

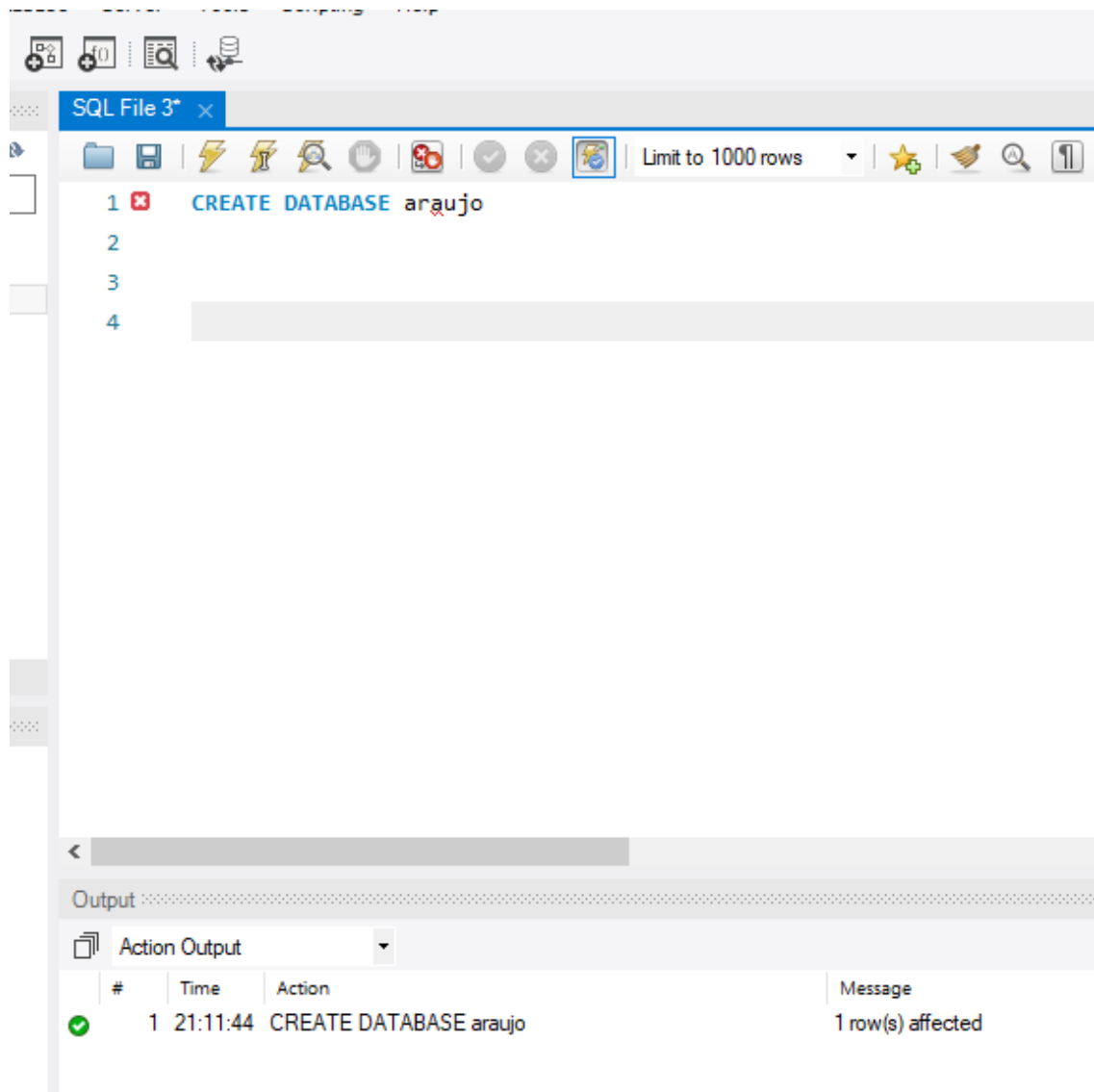
SQL para Análise de Dados

Os gerenciadores de banco de dados é uma das principais ferramentas utilizada em análise de dados, as funcionalidades a seguir descreve algumas funções de SQL que é utilizada em processos de análise de dados. O SGBD utilizado nas funções a seguir é o Mysql.

Para criar um banco de dados utilizamos a instrução:

```
CREATE DATABASE <nome>
```

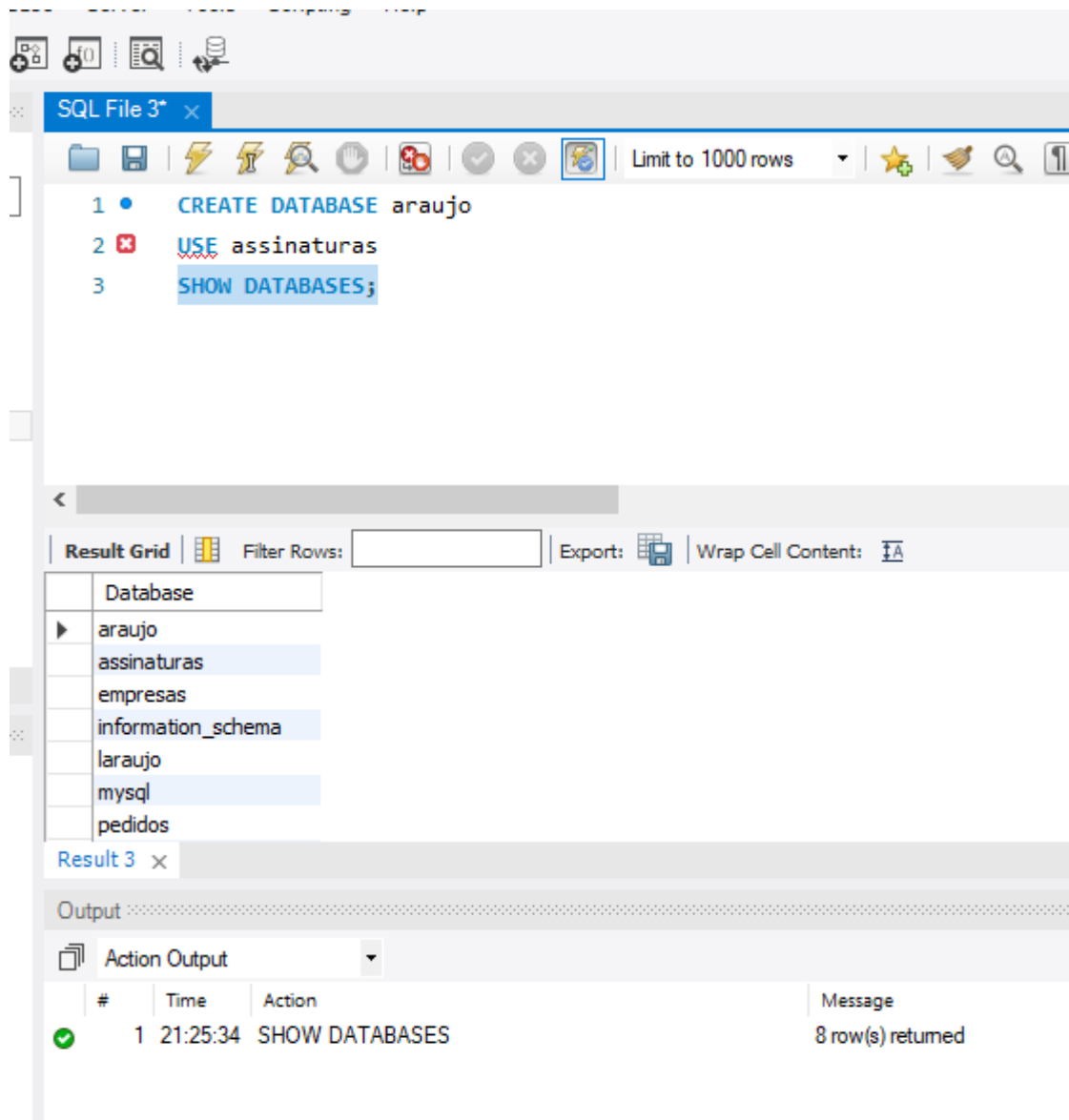
Nesse banco de dados criado podemos criar tabelas e inserir dados



Para verificar se o banco de dados foi criado utilizamos o comando

SHOW DATABASES;

Todos os bancos que possuímos no nosso SGBD serão exibidos;



The screenshot shows a MySQL IDE interface. The top panel, titled "SQL File 3*", contains three lines of SQL code:

```
1 CREATE DATABASE araujo
2 USE assinaturas
3 SHOW DATABASES;
```

The bottom panel displays the "Result Grid" for the third command. It shows a list of databases:

