

# TP : Initiation à MySQL avec PhpMyAdmin

## Objectif du TP

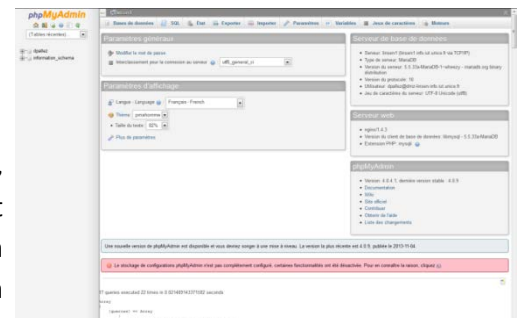
L'objectif de ce TP est de réviser le langage SQL tout en découvrant un autre SGBD que Microsoft Access. Il existe de nombreux SGBDs (un comparatif est disponible [ici](#)) mais nous avons choisi de présenter un équivalent à **MySQL** qui est le 2<sup>e</sup> SGBD le plus utilisé au monde suivant le classement [DB-Engines](#) lors de la rédaction de ce document (identifiez la place de Microsoft Access). MySQL est particulièrement utilisé en programmation web du fait de son packaging avec Apache & Php ([LAMP](#), WAMP...). L'une de ses spécificités est d'inclure plusieurs manières de stocker les données au sein d'un même SGBD (MyISAM, InnoDB, Maria...). Suite au rachat de Sun (distributeur de MySQL) par Oracle (avril 2009), il est apparu [plusieurs variantes](#) (ou *forks*) de MySQL. Ainsi, dans ce TP, nous allons utiliser [MariaDB](#) (un fork OpenSource de MySQL) qui utilise par défaut le moteur XtraDB (un fork de InnoDB).



1. À la fin de la séance, envoyez un fichier texte à votre responsable de TP (respectez le format TP\_SGBD\_NomDuBinome1\_NomDuBinome2.sql). Le fichier contiendra les numéros des questions traitées et les réponses textuelles et/ou SQL correspondantes). Tout ce qui n'est pas SQL devra être mis en commentaire avec le caractère # devant chaque ligne.

## CONNEXION À MARIADB

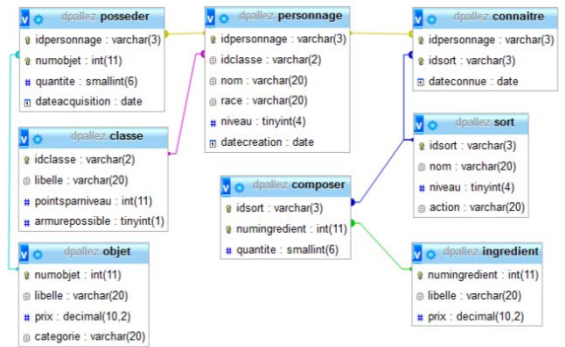
2. Pour se connecter au SGBD, nous allons utiliser phpMyAdmin, correspondant à un ensemble de pages php et permettant d'administrer la base de données. Utilisez le lien <http://wiki.unice.fr/display/IntraSIDI/PhpMyAdmin> dans un navigateur web avec votre login & mps unice pour y accéder.



## DÉCOUVERTE DE PHPMYADMIN

3. Nous allons recréer la base de données Jeu de rôle en créant les tables de différentes façons :
  - a. Allez dans l'onglet Structure pour saisir interactivement la table `classe(idclasse:chaîne(2), libelle:chaîne(20), pointsparniveau:entier, armurepossible:booléen)`; cliquer sur sauvegarder une fois terminé ;
  - b. Affichez à nouveau la structure de la table que vous venez de créer et ajoutez les contraintes d'intégrité suivantes :
    - i. la clé primaire ;
    - ii. tous les attributs ont une valeur nulle par défaut sauf Libelle ;
    - iii. Libelle est indexé. Quelle est la requête SQL correspondante ?
  - c. Dans l'onglet Importer, choisissez le fichier `script_pour_importation.sql` et exécutez-le. Observez bien ce qui a été créé.
  - d. Dans l'onglet SQL, créez les tables `objet` et `ingrédient` grâce à [l'aide en ligne de MariaDB](#). Pour information, le raccourci clavier permettant d'exécuter une requête SQL est <CTRL>+<Enter>.
4. Comme vous avez pu le constater, le script importé ne contenait pas les contraintes d'intégrités (clés primaires, intervalle de valeurs...). À l'aide d'instructions SQL, modifiez les structures des tables en créant les clés primaires et les clés étrangères manquantes.

