

PROVA PRÁTICA DE BANCO DE DADOS I - SQL SERVER (Versão 1)

1. Liste o nome de todos os fornecedores: Selecione a coluna NomeFornecedor da tabela Fornecedores.

```
SELECT NomeFornecedor  
FROM Fornecedores;
```

2. Produtos na categoria 'Áudio' com estoque superior a 30: Selecione o NomeProduto e Estoque para todos os produtos que pertencem à categoria 'Áudio' e cujo estoque atual é maior que 30. Use WHERE.

```
SELECT NomeProduto, Estoque  
FROM Produtos  
WHERE Categoria = 'Áudio' AND Estoque > 30;
```

3. Pedidos realizados no primeiro trimestre de 2024: Selecione o PedidoID, ClienteID e DataPedido de todos os pedidos cuja data esteja entre '2024-01-01' e '2024-03-31' (inclusive). Use WHERE.

```
SELECT PedidoID, ClienteID, DataPedido  
FROM Pedidos  
WHERE DataPedido BETWEEN '2024-01-01' AND '2024-03-31';
```

4. Contagem de endereços por cliente: Conte quantos endereços (EnderecoID) estão cadastrados para cada cliente. Apresente o ClienteID e a contagem de endereços. Use GROUP BY.

```
SELECT ClienteID, COUNT(EnderecoID) AS ContagemEnderecos  
FROM EnderecosCliente  
GROUP BY ClienteID;
```

5. Valor total pago por método de pagamento: Calcule a soma do ValorPago para cada método de pagamento (MetodoPagamento). Apresente o método de pagamento e o valor total pago. Use GROUP BY.

```
SELECT MetodoPagamento, SUM(ValorPago) AS ValorTotalPago  
FROM Pagamentos  
GROUP BY MetodoPagamento;
```

6. Pagamentos com status 'Recusado' ou 'Estornado': Liste o PagamentoID, PedidoID, DataPagamento e StatusPagamento para todos os pagamentos cujo status seja 'Recusado' ou 'Estornado'. Use WHERE com OR ou IN.

```
SELECT PagamentoID, PedidoID, DataPagamento, StatusPagamento  
FROM Pagamentos  
WHERE StatusPagamento IN ('Recusado', 'Estornado');
```

7. Fornecedores que fornecem mais de 2 produtos: Liste o NomeFornecedor para todos os fornecedores que estão associados a mais de 2 produtos diferentes. Use JOIN com Produtos, GROUP BY pelo fornecedor e HAVING para filtrar pela contagem de produtos.

```
SELECT F.NomeFornecedor  
FROM Fornecedores F  
INNER JOIN Produtos P ON F.FornecedorID = P.FornecedorID  
GROUP BY F.FornecedorID, F.NomeFornecedor  
HAVING COUNT(P.ProdutoID) > 2;
```

8. Pedidos e o nome completo do cliente que o realizou: Selecione o PedidoID, DataPedido e o Nome completo do cliente (Nome + Sobrenome) para todos os pedidos. Utilize um JOIN entre as tabelas Pedidos e Clientes.

```
SELECT P.PedidoID, P.DataPedido, C.Nome + ' ' + C.Sobrenome AS  
NomeCompleto  
FROM Pedidos P  
INNER JOIN Clientes C ON P.ClienteID = C.ClienteID;
```

9. Itens de pedido com nome do produto e nome do fornecedor: Selecione o PedidoID, o NomeProduto, o NomeFornecedor e a Quantidade para todos os itens de pedido. Utilize JOINS entre ItensPedido, Produtos e Fornecedores. Ordene os resultados por PedidoID e, secundariamente, por NomeProduto (ascendente).

```
SELECT IP.PedidoID, P.NomeProduto, F.NomeFornecedor, IP.Quantidade  
FROM ItensPedido IP  
INNER JOIN Produtos P ON IP.ProdutoID = P.ProdutoID  
INNER JOIN Fornecedores F ON P.FornecedorID = F.FornecedorID  
ORDER BY IP.PedidoID ASC, P.NomeProduto ASC;
```

10. Pedidos entregues em São Paulo (SP): Liste o PedidoID, DataPedido e o Nome completo do cliente para todos os pedidos cujo endereço de entrega (EnderecoEntregaID) esteja em 'São Paulo' no estado de 'SP'. Use JOINS entre Pedidos, Clientes e EnderecosCliente, e WHERE para filtrar pela cidade e estado do endereço de entrega.

```
SELECT P.PedidoID, P.DataPedido, C.Nome + ' ' + C.Sobrenome AS  
NomeCompleto  
FROM Pedidos P  
INNER JOIN Clientes C ON P.ClienteID = C.ClienteID  
INNER JOIN EnderecosCliente E ON P.EnderecoEntregaID = E.EnderecoID  
WHERE E.Cidade = 'São Paulo' AND E.Estado = 'SP';
```

11. Clientes que possuem mais de um tipo de endereço cadastrado: Liste o Nome e Sobrenome dos clientes que possuem pelo menos dois registros na tabela EnderecosCliente com tipos de endereço (TipoEndereco) distintos. Use GROUP BY por cliente e HAVING na contagem distinta de tipos de endereço.

```
SELECT C.Nome, C.Sobrenome  
FROM Clientes C  
INNER JOIN EnderecosCliente E ON C.ClienteID = E.ClienteID  
GROUP BY C.ClienteID, C.Nome, C.Sobrenome  
HAVING COUNT(DISTINCT E.TipoEndereco) > 1;
```

12. Categorias de produtos com valor total de estoque superior a R\$ 10000.00: Liste as Categorias de produtos onde a soma total do valor do estoque (Estoque * Preço) para todos os produtos nessa categoria é maior que 10000.00. Use GROUP BY pela categoria e HAVING na soma (Estoque * Preço).

```
SELECT Categoria  
FROM Produtos  
GROUP BY Categoria  
HAVING SUM(Estoque * Preco) > 10000.00;
```

13. Pedidos com valor total diferente da soma dos pagamentos aprovados: Liste o PedidoID e o ValorTotal para os pedidos onde a soma dos ValorPago dos pagamentos com StatusPagamento 'Aprovado' associados a este pedido é diferente do ValorTotal do pedido. Use JOINS com Pagamentos, WHERE para filtrar pagamentos aprovados, GROUP BY pelo pedido e HAVING para comparar a soma dos pagamentos com o ValorTotal do pedido.

```
SELECT P.PedidoID, P.ValorTotal  
FROM Pedidos P  
INNER JOIN Pagamentos PG ON P.PedidoID = PG.PedidoID  
WHERE PG.StatusPagamento = 'Aprovado'  
GROUP BY P.PedidoID, P.ValorTotal  
HAVING SUM(PG.ValorPago) != P.ValorTotal;
```

14. Crie uma View chamada PedidosProcessandoView que selecione todas as colunas da tabela Pedidos apenas para os pedidos cujo StatusPedido seja 'Processando'.

```
CREATE VIEW PedidosProcessandoView AS
```

```
SELECT *
```

```
FROM Pedidos
```

```
WHERE StatusPedido = 'Processando';
```

```
SELECT * FROM PedidosProcessandoView;
```

15. Utilize a View PedidosProcessandoView criada na pergunta anterior.

```
SELECT C.Nome, C.Sobrenome
```

```
FROM PedidosProcessandoView PPV
```

```
INNER JOIN Clientes C ON PPV.ClienteID = C.ClienteID;
```