



SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL

SENAI “GASPAR RICARDO JUNIOR”

Curso

**TÉCNICO EM DESENVOLVIMENTO
DE SISTEMAS**

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

Ana Caroline Mena Bezerra de Paula

Sorocaba
Novembro – 2024



SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL

SENAI “GASPAR RICARDO JUNIOR”

Ana Caroline Mena Bezerra de Paula

SQL Views – Conceito, Benefícios e Aplicações Práticas

Entender o que são SQL Views e
seu processo de criação.

Prof. – Emerson Magalhães

Sorocaba
Novembro – 2024

SUMÁRIO

RESUMO	3
As <i>SQL Views</i> (ou Visões) são basicamente consultas salvas em um banco de dados. Elas permitem que você acesse dados de forma simplificada, sem precisar reescrever a consulta toda vez.	3
Em vez de mostrar as tabelas diretamente, as visões podem mostrar apenas os dados que você precisa, o que ajuda na segurança e na clareza das consultas. Também, como as visões são como "atalhos" para consultas complexas, elas tornam o trabalho com dados mais rápido e fácil, evitando cálculos repetidos.	3
Se você criar uma visão, pode usá-la como se fosse uma tabela normal, mas sem armazenar os dados fisicamente. Isso facilita a manutenção do banco de dados, pois mudanças nas tabelas podem ser feitas sem afetar as consultas que usam a visão.	3
OBJETIVO	5
O objetivo desta pesquisa é explorar os fundamentos teóricos das <i>SQL Views</i> , suas diferenças em relação às tabelas comuns, e os diferentes tipos de views disponíveis. A pesquisa abrange desde a definição básica até exemplos práticos de uso.	5
INTRODUÇÃO.....	6
1.1. Definição de <i>SQL Views</i> e sua função em bancos de dados.....	6
1.2. Importância das views em sistemas de banco de dados relacionais.....	6
Simplificam consultas complexas: Permitem que consultas complexas sejam encapsuladas em uma única view, facilitando o acesso aos dados.....	6
Melhoram a segurança: Podem restringir o acesso a colunas específicas de uma tabela, protegendo dados sensíveis.....	6
Facilitam a manutenção: Alterações na lógica de consulta podem ser feitas na view sem necessidade de modificar as aplicações que a utilizam.....	6
Uma das características principais do modelo é a possibilidade de relacionar várias tabelas, evitando a redundância no armazenamento de dados. O relacionamento nesse modelo pode ocorrer de três formas:	6
A) Um-para-um;	6
B) Um-para-muitos;.....	6
C) Muitos-para-muitos	6
2. FUNDAMENTOS TEÓRICOS DAS <i>SQL VIEWS</i>	7
2.1. O que são views e como elas funcionam no <i>SQL</i>	7
2.2. Diferença entre views e tabelas comuns.....	7
Views: Não armazenam dados fisicamente, apenas a consulta que gera os dados.	7
Tabelas comuns: Armazenam dados fisicamente no banco de dados.	7
2.3. Tipos de views	7
Simples: Baseadas em uma única tabela e sem funções agregadas. Sendo úteis para oferecer uma visão resumida ou segmentada dos dados originais.	7
Complexas: Envolvem junções entre múltiplas tabelas e podem incluir funções agregadas. São especialmente úteis para simplificar consultas complexas e fornecer uma visão abrangente dos dados.	8
Materializadas: Armazenam fisicamente os dados resultantes da consulta, melhorando o desempenho em consultas complexas, mas requerem atualização manual ou automática.	8