5. Dans le menu de gauche, cliquez sur votre base de données (correspondant à votre login) et ensuite cliquez sur le menu plus > Concepteur. Vous devez obtenir un diagramme équivalent à celui d'Access qui ressemble au schéma ci-contre. Vous pouvez réorganiser le positionnement des tables afin d'éviter le croisement des relations mais il faudra ensuite sauvegarder vos modifications (utiliser le navigateur chrome plutôt que Internet Explorer).



6. À l'aide d'une requête SQL, modifiez la structure de la table `personnage` en ajoutant la contrainte d'intégrité qui précise que le `niveau` doit être compris entre 1 et 20.
7. Créer la requête SQL qui ajoute l'enregistrement « 'GAL','MA','Galadriel','Humain','2007-07-14','22' » dans `Personnage`. Que constatez-vous ?
8. Nous avons progressé par rapport au SGBD Access qui, lui, n'accepte même pas le mot clé `CHECK`. Pour votre information, MySQL accepte le mot clé `check` mais n'en tient absolument pas compte (cf. documentation [ici](#) qui précise que « the CHECK clause is parsed but ignored by all storage engines » !). Identifiez grâce à l'aide en ligne de MySQL et de votre ami Google, quelles autres instructions/clauses pourraient palier à ce problème ? De cette mauvaise expérience, vous devez en retenir 3 choses :
- Toujours prendre le temps de lire la documentation du logiciel que vous utilisez. Même entre deux logiciels qui respectent une norme, l'implémentation et donc l'utilisation peut différer ;
  - Choisir le logiciel en fonction de vos besoins et pas parce que tout le monde l'utilise (MySQL est le 2<sup>e</sup> SGBD le plus utilisé au monde alors qu'il ne respecte pas la clause `CHECK` pourtant spécifié dans la norme [SQL 92](#)). Plutôt que MySQL, envisagez l'utilisation de [PostgreSQL](#).
  - Comme vous avez choisi le logiciel en fonction de critères, il est nécessaire de vérifier et tester ces critères.
9. Créez la requête SQL qui permet de préciser que par défaut, la valeur de l'attribut `quantite` dans `posseder` est égale à 1 (altérer la structure de la table en changeant l'attribut `quantite` en un autre attribut de même nom et de même type mais auquel on a ajouté une valeur par défaut).
10. Vérifiez que la valeur de l'attribut `quantite` est égale à 1 lors de l'ajout d'un enregistrement dans `posseder`.
11. Nous souhaitons faire en sorte que par défaut l'attribut `dateacquisition` dans `posseder` corresponde à la date du jour. En lisant la documentation de MariaDB ou MySQL (en particulier `create table`), est ce possible et si oui comment ?
12. Afin d'importer des données dans MySQL, ouvrez le fichier Excel contenant les informations de la base de données `JeuDeRole`. Pour chaque onglet (c'est-à-dire pour chaque table), il faut enregistrer ces données dans un format importable par MySQL. Le format le plus standard est le format CSV :
- Cliquez sur le bouton droit de l'onglet et choisissez `déplacer` ou `copier`, puis créez une copie dans un nouveau classeur ;
  - Enregistrez ce nouveau classeur avec le nom correspondant à la table mais en modifiant le type du classeur en CSV ;
13. Nous souhaitons importer les données de la table `classe`. Pour cela :
- Sous phpMyAdmin, cliquez sur la table `classe` et cliquez sur `importer` ;

