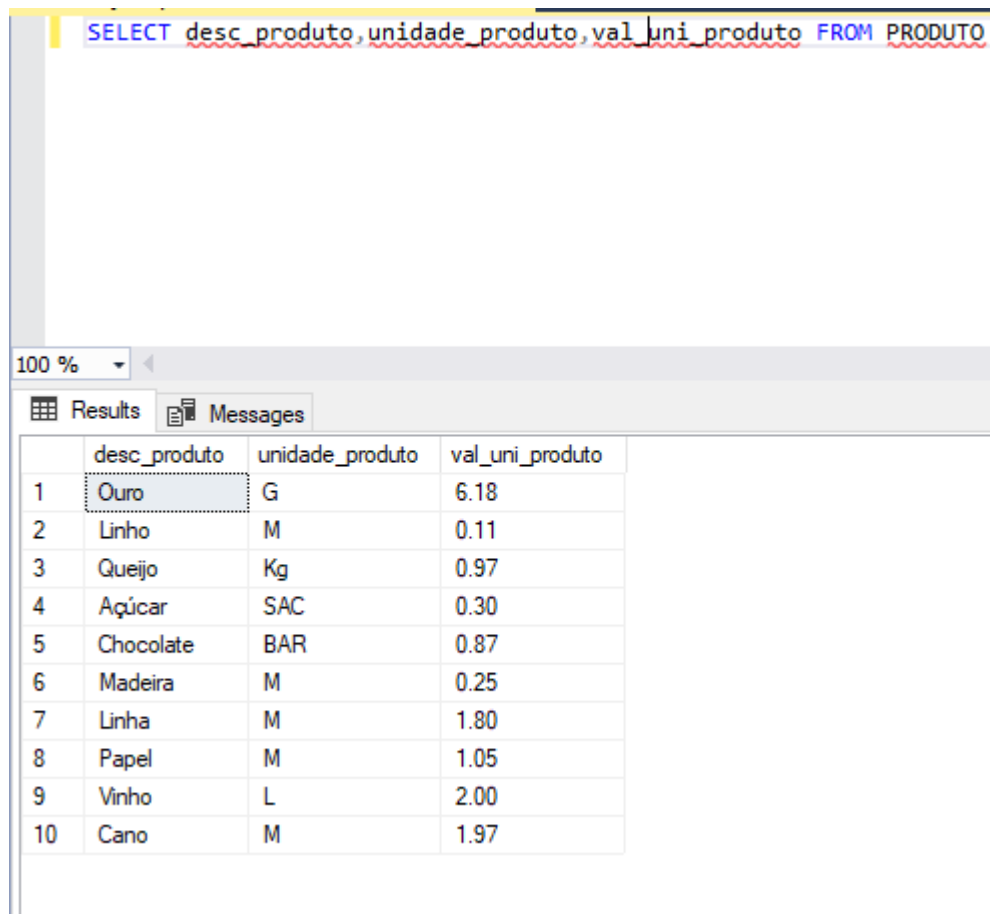


1) Problema: - Listar todos os produtos com respectivas descrições, unidades e valores unitários.

RESP:

```
SELECT desc_produto,unidade_produto,val_uni_produto FROM PRODUTO
```



The screenshot shows a SQL query execution interface. At the top, the query `SELECT desc_produto,unidade_produto,val_uni_produto FROM PRODUTO` is entered. Below the query, there is a zoom level of 100% and two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with 10 rows and 3 columns: `desc_produto`, `unidade_produto`, and `val_uni_produto`. The first row is highlighted.

	desc_produto	unidade_produto	val_uni_produto
1	Ouro	G	6.18
2	Linho	M	0.11
3	Queijo	Kg	0.97
4	Açúcar	SAC	0.30
5	Chocolate	BAR	0.87
6	Madeira	M	0.25
7	Linha	M	1.80
8	Papel	M	1.05
9	Vinho	L	2.00
10	Cano	M	1.97

2) Problema: - Listar da tabela CLIENTE o CNPJ, o nome do cliente e o seu endereço.

RESP:

```
SELECT CNPJ,nome_cliente,endereco FROM CLIENTE
```

SELECT CNPJ, nome_cliente, endereco FROM CLIENTE			
100 %			
Results Messages			
	CNPJ	nome_cliente	endereco
1	3248512673268	Beth	Av. Climério n. 45
2	1451276498349	Jorge	Rua Caiapo 13
3	234632842349	Edmar	Rua da Praia sn/
4	328482233242	Paulo	Tv. Moraes c/3
5	1273657123474	Livia	Av. Beira Mar n.1256
6	2831521393488	Lúcia	Rua Itabira 123 Loja 9
7	2176357612323	Jose	Quadra 3 bl. 3 sl.1003
8	217635712329	Susana	Rua Lopes Mendes 12
9	1327657112314	Renato	Rua Meireles n. 123 b1.2 s1.345
10	321765472133	Sebastião	Rua da Igreja n. 10
11	1283512823469	Rodolfo	Largo da Lapa 27 sobrado
12	12113231000134	Ana	Rua 17 n. 19
13	3281698574656	Mauricio	Av. Paulista 1236 s1/2345
14	225341269378	Flávio	AV. Pres. Vargas 10

3) Problema: - Listar todo o conteúdo de vendedor

RESP:

```
SELECT * FROM VENDEDOR
```

SELECT * FROM <u>VENDEDOR</u>				
00 %				
Results Messages				
	id_vendedor	nome_vendedor	salario_fixo	faixa_comissao
1	11	Joao	2780.00	C
2	101	João	2650.00	C
3	111	Carlos	2490.00	A
4	209	José	1800.00	C
5	213	Jonas	2300.00	A
6	240	Antônio	9500.00	C
7	250	Mauricio	2930.00	B
8	310	Josias	870.00	B
9	720	Felipe	4600.00	A

4) O SQL permite que se apresente a saída de um SELECT com cabeçalhos de colunas ao nosso gosto

RESP:

```
SELECT id_vendedor AS Numero, nome_vendedor AS Nome, salario_fixo
AS Rendimentos, faixa_comissao AS Comicao FROM VENDEDOR
```

SQLQuery1.sql - CT...H (BR\DISRCT (64))* - X

```
SELECT id_vendedor AS Numero, nome_vendedor AS Nome, salario_fixo AS Rendimentos, faixa_comissao AS Comissao FROM VENDEDOR
```

100 %

Results Messages

	Numero	Nome	Rendimentos	Comissao
1	11	Joao	2780.00	C
2	101	João	2650.00	C
3	111	Carlos	2490.00	A
4	209	José	1800.00	C
5	213	Jonas	2300.00	A
6	240	Antônio	9500.00	C
7	250	Mauricio	2930.00	B
8	310	Josias	870.00	B
9	720	Felipe	4600.00	A

5) Traga o nome do vendedor e o salário fixo multiplicado por 2

RESP:

```
SELECT nome_vendedor, salario_fixo*2 FROM VENDEDOR
```

```
SELECT nome_vendedor, salario_fixo*2 FROM VENDEDOR
```

100 %

Results Messages

	nome_vendedor	(No column name)
1	Joao	5560.00
2	João	5300.00
3	Carlos	4980.00
4	José	3600.00
5	Jonas	4600.00
6	Antônio	19000.00
7	Mauricio	5860.00
8	Josias	1740.00
9	Felipe	9200.00

6) Problema: - Listar o num_pedido, o código_produto e a quantidade dos itens do pedido com a quantidade igual a 35 da tabela item_do_pedido.

