



**PUCRS**  
Pontifícia Universidade Católica  
do Rio Grande do Sul

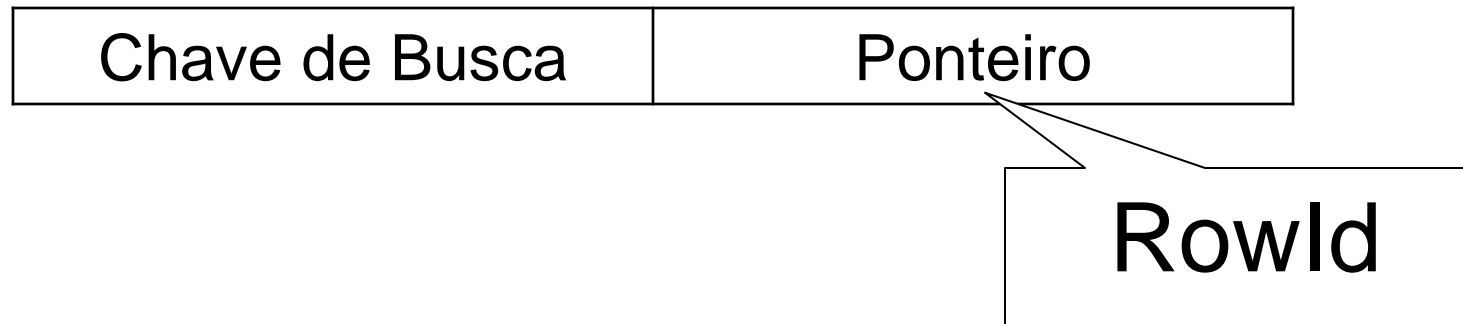
ESCOLA  
**POLITÉCNICA**

# Estruturas de Armazenamento e Indexação

98H00-04 - Infraestrutura para Gestão de Dados

Prof. Msc. Eduardo Arruda  
[eduardo.arruda@pucrs.br](mailto:eduardo.arruda@pucrs.br)

- Estruturas utilizadas para agilizar a recuperação de dados
- Exemplo
  - Catálogo de autores em uma biblioteca
- Chave de Busca
  - atributo ou conjunto de atributos para pesquisar registros em um arquivo
- Um índice consiste em registros (entradas de índice) na forma



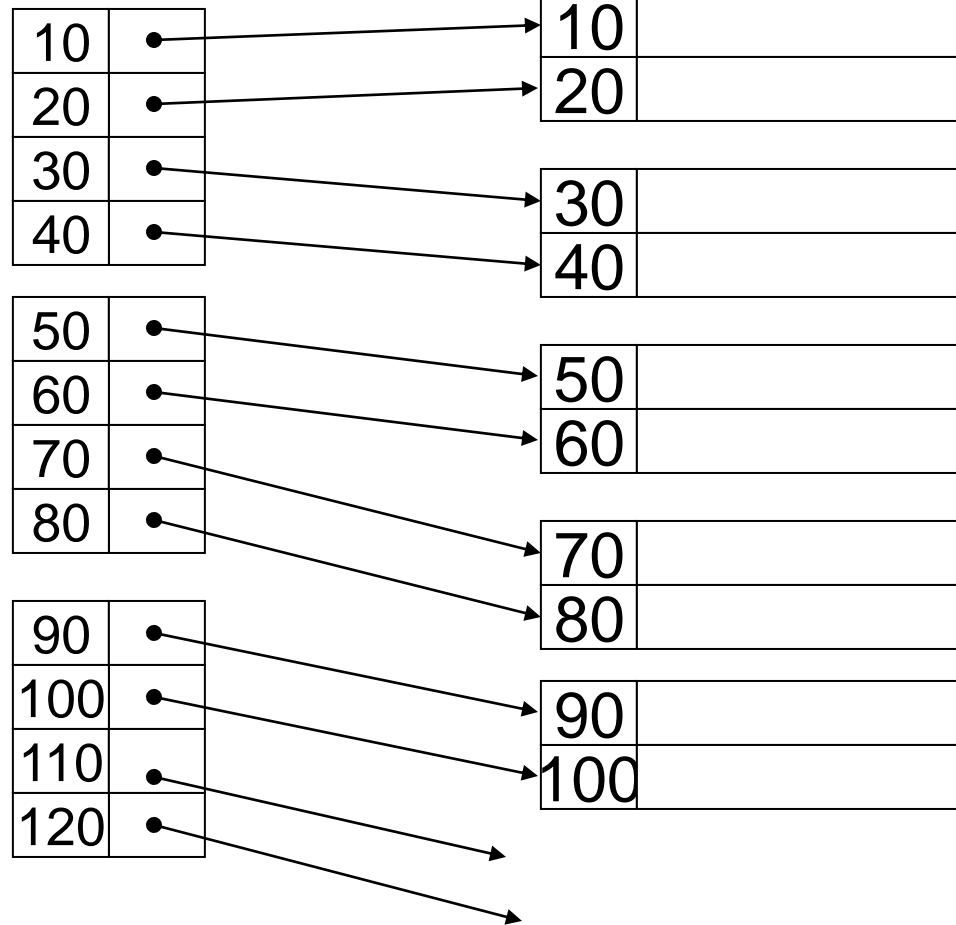
- Diversas classificações possíveis:
  - Denso ou Esparso
  - Primário ou Secundário
  - *Clustered* ou *Non-Clustered*
  - Nível Único ou Multinível

- Densos
  - Possui uma entrada no índice para cada valor da chave de busca, que aponta para o primeiro registro que possui aquele valor
- Esparsos
  - Apenas o primeiro registro de cada bloco do arquivo de dados é representado no índice

- Sequência de blocos contendo todos os valores da chave de busca dos registros e os ponteiros para os próprios registros
- Mantém os valores da chave de busca no índice em ordem
  - Permite a utilização de algoritmos otimizados de busca
  - O arquivo de dados não precisa estar em ordem
- O tamanho do índice é menor dos dados em si
  - Pode caber todo no buffer
  - Com isto, é possível encontrar qualquer registro, dado o valor de sua chave de busca, com poucas operações de E/S

