

FIAP

PHP

Um banco de dados é uma ferramenta para coletar e organizar informações. Bancos de dados podem armazenar informações sobre pessoas, produtos, pedidos ou qualquer outra coisa. Iniciar muitos bancos de dados como uma lista em uma planilha ou um programa de processamento de texto.

- Open Source
- Bem documentado
- Entre os mais utilizados na web



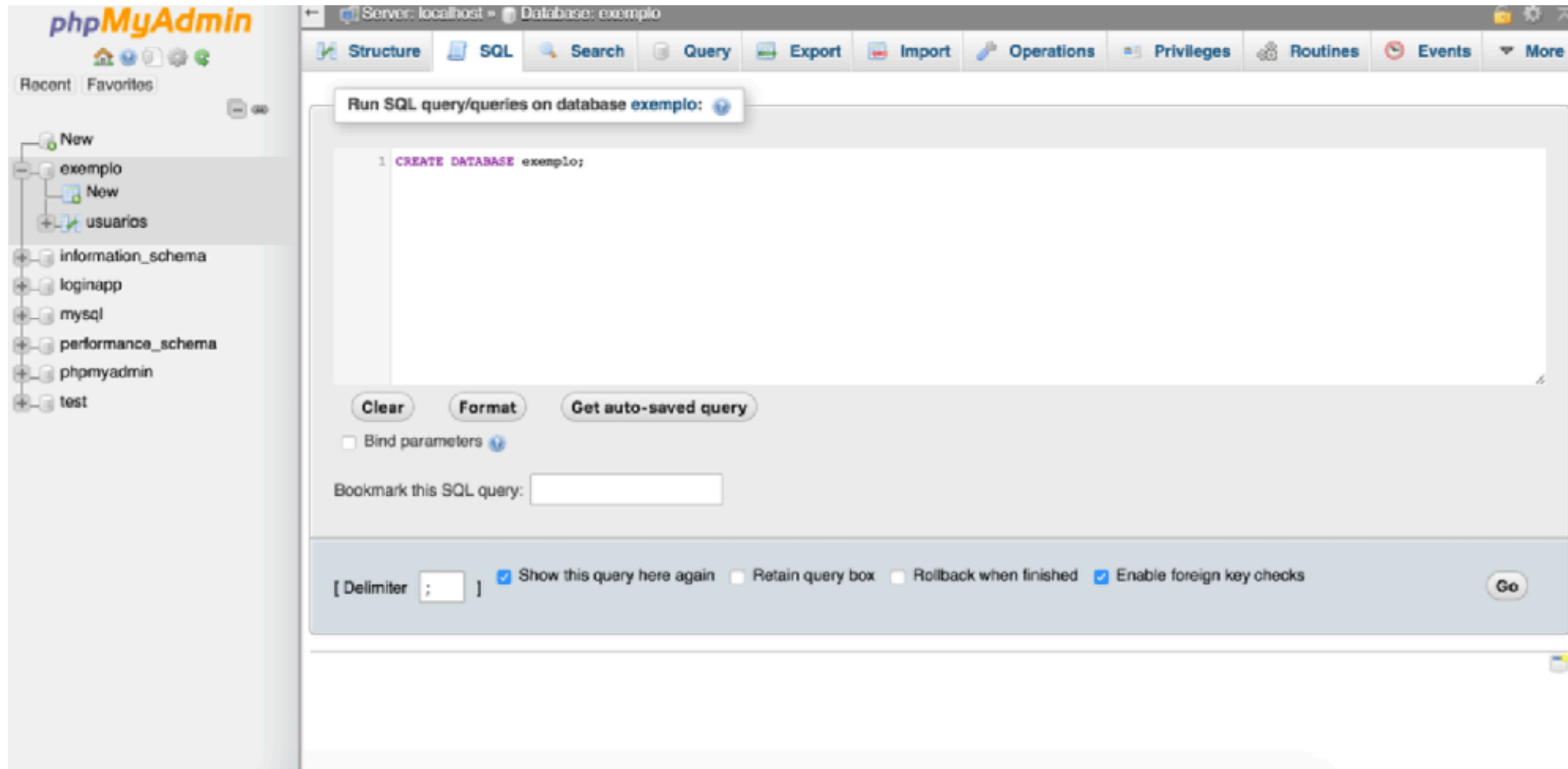
Criando um BDD no PhpMyAdmin

The screenshot shows the phpMyAdmin interface for a local MySQL server. The 'Databases' tab is active. In the left sidebar, the 'New' button is circled in blue and labeled '1º'. In the main content area, the 'Create database' form is shown with 'loginapp' entered in the name field. The 'Create' button is circled in blue and labeled '2º'. Below the form, a table lists existing databases with their collations and a 'Check privileges' link for each.