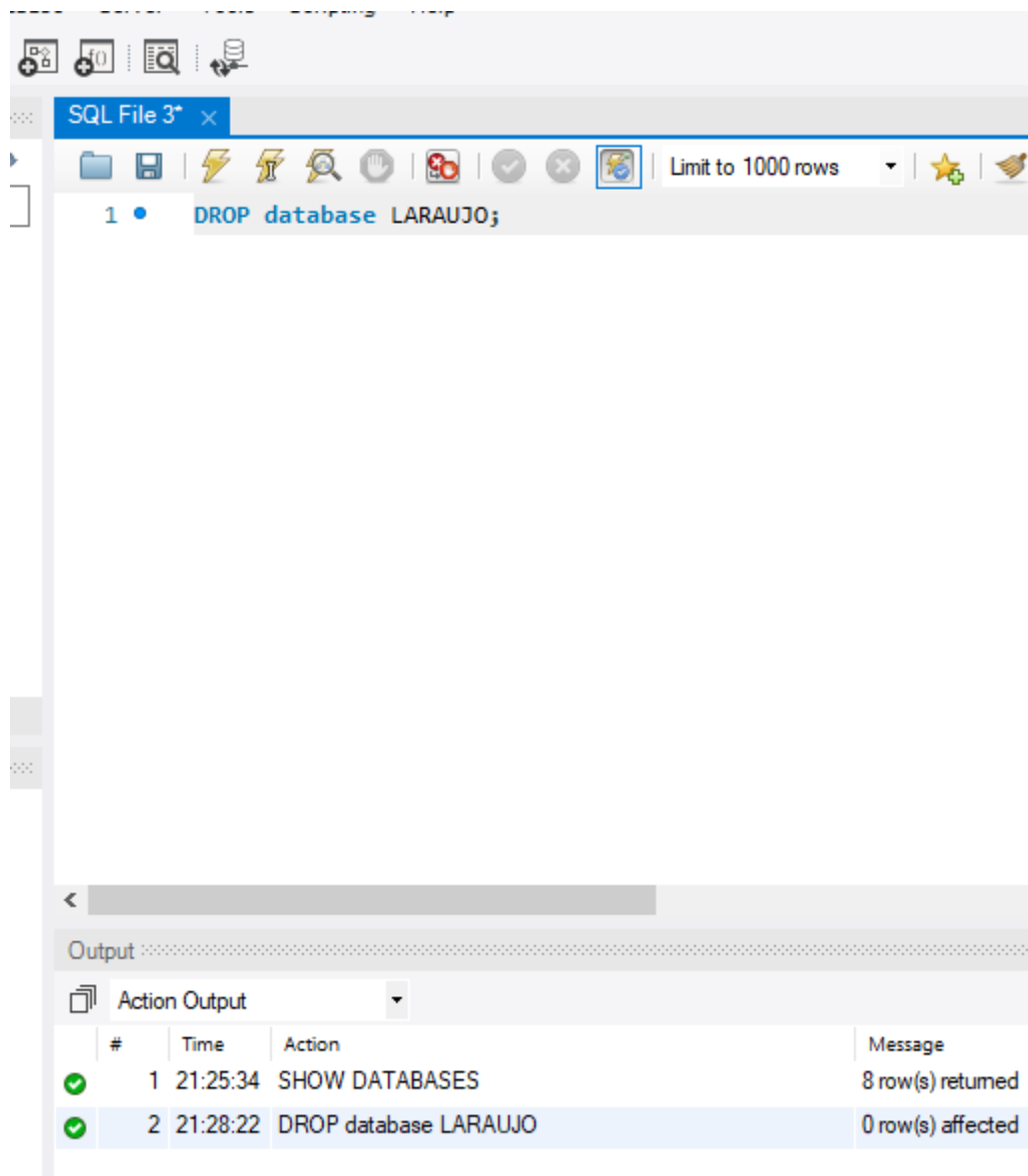
Database
araujo
assinaturas
empresas
information_schema
laraujo
mysql
pedidos

Below the result grid, the "Output" section shows the "Action Output" for the command:

#	Time	Action	Message
1	21:25:34	SHOW DATABASES	8 row(s) returned

Para deletar um banco de dados podemos utilizar o seguinte comando:

`DROP DATABASE <nome>;`



Criando minha primeira tabela, o comando é:

```
CREATE TABLE <nome> ( <coluna> <tipo de dado>);
```

É possível inserir diversas colunas e com diferentes tipos de dados;

O mais utilizado é o VARCHAR, que representa os textos/strings de uma tabela.

The screenshot shows a SQL IDE window titled "SQL File 3* x". The main editor contains the following SQL code:

```
5 • create table if not exists contatos (  
6     ID BIT (4) primary KEY,  
7     nome varchar(50) not null,  
8     email varchar(50) not null,  
9     telefone decimal not null  
10  );  
11 • show tables;
```

Below the editor, there is a "Result Grid" section with a "Filter Rows" input and an "Export" button. It lists the tables in the database:

Tables_in_empresas
clientes
contatos
dadosrh_modificado

At the bottom, the "Output" section shows the "Action Output" table:

#	Time	Action	Message
✓ 1	22:59:10	create table if not exists contatos (ID BIT (4) primary K...	0 row(s) affected
✓ 2	23:00:07	show tables	3 row(s) returned

Inserindo dados na tabela contatos.

SQL File 3*

```
1 • use EMPRESAS;
2 • INSERT INTO contatos (ID, nome, email, telefone) VALUES(1, 'Luana', 'lsa@hotmail.com', '1110503050');
3 • INSERT INTO contatos (ID, nome, email, telefone) VALUES(2, 'Pedro', 'contato@contatos.com', '7197451102');
4 • INSERT INTO contatos (ID, nome, email, telefone) VALUES(3, 'Lucas', 'pdr@contatos.com', '7594050203');
5 • select * from contatos;
6
7
```

Result Grid

ID	nome	email	telefone
1	Luana	lsa@hotmail.com	1110503050
2	Pedro	contato@contatos.com	7197451102
3	Lucas	pdr@contatos.com	7594050203
•	NULL	NULL	NULL

contatos 2

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✗ 1	21:30:11	INSERT INTO contatos (ID, nome, email, telefone) VALU...	Error Code: 1264. Out of range value for column 'telefone' ...	0.000 sec
✓ 2	21:31:03	INSERT INTO contatos (ID, nome, email, telefone) VALU...	1 row(s) affected	0.093 sec
✓ 3	21:31:03	INSERT INTO contatos (ID, nome, email, telefone) VALU...	1 row(s) affected	0.156 sec
✓ 4	21:31:07	select * from contatos LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec

Adicionando uma coluna na tabela.

SQL File 3* x

Limit to 1000 rows

```

1 • ALTER TABLE contatos ADD COLUMN profissão varchar(50);
2 • select * from contatos;
3

```

<

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:

ID	nome	email	telefone	idade	profissão
1	Luana	lsa@hotmail.com	1110503050	NULL	NULL
2	Pedro	contato@contatos.com	7197451102	NULL	NULL
3	Lucas	pdr@contatos.com	7594050203	NULL	NULL

contatos 3 x

Output

Action Output

#	Time	Action	Message
1	23:00:19	ALTER TABLE contatos ADD COLUMN profissão varchar...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
2	23:00:59	select * from contatos LIMIT 0, 1000	3 row(s) returned

Removendo uma coluna

The screenshot shows a SQL IDE interface with a query editor and a result grid. The query editor contains two SQL statements: `ALTER TABLE contatos DROP COLUMN idade;` and `select *from contatos;`. The result grid displays the data from the `contatos` table, which has columns `ID`, `nome`, `email`, `telefone`, and `profissão`. The data is as follows:

ID	nome	email	telefone	profissão
1	Luana	lsa@hotmail.com	1110503050	NULL
2	Pedro	contato@contatos.com	7197451102	NULL
3	Lucas	pdr@contatos.com	7594050203	NULL
*	NULL	NULL	NULL	NULL

Below the result grid, the `Output` pane shows the execution log for the `contatos` table. The log contains three entries:

#	Time	Action	Message
✓ 1	23:03:45	select * from contatos LIMIT 0, 1000	3 row(s) returned
✓ 2	23:04:33	ALTER TABLE contatos DROP COLUMN idade	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 3	23:04:51	select *from contatos LIMIT 0, 1000	3 row(s) returned

Modificando o tipo de dado de uma tabela, no caso abaixo foi alterado o tipo de dados da coluna "ID".

The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and search. The SQL editor contains two lines of code:

```
1 • ALTER TABLE contatos MODIFY COLUMN ID varchar (10);
2 • select *from contatos;
```

Below the editor is the 'Result Grid' section, which displays the data from the 'contatos' table. The table has five columns: ID, nome, email, telefone, and profissão. The data is as follows:

ID	nome	email	telefone	profissão
1	Luana	lsa@hotmail.com	1110503050	NULL
2	Pedro	contato@contatos.com	7197451102	NULL
3	Lucas	pdr@contatos.com	7594050203	NULL
NULL	NULL	NULL	NULL	NULL

At the bottom, the 'Output' section shows the 'Action Output' for the executed queries:

#	Time	Action	Message
1	23:13:01	ALTER TABLE contatos MODIFY COLUMN ID varchar (10)	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0
2	23:13:42	select *from contatos LIMIT 0, 1000	3 row(s) returned

Seleccionando columnas específicas.

