

1 Base de données *MySQL* et script *PHP*

Initialisation de la base de données *MySQL*

Avant toute chose, il est nécessaire de créer et d'initialiser la base de données des photos. Pour cela connectez-vous à l'interface *phpmyadmin* : <http://localhost/phpmyadmin>
Créez ensuite un nouvel utilisateur avec les informations suivantes :

- Login : comweb_project
- Mot de passe : tcejorp_bewmoc_isen29
- Client : local

Cochez la case : « créer une base portant son nom et donner à cet utilisateur tous les privilèges sur cette base. ».

Chargez le contenu du script *SQL* « sql.sql » disponible dans les ressources du TP. Les modèles conceptuel et physique de données sont fournis dans la figure 1.

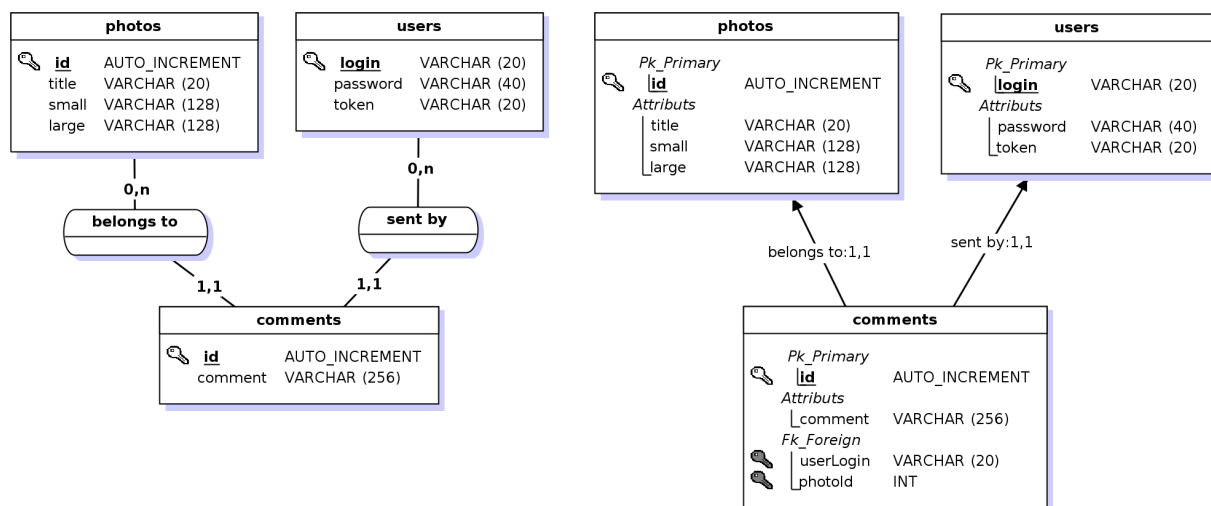


FIGURE 1 – Modèles conceptuel (à gauche) et physique (à droite) de données de la BDD mise en place.

Constantes de connexion à la base de données *MySQL*

Afin de se connecter à la base de données, il est nécessaire de définir les constantes de connexion. Pour cela, créez un script nommé « constantes.php » dans le dossier « php » et ajoutez-y les constantes de connexion suivantes :

- DB_USER qui vaut comweb_project;
- DB_PASSWORD qui vaut tcejorp_bewmoc_isen29;
- DB_NAME qui vaut comweb_project;
- DB_SERVER qui vaut localhost.

Accès à la base de données (connexion et requêtes) en *PHP*

Afin de récupérer dans la base de données les informations nécessaires à l'affichage des photos, créez un script nommé « database.php » dans le dossier « php ». Dans celui-ci, commencez par écrire la fonction nommée `dbConnect` permettant d'établir la connexion et de renvoyer le lien *PDO* nouvellement créé.

Ensuite, écrivez la fonction permettant de récupérer la liste des « petites » photos (c.-à-d. vignettes) :

```
dbRequestPhotos($db)
```

Où `db` est le lien *PDO* de la base de données. Cette fonction doit renvoyer la liste de toutes les photos avec les attributs `id` et `small` nécessaire pour l'affichage des vignettes.

Enfin, écrivez la fonction permettant de récupérer les informations d'une « grande » photo :

```
dbRequestPhoto($db, $id)
```

Où `db` est le lien *PDO* de la base de données et `id` l'identifiant de la photo souhaitée. Cette fonction doit renvoyer les informations de la photo demandée avec les attributs `id`, `title` et `large` nécessaire pour l'affichage de la photo.

