

**Serviço Nacional de Aprendizagem Industrial**

**SENAI “Gaspar Ricardo Junior”**

**Curso**

**TÉCNICO EM DESENVOLVIMENTO**

**DE SISTEMAS**

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

João Alexandre da Silva Pereira

Sorocaba

Março – 2024



**Serviço Nacional de Aprendizagem Industrial**

**SENAI “Gaspar Ricardo Junior”**

Fernanda de Oliveira Nunes

João Alexandre da Silva Pereira

Júlia Garcia de Carvalho

Marco Antônio da Costa Silva

**IoT**

Trabalho sobre como garantir que diferentes dispositivos IoT de vários fabricantes possam se comunicar de maneira eficaz e interoperável em ambientes industriais

Profº Emerson Magalhães

Sorocaba

Março – 2024

**Sumário**

INTRODUÇÃO 3

1. COMO A UTILILIZAÇÃO DO IOT PODE TORNAR A PRODUÇÃO DAS INDÚSTRIAS MAIS EFICIENTE? 4

1.1. COMO SE COMUNICAM? 6

CONCLUSÃO 7

BIBLIOGRAFIA 8

* **INTRODUÇÃO**

**1 - Definição de SQL Views e sua função em bancos de dados.**

Em banco de dados relacionais, uma visão (ou "view"), é uma consulta (instrução SQL) armazenada no banco de dados geralmente chamada de "tabela virtual". Essa tabela virtual é derivada de uma ou mais tabelas já existentes no banco de dados e podem ser consultadas como se fossem uma tabela real. Ao criar uma view, o resultado da consulta não é armazenado fisicamente no banco de dados, a cada execução os dados são acessados diretamente das tabelas de origem. As views geralmente são usadas para simplificar consultas complexas, fornecer uma exibição personalizada dos dados ou limitar acesso a dados das tabelas reais.

**2 - Importância das views em sistemas de banco de dados relacionais.**

As views são muito usadas para ajudar dar entendimento ao projeto lógico do banco de dados. São muitos os motivos e vantagens para utiliza-las nos projetos. a) É possível ter o controle das informações que o usuário terá acesso.

**3 - Objetivo e abrangência da pesquisa.**

Explicitar o objetivo, que pode ser a análise das principais funcionalidades, vantagens, limitações e as práticas recomendadas para uso de SQL Views em projetos reais de banco de dados.

Abrangência: Cobrir conceitos, funcionalidades, exemplos práticos e um estudo de caso realista.

* **Fundamentos Teóricos das SQL Views**

**1 - O que são views e como elas funcionam no SQL.**

View é um resultado originado de uma consulta pré-definida. Essencialmente é um metadado que mapeia uma query para outra, por isto pode ser considerado como uma tabela virtual. Como o próprio nome diz, ela representa uma visão de dados e não contém dados. Com ela você tem a ilusão que está vendo uma tabela que não existe. Claro que o que você vê nesta tabela existe de outra forma no banco

**2 - Diferença entre views e tabelas comuns.**

**1 - views simples:**

Estas são views básicas que consistem em uma consulta direta e, geralmente, não contêm operações complexas, como agregações ou junções de várias tabelas.

**2 - Views Complexas:**

Incluem consultas com operações como JOIN, agregações (SUM, COUNT, etc.), e podem envolver várias tabelas.

Útil para consolidar dados de várias tabelas ou realizar cálculos.

**3 - views materializadas**

Tabela Temporária com Atualização Manual: Crie uma tabela para armazenar os dados da consulta e atualize-a manualmente ou periodicamente.

Triggers para Atualização Automática: Use triggers para atualizar a tabela automaticamente ao ocorrerem alterações nos dados originais.

Ferramentas de ETL: Use ferramentas externas para automatizar a criação e atualização da tabela materializada.

* **Vantagens das Views:**

- Simplificação de consultas complexas : torna consultas complexas mais acessíveis , permitindo que os usuários realizem uma consulta somente para visualização em vez de uma extensa .acessível , permitindo que os usuários realizem uma consulta somente para visualização, em vez de uma consulta extensa .

- Aumentando a segurança: Restrinja o acesso para certas linhas e colunas , salvaguardandoprotegendo dados sensíveis .confidenciais .

- Facilidade de Controle e Manutenção : Permite consultas frequentes centralizando- as em uma única definição .em uma única definição.

**Desvantagens das Views:**

- Impacto no desempenho: como as visualizações são recalculados em cada câmara, eles podem reduzir o desempenho.

- Limitações nas operações de atualização : Alguns pontos pontos de vistade (particularmente os mais complicados ) não nos permitemnos permitem atualizar os dados diretamente.para atualizar os dados diretamente.

- Manutenção de visualizações materializadas : Da mesma forma, uma visão materializadavisualização requerrequer atualização manual ou automatizada para manter os dados atualizados , aumentando a carga de manutenção.atualização manual ou automatizada para manter os dados atualizados , aumentando a carga de manutenção .

