

Travaux Pratiques de Base de données

Création de BDs relationnelles avec MySql (phpMyAdmin)

(2 séances)

Objectif

On va ici s'intéresser à la création de cette base de données en utilisant un SGBD (MySQL). Cela nous permettra en particulier d'expérimenter la création d'une base de données relationnelle en utilisant l'interface fournie avec MySQL (phpMyAdmin).

Prise en main

Accédez à la page web : <http://euterpe.unice.fr/phpmyadmin/>. Votre identifiant est constitué de L2I suivi de votre nom en majuscule (limité à 13 caractères en incluant L2I)¹. Le mot de passe est votre nom d'utilisateur sur la boîte « mail » de l'université.

Ensuite, à partir du menu déroulant à gauche sélectionnez votre base de données, elle a le même nom que votre identifiant. N'oubliez pas que l'interface *phpMyAdmin* utilise le terme de *table* pour désigner une relation, le terme de *champ* pour désigner un attribut, le terme d'*enregistrement* pour désigner un tuple et le terme de *relation* pour désigner une liaison entre tables.

Exercice 1 : Création du schéma (premiers éléments)

Se positionner sur votre base de données et utiliser l'interface proposée pour créer vos tables (c'est assez intuitif). **Attention** : il faut créer les tables en leur donnant le type **InnoDB**.

Créer les relations **Etudiant**, **UE**, **EtudiantUE**, **Adresse**, **Enseignant** de telle manière à ce qu'elles contiennent les attributs suivants. Choisir les types de données qui vous semblent les plus appropriés (se référer au cours et au TP1) et spécifier si la saisie de donnée est obligatoire ou non pour un attribut (**NULL/NOT NULL**):

- **Etudiant** : numéro (clé primaire), nom, prénom.
- **UE** : code, libellé, nombre d'heures associées (cours, TP, TD confondus). Le code est la clé primaire de la relation.
- **Adresse** : numéro et nom de rue, nom de la ville, code postal, caractère ('t' pour Ter, 'b' pour Bis, et ' ' si aucun des deux. On pourra pour simplifier donner un « numéro généré automatiquement » à chaque adresse qui servira de clé primaire.
- **Enseignant** : numéro (clé primaire), nom, prénom, âge, ville où il habite, nombre d'heures enseignées.
- **EtudiantUE** : la note de CC obtenue, la note d'examen obtenue.

Exercice 2 : Création du schéma (compléments)

On veut maintenant compléter la description de ces relations en introduisant les informations qui nécessitent de lier ces relations entre-elles (définir les clés étrangères). Pour cela il faudra rajouter des attributs (avec index) à certaines relations et utiliser le lien « gestion des relations ». Attention choisir l'option SET NULL (si l'attribut l'autorise) ou CASCADE (sinon).

1. **Etudiant**. Il faut i) lier la relation **Etudiant** avec la relation **Adresse** afin d'exprimer le fait que tout étudiant a une adresse et ii) lier la relation **Etudiant** et la relation

¹ Les ' ' sont remplacés par '_'. En cas de doute, l'enseignant de TP a la liste des identifiants.

EtudiantUE afin de préciser les UEs avec leurs notes² qui sont suivies par les étudiants. Eventuellement, un étudiant peut ne suivre aucune UE. *Avant de créer une clé étrangère dans une relation il faut qu'elle ait sa clé primaire définie.*

2. **Enseignant**. Il faut lier la relation **Enseignant** avec la relation **UE** afin d'exprimer le fait que chaque enseignant peut être responsable d'UEs (éventuellement, un enseignant peut être responsable d'aucune UE)³.
3. **UE**. A propos de la liaison de la relation **UE** avec la relation **Enseignant** afin d'exprimer le fait que chaque UE a un enseignant responsable. Que pouvez-vous dire ?
4. **EtudiantUE**. Il faut lier la relation **EtudiantUE** avec la relation **UE** afin de pouvoir faire le lien entre les notes et l'UE concernée. *Avant de créer une clé étrangère dans une relation il faut qu'elle ait sa clé primaire définie.* Définir la clé primaire qui vous semble raisonnable (vous ne choisirez pas ici de numéroter les tuples comme on a pu le faire plus haut). A propos de la liaison de la relation **EtudiantUE** avec la relation **Etudiant** afin d'exprimer le fait que les notes sont obtenues par un certain étudiant. Que pouvez-vous dire ?