RESP:

```
SELECT id_pedido, id_produto, quantidade FROM ITEM_PEDIDO WHERE quantidade = 35
```

```
SELECT id_pedido,id_produto,quantidade FROM ITEM_PEDIDO WHERE quantidade = 35
```

100 %

Results Messages

	id_pedido	id_produto	quantidade
1	138	77	35
2	121	31	35

7) Problema: - liste os nomes e a cidade dos clientes que moram em Niterói?

RESP:

```
SELECT nome_cliente,cidade FROM CLIENTE WHERE cidade = 'Niteroi'
```

```
SELECT nome_cliente,cidade FROM CLIENTE WHERE cidade = 'Niteroi'
```

100 %

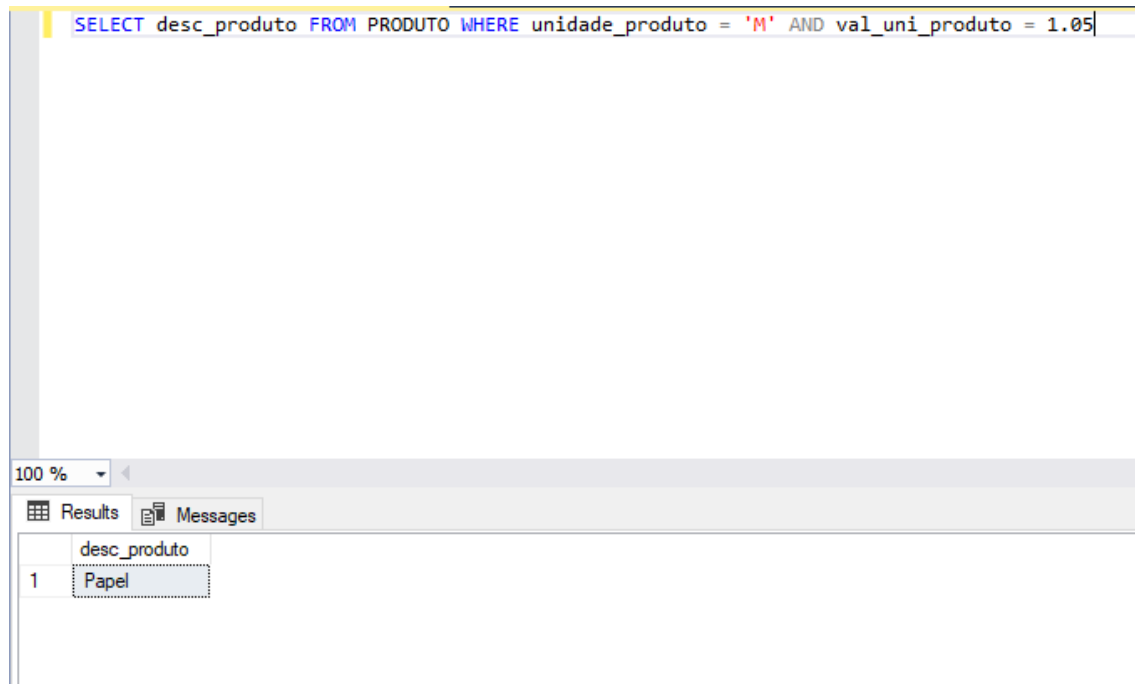
Results Messages

	nome_cliente	cidade
1	Susana	Niteroi
2	Ana	Niteroi

8) Problema: - Listar os produtos que tenham unidade igual a 'M' e valor unitário igual a R\$ 1,05 da tabela produto

RESP:

```
SELECT desc_produto FROM PRODUTO WHERE unidade_produto = 'M' AND val_uni_produto = 1.05
```



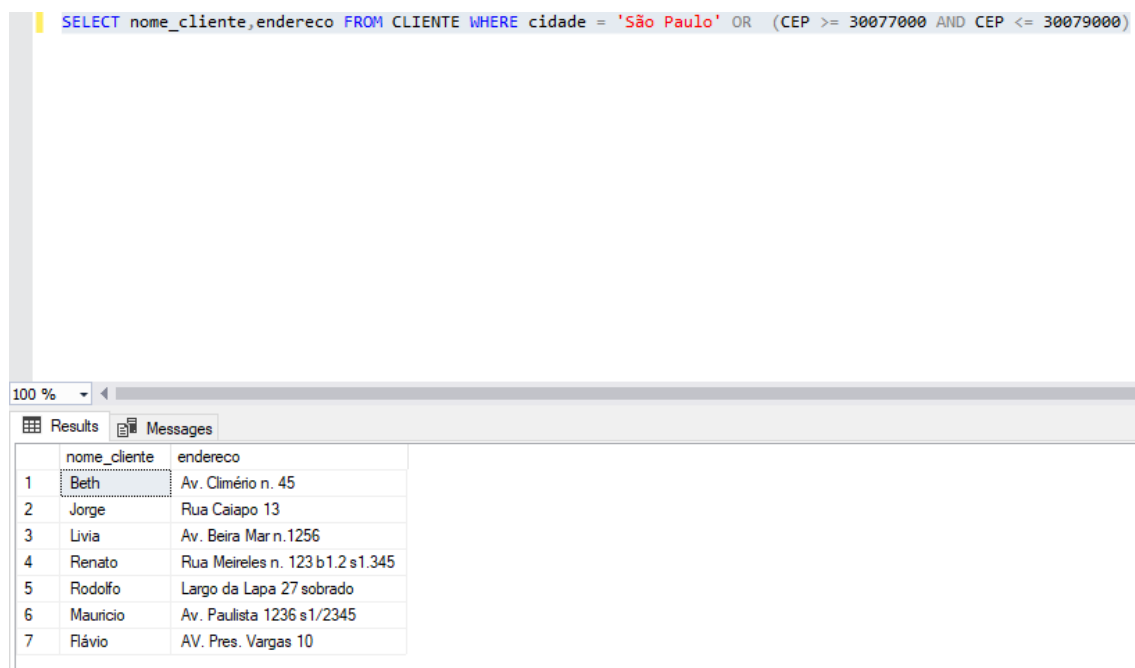
```
SELECT desc_produto FROM PRODUTO WHERE unidade_produto = 'M' AND val_uni_produto = 1.05
```

	desc_produto
1	Papel

9) Problema: Liste os clientes e seus respectivos endereços, que moram em 'SAO PAULO' ou estejam na faixa de CEP entre '30077000' e '30079000'.

RESP:

```
SELECT nome_cliente,endereco FROM CLIENTE WHERE cidade = 'São Paulo' OR (CEP >= 30077000 AND CEP <= 30079000)
```



```
SELECT nome_cliente,endereco FROM CLIENTE WHERE cidade = 'São Paulo' OR (CEP >= 30077000 AND CEP <= 30079000)
```

	nome_cliente	endereco
1	Beth	Av. Clímério n. 45
2	Jorge	Rua Caiapo 13
3	Livia	Av. Beira Mar n.1256
4	Renato	Rua Meireles n. 123 b1.2 s1.345
5	Rodolfo	Largo da Lapa 27 sobrado
6	Maurício	Av. Paulista 1236 s1/2345
7	Flávio	AV. Pres. Vargas 10

10) Problema: - Mostrar todos os pedidos que não tenham prazo de entrega igual a 15 dias.

