MySQL e PHP

Há uma boa possibilidade de encontrar o MySQL associado ao PHP na construção e operação de sistemas web. Este texto vai discorrer sobre essa plataforma. Convivem aqui os seguintes atores:

Servidor HTTP Um computador capaz de receber requisições HTTP.

PHP Um PHP funcional e operando no servidor HTTP.

MySQL Um servidor – que pode ser o mesmo – de banco de dados operando com MySQL.

páginas HTML no servidor HTTP contendo HTML, PHP e MySQL tudo integrado de maneira a responder a requisições chegadas pela WEB.

No caso de Windows é necessário habilitar o módulo do MySQL com os componentes do PHP. Para fazer isso deve-se copíar alguns arquivos do PHP para dentro de pastas do windows. Há também que assinalar algumas variáveis de controle dentro do MySQL e dentro do PHP. A receita específica deve ser buscada na documentação da versão que estiver em uso.

PHP

O PHP se insere entre o cliente e o servidor HTML interpretando algumas tags que são intercaladas no código HTML. Elas sempre tem o aspecto

```
<?php especificação PHP ?>
```

Por exemplo, se você criar um `processorder.php` contendo

```
<?php
echo '<p>Ordem Processada aos ';
echo date('H:i, jS F');
echo '</p>;
?>
```

Está se pedindo ao PHP que através da chamada à função `date()` atualize e mostre o resultado esperado. A função date() é uma da extensa biblioteca de funções PHP que recebem e devolvem dados. A propósito na função acima, `H` é a hora em formato de 24H, `i` é minutos eventualmente com zeros à esquerda se necessário, `j` é o dia do mês sem zero à esquerda e `S` representa o sufixo ordinal (neste caso th) e `F` é o nome completo do mês.

Todos os nomes de variáveis em PHP iniciam pelo caracter $. Esquecer o símbolo $ em nomes de variáveis é um erro de programação muito comum em PHP. Então, se houver uma variável de formulário chamada `numped`, dentro do PHP ela será usada como `$numped` ou `$_POST['numped']` ou ainda `$HTTP_POST_VARS['numedp']`. (Ou GET, dependendo do método usado no HTML). A propósito o PHP é suscetível à caixa dos nomes em geral. Assim a variável `$TA` é diferente da variável `$Ta`. Variáveis em PHP não precisam ser declaradas previamente a seu uso. O operador de atribuição em PHP é `=`.

O PHP faz distinção entre aspas simples e duplas. Um conteúdo entre aspas simples será enviado como está para o navegador. Já o mesmo conteúdo entre aspas duplas será substituído por seu valor antes de ser enviado ao navegador. Acompanhe:

```
echo "$qtdpedido unidades <br />";
echo '$qtdpedido unidades <br />';
```

O primeiro caso informará algo como 22 unidades enquanto o segundo será apresentado literalmente como está no exemplo.

Os tipos aceitos pelo PHP são integer (inteiro), double (reais de precisão dupla), string (cadeia), boolean (lógico), array (ocorrência múltipla homogênea), object (instâncias de classe), resource, pdfinfo, pdfdoc e NULL. A tipagem do PHP é fraca o que significa que o tipo da variável é definido no momento da sua primeira alocação e depende do que é atribuído a ela. Esta regra pode ser modificada usando coerção (cast). O nome da variável pode ser alterado dinamicamente. Veja no exemplo

```
$varname = 'qtdpedido';
```

Agora a variável `$qtdpedido` pode ser acessada pelo seu nome (`$qtdpedido`) ou então por `$$varname`. É a mesma coisa.

O PHP aceita os operadores convencionais: (`+`, `-`, `/`, `*` e `%`). Há também os * e `--`. Em uma operação envolvendo número e string, o PHP tentará converter o string em número e se não conseguir, o string terá valor 0. A concatenação entre strings é o operador `.`. Os operadores de comparação são: `==`, `!=` ou `<>`, `>`, `<`, `>=`, `<=`. Já os operadores lógicos são `!`, `&&` e `||`.

As estruturas de controle são: `if` que é seguida por uma condição entre parênteses e pode ser seguida por um comando ou por um bloco. Neste caso, ele será escrito entre { e }. Pode ou não haver um else. Há também o `while` seguido por uma condição entre parênteses e um comando ou bloco. Há o `for` cuja síntaxe é exatamente igual à da linguagem C.

