

# **Aprofundando em Banco de Dados Relacional e SQL**

## **Aula 14: Consultas avançadas em SQL**

**Prof. MSc. Kizzy Terra**

PretaLab  
Ciclo Formativo Intermediário  
30/09/2025

# Conteúdo

- Lidando com Arquivos, Pacotes e Módulos
- Banco de Dados Relacionais e SQL
- **Aprofundando em Banco de Dados Relacionais e SQL**
- Tratamento de Dados Utilizando Pandas e Numpy
- Estatística com Python - Probabilidade, Amostragem e Testes de Hipóteses
- Análise de Dados do Mundo Real - Projeto Final

## Semanalmente:

- Aulas teóricas (com prática) nas **segundas e terças**
- Aula focadas em exercícios práticos nas **quintas**
  - Se organize para isso!

# Subconsultas

Uma subconsulta SQL (ou subquery, consulta aninhada ou subselect) é **uma instrução SELECT embutida dentro de outra instrução SQL**, que fornece dados para a consulta principal por meio das cláusulas WHERE, FROM, ou SELECT.

Essa consulta interna é executada primeiro, e os resultados são usados como filtros ou valores de comparação pela consulta externa.

As subconsultas podem retornar um valor único (escalar), uma linha com vários valores, ou uma tabela inteira.

## Exemplo 1 – WHERE

```
SELECT titulo  
FROM filmes  
WHERE id IN (  
    SELECT filme_id FROM avaliacoes WHERE nota = 5  
);
```

## Exemplo 2 – SELECT

```
SELECT f.titulo,  
      (SELECT MAX(nota)  
       FROM avaliacoes a  
       WHERE a.filme_id = f.id) AS maior_nota  
FROM filmes f;
```

## Exemplo 3 – HAVING

```
SELECT genero, AVG(a.nota)
FROM filmes f
JOIN avaliacoes a ON f.id = a.filme_id
GROUP BY genero
HAVING AVG(a.nota) > (SELECT AVG(nota) FROM avaliacoes);
```

## Exemplo 4 – FROM

```
SELECT f.titulo, m.media  
FROM filmes f  
JOIN (  
    SELECT filme_id, AVG(nota) AS media  
    FROM avaliacoes  
    GROUP BY filme_id  
) m ON f.id = m.filme_id;
```



# Prática 1

👉 Liste os filmes que possuem média de notas maior que 4.



## Prática 2

👉 Liste os gêneros cuja média de notas é menor que a média geral.

# CASE... WHEN

A instrução CASE WHEN no SQL é uma **expressão condicional** que permite executar lógica de "**se/então/senão**" dentro de uma consulta, atribuindo um valor com base na verificação de condições específicas em dados de uma tabela.

# Exemplo 1 - CASE simples

```
SELECT genero,  
       CASE genero  
         WHEN 'Ação' THEN 'Adrenalina'  
         WHEN 'Drama' THEN 'Emocionante'  
         ELSE 'Outro'  
       END AS categoria  
FROM filmes;
```

## Exemplo 2

```
SELECT titulo, ano,  
       CASE  
         WHEN ano < 1900 THEN 'Clássico'  
         WHEN ano BETWEEN 1900 AND 2020 THEN 'Moderno'  
         ELSE 'Atual'  
       END AS periodo  
FROM filmes;
```

## Exemplo 3 - CASE em agregação

```
SELECT  
    SUM(CASE WHEN nota = 5 THEN 1 ELSE 0 END) AS otimas,  
    SUM(CASE WHEN nota < 3 THEN 1 ELSE 0 END) AS ruins  
FROM avaliacoes;
```

## Prática 3

- ☞ Consulte os filmes e classifique-os em:
- “Antes de 2010”
  - “Depois de 2010”

# Prática 4

- ☞ Consulte as avaliações, classificando-as em:
- “Ótima” (nota = 5)
  - “Boa” (nota entre 3 e 4)
  - “Ruim” (nota < 3)