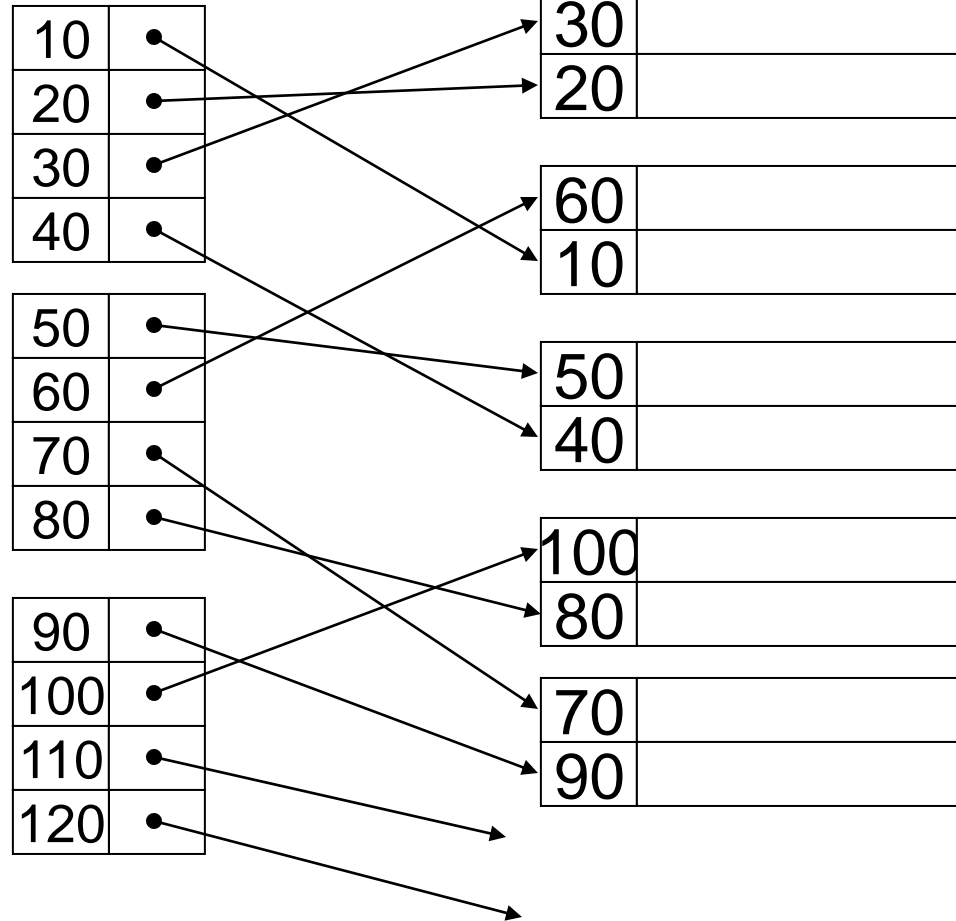
## Índice Denso

## Arquivo de Dados Ordenado



## Índice Denso

## Arquivo de dados Não Ordenado

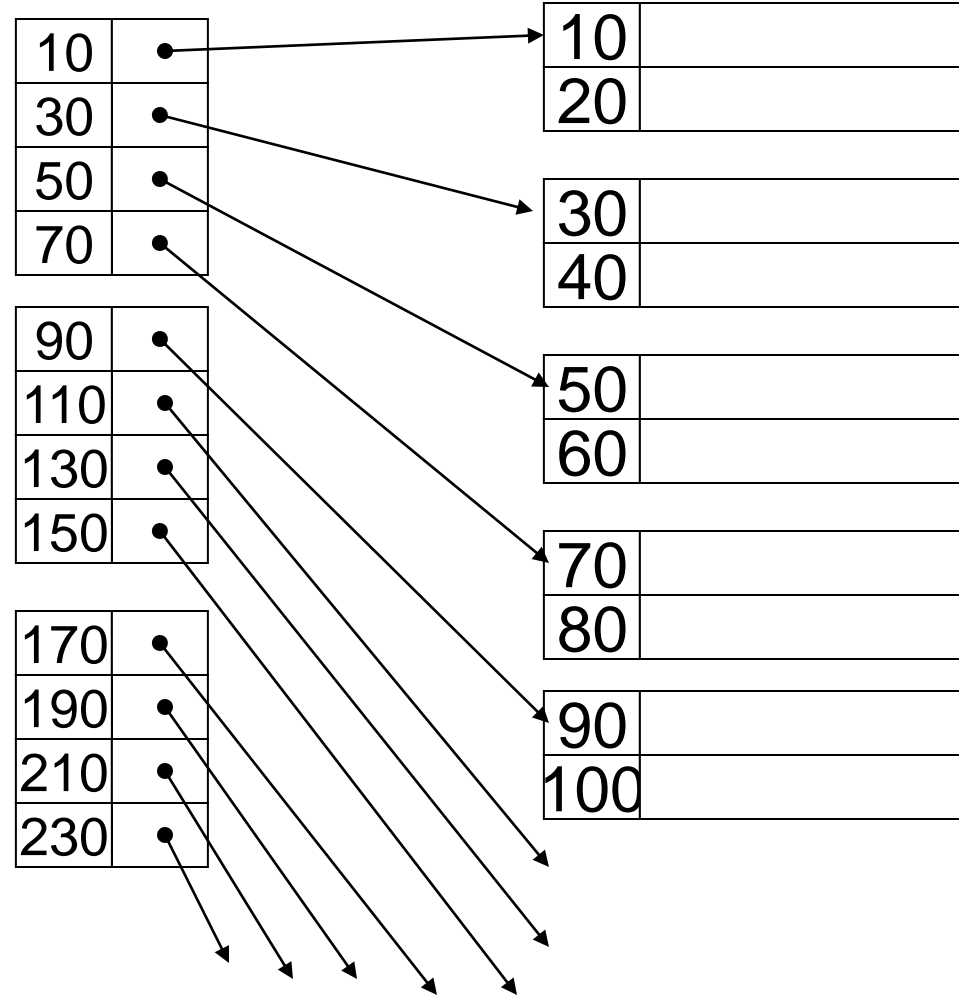


- Contém apenas um par chave-ponteiro por bloco de dados
  - O valor da chave de busca corresponde ao primeiro registro do bloco
- Utiliza menos espaço, porém o custo para encontrar um registro, dado o valor de sua chave de busca, é um pouco maior
- O arquivo de dados deve estar ordenado pelo campo indexado, para fazer sentido



