

PHP com Banco de Dados

Wendell Bento Geraldles

IFGO

16 de outubro de 2018

Sumário

- 1 Conectando com um banco de dados
- 2 Executando comandos SQL em um programa PHP
- 3 Exibindo os resultados de comandos SQL
- 4 Inserir, exibir, alterar e excluir dados com PHP
- 5 Referências bibliográficas

PHP com banco de dados

Conectando com um banco de dados

Antes de acessar um banco de dados e começar a realizar operações, precisamos abrir uma conexão com ele. Para isso, utilizaremos uma função de conexão e uma variável do PHP como ponteiro, para referenciar essa conexão. (NIEDERAUER, 2011)

Figura: PHP e MySQL



Em um programa PHP, para fazer a conexão com um banco de dados MySQL utilizamos a função `mysql_connect`.

Código-Fonte 1: Conectando com um banco de dados

```
$conexao = mysql_connect ("localhost", " usuario", " senha");
```

Se o banco de dados estiver localizado em um servidor diferente, basta substituir `localhost` pelo nome ou endereço IP desse servidor.(NIEDERAUER, 2011)

Logo após conectar com o banco de dados, especificamos com o comando `mysql_select_db` qual o banco de dados será utilizado.(NIEDERAUER, 2011)

Código-Fonte 2: Selecionando um banco de dados

```
mysql_select_db (" bdteste" );
```

Quando não formos mais utilizar uma conexão aberta dentro de um programa, pedemos fechá-la com o comando `mysql_close`.(NIEDERAUER, 2011)

Código-Fonte 3: Fechando a conexão com um banco de dados

```
mysql_close ($conexao);
```

Para não ter de repetir todos esses comandos em diversas páginas do site, seria interessante a criação de uma include que faz a conexão com o banco de dados.(NIEDERAUER, 2011)

Código-Fonte 4: conecta_mysql.inc

```
<?php
    $conexao = mysql_connect("localhost","usuario","senha");
    mysql_select_db("bdteste");
?>
```

Se a sua versão do MySQL for 4.1 ou superior você deverá usar a função `mysqli_connect` e `mysqli_select_db`.(NIEDERAUER, 2011)

PHP com banco de dados

Executando comandos SQL em um programa PHP

Para executar comandos em um banco de dados MySQL, iremos utilizar a função `mysql_query`.(NIEDERAUER, 2011)

Código-Fonte 5: Exemplo da função `mysql_query`

```
<?php
    include "conecta_mysql.inc";
    $resultado = mysql_query ("SELECT * FROM produtos");
    mysql_close($conexao);
?>
```

Se a sua versão do MySQL for 4.1 ou superior, você deverá usar `mysqli_query` em vez de `mysql_query`.(NIEDERAUER, 2011)

PHP com banco de dados

Exibindo os resultados de comandos SQL

Para exibir os resultados de comandos SQL no PHP existem diversas funções específicas.(NIEDERAUER, 2011)

Tabela: Funções para MySQL no PHP

Função	Significado
mysql_affected_rows	Retorna o número de linhas afetadas por uma operação.
mysql_fetch_array	Armazena a linha atual do resultado em um array associativo.
mysql_fetch_object	Retorna uma linha como um objeto.
mysql_fetch_row	Armazena a linha atual do resultado em um array.

Fonte: (NIEDERAUER, 2011)

PHP com banco de dados

Exibindo os resultados de comandos SQL

Tabela: Funções para MySQL no PHP

Função	Significado
mysql_result	Retorna uma coluna do resultado.
mysql_num_rows	Retorna o número de linhas de uma consulta.
mysql_num_fields	Retorna o número de colunas de uma consulta.
mysql_field_name	Retorna o nome de uma coluna em uma consulta.

Fonte: (NIEDERAUER, 2011)

É importante também verificar o número da versão do MySQL que você está usando. No PHP 5 foi adicionada a biblioteca **mysqli**. Neste caso a função **mysql_num_rows** seria **mysqli_num_rows** e **mysql_fetch_array** seria **mysqli_fetch_array**.