Question : Quelles sont vos remarques à propos du travail correspondant fait dans le TP1 (productivité, spécification des liaisons entre classes/rerelations,...) ?

Exercice 3 : Insertion de données

Pour chaque table vous cliquerez sur le bouton ou l'onglet « insérer » selon que vous soyez au niveau de la base de données ou à celui de la table concernée. L'objectif est de créer les tuples. A priori il faut opérer dans un ordre logique : **Adresse**, **Etudiant**, **Enseignant**, **UE**, **EtudiantUE**⁴. Saisir les tuples suivants :

- Adresse(1, 3, 'b', "Jean médecin", "O6000", "Nice")
- Adresse(2, 10, ' ', "Barla", "O6000", "Nice")
- Adresse(3, 10, ' ', "Jean Jaures", "O6200", "Cagnes")
- Etudiant(1001, "Nom1", "prenom1", 1)
- Etudiant(1002, "Nom2", "prenom2", 2)
- Etudiant(1003, "Nom3", "prenom3", 3)
- Enseignant(1, "Menez", "Gilles", 25, 35, "Antibes")
- Enseignant(2, "Lahire", "Philippe", 26, 30, "Nice")
- Enseignant(3, "Kounalis", "Emanuel", 27, 28, "Nice")
- Enseignant(4, "Renevier", "Philippe", 21, 28, "Nice")
- UE("SL2IBD", "Base de Données", 24, 2)
- UE("SL2IPI", "Programmation impérative", 36, 1)
- UE("SL2IAL", "Algorithmique", 32, 3)
- UE("SL2IPW", "Programmation WEB", 26, 4)
- EtudiantUE(1001, "SL2IBD", 10, 11)
- EtudiantUE(1001, "SL2IPI", 8, 10)
- EtudiantUE(1002, "SL2IBD", 10, 11)
- EtudiantUE(1002, "SL2IPI", 8, 10)
- EtudiantUE(1003, "SL2IBD", 10, 11)
- EtudiantUE(1003, "SL2IPI", 8, 10);
- EtudiantUE(1003, "SL2IAL", 12, 13);

² à priori il n'y en aura pas plus de 5 mais pour l'instant on ne s'en préoccupe pas.

³ Même remarque concernant le nombre que pour la relation **Etudiant**.

⁴ Dire pourquoi cet ordre est logique.

Exercice 4 : Consultation et sélection simple de données

Utiliser le bouton « afficher » pour voir l'ensemble des informations contenues dans chacune des tables. Utiliser ensuite successivement le bouton « chercher » et l'onglet « Requête »⁵ pour afficher les informations suivantes (dans les deux cas utiliser les différentes possibilités offertes):

1. Numéro et nom de la rue et ville, pour toutes les adresses
2. Nom et prénom de tous les étudiants
3. Le libellé de l'UE et le numéro de l'enseignant responsable pour toutes les UEs
4. code postal et ville, pour toutes les adresses (avec ou sans suppression des doublons)

Premières sélections :

1. Affichage de tous les étudiants dont le nom est « Nom1 » (toutes les informations)
2. Affichage de tous les numéros d'étudiants qui suivent l'UE « SL2IBD »
3. Afficher tous les enseignants dont le prénom est « Philippe ».
4. Afficher tous les noms de rues de la ville « Nice ».

