Banco de Dados II

Índices

Profa.: Márcia Sampaio Lima

EST - UEA

Índices

- Quando se trabalha com bancos de dados, diariamente é preciso fazer consultas a tabelas com grandes quantidades de registros.
  - Levam tempo para serem lidas e retornado o resultado.
- Independente da plataforma (desktop, web, mobile), o desempenho das aplicações é um fator fundamental e determinante,

Índices

- Há um esforço para que o tempo de resposta, quando consultas a bancos de dados são feitas, seja o menor possível.
- Utiliza-se índices, principalmente em campos numéricos de suas tabelas.
- Essa medida ajuda o gerenciador do banco de dados a localizar os registros com mais facilidade.

Índices

- Entendendo .....
  - A execução de um SELECT sobre um tabela gera uma ação feita pelo banco chamada TABLE SCAN.
  - Essa ação consiste em percorrer toda a tabela, avaliando cada registro. Caso o registro atenda às condições definidas no filtro, ele é incluído no conjunto de retorno, senão, é desconsiderado.

Índices

- Entendendo...
  - SELECT * FROM tabela WHERE “Codigo”=3.

  - A tabela contem 3 registros, porém, se a tabela tivesse milhares de linhas???

Índices

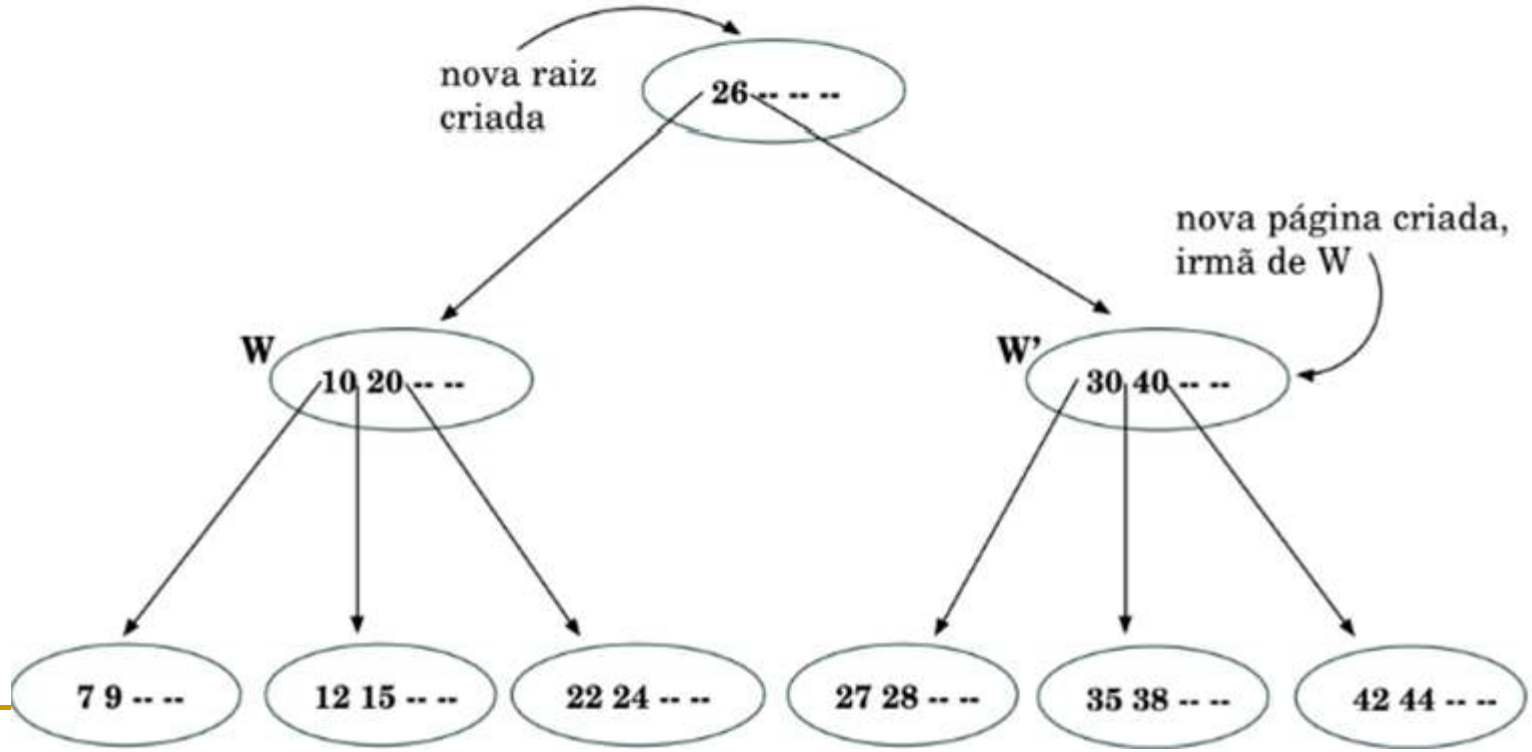
- Para melhorar o desempenho das consultas, utilizamos ÍNDICES.
  - São objetos do banco de dados que facilitam a organização e consulta de uma tabela, “indexando-a” por uma de suas colunas.
- Quando criamos um índice em uma coluna, o gerenciador do banco ordena a tabela por essa coluna e a partir de então os filtros (sobre essa coluna) são feitos através de uma busca mais eficiente.