The screenshot shows the SQL File 3* editor with the following SQL query:

```
1 • SELECT nome, email FROM contatos;
```

Below the query, the results are displayed in a table:

	nome	email
▶	Luana	lsa@hotmail.com
	Pedro	contato@contatos.com
	Lucas	pdr@contatos.com

At the bottom, the Output panel shows the Action Output:

#	Time	Action	Message
1	23:22:09	SELECT nome, email FROM contatos LIMIT 0, 1000	3 row(s) returned

Para atualizar dados vamos utilizar a instrução UPDATE:

The screenshot shows the SQL Server Enterprise Manager interface. The top menu bar includes 'File', 'Server', 'Tools', 'Scripting', and 'Help'. Below the menu is a toolbar with various icons. The main window displays a script titled 'SQL File 3*' with the following SQL code:

```
1 • USE empresas;  
2 • SET SQL_SAFE_UPDATES=0;  
3 • update contatos  
4   SET profissão = 'Engenheira'  
5   WHERE ID = '1';  
6  
7
```

Below the script editor is the 'Output' pane, which shows the 'Action Output' for the executed statement:

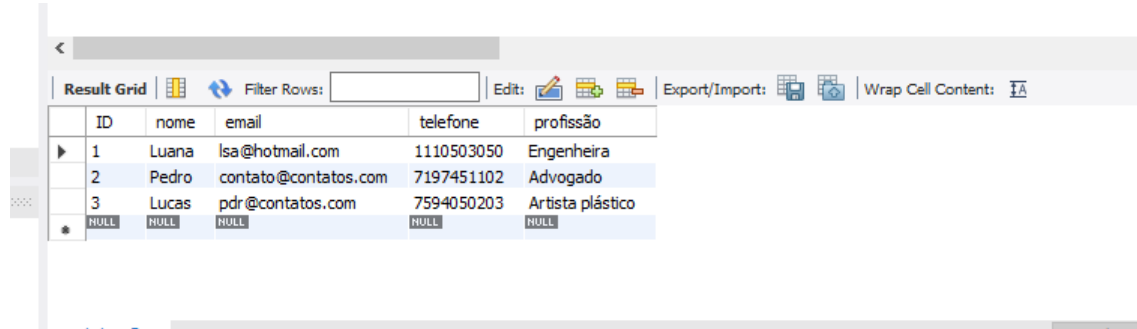
#	Time	Action	Message	Duration / Fet
1	22:13:50	update contatos SET profissão = 'Engenheira' WHERE ID ...	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.047 sec

Pronto arquivo, alterado.

The screenshot shows the 'Result Grid' in SQL Server Enterprise Manager. The grid displays the following data:

ID	nome	email	telefone	profissão
1	Luana	lsa@hotmail.com	1110503050	Engenheira
2	Pedro	contato@contatos.com	7197451102	Advogado
3	Lucas	pdr@contatos.com	7594050203	Artista plástico
NULL	NULL	NULL	NULL	NULL

Deletando uma linha. Nesse caso deletamos a linha com os dados do Lucas.



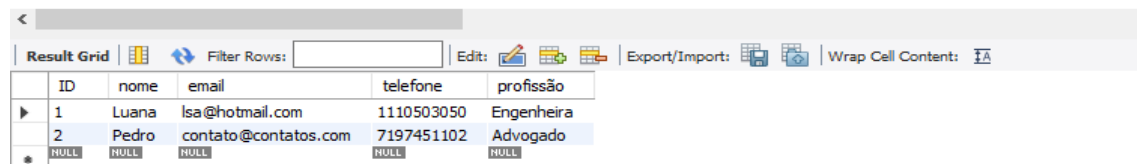
ID	nome	email	telefone	profissão
1	Luana	lsa@hotmail.com	1110503050	Engenheira
2	Pedro	contato@contatos.com	7197451102	Advogado
3	Lucas	pdr@contatos.com	7594050203	Artista plástico
NULL	NULL	NULL	NULL	NULL

Para isso utilizaremos o comando;

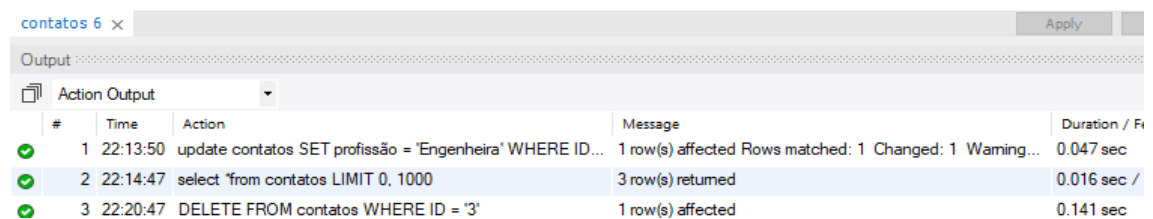
DELETE FROM

WHERE;

```
2 • DELETE FROM contatos WHERE ID = '3';  
3
```

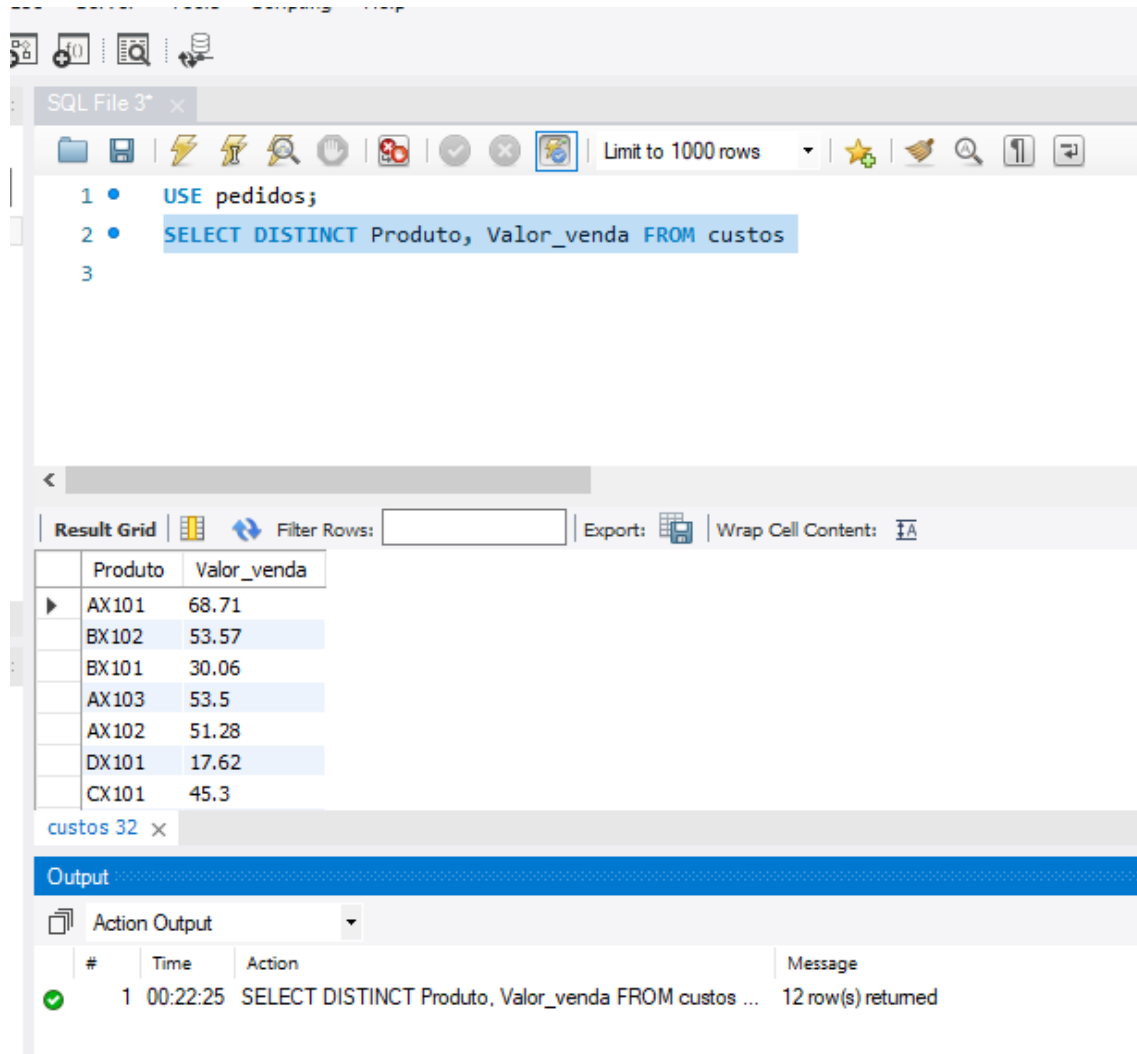


ID	nome	email	telefone	profissão
1	Luana	lsa@hotmail.com	1110503050	Engenheira
2	Pedro	contato@contatos.com	7197451102	Advogado
NULL	NULL	NULL	NULL	NULL



#	Time	Action	Message	Duration / F
✓ 1	22:13:50	update contatos SET profissão = 'Engenheira' WHERE ID...	1 row(s) affected Rows matched: 1 Changed: 1 Warning...	0.047 sec
✓ 2	22:14:47	select *from contatos LIMIT 0, 1000	3 row(s) returned	0.016 sec /
✓ 3	22:20:47	DELETE FROM contatos WHERE ID = '3'	1 row(s) affected	0.141 sec

Selecionando e classificando um produto, pelos valores unicos registrado no banco de dados. Com o comando Distinct os resultados não serão apresentados em duplicidade.



