

INFORMAÇÕES

MÓDULO	AULA	INSTRUTOR
01 Fundamentos de Banco de Dados	04 - Subconsultas, CTEs e Funções Avançadas	Matheus Laureto

CONTEÚDO**1. Revisão da Aula Anterior**

-
- **WHERE** – Filtragem de registros usando operadores (=, <>, >, <, >=, <=, **LIKE**, **BETWEEN**, **IN**, **IS NULL**), podendo usar **NOT** para negação.
 - **ORDER BY** – Ordena resultados (**ASC** é padrão, **DESC** precisa ser declarado).
 - Funções de Agregação:
 - **COUNT()** – retorna à quantidade de registros;
 - **SUM()** – retorna a soma dos registros;
 - **AVG()** – retorna a média dos registros;
 - **MIN()** – retorna o menor valor;
 - **MAX()** – retorna o maior valor.
 - Agrupamento:
 - **GROUP BY** para agrupar e aplicar funções agregadas;
 - **HAVING** para filtrar depois do agrupamento.
-

JOINS:

- Conceito – Combinam registros de duas ou mais tabelas com base em campos chave.
- Tipos:
 - **INNER JOIN** – Apenas registros com correspondência em todas as tabelas.
 - **LEFT JOIN** – Todos os registros da tabela da esquerda + correspondentes da direita; onde não houver correspondência, vem **NULL**.
 - **RIGHT JOIN** – Igual ao **LEFT JOIN**, mas preserva todos os registros da tabela da direita.
 - **FULL JOIN** – Todos os registros de ambas as tabelas, com **NULL** quando não há correspondência.
 - **CROSS JOIN** – Produto cartesiano (todas as combinações possíveis); não usa **ON**.
 - **SELF JOIN** – Relaciona uma tabela com ela mesma, usando apelidos.

2. SUBCONSULTAS

O que são subconsultas e para que servem?

- São consultas dentro de outra consulta.
- Ex: Buscar um dado já processado (ou calculado) para ser utilizado na consulta principal.

Exemplo prático:

Quais funcionários ganham mais do que a média dos salários da empresa?

Sintaxe:

```
SELECT nome, salario
FROM funcionarios
WHERE salario > (SELECT AVG(salario) FROM funcionarios);
```

- Importante:
Quando estamos falando de subconsultas (ou subqueries), devemos lembrar de que os comandos internos têm prioridade na execução, e posteriormente os comandos externos.
Ou seja, o comando **"SELECT AVG(SALARIO) FROM FUNCIONARIOS"** será executado primeiro, com o resultado armazenado em memória e depois o restante da query será executada, utilizando no **WHERE** o resultado do comando.

4. CTEs – COMMON TABLE EXPRESSION

O que são CTEs e para que servem?

- CTEs são comando utilizados para criar tabelas temporárias somente para a consulta em tempo de execução (durante a sessão), sem precisar criar a tabela (tabela física ou tabela temporária).
- Ela é definida usando o comando **WITH**.

Ex: Listar apenas os clientes que já compraram mais de R\$3000 no total.

Sintaxe:

```
WITH total_por_cliente AS (  
    SELECT cliente_id, SUM(valor) AS total_gasto  
    FROM vendas  
    GROUP BY cliente_id  
)  
SELECT cliente_id, total_gasto  
FROM total_por_cliente  
WHERE total_gasto > 3000;
```

- Vantagens de usar CTE:
 - Melhor Legibilidade – comando mais simples, mais fáceis de entendimento, sem a necessidade de utilizar uma query gigante;
 - Reuso dentro da mesma consulta – em caso de utilização de cálculos, é possível utilizar o mesmo cálculo sem repetição de código, podendo ser chamada N vezes;
 - Depuração fácil – é possível testar as queries separadamente para encontrar erros;
 - E principal, PERFORMANCE! – uma CTE bem desenvolvida é extremamente performática, por se tratar de um “único objeto”.

5. Funções Avançadas de Texto

- **UPPER()** e **LOWER()**
 - Transforma o texto para maiúsculo (**UPPER**) ou minúsculo (**LOWER**).

Ex:

```
SELECT UPPER('Itb');  
SELECT LOWER('Itb');
```

- **TRIM()**, **LTRIM()** e **RTRIM()**
 - Remove espaços (ou caracteres específicos) do início e/ou fim do texto

Ex:

```
SELECT TRIM('  ITB  ');  
SELECT LTRIM('  ITB');  
SELECT RTRIM('ITB  ');
```

- **SUBSTRING()**
 - Extrai parte de um texto.

Ex:

```
SELECT SUBSTRING('123.456.789-00' FROM 1 FOR 3);
```

- **REPLACE()**
 - Substitui parte do texto.

Ex:

```
SELECT REPLACE('Meu e-mail é teste@email.com.br', 'email', 'itb');
```

- **POSITION()**
 - Mostra a posição de um caractere ou texto dentro de outro

Ex:

```
SELECT POSITION('@' IN 'teste@email.com');
```

6. Exercícios

1. Liste em vendas as colunas `CLIENTE_ID` e `VALOR`, dos registros que vendas sejam maiores que a média de todas as vendas.
2. Liste todas as vendas mostrando, junto, o total gasto pelo mesmo cliente (subconsulta na tabela `VENDAS`, `SUM` em valor).
3. Use uma CTE para somar o valor total gasto por cada cliente e depois selecione todos os registros dessa CTE.
4. Mostre o nome de todos os clientes em letras maiúsculas.
5. Mostre os 3 primeiros caracteres do nome de cada cliente.