UNIVERSIDADE FEDERAL DE ITAJUBÁ

Banco de Dados II

COM 312

Indexação
Aula 5

Vanessa Cristina Oliveira de Souza

Introdução

- No nível conceitual ou lógico, o banco de dados no modelo relacional, foi visto como uma coleção de tabelas.

- O modelo lógico de um banco de dados é o nível correto para o usuário do banco de dados focalizar.

Introdução

- O usuário de um sistema de banco de dados está normalmente interessado no desempenho do sistema.
- Quão rápido esse sistema responde a uma requisição desse usuário?
- Portanto, o fator principal de satisfação (ou de insatisfação) é o desempenho da base de dados – se o tempo de resposta a uma consulta é grande demais, o sistema é desvalorizado.

Introdução

- A performance do sistema de base de dados depende:
  - Da eficiência das estruturas computacionais usadas para representar os dados na base e,
  - De como o sistema eficientemente é capaz de operar nessas estruturas de dados.

Gerenciador de Arquivos

- Componente funcional de um SGBD que gerencia a alocação do espaço na armazenagem do disco e as estruturas de dados usadas para representar a informação armazenada no disco.

Gerenciador de Buffer

- Componente funcional de um SGBD responsável pela transferência de informações entre o disco de armazenagem e a memória principal.

UNIVERSIDADE FEDERAL DE ITAJUBÁ

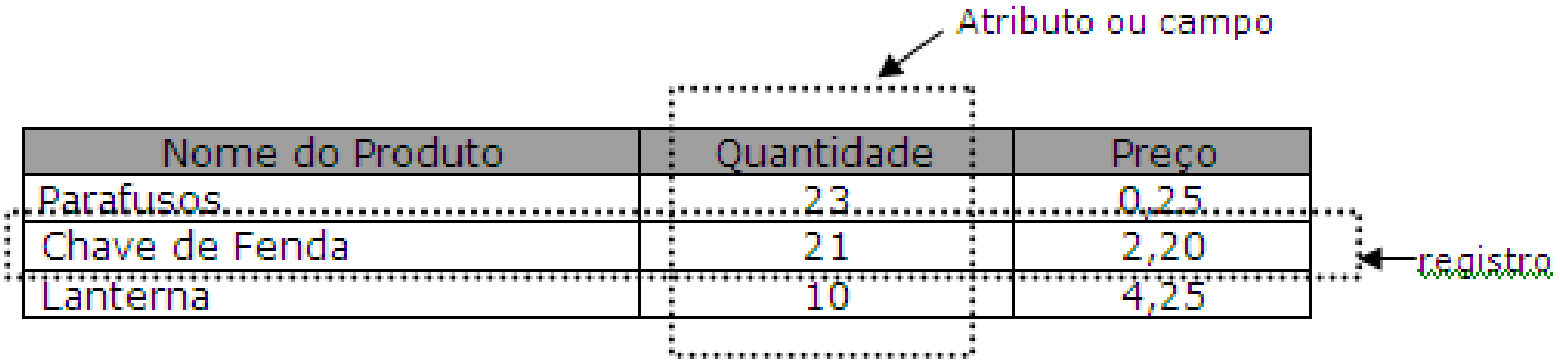
ORGANIZAÇÃO DE ARQUIVOS

CONCEITOS BÁSICOS

Arquivo de dados

- Estrutura de dados que armazena o banco de dados propriamente dito.

Arquivo x Registro



Estrutura de um arquivo.

Operações básicas sobre arquivos

- Inserção
  - Adição de um registro no arquivo
- Alteração
  - Corresponde a localização (consulta) de determinado registro, a alteração de um ou mais de seus atributos e atualização
- Exclusão
  - Eliminação de um registro do arquivo

Operações básicas sobre arquivos

- Consulta
  - Localização de um registro no arquivo.
  - Pode ser de três tipos:
    - *simples* – O usuário especifica o objeto que está procurando (ou sua negação)
    - *por faixa* – O usuário especifica a faixa de valores onde encontram-se os objetos que ele está procurando. Geralmente são utilizados os operados relacionais
    - *booleana* – Corresponde a combinação de uma consulta simples ou por faixa, utilizando um operador lógico

- Reorganização/Ordenação
  - Dependendo da organização de arquivos utilizada é necessário reorganizar ou ordenar os registros, facilitando a sua manipulação.

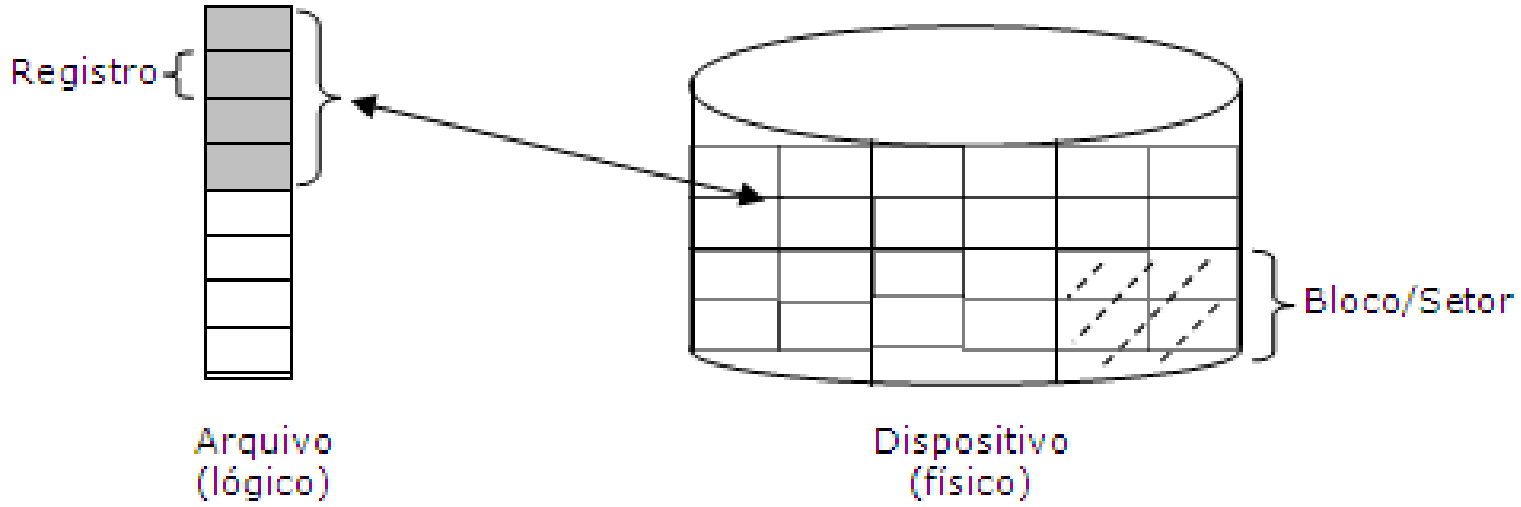
Registros Lógicos
x
Registros Físicos

- O registro **lógico** corresponde às instâncias (objetos) do mundo real, cujos atributos são representados pelos campos. Estes registros é que são manipulados pelos programas (sistemas) de aplicação.

- O registro **físico** corresponde à organização física onde as informações são armazenadas (no HD, setores).

Organização de Arquivos

- Organizar um arquivo significa relacionar (mapear) a estrutura lógica e a estrutura física de um arquivo.



Mapeamento entre estrutura física e lógica.

Organização de Arquivos

- A organização de arquivo trata do arranjo ou a forma de distribuição dos registros dentro do arquivo, objetivando **agilizar o processo de armazenamento e recuperação de dados**.

Arquivos Sequenciais

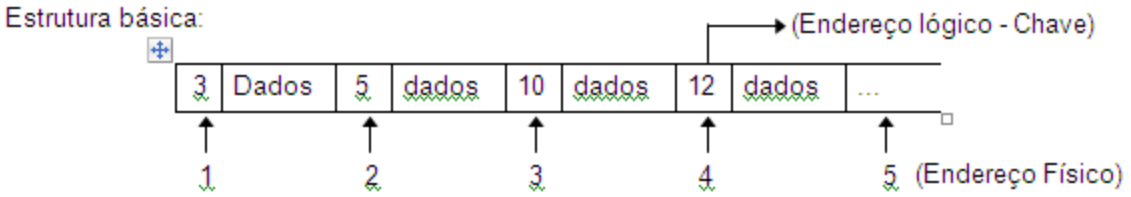
- Em um arquivo sequencial, os registros são dispostos ordenadamente, obedecendo à sequência determinada por uma chave primária, chamada de chave de ordenaçãc

EMPREGADO

Chave de pesquisa: *1030*

Matricula	Nome	Idade	Salário
3	Ademar	32	5000
5	Roberto	25	7500
10	Gerson	43	6000
12	Yeda	23	9000
30	Bernardo	21	4500
50	Ângela	29	5000

Chave de ordenação



Arquivos Sequenciais

- Para permitir uma recuperação rápida de registros na ordem da chave de busca, os registros são encadeados por ponteiros.

- O ponteiro em cada registro aponta para o próximo registro na ordem da chave de busca.

Arquivos Sequenciais

1	3	Ademar	32	5000
2	5	Roberto	25	7500
3	10	Gerson	43	6000
4	12	Yeda	23	9000
5	30	Bernardo	21	4500
6	50	Ângela	29	5000

Overflow

Arquivos Sequenciais

- Vantagem: Acesso sequencial de um conjunto de dados é o mais rápido.
- Útil quando é preciso processar quase todos os elementos.
- Desvantagens: Inclusão e Exclusão é extremamente custosa.

Arquivos Sequenciais
Exemplo de Inserção

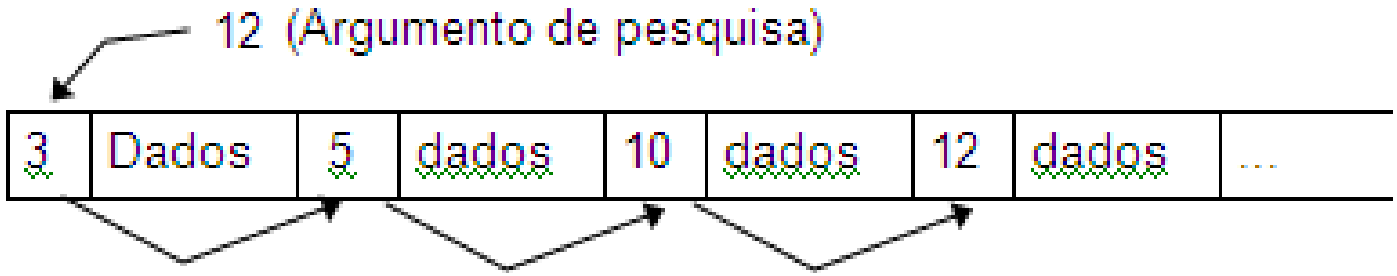
	3	Ademar	32	5000
	5	Roberto	25	7500
	10	Gerson	43	6000
	12	Yeda	23	9000
	30	Bernardo	21	4500
	50	Ângela	29	5000
	15	Vanessa	28	6000

Overflow

Inserir o registro de chave de ordenação 15.

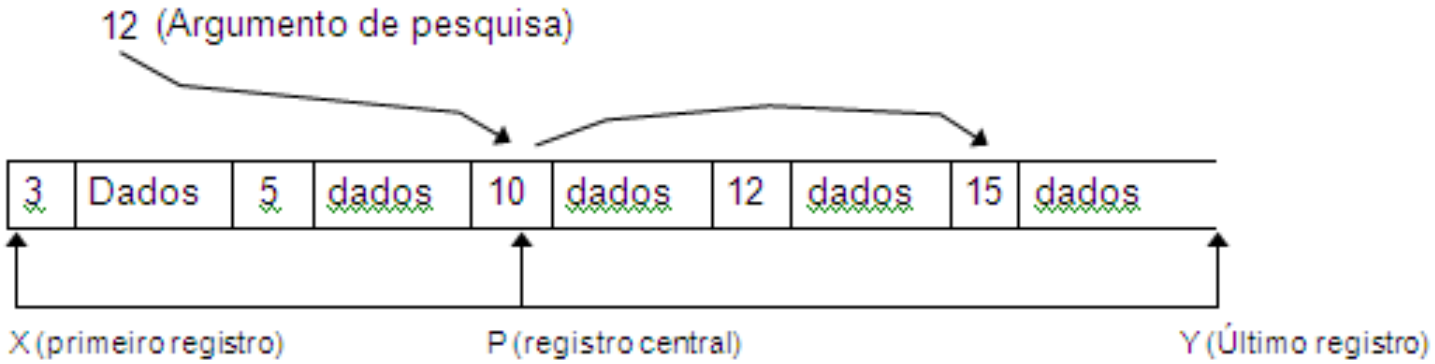
Arquivos Sequenciais Consultas

- Acesso serial (ou sequencial):
  - consiste na obtenção do próximo registro, segundo a chave de ordenação. Este tipo de acesso é *extremamente eficiente* por estarem os registros fisicamente armazenados de acordo com a sequência que é solicitada. *O argumento de pesquisa é comparado com cada registro lido de forma seqüencial*.



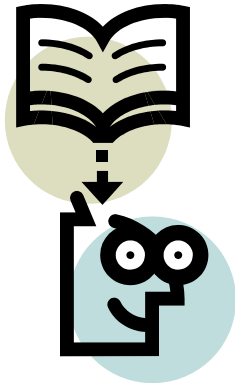
Arquivos Sequenciais Consultas

- Acesso aleatório
  - Divide-se o arquivo pela metade e compara-se o argumento de pesquisa com o registro da divisão. Se este não for o registro desejado, determina-se para qual metade se fará a próxima divisão (compara se o argumento de pesquisa é maior ou menor que o registro da divisão), e novamente se faz a comparação com o registro divisor e assim sucessivamente.



INDEXAÇÃO

Indexação – Problema



Indexação – Problema

- Tempo para encontrar o livro aumenta significativamente com o número de livros que se tem na biblioteca em questão.

- Solução:
  - Um catálogo, organizado de alguma forma (por autor, por título, ...) que diga onde encontrar o livro na prateleira.

Indexação – Solução em BD

- Utilização de índices.

- Objetivo:
  - Permitir um rápido acesso aleatório aos registros num arquivo.

- O que é um índice?
  - Um estrutura de dados **adicional** associada ao arquivo (tabela).

