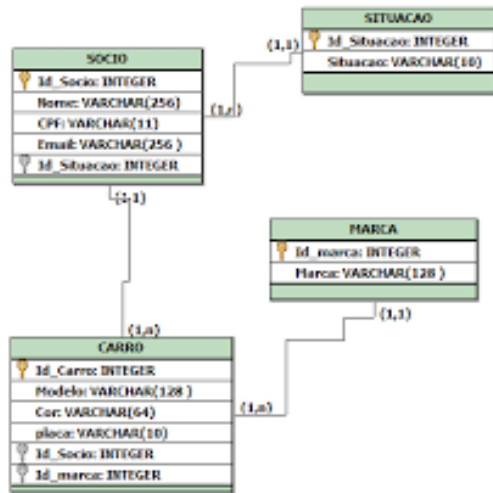


Bancos de Dados

Linguagem SQL – DQL – Data Query Language

Sub Selects



Clovs Ferraro
cferraro@unicid.edu.br

Funções de arredondamento

Funções de arrendodamento

- select pco_produto, **round**(pco_produto,1) from produto;
- select **avg**(pco_produto), **round**(**avg**(pco_produto),2) from produto;
- select pco_produto, **truncate**(pco_produto,2) from produto;
- select pco_produto, **floor**(pco_produto,2) from produto;
- select pco_produto, **ceiling**(pco_produto,2) from produto;

SUB SELECTS

Sub Selects (Subquery)

- Permite selecionar o que está contido em outra pesquisa.
- Subquery é um comando SELECT que foi "embutido" noutro comando SELECT, UPDATE, DELETE ou dentro de outra subquery.
- Uma subconsulta nada mais é do que uma instrução SELECT dentro de outro SELECT que retorna algumas colunas específicas que são usadas em algumas funções como INSERT e UPDATE por exemplo.
- Ou seja, a finalidade da subquery é retornar um conjunto de linhas para a query principal.







IN

```
select * from produto where cod_categoria in (1,2)
```

	COD_PRODUTO	NOM_PRODUTO	DSC_PRODUTO	IMG_PRODUTO	EST_PRODUTO	PCO_PRODUTO
▶	1	Ultrabook Acer Aspire M5-481T-6650 com Intel...	Ultrabook Acer Aspire M5-481T-6417 com Intel...	NULL	10	2199.99
	2	Ultrabook Acer Aspire M5-481T-6417 com Intel...	Ultrabook Acer Aspire M5-481T-6650 com Intel...	NULL	10	1699.99
	3	Ultrabook Acer Aspire M5-481T-6195 com Intel...	Ultrabook Acer Aspire M5-481T-6195 com Intel...	NULL	10	2049.99
	4	Ultrabook Acer Aspire com Intel® Core™ i5	Ultrabook Acer Aspire com Intel® Core™ i5, 6G...	NULL	10	2399.99
★	NULL	NULL	NULL	NULL	NULL	NULL


NOT IN

```
select * from produto where cod_categoria not in (1,2)
```

Result Grid  Filter Rows: <input type="text"/> Edit:   Export/Import:   Wrap Cell Content: 						
	COD_PRODUTO	NOM_PRODUTO	DSC_PRODUTO	IMG_PRODUTO	EST_PRODUTO	PCO_PF
▶	5	Computador HP 402 G1 SFF com Intel Core i3-4...	Computador HP 402 G1 SFF com Intel Core i3-4...	NULL	10	1440.00
	6	Computador HP ProDesk 600 G1 SFF com Intel...	Computador HP ProDesk 600 G1 SFF com Intel...	NULL	10	2814.81
	7	Computador HP All in One Pavilion 20-b410br Pr...	Computador HP All in One Pavilion 20-b410br Pr...	NULL	10	1529.15
	8	Computador HP All in One Pavilion TouchSmart ...	Computador HP All in One Pavilion TouchSmart ...	NULL	10	3229.15
	9	Notebook Acer Aspire E1-572-6_BR648 com Int...	Notebook Acer Aspire E1-572-6_BR648 com Int...	NULL	10	1598.99
	10	Notebook Samsung ATIV Book 2 270E4E-KD6 co...	Notebook Samsung ATIV Book 2 270E4E-KD6 co...	NULL	10	1108.48
	11	Notebook 2 em 1 Touch HP Pavilion x360 11-n0...	Notebook 2 em 1 Touch HP Pavilion x360 11-n0...	NULL	10	1399.00
	12	Notebook HP ProBook 430 G1 com Intel Core i3-4...	Notebook HP ProBook 430 G1 com Intel Core i3-4...	NULL	10	1598.99

