

Criando tabelas com SQL

Fonte: <https://dev.mysql.com/doc/refman/8.0/en/create-table.html>
<https://elias.praciano.com/2014/01/como-criar-tabelas-no-mysql/>

Manipulando tabelas no MySQL

Neste tutorial vamos verificar as características e sintaxe para criação de tabelas; como sabemos, a tabela representa a entidade que, por sua vez, abstrai um objeto do “mundo real” e seus respectivos atributos; como já verificamos anteriormente, o SQL para cada SGBD em particularidades, porém, em a lógica é praticamente a mesma.

Antes de qualquer interação com a ferramenta, vamos entender a estrutura para a criação de uma tabela; segue exemplo abaixo da sintaxe básica que podemos utilizar:

```
CREATE TABLE alunos(  
  codigo int(4) AUTO_INCREMENT,  
  nome varchar(30) NOT NULL,  
  email varchar(50),  
  PRIMARY KEY (codigo)  
);
```

Em primeiro lugar vamos entender os comandos que fazem parte desta criação; veja que o comando “CREATE TABLE alunos” é a indicação de qual objeto estamos criando e seu respectivo nome; a partir do primeiro parênteses estamos definindo as características deste objeto, ou seja, quais serão os atributos da entidade “alunos”; os tipo de dados podem dependendo do SGBD usado; segue alguns exemplos abaixo:

Numéricos:

INT	Número inteiro de tamanho comum;
DECIMAL	Número decimal, de ponto fixo;
FLOAT	Número de ponto flutuante de precisão simples (32 bits);
DOUBLE	Número de ponto flutuante de precisão dupla (64 bits);
BIT	Campo de um bit.

String:

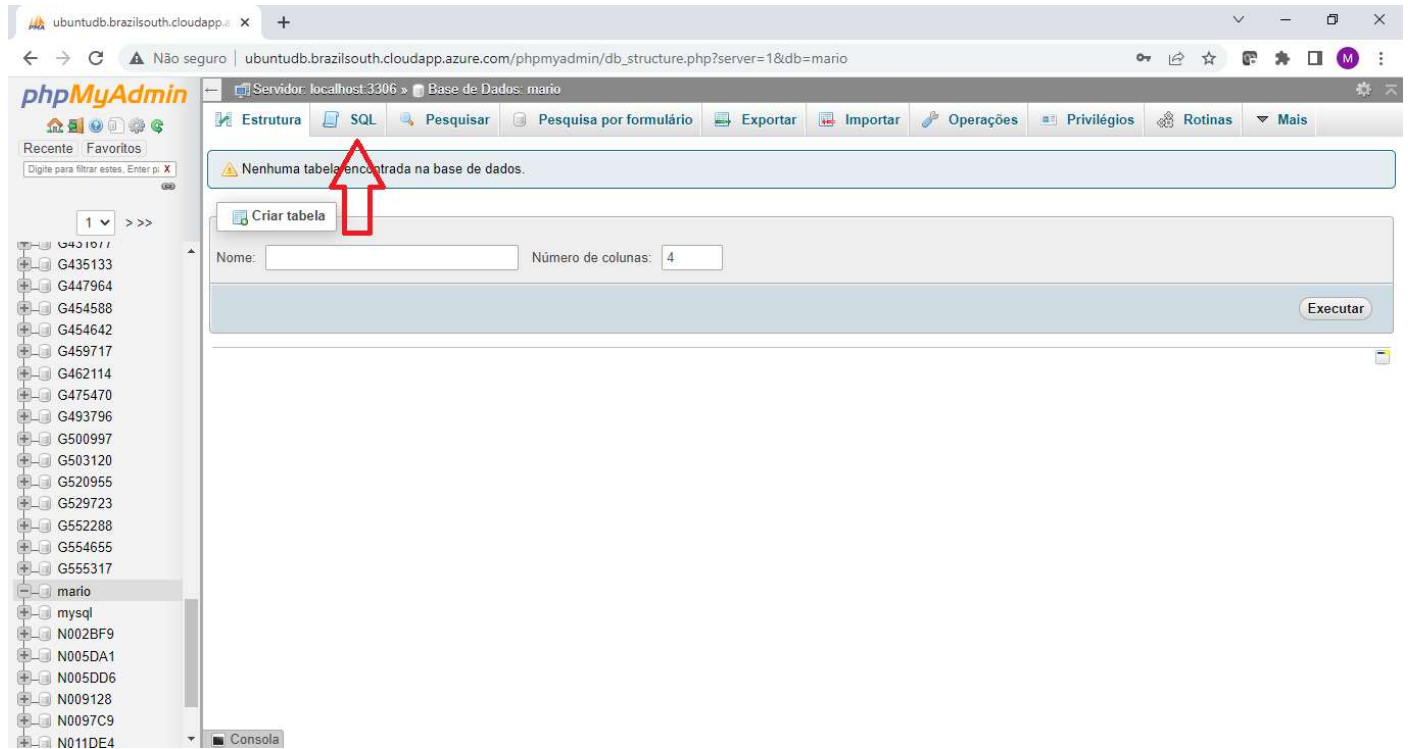
VARCHAR	String de tamanho variável e não-binária;
BINARY	String binária de tamanho fixo;
BLOB	Binary Large Object ou seja, Objeto Grande Binário (pequeno);
LOB	BLOB grande;
TEXT	String não-binária e pequena;
LONGTEXT	String não-binária de tamanho grande;

Data/Hora:

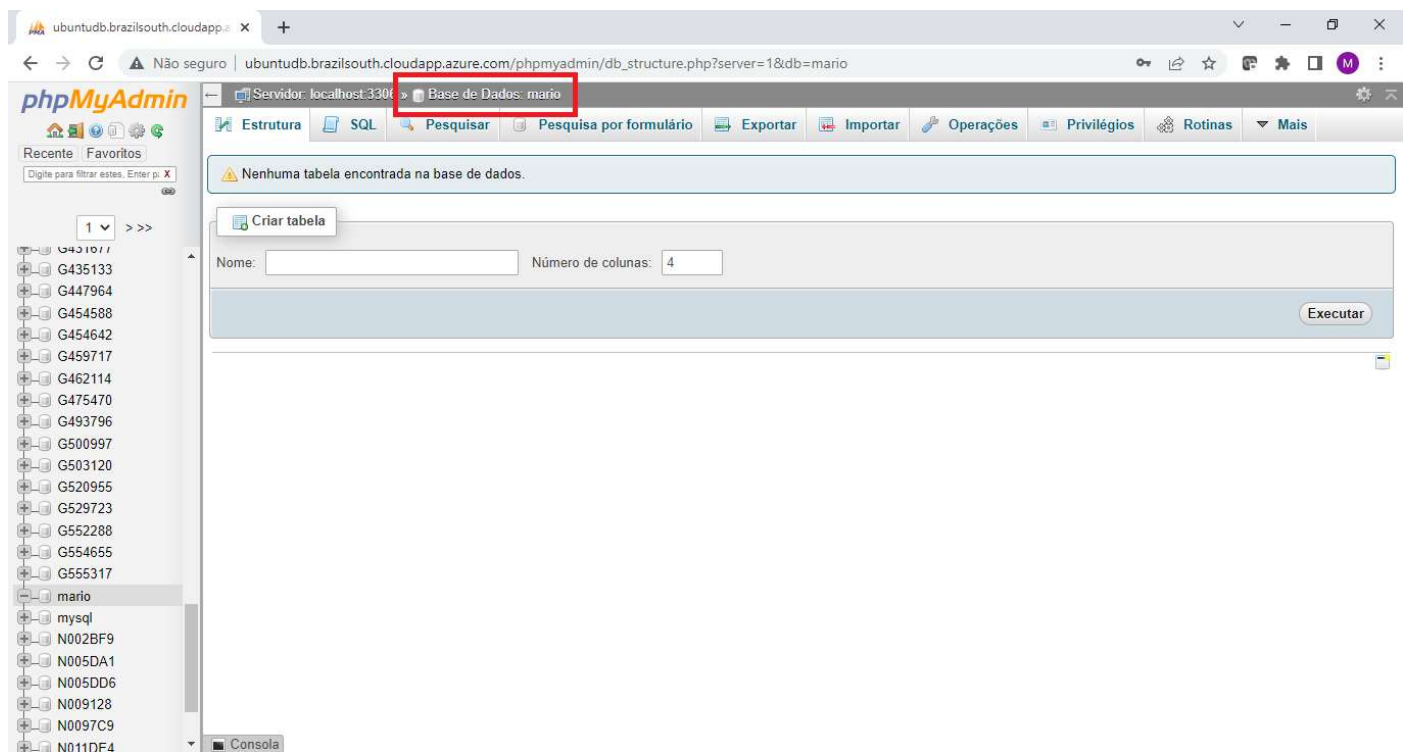
DATE	O valor referente a uma data no formato 'CCYY-MM-DD'. Por exemplo 1985-11-25 (ano-mês-dia). O 'CC' se refere aos dois dígitos do século (Century, em inglês);
TIME	Um valor horário no formato 'hh:mm:ss' (hora:minutos:segundos);

Continuando a análise do nosso comando CREATE, veja que a coluna/atributo “codigo” foi definida como AUTO_INCREMENT; este recurso permite automatizar um código que sirva de chave primária na tabela, sendo responsabilidade do SGBD incrementá-lo conforme a inserção de novos registros; veja que a chave primária da nossa tabela (PRIMARY KEY) foi definida como sendo o campo “codigo”; o segundo parênteses e o ponto e vírgula finalizam a criação.

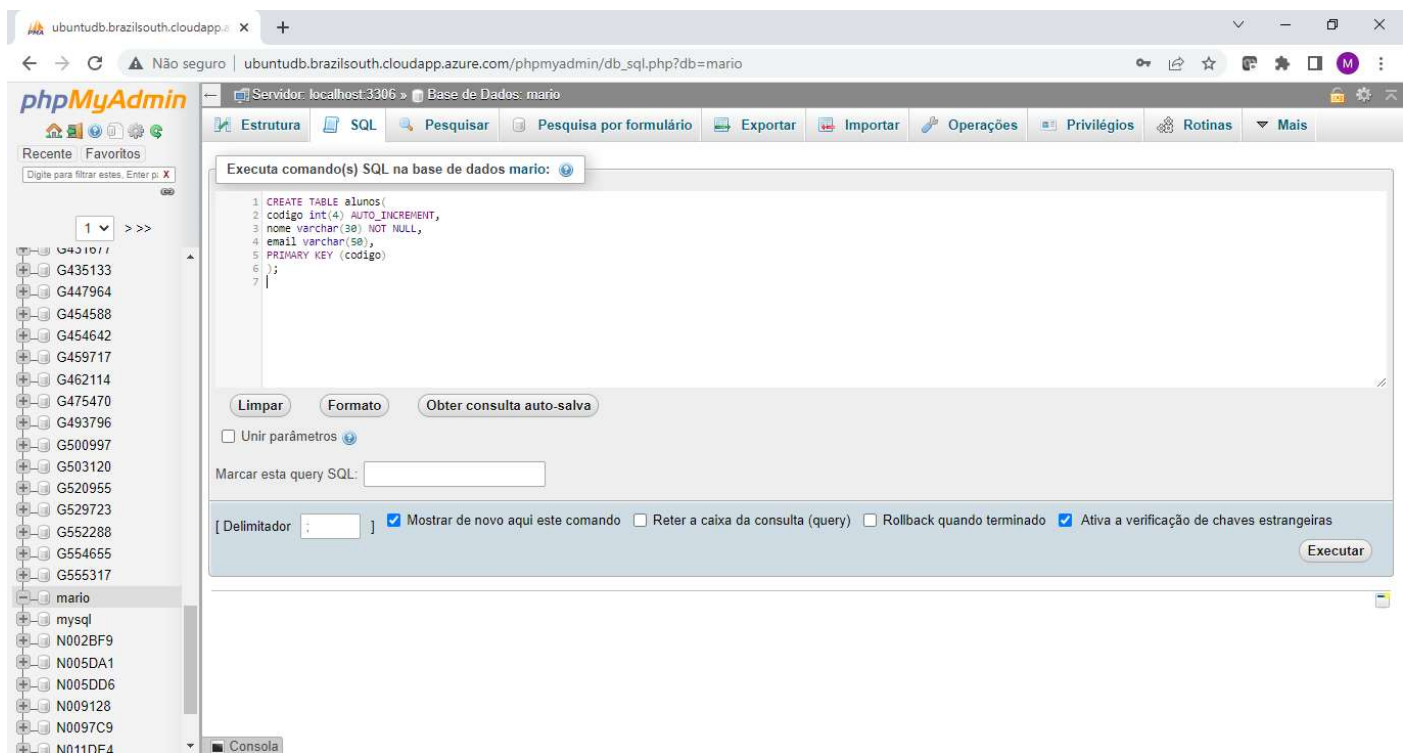
Vale destacar que a entrada NOT NULL define que um determinado campo seja de preenchimento obrigatório, ou seja, a coluna/atributo “nome” SEMPRE deve ser inserida; caso contrário o SGBD retorna uma falha e não conclui a inclusão do novo registro. No *phpMyAdmin*, selecione a aba **SQL**:



Certifique-se que o seu banco de dados foi selecionado:



Agora podemos iniciar a criação de nossa tabela alunos:

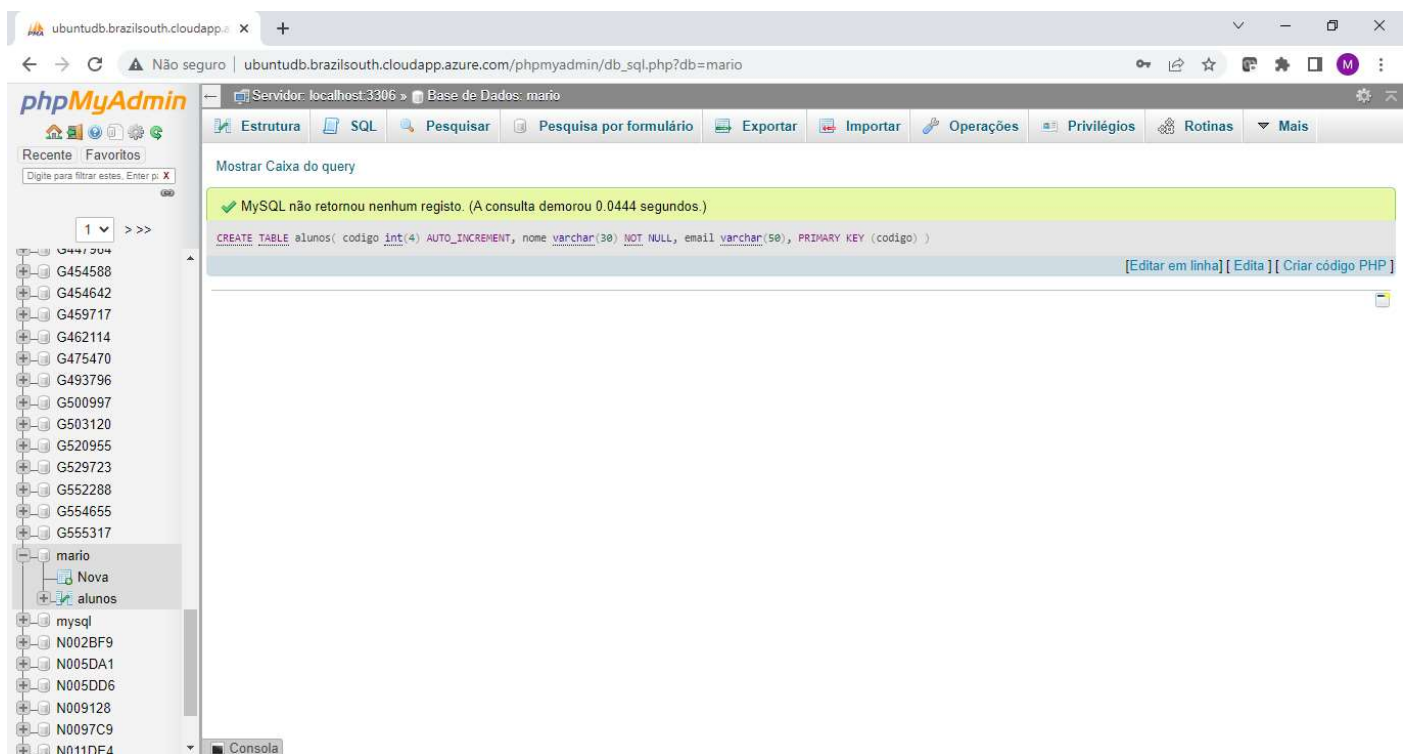


The screenshot shows the phpMyAdmin interface for the 'mario' database. The SQL tab is selected, and the following command is entered in the text area:

```
1 CREATE TABLE alunos(
2   codigo int(4) AUTO_INCREMENT,
3   nome varchar(30) NOT NULL,
4   email varchar(50),
5   PRIMARY KEY (codigo)
6 );
7
```

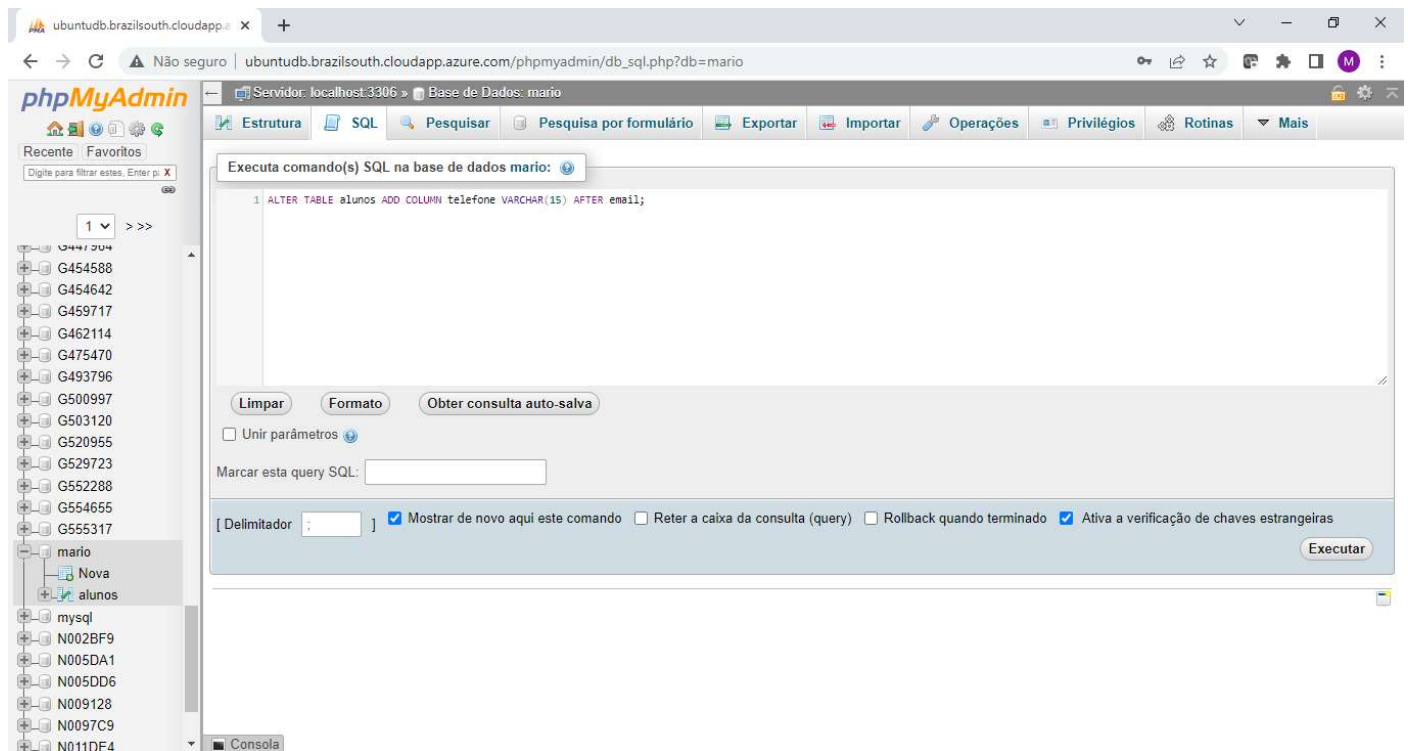
Below the text area, there are buttons for 'Limpar', 'Formato', and 'Obter consulta auto-salva'. There is also a checkbox for 'Unir parâmetros' and a text input for 'Marcar esta query SQL:'. At the bottom, there are checkboxes for 'Mostrar de novo aqui este comando', 'Reter a caixa da consulta (query)', 'Rollback quando terminado', and 'Ativa a verificação de chaves estrangeiras'. An 'Executar' button is located at the bottom right.

Após execução do comando, o resultado deve ser similar ao exemplo abaixo:



The screenshot shows the phpMyAdmin interface after the SQL command has been executed. The 'Mostrar Caixa do query' section displays a green message: 'MySQL não retornou nenhum registro. (A consulta demorou 0.0444 segundos.)'. Below this message, the executed SQL command is shown: 'CREATE TABLE alunos(codigo int(4) AUTO_INCREMENT, nome varchar(30) NOT NULL, email varchar(50), PRIMARY KEY (codigo))'. There are links for '[Editar em linha]', '[Edita]', and '[Criar código PHP]'.

Podemos alterar a tabela criada anteriormente, manipulando seus campos através do comando ALTER; é importante entender que este procedimento deve ser adotado com cautela, evitando que dados importantes sejam removidos indevidamente. Primeiramente vamos adicionar a coluna/atributo “telefone” na tabela alunos:

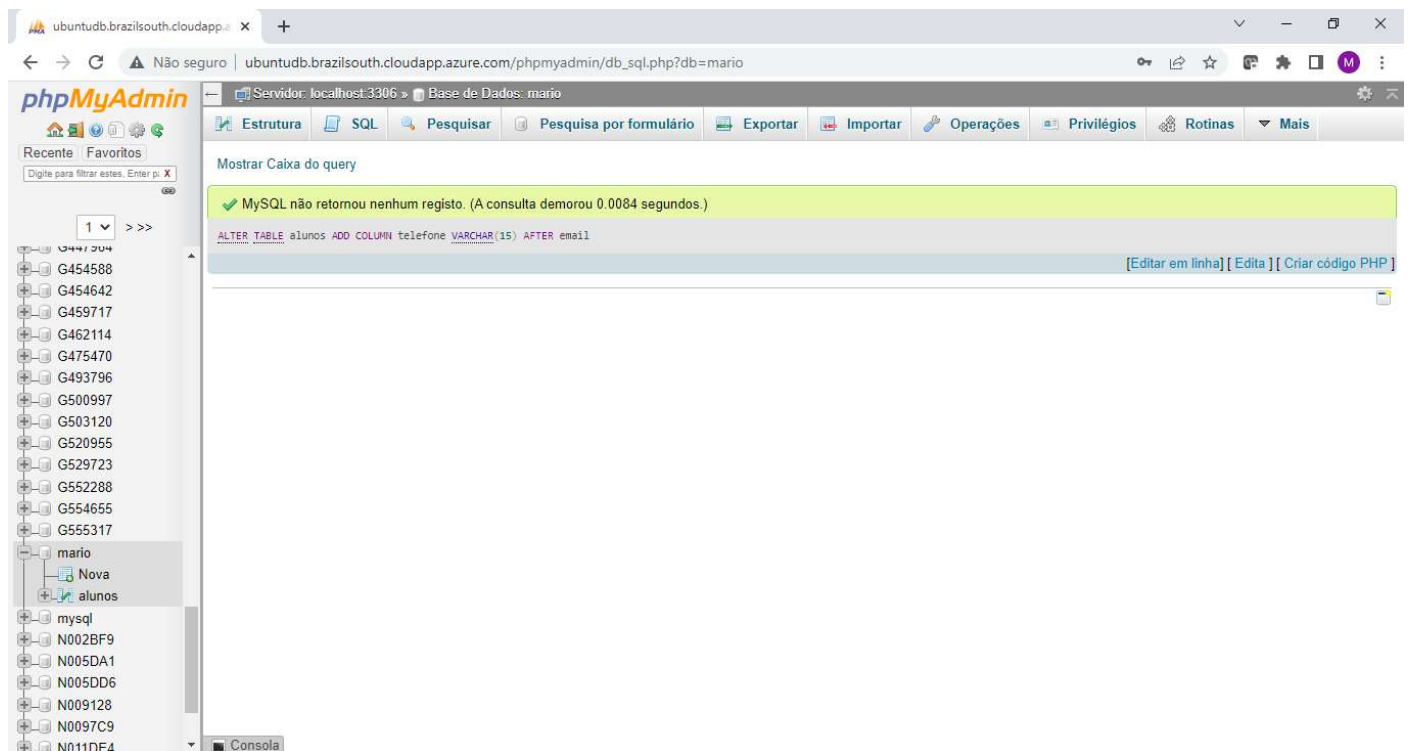


The screenshot shows the phpMyAdmin interface for the 'mario' database. The 'alunos' table is selected in the left sidebar. The SQL command entered in the main area is:

```
1 ALTER TABLE alunos ADD COLUMN telefone VARCHAR(15) AFTER email;
```

