



# Sistemas de Gerenciamento de Banco de Dados

Diogo Bortolini

# Estruturas (Conjuntos) de Linguagens SQL

**DDL** – Linguagem de Definição de Dados: criar novas tabelas e alterar ou excluir elementos associados. Comandos: CREATE – ALTER - DROP

**DML** – Linguagem de Manipulação de Dados: alterações dos dados no registro. Comandos: INSERT - UPDATE - DELETE

**DCL** – Linguagem de Controle de Dados: autorização de acesso ou manipulação dos dados. Comandos: GRANT - REVOKE

**DTL** – Linguagem de Transação de Dados: controlam as transações. Comandos: BEGIN WORK (START TRANSACTION) – COMMIT - ROLLBACK

**DQL** – Linguagem de Consulta de Dados: consulta dos dados. Comando: SELECT

# DQL

**DDL** – Linguagem de Definição de Dados: criar novas tabelas e alterar ou excluir elementos associados. Comandos: CREATE – ALTER - DROP

**DML** – Linguagem de Manipulação de Dados: alterações dos dados no registro. Comandos: INSERT - UPDATE - DELETE

**DCL** – Linguagem de Controle de Dados: autorização de acesso ou manipulação dos dados. Comandos: GRANT - REVOKE

**DTL** – Linguagem de Transação de Dados: controlam as transações. Comandos: BEGIN WORK (START TRANSACTION) – COMMIT - ROLLBACK

**DQL** – Linguagem de Consulta de Dados: consulta dos dados.  
Comando: SELECT

# Funções de Agregação

AVG(): Média de valores

COUNT(): Contagem de registros de uma consulta

COUNT(DISTINCT): Contagem de registros distintos

SUM(): Soma de valores

MAX(): Valor máximo retornado pela consulta

MIN(): Valor mínimo retornado pela consulta

STD(): Desvio Padrão



# Desvio Padrão

Funcionários	Quantidade de peças produzidas por dia				
	Segunda	Terça	Quarta	Quinta	Sexta
A	10	9	11	12	8
B	15	12	16	10	11
C	11	10	8	11	12
D	8	12	15	9	11

Variância → Funcionário D:

$$\text{var}(D) = \frac{(8 - 11)^2 + (12 - 11)^2 + (15 - 11)^2 + (9 - 11)^2 + (11 - 11)^2}{5}$$

$$\text{var}(D) = \frac{30}{5} = 6,0$$

Desvio Padrão → Funcionário D:

$$\begin{aligned}\text{dp}(D) &= \sqrt{\text{var}(D)} \\ \text{dp}(D) &= \sqrt{6,0} \\ \text{dp}(D) &\approx 2,45\end{aligned}$$

Funcionário D:  $11,0 \pm 2,45$  peças por dia

# Funções de Paginação/Apresentação

**DISTINCT:** Seleciona os valores únicos, sem repetição

~~ORDER BY:~~ Ordena o resultado baseado nas colunas informadas

~~LIMIT:~~ Limita o número de resultados retornados

**OFFSET:** Indica quantos registros devem ser avançados

# DISTINCT e COUNT

```
SELECT primeiro_nome FROM `sakila`.`ator` WHERE primeiro_nome LIKE 'An%';
```

```
SELECT DISTINCT primeiro_nome FROM `sakila`.`ator` WHERE primeiro_nome LIKE 'An%';
```

```
SELECT count(primeiro_nome) FROM `sakila`.`ator` WHERE primeiro_nome LIKE 'An%';
```

```
SELECT count(primeiro_nome) FROM `sakila`.`ator` WHERE primeiro_nome LIKE 'An%';
```

```
SELECT count(DISTINCT primeiro_nome) FROM `sakila`.`ator` WHERE primeiro_nome LIKE 'An%';
```

```
SELECT DISTINCT primeiro_nome, ultimo_nome FROM `sakila`.`ator` WHERE primeiro_nome LIKE 'An%';
```

