

Trabalho de Banco de Dados - Projeto – Parte II (Continuação parte I)

1.2 consultas com select e where:

```
select
    *
from
    clientes
where
    id_cliente = 2;

select
    *
from
    produtos
where
    preco > 5000;
```

2. 2 consultas com GROUP BY e ORDER BY com funções de agregação

```
SELECT
    c.Nome AS NomeCliente, SUM(p.Preco) AS ValorTotalPedidos
FROM
    Clientes AS c
JOIN
    Pedidos AS p ON c.Id_cliente = p.Id_cliente
GROUP BY
    c.Nome
ORDER BY
    ValorTotalPedidos DESC;
```

```
SELECT
    prod.Nome AS NomeProduto,
    SUM(ped.Quantidade) AS QuantidadeTotalVendida
FROM
    Produtos AS prod
```

JOIN

Pedidos AS ped ON prod.Id_produto = ped.Id_produto

GROUP BY

prod.Nome

ORDER BY

QuantidadeTotalVendida ASC;

3. 2 consultas com operadores aritméticos (+, -, *, /)

SELECT

prod.Nome AS NomeProduto,

ped.Quantidade AS QuantidadePedida,

prod.Preco AS PrecoUnitario,

(ped.Quantidade * prod.Preco) AS PrecoTotalItem,

(prod.Estoque - 10) AS DiferencaEstoqueMinimo

FROM

Pedidos AS ped

JOIN

Produtos AS prod ON ped.Id_produto = prod.Id_produto;

SELECT

p.Id_pedido,

prod.Nome AS NomeProduto,

p.Quantidade,

p.Preco AS PrecoTotalItem,

(p.Preco / p.Quantidade) AS PrecoMedioPorUnidade,

(p.Preco * 0.95) AS PrecoComDesconto5Porcento

FROM

Pedidos AS p

JOIN

Produtos AS prod ON p.Id_produto = prod.Id_produto;

4. 3 consultas com operadores de comparação (=, !=, , etc.)

```
SELECT
    prod.Nome AS NomeProduto,
    ped.Quantidade AS QuantidadePedida,
    prod.Preco AS PrecoUnitario,
    (ped.Quantidade * prod.Preco) AS PrecoTotalItem,
    (prod.Estoque - 10) AS DiferencaEstoqueMinimo FROM
    Pedidos AS ped
JOIN
    Produtos AS prod ON ped.Id_produto = prod.Id_produto;
```

```
SELECT
    *
FROM
    Clientes
WHERE
    Nome = 'João Silva';
```

```
SELECT
    Id_pedido, Data_pedido, Quantidade
FROM
    Pedidos
WHERE
    Data_pedido < '2025-07-25';
```

5. 3 consultas com operadores lógicos (AND, OR)

```
SELECT
    *
FROM
    Clientes
WHERE
    Endereco LIKE '%São Paulo, SP%' AND Email IS NOT NULL;
```

```
SELECT
    Nome, Estoque, Preco
FROM
    Produtos
WHERE
    Estoque < 20 OR Preco > 5000.00;
```

```
SELECT
    p.Id_pedido, c.Nome AS NomeCliente, p.Data_pedido
FROM
    Pedidos AS p
JOIN
    Clientes AS c ON p.Id_cliente = c.Id_cliente
WHERE
    c.Nome = 'Maria Souza' OR c.Nome = 'Pedro Almeida';
```

6. 2 consultas com operadores lógicos e negação (NOT)

```
SELECT
    Nome
FROM
    Clientes
WHERE
    NOT Nome LIKE 'A%';
```

```
SELECT
    Nome, Cargo
FROM
    Funcionarios
WHERE
    NOT Cargo = 'Gerente de Vendas';
```

7. 3 consultas com operadores auxiliares (IS NULL, BETWEEN, LIKE, IN)

```
SELECT
    Nome,
    Email
FROM
    Clientes
WHERE
    Email IS NULL;
```

```
SELECT
    Nome,
    Preco
FROM
    Produtos
WHERE
    Preco BETWEEN 200.00 AND 1000.00;
```

```
SELECT
    Nome,
    Cargo
FROM
    Funcionarios
WHERE
    Cargo IN ('Atendente', 'Vendedor');
```

8. 3 consultas com funções de agregação (SUM(), AVG(), etc.)

```
SELECT
    SUM(Estoque) AS EstoqueTotal
FROM
    Produtos;
```

```
SELECT
    AVG(Preco) AS PrecoMedioProdutos
```

FROM

Produtos;

SELECT

COUNT(Id_cliente) AS TotalClientes

FROM

Clientes;

9. 2 consultas com funções de datas (NOW(), DATE(), YEAR(), etc.)

