



*UNIVERSIDADE FEDERAL DE ITAJUBÁ*

# **Banco de Dados II**

## **COM 231**

**Indexação**

**Vanessa Cristina Oliveira de Souza**



# 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 indireccionamento
  - ☐ Mononível
  - ☐ Multinível



*UNIVERSIDADE FEDERAL DE ITAJUBÁ*

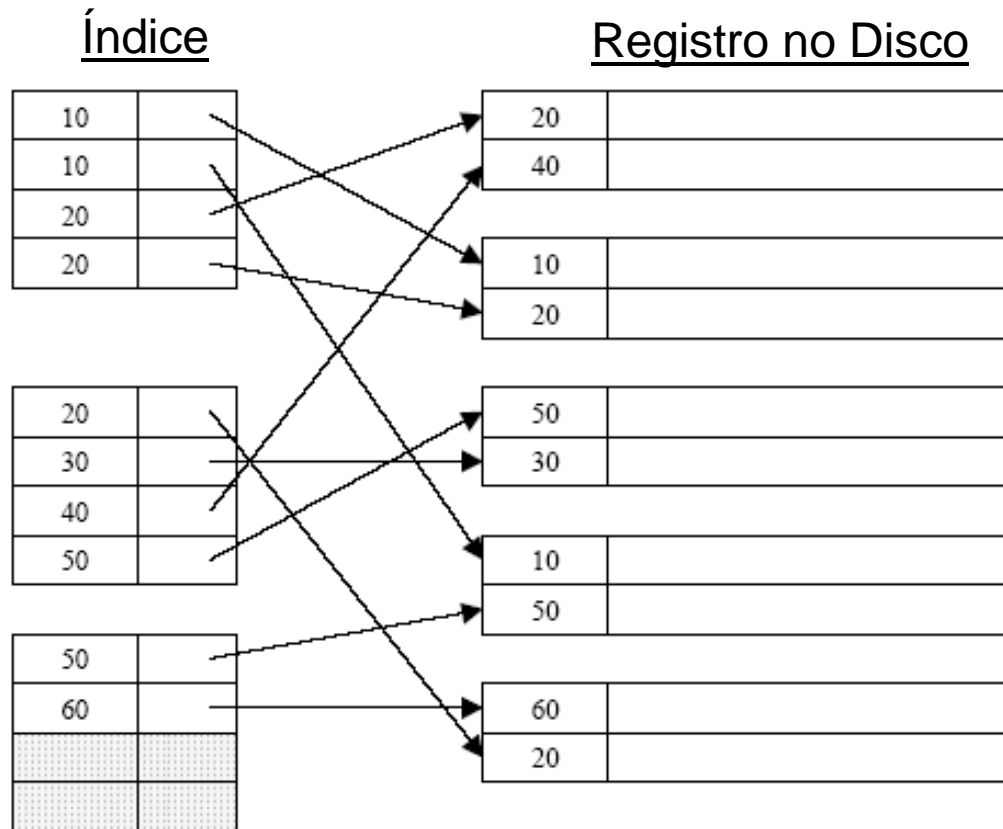