RESP:

```
SELECT id_pedido FROM PEDIDO WHERE prazo_entrega != 15
```

```
SELECT id_pedido FROM PEDIDO WHERE prazo_entrega != 15
```

100 %	
Results Messages	
	id_pedido
1	91
2	97
3	98
4	103
5	104
6	111
7	119
8	121
9	127
10	137
11	138
12	143
13	148
14	203

11) Problema: - Listar o código e a descrição dos produtos que tenham o valor unitário na faixa de R\$ 0,32 até R\$ 2,00.

RESP:

```
SELECT id_produto, desc_produto FROM PRODUTO WHERE val_uni_produto BETWEEN 0.32 AND 2.00
```

```
SELECT id_produto, desc_produto FROM PRODUTO WHERE val_uni_produto BETWEEN 0.32 AND 2.00
```

100 %

Results

Messages

	id_produto	desc_produto
1	25	Queijo
2	31	Chocolate
3	53	Linha
4	77	Papel
5	78	Vinho
6	87	Cano

12) Problema: - Listar todos os produtos que tenham o seu nome começando por Q.

RESP:

```
SELECT * FROM PRODUTO WHERE desc_produto LIKE 'Q%'
```

```
SELECT * FROM PRODUTO WHERE desc_produto LIKE 'Q%'
```

100 %



Results



Messages

	id_produto	unidade_produto	desc_produto	val_uni_produto
1	25	Kg	Queijo	0.97

13) Problema: - Listar os vendedores que não começam por 'Jo'.

RESP:

```
SELECT * FROM VENDEDOR WHERE nome_vendedor NOT LIKE 'Jo%'
```

```
SELECT * FROM VENDEDOR WHERE nome_vendedor NOT LIKE 'Jo%'
```

100 %



Results



Messages

	id_vendedor	nome_vendedor	salario_fixo	faixa_comissao
1	111	Carlos	2490.00	A
2	240	Antônio	9500.00	C
3	250	Mauricio	2930.00	B
4	720	Felipe	4600.00	A

14) Problema: - Listar os vendedores em ordem alfabética e que são da faixa de comissão A e B.

RESP:

```
SELECT * FROM VENDEDOR WHERE faixa_comissao = 'A' OR faixa_comissao = 'B' ORDER BY  
nome_vendedor
```



```
SELECT * FROM VENDEDOR WHERE faixa_comissao = 'A' OR faixa_comissao = 'B' ORDER BY nome_vendedor
```

100 %

Results Messages

	id_vendedor	nome_vendedor	salario_fixo	faixa_comissao
1	111	Carlos	2490.00	A
2	720	Felipe	4600.00	A
3	213	Jonas	2300.00	A
4	310	Josias	870.00	B
5	250	Mauricio	2930.00	B

15) Problema: - Mostrar os clientes que não tenham inscrição estadual.

RESP:

```
SELECT * FROM CLIENTE WHERE IE IS NULL
```

```
SELECT * FROM CLIENTE WHERE IE IS NULL
```

100 %

Results Messages

	id_cliente	nome_cliente	endereco	cidade	CEP	UF	CNPJ	IE
1	110	Jorge	Rua Caiapo 13	Curitiba	30078500	PR	1451276498349	NULL
2	180	Livia	Av. Beira Mar n.1256	Florianopolis	30077500	SC	1273657123474	NULL

16) Problema: - Mostrar em ordem alfabética a lista de vendedores e seus respectivos salários fixos.

```
SELECT nome_vendedor, salario_fixo FROM VENDEDOR ORDER BY nome_vendedor
```

```
SELECT nome_vendedor,salario_fixo FROM VENDEDOR ORDER BY nome_vendedor
```

100 %

Results Messages

	nome_vendedor	salario_fixo
1	Antônio	9500.00
2	Carlos	2490.00
3	Felipe	4600.00
4	Joao	2780.00
5	João	2650.00
6	Jonas	2300.00
7	José	1800.00
8	Josias	870.00
9	Mauricio	2930.00

17) Problema: - Listar os nomes, cidades e estados de todos os clientes, ordenados por estado e cidade de forma descendente.

```
SELECT nome_cliente,cidade,UF FROM CLIENTE ORDER BY UF DESC,cidade DESC
```

```
SELECT nome_cliente,cidade,UF FROM CLIENTE ORDER BY UF DESC,cidade DESC
```

100 %

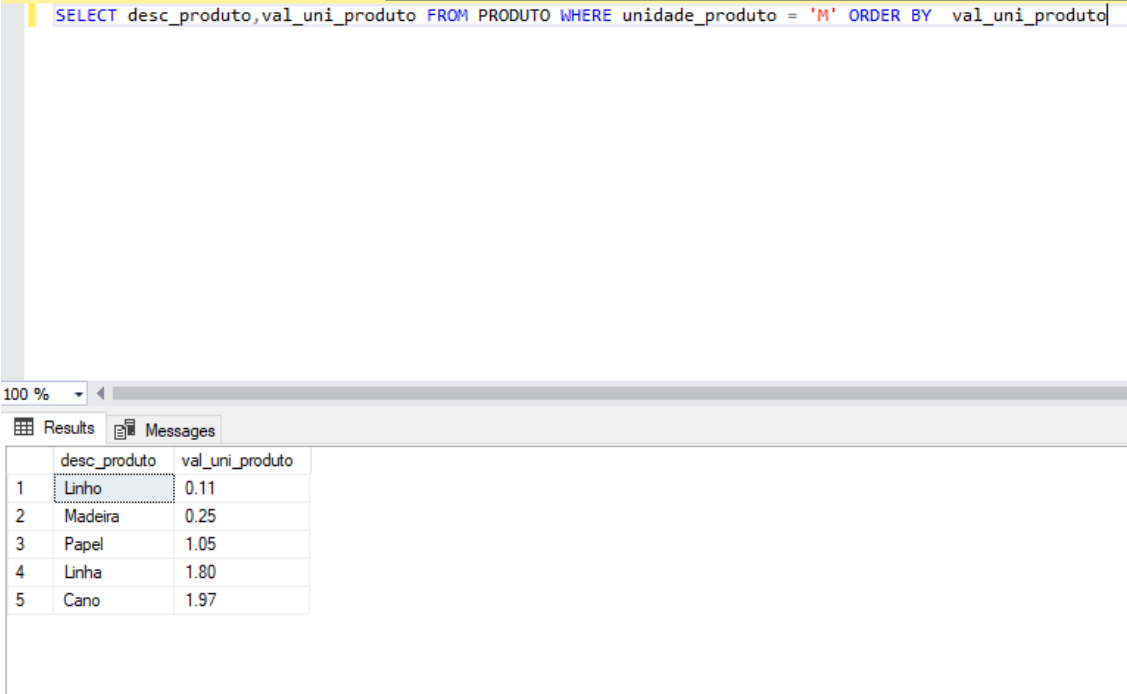
Results Messages

	nome_cliente	cidade	UF
1	Beth	São Paulo	SP
2	Renato	São Paulo	SP
3	Mauricio	São Paulo	SP
4	Flávio	São Paulo	SP
5	Livia	Florianopolis	SC
6	Rodolfo	Rio de Janeiro	RJ
7	Ana	Niteroi	RJ
8	Susana	Niteroi	RJ
9	Paulo	Londrina	PR
10	Jorge	Curitiba	PR
11	Sebastião	Uberaba	MG
12	Lúcia	Belo Horizonte	MG
13	Jose	Brasília	DF
14	Edmar	Salvador	BA