3.	VANTAGENS E DESVANTAGENS DE USAR VIEWS.....	9
3.1.	Vantagens	9
	Simplificação de consultas complexas: Views permitem encapsular consultas complexas, tornando-as mais fáceis de reutilizar e entender.....	9
	Aumento da Segurança: Ao criar views, você pode restringir o acesso a dados sensíveis, exibindo apenas as colunas e linhas necessárias para determinados usuários.....	9
	Facilidade de Manutenção: Views ajudam a centralizar e padronizar consultas frequentemente usadas, facilitando a manutenção e atualização.....	9
3.2.	Desvantagens:	9
	Impactos de Desempenho: Quando forem baseadas em consultas complexas ou se não forem otimizadas, podem introduzir sobrecarga de desempenho.....	9
	Limitações em Operações de Atualização: As que envolvem múltiplas tabelas ou agregações, podem não ser atualizáveis diretamente.	9
	Manutenção de Views Materializadas: Precisam ser atualizadas de tempos em tempos para refletir mudanças nos dados subjacentes, podendo adicionar complexidade e custo de manutenção.	9
4.	PROCESSO DE CRIAÇÃO DE VIEWS NO SQL	10
4.1.	Instrução CREATE VIEW: sintaxe e parâmetros	10
4.2.	Filtragem: seleção de colunas e linhas específicas	10
4.3.	Agregação: uso de funções como SUM, AVG, COUNT	10
4.4.	Junção: combinar dados de várias tabelas para gerar uma nova visão	11
5.	VIEWS ATUALIZÁVEIS E NÃO ATUALIZÁVEIS	12
5.1.	Possibilidade de atualizar dados diretamente em views.....	12
5.2.	Condições para que uma view seja atualizável	12
5.3.	Exemplo prático de uma view atualizável e outra não atualizável.....	12
6.	ESTUDO DO CASO	13
6.1.	Criação de um banco de dados fictício (exemplo: loja de e-commerce ou sistema de RH)	13
	Quando falamos em Views, existem 3 ações envolvidas:	
	1. Criação de uma View	
	2. Alteração de uma View	
	3. Exclusão de uma View	
	Para cada uma dessas ações, temos um comando associado:	
	1. CREATE VIEW	
	2. ALTER VIEW	
	3. DROP VIEW	
	13
	CONCLUSÃO.....	14
	BIBLIOGRAFIA	15
	LISTA DE FIGURAS	16



SQL Views – Conceito, Benefícios e Aplicações Práticas

RESUMO

As *SQL Views* (ou Visões) são basicamente consultas salvas em um banco de dados. Elas permitem que você acesse dados de forma simplificada, sem precisar reescrever a consulta toda vez.

Em vez de mostrar as tabelas diretamente, as visões podem mostrar apenas os dados que você precisa, o que ajuda na segurança e na clareza das consultas. Também, como as visões são como "atalhos" para consultas complexas, elas tornam o trabalho com dados mais rápido e fácil, evitando cálculos repetidos.

Se você criar uma visão, pode usá-la como se fosse uma tabela normal, mas sem armazenar os dados fisicamente. Isso facilita a manutenção do banco de dados, pois mudanças nas tabelas podem ser feitas sem afetar as consultas que usam a visão.

OBJETIVO

O objetivo desta pesquisa é explorar os fundamentos teóricos das SQL Views, suas diferenças em relação às tabelas comuns, e os diferentes tipos de views disponíveis. A pesquisa abrange desde a definição básica até exemplos práticos de uso.

INTRODUÇÃO

1.1. Definição de SQL Views e sua função em bancos de dados

Uma visão (ou "view"), é uma consulta (instrução SQL) armazenada no banco de dados geralmente chamada de "tabela virtual". São usadas para simplificar consultas complexas, melhorar a segurança ao restringir o acesso a dados sensíveis e facilitar a reutilização de consultas frequentes



Imagem1

1.2. Importância das views em sistemas de banco de dados relacionais

Simplificam consultas complexas: Permitem que consultas complexas sejam encapsuladas em uma única view, facilitando o acesso aos dados.

Melhoram a segurança: Podem restringir o acesso a colunas específicas de uma tabela, protegendo dados sensíveis.

Facilitam a manutenção: Alterações na lógica de consulta podem ser feitas na view sem necessidade de modificar as aplicações que a utilizam.

