A dark blue L-shaped frame is positioned on the left and bottom edges of the slide, framing the central text.

# LIGAÇÃO A ESTRUTURAS DE DADOS COM PHP E MYSQL

# Introdução

- Perceber como o PHP interage com bases de dados MySQL
- Aprender a conectar a uma base de dados MySQL utilizando o PHP
- Executar operações básicas:
  - Inserir
  - Atualizar
  - Eliminar
  - Consultar

•

# Contextualização

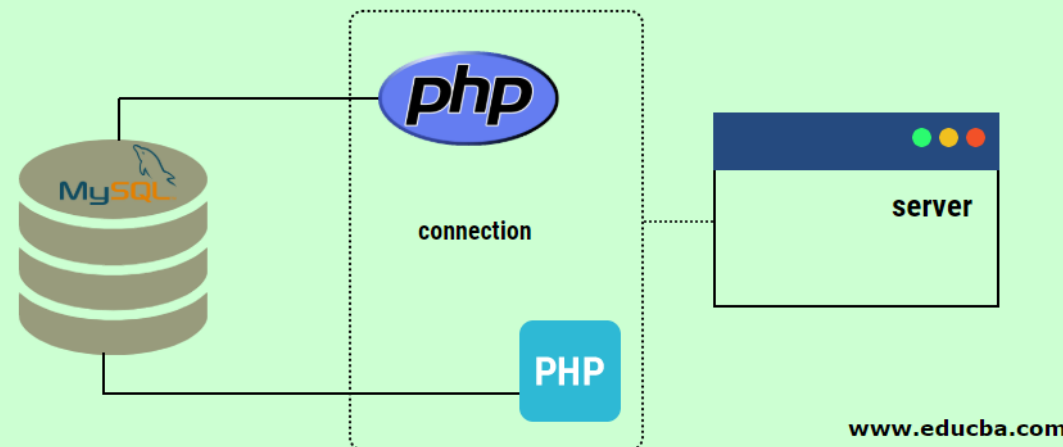
## ■ PHP:

- *Linguagem de script do lado do servidor*
- *Amplamente utilizada para desenvolvimento web*
- *Suporte extenso*
- *Fácil integração com bases de dados*

## ■ MySQL:

- *Sistema de gestão de base de dados relacional (SGBDR)*
- *Popular, robusto e eficiente*
- *Suporte a transações, integridade referencial e segurança*