Group by

```
SELECT cod_produto FROM produto_caracteristica
```

Result Grid 	
	cod_produto
	4
	1
	2
	3
	4
	2
	3
	4

Group by

```
SELECT cod_produto FROM produto_caracteristica  
GROUP BY cod_produto
```

Result Grid	
	cod_produto
▶	1
	2
	3
	4

Group by

```
SELECT cod_produto, COUNT(cod_produto) AS conta_prod  
FROM produto_caracteristica GROUP BY cod_produto
```

	cod_produto	conta_prod
▶	1	6
	2	7
	3	7
	4	7

Subconsultas

Select coluna(s)

From tabela(s)

Where coluna operador (select coluna(s) from
tabela where condições)

Subconsultas

Podem ser usadas em:

Insert;

Update;

Delete;

Select.

Tipos de subconsultas

Subconsultas de uma única linha:

retornam zero ou uma linha para a instrução SQL externa;

Subconsultas de várias linhas:

retornam uma ou mais linhas para a SQL externa;

Subconsultas de várias colunas:

retornam mais de uma coluna para a instrução SQL externa;

Subconsultas correlacionadas:

fazem referência a uma ou mais colunas na instrução SQL externa;

Subconsultas aninhadas:

são feitas dentro de outra subconsulta (podemos aninhar até 255 subconsultas).

Subconsultas em uma cláusula WHERE

```
select cod_produto, nom_produto  
from produto  
where cod_fabricante = (  
select cod_fabricante from fabricante  
where nom_fabricante='acer');
```

Subconsultas em uma cláusula WHERE

```
select cod_fabricante from fabricante where  
nom_fabricante='acer';
```

```
select cod_produto, nom_produto from  
produto where cod_fabricante=1
```

Subconsultas em uma cláusula WHERE

Elabore uma consulta que liste o todos os produtos que tem preço acima do preço médio dos produtos.



Subconsultas em uma cláusula WHERE

```
select pco_produto from produto where  
pco_produto > avg(pco_produto);
```



Subconsultas em uma cláusula WHERE

```
select pco_produto from produto where  
pco_produto > (select avg(pco_produto)  
from produto);
```

Sub Selects (Subquery)

- Tipicamente utilizamos 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.
- Exemplos: **USE LOJA;**

```
SELECT COD_PRODUTO, NOM_PRODUTO  
FROM PRODUTO  
WHERE COD_PRODUTO IN (SELECT COD_CATEGORIA  
FROM CATEGORIA);
```

COD_PRODUTO	NOM_PRODUTO
1	Ultrabook Acer Aspire M5-481T-6650 com Intel® Core™ i3-3227U
2	Ultrabook Acer Aspire M5-481T-6417 com Intel® Core™ i5-3317U
3	Ultrabook Acer Aspire M5-481T-6195 com Intel® Core™ i5-3317U
4	Ultrabook Acer Aspire com Intel® Core™ i5
5	Computador HP 402 G1 SFF com Intel Core i3-4130, 4GB, 500GB...
6	Computador HP ProDesk 600 G1 SFF com Intel® Core™ i3-4130...
7	Computador HP All in One Pavilion 20-h410hr Processador Intel

Sub Selects (Subquery) de uma única linha

- Uma subconsulta de uma única linha é aquela que retorna null ou uma linha para a instrução SQL externa.
- Podemos colocar subconsultas em diversas expressões utilizando WHERE, HAVING ou em uma cláusula FROM.

Sub Selects (Subquery) de uma única linha

- Subconsultas em uma cláusula WHERE. Exemplo:

USE sakila;

```
SELECT first_name, last_name DEPOIS EXECUTA ESTE  
FROM actors  
WHERE actors_id = (SELECT actors_id  
                   FROM actors  
                   WHERE last_name = 'gibson');EXECUTA PRIMEIRO ESTE
```

OU

```
SELECT first_name, last_name  
FROM actor  
WHERE last_name = 'gibson';
```

first_name	last_name
MERYL	GIBSON

Sub Selects (Subquery) de uma única linha

- Subconsultas em uma cláusula WHERE
- Exemplo com **erro**:

```
select cod_produto, nom_produto from  
produto where nom_produto like  
'notebook%';
```



Sub Selects (Subquery) de uma única linha

- Subconsultas em uma cláusula WHERE
- Exemplo com **erro**:

```
select cod_produto, nom_produto  
from produto  
where cod_produto =  
(select cod_produto from produto where nom_produto like  
'notebook%');
```



Error Code: 1242. Subquery returns more than 1 row

Sub Selects (Subquery) de uma única linha

- Podemos também usar outros operadores de comparação em subconsultas de uma única linha. Exemplo – usando o operador '>':

USE loja;

