

Índices

Banco de Dados II

(Capítulos 8, 10 e 11 – Ramakrishnan

Capítulo 12 – Silberschatz

Capítulo 14 – Garcia-Molina, Ullman, Widom)

Denio Duarte





Introdução


- Um banco de dados pode armazenar um grande volume de dados
 - Anualmente, mais de 35 milhões de brasileiros¹ enviam a declaração IRPF à receita
- Como encontrar um conjunto de dados específicos dentro de uma base volumosa?

¹Dados de 2023





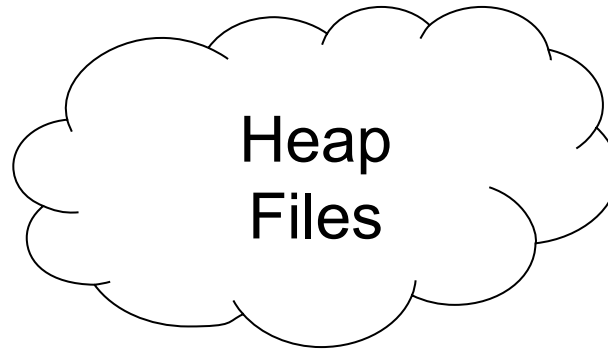
Introdução

- Aspectos que influenciam
 - Organização do arquivo de dados
 - Não ordenado (heap files)
 - Ordenado
 - Hashing
 - Estruturas auxiliares
 - Índices
 - Tamanho da linha (tupla ou registro)
 - Tamanho da tabela (relação ou arquivo)
 - Tamanho do bloco (buffer)
- 

Introdução

Dados sem organização

Paulo, 44, 2000
Pedro, 35, 20000
Carlos, 44, 2000
José, 40, 2500
João, 35, 3000
Ilmério, 40, 3500
Rodrigo, 40, 3500
Maria, 30, 4000
Sara, 35, 4000
Sabrina, 31, 5000



- Bons para inclusão
- Consultas: necessário fazer **scan** (varredura) na tabela

Introdução

Dados Organizados (key: nome)

Carlos, 44, 2000
Ilmério, 40, 3500
João, 35, 3000
José, 40, 2500
Maria, 30, 4000
Paulo, 44, 2000
Pedro, 35, 2000
Rodrigo, 40, 3500
Sabrina, 31, 5000
Sara, 35, 4000



Arquivos Ordenados

- Ruins para inclusão: necessário ter “buracos” no arquivos de dados
- Consultas: eficientes para busca da chave, para outros atributos, **scan** na tabela.

Introdução

Cluster de A à J

Dados
Clusterizados
(key: nome)

Carlos, 44, 2000
José, 40, 2500
João, 35, 3000
Ilmério, 40, 3500
Maria, 30, 4000
Pedro, 35, 2000
Paulo, 44, 2000
Rodrigo, 40, 3500
Sabrina, 31, 5000
Sara, 35, 4000

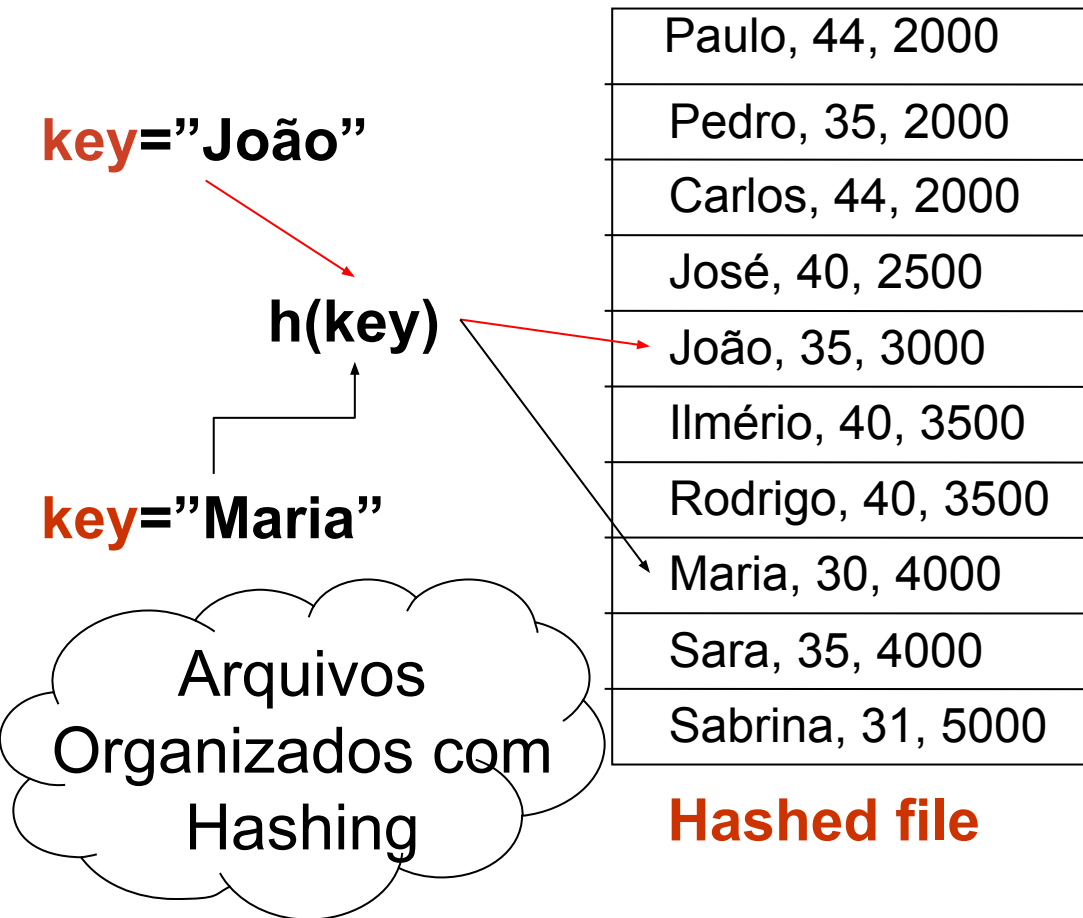
Cluster de K à Q

Cluster de R à Z

Arquivos
Ordenados em
Clusters

- Inclusão: cluster podem ter espaços vagos, mais fácil gerenciar
- Consultas: eficientes para busca da chave (faz **scan** no cluster), para outros atributos, **scan** na tabela (nos clusters).

Introdução



- Inclusão: direta, apenas um acesso. Porém deve haver espaços pré-alocados para novas tuplas
- Consultas: eficientes para busca da chave porém apenas para igualdade. Outras formas, **scan** na tabela.



