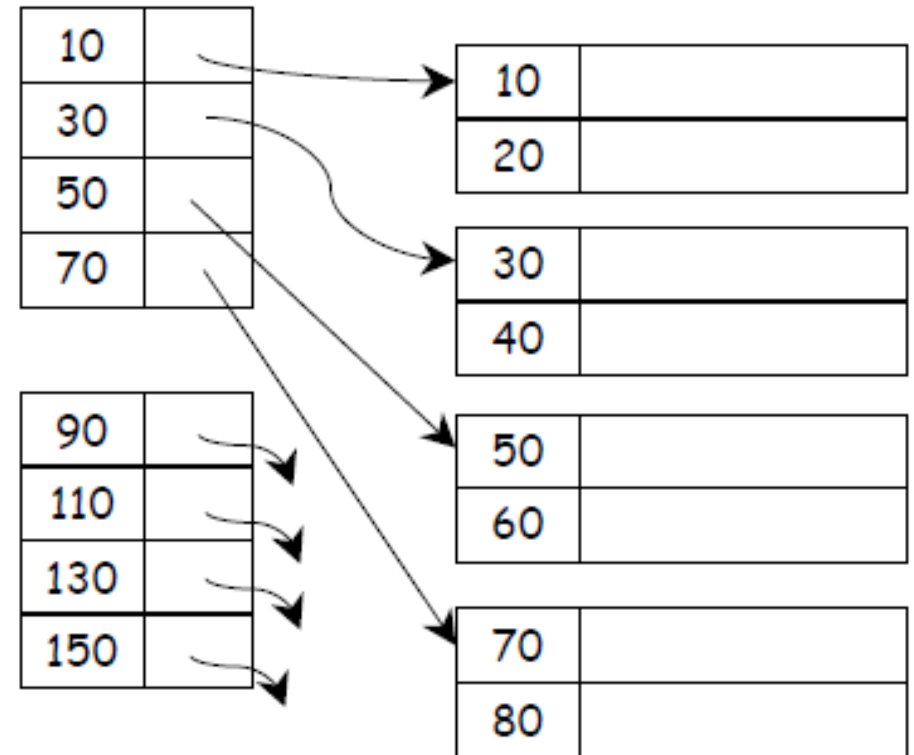
Tipos de Índices

Densos: uma entrada no arquivo de índices p/cada registro no arquivo de dados



Um índice denso sobre um arquivo de dados sequenciais

Esparsos: apenas alguns registros de dados são representados no arquivo de índices



Um índice esparsos sobre um arquivo de dados sequenciais

Tipos

- Índice Denso

- Seqüência de blocos contendo apenas as chaves dos registros e os ponteiros para os próprios registros
- Índice denso = (chave-ponteiro, registro)

- Índice Esparso

- Usa menos espaço de armazenamento que o índice denso ao custo de um tempo um pouco maior para localizar um registro dada a sua chave
- Índice esparsos = (c,
 - Aponta para o 1o. registro do bloco dados)

Tipos de Índices

| Índice | Arquivo de Dados | Atributo Indexado | Registros recuperados |
|-------------|------------------|-------------------|-----------------------|
| Primário | ordenado | Chave primária | 0 ou 1 |
| Agrupamento | ordenado | Não chave | 0 ou vários |
| Secundário | desordenado | Chave primária | 0 ou 1 |
| | | Não chave | 0 ou vários |

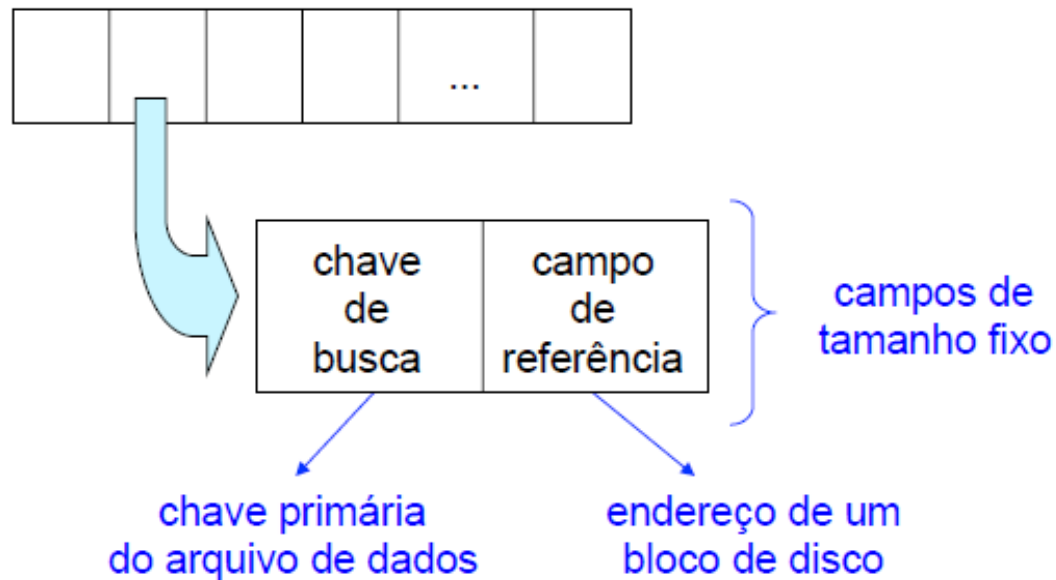
Índice Primário

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

INDEX FILE
($\langle K(i), P(i) \rangle$ entries)

