

TESTE TÉCNICO DE CONHECIMENTOS EM SQL

Getulio Turelli de Melo

Introdução.....	1
Instruções.....	1
Desafios.....	2
1º Desafio.....	2
2º Desafio.....	2
3º Desafio.....	3
Resposta:.....	3
4º Desafio.....	4
Resposta:.....	5
5º Desafio.....	6
Resposta:.....	6
Requisitos.....	7
Resposta usando o compilador.....	7

Introdução

Bem-vindo ao teste técnico de conhecimentos em SQL!

A seguir, você encontrará cinco desafios onde precisará escrever consultas SQL.

Utilize apenas as tabelas e dados presentes no arquivo de SQLs fornecido.

Instruções

1. **Desafios:** Resolva cada um dos desafios propostos escrevendo as queries SQL apropriadas.
2. **Retorno Esperado:** Certifique-se de que as consultas retornem os resultados esperados, incluindo os nomes das colunas especificados.
3. **Validação:** Após finalizar todas as consultas, valide suas respostas no compilador online de PostgreSQL: [SQL Fiddle](#).
4. **Entrega:** Submeta suas consultas finais em um repositório próprio seu!

Desafios

1º Desafio

Consulta de Funcionários: Escreva uma query para listar todos os **funcionários** ativos, mostrando as colunas id, nome, salario. Ordene o resultado pelo nome em ordem ascendente.

Retorno esperado:

id	nome	salario
2	Vendedor B	4000.00
4	Vendedor D	3800.00
1	Vendedor Z	3000.00

Resposta: Não encontrada a tabela funcionários nos arquivos dos testes.

2º Desafio

Funcionários com Salário Acima da Média: Escreva uma query para listar os **funcionários** que possuem um salário acima da média salarial de todos os funcionários. A consulta deve mostrar as colunas id, nome, e salario, ordenadas pelo salario em ordem descendente.

Retorno esperado:

id	nome	salario
5	Vendedor E	4200.00
2	Vendedor B	4000.00
4	Vendedor D	3800.00

Resposta: Não encontrada a tabela funcionários nos arquivos dos testes.

3º Desafio

Resumo por cliente: Escreva uma query para listar todos os **clientes** e o valor total de pedidos já transmitidos. A consulta deve retornar as colunas id, razao_social, total, ordenadas pelo total em ordem descendente.

Retorno esperado:

id	razao_social	total
4	Cliente D	530
3	Cliente C	430
2	Cliente B	350
1	Cliente A	250
5	Cliente E	0

Resposta:

```
1 SELECT c.id_cliente AS id, c.razao_social,
2      COALESCE(SUM(p.valor_total), 0) AS total
3 FROM CLIENTES c
4 LEFT JOIN PEDIDO p ON c.id_cliente = p.id_cliente AND p.data_faturamento IS NOT NULL
5 GROUP BY c.id_cliente, c.razao_social
6 ORDER BY total DESC;
7
```

#	id	razao_social	total
1	4	Cliente D	530,0
2	2	Cliente B	350,0
3	3	Cliente C	220,0
4	1	Cliente A	0,0
5	5	Cliente E	0,0

```
SELECT c.id_cliente AS id, c.razao_social,
      COALESCE(SUM(p.valor_total), 0) AS total
FROM CLIENTES c
LEFT JOIN PEDIDO p ON c.id_cliente = p.id_cliente AND p.data_faturamento IS NOT NULL
GROUP BY c.id_cliente, c.razao_social
ORDER BY total DESC;
```

4º Desafio

Situação por pedido: Escreva uma query que retorne a situação atual de cada **pedido** da base. A consulta deve retornar as colunas id, valor, data e situação. A situação deve obedecer a seguinte regra:

- Se possui data de cancelamento preenchido: **CANCELADO**
- Se possui data de faturamento preenchido: **FATURADO**
- Caso não possua data de cancelamento e nem faturamento: **PENDENTE**