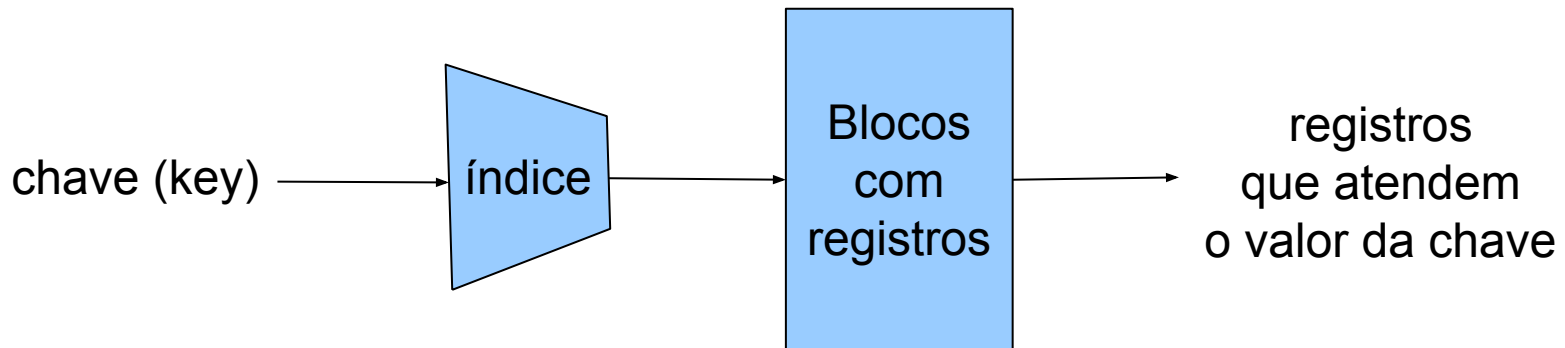
Introdução

- Conclusão organização
 - Uma tabela é mais acessada para consulta
 - Estratégias para melhorar a consulta
 - Inclusão, exclusão e alteração ficam em segundo plano
 - Não devem ser negligenciadas
 - Utilizar uma estrutura auxiliar para consultas (índice)



Introdução

- Índices
 - estrutura auxiliar projetada para agilizar operações de busca, inserção e supressão



- Alteração nos dados pode levar na alteração no índice
- Espaço extra de armazenamento

Índices

- Criação
 - Escolher o(s) atributo(s) que compõe (oram) o índice (a chave)
 - Significados para chave: primária, ordenação ou pesquisa
 - Novo arquivo é criado apenas com a chave e a localização da tupla <key, local>
 - *local* pode ser a localização exata da tupla ou pode ser o cluster ou o bloco
 - Os índices podem ser densos, esparsos (agrupados ou árvores)

Índices

- Criação

- PostgreSQL

```
create unique index <name> on <table> using  
<method> (<attributes>) include (<attributes>)
```

```
create index cust_dtnasc_idx on customer (dtnasc)
```

```
create unique index cust_email_idx on customer  
(email)
```

```
create index cust_state_city_idx on customer  
(state,city)
```

Índices

- Criação

- PostgreSQL


```
create unique index <name> on <table> using  
<method> (<attributes>) include (<attributes>)
```

```
create index cust_dtnasc_inName_idx on customer  
(dtnasc) include (name)
```

```
create index cust_ssn_idx on customer using hash  
(ssn)
```



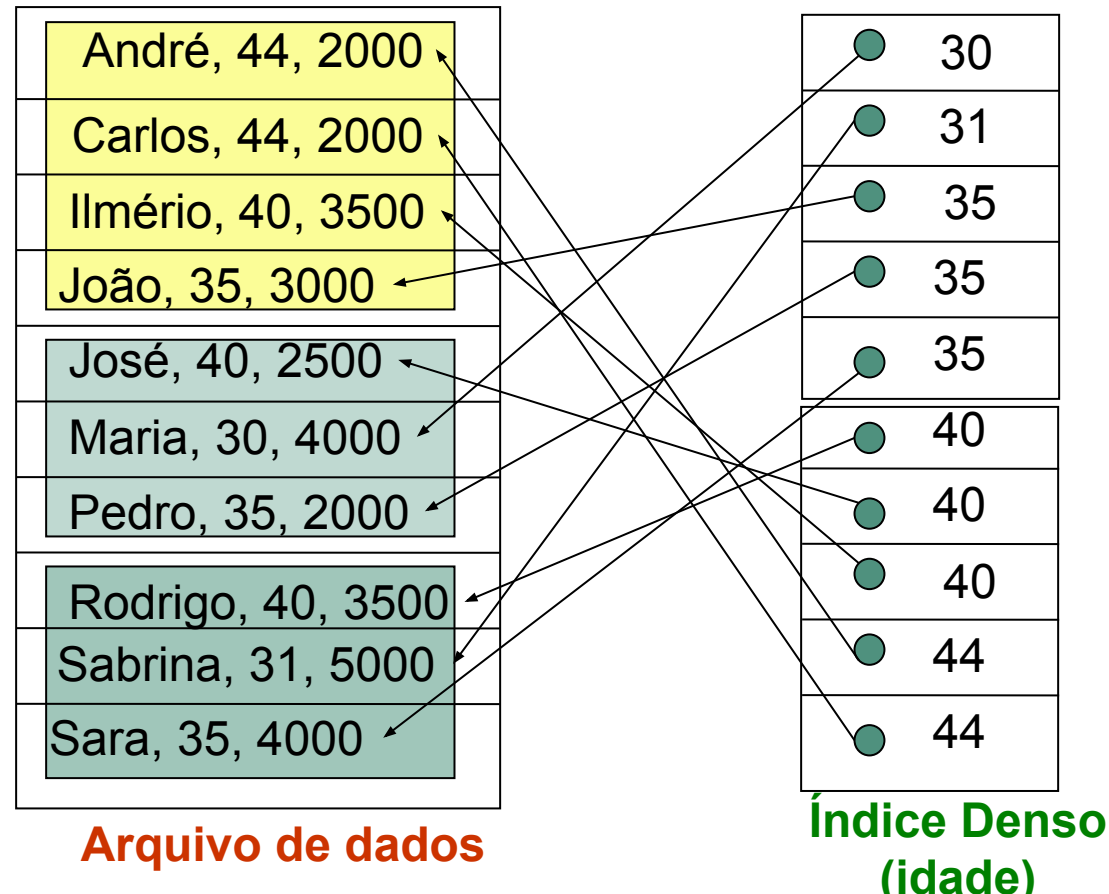
Índices

- O que armazenar em cada entrada do arquivo de índice
 - Entrada = Registro inteiro
 - Entrada = chave, rid
 - Entrada = chave, conjunto de rids
 - Organização das entradas
 - Ordenado
 - Hashing
- 