| BLOCK ANCHOR | PRIMARY KEY VALUE | BLOCK POINTER |
|-----------------|-------------------------|------------------|
| | Aaron, Ed | • |
| | Adams, John | • |
| | Alexander, Ed | • |
| | Allen, Troy | • |
| | Anderson, Zach | • |
| | Arnold, Mack | • |
| | ... | |
| | ... | |
| | Wong, James | • |
| | Wright, Pam | • |
| | ... | |

ordenado pela
chave primária

DATA FILE

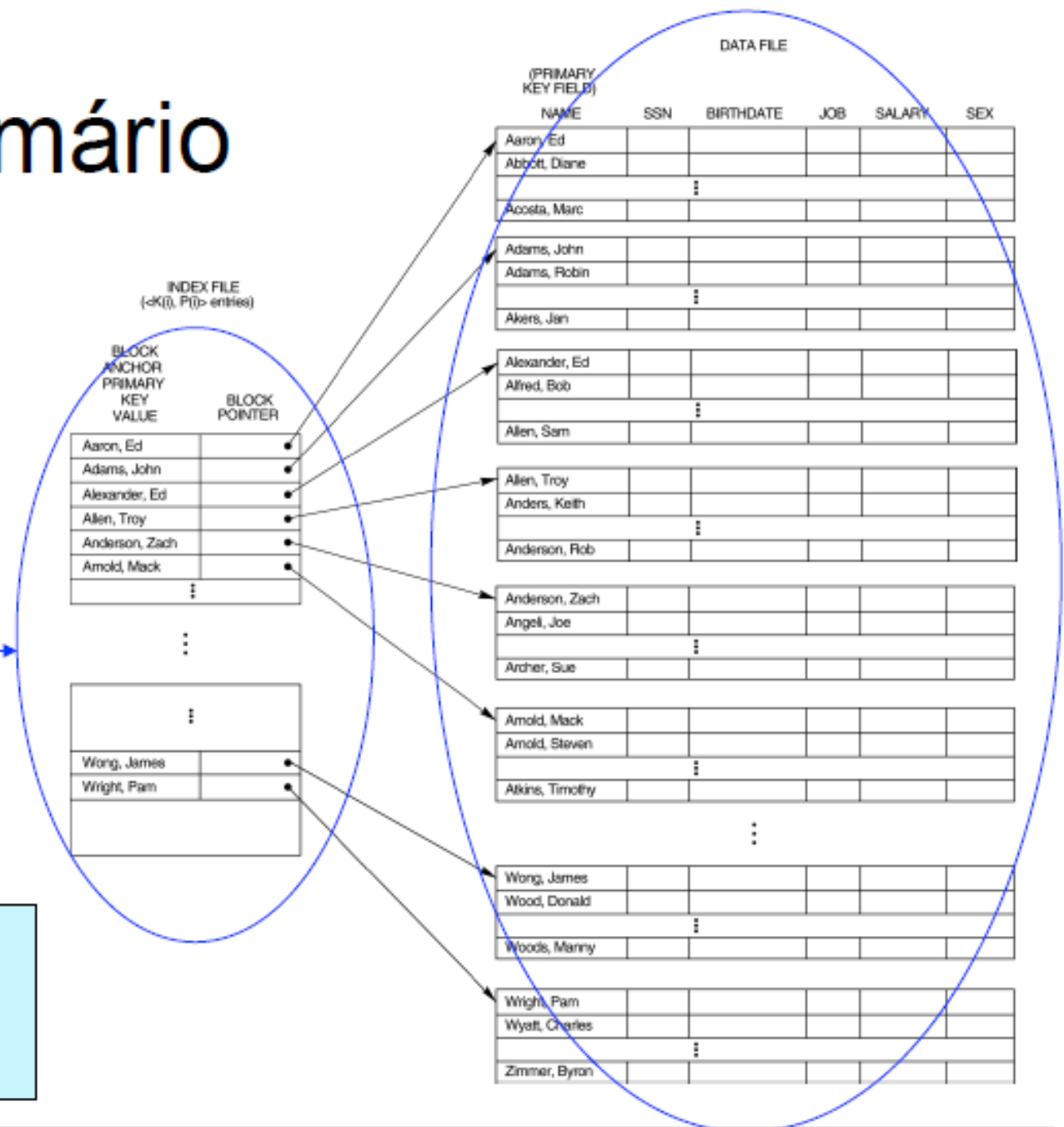
| (PRIMARY KEY FIELD) | NAME | SSN | BIRTHDATE | JOB | SALARY | SEX |
|------------------------|-----------------|-----|-----------|-----|--------|-----|
| | Aaron, Ed | | | | | |
| | Abbott, Diane | | | | | |
| | ... | | | | | |
| | Acosta, Marc | | | | | |
| | Adams, John | | | | | |
| | Adams, Robin | | | | | |
| | ... | | | | | |
| | Akers, Jan | | | | | |
| | Alexander, Ed | | | | | |
| | Alfred, Bob | | | | | |
| | ... | | | | | |
| | Allen, Sam | | | | | |
| | Allen, Troy | | | | | |
| | Anders, Keith | | | | | |
| | ... | | | | | |
| | Anderson, Rob | | | | | |
| | Anderson, Zach | | | | | |
| | Angell, 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) = 30.000
- Tamanho da página (P) = 1.024 bytes
- Tamanho dos registros (R) = 100 bytes
- Fator de página de disco (bfr) = $(\text{ piso})P/R = 10$
 - número de registros que cabem em uma páginas
- Número de páginas (p) = $(\text{ teto}) r/bfr = 3.000$

