کار با Data Object ها در php5-

امروزه کار با database یکی از مسائل روزمره برنامه نویسان وب به شمار می آید،از یک پردازش ساده فرم گرفته تا برنامه های کاربردی در مقیاس بزرگ(large-scale). تقریبا در اکثر موارد از database استفاده می کنیم. بعد از کارکردن روی چند پروژه، زمان زیادی طول نخواهد کشید که متوجه می شوید،چهار عمل اصلی ساده در کار با database به طور مرتب تکرار می شوند.این اعمال عبارتند از:

- پیداکردن و انتخاب رکوردها(**SELECT**)
- به روزرسانی رکوردهای جاری(**UPDATE**)
- افزودن رکوردهای جدید(**INSERT**)

- حذف رکوردها(**DELETE**)

شما در جاهای مختلف برنامه مجبورید که یک Query را بازنویسی کرده و یا آن را از قسمت های قبلی برنامه Copy-Paste کرده و سپس تغییرات لازم را روی آن اعمال کنید.عده ای از برنامه نویسان ترجیح می دهند که از یک لایه انتزاعی(Abstraction Layer) مثل **PEAR::DB** و یا **DBX** استفاده کنند.استفاده از یک لایه انتزاعی چیز خوبی است اما هدف اصلی در آن ،این است که RDMBS را محو کند یا به اصطلاح Transparent سازد و برنامه شما مستقل از RDMBS شود.یک لایه انتزاعی، راهی برای مجردسازی((abstraction ساختار database شما ازائه نمی دهد. اینجاست که یک لایه دسترسی مجزا می تواند مفید واقع شود.شما می توانید لایه دسترسی داده دا از راههای مختلف پیاده سازی کنید، اما در این مقاله، توجه ما معطوف به

Data Object ها می باشد. مفهوم Data Object برگرفته از

الگوهای طراحی به خوبی مستند شده(very well-documented

patterns design) به نام Data Access Object و الگوهای شیءِ

انتقال(Transfer Object) است.این الگوها می توانند بسیار پیچیده

باشند،اما با کمی تفکر می توانیم به راحتی از ایده ها و اهداف پایه ای

آنها در php5 استفاده کنیم.

×یک شیء داده(Object Data) چیست؟

همانطور که ذکر شد، ایده Object Data برگرفته از یک الگوی

طراحی است.اگر نگاهی به الگوهای طراحی کنید، متوجه می شوید که

تقریبا اغلب آنها بر مبنای شیءِگرایی (orientation object

)هستند. در این جا می خواهیم از مدل شی ای جدید php5 استفاده

گسترده ای داشته باشیم.همچنین برای مثالهای خود از MySQL

استفاده می کنیم.

در اصل،هر Data Object یک کلاس کدنویسی شده است که یک

table را در database شما به طور مستقیم مشخص می کند.برای

هر table یک کلاس خاص می نویسیم.کلاس،شامل متغیرهای

عضو(مطابق با فیلدهای جدول) و مجموعه ای از توابع یا متدها(حداقل

برای انجام ٤ عمل اصلی ذکرشده)می باشد.فرض کنید یک جدول

ساده برای نگهداری مشخصات کاربر داریم:

```
TABLE: User
userId  INT
firstName VARCHAR(30)
lastName  VARCHAR(40)
email   VARCHAR(100)
```

برای این جدول کلاسی تعریف می کنیم که شامل متغیرهای عضو

مطابق با فیلدهای جدول باشد.بهتر است که برای Data Object خود

یک پیشوند بگذاریم(مثل DO_)تا آن را با کلاسهای هم نام دیگر اشتباه

نگیریم.(این یک الگوی رایج در php برای شبیه سازی namespae ها

می باشد)

```
class DO_User {
  public $userId;
  public $firstName;
  public $lastName;
  public $email;
}
```

مثال بالا یک سطر ساده از جدول را نشان می دهد.اگر یک شیء

Do_user را نمونه سازی(instantiate) کنیم،می توانیم یک سطر از

داده ها را نگهداری کنیم.حالا چطور اطلاعات را از database واکشی

کرده و در این شیءِ ذخیره کنیم؟ برای این کار یک متد جدید به نام

()get به کلاس خود اضافه می کنیم تا از database برای یک کاربر

خاص Query بگیریم. ما از userID(کلید اولیه) هر کاربر به عنوان

پارامتر استفاده می کنیم:


```
class DO_User {
  public $userId;
  public $firstName;
  public $lastName;
  public $email;

  // This function will perform a select
on the table looking for
    // a specific userId.
  public function get($userId)
  {
    $sql = 'SELECT * FROM User WHERE
```

```
userId='
                .
mysql_escape_string($userId);
     $rs   = mysql_query($sql);
     $row = mysql_fetch_array($rs);

     $this->userId  = $row['userId'];
     $this->firstName =
$row['firstName'];
     $this->lastName  = $row['lastName'];

     $this->email    = $row['email']
   }
}
```

حالا با استفاده از این Data Object می توانیم با database خود

تعامل داشته باشیم.هیچ نیازی با استفاده از SQL نیست.مثال زیر

اطلاعات مربوط به یک کاربر را دریافت و در Web Browser نمایش

می دهد:


```
<?php
include_once('class-DO_User.php');
$user = new DO_User();

// We'll use a literal integer here,
// but this could come from anywhere,
// such as $_POST or $_GET
$user->get(5);

?>
<html>
  <head>
    <title>User Info</title>
  </head>
  <body>
```

```
<p>Here is the user info:</p>

<table border="1">

  <tr>

    <td>User ID</td>

    <td><?=$user->userId?></td>

  </tr>

  <tr>

    <td>First Name</td>

    <td><?=$user->firstName?></td>

  </tr>

  <tr>

    <td>Last Name</td>

    <td><?=$user->lastName?></td>

  </tr>

  <tr>

    <td>Email</td>

    <td><?=$user->email?></td>

  </tr>

</table>
```

```
    </body>
</html>
```

همانطور که مشاهده می کنید استفاده از یک **Data Object** بسیار ساده است. تابع ()**get** ای که اضافه کردیم یک **Query** ساده را انجام می دهد و یک کلید اولیه مشخص(**userID**) جستجو می کند. توجه کنید که چون از فیلد با کلید اولیه استفاده کردیم، تنها یک رکورد از داده ها را دریافت می کنیم و مقادیر آنها را در متغیرهای عضو کلاس قرار می دهیم.در قسمت های بعدی در مورد دریافت چند سطر از رکوردها نیز بحث خواهیم کرد.