SELECT AVG(PCO_PRODUTO) FROM PRODUTO;

AVG(PCO_PRODUTO)

1353.400400

SELECT cod_produto, nom_produto, pco_produto
FROM PRODUTO

WHERE PCO_PRODUTO > (SELECT AVG(PCO_PRODUTO) FROM PRODUTO);

cod_produto	nom_produto	pco_produto
1	Ultrabook Acer Aspire M5-481T-6650 com Intel® Core™ i3-3227U	2199.99
2	Ultrabook Acer Aspire M5-481T-6417 com Intel® Core™ i5-3317U	1699.99
3	Ultrabook Acer Aspire M5-481T-6195 com Intel® Core™ i5-3317U	2049.99
4	Ultrabook Acer Aspire com Intel® Core™ i5	2399.99
5	Computador HP 402 G1 SFF com Intel Core i3-4130, 4GB, 500GB, Gravador de ...	1440.00
6	Computador HP ProDesk 600 G1 SFF com Intel® Core™ i3-4130, 4GB, 500GB, ...	2814.81
7	Computador HP All in One Pavilion 20-b410br Processador Intel® Core™ i3-322...	1529.15
8	Computador HP All in One Pavilion TouchSmart 23-b000br Processador Intel® C	3229.15

Sub Selects (Subquery) de uma única linha

- Os tipos de subconsultas colocadas em uma cláusula FROM também são chamadas de *visões inline*.
- No que diz respeito à cláusula FROM da consulta externa, a saída da subconsulta é apenas uma fonte de dados.

```
SELECT p.cod_produto, p.pco_produto, p_c.conta_prod  
FROM produto p,  
      (SELECT cod_produto, COUNT(cod_produto) AS conta_prod  
       FROM produto_caracteristica  
       GROUP BY cod_produto) p_c  
WHERE p.cod_produto = p_c.cod_produto;
```




cod_produto	pco_produto	conta_prod
1	2199.99	7
2	1699.99	7
3	2049.99	7
4	2399.99	7

Sub Selects (Subquery) de uma única linha

```
SELECT * FROM produto_caracteristica;
```

```
DELETE FROM produto_caracteristica  
WHERE cod_produto = 1 AND cod_caracteristica = 7;
```

```
SELECT p.cod_produto, p.pco_produto, p_c.conta_prod  
FROM produto p,  
      (SELECT cod_produto, COUNT(cod_produto) AS conta_prod  
       FROM produto_caracteristica  
       GROUP BY cod_produto) p_c  
WHERE p.cod_produto = p_c.cod_produto;
```



cod_produto	pco_produto	conta_prod
1	2199.99	6
2	1699.99	7
3	2049.99	7
4	2399.99	7

Sub Selects (Subquery) de várias linhas

- Uma subconsulta de várias linhas retorna uma ou mais linhas para uma instrução SQL externa.
- Para tratar de uma subconsulta que retorna várias linhas, sua consulta externa pode usar o operador **IN, ANY ou ALL**.
- Podemos usar esses operadores para verificar se o valor de uma coluna está contido em uma lista de valores.

Sub Selects (Subquery) de várias linhas

- Uma subconsulta de várias linhas retorna uma ou mais linhas para uma instrução SQL externa.
- Para tratar de uma subconsulta que retorna várias linhas, sua consulta externa pode usar o operador **IN, ANY ou ALL**.

```
SELECT cod_produto, nom_produto  
FROM produto  
WHERE cod_produto IN (1,2,3);
```

cod_produto	NOM_PRODUTO
1	Ultrabook Acer Aspire M5-481T-6650 com Intel® Core™ i3-3227U
2	Ultrabook Acer Aspire M5-481T-6417 com Intel® Core™ i5-3317U
3	Ultrabook Acer Aspire M5-481T-6195 com Intel® Core™ i5-3317U

Sub Selects (Subquery) de várias linhas

- O exemplo simples a seguir usa **IN** para verificar se um valor de `cod_produto` está na lista de valores retornada pela subconsulta; a subconsulta retorna o valor de `cod_produto` dos produtos cujo nome contém a letra “e”:

```
SELECT cod_produto, nom_produto
FROM produto
WHERE cod_produto IN (SELECT cod_produto
                        FROM produto
                        WHERE nom_produto LIKE '%e%')
```

cod_produto	nom_produto
1	Ultrabook Acer Aspire M5-481T-6650 com Intel® Core™ i3-3227U
2	Ultrabook Acer Aspire M5-481T-6417 com Intel® Core™ i5-3317U
3	Ultrabook Acer Aspire M5-481T-6195 com Intel® Core™ i5-3317U
4	Ultrabook Acer Aspire com Intel® Core™ i5
5	Computador HP 402 G1 SFF com Intel Core i3-4130, 4GB, 500GB, Gravador de ...