## Índice Esparso

## Arquivo de Dados Ordenado

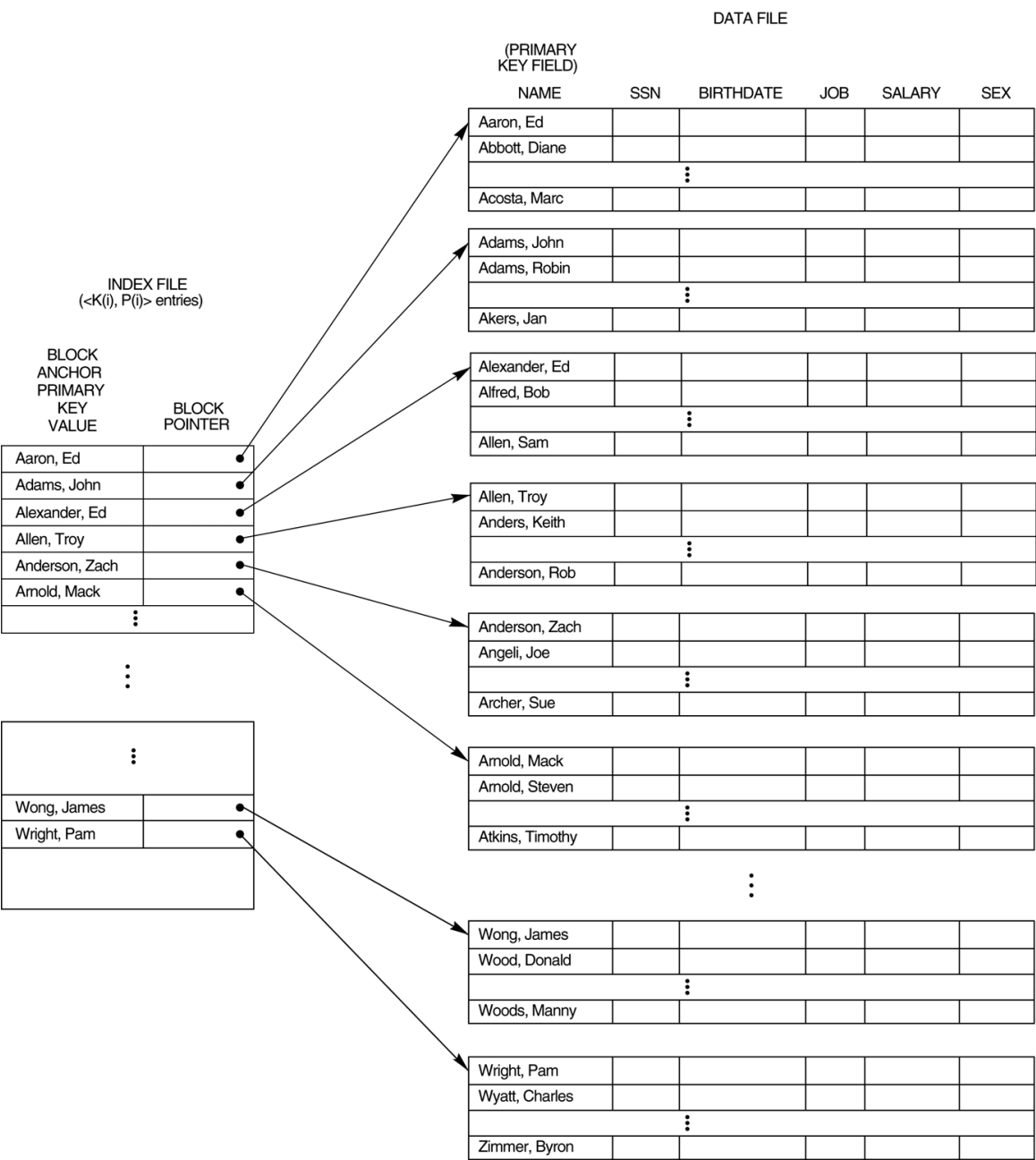


- Índice primário
  - Em um arquivo ordenado sequencialmente, o índice cuja chave de busca é única e especifica a ordem sequencial do arquivo
  - A chave de busca de um índice primário é geralmente a chave primária
  - Exemplo
    - Índice pra numEmp e arquivo de dados ordenado por numEmp
- Índice secundário
  - Índice cuja chave de busca única ou não-única especifica uma ordem diferente da ordem sequencial do arquivo
  - Exemplo
    - Índice pra CPF e arquivo de dados ordenado por codEmp ou não ordenado

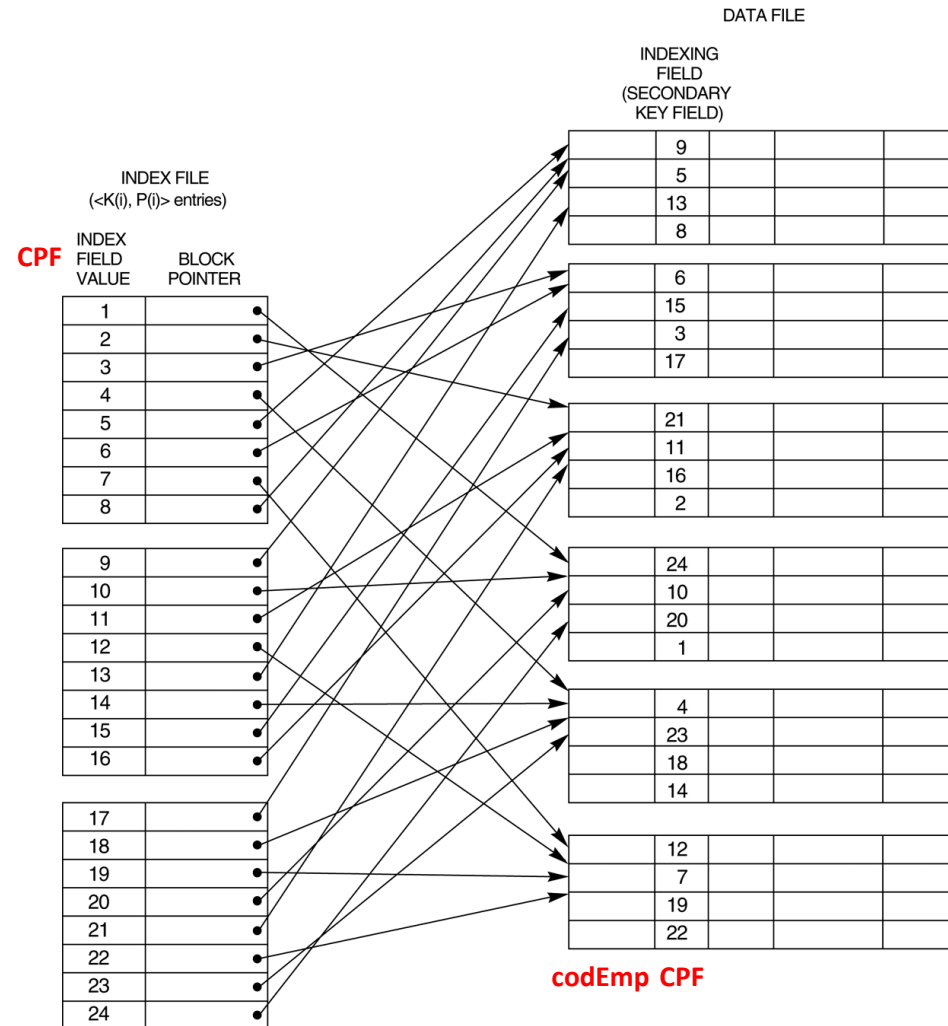
- O arquivo de dados é ordenado pelo campo chave de busca
  - Pode haver apenas um por arquivo de dados
- Um índice primário é normalmente um índice esparsos (não denso)
  - Inclui uma entrada para cada bloco do arquivo de dados e não uma entrada para cada valor de pesquisa
  - Por estar ordenado, sabe-se que o registro está em bloco específico
- Algoritmo de busca
  - Localiza o valor imediatamente inferior ao valor da chave de busca, a entrada do índice tem o valor do campo chave para o primeiro registro no bloco
  - Para buscar um registro é feita leitura sequencial dos registros do bloco ou busca binária até encontrar o registro com valor correspondente da chave.

Considerando o exemplo, alguns blocos de um arquivo ordenado (sequencial) de registros de Empregados tendo nome como campo chave de classificação

Índice primário para o campo chave de classificação



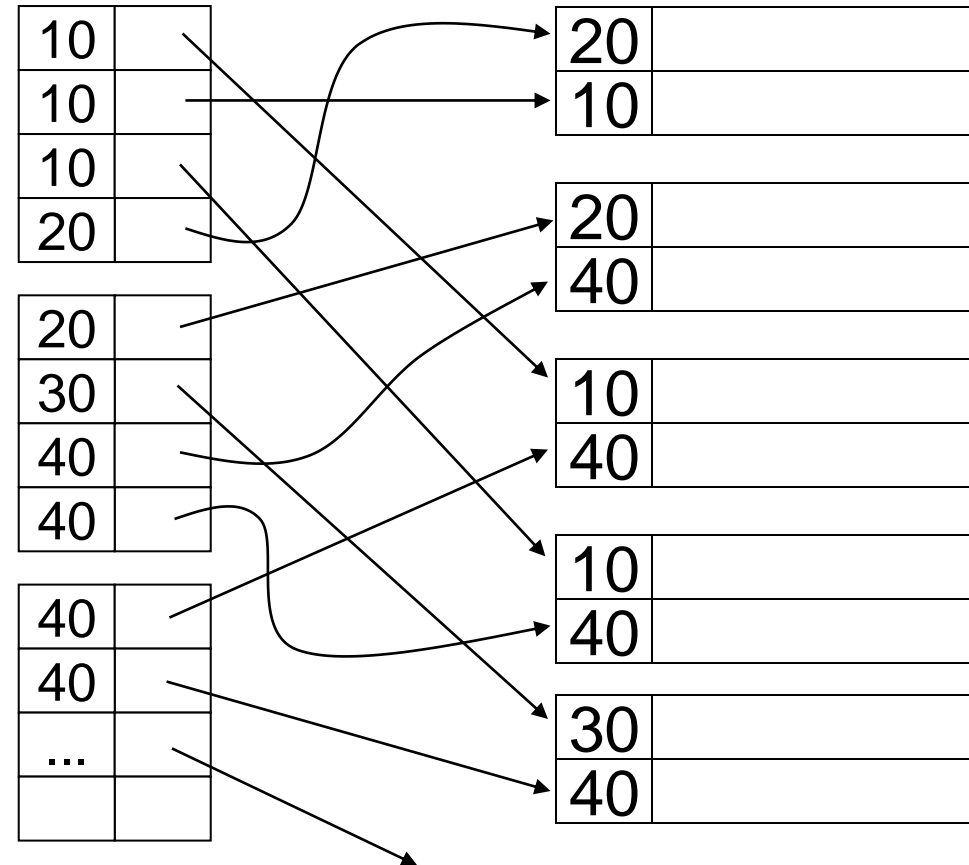
- Um índice cuja chave de busca especifica uma ordem diferente da ordem sequencial do arquivo
  - Pode haver mais de um por arquivo de dados
- Fornece um meio secundário de acessar um arquivo de dados para o qual existe ou não índice primário
- O índice secundário pode ser em um campo que é uma chave candidata e tem um valor único em cada registro, ou uma chave não-única com valores duplicados
- Inclui uma entrada para cada registro no arquivo de dados
  - Índice denso