Arquivo de Índice

- Número de registros (r) = 3.000
 - número de páginas do arquivo de dados
- Tamanho da página (P) = 1.024 bytes
- Tamanho dos registros (R) = 15 bytes
 - chave = 9 bytes
 - endereço = 6 bytes
- Fator de página de disco (bfr) = (piso) $P/R = 68$
- Número de páginas (p) = (teto) $r/bfr = 45$

Desempenho em Acesso a discos

- Sem o uso do índice
 - busca binária no arquivo de dados
 $\log_2 n = \log_2 3000 = 12$
- Com o uso do índice
 - busca binária no arquivo de índice + leitura do registro no arquivo de dados
 $\log_2 n + 1 = \log_2 45 + 1 = 6 + 1 = 7$

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

Remoção

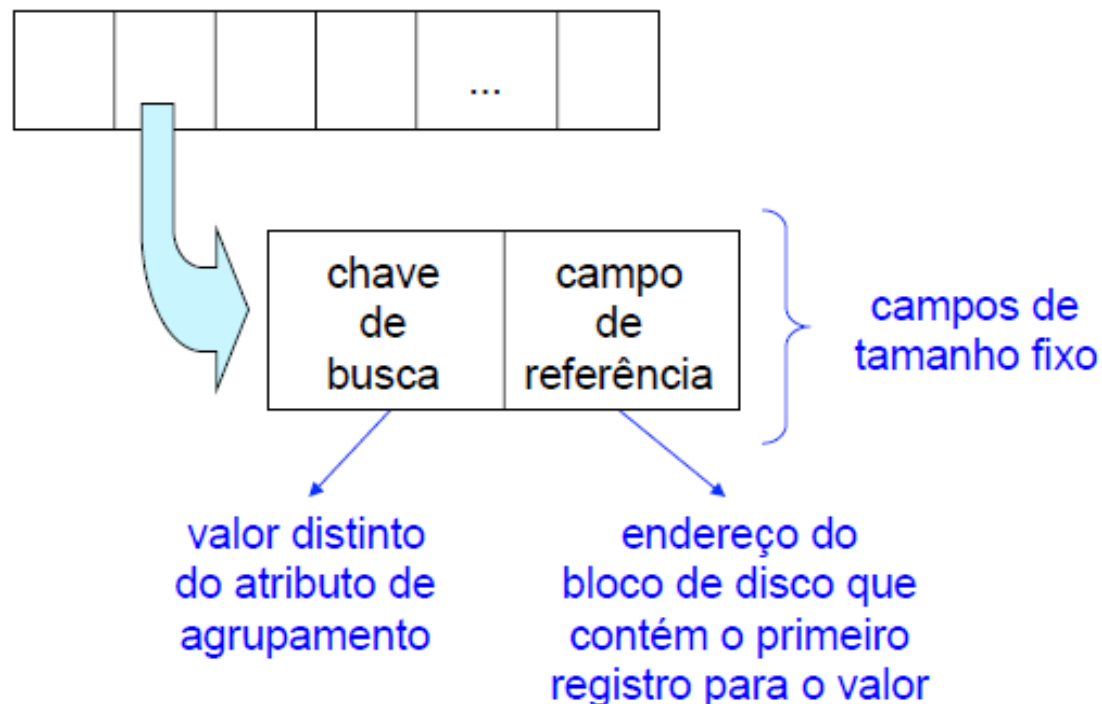
- Características
 - lógica, ao invés de física
 - registros marcados como removidos
- reorganização periódica → recriação do índice

Índice de Agrupamento (Cluster)

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

Índice de Agrupamento

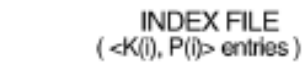
- Estrutura do registro (entrada)



Índice de Agrupamento

ordenado pelo
atributo de
agrupamento do
arquivo de
dados

ordenado pelo atributo
de agrupamento



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

(CLUSTERING
FIELD)

