



Visões



Bruno A. N. Travençolo – FACOM

# VISÕES

---

- ▶ Uma visão é uma descrição alternativa para um subconjunto dos dados selecionados de uma ou mais tabelas da base de dados (tabela virtual). É possível executar operações de seleção, junção, inserção, atualização e remoção sobre uma visão, como se ela fosse uma tabela regular da base de dados, com apenas poucas restrições:
  - ▶ Existem visões atualizáveis e visões “read-only”
  - ▶ No PostgreSQL elas são *read-only* – ou seja, não é possível inserir, deletar ou atualizar por meio de uma visão
- ▶ Sintaxe

```
CREATE [ OR REPLACE ] [ TEMP | TEMPORARY ] VIEW  
name [ ( column_name [, ...] ) ] AS query
```

---



# Exemplo

---

## ► Tabela Matrícula

**MATRICULA**(**codigoturma**,**nmat**,**nota**)

**codigoturma** => código da turma da disciplina

**nmat** => número de matrícula do aluno

**nota** => nota do aluno na disciplina



# Comando CREATE VIEW

---

-- cria a view Aprovações

**CREATE VIEW** Aprovacoes **AS**

**SELECT** \*

**FROM** Matricula

**WHERE** nota >= 60;

-- mostrando os alunos aprovados

**SELECT** \* **FROM** Aprovacoes;

-- Tentando inserir dados na base por meio da VIEW

**INSERT INTO** Aprovacoes **VALUES** (102, 1234,3);

(PostgreSQL 8.4 não deixa fazer a inserção: “ERRO: não pode inserir em uma visão”. Verificar a possibilidade na versão 9)



# Comando CREATE VIEW

## CREATE RULE

---

- ▶ Inserção por meio de visões: deve-se criar uma “rule” (regra)

-- Criando uma regra para inserção

**CREATE RULE** rap **AS**

**ON INSERT TO** Aprovacoes /\* A regra define o que fazer qdo um ‘insert’  
é feito em Aprovações \*/

**DO INSTEAD** -- “faça no lugar”

**INSERT INTO** matricula **VALUES** (

NEW.codigoturma,

NEW.nmat,

NEW.nota);

-- ou seja, quando um ‘insert’ é feito na view Aprovações o que realmente ocorre é um ‘insert’ na tabela Matricula.

-- Apagando a rule “rap”:

-- **DROP RULE** rap **ON** aprovacoes

---



# Variável NEW

---

- ▶ Carrega as informações que foram passadas ao comando INSERT INTO
- ▶ Comando que 'dispara' a regra:

-- Tentando inserir dados na base por meio da VIEW

**INSERT INTO** Aprovacoes **VALUES** (102, 1234, 3);

-- Criando uma regra para inserção

**CREATE RULE** rap **AS**

**ON INSERT TO** Aprovacoes

**DO INSTEAD** -- “faça no lugar”

**INSERT INTO** matricula **VALUES** (

NEW.codigoturma,

NEW.nmat,

NEW.nota);



# Sintaxe para REGRA

---

```
CREATE [ OR REPLACE ] RULE nome AS ON evento  
  TO tabela [ WHERE condição ]  
  DO [ ALSO | INSTEAD ] { NOTHING | comando | ( comando ; comando ... ) }
```

Observe que mais de um comando pode ser executado no DO INSTEAD, desde que coloque parênteses



# Comando CREATE VIEW

## CREATE RULE

---

- ▶ Inserção por meio de visões: deve-se criar uma “rule” (regra)

-- Criando uma regra para inserção

**CREATE RULE** rap **AS**

**ON INSERT TO** Aprovacoes /\* A regra define o que fazer qdo um ‘insert’  
é feito em Aprovações \*/

**DO INSTEAD** -- “faça no lugar”

**INSERT INTO** matricula **VALUES** (

NEW.codigoturma,

NEW.nmat,

NEW.nota);

-- ou seja, quando um ‘insert’ é feito na view Aprovações o que realmente ocorre é um ‘insert’ na tabela Matricula.

-- Apagando a rule “rap”:

-- **DROP RULE** rap **ON** aprovacoes

---





# Comando CREATE VIEW

## CREATE RULE

---

**INSERT INTO** Aprovacoes **VALUES** (102, 1234,3);

-- observe que a nota inserida é  $< 60$ , violando a condição usada na criação da VIEW. Para evitar essa violação deve-se usar "WITH CHECK OPTION"

-- cria a view aprovações

**CREATE VIEW** Aprovacoes  
**AS SELECT \*** **FROM** Matricula  
**WHERE** nota  $\geq 60$

**WITH CHECK OPTION;** -- Não implementado no postgresql 8.4 – verificar nas novas versões

**Exercício:** Verificar se (102, 1234,3) foi gravado na tabela matrícula e se aparece em aprovações.

---




# Comando CREATE VIEW

---

-- VIEW com as médias de cada aluno

```
CREATE VIEW Medias (NumeroMatricula, Media) AS  
SELECT NMat, Avg(nota)  
FROM Matricula  
GROUP BY NMat;
```



Ao usar funções de agregação, deve ser informado o nome da coluna para essa função

-- Selecionando a maior média

```
SELECT NumeroMatricula, media  
FROM Medias  
WHERE media = (SELECT Max(media)  
FROM Medias);
```



# Bibliografia

---

- ▶ Slides adaptados da aula da Profa. Josiane M. Bueno (*in memoriam*)

