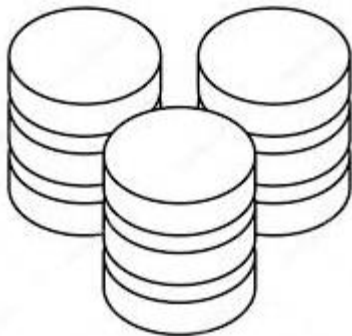


Banco de Dados II

Visões



Profa. Damires Souza
damires@ifpb.edu.br



O que é uma visão?



Visão de Dados

Empregado

| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | EMAIL | PHONE_NUMBER | HIRE_DATE | JOB_ID | SALARY | COMMISSION_PCT | MANAGER_ID | DEPARTMENT |
|-------------|-------------|-----------|----------|--------------|-----------|------------|--------|----------------|------------|------------|
| 100 | Steven | King | SKING | 515.123.4567 | 17/06/03 | AD_PRES | 24000 | (null) | (null) | |
| 101 | Neena | Kochhar | NKOCHHAR | 515.123.4568 | 21/09/05 | AD_VP | 17000 | (null) | 100 | |
| 102 | Lex | De Haan | LDEHAAN | 515.123.4569 | 13/01/01 | AD_VP | 17000 | (null) | 100 | |
| 103 | Alexander | Hunold | AHUNOLD | 590.423.4567 | 03/01/06 | IT_PROG | 9000 | (null) | 102 | |
| 104 | Bruce | Ernst | BERNST | 590.423.4568 | 21/05/07 | IT_PROG | 6000 | (null) | 103 | |
| 105 | David | Austin | DAUSTIN | 590.423.4569 | 25/06/05 | IT_PROG | 4800 | (null) | 103 | |
| 106 | Valli | Pataballa | VPATABAL | 590.423.4560 | 05/02/06 | IT_PROG | 4800 | (null) | 103 | |
| 107 | Diana | Lorentz | DLORENTZ | 590.423.5567 | 07/02/07 | IT_PROG | 4200 | (null) | 103 | |
| 108 | Nancy | Greenberg | NGREENBE | 515.124.4569 | 17/08/02 | FI_MGR | 12008 | (null) | 101 | 1 |
| 109 | Daniel | Faviet | DFAVIET | 515.124.4169 | 16/08/02 | FI_ACCOUNT | 9000 | (null) | 108 | 1 |
| 110 | John | Chen | JCHEN | 515.124.4269 | 28/09/05 | FI_ACCOUNT | 8200 | (null) | 108 | 1 |
| 111 | Ismael | Sciarra | ISCIARRA | 515.124.4369 | | | | | | 1 |
| 112 | Jose Manuel | Urman | JMURMAN | 515.124.4469 | | | | | | 1 |
| 113 | Luis | Popp | LPOPP | 515.124.4567 | | | | | | 1 |

| FIRST_NAME | LAST_NAME | EMAIL | PHONE_NUMBER |
|-------------|-----------|----------|--------------|
| Steven | King | SKING | 515.123.4567 |
| Neena | Kochhar | NKOCHHAR | 515.123.4568 |
| Lex | De Haan | LDEHAAN | 515.123.4569 |
| Alexander | Hunold | AHUNOLD | 590.423.4567 |
| Bruce | Ernst | BERNST | 590.423.4568 |
| David | Austin | DAUSTIN | 590.423.4569 |
| Valli | Pataballa | VPATABAL | 590.423.4560 |
| Diana | Lorentz | DLORENTZ | 590.423.5567 |
| Nancy | Greenberg | NGREENBE | 515.124.4569 |
| Daniel | Faviet | DFAVIET | 515.124.4169 |
| John | Chen | JCHEN | 515.124.4269 |
| Ismael | Sciarra | ISCIARRA | 515.124.4369 |
| Jose Manuel | Urman | JMURMAN | 515.124.4469 |
| Luis | Popp | LPOPP | 515.124.4567 |
| Den | Raphaely | DRAPHEAL | 515.127.4561 |