The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The SQL editor contains the following code:

```
1 • USE pedidos;  
2 • SELECT DISTINCT Produto, Valor_venda FROM custos  
3
```

Below the editor is the 'Result Grid' section, which displays the query results in a table:

	Produto	Valor_venda
▶	AX101	68.71
	BX102	53.57
	BX101	30.06
	AX103	53.5
	AX102	51.28
	DX101	17.62
	CX101	45.3

Below the result grid is the 'Output' section, which shows the execution log:

#	Time	Action	Message
✓ 1	00:22:25	SELECT DISTINCT Produto, Valor_venda FROM custos ...	12 row(s) returned

Selecionando produtos em uma determinada faixa de valor de venda, nesse caso vendas totalizadas entre R\$ 10,00 e R\$30,00.

SQL File 3*

```

1 • USE pedidos;
2 • select *from custos;
3 • SELECT * FROM custos WHERE valor_venda BETWEEN 10 AND 30;

```

Result Grid

Data	Produto	Código	Valor_Venda	Prego_Custo	Duração_Venda_Telefone_(mins)	Duração_Espera_(mins)
6/3/2021	DX101	GF54319	17.62	9.16	28	22
27/02/2021	CX102	GF54233	24.79	11.4	13	22
18/03/2021	DX101	GF54489	17.62	8.11	26	22
20/03/2021	DX101	GF54405	17.62	8.11	23	19
3/3/2021	CX102	GF54664	24.79	12.89	37	18
31/03/2021	DX101	GF54430	17.62	8.28	31	19
24/03/2021	DX101	GF54576	17.62	8.99	11	22

custos 13 x

Action Output

#	Time	Action	Message
✓ 1	23:09:13	SELECT * FROM custos WHERE valor_venda BETWEE...	17 row(s) returned

Ordenando as datas por ordem crescente.

The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and search. The SQL editor contains the following query:

```
1 • USE pedidos;  
2 • select *from custos  
3   order by Data asc
```

Below the editor is the 'Result Grid' section, which displays the query results in a table. The table has 8 columns: Data, Produto, Código, Valor_Venda, Preço_Custo, Duração_Venda_Telefone_(mins), and Duração_Espera_(mins). The results are sorted by the 'Data' column in ascending order.

	Data	Produto	Código	Valor_Venda	Preço_Custo	Duração_Venda_Telefone_(mins)	Duração_Espera_(mins)
▶	1/3/2021	BX103	GF54414	68.93	33.78	0	17
	1/3/2021	BX102	GF54373	53.57	26.25	0	14
	1/3/2021	CX103	GF54371	73.3	40.32	9	15
	10/3/2021	BX103	GF54456	68.93	37.22	0	18
	11/3/2021	BX101	GF54434	30.06	14.73	34	13
	11/3/2021	DX102	GF54374	62.46	31.86	3	21
	13/03/2021	BX103	GF54674	68.93	31.71	32	16

Below the table is the 'Output' section, which shows the execution log. The log indicates that the query was executed successfully and returned 106 rows.

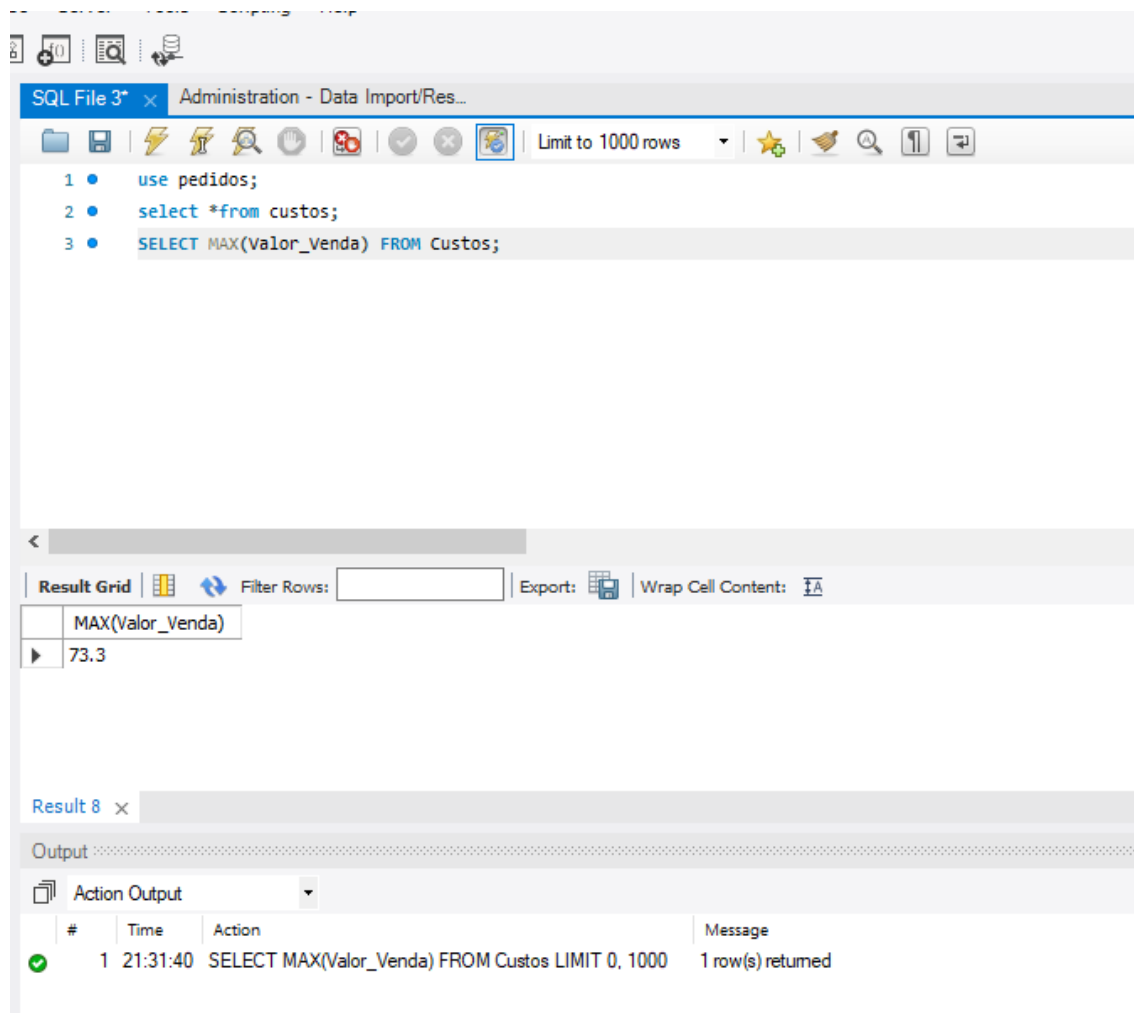
#	Time	Action	Message
✓ 1	21:12:58	select *from custos order by Data asc LIMIT 0, 1000	106 row(s) returned

Verificando o balanceamento das vendas superior a R\$ 60,00 programando para retornar apenas os 10 primeiros.

```
2 • select *from custos;  
3 • SELECT * FROM custos WHERE Valor_Venda > 60 LIMIT 10;
```

Result Grid							
		Filter Rows:		Export:	Wrap Cell Content:		Fetch rows:
	Data	Produto	Código	Valor_Venda	Preço_Custo	Duração_Venda_Telefone_(mins)	Duração_Espera_(mins)
▶	19/03/2021	AX101	GF54309	68.71	34.36	17	11
	2/3/2021	AX101	GF54381	68.71	37.79	27	20
	3/3/2021	AX101	GF54695	68.71	35.73	0	20
	27/03/2021	AX101	GF54354	68.71	31.61	22	22
	13/03/2021	DX102	GF54634	62.46	30.61	27	18
custos 2 x							
Output							
Action Output							
	#	Time	Action			Message	
✓	1	21:14:24	select *from custos LIMIT 0, 1000			106 row(s) returned	
✓	2	21:16:26	SELECT * FROM custos WHERE Valor_Venda > 60 LIM...			10 row(s) returned	

Verificando o maior valor de compra feita por um cliente.



Verificando o menor valor de compra feita por um cliente.