- b. Dans l'interface d'importation, choisissez le fichier `csv` correspondant à la table, modifiez l'encodage des caractères en `windows-1252` (encodage par défaut sous windows), n'importez pas la 1<sup>ère</sup> ligne qui correspond aux intitulés des attributs, remplacer les données de la table surtout si ce n'est pas votre 1<sup>er</sup> essai pour cette table, modifiez les propriétés du CSV : colonnes séparées par des « ; », colonnes entourées par (rien) et précisez le nom des colonnes de la BD MySQL qui sont stockées dans le fichier CSV car l'ordre des attributs ne correspond pas ;
  - c. Cliquez sur le bouton `Exécuter` ;
  - d. Afficher le contenu de la table en cliquant sur son nom dans le menu de gauche. Que constatez-vous ? Le plus simple pour remédier au problème est d'aller modifier le fichier `classe.csv` et de recommencer l'importation.
14. Recommencer l'importation (question 13) pour la table `personnage`. Affichez le contenu de la table en cliquant dessus et analysez les dates ! Pour remédier à cela, il faut encore modifier le fichier `csv` dans Excel en sélectionnant la colonne de date, en choisissant le menu contextuel (bouton droit) `format de cellule` et en définissant un nouveau format personnalisé `aaaa-mm-jj`. Enregistrez et retournez à la question 13.
  15. Importez les données des autres tables.
  16. À l'aide du menu `Exporter`, tester plusieurs exportations de votre BD en variant sur le format (SQL, PDF, CSV, XML...) et sur les paramètres en passant d'une exportation `rapide` à `personnalisée`. Analysez le code SQL généré par MySQL à chacune de vos tentatives. Que remarquez-vous ?
  17. Cliquez sur la Base de Données dans le menu à gauche (votre nom de login) et choisissez le menu `Requête`. Créez la requête interactive qui affiche les identifiants des sorts connus par Aragorn triés par ordre alphabétique décroissant.
  18. À l'aide du menu SQL, créez une *seule* requête SQL qui insère simultanément plusieurs enregistrements dans la table `ingredient` (poussière d'ange et poil de licorne par exemple). Cette requête n'existait pas sous Access.
  19. Créez la requête SQL qui affiche le nom des sorts connus par Aragorn triés par ordre alphabétique inverse.
  20. Donner les noms des `classes` qui ont au moins 2 représentants différents dans `personnage`.
  21. Donner le nom des personnages connaissant tous les sorts (division).
  22. Donner les noms des personnages qui ne connaissent aucun sort (une requête avec `EXISTS` et une autre équivalente avec `IN`) ?
  23. Grâce à la question 22, réécrivez la question 21 en utilisant la clause `IN`. Que constatez-vous ?
  24. Que fait la requête ci-contre ? Ce n'est pas la peine de la tester sur machine car le résultat devrait être *vide*. Prenez le temps de la lire, de la comprendre et de deviner à quoi correspond cette requête (en langage naturel). Si vous ne savez vraiment pas répondre, essayez d'ajouter des données à votre BD pour vous aider. Qu'en déduisez-vous ? Pourquoi cette requête n'était pas possible sous Access ?

```
select p.nom
from posseder po
inner join personnage p
on (po.idpersonnage=p.idpersonnage)
group by po.idpersonnage
having count(distinct po.numobjet) =
(select count(*) from objet) ;
```
  25. Écrire la requête qui donne le nom et le niveau des sorts composés de tous les ingrédients (2 versions différentes) ?
  26. Donner les noms des personnages qui connaissent un sort de niveau supérieur à leur niveau (sans clause `where`).
  27. Donner la composition des sorts qui correspondent à une action d'*augmentation*. La requête devra être la plus indépendante possible à la manière dont l'action a été saisie (gérer la casse = MAJUSCULE ou minuscule).

28. Afficher le nom et le nombre de sorts connus du personnage qui connaît le plus de sort (mot clé `Limit`).
29. Donner le nom des sorts et leur niveau qui sont des soins (gérer l'indépendance à la casse). Il est conseillé d'utiliser une fonction sur les chaînes de caractères.

30. On souhaite calculer la force d'un personnage qui dépend de son âge (en jours) résumé dans le tableau ci-contre. Créez la requête qui permet de calculer cette force pour chaque personnage.