Uma das características principais do modelo é a possibilidade de relacionar várias tabelas, evitando a redundância no armazenamento de dados. O relacionamento nesse modelo pode ocorrer de três formas:

- A) Um-para-um;
- B) Um-para-muitos;
- C) Muitos-para-muitos

2. FUNDAMENTOS TEÓRICOS DAS SQL VIEWS

2.1. O que são views e como elas funcionam no SQL

Uma view são tabelas virtuais (justamente por não fazerem parte do esquema físico da base) que é formado por declarações SELECTs, aonde retornam uma visualização de dados específica de uma ou mais tabelas de um banco de dados, permitindo que os usuários acessem dados de forma simplificada e organizada.

2.2. Diferença entre views e tabelas comuns

Views: Não armazenam dados fisicamente, apenas a consulta que gera os dados.

Tabelas comuns: Armazenam dados fisicamente no banco de dados.

Diferenças		
	Tabela	View
		
Definição	✓ Uma tabela é um objeto que armazena os dados dentro de um banco de dados.	✓ Uma view é um objeto que permite gerar um conjunto de dados a partir da consulta a uma ou mais tabelas.
Dependência	✓ Uma tabela é um objeto independente.	✓ Uma view depende de uma ou mais tabelas.
Manipulação de dados	✓ Podemos adicionar, atualizar ou excluir dados a partir de uma tabela.	✓ Não é possível adicionar, atualizar ou excluir dados a partir de uma view.
Armazenamento	✓ Uma tabela armazena dados.	✓ Uma view armazena uma query (consulta).

Imagem 2.

2.3. Tipos de views

Simples: Baseadas em uma única tabela e sem funções agregadas. Sendo úteis para oferecer uma visão resumida ou segmentada dos dados originais.

```
CREATE VIEW exemplo_simple_view AS
SELECT coluna1, coluna2
FROM tabela_origem
WHERE coluna3 = 'valor';
```

Imagem 3.

Complexas: Envolvem junções entre múltiplas tabelas e podem incluir funções agregadas. São especialmente úteis para simplificar consultas complexas e fornecer uma visão abrangente dos dados.

```
CREATE VIEW exemplo_complex_view AS
SELECT t1.coluna1, t2.coluna2
FROM tabela1 t1
JOIN tabela2 t2 ON t1.chave = t2.chave
WHERE t1.coluna3 = 'valor';
```

Imagem 4.

Materializadas: Armazenam fisicamente os dados resultantes da consulta, melhorando o desempenho em consultas complexas, mas requerem atualização manual ou automática.

```
CREATE MATERIALIZED VIEW exemplo_materialized_view
REFRESH COMPLETE ON DEMAND
AS
SELECT coluna1, COUNT(coluna2) AS total
FROM tabela_origem
GROUP BY coluna1;
```

Imagem 5.

3. VANTAGENS E DESVANTAGENS DE USAR VIEWS

3.1. Vantagens

Simplificação de consultas complexas: Views permitem encapsular consultas complexas, tornando-as mais fáceis de reutilizar e entender.

Aumento da Segurança: Ao criar views, você pode restringir o acesso a dados sensíveis, exibindo apenas as colunas e linhas necessárias para determinados usuários.

Facilidade de Manutenção: Views ajudam a centralizar e padronizar consultas frequentemente usadas, facilitando a manutenção e atualização.

Vantagens

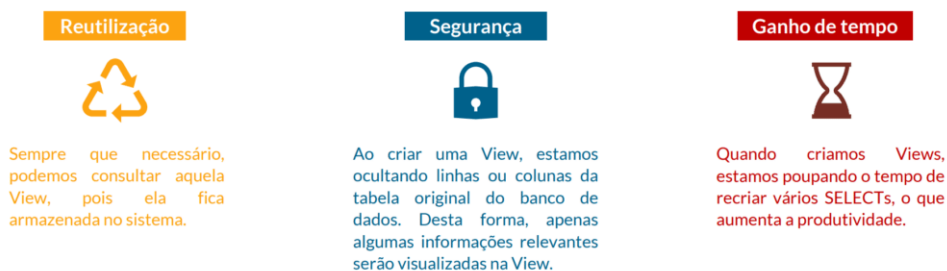


Imagem 6.

