

I54 - TP Transactions

Partie 1

psql est une interface en mode texte pour PostgreSQL
--

On vous demande de faire un compte rendu de chaque TP à rendre à la fin du TP

A. Transaction unique

Comprendre commit et rollback

- 1) Créez la table `Département` (DID, Libelle) où la clé primaire DID est un code à deux lettres, Libelle une chaîne de 20 caractères maximum.

Créez la table `Employe` (EID, Nom, Dept) où la clé primaire EID est un nombre à deux chiffres, Nom une chaîne de 20 caractères maximum et Dept un code à deux lettres. Dept est clé étrangère référençant `Département`.
- 2) Créez un département. Consultez la table `Département`. Effectuez un `rollback`. Consultez la table `Département`. Que se passe – il ?
- 3) Débutez une transaction (`begin transaction`). Créez un département. Consultez la table `Département`. Effectuez un `rollback`. Consultez la table à nouveau. Que se passe – il ?
- 4) Débutez une transaction. Créez un département. Consultez la table `Département`. Effectuez un `Commit`. