

# **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 :

```
$results = $pdo->query(
    'SELECT * FROM user',
    PDO::FETCH_CLASS,
    'User'
);
```

# ORM : PDO

```
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

# Doctrine

## Select avec Doctrine :

```
$users = $em->getRepository('User')->findAll();
```

```
foreach($users as $user) {  
    echo "<h2>$user->name</h2>";  
}
```

## Avec une condition :

```
$users = $em->getRepository('User')->findBy(array('enable' =>  
1));
```

```
$user = $em->getRepository('User')->findOneBy(array('id' => 1));
```

# Doctrine

## Insert avec Doctrine

```
$user = new User();  
$user->firstname = 'Denis';  
$user->lastname = 'Dideros';  
  
$em->persist($user); // L'objet doit est utilisé comme une  
entité dans la base de données  
$em->flush(); // Exécution des requêtes SQL  
  
echo $user->getId();
```

# Doctrine

## Update avec Doctrine :

```
$user =  
$em->getRepository('User')->findOneBy(array('id' =>  
1)); // Récupération d'un utilisateur
```

```
$user->setPhone("+33 1 23 45 67 89");
```

```
$em->persist($user);
```

```
$em->flush();
```



# Doctrine

## Delete avec Doctrine :

```
$user =  
$em→getRepository('User')→findOneBy(array('id'  
' => 1)); // Récupération d'un utilisateur  
  
$em→remove($user);  
  
$em→flush();
```

# ***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.