ESTRUTURAS DE INDEXAÇÃO

ÍNDICES DE 1 NÍVEL

Rodrigo Salvador Monteiro

Slides adaptados do Prof Sean Siqueira (sean@uniriotec.br)

Índices como Caminhos de Acesso

- índice de um-nível
  - arquivo auxiliar que torna a busca de um registro no arquivo de dados mais eficiente.
- O índice é geralmente especificado em um campo do arquivo (embora possa ser especificado em vários campos)
- forma de um índice
  - arquivo de entradas **<valor do campo, ponteiro para o registro>,** ordenado pelo valor do campo
- O índice é chamado de *caminho de acesso* para o campo.

Índices como Caminhos de Acesso

- O arquivo de índice geralmente ocupa consideravelmente menos blocos de disco do que o arquivo de dados
  - entradas muito menores
- Índices densos x esparsos.
  - **índice denso**
    - uma entrada de índice para *cada valor de chave de busca* (e portanto todo registro) no arquivo de dados.
  - **índice esparso** (ou **não-denso**)
    - entradas de índices para apenas alguns dos valores de busca.

Tipos de índice (de um nível)

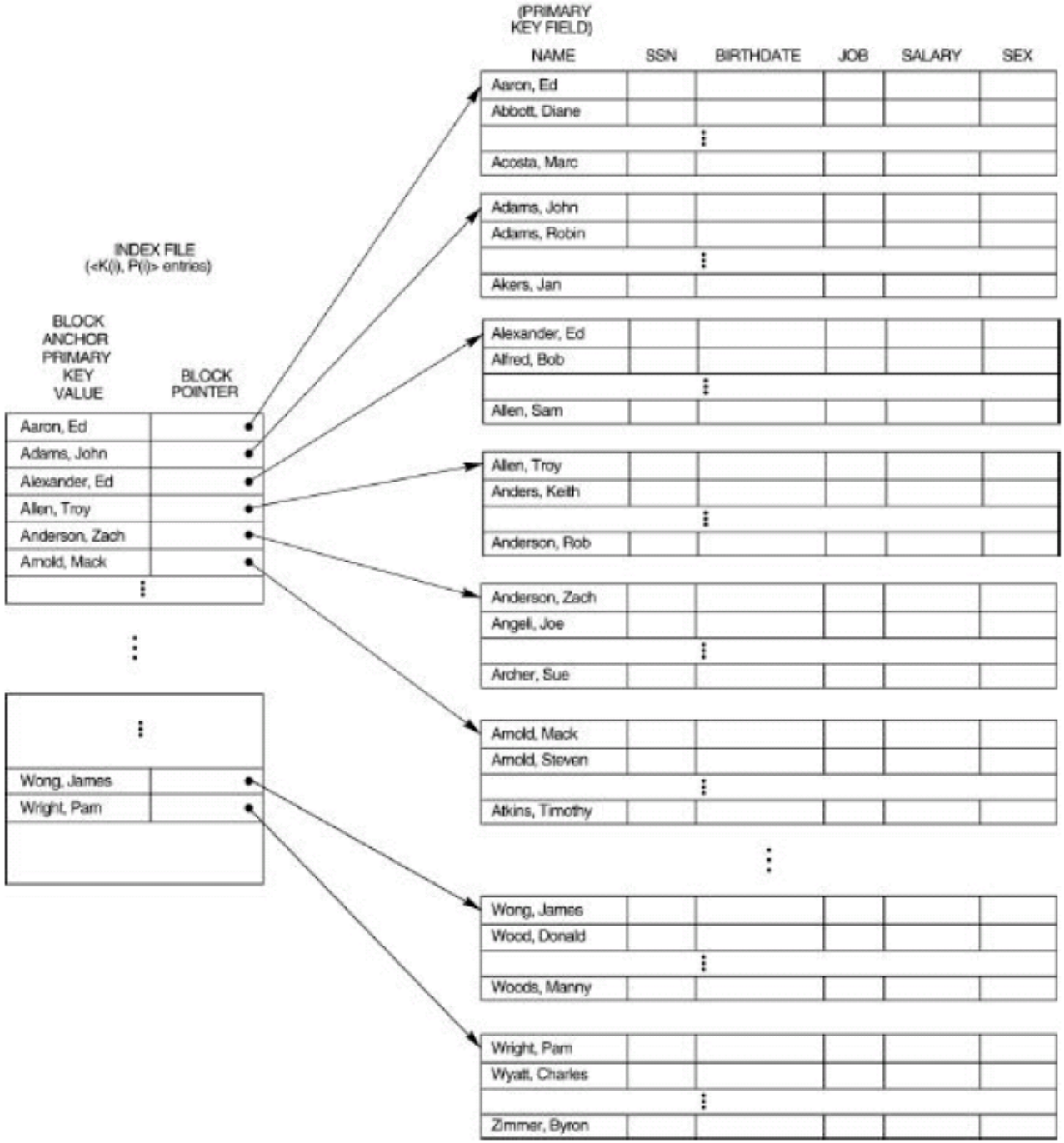
- Índices ordenados
  - Índice principal
    - É especificado no campo chave de ordenação de um arquivo ordenado.
    - campo chave de ordenação
      - é utilizado para ordenar fisicamente os registros de arquivos em disco
      - todo registro possui um valor exclusivo para aquele campo
  - Índice clustering
    - É especificado em um campo de ordenação que não é um campo chave
      - inúmeros registros no arquivo podem possuir o mesmo valor para o campo de ordenação
  - Um arquivo pode possuir no máximo um índice principal ou um índice clustering (não ambos)
    - campo de ordenação física único