Índices

- Índice denso

- Contém entradas com todas as chaves do arquivo de dados





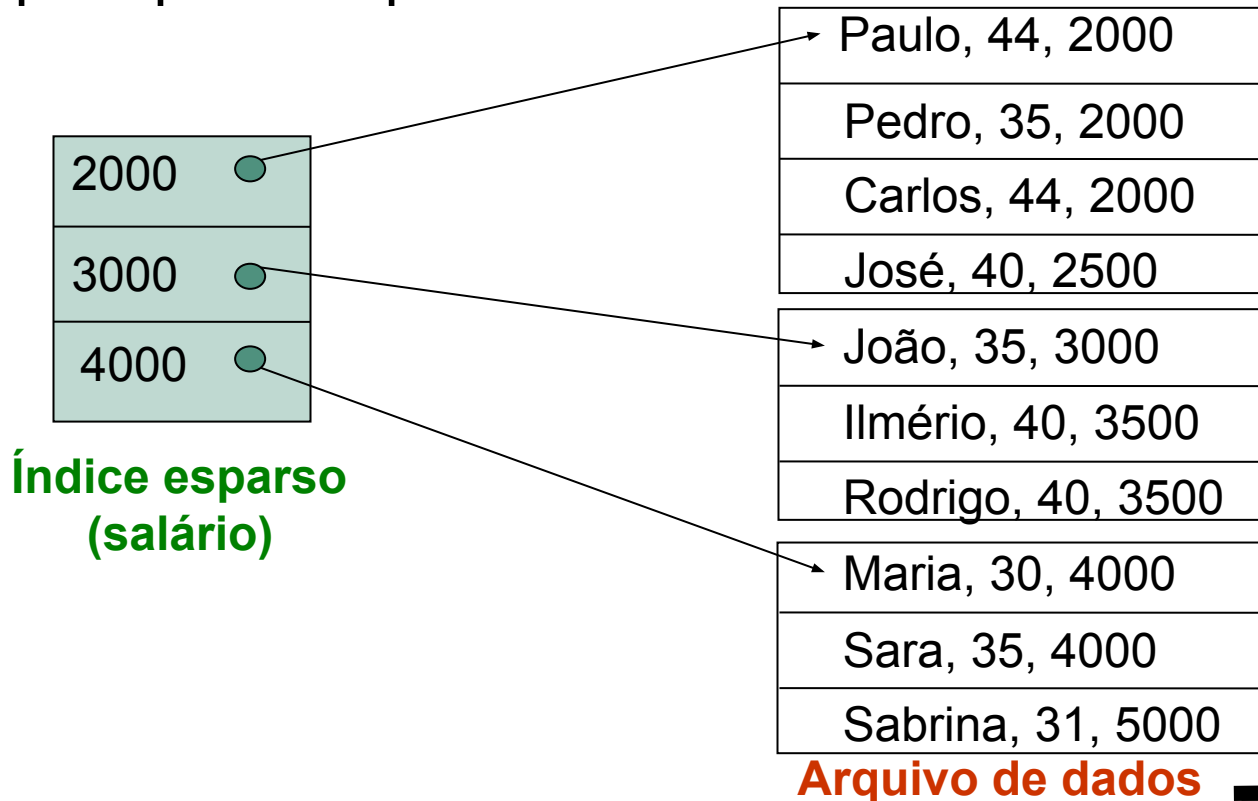
Índices

- Índice denso
 - Vantagens além de otimizar o desempenho da consulta
 - O número de blocos para armazenar o índice é, geralmente, menor que para armazenar os dados
 - Pode-se utilizar a busca binária para buscar um registro
 - O índice pode caber na memória principal (buffer), diminuindo o número de I/O's em uma busca



Índices

- Índice esparsos
 - Contém apenas algumas das chaves arquivo de dados que apontam para blocos com os valores

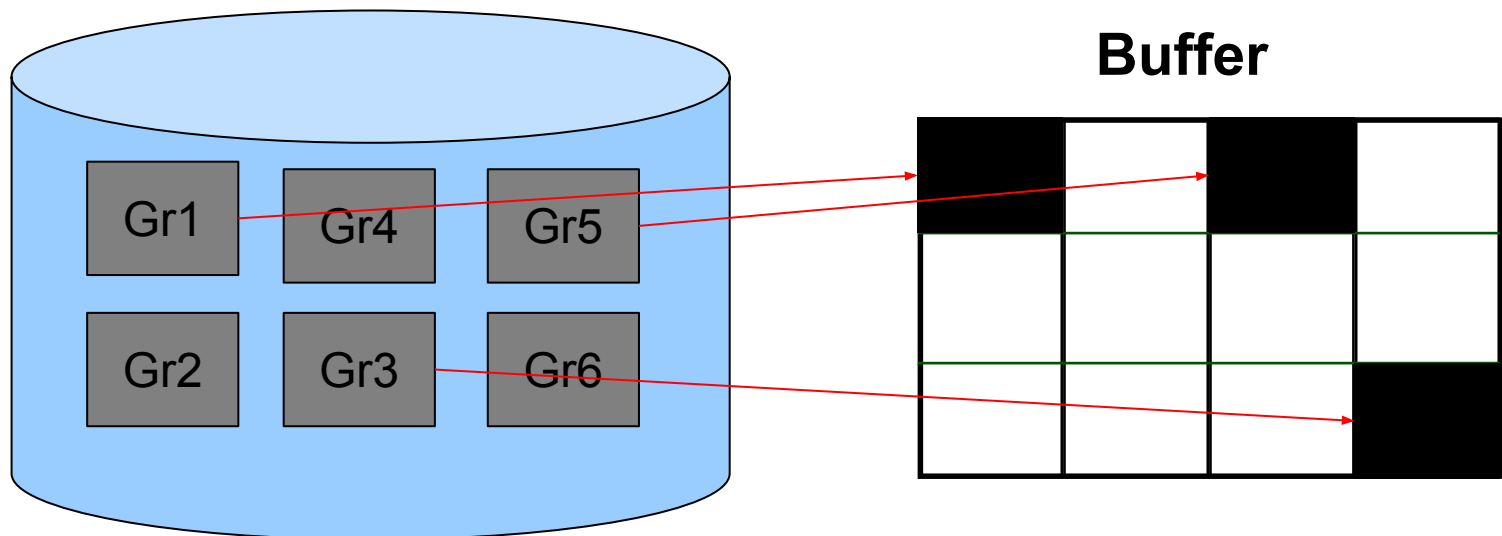


Índices

- Índice esparsos
 - Índices esparsos são menores que os densos (portanto, maior chance de caber no buffer)
 - O arquivo de dados deve estar ordenado pela chave de busca
 - É feito um *scan* no bloco da provável ocorrência dos registros com os valores
 - A busca é feita no índice esparsos *valor \geq key*


Índices

- Estratégias de agrupando podem ser interessantes
 - Um grupo pode ser do tamanho de um bloco
 - Um grupo pode caber no buffer






Índices

- Primário/Único
 - A chave do índice é composta por uma chave (primária ou não) da tabela
 - A maioria dos SGBD cria índices para chave primária automaticamente
 - Não permitem duplicatas
 - Podem ser agrupados
 - Secundário
 - Outras colunas da tabela participam
 - Permitem duplicatas
- 