DATA FILE

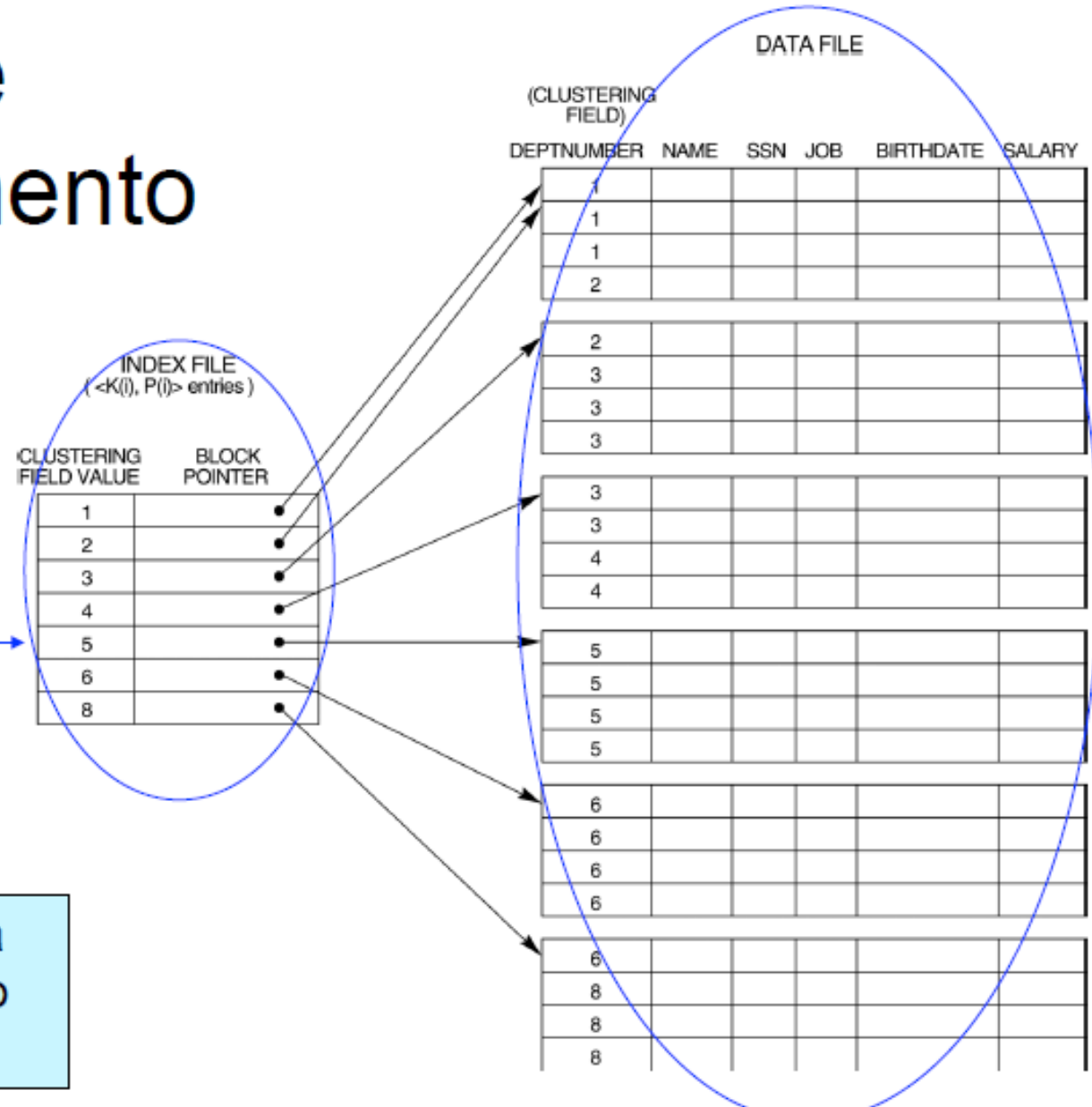
| DEPT | NUMBER | NAME | SSN | JOB | BIRTHDATE | SALARY |
|------|--------|------|-----|-----|-----------|--------|
| 1 | 1 | | | | | |
| 1 | 1 | | | | | |
| 1 | 1 | | | | | |
| 2 | 2 | | | | | |
| 2 | 2 | | | | | |
| 3 | 3 | | | | | |
| 3 | 3 | | | | | |
| 3 | 3 | | | | | |
| 3 | 3 | | | | | |
| 3 | 3 | | | | | |
| 3 | 3 | | | | | |
| 3 | 3 | | | | | |
| 4 | 4 | | | | | |
| 4 | 4 | | | | | |
| 5 | 5 | | | | | |
| 5 | 5 | | | | | |
| 5 | 5 | | | | | |
| 5 | 5 | | | | | |
| 6 | 6 | | | | | |
| 6 | 6 | | | | | |
| 6 | 6 | | | | | |
| 6 | 6 | | | | | |
| 6 | 6 | | | | | |
| 6 | 6 | | | | | |
| 8 | 8 | | | | | |
| 8 | 8 | | | | | |
| 8 | 8 | | | | | |
| 8 | 8 | | | | | |

