



SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL

SENAI “GASPAR RICARDO JUNIOR”

Curso

**TÉCNICO EM DESENVOLVIMENTO
DE SISTEMAS**

**SQL Views - Conceito, Benefícios e
Aplicações Práticas**

Nathalia Almeida Yoshioka

Sorocaba
Novembro – 2024



SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL

SENAI “GASPAR RICARDO JUNIOR”

Nathalia Almeida Yoshioka

SQL Views: Conceito, Benefícios e Aplicações Práticas

Este trabalho é uma introdução
ao SQL Views para facilitar o
aprendizado do conteúdo
Prof. – Emerson Magalhães

Sorocaba
Novembro – 2024

HISTÓRICO DE VERSÕES

[illegible]

SUMÁRIO

INTRODUÇÃO.....	4
1. FUNDAMENTOS TEÓRICOS DAS SQL VIEWS	5
1.1. VANTAGENS E DESVANTAGENS DE USAR VIEWS	5
1.1.1. VANTAGENS.....	5
1.1.2. DESVANTAGENS	5
2. PROCESSO DE CRIAÇÃO DE VIEWS NO SQL	6
2.1. VIEWS ATUALIZÁVEIS E NÃO ATUALIZÁVEIS	7
3. ESTUDO DE CASO	7
CONCLUSÃO.....	9
BIBLIOGRAFIA	10
LISTA DE FIGURAS	12
LISTA DE TABELAS	12

SQL Views - Conceito, Benefícios e Aplicações Práticas

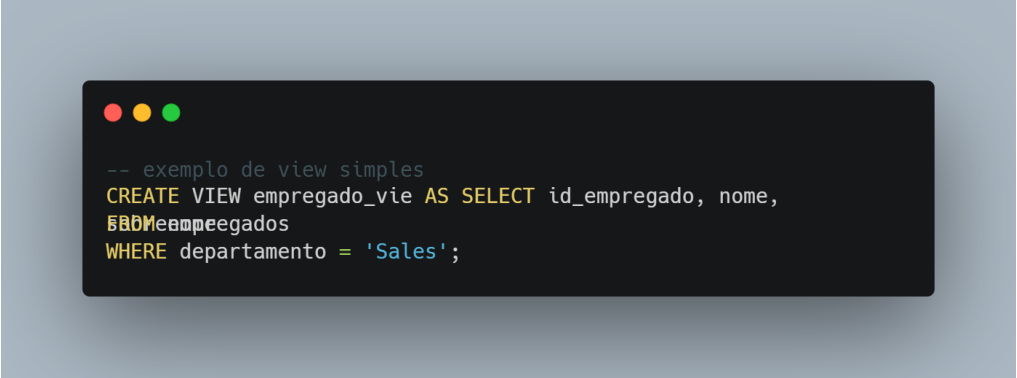
INTRODUÇÃO

SQL Views são consultas armazenadas no banco de dados que permitem a visualização de dados de uma ou mais tabelas como se fossem uma tabela única. Elas são importantes porque simplificam consultas complexas, melhoram a segurança ao restringir o acesso a dados sensíveis e facilitam a manutenção de consultas frequentemente usadas. O objetivo desta pesquisa é explicar a definição, importância, vantagens e desvantagens das SQL Views, bem como fornecer exemplos práticos de sua criação e utilização.

1. FUNDAMENTOS TEÓRICOS DAS SQL VIEWS

As SQL Views são representações virtuais de uma ou mais tabelas no banco de dados. Elas permitem que os usuários consultem os dados como se estivessem acessando uma tabela única. Diferente das tabelas comuns, que armazenam dados, as views armazenam consultas SQL que geram uma visão dos dados.

Existem diferentes tipos de views, como as views simples, que são baseadas em uma única tabela; as views complexas, que envolvem junções e agregações de múltiplas tabelas; e as views materializadas, que armazenam os dados da consulta em si e são atualizadas periodicamente.



```
-- exemplo de view simples
CREATE VIEW empregado_vie AS SELECT id_empregado, nome,
FROM empregados
WHERE departamento = 'Sales';
```

1.1. VANTAGENS E DESVANTAGENS DE USAR VIEWS

1.1.1. VANTAGENS

As vantagens de usar views incluem a simplificação de consultas complexas, o aumento da segurança ao restringir o acesso a certas colunas e linhas, e a facilitação do controle e da manutenção de consultas frequentemente usadas.

1.1.2. DESVANTAGENS

Já as desvantagens de usar views incluem possíveis impactos de desempenho, limitações em views complexas para operações de atualização, e a necessidade de manutenção das views materializadas para a atualização periódica dos dados.