Índices

- O uso de índice é conveniente aos processos que precisam de um **desempenho mais ágil**, mesmo para os registros que não se encontrem ordenados fisicamente.
- **Exemplo:**
- Para uma consulta que deseje encontrar todas as contas somente da agência de Brasília têm-se que:
  - refere-se a uma porção (ou fração) de todos os registros de contas;
  - é ineficiente para o sistema ler todos os registros para localizar somente os que são desta agência;
  - seria eficiente um acesso direto a esses registros;
- Assim torna-se interessante acrescentar uma nova estrutura que permita essa forma de acesso.

Índices

- Índices são, portanto, **estruturas de dados auxiliares** cujo único propósito é tornar mais **rápido** o acesso a registros baseados em certos campos, chamados campos de indexação.

Índices

- Um índice permite localizar um registro sem ter que examinar mais que uma pequena fração dos registros possíveis;

- O(s) campo(s) cujos valores o índice se baseia formam a **chave de pesquisa**;

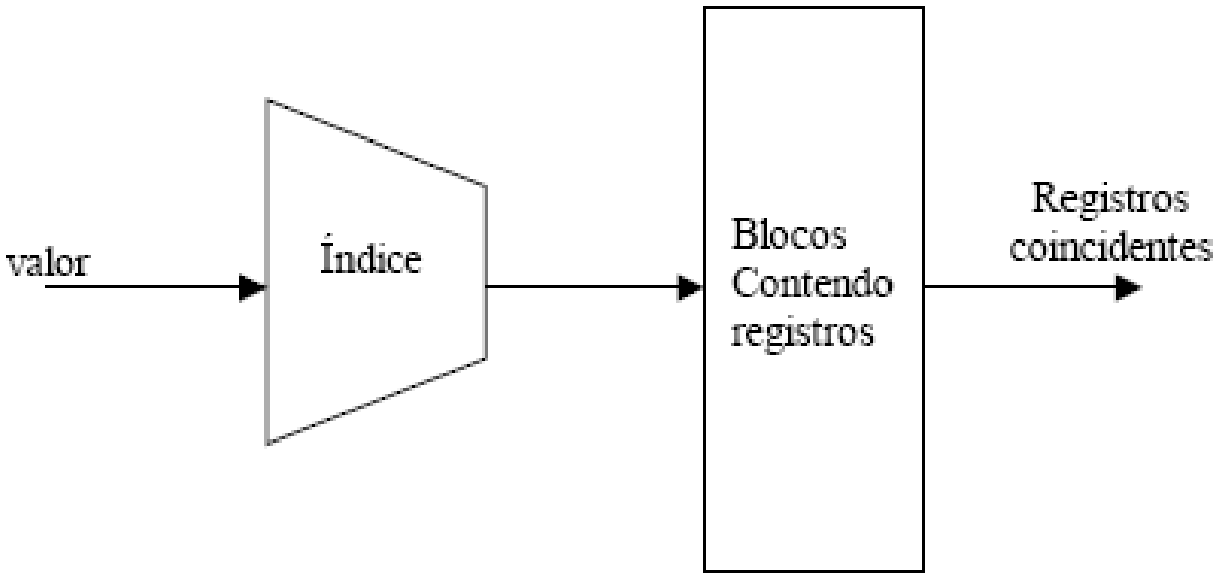
Índices

- Cada índice é formado pelo valor de um atributo chave do registro e pela sua localização física no interior do arquivo.

- A estrutura com a tabela de índices é também armazenada e mantida em disco.

Índices

- Cada estrutura de índice está associada a uma chave de busca particular.



Índices

- No exemplo da biblioteca, poderíamos ter os seguintes índices:
  - Sobrenome do autor
  - Título do livro
  - Assunto do livro
  - ...

Tipos básicos de Índices

- Ordenados
  - Baseiam-se na ordenação dos valores.
- Hash
  - Baseiam-se na distribuição uniforme dos valores determinados por uma função (função de hash).

Classificação de Índices Ordenados

- Considerando a quantidade de entradas
  - Denso
  - Esparso
- Considerando a organização do arquivo
  - Primário
  - Clustering
  - Secundário
- Considerando os níveis de indirecionamento
  - Mononível
  - Multinível

UNIVERSIDADE FEDERAL DE ITAJUBÁ

ÍNDICES ORDENADOS

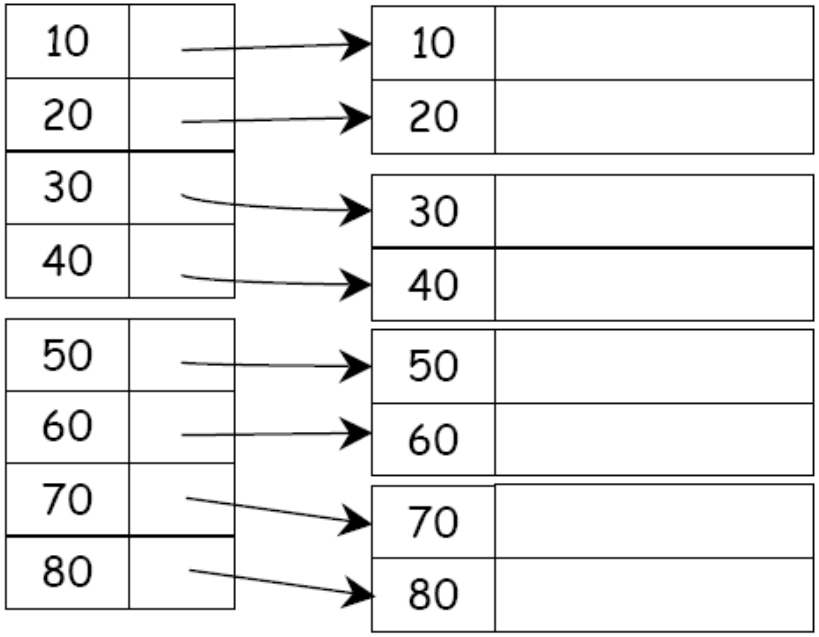
DENSO X ESPARSO

Índices sobre arquivos sequenciais

- Os ponteiros em um arquivo de índices podem se referir a **cada registro** do arquivo ou a um **bloco de disco**, originando dois tipos de índices:
  - Índice denso
  - Índice esparso

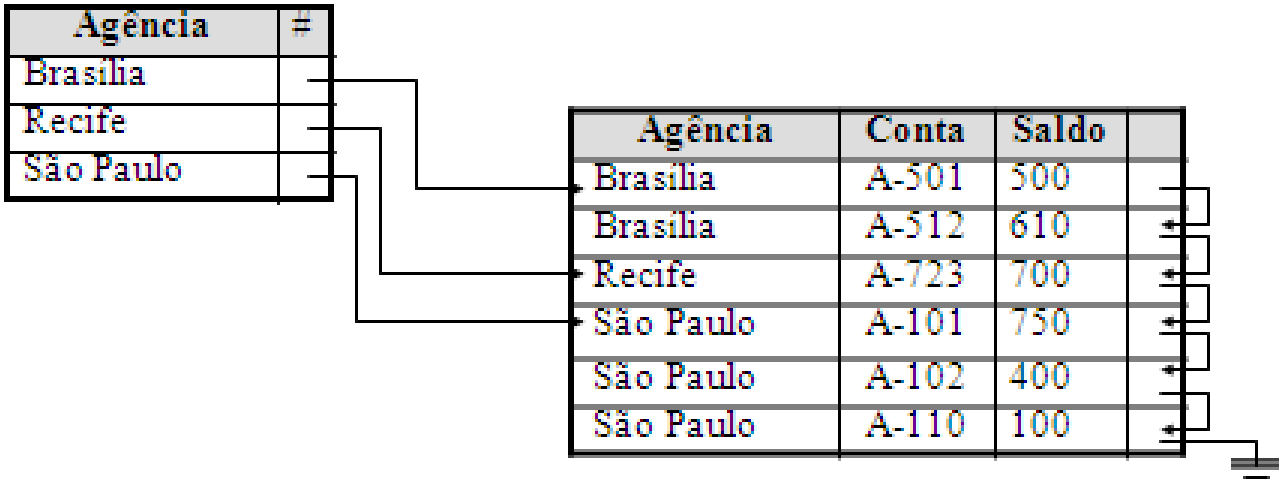
Índice Denso

- Há uma entrada no índice para cada valor de chave que ocorre em um registro de dados.



Esse registro de índice contém o valor da chave de busca e um ponteiro para o registro.

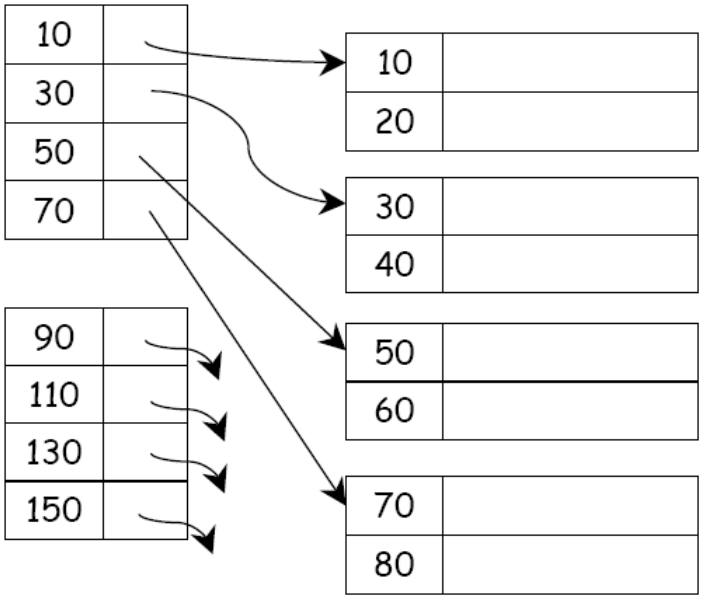
Índice Denso



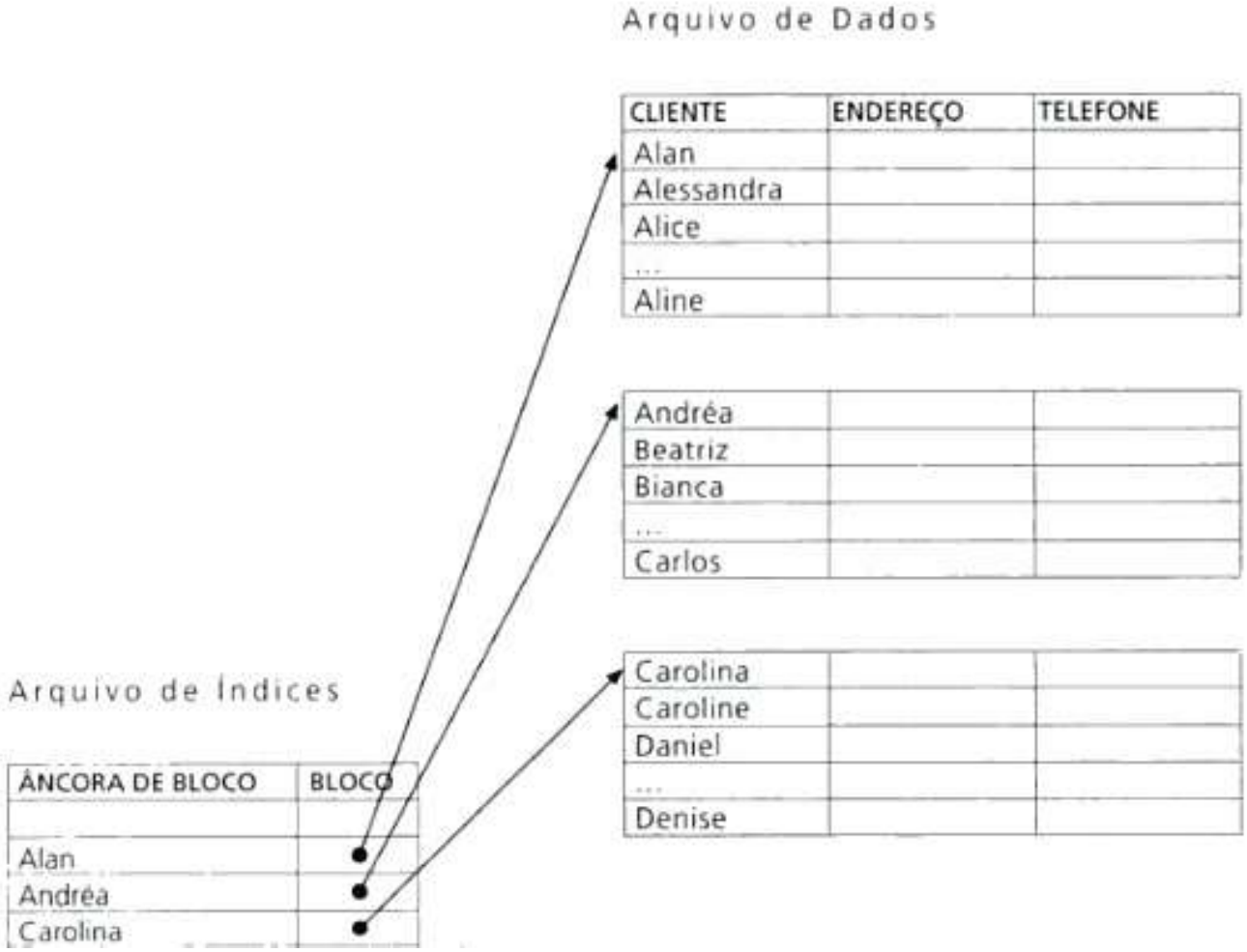
Índice Esparso

- Há um entrada no índice apenas para alguns valores de chave (um para cada bloco).

- Para localizar um registro com chave K, procura-se a entrada E do índice com o maior valor de chave menor ou igual a K e pesquisa-se o arquivo a partir do registro apontado por E.



Índice Esparso



Índice Esparso
x
Índice Denso

- Um índice esparso utiliza menos espaço para armazenamento ao custo de um tempo um pouco maior para encontrar um registro dada sua chave.

- No índice esparso, inserções e remoções são menos custosas quando comparadas ao índice denso.

Exercícios

- Considere uma relação com 1.000.000 registros (ou tuplas) que cabem dez a dez em um bloco de 4.096 bytes.

  a) Qual é o espaço necessário para os dados?

  b) Se o campo da chave tem 32 bytes e um ponteiro 8 bytes.
     a) Quantos pares chave-ponteiro cabem em um bloco?
     b) Qual o espaço ocupado por um índice denso para esta relação?
     c) E se um índice esparso for utilizado, qual o espaço gasto pelo índice?