Índice de Agrupamento

menor número de
blocos

- menos entradas
- registros menores

pesquisa binária
mais eficiente no
índice



Inserção e Remoção

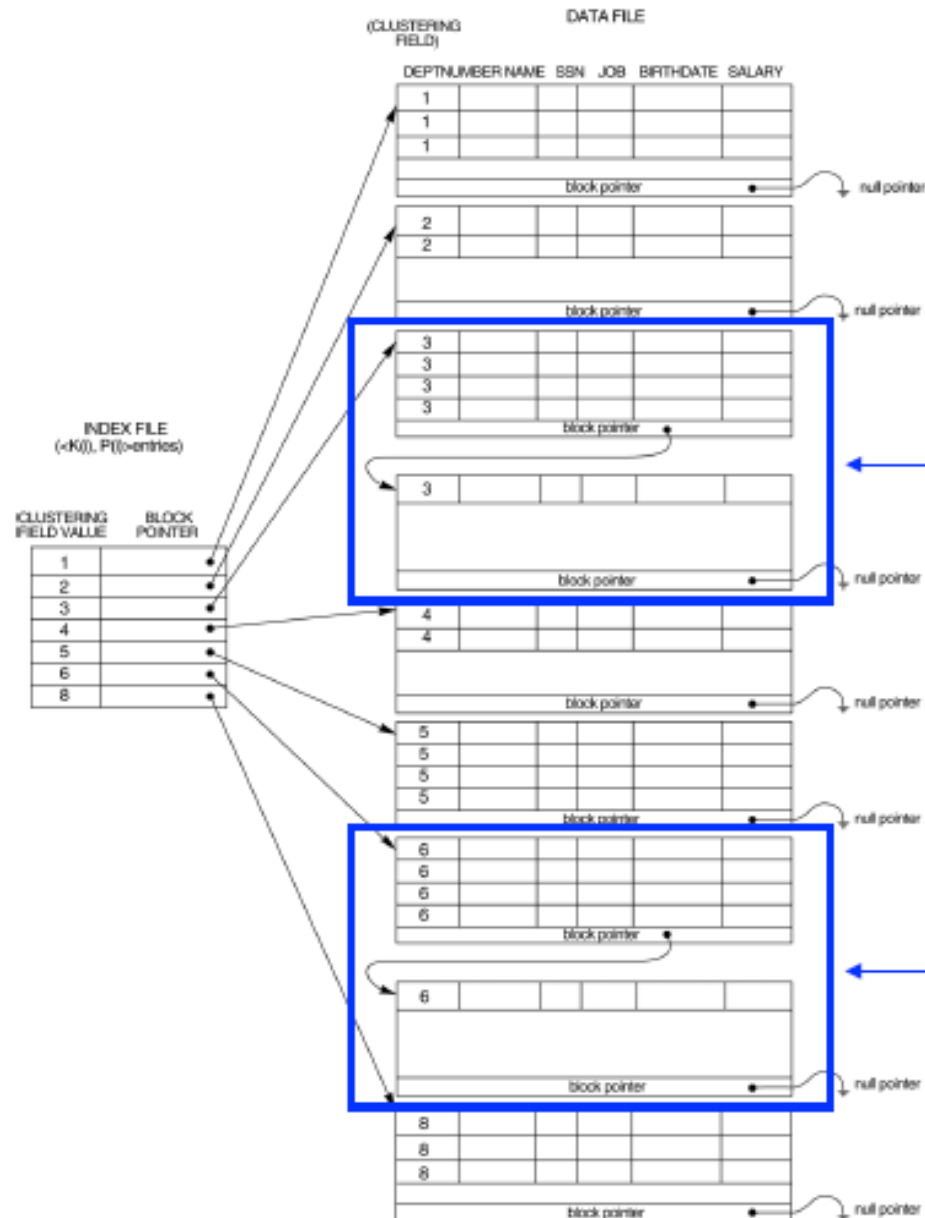
- Problemas

- 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

Desempenho em Acesso a Disco

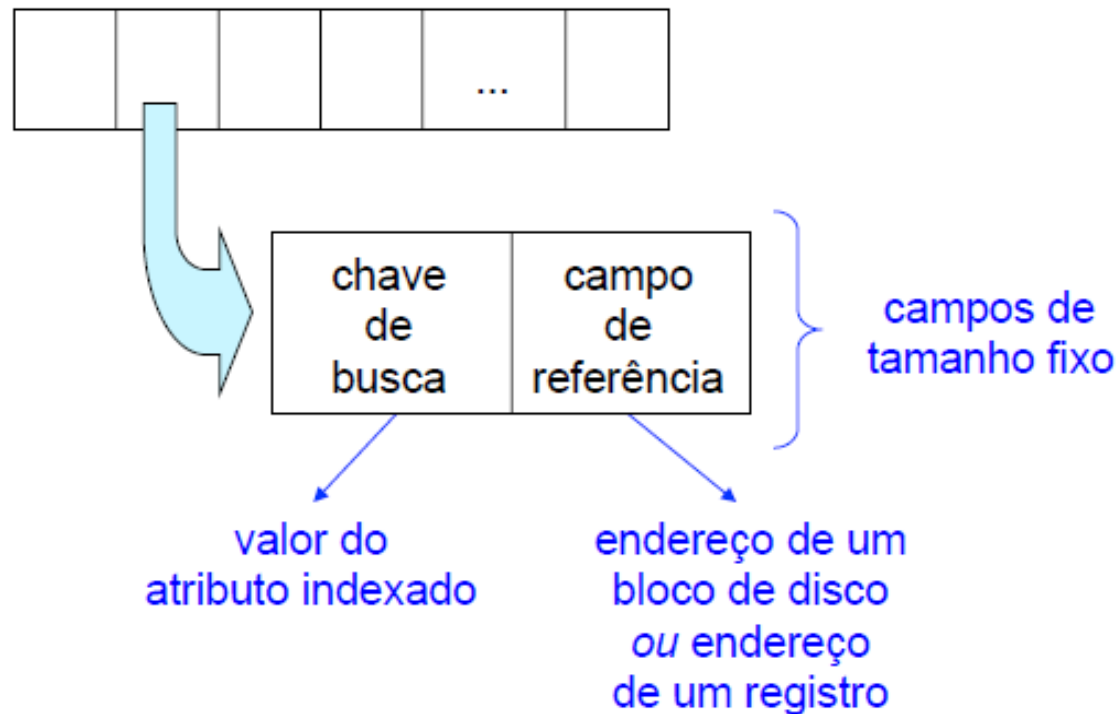
- Índice de agrupamento
 - o desempenho tende a ser ligeiramente pior que o desempenho de um índice primário
 - a repetição de valores pode conduzir à leitura de mais de uma página no arquivo de dados
 - No Exemplo, o desempenho para o campo indexado com valor 3 corresponde a:
 $\log_2 n + 2 = \log_2 45 = 6 + 2 = 8$

