

# Operadores

6

- ❑ Na cláusula WHERE podemos utilizar um conjunto de operadores para filtrar o resultado da consulta.
- ❑ Os operadores podem ser:
  - ❑ Relacionais
  - ❑ Lógicos
  - ❑ Especiais

# Operadores Relacionais

7

- Utilizados para realizar comparações entre valores.

| Operador | Significado      | Exemplo           |
|----------|------------------|-------------------|
| =        | Igual            | Codigo_Autor = 2  |
| <        | Menor que        | Preco_Venda < 10  |
| <=       | Menor ou igual a | Preco_Venda <= 10 |
| >        | Maior que        | Preco_Venda > 10  |
| >=       | Maior ou igual a | Preco_Venda >= 10 |
| != ou <> | Diferente        | Codigo_Autor != 2 |

# Operadores Lógicos

8

- Realiza operações do tipo booleano (verdadeiro/falso).

| Operador | Significado | Apresenta resultado se:                 |
|----------|-------------|---|
| AND      | E           | 2 condições verdadeira                  |
| OR       | OU          | Apenas 1 das condições verdadeira       |
| NOT ou ! | Negação     | Inverte o resultado da expressão lógica |

# Operadores Especiais

9

- Os principais operadores especiais são:
  - IS NULL ou IS NOT NULL;
  - BETWEEN;
  - LIKE;
  - IN.

# Operadores Especiais

10

- IS NULL ou IS NOT NULL, verifica se o conteúdo do campo é nulo (IS NULL) ou não é nulo (IS NOT NULL).

| ID | Nome      | Telefone |
|----|-----------|----------|
| 1  | Maria     | null     |
| 2  | Joana     | 123456   |
| 3  | José      | 222333   |
| 4  | João      | null     |
| 5  | Mariana   | null     |
| 6  | Nickerson | 222222   |

# Operadores Especiais

11

- BETWEEN determina um intervalo para a consulta.
- Semelhante ao  $\geq$  e  $\leq$

| ID | Nome      | Telefone |
|----|-----------|----------|
| 1  | Maria     | null     |
| 2  | Joana     | 123456   |
| 3  | José      | 222333   |
| 4  | João      | null     |
| 5  | Mariana   | null     |
| 6  | Nickerson | 222222   |

# Operadores Especiais

12

- ❑ LIKE utilizado para comparação de texto.
- ❑ Pode utilizar expressões para retornar uma quantidade maior de tuplas.

| Expressão  | Explicação   |
|------------|--|
| LIKE 'A%'  | Todas as palavras que iniciem com a letra A        |
| LIKE '_A'  | String de dois caracteres cuja última letra seja A |
| LIKE '%A_' | Todos que tenham a letra A na penúltima posição    |

# Operadores Especiais

13

- ❑ IN compara o valor de uma coluna com um conjunto informado.
- ❑ Exemplo:

```
SELECT * from funcionario  
WHERE codigo IN (2, 4, 25, 30)
```



# Funções de agregação

14

- ❑ Na SQL existem algumas funções que agrupam valores.
- ❑ São elas:
  - ❑ COUNT: conta a quantidade de linhas
  - ❑ AVG: realiza a média aritmética da coluna
  - ❑ SUM: soma os valores da coluna
  - ❑ MIN: retorna o menor valor da coluna
  - ❑ MAX: retorna o maior valor da coluna

# Funções de agregação

15

| ID | Nome     | Valor | Quantidade |
|----|----------|-------|------------|
| 1  | Lápis    | 1,00  | 10         |
| 2  | Borracha | 3,00  | 10         |
| 3  | Caderno  | 15,00 | 10         |

```
SELECT COUNT(*) AS QTD FROM ITEM;
```

|  |     |
|--|-----|
|  | QTD |
|  | 3   |

```
SELECT MIN(ID) as menor FROM ITEM;
```

|  |       |
|--|-------|
|  | menor |
|  | 1     |

```
SELECT MAX(ID) as maior FROM ITEM;
```

|  |       |
|--|-------|
|  | maior |
|  | 3     |

```
SELECT AVG(valor) as media FROM ITEM;
```

|  |       |
|--|-------|
|  | media |
|  | 6.33  |

```
SELECT SUM(valor * quantidade) as valorTotal FROM ITEM;
```

|  |            |
|--|------------|
|  | valorTotal |
|  | 190.0      |

# Agrupando valores

16

- As funções de agregação também podem agrupar os valores de acordo com determinadas colunas.

```
SELECT COD_PRODUTO, SUM(VALOR_UNIT)  
FROM ITEM  
GROUP BY COD_PRODUTO;
```

|  | CDPROD | TOTAL  |
|--|--------|--------|
|  | 1      | 100.00 |
|  | 2      | 135.00 |

# Agrupando valores

17

- ❑ Podemos restringir os resultados das funções de agregação.
- ❑ Para isso utilizamos a cláusula HAVING.

```
SELECT CDPROD, SUM(VALOR_UNIT) TOTAL  
FROM ITEM  
GROUP BY COD_PRODUTO  
HAVING SUM(VALOR_UNIT) > 100;
```

|  | CDPROD | TOTAL  |
|--|--------|--------|
|  | 2      | 135.00 |

# Ordenando valores

18

- ❑ Para ordenar o resultado de uma pesquisa utilizamos a cláusula ORDER BY.
- ❑ Pode ser ordenado de forma ascendente (ASC) ou descendente (DESC).
- ❑ O padrão é ASC.

```
SELECT CDPROD, SUM(V valor_unit) TOTAL  
FROM ITEM  
GROUP BY COD_PRODUTO  
ORDER BY CDPROD;
```

```
SELECT CDPROD, SUM(V valor_unit) TOTAL  
FROM ITEM  
GROUP BY COD_PRODUTO  
ORDER BY CDPROD ASC;
```