DadosPessoaisEMP

Visão (*View*)

- Uma visão, no **modelo relacional**, é uma **tabela VIRTUAL derivada** de outra tabela ou de outras tabelas
 - **Não** possui dados de forma física
 - Fornece uma **visão particular** de um conjunto de dados
 - Pode ser **consultada** como as demais tabelas
 - É um meio para especificação de uma tabela que precise ser **consultada frequentemente, embora não exista fisicamente.**

- Problema: Criar uma **view** que contenha somente os produtos cuja unidade é o Quilo.

CREATE or replace **VIEW Prquilo**

(codigo, descricao,unidade)

AS Select codprod,descricao,unidade

From **produto**

Where unidade = 'KG';

| codprod [PK] integer | descricao character varying (20) | valor numeric (10,2) | unidade character (2) |
|-------------------------|-------------------------------------|-------------------------|--------------------------|
| 1 | Queijo | 22.00 | KG |
| 4 | Linho | 24.00 | M |
| 5 | Feijao | 12.00 | KG |
| 2 | Chocolate | 6.27 | G |
| 3 | Leite | 10.25 | L |
| 6 | Açucar | 9.23 | KG |
| 7 | XXX | 1.23 | KG |



Select * from Prquilo;

Select descricao from prquilo;

| codigo integer | descricao character varying (20) | unidade character (2) |
|-------------------|-------------------------------------|--------------------------|
| 1 | Queijo | KG |
| 5 | Feijao | KG |
| 6 | Açucar | KG |
| 7 | XXX | KG |

Tipos de views

■ Visão Idêntica

TABELA

| A | B | C |
|---|---|---|
| | | |
| | | |
| | | |

VISÃO

| A | B | C |
|---|---|---|
| | | |
| | | |
| | | |

■ Visão por Seleção de Linhas

TABELA

| A | B | C |
|---|---|---|
| | | |
| | | |
| | | |

VISÃO

| A | B | C |
|---|---|---|
| | | |
| | | |

■ Visão por Seleção de Colunas

TABELA

| A | B | C |
|---|---|---|
| | | |
| | | |
| | | |

VISÃO

| A | C |
|---|---|
| | |
| | |
| | |

■ Visão por Seleção de Linhas e Colunas

TABELA

| A | B | C | D |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |

VISÃO

| A | B |
|---|---|
| | |
| | |

■ Visão por Junção de Tabela

TABELA

| A | B | C |
|---|---|---|
| | | |
| | | |
| | | |
| | | |

VISÃO

| A | B | C | D | E |
|---|---|---|---|---|
| | | | | |
| | | | | |

| A | D | E |
|---|---|---|
| | | |
| | | |

Criando uma view

```
CREATE [ OR REPLACE ] [ TEMP ] VIEW
```

```
  name [ ( column_name [, ...] ) ]
```

```
  AS query
```

```
[ WITH CHECK OPTION ]
```

- **Replace:** recria a view, caso já exista
- **With Check Option:**
 - Esta opção está associada às visões atualizáveis. Novos dados devem satisfazer às condições que definem a visão. Se as condições não forem satisfeitas, a atualização será rejeitada.

Comando DDL

Exemplo

- Problema: Criar uma **view** que contenha o **código** do vendedor, seu **nome** e o **salário**.

```
CREATE OR REPLACE VIEW
```

```
VendSal(codigo,nome,salario)
```