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

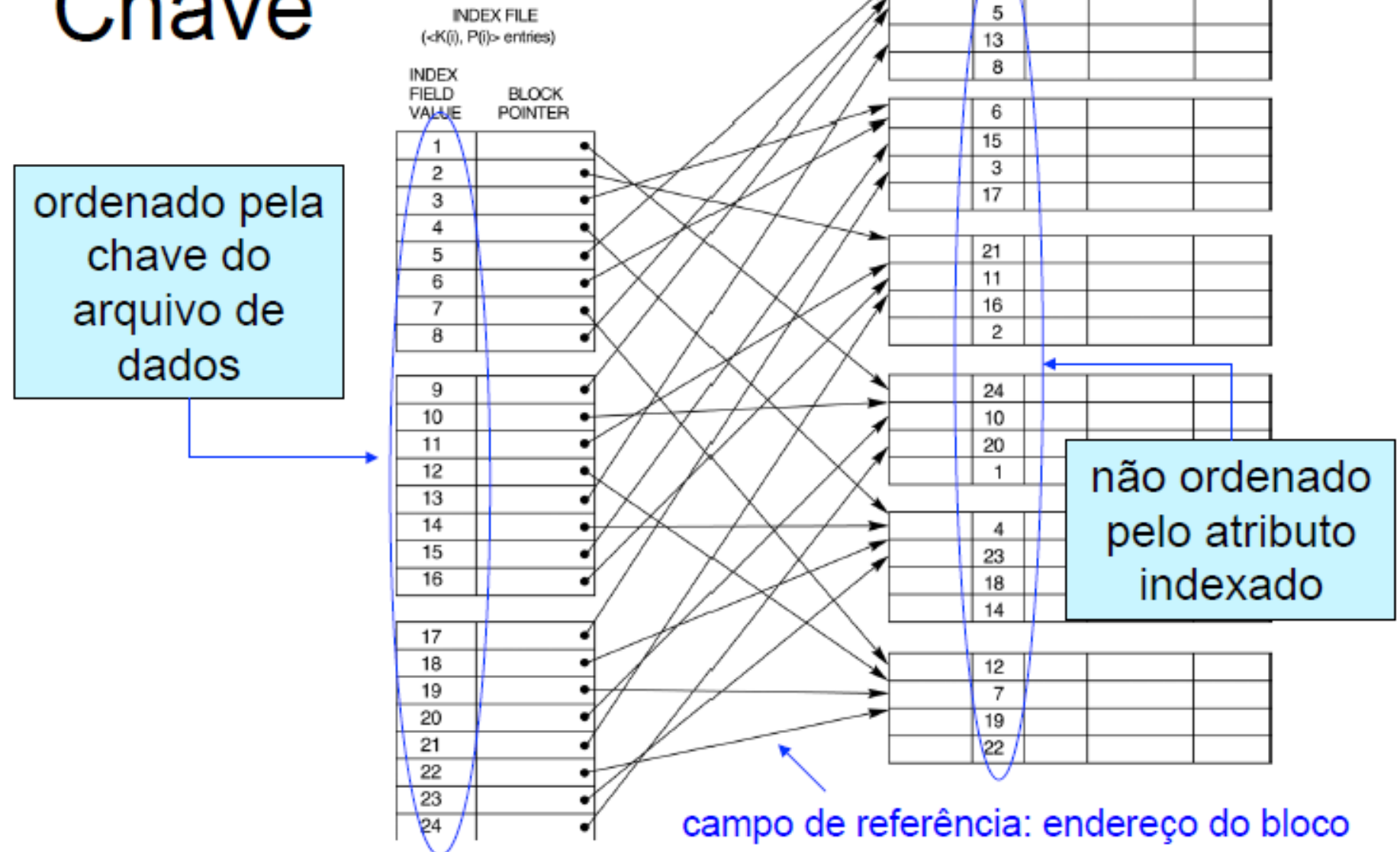
Índice Secundário - Chave

- 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 na 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 Índices

