

Banco de Dados

Rebeca Barros



Subqueries

MySQL



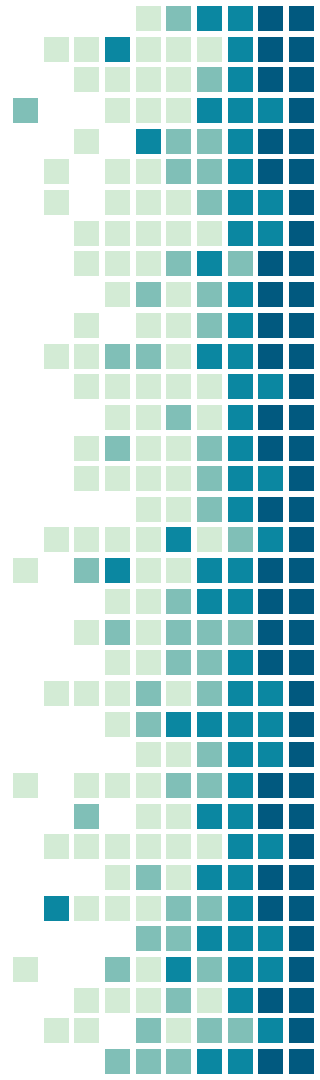
Subqueries

- Uma subquery (ou subconsulta) é uma consulta dentro de outra consulta SQL.
- Uma subquery sempre está dentro de parênteses e usualmente é executada antes da consulta onde ela está contida.



Subqueries

- Como qualquer outra consulta SQL, uma subquery pode retornar:
 - Uma única linha com única coluna;
 - Múltiplas linhas com uma única coluna;
 - Múltiplas linhas e múltiplas colunas.



Subqueries

- Além disso, subqueries também podem ser divididas em:
 - **Não correlatas** - quando não dependem de colunas da consulta base;
 - **Correlatas** - quando referenciam colunas da consulta base

Subquery de uma única linha/coluna

-- consultar todos os cds com preços acima da media

```
SELECT nome, preco_venda FROM cd  
WHERE preco_venda > (SELECT avg(preco_venda) from cd);
```

-- consultar os cds com preços acima da media por gravadora

```
SELECT Gravadora_idGravadora, nome, preco_venda  
FROM cd c WHERE preco_venda >  
(SELECT avg(preco_venda) from cd  
WHERE Gravadora_idGravadora = c.Gravadora_idGravadora);
```

Subquery de uma única linha/coluna

```
/* consulte os cds que são da mesma gravadora do cd com id 2 e  
que tenham preço de venda maior que o cd com id 5 */  
SELECT Gravadora_idGravadora, nome, preco_venda  
FROM cd WHERE Gravadora_idGravadora =  
(SELECT Gravadora_idGravadora FROM cd WHERE idCD = 2)  
AND preco_venda >  
(SELECT preco_venda FROM cd WHERE idCD = 5);
```

Subquery de uma única linha/coluna

/* consulte o id da gravadora e o menor preço de cd de todas as gravadoras em que o menor preço de cd seja maior que o preço do cd de id 6 */

```
SELECT Gravadora_idGravadora, min(preco_venda)
FROM cd group by Gravadora_idGravadora
having min(preco_venda) >
(SELECT preco_venda from cd where idCD = 6);
```

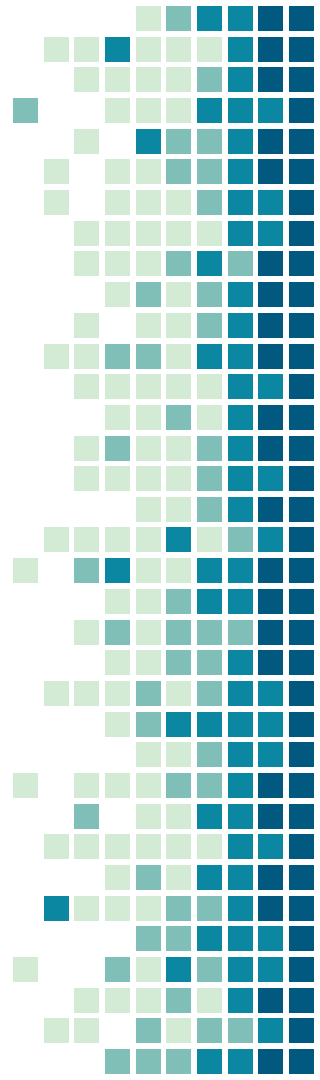

Subquery de uma única linha/coluna

/* consultar o id da gravadora e o preço máximo de cd onde esse preço máximo seja maior que a média de preço da gravadora*/

```
SELECT Gravadora_idGravadora, max(preco_venda)
FROM cd c group by Gravadora_idGravadora
having max(preco_venda) >
(SELECT avg(preco_venda) from cd
where Gravadora_idGravadora = c.Gravadora_idGravadora);
```

O comando EXISTS

- Usado quando você precisa identificar que um relacionamento existe sem precisar saber a quantidade de linhas.
- Se a subquery conter uma ou mais linhas, o resultado é mostrado, se não, não é mostrado.



