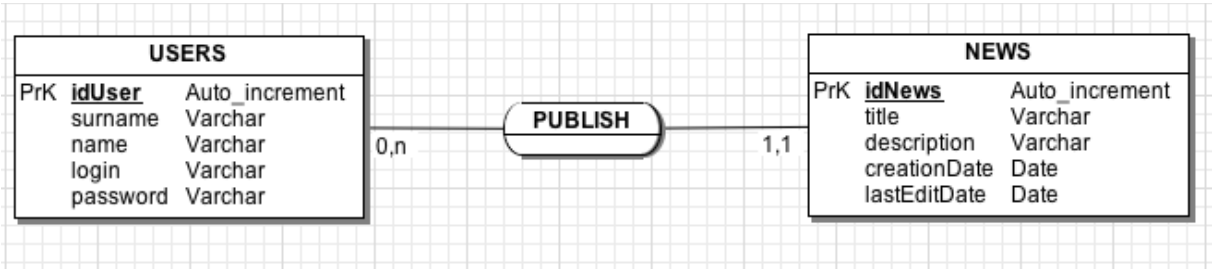
EE, Révision – Forum - Chapitre 4

On peut maintenant poster des nouvelles, c'est bien, mais cela serait encore mieux si nous pouvions les éditer et les supprimer si besoin !

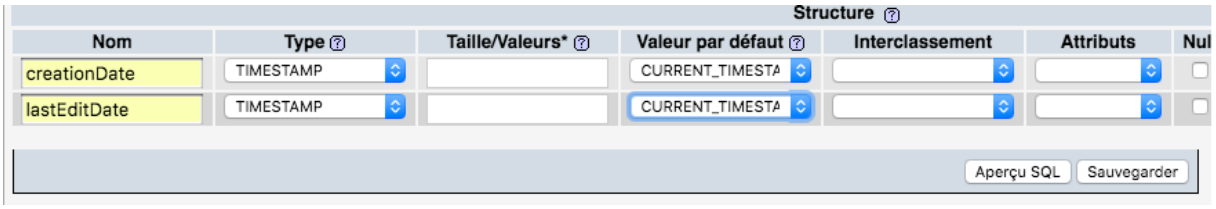
Nous allons également rajouter quelque informations utiles en base pour commencer:

- La date de création du post
- La date de dernière modification du post

Voici donc le MCD mis à jour :



Mettez à jour votre MLD et rajoutez donc les 2 colonnes de la façon suivante dans votre base (n'oubliez pas la valeur par défaut !):



Il ne vous reste plus qu'à ajouter ces informations à l'affichage (cf plus bas pour le détail). Comme nous savons déjà quel est l'auteur du post, nous pouvons directement faire figurer cette information. Cela sera utile étant donné que cette application est destinée à un usage multi-utilisateur. Nous laisserons la possibilité à l'auteur (et rien qu'à l'auteur) d'éditer et de supprimer ses posts.

Nous allons quand même rajouter un peu de mise en page pour casser cet effet « brut de décoffrage ». Vous avez reçu un fichier css contenant le code nécessaire. Utilisez le afin que :

- Tous les messages d'erreurs utilisent l'# « warning » et soient en rouge
- Tous les autres avertissement utilisent l'# « info» et soient en vert
- Chaque post bénéficie de la mise en page de l'# « post »

Il vous faudra donc lier ce fichier et mettre les bons id au bon endroit !

Voilà plus ou moins à quoi devra ressembler votre page principale à la fin de ce chapitre :

Vous remarquez ici que les posts de la personne connectée contiennent des liens de modification et de suppression mais pas les autres. Pour vous aider, voici comment vous pouvez vous y prendre dans la boucle d'affichage de vos posts :

Tout d'abord vous devez récupérer les informations de l'utilisateur connecté. La fonction à développer est la suivante :

```
/**
 * Renvoie les détails d'un utilisateur
 * @param string id de l'utilisateur
```

```
 * @return array {login; surname; name} ou false si non valide
 */
function getUserById($id) {
}
```

