



Subconsultas - Subquerys

1

Banco de Dados II



Banco de Dados II

Subconsultas - Subquerys

O que é uma Subconsulta SQL (Subquery)

Uma subconsulta é uma consulta embutida dentro de outra consulta, de forma aninhada, passando os resultados da consulta mais interna para a consulta mais externa por meio de uma cláusula **WHERE** ou de uma cláusula **HAVING**.

Desta forma, é possível restringir mais ainda os dados retornados por uma consulta, permitindo a criação de filtros bastante aprimorados. A subconsulta retorna os dados que serão empregado pela consulta "principal", alimentando-a com informações a serem utilizadas como condições de filtragem.





Banco de Dados II

Subconsultas - Subquerys

O que é uma Subconsulta SQL (Subquery)

Podemos utilizar subconsultas não apenas em consultas com a cláusula SELECT, mas também em operações INSERT, UPDATE e DELETE.

Quando usamos uma subconsulta em uma query, a subconsulta é resolvida primeiro, e então a consulta externa (principal) é resolvida de acordo com o resultado retornado pela subconsulta (subquery).





Banco de Dados II

Subconsultas - Subquerys

Regras gerais para aplicação de subconsultas

Algumas regras gerais precisam se seguidas para que seja possível empregar subconsultas. Entre elas, temos:

- A subconsulta pode ter apenas uma única coluna em sua cláusula SELECT, exceto quando a consulta principal tiver múltiplas colunas para comparação com as colunas selecionadas
- Se a subconsulta retornar mais de uma linha de dados, é necessário usar operadores de valores múltiplos, como o operador lógico IN
- As subconsultas são escritas dentro de parênteses



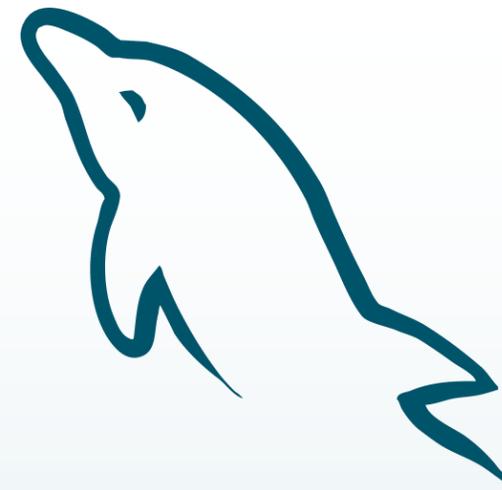


Banco de Dados II

Subconsultas - Subquerys

Regras gerais para aplicação de subconsultas

- **Não é possível usar a cláusula ORDER BY** em uma subconsulta – mas a consulta principal pode.
- **É possível usar uma cláusula GROUP BY** em uma subconsulta.
- **Não podemos usar o operador BETWEEN** com uma subconsulta (na consulta principal), mas podemos usar esse operador dentro da subconsulta.





Banco de Dados II

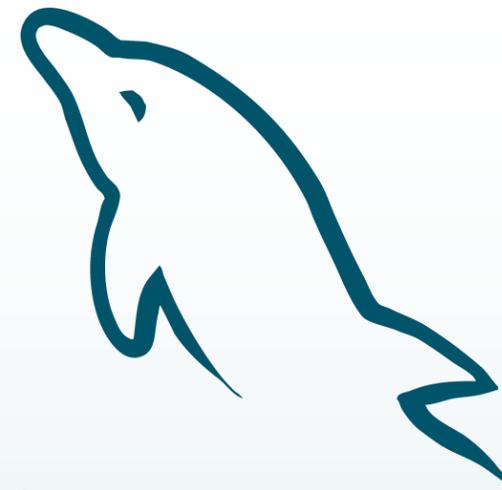
Subconsultas - Subquerys

Sintaxe Básica de uma Subconsulta SQL

A sintaxe básica para a criação de uma subconsulta SQL é a seguinte:

```
SELECT coluna(s)  
FROM tabela(s)  
WHERE coluna  
operador (SELECT coluna FROM tabela  
WHERE condições);
```

Onde operador pode ser qualquer operador lógico ou relacional, como >, <, >=, <=, =, <>, IN, NOT, AND, OR, etc.





