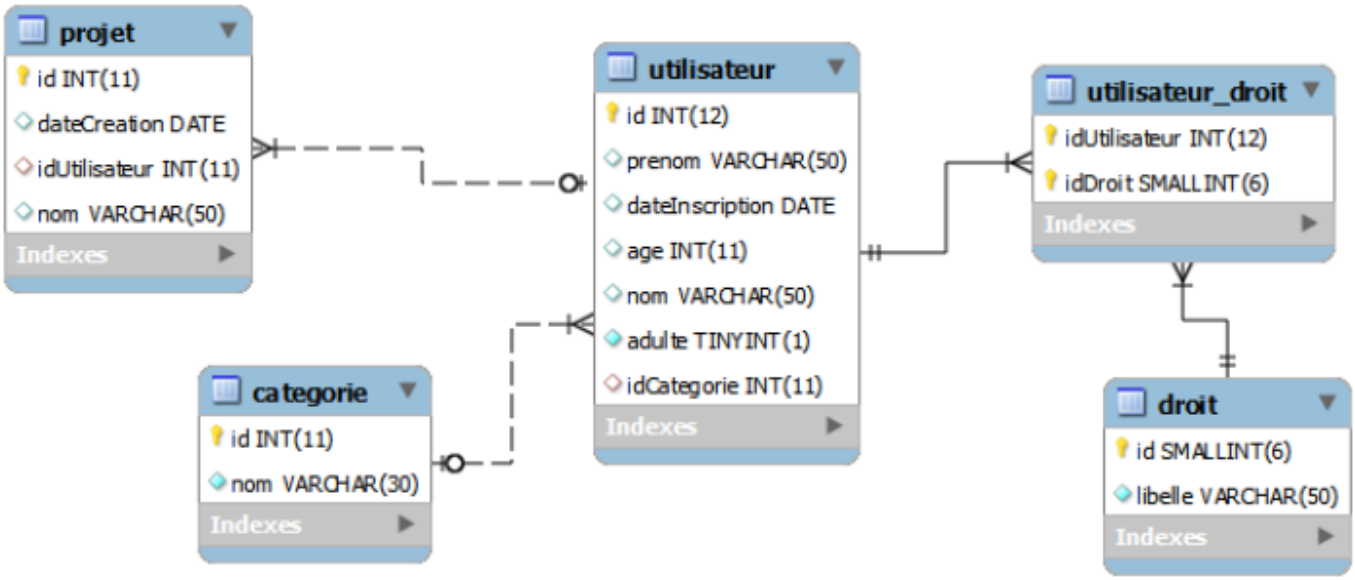
Accès aux données en PHP

Sources

1. slam1td3.zip

-- Base de données

Dans notre exemple, le modèle de données relationnel est le suivant :



- **Utilisateurs et projets :**
  - Chaque projet est la propriété d'un utilisateur (Projet belongsTo → Utilisateur)
  - Un utilisateur peut être le propriétaire de plusieurs projets (Utilisateur hasMany → Projets)
- **Utilisateurs et Catégories :**
  - Chaque utilisateur appartient à une categorie (Utilisateur belongsTo → Categorie)
  - Dans une catégorie, on peut avoir plusieurs Utilisateurs (Categorie hasMany → Utilisateurs)
- **Utilisateurs et Droits :**
  - Chaque Utilisateur dispose de Droits (Utilisateur hasAndBelongsToMany → Droits)
  - Chaque Droit peut-être affecté à plusieurs Utilisateurs

-- Création de la base de données

- Télécharger la base de données
- Exécuter le script de création de la base de données **slam1Td3** téléchargé sur votre serveur Mysql

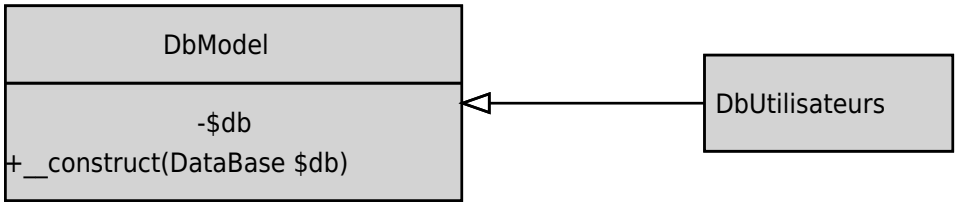
-- Modèle conceptuel

- Élaborer le diagramme de classes UML des classes métier correspondant à la BDD

-- Manipulations de données

-- Contraintes à respecter

- Utiliser la classe **DataBase** pour l'accès à la base de données : database.php
- Créer les classes métier dans le dossier **classes**
- Les opérations entre le modèle Objet et la base de données doivent obligatoirement être implémentées dans des classes dédiées (à créer)
- Le code permettant de générer l'affichage d'objet doit être implémenté dans la classe **GUI** (à créer), via des méthodes statiques



PLEASE MIGRATE

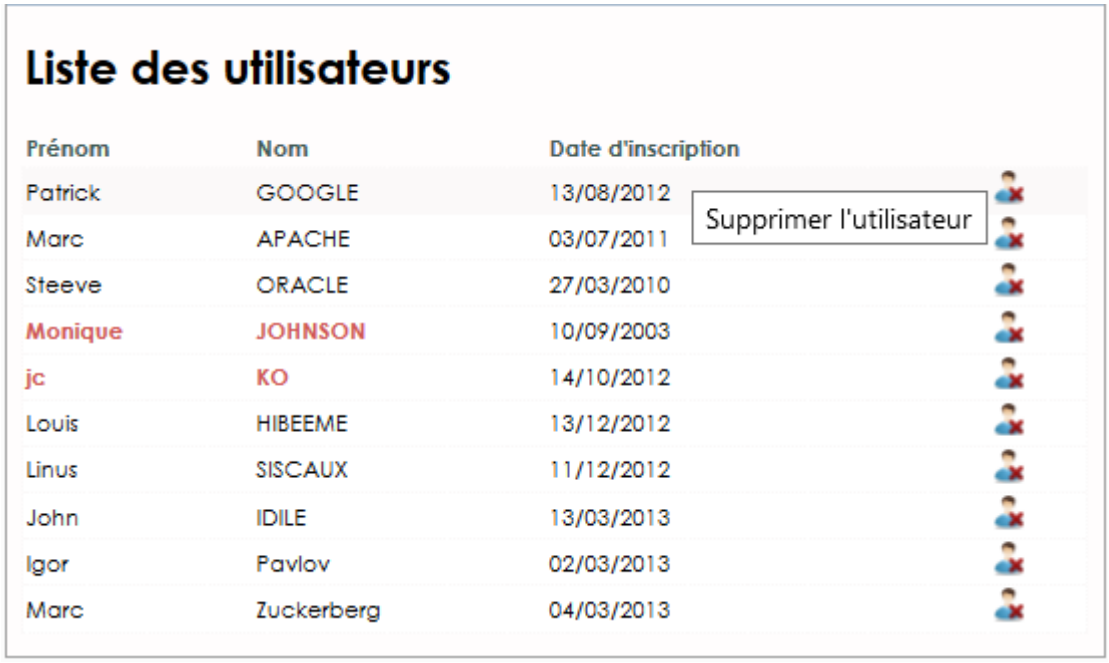
-- Liste des Utilisateurs

1. Créer la classe métier **Utilisateur**, en faisant correspondre les membres de la classe aux champs de la table.
2. Créer la méthode **getAll()** dans la classe **DbUtilisateurs**, retournant un tableau d'instances de la classe Utilisateur correspondant aux enregistrements de la table
3. Créer la page **listUtilisateurs.php** utilisant la méthode getAll et affichant la liste des utilisateurs.

Prénom	Nom	Date d'inscription
Patrick	GOOGLE	13/08/2012
Marc	APACHE	03/07/2011
Steeve	ORACLE	27/03/2010
Monique	JOHNSON	10/09/2003
jc	KO	14/10/2012
Louis	HIBEEME	13/12/2012
Linus	SISCAUX	11/12/2012
John	IDILE	13/03/2013
Igor	Pavlov	02/03/2013
Marc	Zuckerberg	04/03/2013