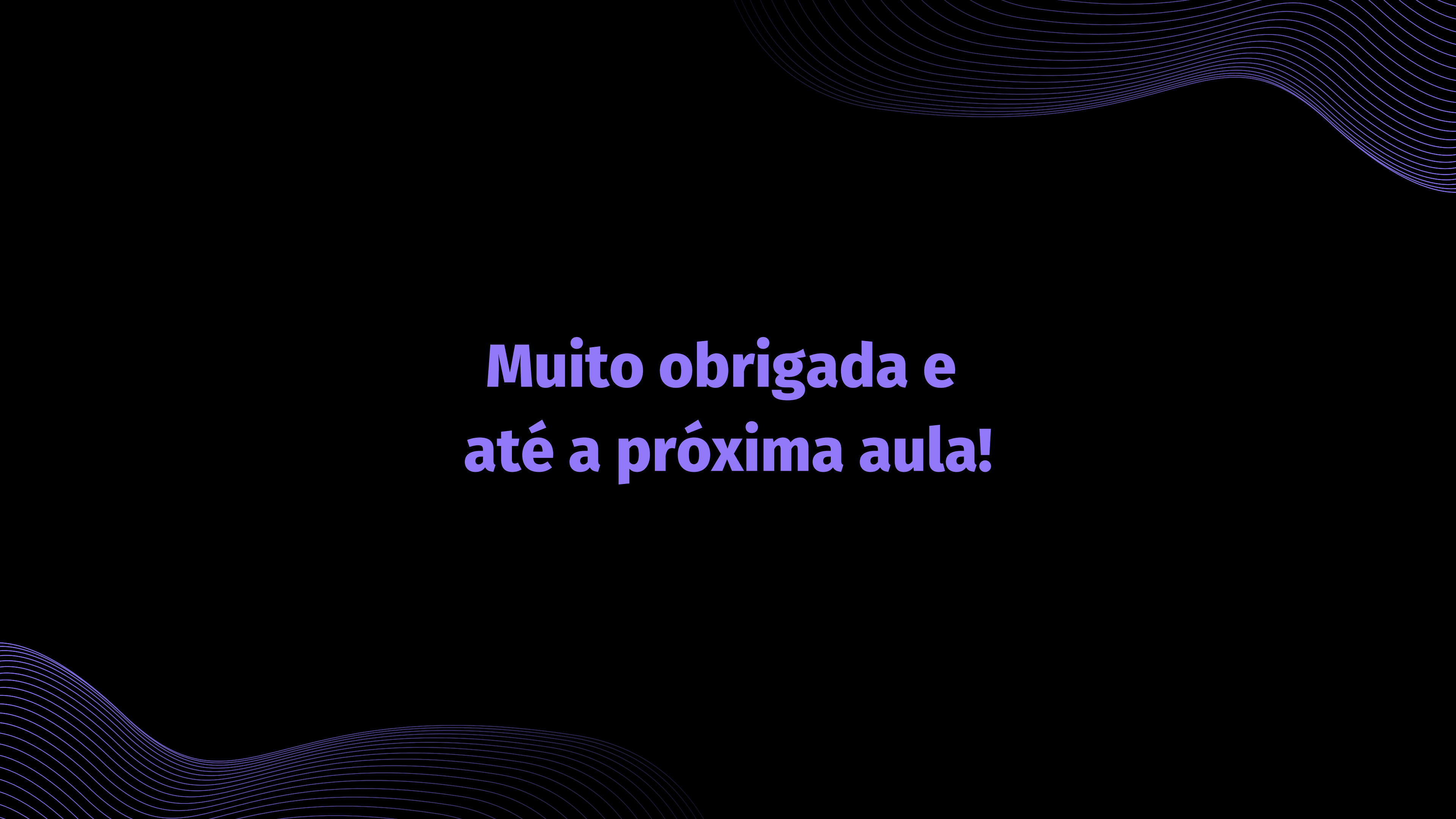
**Exercícios resolvidos:**

<https://www.db-fiddle.com/f/iCENLmdAvCtdTzeJmaFD2T/23>





**Dúvidas?**



**Muito obrigada e  
até a próxima aula!**