## PHP Database Connection

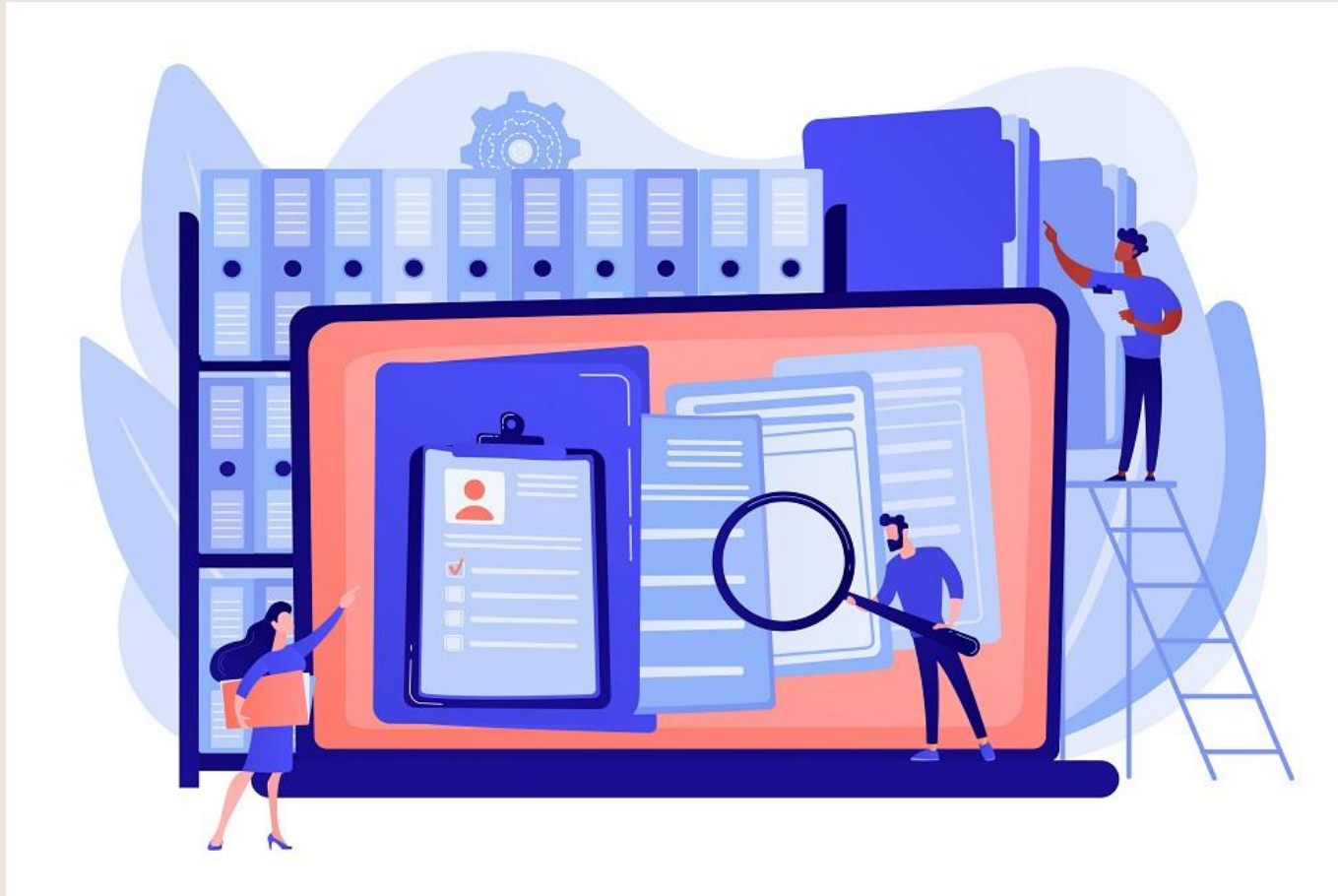


- PHP é uma das linguagens de script mais populares para desenvolvimento web, devido à sua flexibilidade e facilidade de uso.
- MySQL, por outro lado, é um sistema de gestão de bases de dados muito eficiente e amplamente utilizado.
- Juntos, formam uma poderosa combinação para desenvolvimento de aplicações web dinâmicas.
- A integração entre PHP e MySQL é fácil, permitindo criar sites interativos que podem armazenar e recuperar dados eficientemente

# Ações e Comandos de Base de Dados

## ■ Ações:

1. *Conectar à Base de Dados*
2. *Inserir Dados*
3. *Consultar Dados*
4. *Atualizar Dados*
5. *Eliminar Dados*



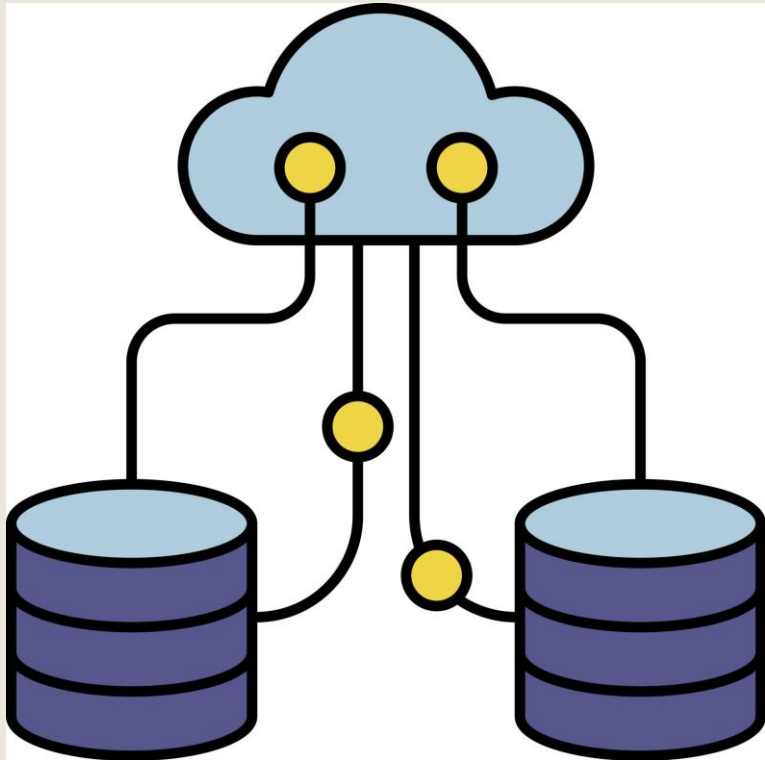
# Ações e Comandos de Base de Dados



## Comandos de Base de Dados:

- **CREATE DATABASE:** Cria uma nova base de dados
- **CREATE TABLE:** Cria uma nova tabela
- **INSERT INTO:** Insere novos dados
- **SELECT:** Consulta dados
- **UPDATE:** Atualiza dados existentes
- **DELETE:** Elimina dados

# Conectando ao MySQL com PHP



- O primeiro passo para interagir com a base de dados é estabelecer uma conexão.
- No exemplo seguinte é utilizada a função `new mysqli` para criar uma nova conexão com o servidor MySQL.
- Precisamos indicar:
  - o nome do servidor (localhost para um servidor local)
  - o nome de utilizador
  - a password e o nome da base de dados.
- Após a indicação dos parâmetros é verificado se a conexão foi bem-sucedida utilizando o comando `connect_error`.
  - Se houver um erro, o script é terminado com a função `die` e exibida a mensagem de erro
  - Caso contrário, confirma-se a conexão bem-sucedida.
- Este é o ponto de partida para todas as operações subsequentes na base de dados.