Um índice secundário denso  
(com ponteiros de bloco) em  
um campo que não é chave de  
classificação de um arquivo

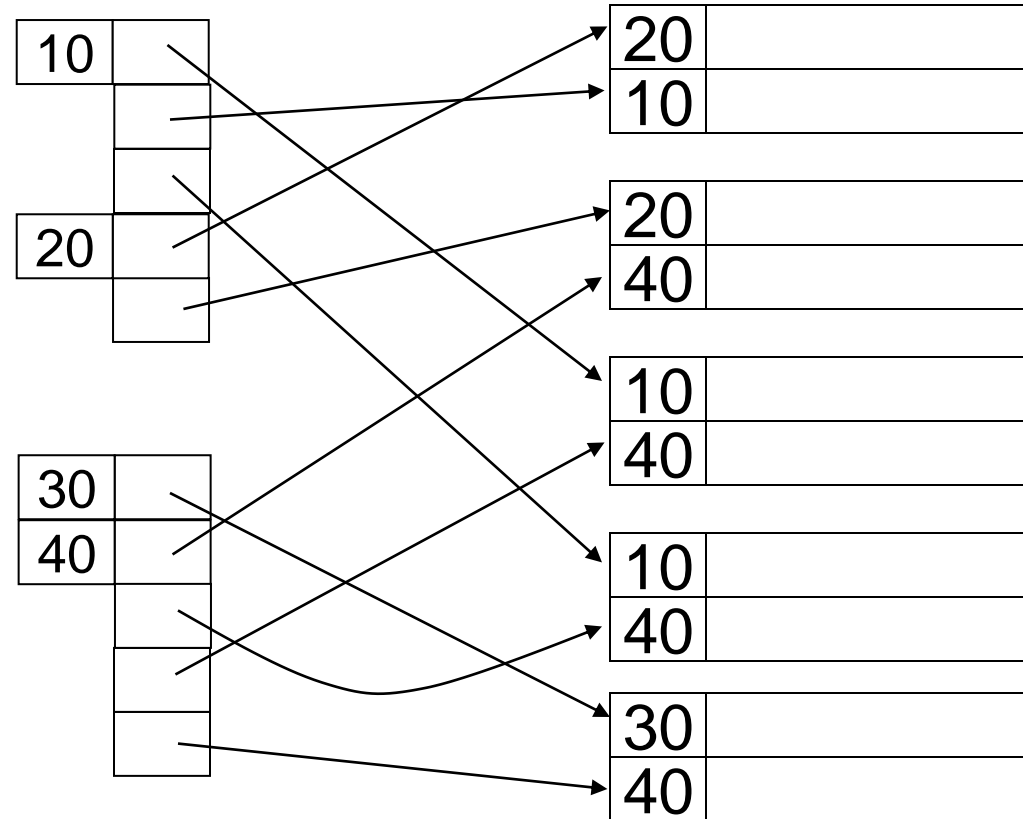
- Uma opção

Não é eficiente

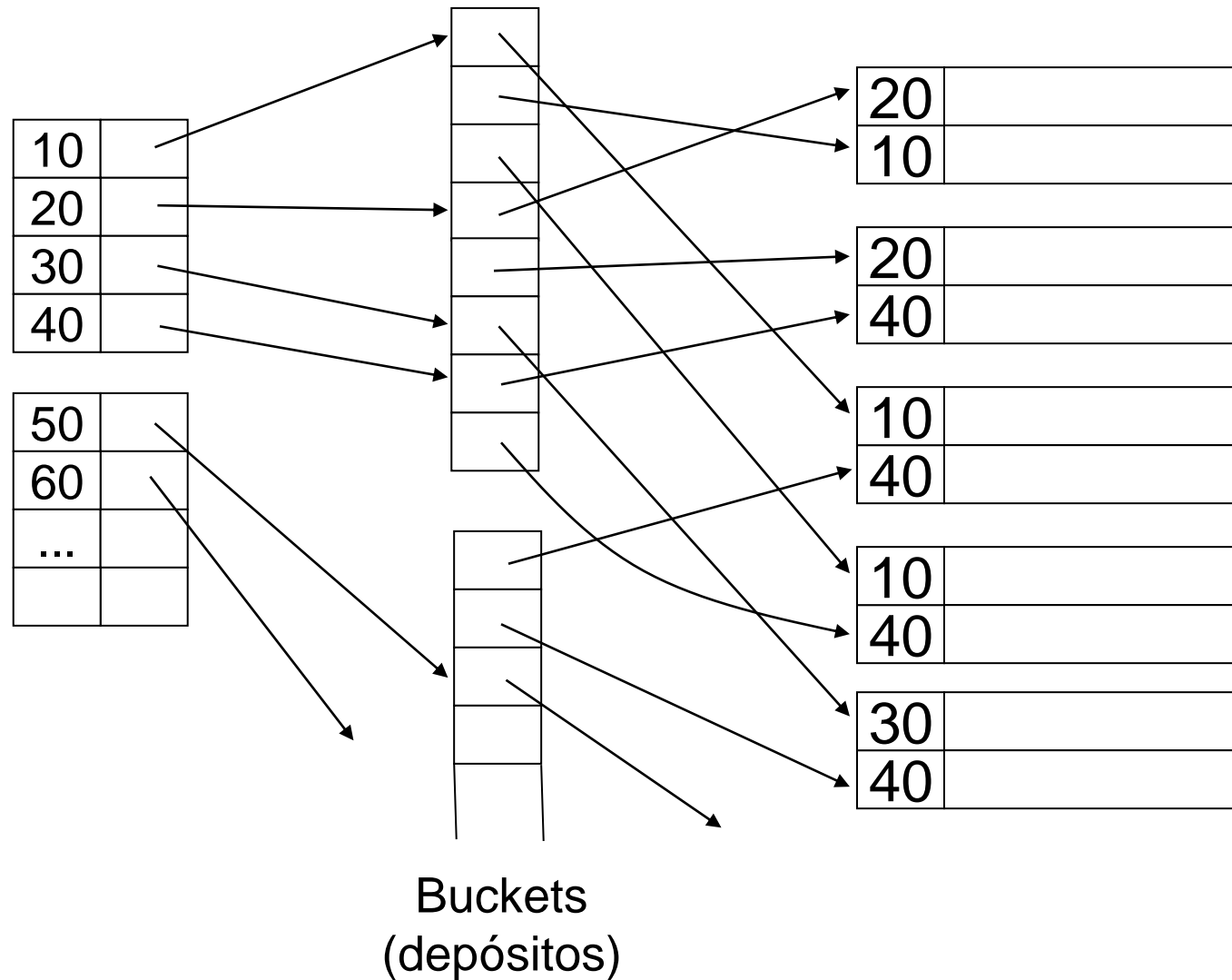


- Outra opção...

