

## TP2

Modèle de base de données v2  
Pages PHP avec requêtes SQL

Ce TP correspond à la mise en œuvre des notions SQL et PHP abordées en cours. Vous devez rendre votre travail en fin de séance.

### Partie I : On utilise la base matos2 !

La première partie consiste à utiliser le script **matos2\_ddl.sql** et les scripts **\*\_dml.sql** associés pour remplir la base avec des enregistrements. Récupérer tous les fichiers nécessaires sur Moodle. Dans votre compte-rendu, indiquer clairement l'ordre d'importation des fichiers **\*\_dml.sql**. Examiner le contenu des tables et identifier ce qui a changé par rapport la version 1 de **matos**.

### Partie II : On prépare le répertoire /var/www/html !

Le répertoire **/var/www/html** a par défaut les droits de l'utilisateur **root**. Pour que vous puissiez ajouter, modifier ou détruire des fichiers dans ce répertoire avec votre identifiant d'utilisateur **pi**, il faut modifier les droits d'accès avec les commandes suivantes :

```
$ cd /var/www
$ sudo chmod -R777 html
```

Récupérer ensuite le fichier **listepersonnes.php** sur Moodle, et modifier le pour être sûr que les informations de connexion (login, password, nom de la base) sont correctes. Tester ensuite avec votre navigateur si l'affichage des élèves se passe bien. Prendre une photo d'écran pour valider ce point. Expliquer avec vos mots le fonctionnement de la fonction **query\_database()** et surtout la structure du résultat rendu par cette fonction (rôle des array...)

### Partie III : On filtre maintenant les élèves par promo !

La troisième partie consiste à écrire les deux pages PHP nécessaires pour afficher les élèves par promotion. La première, nommée **listerpromos.php** s'inspire largement de **listepersonnes.php** mais doit maintenant proposer un formulaire avec une liste déroulante permettant de sélectionner la promotion à afficher, ainsi qu'un bouton de validation/soumission.

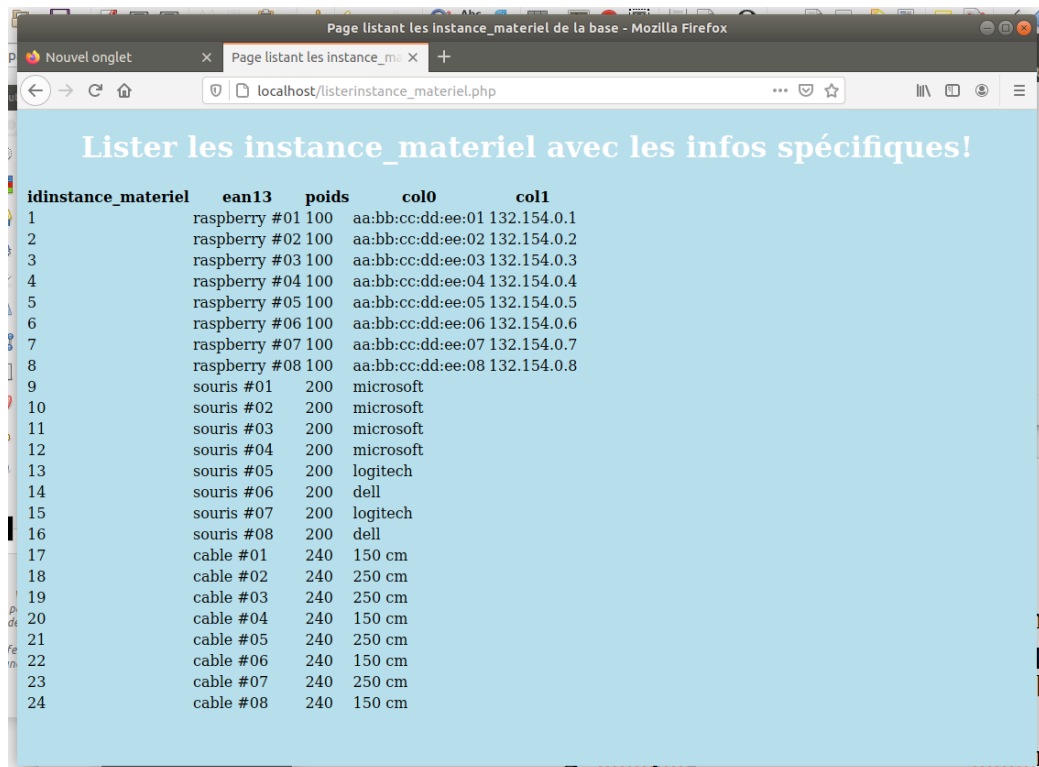
Une fois que l'utilisateur de la page a sélectionné une promo et cliqué sur le bouton de validation, la nouvelle page **affiche\_promo.php** doit être appelée qui, comme son nom l'indique, doit afficher la promo sélectionnée.