# **ÍNDICES ORDENADOS**

## **MONONÍVEL X MULTINÍVEL**



# Índice Mononível

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



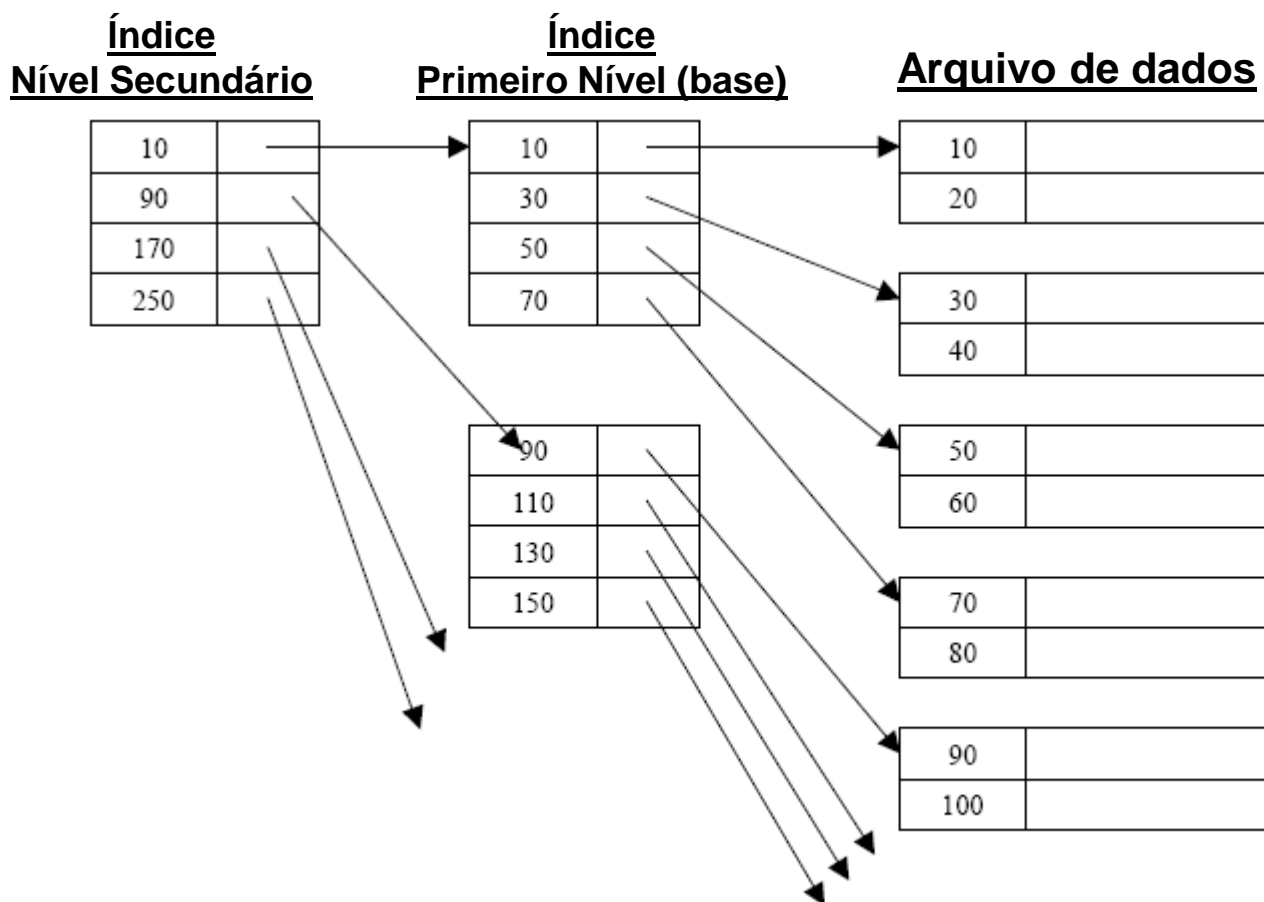


# Índice Multinível

- 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





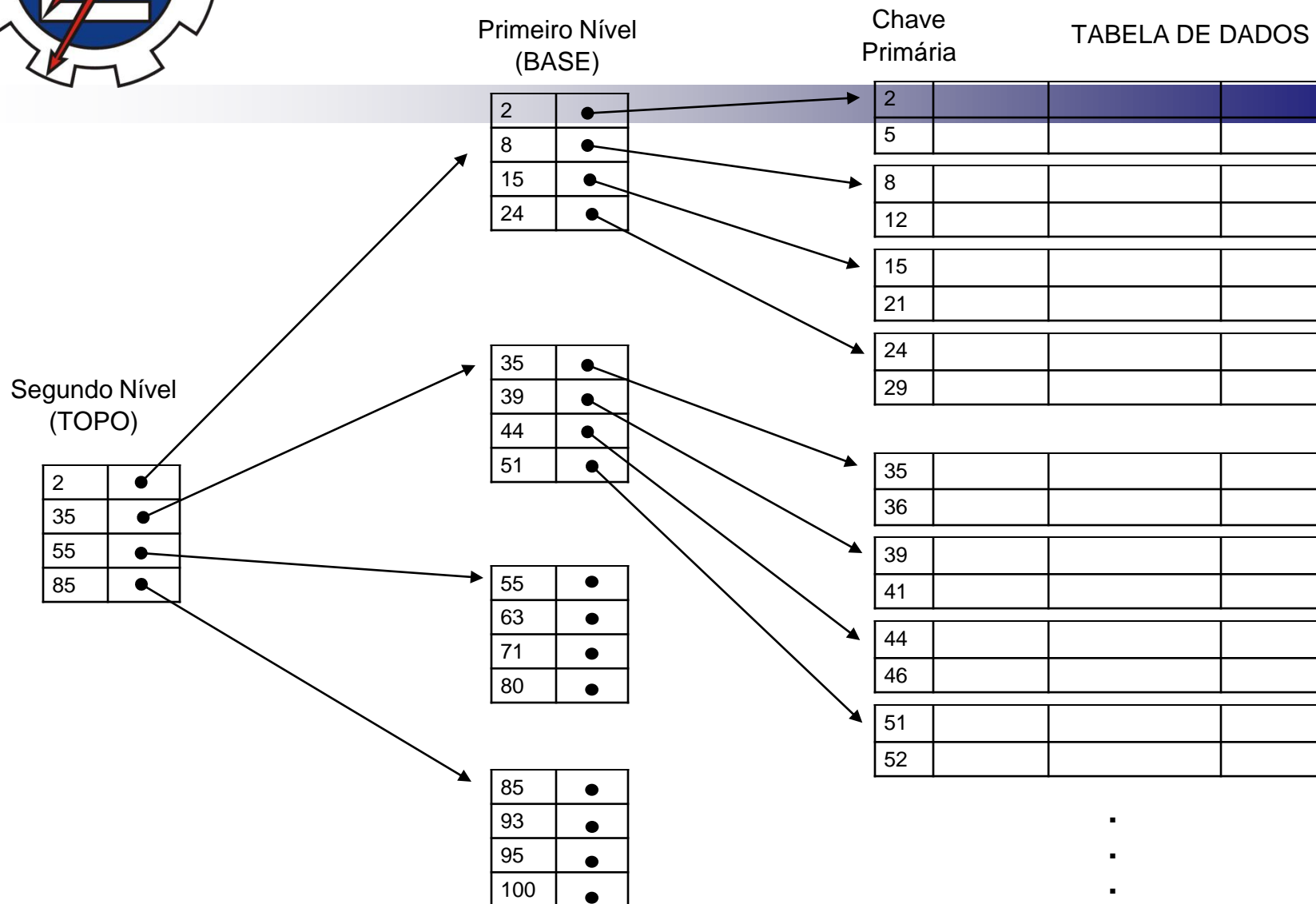
# Índice Multinível

- 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  $t = \lceil \log_{fo}(r) \rceil$ .