UNIVERSIDADE FEDERAL DE ITAJUBÁ

ÍNDICES ORDENADOS

PRIMÁRIO X CLUSTERING X SECUNDÁRIO

Índice Primário

- No arquivo sequencial, um campo chave é usado para ordenar fisicamente os registro de arquivos em disco.

- Se a chave de ordenação for um campo exclusivo (sem duplicatas), então o índice é chamado de **Índice Primário**.

Índice Primário

- A chave de busca de um índice primário é **usualmente** a chave primária.

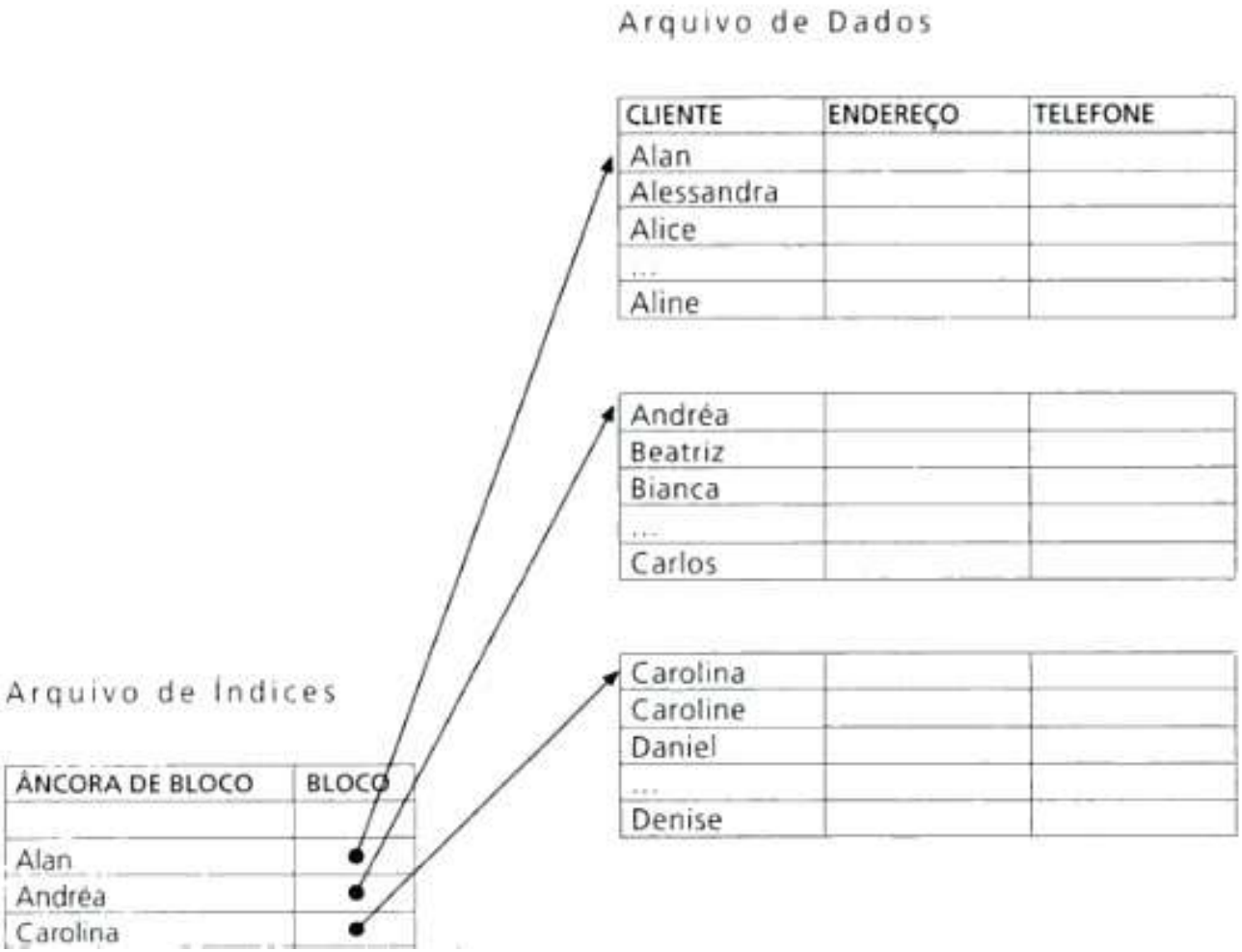
Índice Primário

- Vantagens
  - Economia de acesso aos blocos
  - Busca binária
- Desvantagem
  - Inclusão e remoção de registros

Índices primários em geral são esparsos

(o número de entradas é
igual ao número de blocos do arquivo de dados)

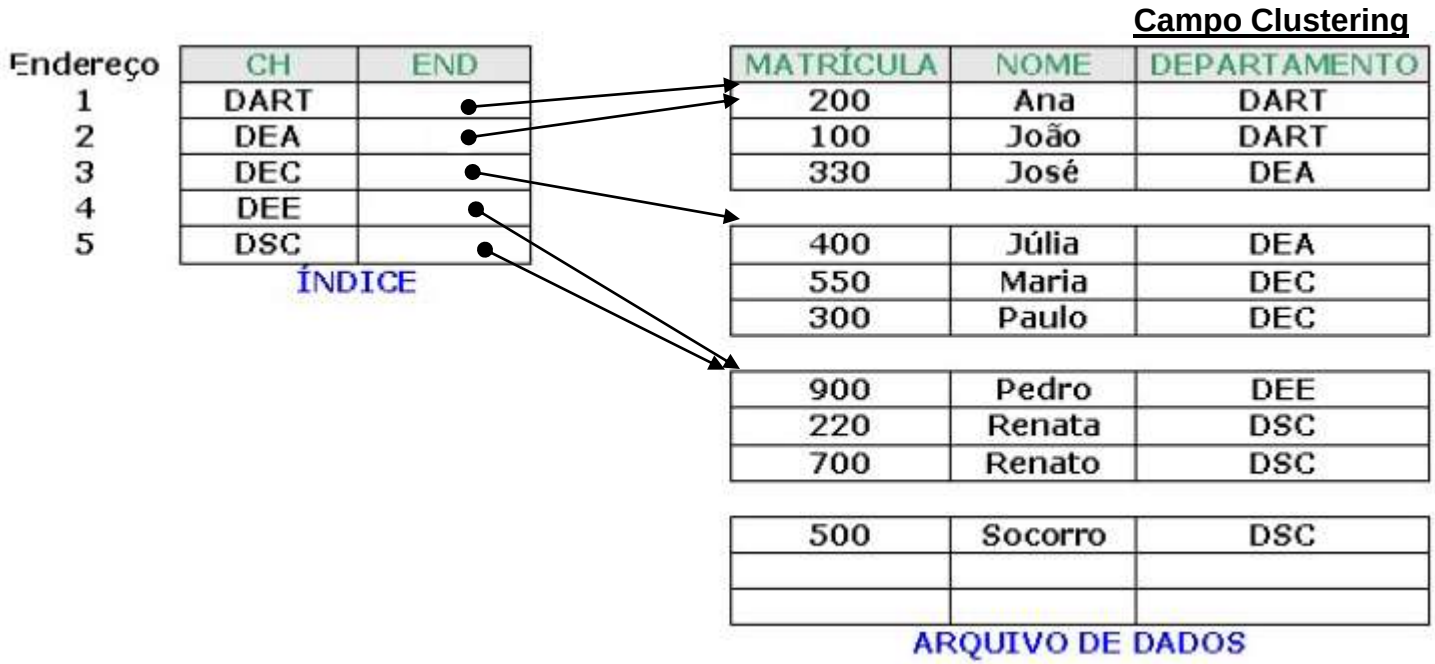
Índice Primário Esparso



Índice Clustering

- Se a chave de ordenação for um campo não chave (com duplicatas), então o índice é chamado de **Índice Clustering**.
- Existe uma entrada no índice clustering para cada valor distinto do campo clustering.
- O índice 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.

Índice de Clustering



No **Índice de Cluster** os registros de um arquivo são fisicamente ordenados por um campo de ordenação que permite duplicatas.

Índices de clustering são esparsos

(o número de entradas é igual ao número de valores distintos do atributo de clustering)

Índice de Clustering

- Vantagens
  - Economia de acesso aos blocos
- Desvantagem
  - Inclusão e remoção de registros

Índices sobre arquivos sequenciais

- Um arquivo pode possuir, no máximo, um índice primário ou um índice de clustering, **porém não ambos**.

- Porque um arquivo pode possuir, no máximo, um campo de ordenação física.

Índice Secundário

- Frequentemente desejamos ter vários índices em uma relação, a fim de facilitar consultas variadas.
- Um terceiro tipo de índice, chamado de índice secundário, pode ser especificado sobre qualquer campo não-ordenado de um arquivo.
- Um arquivo pode possuir diversos índices secundários, além de seu método de acesso principal.

Índice Secundário

- Um índice secundário é um índice cuja chave não é aquela da ordem sequencial do arquivo.

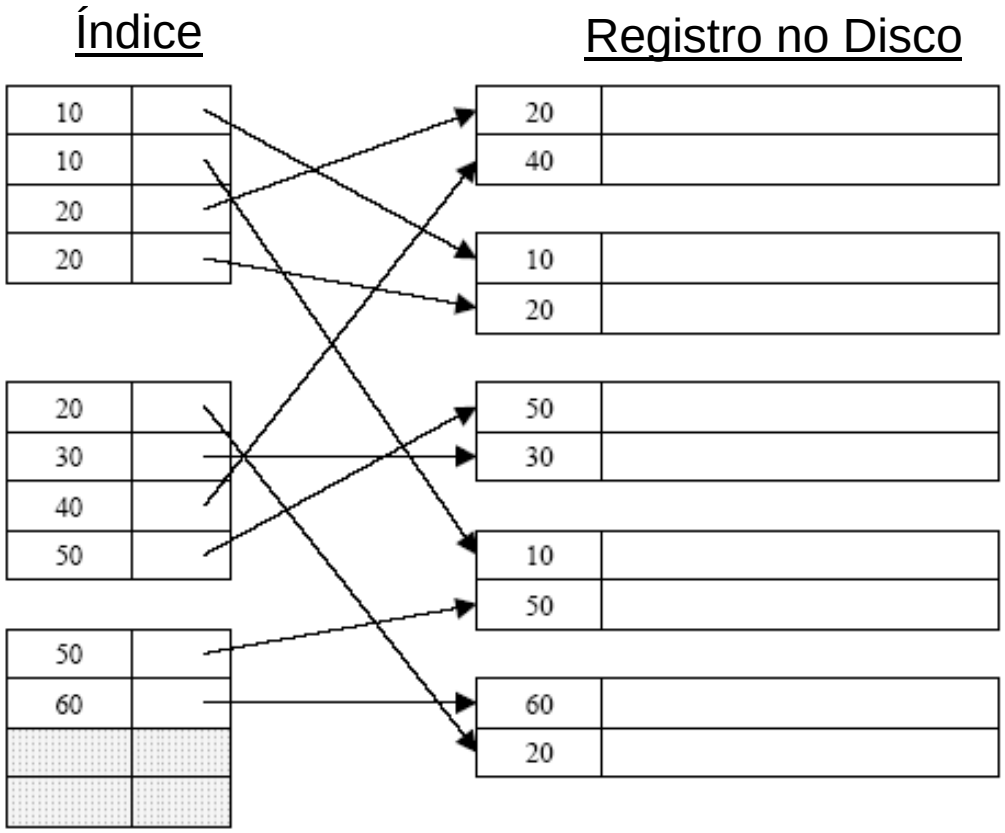
- Portanto, a diferença entre índice secundário e índice primário está no fato de que o índice secundário não determina a colocação de registros no arquivo de dados.

Índices secundários são sempre densos!!!

Projeto de índices secundários

- Um índice secundário é um índice denso, normalmente com duplicatas.
- O índice consiste em pares chave-ponteiro.
- Os pares no arquivo de índices são classificados pelo valor da chave.

Índices Secundários - Exemplo



- Os dados não estão classificados pela chave de pesquisa.
- As chaves no arquivo de índices estão classificadas.

Índice Secundário

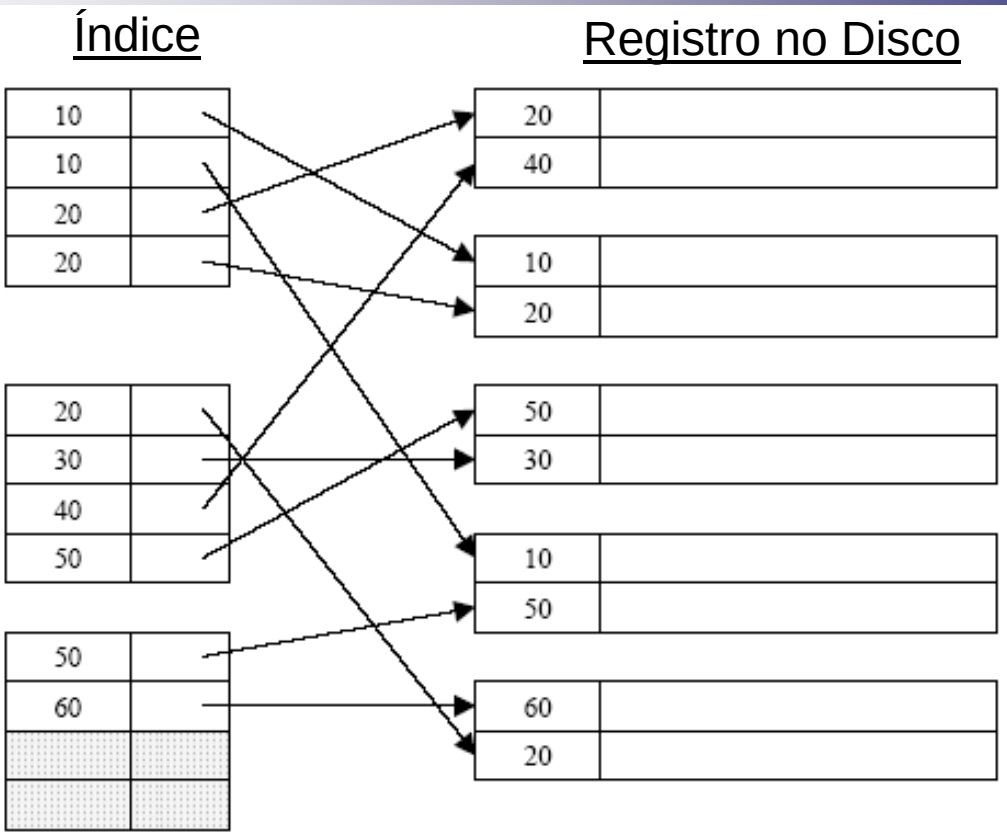
- Se o campo chave que estiver sendo indexado possuir duplicatas, dizemos que temos um **Índice Secundário para Não Chave Primária.**

Índice Secundário para Não Chave Primária

- Neste tipo de índice, vários registros do arquivo de dados podem ter o atributo chave com o mesmo valor, haja vista que o mesmo não é chave primária.