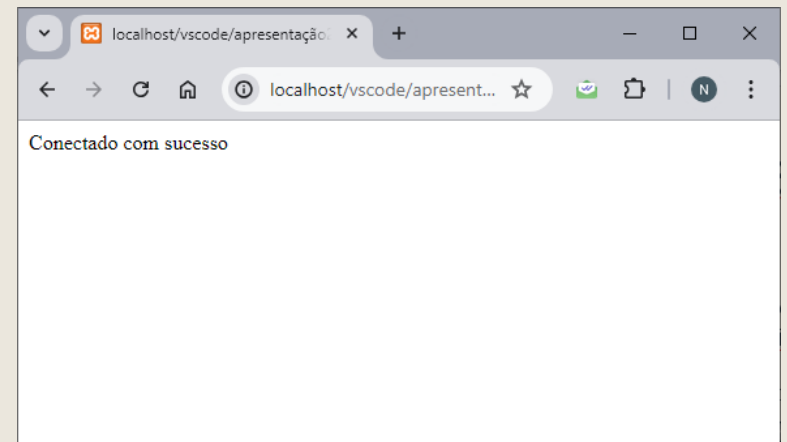
# Conectando-se ao MySQL com PHP

## Exemplo 1

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

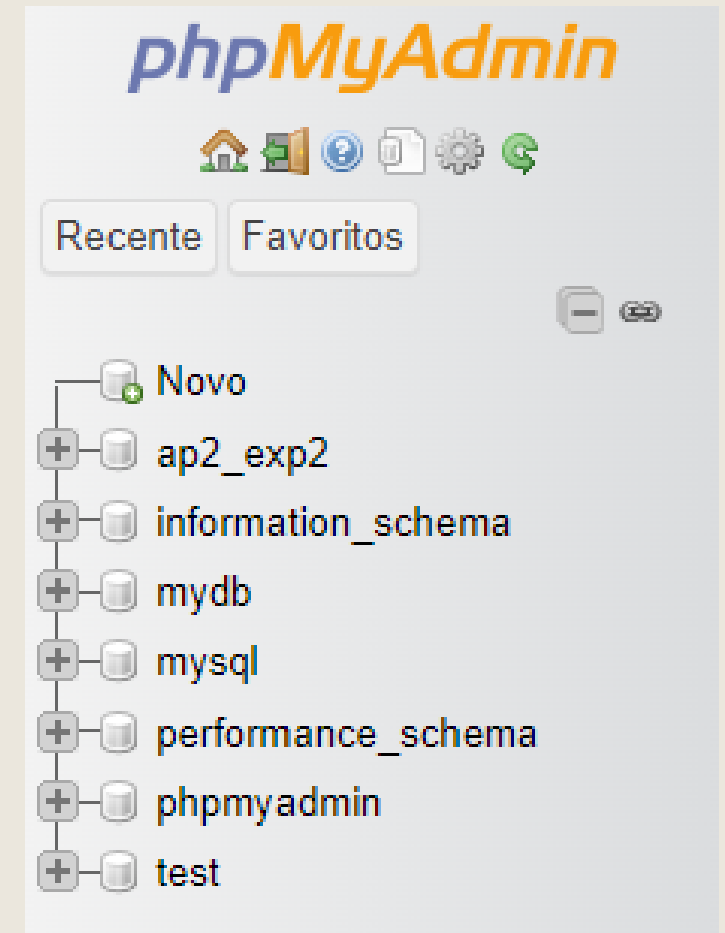
// Criar conexão
$conn = new mysqli($servername,
```

