



**Instituto Federal**  
Campus Goiânia

**Bacharelado em Sistemas de Informação**

# Banco de Dados II



**Prof. Dory Gonzaga Rodrigues**





## Agenda

- Objetos Avançados
  - VIEWS
  - ANÁLISE “TOP-N”





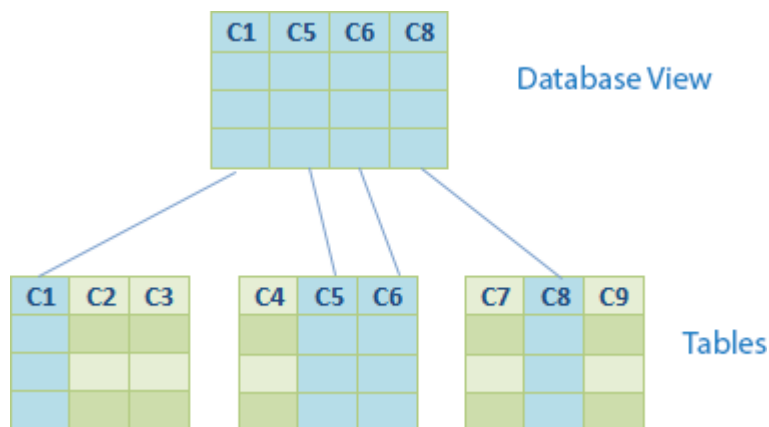
## OBJETOS AVANÇADOS

### SQL – AVANÇADA

#### - Objetos Avançados

Um banco de dados é composto por vários objetos.

Uma visão é um objeto do banco de dados. A View é uma “Tabela Virtual” que permite consultas através dela. Compreender o que é uma VIEW de banco de dados e usá-la corretamente é muito muito importante. Nesta aula, vamos discutir sobre o ponto de vista do banco de dados, como uma VIEW é implementada e como usá-la de forma mais eficaz.





## OBJETOS AVANÇADOS

### SQL – AVANÇADA

#### - Objetos Avançados: **VIEWS**

- Uma VIEW em um banco de dados representa logicamente subconjuntos de dados de uma ou mais tabelas;
- Uma VIEW é uma tabela lógica baseada em uma tabela ou outra view.
- Uma VIEW não contém dados próprios mas é como uma janela através da qual os dados das tabelas podem ser vistos ou alterados.
- As tabelas nas quais uma VIEW é baseada são chamadas tabelas-base.
- A VIEW é armazenada como uma instrução SELECT no dicionário de dados.
- Quando ocorre alteração nos dados de uma tabela, a VIEW reflete automaticamente as mudanças também.





## OBJETOS AVANÇADOS

### SQL – AVANÇADA

#### - Objetos Avançados: **VIEWS**

##### Por Que Usar Views?

- Para restringir o acesso a dados, ou seja, mostrar apenas o que deve ser mostrado;
- Para facilitar as consultas complexas, ou seja, implementamos as consultas e disponibilizamos os dados já tratados para as aplicações ou usuários finais;
- Para permitir a independência dos dados, busca de dados em diversas tabelas;
- Para apresentar diferentes visões dos mesmos dados, ou seja, a informação será apresentada da forma que interessa a cada usuário ou grupo de Usuário;
- Podemos ter colunas de campos derivados, ou seja, podemos ter colunas de dados calculados. Ex: Total da Nota;





## OBJETOS AVANÇADOS

### SQL – AVANÇADA

#### - Objetos Avançados: **VIEWS**

##### Por Que Usar Views?

- Compatibilidade com versões e sistemas legados. Mesmo com alterações nas tabelas do banco de dados (criação/alteração/exclusão) podemos criar visões com o mesmo esquema e tabelas dos sistemas legados;

##### Desvantagens

- Performance: consultas através de VIEWS pode ser lenta, especialmente se a exibição é criada com base em outros visões.
- Dependência entre tabelas: sempre que você alterar a estrutura das tabelas que participam da VIEW, você tem que mudar a visão também.





## OBJETOS AVANÇADOS

### SQL – AVANÇADA

#### - Objetos Avançados: **VIEWS**

As VIEW podem ser classificadas da seguinte forma:

Recurso	View Simples	View Complexa
Número de Tabelas	Uma	Duas ou mais
Contém Funções	NÃO	SIM
Contém Dados Agrupados	NÃO	SIM
Uso de comando DML através da VIEW	SIM	DEPENDE





## OBJETOS AVANÇADOS

### SQL – AVANÇADA

#### - Objetos Avançados: CRIANDO UMA VIEW SIMPLES

A sintaxe:

```
CREATE [OR REPLACE] VIEW nome_da_view AS  
    SELECT colunas  
    FROM tables  
    [WHERE conditions];
```

Opcional:

[OR REPLACE] recria a view se ela já existir!