Problema:  
Registros do  
índice ficam  
com tamanhos  
variáveis



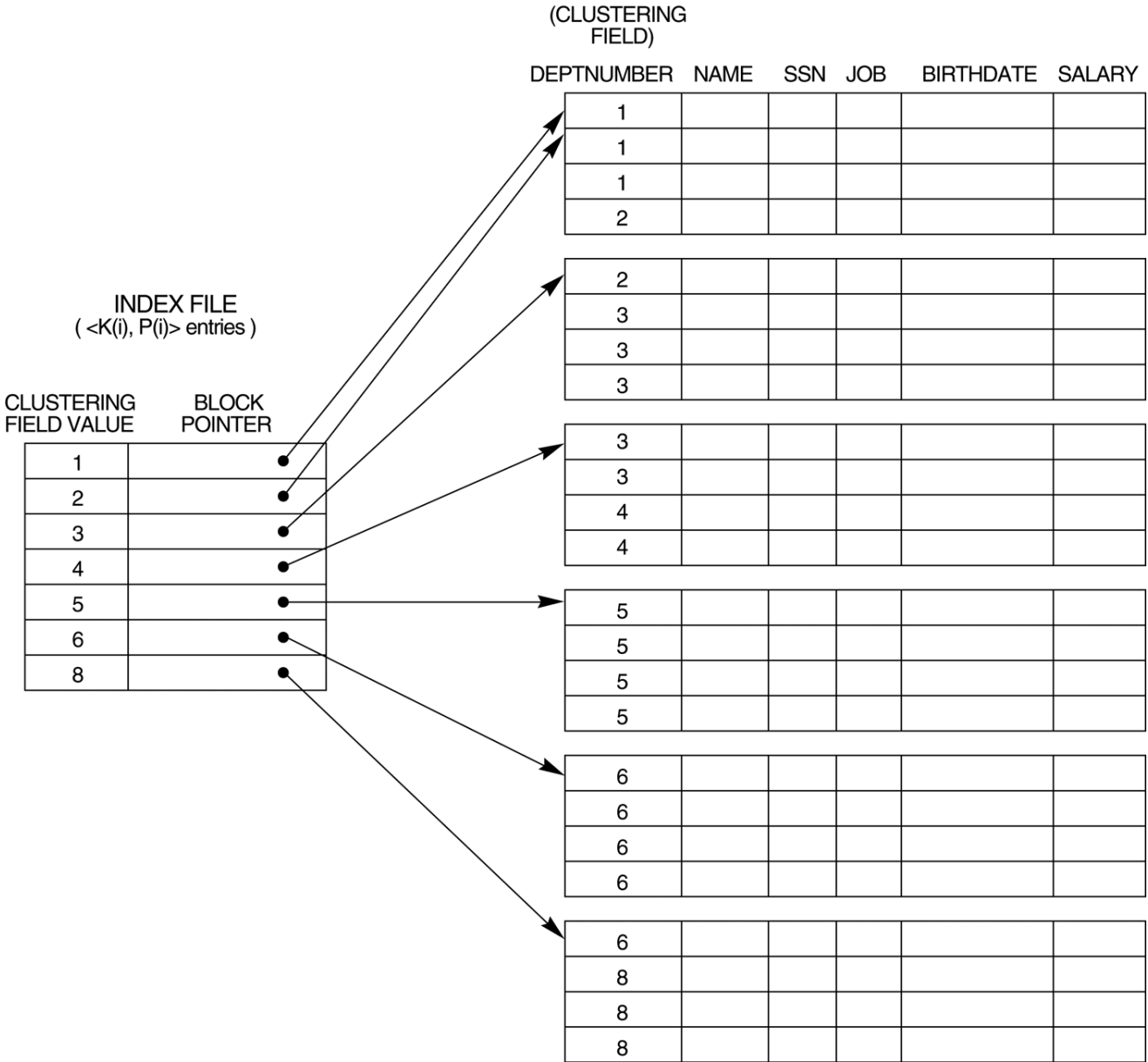




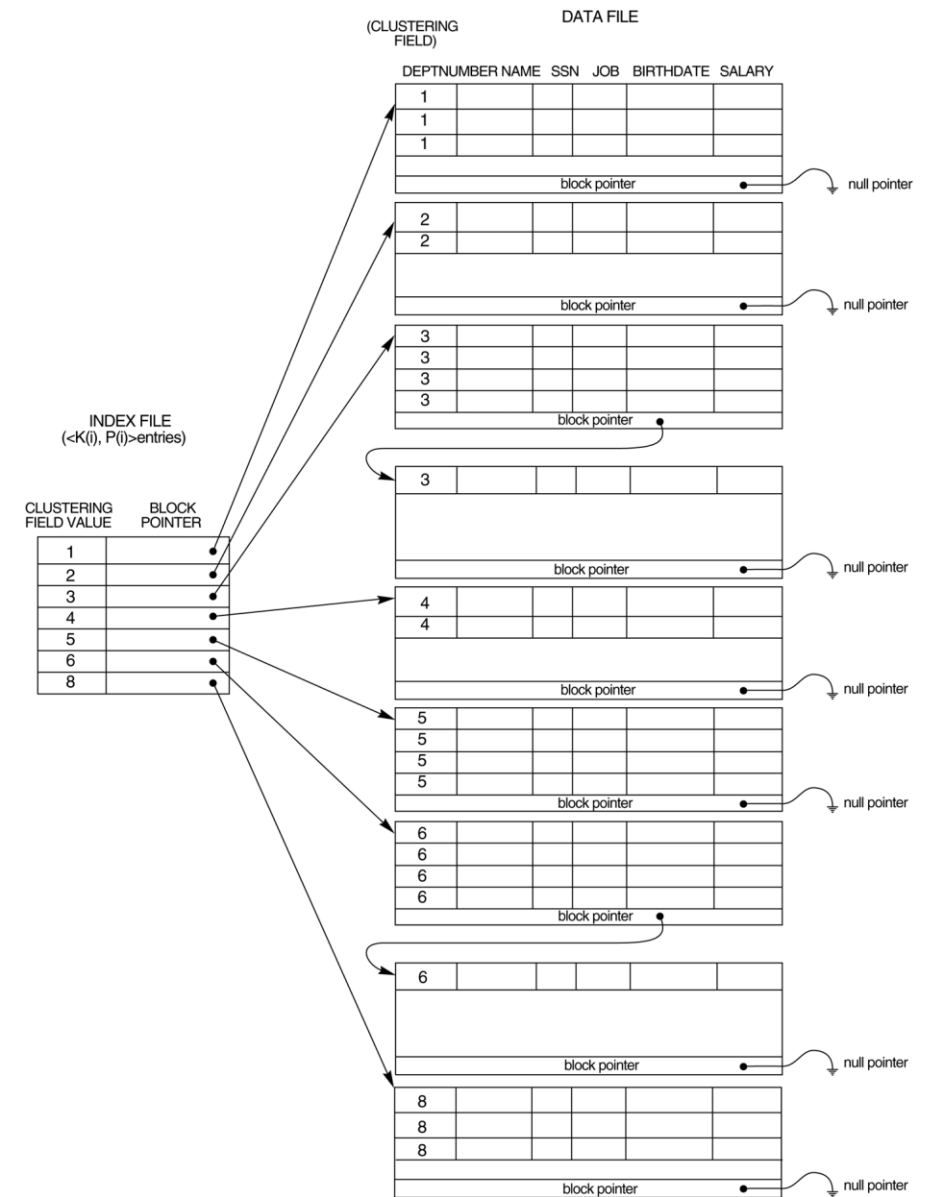
- Índice *Non-Clustered*
  - Índice em que o arquivo de dados é não ordenado e de estrutura simples
- Índice *Clustered*
  - Índice em que o arquivo de dados é ordenado por uma chave de busca e pode conter mais de um tipo de registro intercalado
  - Exemplo
    - Índice pra codDepto em Departamento
    - Intercalados aos registros dos Departamentos, estão os registros de Empregado daquele Departamento (o codDepto se repete em vários registros)
    - O arquivo de dados é ordenado por codDepto