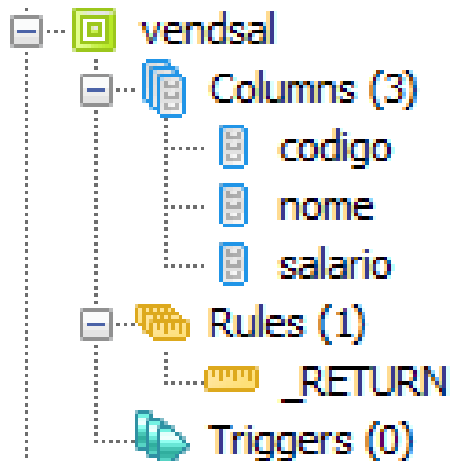
```
AS  Select codvend,nome,salariofixo  
    From vendedor;
```

```
Select * from VendSal;
```

```
Select nome from vendSal order by nome;
```


Internamente...

```
CREATE OR REPLACE RULE "_RETURN" AS
  ON SELECT TO vendsal DO INSTEAD SELECT vendedor.codvend AS codigo,
  vendedor.nome,
  vendedor.salariofixo AS salario
FROM vendedor;
```



| | General | Definition | Condition | Commands | SQL |
|---|---------|------------|-----------|---------------|--|
| 1 | | | | SELECT | vendedor.codvend AS codigo, |
| 2 | | | | | vendedor.nome, |
| 3 | | | | | vendedor.salariofixo AS salario |
| 4 | | | | FROM | vendedor |

Visão (View)

➤ A view **vendsal** é uma **TABELA VIRTUAL**

- Normalmente para **leitura**
- A view reflete as atualizações e dados existentes na(s) **tabela(s) base**
 - ** Nas tabelas base, **tuplas são realmente armazenadas**
- **Isso limita as operações de atualização**
 - Mas, não impõe limites para consultas.

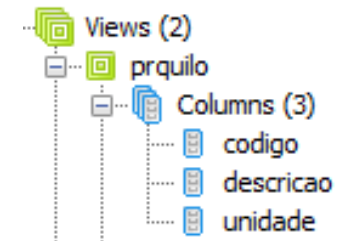
O que acontece com o insert??

**** Veja a view prquilo
no catálogo**

Select * from prquilo
order by codigo;

Select * from produto
order by codprod;

**Insert into PRquilo
values (110,'Arroz','KG');**



View PRQUILO

| codigo integer | descricao character varying (20) | unidade character (2) |
|-------------------|-------------------------------------|--------------------------|
| 1 | Queijo | KG |
| 5 | Feijao | KG |
| 6 | Açucar | KG |
| 7 | XXX | KG |

Tabela Produto

| codprod [PK] integer | descricao character varying (20) | valor numeric (10,2) | unidade character (2) |
|-------------------------|-------------------------------------|-------------------------|--------------------------|
| 1 | Queijo | 22.00 | KG |
| 4 | Linho | 24.00 | M |
| 5 | Feijao | 12.00 | KG |
| 2 | Chocolate | 6.27 | G |
| 3 | Leite | 10.25 | L |
| 6 | Açucar | 9.23 | KG |
| 7 | XXX | 1.23 | KG |

O que acontece com o insert??

**Insert into PRquilo
values (110,'Arroz','KG');**

- Select * from prquilo;
- Select * from produto;

View PRQUILO

| codigo integer | descricao character varying (20) | unidade character (2) |
|-------------------|-------------------------------------|--------------------------|
| 1 | Queijo | KG |
| 5 | Feijao | KG |
| 6 | Açucar | KG |
| 7 | XXX | KG |
| 110 | Arroz | KG |

Tabela Produto

| codprod [PK] integer | descricao character varying (20) | valor numeric (10,2) | unidade character (2) |
|-------------------------|-------------------------------------|-------------------------|--------------------------|
| 1 | Queijo | 22.00 | KG |
| 4 | Linho | 24.00 | M |
| 5 | Feijao | 12.00 | KG |
| 2 | Chocolate | 6.27 | G |
| 3 | Leite | 10.25 | L |
| 6 | Açucar | 9.23 | KG |
| 7 | XXX | 1.23 | KG |
| 110 | Arroz | [null] | KG |

DML por meio de View??

