



Semana 7 – Subconsultas (Subqueries): O Coração do SQL Avançado

"Domine consultas aninhadas e leve seu SQL para o próximo nível"

Professor: Ailson Gomes

Curso: Desenvolvimento de sistemas

Introdução ao Tema

↔ O que é uma Subconsulta?

Consulta **aninhada** dentro de outra consulta SQL

Executada **antes** da consulta principal

Pode retornar um único valor ou múltiplos valores

💡 Quando Usar?

Para **filtrar** dados com base em resultados de outra consulta

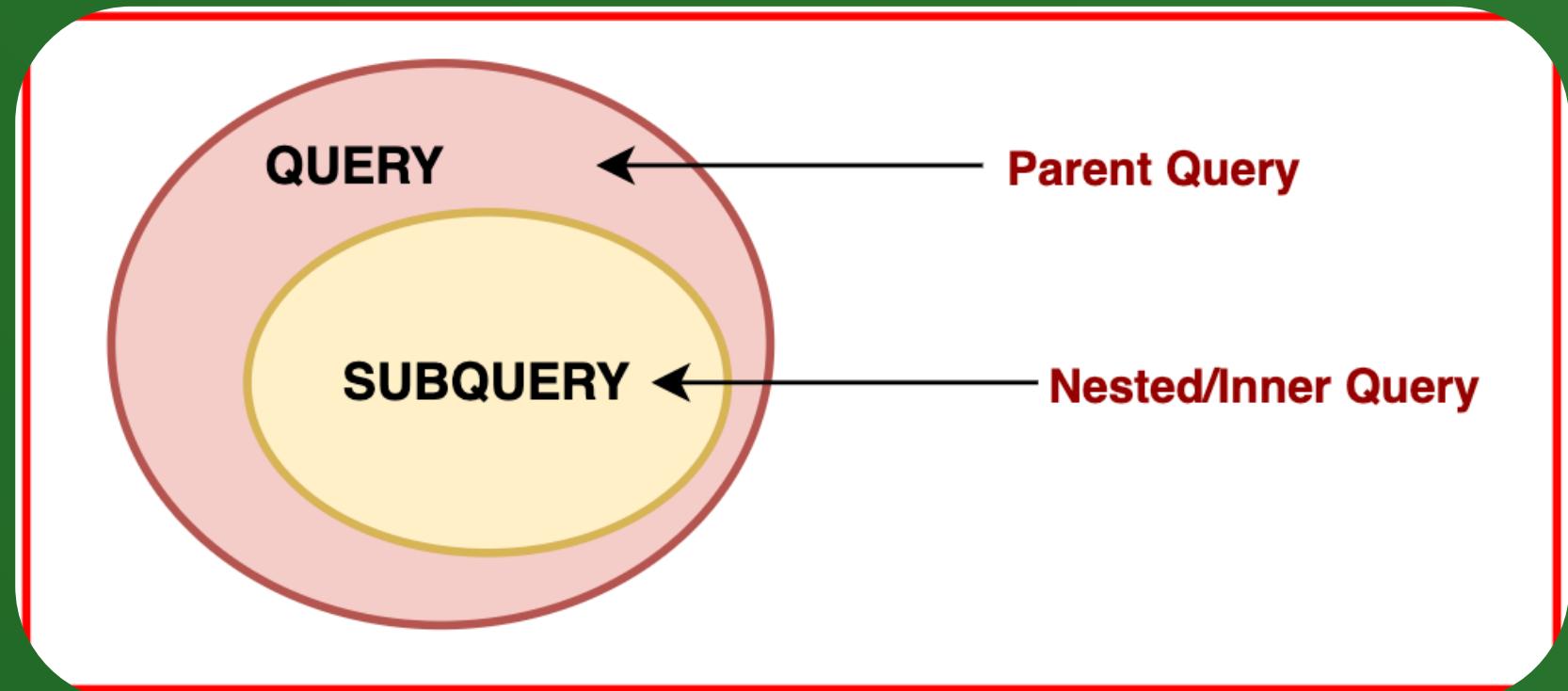
Quando precisa de **agregações** complexas

Para comparar dados entre diferentes tabelas

★ Benefícios

Flexibilidade na construção de consultas

Performance otimizada em certos cenários



Sintaxe Básica de Subconsultas

💡 Estrutura Fundamental

Consulta **interna** entre parênteses

Posicionada em cláusulas **WHERE, FROM, HAVING**

Executada **antes** da consulta externa

⌚ Execução

Processamento **de dentro para fora**

Subconsulta → Resultado → Consulta principal

⚡ Exemplo com IN

```
SELECT coluna1  
FROM tabela1  
WHERE coluna1 IN (  
    SELECT coluna  
    FROM tabela2  
    WHERE condição  
);
```