18) Problema: - Mostrar a descrição e o valor unitário de todos os produtos que tenham a unidade 'M', em ordem de valor unitário ascendente

RESP:

```
SELECT desc_produto, val_uni_produto FROM PRODUTO WHERE unidade_produto = 'M' ORDER BY val_uni_produto
```

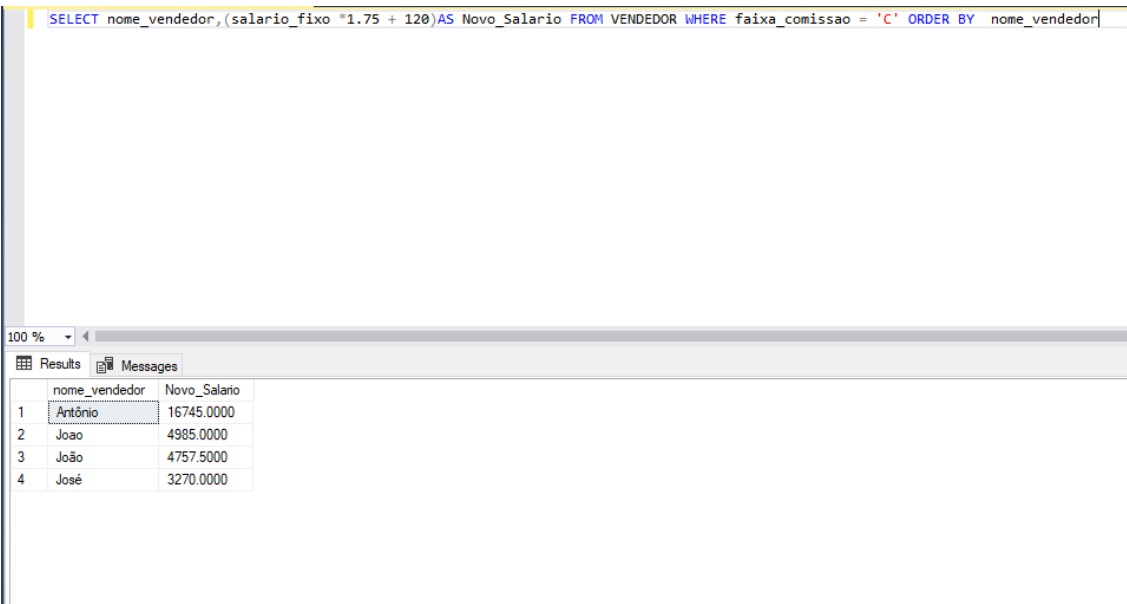


The screenshot shows a SQL query editor with the query: `SELECT desc_produto, val_uni_produto FROM PRODUTO WHERE unidade_produto = 'M' ORDER BY val_uni_produto`. Below the query, the results are displayed in a table with two columns: `desc_produto` and `val_uni_produto`. The results are ordered by `val_uni_produto` in ascending order.

	desc_produto	val_uni_produto
1	Linho	0.11
2	Madeira	0.25
3	Papel	1.05
4	Linha	1.80
5	Cano	1.97

19) Problema: - Mostrar o novo salário fixo dos vendedores, de faixa de comissão 'C', calculado com base no reajuste de 75% acrescido de R\$ 120.00 de bonificação. Ordenar pelo nome do vendedor.

```
SELECT nome_vendedor, (salario_fixo * 1.75 + 120) AS Novo_Salario FROM VENDEDOR WHERE faixa_comissao = 'C' ORDER BY nome_vendedor
```



The screenshot shows a SQL query editor with the query: `SELECT nome_vendedor, (salario_fixo * 1.75 + 120) AS Novo_Salario FROM VENDEDOR WHERE faixa_comissao = 'C' ORDER BY nome_vendedor`. Below the query, the results are displayed in a table with two columns: `nome_vendedor` and `Novo_Salario`. The results are ordered by `nome_vendedor`.

	nome_vendedor	Novo_Salario
1	Antônio	16745.0000
2	Joao	4985.0000
3	João	4757.5000
4	José	3270.0000

20) Problema: Mostrar o menor e o maior salários da tabela vendedor.
RESP:

```
SELECT MIN(salario_fixo) AS 'Minimo', MAX(salario_fixo) AS 'Maximo' FROM VENDEDOR
```

```
SELECT MIN(salario_fixo) AS 'Minimo', MAX(salario_fixo) AS 'Maximo' FROM VENDEDOR
```

100 %

Results			Messages		
	Minimo	Maximo			
1	870.00	9500.00			

21) Problema: - Mostrar a quantidade total pedida para o produto 'VINHO' de código 78' na tabela item_de_pedido.

```
SELECT SUM(quantidade) AS 'Quantidade de Vinho' FROM ITEM_PEDIDO WHERE id_produto = (SELECT id_produto FROM PRODUTO WHERE desc_produto = 'Vinho')
```

```
SELECT SUM(quantidade) AS 'Quantidade de Vinho' FROM ITEM_PEDIDO WHERE id_produto = (SELECT id_produto FROM PRODUTO WHERE desc_produto = 'Vinho')
```

6

Results		Messages	
	Quantidade de Vinho		
	183		

22) Problema: - Qual a média dos salários fixos dos vendedores?

```
SELECT AVG(salario_fixo) AS 'Media de Salarios' FROM VENDEDOR
```

```
SELECT AVG(salario_fixo) AS 'Media de Salarios' FROM VENDEDOR
```

Media de Salarios
3324.444444

23) Problema: - Quantos vendedores ganham acima de R\$ 2.500,00 de salário fixo?

```
SELECT COUNT(salario_fixo) AS 'Salario > 2500' FROM VENDEDOR WHERE salario_fixo > 2500
```

```
SELECT COUNT(salario_fixo) AS 'Salario > 2500' FROM VENDEDOR WHERE salario_fixo > 2500
```

Salario > 2500
5

24) ♦ Problema: - Quais as unidades de produtos, diferentes, na tabela produto?

```
SELECT DISTINCT unidade_produto FROM PRODUTO
```

SELECT DISTINCT unidade_produto FROM PRODUTO	
100 %	
Results	Messages
	unidade_produto
1	BAR
2	G
3	Kg
4	L
5	M
6	SAC

25) Problema: - Listar o número de produtos que cada pedido contém.

```
SELECT id_pedido, COUNT(id_produto) FROM ITEM_PEDIDO GROUP BY id_pedido
```

SELECT id_pedido, COUNT(id_produto) FROM ITEM_PEDIDO GROUP BY id_pedido HAVING COUNT(id_produto) > 3	
133 %	
Results	Messages
	id_pedido (No column name)
1	119 4
2	148 5

26) Problema: - Listar os pedidos que têm mais do que três produtos.

```
SELECT id_pedido, COUNT(id_produto) FROM ITEM_PEDIDO GROUP BY id_pedido HAVING COUNT(id_produto) > 3
```