Update Prquilo

Set descricao = 'Arroz Integral'

Where codigo = 110;

Select * from prquilo order by codigo;

Select * from produto order by codprod;

Delete from PRquilo

where descricao = 'Arroz Integral';

ATENÇÃO

- **Atualizações de visões** podem ser complicadas e ambíguas



- Se a view for do tipo **join** ou for criada com **group by**, operadores **distinct** ou **funções de grupo**, não poderá ser diretamente utilizada para inclusão, exclusão ou alteração.

Exemplo

CREATE or replace VIEW **Listapedidos** AS

Select nome, descricao

From vendedor v join pedido p on **v.codvend = p.codvend**

Join itenspedido i on **p.numped = i.numped**

join produto pr on **i.codprod = pr.codprod**

order by nome;

Select * from listapedidos order by nome;

Podemos inserir por meio desta view???

Outro Exemplo

Create or replace view **totalsalarios** as
select sum(salariofixo) as TotaldeSalarios
from vendedor;

Select * from totalsalarios;

Update totalsalarios

Set total = total + 130;

Faz sentido???

Views são normalmente criadas para apresentar
subconjuntos de dados presentes em tabelas reais

View with check option

- Não permite que as operações violem a **condição de filtro** (where) definida na visão

```
CREATE OR REPLACE VIEW ProdutodescA AS  
  SELECT codprod, descricao  
  FROM produto  
  WHERE descricao like 'A%'  
WITH CHECK OPTION;
```

```
Select * from produtodesca;
```

- Verifique:
 - Insert into ProdutodescA values (40, 'Manteiga');
 - Insert into ProdutodescA values (41, 'Azeite');

Resumo

- Uma visão de uma **única tabela base** é atualizável se a visão contiver, entre seus atributos, o de **chave primária**
 - Deve conter também **todos os atributos com restrição NOT NULL** que não tenham valores default especificados
- As visões definidas a partir de diversas tabelas utilizando-se **JOINS**, ou operações de conjunto, em geral, **não são atualizáveis**
- As visões definidas usando-se **agrupamento e funções de agregação** não são atualizáveis

Lembrar...

➤ A view é um meio...

Select nome from **vendsal** v join pedido p on
v.codigo = p.codvend;

≡

Select nome from **vendedor** v join pedido p on
v.codvend = p.codvend;

Mais DDL

Alter view nomeview1 rename to nomeview2;

e.g., **Alter view** vendsal rename to vendsal2;

DROP VIEW <nome da view>;

e.g., **drop view** vendsal2;

Vantagens no Uso de Visões

- **Salva consultas** (bastante complexas)
 - **“consultas nomeadas”**
 - E.g., visão a partir de um join de várias tabelas
- **Simplifica sentenças** para o usuário/desenvolvedor
 - Ex: Select * from visãoempregado;
- **Provê perspectiva diferente** da tabela base
 - **Pode-se renomear as colunas**



Vantagens no Uso de Visões

- **Isola aplicações** das mudanças sobre tabelas bases
 - Acrescentar uma coluna à tabela base não altera a *view*



**** Provê segurança e confidencialidade**

- **Restringe o acesso** a um conjunto predeterminado de colunas e/ou linhas de uma tabela
 - **Esconde dados**

Segurança de dados

Comando WITH

```
[ WITH [recursive] <common_table_expression>
[ ,...n ] ]
<common_table_expression> ::= expression_name
[ ( column_name [ ,...n ] ) ] AS (
CTE_query_definition )
```

- ***CTE_query_definition***

- Especifica uma instrução SELECT cujo conjunto de resultados popula a expressão de tabela comum

Exemplo

`select * from cliente;`

With ClientesAtivos AS

(SELECT codcli from Cliente WHERE endereco is not null),

ClientesInativos AS

(SELECT codcli from Cliente WHERE endereco is null)

SELECT * FROM ClientesAtivos

UNION

SELECT * FROM ClientesInativos