The screenshot displays the SQL Server Enterprise Manager interface. The top pane shows a query window with the following SQL code:

```
1 use pedidos;  
2 select *from custos;  
3 SELECT MIN(Valor_Venda) FROM Custos;
```

The bottom pane shows the execution results in a grid format:

MIN(Valor_Venda)
17.62

Below the grid, the 'Output' section shows the 'Action Output' for the query:

#	Time	Action	Message
✓ 1	22:04:01	SELECT MIN(Valor_Venda) FROM Custos LIMIT 0, 1000	1 row(s) returned

Verificando o valor médio de vendas

The screenshot shows the SQL Server Enterprise Manager interface. At the top, there's a tab labeled "SQL File 3*" and another labeled "Administration - Users and Privil...". Below the tabs is a toolbar with various icons. The main area displays a SQL query:

```
1 • use pedidos;
2 • select *from custos;
3 • SELECT AVG(Valor_Venda) FROM Custos;
4 |
```

Below the query editor, there's a section for the results. It includes a "Result Grid" button, a "Filter Rows:" input field, and an "Export:" button. The result grid shows one row of data:

	AVG(Valor_Venda)
▶	50.492452830188675

At the bottom, there's a "Result 21" tab. Below it, the "Output" pane shows the execution log:

#	Time	Action	Message
✓ 2	23:03:39	select *from custos LIMIT 0, 1000	106 row(s) returned
✓ 3	23:03:39	select *from custos LIMIT 0, 1000	106 row(s) returned
✓ 4	23:03:49	SELECT AVG(Valor_Venda) FROM Custos LIMIT 0, 1000	1 row(s) returned

Somando todas as vendas

SQL File 3* x Administration - Users and Privil...

Limit to 1000 rows

```
1 • use pedidos;
2 • select *from custos;
3 • SELECT SUM(Valor_Venda) FROM custos;
4
5
```

Result Grid

	SUM(Valor_Venda)
▶	5352.2

Filter Rows: Export: Wrap Cell Content:

Result 22 x

Output

Action Output

#	Time	Action	Message
✓ 1	23:06:12	SELECT SUM(Valor_Venda) FROM custos LIMIT 0, 1000	1 row(s) returned

Fazendo uma busca por um determinado resultado. No caso abaixo foi selecionado que apareça apenas resultados a qual a idade seja 21, 32, 40.

The screenshot shows a SQL client interface with a query editor and a results grid. The query editor contains the following SQL code:

```
1 • use empresas;  
2 • show tables;  
3 • select *from clientes;  
4 • SELECT * FROM clientes WHERE Idade IN ('21', '32', '40');  
5 •
```

The results grid displays the following data:

ID Cliente	Nome	Sobrenome	Sexo	Idade	Regiao	Classificacao	Data Cadastro	Saldo
100000001	Simon	Walsh	Masculino	21	SÃ£o Paulo	Alto	05.Jan.21	113810.21
300000004	Trevor	Parr	Masculino	32	Belo Horizonte	Alto	08.Jan.21	1421.52
100000011	Dominic	Lewis	Masculino	40	SÃ£o Paulo	Alto	12.Jan.21	39667.83
100000019	William	Ince	Masculino	40	SÃ£o Paulo	MÃ©dio	15.Jan.21	65534.69

The interface also shows an "Output" section with "Action Output" and a table of execution logs:

#	Time	Action	Message
1	23:05:42	SELECT * FROM clientes WHERE Idade IN ('21', '32', '40'...	320 row(s) returned

Verificando o número de vezes que um cliente efetuou compras

The screenshot shows the SQL Server Enterprise Manager interface. At the top, there's a toolbar with various icons. Below it, a tab labeled "SQL File 3*" is active. The main area contains a SQL query:

```
1 • use empresas;
2 • select *from clientes;
3 • SELECT Nome, COUNT(Saldo) as 'vendas por pessoa'FROM clientes GROUP BY Nome;
```

Below the query editor, the "Result Grid" section displays the output of the query. It has columns for "Nome" and "vendas por pessoa". The data rows are:

	Nome	vendas por pessoa
▶	Simon	29
	Jasmine	30
	Liam	28
	Trevor	28
	Deirdre	32

At the bottom, the "Output" pane shows the message: "SELECT Nome, COUNT(Saldo) as 'vendas por pessoa'FR... 172 row(s) returned".

Retornando a região com vendas inferiores a R\$ 1.000,00.

The screenshot shows the SQL Developer interface with a query window titled "SQL File 3*" and a results window titled "Result 44".

Query:

```
1 • USE EMPRESAS;  
2 • SELECT *FROM CLIENTES;  
3 • select Regiao, SUM(50) as 'Regiões com menor venda'  
4   from clientes  
5   group by Nome  
6   having SUM(50) < 1000;
```

Result Grid:

Regiao	Regiões com menor venda
Belo Horizonte	700
Rio de Janeiro	850
São Paulo	900
São Paulo	900
São Paulo	850

Output:

#	Time	Action	Message
1	20:23:17	SELECT *FROM CLIENTES LIMIT 0, 1000	1000 row(s) returned
2	20:23:17	select Regiao, SUM(50) as 'Regiões com menor venda' fr...	47 row(s) returned

Verificando através da função CHAR_LENGTH a quantidade de caracteres uma determinada palavra tem.

The screenshot shows a SQL IDE interface. At the top, a tab labeled "SQL File 3*" is active. Below the tab is a toolbar with various icons, including a "Limit to 1000 rows" dropdown. The main editor area contains three lines of SQL code:

```
1 • USE EMPRESAS;  
2 • SELECT *FROM CLIENTES;  
3 • SELECT CHAR_LENGTH('Mackenzie');
```

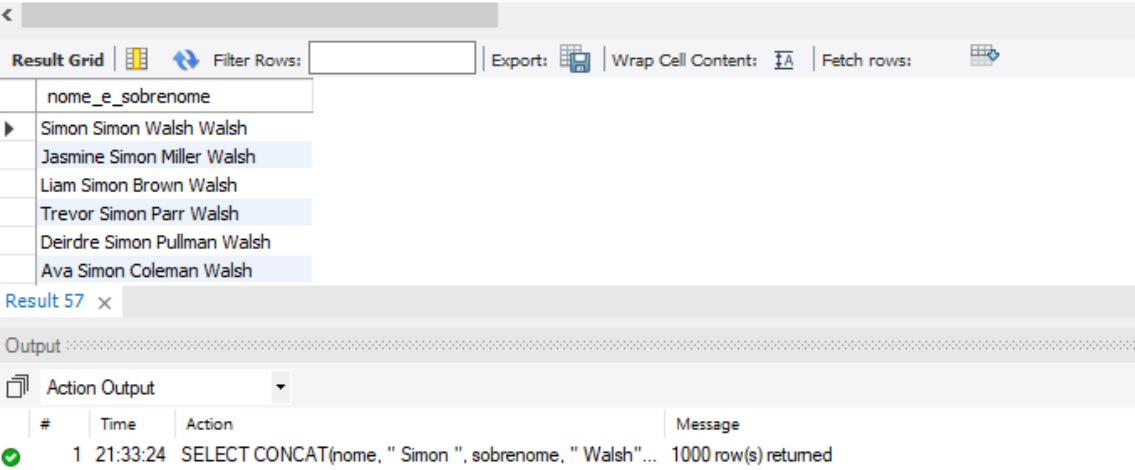
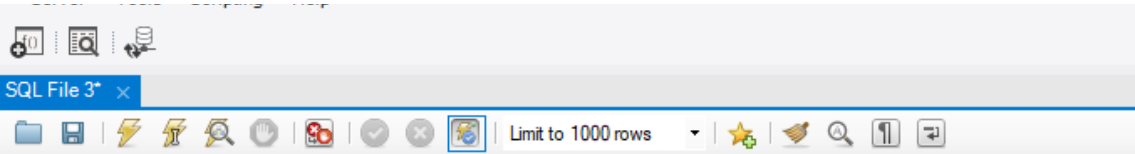
Below the editor is a "Result Grid" section. It has a "Filter Rows:" input field, an "Export:" button, and a "Wrap Cell Content:" checkbox. The grid displays the following data:

	CHAR_LENGTH('Mackenzie')
▶	9

At the bottom, there is a "Result 53" tab and an "Output" section. The "Output" section has a dropdown menu set to "Action Output". Below this is a table with the following data:

#	Time	Action	Message
✓ 1	21:27:44	SELECT CHAR_LENGTH('Mackenzie') LIMIT 0, 1000	1 row(s) returned

Efetuating the concatenation of words.



Concatenando palavras separando-as por virgula.