SELECT id_pedido, COUNT(id_produto) FROM ITEM_PEDIDO GROUP BY id_pedido HAVING COUNT(id_produto) > 3	
110 %	
Results	Messages
	id_pedido (No column name)
1	119 4
2	148 5

27) Problema: Ver os pedidos de cada cliente

```
SELECT CLIENTE.id_cliente,CLIENTE.nome_cliente,PEDIDO.id_pedido
FROM PEDIDO
INNER JOIN CLIENTE ON PEDIDO.id_cliente = CLIENTE.id_cliente;
```

```
SELECT CLIENTE.id_cliente,CLIENTE.nome_cliente,PEDIDO.id_pedido
FROM PEDIDO
INNER JOIN CLIENTE ON PEDIDO.id_cliente = CLIENTE.id_cliente;
```

110 %

	id_cliente	nome_cliente	id_pedido
1	260	Susana	91
2	720	Ana	97
3	410	Rodolfo	98
4	720	Ana	101
5	260	Susana	103
6	110	Jorge	104
7	180	Livia	105
8	290	Renato	108
9	260	Susana	111
10	390	Sebastião	119
11	410	Rodolfo	121
12	410	Rodolfo	127
13	720	Ana	137
14	260	Susana	138
15	20	Beth	143
16	720	Ana	148
17	870	Flávio	189
18	830	Mauricio	203

28) Problema: - Juntar Clientes com Pedidos.

```
SELECT CLIENTE.id_cliente,CLIENTE.nome_cliente,PEDIDO.id_pedido
FROM PEDIDO,CLIENTE;
```

```
SELECT CLIENTE.id_cliente,CLIENTE.nome_cliente,PEDIDO.id_pedido  
FROM PEDIDO,CLIENTE;
```

110 %

Results Messages

	id_cliente	nome_cliente	id_pedido
1	20	Beth	91
2	20	Beth	97
3	20	Beth	98
4	20	Beth	101
5	20	Beth	103
6	20	Beth	104
7	20	Beth	105
8	20	Beth	108
9	20	Beth	111
10	20	Beth	119
11	20	Beth	121
12	20	Beth	127
13	20	Beth	137
14	20	Beth	138
15	20	Beth	143
16	20	Beth	148
17	20	Beth	189
18	20	Beth	203
19	110	Jorge	91
20	110	Jorge	97

29) Problema: Quais são os clientes que têm pedido e os que não têm pedido.

```
SELECT CLIENTE.nome_cliente,PEDIDO.id_cliente,PEDIDO.id_pedido  
FROM CLIENTE  
LEFT JOIN PEDIDO  
ON CLIENTE.id_cliente = PEDIDO.id_cliente;
```



```
SELECT CLIENTE.nome_cliente,PEDIDO.id_cliente,PEDIDO.id_pedido
FROM CLIENTE
LEFT JOIN PEDIDO
ON CLIENTE.id_cliente = PEDIDO.id_cliente;
```

75 %

Results Messages

	nome_cliente	id_cliente	id_pedido
1	Beth	20	143
2	Jorge	110	104
3	Edmar	NULL	NULL
4	Paulo	NULL	NULL
5	Livia	180	105
6	Lúcia	NULL	NULL
7	Jose	NULL	NULL
8	Susana	260	91
9	Susana	260	103
10	Susana	260	111
11	Susana	260	138
12	Renato	290	108
13	Sebastião	390	119
14	Rodolfo	410	98
15	Rodolfo	410	121
16	Rodolfo	410	127
17	Ana	720	97
18	Ana	720	101
19	Ana	720	137
20	Ana	720	148
21	Mauricio	830	203
22	Flávio	870	189

30) Problema: - Quais clientes têm prazo de entrega superior a 15 dias e pertencem aos estados de São Paulo ('SP') ou Rio de Janeiro ('RJ')?

```
SELECT DISTINCT CLIENTE.nome_cliente, CLIENTE.UF, PEDIDO.prazo_entrega
FROM PEDIDO
INNER JOIN CLIENTE ON PEDIDO.id_cliente = CLIENTE.id_cliente
WHERE (CLIENTE.UF = 'SP' OR CLIENTE.UF='RJ') AND PEDIDO.prazo_entrega > 15;
```

```

SELECT DISTINCT CLIENTE.nome_cliente, CLIENTE.UF, PEDIDO.prazo_entrega
FROM PEDIDO
INNER JOIN CLIENTE ON PEDIDO.id_cliente = CLIENTE.id_cliente
WHERE (CLIENTE.UF = 'SP' OR CLIENTE.UF='RJ') AND PEDIDO.prazo_entrega > 15;

```

121 %

Results Messages

	nome_cliente	UF	prazo_entrega
1	Ana	RJ	20
2	Beth	SP	30
3	Mauricio	SP	30
4	Rodolfo	RJ	20
5	Susana	RJ	20

31) Problema - Mostrar os clientes e seus respectivos prazos de entrega, ordenados do maior para o menor

```

SELECT DISTINCT CLIENTE.nome_cliente, PEDIDO.prazo_entrega
FROM PEDIDO
INNER JOIN CLIENTE ON PEDIDO.id_cliente = CLIENTE.id_cliente
ORDER BY PEDIDO.prazo_entrega DESC;

```

```

SELECT DISTINCT CLIENTE.nome_cliente, PEDIDO.prazo_entrega
FROM PEDIDO
INNER JOIN CLIENTE ON PEDIDO.id_cliente = CLIENTE.id_cliente
ORDER BY PEDIDO.prazo_entrega DESC;