```
// Verificar conexão
if ($conn->connect_error) {
    die("Falha na conexão: " . $conn->connect_error);
}
echo "Conectado com sucesso";
?>
```



# Criação de Base de Dados em PHP

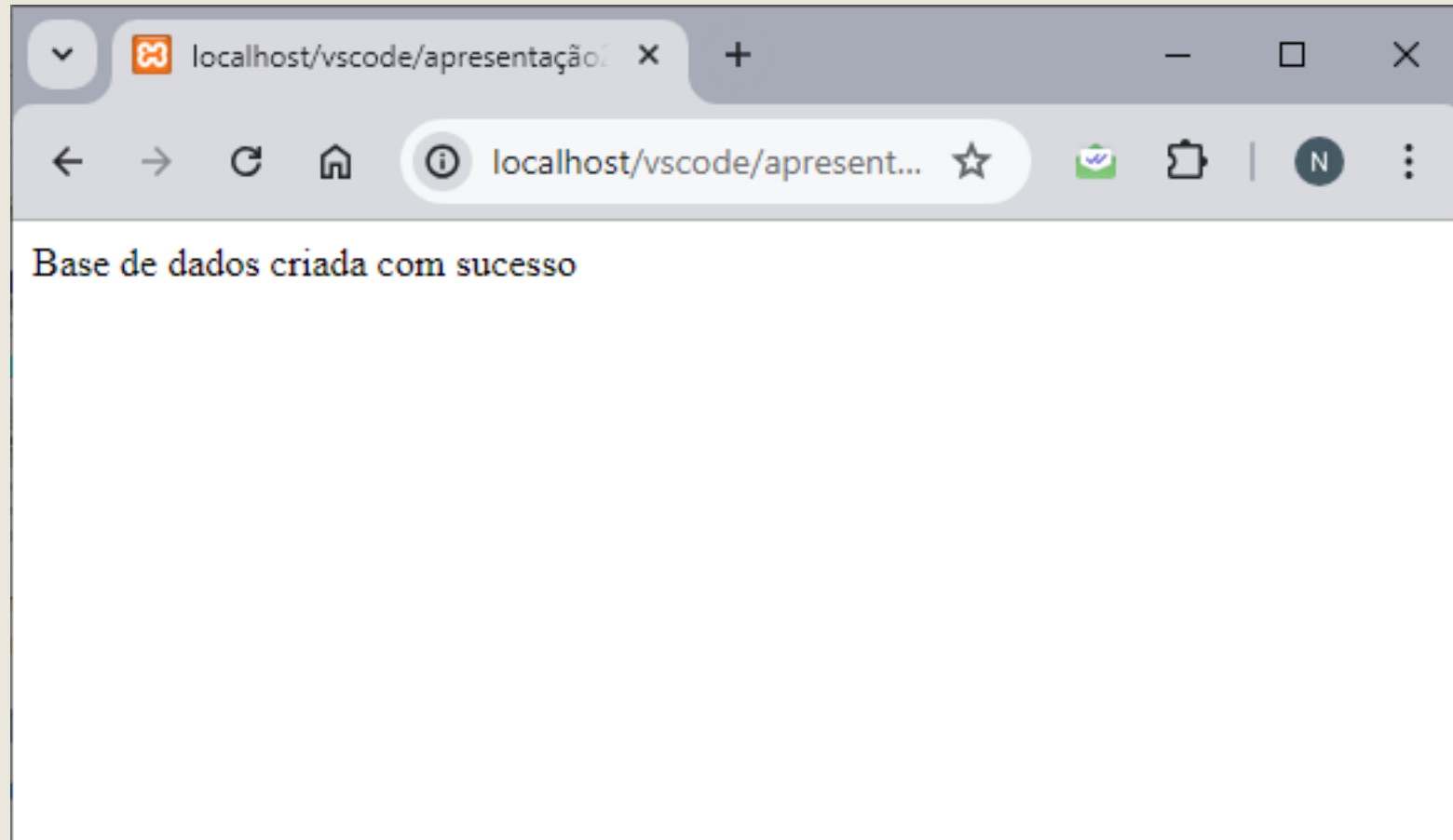
- Para criar uma nova base de dados através de PHP, seguimos um processo semelhante ao da conexão.
- Primeiro, estabelecemos a conexão com o servidor MySQL sem especificar uma base de dados.
- De seguida, usamos o comando CREATE DATABASE dentro da função query para criar a base de dados.
- Verificamos se a operação foi bem-sucedida e exibimos uma mensagem correspondente.
- Este script cria a base de dados myDB no servidor MySQL.





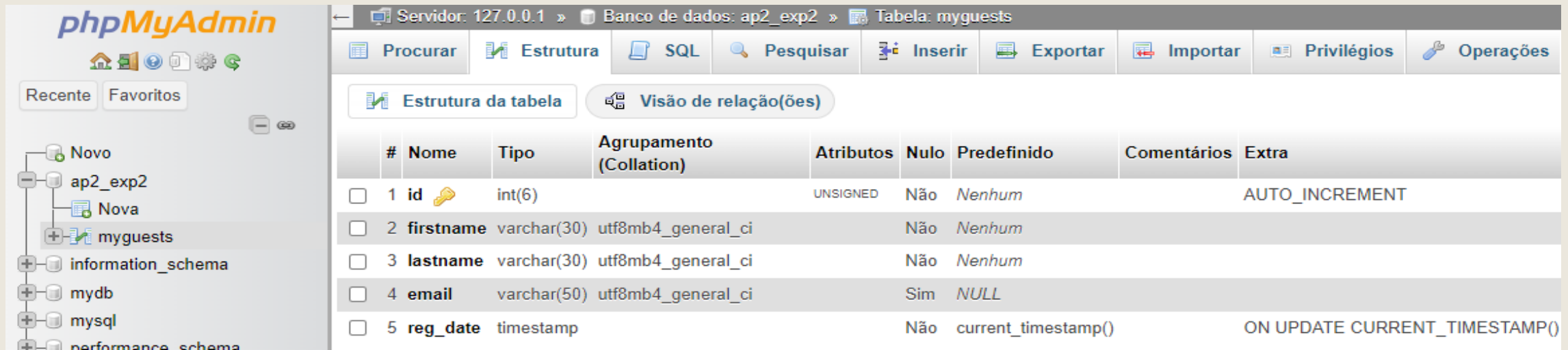
# Criação de Base de Dados em PHP

## Exemplo 2



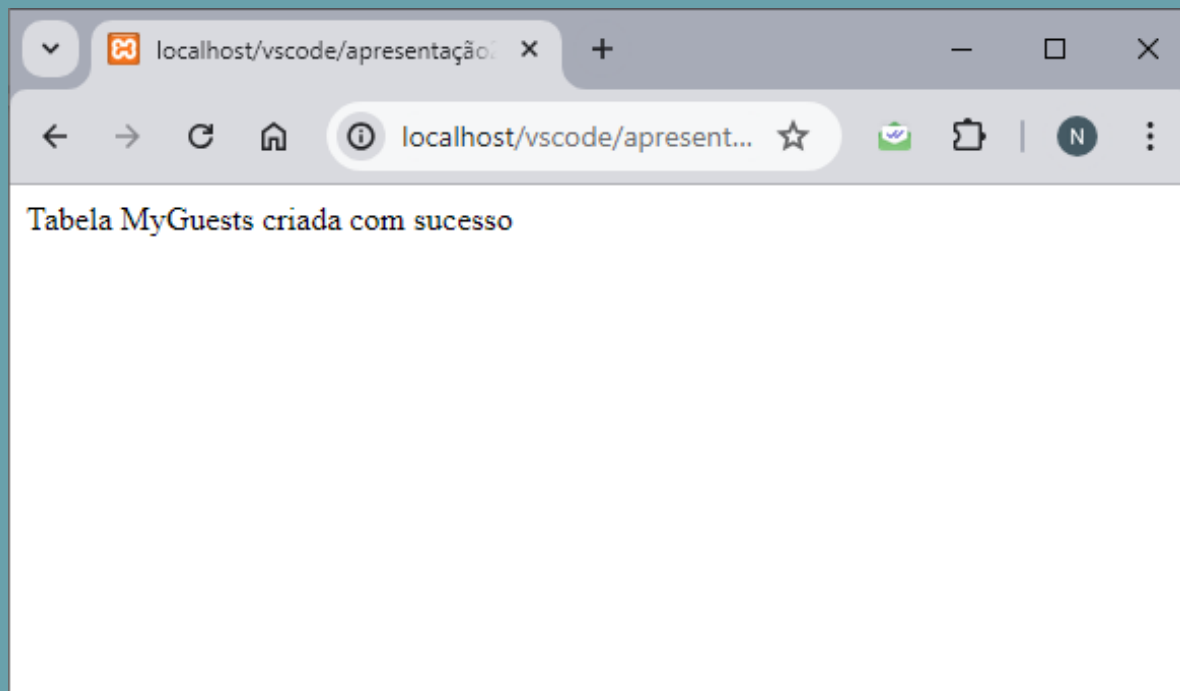
# Criação de Tabela em PHP

- Depois de criar a base de dados, o próximo passo é criar uma tabela para armazenar os dados.
- No exemplo, conectamo-nos à base de dados “ap2\_exp2” e utilizamos o comando CREATE TABLE para definir a estrutura da tabela MyGuests
- A tabela inclui colunas para id, firstname, lastname, email e reg\_date.
