



**SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL**

**SENAI “GASPAR RICARDO JUNIOR”**

**Curso**

**TÉCNICO EM DESENVOLVIMENTO  
DE SISTEMAS**

**Pesquisa SQL Views**

Ana Carolina Castro Ribeiro

Sorocaba  
Novembro – 2024



**SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL**

**SENAI “GASPAR RICARDO JUNIOR”**

Ana Carolina Castro Ribeiro

## **Pesquisa SQL Views**

Pesquisa e levantamento de  
Conhecimento (SQL Views)

Prof. – Emerson Magalhães, Daniel Cintori

Sorocaba  
Novembro – 2024

# SUMÁRIO

INTRODUÇÃO.....	5
1. Fundamentos Teóricos.....	6
1.1. O Que São SQL Views e suas Funções .....	6
1.2. Diferença entre Views e Tabelas Comuns .....	6
1.3. Tipos de Views .....	6
2. Vantagens e Desvantagens de Usar Views.....	7
3. Processo de Criação de Views no SQL .....	7
4. Views Atualizáveis e Não Atualizáveis.....	8
5. Estudo de Caso .....	9
CONCLUSÃO.....	11
BIBLIOGRAFIA .....	12



# SQL Views

## INTRODUÇÃO

Inicialmente, SQL Views são consultas armazenadas em um banco de dados que agem como uma tabela virtual. Elas não armazenam dados por conta própria, mas sim instruções SQL que permitem extrair informações de uma ou mais tabelas. Assim, sua principal função é simplificar consultas de dados complexos, facilitando o acesso e uso das informações.

Elas são de grande importância em sistemas de banco de dados relacionais pois contribuem para a segurança, simplificação de operações e facilitação da manutenção. Dessa forma é promovido a eficiência e clareza na interação com o banco de dados.

Em vista disso, essa pesquisa tem como objetivo promover o aprendizado acerca das SQL Views, com sua definição, importância, tipos, vantagens e desvantagens e exemplos práticos em que podem ser aplicados.

# **1. Fundamentos Teóricos**

## **1.1. O Que São SQL Views e suas Funções**

SQL Views são consultas armazenadas em um banco de dados que agem como uma tabela virtual. Os dados não vão ser armazenados diretamente nelas, mas sim a lógica da consulta que os gera. Assim, uma view é como uma "janela" para dados que são atualizados em tempo real a partir das tabelas subjacentes.

Dessa forma, sempre que uma view é acessada, a consulta é executada e os dados são apresentados de forma dinâmica. Sendo estas úteis para facilitar o acesso a consultas complexas e personalizar a apresentação dos dados em um banco de dados relacional.

## **1.2. Diferença entre Views e Tabelas Comuns**

Basicamente as views se diferenciam das tabelas por não armazenarem dados permanentemente. Então elas são executadas em tempo real sempre que chamadas, enquanto que as tabelas possuem dados persistentes.

## **1.3. Tipos de Views**

- Views simples: Criadas a partir de consultas básicas e tendo dados contidos em uma única tabela;
- Views complexas: Resultantes de consultas com junções, agregações e etc;
- Views materializadas: Elas armazenam resultados de consultas, permitindo acesso mais rápido a dados que não mudam com frequência.

## 2. Vantagens e Desvantagens de Usar Views

### 2.1. Vantagens:

- Simplificação de consultas complexas, uma vez que ajuda a manter a clareza e organização ao acessar dados;
- Segurança, pois permite o controle de acesso, mostrando apenas dados específicos para certos usuários;
- Manutenção simplificada, já que facilita o gerenciamento de consultas frequentemente usadas, centralizando a lógica.

### 2.2. Desvantagens:

- Desempenho, sendo que, dependendo da complexidade, pode impactar o desempenho;
- Atualização, com algumas views podendo não permitir atualizações diretas;
- Complexidade de gerenciamento, uma vez que o uso excessivo de views interligadas pode complicar a administração do banco de dados.

## 3. Processo de Criação de Views no SQL

Com a instrução “CREATE VIEW” é permitido criar uma view com uma estrutura como:

```
CREATE VIEW vendas_simplificadas AS
SELECT produto_id, SUM(quantidade) AS total_vendido
FROM vendas
GROUP BY produto_id;
```

FIGURA 1- Criação de View

Nesse exemplo é criado uma view que contabilize a quantidade vendida de produtos.

```
CREATE VIEW vendas_brasil AS
SELECT *
FROM vendas
WHERE pais = 'Brasil';
```

FIGURA 2- Criação de View Simples

```
CREATE VIEW resumo_vendas AS
SELECT c.nome_cliente, SUM(v.valor) AS total_gasto
FROM clientes c
JOIN vendas v ON c.id_cliente = v.cliente_id
WHERE v.data_venda BETWEEN '2024-01-01' AND '2024-12-31'
GROUP BY c.nome_cliente
HAVING SUM(v.valor) > 5000;
```

FIGURA 3- Criação de View Complexa

## 4. Views Atualizáveis e Não Atualizáveis

As views podem permitir a atualização de dados diretamente, mas isso depende de algumas condições e da complexidade da view.

