

Instituto Federal
Campus Goiânia

Bacharelado em Sistemas de Informação

Banco de Dados II



Prof. Dory Gonzaga Rodrigues





Agenda

- Subqueries
 - Retornando uma Linha
 - Retornando Múltiplas Linhas
 - Como colunas
 - Na cláusula FROM

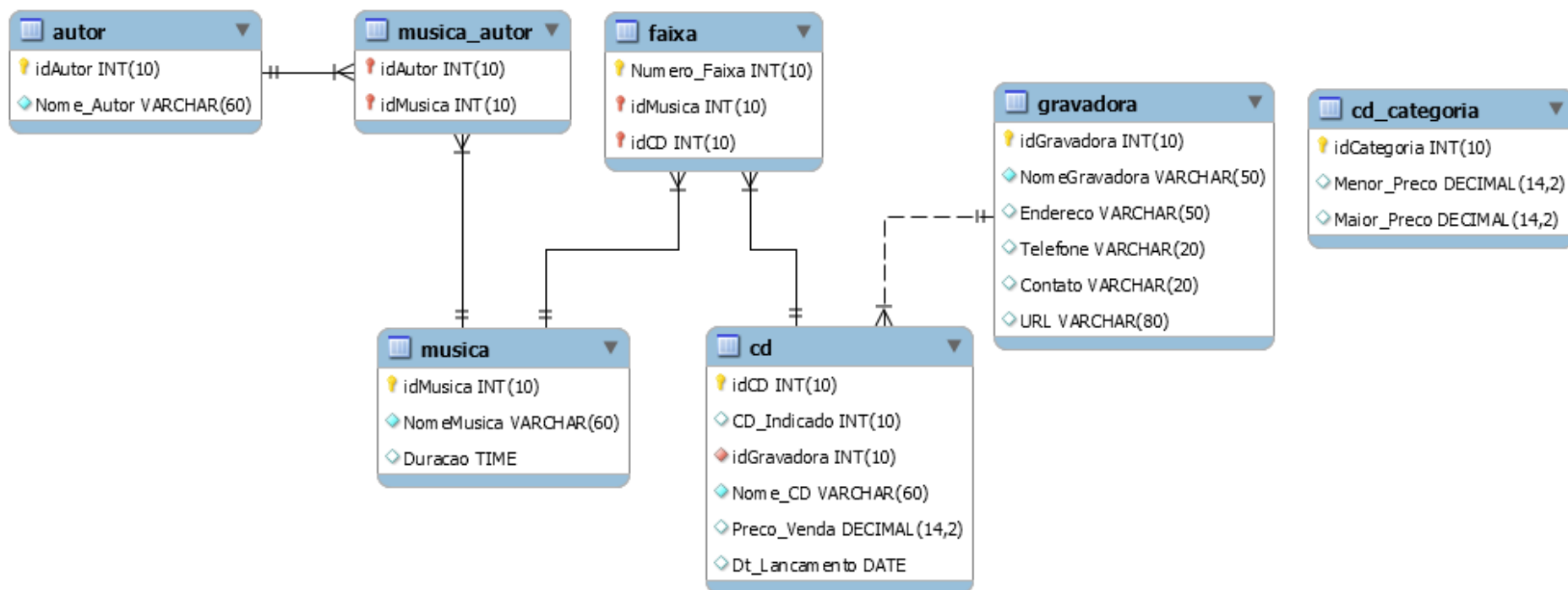




DML BANCO DE DADOS

BANCO DE DADOS REFERÊNCIA

- Utilizaremos nesta aula o banco de dados: Catálogo de CDs



Arquivo enviado por e-mail: [DDL_Catalogo_CDs.sql](#)
[DML_Catalogo_CDs.sql](#)





DML SELECT

SQL – DML

- Manipulação dos dados em Tabela: **SELECT**

O comando **SELECT** é sem sombra de dúvidas o comando mais utilizado e por isso o mais importante da linguagem SQL. Este comando permite selecionar os dados armazenados no Banco de Dados.

