

SUBCONSULTAS – SUBQUERIES

Nas sessões anteriores exploramos largamente o comando SELECT utilizando-o em conjunto com uma gama de funções. Ainda neste contexto, é possível incluir subconsultas dentro das consultas principais.

8.1. SUBQUERY

As subqueries fazem parte do padrão SQL-86 logo, todos os bancos de dados relacionais que utilizam SQL devem permitir essa utilização.

Sintaxe:

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

Há três tipos de subquery:

- a) **Subquery de uma linha:** o retorno do SELECT interno será uma única linha.
- b) **Subquery de múltiplas linhas:** o retorno do SELECT interno será mais de uma linha.
- c) **Subquery de múltiplas colunas:** o retorno do SELECT interno será conterá mais de uma linha e coluna.

8.1.1. Subquery de uma linha

Neste caso, primeiro é realizada a busca em uma tabela e, com base no resultado, pesquisada a outra tabela.

Exemplo:

```
SELECT cd_nome, cd_preco FROM cd
WHERE cd_preco > (SELECT AVG(cd_preco) FROM cd)
```

Acima é utilizada uma subquery para verificar a média de preço dos CD's e, com base nesse resultado, é extraído o resultado da busca do primeiro SELECT.

Exemplo:

```
SELECT grav_id, cd_nome, cd_preco FROM cd c
WHERE cd_preco > (SELECT AVG(cd_preco) FROM cd
                 WHERE grav_id = c.grav_id)
```

Acima são buscados apenas os CD's que tenham preço de venda superior a média da própria gravadora.

Muito cuidado ao usar essa construção, porque ele tende a consumir muito recurso do banco de dados, pois a cada linha do primeiro SELECT será extraído a média no segundo SELECT.

Assim é necessário saber qual a gravadora do primeiro SELECT, para poder calcular o segundo SELECT.

NOTA

- Coloque as subqueries entre parênteses.
- Coloque a subquery à direita do operador.
- Não use a cláusula ORDER BY em uma subquery. Se for necessário ordenar faça-o no SELECT principal.
- Use operadores de grupo apenas em buscas que potencialmente retornem mais de uma linha.
- Use operadores de linha apenas em buscas que retornem uma única linha.

Exemplo:

```
SELECT grav_id, cd_nome, cd_preco FROM cd
WHERE grav_id = (SELECT grav_id FROM cd WHERE cd_id = 2) AND
cd_preco > (SELECT cd_preco FROM cd WHERE cd_id = 5);
```

8.1.1.1. SUBQUERY EM CLÁUSULA HAVING

A utilização é semelhante a cláusula WHERE. A subquery será executada primeiro e o resultado da busca servirá de base para filtrar as linhas do GROUP BY.

Exemplos:

```
SELECT grav_id, MIN(cd_preco) FROM cd
GROUP BY grav_id
HAVING MIN(cd_preco) > (SELECT cd_preco FROM cd
                       WHERE cd_id = 2)

SELECT grav_id, MAX(cd_preco) FROM cd c
GROUP BY grav_id
HAVING MAX(cd_preco) > (SELECT AVG(cd_preco) FROM cd
                       WHERE grav_id = c.grav_id)
```

8.1.1.2. COMANDO EXISTS

Verifica o número de linhas retornadas pela subquery. Caso contenha uma ou mais linhas, então o resultado será mostrado.

Exemplo:

```
SELECT grav_id, grav_nome FROM gravadora
WHERE EXISTS (SELECT * FROM cd
                WHERE cd.grav_id = gravadora.grav_id)
```

Note que quando se está utilizando esse operador, não importa o que o comando SELECT interno irá buscar. Interessa apenas se ele retorna ou não linhas, Por esse motivo é usado o * em vez de uma coluna em especial.

8.1.2. Subquery de múltiplas linhas

Neste caso o SELECT interno retorna mais de uma linha. Não é possível utilizar operadores simples como igualdade, diferença, maior ou menor.

Utilizam-se operadores de grupo para realizar a comparação, como:

- ANY
- ALL
- IN

8.1.2.1. IN

Imagine que queremos saber quais CD's têm o preço igual ao menor preço de cada gravadora. Inicialmente devemos saber qual o menor preço de cada gravadora.

```
SELECT MIN(cd_preco) FROM cd
GROUP BY grav_id
```

Para saber quais são os CD's escreveríamos o seguinte comando:

```
SELECT cd_id, cd_nome, cd_preco FROM cd
WHERE cd_preco IN (19, 23, 11)
```

Para fazer isso em um único comando, escreveríamos o comando da seguinte forma:

```
SELECT cd_id, cd_nome, cd_preco FROM cd
WHERE cd_preco IN (SELECT MIN(cd_preco) FROM cd
                  GROUP BY grav_id)
```

8.1.2.2. ANY (Compara com qualquer valor)

Esse operador permite comparar operadores simples (=,<,>!=) com um grupo de linhas. Imagine que desejamos saber quais CD's têm preço inferior a qualquer outro da gravadora com código 2, mas que não sejam da gravadora 2.