Programas PHP podem manusear arquivos locais ao servidor. Podem também manusear arquivos remotos. Se o nome do arquivo começa com `ftp://` a conexão é estabelecida. Fazem OPEN, READ, WRITE, CLOSE de maneira muito parecida a C.;

Podem ser criadas funções de usuário dentro de programas PHP. Novamente a síntaxe e a semântica é parecida à da linguagem C. O PHP aceita recursividade. PHP é orientado a objetos e aceita herança e polimorfismo.

Resumo

Eis um resumo desta arquitetura funcionando:

1. Alguém se conecta a um servidor de páginas web através de um formulário, por exemplo, consultando os preços de um aparelho qualquer. A página que vai responder chama-se `precos.php`.

2. O Apache recebe a solicitação para `precos.php`, recupera este arquivo e o passa ao PHP.

3. o PHP analisa `precos.php`. Dentro dele há um comando de conexão para o `MySQL` para fazer a pesquisa. O PHP abre a conexão e manda a pesquisa.

4. O MySQL recebe a pesquisa, a faz e retorna os resultados para o PHP.

5. o PHP continua processando o script `precos.php`. Recebendo os dados do MySQL intercála-os ao código HTML pré-existente compondo uma página HTML bonita e coerente. Retorna isso tudo para o Apache

6. Que o devolve para o consultante original na forma de HTML puro.

Uma sessão completa

A seguir, devidamente comentada uma sessão completa de consulta a um banco MySQL a partir de um aplicativo PHP.

Seja a seguinte página HTML que consulta um cadastro de produtos

```
<html> <head> <title> Consulta a preços
</title> </head> <body>
<h1> Consulta a preços no catálogo </h1>
<form action="precos.php" method="post">
Escolha o produto: <br />
<select name="produto">
<option value="som" Som </option>
<option value="tv" Televisor </option>
</select> <br />
Informe palavras da pesquisa: </ br>
<input name="termop" type="text"> </ br>
<input type="submit" value "pesquisa">
</form> </body> </html>
```

Agora vamos estudar `precos.php`:

```
<html> <head> <title> Resposta </title>
</head> <body> <h1> Resultados: </h1>
<?php
// cria nome de variavel abreviado
$produto = $HTTP_POST_VARS['produto'];
$termop = $HTTP_POST_VARS['termop'];
$termop = trim($termop);
if (!$produto || $termop) {
   echo 'Consulta inválida, reentre';
   exit; }
$produto=addslashes($produto);
$termop=addslashes($termop);
@ $db = mysql_pconnect('host', 'user' 'psw');
if (!$db) {
   echo 'Erro de conexão. Tente novamente'; }
mysql_select_db('prods');
$query = "select * from produs where
".$produto." like '%".$termop."%'";
$result = mysql_query($query);
$num_results = mysql_num_rows($result);
echo '<p>Prods achados:',.$num_results.'</p>';
for ($i=0; i<$num_results;$i++) {
  $row=mysql_fetch_array($result);
  echo '<p><strong>'.($i+1).'. Produto: ';
  echo (stripslashes($row['marca']));
  echo (stripslashes($row['preco']));
  echo '</p>';
  } ?>
</body>
</html>
```

Mais um exemplo completo

Já usando a interface mysqli (MySQL Improved) que é um driver moderno e relacional do PHP para interfacear com o MySQL. Ele tem 2 modos de usar: orientado a objeto e procedural (este último mais parecido com o jeito antigo de usar). O exemplo a seguir é procedural. Veja a documentação para mais informações.

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

// Create connection
$conn = mysqli_connect($servername, $username,
    $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " .
    mysqli_connect_error());
}

$sql = "SELECT id, firstname, lastname
    FROM MyGuests";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
    // output data of each row
    while($row = mysqli_fetch_assoc($result)) {
        echo "id: " . $row["id"]. " - Name: " .
        $row["firstname"]. " " . $row["lastname"].
        "<br>";
    }
} else {
    echo "0 results";
}

mysqli_close($conn);
?>
```

☞ Para você fazer

1. Consulte a documentação PHP-MySQL e elabore uma lista com as 5 ou 6 principais funções que fazem a ligação entre esses dois ambientes.

2. Faça um esboço de projeto para um aplicativo WEB usando MySQL e PHP que permita cadastrar e consultar coisas. Descreva o diagrama de contexto e os componentes do sistema.