* **Processo de Criação de Views no SQL**

As Views são uma forma de armazenar uma consulta SQL em um objeto de banco de dados, permitindo fácil reutilização e simplificação do código. Elas ajudam a criar "visões" específicas dos dados, sem a necessidade de duplicar os dados no banco.

**Instrução CREATE VIEW: Sintaxe e Parâmetros**

A sintaxe básica para criar uma view é:

CREATE VIEW nome\_da\_view AS

SELECT colunas

FROM tabela WHERE condições;

**Parâmetros:**

nome\_da\_view: Nome que identifica a view.

SELECT colunas FROM tabela WHERE condições: Consulta que define quais dados a view irá exibir.

**Exemplos de Views Simples**

**View de Filtragem**

Uma view de filtragem seleciona colunas e linhas específicas. Por exemplo:

CREATE VIEW vendas\_filtradas AS

SELECT produto\_id, quantidade, preco

FROM vendas WHERE quantidade > 10;

Neste exemplo, criamos uma view chamada vendas\_filtradas que exibe apenas os registros de vendas com uma quantidade maior que 10.]

**View de Agregação**

Views de agregação usam funções como SUM, AVG, COUNT, etc., para resumir dados. Exemplo:

CREATE VIEW total\_vendas\_por\_produto AS

SELECT produto\_id, SUM(quantidade) AS total\_quantidade, AVG(preco) AS preco\_medio

FROM vendas

GROUP BY produto\_id;

Aqui, a view total\_vendas\_por\_produto exibe o total de quantidade vendida e o preço médio para cada produto.

**View de Junção**

Uma view de junção combina dados de várias tabelas. Exemplo:

CREATE VIEW informacoes\_vendas AS

SELECT v.venda\_id, v.produto\_id, p.nome AS nome\_produto, v.quantidade, v.preco

FROM vendas v

JOIN produtos p ON v.produto\_id = p.produto\_id;

Neste caso, a view informacoes\_vendas une as tabelas vendas e produtos para exibir informações adicionais do produto em cada venda.

Essas views podem ser reutilizadas em consultas, facilitando a organização e a legibilidade do código SQL.

**Views Atualizáveis e Não Atualizáveis**

Visualização Atualizável: permite entrada, exclusão ou atualização direta de dados .entrada, exclusão ou atualização. Geralmente, ele se com base em uma única tabela sem adições complexas ouadições ou junções junções.

CREATE VIEW clientes\_view AS

SELECT cliente\_id, nome, email FROM clientes;

É possível atualizar dados diretamente:

UPDATE clientes\_view SET email = 'novoemail@dominio.com' WHERE cliente\_id = 1;

Exibir não atualizável: não permite modificação direta de permitirdados .modificação direta de dados . Isso ocorre quando a visão envolve juntas, agregações ou outras operações complexas .operações

CREATE VIEW vendas\_resumo AS

SELECT v.produto\_id, p.nome AS produto\_nome, SUM(v.quantidade) AS total\_vendido

FROM vendas v

JOIN produtos p ON v.produto\_id = p.produto\_id

GROUP BY v.produto\_id, p.nome;

* **CONCLUSÃO**

**Principais Pontos:**

Views são consultas SQL armazenadas no banco de dados, oferecendo uma "tabela virtual" que simplifica o acesso aos dados.

São úteis para simplificar consultas, melhorar a segurança e facilitar a manutenção de consultas frequentes.

No entanto, podem impactar o desempenho e limitar atualizações, especialmente em views complexas com junções ou agregações.

**Considerações Finais:**

Views são essenciais para organizar dados e controlar acessos em projetos de banco de dados, mas devem ser usadas com cautela para evitar problemas de desempenho e atualização.

**Práticas Recomendadas:**

Priorizar views simples e eficientes.

Evitar complexidade desnecessária, que pode prejudicar o desempenho.

Planejar atualizações de views materializadas adequadamente.

Monitorar o impacto no desempenho.

Utilizar views para controlar acessos a dados sensíveis.

Em resumo, views são uma ferramenta valiosa quando bem utilizadas, ajudando a organizar e otimizar a consulta de dados em sistemas de banco de dados.

**BIBLIOGRAFIA**

[**https://sae.unb.br/cae/conteudo/unbfga/lbd/banco2\_visoes.html#:~:text=Em%20banco%20de%20dados%20relacionais,se%20fossem%20uma%20tabela%20real**](https://sae.unb.br/cae/conteudo/unbfga/lbd/banco2_visoes.html#:~:text=Em%20banco%20de%20dados%20relacionais,se%20fossem%20uma%20tabela%20real)

[**https://pt.stackoverflow.com/questions/35413/o-que-s%C3%A3o-views-em-sql-quais-vantagens-e-desvantagens-em-utilizar#:~:text=View%20%C3%A9%20um%20resultado%20originado,dados%20e%20n%C3%A3o%20cont%C3%A9m%20dados**](https://pt.stackoverflow.com/questions/35413/o-que-s%C3%A3o-views-em-sql-quais-vantagens-e-desvantagens-em-utilizar#:~:text=View%20%C3%A9%20um%20resultado%20originado,dados%20e%20n%C3%A3o%20cont%C3%A9m%20dados)