O comando EXISTS

```
-- consultar id e nome das gravadoras que possuem cds cadastrados
SELECT idGravadora, nomeGravadora
FROM gravadora g WHERE EXISTS
(SELECT * FROM CD where cd.Gravadora_idGravadora = g.idGravadora);

-- consultar id e nome das gravadoras que não possuem cds cadastrados
SELECT idGravadora, nomeGravadora
FROM gravadora g WHERE NOT EXISTS
(SELECT * FROM CD where cd.Gravadora_idGravadora = g.idGravadora);
```

Subquery de múltiplas linhas

- Em subqueries que retornam mais de uma linha não é possível utilizar operadores simples (**=**, **<>**, **>** e **<**).
- Deve-se utilizar um operador de grupo que permita realizar a comparação com um conjunto e não com um único valor.

Subquery de múltiplas linhas

- Os operadores são:
 - **IN** (e **NOT IN**): checa se o valor está dentro de um conjunto de valores
 - **ANY/ALL**: permite comparações entre um valor único com cada valor dentro do conjunto. A diferença é que o ANY retorna *true* assim que encontrar uma comparação verdadeira, enquanto que o ALL só retorna se a comparação for verdadeira para todos os membros do conjunto.

Subquery de múltiplas linhas

```
/* consultar o cd que tem preço igual ao menor preço  
de cd de alguma gravadora */  
SELECT idCD, nome, preco_venda  
FROM cd where preco_venda IN  
(SELECT min(preco_venda) FROM cd group by Gravadora_idGravadora);  
  
-- consultar cd e faixa que possuam uma música com Amor no título  
  
SELECT CD_idCD, numero_faixa FROM faixa WHERE Musica_idMusica IN  
(SELECT idMusica FROM musica where nome like '%amor%');
```

Subquery de múltiplas linhas

/* consultar qual cd tem preço inferior a qualquer outro da gravadora 2 mas que não seja da gravadora 2 */

```
SELECT idCD, nome, preco_venda FROM CD
WHERE preco_venda < ANY
(SELECT preco_venda FROM cd where Gravadora_idGravadora = 2)
AND Gravadora_idGravadora <> 2;
```

Subquery de múltiplas linhas

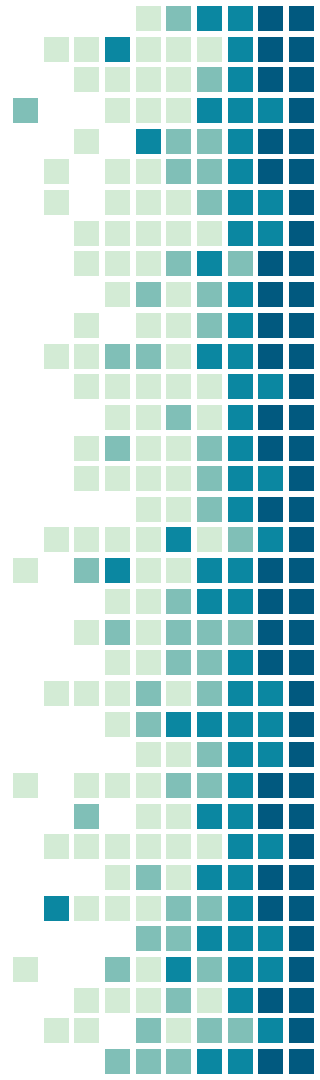
/* consultar quais cds tem preço de venda menor que a média de preço de todas as gravadoras */

```
SELECT idCD, nome, preco_venda FROM CD
WHERE preco_venda < ALL
(SELECT avg(preco_venda) FROM cd group by Gravadora_idGravadora);
```


Subquery de múltiplas linhas

Importante:

- Ao utilizar **not in** ou **<> all** para comparar um valor a um conjunto, é importante garantir que o conjunto de valores não contenha um valor ***null*** pois qualquer comparação com ***null*** retorna um valor desconhecido.



Subquery de múltiplas linhas/columnas

```
/* Considerando o menor id de cd indicado de uma gravadora,  
retorne as informações sobre os cds que o indicam */  
SELECT idCD, nome, Gravadora_idGravadora, CD_indicado FROM CD  
WHERE (Gravadora_idGravadora, CD_indicado) IN  
(SELECT Gravadora_idGravadora, min(CD_indicado) FROM cd  
group by Gravadora_idGravadora);
```

Subquery como fontes de dados

/* consultar nome do cd, preço de venda e preço médio da gravadora quando o preço do cd for maior que a média da sua gravadora */

```
SELECT c.nome, c.preco_venda, g.preco_medio
FROM cd c INNER JOIN (SELECT Gravadora_idGravadora, avg(preco_venda) as
preco_medio FROM cd group by Gravadora_idGravadora) g ON
c.Gravadora_idGravadora = g.Gravadora_idGravadora WHERE
c.preco_venda > g.preco_medio;
```

Subquery na manipulação dados

```
UPDATE conta c
SET c.ultima_atividade =
(SELECT MAX(t.tr_data) FROM transacao t
WHERE t.conta_id = c.conta_id)
WHERE EXISTS (SELECT 1 FROM transacao t
WHERE t.conta_id = a.conta_id);
```

Subquery na manipulação dados

```
DELETE FROM departamento WHERE NOT EXISTS  
(SELECT 1 FROM empregado  
WHERE empregado.dept_id = departamento.dept_id);
```

Referências

- Beaulieu, A. (2009). *Learning SQL: Master SQL Fundamentals*. "O'Reilly Media, Inc."
- de OLIVEIRA, C. H. P. (2002). *SQL: curso prático*. Novatec.
- Documentação oficial do MySQL. Disponível em: <https://dev.mysql.com/doc/refman/8.0/en/>
- Curso Completo de MySQL (Bóson Treinamentos). Disponível em: <http://www.bosontreinamentos.com.br/curso-completo-de-mysql/>