SQL File 3

Limit to 1000 rows

```

1 • USE EMPRESAS;
2 • SELECT *FROM CLIENTES;
3 • SELECT CONCAT_WS(" ", Nome, Idade) AS cliente
4     FROM clientes;
5

```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

Fetch rows:

	cliente
▶	Simon, 21
	Jasmine, 34
	Liam, 46
	Trevor, 32
	Deirdre, 38
	Ava, 30

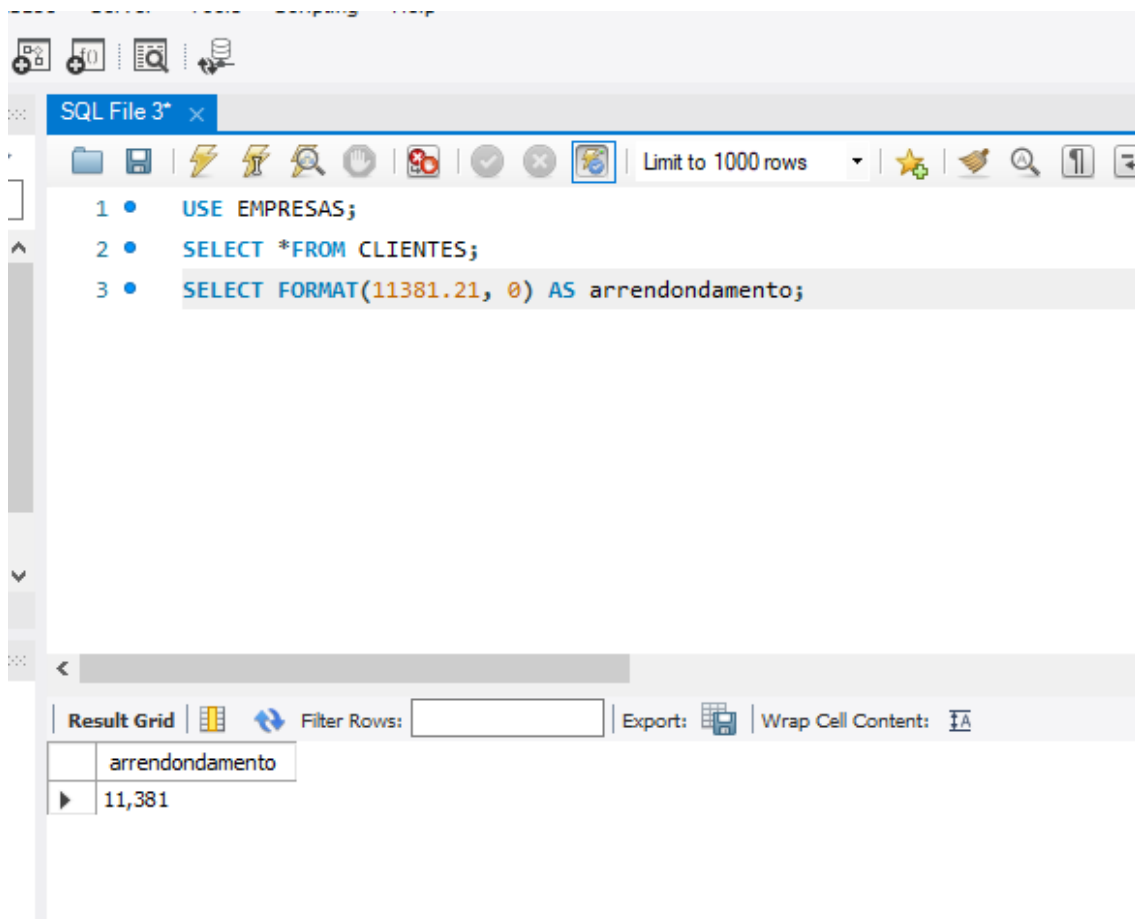
Result 66

Output

Action Output

#	Time	Action	Message
✓ 1	00:34:32	SELECT CONCAT_WS(" ", Nome, Idade) AS cliente FR...	1000 row(s) returned

Arredondando o valor de um Resultado do saldo.



The screenshot shows a SQL IDE interface. The top toolbar includes icons for running queries, saving, and other standard functions. The main editor area contains the following SQL code:

```
1 • USE EMPRESAS;  
2 • SELECT *FROM CLIENTES;  
3 • SELECT FORMAT(11381.21, 0) AS arredondamento;
```

Below the editor, the 'Result Grid' is displayed. It has a 'Filter Rows' input field and buttons for 'Export' and 'Wrap Cell Content'. The grid shows the result of the query:

arredondamento
11,381

Verificando em qual posição um determinado caractere se encontra. No caso abaixo será o "Ã".

The screenshot shows the SQL Developer interface. The top pane displays a SQL script with three lines:

```
1 • USE EMPRESAS;
2 • SELECT *FROM CLIENTES;
3 • SELECT Regiao, INSTR(regiao, 'ã') FROM clientes;
```

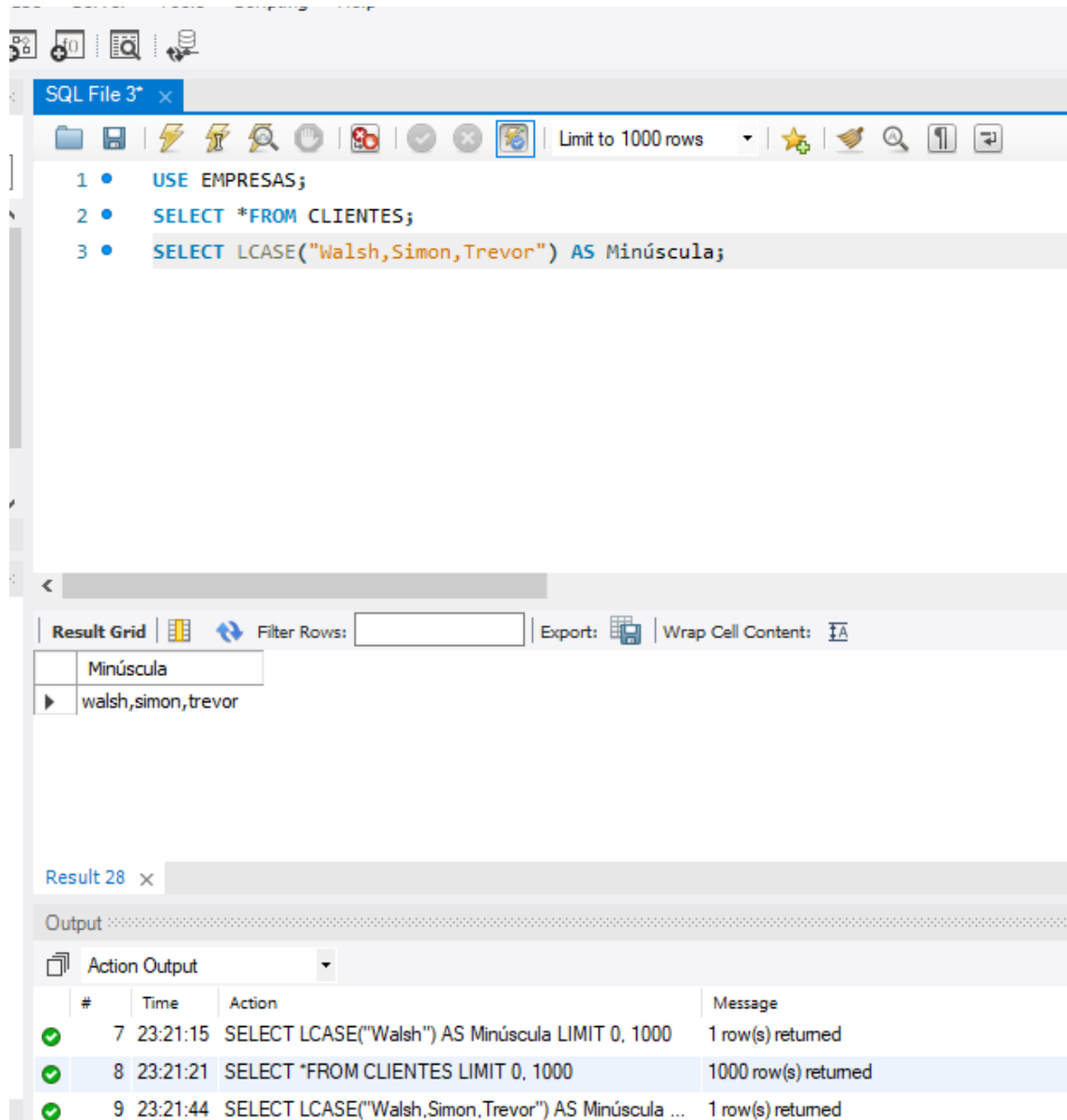
The third line is highlighted in blue. The bottom pane shows the "Result Grid" with the following data:

Regiao	INSTR(regiao, 'ã')
SÃ£o Paulo	2
VitÃ³ria	4
SÃ£o Paulo	2
Belo Horizonte	0
SÃ£o Paulo	2

Below the result grid, the "Output" pane shows the "Action Output" table:

#	Time	Action	Message
5	22:56:29	INSTR(Regiao, 'ã')	Error Code: 1064. You have an error in your SQL syntax;...
6	22:58:07	SELECT *FROM CLIENTES LIMIT 0, 1000	1000 row(s) returned
7	22:58:39	SELECT Regiao, INSTR(regiao, 'ã') FROM clientes LIM...	1000 row(s) returned

Transformando os caractere de determinadas palavras em caractere Minúsculos.



Extraindo uma determinada quantidade de caractere, da esquerda para direita.

The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The SQL editor contains the following code:

```
1 • USE EMPRESAS;  
2 • SELECT *FROM CLIENTES;  
3 • SELECT Nome, LEFT(Nome, 2) FROM clientes;  
4
```

Below the editor, the 'Result Grid' tab is active, displaying the results of the third query. The grid has two columns: 'Nome' and 'LEFT(Nome, 2)'. The data rows are:

	Nome	LEFT(Nome, 2)
▶	Simon	Si
	Jasmine	Ja
	Liam	Li
	Trevor	Tr
	Deirdre	De

At the bottom, the 'Output' pane shows the 'Action Output' for the query, indicating that 1000 row(s) were returned.

#	Time	Action	Message
✓ 1	23:42:16	SELECT Nome, LEFT(Nome, 2) FROM clientes LIMIT 0, 1...	1000 row(s) returned

Trocando alguma parte da escrita em uma string.