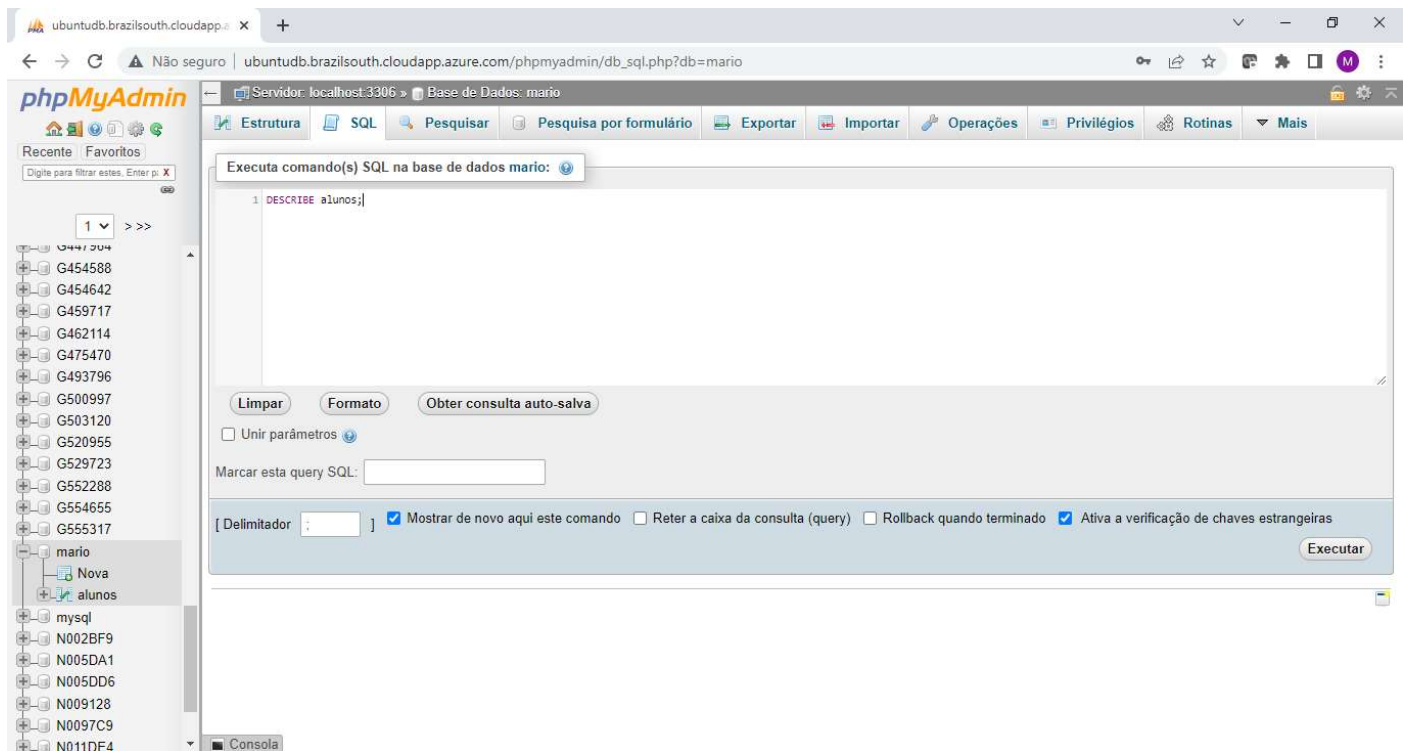
Below the command, there are buttons for 'Limpar', 'Formato', and 'Obter consulta auto-salva'. There is also a checkbox for 'Unir parâmetros' and a text input for 'Marcar esta query SQL:'. At the bottom, there are checkboxes for 'Mostrar de novo aqui este comando', 'Reter a caixa da consulta (query)', 'Rollback quando terminado', and 'Ativa a verificação de chaves estrangeiras'. The 'Executar' button is at the bottom right.

Veja que o comando é acatado pelo SGBD:



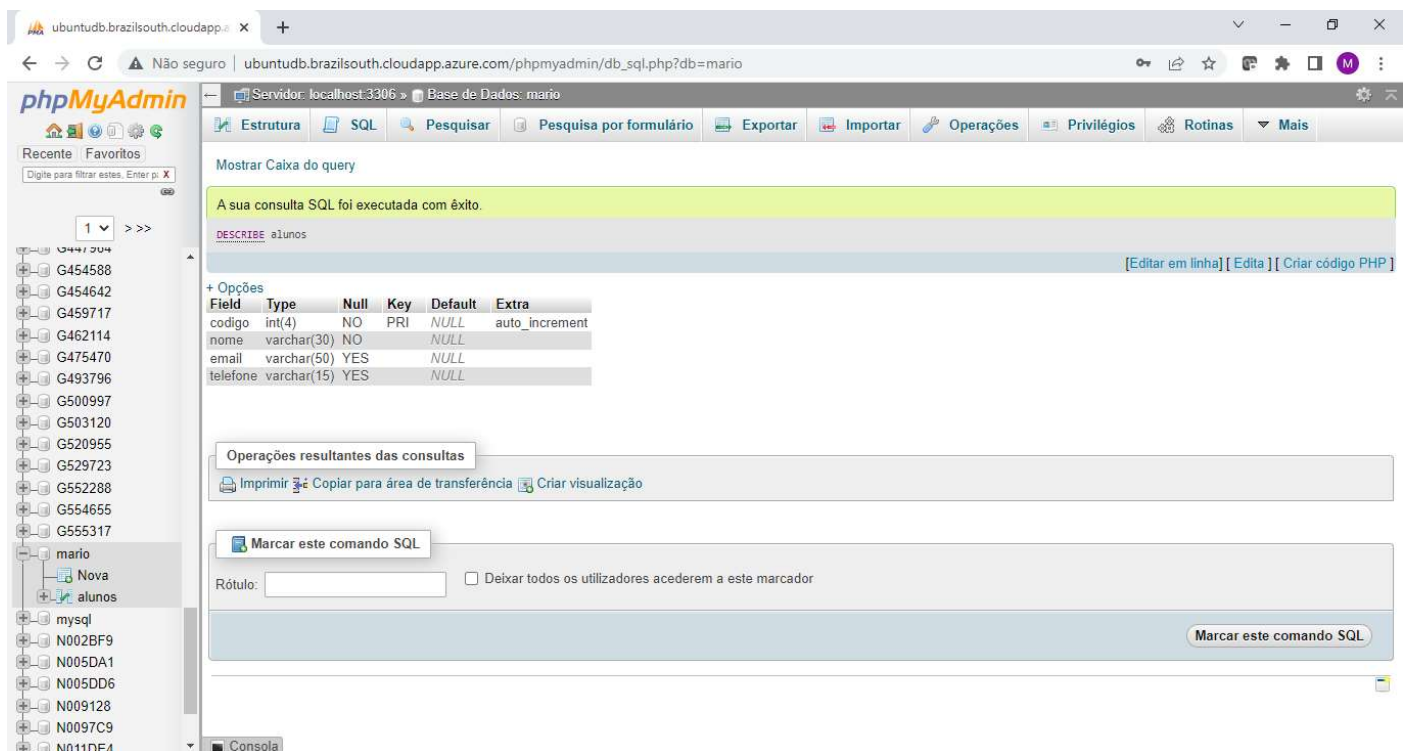
The screenshot shows the phpMyAdmin interface after the SQL command has been executed. A green message box at the top states: 'MySQL não retornou nenhum registro. (A consulta demorou 0.0084 segundos.)'. Below this, the executed SQL command is displayed: 'ALTER TABLE alunos ADD COLUMN telefone VARCHAR(15) AFTER email'. There are links for '[Editar em linha]', '[Edita]', and '[Criar código PHP]'.

Podemos verificar alteração utilizando comando DESCRIBE:



The screenshot shows the phpMyAdmin interface for a database named 'mario'. The 'SQL' tab is selected, and the query box contains the command: `1 DESCRIBE alunos;`. Below the query box, there are buttons for 'Limpar', 'Formato', and 'Obter consulta auto-salva'. There is also a checkbox for 'Unir parâmetros' and a text field for 'Marcar esta query SQL:'. At the bottom, there are checkboxes for 'Mostrar de novo aqui este comando', 'Reter a caixa da consulta (query)', 'Rollback quando terminado', and 'Ativa a verificação de chaves estrangeiras'. An 'Executar' button is located at the bottom right.

Segue o resultado esperado:

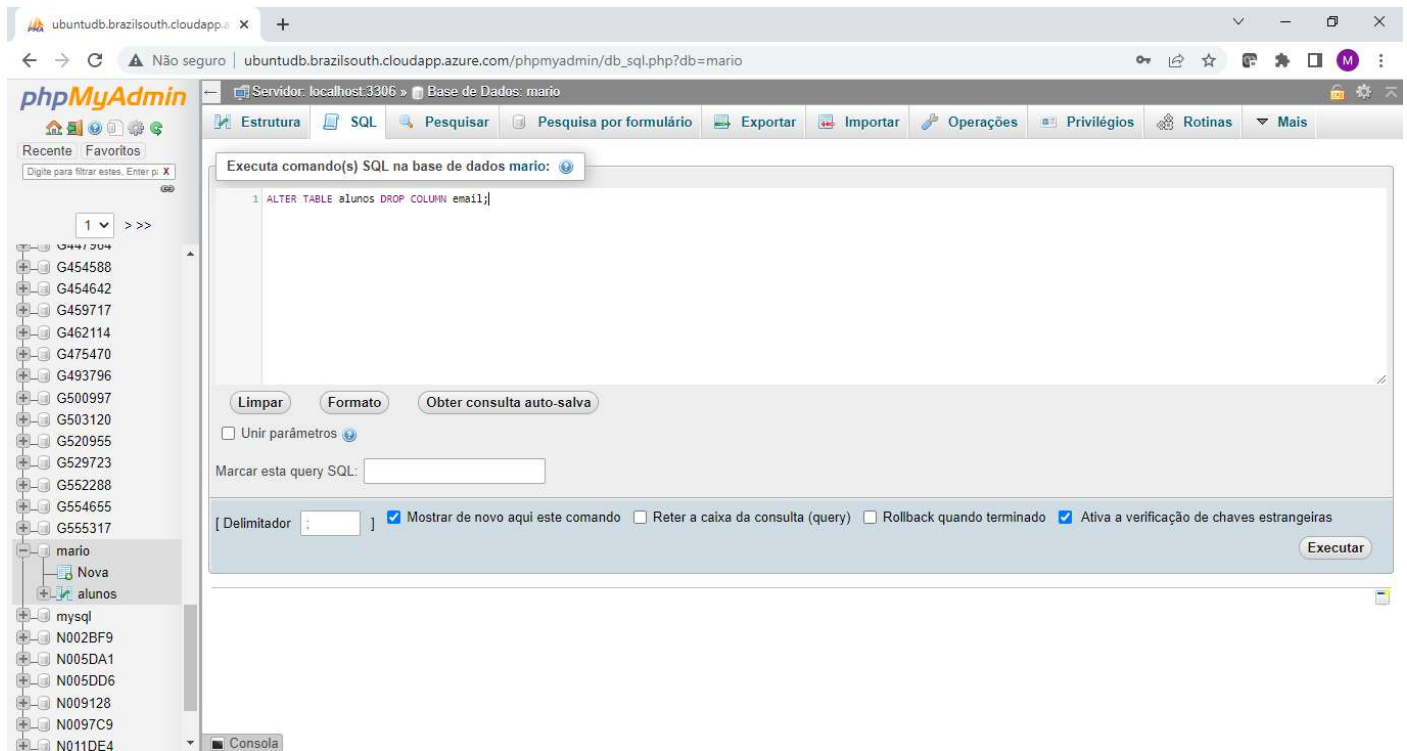


The screenshot shows the phpMyAdmin interface after executing the 'DESCRIBE alunos;' command. A green message box at the top states: 'A sua consulta SQL foi executada com êxito.' Below this, the command 'DESCRIBE alunos' is displayed. A table titled '+ Opções' shows the structure of the 'alunos' table:

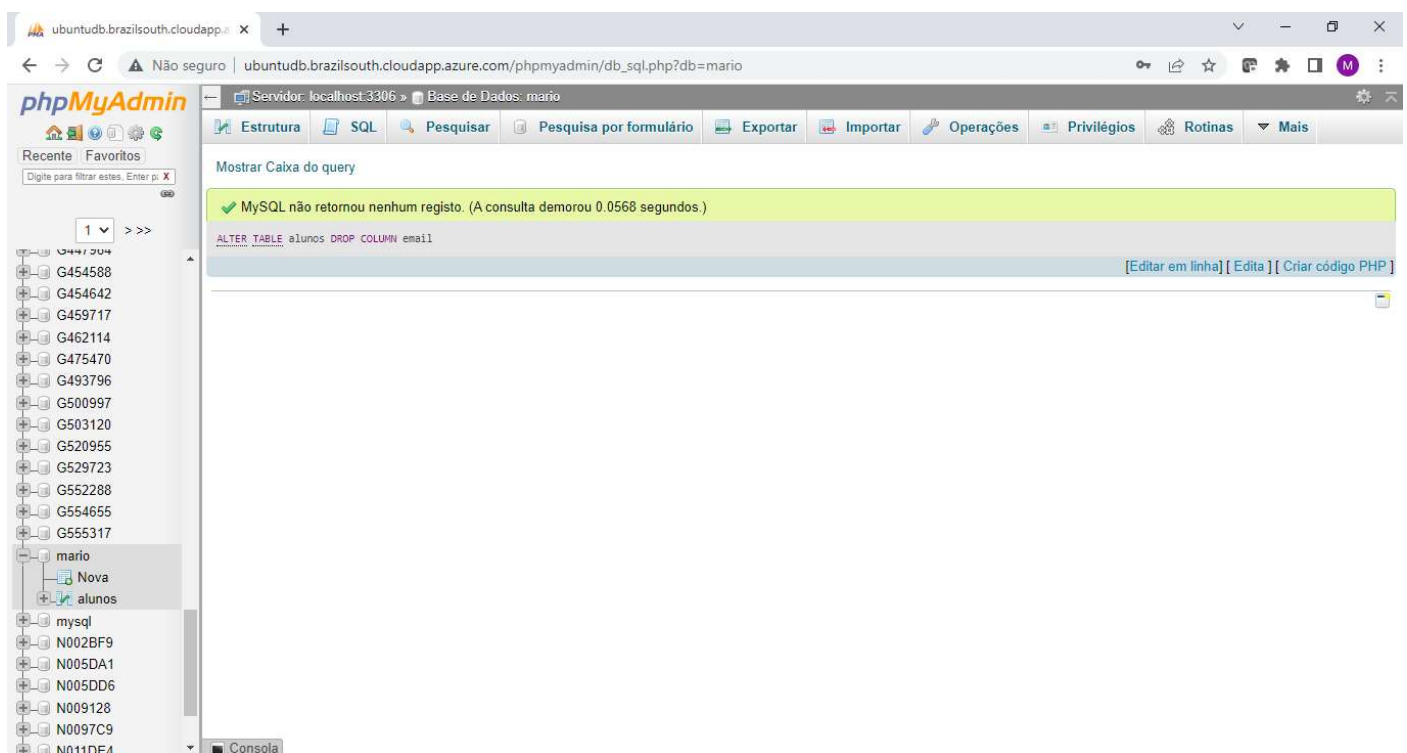
Field	Type	Null	Key	Default	Extra
codigo	int(4)	NO	PRI	NULL	auto_increment
nome	varchar(30)	NO		NULL	
email	varchar(50)	YES		NULL	
telefone	varchar(15)	YES		NULL	

Below the table, there are buttons for 'Imprimir', 'Copiar para área de transferência', and 'Criar visualização'. At the bottom, there is a section titled 'Operações resultantes das consultas' and a 'Marcar este comando SQL' section with a 'Rótulo:' text field and a checkbox 'Deixar todos os utilizadores acederm a este marcador'. A 'Marcar este comando SQL' button is at the bottom right.

