

Bancos de Dados

Roteiro de Exercícios 3

Versão: 0.2

Data: nov/2024

Autor: Oclair Prado

Oclair Prado

oclairprado@gmail.br

Objetivos:

- O objetivo fundamental deste material de apoio complementar é fornecer uma oportunidade adicional para o aluno praticar os conceitos estudados em sala de aula.
- Os exercícios estão divididos em ordem crescente de dificuldade e foram distribuídos em seções.
- Para melhor aproveitamento do material fornecido, o aluno deve resolver os exercícios propostos em casa e trazer os resultados obtidos para serem discutidos em sala de aula.

Orientações iniciais:

- Este material é continuação do GADS2022_BD_roteiro_1.pdf e do GADS2022_BD_roteiro_2.pdf, que foram criados considerando as vendas em uma loja de alguns produtos para alguns clientes realizadas por alguns vendedores.
- Consulte os documentos anteriores para orientações sobre a inicialização do SGBD MySQL (Maria DB) e também para o preparo das tabelas com os dados necessários para a realização dos exercícios deste roteiro.

g) Select dentro de select (aninhado), subquery ou subconsulta:

Estrutura básica

```
SELECT    select_list
FROM      table
WHERE     expr operator      (SELECT    select_list
                              FROM      table);
```

Exemplo: mostre os produtos com preço de custo acima da média

The screenshot shows a database query editor with the following SQL query:

```
1 • use aulas;
2 • SELECT pr.cod_prod, pr.nome_prod, pr.valor_custo
3 FROM produto pr
4 WHERE pr.valor_custo > (SELECT AVG(pr.valor_custo) media
5                        FROM produto pr);
```

Below the query editor, the 'Result Grid' is displayed with the following data:

	cod_prod	nome_prod	valor_custo
▶	1	prod_1	24
	3	prod_3	26
	5	prod_5	30
*	NULL	NULL	NULL

Forma alternativa, usando a subconsulta no FROM:

The screenshot shows a database query editor with the following SQL query:

```
1 • use aulas;
2 • SELECT pr.cod_prod, pr.nome_prod, pr.valor_custo, sub.media
3 FROM produto pr, (SELECT AVG(pr.valor_custo) media
4                  FROM produto pr) sub
5 WHERE pr.valor_custo > sub.media
```

Below the query editor, the 'Result Grid' is displayed with the following data:

	cod_prod	nome_prod	valor_custo	media
▶	1	prod_1	24	21.2
	3	prod_3	26	21.2
	5	prod_5	30	21.2

Note que agora é possível usar a nova coluna, “media” criada na sub consulta chamada “sub”, como se fosse uma coluna de uma nova tabela.

g.1) Para a próxima promoção quero dar grande desconto para os produtos mais baratos. Quais são os produtos com valor de venda abaixo da média dos valores de venda?

Resultado esperado:

	cod_prod	nome_prod	valor_custo	valor_venda	media_venda
▶	1	prod_1	24	46	47.6
	2	prod_2	12	42	47.6
	5	prod_5	30	45	47.6

g.2) Também é possível fazer várias subconsultas no FROM, desde que tenham nomes diferentes.

Mostre o valor de custo médio e mínimo dos nossos produtos.

Resultado esperado e exemplo de resolução:

```

1 • USE aulas;
2 • SELECT sub_media.media_custo, sub_min.min_custo
3   FROM (SELECT AVG(pr.valor_custo) media_custo FROM produto pr) sub_media,
4        (SELECT MIN(pr.valor_custo) min_custo FROM produto pr) sub_min;
5
6
7

```

	media_custo	min_custo
▶	21.2	12

g.3) Mostre o valor de custo médio dos produtos vendidos em 2022. Mostre também o valor mínimo e o máximo.

Resultado esperado:

	ano_venda	media_custo	min_custo	max_custo
▶	2022	20.8	14	26

g.4) Encontre o produto com menor valor de venda em 2020 e mostre os clientes que compraram este produto em 2020.

Resultado esperado:

Resultado #1 (2r x 3c)			
fk_cod_cli	ano_venda	valor_venda	
2	2.020	45	
3	2.020	45	

g.5) Encontre o produto com maior valor de custo em 2020 e mostre os vendedores que venderam este produto em 2020.

Resultado esperado:

Resultado #1 (2r x 4c)				
fk_cod_vendedor	ano_venda	valor_custo	valor_venda	
2	2.020	30	45	
4	2.020	30	45	

h) UNION: É possível unir os resultados de duas ou mais consultas com UNION. Mas é necessário que as colunas sejam idênticas.

UNION não traz valores duplicados.

Quando for necessário receber até as linhas duplicadas então deve ser usado UNION ALL.

h.1) Mostre os clientes e os vendedores com seus nomes e CPF.

Resolução e resultado esperado:

```

1  USE aulas;
2  SELECT 'Cliente' tipo, cod_cli cod, fk_cpf CPF FROM cliente
3  UNION
4  SELECT 'Vendedor' tipo, cod_vendedor cod, fk_cpf CPF FROM vendedor

```

cliente (10r x 3c)		
tipo	cod	CPF
Cliente	1	101
Cliente	2	102
Cliente	3	103
Cliente	4	104
Cliente	5	105
Vendedor	1	201
Vendedor	2	202
Vendedor	3	203
Vendedor	4	204
Vendedor	5	205

Notem que foi criada uma nova coluna que informa a origem da informação.

h.2) Mostre os clientes que compraram em 2022 e os vendedores que venderam em 2022.

Resolução e resultado esperado

tipo	ano_venda	cod
Cliente	2.020	2
Cliente	2.020	3
Cliente	2.020	4
Vendedor	2.020	3
Vendedor	2.020	1
Vendedor	2.020	5
Vendedor	2.020	2
Vendedor	2.020	4

tipo	ano_venda	cod
Cliente	2.020	2
Cliente	2.020	3
Cliente	2.020	4
Cliente	2.020	2
Cliente	2.020	3
Vendedor	2.020	3
Vendedor	2.020	1
Vendedor	2.020	5
Vendedor	2.020	2
Vendedor	2.020	4

Diferença devida ao uso do **UNION ALL**

h.3) Mostre os produtos vendidos pelo vendedor 4 em 2021 e os produtos comprados pelo cliente 3 em 2022.

Resultado esperado

tipo	ano_venda	cod	cod_prod	nome_prod
Cliente	2.022	3	4	prod_4
Vendedor	2.021	4	1	prod_1
Vendedor	2.021	4	5	prod_5

h.4) Mostre os produtos com menor preço de venda em 2019 e os produtos com maior preço de venda em 2022.

Resultado esperado

	ano_venda	cod_prod	nome_prod	valor_custo	valor_venda
Mais baratos de 2019	2.019	5	prod_5	30	45
Mais caros de 2022	2.022	3	prod_3	26	53

Agradecimentos:

Agradecemos a colaboração e atenção dos amigos que contribuíram para a elaboração deste material de apoio.

Toda e qualquer contribuição será sempre muito bem-vinda.