Exemplo Prático Inicial

Cenário

Encontrar alunos com idade **acima da média** da turma

Problema: Precisamos primeiro calcular a média e depois comparar

Explicação Passo a Passo

- 1 Subconsulta **calcula a média** de idade
- 2 Consulta **principal filtra** alunos acima desse valor
- 3 Resultado: **apenas alunos** com idade acima da média

Subconsulta Escalar

```
SELECT nome, idade  
FROM alunos  
WHERE idade > (  
    SELECT AVG(idade)  
    FROM alunos  
)
```

Vantagens

- ✓ **Uma única consulta** resolve o problema
- ✓ Código **mais legível** e organizado

Tipos de Subconsultas



Escalares

Retornam **um único valor**

Usadas em comparações diretas

```
SELECT AVG(idade)  
FROM alunos
```



Lista

Retornam **múltiplos valores**

Usadas com IN, ANY, ALL

```
SELECT curso  
FROM alunos  
GROUP BY curso
```



Correlacionadas

Dependem da consulta externa

Executadas linha a linha

```
SELECT MAX(nota)  
FROM alunos b  
WHERE b.curso = a.curso
```



Comparação entre Tipos

Tipo	Retorno	Execução	Complexidade
Escalar	1 valor	1 vez	Baixa
Lista	N valores	1 vez	Média
Correlacionada	Depende	N vezes	Alta

Subconsultas Escalares

1 Definição

- ✓ Retorna **um único valor**
- ✓ Usada em **comparações diretas**
- ✓ Posicionada onde se espera **um valor**

2 Uso Comum

- Comparações com `=, >, <, >=, <=, <>`
- Funções de **agregação** (AVG, SUM, MAX, MIN)
- Cláusulas **WHERE, HAVING**

3 Execução Única

- Subconsultas escalares são executadas **apenas uma vez**, tornando-as mais eficientes que outros tipos

↔ Exemplo: Idade Acima da Média

```
SELECT nome, idade
FROM alunos
WHERE idade > (
    SELECT AVG(idade)
    FROM alunos
);
```

4 Resultado

Nome	Idade
Maria	22
João	23
Carlos	24

Subconsultas de Lista

☰ Definição

- ✓ Retorna **múltiplos valores**
- ✓ Usada com operadores **específicos**
- ✓ Executada **uma única vez**

🔧 Operadores

IN

Verifica se valor está na lista

NOT IN

Verifica se valor não está na lista

ANY

Compara com qualquer valor da lista

ALL

Compara com todos os valores

↔ Exemplo: Cursos com Mais de 3 Estudantes

```
SELECT nome, curso  
FROM alunos  
WHERE curso IN (  
    SELECT curso  
    FROM alunos  
    GROUP BY curso  
    HAVING COUNT(*) > 3  
);
```

📊 Resultado

Nome	Curso
Maria	Engenharia
João	Engenharia
Carlos	Medicina
Ana	Medicina

Subconsultas Correlacionadas

→ Definição

- ✓ **Depende** da consulta externa
- ✓ Referencia colunas da **consulta pai**
- ✓ Executada **linha a linha**

↻ Execução Linha a Linha

- 1 Consulta externa processa **uma linha**
- 2 Subconsulta usa valores **dessa linha**
- 3 Processo se **repete** para cada linha

Alta Complexidade

- ⚠ Executada **N vezes** (uma para cada linha da consulta externa)

↔ Exemplo: Melhor Aluno de Cada Curso

```
SELECT a.nome, a.curso, a.nota_media
FROM alunos a
WHERE a.nota_media = (
  SELECT MAX(nota_media)
  FROM alunos b
  WHERE b.curso = a.curso
);
```

Resultado

nome	curso	nota_media
Maria	Engenharia	9.5
João	Medicina	9.8
Ana	Direito	9.2

Operadores Especiais



EXISTS / NOT EXISTS

Verifica a **existência** de registros na subconsulta

💡 Quando Usar

- › Para testar **presença/ausência** de dados
- › Em substituição a **JOINS complexos**
- › Para **semi-joins** eficientes



ANY / ALL

Compara um valor com **cada elemento** de uma lista

💡 Quando Usar

- › **ANY**: comparação com qualquer valor
- › **ALL**: comparação com todos valores
- › Para **validações de limite**

Operadores Especiais

«» EXISTS

```
SELECT nome
FROM alunos a
WHERE EXISTS (
  SELECT *
  FROM matriculas m
  WHERE m.id_aluno = a.id
);
```

«» ANY / ALL

```
-- ANY: qualquer nota acima da média
SELECT nome
FROM alunos
WHERE nota > ANY (
  SELECT nota
  FROM medias
);
```



Performance em Tabelas Grandes

EXISTS geralmente **mais eficiente** que IN para grandes volumes de dados, pois para na primeira correspondência

Aliases em Subconsultas

① Por que Usar Aliases?

