



MODELAGEM DE BANCO DE DADOS

Emerson Alberto Marconato

marconato@univem.edu.br

Maio 2024

OBJETIVOS

- ❑ Conceitos sobre Subqueries e Queries Correlatas
- ❑ Cláusula CASE
- ❑ Cláusula HAVING

SUBQUERIES

Subquery é um comando SELECT que foi "embutido" em outro comando SELECT, UPDATE, DELETE ou dentro de outra subquery.

A finalidade da subquery é retornar um conjunto de linhas para a query principal.

Utilização

Tipicamente utiliza-se subqueries na filtragem de pesquisas (=cláusula WHERE) nas cláusulas IN() e EXISTS(), mas subqueries também podem aparecer também na cláusula FROM ou como substituto de expressões.

UTILIZANDO UMA SUBQUERY EM CONJUNTO COM A CLÁUSULA **IN**

NFE

numero	data	iddiente
1	2023-04-11	1
2	2023-04-12	1
3	2023-04-13	2
4	2023-04-13	3

Item_NFE

numero	idproduto	quantidade
1	1	10.0000
1	2	3.0000
1	5	1.0000
2	4	5.0000
3	1	10.0000
4	1	100.0000
4	4	7.0000
4	5	3.0000

```
SELECT numero  
FROM nfe  
WHERE numero IN (SELECT numero  
                  FROM item_nfe  
                  WHERE quantidade >= 10);
```

Quais nfe possuem quantidades de itens maiores que 10??

UTILIZANDO UMA SUBQUERY COMO "ALVO" DA CLÁUSULA FROM

Cidade

idcidade	nome	uf
1	Marília	SP
2	Vera Cruz	SP
3	Garça	SP
4	Pompéia	SP
5	Assis	SP

Bairro

idbairro	descricao	idcidade
100	Centro	1
200	Alto Cafezal	1
300	Campus Universitário	1
400	Maria Isabel	1
500	Centro	2
600	Aeroporto	2
700	Centro	3
800	Distrito Industrial	3
900	Centro	4
950	CECAP	5

```
SELECT cidade.idcidade, cidade.nome, TB.Qtde_Bairro
FROM cidade
JOIN (SELECT idcidade, COUNT(*) AS Qtde_Bairro
      FROM bairro
      GROUP BY idcidade) TB
ON (cidade.idcidade = TB.idcidade);
```

UTILIZANDO UMA SUBQUERY PARA SUBSTITUIR UMA EXPRESSÃO

Produto

idproduto	descricao	valor_unitario
1	Lápis Nº 2	5.80
2	Caneta esferográfica Azul	1.20
3	Caneta esferográfica Preta	1.20
4	Caneta esferográfica Vermelha	0.98
5	Caderno com 200 folhas	38.59

```
SELECT descricao, valor_unitario,  
      (SELECT AVG(valor_unitario)  
       FROM produto) Media  
FROM produto;
```

QUERY CORRELATA

Quando uma subquery referencia colunas da query principal, recebe o nome de Subquery Correlata. Diferentemente das subqueries convencionais, a Subquery Correlata será executada tantas vezes quantas forem as linhas de output da query principal, num processo de Nested Loop Join.

QUERY CORRELATA

NFE

numero	data	idcliente
1	2023-04-11	1
2	2023-04-12	1
3	2023-04-13	2
4	2023-04-13	3

Item_NFE

numero	idproduto	quantidade
1	1	10.0000
1	2	3.0000
1	5	1.0000
2	4	5.0000
3	1	10.0000
4	1	100.0000
4	4	7.0000
4	5	3.0000

Produto

idproduto	descricao	valor_unitario
1	Lápis Nº 2	5.80
2	Caneta esferográfica Azul	1.20
3	Caneta esferográfica Preta	1.20
4	Caneta esferográfica Vermelha	0.98
5	Caderno com 200 folhas	38.59

```
SELECT numero, data,  
      (SELECT SUM(quantidade * valor_unitario)  
       FROM item_nfe  
        JOIN produto ON (item_nfe.idproduto = produto.idproduto)  
        WHERE item_nfe.numero = nfe.numero ) as Total  
FROM nfe;
```


QUERY CORRELATA EM CONJUNTO COM A CLÁUSULA **EXISTS**

Bairro

idbairro	descricao	idcidade
100	Centro	1
200	Alto Cafezal	1
300	Campus Universitário	1
400	Maria Isabel	1
500	Centro	2
600	Aeroporto	2
700	Centro	3
800	Distrito Industrial	3
900	Centro	4
950	CECAP	5

CEPs

cep	logradouro	auxiliar	idbairro
17502053	Rua Coroados	até 390/391	200
17504060	Rua Coroados	de 392/393 ao fim	200
17509210	Avenida Ipiranga	NULL	100

```
SELECT *  
  FROM bairro  
 WHERE EXISTS (SELECT 1  
                FROM ceps  
               WHERE ceps.idbairro = bairro.idbairro  
                  AND auxiliar IS NOT NULL) ;
```

CASE

CASE :

Sintaxe

case(expression , WHEN search THEN result [, WHEN search THEN result]... [, ELSE])

expression o valor a ser comparado.

Search o valor que será comparado com a ***expression***.

Result o valor retornado, se a ***expression*** for igual a ***search***.

ELSE - opcional. Monstrada quando o CASE não encontra um resultado compatível.

Ex.:

```
SELECT valor_unitario,  
       CASE valor_unitario  
         WHEN 0.98 THEN 'Baixo'  
         WHEN 1.20 THEN 'Medio'  
         WHEN 5.8 THEN 'Alto'  
         WHEN 38.59 THEN 'Muito alto'  
         ELSE 'Indeterminado'  
       END as Classificacao  
FROM produto;
```

HAVING

Having :

Utilizado para filtrar valores obtidos com funções de grupo

Exemplo:

```
SELECT numero, SUM(quantidade) Qtde  
  FROM item_nfe  
 GROUP BY numero  
HAVING Qtde > 10;
```