The screenshot shows a SQL editor window titled "SQL File 3*" with a toolbar at the top. The SQL query is as follows:

```
1 • USE EMPRESAS;  
2 • SELECT *FROM CLIENTES;  
3 • SELECT REPLACE(Regiao, "VitÃ³ria", "Vitória") FROM clientes;  
4
```

Below the editor, the "Result Grid" is displayed, showing the results of the query. The first row is the SQL statement itself, and the subsequent rows are the results of the REPLACE function applied to the 'Regiao' column of the 'CLIENTES' table.

	REPLACE(Regiao, "VitÃ³ria", "Vitória")
▶	SÃ£o Paulo
	Vitória
	SÃ£o Paulo
	Belo Horizonte
	SÃ£o Paulo

Below the result grid, the "Output" section is visible, showing the execution of the query. The output is as follows:

```
Result 6 x  
Output  
Action Output  
# Time Action Message  
✓ 1 22:49:47 SELECT REPLACE(Regiao, "VitÃ³ria", "Vitória") FROM cli... 1000 row(s) returned
```

Extraindo uma determinada quantidade de caractere, da direita para esquerda.

The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The SQL editor contains the following query:

```
1 • USE EMPRESAS;  
2 • SELECT *FROM CLIENTES;  
3 • SELECT RIGHT(Regiao, 03) AS Redução FROM clientes;
```

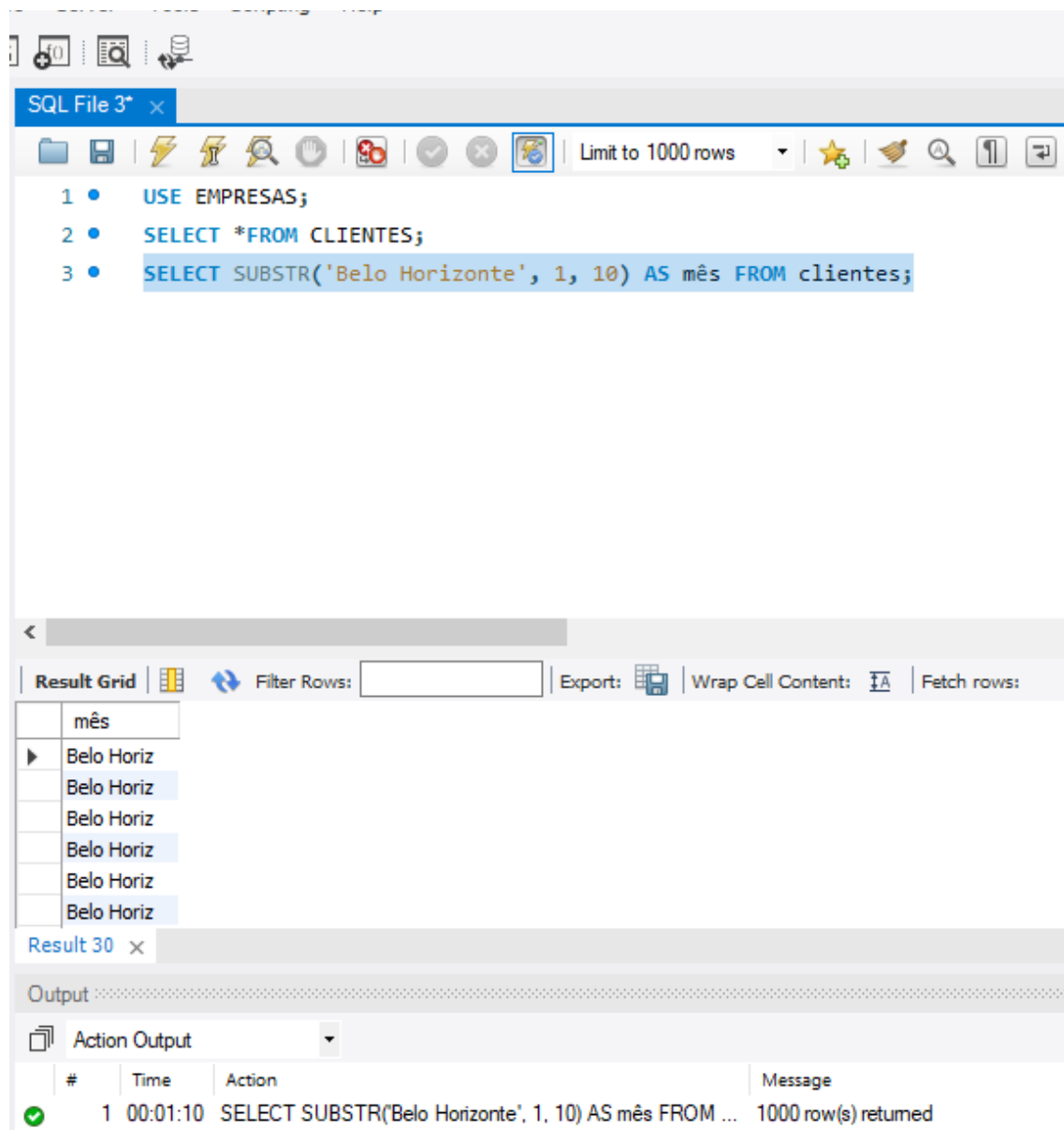
Below the editor is the 'Result Grid' section, which displays the results of the third query. The column header is 'Redução'. The visible data rows are:

Redução
ulo
ria
ulo
nte
ulo
nte

At the bottom, the 'Output' pane shows the 'Action Output' for the executed query:

#	Time	Action	Message
1	23:31:57	SELECT RIGHT(Regiao, 03) AS Redução FROM clientes...	1000 row(s) returned

Efetuada a abreviação de uma determinada palavra. A partir de um determinado ponto.



Transformando todas as letras das palavras de uma determinada coluna em maiúscula

The screenshot shows a SQL IDE interface. At the top, a toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The SQL editor contains the following code:

```
1 • USE EMPRESAS;  
2 • SELECT *FROM CLIENTES;  
3 • SELECT UCASE(Nome) FROM clientes;
```

Below the editor is the 'Result Grid' section, which displays the results of the query in a table:

UCASE(Nome)
SIMON
JASMINE
LIAM
TREVOR
DEIRDRE
AVA

At the bottom, the 'Output' pane shows the 'Action Output' for the query:

#	Time	Action	Message
✓ 1	00:05:53	SELECT UCASE(Nome) FROM clientes LIMIT 0, 1000	1000 row(s) returned

Efetuando o arredondamento com casas decimais para cima.

SQL File 3*

Limit to 1000 rows

1 • USE EMPRESAS;

2 • SELECT *FROM CLIENTES;

3 • SELECT Saldo, CEIL(Saldo) AS arredondado FROM clientes;

Result Grid

Filter Rows:

Export

Wrap Cell Content:

Fetch rows:

	Saldo	arredondado
▶	113810.21	113811
	36919.73	36920
	101536.83	101537
	1421.52	1422
	35639.79	35640
	122443.77	122444

Result 2

Output

Action Output

#	Time	Action	Message
✓ 1	23:35:54	SELECT Saldo, CEIL(Saldo) AS arredondado FROM client...	1000 row(s) returned

Efetuating o arredondamento com casas decimais para baixo.

The screenshot shows the SQL File 3+ editor with the following SQL query:

```
1 • USE EMPRESAS;
2 • SELECT *FROM CLIENTES;
3 • SELECT Saldo, CEIL(Saldo) AS para_cima, FLOOR(Saldo) AS para_baixo FROM clientes;
4 •
```











The results are displayed in the Result Grid below the query:

	Saldo	para_cima	para_baixo
▶	113810.21	113811	113810
	36919.73	36920	36919
	101536.83	101537	101536
	1421.52	1422	1421
	35639.79	35640	35639
	122443.77	122444	122443






The Output pane shows the message: "1 23:47:59 SELECT Saldo, CEIL(Saldo) AS para_cima, FLOOR(Saldo) AS para_baixo FROM clientes; 1000 row(s) returned".

Adicionando uma Quantidade de dias, a partir de uma data.

SQL File 3*



Limit to 1000 rows





1 • `USE EMPRESAS;`

2 • `SELECT *FROM CLIENTES;`


3 • `SELECT ADDDATE("2021-01-05", INTERVAL 10 DAY);`


<

Result Grid



Filter Rows:

Export: 

Wrap Cell Content: 

	ADDDATE("2021-01-05", INTERVAL 10 DAY)
▶	2021-01-15

Result 12

Output

Action Output

#	Time	Action	Message
✓ 1	00:22:17	SELECT ADDDATE("2021-01-05", INTERVAL 10 DAY) L...	1 row(s) returned
✓ 2	00:22:28	SELECT ADDDATE("2021-01-05", INTERVAL 10 DAY) L...	1 row(s) returned

Calculando a diferença de dias entre duas datas.

[illegible]

Alterando a Configuração do formato de uma data.