Uma view é atualizável quando permite operações, sendo elas: “INSERT”, “UPDATE” e “DELETE”. Entretanto, nem todas as views são atualizáveis; em alguns casos, as operações precisam ser realizadas diretamente nas tabelas originais.

A possibilidade de atualização das views depende de fatores como:

A view deve se basear em uma única tabela;

A tabela de origem deve ter uma chave primária, que também deve estar incluída na view;

Não pode conter funções agregadas;

Não deve incluir “DISTINCT”, “GROUP BY”, ou subconsultas complexas.

Já as views que combinam múltiplas tabelas ou usam funções de agregação não são atualizáveis diretamente.

Exemplo de view atualizável:

```
CREATE VIEW clientes_vips AS
SELECT id_cliente, nome_cliente, email
FROM clientes
WHERE status_vip = 'Sim';
```

FIGURA 4- View atualizável

Exemplo de view não atualizável:

```
CREATE VIEW vendas_resumo AS
SELECT c.nome_cliente, SUM(v.valor_total) AS total_gasto
FROM clientes c
JOIN vendas v ON c.id_cliente = v.cliente_id
GROUP BY c.nome_cliente;
```

FIGURA 5- View não atualizável

## 5. Estudo de Caso

Entende-se que as views são recomendadas em situações em que é necessário a: padronização de consultas, mantendo a consistência em consultas complexas e repetidas; segurança de dados, fornecendo acesso controlado a informações específicas; e separação de camadas lógicas, permitindo que desenvolvedores e analistas de dados trabalhem com abstrações sem acesso direto às tabelas subjacentes.

Como exemplo, em um banco de dados de um e-commerce (comércio eletrônico), as views podem ser usadas para gerar relatórios de vendas e consultas de estoque. Simplificando a análise de dados e melhorando a segurança ao expor somente as informações necessárias aos usuários finais.

```
CREATE VIEW relatorio_vendas_cliente AS
SELECT c.nome_cliente, COUNT(v.id_venda) AS num_compras, SUM(v.valor_total) AS total_gasto
FROM clientes c
JOIN vendas v ON c.id_cliente = v.cliente_id
GROUP BY c.nome_cliente;
```

FIGURA 6- View de monitoramento de vendas

Essa é uma view que ajuda a monitorar o desempenho de vendas, agrupando as transações por cliente.



```
CREATE VIEW estoque_baixo AS
SELECT produto_id, nome_produto, quantidade
FROM produtos
WHERE quantidade < 10;
```

FIGURA 7- View de visualização de produtos

Aqui encontra-se uma view é possível visualizar apenas os produtos com quantidade limitada em estoque.

```
CREATE VIEW pagamentos_pendentes AS
SELECT v.id_venda, c.nome_cliente, p.valor, p.status_pagamento
FROM pagamentos p
JOIN vendas v ON p.venda_id = v.id_venda
JOIN clientes c ON v.cliente_id = c.id_cliente
WHERE p.status_pagamento = 'Pendente';
```

FIGURA 8- View de relatório de pagamento pendente

Já essa view é a de relatório de pagamentos pendentes

## CONCLUSÃO

Com isso, conclui-se que as views são ferramentas de grande utilidade para estruturar e simplificar o acesso de dados em sistemas SQL. Observando o que são, suas funcionalidades, tipos, diferenças com tabelas, vantagens e desvantagens, bem como exemplos práticos aplicados a contextos específicos, é possível um melhor aprendizado sobre o tema.

Assim a importância das views se dá pelo aumento de segurança que elas fornecem, restringindo o acesso, pela simplificação do sistema e a melhora na manutenção. Entretanto, isso é alcançado de forma plena por meio de um uso consciente, com práticas como: nomeação clara; documentação; simplificação de consultas; atualizações; otimização; revisão regular; e etc.

Com esse conhecimento será possível o uso das views em ambientes que necessitam de sua aplicação para a melhora de gerenciamento e armazenamento de dados, como observado no exemplo do e-commerce, em que o uso das views possibilitaria o controle dos produtos. Dessa forma, termina-se o levantamento de conhecimento acerca das SQL views.

## BIBLIOGRAFIA

- <https://ramosdainformatica.com.br/views-em-sql-vantagens-e-desvantagens/>
- <https://dfilitto.blog.br/desenvolvimento/views-em-banco-de-dados-desvendando-as-tabelas-virtuais/>
- <https://www.devmedia.com.br/conceitos-e-criacao-de-views-no-sql-server/22390>
- <https://www.hashtagtreinamentos.com/diferenca-tabelas-e-views-no-sql>
- <https://www.youtube.com/watch?app=desktop&v=eb71ZN5HX5M>