Índices

- Índices são utilizados nos seguintes modos:
  - Para encontrar rapidamente os registros que coincidam com uma cláusula **WHERE**.
  - Para recuperar registros de outras tabelas ao realizar joins.

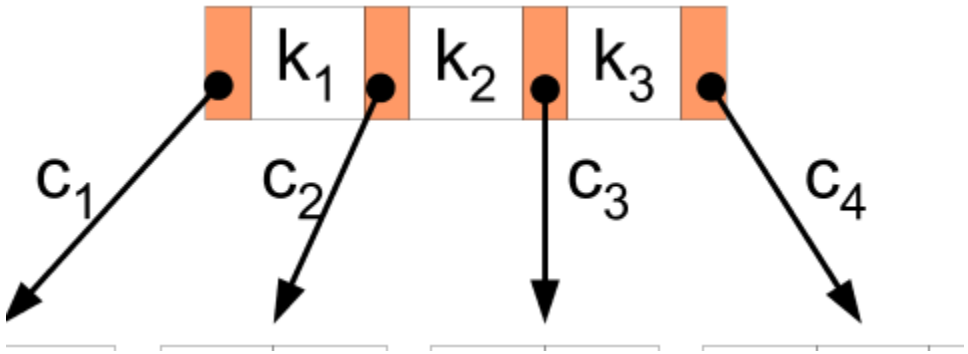
Índices

- Estrutura usada para criação de índices:
  - B-tree



Árvores B

- Nomenclatura:
  - Nós de uma árvore B são ***páginas***
  - Cada página armazena várias ***chaves***