2 Traitement des requêtes *AJAX* en *PHP*

Dans cette partie, il va être nécessaire de mettre en place le code *PHP* permettant de prendre en compte les requêtes suivantes côté serveur :

GET

php/request.php/photos/ Récupération des vignettes

GET

php/request.php/photos/i Récupération d'une photos

Connexion à la base de données en *PHP*

Ajoutez dans votre script « request.php » les instructions nécessaires à la connexion à la base de données en vous appuyant sur la fonction `dbConnect` du fichier « database.php ». Lors de cette tentative de connexion, renvoyez un statut 503 en cas d'erreur avant de quitter le script grâce à la fonction `exit` :

```
header('HTTP/1.1 503 Service Unavailable');
```

Étude de la requête du client en *PHP*

Afin de savoir ce que souhaite le client, il est nécessaire d'extraire les informations formulées en *REST* au travers de l'url utilisée lors de sa requête *AJAX*. Pour cela, vous pouvez extraire l'information après la ressource (c.-à-d. après le dernier slash '/') grâce aux instructions suivantes :

```
$id = array_shift($request);  
if ($id == '')  
    $id = NULL;
```

Écrivez ensuite, les instructions permettant :

- d'appeler la fonction `dbRequestPhotos`, du script « database.php », lorsque la ressource demandée est « photos » et que l'id est nul ;
- d'appeler la fonction `dbRequestPhoto`, du script « database.php », lorsque la ressource demandée est « photos » et que l'id est non nul.

Finalement, si les données récupérées par l'appel de ces fonctions sont valides, envoyez les données au client après leur encodage en *JSON* (pensez à modifier le type des données renvoyées dans le *header HTTP* du script « request.php »). Dans le cas contraire, renvoyez un statut 400 :

```
header('HTTP/1.1 400 Bad Request');
```

Pour plus de lisibilité, il est conseillé d'écrire une fonction permettant de renvoyer les données au client en *JSON* (le code correspondant aux statuts *HTTP* 200 ou 201) :

```
sendJsonData($data, $code);
```

3 Traitement des réponses *AJAX* en *JavaScript*

Affichage des vignettes

Dans cette partie, nous souhaitons afficher les vignettes des photos dans la section d'id `thumbnails` (définie dans le code *HTML*). Avant toute chose, modifiez la fonction `requestThumbnails`, du fichier « photos.js », pour utiliser la version *JSON* de la réponse du serveur plutôt que sa version texte. Ensuite, ajoutez dans la fonction `displayThumbnails` du fichier « loading.js » les instructions suivantes :

- Création des miniatures qui auront la forme suivante :

```
<div class="col-xs-2 col-md-2">
  <a href="#">
    
    </a>
  </div>
```

Où les deux premiers emplacements ?? sont à remplacer par l'id de la photo et le second par son chemin.

- Ajout d'une gestion de l'évènement *click* sur les vignettes :

```
document.getElementById('thumbnails').addEventListener('click', ??);
```

Où les ?? sont à remplacer par la fonction suivante :

```
requestPhoto(event)
```

Où `event` est l'évènement généré par le clic sur la vignette. La fonction `requestPhoto` doit permettre de réaliser la requête *AJAX* permettant de récupérer les informations détaillées de la photo cliquée. On obtient l'id de cette dernière grâce à l'information stockée dans l'attribut `photoid`. Cela peut être réalisé avec l'instruction suivante :

```
let id = document.getElementById(event.target.id).getAttribute('photoid');
```

En cas de succès de la requête *AJAX*, appelez une nouvelle fonction nommée `displayPhoto` à ajouter au script « loading.js ».

Affichage de la photo grand format

Dans cette partie, nous souhaitons afficher la photo en grand format lorsque l'utilisateur clic sur une miniature. Cet affichage doit se faire dans la `div` d'id `photo` (définie dans le code *HTML*). Pour cela, ajoutez à la fonction `displayPhoto` l'affichage de la photo grand format avec son titre dans une structure *HTML* de la forme suivante :

```
<div class="card col-xs-12 col-md-12">
  <div class="card-body">
    <h4>??</h4>
    
  </div>
</div>
```

Où les trois ensemble ?? sont respectivement à remplacer par le titre, le chemin et l'id de la photo.