- Nesse caso, o índice aponta para o registro.

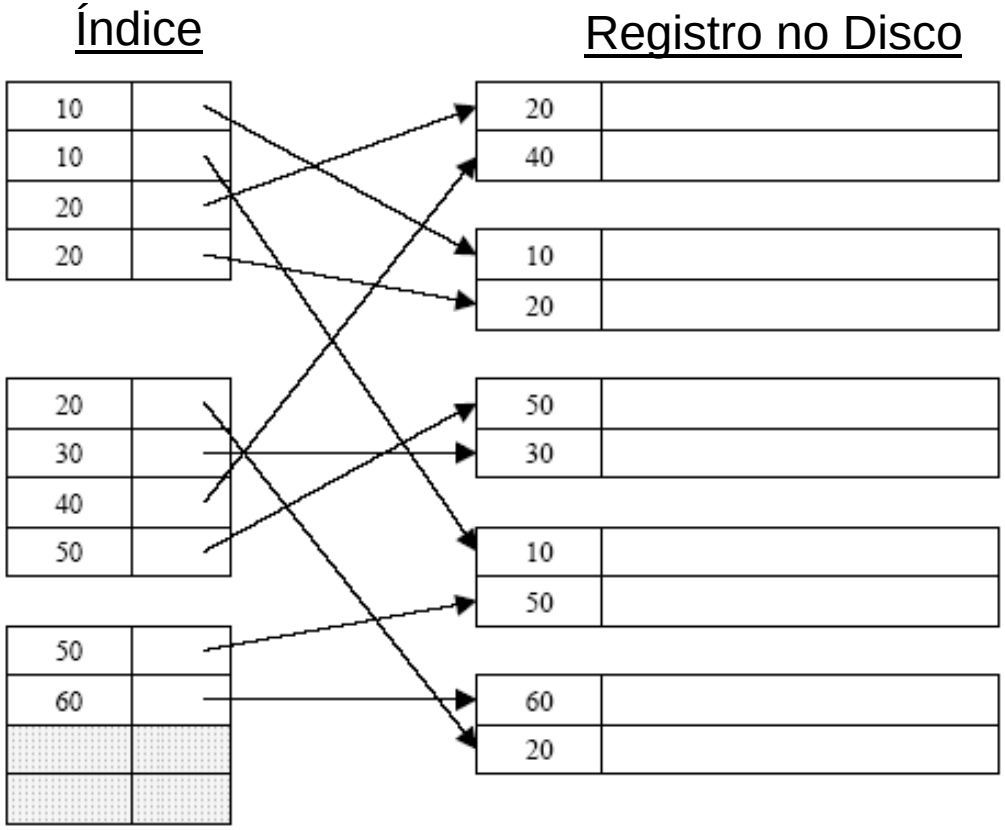
Índice Secundário para Não Chave Primária



Índice Secundário para Não Chave Primária

- Como as ordens da chave secundária e da chave física diferem, ao tentarmos varrer o arquivo sequencialmente na ordem da chave secundária, a leitura de cada registro provavelmente requererá a leitura de um novo bloco do disco.

Índice Secundário para Não Chave Primária

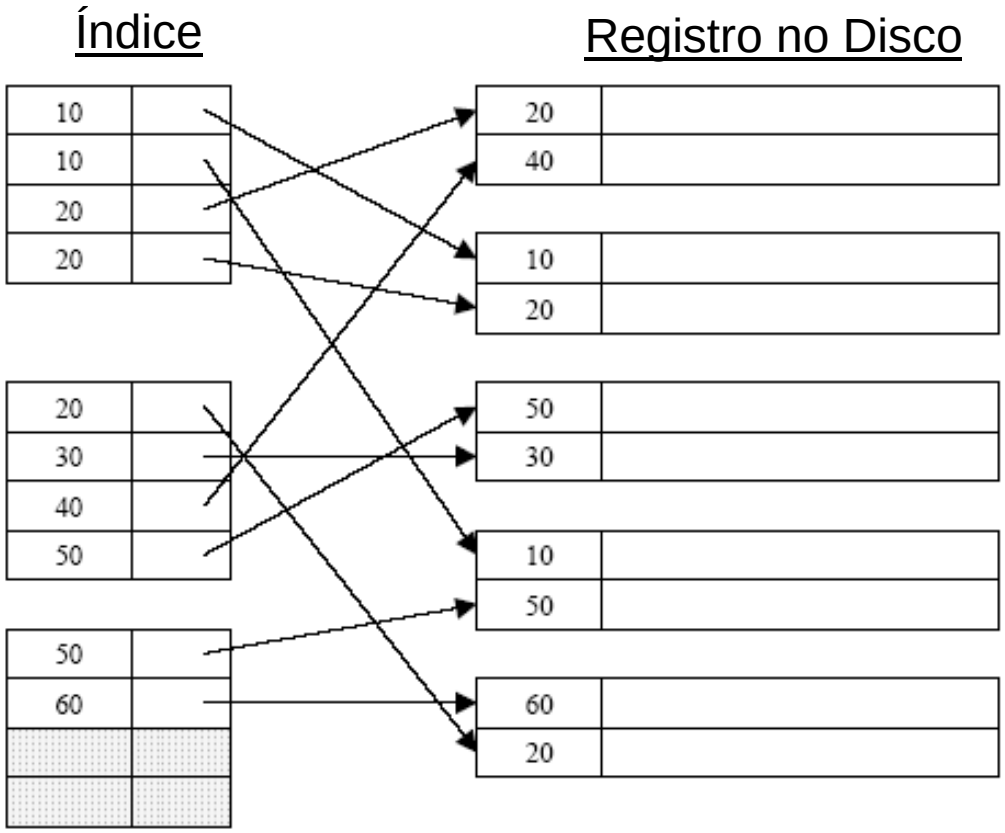


- Selecione todos os registros de valor 20.

Índice Secundário

- Problemas:
  - Os ponteiros de um bloco de índices podem ir para muitos blocos de dados diferentes, em vez de irem para um ou para alguns blocos sucessivos.
  - Se um valor de chave de pesquisa aparece *n* vezes no arquivo de dados, então o valor é escrito n vezes no arquivo de índice.
  - Seria melhor se pudéssemos escrever o valor da chave uma vez para todos os ponteiros para registros de dados com esse valor.

Índice Secundário



- Selecione todos os registros de valor 20.
  - Recuperar 2 blocos de índice + 3 blocos de dados.

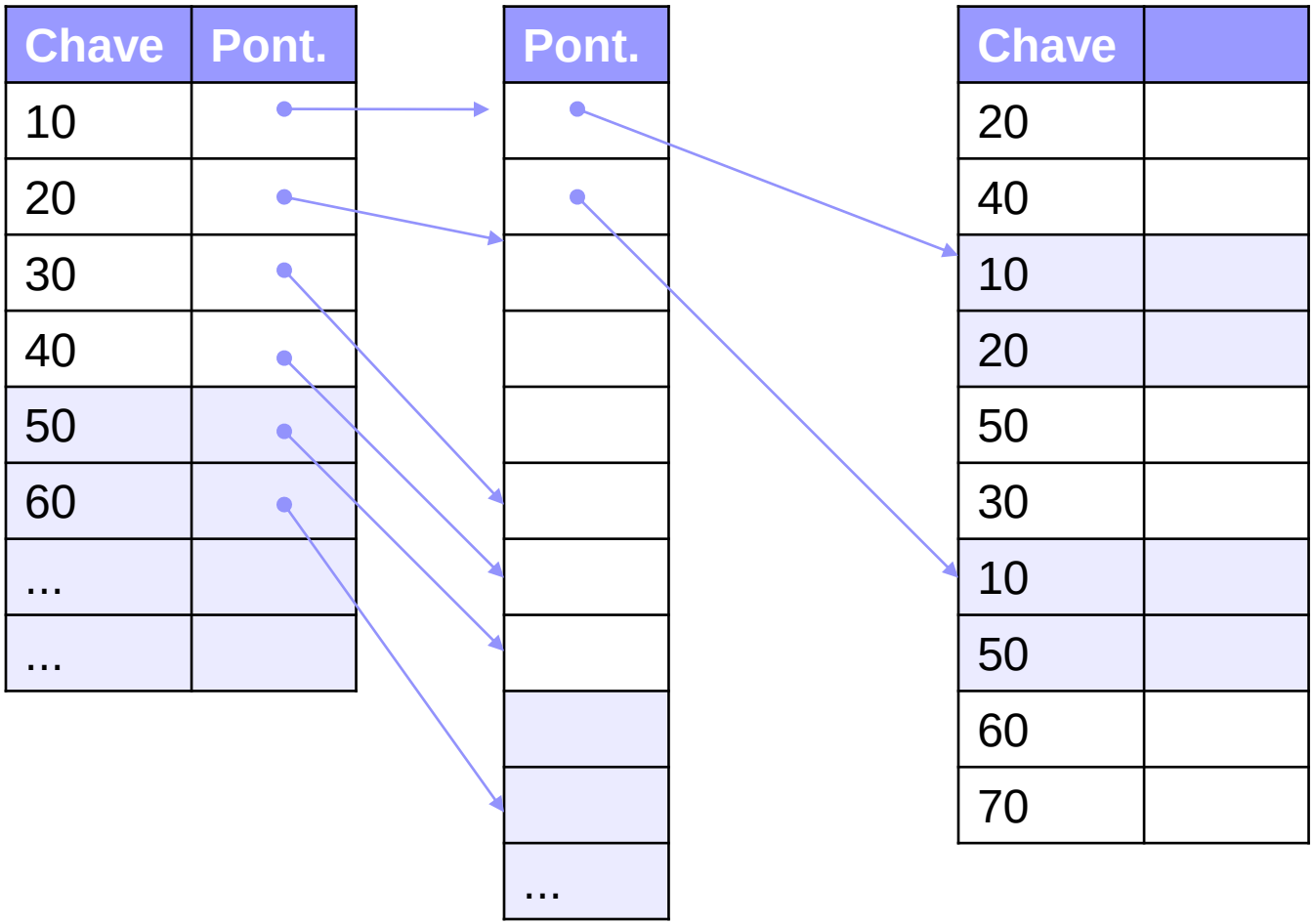
Índice Secundário

- Problema:
  - O uso de um índice secundário pode resultar em muito mais operações de E/S do disco.

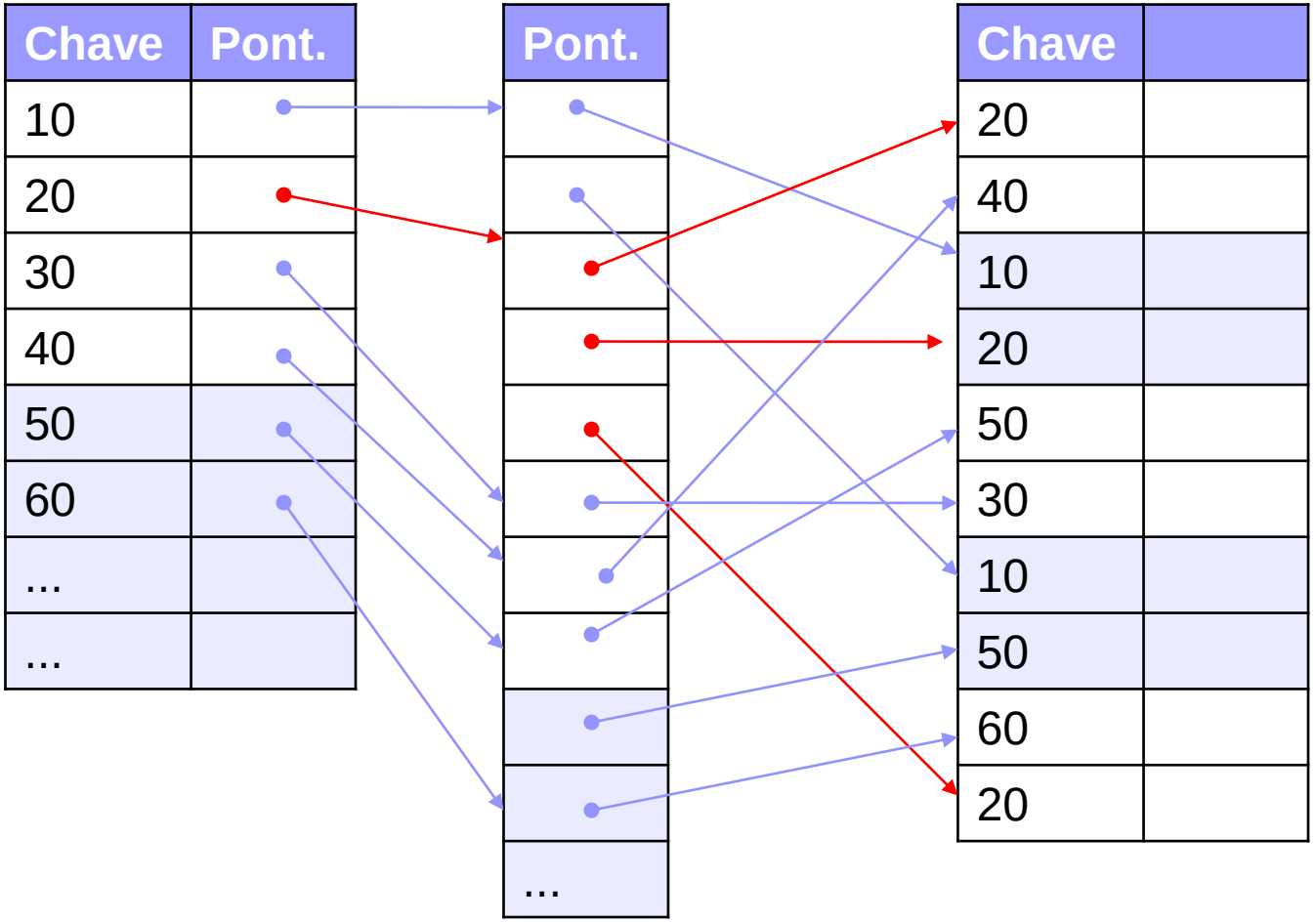
Índice Secundário para Não Chave Primária

- Solução:
  - Criar estruturas que ‘agrupem’ os valores de chave.
  - Uma forma conveniente de evitar a repetição de valores é usar um nível de caráter indireto, chamado depósito (*bucket*), entre o arquivo de índices secundários e o arquivo de dados.
    - Caráter indireto em índices secundários.