## OBJETOS AVANÇADOS

### SQL – AVANÇADA

#### - Objetos Avançados: CRIANDO UMA VIEW SIMPLES

Exemplo: Crie uma visão que projete o nome do CD e o valor de venda dos CDs da gravadora com código igual a 2.

```
CREATE OR REPLACE VIEW vCD_Gravadora AS  
  SELECT Nome_CD, Preco_Venda  
  FROM CD  
  WHERE idGravadora = 2;
```





## OBJETOS AVANÇADOS

### SQL – AVANÇADA

#### - Objetos Avançados: CRIANDO UMA VIEW SIMPLES

Existem algumas regras no uso da instrução SELECT na criação de uma VIEW:

- O SELECT pode conter subconsulta na cláusula WHERE, mas não no FROM cláusula.
- O SELECT não pode fazer referência a variáveis, incluindo variável local, variável de usuário ou variável de sessão.
- O SELECT não pode se referir aos parâmetros de Prepared Statements.





## OBJETOS AVANÇADOS

### SQL – AVANÇADA

#### - Objetos Avançados: **CRIANDO UMA VIEW COM SUBCONSULTA**

Exercício: Crie uma visão que projete o código da gravadora, o nome do CD e o valor de venda do CD que estão acima da média de preço de venda da gravadora.





## OBJETOS AVANÇADOS

### SQL – AVANÇADA

#### - Objetos Avançados: CRIANDO UMA VIEW COM SUBCONSULTA

Exercício: Crie uma visão que projete o código da gravadora, o nome do CD e o valor de venda do CD que estão acima da média de preço de venda da gravadora.

```
SELECT idGravadora, Nome_CD, Preço_Venda  
FROM CD a  
WHERE Preço_Venda > (SELECT AVG(Preço_Venda)  
                      FROM cd b  
                      WHERE b.idgravadora = a.idgravadora)  
ORDER BY idGravadora;
```





## OBJETOS AVANÇADOS

### SQL – AVANÇADA

#### - Objetos Avançados: **CRIANDO UMA VIEW COMPLEXA**

Exercício: Crie uma visão que projete o nome da gravadora e a quantidade de CDs gravados por gravadora, mesmo que exista gravadoras sem CDs cadastrados.





## OBJETOS AVANÇADOS

### SQL – AVANÇADA

#### - Objetos Avançados: CRIANDO UMA VIEW COMPLEXA

Exercício: Crie uma visão que projete o nome da gravadora e a quantidade de CDs gravados por gravadora, mesmo que exista gravadoras sem CDs cadastrados.

```
CREATE OR REPLACE VIEW v_GravadoraQtdeCDs AS  
    SELECT NomeGravadora, COUNT(idCD) AS Qtde_CDs  
    FROM GRAVADORA left join CD using(idGravadora)  
    GROUP BY idGravadora;
```





## OBJETOS AVANÇADOS

### SQL – AVANÇADA

#### - Objetos Avançados: **CRIANDO UMA VIEW COMPLEXA**

Exercício: Crie uma visão que mostre as Gravadoras ordenadas pelo nome, com seus respectivos CDs com a quantidade de músicas e o tempo total das músicas.





## OBJETOS AVANÇADOS

### SQL – AVANÇADA

#### - Objetos Avançados: CRIANDO UMA VIEW COMPLEXA

Exercício: Crie uma visão que mostre as Gravadoras ordenadas pelo nome, com seus respectivos CDs com a quantidade de músicas e o tempo total das músicas.

```
CREATE OR REPLACE VIEW vInfoCDs AS
```

```
  SELECT G.NomeGravadora,  
         C.Nome_CD,  
         COUNT(F.idMusica) AS Faixas,  
         SUM(M.duracao)   AS Duracao
```

```
FROM CD C
```

```
INNER JOIN GRAVADORA G ON G.idGravadora = C.idGravadora
```

```
INNER JOIN FAIXA F     ON F.idCD         = C.idCD
```

```
INNER JOIN MUSICA M     ON F.idMusica     = M.idMusica
```

```
GROUP BY G.NomeGravadora, C.idCD
```

```
ORDER BY G.NomeGravadora ASC;
```







## OBJETOS AVANÇADOS

### SQL – AVANÇADA

#### - Objetos Avançados: SELECIONANDO DADOS DE UMA VIEW

Podemos utilizar o comando SELECT para realizar consultas em uma VIEW.

```
SELECT *  
FROM vTOP_CD_Gravadora;
```

```
SELECT idGravadora, Nome_CD, MAX(Venda) AS Venda  
FROM vTOP_CD_Gravadora;
```

```
SELECT *  
FROM vCD_Gravadora;
```

```
SELECT *  
FROM vInfoCDs;
```