Retorno esperado:

id	valor	data	situacao
1	120.00	2023-07-06	PENDENTE
2	130.00	2023-07-07	PENDENTE
3	170.00	2023-07-08	FATURADO
4	180.00	2023-07-09	CANCELADO
5	210.00	2023-07-10	PENDENTE
6	220.00	2023-07-11	FATURADO
7	260.00	2023-07-12	CANCELADO
8	270.00	2023-07-13	FATURADO

Resposta:

```
1 SELECT p.id_pedido AS id, p.valor_total AS valor, p.data_emissao AS data,  
2     CASE  
3         WHEN p.data_cancelamento IS NOT NULL THEN 'CANCELADO'  
4         WHEN p.data_faturamento IS NOT NULL THEN 'FATURADO'  
5         ELSE 'PENDENTE'  
6     END AS situacao  
7 FROM PEDIDO p;  
8
```

PEDIDO (8r x 4c)					
#	id		valor	data	situacao
1	1		120,0	2023-07-06	PENDENTE
2	2		130,0	2023-07-07	PENDENTE
3	3		170,0	2023-07-08	FATURADO
4	4		180,0	2023-07-09	CANCELADO
5	5		210,0	2023-07-10	PENDENTE
6	6		220,0	2023-07-11	FATURADO
7	7		260,0	2023-07-12	CANCELADO
8	8		270,0	2023-07-13	FATURADO

```
SELECT p.id_pedido AS id, p.valor_total AS valor, p.data_emissao AS data,  
    CASE  
        WHEN p.data_cancelamento IS NOT NULL THEN 'CANCELADO'  
        WHEN p.data_faturamento IS NOT NULL THEN 'FATURADO'  
        ELSE 'PENDENTE'  
    END AS situacao  
FROM PEDIDO p;
```

5º Desafio

Produtos mais vendidos: Escreva uma query que retorne o **produto** mais vendido (em quantidade), incluindo o valor total vendido deste produto, quantidade de pedidos em que ele apareceu e para quantos clientes diferentes ele foi vendido. A consulta deve retornar as colunas id_produto, quantidade_vendida, total_vendido, clientes, pedidos. Caso haja empate em quantidade de vendas, utilizar o total vendido como critério de desempate.

Retorno esperado:

id_produto	quantidade_vendida	total_vendido	pedidos	clientes
2	12	220	3	2

Resposta:

```
1 SELECT ip.id_produto, SUM(ip.quantidade) AS quantidade_vendida,
2       SUM(ip.preco_praticado * ip.quantidade) AS total_vendido,
3       COUNT(DISTINCT p.id_pedido) AS pedidos,
4       COUNT(DISTINCT o.id_cliente) AS clientes
5 FROM ITENS_PEDIDO ip
6 JOIN PEDIDO p ON ip.id_pedido = p.id_pedido
7 JOIN CLIENTES o ON p.id_cliente = o.id_cliente
8 GROUP BY ip.id_produto
9 ORDER BY quantidade_vendida DESC, total_vendido DESC
10 LIMIT 1;
```

ITENS_PEDIDO (1r x 5c)					
#	id_produto	quantidade_vendida	total_vendido	pedidos	clientes
1	2	12	220,0	3	2

```
SELECT ip.id_produto, SUM(ip.quantidade) AS quantidade_vendida,
       SUM(ip.preco_praticado * ip.quantidade) AS total_vendido,
       COUNT(DISTINCT p.id_pedido) AS pedidos,
       COUNT(DISTINCT o.id_cliente) AS clientes
FROM ITENS_PEDIDO ip
JOIN PEDIDO p ON ip.id_pedido = p.id_pedido
JOIN CLIENTES o ON p.id_cliente = o.id_cliente
GROUP BY ip.id_produto
ORDER BY quantidade_vendida DESC, total_vendido DESC
LIMIT 1;
```

Requisitos

- Utilize apenas as tabelas e dados presentes no arquivo **schema.sql**.
- Certifique-se de que os nomes das colunas no resultado sejam exatamente os especificados e retornados nos desafios.
- Valide suas respostas utilizando o compilador online [SQL Fiddle](#).

Boa sorte!

Resposta usando o compilador

Abaixo o link com a resposta do teste usando o compilador online SQL Fiddle
Resposta do compilador SQL Fiddle