### Partie IV : On rajoute un filtrage par promo et par genre (homme/femme) !

La quatrième partie consiste à écrire les deux pages PHP nécessaires pour afficher les élèves par promotion et par sexe. La première, nommée **listerpromos\_genre.php** s'inspire largement de **listerpromos.php** mais doit maintenant proposer un formulaire avec une liste déroulante permettant de sélectionner la promotion à afficher, des boutons checkbox pour sélectionner les genres souhaités (homme/femme, ainsi qu'un bouton de validation/soumission).

Une fois que l'utilisateur de la page a sélectionné une promo, sélectionné les checkbox souhaitées et cliqué sur le bouton de validation, la nouvelle page **affiche\_promo\_genre.php** doit être appelée qui, comme son nom l'indique, doit afficher la promo sélectionnée avec les bonnes valeurs de genre.

## Partie V : On réalise un affichage complexe : la liste des instance\_materiel !



idinstance_materiel	ean13	poids	col0	col1
1	raspberry #01	100	aa:bb:cc:dd:ee:01	132.154.0.1
2	raspberry #02	100	aa:bb:cc:dd:ee:02	132.154.0.2
3	raspberry #03	100	aa:bb:cc:dd:ee:03	132.154.0.3
4	raspberry #04	100	aa:bb:cc:dd:ee:04	132.154.0.4
5	raspberry #05	100	aa:bb:cc:dd:ee:05	132.154.0.5
6	raspberry #06	100	aa:bb:cc:dd:ee:06	132.154.0.6
7	raspberry #07	100	aa:bb:cc:dd:ee:07	132.154.0.7
8	raspberry #08	100	aa:bb:cc:dd:ee:08	132.154.0.8
9	souris #01	200	microsoft	
10	souris #02	200	microsoft	
11	souris #03	200	microsoft	
12	souris #04	200	microsoft	
13	souris #05	200	logitech	
14	souris #06	200	dell	
15	souris #07	200	logitech	
16	souris #08	200	dell	
17	cable #01	240	150 cm	
18	cable #02	240	250 cm	
19	cable #03	240	250 cm	
20	cable #04	240	150 cm	
21	cable #05	240	250 cm	
22	cable #06	240	150 cm	
23	cable #07	240	250 cm	
24	cable #08	240	150 cm	

On souhaite réaliser une page PHP permettant d'afficher la liste des instance\_materiel. Pour chaque enregistrement, on souhaite afficher sur la même ligne les infos de la table **instance\_materiel** ainsi que les infos spécifiques. Par exemple, pour l'enregistrement 1, le code ean13 **raspberry #01** et le poids **100** viennent de la table *instance\_materiel*, les données spécifiques depuis la table *tm\_raspberry* **aa:bb:cc:dd:ee:01** et **132.154.0.1**.

Il faut ici écrire une page PHP réalisant plusieurs requêtes SQL consécutives.

Avant d'écrire le code PHP, écrire dans votre compte-rendu en français les grandes idées de votre solution. Quelles requêtes faire ? Dans quel ordre ?

Procéder ensuite par étape, et vérifier à chaque fois avec des echo, si l'étape donne bien les résultats.

Il y a en gros deux manières de faire, une relativement simple où on ne cherche pas à récupérer les noms des colonnes d'une table spécifique (select \*) et une où on utilise la requête « show columns from tm\_XXX » qui entraîne un peu de manipulation de chaîne de caractères en PHP. Les deux méthodes sont acceptées, si le code PHP rendu est commenté.