



SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL

SENAI “GASPAR RICARDO JUNIOR”

**CURSO TÉCNICO EM DESENVOLVIMENTO DE
SISTEMAS**

SQL Views

Luany Urtado Santos

Sorocaba
Novembro – 2024



SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL

SENAI “GASPAR RICARDO JUNIOR”

Luany Urtado Santos

SQL Views

Breve pesquisa sobre os conceitos, benefícios e aplicações
práticas do SQL Views

Prof. – Emerson Magalhães

Sorocaba
Novembro – 2024

INTRODUÇÃO

As SQL Views são objetos de banco de dados que apresentam uma consulta como se fossem uma tabela. Elas permitem que os usuários acessem e manipulem dados de forma simplificada. Elas são formadas por declarações `SELECTs`, que retornam uma visualização de dados específica de uma ou mais tabelas de um banco de dados.

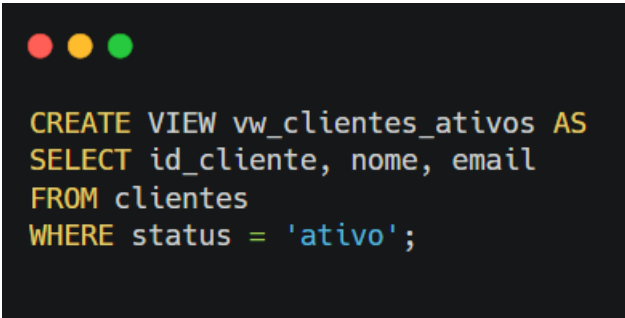
Elas são essenciais para que seja possível abstrair a complexidade do banco de dados, proporcionando uma interface mais amigável para os usuários e facilitando a segurança e o gerenciamento dos dados.

FUNDAMENTOS TEÓRICOS

Views são consultas salvas que podem ser tratadas como tabelas em comandos SQL subsequentes. Elas fornecem uma forma de organizar e simplificar consultas complexas.

Diferença entre Views e Tabelas: Diferente das tabelas, as views não armazenam dados diretamente, elas são uma representação dinâmica dos dados que residem nas tabelas.

Views simples: Baseadas em uma única tabela, com filtrações ou seleções de colunas.

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. It displays SQL code to create a view named 'vw_clientes_ativos' from a table named 'clientes', selecting columns 'id_cliente', 'nome', and 'email' where the status is 'ativo'.

```
CREATE VIEW vw_clientes_ativos AS
SELECT id_cliente, nome, email
FROM clientes
WHERE status = 'ativo';
```

Views complexas: Envolvem junções de múltiplas tabelas ou agregações.

```
CREATE VIEW vw_vendas_por_cliente AS
SELECT c.id_cliente, c.nome, COUNT(v.id_venda) AS total_vendas, SUM(v.valor_total) AS total_gasto
FROM clientes c
JOIN vendas v ON c.id_cliente = v.id_cliente
GROUP BY c.id_cliente, c.nome
HAVING SUM(v.valor_total) > 1000;
```

Views materializadas: Armazenam fisicamente os dados resultantes da consulta, o que pode melhorar o desempenho, mas exigem atualizações periódicas.

```
CREATE MATERIALIZED VIEW mv_vendas_resumo AS
SELECT id_cliente, SUM(valor_total) AS total_vendas
FROM vendas
GROUP BY id_cliente;
```

VANTAGENS E DESVANTAGENS NO USO

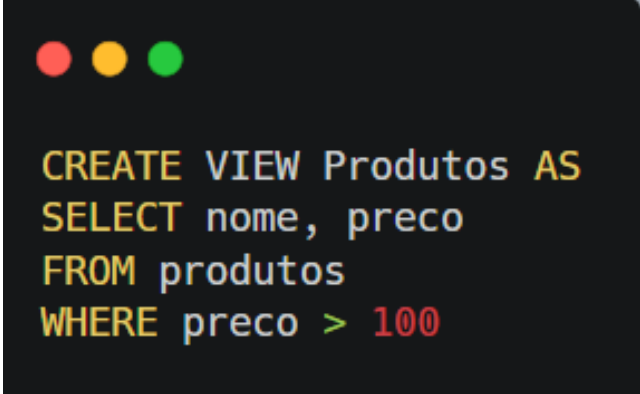
Vantagens: Simplificação de consultas complexas, aumento da segurança ao restringir acesso a certas colunas e linhas, facilita o controle e a manutenção de consultas frequentemente usadas.

