

# Projeto de Banco de Dados



## Visões de Banco de Dados

**PROF. DR. THIAGO ELIAS**

# Visões de BD no PostgreSQL



- São uma maneira simples de executar e exibir dados selecionados de consultas complexas em bancos.
- Em que elas são úteis? Elas economizam grande quantidade de digitação e esforço e apresentam somente os dados que desejamos.

# Visões de BD no PostgreSQL



- A visão não é materializada fisicamente (não ocupa espaço em disco). Em vez disso, a consulta é executada toda vez que a visão é referenciada em uma consulta.
- As visões podem ser utilizadas em praticamente todos os lugares onde uma tabela real pode ser utilizada.
- É possível construir visões baseadas em visões.

# Visões de BD no PostgreSQL



- **Vantagem das Views:**
  - **Reuso:** as views são objetos de caráter permanente. Pensando pelo lado produtivo isso é excelente, já que elas podem ser lidas por vários usuários simultaneamente.
  - **Segurança:** as views permitem que ocultemos determinadas colunas de uma tabela. Para isso, basta criarmos uma view com as colunas que achamos necessário que sejam exibidas e as disponibilizarmos para determinados usuários.
  - **Simplificação do código:** as views nos permitem criar um código de programação muito mais limpo, na medida em que podem conter um SELECT complexo. Assim, criar views para os programadores a fim de poupá-los do trabalho de criar SELECT's é uma forma de aumentar a produtividade da equipe de desenvolvimento.

# Visões de BD no PostgreSQL



- Exemplos:

```
CREATE OR REPLACE VIEW TESTE1  
AS  
SELECT * FROM CLIENTE WHERE NOME ILIKE 'M%'
```

```
CREATE OR REPLACE VIEW TESTE1 (CODIGO, NOME_CLI)  
AS  
SELECT * FROM CLIENTE WHERE NOME ILIKE 'M%'
```

```
CREATE OR REPLACE VIEW TESTE1  
AS  
SELECT * FROM CLIENTE WHERE NOME ILIKE 'M%'  
WITH CHECK OPTION
```

# Visões MATERIALIZADAS de BD no PostgreSQL



- A partir da versão 9.3
- Define uma visualização materializada de uma consulta.
- Vantagem?
  - Melhora performance uma vez que índices podem ser criados para a visão
- A consulta é executada e usada para preencher a visão no momento em que o comando é executado (a menos que WITH NO DATA seja usado).
- Pode ser atualizada posteriormente usando REFRESH MATERIALIZED VIEW.

# Visões MATERIALIZADAS de BD no PostgreSQL



```
CREATE MATERIALIZED VIEW TESTE
AS
SELECT * FROM CLIENTE WHERE NOME ILIKE 'M%'
WITH [NO] data
```

- Teste (erro):

```
CREATE MATERIALIZED VIEW TESTE2
AS
SELECT * FROM CLIENTE WHERE NOME ILIKE 'M%'
WITH NO DATA;

SELECT * FROM TESTE2;

REFRESH MATERIALIZED VIEW TESTE2;

SELECT * FROM TESTE2;
```

# Visões de BD no PostgreSQL



- Exercício:

- Crie a tabela venda com os seguintes atributos:
  - ✦ Cod\_venda
  - ✦ Nome\_vendedor
  - ✦ data\_venda
  - ✦ Valor\_vendido
- 1) Povoie a tabela com 10 vendas, considerando que existam apenas 4 vendedores na loja.
- 2.1) Mostre o nome dos vendedores que venderam mais de X reais no mês de março de 2024.
- 2.2) Mostre o nome de um dos vendedores que mais vendeu no mês de março de 2024
- 3) Sem usar “select na cláusula from,” qual o nome do(s) vendedor(es) que mais vendeu no mês de março de 2024?