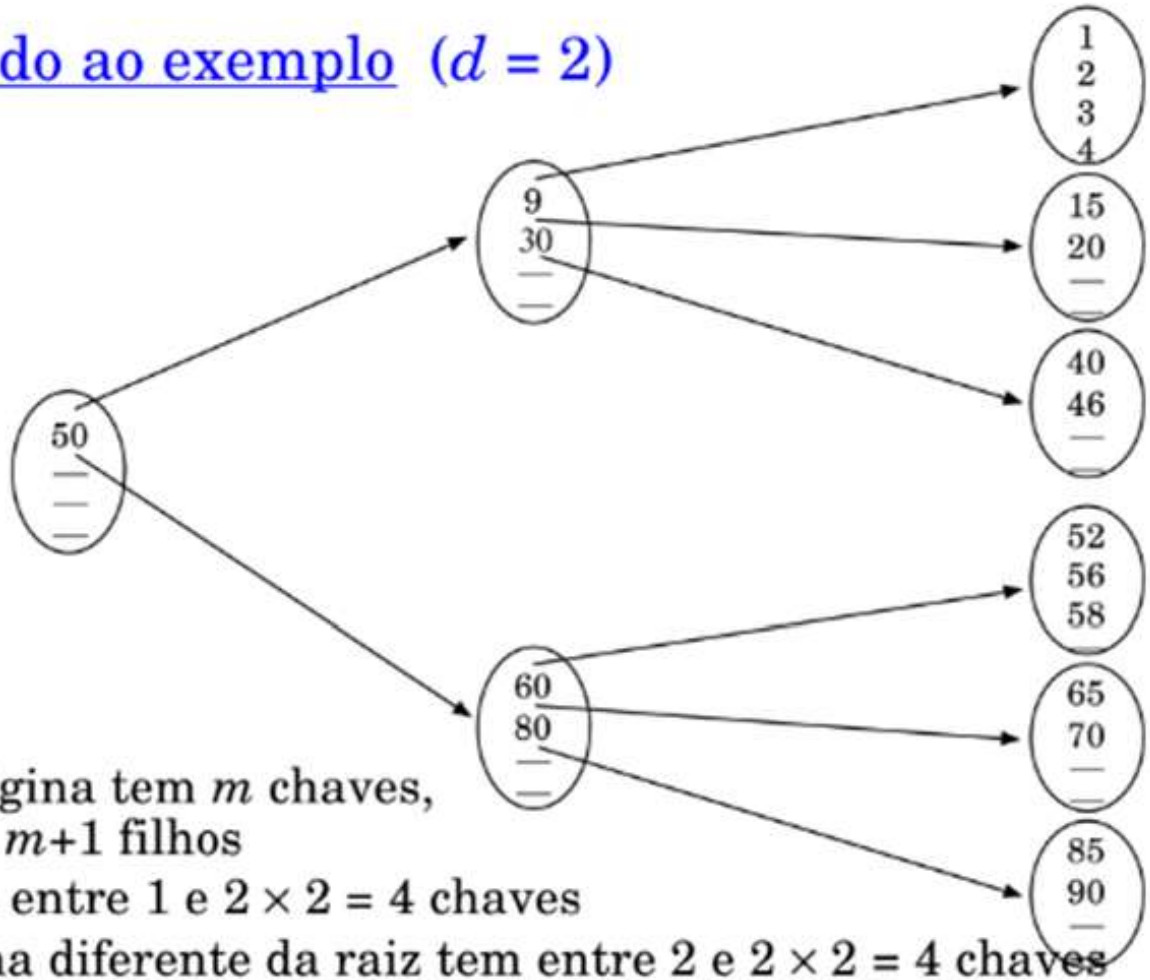
Árvore B

- Propriedades:
  - Se uma página **P** não folha possui **m** chaves, então **P** possui ***m + 1*** filhos.
  - A raiz possui entre **1** e **2d** chaves. Sendo d a ordem da árvore a ser definida.
  - Cada página ≠ da raiz possui entre **d** e **2d** chaves
  - Em cada página **P** com **m** chaves, as chaves estão ordenadas:

$$s_1 < s_2 < .... < s_m$$

  - **P** contém **m + 1** ponteiros (***p_0, p_1, ..., p_m***) apontando para seus filhos.

Voltando ao exemplo ($d = 2$)



Observe:

a) se uma página tem m chaves, então tem $m+1$ filhos

b) a raiz tem entre 1 e $2 \times 2 = 4$ chaves

c) cada página diferente da raiz tem entre 2 e $2 \times 2 = 4$ chaves

d) em cada página as chaves estão ordenadas

e) de cada página com m chaves partem $m+1$ ponteiros (as folhas têm ponteiros nulos)

Observe a estrutura das entradas, de acordo com a tela anterior.