Sub Selects (Subquery) de várias linhas

- Podemos também usar **NOT IN** para executar a lógica oposta de IN: verificar se um valor **não está em uma lista** de valores especificada.

```
SELECT cod_produto, nom_produto  
FROM produto  
WHERE cod_produto NOT IN (SELECT cod_produto  
FROM promocao_produto);
```

24 row(s) returned

cod_produto	nom_produto
2	Ultrabook Acer Aspire M5-481T-6417 com Intel® Core™ i5-3317U
3	Ultrabook Acer Aspire M5-481T-6195 com Intel® Core™ i5-3317U
4	Ultrabook Acer Aspire com Intel® Core™ i5
5	Computador HP 402 G1 SFF com Intel Core i3-4130, 4GB, 500GB, Gravador de ...
6	Computador HP ProDesk 600 G1 SFF com Intel® Core™ i3-4130, 4GB, 500GB, ...

Sub Selects (Subquery) de várias linhas

- Podemos também usar **IN** para executar a lógica oposta de IN: verificar se um valor **está em uma lista** de valores especificada.

```
SELECT count(*) FROM produto;
```



count(*)
25

```
SELECT cod_produto, nom_produto  
FROM produto  
WHERE cod_produto IN (SELECT cod_produto  
                        FROM item_compra);
```

cod_produto	nom_produto
1	Ultrabook Acer Aspire M5-481T-6650 com Intel® Core™ i3-3227U

Sub Selects (Subquery) de várias linhas

- O operador **ANY** é usado para comparar um valor com qualquer valor presente em uma lista.
- Você deve colocar um operador **=, <>, <, >, <= ou >=** antes de **ANY** em sua consulta.
- O exemplo a seguir usa ANY para obter os produtos cujo preço é menor do que qualquer um dos produtos em promoção.

```
SELECT cod_produto, nom_produto  
FROM produto  
WHERE pco_produto < ANY (SELECT pco_produto  
FROM promocao_produto);
```

19 row(s) returned

cod_produto	nom_produto
2	Ultrabook Acer Aspire M5-481T-6417 com Intel® Core™ i5-3317U
5	Computador HP 402 G1 SFF com Intel Core i3-4130, 4GB, 500GB, Gravador de ...
7	Computador HP All in One Pavilion 20-b410br Processador Intel® Core™ i3-322...
9	Notebook Acer Aspire E1-572-6_BR648 com Intel® Core™ i5-4200U, 6GB, 500...
10	Notebook Samsung ATIV Book 2 270E4E-KD6 com Intel® Dual Core, 4GB, 500G...

Sub Selects (Subquery) de várias linhas

- O operador **ALL** é usado para comparar um valor com todos os valores presentes em uma lista.
- Você deve colocar um operador **=, <>, <, >, <= ou >=** antes de **ALL** em sua consulta.
- O exemplo a seguir usa esse operador para obter os produtos cujo preço é maior do que todos os produtos em promoção.

```
SELECT cod_produto, nom_produto  
FROM produto  
WHERE pco_produto > ALL(SELECT pco_produto  
FROM promocao_produto);
```

6 row(s) returned

cod_produto	nom_produto
1	Ultrabook Acer Aspire M5-481T-6650 com Intel® Core™ i3-3227U
3	Ultrabook Acer Aspire M5-481T-6195 com Intel® Core™ i5-3317U
4	Ultrabook Acer Aspire com Intel® Core™ i5
6	Computador HP ProDesk 600 G1 SFF com Intel® Core™ i3-4130, 4GB, 500GB, ...
8	Computador HP All in One Pavilion TouchSmart 23-h000br Processador Intel® C...

Sub Selects (Subquery) de várias colunas

- As subconsultas vistas até aqui retornaram linhas contendo apenas uma coluna.
- Não estamos limitados a uma única coluna: é possível escrever subconsultas que retornam várias colunas.

Sub Selects (Subquery) de várias colunas

- O exemplo a seguir recupera os produtos com o menor preço para cada categoria de produtos:

```
SELECT cod_produto, cod_categoria, nom_produto, pco_produto
FROM produto
WHERE (cod_categoria, pco_produto)
      IN (SELECT cod_categoria, MIN(pco_produto)
          FROM produto
          GROUP BY cod_categoria);
```

cod_produto	cod_categoria	nom_produto	pco_produto
2	2	Ultrabook Acer Aspire M5-481T-6417 com Intel® Core™ i5-3317U	1699.99
5	5	Computador HP 402 G1 SFF com Intel Core i3-4130, 4GB, 500GB, Gravador de ...	1440.00
12	3	Notebook Asus X551MA-BRAL-SX207H com Intel® Quad Core, 4GB, 500GB, Gra...	1061.65
14	8	Impressora Monocromática Laser - Samsung ML-2165	269.00
21	14	Pen Drive Multilaser Nano 8GB - Preto	27.30
22	10	Tablet Samsung Galaxy Tab 3 Lite SMT111M Branco com Tela 7", Wi-Fi, 3G, 8GB...	551.65

- Note que a subconsulta retorna o valor de cod_categoria e o preço mínimo para cada categoria de produtos e que eles são comparados com os valores de cod_categoria e preço para cada produto na cláusula WHERE da consulta externa.