SELECT

Id_pedido, Data_pedido, Quantidade

FROM

Pedidos

WHERE

YEAR(Data_pedido) = 2025;

SELECT

Id_pedido, Data_pedido, Quantidade

FROM

Pedidos

WHERE

Data_pedido = CURDATE();

10. 3 sub-consultas com agrupamento e união de dados

SELECT

Nome

FROM

Clientes

WHERE

Id_cliente IN (

SELECT p.Id_cliente

```
FROM Pedidos AS p
JOIN Produtos AS prod ON p.Id_produto = prod.Id_produto
WHERE prod.Preco > (SELECT AVG(Preco) FROM Produtos)
GROUP BY p.Id_cliente);
```

```
SELECT
    Nome,
    Cargo
FROM
    Funcionarios
WHERE
    Id_funcionario IN (
        SELECT Id_funcionario
        FROM Pedidos
        WHERE Quantidade > 1
        GROUP BY Id_funcionario);
```

```
SELECT
    Nome,
    Preco
FROM
    Produtos
WHERE
    Id_produto IN (
        SELECT Id_produto
        FROM Pedidos
        WHERE Preco > (SELECT AVG(Preco) FROM Pedidos)
        GROUP BY Id_produto);
```

11. 3 consultas com JOIN e visualização de tabelas

```
SELECT
    ped.Id_pedido,
```

```
    ped.Data_pedido,
    cli.Nome AS NomeCliente,
    func.Nome AS NomeFuncionario,
    prod.Nome AS NomeProduto,
    ped.Quantidade,
    ped.Preco AS PrecoTotalItem
FROM
    Pedidos AS ped
JOIN
    Clientes AS cli ON ped.Id_cliente = cli.Id_cliente
JOIN
    Funcionarios AS func ON ped.Id_funcionario = func.Id_funcionario
JOIN
    Produtos AS prod ON ped.Id_produto = prod.Id_produto;
```

```
SELECT
    p.Nome AS NomeProduto,
    p.Preco AS PrecoUnitario,
    c.Nome AS ClienteUltimoPedido
FROM
    Produtos AS p
LEFT JOIN
    Pedidos AS ped ON p.Id_produto = ped.Id_produto
LEFT JOIN
    Clientes AS c ON ped.Id_cliente = c.Id_cliente;
```

```
SELECT
    f.Nome AS NomeFuncionario,
    f.Cargo,
    c.Nome AS NomeCliente,
    c.Email
```


FROM

Funcionarios AS f

CROSS JOIN

Clientes AS c

LIMIT 10;

12. 4 consultas com tipos de JOIN: INNER, LEFT, RIGHT

SELECT

ped.Id_pedido,

ped.Data_pedido,

cli.Nome AS NomeCliente,

func.Nome AS NomeFuncionario,

prod.Nome AS NomeProduto

FROM

Pedidos AS ped

INNER JOIN Clientes AS cli ON ped.Id_cliente = cli.Id_cliente

INNER JOIN Funcionarios AS func ON ped.Id_funcionario = func.Id_funcionario

INNER JOIN Produtos AS prod ON ped.Id_produto = prod.Id_produto;

SELECT

c.Nome AS NomeCliente,

p.Id_pedido,

p.Data_pedido,

p.Preco AS PrecoTotalItem

FROM

Clientes AS c

LEFT JOIN

Pedidos AS p ON c.Id_cliente = p.Id_cliente;

SELECT

prod.Nome AS NomeProduto,

prod.Preco AS PrecoUnitario,

```
ped.Id_pedido,  
ped.Quantidade
```

```
FROM
```

```
Pedidos AS ped
```

```
RIGHT JOIN -- Pedidos é a tabela da esquerda, Produtos é a da direita
```

```
Produtos AS prod ON ped.Id_produto = prod.Id_produto;
```

```
SELECT
```

```
prod.Nome AS NomeProduto,
```

```
prod.Preco AS PrecoUnitario,
```

```
ped.Id_pedido,
```

```
ped.Quantidade
```

```
FROM
```

```
Produtos AS prod -- Produtos agora é a tabela da esquerda
```

```
LEFT JOIN
```

```
Pedidos AS ped ON prod.Id_produto = ped.Id_produto;
```