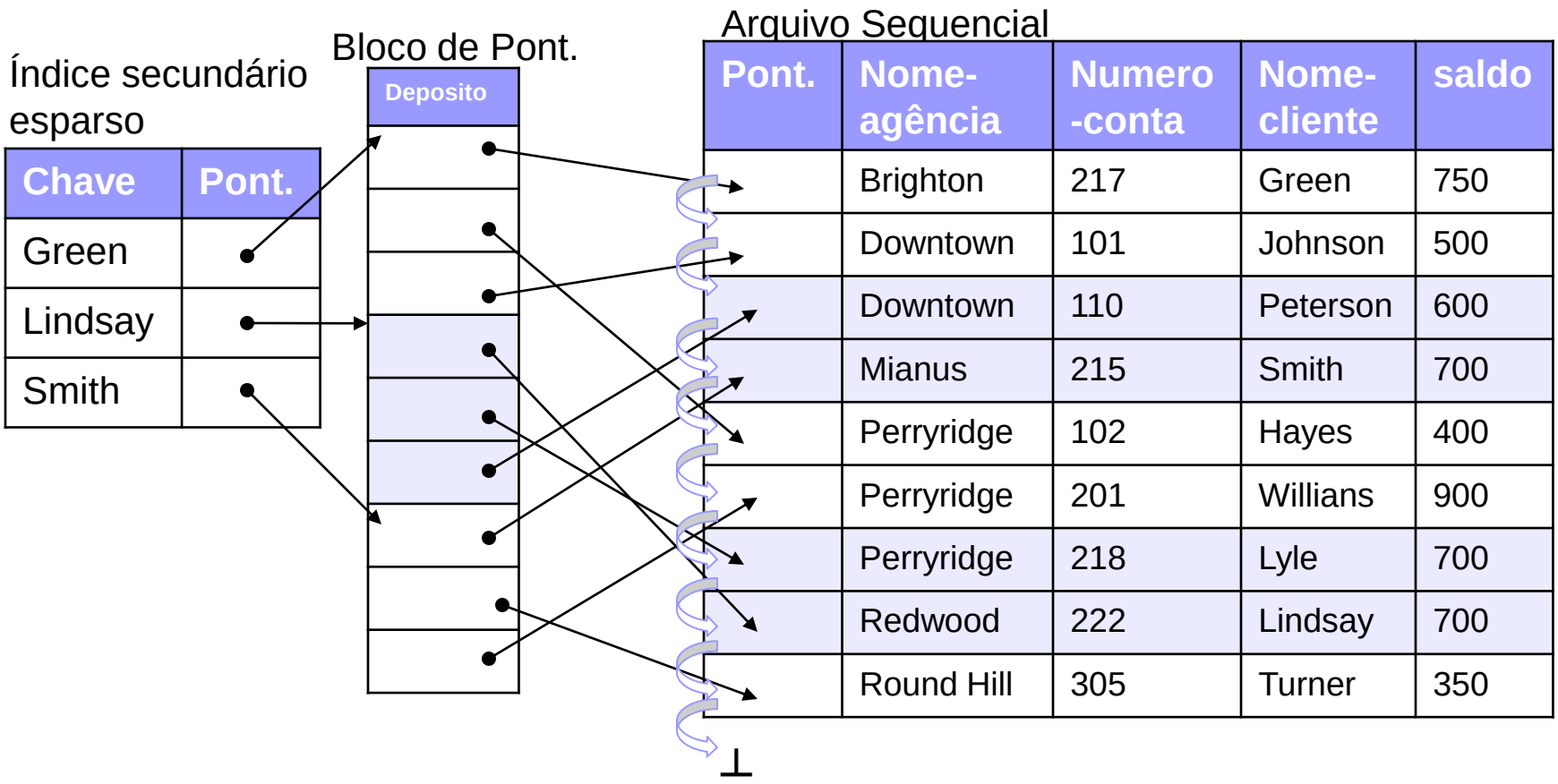
Caráter indireto em índices secundários



Caráter indireto em índices secundários



Exemplo – Índice Secundário



Índice Secundário para Não Chave Primária

Endereço	CH	END
1	DART	2
2	DEA	3
3	DEC	5
4	DEE	4
5	DSC	1

ÍNDICE

Endereço	End_1	End_2	End_3	Próx
1	1	3	8	6
2	2	5		0
3	4			0
4	6	7	10	0
5	9	11	14	7
6	12	13		0
7	15			0

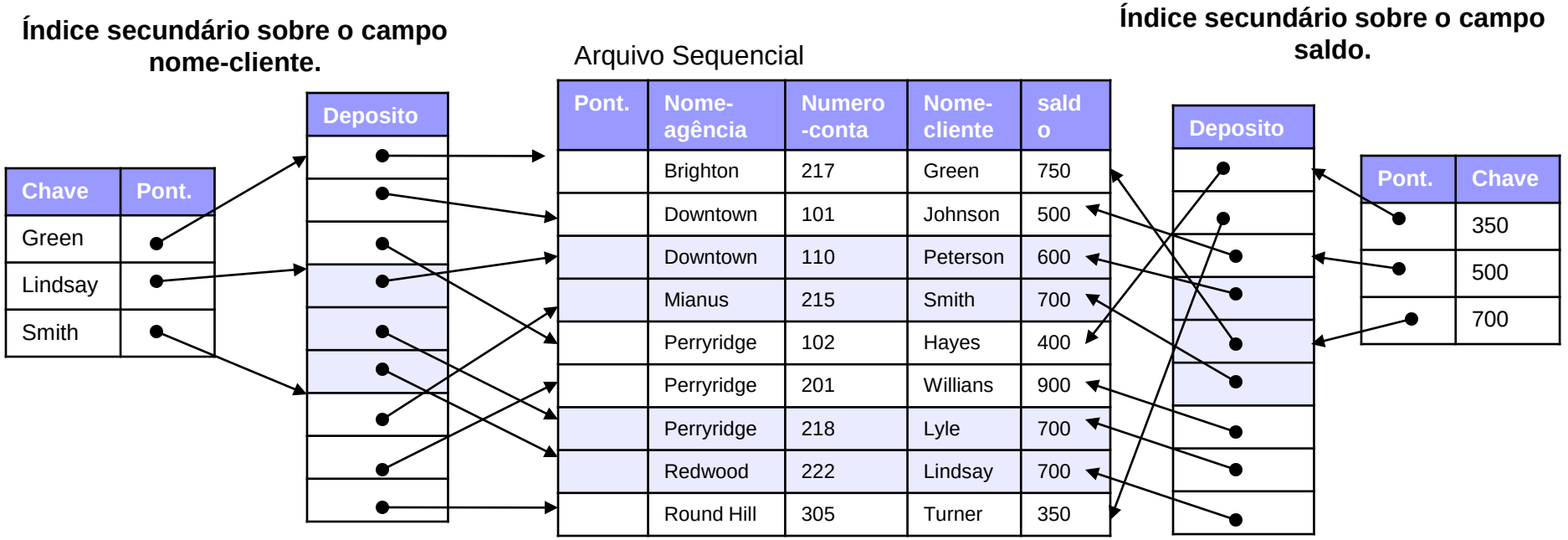
BL. REG. DE APONTADORES

Endereço	CÓDIGO	NOME	DEPTO
1	100	João	DSC
2	200	Maria	DART
3	400	Ana	DSC
4	440	Lucas	DEA
5	500	Pedro	DART
6	560	Renan	DEE
7	600	Mariana	DEE
8	660	Luana	DSC
9	700	José	DEC
10	720	Paulo	DEE
11	810	Beatriz	DEC
12	880	Yara	DSC
13	900	Yalli	DSC
14	990	Júlia	DEC
15	999	Júlio	DEC

ARQUIVO DE DADOS

Exemplo – Índice Secundário

- Um mesmo arquivo pode ter vários índices secundários.



Índices Secundários

- A cada mudança no banco de dados, todos os índices secundários (e depósitos) precisam ser modificados.

Fatores que tornam a pesquisa baseada em índices mais eficiente do que parece

- O número de blocos de índices em geral é pequeno quando comparado com o número de blocos de dados.
- Tendo em vista que as chaves são classificadas, podemos usar a pesquisa binária para encontrar uma determinada chave.
- Se houver *n* blocos do índice, bastará examinar $\log_2 n$ deles.
- O índice pode ser pequeno o bastante para ser mantido permanente em buffers da memória principal.

Classificação de Índices Ordenados

- Considerando a quantidade de entradas
  - Denso
  - Esparso
- Considerando a organização do arquivo
  - Primário
  - Clustering
  - Secundário
- Considerando os níveis de indirecionamento
  - Mononível
  - Multinível

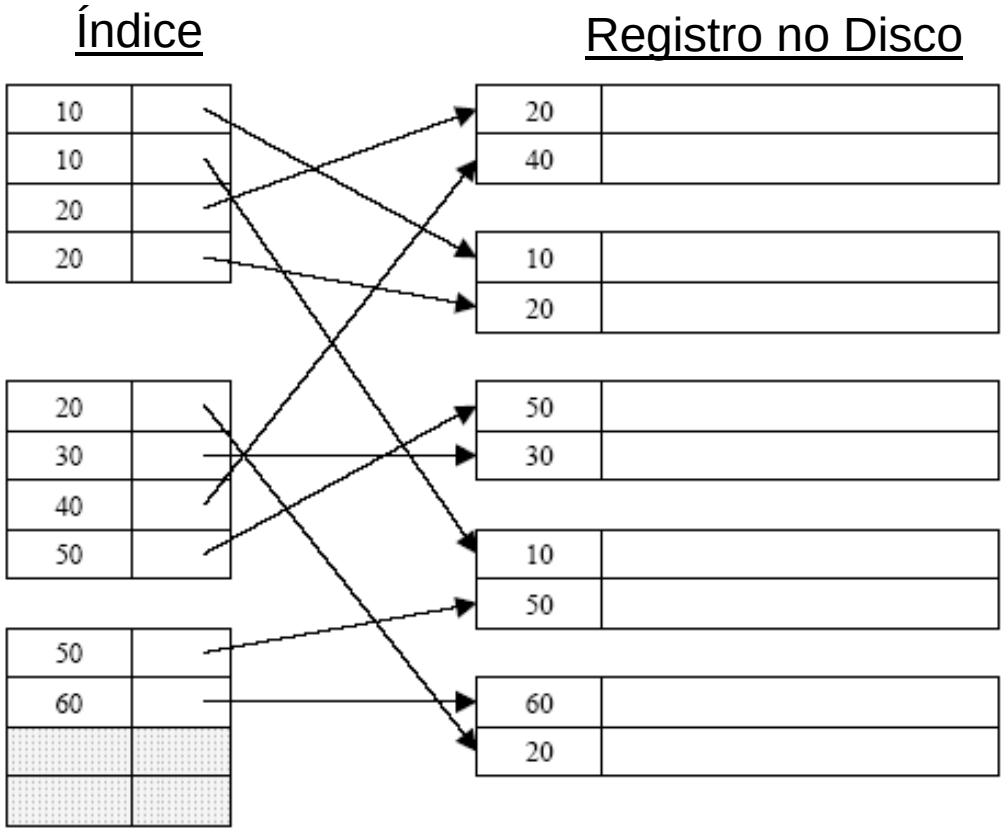
UNIVERSIDADE FEDERAL DE ITAJUBÁ

ÍNDICES ORDENADOS

MONONÍVEL X MULTINÍVEL

Índice Mononível

- Todos os índices visto até agora possuem apenas um nível de indireção.



Índice Multinível

- Um índice pode cobrir sozinho muitos blocos (índices muito grandes).
- Se esses blocos não estiverem em algum lugar onde saibamos que é possível encontrá-los, por exemplo em cilindros designados de um disco, então talvez seja necessária outra estrutura de dados para localizá-los.
- Mesmo que isto ocorra, talvez ainda seja preciso executar muitas operações de E/S de disco para alcançar o registro que queremos localizar.
- Inserindo um índice no índice, poderemos tornar o uso do primeiro nível de índices mais eficiente.