Desvantagens: Possíveis impactos no desempenho, especialmente com views complexas, limitações em operações de atualização em views complexas, necessidade de manutenção para views materializadas.

PROCESSO DE CRIAÇÃO

Para a criação de uma view, é necessário iniciar o comando com o CREATE VIEW. Antes de tudo é preciso criar uma consulta na qual você

quer que ela retorne. O CREATE VIEW deve seguir pelo nome da view e da consulta SQL, como é possível ver no exemplo abaixo.

A terminal window with a dark background and three colored window control buttons (red, yellow, green) at the top left. It displays the following SQL code:

```
CREATE VIEW Produtos AS
SELECT nome, preco
FROM produtos
WHERE preco > 100
```

ATUALIZÁVEIS E NÃO ATUALIZÁVEIS

Views atualizáveis são aquelas que aceitam comandos DML (DELETE/INSERT/UPDATE). Já as não atualizáveis são aquelas que não aceitam.

Views Atualizáveis: Podem ser modificadas diretamente, desde que atendam a certos critérios, como não envolverem funções agregadas ou junções complexas.

Views Não Atualizáveis: Quando uma view é baseada em agregações ou junções complexas, não é possível atualizar os dados diretamente por meio dela.

ESTUDO DE CASO

```
DROP DATABASE db_farmacia;

CREATE DATABASE db_farmacia;
USE db_farmacia;

CREATE TABLE medicamentos (
  id_medicamento INT AUTO_INCREMENT PRIMARY KEY,
  nome VARCHAR(100) NOT NULL,
  preco DECIMAL(10, 2),
  estoque INT
);

INSERT INTO medicamentos (id_medicamento, nome, preco, estoque) VALUES
(1, 'Amoxicilina', 26, 98),
(2, 'Escitalopram', 15, 102),
(3, 'Aзитromicina', 23, 45),
(4, 'Alprazolam', 12, 18);

SELECT * FROM medicamentos;

CREATE TABLE vendas (
  id_venda INT AUTO_INCREMENT PRIMARY KEY,
  id_medicamento INT,
  quantidade INT,
  data_venda DATE,
  total_venda DECIMAL(10, 2),
  FOREIGN KEY (id_medicamento) REFERENCES medicamentos(id_medicamento)
);

INSERT INTO vendas (id_venda, id_medicamento, quantidade, data_venda, total_venda) VALUES
(5, 1, 13, '2024-10-08', 78.19),
(6, 2, 25, '2024-04-09', 129.99),
(7, 3, 99, '2024-07-23', 346.78),
(8, 4, 203, '2024-06-06', 200.5);

SELECT * FROM vendas;

-- View que exibi o relatório de vendas com o nome do medicamento --
CREATE VIEW relatorio_vendas AS
SELECT v.id_venda, m.nome AS nome_medicamento, v.quantidade, v.data_venda, v.total_venda
FROM vendas v
JOIN medicamentos m ON v.id_medicamento = m.id_medicamento;

select * from relatorio_vendas;
```

CONCLUSÃO

SQL Views é ferramenta para organizar, simplificar e proteger consultas em bancos de dados. Elas permitem criar abstrações úteis sobre os dados, tornando o sistema mais eficiente e seguro. Ao criar views, é importante seguir boas práticas como manter a simplicidade, evitar views desnecessárias e garantir que a manutenção de views materializadas seja feita de forma eficiente.

REFERÊNCIAS BIBLIOGRÁFICAS

[https://sae.unb.br/cae/conteudo/unbfga/lbd/banco2_visoes.html#:~:text=Uma%20vis%C3%A3o%20materializada%20\(ou%20%22materialized,tempo%20de%20resposta%20mais%20r%C3%A1pido](https://sae.unb.br/cae/conteudo/unbfga/lbd/banco2_visoes.html#:~:text=Uma%20vis%C3%A3o%20materializada%20(ou%20%22materialized,tempo%20de%20resposta%20mais%20r%C3%A1pido)

<https://www.profissionaloracle.com.br/2023/07/29/guia-das-views-no-banco-de-dados-oracle-tipos-funcionalidades-e-exemplos-praticos/>

<https://pt.stackoverflow.com/questions/35413/o-que-s%C3%A3o-views-em-sql-quais-vantagens-e-desvantagens-em-utilizar>

<https://king.host/wiki/artigo/views-mysql/#:~:text=Uma%20view%20%C3%A9%20um%20objeto,do%20esquema%20f%C3%ADsico%20da%20base>