Database	Collation	
<input type="checkbox"/> exemplo	latin1_swedish_ci	Check privileges
<input type="checkbox"/> information_schema	utf8_general_ci	Check privileges
<input type="checkbox"/> loginapp	latin1_swedish_ci	Check privileges
<input type="checkbox"/> mysql	latin1_swedish_ci	Check privileges
<input type="checkbox"/> performance_schema	utf8_general_ci	Check privileges
<input type="checkbox"/> phpmyadmin	utf8_bin	Check privileges
<input type="checkbox"/> test	latin1_swedish_ci	Check privileges
Total: 7		

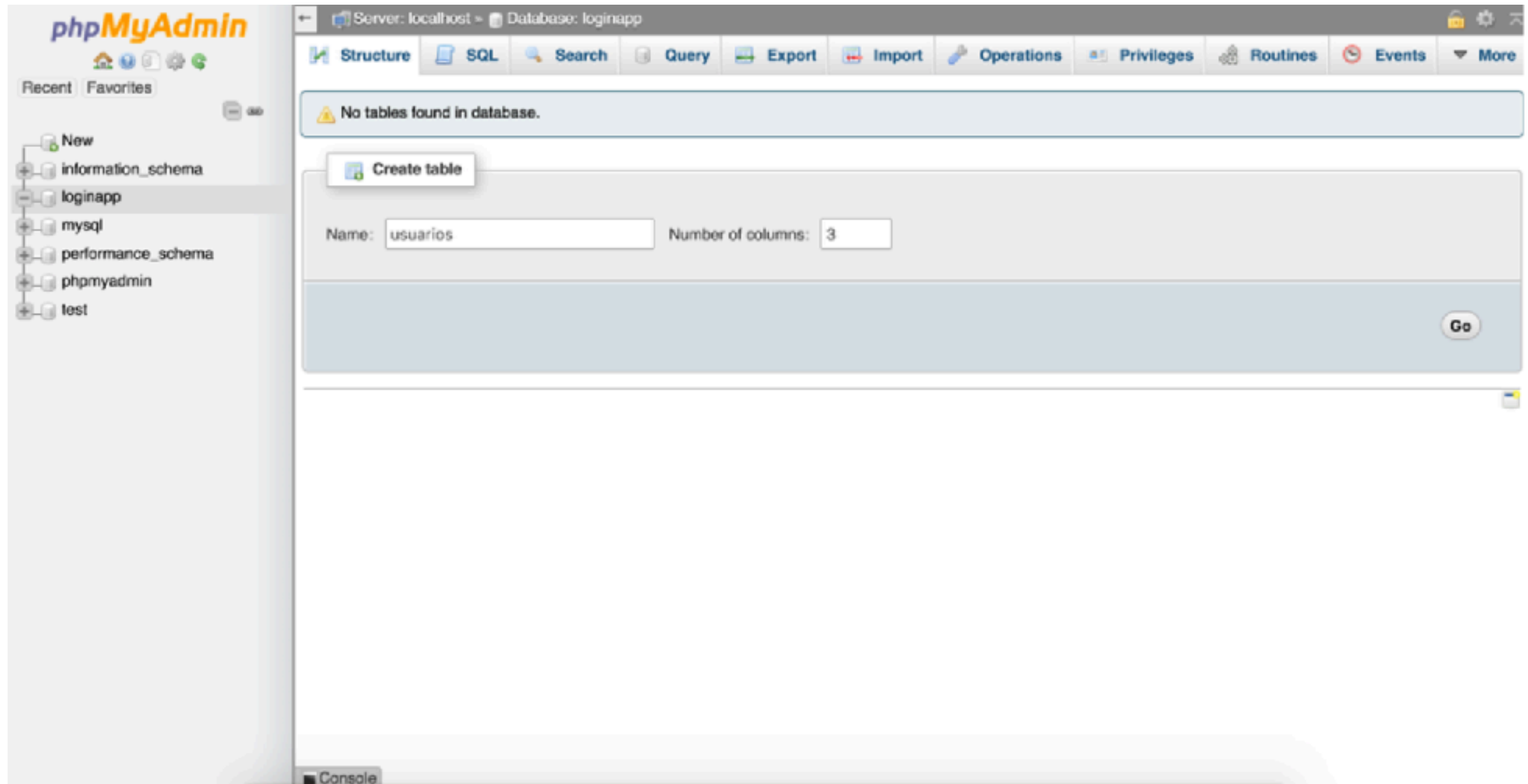
Para criar um novo banco de dados de forma automática é simples, basta clicar em **new**, digitar um nome e por fim clicar em **create**.

