



Consulta em Banco de Dados

Subqueries

Consulta em Banco de Dados

Subqueries

- Subqueries (Subconsultas) - utilização de uma consulta dentro de outra;
- Na prática seria um SELECT dentro de outro;
- Forma básica:

SELECT colunas

FROM tabela WHERE expressão operador (SELECT colunas FROM tabela WHERE ...);

***I.** colunas - Colunas que serão mostradas;*

***II.** tabela - Nome da tabela;*

***III.** expressão - Condição ou predicado da busca;*

***IV.** operador - Pode ser de linha (>, <, >=, <=, =, <>) ou de grupo (IN, ANY, ALL);*

Consulta em Banco de Dados

Subqueries

- A subquery pode ser colocada na cláusula HAVING, como será visto adiante;
- São três tipos de subquery:
 - * 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: o retorno do SELECT interno conterá mais de uma linha e coluna;

Consulta em Banco de Dados

Subqueries

- Subquery de uma linha:
- * O resultado do SELECT mais interno servirá de base para o primeiro SELECT.;
- * Esta forma de consulta pode até melhorar o desempenho, pois uma alternativa seria realizar uma união regular de tabelas, que combina todas as linhas de ambas as tabelas.

Exemplos:

```
SELECT nome_cd, preco_venda  
FROM cd
```

```
WHERE preco_venda > (SELECT AVG(preco_venda) FROM cd);
```

```
SELECT codigo_gravadora, nome_cd, preco_venda FROM cd a  
WHERE preco_venda > (SELECT AVG(preco_venda) FROM cd WHERE  
codigo_gravadora = a.codigo_gravadora);
```

Consulta em Banco de Dados

Subqueries

- Para realizar subqueries tenha em mente:
 - * Coloque as subqueries entre parênteses;
 - * Coloque a subquery à direita do operador;
 - * Não coloque a cláusula ORDER BY em uma subquery. Não esqueça que deve haver apenas uma cláusula ORDER BY em todo o comando SELECT. Logo, se for necessário ordenar o resultado, faça isso no SELECT principal;
 - * Utilize operadores de linha apenas em buscas que retornem uma única linha;
 - * Utilize operadores de grupo apenas em buscas que potencialmente retornem mais de uma linha;
- ```
SELECT codigo_gravadora, nome_cd, preco_venda
FROM cd WHERE codigo_gravadora = (SELECT codigo_gravadora FROM cd
WHERE codigo_cd = 2) AND
preco_venda > (SELECT preco_venda FROM cd WHERE codigo_cd = 5);
```

# Consulta em Banco de Dados

## Subqueries

- Utilizando subquery em cláusula HAVING:
- \* A subquery é executada primeiro e o resultado da busca servirá de base para filtrar as linhas do GROUP BY.

```
SELECT codigo_gravadora, MIN(preco_venda)
FROM cd GROUP BY codigo_gravadora HAVING MIN (preco_venda) > (SELECT
preco_venda FROM cd WHERE codigo_cd = 6);
```

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

# Consulta em Banco de Dados

## Subqueries

- EXISTS

- \* Verifica o número de linhas retornadas pela subquery. Caso contenha uma ou mais linhas, então o resultado será mostrado. Do contrário, não o será.

*SELECT codigo\_gravadora, nome\_gravadora FROM gravadora WHERE EXISTS (SELECT \* FROM cd WHERE cd.codigo\_gravadora = gravadora.codigo\_gravadora);*

- Subquery de multiplas linhas

- \* Neste caso, o SELECT interno retorna mais de uma linha. Não se pode utilizar operadores simples como igualdade, diferença, maior ou menor. Deve-se utilizar um operador de grupo para realizar a comparação. Esses operadores são ANY, ALL e IN.





# Consulta em Banco de Dados

## Subqueries

- IN

*SELECT MIN(preco\_venda) FROM cd GROUP BY codigo\_gravadora;*

*SELECT codigo\_cd, nome\_cd, preco\_venda FROM cd WHERE preco\_venda IN (15, 9, 10.5);*

*SELECT codigo\_cd, nome\_cd, preco\_venda FROM cd WHERE preco\_venda IN (SELECT MIN(preco\_venda) FROM cd GROUP BY codigo\_gravadora);*

*SELECT codigo\_musica FROM musica WHERE UPPER (nome\_musica) LIKE '%AMOR%';*

*SELECT codigo\_cd, numero\_faixa FROM faixa WHERE codigo\_musica IN (25, 41);*

*SELECT codigo\_cd, numero\_faixa FROM faixa WHERE codigo\_musica IN (SELECT codigo\_musica FROM musica WHERE UPPER (nome\_musica) LIKE '%AMOR%');*





# Consulta em Banco de Dados

## Subqueries

- ANY

*SELECT preco\_venda FROM cd WHERE codigo\_gravadora = 2;*

*SELECT codigo\_cd, nome\_cd, preco\_venda FROM cd WHERE preco\_venda < ANY (12, 13.5, 9.5, 9) AND codigo\_gravadora != 2;*

*SELECT codigo\_cd, nome\_cd, preco\_venda FROM cd WHERE preco\_venda < ANY (SELECT preco\_venda FROM cd WHERE codigo\_gravadora = 2) AND codigo\_gravadora != 2;*