Inicialmente é preciso saber qual é o preço de venda de cada CD que não seja da gravadora 2:

```
SELECT cd_id, cd_nome, cd_preco FROM cd
WHERE grav_id != 2
```

Depois precisamos saber quais os preços dos CD's da gravadora 2 com o comando da seguinte forma:

```
SELECT cd_preco FROM cd
WHERE grav_id = 2
```

Agora realizamos a pesquisa na tabela de CD para comparar os preços inferiores a esses:

```
SELECT cd_id, cd_nome, cd_preco FROM cd
WHERE cd_preco < ANY (SELECT cd_preco FROM cd
                      WHERE grav_id = 2)
AND grav_id != 2
```

8.1.2.3. ALL

Esse operador é usado 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 CD's 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(cd_preco) FROM cd
GROUP BY grav_id;
```

Depois verificamos quais são os CD's com preço inferior a todos anteriores.

```
SELECT cd_id, cd_nome, cd_preco FROM cd
WHERE cd_preco < ALL (SELECT AVG(cd_preco) FROM cd
                     GROUP BY grav_id)
```

8.1.3. Subquery de múltiplas colunas

A técnica consiste em colocar as colunas unidas na cláusula WHERE do SELECT externo e realizar a busca no SELECT interno dessas colunas unidas da mesma forma. Essa técnica tende a ser muito lenta, mas pode ser feita em qualquer banco de dados.

Temos vários CD's indicados no cadastro e queremos conhecer os dados do CD dentre os menores CD's indicados de cada gravadora. Para isso temos que saber o menor CD indicado de cada gravadora. Utilizamos o || para unir as colunas.

Exemplo:

```
SELECT grav_id || cast(MIN(cd_indicado) as varchar)
FROM cd
GROUP BY grav_id
```

Agora realizamos a busca dos dados que atendem essas características:

```
SELECT cd_id, cd_nome, grav_id, cd_indicado
FROM cd
WHERE (grav_id || cast (cd_indicado as varchar)) IN
      (SELECT grav_id || cast(MIN(cd_indicado) as varchar) FROM cd
       GROUP BY grav_id)
```

Outra técnica consiste em colocar as colunas de busca entre parênteses no SELECT externo, entretanto não está disponível em alguns bancos de dados.

Exemplo:

```
SELECT cd_id, cd_nome, grav_id, cd_indicado FROM cd
WHERE (grav_id, cd_indicado) IN
      (SELECT grav_id, MIN(cd_indicado) FROM cd
       GROUP BY grav_id)
```

NOTA:

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

8.1.4. Subquery na Cláusula FROM

Podemos utilizar uma subquery na cláusula FROM de um comando SELECT.

Essa estrutura não está disponível para todos os bancos de dados, mas faz parte do padrão SQL. Sua utilização é muito parecida com as visões (Views).

Imagine que queremos extrair o nome do CD, seu preço e o preço médio da gravadora, e o preço médio da gravadora é extraído por uma subquery na própria cláusula FROM:

Exemplo:

```
SELECT a.cd_nome, a.cd_preco, b.preco_medio
FROM cd a, (SELECT grav_id, AVG(cd_preco) as preco_medio FROM cd
           GROUP BY grav_id) b
WHERE a.grav_id = b.grav_id
AND a.cd_preco > b.preco_medio
```

Veja que foi substituída a segunda tabela por um comando SELECT e que a tabela recebeu o apelido de b.

8.2. EXERCÍCIOS

- a) Faça uma lista de CD's usando subquery que foram gravados pela gravadora 2.
- b) Faça uma lista que mostre os CD's que custam mais que a média de preço dos CD's gravados pela gravadora 1.
- c) Faça uma consulta usando a cláusula HAVING que retorne o CD mais barato que a média de preço dos CD's.
- d) Faça uma lista que mostre os CD's cujas músicas tem duração maior que 00:02:00.
- e) Faça uma lista que mostre quais CD's tem preços superiores a qualquer CD da gravadora 1 excluindo os da própria gravadora 1.