- Definido em um arquivo de dados ordenado.
- O arquivo de dados é ordenado por um campo não chave (ao contrário do índice primário).
- Inclui uma entrada de índice para cada valor distinto do campo (mas não necessariamente para cada registro).
- Usualmente é esparso.
- Algoritmo de busca pela chave:
  - Localiza o valor imediatamente inferior a chave de busca. A entrada do índice aponta para o primeiro bloco de dados que contém registros com o valor do campo. Para buscar os registros é feita leitura sequencial dos registros até achar um valor que seja diferente da chave buscada.

Poderia  
Intercalar  
registros de  
Departamentos  
com registros de  
Empregados



Índice *clustered* com um grupo (*cluster*) separado de blocos para cada grupo de registros que compartilhem o mesmo valor de campo de *clustering*

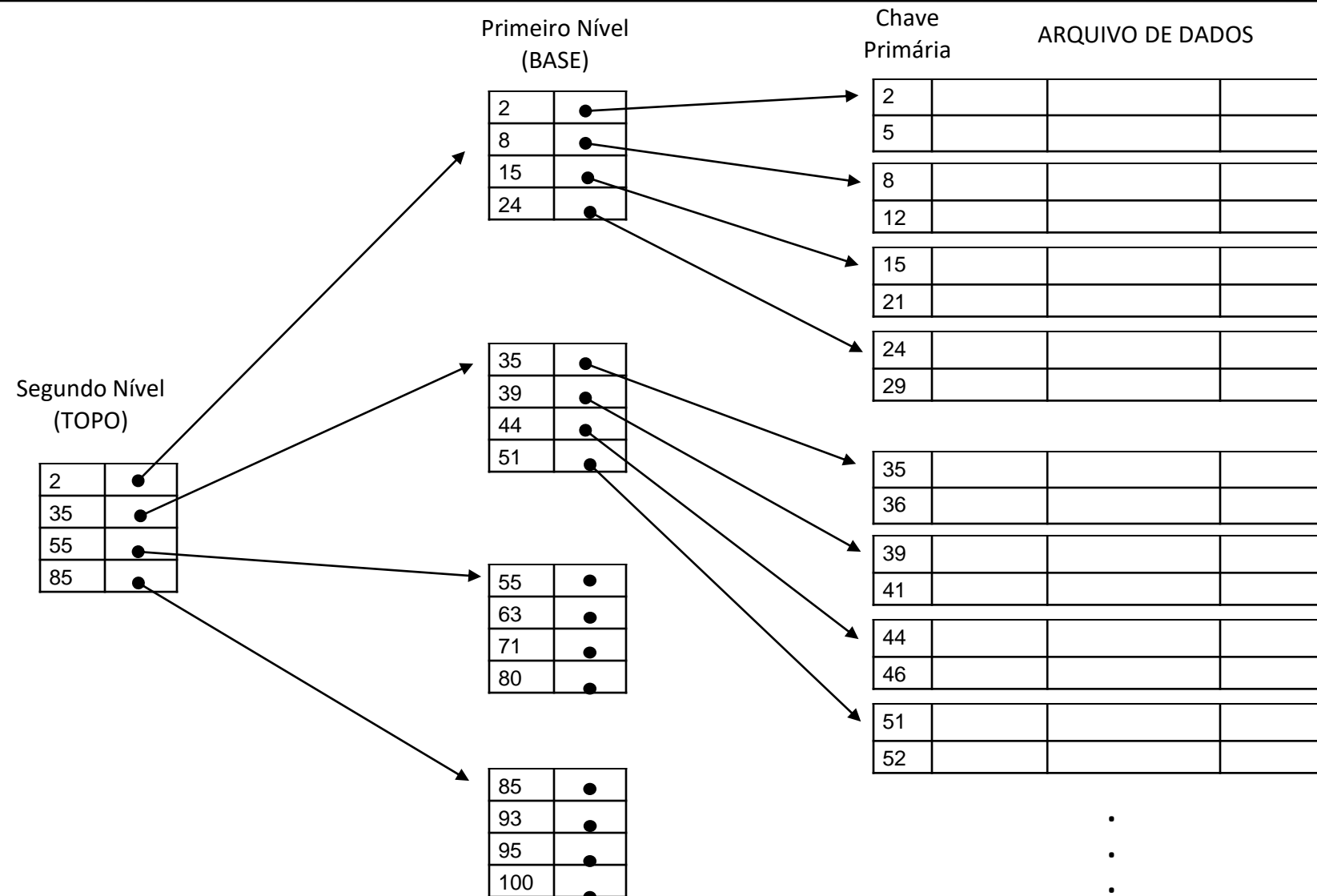


```
CREATE CLUSTER c_pedidos (numero_pedido NUMBER(7)
SIZE 512; /* clusters de 512B, se não especificar, usa um datablock */
```