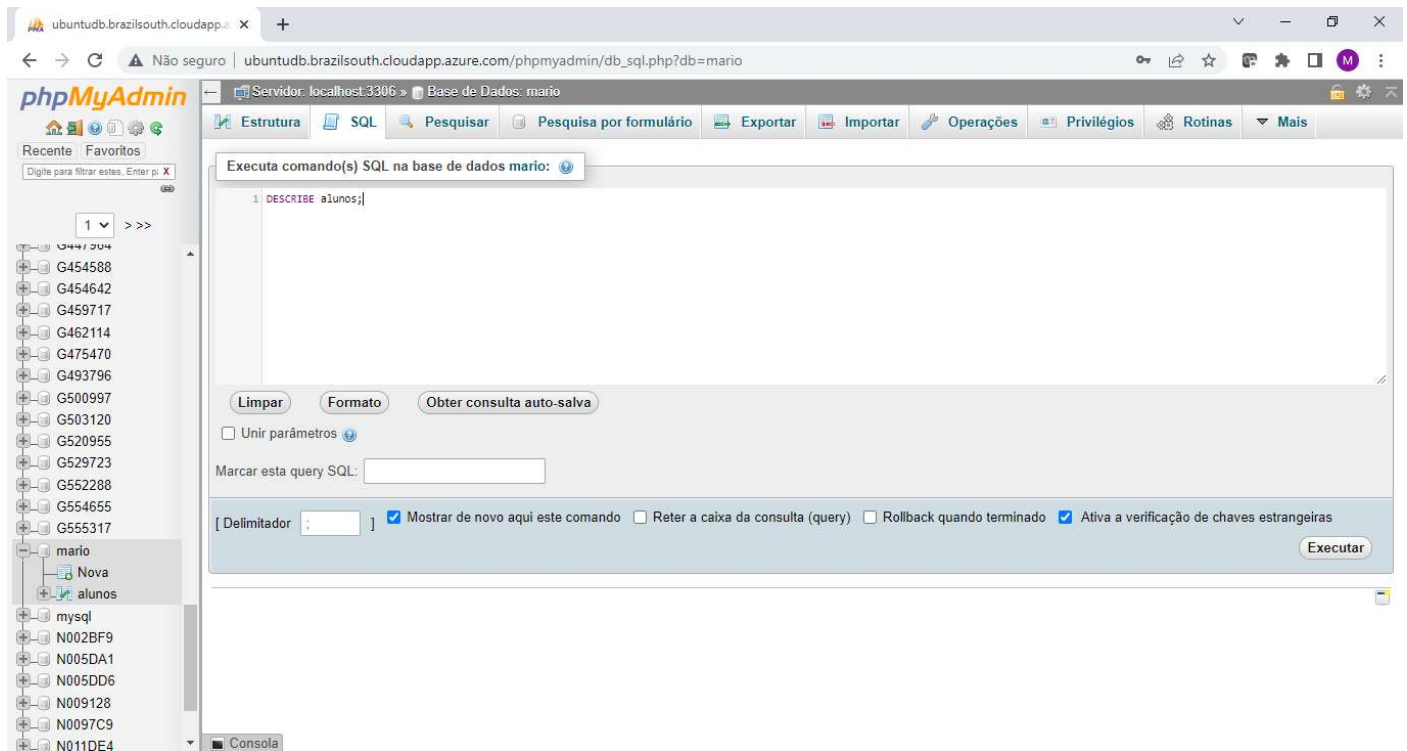
Agora vamos remover o campo “email” utilizando mesmo comando, porém, concatenado com a função DROP:



Veja que o comando é acatado pelo SGBD:

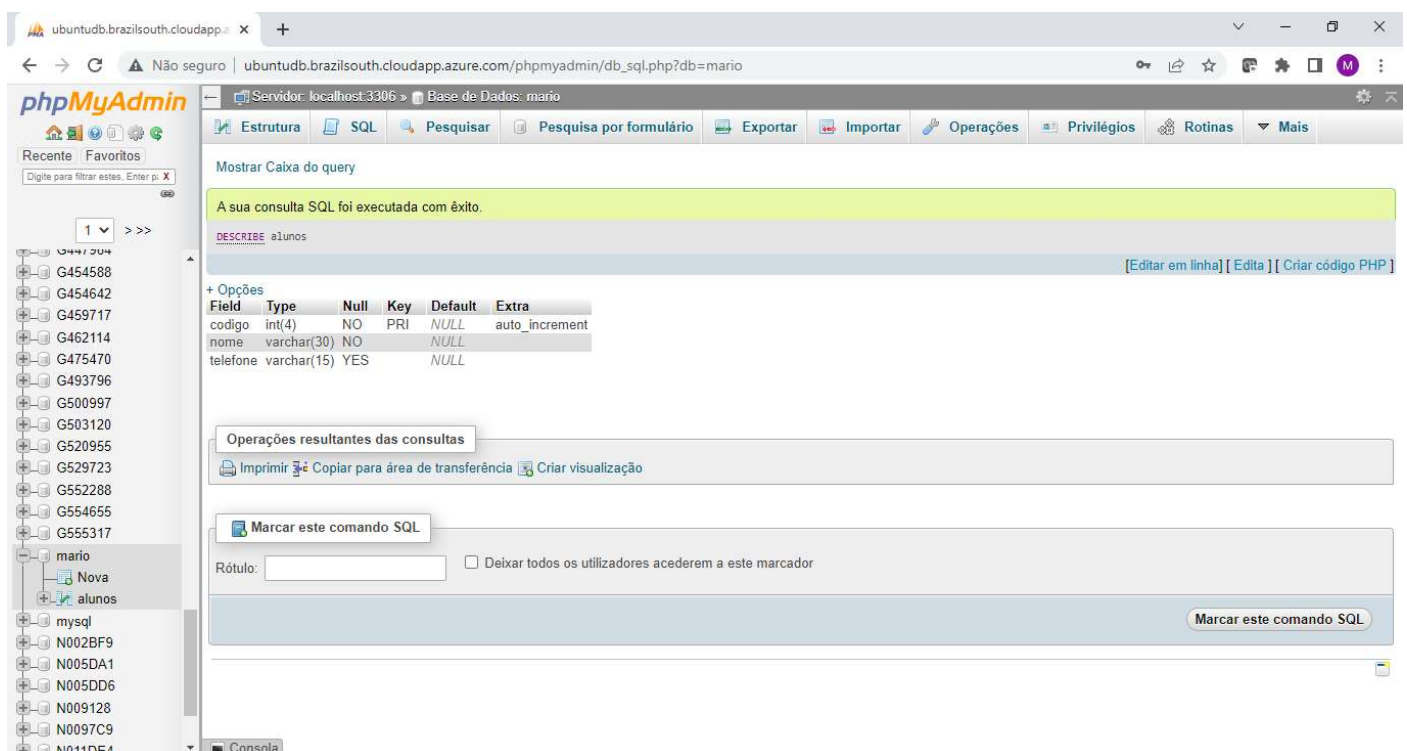


Verificando novamente a estrutura da tabela com o comando DESCRIBE:



The screenshot shows the phpMyAdmin interface with the 'SQL' tab selected. The query box contains the command: `1 DESCRIBE alunos;`. Below the query box, there are buttons for 'Limpar', 'Formato', and 'Obter consulta auto-salva'. There is also a checkbox for 'Unir parâmetros' and a text input for 'Marcar esta query SQL:'. At the bottom, there are checkboxes for 'Mostrar de novo aqui este comando', 'Reter a caixa da consulta (query)', 'Rollback quando terminado', and 'Ativa a verificação de chaves estrangeiras'. The 'Executar' button is at the bottom right.

Segue o resultado esperado:



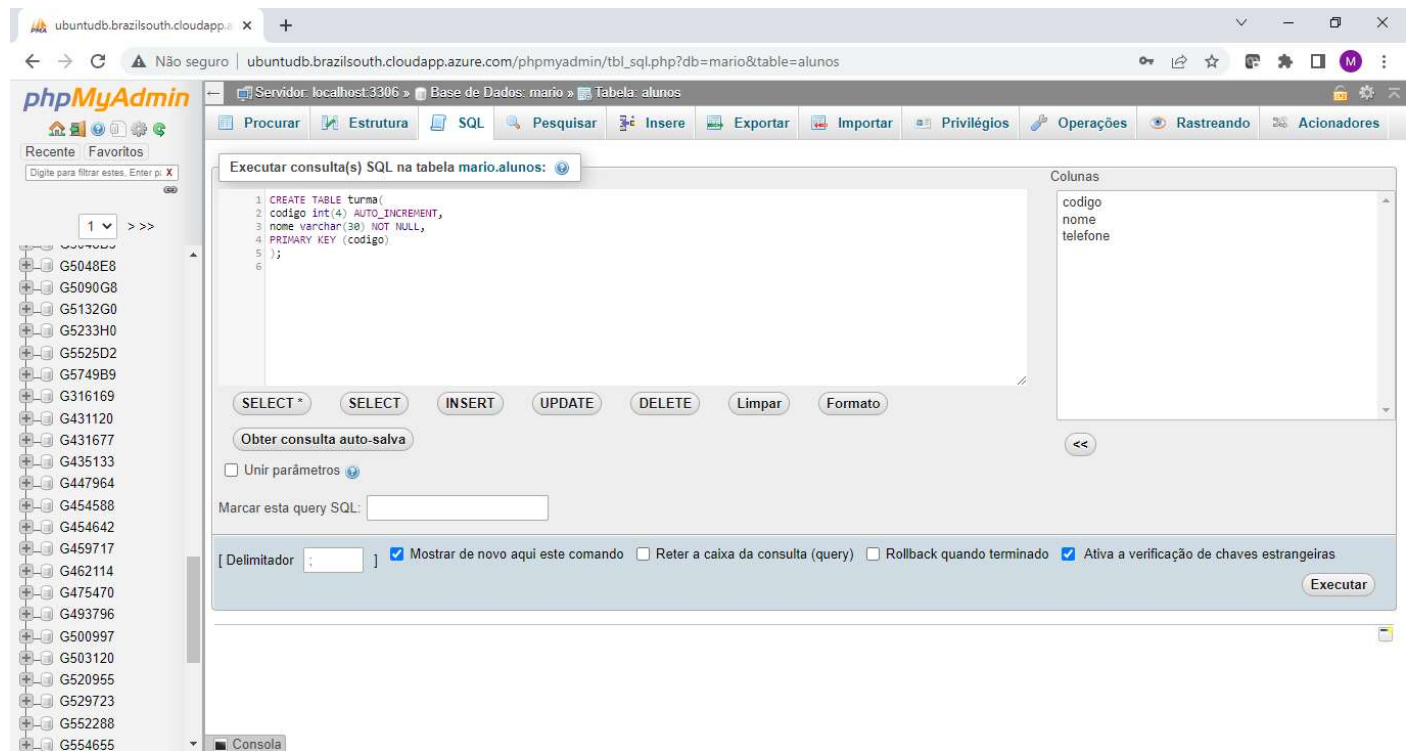
The screenshot shows the phpMyAdmin interface with the 'SQL' tab selected. The query box contains the command: `DESCRIBE alunos`. Below the query box, there is a message: 'A sua consulta SQL foi executada com êxito.' and a link to '[Editar em linha] [Edita] [Criar código PHP]'. Below this, there is a table showing the structure of the 'alunos' table:

Field	Type	Null	Key	Default	Extra
codigo	int(4)	NO	PRI	NULL	auto_increment
nome	varchar(30)	NO		NULL	
telefone	varchar(15)	YES		NULL	

Below the table, there are buttons for 'Operações resultantes das consultas', 'Imprimir', 'Copiar para área de transferência', and 'Criar visualização'. There is also a section for 'Marcar este comando SQL' with a text input for 'Rótulo:' and a checkbox for 'Deixar todos os utilizadores acederem a este marcador'. The 'Marcar este comando SQL' button is at the bottom right.