Exercice 5 : Sélection plus complexes

C'est le moment d'utiliser de manière plus fine les possibilités de MySQL pour réaliser des requêtes impliquant plusieurs tables. Pour cela, se placer au niveau de la base de données et cliquer sur l'onglet « Requête »

1. Afficher tous les noms et prénoms d'étudiants qui suivent l'UE « SL2IBD » ou l'UE « SL2IPI »
2. Afficher tous les noms et prénoms d'étudiants ainsi que le libellé des UEs toujours pour ceux qui suivent l'UE « SL2IBD » ou l'UE « SL2IPI »
3. Afficher tous les étudiants dont le nom est « Nom1 », mais en affichant que le prénom de l'étudiant et le nom de la rue où il habite.
4. Afficher le nom du responsable de chacune des UEs « SL2IBD » et « SL2IPI »
5. Afficher le nom de l'étudiant, le libellé de l'UE et la note de CC de chacun des étudiants.
6. Afficher le nom de l'étudiant, le libellé de l'UE et la note de CC de chacun des étudiants dont le nom commence par « Nom »

Exercice 6 : Modification de données

Afficher le contenu d'une table en utilisant le bouton « afficher » présent pour chaque table, puis pour chacun des champs utiliser le bouton « modifier » afin de réaliser les opérations suivantes (tuple par tuple⁶) :

1. Augmenter toutes les notes d'examen de 1 point pour tous les étudiants et toutes les UEs.
2. Pour les étudiants qui suivent l'UE « SL2IBD », augmenter la note de contrôle continu de 1 point.

Exercice 7 : Modification du schéma

Utiliser le bouton « structure » présent pour chaque table afin de vous mettre en position de travailler sur la structure d'une table. Ensuite, utilisez les boutons « modifier » ou

⁵ « chercher » une fois que vous êtes positionné sur la bonne table et « requête » au niveau de la base de données.

⁶ On verra comment automatiser ces requêtes avec le langage SQL dans un prochain TP.

« supprimer » accessible pour chaque champ d'une table et le bandeau d'ajout de colonne placé sous la table pour réaliser les opérations suivantes :

1. Changer le type de l'attribut « ville » de la table **Adresse** de manière à en augmenter la taille de 5 caractères.
2. Ajouter une colonne « age » à la table **Etudiant** et compléter son contenu.
3. Supprimer la colonne « age » de la table Etudiant

Exercice 8 : Nouveau contexte

On désire réaliser une base de données pour la gestion d'une bibliothèque de DVDs.

Chaque société de location de DVD a un nom et une adresse (nom de rue et ville), ainsi qu'un nom de directeur. Plusieurs sociétés de location peuvent porter le même nom.

Tout DVD possède un titre, un auteur, une année de sortie, un numéro qui permet de l'identifier, une catégorie (ex : action, aventure); on enregistre également sa date d'acquisition, le nombre d'exemplaires achetés et la société qui l'a acquis.

Pour un acteur on mémorise son nom, son prénom, son âge et son sexe. Pour chaque DVD et pour chaque acteur principal du film concerné, on mémorise le rôle tenu dans le film.

Un DVD est emprunté par un client (nous ne mémoriserons pas d'autre information que le numéro du client) à une date donnée et pour une durée déterminée (nombre de jours). Un même client ne peut emprunter un même DVD plus d'une fois par jour.

Société (numS, nomS, rueS, villeS, directeurS)
Dvd (numD, titreD, auteurD, annéeD, catégorieD, dateAchatD, nombreD, *sociétéD*)
Acteur (numA, nomA, prénomA, âgeA, sexeA)
Casting (dvdC, acteurC, rôleC)
Emprunt (dvdE, dateE, clientE, duréeE)

(un trait souligné correspond à une clé primaires et l'italique à une clé étrangère)

Travail à faire :

- Créer la base de données
- Mettre en œuvre les requêtes suivantes
 1. Liste des DVDs acquis par la société 'DVDStore'.
 2. Liste des DVDs acquis par la société 'DVDStore', avec pour chacun le nom et le prénom des acteurs qui ont participé au film.
 3. Liste des acteurs de chaque DVD.
 4. Liste des clients ayant loué un DVD dont un des acteurs a « xavier » pour prénom
 5. ...
- Ajouter une information sur le numéro de la rue où se trouve une société de location.