A sintaxe:

```
SELECT      [ DISTINCT | ALL ]      { * , coluna1, coluna1, ..., colunaN }  
FROM       nome da tabela  
[ ORDER BY <campo>      ]  
[ WHERE     <condição>  ]  
[ GROUP BY <campo>      ]  
[ HAVING    <condição>  ]
```

[] Opcional





DML SELECT SUBQUERIES

SQL – DML – Subqueries

- Subqueries é o uso de consultas aninhadas no comando SELECT

- Como esta funcionalidade faz parte do padrão SQL, praticamente todos os bancos de dados devem permitir sua utilização;
- Quais os tipos de retorno de uma subquery ?
 - o retorno do SELECT interno será de uma única linha;
 - o retorno do SELECT interno conterá mais de uma linha;
 - o retorno do SELECT interno conterá mais de uma linha e coluna;





DML SELECT SUBQUERIES

SQL – DML – Subqueries

- Subqueries é o uso de consultas aninhadas no comando SELECT

- Como podemos utilizar Subquery ?
 - como filtro de uma consulta principal;
 - como uma relação da consulta principal; e
 - como uma nova coluna da consulta principal.





DML SELECT SUBQUERIES

SQL – DML – Subqueries

- Atenção no uso de subqueries:

- A Subquery devem vir entre parênteses;
- A Subquery deve estar à direita do operador quando usada como filtro em uma consulta principal ;
 - Utilize operadores de linha (Ex: <, >, !=, ==) apenas em buscas que retornem uma única linha;
 - Utilize operadores de grupo (Ex: IN, ANY, ALL) apenas em buscas que potencialmente retornem mais de uma linha;
- A Subquery não pode conter a cláusula ORDER BY;





DML SELECT SUBQUERIES

SQL – DML – Subqueries

- Subquery como FILTRO da consulta principal e retornando apenas 01 valor:

- Quando o retorno do SELECT interno será de uma única linha

EXEMPLO

Retorne o nome e o preço dos CD's que estão acima da média de preço de venda dos CD's.





DML SELECT SUBQUERIES

SQL – DML – Subqueries

- Subquery como FILTRO da consulta principal e retornando apenas 01 valor:

- Quando o retorno do SELECT interno será de uma única linha

EXEMPLO

Retorne o nome e o preço dos CD's que estão acima da média de preço de venda dos CD's.

Resp: **SELECT** nome_CD, preco_venda
 FROM cd
 WHERE preco_Venda > (
 SELECT AVG(preco_venda)
 FROM cd
);





DML SELECT SUBQUERIES

SQL – DML – Subqueries

- Subquery como FILTRO da consulta principal e retornando apenas 01 valor:

UTILIZANDO cláusula HAVING

- Utilizamos a subquery da mesma forma que na cláusula WHERE;
- EXEMPLO
Retorne o código da Gravadora e o maior preço de venda dos CD's que estão acima da média de preço de venda da Gravadora.





DML SELECT SUBQUERIES

SQL – DML – Subqueries

- Subquery como FILTRO da consulta principal e retornando apenas 01 valor:

UTILIZANDO cláusula HAVING

- Utilizamos a subquery da mesma forma que na cláusula WHERE;

- EXEMPLO

Retorne o código da Gravadora e o maior preço de venda dos CD's que estão acima da média de preço de venda da Gravadora.

```
SELECT idgravadora, MAX(preco_venda)
FROM cd a
GROUP BY idgravadora
HAVING MAX(preco_venda) > ( SELECT AVG(preco_venda)
                             FROM cd
                             WHERE idgravadora = a.idgravadora
                           );
```





DML SELECT SUBQUERIES

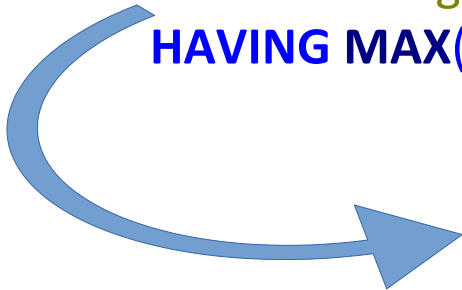
SQL – DML – Subqueries

- Subquery como FILTRO da consulta principal e retornando apenas 01 valor:

UTILIZANDO cláusula HAVING

- Observe que neste caso, estamos utilizando dados do primeiro SELECT (a.idgravadora) para realizar a média no segundo SELECT;

Resp: **SELECT** idgravadora, **MAX**(preco_venda)
FROM cd a
GROUP BY idgravadora
HAVING **MAX**(preco_venda) > (
 SELECT **AVG**(preco_venda)
 FROM cd
 WHERE idgravadora = a.idgravadora
);





DML SELECT SUBQUERIES

SQL – DML – Subqueries

- Subquery como **FILTRO** da consulta principal e podendo retornar **MAIS DE 01** valor:

- O SELECT INTERNO retorna mais de uma linha (tupla) e sempre uma coluna;
- O SELECT EXTERNO recebe valores que funcionam como uma lista, sendo assim:
 - Podemos utilizar a cláusula EXISTS
 - Podemos utilizar os operadores de grupo: IN, ALL e ANY
 - Já os operadores de linha não irão funcionar: = , > , < ou !=





DML SELECT SUBQUERIES

SQL – DML – Subqueries

- Subquery como **FILTRO** da consulta principal e podendo retornar **MAIS DE 01** valor:

UTILIZANDO cláusula EXISTS

- Na subqueries, a cláusula EXISTS verifica o número de linhas retornadas. Caso seja retornada uma ou mais linhas, então o resultado da primeira SELECT será apresentado;
- EXEMPLO
Retorne o código e o nome das gravadoras que possuem CD's cadastrados.





DML SELECT SUBQUERIES

SQL – DML – Subqueries

- Subquery como FILTRO da consulta principal e podendo retornar MAIS DE 01 valor:

UTILIZANDO cláusula EXISTS

- Na subqueries, a cláusula EXISTS verifica o número de linhas retornadas. Caso seja retornada uma ou mais linhas, então o resultado da primeira SELECT será apresentado;
- EXEMPLO

Retorne o código e o nome das gravadoras que possuem CD's cadastrados.

Resp: **SELECT** idGravadora, NomeGravadora

FROM gravadora g

WHERE EXISTS (SELECT *

FROM cd

WHERE cd.idGravadora = g.idGravadora

);





DML SELECT SUBQUERIES

SQL – DML – Subqueries

- Subquery como FILTRO da consulta principal e podendo retornar MAIS DE 01 valor:

UTILIZANDO cláusula EXISTS

- Os exemplos abaixo são equivalentes:

```
SELECT DISTINCT gravadora.idGravadora, gravadora.NomeGravadora  
FROM gravadora, cd  
WHERE gravadora.idGravadora = cd.idGravadora;
```

```
SELECT idGravadora, NomeGravadora  
FROM gravadora g  
WHERE EXISTS ( SELECT *  
                FROM cd  
                WHERE cd.idGravadora = g.idGravadora  
              );
```





DML SELECT SUBQUERIES

SQL – DML – Subqueries

- Subquery como **FILTRO** da consulta principal e podendo retornar **MAIS DE 01** valor:

UTILIZANDO operadores de grupo

- EXEMPLO

Retorne o código, o nome e o preço de venda dos CD's que têm o preço igual a um dos menores preço das gravadoras:





DML SELECT SUBQUERIES

SQL – DML – Subqueries

- Subquery como **FILTRO** da consulta principal e podendo retornar **MAIS DE 01** valor:

UTILIZANDO operadores de grupo

- EXEMPLO

Retorne o código, o nome e o preço de venda dos CD's que têm o preço igual ao menor preço de uma das gravadoras:

Passo:

1 – Encontramos o menor preço de venda de cada gravadora (lista de valores);

```
SELECT MIN(Preco_Venda)
FROM cd
GROUP BY idGravadora
```

2 – Depois encontramos os CDs com preços iguais a um dos valores da lista.

```
SELECT idCD, Nome_CD, Preco_Venda
FROM cd
WHERE Preco_Venda IN (9,10.5,15);
```





DML SELECT SUBQUERIES

SQL – DML – Subqueries

- Subquery como **FILTRO** da consulta principal e podendo retornar **MAIS DE 01** valor:

UTILIZANDO operadores de grupo

- EXEMPLO

Retorne o código, o nome e o preço de venda dos CD's que têm o preço igual ao menor preço de uma das gravadoras:

Resp:

```
SELECT idCD, Nome_CD, Preço_Venda
FROM cd
WHERE Preço_Venda IN (
    SELECT MIN(Preço_Venda)
    FROM cd
    GROUP BY idGravadora
);
```





DML SELECT SUBQUERIES

SQL – DML – Subqueries

- Subquery como **FILTRO** da consulta principal e podendo retornar **MAIS DE 01** valor:

UTILIZANDO operadores de grupo

- EXEMPLO

Retorne o nome do Cd e o número da faixa que contenha, no nome da música, a palavra “EU”:





DML SELECT SUBQUERIES

SQL – DML – Subqueries

- Subquery como FILTRO da consulta principal e podendo retornar MAIS DE 01 valor:

UTILIZANDO operadores de grupo

- EXEMPLO

Retorne o nome do Cd e o número da faixa que contenha, no nome da música, a palavra “EU”:

Passo:

1 – Encontrar a lista das músicas que possuem “EU” no nome da música;

```
SELECT idMusica FROM musica WHERE UPPER(nomeMusica) LIKE "%EU %";
```

2 – Depois encontramos quais são os CDs e as faixas que contenham tais músicas (lista com os id's das músicas):

```
SELECT idCD, Numero_Faixa  
FROM faixa  
WHERE idMusica IN (17,18,36,45,62);
```





DML SELECT SUBQUERIES

SQL – DML – Subqueries

- Subquery como FILTRO da consulta principal e podendo retornar MAIS DE 01 valor:

UTILIZANDO operadores de grupo

- EXEMPLO

Retorne o nome do Cd e o número da faixa que contenha, no nome da música, a palavra “EU”:

Resp:

```
SELECT cd.idCD, cd.Nome_CD, faixa.Numero_Faixa
```

```
FROM cd, faixa
```

```
WHERE faixa.idCD = cd.idCD
```

```
AND faixa.idMusica IN (
```

```
SELECT idMusica
```

```
FROM musica
```

```
WHERE UPPER(nomeMusica) LIKE '%EU %'
```

```
);
```





DML SELECT SUBQUERIES

SQL – DML – Subqueries

- Subquery como **FILTRO** da consulta principal e podendo retornar **MAIS DE 01** valor:

UTILIZANDO operador ANY

- O operador ANY permite o uso dos operadores simples (=, >, <, !=) com uma lista de valores;
- O operador ANY realiza a comparação para cada valor do SELECT EXTERNO com cada valor apresentado na lista retornada pelo SELECT INTERNO;
- Lista de Valores Especiais
 - = ANY : igual a algum dos elementos da lista (o mesmo que IN)
 - > ANY : maior que algum dos elementos da lista
 - < ANY: menor que algum dos elementos da lista
 - <> ANY : diferente de algum dos elementos da lista





DML SELECT SUBQUERIES

SQL – DML – Subqueries

- Subquery como FILTRO da consulta principal e podendo retornar MAIS DE 01 valor:

UTILIZANDO operador ANY

- EXEMPLO

Retorne o nome do CD e o preço de venda dos CD's que tenham preço de venda inferior a um dos cds da gravadora com código 3:

Resp:





DML SELECT SUBQUERIES

SQL – DML – Subqueries

- Subquery como FILTRO da consulta principal e podendo retornar MAIS DE 01 valor:

UTILIZANDO operador ANY

- EXEMPLO

Retorne o nome do CD e o preço de venda dos CD's que tenham preço de venda inferior a um dos cds da gravadora com código 3:

Resp:

```
SELECT idCD, cd.Nome_CD, cd.Preco_Venda, cd.idGravadora
FROM cd
WHERE idGravadora != 3
      AND Preco_Venda < ANY ( SELECT Preco_Venda
                              FROM cd
                              WHERE idGravadora = 3
                              );
```