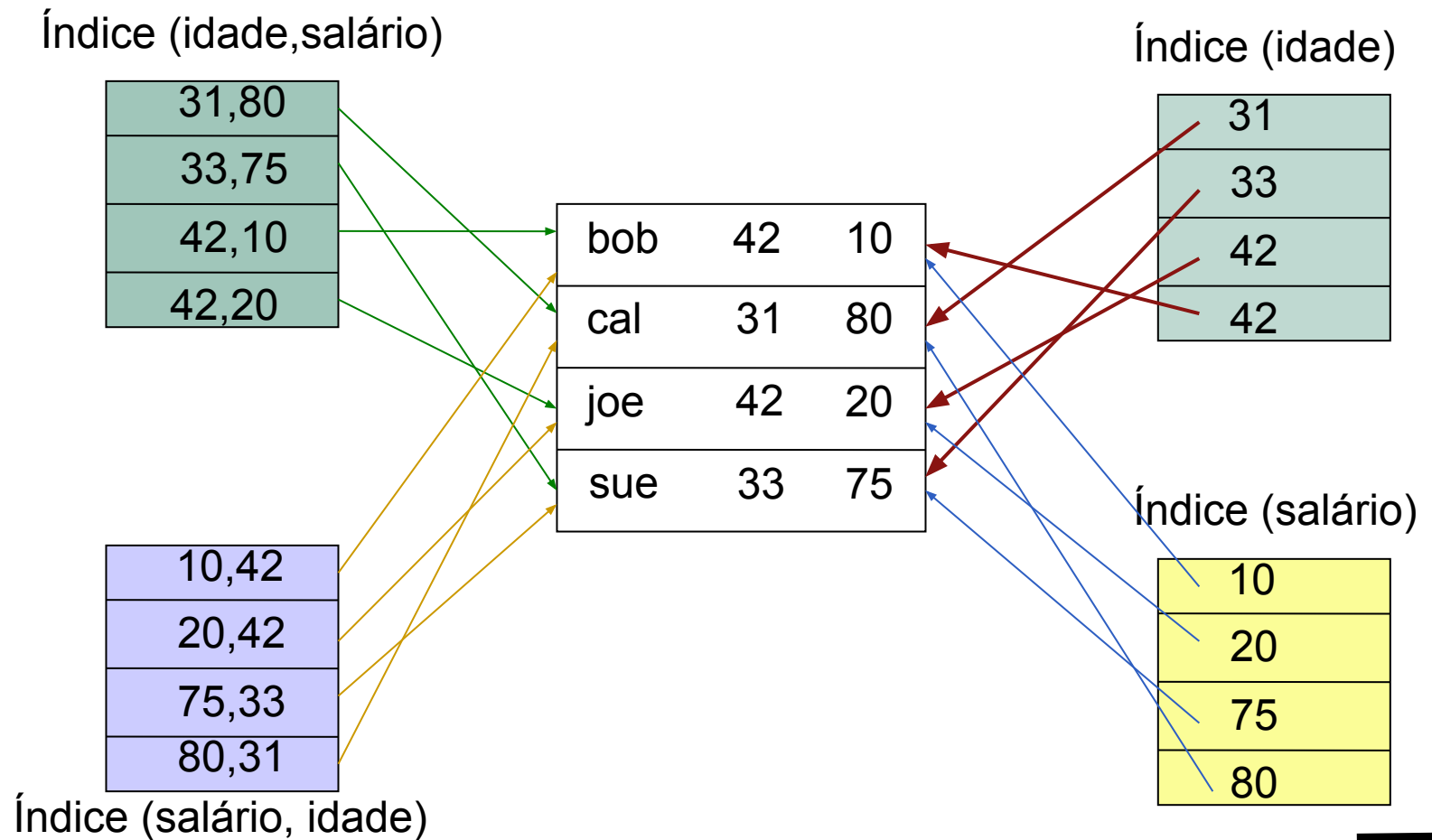
Índices

- Compostos
 - Mais de uma coluna compõem a chave
 - Podem ser primários ou secundários
 - Índices com conteúdo
 - Permite colocar valores mais acessados juntos com o(s) atributos que compõe(m) o índice
- 

Índices

- Compostos

Simple





Índices

- Níveis simples
 - Apenas uma camada para acesso ao arquivo de dados
 - Índices densos
 - Índices agrupados (esparsos) – alguns casos
- Múltiplos níveis
 - Várias camadas de índices até o arquivo de dados
 - Árvores (veremos mais adiante)





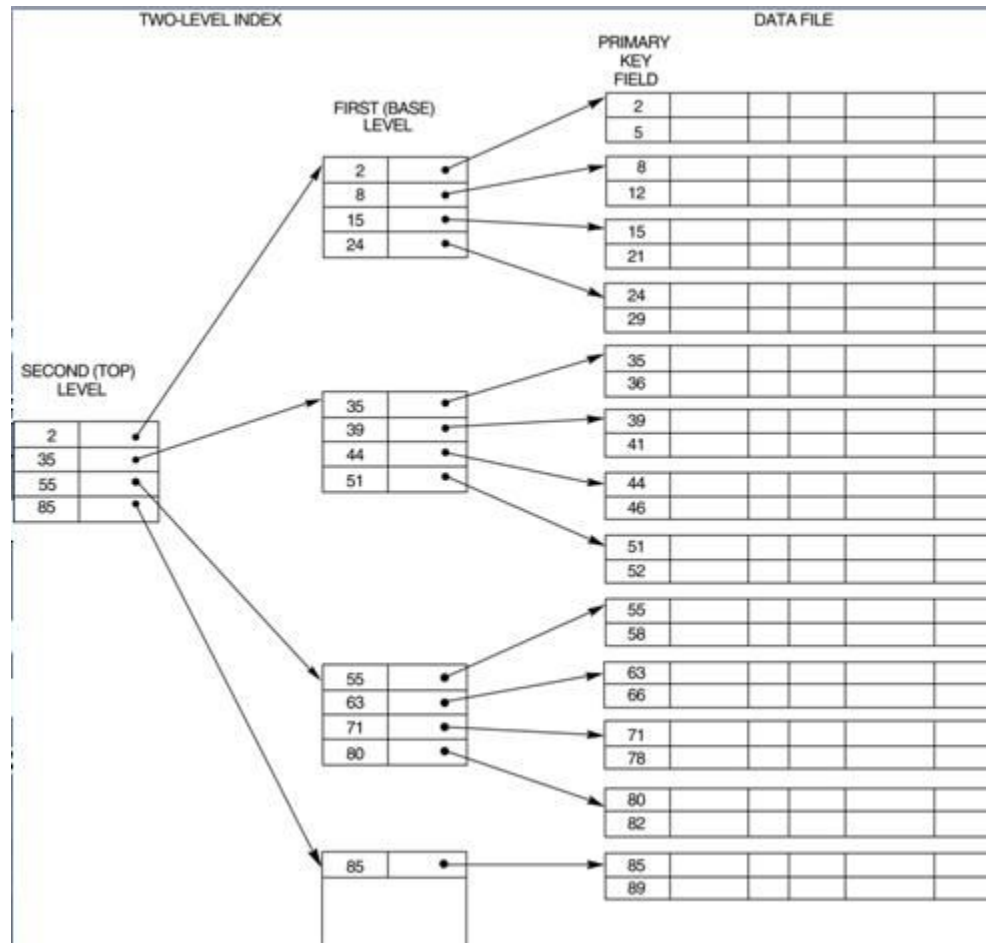
Índices

- Índices de múltiplos níveis
 - Apontar para índices densos
 - Arquivos ordenados
 - Otimizar o acesso
 - Outros níveis acima de níveis anteriores
 - Hierarquização do acesso