```

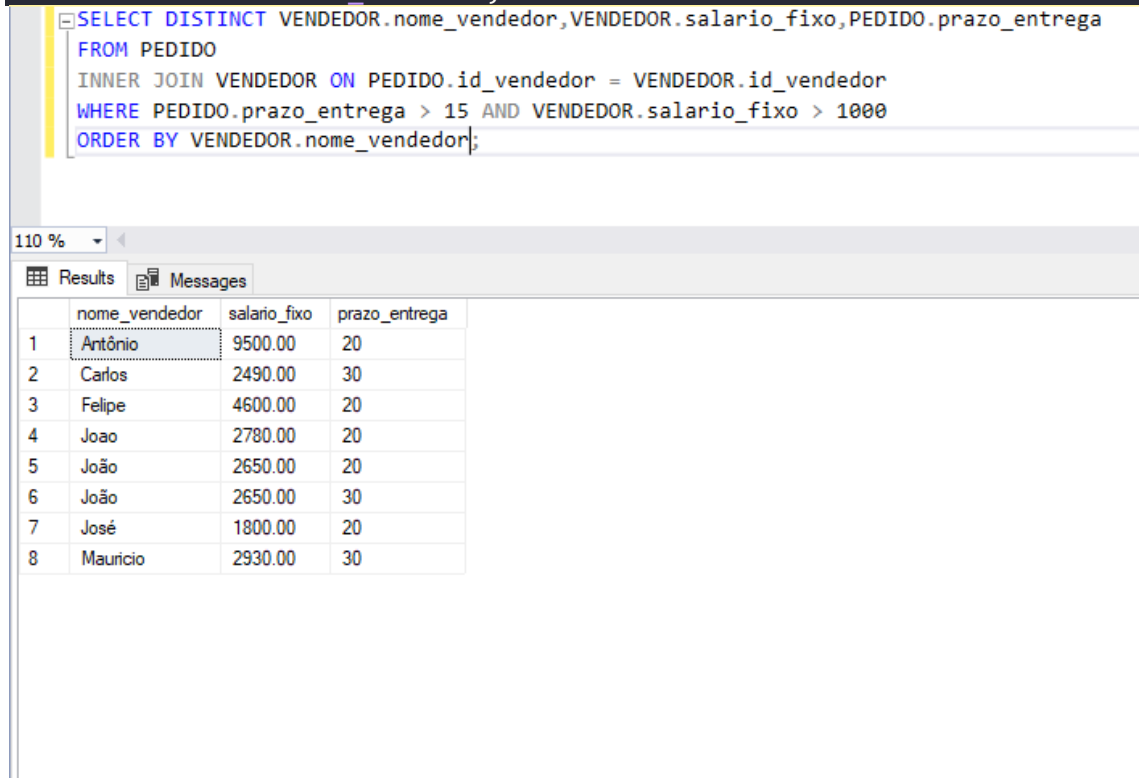
121 %

Results Messages

	nome_cliente	prazo_entrega
1	Beth	30
2	Jorge	30
3	Mauricio	30
4	Sebastião	30
5	Ana	20
6	Rodolfo	20
7	Susana	20
8	Ana	15
9	Flávio	15
10	Livia	15
11	Renato	15
12	Rodolfo	10

32) Problema: - Apresentar os vendedores (ordenados) que emitiram pedidos com prazos de entrega superiores a 15 dias e tenham salários fixos iguais ou superiores a R\$ 1.000,00.

```
SELECT DISTINCT VENDEDOR.nome_vendedor, VENDEDOR.salario_fixo, PEDIDO.prazo_entrega
FROM PEDIDO
INNER JOIN VENDEDOR ON PEDIDO.id_vendedor = VENDEDOR.id_vendedor
WHERE PEDIDO.prazo_entrega > 15 AND VENDEDOR.salario_fixo > 1000
ORDER BY VENDEDOR.nome_vendedor;
```



The screenshot shows a database query interface. At the top, the SQL query is displayed in a text area. Below the query, there are tabs for 'Results' and 'Messages'. The 'Results' tab is active, showing a table with 4 columns: 'nome_vendedor', 'salario_fixo', and 'prazo_entrega'. The table contains 8 rows of data, numbered 1 to 8. The first row is highlighted with a dashed border.

	nome_vendedor	salario_fixo	prazo_entrega
1	Antônio	9500.00	20
2	Carlos	2490.00	30
3	Felipe	4600.00	20
4	Joao	2780.00	20
5	João	2650.00	20
6	João	2650.00	30
7	José	1800.00	20
8	Mauricio	2930.00	30

33) Problema: - Mostre os clientes (ordenados) que têm prazo de entrega maior que 15 dias para o produto 'QUEIJO' e sejam do Rio de Janeiro.

```
SELECT DISTINCT CLIENTE.nome_cliente
FROM PEDIDO
INNER JOIN CLIENTE ON PEDIDO.id_cliente = CLIENTE.id_cliente
FULL OUTER JOIN ITEM_PEDIDO ON PEDIDO.id_pedido = ITEM_PEDIDO.id_pedido
WHERE prazo_entrega > 15 AND ITEM_PEDIDO.id_produto = 25 AND UF = 'RJ'
ORDER BY CLIENTE.nome_cliente;
```

```
SELECT DISTINCT CLIENTE.nome_cliente
FROM PEDIDO
INNER JOIN CLIENTE ON PEDIDO.id_cliente = CLIENTE.id_cliente
FULL OUTER JOIN ITEM_PEDIDO ON PEDIDO.id_pedido = ITEM_PEDIDO.id_pedido
WHERE prazo_entrega > 15 AND ITEM_PEDIDO.id_produto = 25 AND UF = 'RJ'
ORDER BY CLIENTE.nome_cliente;
```

110 %

Results Messages

	nome_cliente
1	Ana
2	Rodolfo
3	Susana

34) Problema: - Mostre todos os vendedores que venderam chocolate em quantidade superior a 10 Kg

```
SELECT DISTINCT VENDEDOR.nome_vendedor
FROM PEDIDO
INNER JOIN VENDEDOR ON PEDIDO.id_vendedor = VENDEDOR.id_vendedor
FULL OUTER JOIN ITEM_PEDIDO ON PEDIDO.id_pedido = ITEM_PEDIDO.id_pedido
WHERE prazo_entrega > 15 AND ITEM_PEDIDO.id_produto = (SELECT id_produto FROM PRODUTO WHERE desc_produto = 'Chocolate') AND quantidade > 10
```

```
SELECT DISTINCT VENDEDOR.nome_vendedor
FROM PEDIDO
INNER JOIN VENDEDOR ON PEDIDO.id_vendedor = VENDEDOR.id_vendedor
FULL OUTER JOIN ITEM_PEDIDO ON PEDIDO.id_pedido = ITEM_PEDIDO.id_pedido
WHERE prazo_entrega > 15 AND ITEM_PEDIDO.id_produto = (SELECT id_produto FROM PRODUTO WHERE desc_produto = 'Chocolate') AND
```

110 %

Results Messages

	nome_vendedor
1	Carlos
2	José

35) Problema: - Quantos clientes fizeram pedido com o vendedor João?

RESP:

```
SELECT COUNT(DISTINCT id_cliente) FROM PEDIDO
WHERE id_vendedor = (SELECT id_vendedor FROM VENDEDOR WHERE nome_vendedor = 'João')
```

133 %

Results Messages

	(No column name)
1	2

36) Problema: - Quantos clientes da cidade do Rio de Janeiro e de Niterói tiveram seus pedidos tirados com o vendedor João? (Exiba o nome das Cidades e a quantidade.)

RESP:

```
SELECT CLIENTE.cidade, COUNT( DISTINCT CLIENTE.id_cliente)AS 'TOTAL'
FROM PEDIDO
INNER JOIN CLIENTE ON PEDIDO.id_cliente = CLIENTE.id_cliente
FULL OUTER JOIN VENDEDOR ON PEDIDO.id_vendedor = VENDEDOR.id_vendedor
WHERE cidade = 'Rio de Janeiro' OR cidade = 'Niteroi' AND VENDEDOR.nome_vendedor = 'João'
GROUP BY cidade
```

133 %

Results Messages

	cidade	TOTAL
1	Niteroi	1
2	Rio de Janeiro	1

37) Problema: - Listar todo o conteúdo de vendedor

RESP:

```
SELECT * FROM VENDEDOR
```

```
SELECT * FROM VENDEDOR
```

133 %

Results Messages

	id_vendedor	nome_vendedor	salario_fixo	faixa_comissao
1	11	João	2780.00	C
2	101	João V	2650.00	C
3	111	Carlos	2490.00	A
4	209	José	1800.00	C
5	213	Jonas	2300.00	A
6	240	Antônio	9500.00	C
7	250	Mauricio	2930.00	B
8	310	Josias	870.00	B
9	720	Felipe	4600.00	A

38) Que produtos participam de qualquer pedido cuja quantidade seja 10? (Exibe os nomes – **Uso consulta encadeada, utilize IN**)