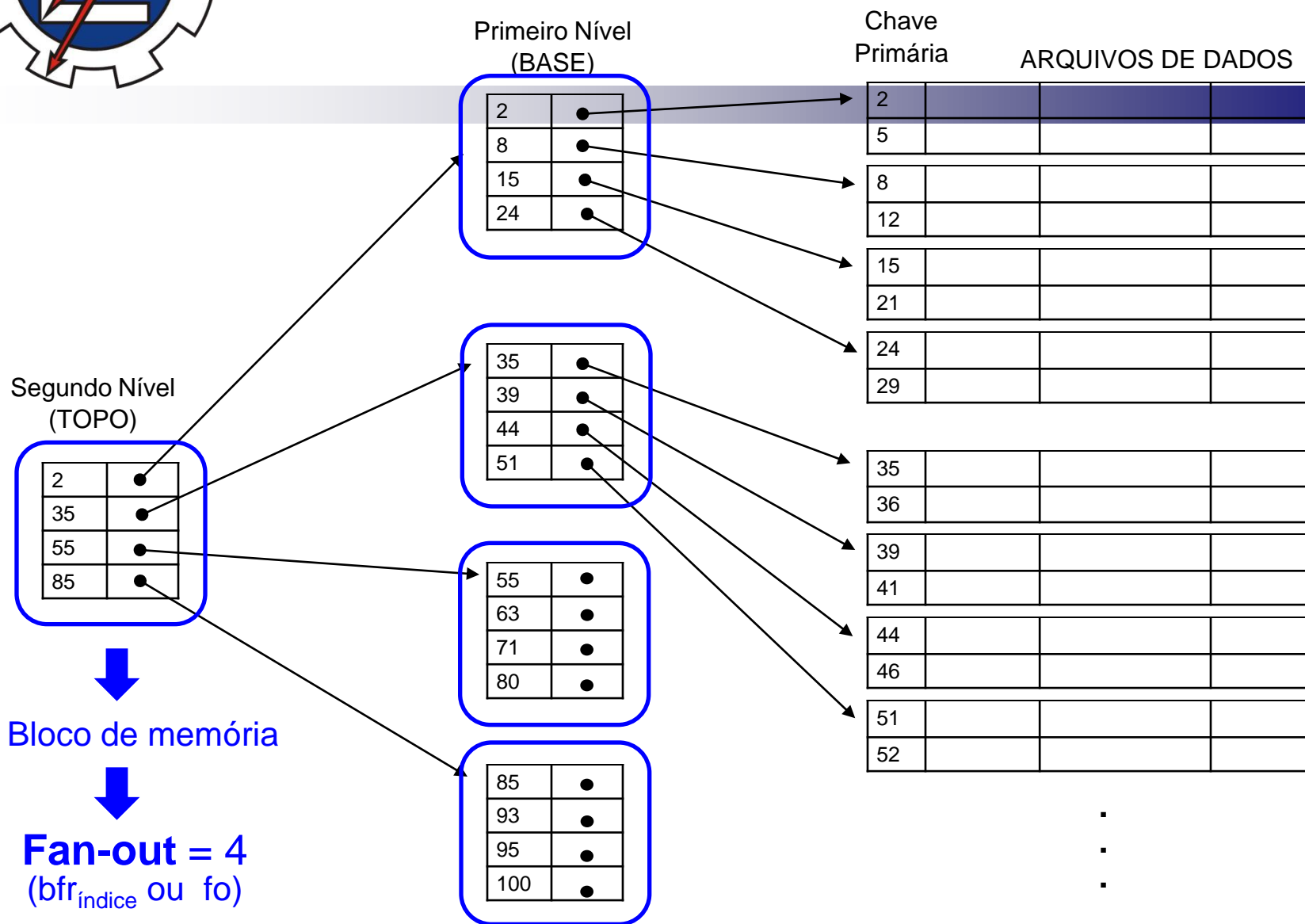


# Índices Multinível – $t = 2$





# Índices Multinível – $t = 2$





# Índice Multinível

- Embora um ou dois níveis de índices sejam com frequência 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<sup>+</sup>**



# 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.



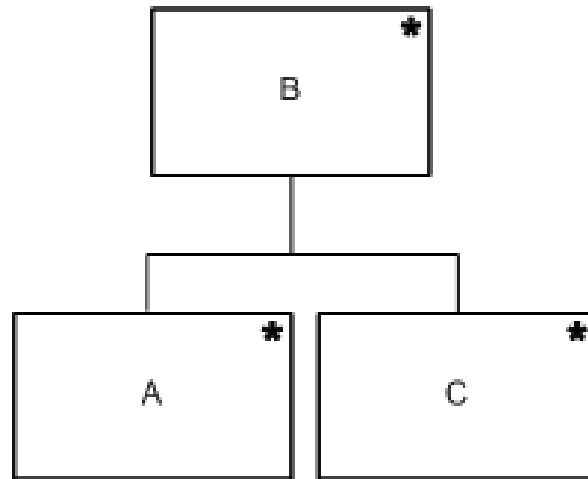
# Árvore B

- 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)