Banco de Dados II

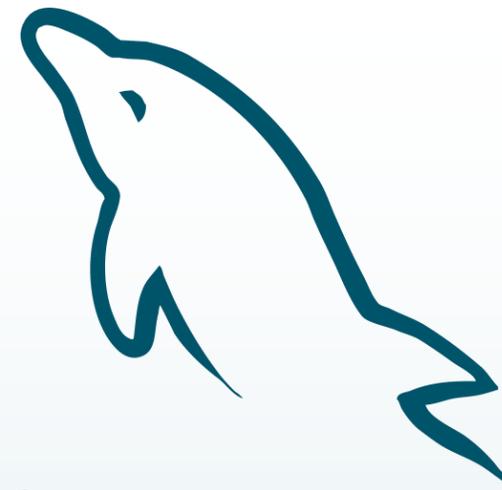
Subconsultas - Subquerys

Sintaxe Básica de uma Subconsulta SQL

A sintaxe básica para a criação de uma subconsulta SQL é a seguinte:

```
SELECT coluna(s)  
FROM tabela(s)  
WHERE coluna  
operador (SELECT coluna FROM tabela  
WHERE condições);
```

Onde operador pode ser qualquer operador lógico ou relacional, como >, <, >=, <=, =, <>, IN, NOT, AND, OR, etc.





Banco de Dados II

Subconsultas - Subqueries

Subconsulta com operação SELECT

Vamos efetuar uma consulta à tabela de livros de nosso banco de dados db_biblioteca, retornando os livros (tbl_livro) publicados pela editora Wiley (tbl_editora), porém usando uma subconsulta em vez de usar INNER JOIN:

```
SELECT NomeLivro, PrecoLivro, IdEditora
FROM tbl_livro
WHERE IdEditora = (SELECT IdEditora FROM
tbl_editora WHERE NomeEditora = 'Wiley');
```

Resultado.

NomeLivro	PrecoLivro	IdEditora
Linux Command Line and Shell Scripting	88.74	4
Fedora and Red Hat Linux	68.46	4





Banco de Dados II

Subconsultas - Subquerys

Subconsulta com operação INSERT INTO

Podemos usar subconsultas também com a instrução de inserção de registros INSERT INTO, e outras instruções de linguagem de manipulação de dados (DML). Com a instrução INSERT INTO, os dados retornados por uma subconsulta são usados para realizar a inserção de um registro em outra tabela.

```
INSERT INTO tabela (colunas)  
(SELECT coluna(s) FROM tabela(s)  
WHERE coluna operador valor);
```





Banco de Dados II

Subconsultas - Subquerys

Subconsulta com operação INSERT INTO

Vamos criar uma tabela de exemplo chamada de LivrosAssuntos, contendo os campos ID, NomeLivro e NomeAssunto:

```
CREATE TABLE LivrosAssuntos (  
    ID TINYINT PRIMARY KEY AUTO_INCREMENT,  
    NomeLivro VARCHAR(50), NomeAssunto VARCHAR(40)  
)
```





Banco de Dados II

Subconsultas - Subquerys

Subconsulta com operação INSERT INTO

Usando uma subconsulta, vamos popular esta nova tabela com os livros que tenham mais de 600 páginas e seus respectivos assuntos:

```
INSERT INTO LivrosAssuntos (NomeLivro, NomeAssunto)
(SELECT L.NomeLivro, A.NomeAssunto FROM tbl_livro L
INNER JOIN tbl_assunto A ON L.IdAssunto = A.IdAssunto
WHERE L.NumPaginas >= 600);
```

Após executar o comando, efetuamos uma consulta à nova tabela para verificar se os dados foram inseridos com sucesso:

```
SELECT * FROM LivrosAssuntos
```