Sub Selects - Correlacionadas

- Uma subconsulta correlacionada referencia uma ou mais colunas na instrução SQL externa. Elas são chamadas de subconsultas correlacionadas, porque são relacionadas à instrução SQL externa por meio das mesmas colunas.
- Normalmente, usamos uma subconsulta correlacionada quando precisa de uma resposta para uma pergunta que depende de um valor em cada linha contida em uma consulta externa.

Sub Selects - Correlacionadas

- Por exemplo, talvez você queira ver se existe uma relação entre os dados, mas não se preocupa com o número de linhas retornadas pela subconsulta, isto é, quer apenas verificar se alguma linha é retornada, mas não se importa com a quantidade retornada.
- Além disso, uma subconsulta correlacionada **pode trabalhar com valores nulos**.

Sub Selects - Correlacionadas

- Exemplo: A subconsulta correlacionada a seguir recupera os produtos que têm preço maior do que a média para seu tipo de produto:

```
SELECT cod_produto, cod_categoria, nom_produto, pco_produto
FROM produto externa
WHERE pco_produto >
(SELECT avg(pco_produto)
FROM produto interna
WHERE interna.cod_categoria = externa.cod_categoria);
```

cod_produto	cod_categoria	nom_produto	pco_produto
1	2	Ultrabook Acer Aspire M5-481T-6650 com Intel® Core™ i3-3227U	2199.99
4	2	Ultrabook Acer Aspire com Intel® Core™ i5	2399.99
6	5	Computador HP ProDesk 600 G1 SFF com Intel® Core™ i3-4130, 4GB, 500GB, ...	2814.81
8	5	Computador HP All in One Pavilion TouchSmart 23-h000br Processador Intel® C...	3229.15
9	3	Notebook Acer Aspire E1-572-6_BR648 com Intel® Core™ i5-4200U, 6GB, 500...	1598.99
11	3	Notebook 2 em 1 Touch HP Pavilion x360 11-n022br Processador Intel® Celero...	1399.00
13	3	Notebook Lenovo G400S com Intel® Core™ i5-3230M, 4GB, 1TB, Gravador de ...	1399.00
15	8	Impressora Jato de Tinta HP Designjet T120 ePrinter Series CQ891A#B1K	3959.00
19	14	Pen Drive SanDisk Ultra® Dual Drive com USB 2.0 e Micro-USB 32GB - Preto	76.42
20	14	Pen Drive Kingston Data Traveler 101G2 32GB - Roxo	103.62
24	10	iPad Air com Tela Retina Apple Wi-Fi com 32GB, Bluetooth 4.0, Câmera HD, Búss...	1528.22
25	10	iPad Air com Tela Retina Apple Wi-Fi + 3G/4G* com 32GB, Bluetooth 4.0, Câmer...	1869.15

Sub Selects - Correlacionadas

- Exemplo: A subconsulta correlacionada a seguir recupera os produtos que têm preço maior do que a média para seu tipo de produto:
- Para termos certeza.....**

```
SELECT cod_categoria, avg(pco_produto)
FROM produto
GROUP BY cod_categoria;
```

cod_categoria	avg(pco_produto)
2	2087.490000
3	1313.424000
5	2253.277500
8	1264.162500
10	1152.792500
14	59.250000