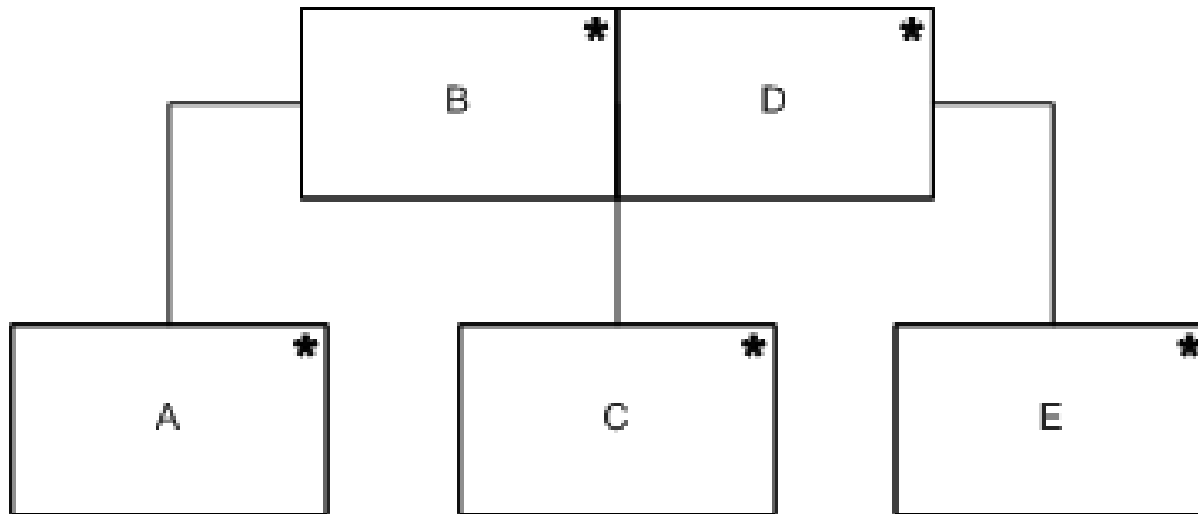






# Árvore B

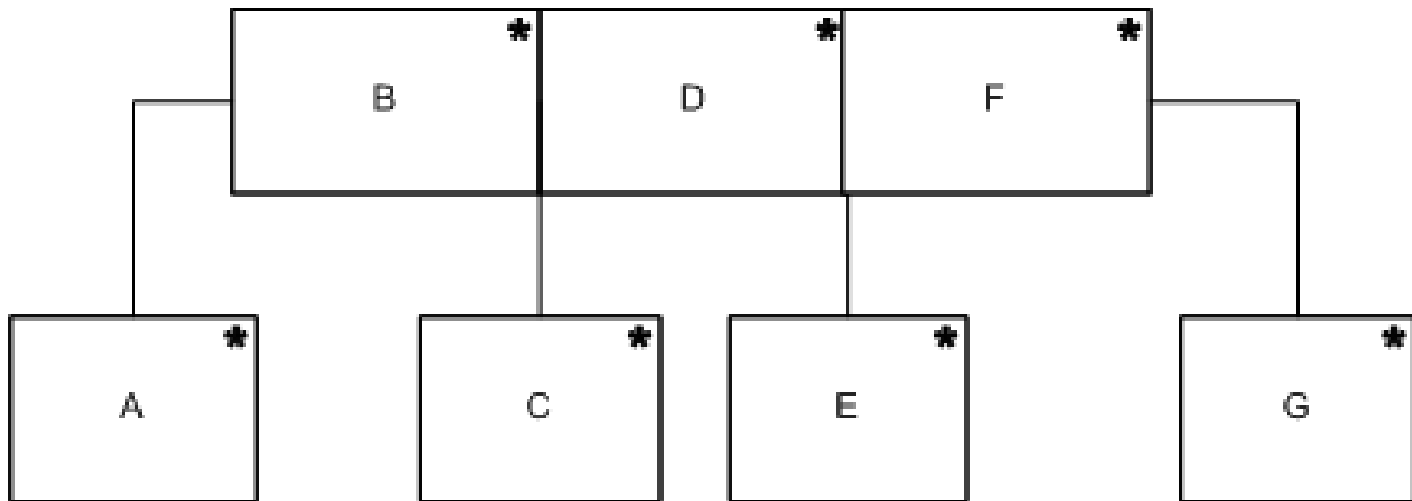
## ■ Exemplo – ordem 3





# Árvore B

## ■ Exemplo – ordem 4





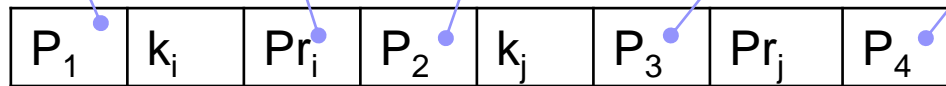
# Árvore B

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.

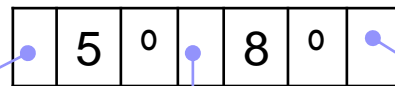


# Nó de uma árvore B

Ponteiro de árvore    Ponteiro de dados    Ponteiro de árvore    Ponteiro de dados    Ponteiro de árvore



Nós Intermediário

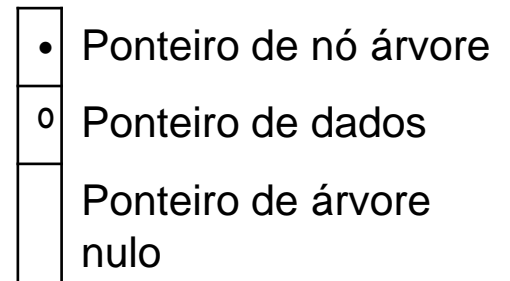
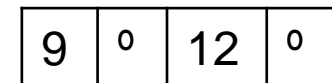
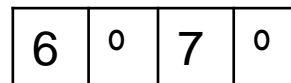
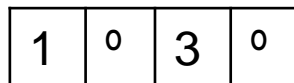