Índice Multinível

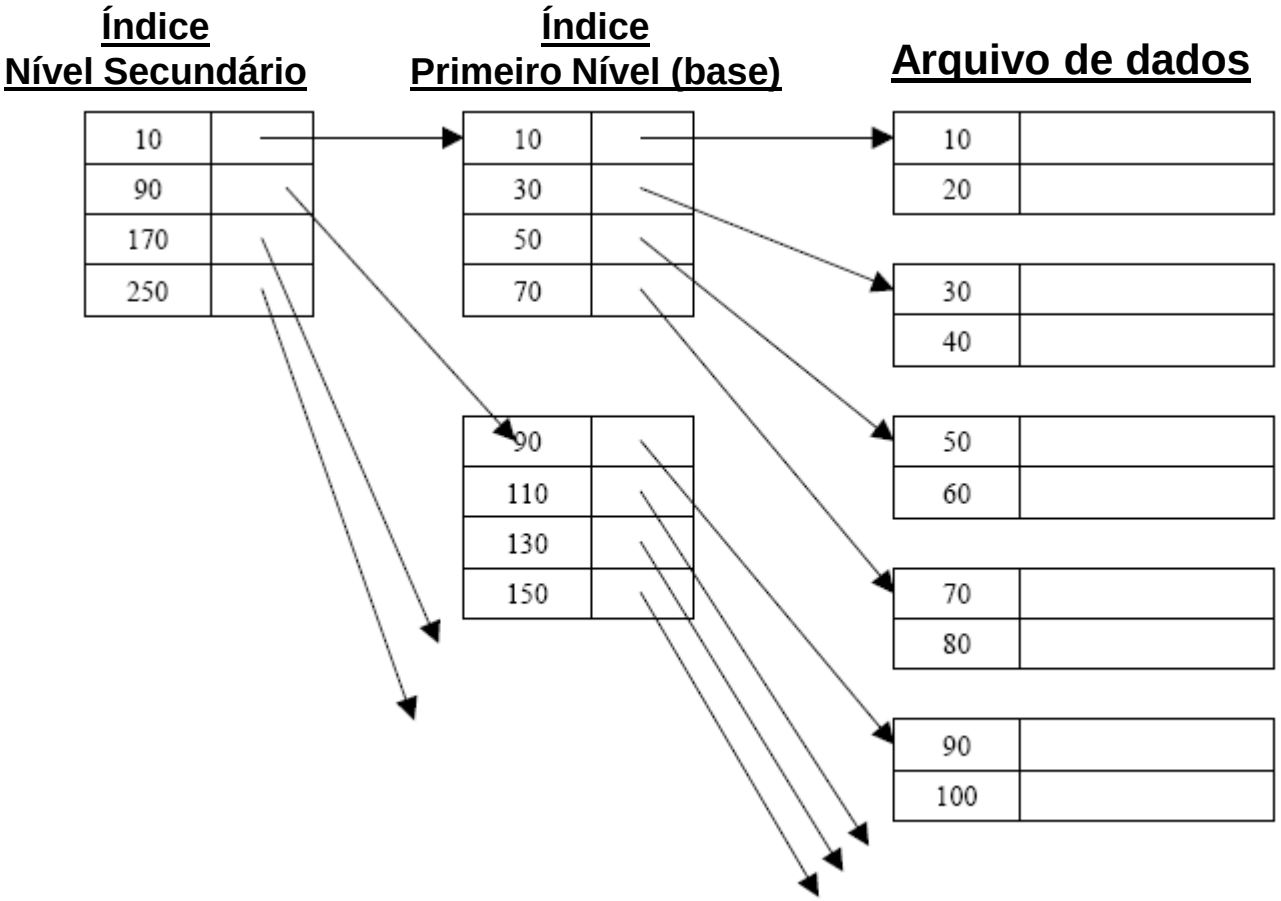
- Um índice multinível considera o arquivo índice, o qual chamaremos agora de primeiro nível (ou base) do índice multinível, **como um arquivo ordenado com um valor distinto para cada k($_i$).**

- Portanto, podemos criar um índice principal para o primeiro nível.

- Esse índice para o primeiro nível é chamado de segundo nível do índice multinível.

- Uma vez que o segundo nível é um índice principal, podemos utilizar âncoras de blocos de forma que o segundo nível possua uma entrada para cada bloco do primeiro nível.

Índice Multinível



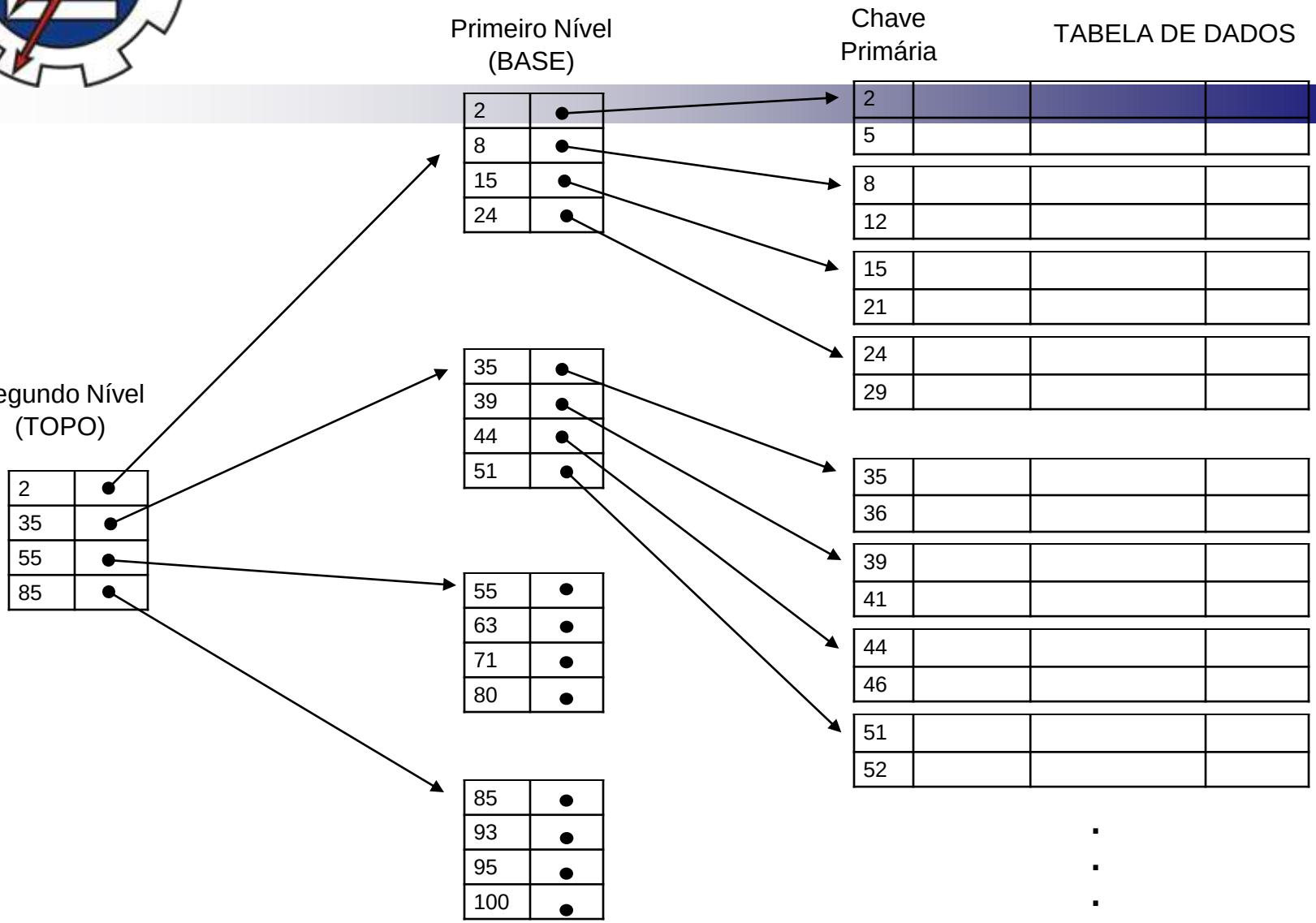
Índice Multinível

- Dá-se o nome de ***fator de bloco do índice*** (bfr_i) para o número de registros lógicos que cabem em um registro físico.

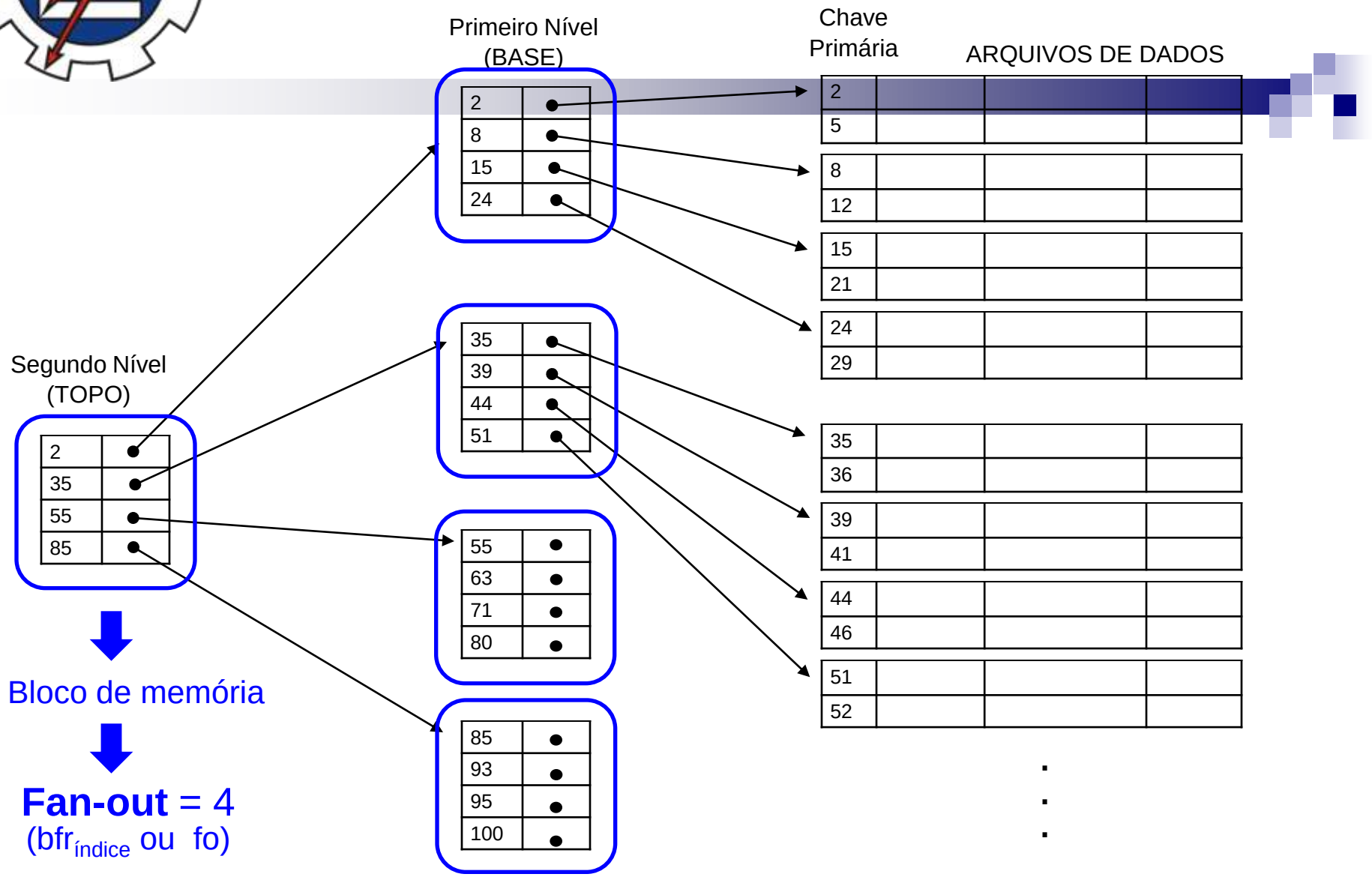
- O valor de bfr_i é chamado de ***fan out*** (fo) do índice multinível.

- Um índice multinível com r entradas de primeiro nível terá aproximadamente t níveis, onde $t = \lceil \log_{fo}(r) \rceil$.

Índices Multinível – t = 2



Índices Multinível – t = 2



Índice Multinível

- Embora um ou dois níveis de índices sejam com frequencia muito úteis para acelerar consultas, há uma estrutura mais geral usada comumente em SGBDs comerciais por serem mais flexíveis e escaláveis.

- A família geral de estruturas de dados é chamada árvore B, e a variante utilizada com mais frequência é conhecida como árvore B+.

Índice Multinível

- Basicamente, a árvore B:
  - Automaticamente mantém os níveis balanceados para a quantidade de dados que está sendo indexada, e,
  - Gerencia o espaço usado por seus blocos para que eles sempre estejam ocupados com pelo menos a metade de sua capacidade.

UNIVERSIDADE FEDERAL DE ITAJUBÁ

ÍNDICES ORDENADOS MULTINÍVEL

ÁRVORE B X ÁRVORE B+

Árvore B

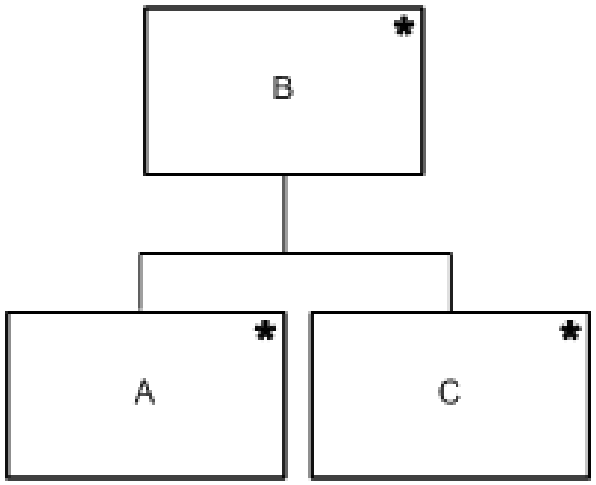
- As árvores B são árvores **balanceadas** que visam otimizar as operações de entrada e saída nos dispositivos.
- Em uma aplicação comum de uma árvore B, a quantidade de dados é tão grande que provavelmente não caberia na memória principal.
- A árvore B copia blocos específicos para a memória principal quando necessário e os grava no disco se os blocos tiverem sido alterados.