Antes de prosseguir, é importante que façamos a distinção entre as funções **mysql_affected_rows** e **mysql_num_rows**. A primeira é utilizada nas operações de inserção (**INSERT**), atualização (**UPDATE**) ou exclusão (**DELETE**). A segunda é utilizada em consultas (**SELECT**) para obtermos o número de linhas que foram retornadas.(NIEDERAUER, 2011)

PHP com banco de dados

Exemplo de um programa que apaga todo conteúdo de uma tabela

Código-Fonte 6: apaga_tabela.php

```
<?php
    include "conecta_mysql.inc";
    $resultado = mysql_query ("DELETE FROM produtos");
    $linhas = mysql_affected_rows();
    mysql_close($conexao);
    echo "Foram excluidos $linhas produtos!";
?>
```

A variável `$linhas` conterà o número de registros afetados pelo DELETE executado pelo comando `mysql_query`. Do mesmo modo podemos utilizar os comandos INSERT e UPDATE. (NIEDERAUER, 2011)

PHP com banco de dados

Exemplo de um programa que consulta dados em uma tabela

Código-Fonte 7: consulta_tabela.php

```
<?php
    include "conecta_mysql.inc";
    $resultado = mysql_query ("SELECT * FROM produtos");
    $linhas = mysql_num_rows($resultado);
    mysql_close($conexao);
    echo "A consulta retornou $linhas registros !";
?>
```

Se em vez de utilizar a função **mysql_num_rows**, utilizássemos a **mysql_num_fields**, teríamos o número de campos retornados na consulta.(NIEDERAUER, 2011)

PHP com banco de dados

Exibindo os resultados de comandos SQL

Para obter valores dos campos retornados por uma consulta, existem diversas alternativas. Uma delas é utilizar a função `mysql_result`. (NIEDERAUER, 2011)

`mysql_result (recurso resultado, int linha [, misto couna])`

Tabela: Parâmetros da função `mysql_result`

Parâmetro	Significado
resultado	Nome da variável que recebeu o resultado da função <code>mysql_query</code> .
linha	Número da linha que desejamos recuperar (começando com 0).
coluna	Número da coluna (começando com 0) ou nome do campo da tabela.

Fonte: (NIEDERAUER, 2011)

PHP com banco de dados

Exemplo de um programa que exibe alguns valores da primeira linha retornada de uma consulta

Código-Fonte 8: mostra_registros.php

```
<?php
include "conecta_mysql.inc";
$resultado = mysql_query ("SELECT * FROM produtos");
$codigo = mysql_result ($resultado , 0, 0);
$nome = mysql_result($resultado, 0, "nome_produto");
$preco = mysql_result ($resultado , 0, "preco");
mysql_close($conexao);
echo "Codigo do produto: $codigo <br>";
echo "Nome do produto: $nome <br>";
echo "Preco do produto: $preco";
?>
```

PHP com banco de dados

Exemplo de um programa que exibe o nome e o preço de todos os produtos.

Código-Fonte 9: mostra_registros2.php

```
<?php
include "conecta_mysql.inc";
$resultado = mysql_query ("SELECT * FROM produtos");
$linhas = mysql_num_rows($resultado);
for ($i=0; $i<$linhas; $i++) {
    $nome = mysql_result($resultado, $i, "nome_produto");
    $preco = mysql_result ($resultado, $i, "preco");
    echo "Nome do produto: $nome <br>";
    echo "Preco: $preco <br>";
}
mysql_close($conexao);
?>
```

Outra forma (mais recomendada) de obter os valores retornados de uma consulta é por meio das funções **mysql_fetch_row** e **mysql_fetch_array**. Estas funções colocam as linhas retornadas (uma por vez) em uma array. A diferença entre as duas é que a função `mysql_fetch_array` permite o acesso aos índices do array por chave associativa (nome dos campos), enquanto a função `mysql_fetch_row` utiliza somente índices numéricos. A sintaxe básica dessas função é seguinte:(NIEDERAUER, 2011)

```
mysql_fetch_row (recurso resultado)
```

```
mysql_fetch_array (recurso resultado)
```


PHP com banco de dados

Exemplo de um programa que exibe todos os produtos.

Código-Fonte 10: mostra_registros3.php

```
<?php
include "conecta_mysql.inc";
$resultado = mysqli_query($conexao,"SELECT * FROM produtos");
while ( $linhas = mysqli_fetch_row( $resultado) ) {
    echo "<p>". $linhas[0]. " - ". $linhas [1]. " - ". $linhas [2]. "<p>";
}
mysqli_close ($conexao);
?>
```

PHP com banco de dados

Exemplo de um programa que exibe todos os produtos.

Código-Fonte 11: mostra_registros3.php

```
<?php
include "conecta_mysql.inc";
$resultado = mysqli_query($conexao,"SELECT * FROM produtos");
while ( $linhas = mysqli_fetch_row($resultado) ) {
    echo "<p>";
    foreach ( $linhas as $valor ) {
        echo $valor." - ";
    }
    echo "<p>";
}
mysqli_close ($conexao);
?>
```

PHP com banco de dados

Inserir, exibir, alterar e excluir dados com PHP

As principais operações realizadas em um banco de dados são inclusão, alteração, consulta e exclusão de registros. Em inglês a sigla CRUD (**Create, Read, Update e Delete**) representam estas quatro operações. Os exemplos a seguir mostram como criar programas que realizam estas operações.

