

FUNÇÕES DE GRUPO E AGRUPAMENTO

Até agora trabalhamos com funções que tratavam apenas de uma linha. Contudo, em diversas situações é necessário trabalhar com várias linhas da consulta. Dessa forma, é necessário conhecer funções especiais denominadas funções de grupo e agrupamento.

7.1. FUNÇÕES DE GRUPO

Funções de grupo operam conjuntos de linhas visando a fornecer um resultado para o grupo. Estes grupos podem ser constituídos desde toda a tabela até subgrupos da tabela. Existem diversas funções de grupo que são implementadas pelo padrão SQL.

As principais funções são:

FUNÇÃO	DESCRIÇÃO
COUNT	Retorna número de linhas afetadas pelo comando.
SUM	Retorna o somatório do valor das colunas especificadas
AVG	Retorna a média aritmética dos valores das colunas
MIN	Retorna o menor valor da coluna de um grupo de linhas
MAX	Retorna o maior valor da coluna de um grupo de linhas
STDDEV	Retorna o desvio-padrão da coluna
VARIANCE	Retorna a variância da coluna

7.1.1. COUNT

Diferente das outras funções de grupo, o **COUNT** retorna o número de linhas que atende a uma determinada condição.

Podemos utilizá-lo com um asterisco entre parênteses, para indicar que queremos saber a quantidade total de linhas independente de haver linhas com colunas nulas ou não.

Exemplos:

```
SELECT COUNT(*)
FROM gravadora;

SELECT COUNT(grav_contato)
FROM gravadora;
```

Efeito semelhante pode ser conseguido com o seguinte comando:

Exemplos:

```
SELECT COUNT(grav_contato)
FROM gravadora
WHERE grav_contato IS NOT NULL;
```

Diferente de:

```
SELECT COUNT(*)
FROM gravadora
WHERE grav_contato IS NULL;
```

Outra maneira é utilizando o comando DISTINCT. Observe a diferença entre os dois comandos:

Exemplos:

```
SELECT COUNT(DISTINCT aut_id)
FROM musica_autor;

SELECT COUNT(aut_id)
FROM musica_autor;
```

7.1.2. SUM

Retorna o valor total de uma determinada coluna em um determinado grupo de linhas.

Assim se quisermos saber o total do preço de venda dos CD's, utilizamos o comando:

Exemplo:

```
SELECT SUM(cd_preco)
FROM cd;
```

É possível realizar cálculos baseados na somatória ou mesmo incluir outras colunas e operações no comando.

Exemplo:

```
SELECT SUM(cd_preco) * 1.2
FROM cd;
```

7.1.3. AVG

Extrai a média aritmética de um determinado grupo de linhas.

Para saber o preço médio dos CD's da loja execute o seguinte comando:

Exemplo:

```
SELECT AVG(cd_preco)
FROM cd;
```

7.1.4. MIN

Retorna o menor valor de uma coluna em um grupo de linhas. Podemos utilizá-la em colunas do tipo data ou alfanuméricas.

Para saber o preço de venda do CD mais barato execute o seguinte comando:

Exemplos:

```
SELECT MIN(cd_preco)
FROM cd;
```

```
SELECT MIN(cd_dt_lancamento)
FROM cd;
```

```
SELECT MIN(cd_nome)
FROM cd;
```

7.1.5. MAX

Retorna o maior valor de uma coluna em um grupo de linhas. Podemos utilizá-la em colunas do tipo data ou alfanuméricas.

Para saber o preço de venda do CD mais caro execute o seguinte comando:

Exemplos:

```
SELECT MAX(cd_preco)
FROM cd;
```

```
SELECT MAX(cd_dt_lancamento)
FROM cd;
```

```
SELECT MAX(cd_nome)
FROM cd;
```

7.1.6. STDDEV (Não funciona em alguns bancos de dados.)

Retorna o desvio padrão de uma determinada coluna.

Para saber o desvio padrão dos preços de venda dos CD's da loja execute o seguinte comando:

Exemplo:

```
SELECT STDDEV(cd_preco)
FROM cd;
```

7.1.7. VARIANCE (Não funciona em alguns bancos de dados.)

Retorna a variância de uma determinada coluna.

