

Banco de dados

**Visões**

Prof. Eldane Vieira

# Introdução

- Quando precisamos trabalhar com os dados de um Banco de Dados, montamos consultas para termos as informações desejadas.
- Em alguns casos não é conveniente que o usuário ou pessoas não autorizadas tenham acesso a todos os dados diretamente, para isso podemos utilizar visões ou *views*.

# Definição

- Uma visão é um procedimento fornecido pelos SGBDs para que possamos trabalhar com tabelas virtuais, que não armazenam dados.
- Com a visão, somente a definição da tabela virtual é armazenada, ou seja, ela não ocupa espaço em disco.

# Definição

- Ao trabalhar com visão, temos a impressão de lidar com uma tabela real, pois o conteúdo e o comportamento da visão é o mesmo de uma tabela real.
- Uma visão pode referenciar a tabela resultante de uma consulta ou de uma tabela derivada de uma visão ou de algumas tabelas.
- Em uma visão são somente apresentados os dados relevantes, e não toda a base de dados.

# Definição

- Os dados apresentados mediante a visão podem ser alterados a todo instante no banco de dados, sendo assim, o conteúdo o conteúdo da visão reflete as informações daquele determinado momento.
- Trabalhamos com visões da mesma maneira como manipulamos tabelas reais.

# Vantagens

- Possibilidade de se ter um acesso limitado e controlado dos dados, uma vez que não estamos trabalhando com esses dados diretamente nas tabelas;
- Uma performance mais rápida, visto que trabalhamos com uma consulta previamente pronta;
- Segurança dos dados, pois por meio das visões é possível restringir o acesso do usuário a somente os dados apresentados na visão.
- Reduz o número de repetições de comandos SQL, como consultas que são feitas constantemente.

# Criando Visões

- O primeiro passo é definir uma visão conforme a seguinte sintaxe:
  - **CREATE VIEW <nome da visão> [(campos)]  
AS <comando SELECT>;**
  - A declaração de campos é opcional, mas se for feita, ela servirá como uma renomeação dos atributos do comando SELECT.

# Esquema de tabelas

- Os exemplos nesta aula serão feitos sobre o seguinte esquema de tabela (essa base está disponível no disco virtual):
  - Cliente (ID\_Cliente, nome, endereco, cidade, telefone, tipo)
  - Empresa (ID\_Empresa, CNPJ, nome, endereco, cidade, telefone)
  - Cliente\_Empresa (ID\_Empresa, ID\_Cliente)
  - Produto (ID\_Produto, nome, qnt\_vendida, valor, total\_produzido)
  - Produto\_Empresa(ID\_Produto,ID\_Empresa)
  - Produto\_Comprado\_Cliente(ID\_Produto,ID\_Cliente, quantidade)



# Exemplo 1

- Visão MenorValorProduto: visão criada com base na consulta que apresenta o menor valor de um produto e seu nome.
  - CREATE VIEW MenorValorProduto AS  
select nome, valor  
from produto  
where valor = (select min(valor)  
from produto);
- Consulta na view: Lista o nome e o produto de menor valor.
  - select \*  
from MenorValorProduto;

## Exemplo 2

- Visão ClientesCompradores: visão criada com base na consulta que contém os clientes que já compraram algum produto.
  - CREATE VIEW ClientesCompradores AS  
select distinct cliente.nome  
from cliente, produto\_comprado\_cliente  
where cliente.ID\_Cliente = produto\_comprado\_cliente.ID\_Cliente;
- Consulta na view: Apresente o nome de todos os cliente compradores.
  - select \*  
from ClientesCompradores;

# Exemplo 3

- Visão ProdutosComprados: visão criada com base na consulta que contém os produtos já comprados ordenados pelo cliente.
  - CREATE VIEW ProdutosComprados AS  
select c.nome AS Cliente, p.nome AS Produto  
from cliente c, produto p, produto\_comprado\_cliente pcp  
where c.ID\_Cliente = pcp.ID\_Cliente AND  
p.ID\_Produto = pcp.ID\_Produto  
order by c.nome;
- Consulta: Apresente o nome dos produtos já comprados por algum cliente.
  - select Produto  
from ProdutosComprados;