RESP:

```
SELECT DISTINCT desc_produto FROM PRODUTO  
INNER JOIN ITEM_PEDIDO ON PRODUTO.id_produto = ITEM_PEDIDO.id_produto  
WHERE quantidade IN(10)
```

```
SELECT DISTINCT desc_produto FROM PRODUTO  
INNER JOIN ITEM_PEDIDO ON PRODUTO.id_produto = ITEM_PEDIDO.id_produto  
WHERE quantidade IN(10)
```

133 %

Results Messages

	desc_produto
1	Linho
2	Queijo
3	Vinho

39) Quais vendedores ganham um salário fixo abaixo da média? (Exibe os nomes
– **Uso consulta encadeada, utilize AVG**)

RESP:

```
SELECT nome_vendedor FROM VENDEDOR WHERE salario_fixo < (SELECT AVG(salario_fixo) FROM VENDEDOR)
```

```
SELECT nome_vendedor FROM VENDEDOR WHERE salario_fixo < (SELECT AVG(salario_fixo) FROM VENDE
```

133 %
Results Messages

	nome_vendedor
1	João
2	João V
3	Carlos
4	José
5	Jonas
6	Mauricio
7	Josias

40) Problema: Quais os produtos que não estão presentes em nenhum pedido?
(**Uso consulta encadeada, utilize NOT IN**)

RESP:

```
SELECT * FROM PRODUTO WHERE id_produto NOT IN(SELECT id_produto FROM ITEM_PEDIDO)
```

```
SELECT * FROM PRODUTO WHERE id_produto NOT IN(SELECT id_produto FROM ITEM_PEDIDO)|
```

133 %
Results Messages

	id_produto	unidade_produto	desc_produto	val_uni_produto
1	30	SAC	Açúcar	0.30
2	87	M	Cano	1.97

41) Problema: - Quais vendedores que só venderam produtos por grama(G)?
(Exibir o Cod do Vendedor e o nome)

RESP:

```
SELECT DISTINCT VENDEDOR.id_vendedor, nome_vendedor
FROM VENDEDOR
INNER JOIN PEDIDO ON PEDIDO.id_vendedor = VENDEDOR.id_vendedor
FULL OUTER JOIN ITEM_PEDIDO ON PEDIDO.id_pedido = ITEM_PEDIDO.id_pedido
FULL OUTER JOIN PRODUTO ON ITEM_PEDIDO.id_produto = PRODUTO.id_produto
WHERE unidade_produto = 'G'
```

```
SELECT DISTINCT VENDEDOR.id_vendedor, nome_vendedor
FROM VENDEDOR
INNER JOIN PEDIDO ON PEDIDO.id_vendedor = VENDEDOR.id_vendedor
FULL OUTER JOIN ITEM_PEDIDO ON PEDIDO.id_pedido = ITEM_PEDIDO.id_pedido
FULL OUTER JOIN PRODUTO ON ITEM_PEDIDO.id_produto = PRODUTO.id_produto
WHERE unidade_produto = 'G'
```

133 %

Results Messages

	id_vendedor	nome_vendedor
1	101	João V
2	250	Mauricio
3	310	Josias
4	720	Felipe

42) Problema: - Quais Clientes estão presentes em mais de TRÊS pedidos? (Exibir os nomes)

RESP:

```
SELECT CLIENTE.nome_cliente, COUNT(PEDIDO.id_cliente) FROM PEDIDO
INNER JOIN CLIENTE ON PEDIDO.id_cliente = CLIENTE.id_cliente
GROUP BY nome_cliente
HAVING COUNT(PEDIDO.id_cliente) > 3
```

```
SELECT CLIENTE.nome_cliente, COUNT(PEDIDO.id_cliente) FROM PEDIDO
INNER JOIN CLIENTE ON PEDIDO.id_cliente = CLIENTE.id_cliente
GROUP BY nome_cliente
HAVING COUNT(PEDIDO.id_cliente) > 3
```

121 %

Results Messages

	nome_cliente	(No column name)
1	Ana	4
2	Susana	4

43) Problema: Adicione o produto “**PARAFUSO → Kg , valor R\$1,25. Cod 108**” a tabela **PRODUTO**

RESP:

```
INSERT INTO PRODUTO(id_produto,unidade_produto,desc_produto,val_uni_produto)
VALUES
(108,'Kg','Parafuso', 1.25)
```

```
INSERT INTO PRODUTO(id_produto,unidade_produto,desc_produto,val_uni_produto)
VALUES
(108,'Kg','Parafuso', 1.25)
```

121 %
Messages

(1 row affected)

Completion time: 2021-12-10T16:16:39.4876525-03:00

44) Problema: - Alterar o valor unitário do parafuso de R\$1,25 para R\$1,62

RESP:

```
UPDATE PRODUTO
SET val_uni_produto='1.62'
WHERE desc_produto = 'Parafuso';
SELECT * FROM PRODUTO WHERE desc_produto = 'Parafuso'
```

```
UPDATE PRODUTO
SET val_uni_produto='1.62'
WHERE desc_produto = 'Parafuso';
SELECT * FROM PRODUTO WHERE desc_produto = 'Parafuso'
```

