



**Universidade Federal de Sergipe**  
**Departamento de Sistemas de Informação**  
Banco de Dados II



# Visões em SGBDs

---

André Vinicius R. P. Nascimento  
[andreviniciusnascimento@gmail.com](mailto:andreviniciusnascimento@gmail.com)



# Conteúdo

---

- Introdução
- Definição de Visões
- Principais Objetivos
- Removendo Visões
- Atualizando Visões
- Modificando dados através de Visões
- Visões Materializadas



# Introdução

---

- O que é uma Visão em um SGBD ?
  - Representa uma tabela virtual derivada de outras tabelas básicas ou mesmo de outras visões.
  - Essa derivação é alcançada através de uma consulta SQL.



# Definição

- O comando SQL CREATE VIEW é utilizado para a definição de Visões (Views)

```
CREATE VIEW VW_LOJA
```

```
AS
```

```
SELECT CD_LOJA, NM_LOJA, CD_CATEGORIA
```

```
FROM TB_LOJA
```

TB_LOJA
---------

CD_LOJA	NM_LOJA	CD_CATEGORIA	ESTADO
10	CENTRO	1	SE
20	SUL	2	SE
30	NORTE	3	PB



# Definição

---

- A visão VW\_LOJA comporta-se, para consultas, como uma tabela normal.

```
SELECT CD_LOJA, NM_JOJA  
FROM VW_LOJA
```

```
SELECT NM_LOJA  
FROM VW_LOJA  
WHERE CD_CATEGORIA = 1
```



# Definição

---

- A visão VW\_LOJA não existe fisicamente como uma tabela.
- Seus dados ou sua consulta existem somente no momento de utilização da visão.

```
CREATE VIEW VW_LOJA
AS
SELECT CD_LOJA, NM_LOJA, CD_CATEGORIA
FROM TB_LOJA
```



# Definição

---

- Podemos utilizar junções entre visões e tabelas, e entre visões e visões.

```
SELECT CD_LOJA, NM_LOJA, NM_CATEGORIA  
FROM  
VW_LOJA VL JOIN TB_CATEGORIA C  
ON (VL.CD_CATEGORIA = C.CD_CATEGORIA)
```



# Definição

- O comando CREATE VIEW pode ser utilizado com atributos.

```
CREATE VIEW VW_LOJA (CODIGO, NOME, CATEGORIA)
AS
SELECT CD_LOJA, NM_LOJA, CD_CATEGORIA
FROM TB_LOJA
```

TB\_LOJA

```
SELECT CODIGO, NOME
FROM VW_LOJA
```

CD_LOJA	NM_LOJA	CD_CATEGORIA
10	CENTRO	1
20	SUL	2
30	NORTE	3





# Principais Objetivos

---

- Segurança
  - Ocultar informações que não sejam relevantes para um usuário ou aplicação.
- Transparência ou Baixo Acoplamento entre aplicações.
- Combinação e Otimização (visões particionadas)
- Simplicidade
  - Criar consultas pré-definidas que omitam a complexidade dos esquemas de dados.



# Removendo Visões

---

- Utilizamos o comando DROP VIEW

```
DROP VIEW VW_LOJA
```



# Atualizando Visões

---

- Utilizamos o comando ALTER VIEW

```
ALTER VIEW VW_LOJA  
AS  
SELECT NM_LOJA, CATEGORIA  
FROM TB_LOJA  
WHERE ESTADO = 'SE'
```

- Por que não utilizar DROP e CREATE VIEW ?



# Modificando dados em Visões

---

- A priori, o objetivo principal das Visões (Views) é servir de base para consultas.
- No entanto, algumas visões podem também ser utilizadas para modificação de dados.
- Quais visões não podem sofrer atualizações ?



# Modificando dados em Visões

---

```
CREATE VIEW VW_LOJA
```

```
AS
```

```
SELECT CD_LOJA, NM_LOJA, CD_CATEGORIA,  
        ESTADO
```

```
FROM TB_LOJA
```

```
INSERT INTO VW_LOJA
```

```
VALUES (40, 'SUL', 2, 'AL')
```

```
UPDATE VW_LOJA
```

```
SET NM_LOJA = 'SUDESTE'
```

```
WHERE CD_LOJA = 40
```



# Visões Materializadas

---

- Atualmente existe um conceito muito importante para aplicações que necessitam de muito desempenho para consultas complexas.
- Esse conceito é conhecido como Visões Materializadas ou Agregados



# Visões Materializadas

---

- Visões Materializadas não são efetivamente Visões.
- Representam consultas que, por sua complexidade, terão seus dados fisicamente armazenados em tabelas reais.



# Visões Materializadas

---

- Considere a seguinte tabela.

```
CREATE TABLE TB_VENDAS (  
    CODIGO INT NOT NULL PRIMARY KEY,  
    DATA_VENDA DATETIME NOT NULL,  
    MATRICULA INT NOT NULL,  
    COD_PRODUTO INT NOT NULL,  
    QUANTIDADE INT NOT NULL,  
    VALOR_UNITARIO NUMERIC(10,2) NOT NULL,  
    VALOR_TOTAL NUMERIC(10,2)  
)
```





# Visões Materializadas

---

- Imagine que essa tabela possui alguns milhões de registros e que precisássemos de uma consulta que apresentasse o total de vendas por produto por ano.
- Poderíamos Criar uma Visão ?
- Isso ajudaria a melhorar o desempenho da nossa consulta ?



# Visões Materializadas

---

- Qual seria a solução ?
  - Criar uma Visão Materializada

```
CREATE TABLE TB_VENDAS_ANUAL (  
    COD_PRODUTO INT NOT NULL,  
    ANO INT NOT NULL,  
    QUANTIDADE INT NOT NULL,  
    VALOR_TOTAL NUMERIC(10,2)  
)
```



# Visões Materializadas

---

- Quando a Visão Materializada deve ser atualizada?
  - Por não representar uma consulta operacional, não existe a necessidade que ela seja atualizada a cada modificação da tabela de Vendas.
  - A atualização pode ser feita uma vez ao dia através de um job.



# Referências

---

- Visões

- Natavathe 4ª Edição – Capítulo 9, Seção 9.2
- Silberschatz 5ª Edição – Capítulo 3, Seção Views