The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and search. The main editor displays a SQL script with four lines:

```
1 • USE EMPRESAS;  
2 • SELECT *FROM CLIENTES;  
3 • SELECT DATE_FORMAT("2021-01-05", '%D %M %Y')  
4
```

Below the editor, the 'Result Grid' section shows the output of the third query. It includes a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' checkbox. The result is displayed in a table:

	DATE_FORMAT("2021-01-05", '%D %M %Y')
▶	5th January 2021

At the bottom, the 'Result 13' tab is active, showing the 'Output' section. It contains a table with columns for '#', 'Time', 'Action', and 'Message'.

#	Time	Action	Message
✓ 1	21:09:40	SELECT DATE_FORMAT("2021-01-05", '%D %M %Y') LI...	1 row(s) returned

Verificando o dia de uma data.

The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and search. The main editor displays a SQL script with four lines: `USE EMPRESAS;`, `SELECT *FROM CLIENTES;`, `select day("2021-01-07");`, and a blank line. The third line is highlighted. Below the editor, the 'Result Grid' shows a single row with the value '7' for the query `day("2021-01-07")`. The bottom section, titled 'Result 18', shows the 'Action Output' with a table containing execution details.

SQL File 3*

Limit to 1000 rows

```
1 • USE EMPRESAS;
2 • SELECT *FROM CLIENTES;
3 • select day("2021-01-07");
4
```

Result Grid

day("2021-01-07")
7

Result 18

Output

Action Output

#	Time	Action	Message
✓ 1	22:27:20	select day("2021-01-07") LIMIT 0, 1000	1 row(s) returned

Verificando qual dia da semana é uma determinada data.

The screenshot shows a SQL IDE interface. At the top, a toolbar contains various icons for file operations, execution, and search. Below the toolbar, a tab labeled "SQL File 3*" is active. The main editor area contains the following SQL code:

```
1 • USE EMPRESAS;  
2 • SELECT *FROM CLIENTES;  
3 • select dayofweek("2021-06-26");
```

The third line is highlighted. Below the editor, a "Result Grid" section shows the output of the query. It contains a single row with the value "7" under the column header "dayofweek('2021-06-26')".

Below the result grid, there is an "Output" section with a dropdown menu set to "Action Output". It displays a log of actions performed during the execution:

#	Time	Action	Message
2	22:42:12	SELECT *FROM CLIENTES LIMIT 0, 1000	1000 row(s) returned
3	22:42:12	select dayofweek("2021-06-26") LIMIT 0, 1000	1 row(s) returned
4	22:42:18	select dayofweek("2021-06-26") LIMIT 0, 1000	1 row(s) returned

Verificando o dia do ano de uma determinada data

The screenshot shows the SQL Server Enterprise Manager interface. At the top, there's a toolbar with various icons. Below it, a tab labeled "SQL File 3*" is active. The main area contains a query editor with three lines of SQL code:

```
1 • USE EMPRESAS;  
2 • SELECT *FROM CLIENTES;  
3 • select dayofyear("2021-02-21");
```

Below the query editor, there's a section for the query results. It includes a "Result Grid" button, a "Filter Rows:" input field, an "Export:" button, and a "Wrap Cell Content:" checkbox. The results are displayed in a table with two columns: "dayofyear('2021-02-21')" and "52".

Verificando a semana do ano de uma determinada data.

The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and search. The main editor displays a SQL script with three lines:

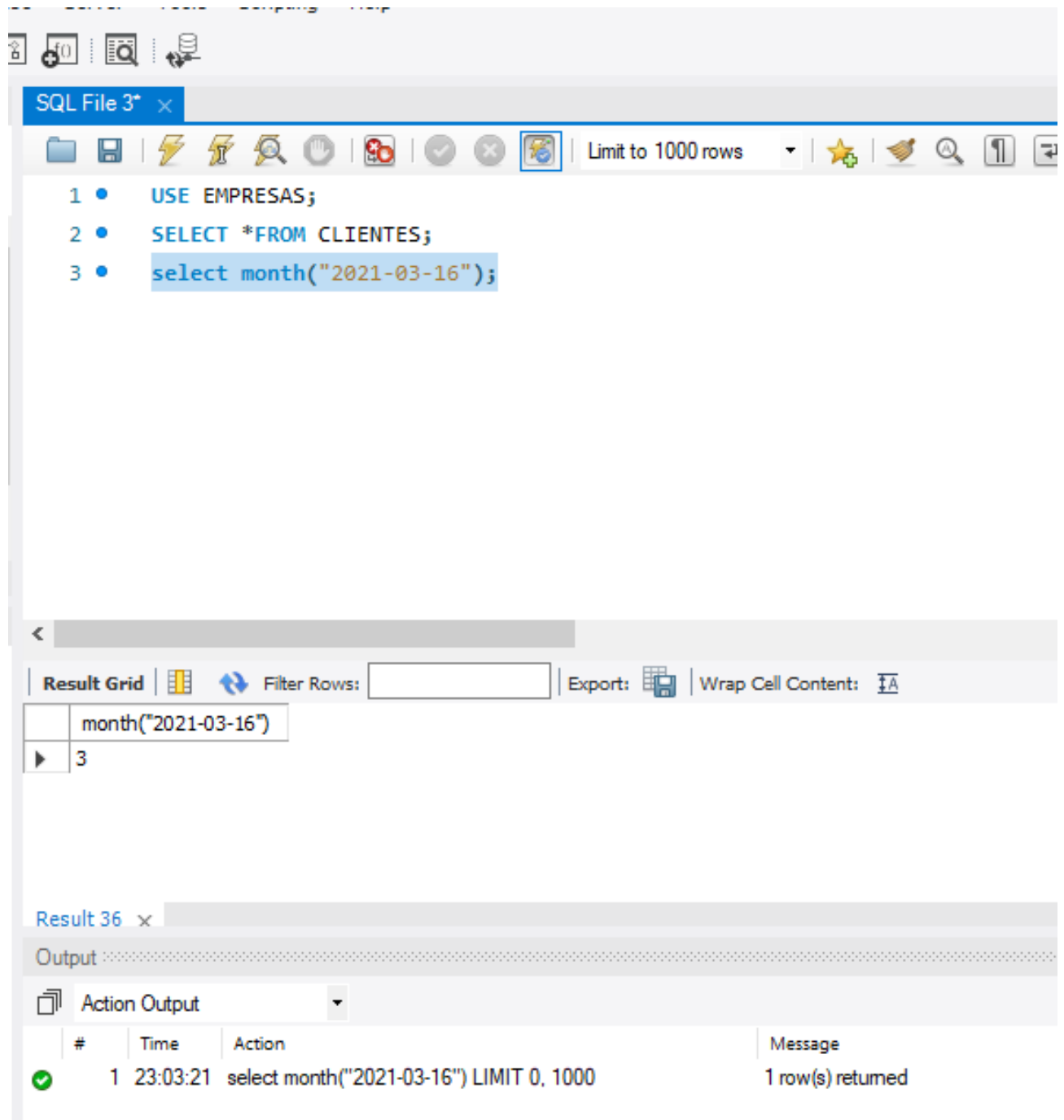
```
1 • USE EMPRESAS;  
2 • SELECT *FROM CLIENTES;  
3 • select weekofyear("2021-05-14");
```

Below the editor, the 'Result Grid' tab is active, showing the results of the third query. The first row contains the text 'weekofyear("2021-05-14")' and the second row contains the value '19'.

At the bottom, the 'Output' tab is active, displaying a table with the following data:

#	Time	Action	Message
✓ 1	22:54:52	select weekofyear("2021-05-14") LIMIT 0, 1000	1 row(s) returned

Retornando o Mês de uma data



The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and settings. The main editor displays a SQL script with three lines: `USE EMPRESAS;`, `SELECT *FROM CLIENTES;`, and `select month("2021-03-16");`. The third line is highlighted. Below the editor, the 'Result Grid' shows a single row with the value '3' for the query `month("2021-03-16")`. The bottom section, 'Result 36', shows the 'Action Output' table with one entry indicating the query was executed successfully and returned 1 row(s).

SQL File 3* x

Limit to 1000 rows

```
1 • USE EMPRESAS;
2 • SELECT *FROM CLIENTES;
3 • select month("2021-03-16");
```

Result Grid

	month("2021-03-16")
▶	3

Result 36 x

Output

Action Output

#	Time	Action	Message
✓ 1	23:03:21	select month("2021-03-16") LIMIT 0, 1000	1 row(s) returned

Verificando o ano de uma determinada data.

The screenshot shows a SQL IDE interface. At the top, there's a toolbar with various icons. Below it, a tab labeled "SQL File 3*" is active. The main editor area contains the following SQL code:

```
1 • USE EMPRESAS;  
2 • SELECT *FROM CLIENTES;  
3 • select year("2021-04-05");
```

Below the editor, there's a "Result Grid" section. It has a "Filter Rows:" input field and an "Export:" button. The grid displays the result of the query:

	year("2021-04-05")
▶	2021

Below the result grid, there's a "Result 42" tab. It has an "Output" section with a dropdown menu set to "Action Output". Below this, there's a table showing the execution details:

#	Time	Action	Message
✓ 1	23:12:22	select year("2021-04-05") LIMIT 0, 1000	1 row(s) returned