Tipos de índice (de um nível)

- Índices não-ordenados
  - Índice secundário
    - É especificado em qualquer campo não-ordenado de um arquivo.
    - Um arquivo pode possuir diversos índices secundários além de seu método de acesso principal.

Índice Principal

- arquivo ordenado cujos registros são de tamanho fixo com dois campos
  - <K(i), P(i)>
  - K(i) é do mesmo tipo de dado que o campo chave de ordenação (chamado de **chave primária**)
  - P(i) é um ponteiro para um bloco do disco (um endereço de bloco)
- Existe uma entrada de índice (ou registro de índice) no arquivo índice para cada bloco no arquivo de dados.
- Cada entrada de índice possui o valor do campo chave primária para o primeiro registro num bloco e um ponteiro para aquele bloco como seus dois valores de campo.
- O número total de entradas no índice é o mesmo que o número de blocos de discos no arquivo ordenado de dados.
- Um índice principal é um índice não-denso (ou esparso).


(PRIMARY KEY FIELD)
NAME
SSN
BIRTHDATE
JOB
SALARY
SEX
INDEX FILE (<K(i), P(i)> entries)
BLOCK ANCHOR PRIMARY KEY VALUE
BLOCK POINTER
Aaron, Ed
Adams, John
Alexander, Ed
Allen, Troy
Anderson, Zach
Arnold, Mack
Wong, James
Wright, Pam
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

Índices como Caminhos de Acesso

Exemplo: Dado o seguinte arquivo ordenado de dados:

- EMPREGADO(Nome, NSS, Endereço, Profissão, Sal, ...)

Suponha que:

- Tamanho do registro R=100 bytes (tamanho fixo e não espalhado)
- Tamanho do bloco B=1024 bytes
- r=30000 registros

Perguntas:

- Calcule o bfr (blocking factor = número de registros por bloco)

  $Bfr = \lfloor (B/R) \rfloor = 1024/100 = 10$ registros por bloco

- Quantos blocos são necessários para o arquivo?

  $b = \lceil r/Bfr \rceil = \lceil 30000/10 \rceil = 3000$ blocos

- Quantos acessos serão necessários para buscar um determinado valor?

