

ORM

Object Relational Mapping

Introductions aux Bases de Données
Nathanaël Martel

ORM : PDO

Dans le cadre d'une application programmé avec des classes, les enregistrements de la base correspondent souvent à des objets. Par exemple une table utilisateur :

```
CREATE TABLE `user` (  
    `id` int NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    `firstname` varchar(255) NOT NULL,  
    `lastname` varchar(255) NOT NULL  
);  
  
INSERT INTO `user` (`firstname`, `lastname`) VALUES  
('John', 'Doe');
```

ORM : PDO

Et une classe associée :

```
class User {  
    private $id, $firstname, $lastname;  
  
    public function getId() {  
        return $this->id;  
    }  
    public function getFirstname() {  
        return $this->firstname;  
    }  
    public function getLastName() {  
        return $this->lastname;  
    }  
    public function getFullname() {  
        return $this->firstname.' '.$this->lastname;  
    }  
}
```

ORM : PDO

Il est possible de récupérer des objets, instances de cette classe :

```
foreach ($pdo->query('SELECT * FROM user', PDO::FETCH_CLASS, 'User') as $User) {
    var_dump($User);
    echo $User->getFullname();
}
object(User)#5 (3) {
    ["id":"User":private]=>
    string(1) "1"
    ["firstname":"User":private]=>
    string(4) "John"
    ["lastname":"User":private]=>
    string(3) "Doe"
}
John Doe
```

ORM : PDO

Avec un peu plus de PHP, Il est possible de faire de la création d'objet :

```
$user = new User();  
$user->firstname = 'Denis';  
$user->lastname = 'Dideros';  
$user->save(); → insert in database
```

Doctrine

Select avec Doctrine :

```
$users = Doctrine::getTable('users')->findAll();  
    foreach($users as $user) {  
        echo "<h2>$user->name</h2>";  
    }  
}
```

Avec une condition :

```
function returnDescriptionDeCategorie ($id) {  
    $q = Doctrine_Query::CREATE()  
    ->select('description')  
    ->from('Categorie c')  
    ->where('c.id_categorie = ?', $id);  
    return $q->fetchOne();  
}
```

Doctrine

Insert avec Doctrine :

```
$user = new User();  
$user->name = "john";  
$user->password = "doe";  
$user->save();  
  
echo "L'utilisateur avec l'id " . $user->id .  
" a été ajouté";
```

Doctrine

Update avec Doctrine :

```
$user = Doctrine::getTable('user')-  
>findOneById(2);  
$user->phone = "+33 1 23 45 67 89";  
$user->save();
```


Doctrine

Delete avec Doctrine :

```
$user = Doctrine::getTable('user')-  
>findBySlug("kevin-smith");  
  
$user->remove();
```

Java Hibernate

Pour Java, il existe un ORM similaire :

- **Java Hibernate**

Conclusion

Il est possible de manipuler les enregistrements comme des objets.
Le driver de la base de données va fournir les outils pour cela.