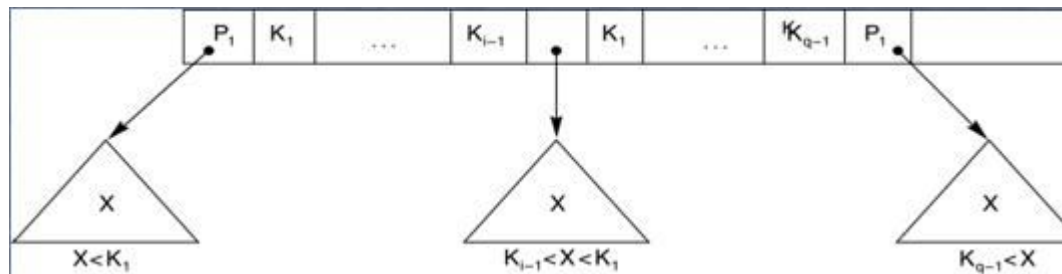
Índices

- ISAM - Indexed Sequential Access Method (IBM)



Índices

- Os índices de múltiplos níveis podem formar uma árvore



- Atualização nos dados, implica na atualização em todos os níveis



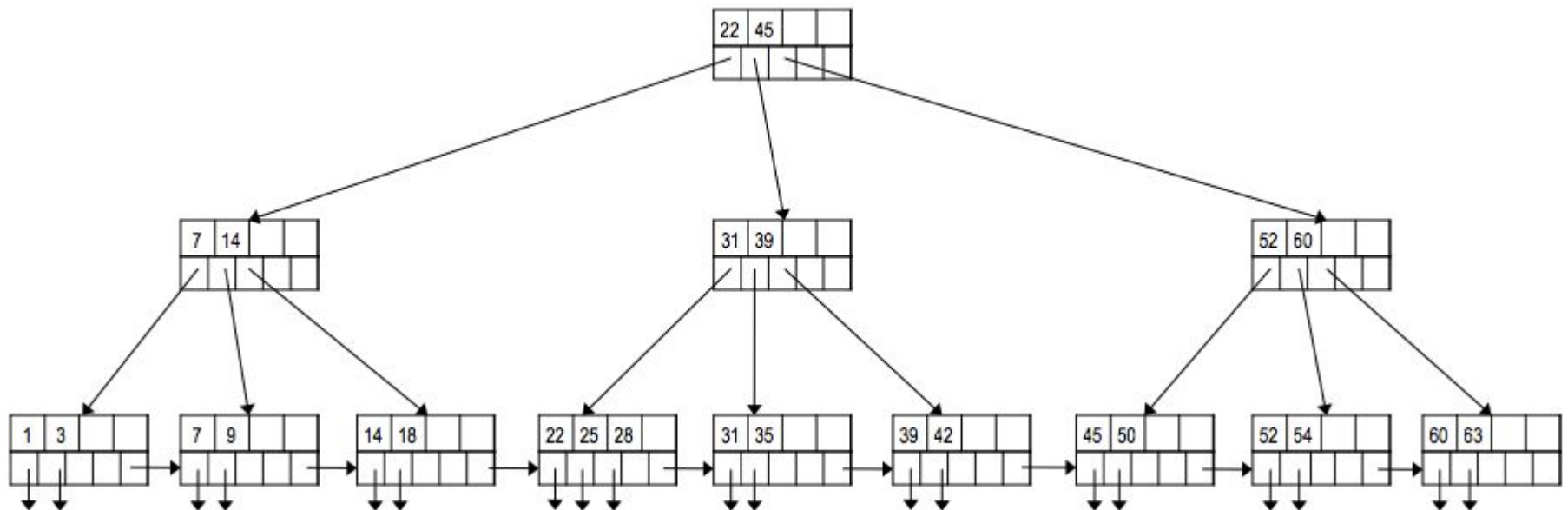
Índices

- Os SGBDs implementam índices multi-níveis através de árvores B+
 - Atualização dos níveis mais eficientes
 - Cada nível elimina vários acessos
 - O grau da árvore indica o número de acessos



Índices

- Árvores B+



Índices

- Árvores B+
 - Qual a altura máxima de uma árvore B com m chaves?
 - Esta questão é importante pois a altura da árvore dará o limite máximo de acesso ao disco
 - Sendo N o maior número de chaves na árvore, N' o menor e m o número máximo de chaves em um nó
 - Assim, a menor altura $h = \log_{\frac{m}{2}}(\frac{N' + 1}{2})$
 - A maior $h = \log_{\frac{m}{2}}(\frac{N + 1}{2})$

Índices

- Dada um árvores B+ de ordem b , os tempos serão:
 - Inserção $h = \log_b(n)$
 - Busca $h = \log_b(n)$
 - Onde n é o número de chaves



Índices

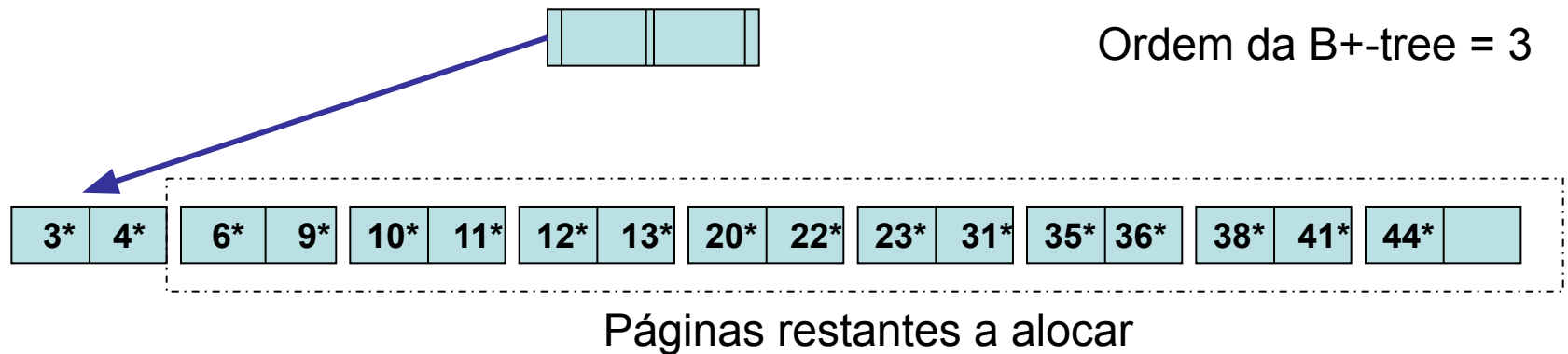
- Let's check it out:
 - Animação ([here](#))
 - Or type `http://www.cs.usfca.edu/~galles/visualization/BPlusTree.html`