*SELECT codigo\_cd, nome\_cd, preco\_venda FROM cd WHERE preco\_venda = ANY (SELECT preco\_venda FROM cd WHERE codigo\_gravadora = 2);*



# Consulta em Banco de Dados

## Subqueries

- **ALL**

*SELECT AVG(preco\_venda) FROM cd GROUP BY codigo\_gravadora;*

*SELECT codigo\_cd, nome\_cd, preco\_venda FROM cd WHERE preco\_venda < ALL (15, 11, 11.75);*

*SELECT codigo\_cd, nome\_cd, preco\_venda FROM cd WHERE preco\_venda < ALL(SELECT AVG(preco\_venda) FROM cd GROUP BY codigo\_gravadora);*

- **Subquery de múltiplas colunas**

*SELECT codigo\_gravadora || MIN(cd\_indicado) FROM cd GROUP BY codigo\_gravadora;*

*SELECT codigo\_cd, nome\_cd, codigo\_gravadora, cd\_indicado FROM cd WHERE codigo\_gravadora || cd\_indicado IN (15, 21, 31);*

# Consulta em Banco de Dados

## Subqueries

- Subquery de múltiplas colunas

```
SELECT codigo_cd, nome_cd, codigo_gravadora, cd_indicado FROM cd WHERE
codigo_gravadora || cd_indicado IN (SELECT codigo_gravadora ||
MIN(cd_indicado) FROM cd GROUP BY codigo_gravadora);
```

```
SELECT codigo_cd, nome_cd, codigo_gravadora, cd_indicado FROM cd WHERE
(codigo_gravadora, cd_indicado) IN (SELECT codigo_gravadora, MIN(cd_indicado)
FROM cd GROUP BY codigo_gravadora);
```

- \* NULL - Quando um SELECT interno contiver valores nulos, não deve ser utilizado NOT IN como operador de comparação. Isso porque qualquer comparação com nulo retorna valor nulo.

```
SELECT codigo_cd, nome_cd, cd_indicado FROM cd WHERE cd_indicado IN
(SELECT cd_indicado FROM cd);
```

```
SELECT codigo_cd, nome_cd, cd_indicado FROM cd WHERE cd_indicado NOT IN
(SELECT cd_indicado FROM cd);
```

# Consulta em Banco de Dados

## Subqueries

- Subquery de múltiplas colunas

- ✱ Pares de comparação - Sempre que são realizadas comparações em subquery de múltiplas colunas, devemos saber se realmente o que queremos é a comparação entre pares de colunas, ou não.

*SELECT codigo\_cd, nome\_cd, cd\_indicado, codigo\_gravadora FROM cd WHERE (codigo\_gravadora, cd\_indicado) IN (SELECT codigo\_gravadora, cd\_indicado FROM cd WHERE codigo\_cd = 3);*

*SELECT codigo\_cd, nome\_cd, cd\_indicado, codigo\_gravadora FROM cd WHERE codigo\_gravadora IN (SELECT codigo\_gravadora FROM cd WHERE codigo\_cd = 3) OR cd\_indicado IN (SELECT cd\_indicado FROM cd WHERE codigo\_cd = 3);*



# Consulta em Banco de Dados

## *Subqueries*

- Subquery na cláusula FROM

✱ Pode-se utilizar uma subquery cláusula FROM de um comando SELECT.

```
SELECT a.nome_cd, a.preco_venda, b.preco_medio FROM cd a, (SELECT
codigo_gravadora, AVG(preco_venda) preco_medio FROM cd GROUP BY
codigo_gravadora) b WHERE a.codigo_gravadora = b.codigo_gravadora AND
a.preco_venda > b.preco_medio;
```



# Consulta em Banco de Dados

## *Subqueries*

### ● Atividades:

- Faça uma lista de imóveis do mesmo bairro do imóvel 2. Exclua o imóvel 2 da sua busca;
- Faça uma lista que mostre todos os imóveis que custam mais que a média de preço dos imóveis;
- Faça uma lista com todos os compradores que tenham ofertas cadastradas com valor superior a 70 mil;
- Faça uma com todos os imóveis com oferta com oferta superior à média do valor das ofertas;
- Faça uma lista com todos os imóveis com preço superior à média de preço dos imóveis do mesmo bairro;
- Faça uma lista dos imóveis com maior preço agrupado por bairro, cujo maior preço seja superior à média de preços dos imóveis;
- Faça uma lista com os imóveis que têm o preço igual ao menor preço de cada vendedor;
- Faça uma lista com as ofertas dos imóveis com data de lançamento do imóvel inferior a 30 dias e superior a 180 dias, a contar de hoje e cujo código vendedor seja 2;
- Faça uma lista com os imóveis que têm o preço igual ao menor preço de todos os vendedores, exceto os imóveis do próprio vendedor;
- Faça uma lista com as ofertas menores que todas as ofertas do comprador 2, exceto as ofertas do próprio comprador;
- Faça uma lista de todos os imóveis cujo estado e cidade sejam os mesmos do vendedor 3, exceto os imóveis do vendedor 3;
- Faça uma lista com todos os nomes de bairro cujos imóveis sejam do mesmo estado, cidade e bairro do imóvel código 5;