  $\lceil \log_2 b \rceil = \lceil \log_2 3000 \rceil = 12$ acessos a blocos

Índices como Caminhos de Acesso

Suponha que:

- Campo chave de ordenação seja de tamanho V = 9 bytes
- Ponteiro de bloco seja de tamanho P = 6 bytes
- Construímos um índice principal para o arquivo

Perguntas:

- Qual é o tamanho de cada entrada de índice?

  $R_i = (9+6) = 15$ bytes
- Qual é o fator de bloco para o índice?

  $Bfr_i = \lfloor (B/R_i) \rfloor = \lfloor 1024/15 \rfloor = 68$ entradas por bloco
- Qual é o número total de entradas de índice?

  $r_i = b = 3000$
- Quantos blocos de índice são necessários?

  $b_i = \lceil r_i / Bfr_i \rceil = \lceil 3000/68 \rceil = 45$ blocos
- Para realizar uma busca no arquivo de índice quantos acessos seriam necessários?

  $\lceil Log_2 b_i \rceil = \lceil Log_2 45 \rceil = 6$ acessos a blocos
- Para pesquisar um registro usando o índice, quantos acessos seriam necessários?

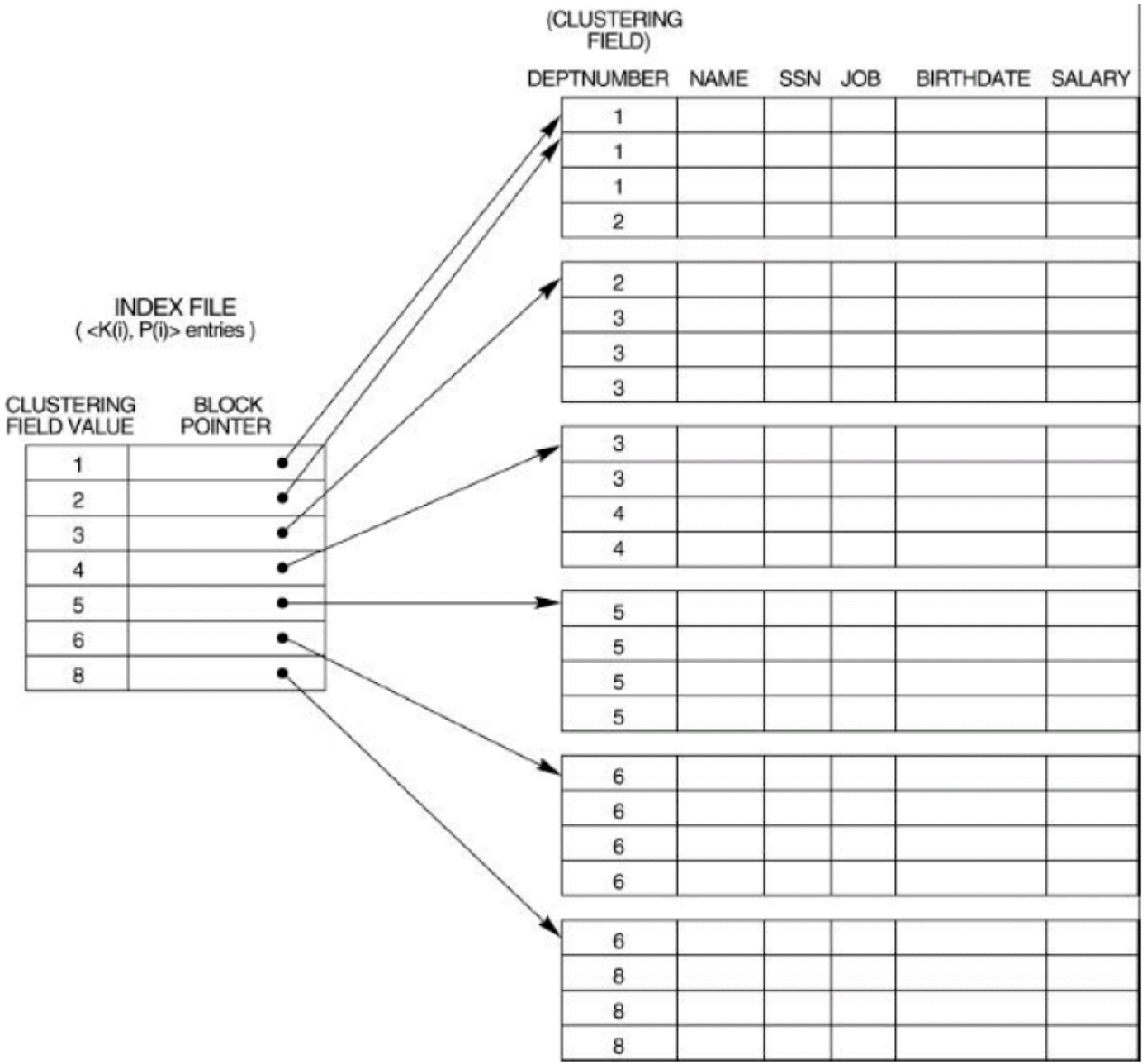
  Acessos ao índice + acesso ao bloco do arquivo de dados = 6 +1 = 7 acessos