# Agregações Exemplos

```
SELECT COUNT(*) FROM sakila.ator;
```

```
SELECT COUNT(*) FROM sakila.ator WHERE primeiro_nome like 'B%';
```

```
SELECT SUM(valor) FROM sakila.pagamento WHERE cliente_id = 1;
```

```
SELECT AVG(valor) FROM sakila.pagamento;
```

```
SELECT MAX(preco_da_locacao) FROM filme WHERE titulo LIKE "A%";
```

```
SELECT MIN(preco_da_locacao) FROM filme WHERE titulo LIKE "A%";
```



# Funções Matemáticas

CEILING(): Arredondar para cima

FLOOR(): Arredondar para baixo

POW(x,y): Retorna x elevado a y

SIGN(X) Retorna o sinal do argumento (-1, 0, ou 1, para negativo, zero ou positivo)

SQRT(): Raiz quadrada de um argumento

ROUND() Arredondar

TRUNCATE() Retorna a parte truncada (cortada)

```
SELECT ROUND(MAX(preco_da_locacao),1) FROM filme WHERE titulo LIKE "A%";
```

```
SELECT CEILING(MIN(preco_da_locacao)) FROM filme WHERE titulo LIKE "A%";
```

# Formatação do Resultados

CONCAT: unir na apresentação um número de colunas

UPPER ou UCASE: apresenta em letras maiúsculas

LOWER ou LCASE: apresenta em letras minúsculas

AS: apelido, serve para dar um nome provisório para exibição ou menção de tabela/coluna

# Formatação do Resultados

```
SELECT CONCAT(primeiro_nome, ' ', ultimo_nome) AS 'Nome Ator' FROM ator LIMIT 10;
```

```
SELECT UPPER(CONCAT(primeiro_nome, ' ', ultimo_nome)) AS 'Nome Ator' FROM ator LIMIT 10;
```

```
SELECT UCASE(CONCAT(primeiro_nome, ' ', ultimo_nome)) AS 'Nome Ator' FROM ator LIMIT 10;
```

```
SELECT LOWER(CONCAT(primeiro_nome, ' ', ultimo_nome)) AS 'Nome Ator' FROM ator LIMIT 10;
```

```
SELECT LCASE(CONCAT(primeiro_nome, ' ', ultimo_nome)) AS 'Nome Ator' FROM ator LIMIT 10;
```

```
SELECT CONCAT('R$ ', valor) AS 'Valor Pagamento', DATE_FORMAT(data_de_pagamento, "%d/%m/%Y") AS 'Data do Pagamento'  
FROM sakila.pagamento LIMIT 10;
```

# Funções de Agrupamento

GROUP BY: Agrupamento de registros por categoria

HAVING: Seleção de agrupamento

```
SELECT ultimo_nome FROM sakila.ator GROUP BY ultimo_nome;
```

```
SELECT ultimo_nome, COUNT(*) FROM sakila.ator GROUP BY ultimo_nome;
```

```
SELECT ultimo_nome, COUNT(*) FROM sakila.ator GROUP BY ultimo_nome HAVING COUNT(*) > 3;
```

# Referências

- RODRIGUEZ, F. T. **Um pouco sobre a história dos bancos de dados**. Save Point 2017.
- GOMES, Daniella. **Modelagem de dados: 1:N ou N:N?** Disponível em: <https://www.devmedia.com.br/modelagem-1-n-ou-n-n/38894>
- FERREIRA, Nickerson Fonseca. **Normalização SQL**. IFRN, 2019
- DATE, C. J. **Introdução a sistemas de bancos de dados**. Rio de Janeiro: Campus, 2004.
- RAMAKRISHNAN, Raghu. **Sistemas de gerenciamento de banco de dados**. São Paulo: McGraw Hill, 2009.

# Referências

- MILANI, ANDRÉ. **Construindo Aplicações Web com PHP e Mysql**. Novatec, 2010.
- SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. **Sistema de banco de dados**. Rio de Janeiro: Elsevier, 2006.
- MACHADO, Felipe Nery R.; ABREU, M. **Projeto de Banco de Dados: Uma visão prática**. 15ª ed. São Paulo: Érica, 2008.
- <https://dev.mysql.com/doc/refman/8.0/en/storage-engines.html>