

Pesquisa básica em tabelas

José Antônio da Cunha

Disciplina: Banco de Dados

Introdução

- O comando utilizado para realizar pesquisas em tabelas, ou seja, extrair informações do banco de dados, é o **SELECT**.

```
SELECT [DISTINCT | ALL] { * | coluna [, coluna,...] }  
FROM tabela
```

Pesquisa básica em tabelas

- Ex01: para visualizar todas as linhas e colunas da tabela Autor, devemos utilizar o seguinte comando:

```
SELECT * FROM Autor
```

```
SELECT codigo, nome_autor FROM Autor
```

```
SELECT codigo AS 'ID', nome_autor AS 'Nome'  
FROM Autor
```

Pesquisa básica em tabelas

- Ex01: para visualizar todas as linhas e colunas da tabela Autor, devemos utilizar o seguinte comando:

```
SELECT 'ID'=codigo, 'Nome'=nome_autor AS  
FROM Autor
```

Order By

- Em SQL, devemos utilizar a cláusula ORDER BY para determinar a ordem em que são mostradas as linhas de uma consulta.

```
SELECT [DISTINCT | ALL] {* | coluna [, coluna,...]}  
FROM tabela  
ORDER BY coluna [, coluna,... ] [asc, desc]
```

Order By

- EX02: Liste os autores em ordem alfabética:

```
SELECT codigo, nome_autor FROM Autor  
ORDER BY nome_autor
```

```
SELECT codigo, nome_autor FROM Autor  
ORDER BY nome_autor desc
```

Order By

- EX02: mostre-me os CDs ordenados por código da gravadora e nome do CD:

```
SELECT codigo_gravadora, nome_cd FROM CD  
ORDER BY codigo_gravadora, nome_cd
```

Filtrando linhas - WHERE

- Para filtramos linhas em uma pesquisa, utilizamos a cláusula **WHERE**.

```
SELECT [DISTINCT | ALL {* | coluna [, coluna,... ]}  
FROM tabela  
WHERE condição
```


Filtrando linhas - WHERE

- Ex03: mostre-me todos os CDs cujo preço de venda seja menor que R\$ 10,00.

```
SELECT * FROM CD WHERE preco_venda < 10
```

- Ex03: mostre-me todos os CDs cujo preço de venda seja menor que R\$ 10,00 e a gravadora seja a 2.

```
SELECT * FROM CD  
WHERE preco_venda < 10 AND codigo_gravadora = 2
```

Filtrando linhas - WHERE

- Ex03: mostre-me todos os CDs cujo preço de venda seja maior que R\$ 11,00 e gravadora seja 2 ou 3.

```
SELECT * FROM CD
```

```
WHERE codigo_gravadora = 2 OR codigo_gravadora = 3  
      AND preco_venda > 11
```

Not ou !

- É utilizado para inverter o resultado de uma expressão lógica, negando o resultado da condição. Caso a condição seja verdadeira, será retornado falso e vice-versa.

```
SELECT nome_cd, preco_venda FROM CD  
WHERE NOT (preco_venda < 15)
```

```
SELECT nome_cd, codigo_gravadora, preco_venda FROM CD  
WHERE NOT (codigo_gravadora = 2 OR codigo_gravadora = 3) AND  
preco_venda > 11
```

IS NULL

- Sabemos que todas as colunas que não têm valor inicializado são nulas em banco de dados SQL. Logo, esse comando é utilizado para saber se o conteúdo da coluna foi ou não inicializado.

```
SELECT nome_gravadora, endereco FROM Gravadora  
WHERE endereco IS NULL
```

IS NOT NULL

- Somente aqueles que tiverem conteúdo atribuído serão mostrados.

```
SELECT nome_gravadora, endereco FROM Gravadora  
WHERE endereco IS NOT NULL
```

BETWEEN

- Esse operador serve para determinar um intervalo de busca.

```
SELECT nome_cd, data_lancamento FROM CD
      WHERE data_lancamento BETWEEN '01/01/1999' AND
                                     '01/12/2001'
```

```
SELECT nome_cd, preco_venda FROM CD
      WHERE preco_venda BETWEEN 9 AND 11
```

```
SELECT nome_cd, preco_venda FROM CD
      WHERE preco_venda NOT BETWEEN 9 AND 11
```

LIKE

- Com esse operador, podemos comparar cadeias de caracteres utilizando padrões de comparação (wildcard) para um ou mais caracteres. Normalmente, o caractere percentual (%) substitui zero, um ou mais caracteres e sublinha (_) substitui um caractere.

LIKE

Expressão	Explicação
Like 'A%'	Todas as palavras que iniciem com a letra A .
Like '%A'	Todas que terminem com a letra A .
Like '%A%'	Todas que tenham a letra A em qualquer posição.
Like 'A_'	String com dois caracteres que tenham a letra A na primeira posição.
Like '_A'	String de dois caractere cujo primeiro caractere seja qualquer um e a última letra seja A .
Like '_A_'	String de três caractere cuja segunda letra seja A .
Like '%A_'	Todas as strings que tenha a letra A na penúltima posição.
Like '_A%'	Tenha a letra A na segunda posição.

LIKE

```
SELECT * FROM Autor
```

```
WHERE nome_autor Like 'R%'
```

```
SELECT * FROM Autor
```

```
WHERE nome_autor Like '_1%'
```

```
SELECT * FROM Autor
```

```
WHERE nome_autor Like 'C_R%'
```

LIKE

```
SELECT codigo_cd, nome_cd, preco_venda FROM CD  
WHERE preco_venda Like '1_'
```

IN

- Permite comparar o valor de uma coluna com um conjunto de valores. Normalmente, utilizamos o IN para substituir uma série de comparações seguidas da cláusula OR.

```
SELECT * FROM Autor  
WHERE codigo_autor IN (1, 10, 20)
```

Nota: a maior utilização do IN, é quando o utilizamos em subquery.

Fim