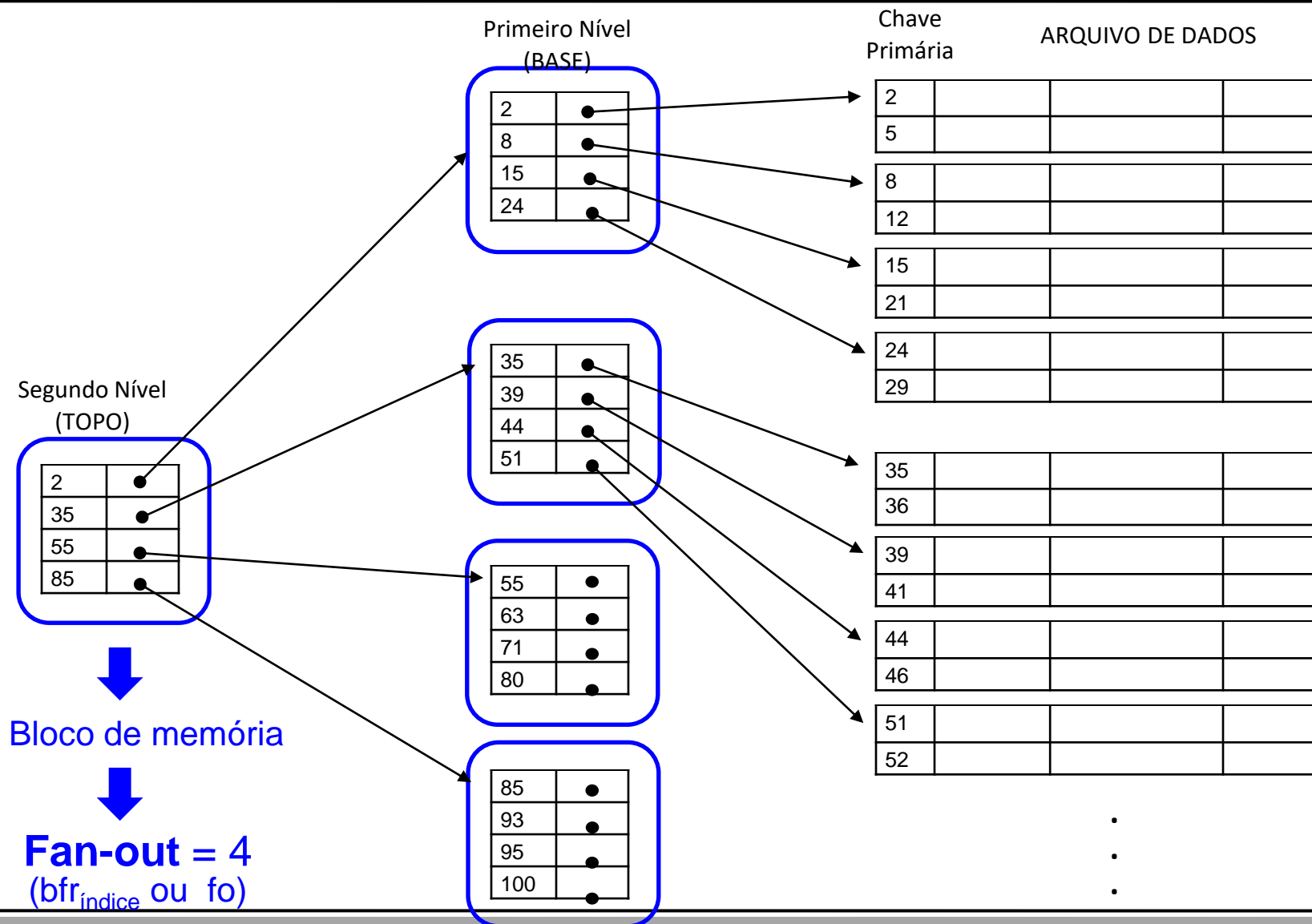
```
CREATE TABLE pedidos (
    num_pedido NUMBER (7) NOT NULL,
    cod_cliente NUMBER (6),
    data_emissao DATE NOT NULL,
    valor_total SMALLMONEY NOT NULL,
    CONSTRAINT PK_PEDIDOS PRIMARY KEY (num_pedido)
) cluster c_pedidos (num_pedido);
```

```
CREATE TABLE pedidos_produtos (
    num_pedido NUMBER (7) NOT NULL,
    cod_produto NUMBER (5) NOT NULL,
    quantidade NUMBER (3) NOT NULL,
    valor_unitario NUMBER (10,2) NOT NULL,
    CONSTRAINT PK_PEDIDOS_PRODUTOS PRIMARY KEY
        (num_pedido, cod_produto)
) cluster c_pedidos (num_pedido);
```

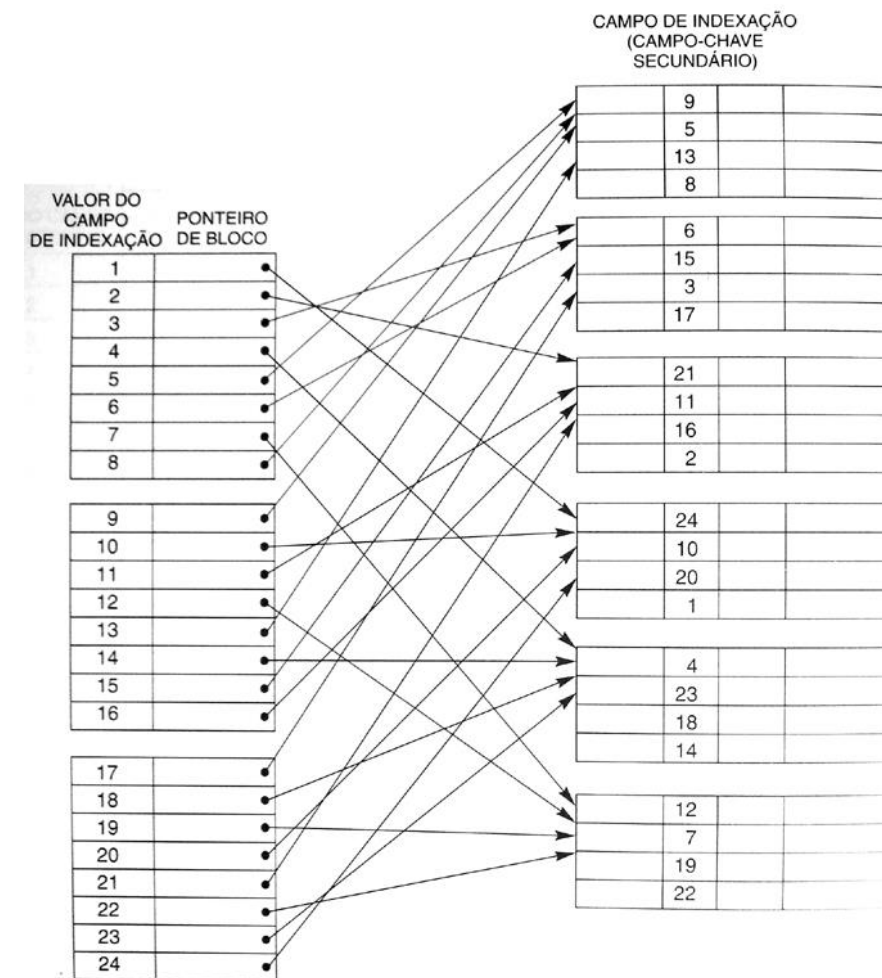
- Índices em nível único
  - Apenas um nível de indireção entre a estrutura do índice e o arquivo de dados
- Índice multinível
  - Duas ou mais indireções entre a estrutura do índice e o arquivo de dados







- Índice secundário sobre um campo-chave secundária
- Arquivo de Dados:
  - Arquivo não ordenado com  $r = 30.000$  registros
  - Disco com tamanho de bloco  $B = 1024$  bytes
  - Registros de tamanho fixo  $R = 100$  bytes
  - Fator de bloco  $bfr = \lfloor B/R \rfloor = 10$  registros por bloco
  - Número de blocos do arquivo  $b = \lceil r/bfr \rceil = 3.000$  blocos
  - Pesquisa de um registro dentro do arquivo sequencial
    - Média  $b/2$  acessos = 1500 blocos/acessos
- Arquivo de Índice Secundário:
  - Tamanho campo chave  $V = 9$  bytes
  - Tamanho de um ponteiro de bloco  $P = 6$  bytes
  - Tamanho de cada entrada do índice  $R_i = (9 + 6) = 15$  bytes
  - Fator de bloco do índice  $bfri = \lfloor B/R_i \rfloor = \lfloor 1024/15 \rfloor = 68$  entradas por bloco
  - Índice denso, logo o número total de entradas do índice é  $r_i = r = 30.000$
  - O número de blocos do arquivo índice é  $b_i = \lceil r_i/bfri \rceil = \lceil 30.000/68 \rceil = 442$  blocos
  - Busca binária no índice  $\lceil \log_2 b_i \rceil = \lceil \log_2 442 \rceil = 9$  acessos + 1 = 10 acessos