Agora vamos criar a tabela turma para estabelecer o relacionamento com a entidade alunos:

```
CREATE TABLE turma(
codigo int(4) AUTO_INCREMENT,
nome varchar(30) NOT NULL,
PRIMARY KEY (codigo)
);
```

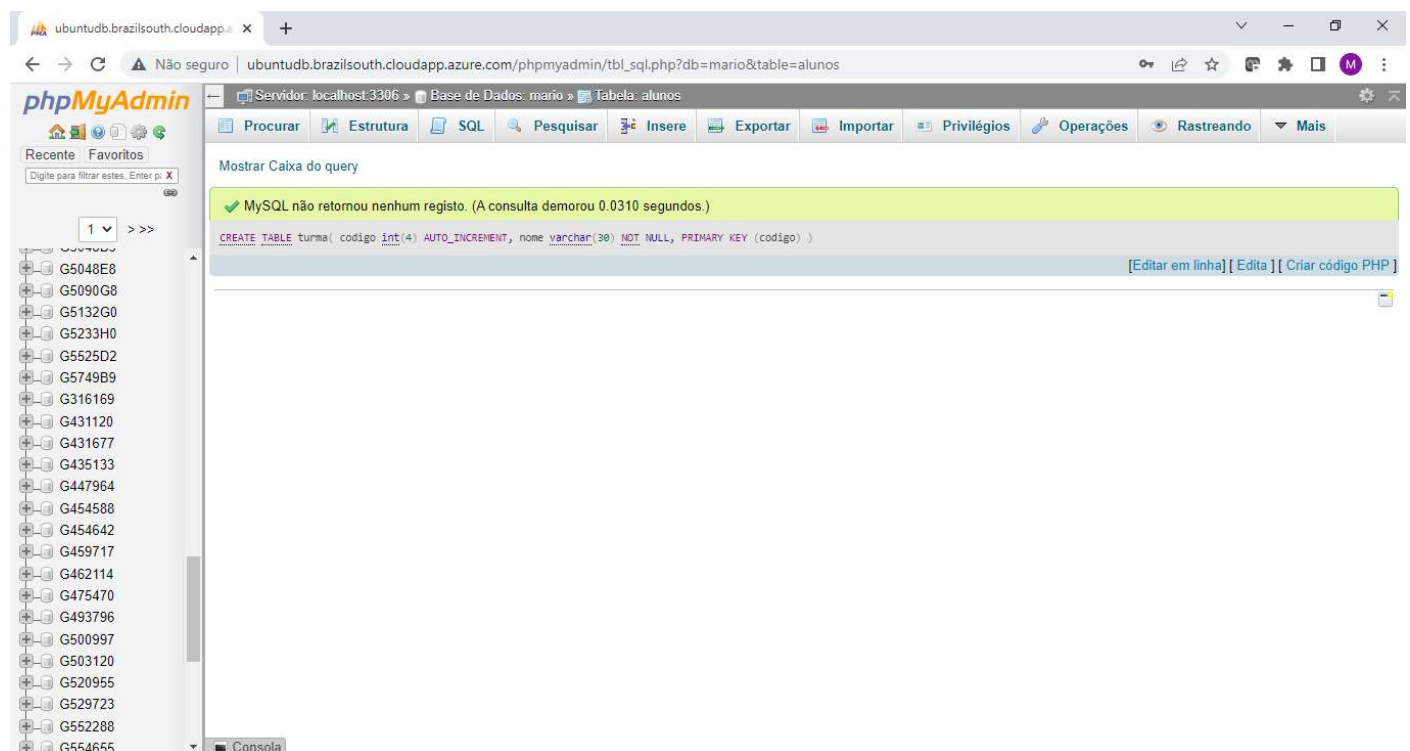


The screenshot shows the phpMyAdmin interface for a database named 'mario'. The 'Tabela: alunos' tab is selected. The SQL query editor contains the following code:

```
1 CREATE TABLE turma(
2 codigo int(4) AUTO_INCREMENT,
3 nome varchar(30) NOT NULL,
4 PRIMARY KEY (codigo)
5 );
6
```

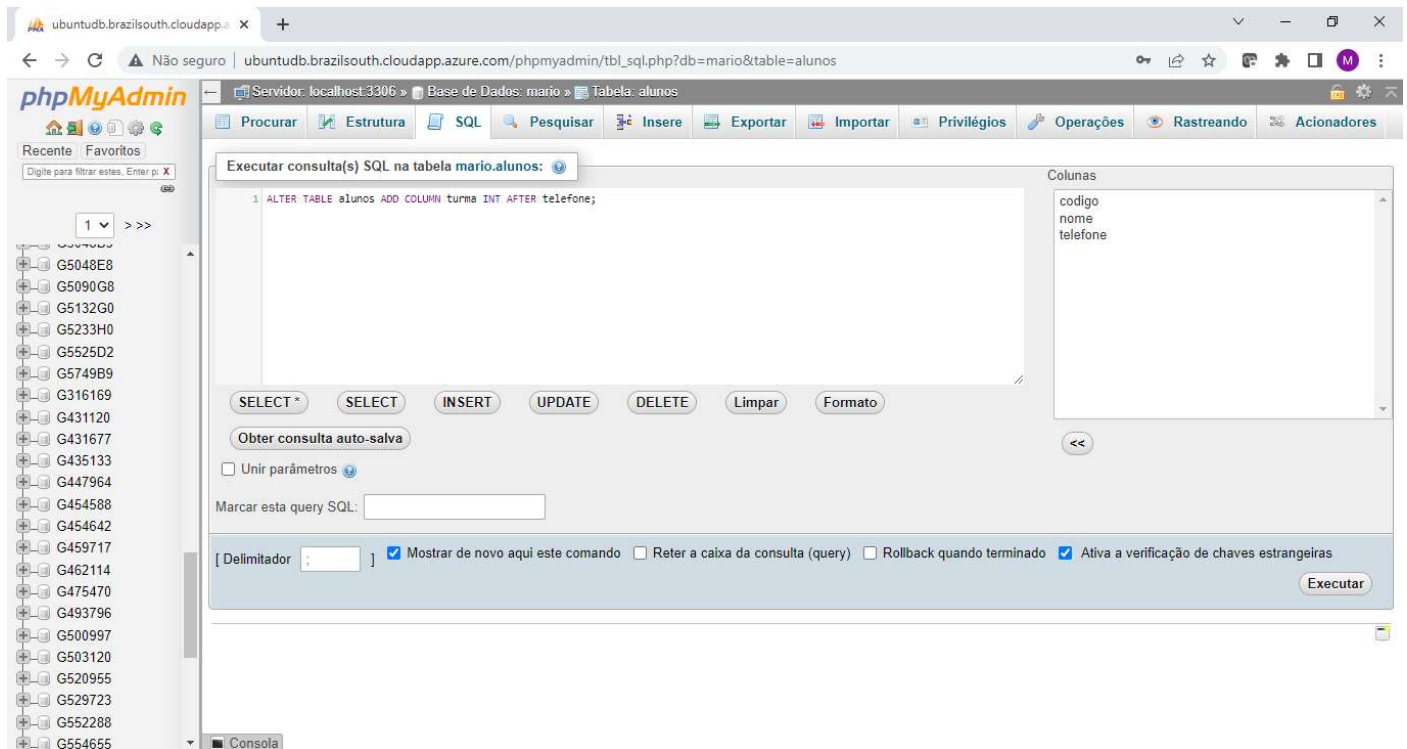
The 'Executar consulta(s) SQL na tabela mario.alunos:' button is highlighted. The 'Colunas' panel on the right shows the columns: 'codigo', 'nome', and 'telefone'. The 'Mostrar de novo aqui este comando' checkbox is checked.

Abaixo o resultado esperado:

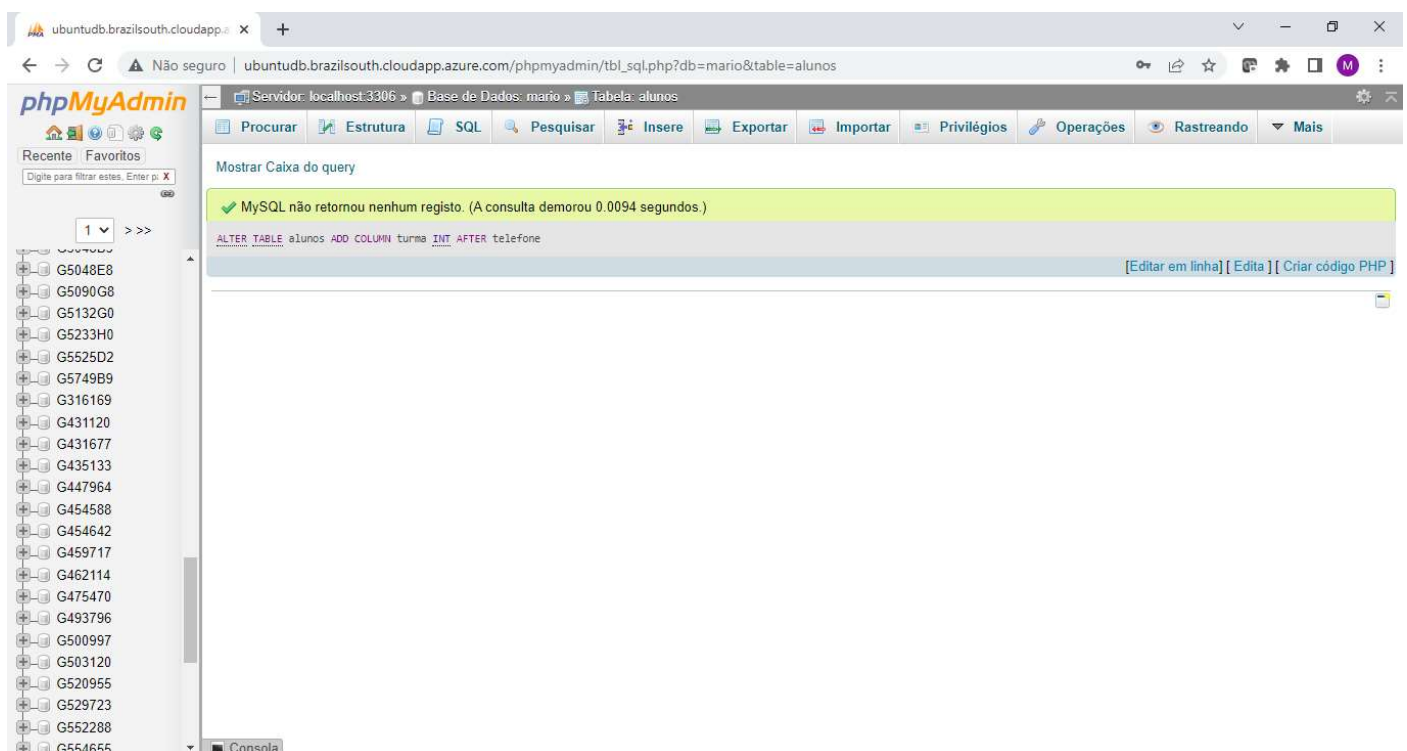


The screenshot shows the phpMyAdmin interface after the SQL query has been executed. A green message box indicates: 'MySQL não retornou nenhum registro. (A consulta demorou 0.0310 segundos.)'. Below the message, the executed SQL query is displayed: 'CREATE TABLE turma(codigo int(4) AUTO_INCREMENT, nome varchar(30) NOT NULL, PRIMARY KEY (codigo))'. The 'Mais' button is visible in the top right corner.