Problema com índice principal

- Inclusão e exclusão de registros
  - Inserir um registro em sua posição correta no arquivo de dados implica em movimentar registros para abrir espaço para o novo registro e alterar algumas entradas de índices.
  - Pode-se utilizar um arquivo de overflow para reduzir esse problema.
  - A exclusão é manipulada através do uso de indicadores de exclusão.

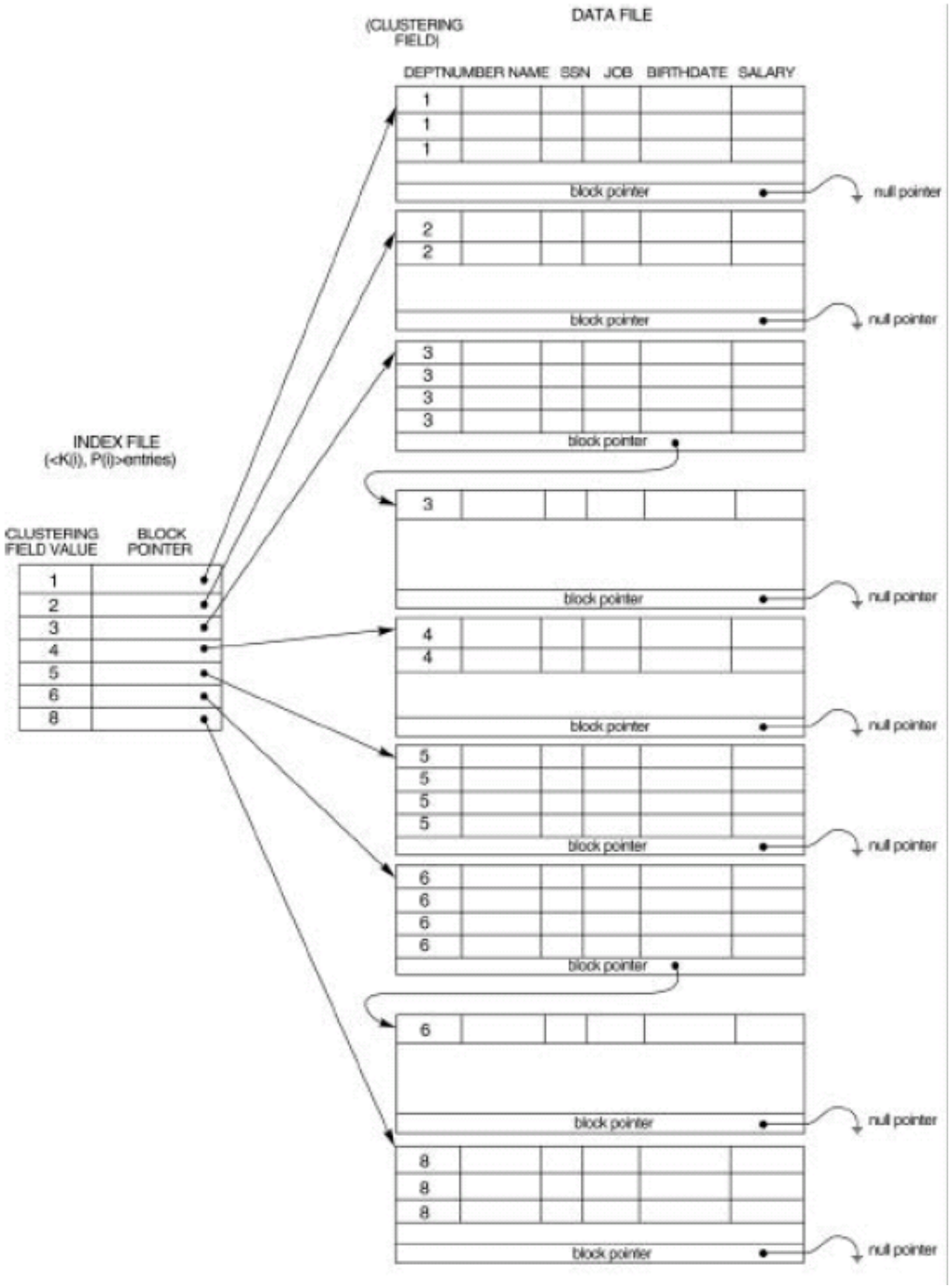
Índices por Clustering

- Os registros de um arquivo estão fisicamente ordenados por um campo que não seja chave.
- Também é um arquivo ordenado com dois campos
  - O primeiro campo é do mesmo tipo do campo clustering do arquivo de dados
  - O segundo campo é um ponteiro para o bloco
- Existe uma entrada no índice clustering para cada valor distinto do campo clustering, que contém o valor e um ponteiro para o primeiro bloco no arquivo de dados que possua um registro com aquele valor para seu campo clustering.
- É um índice não-denso (ou esparso).


(CLUSTERING FIELD)
DEPTNUMBER
NAME
SSN
JOB
BIRTHDATE
SALARY
INDEX FILE
(<K(i), P(i)> entries)
CLUSTERING FIELD VALUE
BLOCK POINTER
1
2
3
4
5
6
8

Problema com índice por clustering

- Inclusão e exclusão de registros
  - Os registros de dados estão fisicamente ordenados
  - É comum reservar um bloco inteiro (ou conjunto de blocos contíguos) para cada valor do campo clustering; todos os registros com aquele valor são posicionados no bloco (ou conjunto de blocos).