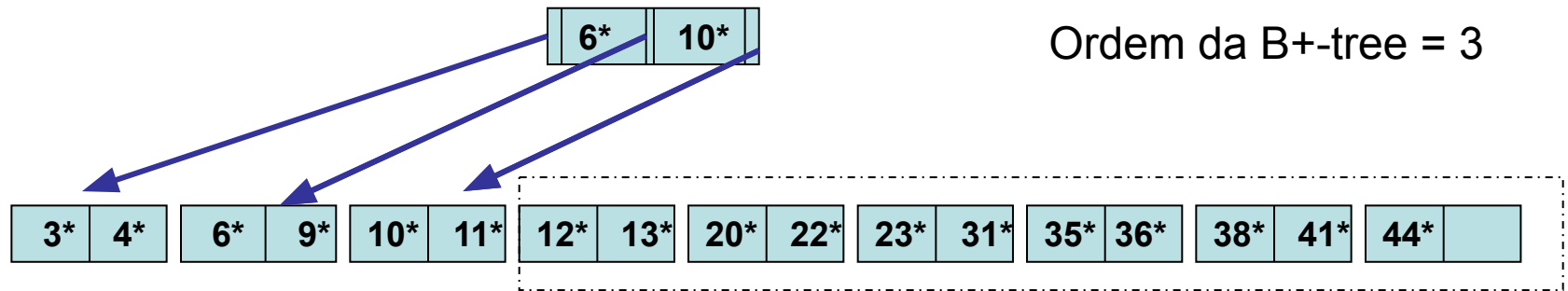
Bulking Load (Construção da B+ Tree)

- Utilizando o arquivo de índice denso (ou arquivo ordenado)
- Aloca-se uma página vazia para a raiz
- Insere nesta página um ponteiro para a primeira página do arquivo contendo as entradas.

Bulking Load (Construção da B+ Tree)

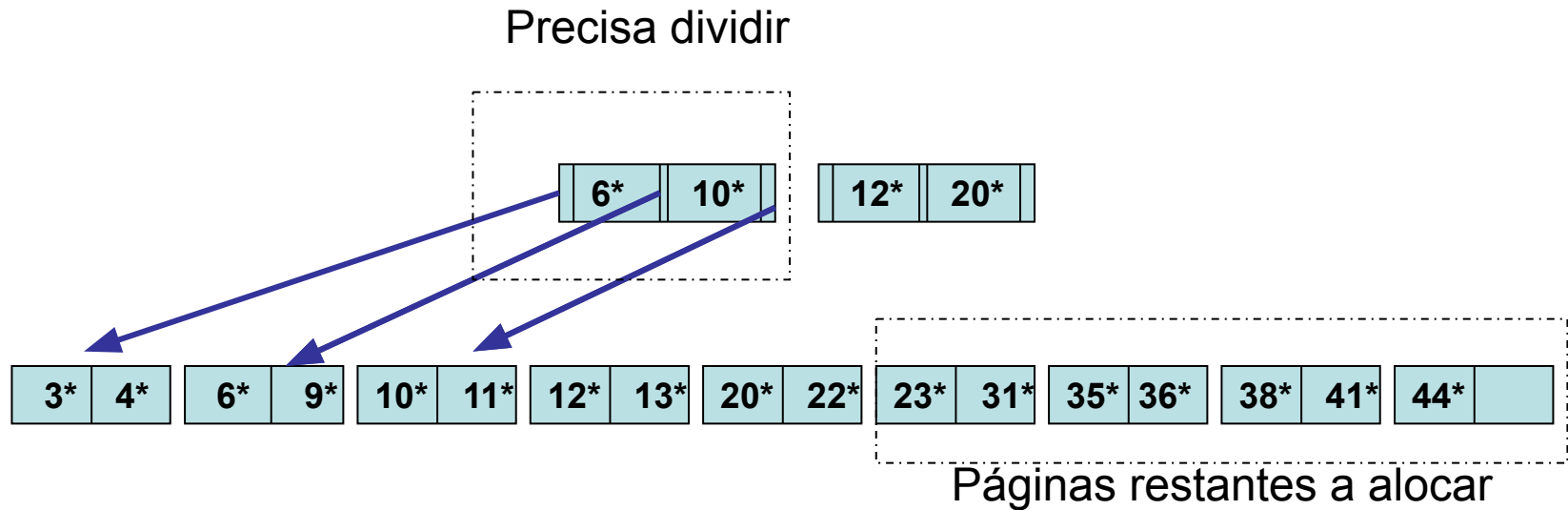


Bulking Load (Construção da B+ Tree)

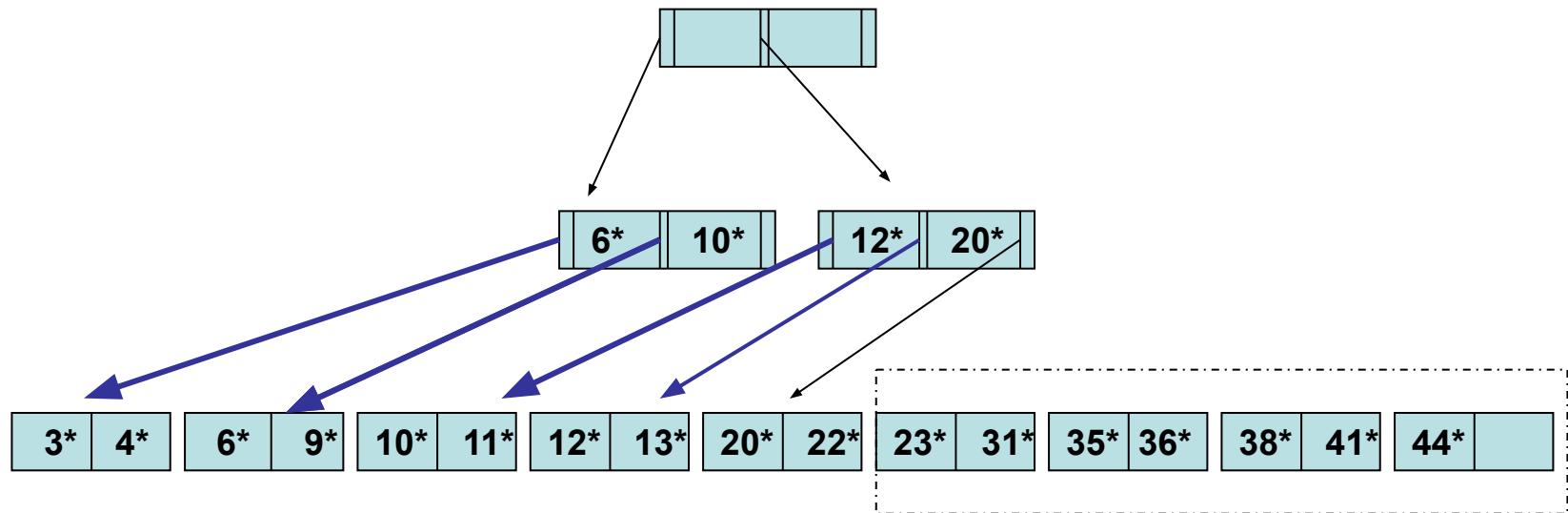


Páginas restantes a alocar

Bulking Load (Construção da B+ Tree)

