

Subconsultas

Introdução

- ▶ Realizar um select dentro do resultado de um outro select.
- ▶ Palavras chave
 - ▶ IN
 - ▶ ANY
 - ▶ ALL

Palavra chave ANY

- ▶ A palavra-chave ANY, que deve seguir um operador de comparação, significa "retornar TRUE se a comparação é verdadeira para qualquer dos valores na coluna que a subconsulta retorna." Por exemplo:
- ▶ `SELECT s1 FROM t1 WHERE s1 > ANY (SELECT s1 FROM t2);`
- ▶ Suponhamos que há uma linha na tabela t1 contendo (10). A expressão é True se a tabela t2 contém (21,14,7) porque há um valor 7 em t2 que é inferior a 10. A expressão é FALSE se a tabela t2 contém (20,10), ou se a tabela t2 está vazia. A expressão é desconhecida (ou seja, NULL) se a tabela t2 contém (NULL, NULL, NULL).

Palavra chave ANY

- ▶ Esse operador permite comparar operadores simples (=, >, <, !=) com um grupo de linhas. Será feita a comparação com cada valor retornado do SELECT interno.
- ▶ Imagine que desejamos saber quais CDs têm preço inferior a qualquer outro da gravadora com código 2, mas que não sejam da 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
```

Palavra chave ANY

- ▶ Podemos ainda utilizar o mesmo operador any com os outros operadores simples.
- ▶

```
select codigo_cd, nome_cd, preco_venda from cd  
where preco_venda = any (select preco_venda from cd  
where codigo_gravadora = 2)
```
- ▶ neste exemplo não existe mais a condição do código da gravadora ser diferente de 2, caso contrário, não existiriam linhas de retorno. Apenas os Cds da gravadora 2 aparecem na última listagem.

Palavra chave IN

- ▶ Quando usado com uma subconsulta, a palavra IN é um apelido para = ANY. Assim, estas duas declarações são as mesmas:
 - ▶ `SELECT s1 FROM t1 WHERE s1 = ANY (SELECT s1 FROM t2);`
 - ▶ `SELECT s1 FROM t1 WHERE s1 IN (SELECT s1 FROM t2);`
- ▶ Para fazer `!= ANY` usa-se NOT IN

Palavra chave IN

- ▶ Imagine que gostaríamos de saber quais CDs têm o preço igual ao menor preço de cada gravadora. Inicialmente devemos saber qual o menor preço de cada gravadora. Isso pode ser feito com o comando:
- ▶ `Select min(preço_venda) from CD Group byCodigo_gravadora`

Palavra chave IN

- ▶ Para saber quais são os CDs com esses preço, poderíamos escrever o seguinte comando:
- ▶ `Select código_cd, nome_cd, preço_venda from cd Where preco_venda in (15,9,10.5)`

Palavra chave IN

Código_CD	Nome_CD	Preço_Venda
7	Listen without	9
5	Perfil	10,5
1	Mais do mesmo	15

Palavra chave IN

- ▶ Queremos saber em qual CD e faixa há músicas que tenham AMOR no nome.
- ▶ Para isso, temos que utilizar as linhas da tabela MUSICA e compará-las com AMOR.
- ▶ Não esquecer de colocar a função UPPER do nome do campo pois, para efetuar a comparação com sucesso, devemos ter certeza de que o conteúdo da coluna esteja em letras maiúsculas, uma vez que escrevemos AMOR em letras maiúsculas.

Palavra chave IN

- ▶ Select código_musica from musica

Where upper(nome_musica) like '%AMOR%'

- ▶ Sabendo quais são os código da música, podemos realizar uma busca na tabela FAIXA que nos informe em quais CDs e faixas estão essas músicas.

- ▶ Select código_cd, numero_faixa from faixa

Where código_musica in (25,41)

Palavra chave IN

- ▶ Da mesma forma, podemos simplificar o comando realizando tudo isso em uma única operação:

- ▶

```
Select codigo_cd, numero_faixa from faixa  
where codigo_musica in (select codigo_musica from musica  
                        where upper(nome_musica) like '%AMOR%')
```

Palavra chave ALL

- ▶ A palavra ALL, que deve seguir um operador de comparação, significa "retornar TRUE se a comparação é verdadeiro para todos os valores na coluna que a subconsulta retorna." Por exemplo:
- ▶ `SELECT s1 FROM t1 WHERE s1 > ALL (SELECT s1 FROM t2);`
- ▶ Suponhamos que há uma linha na tabela t1 contendo (10). A expressão é TRUE se a tabela t2 contém (-5,0, 5), porque 10 é maior do que todos os três valores em t2. A expressão é FALSE se a tabela t2 contém (12,6, NULL, -100) porque existe um único valor 12 na tabela t2 que é maior que 10. A expressão é desconhecida (que, é NULL) se a tabela t2 contém (0, NULL, 1).
- ▶ Finalmente, a expressão é TRUE se a tabela t2 está vazia.

Palavra chave 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.
- ▶ Imagine que queiramos saber quais CDs têm o preço de venda menor que a média de preço de venda de todas as gravadoras. Inicialmente é necessário saber a média de preços por gravadora:
- ▶ `Select avg(preço_venda) from cd Group by código_gravadora`

Palavra chave ALL

- ▶ Depois de verificarmos quais são os CDs com preço inferior a todos anteriores. O comando seria o seguinte:
- ▶ `Select código_cd, nome_cd, preço_venda from cd
where preço_venda < all (15,11,11.75)`
- ▶ Em um único comando teríamos:
- ▶ `select codigo_cd, nome_cd, preco_venda from cd
where preco_venda < all (select avg(preco_venda) from cd
group by codigo_gravadora)`

Palavra chave ALL

- ▶ Para fazer diferente de ALL usamos NOT IN
- ▶ `SELECT s1 FROM t1 WHERE s1 <> ALL (SELECT s1 FROM t2);`
- ▶ `SELECT s1 FROM t1 WHERE s1 NOT IN (SELECT s1 FROM t2);`

Subconsultas depois de FROM

- ▶ CREATE TABLE t1 (s1 INT, s2 CHAR(5), s3 FLOAT);
- ▶ INSERT INTO t1 VALUES (1,'1',1.0);
- ▶ INSERT INTO t1 VALUES (2,'2',2.0);

- ▶ SELECT sb1,sb2,sb3 FROM
(SELECT s1 AS sb1, s2 AS sb2, s3*2 AS sb3 FROM t1) AS sb
WHERE sb1 > 1;

- ▶ Resultado: 2, '2', 4.0.

Subconsultas depois de FROM

- ▶ Suponha que queremos saber a média de um conjunto de somas para uma tabela agrupada.
- ▶ Isso não funciona:
- ▶ `SELECT AVG(SUM(column1)) FROM t1 GROUP BY column1;`
- ▶ Isso funciona:
- ▶ `SELECT AVG(sum_column1) FROM
(SELECT SUM(column1) AS sum_column1 FROM
t1 GROUP BY column1) AS t1;`

Material copiado de

- ▶ Rubens Antunes. Subquery de múltiplas linhas.
<http://www.devmedia.com.br/subquery-de-multiplas-linhas/17378>
- ▶ MySQL Manual. <http://dev.mysql.com/doc/refman/5.7/en/subqueries.html>