



# Consultas aninhadas

## Prof. Ramon Souza

Em alguns casos, precisamos realizar uma consulta que é comparada com o resultado de outra consulta. Aqui teremos o uso de uma **consulta dentro de outra consulta** ou **consulta aninhada**. A consulta que é realizada dentro de outra é chamada subconsulta.



A consulta que é realizada dentro de outra é chamada subconsulta. Uma **subconsulta, consulta interna ou seleção interna** é uma consulta que está aninhada dentro de uma instrução SELECT, INSERT, UPDATE ou DELETE ou em outra subconsulta. Uma subconsulta pode ser usada em qualquer lugar em que é permitida uma expressão.

As subconsultas podem ser comparadas com a consulta externa com o uso de operadores IN (ou NOT IN) ou EXISTS (ou NOT EXISTS), além dos operadores básicos =, <, <=, >, >=, <>.

# Consultas aninhadas (exemplo)

ProdutoID	Nome_do_Produto	FornecedorID	CategoriaID	Unidade	Preco
1	Leite	1	1	Litros	3
2	Banana	1	1	Kilogramas	5
3	Melancia	1	2	Unidade	6
4	Pão	2	2	Pacote	4
5	Suco	2	2	Litros	8

ProdutoID	Nome_do_Produto
1	Leite
5	Suco

```
SELECT Nome_do_Produto, AVG(Preco) FROM
Produtos WHERE ProdutoID IN (SELECT ProdutoID
FROM ProdutosLiquidos);
```

O retorno dessa consulta será 5,5.

A cláusula **EXISTS** faz uma **verificação se existe algum resultado para a subconsulta informada**. Caso haja, o **resultado da consulta principal é exibido**. É muito comum sua utilização quando se deseja trazer resultados onde um valor específico existe dentro de outra tabela.

```
SELECT colunas FROM tabela WHERE EXISTS (SELECT colunas FROM tabela WHERE condição);
```

Da mesma forma, também há a cláusula **NOT EXISTS**, somente **retorna o resultado da consulta principal, se não houver nenhum resultado para a subconsulta.**

```
SELECT colunas FROM tabela WHERE NOT EXISTS (SELECT colunas  
FROM tabela WHERE condição);
```

# EXISTS (exemplo)

<b>id</b>	<b>nome</b>	<b>preco</b>	<b>Id_categoria</b>
1	Bola	35.00	1
2	Patinete	120.00	1
3	Carrinho	15.00	1
4	Skate	296.00	1
5	Notebook	3500.00	2
6	Monitor LG 19	450.00	2
7	O Diário de Anne Frank	45.00	3
8	O dia do Curinga	65.00	3
9	O mundo de Sofia	48.00	3
10	Através do Espelho	38.00	3

**Tabela 1.** Tabela produto

<b>id</b>	<b>nome</b>
1	Infantil
2	Informatica
3	Educacional

**Tabela 2.** abela categoria\_produto

<b>id</b>	<b>id_produto</b>	<b>valor</b>	<b>data</b>
1	1	35.00	2018-05-15
2	1	35.00	2018-06-15
3	1	35.00	2018-07-15
4	2	120.00	2018-07-15
5	2	120.00	2018-07-14
6	3	15.00	2018-07-15
7	7	45.00	2018-07-15
8	8	65.00	2018-07-15
9	8	65.00	2018-07-16
10	9	48.00	2018-07-16
11	5	3500.00	2018-07-16
12	5	3500.00	2018-07-16
13	6	450.00	2018-07-16

**Tabela 3.** Tabela venda\_produto

<b>id</b>	<b>nome</b>
1	Bola
2	Patinete
3	Carrinho
5	Notebook
6	Monitor LG 19
7	O Diário de Anne Frank
8	O dia do Curinga
9	O mundo de Sofia

**Tabela 4.** Lista de produtos que tiveram alguma venda

```
SELECT p.id, p.nome FROM produto p
WHERE EXISTS (SELECT
v.id_produto FROM venda_produto v
WHERE v.id_produto = p.id);
```

## ATENÇÃO!!!

A cláusula EXISTS retorna TRUE (verdadeiro) se existir algum registro que cumpre a subconsulta. A consulta externa, por sua vez, retorna o registro. Caso contrário, o EXISTS retorna FALSE (falso) e a consulta externa não irá exibir o registro.