- Arquivo de Índice:
  - Fan-out do índice multinível  $fo = bfri = \lfloor B/Ri \rfloor = \lfloor 1024/15 \rfloor = 68$  entradas por bloco
  - Número de blocos do primeiro nível é  $b1 = \lceil ri/fo \rceil = \lceil 30.000/68 \rceil = 442$  blocos

Primeiro Nível  
 $b_1 = 442$  blocos




⋮

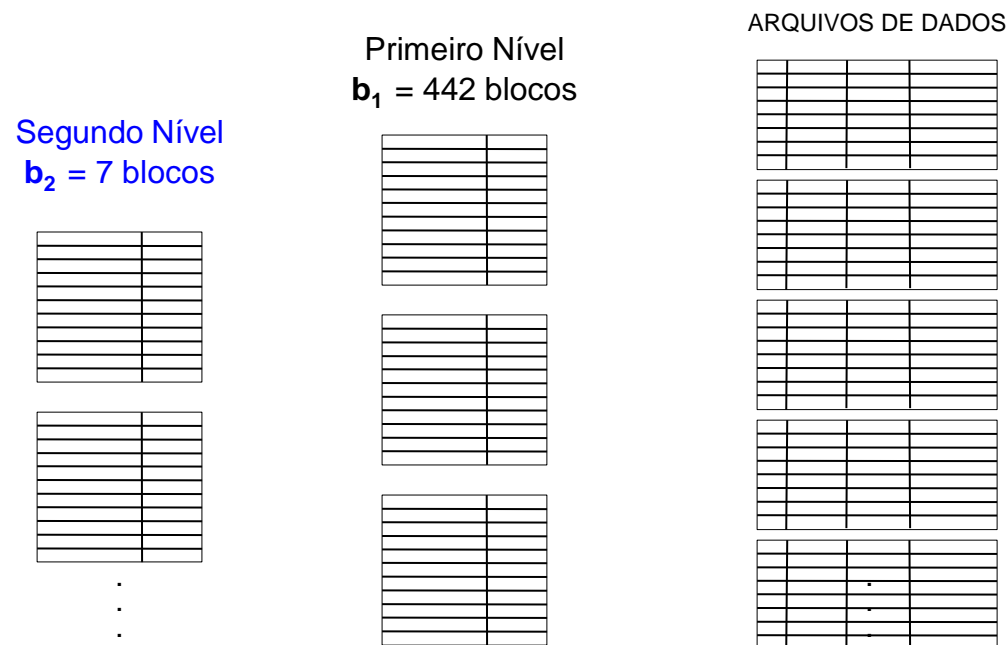
ARQUIVO DE DADOS






⋮

- Fan-out do índice multinível  $f_o = b_{fri} = \lfloor B/R_i \rfloor = \lfloor 1024/15 \rfloor = 68$  entradas por bloco
- Número de blocos do primeiro nível é  $b_1 = \lceil r_i / f_o \rceil = \lceil 30.000/68 \rceil = 442$  blocos
- Número de blocos do segundo nível é  $b_2 = \lceil b_1 / f_o \rceil = \lceil 442/68 \rceil = 7$



- Arquivo de Índice:

- Fan-out do índice multinível  $fo = bfri = \lfloor B/Ri \rfloor = \lfloor 1024/15 \rfloor = 68$  entradas por bloco
- Número de blocos do primeiro nível é  $b_1 = \lceil ri/fo \rceil = \lceil 30.000/68 \rceil = 442$  blocos
- Número de blocos do segundo nível é  $b_2 = \lceil b_1/fo \rceil = \lceil 442/68 \rceil = 7$
- Número de blocos do terceiro nível é  $b_3 = \lceil b_2/fo \rceil = \lceil 7/68 \rceil = 1$

Terceiro Nível  
 $b_3 = 1$  bloco


Segundo Nível  
 $b_2 = 7$  blocos



⋮

Primeiro Nível  
 $b_1 = 442$  blocos




⋮

ARQUIVOS DE DADOS






⋮

- Número de acessos a blocos ?

Terceiro Nível  
 $b_3 = 1$  bloco


Segundo Nível  
 $b_2 = 7$  blocos



⋮

Primeiro Nível  
 $b_1 = 442$  blocos




⋮

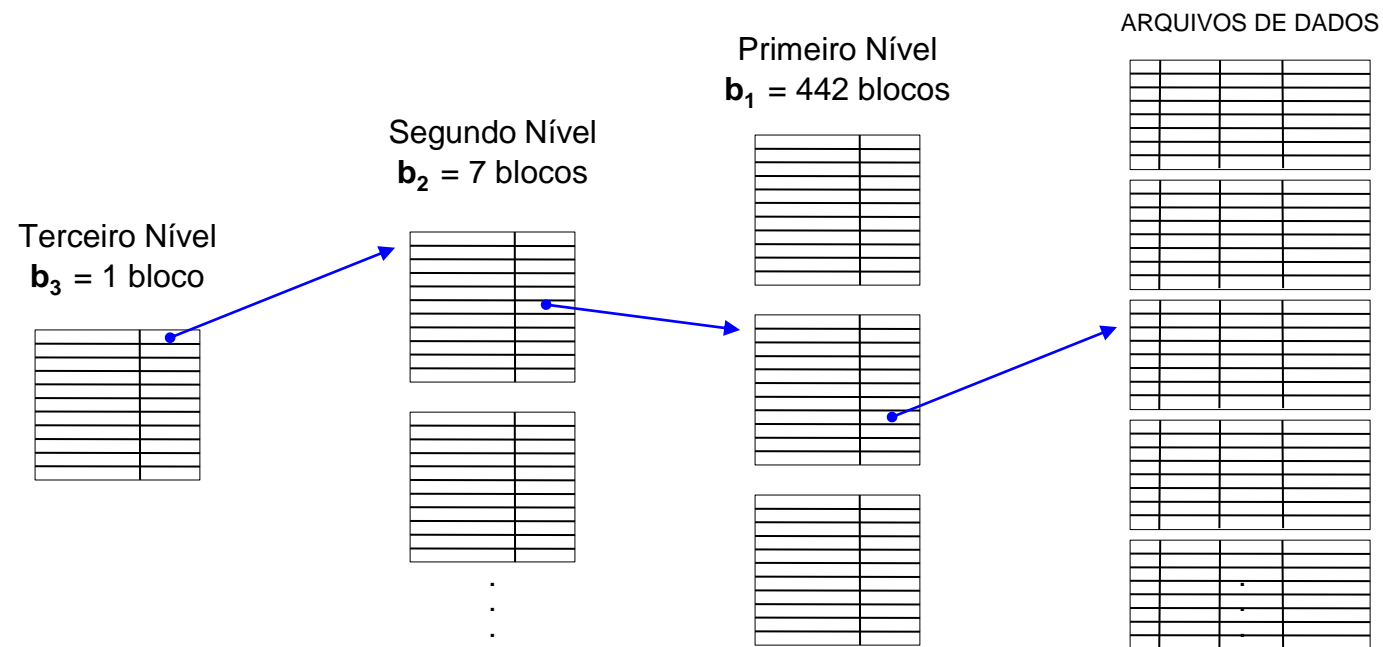
ARQUIVOS DE DADOS






⋮

- Número de acessos a blocos ?
  - Número de níveis  $x + 1 = 4$  acessos
  - Número de blocos adicionais = 8 blocos.
  - Lembram dos 1500 sem índice?





- E como se implementa um índice multinível?