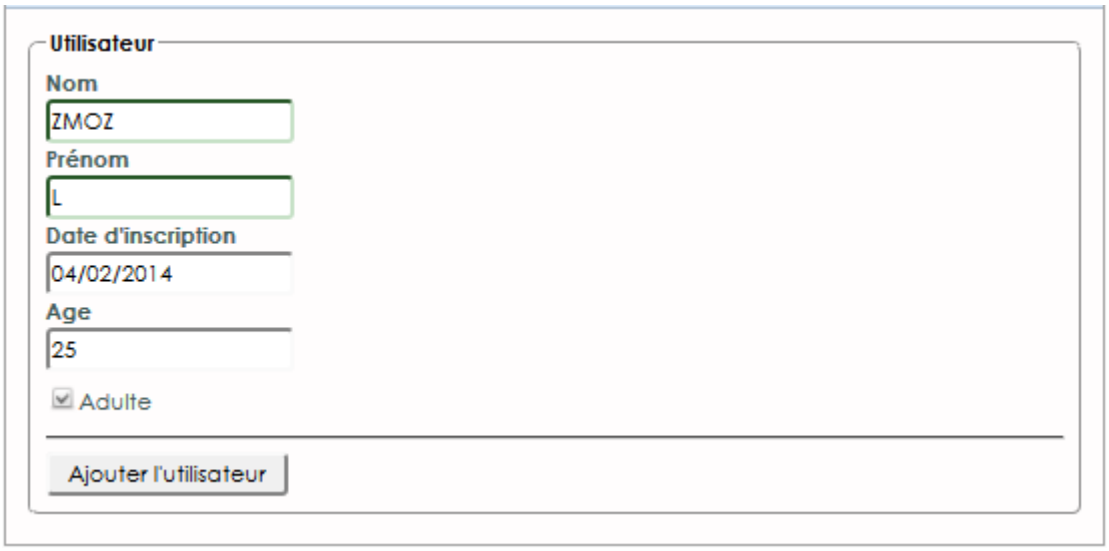
-- Liste et suppression

1. Créer la méthode **delete($id)** dans la classe **DbUtilisateurs**, supprimant l'utilisateur de clé primaire $id
2. Créer une page utilisant cette méthode pour supprimer l'utilisateur dont l'id est passé dans la requête (GET), et affichant le résultat de la suppression
3. Ajouter un lien (image) pour la suppression dans la liste



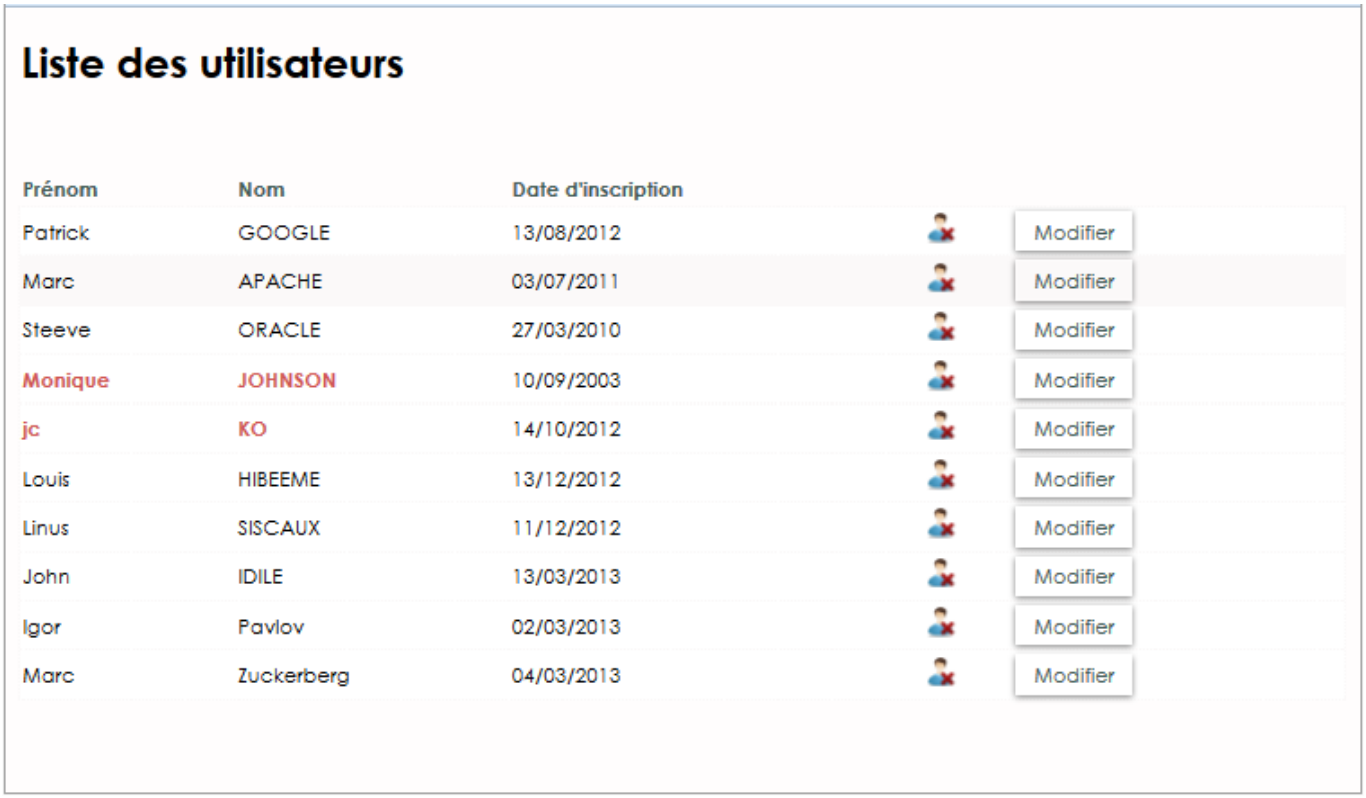
-- Ajout

1. Créer la méthode **addNew($user)** dans la classe **DbUtilisateurs**, ajoutant l'utilisateur $user
2. Créer le formulaire de saisie d'un utilisateur
3. Créer une page d'ajout d'utilisateur traitant la soumission du formulaire et réalisant l'ajout (à partir de la méthode addUtilisateur)



