Apache MySQL

php

phpMyAdmin

Prof. Leandro Martins Dallanora
leandro.dallanora@iffarroupilha.edu.br

Introdução

- Com o PHP podemos acessar diversos banco de dados, como o MySQL, PostgreSQL, Oracle, SQL Server, Firebird, Sysbase, Informix, SQLite e outros mais.

- Um dos bancos de dados mais utilizados com o PHP é o MySQL.

- O PHP possui um módulo específico para esse banco. Utilizaremos o MySQL em nossos exemplos por ser um banco simples de operar e utilizar pouco processamento e memória, em comparação com os outros.

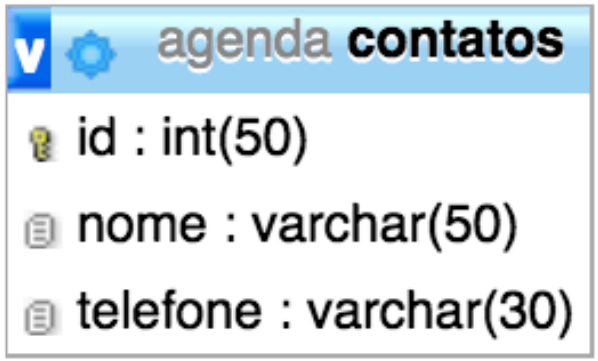
Criando o Banco de Dados

- Antes de construir o *site*, devemos modelar os dados a serem manipulados pelas páginas, porque assim teremos uma melhor visão das informações a serem acessadas em cada página.

- Para usarmos em nossos exemplos, vamos criar um banco de dados no MySQL com o nome **agenda** e construir as tabelas descritas no slide a seguir.

Criando o Banco de Dados

- Nosso banco de dados terá somente uma tabela.



Criando o Banco de Dados

```
CREATE DATABASE IF NOT EXISTS `agenda` DEFAULT CHARACTER
SET latin1 COLLATE latin1_swedish_ci;

USE `agenda`;

DROP TABLE IF EXISTS `contatos`;
CREATE TABLE `contatos` (
  `id` int(50) NOT NULL,
  `nome` varchar(50) NOT NULL,
  `telefone` varchar(30) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `contatos`
  ADD PRIMARY KEY (`id`);
```

MySQLi

- Para realizar a conexão e o acesso ao Banco de Dados, usaremos a extensão MySQLi.
- MySQLi é uma extensão do MySQL para a linguagem de programação PHP.
- A extensão MySQL (MySQLImproved) é um driver de banco de dados relacional utilizado na linguagem de programação PHP para fornecer uma interface com bancos de dados MySQL.

Conectando

- Em uma página PHP, o primeiro passo é conectar com o banco de dados. Podemos realizar isso através do seguinte comando:

```
$conexao = mysqli_connect(“servidor”, “usuário”, “senha”, “banco”);
```

Conectando

onde:

- **servidor – IP** (ou *hostname*) é a porta do servidor onde está o banco de dados, no formato **servidor: porta**. Se o banco de dados estiver no mesmo computador, pode usar ***localhost***. Se a porta não for informada, será utilizada a porta padrão, que no MySQL é a 3306.
- **usuário** e **senha** cadastrados no banco de dados.
- **banco** informa qual banco de dados queremos acessar para executar os comandos SQL.

- Uma vez estabelecida a conexão, esta será encerrada automaticamente ao final da execução da página.

Exemplo:

```
<?php
$bdServidor = “localhost”;
$bdUsuario = “root”;
$bdSenha = “”;  // senha vazia
$bdBanco = “agenda”;

$conexao = mysqli_connect($bdServidor, $bdUsuario, $bdSenha,
$bdBanco);

if (mysqli_connect_errno($conexao)) {
        echo "Problemas para conectar no banco. Erro: ";
        echo mysqli_connect_error();
        die();
}
```

Include

- Cada página que necessitar acessar o banco de dados deverá ter esse comando no início.
- Como um sistema *web* geralmente possui várias páginas, replicar esse comando não será uma boa solução. Caso tenha que mudar algum parâmetro, como por exemplo o IP do servidor, todas as páginas sofrerão modificações. Um trabalho e tanto!
- Para evitar esse trabalho de manutenção, colocam-se os dados de conexão com o banco em um único arquivo e todas as páginas fazem acesso a esse arquivo utilizando o comando **include**.