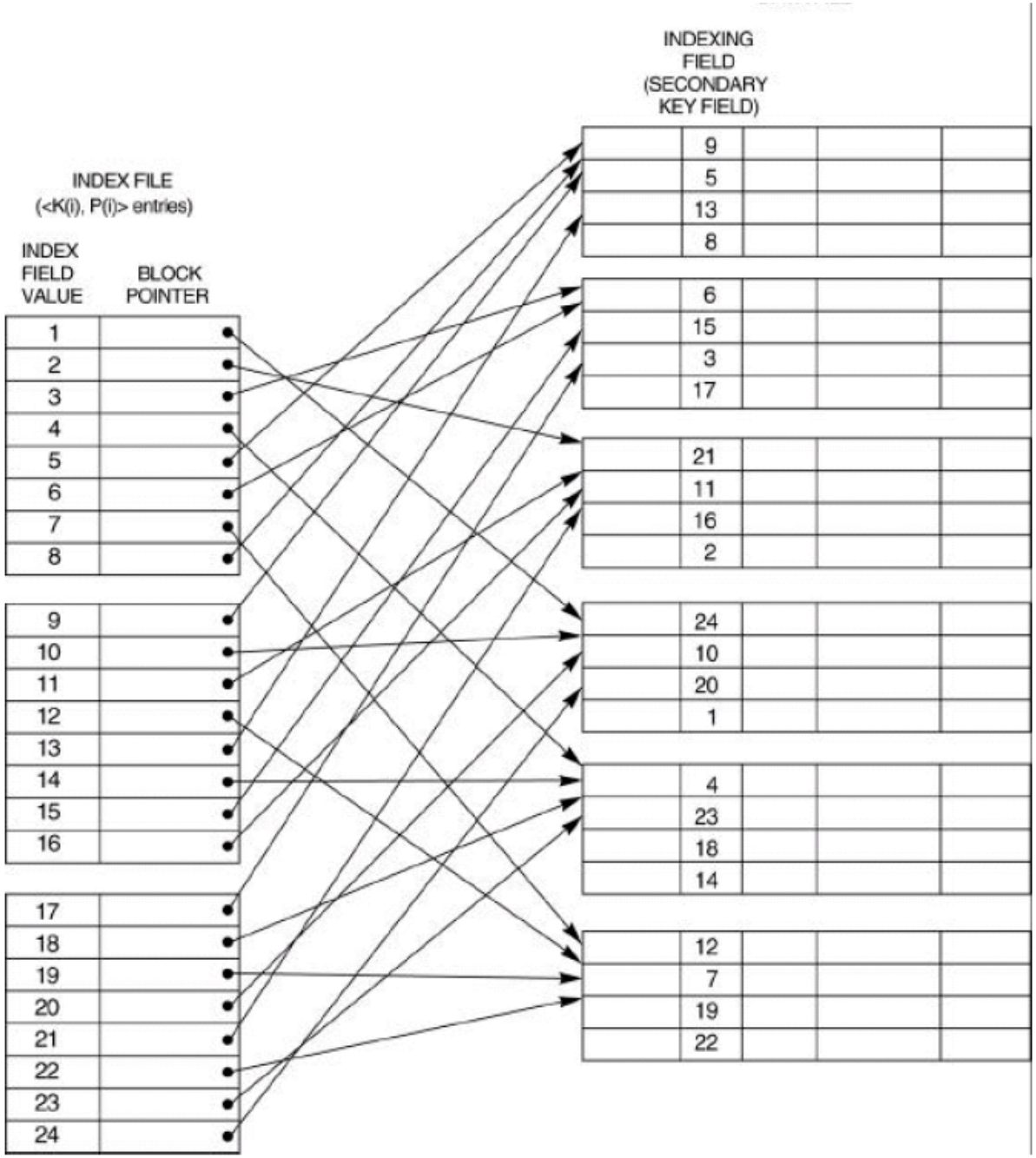

DATA FILE
(CLUSTERING FIELD)
DEPTNUMBER NAME SSN JOB BIRTHDATE SALARY
block pointer
null pointer
INDEX FILE
(<K(i), P(i)>entries)
CLUSTERING FIELD VALUE
BLOCK POINTER
1
2
3
4
5
6
8

Pergunta

- Qual a diferença entre índice por clustering e hashing?
  - Uma pesquisa num índice clustering utiliza os valores do próprio campo de pesquisa enquanto que uma pesquisa num índice hash utiliza o valor calculado através da aplicação da função hash no campo de pesquisa.