3.2. Desvantagens:

Impactos de Desempenho: Quando forem baseadas em consultas complexas ou se não forem otimizadas, podem introduzir sobrecarga de desempenho.

Limitações em Operações de Atualização: As que envolvem múltiplas tabelas ou agregações, podem não ser atualizáveis diretamente.

Manutenção de Views Materializadas: Precisam ser atualizadas de tempos em tempos para refletir mudanças nos dados subjacentes, podendo adicionar complexidade e custo de manutenção.

4. PROCESSO DE CRIAÇÃO DE VIEWS NO SQL

4.1. Instrução CREATE VIEW: sintaxe e parâmetros

A instrução CREATE VIEW é usada para criar uma view, que é uma tabela virtual baseada no resultado de uma consulta SQL. A sintaxe básica é:

Exemplo simples:

```
-- simples
CREATE VIEW Departamento_BH AS
SELECT nome, endereco, funcionarios
FROM departamentos
WHERE endereco = 'Bahia' and nome;
```

Imagem 7

4.2. Filtragem: seleção de colunas e linhas específicas

Uma view que seleciona colunas e linhas específicas:

```
-- filtragem
CREATE VIEW ClientesAtivos AS
SELECT ClienteID, Nome, Cidade
FROM Clientes
WHERE Status = 'Ativo';
```

Imagem 8

4.3. Agregação: uso de funções como SUM, AVG, COUNT

Uma view que usa funções de agregação como SUM, AVG, COUNT:

```
-- agregação
CREATE VIEW vwTotalVendas AS
SELECT ClienteID, SUM(Valor) AS TotalVendas
FROM Vendas
GROUP BY ClienteID;
```

Imagem 9

4.4. Junção: combinar dados de várias tabelas para gerar uma nova visão

Uma view que combina dados de várias tabelas:

```
-- junção
CREATE VIEW vwDetalhesPedidos AS
SELECT Pedidos.PedidoID, Clientes.Nome, Produtos.Nome AS Produto, Pedidos.Quantidade
FROM Pedidos
JOIN Clientes ON Pedidos.ClienteID = Clientes.ClienteID
JOIN Produtos ON Pedidos.ProdutoID = Produtos.ProdutoID;
```

Imagem 10

Exemplo complexa:

```
-- complexa
CREATE VIEW vwRelatorioVendas AS
SELECT Clientes.Nome, Produtos.Nome AS Produto, SUM(Pedidos.Quantidade) AS QuantidadeTotal, SUM(Pedidos.Quantidade * Produtos.Preco) AS ValorTotal
FROM Pedidos
JOIN Clientes ON Pedidos.ClienteID = Clientes.ClienteID
JOIN Produtos ON Pedidos.ProdutoID = Produtos.ProdutoID
WHERE Pedidos.Data BETWEEN '2024-01-01' AND '2024-12-31'
GROUP BY Clientes.Nome, Produtos.Nome;
```

Imagem 11

5. VIEWS ATUALIZÁVEIS E NÃO ATUALIZÁVEIS

5.1. Possibilidade de atualizar dados diretamente em views

```
-- atualizavel
CREATE VIEW vwTotalVendas AS
SELECT ClienteID, SUM(Valor) AS TotalVendas
FROM Vendas
GROUP BY ClienteID;
```

Imagem 12

5.2. Condições para que uma view seja atualizável

- A visualização é deletável.
- A coluna resolve para uma coluna de uma tabela (não utilizar uma operação de referência) e a opção READ ONLY não é especificado.
- Todas as colunas correspondentes dos operandos de um UNION ALL têm exatamente correspondência de tipos de dados (incluindo comprimento ou precisão e escala) e correspondência de valores padrão se a seleção completa da visualização incluir um UNION ALL.

