



Exercício: Visões Virtuais e Materializadas

Objetivos: Criação e atualização de visões.

Data da Entrega: 02/12/2025

## **Criação e Atualização de Visões**

### **1 - INTRODUÇÃO**

Uma visão é uma maneira alternativa para a visualização dos dados derivados de uma ou mais tabelas em um banco de dados. Tem por objetivo disponibilizar dados para o usuário final com segurança.

O usuário é habilitado a visualizar partes selecionadas das informações no banco de dados. As visões também permitem que você defina informações calculadas e derivadas.

As visões apresentam as seguintes vantagens:

- Permitem ao usuário visualizar somente os dados do seu interesse. Dados que não são necessários ou úteis podem ser deixados de fora da visão.
- Simplifica a manipulação dos dados pelo usuário.
- Otimiza as visões que os usuários têm do banco de dados. Isto permite que diferentes usuários visualizem as informações no banco de dados de maneira específica para as suas necessidades.
- Provê segurança, pois permite que os usuários consultem e modifiquem somente os dados que vêm; o restante do banco de dados não é visível nem acessível.
- Podemos ter uma visão de visões.

As visões apresentam as seguintes restrições de sintaxe:

- Tem que ter a clausula *SELECT*.
- Não podemos usar: *Select Into*, *Order By*, *Compute*, *Compute By* ou *Union*.
- Não podemos referenciar tabelas temporárias.
- Não podemos criar *triggers* para serem disparados pelas visões.
- Não podemos criar índices (Alguns SGBDs já permitem).
- Máximo de 250 colunas.

A modificação dos dados através de visões apresenta (em geral) as seguintes restrições.

- As modificações não podem afetar mais de um objeto básico (tabela).
- As modificações não podem ser feitas em determinadas colunas, como por exemplo:
  - Valores calculados
  - *Built-in functions*
  - *Row aggregate functions*
- As modificações que afetam tabelas com colunas *Not Null* podem causar erros.

Obs: As colunas fora da visão devem permitir valores Nulos ou *Defaults*.

Obs: Se você define uma visão com a cláusula *SELECT \** e depois altera a estrutura da tabela base adicionando colunas, as novas colunas não aparecerão na visão. O asterisco é interpretado e expandido quando a visão é inicialmente criada. Para ver as novas colunas através da visão, execute um drop na visão e crie-a novamente.

Obs: Se a visão referencia uma tabela que tenha sido *dropped*, você receberá uma mensagem de erro.

## **2. ESQUEMA DE TRABALHO**

### ***empregado***

ename	cpf	endereco	nasc	sexo	salario	chefe	cdep
Chiquin	1234	rua 1, 1	02/02/62	M	10000,00	8765	3
Helenita	4321	rua 2, 2	03/03/63	F	12000,00	6543	2
Pedrin	5678	rua 3, 3	04/04/64	M	9000,00	6543	2
Valtin	8765	rua 4, 4	05/05/65	M	15000,00	Null	4
Zulmira	3456	rua 5, 5	06/06/66	F	12000,00	8765	3
Zefinha	6543	rua 6, 6	07/07/67	F	10000,00	8765	2

### ***departamento***

dnome	codigo	gerente
Pesquisa	3	1234
Marketing	2	6543
Administração	4	8765

### ***projeto***

pnome	pcodigo	cidade	cdep
ProdutoA	PA	Cumbuco	3
ProdutoB	PB	Icapuí	3
Informatização	Inf	Fortaleza	4
Divulgação	Div	Morro Branco	2

***tarefa***

<b>cpf</b>	<b>pcodigo</b>	<b>horas</b>
1234	PA	30.0
1234	PB	10.0
4321	PA	5.0
4321	Div	35.0
5678	Div	40.0
8765	Inf	32.0
8765	Div	8.0
3456	PA	10.0
3456	PB	25.0
3456	Div	5.0
6543	PB	40.0

**dunidade**

dcodigo	dcidade
2	Morro Branco
3	Cumbuco
3	Prainha
3	Taíba
3	Icapuí
4	Fortaleza

**3 - EXERCÍCIO**

3.1. Crie as visões solicitadas a seguir.

3..2. Para cada uma das operações de atualização (*update, insert e delete*), sobre as visões criadas, solicitadas a seguir, indique:

- a) Se a atualização foi permitida (ou não) na visão. Justifique sua resposta.
- b) Caso a atualização seja permitida, especifique qual a tradução da atualização nas relações de base correspondente.
- c) Caso a atualização não seja permitida, justifique o motivo e indique como seria a semântica desta atualização nas tabelas base correspondentes.
- d) Desafio Opcional: Crie *Stored Procedures* para implementar as atualizações que não forem permitidas.

## **4 - TAREFAS**

- 4.1. Crie uma visão que recupere para cada empregado o seu nome, o seu sexo e o seu salário.
- 4.2. Crie uma consulta usando a visão anterior que recupere o nome do funcionário com maior salário.
- 4.3. Usando a visão anterior, efetue uma atualização que conceda um aumento de salário de 10%.
- 4.4. Crie uma visão que recupere para cada departamento o seu nome e o nome do seu gerente.
- 4.5. Usando a visão anterior, altere o nome do gerente do departamento de informática.
- 4.6. Usando a visão anterior, altere o nome do departamento do empregado Chiquim.
- 4.7. Crie uma visão que recupere para cada empregado com salário menor que 500: o seu nome, o seu sexo e o seu salário.
- 4.8. Usando a visão anterior, efetue uma atualização que conceda um aumento de 600 a todos os empregados que aparecem na visão.
- 4.9. Ainda usando esta visão exclua os empregados com salário entre 1000 e 2000.
- 4.10. Agora inclua uma nova *tupla* usando a visão anterior.
- 4.11. Crie uma visão que recupere para cada departamento o seu código, o seu nome, além do CPF e do nome do seu gerente.
- 4.12. Usando a visão anterior altere o código do departamento de informática.
- 4.13. Agora altere o CPF do empregado Chiquim.
- 4.14. Ainda usando esta visão altere o nome do empregado Chiquim.
- 4.15. Crie uma visão que recupere para cada departamento o seu código, o seu nome, a quantidade de empregados, o maior salário do departamento, o menor salário do departamento e a média salarial do departamento.
- 4.16. Utilizando a visão anterior:
  - Altere a média salarial do departamento de código igual a 1.
  - Altere a quantidade de empregados do departamento de código 1.
  - Exclua o departamento de código 1.
  - Insira um novo departamento.
  - Altere o nome de um dos departamentos que aparecem na visão.