



SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL

SENAI “GASPAR RICARDO JUNIOR”

Curso

**TÉCNICO EM DESENVOLVIMENTO
DE SISTEMAS**

Views SQL

Helena Padilha Bueno

Sorocaba
Novembro – 2024



SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL

SENAI “GASPAR RICARDO JUNIOR”

Helena Padilha Bueno

Views SQL

Pesquisa da matéria de banco de dados

Prof. – Emerson Magalhães

Sorocaba
Novembro – 2024

SUMÁRIO

INTRODUÇÃO.....	2
1. FUNDAMENTOS TEÓRICOS	3
1.1. O que são views e como elas funcionam no SQL:	3
1.2. Diferença entre views e tabelas comuns:	3
1.3. Tipos de views:	3
2. VANTAGENS E DESVANTAGENS	3
2.1. Vantagens:	3
2.2. Desvantagens:	4
3. PROCESSO DE CRIAÇÃO DE VIEWS NO SQL	4
3.1. Exemplos de views simples:	4
3.1.1. View de filtragem: Seleciona colunas e linhas específicas de uma tabela.	4
3.1.2. View de agregação: Utiliza funções como SUM, AVG, COUNT.	4
3.1.3. View de junção: Combina dados de várias tabelas.	5
3.1.4. Exemplo de criação de uma view complexa:	5
4. VIEWS ATUALIZÁVEIS E NÃO ATUALIZÁVEIS	5
4.1. ATUALIZÁVEIS	5
5. ESTUDO DE CASO	6
6. DISCUSSÃO SOBRE AS VANTAGENS DAS VIEWS.....	6
CONCLUSÃO.....	7
BIBLIOGRAFIA	8



Views SQL

INTRODUÇÃO

SQL Views (ou simplesmente “views”) são representações virtuais de dados de uma ou mais tabelas em um banco de dados. Elas atuam como consultas salvas, permitindo que dados complexos sejam apresentados de forma mais simples e acessível. A view não armazena dados por si só, exceto em casos de views materializadas; ela executa uma consulta que retorna dados no momento do acesso. Importância das views em sistemas de banco de dados relacionais: Views são ferramentas essenciais para simplificar o acesso a dados complexos, melhorar a segurança ao restringir a visualização de certas colunas e linhas e facilitar a manutenção de consultas frequentemente utilizadas.

1. FUNDAMENTOS TEÓRICOS

1.1. O que são views e como elas funcionam no SQL:

Views são consultas SQL armazenadas no banco de dados. Elas permitem a criação de uma “visão” customizada dos dados sem modificar as tabelas subjacentes. Sempre que a view é acessada, o banco de dados executa a consulta associada a ela e retorna o resultado.

1.2. Diferença entre views e tabelas comuns:

A principal diferença entre uma view e uma tabela está no armazenamento dos dados. Tabelas armazenam dados diretamente, enquanto views são apenas definições de consulta. Em algumas implementações, existem views materializadas, que armazenam uma cópia dos dados.

1.3. Tipos de views:

- Views simples: Representam dados de uma única tabela e não contêm funções agregadas ou junções complexas.
- Views complexas: Incluem junções entre várias tabelas, agregações e funções, permitindo um processamento mais avançado dos dados.
- Views materializadas: Armazenam o resultado da consulta no banco de dados, o que melhora o desempenho ao custo de manutenção periódica.

2. VANTAGENS E DESVANTAGENS

2.1. Vantagens:

Simplificação de consultas complexas: Views encapsulam consultas complexas, tornando-as acessíveis por meio de uma simples referência.

Aumento da segurança: Elas permitem limitar o acesso a dados confidenciais, exibindo apenas as colunas e linhas necessárias para cada usuário ou aplicação.

Facilidade de manutenção: Ao encapsular consultas, views centralizam o código e facilitam atualizações de lógica sem necessidade de alterar várias partes do sistema.

2.2. Desvantagens:

Possíveis impactos de desempenho: Views complexas podem afetar o desempenho, pois exigem processamento adicional no momento da consulta.