Include

- Então, o código ficará em um arquivo, por exemplo, **“conecta.php”**, e todas as outras páginas incluirão esse código da seguinte forma:

```
include(“conecta.php”);
```

- **PRONTO!!!** Uma vez conectado com o banco de dados, podemos realizar todas as operações para manipulação dos dados: inserir, consultar, editar e excluir. O que precisaremos saber para realizar essas operações é sobre a linguagem SQL.

Executando comandos SQL

- A linguagem padrão de comunicação com os bancos de dados é a linguagem SQL.
- Para fazer com que o PHP execute os comandos SQL no banco de dados MySQL, utiliza-se o comando:

mysqli_query($conexao, $sql);

Onde:

$conexao é a conexão MySQLi criada anteriormente para conectar ao MySQL;

mysqli_query() é a função que executa uma consulta ao BD;

$sql é a variável que contém a string com o comando SQL;

Inserindo dados

- Portanto, para inserir os dados no banco, o comando em SQL que é utilizado é o INSERT.

- Para testarmos, vamos criar uma página contendo HTML e PHP, com um formulário para digitar os dados e depois inserir no banco.

- O formulário conterá os mesmos campos que a tabela **contatos** do nosso banco de dados **agenda**.

Inserindo dados

- Vamos criar uma variável ($sql) que receberá o comando INSERT com os dados do formulário.

- Na sintaxe do INSERT, a parte do VALUES será onde passaremos os valores para o banco de dados. É nesse ponto que usaremos as variáveis com os valores obtidos dos formulários.

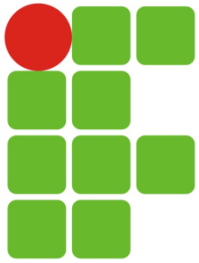
```
$sql = "INSERT INTO contatos (nome, telefone) VALUES
(\"$nome\", \"$telefone\")";
```

Inserindo dados

- Logo em seguida chamaremos o comando em PHP para executar esse SQL no banco de dados.

mysqli_query($conexao, $sql);

- Os campos no banco de dados do tipo "autonumeração" (ou "autoincremento") não devem ser passados para o comando INSERT.
- Verifique que em nossa tabela de exemplo, o campo **id** é um campo autoincremento e ele não foi passado para o INSERT.



Listando os dados

- Após inserir os dados no banco, temos a possibilidade de recuperá-los e mostrá-los para o usuário. O comando em SQL que faz isso é o SELECT.
- O comando em PHP para recuperar os dados é o mesmo usado no inserir. O que muda é o comando SQL passado para o banco, que agora é o SELECT.
- O retorno de um SELECT no banco é um conjunto de registros. Precisamos percorrer todos esses registros, pegando o primeiro, passando para o próximo, e assim por diante até o último. O comando em PHP que faz isso é o **mysqli_fetch_assoc()**

Listando os dados

- mysqli_fetch_assoc() – retorna um registro de uma consulta e aponta para o próximo registro. Retorna *false* quando não existir mais registros, ou seja, quando chegar ao último. Retorna o registro em forma de *array*, em que as chaves são os nomes das colunas da tabela no banco de dados.

  Sua sintaxe é:

mysqli_fetch_assoc($resultado)

em que: **$resultado** é o resultado do SELECT, ou seja, o conjunto de registros.

Exemplo

```
// Seleciona todos os dados da tabela contatos
$sql = "SELECT * FROM contatos";

// Executa o Select
$resultado = mysqli_query($conexao,$sql);

//Lista os contatos
while ($dados = mysqli_fetch_assoc($resultado)) {
        echo $dados['id']." <br> ";
        echo $dados['nome'] ." <br> ";
        echo $dados['telefone'] ." <br> ";
}
```

Outros…

- Ainda com o exemplo anterior em mente, podemos utilizar o comando:

mysqli_num_rows($resultado);

Para verificar o número de linhas retornadas pelo comando Select… Ou ainda… se utilizarmos um comando SQL como Delete ou Update, podemos utilizar o comando:

$linhasAfetadas = mysqli_affected_rows($conexao);

Para verificar quantas linhas foram afetadas pelo último comando SQL utilizado…