DML SELECT SUBQUERIES

SQL – DML – Subqueries

- Subquery como **FILTRO** da consulta principal e podendo retornar **MAIS DE 01** valor:

UTILIZANDO operador ALL

- O operador ALL é utilizado em conjunto com os operadores simples: > e < ;
- O operador ALL realiza a comparação para cada valor do SELECT EXTERNO com cada valor apresentado na lista retornada pelo SELECT INTERNO;
- O resultado do SELECT EXTERNO são as linhas que atendam à condição definida para TODOS os valores da lista retornada pelo SELECT INTERNO;
- Lista de Valores Especiais
 - > ALL : maior que todos
 - < ALL : menor que todos
 - <> ALL : diferente de todos (igual a NOT IN)
 - != ALL : diferente de todos (igual a NOT IN)





DML SELECT SUBQUERIES

SQL – DML – Subqueries

- Subquery como FILTRO da consulta principal e podendo retornar MAIS DE 01 valor:

UTILIZANDO operador ALL

- EXEMPLO

Retorne o nome do CD e o preço de venda dos CD's que tenham preço de venda inferior à média de preço de venda de TODAS as gravadoras.

Resp:





DML SELECT SUBQUERIES

SQL – DML – Subqueries

- Subquery como FILTRO da consulta principal e podendo retornar MAIS DE 01 valor:

UTILIZANDO operador ALL

- EXEMPLO

Retorne o nome do CD e o preço de venda dos CD's que tenham preço de venda inferior à média de preço de venda de TODAS as gravadoras.

Resp: **SELECT** Nome_CD, Preço_Venda
FROM cd
WHERE Preço_Venda < **ALL** (

SELECT AVG(Preço_Venda)
FROM cd

);





DML SELECT SUBQUERIES

SQL – DML – Subqueries

- Subquery como **RELAÇÃO** da consulta principal:

- Podemos utilizar uma subquery na cláusula FROM do comando SELECT;
- Uma subquery na cláusula FROM, funciona como uma tabela.
- EXEMPLO
Retorne o nome do CD, o preço de venda e o preço médio da gravadora.





DML SELECT SUBQUERIES

SQL – DML – Subqueries

- Subquery como **RELAÇÃO** da consulta principal:

- Podemos utilizar uma subquery na cláusula FROM do comando SELECT;
- Uma subquery na cláusula FROM, funciona como uma tabela.
- EXEMPLO

Retorne o nome do CD, o preço de venda e o preço médio da gravadora.

```
SELECT cd.Nome_CD, cd.Preco_Venda, b.Preco_Medio
FROM cd, ( SELECT idGravadora, AVG(Preco_Venda) AS Preco_Medio
           FROM cd
           GROUP BY idGravadora
         ) b
WHERE cd.idGravadora = b.idGravadora
```





DML SELECT SUBQUERIES

SQL – DML – Subqueries

- Subquery como NOVA COLUNA da consulta principal:

- Podemos utilizar uma subquery como COLUNA do comando SELECT;
- EXEMPLO
Retorne o nome da Gravadora e a respectiva quantidade de CDs.





DML SELECT SUBQUERIES

SQL – DML – Subqueries

- Subquery como NOVA COLUNA da consulta principal:

- Podemos utilizar uma subquery como COLUNA do comando SELECT;
- EXEMPLO
Retorne o nome da Gravadora e a respectiva quantidade de CDs.

```
SELECT nomegravadora ,  
      (  
        SELECT COUNT(*)  
        FROM cd  
        WHERE cd.idgravadora = g.idgravadora  
      ) AS qtde_cds  
FROM gravadora g
```





DML SELECT SUBQUERIES

SQL – DML – Subqueries

Os exemplos abaixo são equivalentes:

```
SELECT nomegravadora ,  
      (  
        SELECT COUNT(*)  
        FROM cd  
        WHERE cd.idgravadora = g.idgravadora  
      ) AS qtde_cds  
FROM gravadora g
```

```
SELECT nomegravadora, COUNT(cd.*) AS qtde_cds  
FROM gravadora g  
LEFT JOIN cd USING(idgravadora)  
GROUP BY g.idgravadora
```