Criando um BDD no PhpMyAdmin



Na aba **SQL**, você consegue criar de forma manual inserindo comandos.

Criando uma tabela



Selecione o banco de dados desejado, clique em ***create table***.
Forneça o nome da tabela e a quantidade de colunas.

Criando uma tabela

The screenshot shows the phpMyAdmin interface for creating a table named 'usuarios' in the 'loginapp' database. The table structure is defined with three columns: 'id' (INT, 11), 'username' (VARCHAR, 32), and 'password' (VARCHAR, 32). The 'id' column is set as the primary key (PRIMARY) and has the 'Null' checkbox unchecked. The 'Storage Engine' is set to InnoDB. A blue circle highlights the 'PRIMARY' dropdown for the 'id' column, with an arrow pointing to a text box containing the note: 'Atenção para o **id**. Chave primária e auto incremento foram definidos'.

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A_I	Comments
id	INT	11	None			<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>	
username	VARCHAR	32	None			<input type="checkbox"/>		<input type="checkbox"/>	
password	VARCHAR	32	None			<input type="checkbox"/>		<input type="checkbox"/>	

Table comments:

Collation:

Storage Engine: InnoDB

PARTITION definition:

Preview SQL Save

Neste exemplo com 3 colunas definimos os campos: **id**, **username** e **password** (int, varchar, varchar, respectivamente).

Como conectar o banco no php?

```
1  <?php
2
3  if(isset($_POST['enviar'])){
4      $username = $_POST['username'];
5      $password = $_POST['password'];
6
7      $connection = mysqli_connect('localhost', 'root', '', 'exemplo');
8
9  ?>
```

endereço, usuário, senha, nome do banco

A função ***mysqli_connect*** estabelece a conexão com nosso banco de dados que criamos via PHPmyAdmin. Repare que neste comando passamos 4 parâmetros (veja na imagem).

CRUD - Create

```
<?php
    if(isset($_POST['enviar'])){
        $username = $_POST['username'];
        $password = $_POST['password'];

        //Estabelecendo a conexão
        $connection = mysqli_connect('localhost', 'root', '', 'loginapp');

        if(!$connection){
            die("Não conectou");
        }

        //Query para inserir os dados de acordo com os nomes das variáveis.
        $query = "INSERT INTO usuarios(username,password) VALUES ('$username',
        '$password')";

        //Função para executar uma query em nosso banco.
        $resultado = mysqli_query($connection, $query);

        //Validação
        if(!$resultado){
            die("Não deu certo a inclusão" .mysqli_error());
        } else {
            echo "Dados criados com sucesso";
        }
    }
?>
```



```
<?php
$connection = mysqli_connect('localhost', 'root', '', 'exemplo');
if(!$connection){
    die("Conexão falhou");
}

$query = "SELECT * FROM usuarios";
$resultado = mysqli_query($connection, $query);
?>
```

Através do comando SELECT você seleciona todos os dados da sua tabela

```
<body>
<div class="container">

    <div class="col-sm-6">

        <pre>
        <?php
            while($row = mysqli_fetch_assoc($resultado))
            {
                print_r($row);
            }
        ?>
        </pre>
    </div>

</div>
</body>
```

A função *mysqli_fetch_assoc* irá retornar as informações da sua tabela para dentro de um array, feito isso é só imprimir ele com o comando *print_r*

Desafio 8

Crie um banco de dados (*desafio*) de forma manual no PHPmyAdmin.