PHP com banco de dados

Inserir dados do formulário no MySQL

Código-Fonte 12: inserir.php

```
<?php
include "conecta_mysql.inc";
$codigo = $_POST["codigo"];
$produto = $_POST["produto"];
$preco = $_POST["preco"];
$resultado = mysqli_query($conexao,"INSERT INTO produtos VALUES
                        ($codigo,'$produto',$preco)");
$linhas = mysqli_affected_rows($conexao);
mysqli_close($conexao);
echo "<p>";
echo "Foram inseridos  " . $linhas . "  registros ";
?>
```

PHP com banco de dados

Inserir dados do formulário no MySQL

Código-Fonte 13: inserir.php

```
<?php
//Este trecho pode ser inserido para validar o formulario
if (empty($codigo) or empty($produto) or empty($preco)) {
    $linhas = 0;
    echo "<p>Todos os campos devem ser preenchidos</p>";
} else {
    $resultado = mysqli_query($conexao,"INSERT INTO produtos
        VALUES ($codigo,'$produto',$preco)");
    $linhas = mysqli_affected_rows($conexao);
    mysqli_close($conexao);
}
?>
```

PHP com banco de dados

Exibir dados do MySQL com PHP

Código-Fonte 14: listar.php

```
<?php
include "conecta_mysql.inc";
$resultado = mysqli_query($conexao,"SELECT * FROM produtos");
while ( $linhas = mysqli_fetch_row( $resultado) ) {
    echo "<p>". $linhas[0]. " – ". $linhas [1]. " – ". $linhas [2]. " – ". "
        <a href='alterar.php?id=$linhas[0] '> Editar </a>". " – ". " <
        a href='exclui.php?id=$linhas[0] '> Excluir </a>". " <p>";
}
mysqli_close ( $conexao);
echo("<a href='formproduto.html'>Formulario</a>");
?>
```

PHP com banco de dados

Alterar dados do MySQL com PHP

Código-Fonte 15: alterar.php

```
<?php
$id=$_GET['id'];
include "conecta_mysql.inc";
$resultado = mysqli_query($conexao,"SELECT * FROM produtos WHERE codigo=$id");
echo "<form action='grava.php' method='POST'>";
while( $linhas = mysqli_fetch_array( $resultado )) {
?>
//Aqui o codigo que vai preencher o formulario com os dados vindos da consulta SQL
<?php } ?>
```

PHP com banco de dados

Alterar dados do MySQL com PHP

Código-Fonte 16: alterar.php

```
<p>Codigo</p>
<input type='text' name='codigo' readonly size='10' value=" <?php echo $linhas[0] ?>"
    ><br>
<p>Nome do produto</p>
<input type='text' name='produto' size='50' value=" <?php echo $linhas[1] ?>" ><br>
<p>Preco</p>
<input type='text' name='preco' size='10' value=" <?php echo $linhas[2] ?>" ><br>
<input type='submit' value='Salvar' ><p>
<a href='listar.php'>Listar produtos</a><p>
<a href='formproduto.html'>Formulario</a>
```


PHP com banco de dados

Alterar dados do MySQL com PHP

Código-Fonte 17: grava.php

```
<?php
include "conecta_mysql.inc";
$codigo=$_POST['codigo'];
$produto=$_POST['produto'];
$preco=$_POST['preco'];
$resultado=mysqli_query($conexao," UPDATE produtos SET codigo='
    $codigo', nome_produto='$produto', preco='$preco' WHERE codigo=
    $codigo");
$registro = mysqli_affected_rows($conexao);
mysqli_close($conexao);
echo $registro ." registro alterado com sucesso!";
?>
```

PHP com banco de dados

Excluir dados do MySQL com PHP

Código-Fonte 18: exclui.php

```
<?php
$id=$_GET['id'];
include "conecta_mysql.inc";
$resultado=mysqli_query($conexao,"DELETE FROM produtos WHERE
    codigo=$id");
$registro = mysqli_affected_rows($conexao);
echo "<p>";
echo $registro ." registro excluido com sucesso!";
echo "<p>";
mysqli_close($conexao);
?>
```

 NIEDERAUER, J. **Desenvolvendo Websites com PHP**. 2. ed. São Paulo: Novatec Editora, 2011. 301 p.