# Alterando uma Visão

- Para alterar uma visão deve-se seguir a seguinte sintaxe:
  - ALTER VIEW <nome da visão>  
AS <comando SELECT>;

# Modificação no banco de dados via visão

- Se for necessário, pode ser usado os comandos INSERT, DELETE e UPDATE na visão com o objetivo de alterar a base de dados. Mas a visão deve respeitar o esquema da tabela base, como o tipo de dados do atributo, restrições de chaves primárias e estrangeiras.
- Contudo, para fazer a modificação, é necessário que se use a cláusula WITH CHECK OPTION ao final do comando de criação da visão.
- Essa cláusula permite que o sistema verifique a possibilidade da atualização indicada na visão.
- Com esta cláusula as operações sobre a visão podem ser rejeitadas se violarem qualquer restrição de integridade da tabela base.

# Exemplo de Inserção de dados na Visão

- `CREATE TABLE t1 (a INT);`
- `CREATE VIEW v1 AS SELECT * FROM t1 WHERE a < 2`  
`WITH CHECK OPTION;`
- Depois de criada a visão insira um dado na visão:
  - `INSERT INTO v1 VALUES (1);`
    - Essa inserção acontece na visão v1 e na tabela t1.
- Exemplo de deleção:
  - `delete from v1 where a=1 ;`
    - Se preciso execute: `SET SQL_SAFE_UPDATES = 0;` (Pode ser necessário para permitir a deleção)
    - Essa deleção acontece na visão v1 e na tabela t1.

# Excluindo uma Visão

- Para excluir uma visão deve-se seguir a seguinte sintaxe:
  - DROP VIEW <nome da visão>;

# **Exercícios**



# Exercício 1

- A) Crie uma visão, para o esquema de tabelas apresentado, que contenha uma relação entre o ID\_Empresa, o nome das empresas, os produtos produzidos por elas e seus valores.
- B) Dê o comando que consulta a visão criada apresentando a quantidade de tipos de produtos produzidos por cada empresa.

```
Cliente (ID_Cliente, nome, endereco, cidade, telefone, tipo)
Empresa (ID_Empresa, CNPJ, nome, endereco, cidade, telefone)
Cliente_Empresa (ID_Empresa, ID_Cliente)
Produto (ID_Produto, nome, qnt_vendida, valor, total_produzido)
Produto_Empresa(ID_Produto,ID_Empresa)
Produto_Comprado_Cliente(ID_Produto,ID_Cliente, quantidade)
```

## Exercício 2

- A) Crie uma visão, para o esquema de tabelas apresentado, que apresenta uma relação entre as empresas e seus clientes, tendo na visão os IDs das empresas e dos clientes, além de seus respectivos nomes.
- B) Dê o comando que consulta a visão criada apresentando a quantidade de clientes de cada empresa.

Cliente ( <u>ID_Cliente</u> , nome, endereco, cidade, telefone, tipo)
Empresa ( <u>ID_Empresa</u> , CNPJ, nome, endereco, cidade, telefone)
Cliente_Empresa ( <u>ID_Empresa</u> , <u>ID_Cliente</u> )
Produto ( <u>ID_Produto</u> , nome, qnt_vendida, valor, total_produzido)
Produto_Empresa( <u>ID_Produto</u> , <u>ID_Empresa</u> )
Produto_Comprado_Cliente( <u>ID_Produto</u> , <u>ID_Cliente</u> , quantidade)

## Exercício 3

- A) Crie uma visão, para o esquema de tabelas dado, que apresente o nome dos produtos e a quantidade de cada produto em estoque.
- B) Dê o comando que consulta a visão criada apresentando os produtos que tenham pelo menos 80 unidades no estoque.

Cliente ( <u>ID_Cliente</u> , nome, endereco, cidade, telefone, tipo)
Empresa ( <u>ID_Empresa</u> , CNPJ, nome, endereco, cidade, telefone)
Cliente_Empresa ( <u>ID_Empresa</u> , <u>ID_Cliente</u> )
Produto ( <u>ID_Produto</u> , nome, qnt_vendida, valor, total_produzido)
Produto_Empresa( <u>ID_Produto</u> , <u>ID_Empresa</u> )
Produto_Comprado_Cliente( <u>ID_Produto</u> , <u>ID_Cliente</u> , quantidade)