**Conexão com o MySQL via PHP**

*(versão do MySQL: 5.x)*

Para interagir com uma base de dados SQL existem três comandos básicos que devem ser utilizados:

* um que faz a conexão com o servidor de banco de dados;
* um que seleciona a base de dados a ser utilizada;
* um terceiro que executa uma “query” (consulta) SQL.

Existe, ainda, um quarto comando que pode ou ser utilizado, que é o comando que fecha a conexão com o banco de dados.

**Abrindo e fechando conexão com o MySQL**

*mysql\_connect()*

Este comando é utilizado para criar a conexão com a base de dados num servidor MySQL.

O comando mysql\_pconnect também pode ser utilizado para criar a conexão. A diferença entre os dois comandos é que o mysql\_pconnect estabelece uma conexão permanente, ou seja, que não é encerrada ao final da execução do script. Utilizaremos a primeira opção: mysql\_connect.

Sintaxe:

mysql\_connect(string [host[:porta]], string [login], string [senha])

[or die (“mensagem de erro”)];

Onde:

* string [host[:porta]] – é o endereço do servidor, onde o banco está armazenado;
* string [login] – é o usuário do banco de dados;
* string [senha] – é a senha do usuário do banco de dados MySQL;
* die – exibe uma mensagem indicando que a conexão não foi efetuada.

Exemplo:

<?php

$conexao=mysql\_connect(“localhost”,“root”, focus”)

or die (“Conexão não efetuada”);

?>

O valor de retorno é um inteiro que identifica a conexão, ou falso se a conexão falhar. Antes de tentar estabelecer uma conexão, o interpretador PHP verifica se já existe uma conexão estabelecida com o mesmo host, o mesmo login e a mesma senha. Se existir, o identificador desta conexão é retornado. Senão, uma nova conexão é criada.

Assim, se a conexão for bem sucedida (se existir um servidor no endereço especificado que possua o usuário com a senha fornecida), o identificador da conexão fica armazenado na variável $conexao, caso contrário, será mostrada a mensagem “Conexão não efetuada”.

*mysql\_close()*

Comando utilizado para encerrar uma conexão estabelecida com o comando mysql\_connect antes de chegar ao final do script. Caso esse comando não seja utilizado, a conexão é encerrada no final do script.

Sintaxe:

mysql\_close(int [string da conexão]);

Se o identificador não for fornecido, a última conexão estabelecida será encerrada.

Exemplo:

<?php

mysql\_close($conexao);

?>

Obs.: O comando mysql\_close não encerra conexões estabelecidas com o mysql\_pconnect.

*mysql\_select\_db()*

Comando utilizado para selecionar a base de dados depois que a conexão for estabelecida. Se nenhuma de conexão é especificada, a última conexão aberta é assumida. Se nenhuma conexão esta aberta, a função irá tentar abrir uma conexão como se mysql\_connect() fosse chamada sem argumentos e usá-la.

Sintaxe:

mysql\_select\_db(banco de dados,[string de conexão])

or die [(“mensagem”)];

Onde:

* banco de dados – é o banco de dados que será utilizado;
* string de conexão – é a conexão criada com o servidor MySQL.
* die – exibe uma mensagem indicando que a conexão não foi efetuada.

Exemplo:

<?php

$conexao=mysql\_connect(“localhost”,“root”,“focus”);

if($conexao){

die(“Conexão não estabelecida”);

}

db\_select=mysql\_select\_db(“focus”,$conexao);

?>

*mysql\_query()*

Este comando é utilizado para realizar uma consulta SQL no MySQL.

Sintaxe:

mysql\_query(string da consulta);

Onde:

* String da consulta – é um dos comandos utilizados do SQL para efetuar uma consulta, uma inclusão, uma alteração ou uma exclusão no banco de dados.

O comando mysql\_query() envia uma query (consulta) para o banco de dados ativo no servidor da conexão informada em string da consulta. Se o parâmetro string da consulta não é especificado, a última conexão aberta é usada. Se nenhuma conexão esta aberta, a função tenta estabelecer uma conexão como mysql\_connect() seja chamada sem argumentos e usá-la. O resultado é guardado em buffer.

Exemplo:

<?php

sql=“select \* from alunos where id\_aluno=10”;

mysql\_query (sql);

?>

Vamos ver um exemplo de um script PHP utilizando os comandos vistos acima que cria uma conexão com o banco de dados Focus, seleciona a tabela alunos e insere alguns dados na tabela.

<?php

$conexao=mysql\_connect(“localhost”,“root”,“focus”);

mysql\_select\_db(“focus”,$conexão);

$insere=“Insert into alunos (id\_aluno, nome, endereço, turma, turno)

Values(1, ‘Glaucio’, ‘Av. das Américas’, ‘1101’, ‘manhã’)”;

$insere1=“Insert into alunos (id\_aluno, nome, endereço, turma, turno)

Values(2, ‘Alexandre’, ‘Av. das Américas’, ‘1101’, ‘tarde’)”;

$insere2=“Insert into alunos (id\_aluno, nome, endereço, turma, turno)

Values(3, ‘Lucinaldo’, ‘Av. das Américas’, ‘1101’, ‘tarde’)”;

mysql\_query($insere,$conexao)

or die (“Não foi possível executar a inserção.”);

mysql\_query($insere1, $conexao)

or die (“Não foi possível executar a inserção.”);

mysql\_query($insere2, $conexao)

or die (“Não foi possível executar a inserção.”);

$delete=Delete from alunos where turno=“tarde”);

mysql\_query($delete,$conexao);

echo(“Todos os alunos do turno da tarde foram excluídos.”);

mysql\_close($conexao);

?>

**Tratamento de resultados de query Select**

Ao executar uma query SQL SELECT através do comando mysql\_query, o identificador do resultado deve ser armazenado numa variável que pode ser tratada de diversas formas.