$K_i < 5$

$8 < K_i < 5$

$K_i > 8$

Nós Folhas





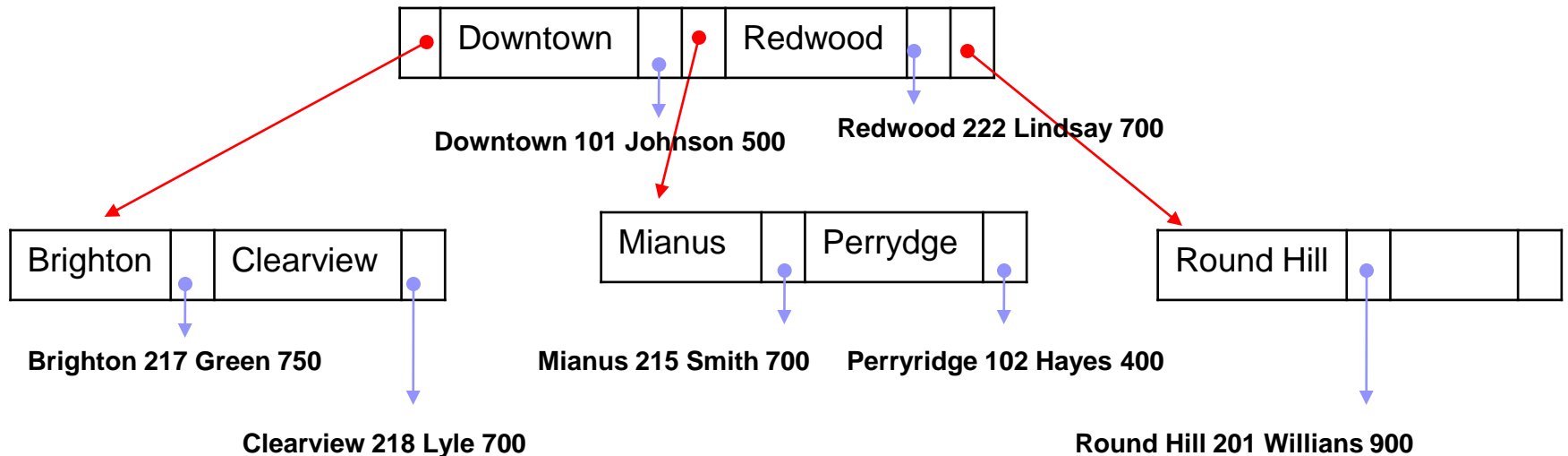
# Exemplo - Relação

## ■ 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	Williams	900
6	Brighton	217	Green	750
7	Clearview	218	Lyle	700