- O campo id é uma chave primária auto-incrementável
- Os campos firstname e lastname são obrigatórios
- Executamos a consulta e verificamos se a criação da tabela foi bem-sucedida.



The screenshot shows the phpMyAdmin interface. On the left, the database 'ap2\_exp2' is selected, and the table 'myguests' is highlighted. The main panel displays the 'Estrutura da tabela' (Table Structure) view for 'myguests'. The table has 5 columns: 'id' (int(6), UNSIGNED, NOT NULL, AUTO\_INCREMENT), 'firstname' (varchar(30), utf8mb4\_general\_ci, NOT NULL), 'lastname' (varchar(30), utf8mb4\_general\_ci, NOT NULL), 'email' (varchar(50), utf8mb4\_general\_ci, NULL), and 'reg\_date' (timestamp, NOT NULL, current\_timestamp(), ON UPDATE CURRENT\_TIMESTAMP()).

#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Comentários	Extra
<input type="checkbox"/>	1 id	int(6)		UNSIGNED	Não	Nenhum		AUTO_INCREMENT
<input type="checkbox"/>	2 firstname	varchar(30)	utf8mb4_general_ci		Não	Nenhum		
<input type="checkbox"/>	3 lastname	varchar(30)	utf8mb4_general_ci		Não	Nenhum		
<input type="checkbox"/>	4 email	varchar(50)	utf8mb4_general_ci		Sim	NULL		