121 %

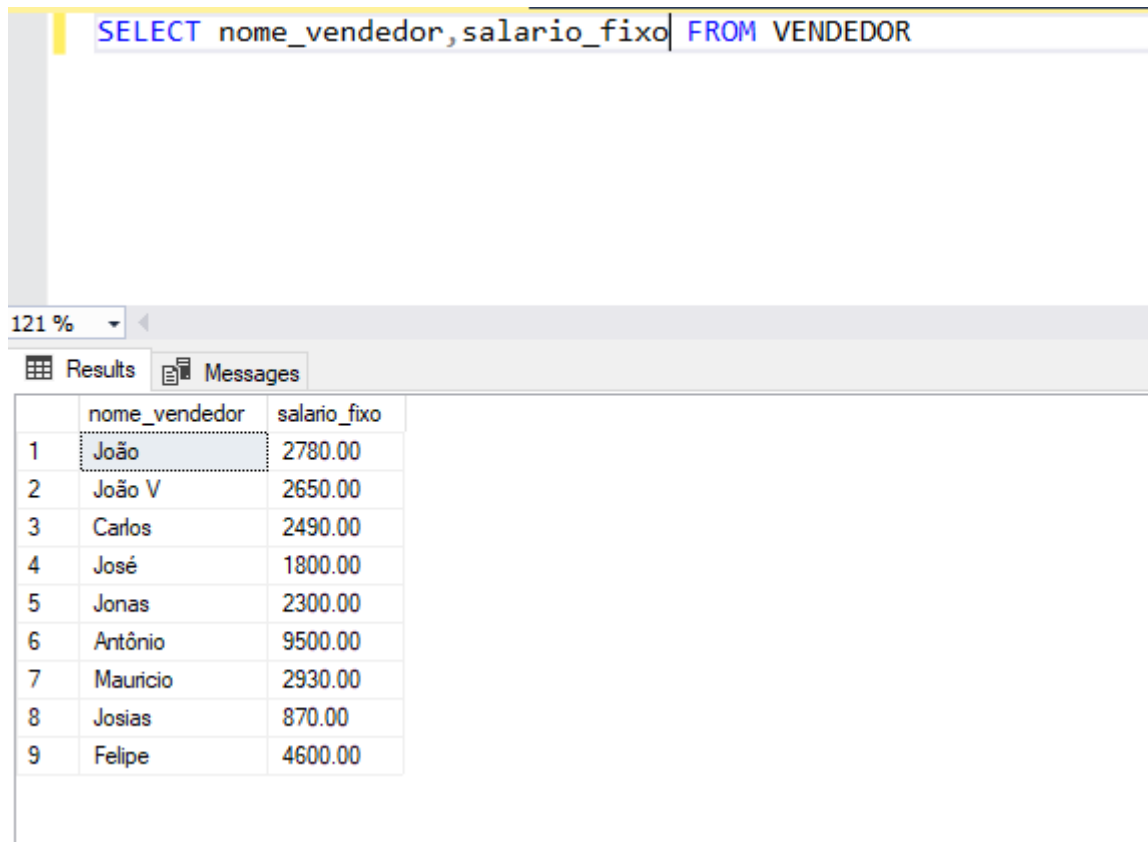
Results Messages

	id_produto	unidade_produto	desc_produto	val_uni_produto
1	108	Kg	Parafuso	1.62

45) Problema: - Exibir o nome e Salário fixo de todos os vendedores.

RESP:

```
SELECT nome_vendedor,salario_fixo FROM VENDEDOR
```



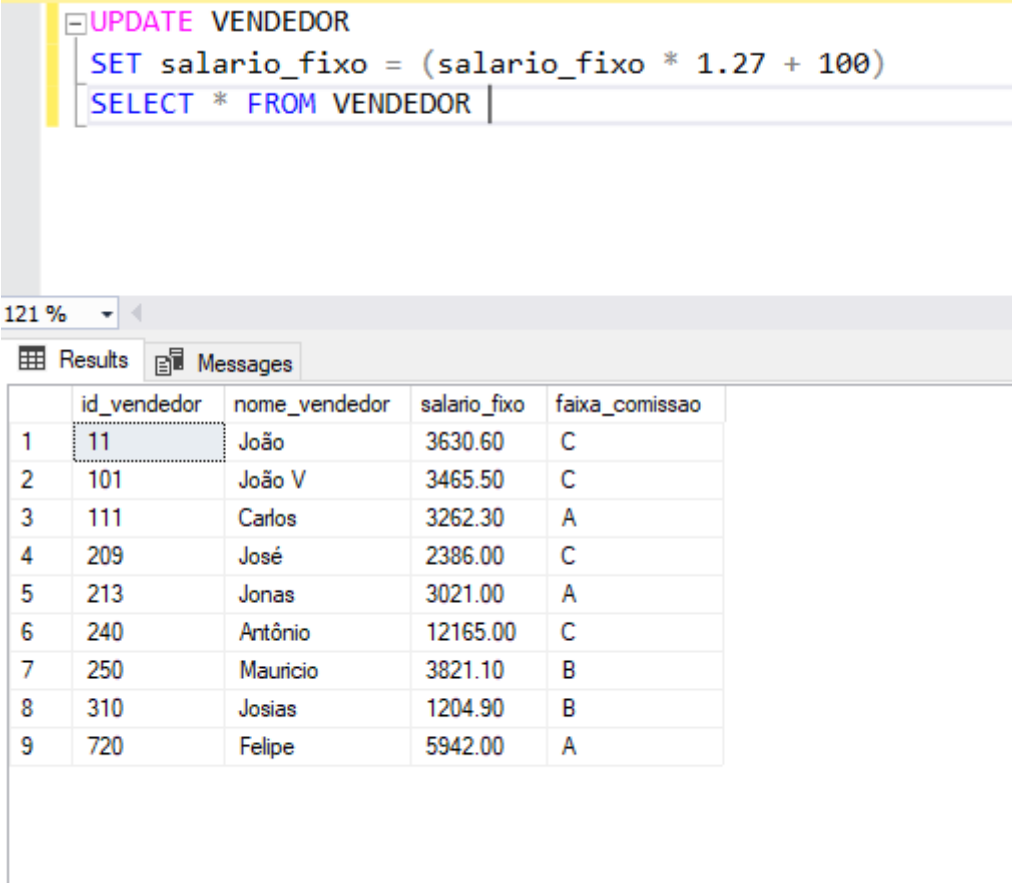
The screenshot shows a SQL query execution interface. At the top, the query `SELECT nome_vendedor,salario_fixo FROM VENDEDOR` is entered in a text box. Below the text box, there is a zoom level indicator set to 121%. Underneath, there are two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with the query results. The table has two columns: 'nome_vendedor' and 'salario_fixo'. It contains 9 rows of data, with the first row highlighted.

	nome_vendedor	salario_fixo
1	João	2780.00
2	João V	2650.00
3	Carlos	2490.00
4	José	1800.00
5	Jonas	2300.00
6	Antônio	9500.00
7	Mauricio	2930.00
8	Josias	870.00
9	Felipe	4600.00

46) Problema: - Atualizar o salário fixo de todos os vendedores em 27% mais uma bonificação de R\$100,00.

RESP:

```
UPDATE VENDEDOR
SET salario_fixo = (salario_fixo * 1.27 + 100)
SELECT * FROM VENDEDOR
```



The screenshot shows a database management interface. At the top, there is a text area containing the SQL query: `UPDATE VENDEDOR SET salario_fixo = (salario_fixo * 1.27 + 100) SELECT * FROM VENDEDOR`. Below the text area, there is a tabbed interface with 'Results' and 'Messages' tabs. The 'Results' tab is active, displaying a table with 5 columns: `id_vendedor`, `nome_vendedor`, `salario_fixo`, and `faixa_comissao`. The table contains 9 rows of data. The first row is highlighted with a dotted border.

	id_vendedor	nome_vendedor	salario_fixo	faixa_comissao
1	11	João	3630.60	C
2	101	João V	3465.50	C
3	111	Carlos	3262.30	A
4	209	José	2386.00	C
5	213	Jonas	3021.00	A
6	240	Antônio	12165.00	C
7	250	Mauricio	3821.10	B
8	310	Josias	1204.90	B
9	720	Felipe	5942.00	A

47) Problema: - Acrescentar 2,5% ao preço unitário dos produtos que estejam abaixo da média dos preços para aqueles comprados a QUILO.

RESP:

```
UPDATE PRODUTO
SET val_uni_produto = (val_uni_produto * 1.025)
WHERE val_uni_produto < (SELECT AVG(val_uni_produto) FROM PRODUTO) AND
unidade_produto = 'Kg'

SELECT * FROM PRODUTO
```

```
UPDATE PRODUTO
SET val_uni_produto = (val_uni_produto * 1.025)
WHERE val_uni_produto < (SELECT AVG(val_uni_produto) FROM PRODUTO) AND unidade_produto = 'Kg'

SELECT * FROM PRODUTO
```

121 %

Results Messages

	id_produto	unidade_produto	desc_produto	val_uni_produto
1	13	G	Ouro	6.18
2	22	M	Linho	0.11
3	25	Kg	Queijo	0.99
4	30	SAC	Açúcar	0.30
5	31	BAR	Chocolate	0.87
6	45	M	Madeira	0.25
7	53	M	Linha	1.80
8	77	M	Papel	1.05
9	78	L	Vinho	2.00
10	87	M	Cano	1.97
11	108	Kg	Parafuso	1.62