**Manutenção de Registros em Tabelas de Banco de Dados (HTML/PHP)**

A manutenção de registros em tabelas de banco de dados compreende 4 procedimentos básicos:

* Inclusão (ou inserção) de novos registros;
* Alteração (ou edição) de registros;
* Exclusão de registros;
* Consulta de registros;

Onde se entende por registro o conjunto de campos que compõe uma linha de uma tabela de um banco de dados. Por padrão um desses campos deve ser definido como chave primária (campo cujo valor não se repete para mais de um registro e usado como dado de acesso ao registro). A chave primária geralmente é um campo numérico inteiro (ID ou CODIGO) mas também pode ser definida como outra informação (CPF, RG, e-mail, login, etc...)

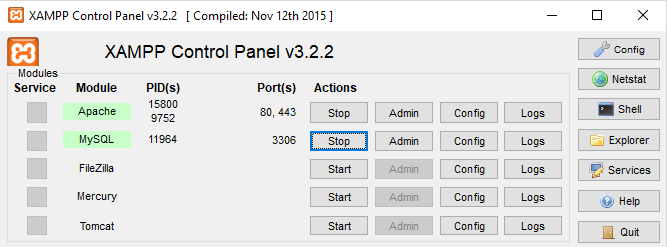
Na operação de inclusão de novos registros em caso de chave primária definida como campo numérico inteiro esse valor deverá ser gerado pelo software gerenciador de banco de dados ou pelo próprio programa. Se for usado outro tipo de dado o programa deverá verificar se esse valor já existe ou não antes de efetuar a operação.

Nas operações de alteração e exclusão deve-se utilizar o valor do campo chave primária para acessar o registro antes de efetuar a operação. Daí a necessidade de uma rotina de consulta de dados.

No exemplo trabalhado nessa apostila usaremos uma tabela com campos básicos e utilizaremos os comandos e recursos mais simplificados possíveis das linguagens HTML e PHP sem a utilização de recursos mais sofisticados como scripts e sem a preocupação com acabamento e layout.

Para emular os serviços de um servidor web para hospedagem, acesso e execução dos scripts de PHP utilizaremos o software XAMPP. Os arquivos gerados deverão estar gravados em uma pasta no diretório C:\xampp\htdocs, o banco de dados e sua tabela criados a partir dos recursos phpmyadmin (localhost/phpmyadmin) e a programação acessada a partir que qualquer navegador web abertos em localhost/<nome\_pasta>.

Para iniciar execute o Xampp a e a partir do seu painel de controle (XAMPP Control Panel) coloque em execução os serviços Apache (para execução dos scripts PHP) e MySQL (para execução do PHPMyadmin).



Em seguida crie uma pasta chamada cad em C:\xampp\htdocs. Nesta pasta iremos salvar todos os arquivos HTML e PHP. O acesso será via navegador web em localhost\cad.

Criação da Base de Dados e Tabela: usaremos nesse exemplo uma base de dados chamada bdcadastro que terá uma tabela chamada tbcadastro com os seguintes campos:

id int (4) auto increment primary key

nome varchar(40)

cidade varchar(35)

telefone varchar(15)

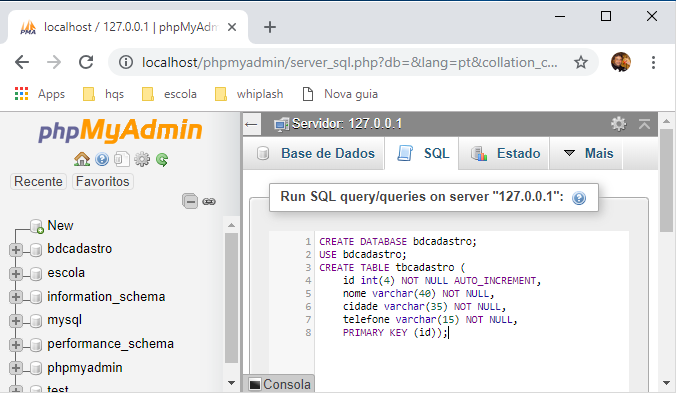
Para criar a base de dados e a tabela devemos acessar localhost/phpmyadmin no navegador web e utilizar os botões de atalho ou executar o código SQL de criação na aba SQL da página:

CREATE DATABASE bdcadastro;

USE bdcadastro;

CREATE TABLE tbcadastro (id int(4) NOT NULL AUTO\_INCREMENT,

nome varchar(40) NOT NULL, cidade varchar(35) NOT NULL, telefone varchar(15) NOT NULL, PRIMARY KEY (id));



Com a base de dados e tabela criados podemos iniciar a criação dos arquivos com os códigos HTML e PHP que permitirão a manutenção dos registros dessa tabela. Todos os arquivos deverão estar gravados na pasta c:\xampp\htdocs\cad e serão abertos via navegador web em localhost/cad.

Novos comandos (funções) PHP para acesso e manutenção de registros em tabelas de bancos de dados:

**mysqli\_connect** (nome\_servidor, login, senha\_login, nome\_bd)

Estabelece uma conexão com banco de dados. A partir dessa conexão temos acesso às tabelas do banco de dados e seus registros. Essa função retorna true (verdadeiro) se a conexão foi estabelecida com sucesso ou false (falso) no caso da conexão falhar. Deve-se associar o resultado da conexão à uma variável que será usada posteriormente no acesso aos registros. Exemplo:

$conecta = mysqli\_connect ("localhost","root","","bdcadastro");

if ($conecta == false)

{

echo "Problemas na conexão....";

exit;

}

else

{

echo "Conexão ok....";

}

**mysqli\_query** (conexão, instrução\_sql)

Executa uma instrução SQL (insert, delete, update, select) que será implementada nas tabelas e registros configurados na instrução. A conexão será o nome da variável que recebeu a conexão realizada pelo mysqli\_connect. Também retorna true ou false dependendo do sucesso ou fracasso da execução da instrução. Exemplo:

$sql = "SELECT \* FROM tbcadastro";

$consulta = mysqli\_query ($conecta, $sql);

if ($consulta == false)

{

echo "Erro ao realizar a consulta";

exit;

}

**mysqli\_fetch\_assoc**(nome\_variável\_consulta)

Utilizado após uma operação de consulta (query) bem sucedida essa função gera um vetor com um elemento para cada campo do registro selecionado na consulta. O nome do vetor é usado para receber os dados deve ser definido como variável que recebe o resultado da execução da função.

Exemplo:

$consulta = mysqli\_query ($conecta,$sql);

Onde: $conecta contém o retorno de mysqli\_connect e $sql contém a consulta SQL

Supondo que essa pesquisa retornou o registro da linha 7 da tabela:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linha | id | nome | cidade | telefone |
| 7 | 33 | Carlos | Sta Cruz | 1111-1111 |
| 8 | 72 | Luciana | Ourinhos | 2222-1111 |

A linha de comando: $dados = mysqli\_fetch\_assoc($consulta)

gera um vetor $dados com 4 elementos onde o índice é o nome do campo e o conteúdo é o conteúdo daquele campo:

$dados [“id”] = 33

$dados [“nome”] = Carlos

$dados [“cidade”] = Sta Cruz

$dados [“telefone”] = 1111-1111

Após a criação do vetor o foco passa automaticamente para o próximo registro (no exemplo linha 8). Por isso essa função é utilizada como condição em um laço de repetição (WHILE) para que todos os registros retornados pela consulta possam ser trabalhados. Esse procedimento é considerado padrão para rotinas de consulta de dados, onde todos os registros selecionados devem ser exibidos:

while ( $dados = mysqli\_fetch\_assoc($consulta))

{

echo $dados['id'] . "<br>";

echo $dados['nome'] . "<br>";

echo $dados['cidade'] . "<br>";

echo $dados['telefone'] . "<br>";

echo "<hr>";

}

**mysqli\_affected\_rows** (conexão, instrução\_sql)

Retorna o número de linha (registros) afetado (acessado) por uma instrução SQL em uma conexão de banco de dados. Utilizado para verificar se um registro foi encontrado (linhas afetadas=1) ou não (linhas afetadas=0).

Exemplo:

$linhas = mysqli\_affected\_rows($conecta);

if ($linhas == 0)

echo "Codigo nao cadastrado";

else

echo "Codigo cadastrado";