<input type="checkbox"/>	5 reg_date	timestamp			Não	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()



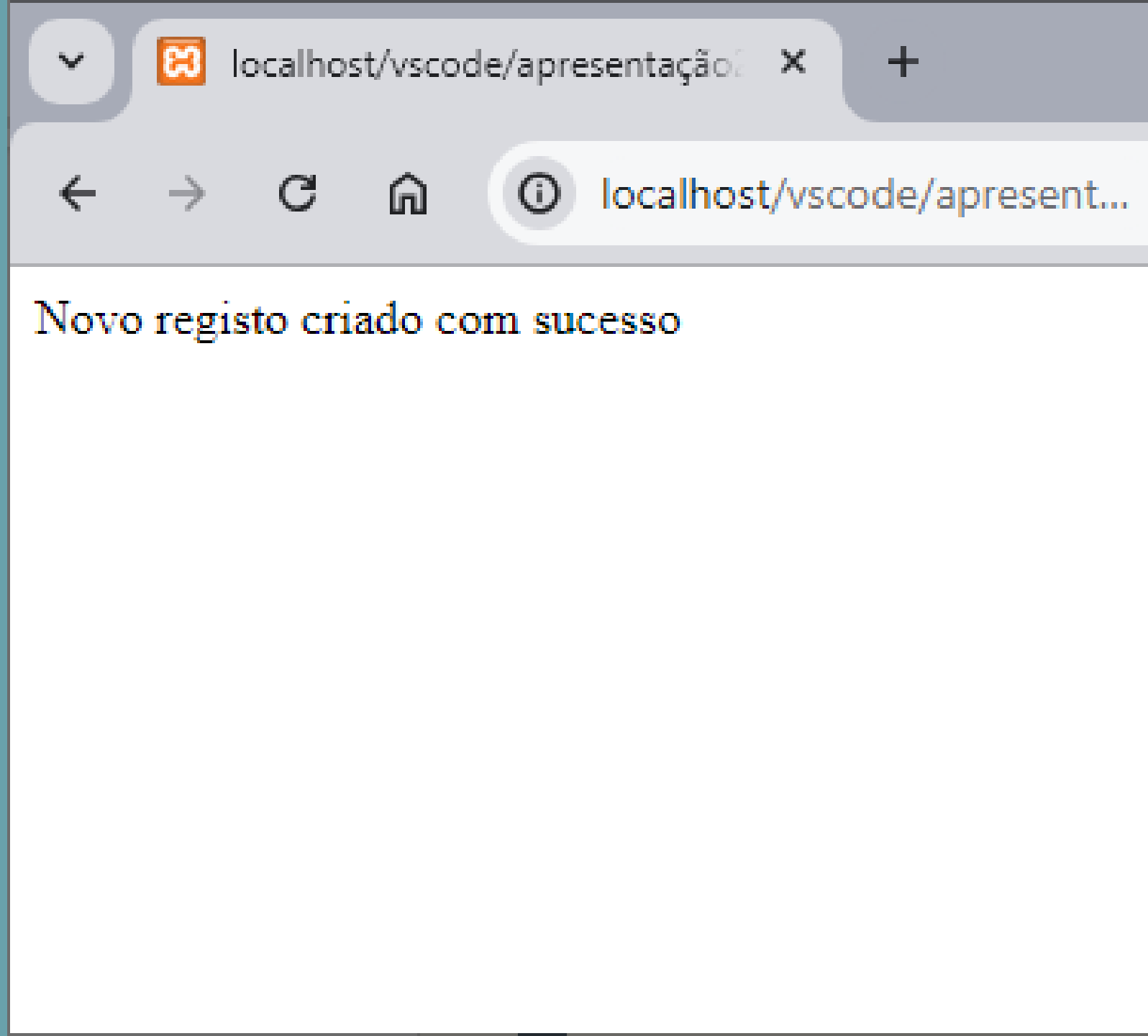
# CRIAÇÃO DE TABELA EM PHP EXEMPLO 3

# Inserção de Dados com PHP

- Para inserir dados numa tabela, usamos o comando INSERT INTO
- No exemplo:
  - Conectamo-nos à base de dados “ap2\_exp2
  - Inserimos um novo registo na tabela MyGuests
  - Os valores Nuno, Godinho e nuno@exemplo.pt são inseridos nas colunas
    - firstname
    - lastname
    - email
  - Verificamos se a inserção foi bem-sucedida e exibimos uma mensagem apropriada.

</

# INSERÇÃO DE DADOS COM PHP EXEMPLO 4



# Consulta de Dados com PHP

- Para consultar dados de uma tabela, utilizamos o comando SELECT.
- No exemplo:
  - Conectamo-nos à base de dados “ap2\_exp2
  - Seleccionamos as colunas *id*, *firstname* e *lastname* da tabela *MyGuests*
  - Executamos a consulta e verificamos se há resultados
  - Se houver, usamos um loop *while* para percorrer cada linha do resultado e exibir os dados
  - Este script mostra como recuperar e apresentar dados de uma tabela.

</

# Consulta de Dados com PHP

## Exemplo 5

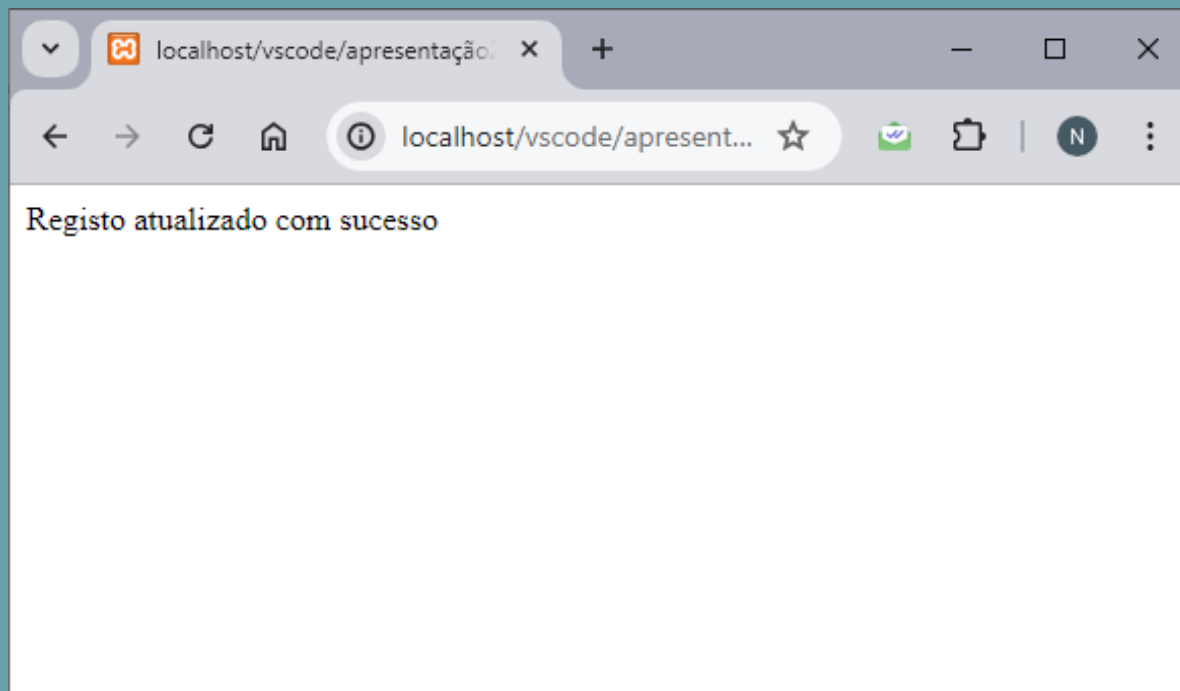


