Views

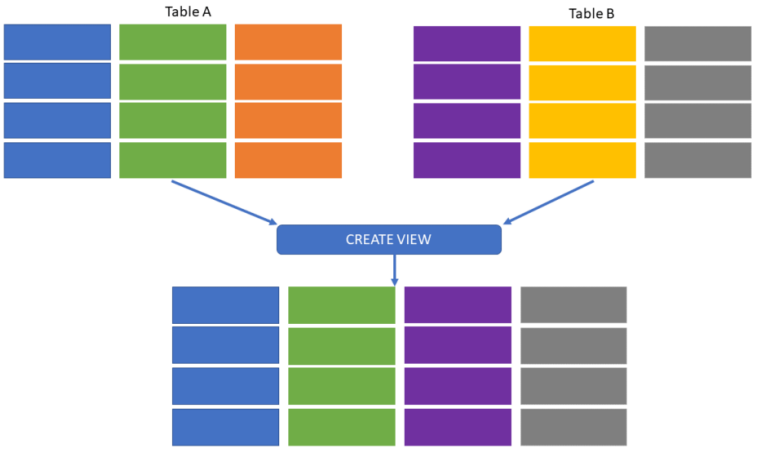
Programação e Administração de Banco de Dados

Prof. Jeferson Queiroga

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE

Introdução

- **O que são Views?**
- Views são representações virtuais de uma ou mais tabelas baseadas em consultas SQL.
- Elas não armazenam dados fisicamente, mas fornecem uma visão filtrada e/ou formatada das tabelas subjacentes.



Introdução

Por que usar Views?

- Simplificar consultas complexas: *Views* podem consolidar várias tabelas e operações em uma única consulta, tornando o acesso a dados mais fácil e organizado.

- Facilitar o acesso a dados específicos: *Views* permitem que os usuários vejam apenas as colunas ou linhas de interesse, mantendo a estrutura original das tabelas intacta.

- Melhorar a segurança e o desempenho das consultas: *Views* podem ser usadas para limitar o acesso a dados sensíveis e otimizar o desempenho de consultas, quando aplicadas corretamente.

Sintaxe básica

```
1. CREATE [OR REPLACE] [TEMP | TEMPORARY] VIEW view_name AS
2. SELECT column1, column2, ...
3. FROM table_name
4. WHERE condition;
5.
```

Criando o view

```
1  CREATE VIEW sales_by_customer AS
2  SELECT c."CustomerId", c."FirstName", c."LastName", SUM(i."Total") as total_sales
3  FROM "Customer" c
4  INNER JOIN "Invoice" i ON c."CustomerId" = i."CustomerId"
5  GROUP BY c."CustomerId", c."FirstName", c."LastName"
6  ORDER BY total_sales DESC;
7
8  SELECT * FROM sales_by_customer;
```

Isso retornará uma lista dos clientes e seus respectivos totais de vendas, ordenados em ordem decrescente pelo total de vendas.

Views Temporárias

- Definição: São views que existem apenas durante a sessão atual e são automaticamente removidas ao encerrar a sessão.
- Utilidade: Úteis para consultas e análises temporárias, sem afetar o esquema do banco de dados permanentemente.

```
CREATE TEMPORARY VIEW temp_employees_view AS
SELECT employee_id, first_name, last_name, department
FROM employees;
```

Views Materializadas

- Definição: São views que armazenam fisicamente o resultado da consulta em disco, o que pode melhorar o desempenho de consultas frequentes.

- Utilidade: Úteis para otimizar consultas com dados que não mudam frequentemente, reduzindo o tempo de consulta em operações repetidas.

```
CREATE MATERIALIZED VIEW sales_summary_view AS
SELECT customer_name, SUM(quantity) AS total_sales
FROM orders
GROUP BY customer_name;
```

Views Materializadas

- Exemplo 1: Relatório de vendas mensal - Suponha que você tenha um banco de dados de vendas e queira gerar um relatório mensal de vendas por produto. Nesse caso, você pode criar uma view materializada que agrupe as vendas por mês e produto, já que esses dados só precisam ser atualizados uma vez por mês.

```
CREATE MATERIALIZED VIEW monthly_sales_report AS
SELECT
  product_id,
  date_trunc('month', order_date) AS month,
  SUM(quantity) AS total_quantity,
  SUM(quantity * price) AS total_sales
FROM
  sales
GROUP BY
  product_id,
  month;
```

Views Materializadas

- Lembre-se de que as views materializadas não são atualizadas automaticamente quando os dados subjacentes são alterados.

```
Comando para atualizar:
REFRESH MATERIALIZED VIEW view_name;
```

Atualizando Views Regulares

- Para modificar a definição de uma view, utilize a cláusula OR REPLACE ao criar a view:

Comando:

```
CREATE OR REPLACE VIEW view_name AS
SELECT ...
```

Excluir Views Regulares

- Para excluir uma view (regular ou materializada), utilize o comando DROP VIEW ou DROP MATERIALIZED VIEW, respectivamente:

Comando:

```
DROP VIEW view_name;
DROP MATERIALIZED VIEW materialized_view_name;
```

Views e Controle de Acesso

Limitando o acesso a dados sensíveis

- Views podem ser usadas para restringir o acesso a informações sensíveis.
- Ao criar views, selecione apenas as colunas necessárias e aplique condições para limitar o acesso aos dados.
- Exemplo: Ocultar informações pessoais dos funcionários e mostrar apenas seus IDs e departamentos.

Passo 01:

```
CREATE VIEW limited_employee_data AS
SELECT employee_id, department
FROM employees;
```

Views e Controle de Acesso

Revogar permissões de INSERT, UPDATE e DELETE para um usuário em uma view específica.

Passo 02:

```
REVOKE INSERT, UPDATE, DELETE ON
limited_employee_data FROM user_name;
```

Exercício 1

- Neste exercício, você criará uma view que mostra os artistas mais vendidos no banco de dados Chinook. A view deve incluir o nome do artista, o número total de faixas vendidas e o valor total de vendas. O nome da view top_selling_artists.
- A view deve listar os artistas em ordem decrescente com base no número total de faixas vendidas.

Exercício 2

- Vamos supor que você possui o banco de dados Chinook com informações sobre artistas, álbuns e faixas. Você criou uma view chamada artist_albums que mostra informações detalhadas sobre os álbuns de cada artista. Agora, você deseja gerenciar o acesso a essa view para diferentes usuários:

a) Conceda permissão de SELECT na view artist_albums para o usuário "user1".

b) Conceda todas as permissões na view artist_albums para um grupo de usuários chamado "music_team".