Resultado

ID	NomeLivro	NomeAssunto
1	Linux Command Line and Shell ...	Tecnologia
2	Fedora and Red Hat Linux	Tecnologia
3	Windows Server 2012 Inside Out	Tecnologia
4	Microsoft Exchange Server 2010	Tecnologia
5	Practical Electronics for Inventors	Tecnologia





Banco de Dados II

Subconsultas - Subquerys

Subconsulta com operação UPDATE

Também podemos usar uma subconsulta com uma instrução UPDATE, permitindo atualizar os valores de registros de uma tabela com base no resultado retornado pela subconsulta.

```
UPDATE tabela SET coluna(s) =  
valor WHERE coluna operador  
(SELECT coluna(s) FROM tabela WHERE coluna operador  
valor);
```





Banco de Dados II

Subconsultas - Subquerys

Subconsulta com operação UPDATE

Vamos aumentar os preços dos livros da editora Microsoft Press em 12% usando a declaração UPDATE. Como não sabemos o ID desta editora de memória, vamos usar uma subconsulta para nos auxiliar:

```
UPDATE tbl_livro SET PreçoLivro = PreçoLivro * 1.12  
WHERE IdEditora = (SELECT IdEditora FROM tbl_Editora  
WHERE NomeEditora = 'Microsoft Press');
```

A subconsulta retorna o ID da editora de acordo com o nome da editora fornecido, e este ID é então empregado na consulta principal para aplicar a atualização nos valores dos registros desejados.





Banco de Dados II

Subconsultas - Subquerys

Subconsulta com operação DELETE

Finalmente, podemos usar subconsultas em instruções DELETE, selecionando o valor a ser excluído de uma tabela de acordo com o retorno da subconsulta.

```
DELETE FROM tabela  
WHERE coluna operador (SELECT coluna FROM tabela WHERE  
coluna operador valor);
```





Banco de Dados II

Subconsultas - Subqueries

Subconsulta com operação DELETE

Desejamos excluir os livros publicados cujo assunto é "Dinossauros", porém não sabemos o ID deste assunto na tabela de livros. Suponha também que não sabemos com exatidão se o nome do assunto é "Dinossauros", "Dinossauro" (no singular), "Dinos" ou algo do gênero – apenas os lembramos que o nome do assunto começa com "Dino":

```
DELETE FROM tbl_livro  
WHERE IdAssunto =  
(SELECT IdAssunto FROM tbl_Assunto WHERE NomeAssunto  
LIKE 'Dino%');
```





Banco de Dados II

Subconsultas - Subquerys



Subconsultas encadeadas

Podemos aninhar uma subconsulta dentro de outra subconsulta, da mesma maneira que uma subconsulta é aninhada dentro de uma consulta principal. Neste caso, a subconsulta mais interna é resolvida primeiro, depois a subconsulta intermediária, e finalmente o resultado é repassado para a consulta principal.

```
SELECT coluna(s) FROM tabela(s)
WHERE coluna operador
    ( SELECT coluna FROM tabela
      WHERE coluna operador (
        SELECT coluna FROM tabela
        WHERE coluna operador valor));
```

Na prática, podemos pensar na subconsulta intermediária como fazendo o papel de uma consulta principal para a subconsulta de nível mais interno, e, assim, o processo é similar para quantos níveis de subconsultas forem aninhados.



Banco de Dados II

Subconsultas - Subqueries

Subconsultas encadeadas

Há um limite para a quantidade das subconsultas que podem ser aninhadas?

Isso depende muito da implementação de banco de dados que está sendo empregada. Por exemplo, o SGBDR **Microsoft SQL Server suporta até 32 níveis de aninhamento** (subconsulta dentro de subconsulta), porém esse limite pode ser diferente em outros sistemas, muitas vezes por conta de restrições de memória disponível ou da complexidade em si das consultas. Consulte sempre a documentação do sistema em uso para saber mais a respeito.





Referências

- **Boson Treinamentos**

www.bosontreinamentos.com.br/mysql