- 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

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

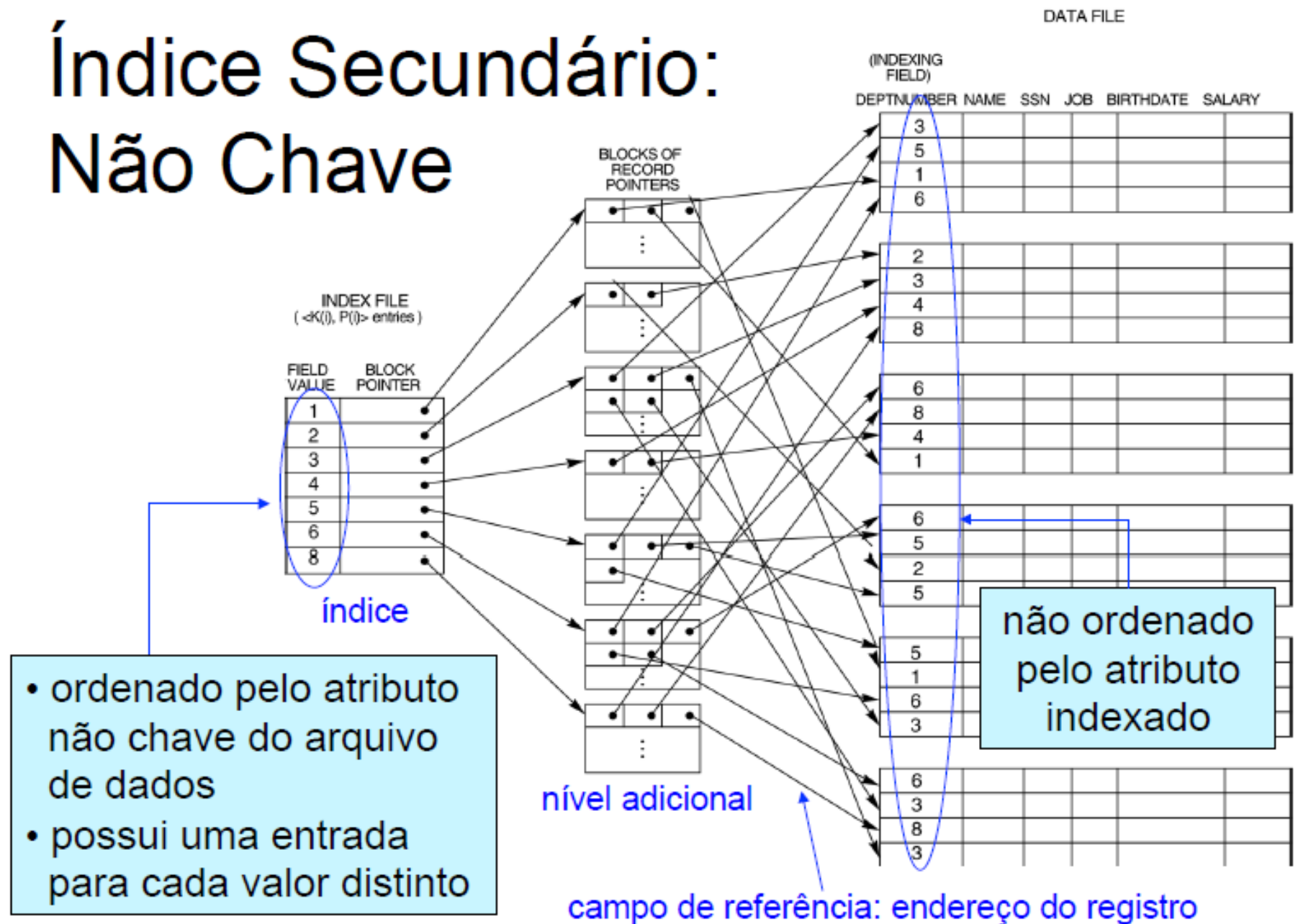
Índice Secundário: Não Chave

- 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

Índice Secundário: Não Chave



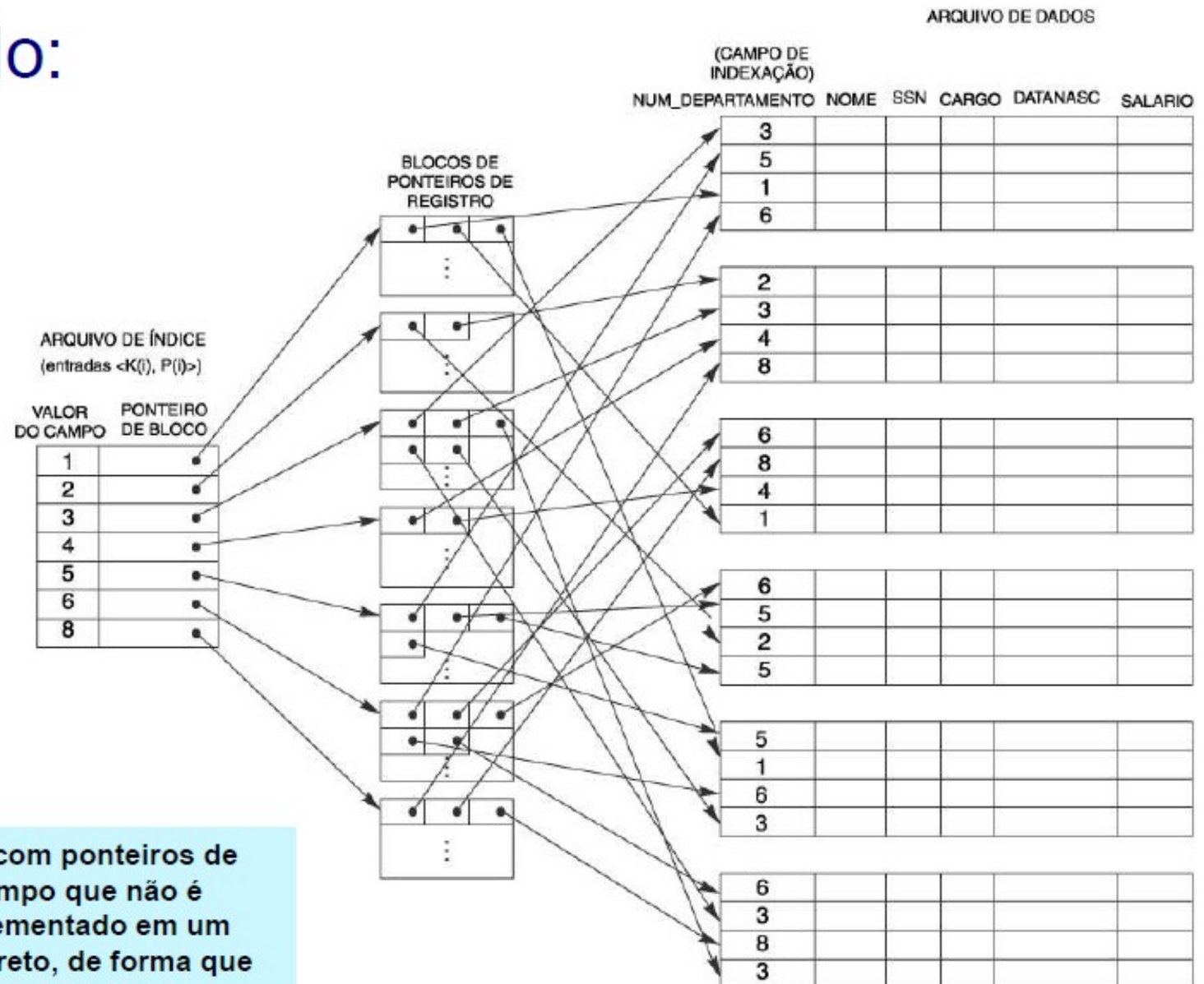
Número de Índices

- 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

Propriedades dos Índices

| Tipo de Índice | Número de Entradas | Denso ou Esparso |
|------------------------|---|------------------|
| Primário | Número de blocos no arquivo de dados | Esparso |
| Agrupamento | Número de valores distintos do campo de indexação | Esparso |
| Secundário (chave) | Número de registros no arquivo de dados | Denso |
| Secundário (não chave) | Número de registros no arquivo de dados ou | Denso |
| | Número de valores distintos do campo de indexação | Esparso |