- ✓ Diferenciar tabelas repetidas
- ✓ Referenciar a mesma tabela múltiplas vezes
- ✓ Simplificar nomes longos de tabelas



Legibilidade

Código mais claro e fácil de entender



Manutenção

Facilita alterações e correções futuras

↔ Exemplo Prático

```
SELECT a.nome, a.curso  
FROM alunos a  
WHERE a.nota_media > (  
    SELECT AVG(b.nota_media)  
    FROM alunos b  
    WHERE b.curso = a.curso  
);
```

b =

alunos (subconsulta)

a =

alunos (consulta externa)



Boas Práticas

- ★ Use aliases **curtos e descritivos**
- ★ Seja **consistente** na nomenclatura
- ★ Documente aliases **complexos**

Ordenação com ORDER BY

Sintaxe Básica



`ORDER BY coluna [ASC|DESC]`

Combinação com Subconsultas

- ✓ Subconsulta **filtra** dados
- ✓ `ORDER BY` **organiza** resultados
- ✓ Combinados para **rankings** e listas

ASC

Crescente (A-Z, 0-9)

DESC

Crescente (A-Z, 0-9)

↔ Exemplo: Top 5 Alunos Acima da Média

```
SELECT nome, curso, nota_media
FROM alunos
WHERE nota_media > (
  SELECT AVG(nota_media)
  FROM alunos
)
ORDER BY nota_media DESC
LIMIT 5;
```

Resultado

nome	curso	nota_media
Maria	Engenharia	9.8
João	Medicina	9.7
Carlos	Direito	9.5
Ana	Engenharia	9.3
Pedro	Medicina	9.2

Exercícios Práticos

Subconsulta Escalar Fácil

Encontrar alunos com nota **acima da média** geral

 Use AVG() na subconsulta

Subconsulta Correlacionada Médio

Identificar o **melhor aluno** de cada curso

 Compare com MAX() por curso

Comparativa Difícil

Alunos com nota **acima da média** do seu próprio curso

 Use subconsulta correlacionada com AVG()

Subconsulta de Lista Fácil

Listar cursos com **mais de 4 alunos** matriculados

 Use GROUP BY + HAVING + IN

Ranking Médio

Exibir o **top 3 alunos** de cada curso

 Combine subconsulta com ORDER BY + LIMIT



Desafio 1: Média por Cidade

Médio

Calcule a **média de notas** dos alunos agrupados por cidade

- 💡 Use GROUP BY com subconsulta para filtrar cidades com média acima da geral



Desafio 2: Relatório Executivo

Difícil

Crie um **relatório completo** por curso com estatísticas detalhadas

- 💡 Combine múltiplas subconsultas para calcular médias, máximos e mínimos



Desafio 3: Ranking Completo

Difícil

Desenvolva um **ranking geral** com posição de cada aluno

- 💡 Use subconsulta correlacionada para contar alunos com notas maiores



Desafio 4: Top/Bottom 20%

Expert

Identifique os **20% melhores e piores** alunos de cada curso

- 💡 Combine subconsultas com cálculo de percentis e LIMIT/OFFSET



Desafio 5: Dashboard por Turma

Expert

Construa um **painel completo** com métricas por turma e período

- 💡 Use subconsultas aninhadas para criar múltiplas métricas em uma única consulta

Boas Práticas e Performance



Índices

Use **índices** nas colunas referenciadas em subconsultas para melhorar performance



Subconsultas Correlacionadas

Tenha **cuidado** com subconsultas correlacionadas em tabelas grandes



JOINs

Prefira JOINs em casos críticos de performance



Aliases

Use **aliases** e nomes claros para facilitar manutenção



Ordenação

Ordene apenas quando necessário para otimizar recursos



Análise de Performance

Monitore o plano de execução para identificar gargalos



Lembre-se: a **legibilidade** do código é tão importante quanto a performance. Escreva consultas que outros desenvolvedores possam entender facilmente.

Conclusão e Próximos Passos

Você Aprendeu:

-  **Conceito** e importância das subconsultas
-  **3 tipos principais:** escalares, lista e correlacionadas
-  Uso de **aliases** e ordenação
-  **Exercícios práticos** e desafios avançados

 **Escalares**

 **Lista**

 **Correlacionadas**

 **EXISTS**

 **ANY/AL
L**



Próxima Aula

JOINS Avançados + Subconsultas Combinadas



Aplicações Práticas

-  Análises de **dados complexos**
-  Criação de **relatórios dinâmicos**
-  Otimização de **consultas SQL**



Domine subconsultas e eleve suas habilidades
em SQL para o próximo nível!