Árvore B

- O nó raiz tem no mínimo 2 sub-árvores e no máximo, n sub-árvores.
  - n é a **ordem** da árvore B

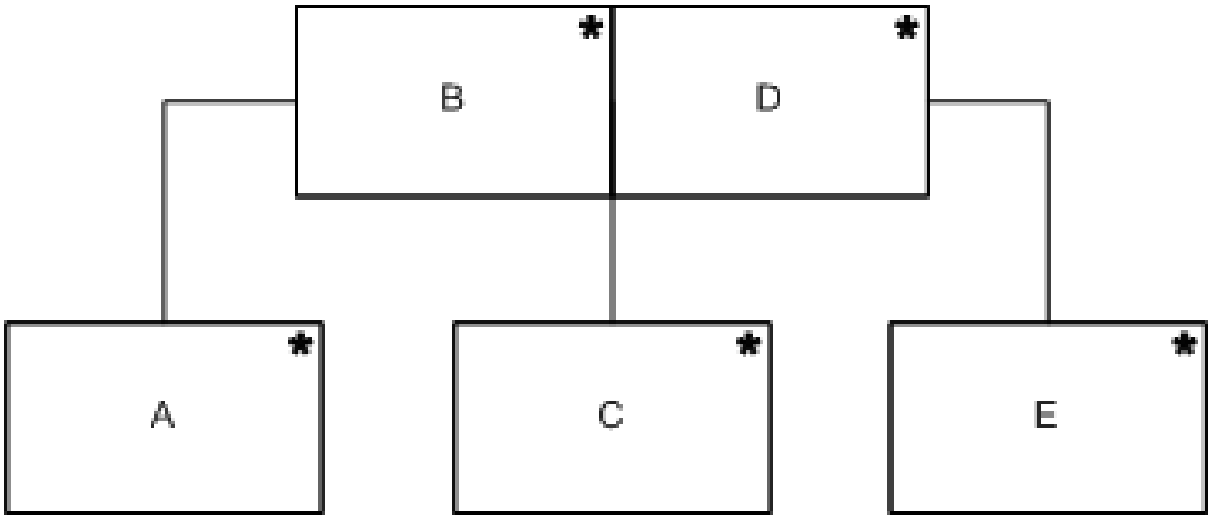
Árvore B

- Exemplo – ordem 2 (mínimo)



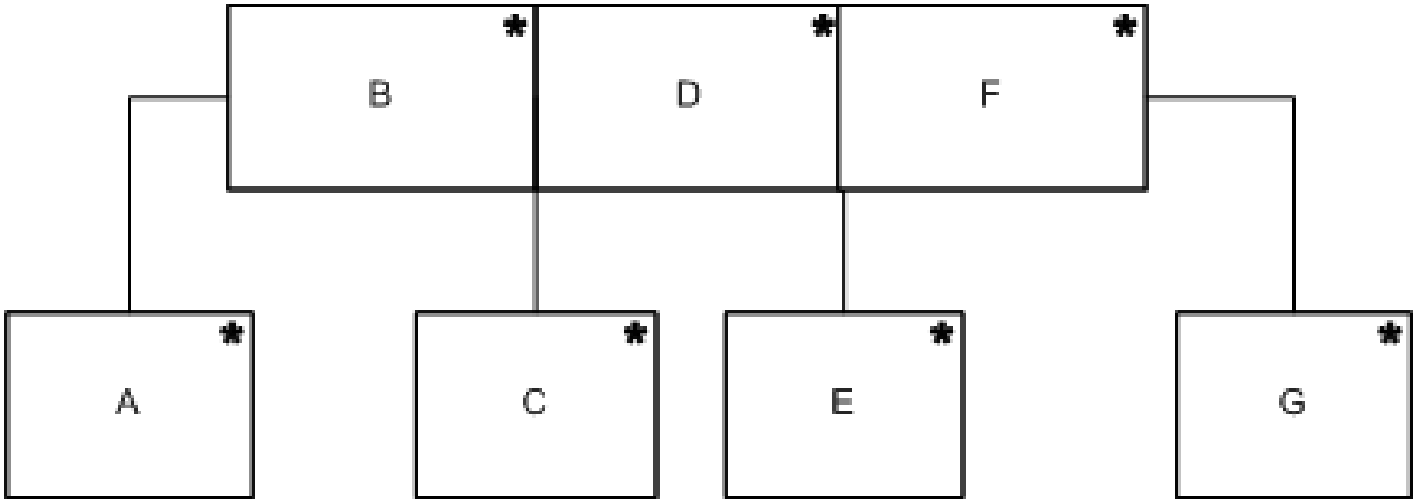
Árvore B

- Exemplo – ordem 3



Árvore B

- Exemplo – ordem 4

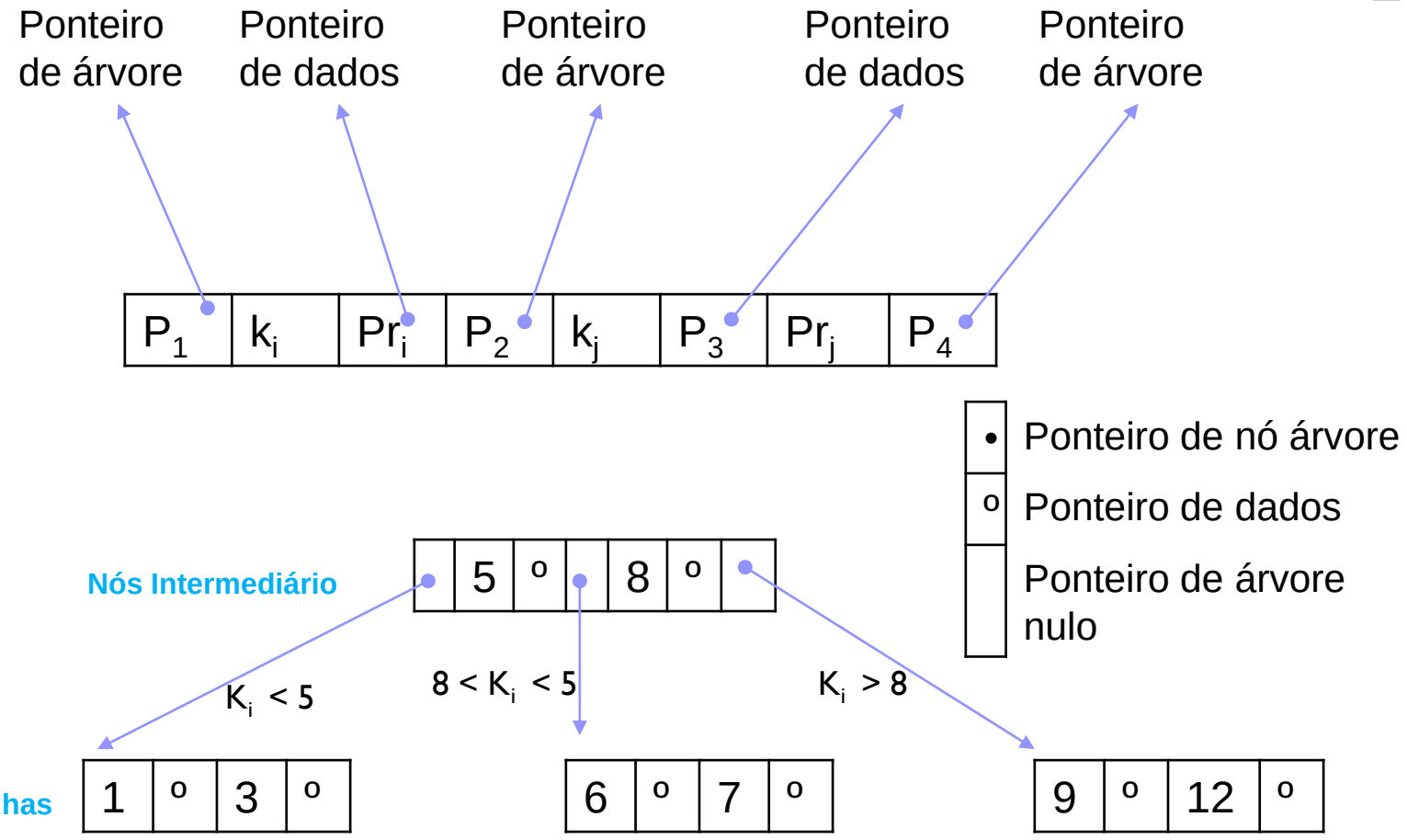


Árvore B

4. Cada nó da Árvore B, por definição, possui restrições quanto à quantidade de chaves.
   - Existe um número máximo e mínimo de filhos em um nó. Este número pode ser descrito em termos de um inteiro fixo ***t*** maior ou igual a 2 chamado **grau mínimo**.
   - Cada nó, exceto a raiz, precisa ter pelo menos t-1 chaves.
   - Cada nó possui no máximo 2t-1 chaves, para que assim cada nó interno tenha no máximo 2t filhos.

5. O número máximo de filhos para cada nó determina a ordem “m” de uma árvore B.

Nó de uma árvore B

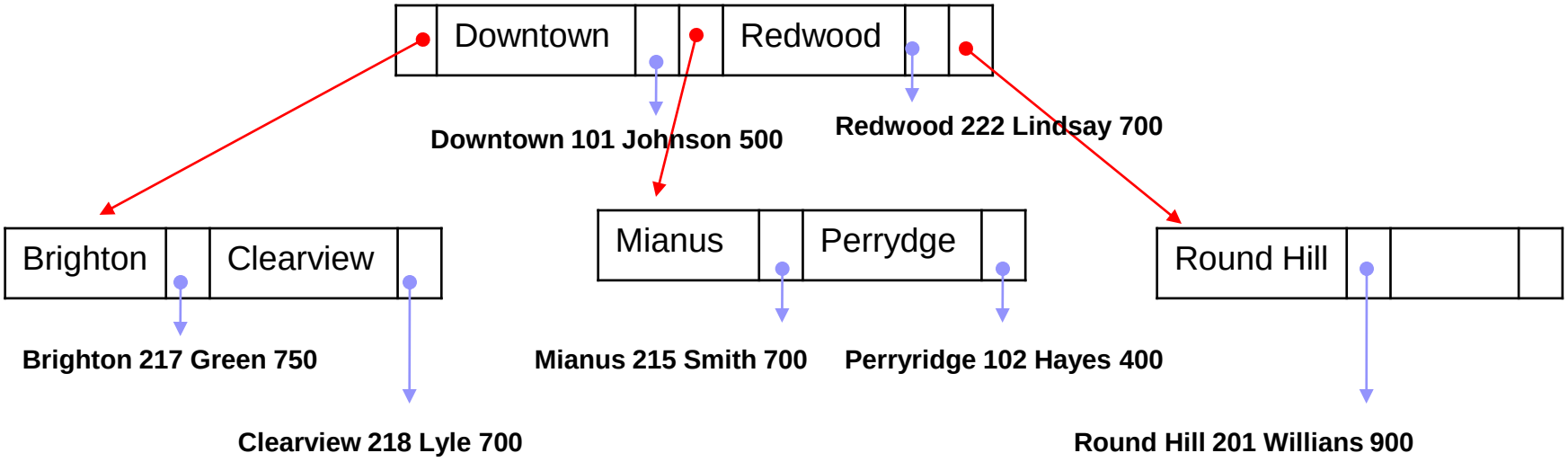


Exemplo - Relação

- Relação depósito no banco de dados bancário.

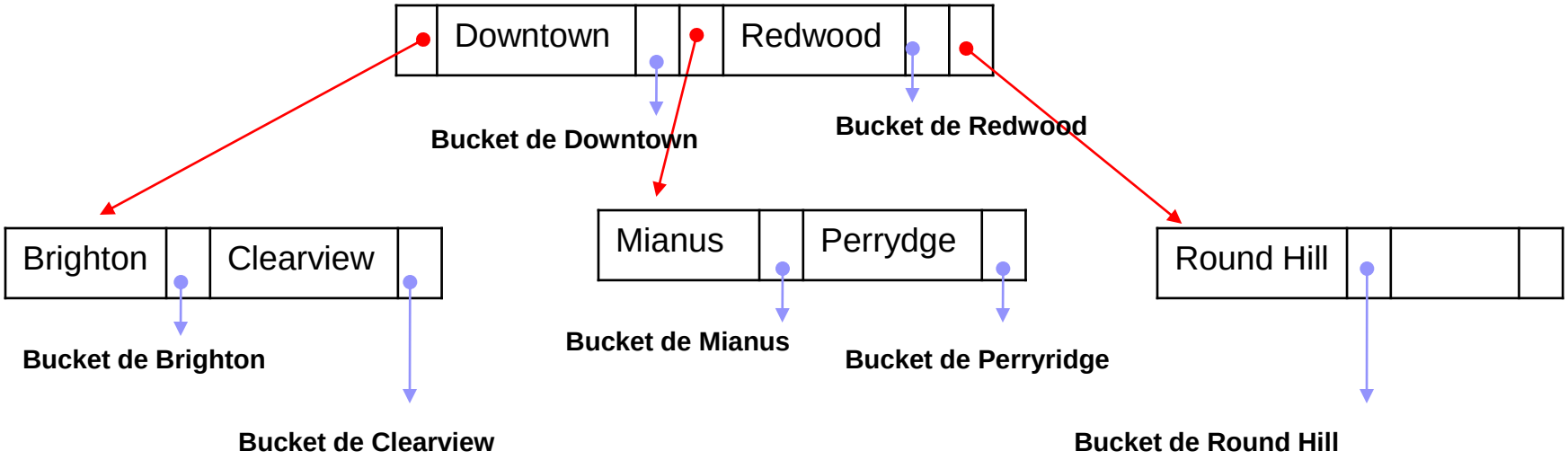
Registro	Nome-agência	Numero-conta	Nome-cliente	saldo
0	Perryridge	102	Hayes	400
1	Round Hill	305	Turner	350
2	Mianus	215	Smith	700
3	Downtown	101	Johnson	500
4	Redwood	222	Lindsay	700
5	Round Hill	201	Willians	900
6	Brighton	217	Green	750
7	Clearview	218	Lyle	700

Exemplo



A árvore B gerencia o espaço usado por seus blocos para que eles sempre estejam ocupados com pelo menos a metade de sua capacidade.

Exemplo
E se a chave se repetir?

