Consultas preparadas

Cada vez que se envía unha consulta ao servidor, este debe analizala antes de executala. Algunhas sentenzas SQL, como as que insiren valores nunha táboa, deben repetirse de forma habitual nun programa. Para acelerar este proceso, MySQL admite consultas preparadas. Estas consultas almacénanse no servidor listas para ser executadas cando sexa necesario e presentan as seguintes vantaxes:

- *Optimización*: xa que reducen o gasto de recursos na análise e execución de cada consulta.
- *Seguridade*: xa que ofrecen máis seguridade ante posibles inxeccións de SQL.

Para traballar con consultas preparadas coa extensión MySQLi de PHP empregando a interface orientada a obxectos, debes utilizar a clase `mysqli_stmt`. Utilizando o método `stmt_init` da clase mysqli obtense un obxecto da devandita clase.

Os pasos que debes seguir para executar unha consulta preparada coa interface orientada a obxectos son:

- Preparar a consulta no servidor MySQL utilizando o método `prepare`.
- Executar a consulta, tantas veces como sexa necesario, co método `execute`.
- Unha vez que xa non se necesita máis, débese executar o método `close`.

Imos ver un exemplo:

```
$consulta = $db->stmt_init();
$consulta->prepare('INSERT INTO PROVINCIA (codigo, nome)
                    VALUES ("50", "Zaragoza")');
$consulta->execute();
$consulta->close();
$db->close();
```

O problema é que de pouco serve preparar unha consulta de inserción de datos como a anterior, se os valores que insire son sempre os mesmos. Por este motivo as consultas preparadas admiten parámetros. Para preparar unha consulta con parámetros, en lugar de poñer os valores debes indicar cun signo de interrogación a súa posición dentro da sentenza SQL.

```
$consulta->prepare('INSERT INTO PROVINCIA (codigo, nome) VALUES
(?, ?) ');
```

E antes de executar a consulta tes que utilizar o método `bind_param` para substituír cada parámetro polo seu valor. O primeiro parámetro do método `bind_param` é unha cadea de texto na que cada carácter indica o tipo dun parámetro, segundo a seguinte táboa:

CARACTER	TIPO DO PARÁMETRO
I	Número enteiro

D	Número real (dobre precisión)
S	Cadea de texto
B	Contido en formato binario (BLOB)

Por exemplo:

```
$consulta = $db->stmt_init();
$consulta->prepare('INSERT INTO PROVINCIA (codigo, nome) VALUES
(?, ?)');
$codigo = "50";
$provincia = "Zaragoza";
$consulta->bind_param('ss', $codigo, $provincia);
$consulta->execute();
$consulta->close();
$db->close();
```

No caso das consultas que devolven valores temos que vincular as variables resultado da consulta preparada para almacenar o seu resultado. Temos dous métodos para extraer o resultado das columnas da consulta preparada:

- `bind_result`: vincula as columnas do resultado da columna coas variables que gardarán ese resultado.
- `fetch`: Permite obter os resultado desas variables, para o cal deberemos recorrer un bucle que permita obter os datos de todas as filas resultantes da consulta.

```
$consulta = $db->stmt_init();
$consulta->prepare('SELECT nome, dificultade, tempo
                    FROM receita WHERE tempo<50');
$consulta->execute();
$consulta->bind_result($receita, $dificultade, $tempo);
while($consulta->fetch()) {
   print "<p> $receita ( $dificultade ) -  $tempo minutos</p>";
}
$consulta->close();
$db->close();
```

Para empregar consultas preparadas coa extensión MySQLi de PHP empregando a interface procedemental usaremos a función `mysqli_stmt_init()`. Os pasos a seguir son os mesmos que se indicaron anteriormente, pero as funcións a empregar neste caso son: `mysqli_stmt_prepare`, `mysqli_stmt_execute` e `mysqli_stmt_close` respectivamente.

Imos ver un exemplo:

```
$sql = "INSERT INTO PROVINCIA (codigo, nome) VALUES (?, ?)";
$stmt = mysqli_stmt_init($db);
if(mysqli_stmt_prepare($stmt,$sql)){  //para comprobar erros no
prepare
```

A función para substituir cada parámetro polo seu valor é

`mysqli_stmt_bind_param`:

```
mysqli_stmt_bind_param($stmt, 'ss', $codigo, $provincia);
$codigo = "50";
$provincia = "Zaragoza";
mysqli_stmt_execute($stmt);
$codigo = "38";
$provincia = " Santa Cruz de Tenerife";
mysqli_stmt_execute($stmt);
mysqli_stmt_close($stmt);
} //fin do if do prepare
```

E para para asignar a variables os campos que se obteñen tras a execución temos `mysqli_stmt_bind_result`, usando `mysqli_stmt_fetch()` para recorrelos:

```
$db = mysqli_connect("localhost", "alumno", "abc123.",
"receitas");
 $sql = "SELECT nome, dificultade, tempo FROM receita WHERE
tempo<50";
$stmt = mysqli_stmt_init($db);
if(mysqli_stmt_prepare($stmt,$sql)){
   mysqli_stmt_execute($stmt);
   $receita = "";
   $dificultade = "";
   $tempo = 0;
   mysqli_stmt_bind_result($stmt,$receita, $dificultade,
$tempo);
   while (mysqli_stmt_fetch($stmt)) {
      print "<p> $receita ( $dificultade ) -  $tempo
minutos</p>";
   }
   mysqli_stmt_close($stmt);
   $db->close();
}
```

Tarefa: Realización de operacións con consultas preparadas

Modifica a tarefa de Transaccións para que empregue consultas preparadas cando o consideres necesario.