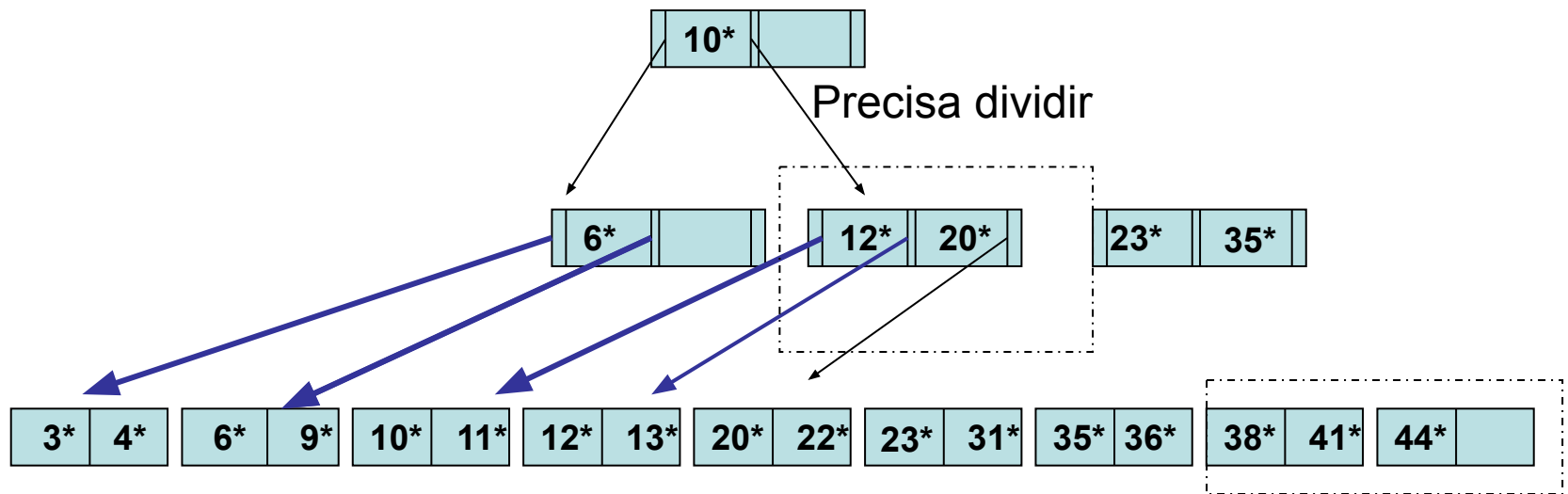


Bulking Load (Construção da B+ Tree)

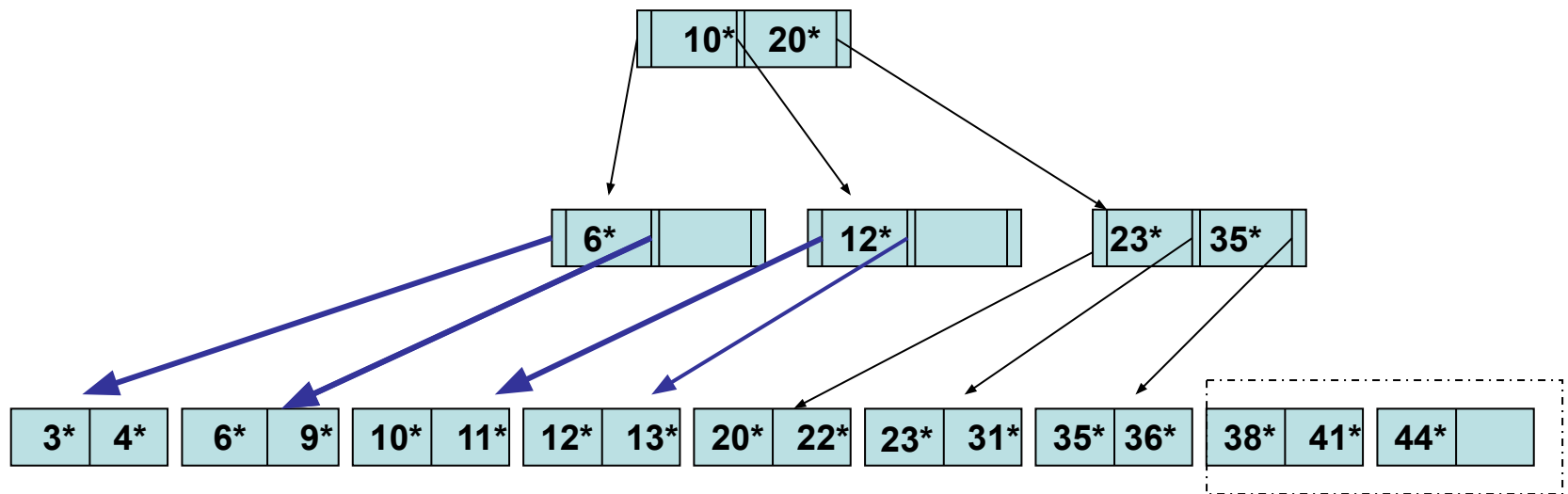


Páginas restantes a alocar

Bulking Load (Construção da B+ Tree)

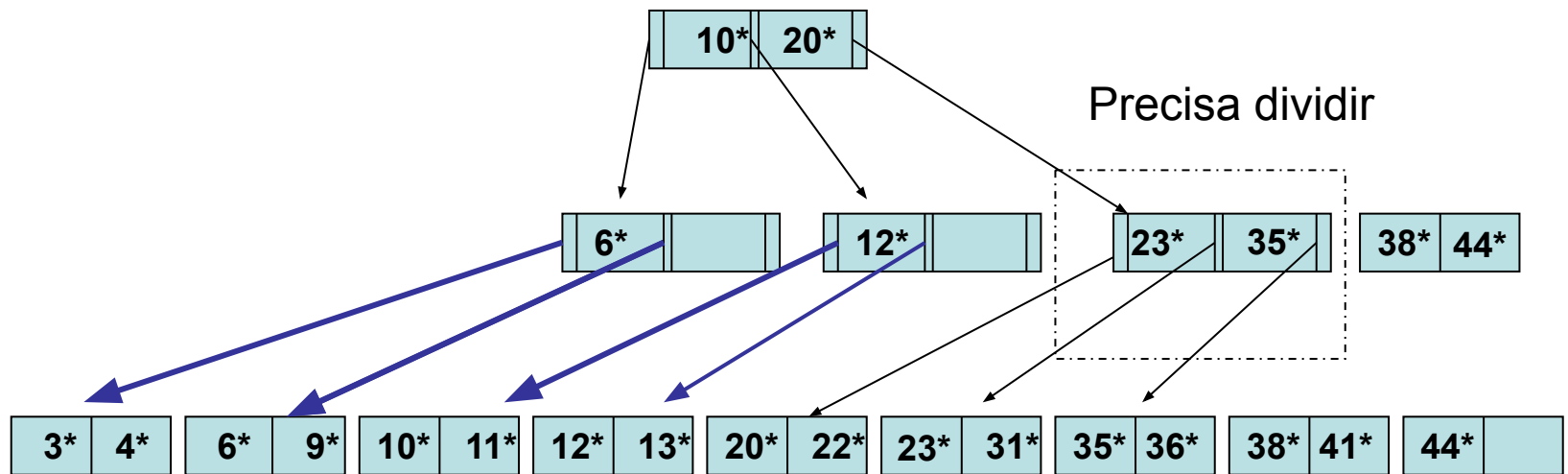


Bulking Load (Construção da B+ Tree)