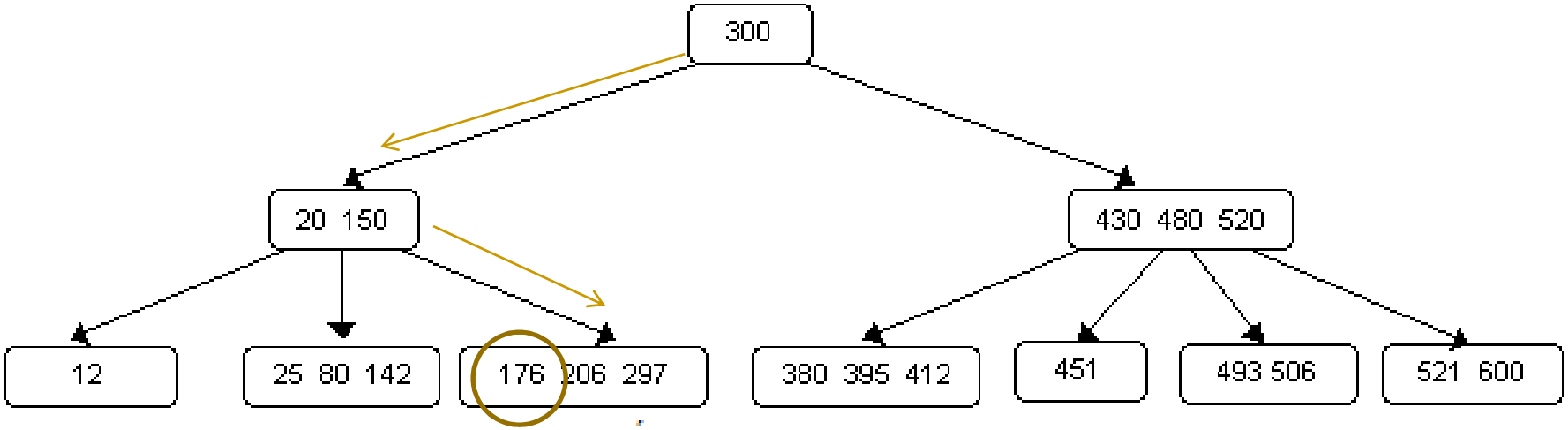
Árvore B -Busca

- Seja **s** a chave procurada:

1. Procura na lista ordenada de chaves da página raiz.
2. Se s for encontrada, interrompa.
3. Senão: busca na página filha seguindo o ponteiro adequado.
   - Se $s < s_1$, usar p_0
   - Se $s_i < s < s_{i+1}$, usar p_i $(1 \leq i \leq m\text{-}1)$
   - Se $s > s_m$, usar p_m
4. Repete a busca até que:
   - ou s seja encontrada
   - ou atinja-se um ponteiro nulo

Árvore B

- Buscar a chave 176 :



Índices

- Criação de Índices:
  - No momento da concepção da tabela
  - Em uma tabela já existente.

- Tipos de Índices:
  - (**PRIMARY**, **UNIQUE** e **INDEX**)

- Todos são armazenados em árvores B.

Índices

- *INDEX:*
  - *Índice não único é um no qual qualquer valor da chave pode ocorrer múltiplas vezes. Este tipo de índice é definido com a palavra chave INDEX.*
- *UNIQUE :*
  - *possui valor único, ou seja, cada valor da chave deve ser diferente de todos os outros (a exceção é que valores NULL podem ocorrer múltiplas vezes);*

Índices

- *PRIMARY KEY*
  - *também é um índice de valores únicos. Ela é semelhante a um índice UNIQUE, mas não aceita valores NULL;*

- *Uma tabela pode conter múltiplos índices UNIQUE, mas no máximo uma PRIMARY KEY;*

Índices

- Exemplo:

```
CREATE TABLE `USUARIO` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `NOME` varchar(45) NOT NULL,
  `USER` varchar(20) NOT NULL,
  `SENHA` varchar(50) NOT NULL,
  `STATUS` enum('A','I') NOT NULL,

  PRIMARY KEY (`ID`),
  UNIQUE KEY `USER_UNIQUE` (`USER`),
  INDEX `NOME_INDEX` (`NOME`)
 )
```

Índices

- Se a tabela já existe:
  - *ALTER TABLE ou CREATE INDEX*:

```
ALTER TABLE nome_tabela ADD INDEX nome_indice
  (campo_tabela);

CREATE INDEX nome_indice ON nome_tabela
  (campo_tabela);
```

Índices

- Excluir Índices:

```
ALTER TABLE nome_tabela DROP INDEX nome_indice;

 DROP INDEX nome_indice;
```

Índices

- Mostra índices:

- SHOW INDEX FROM nome_tabela;

Índices

- Observação:
- Algumas vezes o MySQL não utilizará um índice, mesmo se algum estiver disponível.
  - Exemplo: quando o uso do índice necessita que o MySQL lesse mais de 30% dos registros na tabela.
  - A varredura da tabela é provavelmente mais rápido, já que ela necessitará de menos pesquisas em discos.