

# Linguagem de Definição de Dados (DDL) - Continuação



Prof. Jeferson Souza, MSc.

*(jefecomp)*

[jefecomp.official@gmail.com](mailto:jefecomp.official@gmail.com)

# Visões (Views)

O que são visões (views)?

Visões são uma forma alternativa de acessar dados no modelo de dados. As visões são criadas a partir de consultas válidas realizadas sobre tabelas ou outras visões;

# Visões (Views)

O que são visões (views)?

Visões são uma forma alternativa de acessar dados no modelo de dados. As visões são criadas a partir de consultas válidas realizadas sobre tabelas ou outras visões;

Para que servem as visões (views)?

- ▶ Fornecem um meio de acesso mais simples e direto ao dados;
- ▶ Permitem acesso controlado e limitado a dados com restrições de acesso;
- ▶ Permitem "estender" virtualmente o modelo de dados, e criar relações virtuais que podem ser utilizadas exatamente da mesma forma que tabelas.

# Diferença entre tabelas e visões (views)

- ▶ Tabelas são estruturas criadas para armazenar dados;
- ▶ Visões são o resultado de consultas que podem ser manipuladas posteriormente da mesma forma que tabelas;
- ▶ Caso não exista a necessidade de relacionamentos, tabelas podem ser criadas sem a existência de outras estruturas na base de dados;
- ▶ A criação de uma visão depende da existência de, ao menos, uma tabela;

# Diferença entre tabelas e visões (views)

## (Continuação)

- ▶ Os dados mostrados por uma visão podem ser modificados (em alguns casos) por comandos de inserção, remoção, e atualização. Entretanto, modificações não são recomendadas, já que as mesmas devem refletir modificações nas tabelas que dão origem a visão alvo;

# Criar visões (views)

**CREATE VIEW**

comando

NOME\_DA VISÃO

argumento  
obrigatório

(nome das colunas)

argumento  
opcional

AS

&lt;Consulta&gt;;

argumento  
obrigatório

# Criar visões (views)

Exemplos:

OBS: cargo\_id = 1 é o identificador para o cargo de Professor.

```
CREATE VIEW professores_view as SELECT nome, sobrenome,  
email, cpf, cargo_id, departamento_id FROM usuario where  
cargo_id = 1;
```

## Criar visões (views)

Exemplos:

OBS: cargo\_id = 1 é o identificador para o cargo de Professor.

```
CREATE VIEW professores_view as SELECT nome, sobrenome,  
email, cpf, cargo_id, departamento_id FROM usuario where  
cargo_id = 1;
```

Pode-se depois realizar consultas diretamente na Visão(View)

```
SELECT * FROM professores_view where departamento_id = 1;
```

OBS: departamento\_id=1 é o identificador do DCC.



# Visões Materializadas (Materialised Views)

Alguns SGBDs permitem que visões sejam armazenadas em disco. Esse tipo de visão é chamada de **visão materializada (materialised view)**. Caso dados sejam inseridos na base de dados, e o resultado da consulta que define a visão mude, a visão materializada também é atualizada.

# Criar Visões Materializadas (Materialised Views)

**CREATE MATERIALIZED VIEW** **NOME\_DA VISÃO** **(nome das colunas)** AS **<Consulta>;**  
comando                      argumento obrigatório                      argumento opcional                      argumento obrigatório

## Criar Visões Materializadas (Materialised Views)

```
CREATE MATERIALIZED VIEW professores_view as SELECT  
nome, sobrenome, email, cpf, cargo_id, departamento_id FROM  
usuario where cargo_id = 1;
```

```
SELECT * FROM professores_view where departamento_id = 1;
```

# Bibliografia



Garcia-Molina, H. and Ullman, J. D. and Widom, J.  
*"Database Systems: The Complete Book"*. 2nd edition.  
Prentice Hall, 2008.

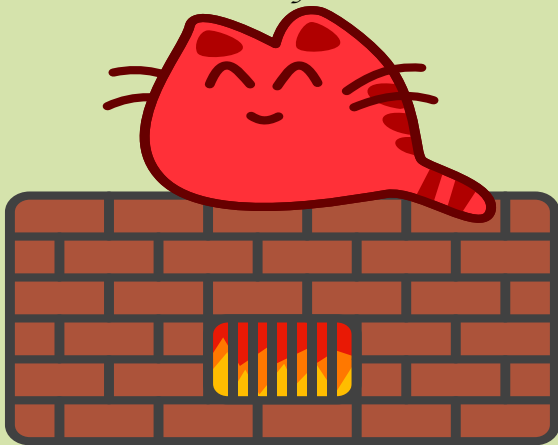


PostgreSQL Development Group.  
*"PostgreSQL 10.2 Documentation"*. 2018.



Silberschatz, A. and Korth, H.F. and Sudarshan, S.  
*"Database Systems"*. 6th edition. McGrawHill, 2011.

*That's it folks!*



*Thank you for your attention!*