Crie uma tabela com 4 colunas (*id*, *username*, *password*, *email*).

Crie um arquivo `desafio_8.php` que conecte ao banco de dados *desafio* e através de uma query realize inserções.

Adicione através do formulário 10 cadastros.

Crie um arquivo `desafio_8_read.php` que leia todos os cadastros.



CRUD - UPDATE



O UPDATE pode ser dividido em 2 etapas.

A 1ª é a inserção da quantidade de elementos que temos em nossa tabela em um select/option em nossa estrutura HTML:

```
<div class="form-group">
  <select name="id" id="">
    <!--<option value="">1</option>          -->
    <?php
      while($row = mysqli_fetch_assoc($resultado)){
        $id = $row['id'];
        echo "<option value='$id'>$id</option>";
      }
    ?>
  </select>
</div>
```

CRUD - UPDATE



Agora que nós podemos selecionar os elementos da tabela através do ID, vamos executar a query de atualização dos dados:

```
if(isset($_POST['update'])){
    $username = $_POST['username'];
    $password = $_POST['password'];
    $id = $_POST['id'];

    $query = "UPDATE usuarios SET ";
    $query .= "username = '$username', ";
    $query .= "password = '$password' ";
    $query .= "WHERE id = $id ";

    $resultado = mysqli_query($connection, $query);

    if(!$resultado){
        die("Falha na atualização");
    } else {
        echo "Dados atualizados com sucesso!";
    }
}
```


CRUD - DELETE



O nosso DELETE mantém a estrutura HTML do UPDATE, pois precisamos selecionar o ID a ser excluído. Neste caso só iremos alterar a nossa query:

```
39 v function deletaLinha(){
40     global $connection;
41     $username = $_POST['username'];
42     $password = $_POST['password'];
43     $id = $_POST['id'];
44
45     $query = "DELETE from usuarios ";
46     $query .= "WHERE id = $id ";
47
48     $resultado = mysqli_query($connection, $query);
49
50 v if(!$resultado){
51     die("Falha na atualização");
52 v } else {
53     echo "Dados removidos com sucesso!";
54 }
55
56 }
```

*este trecho de código pertence ao arquivo *functions.php*, onde reunimos todas as funções em um único arquivo, não esqueça de chamar esta função no seu arquivo principal (*delete.php*)*

Para facilitar a leitura do nosso código e otimizar-mos a nossa sintaxe, contamos com a ajuda de arquivos externos como é o caso do *db.php*, que contém o script de conexão ao banco de dados:

```
• mysql/db.php (exercicios) — Brackets

1  <?php
2  $connection = mysqli_connect('localhost', 'root', '', 'loginapp');

3  if(!$connection){
4      die("Vish.. Deu ruim");
5  }
6  ?>
7
8
```

Assim como o *db.php*, também podemos criar um arquivo *functions.php* que reúne todas as funções do nosso CRUD, só não se esqueça de incluir estes arquivos com a função pré definida **include** e depois chamar as funções criadas dentro deste arquivo. (veja mais no arquivo CRUD.zip em nosso portal)

Desafio 9 - BDD

- > Crie um novo banco de dados (desafio) no PhpMyAdmin
- > Crie uma tabela ('*pokemons*') com 4 rows (id, name, type, hp)
- Através de scripts em PHP, crie um arquivo que faça a conexão entre a página e o bdd e insira os dados de cada pokémon.
- Crie um arquivo que exiba todos os pokémons do tipo fogo.
- Crie um arquivo que permita a atualização de um pokémon.
- Crie um arquivo que permita a exclusão de um pokémon.