# Atualização de Dados com PHP

- Para atualizar dados existentes numa tabela, usamos o comando UPDATE
- No exemplo:
  - Conectamo-nos à base de dados “ap2\_exp2”
  - Atualizamos o campo email para godinho@exemplo.pt onde o id é 1.
- Utilizar a cláusula WHERE é crucial para especificar qual registo deve ser atualizado
- Após executar a consulta, verificamos se a operação foi bem-sucedida e exibimos uma mensagem correspondente
- Este script demonstra como modificar dados em uma tabela.

</





# ATUALIZAÇÃO DE DADOS COM PHP EXEMPLO 6

# Eliminação de Dados com PHP

- Para eliminar dados de uma tabela, utilizamos o comando DELETE
- No exemplo:
  - *Conectamo-nos à base de dados ap2\_exp2*
  - *Eliminamos o registo onde o id é 1*
- Assim como na atualização, a cláusula WHERE é essencial para especificar qual registo deve ser eliminado
- Executamos a consulta e verificamos se a operação foi bem-sucedida, exibindo uma mensagem apropriada
- Este script demonstra como remover dados de uma tabela

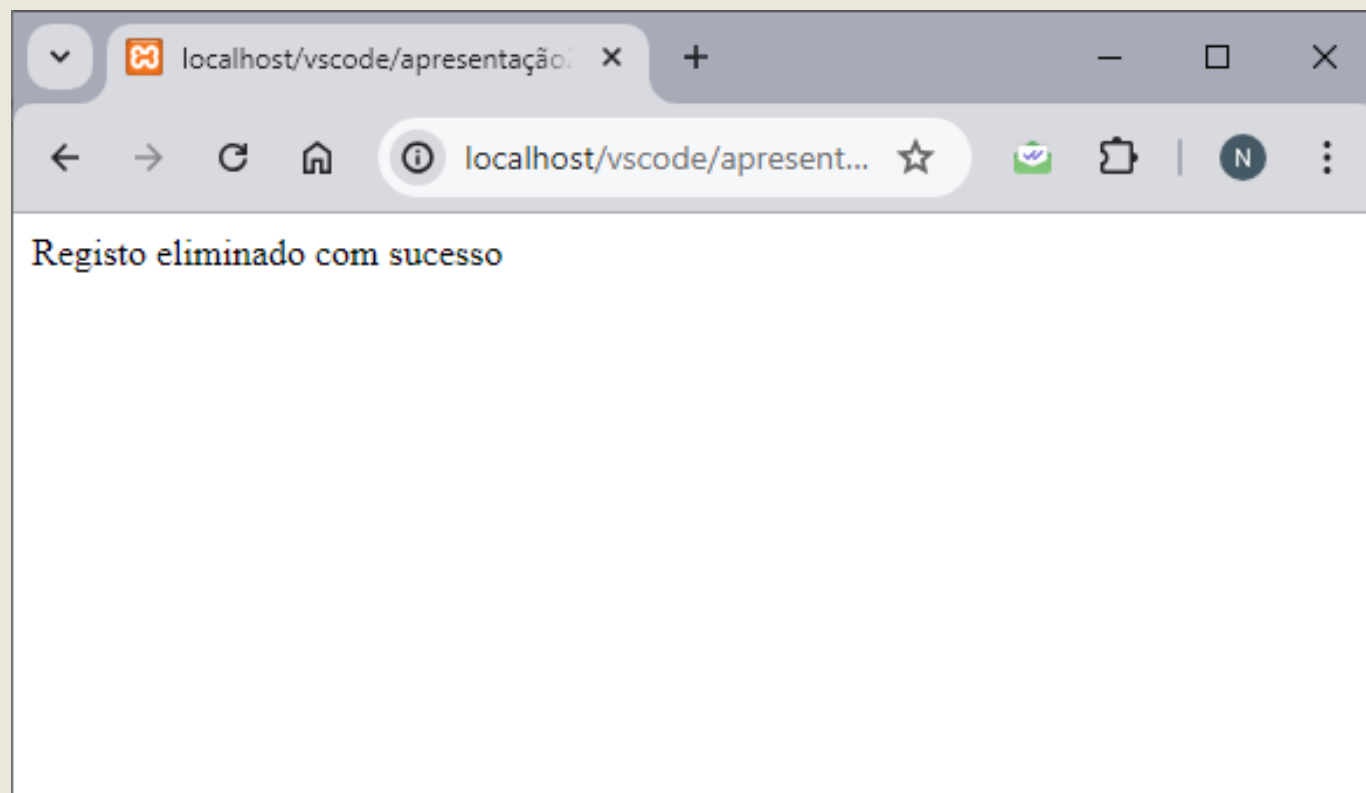
✓ MySQL não retornou nenhum registo. (A consulta demorou 0,0005 segundos.)