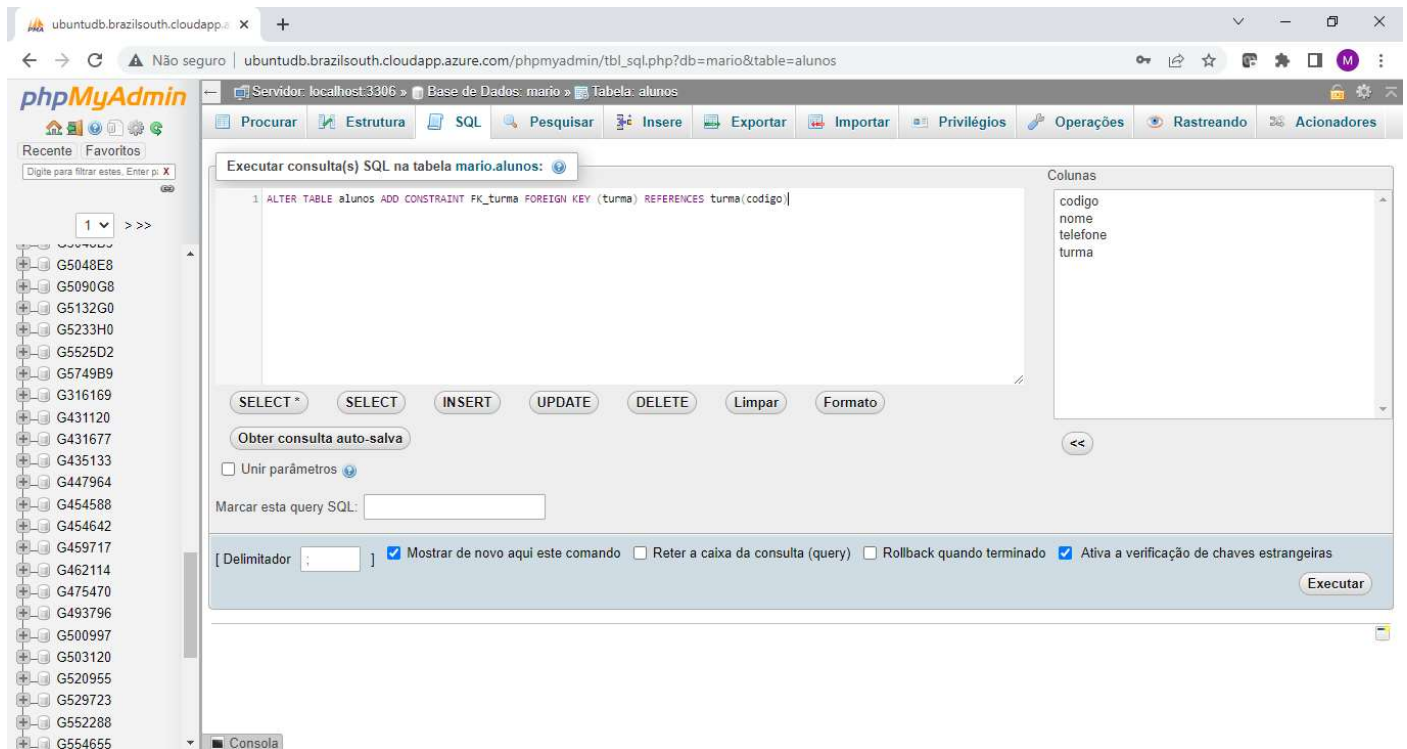
Na tabela alunos, insira o atributo turma:



Abaixo o resultado esperado:



Agora estabeleça o relacionamento entre as entidades, criando a chave estrangeira na tabela alunos:

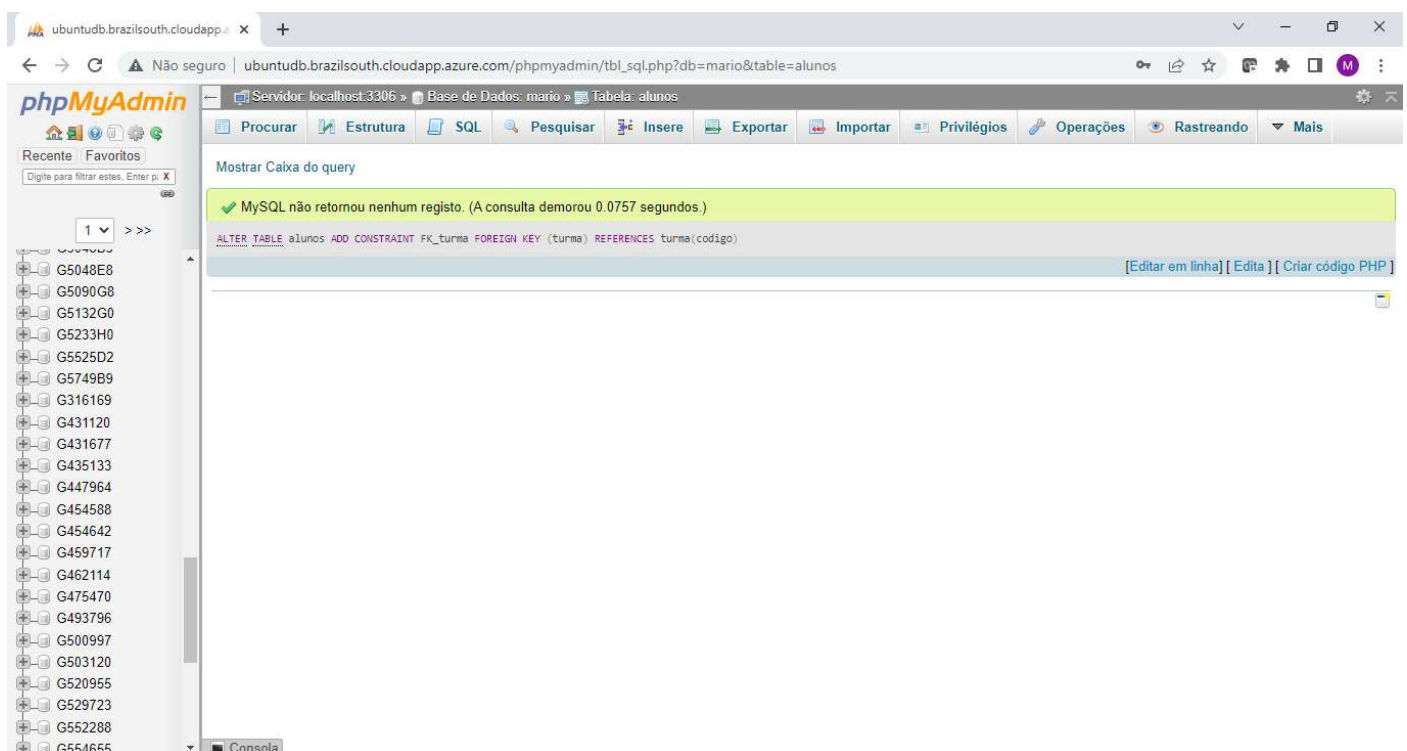


The screenshot shows the phpMyAdmin interface for the 'alunos' table. The SQL query editor is active, displaying the following query:

```
1 ALTER TABLE alunos ADD CONSTRAINT FK_turna FOREIGN KEY (turna) REFERENCES turma(codigo);
```

The interface includes a sidebar with a list of tables, a top navigation bar with various tools (Procurar, Estrutura, SQL, etc.), and a right sidebar showing the columns of the 'alunos' table: codigo, nome, telefone, and turma.

O resultado deve ser similar a tela abaixo:



The screenshot shows the phpMyAdmin interface after executing the SQL query. A green message box displays the result:

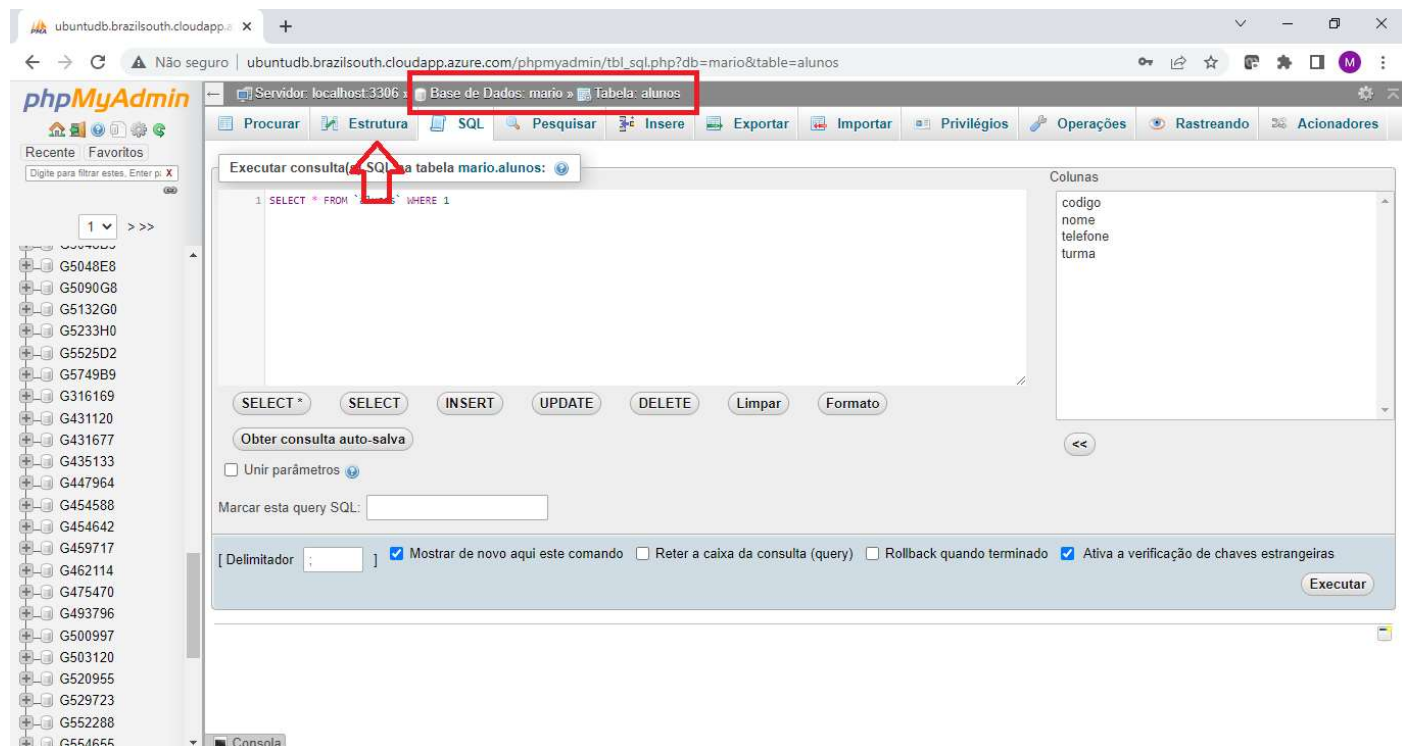
MySQL não retornou nenhum registro. (A consulta demorou 0.0757 segundos.)

Below the message, the executed SQL query is shown:

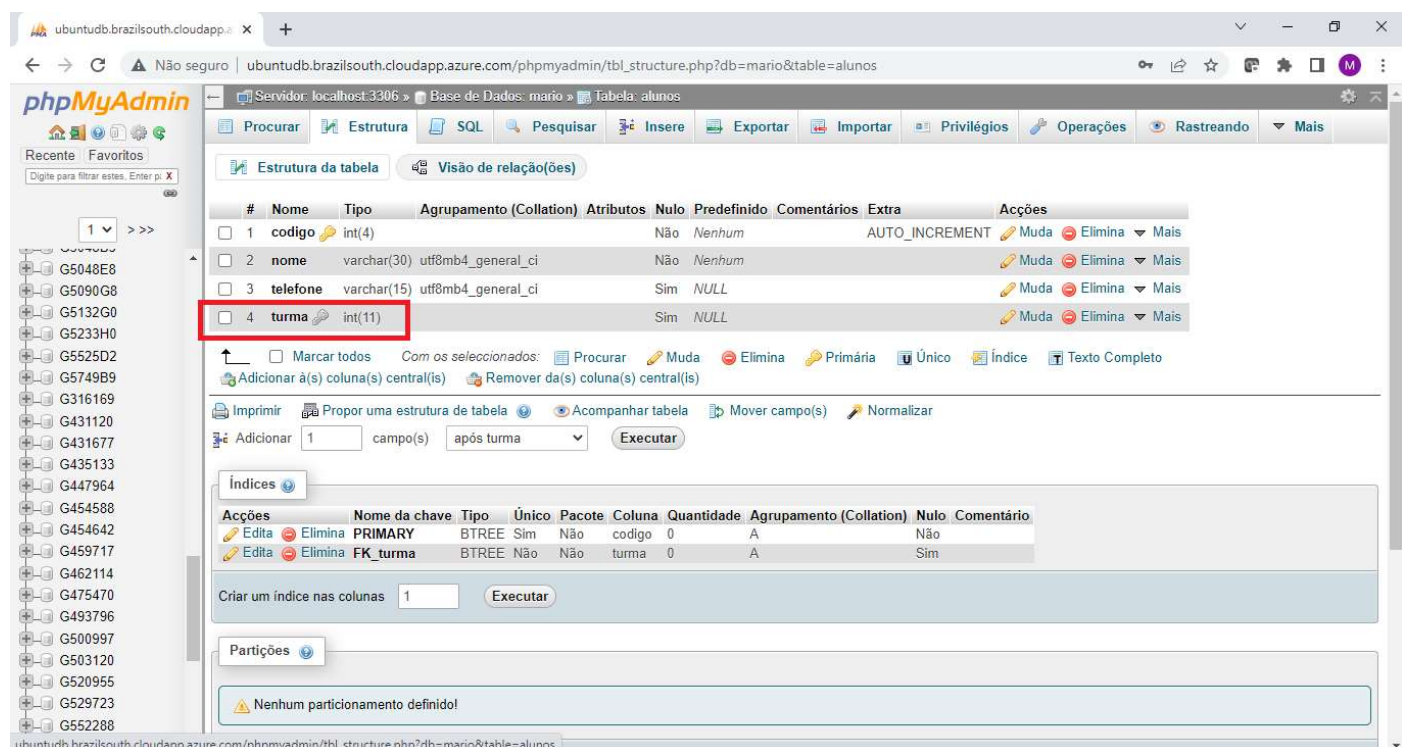
```
ALTER TABLE alunos ADD CONSTRAINT FK_turna FOREIGN KEY (turna) REFERENCES turma(codigo);
```

The interface also includes a sidebar with a list of tables, a top navigation bar, and a right sidebar with a 'Mais' button.