Force	Age
2*niveau	< 2600 jours
Nb de jours d'existence/ niveau	>= 2600 jours

31. Tester la requête suivante `select * from sort s inner join connaitre c using(idsort) ;` et dites à quoi elle correspond. Prenez le temps de lire cette [aide en ligne](#).
32. Quels sont les sorts qui sont connus par un personnage ne nécessitant aucun ingrédient ?
33. Donnez les noms des objets possédés par au moins 3 personnages. Profitez en pour tester la jointure naturelle (`natural join`).
34. On souhaite connaître le nom des personnages qui connaissent tous les sorts et qui possèdent tous les objets. Pour cela :
- En vous inspirant de la question 21, créez une vue nommée `Personnage_Intelligent` qui donne toutes les informations des personnages connaissant tous les sorts.
  - Créez une autre vue nommée `Personnage_Riche` ;
  - Créez une requête basée sur les deux vues précédentes afin de répondre à la question initiale. Pour vérifier qu'elle fonctionne correctement, il sera peut être nécessaire d'ajouter des données dans la table.
35. Créez une vue nommée `Personnage_PremierSort` qui permet d'afficher l'identifiant d'un personnage ainsi que la date du premier sort qu'il a appris.
36. Comme vous le savez, votre BD contient des personnages qui ne connaissent aucun sort. Toutefois, pour faire un inventaire, nous souhaitons afficher *tous* les personnages avec si possible le premier sort appris. Si le personnage ne connaît aucun sort, alors il doit malgré tout apparaître dans la liste mais avec aucune information sur la date du sort (`NULL`).
- Créez la requête qui donne le nom des personnages et la date d'apprentissage de leur premier sort. Respectez l'ordre de jointure `personnage`  $\bowtie$  `autre table`.
  - Dans la requête précédente, remplacez la clause `inner` par la clause `left outer`. Qu'obtenez-vous ?
  - Remplacez `left` par `right` dans la requête précédente. Qu'obtenez-vous ?
  - Dans la requête précédente (en 36.c), changez l'ordre de la jointure (`autre table`  $\bowtie$  `personnage`). Qu'obtenez-vous ?
37. On souhaite comparer le temps d'exécution d'une jointure entre deux tables en faisant un produit cartésien puis une restriction d'une part et en utilisant un `inner join` d'autre part. Pour cela :
- Créez deux vues `Test_jointure_cartesien` et `Test_jointure_join` qui affiche les objets qui ont la même catégorie que l'objet `Dague`. Vous l'aurez compris, la première vue utilisera un produit cartésien (`... from Objet o1, objet o2...`) et la seconde vue utilisera une jointure (`... Objet o1 inner join Objet o2...`).

- b. Pour réellement avoir une chance de différencier les temps d'exécution, nous devons ajouter énormément d'enregistrements dans la BD. Pour cela, exécuter le programme ci-contre. Que

```
#re définit l'opérateur de fin de ligne
Delimiter $$
# supprime la procédure si nécessaire
drop procedure IF EXISTS manyrecords$$
# définit une nouvelle procédure
CREATE PROCEDURE manyrecords (IN nb INT)
BEGIN
    REPEAT
        insert into objet (libelle, categorie, prix) values ('Test','Arme', 10000);
        SET nb = nb - 1;
    UNTIL nb=0
    END REPEAT ;
END $$
Delimiter ;
```

constatez-vous dans votre SGBD ?

- c. Vous pouvez exécuter ce programme en l'appelant avec l'instruction `call` et en précisant le nombre de lignes à créer (essayer avec 100000).
- d. Vérifiez que les enregistrements ont bien été ajoutés à la table en les comptant.
- e. Affichez alternativement tous les enregistrements de chaque vue et noter le temps mis pour l'exécution. Répétez plusieurs fois l'opération et faites une moyenne.
- f. Supprimer tous les enregistrements de la table objet dont le libellé est `Test`.

## Sources pour ce TP

Norme ANSI SQL-92 : <http://en.wikipedia.org/wiki/SQL-92>

MySQL : <http://www.mysql.fr/>

Comparatif des SGBD : <http://fadace.developpez.com/sbgdcmp/>

Compatibilité entre MariaDB et MySQL : <https://kb.askmonty.org/en/mariadb-versus-mysql-compatibility/>

Différences entre MariaDB & MySQL : <https://kb.askmonty.org/en/mariadb-versus-mysql-features/>

Utiliser PhpMyAdmin (avec MariaDB) sans installation : <http://demo.phpmyadmin.net/>

Aide en ligne de MariaDB : <https://kb.askmonty.org/en/mariadb-documentation/>