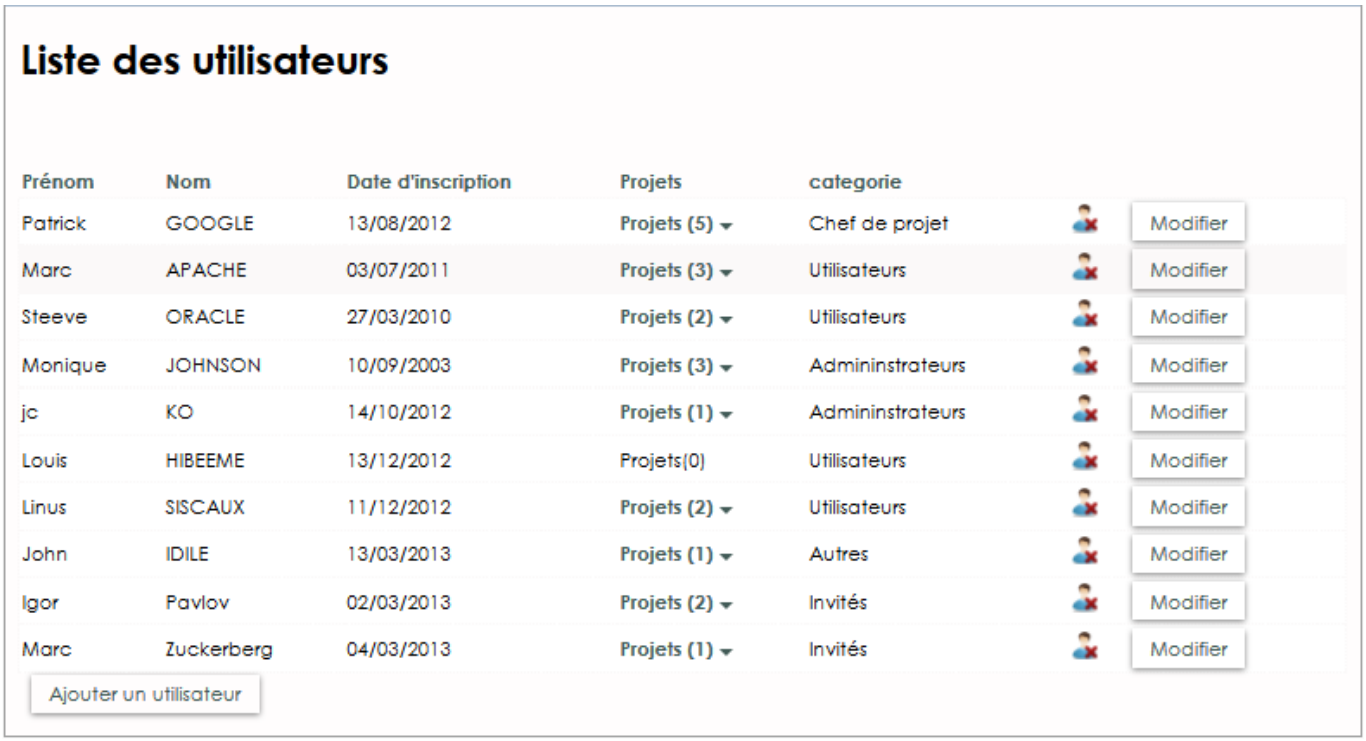
-- Modification

1. Créer la méthode **update($user)** dans la classe **DbUtilisateurs**, modifiant l'utilisateur $user
2. Créer le formulaire affichant les informations de l'utilisateur dont l'id est passé dans la requête, par clic sur le bouton Modifier de la liste des utilisateurs
3. Créer une page de modification d'utilisateur traitant la soumission du formulaire et réalisant la modification (à partir de la méthode update)



-- Chargement des objets associés

1. Modifier la liste pour qu'elle affiche le libellé de la catégorie de l'utilisateur, et le nombre de projets



From:
http://slamwiki2.kobject.net/ - **SlamWiki 2.1**

Permanent link:
http://slamwiki2.kobject.net/slam1/td4?rev=1519034838

Last update: **2019/08/31 14:38**