Índices Secundários

- Fornece um meio secundário de acesso a um arquivo para o qual já existe algum acesso primário;
- O índice secundário pode ser utilizado sobre
  - um campo que é uma chave candidata e possui um valor único em cada registro, ou
  - um campo que não é chave e tem valores duplicados;


INDEXING FIELD (SECONDARY KEY FIELD)
INDEX FILE (<K(i), P(i)> entries)
INDEX FIELD VALUE
BLOCK POINTER
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
9
5
13
8
6
15
3
17
21
11
16
2
24
10
20
1
4
23
18
14
12
7
19
22

Índices Secundários

- Os índices secundários tem que ser densos
  - (relembrando) com uma entrada para cada valor de chave de procura e um ponteiro para cada registro do arquivo
  - Por quê?
    - Se um índice secundário armazenar apenas alguns dos valores da chave de procura, os registros com valores intermediários da chave de procura podem estar em qualquer lugar do arquivo, e neste caso, não é possível localizá-lo sem procurar em todo o arquivo;

Índices como Caminhos de Acesso

Considere o arquivo do exemplo anterior com r = 30000 registros de tamanho fixo R = 100 bytes armazenados em um disco com tamanho de bloco B = 1024 bytes.

- O arquivo possui b = 3000 blocos (calculado anteriormente)
- Para realizar uma pesquisa linear no arquivo precisaríamos de em média b/2 = 3000/2 = 1500 acessos a blocos

Suponha que:

- Construímos um índice secundário num campo não-ordenado chave do arquivo
- V = 9 bytes, P = 6 bytes

Perguntas:

- Calcule o tamanho de cada entrada R_i de índice
  $R_i = (9+6) = 15$ bytes
- Calcule o fator de bloco para o arquivo
  $Bfr_i = \lfloor (B/R_i) \rfloor = \lfloor 1024/15 \rfloor = 68$ entradas por bloco
- Quantas entradas de índice são necessárias?
  r_i = número de registros no arquivo de dados (r) = 30000

Índices como Caminhos de Acesso

Perguntas (cont.):

- Quantos blocos são necessários para o índice?

  $b_i = \lceil r_i / Bfr_i \rceil = \lceil 30000/68 \rceil = 442$ blocos

- Quantos acessos serão necessários para buscar um determinado valor no índice?

