

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE
Campus João Câmara

Disciplina:
Banco de Dados



AULA 13

Subqueries

Consultas aninhadas no SQL

Baseado no livro SQL: Curso Prático, do Celso Henrique.

Edmilson Campos, Prof.

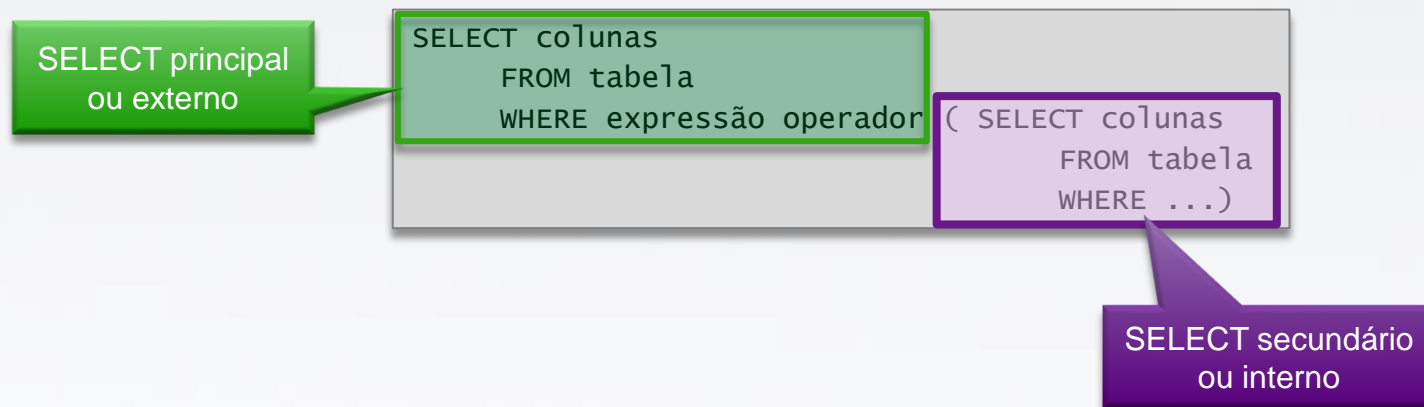
<http://docente.ifrn.edu.br/edmilsoncampos/>
edmilson.campos@ifrn.edu.br
[@edmilsonbcn](#)

Introdução

- Subconsultas ou *Subqueries* são usadas para realizar buscas a partir do resultados de outras buscas
- Na prática estaremos tratando de SELECTs aninhados
- Há três tipos de *subqueries*:
 - » *Subquery* de uma linha
 - O retorno do SELECT interno será uma única linha
 - » *Subquery* de múltiplas linhas
 - O retorno do SELECT interno será mais de uma linha
 - » *Subquery* de múltiplas colunas (só alguns SGBDs suportam)
 - O retorno do SELECT interno conterá mais de uma linha e coluna

Sintaxe geral

- Essa é a forma básica para se construir uma *subquery*:



Subquery de uma linha

- O resultado SELECT interno servirá de base para o SELECT principal
- Exemplos:

```
SELECT NOME_CD AS 'Nome do CD',  
       PRECO_VENDA AS Preço  
FROM CD  
WHERE PRECO_VENDA > (SELECT AVG(PRECO_VENDA) FROM CD)
```

Nome do CD	Preço
Mais do Mesmo	15.00
Elis Regina - Essa Mulher	13.00
A Força que nunca Seca	13.50
Eternamente	15.60



No exemplo é extraído com o *subquery* a média de preço de venda de todos os CDs e com esse resultado é exibido, pelo primeiro SELECT, apenas os CDs que possuírem preço maior que essa média

Subquery de uma linha

- O resultado SELECT interno servirá de base para o SELECT principal
- Exemplos:

```
SELECT CODIGO_GRAVADORA AS 'Código da Gravadora',  
       NOME_CD AS 'Nome do CD',  
       PRECO_VENDA AS Preço  
FROM CD a  
WHERE PRECO_VENDA > (SELECT AVG(PRECO_VENDA) FROM CD  
                     WHERE CODIGO_GRAVADORA = a.CODIGO_GRAVADORA)
```

Código da Gravadora	Nome do CD	Preço
2	Bate-Boca	12.00
2	A Força que nunca Seca	13.50
2	Eternamente	15.60
3	Elis Regina - Essa Mulher	13.00



Agora estamos buscando apenas os CDs que tenham o preço de venda superior à média de preço **da própria gravadora**. Isso ocorre porque foi colocado um apelido na tabela CD da primeira busca (**a**) que foi incluído na comparação do segundo SELECT com o código da gravadora

Utilizando *subquery* em cláusula HAVING

- Utilizado do mesmo modo que a cláusula WHERE em conjunto com um GROUP BY
- Exemplos:

```
SELECT CODIGO_GRAVADORA AS 'Código da Gravadora',  
       MIN(PRECO_VENDA) AS 'Menor Preço'  
FROM CD  
GROUP BY CODIGO_GRAVADORA  
HAVING MIN(PRECO_VENDA) > (SELECT PRECO_VENDA FROM CD  
                           WHERE CODIGO_CD = 6)
```

Código da Gravadora	Menor Preço
1	15.00
3	10.50
5	11.00



No exemplo o valor do CD de código 6, que é R\$ 9,50, está sendo usado como parâmetro para exibir apenas as gravadoras que possuem preços mínimos superiores a este