Une fois les infos à disposition, ajoutez le nom et prénom comme dans l'exemple sur la première ligne. Puis pour afficher les dates c'est un peu plus compliqué. La date retournée par mysql n'est pas par défaut dans le format accepté par la fonction date de php. Il faut auparavant la convertir (cf ressources). Passez le bon format également à la fonction date pour l'afficher tel que dans l'exemple.
L'affichage du titre et de la description ne change pas par rapport à votre version actuelle. Finalement, le gros du travail se situe sur la modification et la suppression. Pour afficher les liens uniquement sur les posts de l'utilisateur connecté vous pouvez vous baser sur l'identifiant de l'utilisateur stocké en session que vous allez comparer à celui de l'utilisateur du post que vous avez récupéré en base. Et voilà, il ne vous reste plus qu'à mettre les liens sur les 2 pages que nous allons voir un peu plus bas ! N'oubliez pas de passer en get l'identifiant du post (cf ressources) .

Pour les 2 nouveaux scripts, vous avez à disposition des exemples de Larry Ullmann qui est un gourou du développement web (difficile de faire mieux que lui). Je vous conseille donc de vous en inspirer dans le fonctionnement global, regardez bien son code vous allez voir que vous pourrez réutiliser beaucoup de choses ! Attention cependant car il n'utilise pas PDO mais mysqli, ne vous mélangez pas les pinceaux avec ce qui est de l'accès à la base de données ! Ci-dessous quelques informations supplémentaires sur les choses à faire dans ces 2 pages.

updateNews.php

Vous devrez récupérer les informations sur le post à modifier. La seule information que vous avez est l'identifiant du post, voici donc la méthode à développer :

```
/**
 * Retourne le post qui possède l'id passé en paramètre
 * @param int $id L'identifiant du post recherché
 * @return array Les données du post
 */
function getPostById($id) {
}
```

Une fois que l'utilisateur valide la modification n'oubliez pas de refaire les mêmes vérifications que lors de l'insertion. Bloquez la modification et affichez les erreurs dans ce cas !

Pour enregistrer vous allez également devoir développer une nouvelle fonction que vous ajouterez comme les autres à votre fichier prévu pour :

```
/**
 * Met à jour un post en base
 * @param int $idPost Id du post à mettre à jour
 * @param string $title Titre du post
 * @param string $description Description du post
 * @return boolean si tout ok ou pas
 */
function updatePost($idPost, $title, $description) {
}
```

Attention n'oubliez pas de rendre le formulaire « sticky » ! Voici à quoi cela doit ressembler :

deleteNews.php

Ici vous allez aussi utiliser votre fonction « GetPostById ». Affichez un petit formulaire tout simple demandant confirmation à l'utilisateur tel que ci-dessous :

Suppression d'une nouvelle

Etes-vous sûr de vouloir supprimer le post intitulé: "My first post++"?

Oui Non Valider

Retour

Vous avez maintenant besoin d'une nouvelle fonction, la voici :

```
/**
 * Supprime le post dont l'id est passé en paramètre
 * @param int $id l'identifiant du post à supprimer
 * @return boolean true si la suppression a eu lieu, false sinon
 */
function deletePost($id) {
}
```

Si l'utilisateur valide avec Non, un message tout simple s'affiche :

Le post n'a pas été supprimé.

Retour

S'il valide avec Oui, un message de confirmation s'affiche :

Le post a bien été supprimé!

Retour

Si vous êtes arrivé à ce point et que tout fonctionne vous commencez à vraiment maitriser ! Dans ce cas n'oubliez pas d'uploader votre travail sur Classroom (export BDD + votre site, le tout zipé) et passez à la suite !

Ressources :

- Transmettre des paramètres à travers une url : https://openclassrooms.com/courses/concevez-votre-site-web-avec-php-et-mysql/transmettre-des-donnees-avec-l-url

- Update SQL : http://sql.sh/cours/update
- Delete SQL : http://sql.sh/cours/delete
- Un exemple de gestion de la suppression : script « delete_user.php » de Larry Ullman
- Un exemple de mise à jour de données : script « edit_user.php » de Larry Ullman
- Fonctions « now » de mysql pour insérer des dates en base : https://www.w3schools.com/sql/func_mysql_now.asp
- Fonction « date » de php : http://php.net/manual/fr/function.date.php
- Fonction « strToTime » pour convertir une date str en timestamp Unix nécessaire à la fonction « date » : http://php.net/manual/fr/function.strtotime.php