# Exemplo

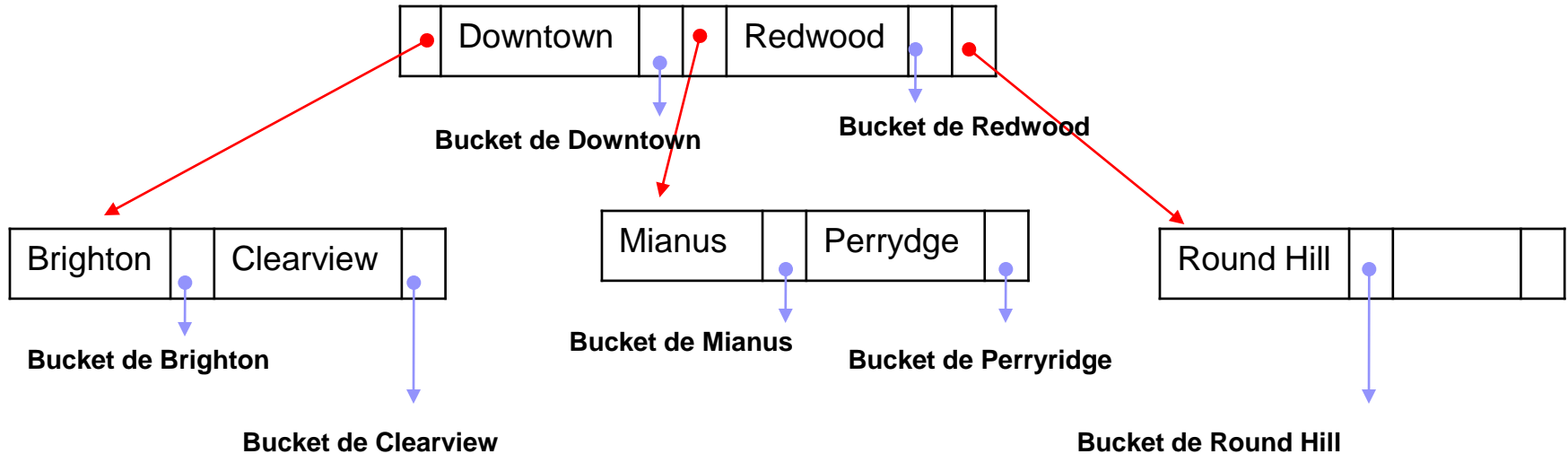


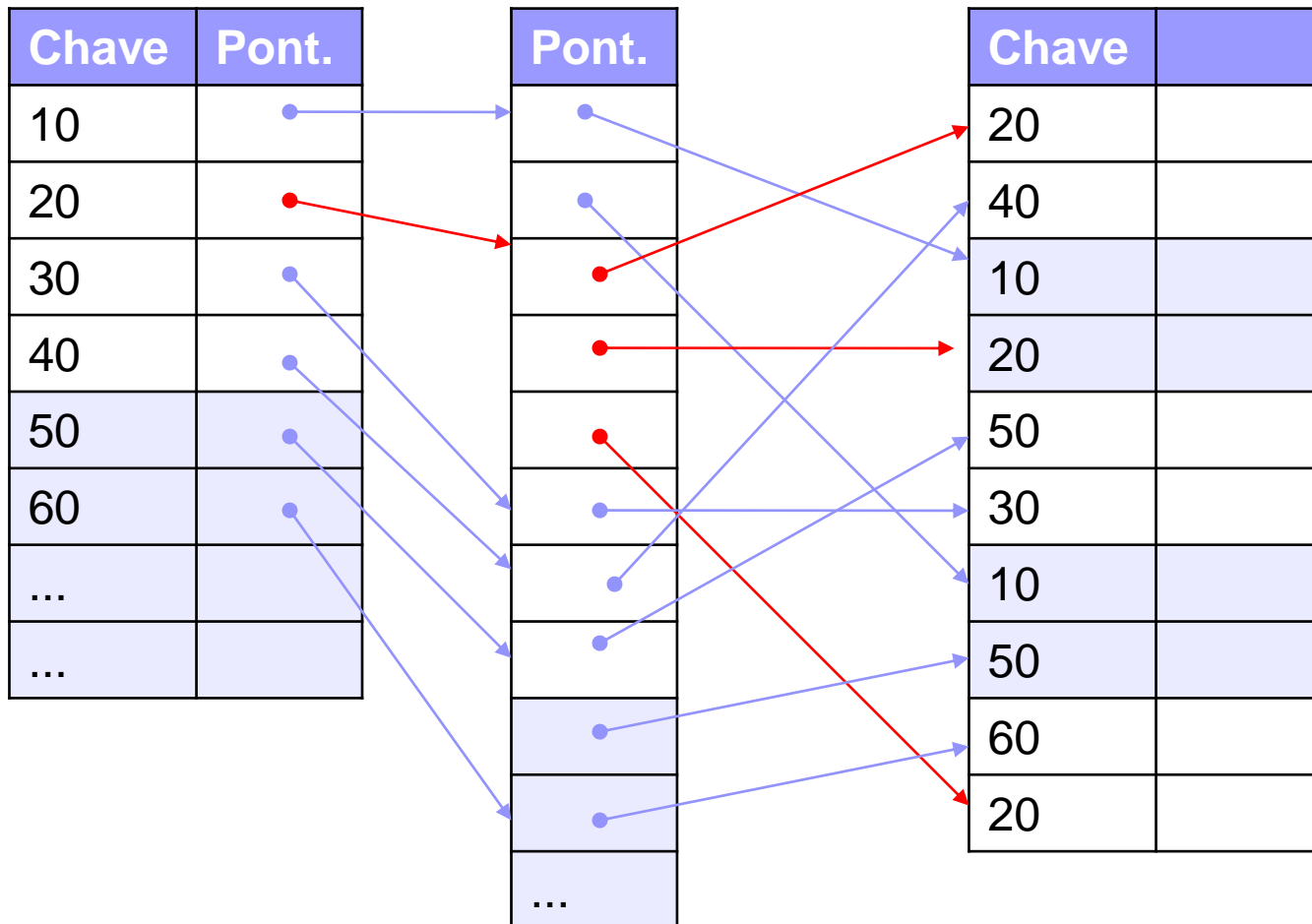
*A árvore B gerencia o espaço usado por seus blocos para que eles sempre estejam ocupados com pelo menos a metade de sua capacidade.*



# Exemplo

## E se a chave se repetir?









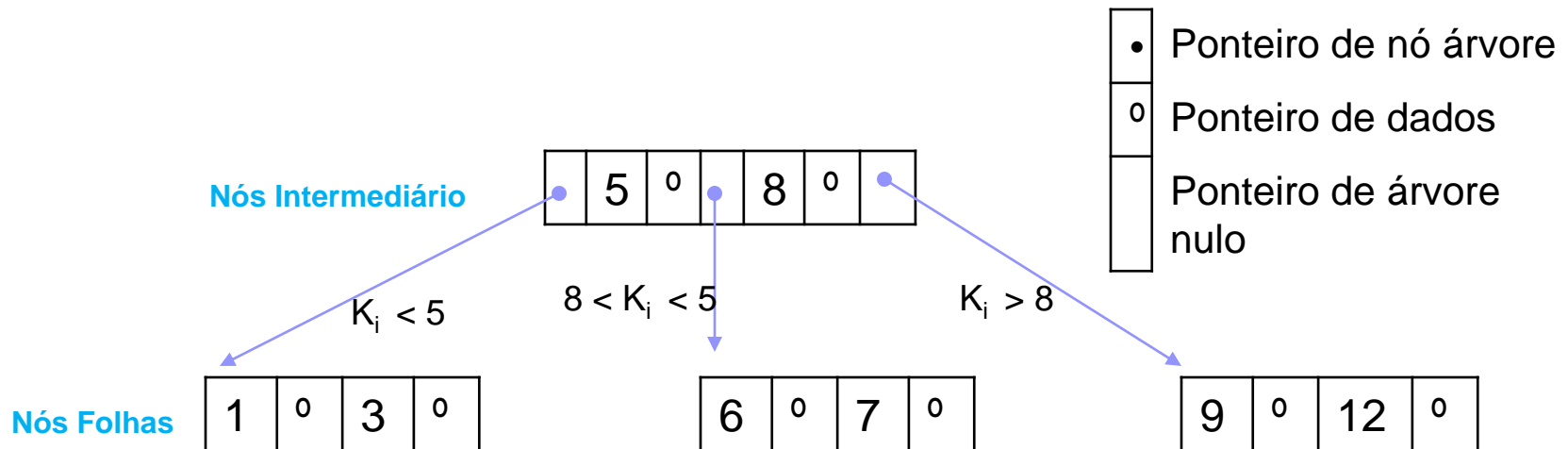
# Árvore B+

- 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.



