

**Banco de Dados**

# **VISÕES**

(VIEW)

# Introdução

- **Visões** em SQL são consultas armazenadas em uma estrutura de fácil acesso baseadas num comando SELECT;
- Essa consulta armazenada funciona como uma **tabela virtual**, com comportamento similar a uma tabela real, entretanto, sem armazenar dados ou modificar as tabelas de origem da consulta;
- Os dados que são exibidos nas **visões** são gerados dinamicamente toda vez que a **visão** é consultada;

# Sintaxe do Comando

- ❑ **CREATE VIEW** nome\_da\_visão **AS**
- ❑ **SELECT**
- ❑ atributo1,
- ❑ atributo2
- ❑ **FROM**
- ❑ tabela1
- ❑ **WHERE**
- ❑ condição;

# Exemplo – BD\_Mecanica 5.0

- ❑ Armazena os dados das vendas de 2019:
- ❑ **CREATE VIEW *Vendas\_2019* AS**
- ❑ **SELECT**
- ❑ **cod\_vend as ID,**
- ❑ **data\_vend as Data\_Venda,**
- ❑ **valortotal\_vend as Valor\_Venda**
- ❑ **FROM**
- ❑ **Venda**
- ❑ **WHERE**
- ❑ **(data\_vend BETWEEN '2019-01-01' AND '2019-12-31');**



Nas VIEWS não coloque espaço no nome de atributos. Pois isso dificulta a consulta depois.

# Exemplo – BD\_Mecanica 5.0

- ❑ Armazena os dados das vendas de 2019 por vendedor:
- ❑ **CREATE VIEW Vendas\_Vendedor\_2019 AS**
- ❑ **SELECT**
- ❑ Venda.cod\_vend as ID,
- ❑ Venda.data\_vend as Data\_Venda,
- ❑ Venda.valortotal\_vend as Valor\_Venda,
- ❑ Funcionario.nome\_func as Vendedor
- ❑ **FROM**
- ❑ Venda, Funcionario
- ❑ **WHERE**
- ❑ (Venda.cod\_func\_fk = Funcionario.cod\_func) AND
- ❑ (data\_vend BETWEEN '2019-01-01' AND '2019-12-31');

# DESC VIEW

- Para verificar a estrutura de uma visão, é necessário utilizar o comando DESC conforme apresentado no destaque a seguir.
- **DESC nome\_da\_visão;**

# DROP VIEW

- Quando uma visão não é mais necessária, pode-se excluí-la utilizando o comando descrito abaixo.
- **DROP VIEW nome\_da\_visão;**

# SELECT VIEW

- A sintaxe para visualizar todos os dados de uma VIEW é descrita no destaque abaixo;
- **SELECT \* FROM nome\_da\_visão;**



# SELECT VIEW

- A resposta do sistema é exatamente a mesma dada pelo comando da **consulta** em si. A diferença é que com a utilização de VISÕES, o comando necessário para visualizar os dados é simples, tornando a consulta muito mais simples e prática.
- É importante ressaltar que o SGBD armazena apenas o nome da visão e o comando SELECT associado. Os dados visualizados nas visões são gerados dinamicamente toda vez que é solicitada uma consulta sobre a VIEW. Isso implica que uma visão está sempre atualizada, ou seja, ao se modificar dados nas tabelas referenciadas na descrição da visão, uma consulta a visão reflete automaticamente essas alterações.

# Select View

- Vale salientar que a consulta a uma visão pode ser realizada da mesma forma que uma consulta a uma tabela. O segredo é que a visão se comporta como uma tabela de verdade, no entanto, sem guardar os dados, por isso chamada de uma tabela virtual.
- **Exemplo:** Consulta as vendas de 2019 com valor superior a R\$ 455.
- **Código:** `SELECT * FROM vendas_2019 WHERE (valor_venda > 455);`



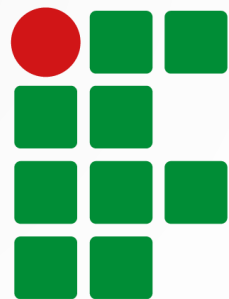
Nome da VIEW



Nome do atributo nomeado  
na VIEW

# Vantagens da VIEWS

- ❑ **Reuso:** as VIEWS são objetos de caráter permanente. Pensando pelo lado produtivo isso é excelente, já que elas podem ser lidas por vários usuários simultaneamente.
- ❑ **Segurança:** as VIEWS permitem que ocultemos determinadas colunas de uma tabela. Para isso, basta criarmos uma VIEW com as colunas que achamos necessário que sejam exibidas e as disponibilizarmos para o usuário.
- ❑ **Simplificação do código:** as VIEWS nos permitem criar um código de programação muito mais limpo, na medida em que podem conter um SELECT complexo. Assim, criar views para os programadores a fim de poupá-los do trabalho de criar SELECT's é uma forma de aumentar a produtividade da equipe de desenvolvimento.



**INSTITUTO  
FEDERAL**  
Rondônia

**Jackson Henrique**

Professor de Informática

**[Jackson.henrique@ifro.edu.br](mailto:Jackson.henrique@ifro.edu.br)**