cod_produto	cod_categoria	nom_produto	pco_produto
1	2	Ultrabook Acer Aspire M5-481T-6650 com Intel® Core™ i3-3227U	2199.99
4	2	Ultrabook Acer Aspire com Intel® Core™ i5	2399.99
6	5	Computador HP ProDesk 600 G1 SFF com Intel® Core™ i3-4130, 4GB, 500GB, ...	2814.81
8	5	Computador HP All in One Pavilion TouchSmart 23-h000br Processador Intel® C...	3229.15
9	3	Notebook Acer Aspire E1-572-6_BR648 com Intel® Core™ i5-4200U, 6GB, 500...	1598.99
11	3	Notebook 2 em 1 Touch HP Pavilion x360 11-n022br Processador Intel® Celero...	1399.00
13	3	Notebook Lenovo G400S com Intel® Core™ i5-3230M, 4GB, 1TB, Gravador de ...	1399.00
15	8	Impressora Jato de Tinta HP Designjet T120 ePrinter Series CQ891A #B1K	3959.00
19	14	Pen Drive SanDisk Ultra® Dual Drive com USB 2.0 e Micro-USB 32GB - Preto	76.42
20	14	Pen Drive Kingston Data Traveler 101G2 32GB - Roxo	103.62
24	10	iPad Air com Tela Retina Apple Wi-Fi com 32GB, Bluetooth 4.0, Câmera HD, Búss...	1528.22
25	10	iPad Air com Tela Retina Apple Wi-Fi + 3G/4G* com 32GB, Bluetooth 4.0, Câmer...	1869.15

Sub Selects - Correlacionadas

- Usamos o apelido externa para rotular a consulta externa e o apelido interna para a subconsulta interna.
- A referência à coluna `cod_categoria` nas partes interna e externa é o que torna a subconsulta interna correlacionada à consulta externa.
- Além disso, a subconsulta retorna uma única linha contendo o preço médio do produto.

Sub Selects - Correlacionadas

- Em uma subconsulta correlacionada, cada linha da consulta externa é passada por vez para a subconsulta.
- Então, ela lê uma linha cada vez da consulta externa e a aplica na subconsulta até que todas as linhas da consulta externa sejam processadas.
- Então, os resultados da consulta inteira são retornados.

Sub Selects - Correlacionadas

- No exemplo anterior, a consulta externa recupera cada linha da tabela produto e passa para a consulta interna.
- Cada linha é lida pela consulta interna, a qual calcula o preço médio de cada produto onde o valor de `cod_categoria` na consulta interna é igual ao valor de `cod_categoria` na consulta externa.

Sub Selects – Exists e Not Exists

- O operador **EXISTS** verifica a existência de linhas retornadas por uma subconsulta. Embora você possa usar EXISTS em subconsultas não correlacionadas, em geral ele é utilizado em subconsultas correlacionadas.
- O operador **NOT EXISTS** executa a lógica oposta de EXISTS: ele verifica se linhas não existem nos resultados retornados por uma subconsulta.

Sub Selects – Exists e Not Exists

- Sintaxe

SELECT

[coluna1, coluna2, ... | *]

FROM

[tabela1, tabela2, ...]

WHERE

EXISTS (

 SELECT

 [coluna1, coluna2, ... | *]

 FROM

 [tabela1, tabela2, ...]

 WHERE [condicao])

Sub Selects – Exists e Not Exists

- O exemplo a seguir usa **EXISTS** para recuperar produtos que foram comprados na promoção; observe que não nos preocupamos com a quantidade de linhas retornadas pela subconsulta, só nos preocupamos em saber se alguma linha é retornada.

```
SELECT cod_produto, nom_produto
FROM produto externa
WHERE EXISTS
(SELECT cod_produto
FROM promocao_produto interna
WHERE interna.cod_produto = externa.cod_produto);
```

cod_produto	nom_produto
1	Ultrabook Acer Aspire M5-481T-6650 com Intel® Core™ i3-3227U

Sub Selects – Exists e Not Exists

- Como EXISTS apenas verifica a existência de linhas retornadas pela subconsulta; uma subconsulta não precisa retornar uma coluna, ela pode retornar apenas um valor literal. Essa característica pode **melhorar o desempenho da sua consulta.**
- Por exemplo, a consulta a seguir reescreve o exemplo anterior com a subconsulta retornando o valor literal 1: Essa mudança, nos exemplos da aula, não faz muita diferença, mas fará com grandes quantidades de registros.

```
SELECT cod_produto, nom_produto
FROM produto externa
WHERE EXISTS
      (SELECT 1
       FROM promocao_produto interna
       WHERE interna.cod_produto = externa.cod_produto);
```

cod_produto	nom_produto
1	Ultrabook Acer Aspire M5-481T-6650 com Intel® Core™ i3-3227U

Sub Selects – Exists e Not Exists

- O exemplo abaixo usa NOT EXISTS para retornar os produtos que não foram comprados em promoção.

```
SELECT cod_produto, nom_produto
FROM produto externa
WHERE NOT EXISTS
  (SELECT cod_produto
   FROM promocao_produto interna
   WHERE interna.cod_produto = externa.cod_produto);
```

24 row(s) returned

1 registro do EXISTS + 24 registros do NOT EXISTS

cod_produto	nom_produto
2	Ultrabook Acer Aspire M5-481T-6417 com Intel® Core™ i5-3317U
3	Ultrabook Acer Aspire M5-481T-6195 com Intel® Core™ i5-3317U
4	Ultrabook Acer Aspire com Intel® Core™ i5
5	Computador HP 402 G1 SFF com Intel Core i3-4130, 4GB, 500GB, Gravador de ...
6	Computador HP ProDesk 600 G1 SFF com Intel® Core™ i3-4130, 4GB, 500GB, ...

