

# 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
$conexao=mysql_connect("localhost","root","focus");
mysql_select_db("focus",$conexao);

$inserte="Insert into alunos (id_aluno, nome, endereço, turma, turno)
Values(1, 'Glaucio', 'Av. das Américas', '1101', 'manhã')";
$inserte1="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", $conexao);

$insere="Insert into alunos (id_aluno, nome, endereço, turma, turno)
```

```

Values(1, 'Glaucio', 'Av. das Américas', '1101', 'manhã');
$insert= "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($insert,$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_fetch_array(string da consulta SQL);
```

Exemplo:

```

<?php
$conexao=mysql_connect("localhost","root","focus");
mysql_select_db("focus",$conexao);

$insere="Insert into alunos (id_aluno, nome, endereço, turma, turno)
Values(1, 'Glaucio', 'Av. das Américas', '1101', 'manhã');

$insert= "Insert into alunos (id_aluno, nome, endereço, turma, turno)
Values(2, 'Alexandre, 'Av. das Américas', '1101', 'tarde')";

mysql_query($insere, $conexao)
or die ("Não foi possível executar a inserção.");
mysql_query ($insert, $conexao)
or die ("Não foi possível executar a inserção.");

```

```

$consulta="Select Id_Aluno, nome, turno from alunos";
$resultado=mysql_query($consulta,$conexao);

$dados=mysql_fetch_array($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_fetch_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", $conexao);

$inserte="Insert into alunos (id_aluno, nome, endereço, turma, turno)
Values(1, 'Glaucio', 'Av. das Américas', '1101', 'manhã')";
$inserte1="Insert into alunos (id_aluno, nome, endereço, turma, turno)
Values (2, 'Alexandre', 'Av. das Américas', '1101', 'tarde')";

```

```
mysql_query($insere, $conexao)
or die ("Não foi possível executar a inserção.");
mysql_query($inserel, $conexao)
or die ("Não foi possível executar a inserção.");

$consulta="Select Id_Aluno, nome, turno from alunos";
$resultado=mysql_query($consulta,$conexao);

$dados=mysql_fetch_array($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 está preocupado em quanta memória está sendo usada para query num grande conjunto de resultados. Toda a memória usada do resultado é liberada automaticamente ao final da execução do script.