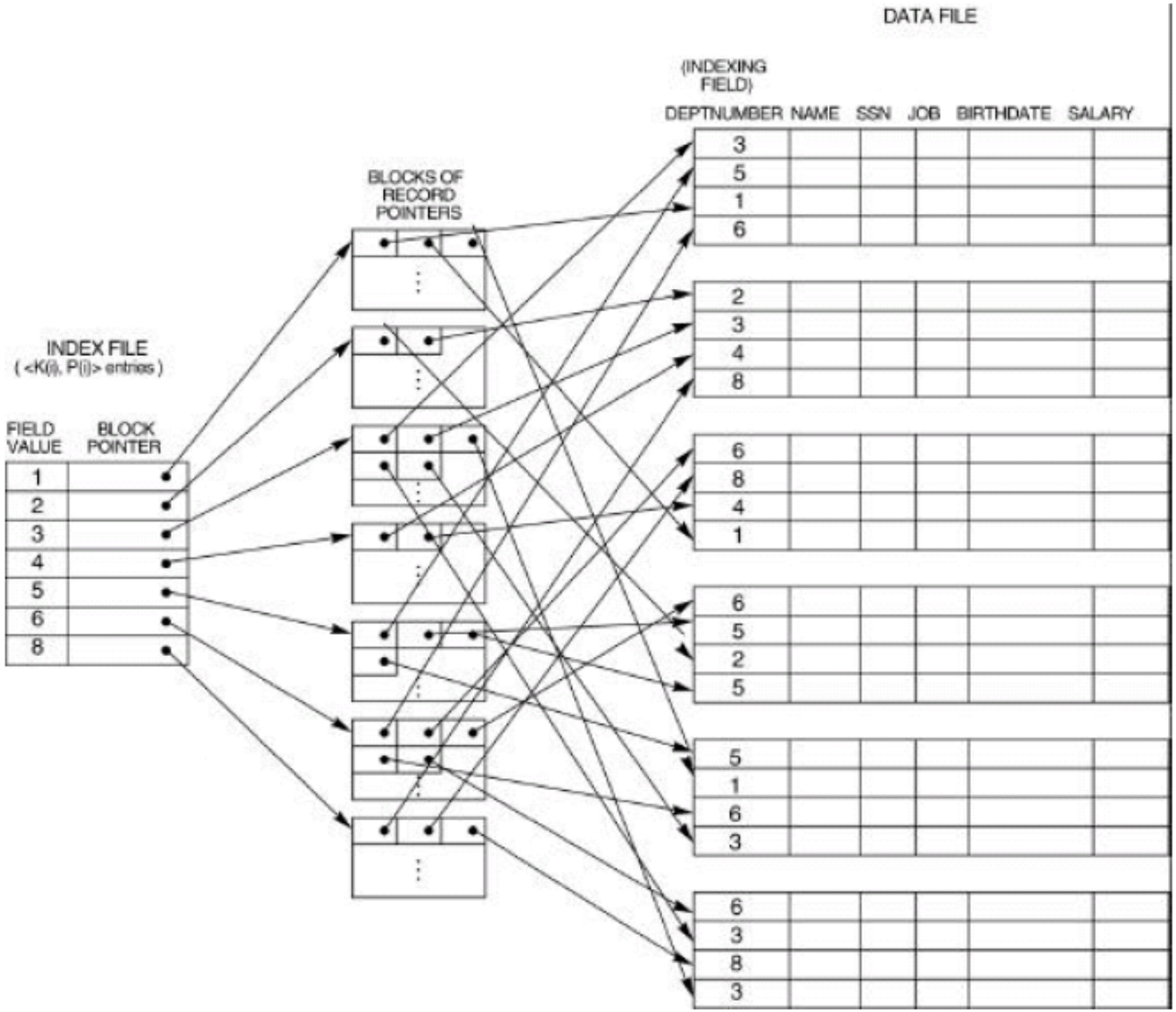
  $\lceil \text{Log}_2 b_i \rceil = \lceil \text{Log}_2 442 \rceil = 9$ acessos a blocos

- Quantos acessos são necessários para pesquisar um determinado registro utilizando o índice?

  acessos ao índice + acesso ao bloco de dados = 9+1 = 10

Índices Secundários

- Podemos criar um índice secundário em um campo que não seja chave? Neste caso, como implementar este índice?
  - Opção 1: Incluir diversas entradas de índice com o mesmo valor K(i) – uma para cada registro
  - Opção 2: Possuir registros de tamanho variável para as entradas de índice, com um campo de repetição para os ponteiros. Mantemos uma lista de ponteiros <P(i,1), ..., P(i,k)> na entrada de índice para K(i)
  - Opção 3: Manter as entradas de índice num tamanho fixo e ter uma única entrada para cada valor de campo de indexação, criando um nível adicional de acesso indireto para lidar com os diversos ponteiros. Os ponteiros não apontam diretamente para o arquivo, mas sim para um bucket que contém ponteiros para o arquivo de dados.


DATA FILE
(INDEXING FIELD)
DEPTNUMBER
NAME
SSN
JOB
BIRTHDATE
SALARY
BLOCKS OF RECORD POINTERS
INDEX FILE
(<K(i), P(i)> entries)
FIELD VALUE
BLOCK POINTER
1
2
3
4
5
6
8
3
5
1
6
2
3
4
8
6
8
4
1
6
5
2
5
5
1
6
3
6
3
8
3

Índices Secundários

Vantagens e desvantagens

- Os índices secundários melhoram o desempenho das consultas, mas impõe sobrecarga significativa na atualização do BD;
- Quem decide é o projetista do BD
- Algumas heurísticas
  - freqüência de consultas sobre a tabela > frequencia de atualizações sobre a tabela
  - Colunas frequentemente mencionadas nas cláusulas where das consultas
  - Chaves estrangeiras para tabelas com cardinalidade grande