Limitações em operações de atualização: Algumas views não permitem operações de inserção, atualização ou exclusão devido à complexidade das consultas.

Manutenção de views materializadas: As views materializadas precisam ser atualizadas regularmente para refletir mudanças nas tabelas subjacentes, o que pode gerar custos adicionais.

3. PROCESSO DE CRIAÇÃO DE VIEWS NO SQL

Instrução CREATE VIEW: sintaxe e parâmetros:

A criação de uma view é feita com a instrução CREATE VIEW. A sintaxe básica é:

```
CREATE VIEW nome_da_view AS
SELECT coluna1, coluna2 FROM tabela
WHERE condição;
```

3.1. Exemplos de views simples:

3.1.1. View de filtragem: Seleciona colunas e linhas específicas de uma tabela.

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

3.1.2. View de agregação: Utiliza funções como SUM, AVG, COUNT.

```
CREATE VIEW total_vendas_por_cliente AS
SELECT cliente_id, SUM(valor) AS total_vendas
FROM vendas
GROUP BY cliente_id;
```

3.1.3.View de junção: Combina dados de várias tabelas.

```
CREATE VIEW detalhes_pedidos AS
SELECT pedidos.id, clientes.nome, produtos.nome AS produto, pedidos.quantidade
FROM pedidos
JOIN clientes ON pedidos.cliente_id = clientes.id
JOIN produtos ON pedidos.produto_id = produtos.id;
```

3.1.4.Exemplo de criação de uma view complexa:

```
CREATE VIEW resumo_financeiro AS
SELECT vendas.cliente_id, SUM(vendas.valor) AS receita_total, AVG(vendas.valor) AS ticket_medio
FROM vendas
JOIN clientes ON vendas.cliente_id = clientes.id
GROUP BY vendas.cliente_id
HAVING receita_total > 1000;
```

4. VIEWS ATUALIZÁVEIS E NÃO ATUALIZÁVEIS

Algumas views permitem operações de atualização diretamente nas tabelas subjacentes, mas isso depende da simplicidade da view e da compatibilidade com as regras do banco de dados.

4.1. ATUALIZÁVEIS

Em views atualizáveis, é possível realizar operações de UPDATE e DELETE diretamente na view, e as alterações são automaticamente refletidas nas tabelas subjacentes (ou seja, nas tabelas de origem dos dados da view). No entanto, há algumas considerações importantes sobre como o UPDATE e o DELETE funcionam em views e quais restrições podem existir.

```
CREATE VIEW Clientes_Simples AS
SELECT id, nome, email
FROM clientes
WHERE status = 'ativo';
UPDATE Clientes_Simples SET email = 'novo_email@exemplo.com' WHERE id = 1;
```

5. ESTUDO DE CASO

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

View de consulta de estoque:

```
CREATE VIEW estoque_baixo AS
SELECT id, nome FROM produtos
WHERE quantidade < 10;
```

6. DISCUSSÃO SOBRE AS VANTAGENS DAS VIEWS

Nesse cenário, views simplificam relatórios frequentes, como o monitoramento de estoque e o acompanhamento das vendas, e tornam o sistema mais seguro ao exibir dados restritos a determinadas colunas e linhas.

CONCLUSÃO

SQL Views são ferramentas essenciais para organizar, simplificar e proteger dados em bancos de dados relacionais. Elas permitem criar representações virtuais de dados complexos, trazendo praticidade para consultas frequentes e simplificando o desenvolvimento de relatórios e análises. As views são particularmente úteis em sistemas onde o acesso aos dados precisa ser controlado e onde as consultas complexas devem ser abstraídas.

BIBLIOGRAFIA

W3 Schools: https://www.w3schools.com/sql/sql_view.asp

DevMedia: <https://www.devmedia.com.br/conceitos-e-criacao-de-views-no-sql-server/22390>

GeeksForGeeks: <https://www.geeksforgeeks.org/sql-views/>

DevMedia: <https://www.devmedia.com.br/introducao-a-views/1614>

Microsoft Learn: <https://learn.microsoft.com/pt-br/sql/t-sql/statements/create-view-transact-sql?view=sql-server-ver16>