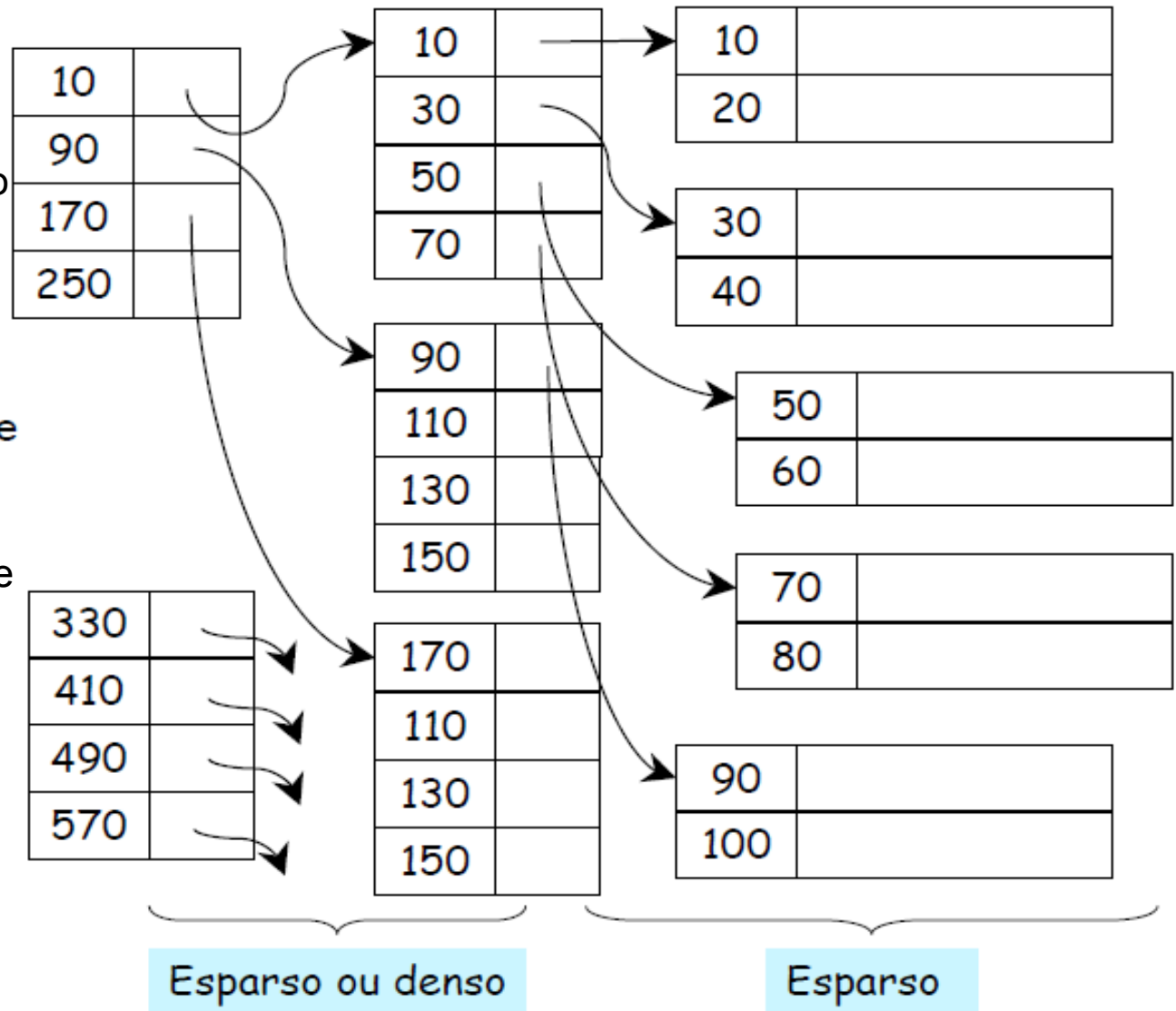
Exemplo:



Índice secundário (com ponteiros de registro), em um campo que não é campo-chave, implementado em um nível adicional, indireto, de forma que as entradas de índice sejam de tamanho fixo e possuam valores de campo únicos.

Múltiplos Níveis

- Motivação: se o arquivo de índices se torna muito grande para ser armazenado em bloco de disco, é interessante indexá-lo em mais de um nível
- Vantagem: índice pequeno pode ser mantido em memória e o tempo de busca é mais baixo
- Desvantagem: muitos níveis de índices podem aumentar a complexidade do sistema



Múltiplos Níveis

- Um índice multinível é um “Índice de índice”.
- Primeiro nível: arquivo ordenado pela chave de indexação, valores distintos, entradas de tamanho fixo.
- Demais níveis: índice primário sobre o índice do nível anterior e assim sucessivamente até que no último nível o índice ocupe apenas um bloco.
- Número de acessos a bloco: um a cada nível de índice, mais um ao bloco do arquivo de dados