Duas maneiras interessantes de fazê-lo usam um dos comandos a seguir: mysql\_result, mysql\_fetch\_row ou mysql\_fetch\_array.

*mysql\_result()*

Esta função retorna o resultado de uma query SQL.

Sintaxe:

mysql\_result(resultado,linha,mixed[campo]);

Onde:

* *resultado:* é o identificador do resultado, obtido com o retorno da função mysql\_query;
* *linha*: especifica o registro a ser exibido, já que uma query SELECT pode retornar diversos registros;
* *campo:* - é o identificador do campo a ser exibido, sendo o tipo descrito como mixed pela possibilidade de ser de diversos tipos (neste caso, inteiro ou string).

Exemplo:

<? php

$conexao=mysql\_connect(“localhost”, “root”, “focus”);

mysql\_select\_db(“focus”, $conexão);

$insere=“Insert into alunos (id\_aluno, nome, endereço, turma, turno)

Values(1, ‘Glaucio’, ‘Av. das Américas’, ‘1101’, ‘manhã’)”;

$insere1=“Insert into alunos (id\_aluno, nome, endereço, turma, turno)

Values(2, ‘Alexandre, ‘Av. das Américas’, ‘1101’, ‘tarde’)”;

mysql\_query($insere,$conexao)

or die (“Não foi possível executar a inserção.”);

mysql\_query($insere1,$conexao)

or die (“Não foi possível executar a inserção.”);

$consulta=“Select Id\_Aluno, nome, turno from alunos”;

$resultado=mysql\_query ($consulta, $conexao);

$nome=mysql\_result($resultado,0,“nome”);

$turno=mysql\_result($resultado,0,“turno”);

echo (“Nome: ”.$nome.“<p>”.“Turno: ”.$turno);

mysql\_close($conexao);

?>

Com o exemplo acima, o resultado será:

|  |
| --- |
| Nome: Gláucio  Turno: Manhã |

*mysql\_fetch\_array()*

Esta função lê uma linha do resultado e devolve um array, cujos índices são os nomes dos campos. A execução seguinte do mesmo comando lerá a próxima linha, até chegar ao final do resultado.

Sintaxe:

mysql\_fecth\_array(string da cosulta SQL);

Exemplo:

<?php

$conexao=mysql\_connect(“localhost”,“root”,“focus”);

mysql\_select\_db(“focus”,$conexão);

$insere=“Insert into alunos (id\_aluno, nome, endereço, turma, turno)

Values(1, ‘Glaucio’, ‘Av. das Américas’, ‘1101’, ‘manhã’)”;

$insere1=“Insert into alunos (id\_aluno, nome, endereço, turma, turno)

Values(2, ‘Alexandre, ‘Av. das Américas’, ‘1101’, ‘tarde’)”;

mysql\_query($insere, $conexao)

or die (“Não foi possível executar a inserção.”);

mysql\_query ($insere1, $conexao)

or die (“Não foi possível executar a inserção.”);

$consulta=“Select Id\_Aluno, nome, turno from alunos”;

$resultado=mysql\_query($consulta,$conexao);

$dados=mysql\_fecth\_array($resultado);

$nome=$dados[“nome”];

$turno=$dados[“turno”];

echo (“Nome: “.$nome.”<p>”.“Turno: “.$turno);

mysql\_close($conexao);

?>

Com o exemplo acima, o resultado será:

|  |
| --- |
| Nome: Gláucio  Turno: Manhã |

*mysql\_fetch\_row()*

Esta função é semelhante à função mysql\_fetch\_array, com a diferença que os índices do array são numéricos, iniciando pelo 0 (zero).

Sintaxe:

mysql\_fetch\_row(string de consulta);

Exemplo:

O exemplo anterior substituindo a função mysql\_fecth\_array pela mysql\_fetch\_row.

*mysql\_free\_result()*

Esta função libera a memória do resultado de uma consulta. Sintaxe:

mysql\_free\_result(resource result)

Exemplo:

<?php

$conexao = mysql\_connect (“localhost”, “root”, “focus”);

mysql\_select\_db(“focus”, $conexão);

$insere=“Insert into alunos (id\_aluno, nome, endereço, turma, turno)

Values(1, ‘Glaucio’, ‘Av. das Américas’, ‘1101’, ‘manhã’)”;

$insere1=“Insert into alunos (id\_aluno, nome, endereço, turma, turno)

Values (2, ‘Alexandre, ‘Av. das Américas’, ‘1101’, ‘tarde’)”;

mysql\_query($insere, $conexao)

or die (“Não foi possível executar a inserção.”);

mysql\_query($insere1, $conexao)

or die (“Não foi possível executar a inserção.”);

$consulta=“Select Id\_Aluno, nome, turno from alunos”;

$resultado=mysql\_query($consulta,$conexao);

$dados=mysql\_fecth\_array($resultado);

$nome=$dados[“nome”];

$turno=$dados[turno];

echo (“Nome: ”.$nome.“<p>”.“Turno: ”.$turno);

mysql\_free\_result($resultado);

mysql\_close ($conexao);

?>

No exemplo acima, a função mysql\_free\_result() irá liberar toda a memória usada com o identificador de resultado $resultado.

Obs.: mysql\_free\_result() somente precisa ser chamado se você esta preocupado em quanta memória esta sendo usada para query num grande conjunto de resultados. Toda a memória usada do resultado é liberada automaticamente ao final da execução do script.