Utilizando *subquery* em cláusula HAVING

- Utilizado do mesmo modo que a cláusula WHERE em conjunto com um GROUP BY
- Exemplos:

```
SELECT CODIGO_GRAVADORA AS 'Código da Gravadora',  
       MAX(PRECO_VENDA) AS 'Preço máximo'  
FROM CD a  
GROUP BY CODIGO_GRAVADORA  
HAVING MIN(PRECO_VENDA) > (SELECT AVG(PRECO_VENDA) FROM CD  
                           WHERE CODIGO_GRAVADORA = CODIGO_GRAVADORA)
```

Código da Gravadora	Preço máximo
1	15.00
2	15.60
3	13.00



Agora, estamos buscando dados da primeira busca para realizar a busca do segundo SELECT. Assim, está sendo exibido apenas as gravadoras que possuem preços de venda superiores à média daquela gravadora

Utilizando o comando EXISTS

- O comando verifica o número de linhas retornadas pela *subquery* e caso ela contenha uma ou mais linhas, então o resultado será mostrado.
- Exemplo:

```
SELECT CODIGO_GRAVADORA AS Código,  
       NOME_GRAVADORA AS Gravadora  
FROM GRAVADORA  
WHERE EXISTS (SELECT * FROM CD  
              WHERE CD.CODIGO_GRAVADORA = GRAVADORA.CODIGO_GRAVADORA)
```

Código	Gravadora
1	EMI
2	BMG
3	SOM LIVRE
5	UOL



O comando SQL acima está mostrando apenas as gravadoras que têm CDs cadastrados, eliminando do resultado da busca as que não "existem" CDs

Subquery de múltiplas linhas

- Como o SELECT interna retorna mais de uma linha, não se pode utilizar operadores de igualdade simples
- Deve-se utilizar operadores de grupos para realizar a comparação, tais como:
 - » ANY
 - » ALL
 - » IN

Utilizando o operador IN

- Exemplos:

```
SELECT CODIGO_CD AS Código,  
       NOME_CD AS Nome,  
       PRECO_VENDA AS Preço  
FROM CD  
WHERE PRECO_VENDA IN (SELECT MIN(PRECO_VENDA) FROM CD  
                      GROUP BY CODIGO_GRAVADORA)
```

Está sendo exibido quais CDs têm preço igual ao menor preço de cada gravadora

Código	Nome	Preço
1	Mais do Mesmo	15.00
5	Perfill	10.50
7	Listen Without Prejudice	9.00
9	Ilusão	11.00

```
SELECT CODIGO_CD AS Código,  
       NUMERO_FAIXA AS Faixa  
FROM FAIXA  
WHERE CODIGO_MUSICA IN (SELECT CODIGO_MUSICA FROM MUSICA  
                        WHERE UPPER(NOME_MUSICA) LIKE '%AMOR%')
```

Agora estamos buscando os códigos dos CDs e faixas que contenham músicas com a palavra 'AMOR'

Código	Faixa
2	9
4	1

Utilizando o operador ANY

- Esse operador permite comparar operadores simples (=, >, <, !=) com um grupo de linha
- Exemplo:

```
SELECT CODIGO_CD AS Código,  
       NOME_CD AS Nome,  
       PRECO_VENDA AS Preço  
FROM CD  
WHERE PRECO_VENDA < ANY (SELECT PRECO_VENDA FROM CD  
                          WHERE CODIGO_GRAVADORA = 2)  
      AND CODIGO_GRAVADORA !=2
```

Exibir os CDs que têm preço inferior a qualquer outro da gravadora, mas que não sejam da gravadora 2

Código	Nome	Preço
1	Mais do Mesmo	15.00
3	Elis Regina - Essa Mulher	13.00
5	Perfill	10.50
9	Ilusão	11.00

Utilizando o operador ALL

- Este operador é utilizado em combinação com operadores simples (>, <) para que os valores retornados de todas as linhas do SELECT interno sejam comparados com o SELECT externo
- Exemplo:

```
SELECT CODIGO_CD AS Código,  
       NOME_CD AS Nome,  
       PRECO_VENDA AS Preço  
FROM CD  
WHERE PRECO_VENDA < ALL (SELECT AVG(PRECO_VENDA) FROM CD  
                        GROUP BY CODIGO_GRAVADORA)  
ORDER BY CODIGO_GRAVADORA
```

Exibir quais CDs têm preço de venda menor que a média de preço de venda de

Código	Nome	Preço
6	Barry Manilow Greatest Hits Vol I	9.50
7	Listen Without Prejudice	9.00
5	Perfill	10.50

Subquery na cláusula FROM

- Podemos utilizar uma subquery na cláusula FROM de um comando SELECT
- Exemplo:

```
SELECT a.NOME_CD AS 'Nome do CD',  
       a.Preco_Venda AS 'Preço de Venda',  
       b.Preco_Medio AS 'Preço Médio'  
FROM CD a, (SELECT CODIGO_GRAVADORA,  
                  AVG(PRECO_VENDA) AS PRECO_MEDIO  
            FROM CD  
            GROUP BY CODIGO_GRAVADORA) b  
WHERE a.CODIGO_GRAVADORA = b.CODIGO_GRAVADORA  
      AND a.PRECO_VENDA > b.PRECO_MEDIO
```

Exibir os nomes dos CDs com seus respectivos preços de venda e o preço médio da Gravadora de cada CD

Nome do CD	Preço de Venda	Preço Médio
Bate-Boca	12.00	11.920000
A Força que nunca Seca	13.50	11.920000
Eternamente	15.60	11.920000
Elis Regina - Essa Mulher	13.00	11.750000

Referências

- Oliveira, Celso Henrique Poderoso. SQL: Curso Básico. Editora Novatec: 2002.
- NAVATHE, S. B.; ELMASRI, R.. Sistemas de banco de dados – Fundamentos e aplicações. LTC: 2002.
- HEUSER, Carlos A. Projeto de Banco de Dados. Sagra Luzzato, 2009.