

Labo2 : Création d'une base de données relationnelle avec MySQL

Objectifs

Ce laboratoire a pour but l'implémentation d'une base de données relationnelle et la mise en œuvre des contraintes d'intégrité référentielle appropriées avec le SGBD MySQL. Nous allons mettre en place la base de données COMPANY introduite dans le cours.

Plus précisément, les points étudiés dans ce laboratoire sont :

1. Création d'une base de données avec le SGBD MySQL (MySQL Server ou MySQL Workbench).
2. Mise en œuvre et utilisation des fonctionnalités suivantes :
 - Définition des tables
 - Définition des contraintes d'intégrité
 - Insertion des données
3. Tests des contraintes d'intégrité.

A rendre

Le laboratoire doit être réalisé **en groupe de 2 étudiants** (maximum un groupe de 3 si le nombre est impair).

Créer un fichier zip contenant vos scripts pour chaque partie et vos réponses aux questions en format BDR_[A|B]_LABO2_NOM1_NOM2.zip où A ou B représente votre classe et NOM1 < NOM2 (dans l'ordre alphabétique). Déposer votre fichier zip sur le site [cyberlearn](http://cyberlearn.ch).

Le laboratoire est à rendre le **lundi 12 octobre 2015 avant minuit**.

1. Implémentation des tables sous MySQL

Prenez le schéma relationnel de la base de donnée COMPANY fourni en Annexe 1.

- Créer les tables. Vous devez respecter le type exact des champs fourni dans l'Annexe.

Attention : Dans cette première phase n'ajoutez pas encore les contraintes d'intégrité référentielle !

Indiquez les commandes SQL utilisées.

2. Insertion des données

- 3.1. Les commandes d'insertion vous sont fournies dans des fichiers .sql (un fichier par table). Exécutez les fichiers dans la fenêtre SQL afin de peupler la base de données. Prenez connaissance des différents tuples ajoutés à la base de données.
- 3.2. Effectuez les modifications suivantes en utilisant les commandes SQL :

- Insérez les projets numéro 3 et 5 pour l'employé dans la table `works_on` avec le `Ssn` '123456789' et attribuez 10 heures de travail sur chaque projet. Indiquez les commandes SQL utilisées. Que constatez-vous ? Critiquez le résultat obtenu.
- Supprimez le département numéro 5 dans la table `department`. Que constatez-vous ? Critiquez le résultat obtenu.

3. Implémentation des contraintes d'intégrité référentielle

- 4.1. Premièrement, il vous faut vider toutes vos tables des différents tuples existants. Indiquez les commandes SQL utilisées.
- 4.2. Ensuite, pour chacune des tables, ajoutez les contraintes d'intégrité référentielle (voir Figure 1 en Annexe 1). Indiquez les commandes SQL utilisées.
- 4.3. Une fois toutes vos contraintes établies, vous devez peupler à nouveau vos tables avec les fichiers `.sql` fournis.
 - Est-ce que ceci est possible ?
 - Que faut-il faire pour pouvoir insérer des tuples ?

Remarque : vous pouvez utiliser la commande `SET FOREIGN_KEY_CHECKS = 0` pour désactiver la vérification des contraintes d'intégrité référentielle. Ceci est utile pour le rechargement d'une base existante sans respecter l'ordre de dépendance entre les tables. A la fin de vos insertions il faut réactiver les contraintes avec `SET FOREIGN_KEY_CHECKS = 1`.

- 4.4. Exécutez les commandes suivantes, observez les résultats et expliquez le comportement :
 - Dans la table `employee`, modifiez le `Dno` du tuple qui a le `Ssn` = '999887777' à 7.
 - Dans la table `employee`, supprimez le tuple correspondant à `Ssn` = '999887777'.
 - Insérez les projets numéro 3 et 5 pour l'employé dans la table `works_on` avec le `Ssn` '123456789' et attribuez 10 heures de travail sur chaque projet.
 - Supprimez le département numéro 5 dans la table `department`.
 - Insérez un projet "Tera Data" dans le département Research. Ajoutez vous et votre coéquipier comme les employés qui travaillent chacun 10 heures sur ce projet.

4. Affinement des contraintes d'intégrité référentielle

- 5.1. Que se passe-t-il si on essaie de supprimer dans la table `employee` un employé qui est superviseur ? Testez à l'aide des commandes SQL et expliquez le comportement. Modifiez les contraintes d'intégrité référentielle pour pouvoir supprimer un employé qui est superviseur, mais dans ce cas, tous les supervisés de cet employé auront NULL comme valeur de superviseur.
- 5.2. Que se passe-t-il si on essaie de mettre à jour dans la table `department` le numéro d'un département ? Testez à l'aide des commandes SQL et expliquez le comportement. Modifiez les contraintes d'intégrité référentielle pour pouvoir mettre à jour le numéro d'un département et assurer que tous les employés qui travaillent dans ce département auront le numéro de département mis-à-jour.