**The MySQL Methods in Spine**

**Calling a Query:**

The MySQL class has been improved on to allow for prepared statements in Spine. You can prepare a statement similar to how you do in MySQLi.

$this->DB->prepare(“SELECT \* FROM test WHERE id=’?’”);

The prepare method can take the binding variables in its 2nd argument, however if you wish to bind variables later on then you can use bind().

$this->DB->bind(array(“i”, 5));

Unlike MySQLi, this bind method needs to take its argument as an array.

Once data is bound to the query, the query can be executed:

$this->DB->execute();

This will help to prevent SQL Injection.

To retrieve data, it can be done a few ways.

Most simply:

$data = $this->DB->fetch();

That will populate the $data variable with an array of the fetched results. Of course, if there are more then 1 results you’ll need to run it in a while loop.

If you want to specify key names for your array, you can add in key names in order of column names as arguments in the fetch method.

$data = $this->DB->fetch(‘id’,’username’);

Now you can access each property at $data[‘id’], or $data[‘username’] etc.

A cleaner method is to use PHPs extrat().

extract($this->DB->fetch());

Again, you can use your own key names in the fetch method.

One last thing, if you are not sure if there will be any results, you can call fetch() in an if statement, then again after the if statement. Calling fetch twice will not invoke 2 queries, the 2nd call will simply use the data returned by the first call.

if($this->DB->fetch()){  
//call fetch again  
}

Instead of preparing the same querying, you can just bind different values to the current query:

$this->DB->prepare("SELECT name FROM test WHERE id='?'");

$this->DB->bind(array('i', 3));

$this->DB->execute();

if($this->DB->fetch()){

extract($this->DB->fetch('username'));

echo $username;

}

$this->DB->bind(array('i', 2));

$this->DB->execute();

if($this->DB->fetch()){

extract($this->DB->fetch('username'));

echo $username;

}