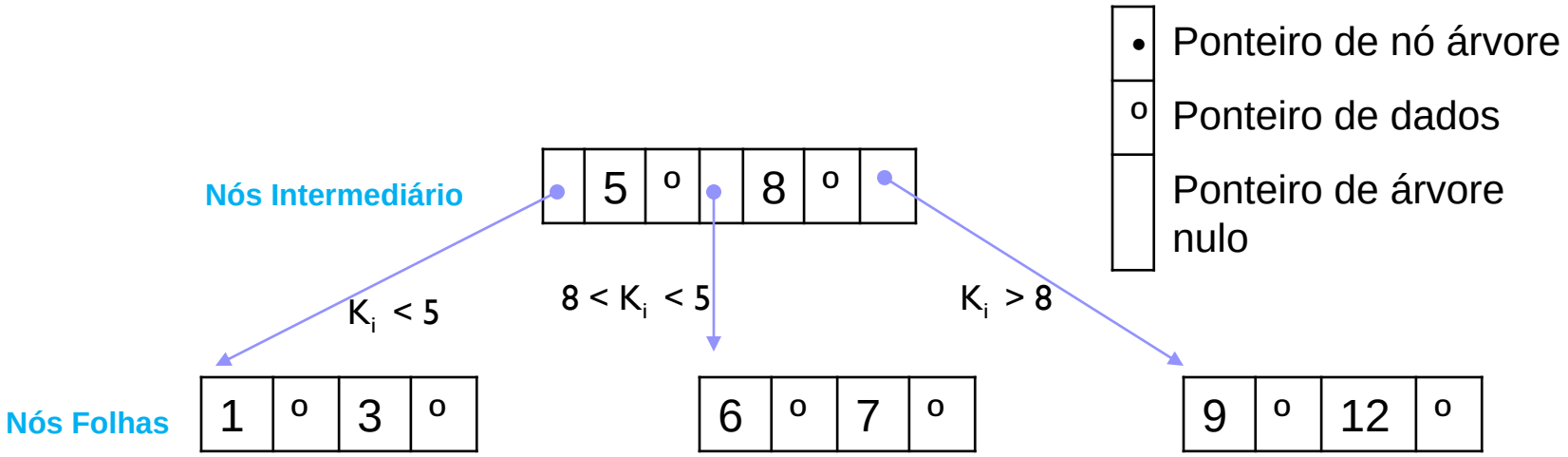


Árvore B+

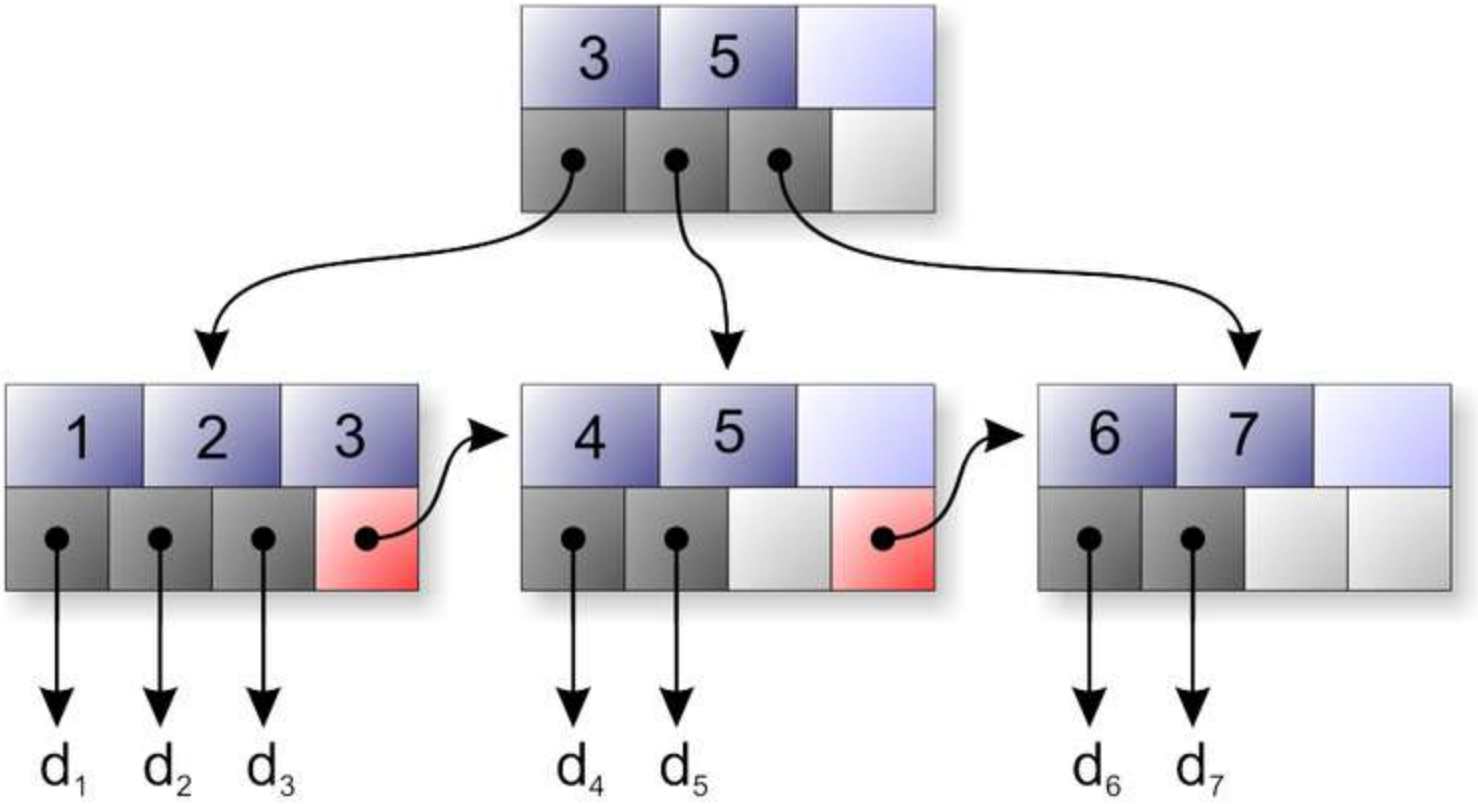
- Na árvore B, uma chave somente é entrada uma vez em algum nível da árvore.
- Já na árvore B+, todos os dados só são armazenados nas *folhas.*
- Desta maneira, a estrutura conceitual das folhas difere da estrutura dos nós internos.
- As folhas da árvore B+ estão ligadas em sequência, tornando possível o acesso ordenado a seus campos.

Nó de uma árvore B

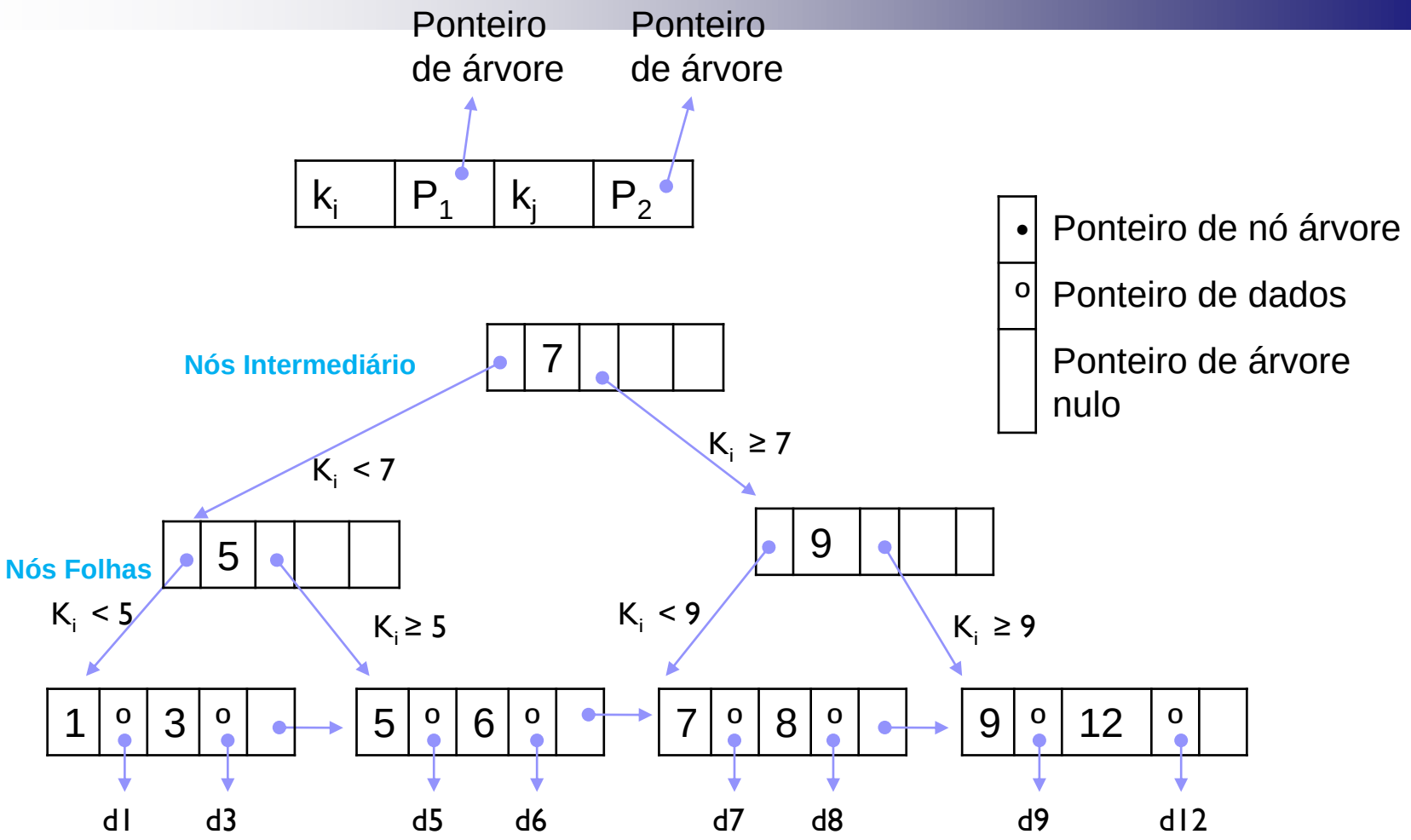
- A chave ‘5’ aparece uma única vez na árvore.
- O nó da árvore B tem um ponteiro para os dados referentes a chave.



Árvore B+



Nó de uma árvore B+

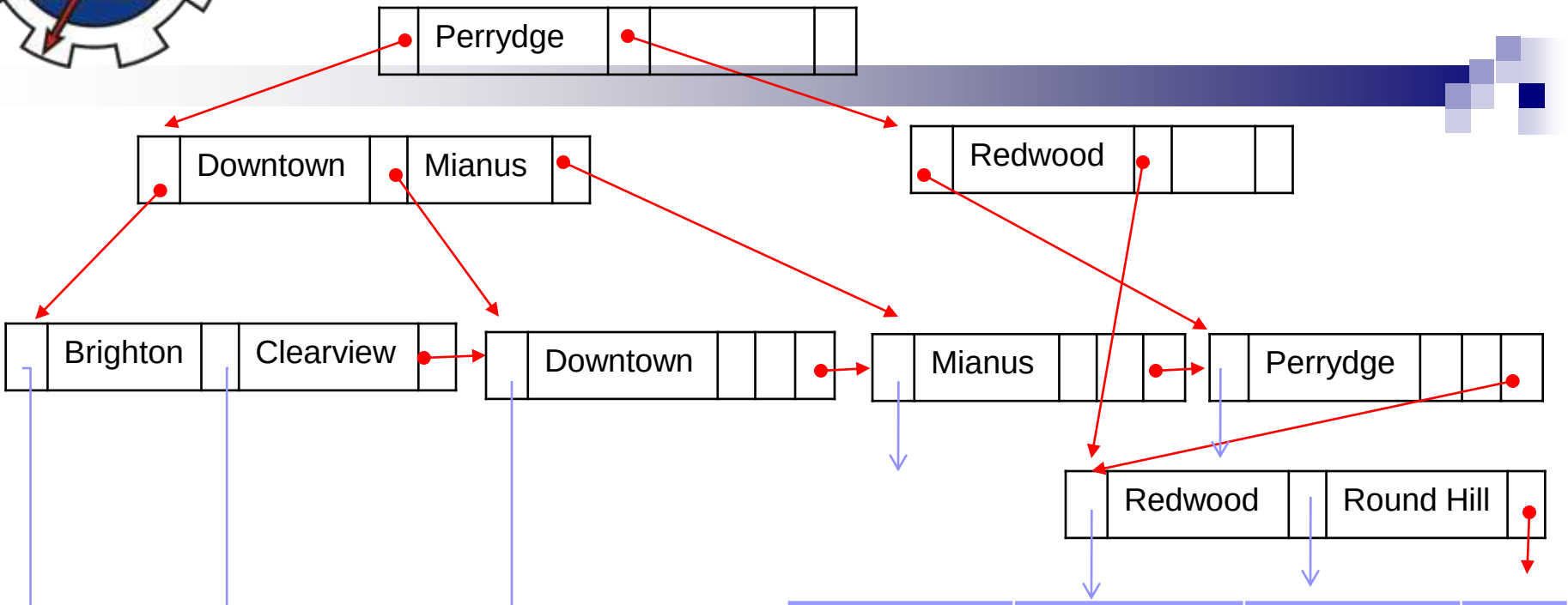


Exemplo - Relação

- Relação depósito no banco de dados bancário.

Registro	Nome-agência	Numero-conta	Nome-cliente	saldo
0	Perryridge	102	Hayes	400
1	Round Hill	305	Turner	350
2	Mianus	215	Smith	700
3	Downtown	101	Johnson	500
4	Redwood	222	Lindsay	700
5	Round Hill	201	Willians	900
6	Brighton	217	Green	750
7	Clearview	218	Lyle	700

Exemplo



Nome-agência	Numero-conta	Nome-cliente	saldo
Brighton	217	Green	750
Clearview	218	Lyle	700
Downtown	101	Johnson	500
Mianus	215	Smith	700
Perryridge	102	Hayes	400
Redwood	222	Lindsay	700
Round Hill	305	Turner	350

Vantagens Árvore B+

- Embora a inserção e remoção em árvore B+ sejam complicadas, elas requerem relativamente poucas operações.

- É a velocidade de operações em árvores B+ que as torna uma estrutura de índice usada frequentemente em implementações de bancos de dados.

Vantagens Árvore B+

- Mecanismo para percorrer sequencialmente o arquivo de registros de dados sem que seja necessário visitar toda a árvore.

- Mecanismo para percorrer sequencialmente o arquivo de registros de dados sem que seja necessário ordenar o arquivo de registro de dados.

Vantagens
Árvore B sobre a B+

- Ausência de armazenamento redundante de chaves de busca;
- Possibilidade de encontrar uma chave sem chegar até um nó folha;
  - Busca mais rápida

Vantagens
Árvore B+ sobre a B

- Nó folha e não-folha são do mesmo tamanho
  - Facilita o gerenciamento do armazenamento para o índice;
- A remoção é mais simples, pois a entrada a ser removida sempre estará numa folha.

Árvore B X Árvore B+

- As vantagens da árvore B acabam quando os índices são muito grandes.

- Assim, a simplicidade estrutural de uma árvore B+ é preferida por muitos implementadores de sistemas de banco de dados.

Índices multinível x mononível

- O uso de índices multinível diminui o número de acessos ao disco.
- Por outro lado, esse tipo de índice ocupa mais espaço em disco.
- O problema de inserções e deleções continua porque todos os níveis de índices são arquivos fisicamente ordenados.
- Para resolver esse problema, os projetistas adotaram um índice multinível que deixa algum espaço em cada um de seus blocos para inserir novos registros.
- Esses são chamados índices multinível dinâmicos.

EXERCÍCIOS

2ª lista de exercícios