Sub Selects (Subquery) de várias linhas

- O exemplo simples a seguir usa **EXISTS** para verificar se um valor de `cod_fornecedor` existe na lista de valores retornada pela subconsulta.

```
select cod_fabricante from fabricante as f
  where EXISTS
    (select cod_fabricante from produto as p where f.cod_fabricante = p.cod_fabricante);
```

	cod_fabricante
▶	1
	2
	3
	4
	5
	6
	o

Sub Selects (Subquery) de várias linhas

- O exemplo simples a seguir usa **NOT EXISTS** para verificar se um valor de `cod_fornecedor` não existe na lista de valores retornada pela subconsulta.

```
select cod_fabricante from fabricante as f
  where NOT EXISTS
    (select cod_fabricante from produto as p where f.cod_fabricante = p.cod_fabricante)
order by 1 desc;
```

Sub Selects – Aninhadas

- Podemos aninhar subconsultas dentro de outras subconsultas até uma profundidade de **255**. Muito raro usar tudo isso.
- Essa técnica deve ser usada com moderação, podemos verificar que sua consulta tem desempenho melhor usando junções de tabela.
- O exemplo a seguir contém uma subconsulta aninhada; observe que ela está contida dentro de uma subconsulta, que por sua vez está contida em uma consulta externa:

Sub Selects – Aninhadas

- Podemos colocar uma subconsulta na cláusula **HAVING** de uma consulta externa. Isso nos permite filtrar grupos de linhas com base no resultado retornado pela subconsulta.
- O exemplo abaixo recupera o valor de cod_categoria e o preço médio dos produtos cujo preço médio é menor do que o máximo da média das categorias dos produtos. Estou usando apenas 3 subconsultas e já causa uma certa confusão.... Imagine usar 255!!!

```
SELECT cod_categoria, AVG(pco_produto)
FROM produto
GROUP BY cod_categoria
HAVING AVG(pco_produto) < (SELECT MAX(media_prod)
                           FROM (SELECT AVG(pco_produto) AS media_prod
                                FROM produto
                                GROUP BY cod_categoria) t);
```


Sub Selects – Aninhadas

- O exemplo abaixo recupera o valor de cod_categoria e o preço médio dos produtos cujo preço médio é menor do que o máximo da média das categorias dos produtos.
- **HAVING é utilizado quando se deseja o filtro depois do agrupamento e não antes.**

```
SELECT cod_categoria, AVG(pco_produto)
FROM produto
GROUP BY cod_categoria
HAVING AVG(pco_produto) < (SELECT MAX(media_prod)
                           FROM (SELECT AVG(pco_produto) AS media_prod
                                FROM produto
                                GROUP BY cod_categoria) t);
```

Sub Selects – Aninhadas

```
SELECT cod_categoria, AVG(pco_produto)
FROM produto
GROUP BY cod_categoria
HAVING AVG(pco_produto) <
(SELECT MAX(media_prod)
FROM (SELECT AVG(pco_produto) AS media_prod
FROM produto
GROUP BY cod_categoria) t);
```



cod_categoria	media_prod
2	2087.490000
3	1313.424000
5	2253.277500
8	1264.162500
10	1152.792500
14	59.250000

- Essa subconsulta retorna a média dos preços por categoria.
- Essas linhas serão retornadas para a seguinte consulta externa.

Sub Selects – Aninhadas

```
SELECT cod_categoria, AVG(pco_produto)
FROM produto
GROUP BY cod_categoria
HAVING AVG(pco_produto) <
(SELECT MAX(media_prod)
FROM (SELECT AVG(pco_produto) AS media_prod
FROM produto
GROUP BY cod_categoria) t);
```

cod_categoria	media_prod
2	2087.490000
3	1313.424000
5	2253.277500
8	1264.162500
10	1152.792500
14	59.250000

MAX(media_prod)
2253.277500

- Essa subconsulta retorna o preço máximo entre a média dos preços por categoria.
- Essa linha é retornada para a seguinte consulta externa.