5.3. Exemplo prático de uma view atualizável e outra não atualizável

Atualizável

```
CREATE VIEW updatable_view
(number, current_date, current_time, temperature)
AS
SELECT number, CURRENT DATE, CURRENT TIME, temperature)
FROM weather.forecast
WHERE number = 300
```

Imagem 13

Não atualizável

```
CREATE VIEW MinhaView
AS
SELECT TOP 100 PERCENT Coluna1, Coluna2, Coluna3
FROM tabela
WITH CHECK OPTION
```

Imagem 14

6. ESTUDO DO CASO

6.1. Criação de um banco de dados fictício (exemplo: loja de e-commerce ou sistema de RH)

```
-- criadas
CREATE VIEW vwRelatorioVendas AS
SELECT Data, Produto, SUM(Quantidade) AS QuantidadeVendida, SUM(Quantidade * Preço) AS ValorTotal
FROM Vendas
GROUP BY Data, Produto;

-- consulta de estoque
CREATE VIEW vwEstoqueAtual AS
SELECT ProdutoID, Nome, Quantidade
FROM Produtos
WHERE Quantidade > 0;

-- Folha de Pagamento
CREATE VIEW vwFolhaPagamento AS
SELECT FuncionarioID, Nome, Salario
FROM Funcionarios;
```

Imagem 15

Funções básicas

```
Quando falamos em Views, existem 3 ações envolvidas:

1. Criação de uma View
2. Alteração de uma View
3. Exclusão de uma View

Para cada uma dessas ações, temos um comando associado:
1. CREATE VIEW
2. ALTER VIEW
3. DROP VIEW
```

Imagem 16

CONCLUSÃO

As **SQL Views** são uma forma de simplificar o acesso a dados no banco de dados, permitindo que consultas complexas sejam armazenadas e reutilizadas, sem precisar duplicar informações. Elas ajudam a organizar o código e podem melhorar a segurança, limitando o acesso a dados sensíveis.

No entanto, é preciso ter cuidado, pois elas são muito complexas podendo afetar o desempenho e precisam ser mantidas sempre que houver mudanças nas tabelas. Existem diferentes tipos, como as simples (que envolvem uma tabela), as complexas (que envolvem várias tabelas) e as materializadas (que armazenam dados).

Para usá-las de forma eficiente, é importante evitar criar muitas delas, documentar bem e controlar o acesso aos dados. Também é essencial lembrar que as alterações no banco de dados devem ser feitas diretamente nas tabelas.

BIBLIOGRAFIA

LABORATORIO DE BANCO DE DADOS

https://sae.unb.br/cae/conteudo/unbfga/lbd/banco2_visoes.html#:~:text=Vis%C3%A3o%20em%20SQL&text=Em%20banco%20de%20dados%20relacionais,s e%20fossem%20uma%20tabela%20real

KINGHOST

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

GPO

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

RIC-CPS

https://ric.cps.sp.gov.br/bitstream/123456789/9920/1/bancodedados_2020_1_a_nabeatrizsantos_osbeneficiosdautilizacaodeviews.pdf

STACK OVERFLOW

<https://pt.stackoverflow.com/questions/197842/view-n%C3%A3o-atualiz%C3%A1vel-em-sql-server>

IBM

<https://www.ibm.com/docs/pt-br/db2/11.1?topic=views-updatable>

LISTA DE FIGURAS

Figura 1 – NORMAS ABNT, 2024. Disponível em:

https://sae.unb.br/cae/conteudo/unbfga/lbd/imagens/view_logica.png

Figura 2 – NORMAS ABNT, 2024. Disponível em:

<https://www.hashtagtreinamentos.com/wp-content/uploads/2022/09/Diferenca-de-Tabelas-e-Views-no-SQL-1-1024x489.png>

Figura 3, 4, 5 – NORMAS ABNT, 2024. Disponível em:

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

Figura 6 – NORMAS ABNT, 2024. Disponível em:

<https://www.hashtagtreinamentos.com/wp-content/uploads/2022/09/Diferenca-de-Tabelas-e-Views-no-SQL-2-1024x485.png>