

ISEN

ALL IS DIGITAL!

OUEST

TP n°3 - Bases de données Requêtes SQL de manipulation de données

benoit.lardeux@isen-ouest.yncrea.fr

Objectifs

- gérer le "commit" manuellement
- insérer des données
- supprimer des données en maintenant l'intégrité référentielle

Mise en place

Ouvrez une session *PostgreSQL* et allez dans la base de données créée lors du Tp1.

Attention : en début de session, avant toute requête de manipulation de données, il faut commencer une transaction avec la commande *start transaction*;. Lors d'une transaction, une manipulation peut-être annulée en utilisant la commande *rollback*;. En cours de transaction, vous ferez des *commit* explicites (c'est-à-dire que vous déciderez vous-même de la validation ou non d'une transaction). Les manipulations sont automatiquement committées à la fin de la transaction (*end transaction*;;).

1 Insertion de données

Attention : Ne faites pas de *commit* avant que ce ne soit demandé.

- a) Insérer un nouvel employé, de fonction *directeur*, embauché à la date d'aujourd'hui, dans le service 40. Son supérieur hiérarchique est le PDG (*numemp_sup=839*). Vous pouvez choisir son numéro (*numemp*), son nom et son salaire.
- b) Insérer un second employé, de fonction *ingenieur*, encore embauché à la date d'aujourd'hui dans le service 40. Son supérieur hiérarchique est le directeur embauché précédemment.
- c) Vérifiez, avec un *select*, que vous voyez bien ces deux employés.
- d) Ouvrez une seconde session sur votre base de données (sans fermer la première, qui reste votre session principale) et vérifiez que ces deux employés n'existent pas dans cette nouvelle session (ils n'ont pas encore été "committés").
- e) Faites, dans la session principale, un *commit* pour valider la transaction (=insertion des deux nouveaux employés).
- f) Vérifiez, avec un *select*, que maintenant vous voyez bien ces deux nouveaux employés dans la seconde session ouverte.
- g) Dans la session principale, insérer un nouvel employé, de fonction *directeur*, embauché à la date d'aujourd'hui, dans le service 50. Est-ce que *PostgreSQL* l'accepte? Si non, expliquer, puis passer au paragraphe 2 suivant

2 Mise à jour

- a) Augmentez le salaire de tous les employés de 10%. Vérifiez par un *select* tous les salaires
- b) Faites un *rollback*
- c) Vérifiez par un *select* (avant que votre chef comptable se réveille) que les salaires des employés n'ont pas été modifiés
- d) Mettez à jour la colonne *numemp_sup* de l'employé *Dupont* : donnez-lui la valeur 999. Est-ce que *Postgresql* l'accepte ? Expliquer
- e) Faites un *commit*. Regardez, avec un *select*, si *Dupont* a été modifié. Si non, pourquoi ?

3 Suppression de lignes

- a) Supprimez les employés du service 40. Est-ce que *Postgresql* l'accepte ?
- b) Supprimez le PDG. Est-ce que *Postgresql* l'accepte ? Si non, trouver une méthode pour pouvoir réaliser cette suppression en maintenant l'intégrité référentielle de la base
- c) Faites un *rollback*

4 Bonus

- a) Recréez la clé étrangère sur *numemp_sup* (`alter table drop foreign key ...` suivi de `alter table add foreign key ...`, cherchez la syntaxe précise en ligne), pour y rajouter une action référentielle de type `on delete cascade`
- b) Réessayez de supprimer le PDG