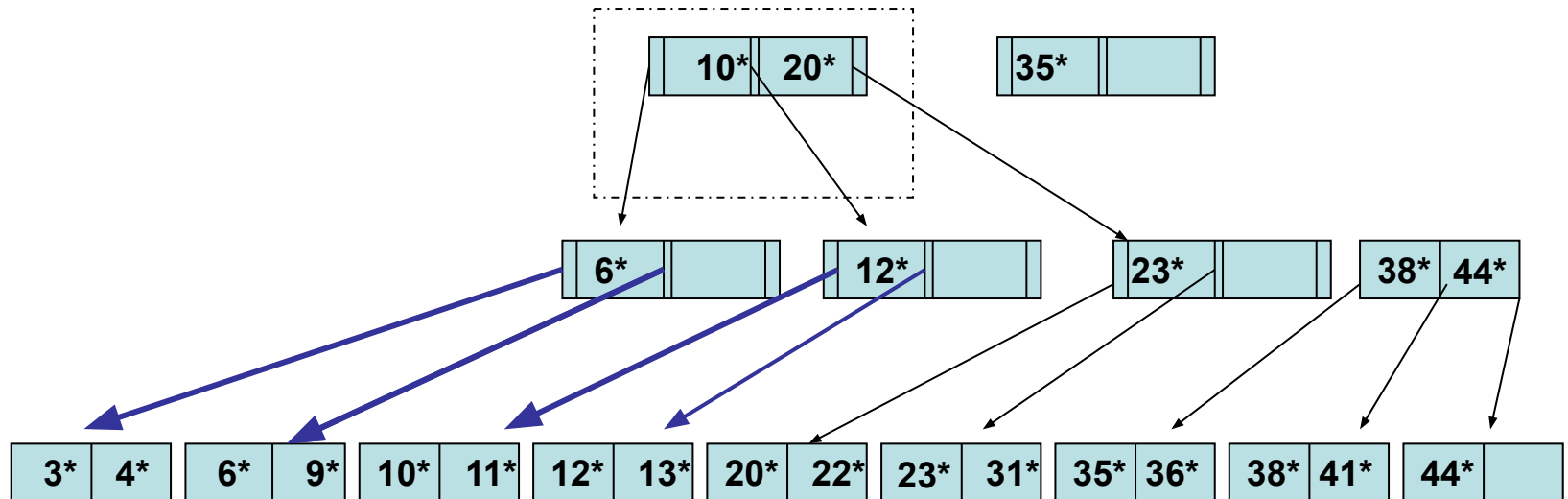
Páginas restantes a alocar

Bulking Load (Construção da B+ Tree)

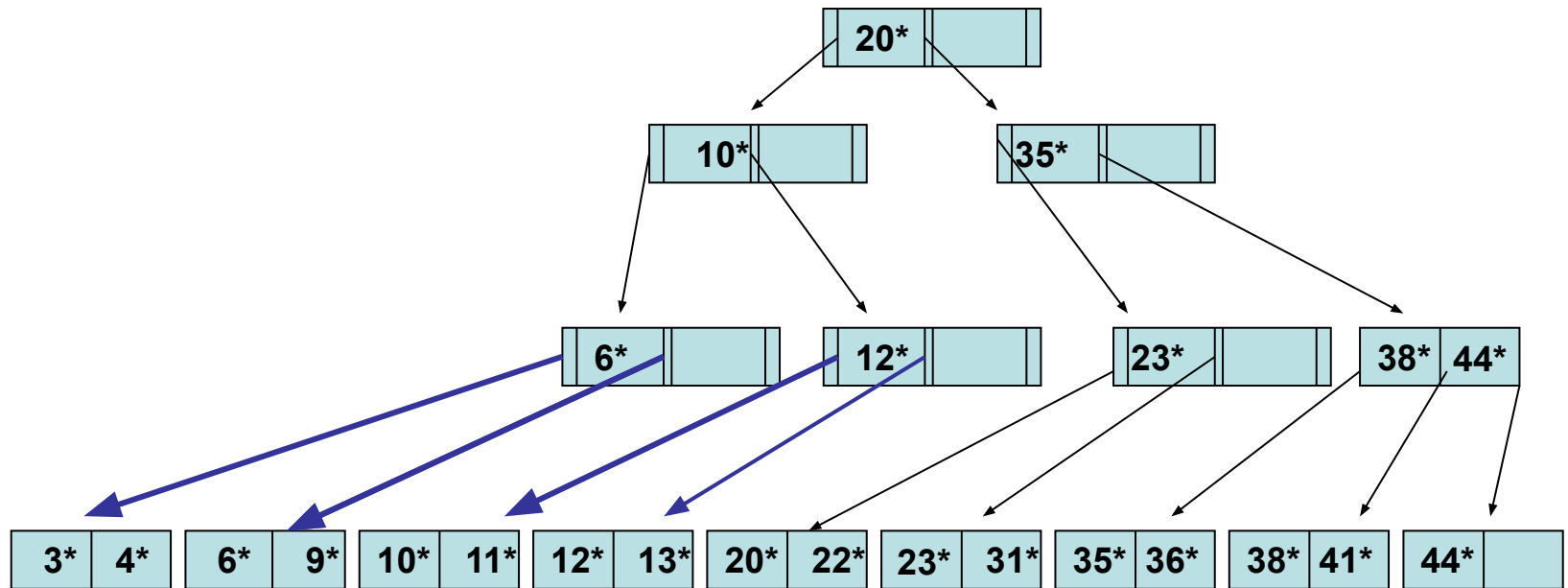


Bulking Load (Construção da B+ Tree)

Precisa dividir



Bulking Load (Construção da B+ Tree)



Árvore construída!



Índices

- Conclusões
 - Os dados são mais acessados que atualizados
 - Necessário existir uma estrutura auxiliar para melhorar o desempenho das consultas
 - Para dados relacionais árvores B+ são os índices mais utilizados
 - O Otimizador de consultas utiliza índices sempre que possível
- 