

O que é uma View no MySQL?

Uma **view** é uma **tabela virtual** baseada em uma consulta SQL. Ela **não armazena os dados fisicamente**, mas permite reutilizar consultas complexas de forma simples e organizada.

Benefícios das Views

- Facilita a leitura de dados complexos.
 - Esconde detalhes sensíveis ou irrelevantes.
 - Permite reuso de lógica SQL.
 - Pode servir como camada de segurança e organização.
-

Exemplos com as tabelas **clientes** e **pedidos**

Criando uma View simples com JOIN

sql

CopiarEditar

```
CREATE VIEW vw_clientes_pedidos AS
SELECT c.id AS cliente_id, c.nome, p.id AS pedido_id, p.valor_total
FROM clientes c
INNER JOIN pedidos p ON p.cliente_id = c.id;
```

Consultando a View

sql

CopiarEditar

```
SELECT * FROM vw_clientes_pedidos;
```

View com filtro

sql

CopiarEditar

```
CREATE VIEW vw_clientes_recente AS
SELECT nome, email, data_cadastro
FROM clientes
WHERE data_cadastro >= '2025-01-01';
```

✖ Remover uma View

sql

CopiarEditar

```
DROP VIEW vw_clientes_recente;
```



Lista de Exercícios sobre Views no MySQL



Básico

1. Crie uma view chamada **vw_emails_clientes** que retorne apenas os campos **nome** e **email** da tabela **clientes**.
2. Crie uma view chamada **vw_pedidos_acima_500** que mostre os pedidos cujo **valor_total** seja maior que 500.
3. Use a view **vw_clientes_recente** (criada acima) para listar apenas os nomes dos clientes cadastrados a partir de 2025.



Intermediário

4. Crie uma view chamada **vw_relatorio_completo** com as seguintes colunas:
 - nome do cliente
 - email
 - id do pedido
 - valor total do pedido
5. Use **INNER JOIN** entre **clientes** e **pedidos**.
6. Atualize o nome da view **vw_pedidos_acima_500** para **vw_pedidos_vip**.

Dica: no MySQL, você precisa dropar e recriar a view com o novo nome.

Liste todas as views existentes no banco atual.

Dica: use:

```
sql
CopiarEditar
SHOW FULL TABLES WHERE Table_type = 'VIEW';
```

7.

Avançado

7. **Crie uma view chamada `vw_estatisticas_pedidos` que exiba:**

- total de pedidos
- valor médio dos pedidos
- valor máximo e mínimo

8. Utilize as funções agregadas `COUNT`, `AVG`, `MAX`, `MIN`.

9. **Crie uma view que mostre o total de pedidos por cliente.**

Use `GROUP BY` com `cliente_id`.

10. **Tente atualizar o nome do cliente através da view `vw_emails_clientes`. O que acontece? Por quê?**

Escreva sua resposta com base no erro que o MySQL retornar.