

Labo 2

Implémentation d'un schéma relationnel

1 But de l'exercice

Il s'agira dans cet exercice d'implémenter un schéma relationnel et mettre en oeuvre les contraintes d'intégrité référentielles appropriées. Ceci se fera à l'aide du SGBD PostgreSQL. L'ensemble des exercices se feront sur la base du schéma de COMPANY qui a été introduit en cours.

1.1 Points étudiés

1. Création d'une base de données avec le SGBD PostgreSQL et un client SQL (par exemple Data-Grip)
2. Définition de la structure des tables
3. Définition des différentes contraintes d'intégrité
4. Insertion de données dans les tables
5. Tester les contraintes d'intégrité

1.2 Implémentation des tables sous PostgreSQL

L'implémentation des tables se fera en se basant sur le schéma relationnel de la base de données Company fourni en annexe. Vous devez créer les tables manuellement tout en respectant le type exacte des champs comme décrit dans le schéma. Les tables doivent être créées dans la base de données *bdr* et dans le schéma *company*. Le schéma et les tables doivent être créés par le rôle *BDR* (pas le rôle *postgres*).

Attention : N'ajoutez pas encore les contraintes d'intégrité référentielles (clés étrangères) ici !

1.3 Insertion des données

Les commandes d'insertion vous sont fournies dans un fichier *.sql*. Exécutez les fichiers afin de remplir votre base de données. Prenez ensuite connaissance des différents tuples qui y ont été ajoutés puis effectuez les modifications suivantes en utilisant les commandes SQL.

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

1.4 Implémentation des contraintes d'intégrité

1. Premièrement, il vous faut vider toutes vos tables des différents tuples existants. Indiquez les commandes SQL utilisées.
2. Ensuite, pour chacune des tables, ajoutez les contraintes d'intégrité référentielle (voir Figure 1 du schéma relationnel COMPANY)
3. Une fois toutes vos contraintes établies, vous devez peupler à nouveau vos tables avec les fichiers .sql fournis.

- (a) Est-ce que ceci est possible? Si non, expliquez le problème.
- (b) Que faut-il faire pour pouvoir insérer des tuples?

Remarque : Pour exécuter une migration, il peut être utile de désactiver les contraintes d'intégrité référentielles. Deux méthodes existent (les deux nécessitent d'être super-utilisateur) :

- (a) Désactiver les contraintes sur chaque table avec :

```
ALTER TABLE ma_table DISABLE TRIGGER ALL;
```

N'oubliez ensuite pas de les réactiver avec :

```
ALTER TABLE ma_table ENABLE TRIGGER ALL;
```

- (b) Une autre manière est en exécutant

```
SET session_replication_role = 'replica';
```

avant la migration, puis en réactivant les triggers à l'aide de

```
SET session_replication_role = 'origin';
```

4. On souhaite insérer un nouveau département "IT" (avec `dnumber = 10`) et son manager qui est un nouvel employé (Steve Jobs avec `ssn = '555444333'`) qui travaille dans ce nouveau département. Expliquer la difficulté et proposer une solution.

Remarque : Par défaut, les contraintes d'intégrité référentielle sont vérifiées après chaque *statement*. On peut exécuter des insertions dans plusieurs tables en un *statement* en utilisant la clause `WITH : WITH first_insert AS (INSERT INTO table1 ...) INSERT INTO table2 ...`

Il est possible de retarder la vérification des contraintes (voir "Chapitre 3 : Annexe – contraintes déférables" sur Cyberlearn).

5. Exécutez les commandes suivantes, observez les résultats et expliquez le comportement
 - (a) Dans la table `employee`, modifiez le `dno` du tuple qui a le `ssn = '999887777'` à 7
 - (b) Dans la table `employee`, supprimez le tuple correspondant à `ssn = '999887777'`
 - (c) Insérez les projets numéro 3 et 5 dans la table `works_on` pour l'employé avec le `ssn = '123456789'` et attribuez 10 heures de travail sur chaque projet
 - (d) Supprimez le département numéro 5 dans la table `department`

1.5 Affinement des contraintes d'intégrité référentielles

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.
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.