2. PROCESSO DE CRIAÇÃO DE VIEWS NO SQL

Para criar uma view no SQL, utiliza-se a instrução **CREATE VIEW**. A sintaxe básica é “**CREATE VIEW view_name AS SELECT column1, column2, ... FROM table_name WHERE condition;**”. Por exemplo, uma view de filtragem pode selecionar colunas e linhas específicas de uma tabela; uma view de agregação pode usar funções como SUM (soma), AVG (média) e COUNT (contagem) para resumir os dados; e uma view de junção pode combinar dados de várias tabelas para gerar uma nova visão.

```
CREATE VIEW view_name AS SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```

```
-- exemplo de view simples  
CREATE VIEW sales_summary AS  
SELECT department, SUM(sales) AS total_sales  
FROM sales  
GROUP BY department;
```

```
-- exemplo de view de junção  
CREATE VIEW employee_department AS  
SELECT e.employee_id, e.first_name, e.last_name, d.department_name  
FROM employees e  
JOIN departments d ON e.department_id = d.department_id;
```

2.1. VIEWS ATUALIZÁVEIS E NÃO ATUALIZÁVEIS

Algumas views permitem que os dados subjacentes sejam atualizados diretamente. Para que uma view seja atualizável, ela deve ser baseada em uma única tabela e não deve conter funções agregadas, junções, subconsultas ou cláusulas ***DISTINCT***. Por exemplo, uma view simples que seleciona colunas de uma única tabela pode ser atualizável, enquanto uma view complexa que combina dados de várias tabelas não seria atualizável.

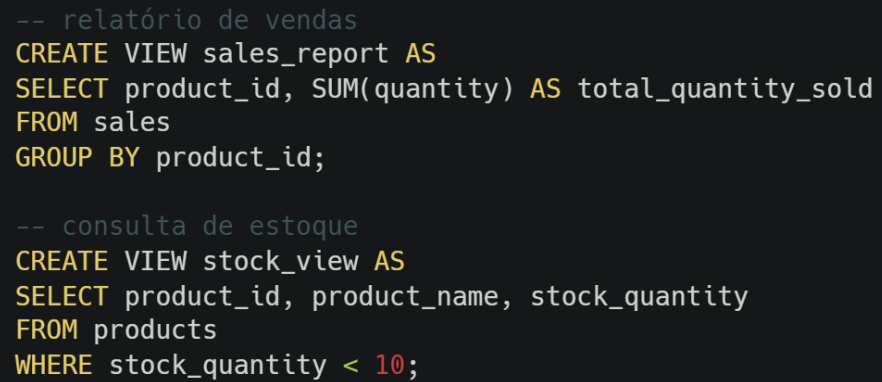
```
-- exemplo de view não atualizável
CREATE VIEW complex_view AS
SELECT e.employee_id, e.first_name, e.last_name, d.department_name
FROM employees e
JOIN departments d ON e.department_id = d.department_id;
```

```
-- exemplo de view atualizável
CREATE VIEW simple_view AS
SELECT employee_id, first_name, last_name
FROM employees;
```

3. ESTUDO DE CASO

Considerando um banco de dados de uma loja de e-commerce, podemos criar views para relatórios de vendas ou consulta de estoque. Por exemplo, uma view para relatório de vendas pode sumarizar a quantidade de produtos vendidos por categoria, enquanto uma view para consulta de estoque pode listar os produtos com baixa quantidade em estoque. Essas views facilitam a geração de

relatórios e a manutenção de dados críticos, melhorando a eficiência e a segurança das operações.

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. It contains two SQL queries. The first query creates a view named 'sales_report' by selecting 'product_id' and the sum of 'quantity' from the 'sales' table, grouped by 'product_id'. The second query creates a view named 'stock_view' by selecting 'product_id', 'product_name', and 'stock_quantity' from the 'products' table where 'stock_quantity' is less than 10.

```
-- relatório de vendas
CREATE VIEW sales_report AS
SELECT product_id, SUM(quantity) AS total_quantity_sold
FROM sales
GROUP BY product_id;

-- consulta de estoque
CREATE VIEW stock_view AS
SELECT product_id, product_name, stock_quantity
FROM products
WHERE stock_quantity < 10;
```

CONCLUSÃO

As SQL Views são ferramentas poderosas para simplificação de consultas, aumento de segurança e manutenção de dados. O uso adequado de views pode otimizar o desempenho e a gestão de bancos de dados relacionais. É importante seguir boas práticas na criação e manutenção de views, como evitar views excessivamente complexas e atualizar regularmente views materializadas.

BIBLIOGRAFIA

W3SCHOOLS. SQL CREATE VIEW, REPLACE VIEW, DROP VIEW Statements.

Disponível em: https://www.w3schools.com/SQL/sql_view.asp. Acesso em: 12 nov. 2024.

WILLIAMDASSAFMSFT. Views - SQL Server. Disponível em:

<https://learn.microsoft.com/en-us/sql/relational-databases/views/views?view=sql-server-ver16>. Acesso em: 12 nov. 2024.

Diferença de Tabelas e Views no SQL - Entenda de Uma Vez! Disponível em:

https://www.hashtagtreinamentos.com/diferenca-tabelas-e-views-no-sql?gad_source=1&gclid=CjwKCAiAudG5BhAREiwAWMISjP7Vkgkf5Vi00o256bXLsUhXZB1R7gOBmDJvhXzulgyjrL1MHthNlXoCcdYQAvD_BwE>. Acesso em: 13 nov. 2024.

Conceitos e criação de views no SQL Server. Disponível em:

<https://www.devmedia.com.br/conceitos-e-criacao-de-views-no-sql-server/22390>. Acesso em: 13 nov. 2024.

LISTA DE FIGURAS

Figura 1 – NORMAS ABNT, 2024. Disponível em: <https://www1.folha.uol.com.br>. Acesso em: 08 jun. 2020

LISTA DE TABELAS

TABELA 1 – VALOR UNITÁRIO – Calculo de Valores realizados na pesquisa