Com a tabela alunos selecionada, verifique via opção “Estrutura” se a chave foi criada:



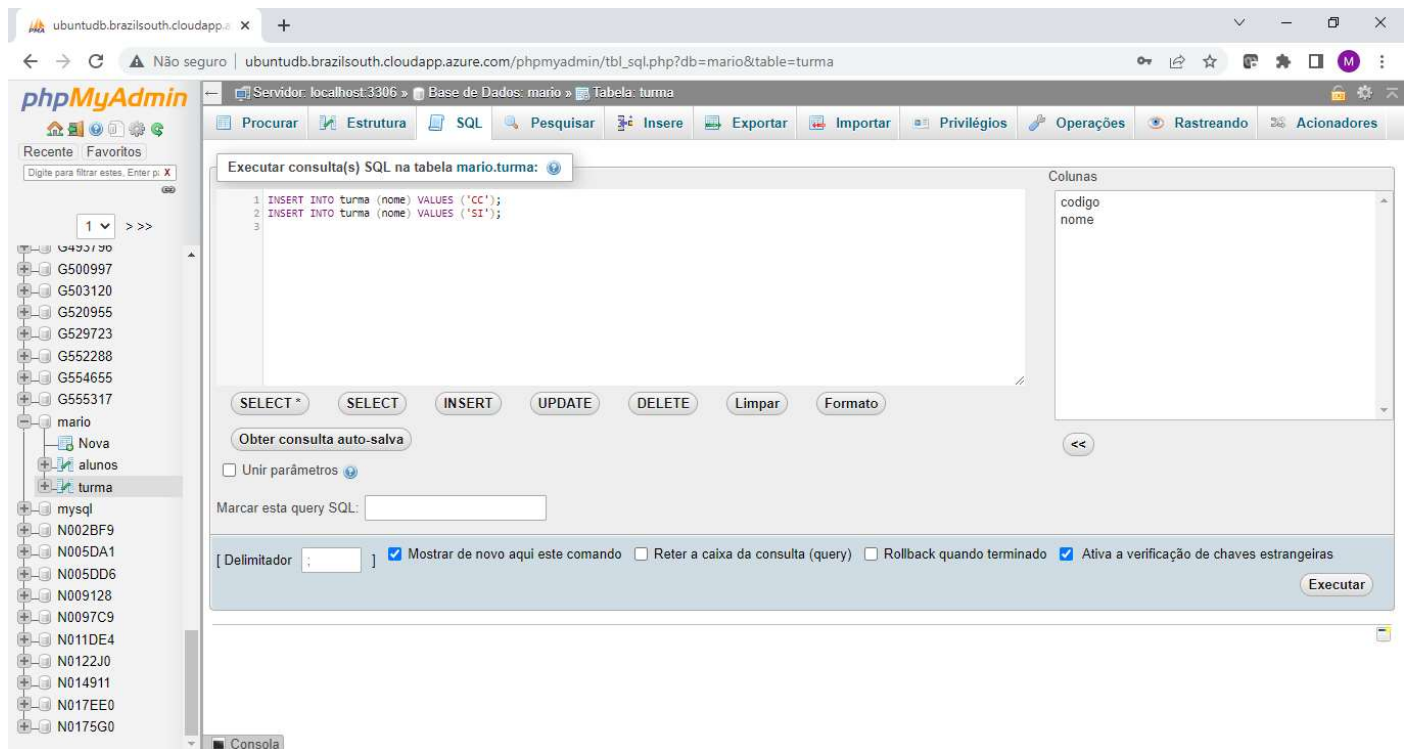
Veja que a chave foi associado ao atributo turma conforme nossa solicitação:



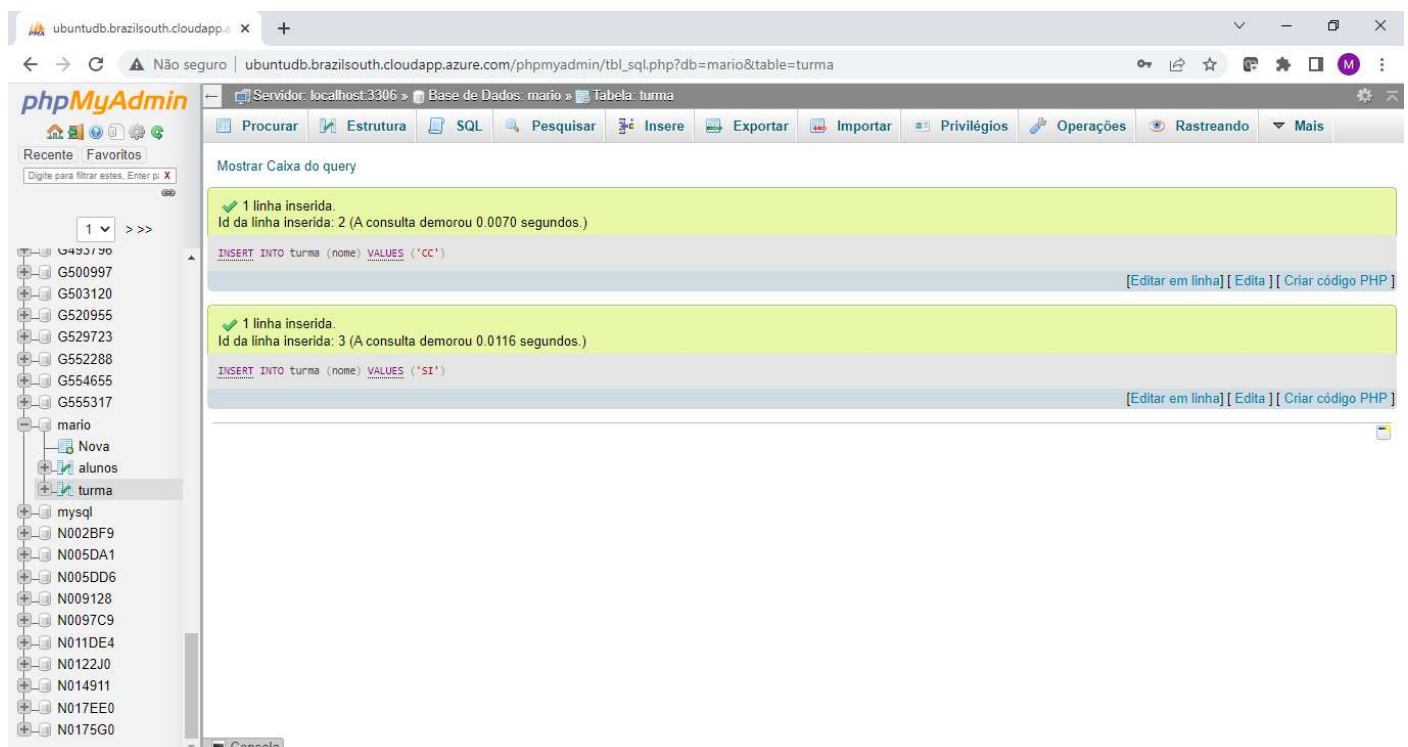
Agora vamos inserir dois registros na tabela através do comando INSERT; como citado anteriormente, este comando pode ter variações de sintaxe, porém, o fundamento lógico é basicamente o mesmo.

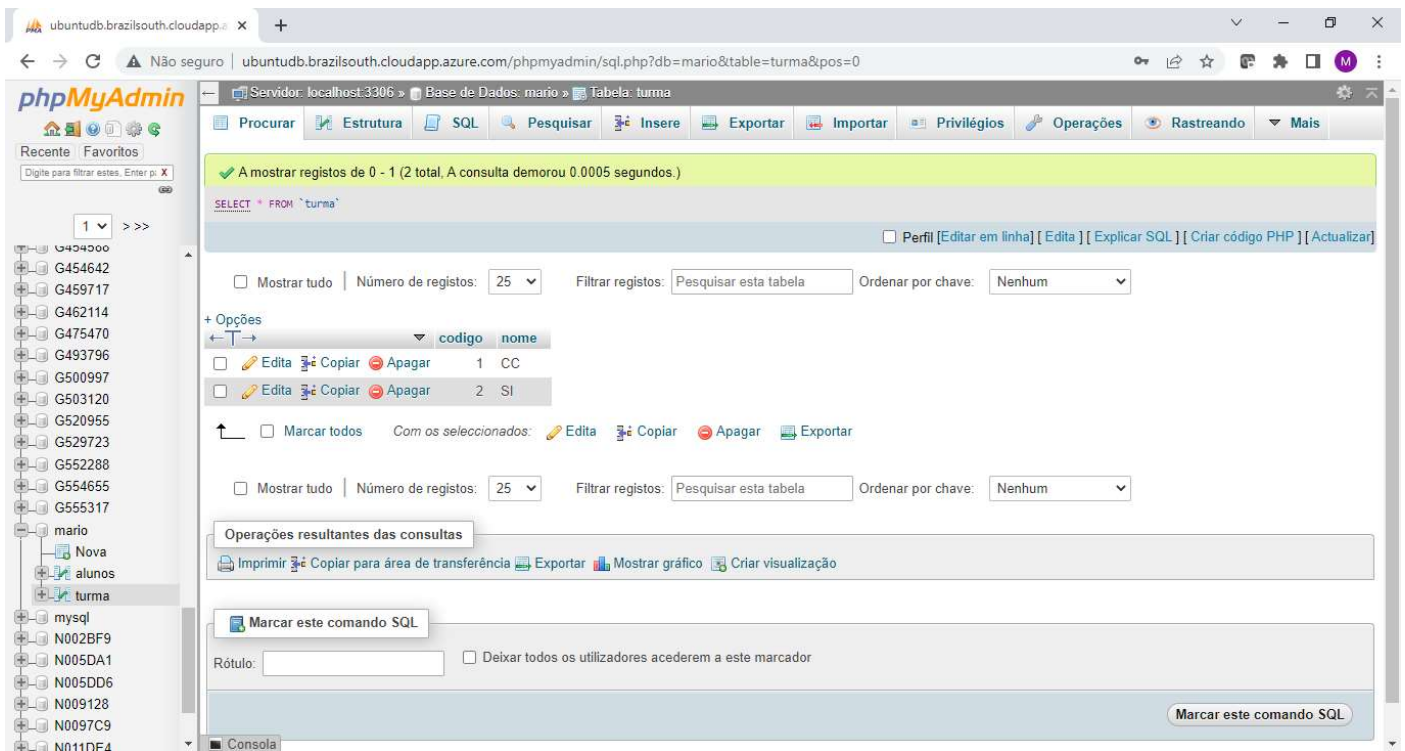
Como as tabelas alunos e turma estão relacionadas, é necessário efetuar primeiro a inserção na tabela que será referenciada:

```
INSERT INTO turma (nome) VALUES ('CC');
INSERT INTO turma (nome) VALUES ('SI');
```



Segue o resultado:





The screenshot shows the phpMyAdmin interface for a database named 'mario'. The 'turma' table is selected, and its structure and data are displayed. The table has two columns: 'codigo' and 'nome'. The data shows two records: '1' with 'CC' and '2' with 'SI'.

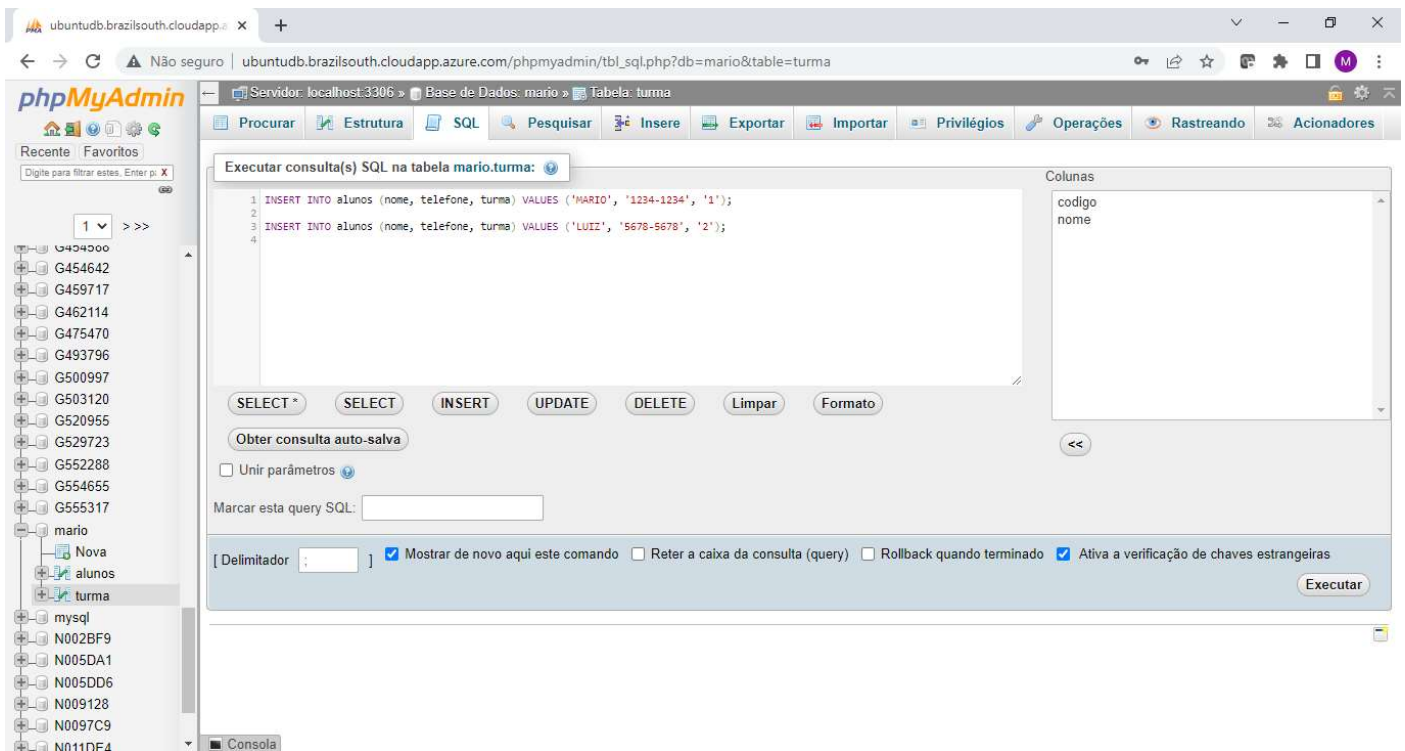
Agora podemos inserir a informação abaixo na tabela alunos:

```
INSERT INTO alunos (nome, telefone, turma) VALUES ('MARIO', '1234-1234', '1');
INSERT INTO alunos (nome, telefone, turma) VALUES ('LUIZ', '5678-5678', '2');
```

