

Labo2 : Modèle Relationnelle

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 introduit dans le cours.

Plus précisément, les points étudiés dans ce laboratoire sont :

1. Création d'une base de données avec le SGBD MySQL (sous MAMP, WAMP ou EasyPHP).
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.

1. Transformation de schéma conceptuel vers le schéma relationnel

Prenez le schéma conceptuel de l'exemple COMPANY en Annexe 1.

- Vérifiez si les relations de la base de données COMPANY fournies en Annexe 1 représentent une transformation correcte du schéma conceptuel donné.
- Complétez le schéma relationnel si nécessaire. Indiquez vos modifications.

2. Implémentation des tables sous MySQL

Prenez le schéma relationnel de la base de donnée COMPANY fourni en Annexe 2.

- Définissez avec le SGBD MySQL, une base de données ainsi que les différentes tables qui la composent. Vous devez respecter le type exact des champs fourni dans l'Annexe.

Attention : Pour cette première phase n'ajouter pas encore de contraintes d'intégrité référentielle sur les champs.

Indiquez les commandes SQL utilisées.

3. 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 votre base de données. Prenez connaissance des différents tuples ajoutés à votre 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 ? Est-ce que ce comportement est juste ? Critiquez le résultat obtenu.

- Supprimez le département numéro 5 dans la table `department`. Que constatez-vous ? Est-ce que ce comportement est juste ? Critiquez le résultat obtenu.

4. 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 2). 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 et observez les résultats et expliquez le comportement :
 - Dans la table `employee`, supprimez le tuple correspondant à `Ssn = '999887777'`.
 - Dans la table `employee`, modifiez le `Dno` du tuple qui a le `Ssn = '999887777'` à 7.
 - 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.

5. Affinement des contraintes d'intégrité référentielle

- 5.1. Que se passe-t-il si on essaie de supprimer dans la table `employé` 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.