Sub Selects – Aninhadas

```
SELECT cod_categoria, AVG(pco_produto)
FROM produto
GROUP BY cod_categoria
HAVING AVG(pco_produto) <
(SELECT MAX(media_prod)
FROM (SELECT AVG(pco_produto) AS media_prod
FROM produto
GROUP BY cod_categoria) t);
```

cod_categoria	media_prod
2	2087.490000
3	1313.424000
5	2253.277500
8	1264.162500
10	1152.792500
14	59.250000

MAX(media_prod)
2253.277500

cod_categoria	AVG(pco_produto)
2	2087.490000
3	1313.424000
8	1264.162500
10	1152.792500
14	59.250000

- Finalmente a consulta é executada.

EXERCÍCIO

Criar Banco para funcionário

Esse banco serve para muitos exemplos no Oracle, vamos criá-lo aqui no MySQL.

Para facilitar para vocês quando houver exemplos na internet.

Fazer “dump” do banco funcionário

Referências

Referências

- DEPARTMENT OF COMPUTER SCIENCE (Canadá). University Of Regina. Crow's Foot Notation. 2016. Disponível em: <<http://www2.cs.uregina.ca/~bernatja/crowsfoot.html>>. Acesso em: 22 mar. 2016.
- ELMASRI, Ramez; NAVATHE, Shamkant B.. Parte 7 – Estruturas de arquivo, indexação e hashing: Capítulo 17 – Armazenamento de disco, estruturas de arquivos básicas e hashing. In: ELMASRI, Ramez; NAVATHE, Shamkant B.. Sistema de Banco de Dados. 6. ed. São Paulo: Pearson Education, 2011. Cap. 17. p. 389-423.
- ELMASRI, Ramez; NAVATHE, Shamkant B.. Parte 7 – Estruturas de arquivo, indexação e hashing: Capítulo 18 – Estruturas de arquivos básicas e hashing. In: ELMASRI, Ramez; NAVATHE, Shamkant B.. Sistema de Banco de Dados. 6. ed. São Paulo: Pearson Education, 2011. Cap. 18. p. 424-455.

Referências

- FANDERUFF, D. Índices e Controle de Concorrência. In: FANDERUFF, D. Dominando o Oracle 9i: Modelagem e Desenvolvimento. São Paulo: Pearson Education do Brasil, 2003. Cap. 17. p. 198-202
- MICROSOFT. CREATE TABLE: Transact-SQL. 2019. Disponível em: <<https://docs.microsoft.com/pt-br/sql/t-sql/statements/create-table-transact-sql?view=sql-server-2017>>. Acesso em: 18 jun. 2019.
- MICROSOFT. Tipos de dados: Transact-SQL. 2017. Disponível em: <<https://docs.microsoft.com/pt-br/sql/t-sql/data-types/data-types-transact-sql?view=sql-server-2017>>. Acesso em: 18 jun. 2019.
- RAMAKRISHNAN, Raghu; GEHRKE, Johannes. Parte 3 - Armazenamento e Indexação: Capítulo 10 - Indexação Estruturada em Árvore. In: RAMAKRISHNAN, Raghu; GEHRKE, Johannes. Sistema de Gerenciamento de Banco de Dados. 3. ed. São Paulo: Mc Graw Hill, 2008. Cap. 10. p. 281-308.

Referências

- RAMAKRISHNAN, Raghu; GEHRKE, Johannes. Parte 3 - Armazenamento e Indexação: Capítulo 11 - Indexação Baseada em Hash. In: RAMAKRISHNAN, Raghu; GEHRKE, Johannes. Sistema de Gerenciamento de Banco de Dados. 3. ed. São Paulo: Mc Graw Hill, 2008. Cap. 11. p. 308-321.
- SILBERSCHATZ, Abraham; KORTH, Henry F.; SUNDARSHAN, S.. Parte 2 - Projeto de Banco de Dados: Projeto de Banco de Dados e o Modelo E-R. In: SILBERSCHATZ, Abraham; KORTH, Henry F.; SUNDARSHAN, S.. Sistema de Banco de Dados. 5. ed. Rio de Janeiro: Campus, 2006. Cap. 11. p. 133-174.
- W3SCHOOLS. SQL ALTER TABLE Statement. Disponível em: <https://www.w3schools.com/sql/sql_alter.asp>. Acesso em: 18 jun. 2019.
- _____. SQL DROP TABLE Statement. Disponível em: <https://www.w3schools.com/sql/sql_drop_table.asp>. Acesso em: 18 jun. 2019.
- WIKIPEDIA. Ferramenta CASE. 2016. Disponível em: <https://pt.wikipedia.org/wiki/Ferramenta_CASE>. Acesso em: 22 mar. 2016.
- _____. IDEF1X. 2016. Disponível em: <<https://en.wikipedia.org/wiki/IDEF1X>>. Acesso em: 22 mar. 2016.

Obrigado!

Prof. Clovis Ferraro
(Adaptado de Alexandre Rangel)