```
SELECT * FROM `myguests`
```

☐ Perfil [ [Editar em linha](#) ] [ [Editar](#) ] [ [Explicar SQL](#) ] [ [Criar código PHP](#) ] [ [Actualizar](#) ]

id	firstname	lastname	email	reg_date
----	-----------	----------	-------	----------

# Eliminação de Dados com PHP Exemplo 7



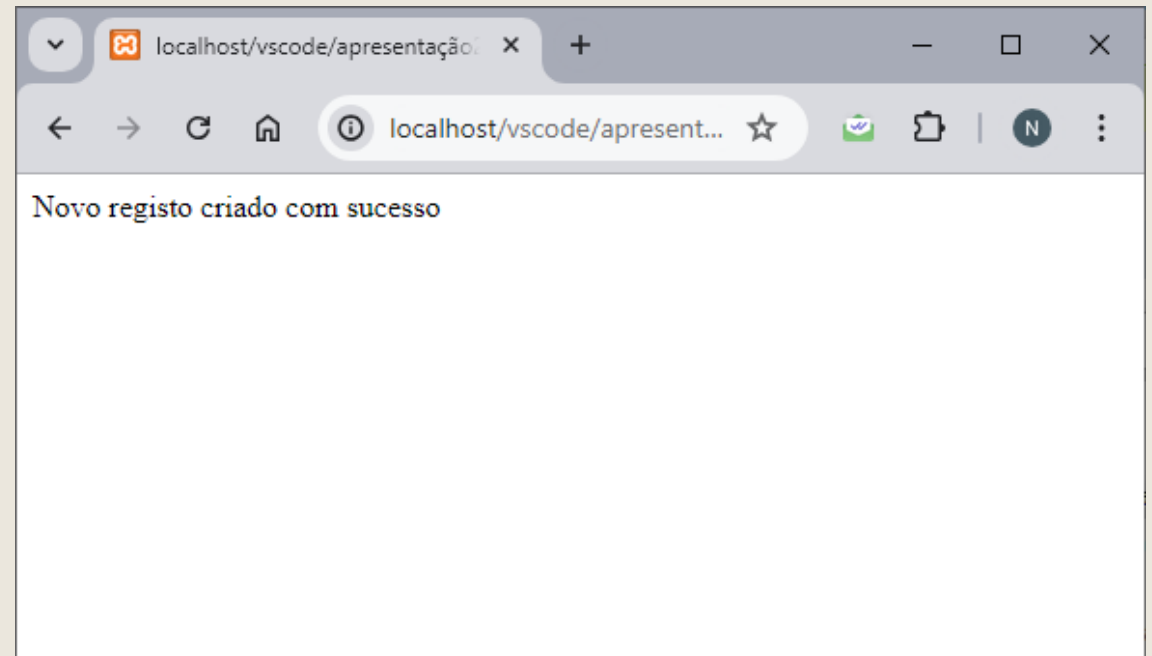
# Boas Práticas

- Utilizar Prepared Statements para evitar SQL Injection – utilizar “Prepared Statements” é uma forma eficaz de prevenir ataques de SQL Injection, que ocorrem quando um invasor insere código malicioso através dos campos de entrada de dados
- No exemplo seguinte são utilizados:
  - *prepare* - para criar uma declaração preparada
  - *bind\_param* - para associar variáveis aos parâmetros da consulta
- Isto assegura que os dados do utilizador sejam tratados corretamente e não possam alterar a estrutura da consulta SQL
- Fechar sempre a conexão à base de dados


</

# Boas Práticas

## Exemplo 8




## ▼ Exemplo 9

 criar\_bd\_tabela.php

 db\_connect.php


 delete.php


 index.php

 insert\_form.html

 insert.php

 select.php

 update\_form.php

 update.php

## Exemplo Completo

- Base de dados

### Exemplo 9

#### ■ Site com:

- *Inserção de dados*
- *Consulta*
- *Atualização*
- *Eliminação*

# Exemplo Completo

## – Base de dados

## Exemplo 9