A sintaxe do comando precisa:

- Declaração da tabela alvo, no caso, “alunos”;

- Declaração dos campos que irão receber os dados; veja que o campo “codigo”, configurado como Auto Incremento, não é declarado, visto que o próprio SGBD se encarrega de associar o próximo valor ao novo registro que esta sendo inserido;

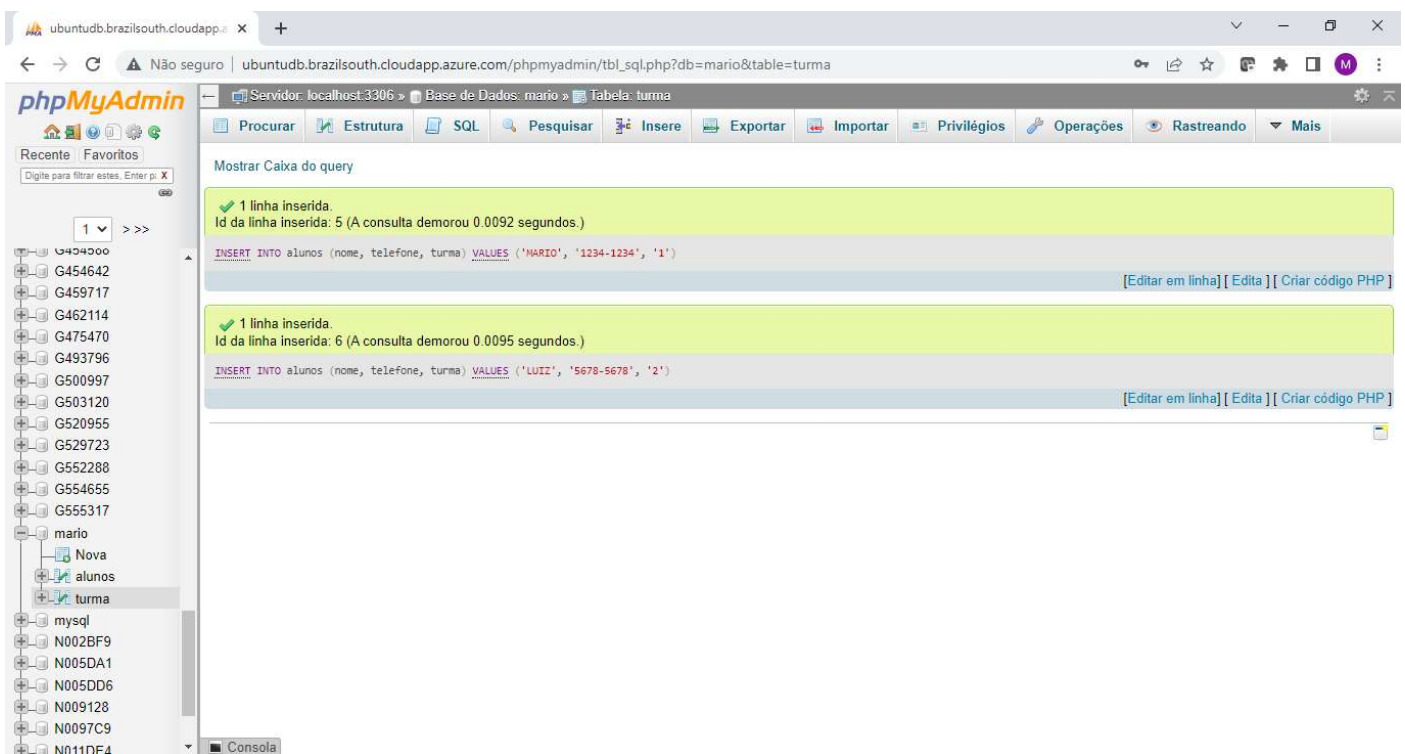


The screenshot shows the phpMyAdmin interface with the 'alunos' table selected in the 'mario' database. The SQL query editor contains the following code:

```
1 INSERT INTO alunos (nome, telefone, turma) VALUES ('MARIO', '1234-1234', '1');
2
3 INSERT INTO alunos (nome, telefone, turma) VALUES ('LUIZ', '5678-5678', '2');
4
```

The interface includes a sidebar with a tree view of databases and tables, a top navigation bar with various tools (Procurar, Estrutura, SQL, etc.), and a main area for executing queries. The 'Executar consulta(s) SQL na tabela mario.turma:' section is active, showing the query and execution options.

Abaixo o resultado esperado:

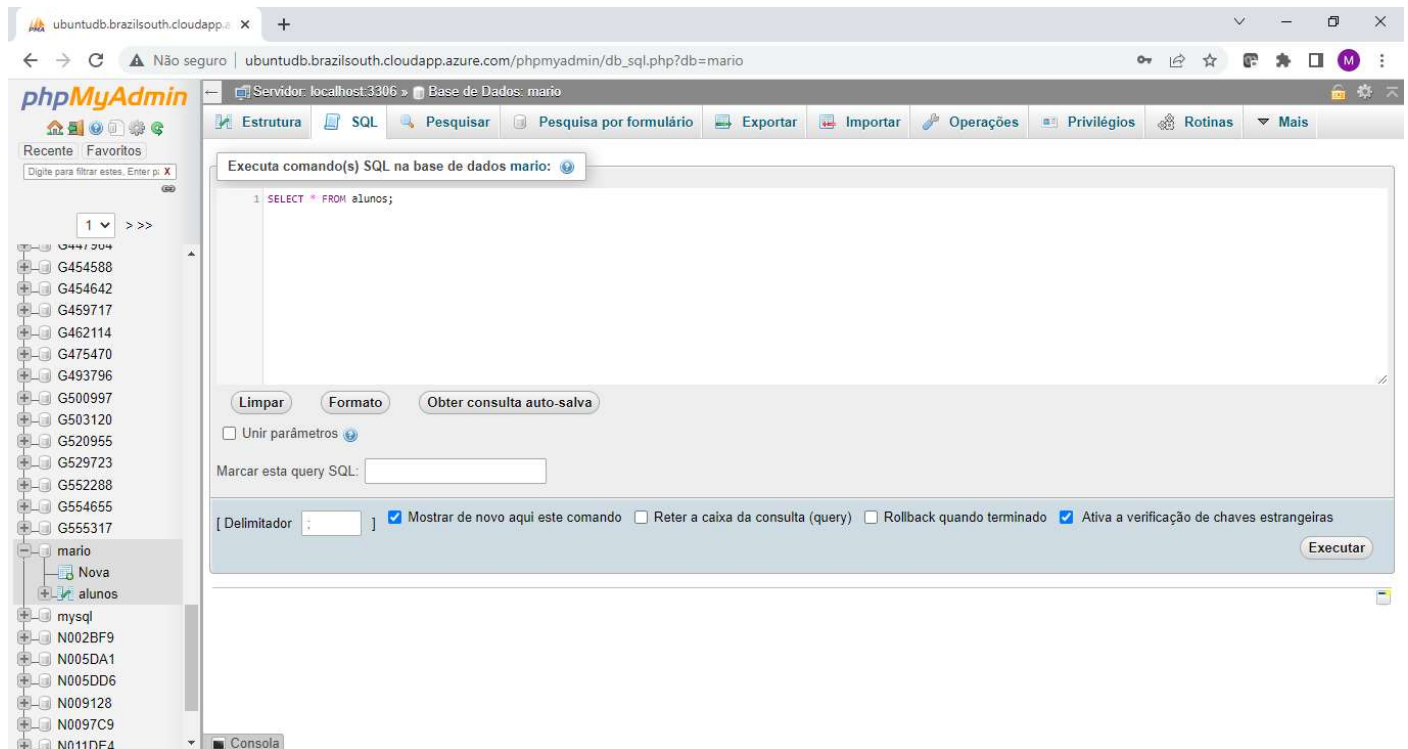


The screenshot shows the phpMyAdmin interface displaying the results of the SQL queries. The 'Mostrar Caixa do query' section shows two successful insertions:

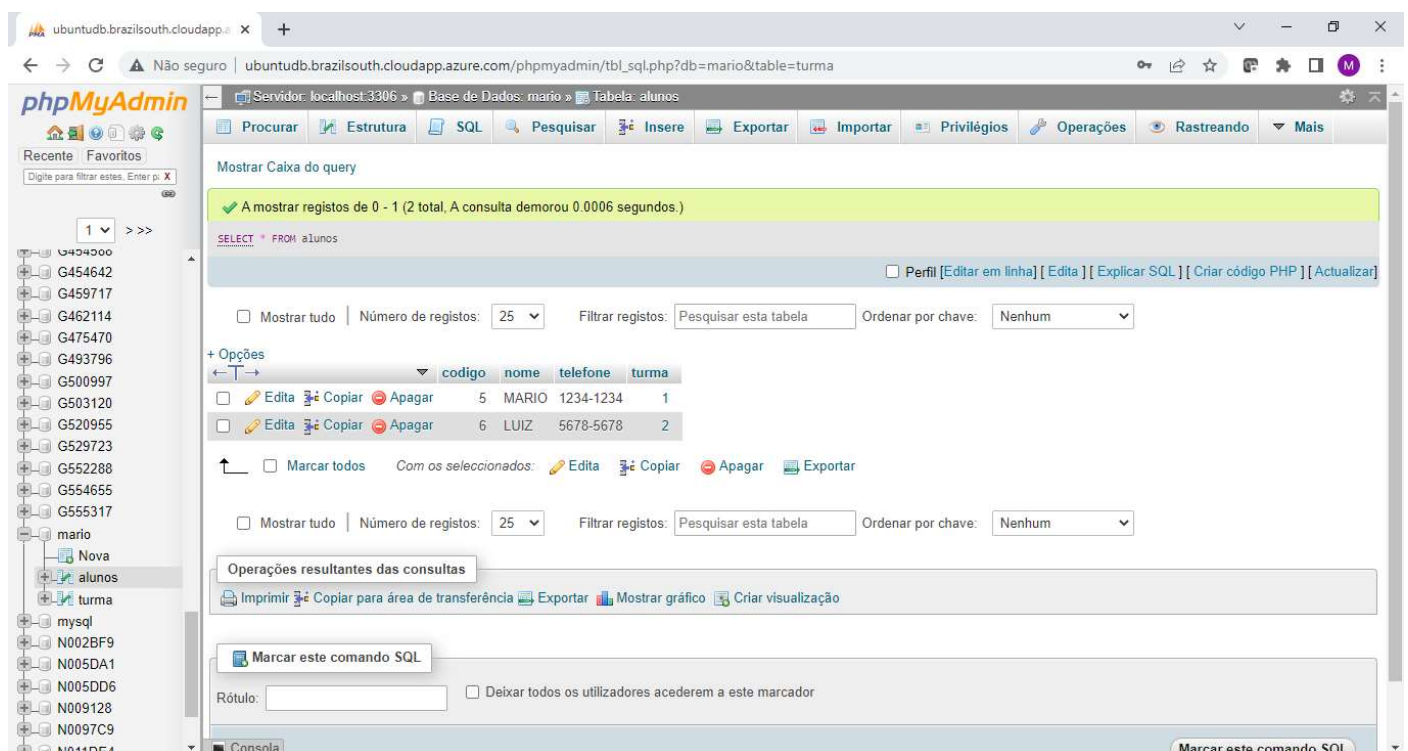
- 1 linha inserida. Id da linha inserida: 5 (A consulta demorou 0.0092 segundos.)
- 1 linha inserida. Id da linha inserida: 6 (A consulta demorou 0.0095 segundos.)

Below each success message, the corresponding SQL query is displayed, and there are links to 'Editar em linha', 'Edita', and 'Criar código PHP'.

Podemos verificar o conteúdo da tabela através do comando SELECT:



O retorno deve ser similar a tela abaixo:



Exercícios

1- Criar a tabela funcionarios com os seguintes atributos:

- Nome
- Sobrenome
- Data de Nascimento
- **Cargo**
- **Departamento**

2- Inserir as informações abaixo na tabela criada anteriormente:

- Netanias Alves Mostarda, nascido em 19/03/2000, Operador de Empilhadeira, atua no departamento de Logística
- Alfredo dos Santos e Silva, nascido em 20/11/1976, Contador, atua no departamento Financeiro
- Adalberto Oliveira Gastardo, nascido em 05/02/2000, Vendedor, atua no departamento Comercial