<b>id</b>	<b>nome</b>	<b>preco</b>	<b>Id_categoria</b>
1	Bola	35.00	1
2	Patinete	120.00	1
3	Carrinho	15.00	1
4	Skate	296.00	1
5	Notebook	3500.00	2
6	Monitor LG 19	450.00	2
7	O Diário de Anne Frank	45.00	3
8	O dia do Curinga	65.00	3
9	O mundo de Sofia	48.00	3
10	Através do Espelho	38.00	3

**Tabela 1.** Tabela produto

<b>id</b>	<b>id_produto</b>	<b>valor</b>	<b>data</b>
1	1	35.00	2018-05-15
2	1	35.00	2018-06-15
3	1	35.00	2018-07-15
4	2	120.00	2018-07-15
5	2	120.00	2018-07-14
6	3	15.00	2018-07-15
7	7	45.00	2018-07-15
8	8	65.00	2018-07-15
9	8	65.00	2018-07-16
10	9	48.00	2018-07-16
11	5	3500.00	2018-07-16
12	5	3500.00	2018-07-16
13	6	450.00	2018-07-16

**Tabela 3.** Tabela venda\_produto

```
SELECT p.id, p.nome FROM produto p WHERE EXISTS (SELECT v.id_produto
FROM venda_produto v WHERE v.id_produto = p.id);
```

**(CESPE - 2016 - TCE-PA - Auditor de Controle Externo - Área Informática - Analista de Suporte)** No que concerne à linguagem SQL, julgue o item seguinte.

A inserção de uma consulta na cláusula from de outra consulta não configura erro de sintaxe.

**(FCC - 2020 - AL-AP - Analista Legislativo - Desenvolvedor de Banco de Dados)** Em um ambiente Oracle, em condições ideais, um Analista precisa criar uma consulta para exibir o nome e o salário dos funcionários cujos números de departamento e salários correspondam a números de departamento e salários de funcionários que ganham comissão. Considerando que os campos usados na instrução existem na tabela colaboradores, a instrução PL/SQL correta que o Analista deve utilizar é:

- a) SELECT nome, salario FROM colaboradores WHERE (salario, idDepartamento) HAVE comissao NOT NULL;
- b) SELECT nome, salario FROM colaboradores WHERE (salario, idDepartamento) = (SELECT salario, idDepartamento FROM colaboradores WHERE comissao IS NOT NULL);
- c) SELECT nome, salario FROM colaboradores WHERE (salario, idDepartamento) IN (SELECT salario, idDepartamento FROM colaboradores WHERE comissao IS NOT NULL);
- d) SELECT nome, salario FROM colaboradores WHERE (comissão!=0 OR comissão IS NOT NULL);
- e) SELECT nome, salario FROM colaboradores WHERE (salario, idDepartamento) IS (SELECT salario, idDepartamento FROM colaboradores WHERE comissao!=0);

**(FCC - 2020 - AL-AP - Analista Legislativo - Desenvolvedor de Banco de Dados)** Em um ambiente Oracle, em condições ideais, um Analista precisa criar uma consulta para exibir o nome e o salário dos funcionários cujos números de departamento e salários correspondam a números de departamento e salários de funcionários que ganham comissão. Considerando que os campos usados na instrução existem na tabela colaboradores, a instrução PL/SQL correta que o Analista deve utilizar é:

- a) SELECT nome, salario FROM colaboradores WHERE (salario, idDepartamento) HAVE comissao NOT NULL;
- b) SELECT nome, salario FROM colaboradores WHERE (salario, idDepartamento) = (SELECT salario, idDepartamento FROM colaboradores WHERE comissao IS NOT NULL);
- c) SELECT nome, salario FROM colaboradores WHERE (salario, idDepartamento) IN (SELECT salario, idDepartamento FROM colaboradores WHERE comissao IS NOT NULL);
- d) SELECT nome, salario FROM colaboradores WHERE (comissão!=0 OR comissão IS NOT NULL);
- e) SELECT nome, salario FROM colaboradores WHERE (salario, idDepartamento) IS (SELECT salario, idDepartamento FROM colaboradores WHERE comissao!=0);

(FGV - 2016 - CODEBA - Analista Portuário - Analista de Tecnologia da Informação)

Analise o comando SQL a seguir.

```
select nome
from x
where exists
    (select * from x x1 where x1.nome > x.nome)
```

Sabe-se que o atributo nome é a chave primária da tabela x e que há mais de dois registros na tabela, assinale a opção que indica a descrição correta do que é produzido pela execução desse comando no MS SQL Server.

- a) Todos os nomes.
- b) Um resultado vazio.
- c) O primeiro nome, na ordem alfabética.
- d) Todos os nomes, à exceção do último na ordem alfabética.
- e) O último nome, na ordem alfabética.



# Consultas aninhadas

## Prof. Ramon Souza