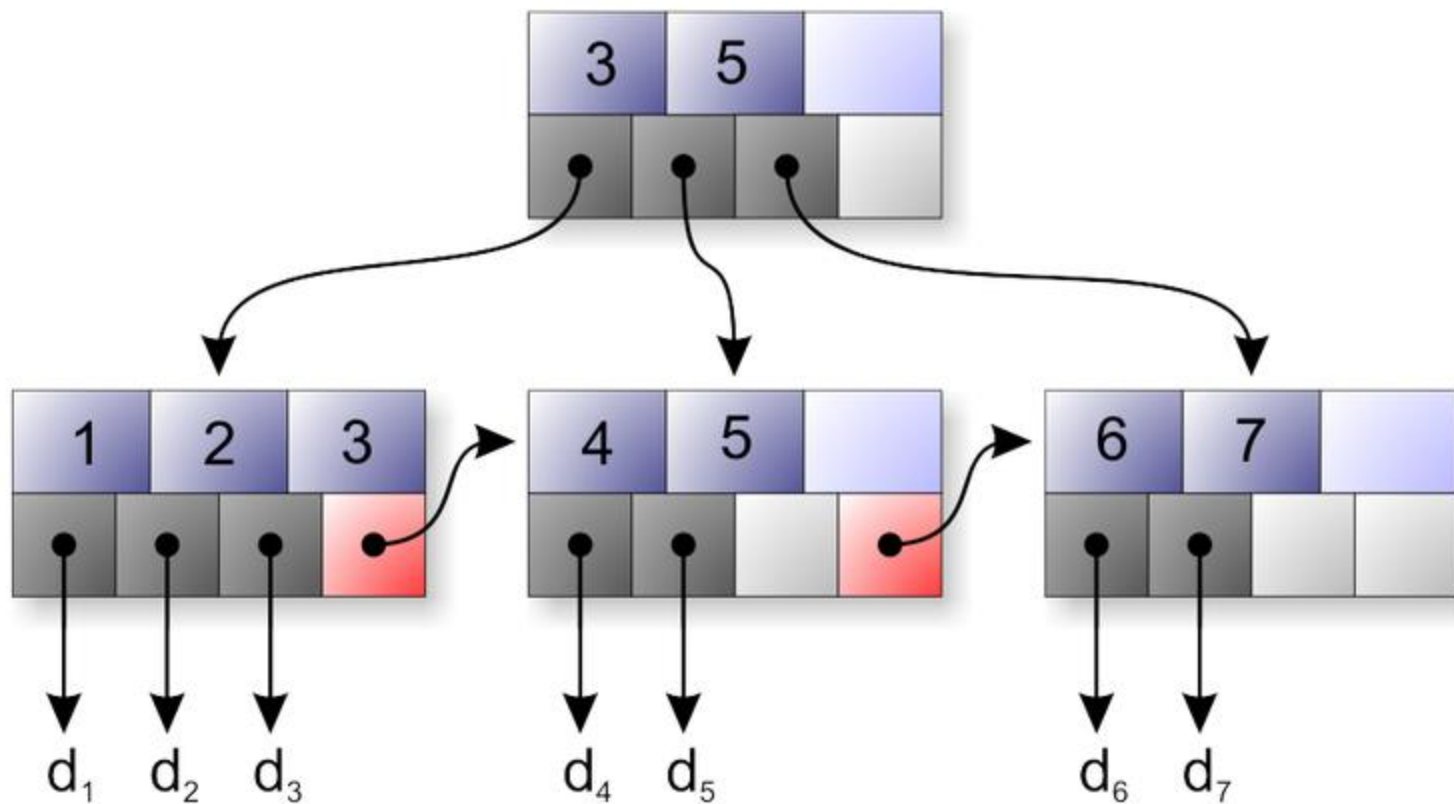
# Nó de uma árvore B

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



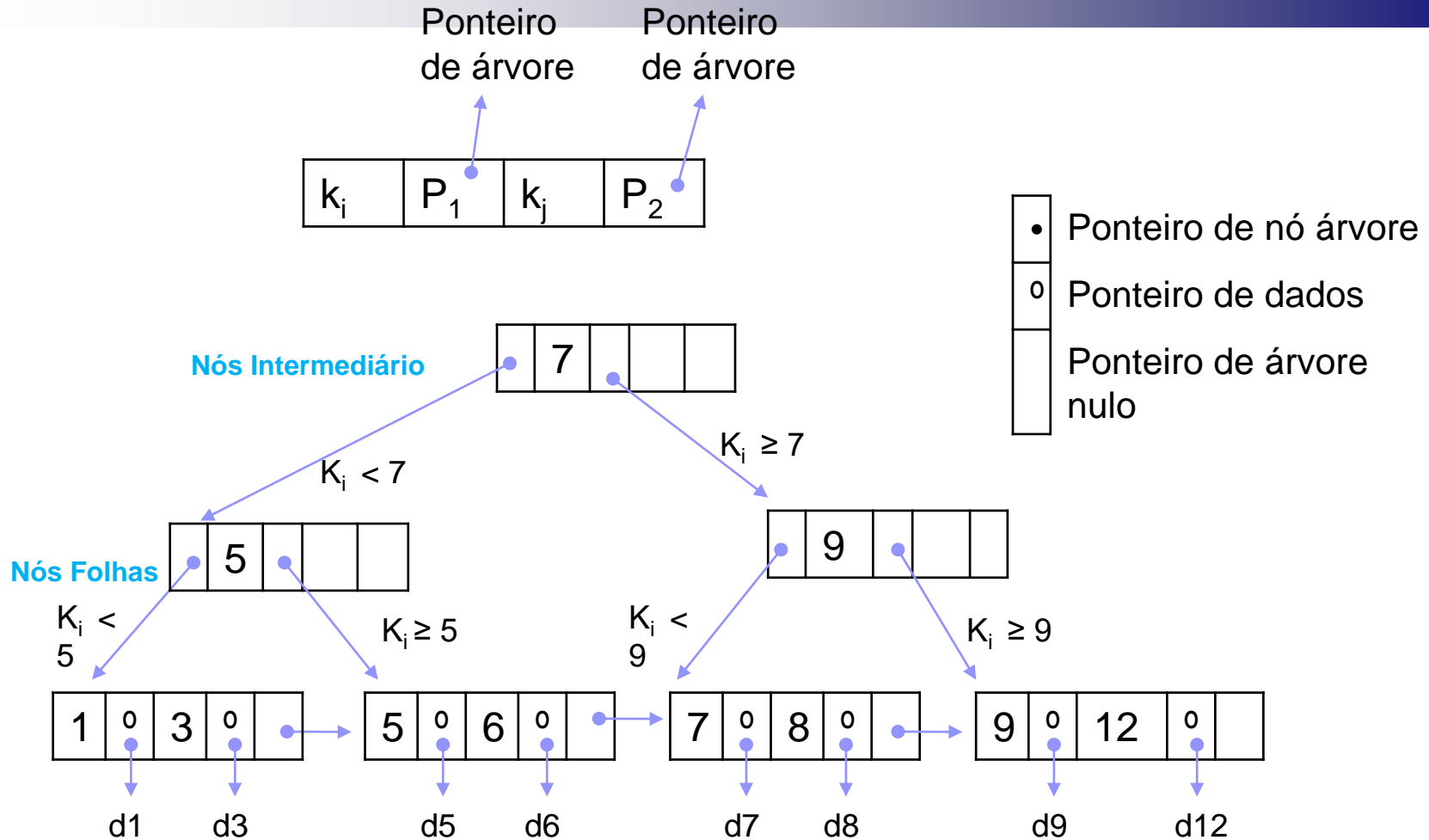


# Árvore B+





# Nó de uma árvore B+





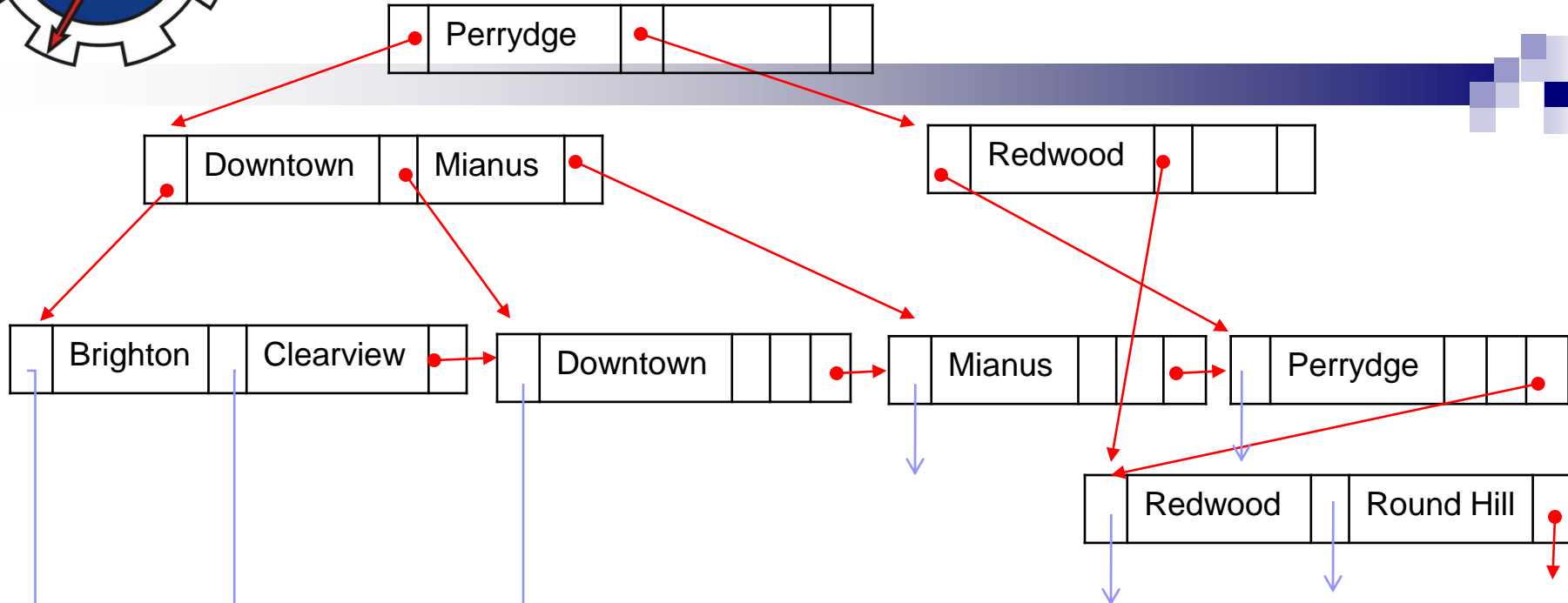
# Exemplo - Relação

## ■ 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	Williams	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.



# Para Casa



- Ler o Capítulo 11 – Indexação e Hashing do Siberschatz, Korth e Sudarshan – 6ª Edição – Sistema de Banco de Dados