Para saber a variância do preço de venda dos CD's da loja execute o seguinte comando:

Exemplo:

```
SELECT VARIANCE (cd_preco)
FROM cd;
```

7.2. AGRUPANDO RESULTADOS

É possível agrupar valores com base em determinadas colunas. Desta forma, estaremos trabalhando um pequeno grupo de dados. Para isso usamos a cláusula **GROUP BY**.

7.2.1. GROUP BY

O comando GROUP BY deve vir antes da cláusula ORDER BY e depois do WHERE (se houver necessidade de utilizá-lo). É utilizado com uma função de grupo (COUNT, SUM, AVG, MIN ou MAX).

Para saber quantas músicas há em cada CD:

Exemplo:

```
SELECT cd_id, COUNT(*)
FROM faixa
GROUP BY cd_id;
```

Para o preço médio de venda de cada CD agrupado por gravadora:

Exemplo:

```
SELECT grav_id, AVG(cd_preco)
FROM cd
GROUP BY grav_id;
```

É possível realizar mais de uma função de grupo dentro de um mesmo SELECT. Neste exemplo, além do preço de venda, temos a quantidade de CD's de cada gravadora:

Exemplo:

```
SELECT grav_id, AVG(cd_preco), COUNT(*)
FROM cd
GROUP BY grav_id;
```

7.3. AGRUPAMENTOS COM MAIS DE UMA TABELA

Podemos unir mais de uma tabela com as regras já citadas.

Exemplo:

```
SELECT c.grav_id, g.grav_nome, AVG(c.cd_preco)
FROM cd c, gravadora g
WHERE c. grav_id = g.grav_id
GROUP BY c.grav_id, g.grav_nome
```

NOTA

É necessário colocar todas as colunas que fazem parte do comando SELECT na cláusula GROUP BY.

Caso se deseje ordenar o resultado, basta acrescentar o comando ORDER BY no final do comando após o comando GROUP BY:

Exemplo:

```
SELECT c.grav_id, g.grav_nome, AVG(c.cd_preco)
FROM cd c, gravadora g
WHERE c. grav_id = g.grav_id
GROUP BY c.grav_id, g.grav_nome
ORDER BY g.grav_nome desc
```

7.4. RESTRINGINDO O RESULTADO

Em algumas situações os dados agrupados podem ser restringidos ou filtrados. Até agora vimos que toda tabela era afetada pelo comando GROUP BY. Contudo, nem sempre isso é desejável.

Para isso podemos utilizar a cláusula WHERE em conjunto com GROUP BY ou utilizar a cláusula HAVING.

Retornar o total de autores agrupados por MUS_ID sendo este menor que 3:

Exemplo:

```
SELECT mus_id, COUNT(*)
FROM musica_autor
WHERE mus_id < 3
GROUP BY mus_id;
```

ou

```
SELECT mus_id, COUNT(*)
FROM musica_autor
GROUP BY mus_id
HAVING mus_id < 3;
```

NOTE BEM

Usando o filtro da cláusula WHERE as linhas são filtradas **antes** do agrupamento;

Usando HAVING, as linhas são filtradas **depois** do agrupamento.

A única restrição é que a cláusula HAVING **só pode usar as colunas que fazem parte do GROUP BY**. Para o WHERE isso não acontece.

Nunca use na cláusula WHERE uma função de grupo para filtrar os grupos. Dessa forma, o comando abaixo é inválido:

Exemplos:

```
SELECT GRAV_ID, AVG(cd_preco) FROM cd
WHERE AVG(cd_preco) > 12
GROUP BY grav_id
```

A forma correta seria:

```
SELECT grav_id, AVG(cd_preco) FROM cd
GROUP BY grav_id
HAVING AVG(cd_preco) > 12
```

7.5. EXERCÍCIOS

- a. Verifique o maior e menor valor entre os preços de venda dos CD's.
- b. Verifique a diferença entre o maior e menor valor dos preços de venda dos CD's.
- c. Verifique a quantidade de dias compreendidos entre a data mais atual e a data mais antiga das datas de lançamento dos CD's;
- d. Mostre através de uma consulta o tempo total do CD 1 (cd_id = 1);
- e. Mostre através de uma consulta a média de duração das músicas cujo nome comece com a letra A;
- f. Faça uma consulta que retorne a quantidade de CD's cadastrados.
- g. Faça uma consulta que retorne a o nome do CD e a média de duração das músicas em cada CD, desde que esta média não seja superior a 4.
- h. Faça uma consulta que retorne os autores que são autores de mais de uma música.