

Banco de Dados

Rebeca Barros



Grupos e Agrupamentos

MySQL



Grupos de dados

- Dados geralmente são armazenados no menor nível de granularidade necessário para algum usuário, permitindo ao menos analisar um **registro individual**.
- No entanto, existem situações em que é mais interessante analisar os dados de formas combinadas, agrupando e agregando em níveis maiores de granularidade.



Funções de Agregação

| Função | Ação |
|---------|---|
| MAX() | Retorna o maior valor da coluna de um grupo de linhas |
| MIN() | Retorna o menor valor da coluna de um grupo de linhas |
| AVG() | Retorna a média aritmética dos valores das colunas |
| SUM() | Retorna a soma do valor das colunas especificadas |
| COUNT() | Retorna o número de linhas afetadas pelo comando. |

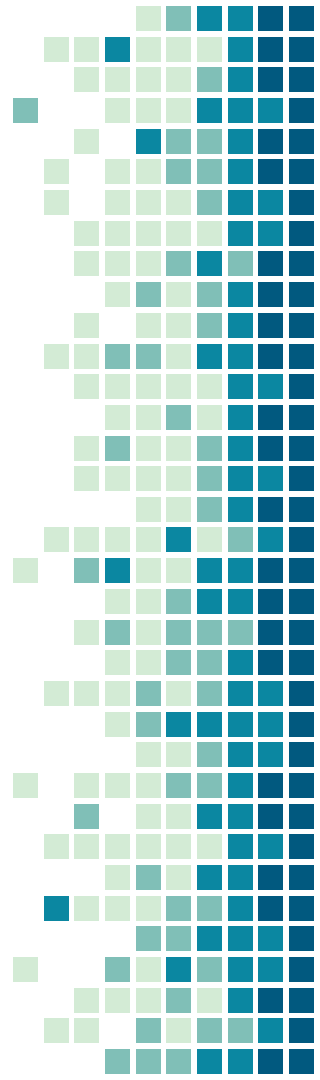
Funções de Agregação

-- exemplos de agregacao na tabela cd

```
SELECT min(preco_venda) AS menor, max(preco_venda) AS maior,  
SUM(preco_venda) AS soma, AVG(preco_venda) AS media FROM cd;
```

Funções de Agregação – Count

- Retorna o número de linhas que atende a uma determinada condição. Pode ser utilizado com um asterisco para indicar que busca-se saber a quantidade total de linhas, independente de possuir linhas com colunas nulas ou não.
- Caso busque-se saber a quantidade em uma determinada coluna, especifica-se essa coluna.



Funções de Agregação – Count

-- quantidade de registros de gravadora

```
SELECT count(*) FROM gravadora;
```

-- quantidade de registros com endereço na tabela Gravadora

```
SELECT count(enderecoGravadora) FROM gravadora;
```

Funções de Agregação – Count

-- quantidade de registros na tabela musica_compositor

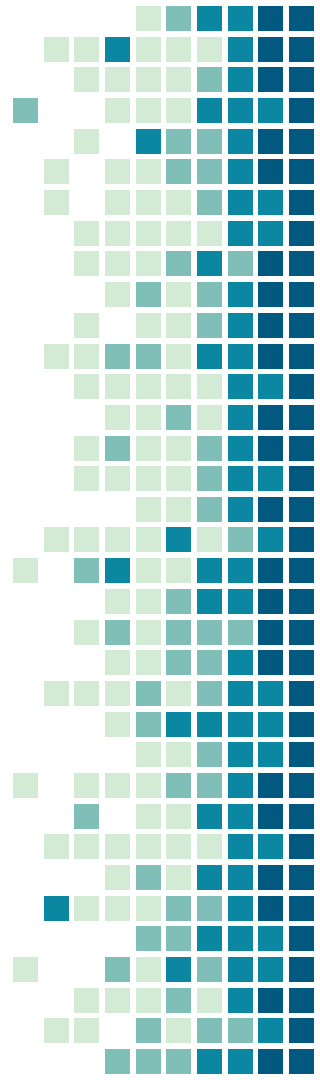
```
SELECT count(*) FROM musica_compositor;
```

-- quantidade de musicas distintas na tabela musica_compositor

```
SELECT count(DISTINCT Musica_idMusica) FROM musica_compositor;
```


Agrupamento – GROUP BY

- O SQL permite agrupar linhas com base em valores de determinadas colunas. Para isso é usado a cláusula **GROUP BY**.
- Esse comando deve vir antes do ORDER BY e depois do WHERE (caso haja necessidade de utilizá-los).



Agrupamento – GROUP BY

-- consulta para saber quantas faixas há em cada cd
`SELECT Cd_idCD, count(*) FROM faixa GROUP BY CD_idCD;`

-- preco medio de um cd por gravadora
`SELECT Gravadora_idGravadora, avg(preco_venda) FROM cd
GROUP BY Gravadora_idGravadora;`

-- preco medio e quantidade de cds por gravadora
`SELECT Gravadora_idGravadora AS Gravadora,
avg(preco_venda) AS Media, count(idCd) AS Quantidade FROM cd
GROUP BY Gravadora_idGravadora;`

Agrupamento – HAVING

- Pode ser necessário filtrar o resultado após um agrupamento, para isso é usado a cláusula **HAVING**.
- Enquanto na cláusula WHERE não é possível referenciar alguma função de agregação, na HAVING isso é possível, já que ela é executada após o agrupamento.

Agrupamento – HAVING

-- filtrando por grupos

```
SELECT Gravadora_idGravadora, avg(preco_venda) AS media FROM cd  
GROUP BY Gravadora_idGravadora HAVING media > 12;
```

/ preco medio e quantidade de cds por gravadora
ordenado por media de preco */*

```
SELECT Gravadora_idGravadora AS Gravadora,  
avg(preco_venda) AS Media, count(idCd) AS Quantidade  
FROM cd GROUP BY Gravadora_idGravadora  
ORDER BY Media;
```

Referências

- Beaulieu, A. (2009). *Learning SQL: Master SQL Fundamentals*. "O'Reilly Media, Inc."
- de OLIVEIRA, C. H. P. (2002). *SQL: curso prático*. Novatec.
- Documentação oficial do MySQL. Disponível em: <https://dev.mysql.com/doc/refman/8.0/en/>
- Curso Completo de MySQL (Bóson Treinamentos). Disponível em: <http://www.bosontreinamentos.com.br/curso-completo-de-mysql/>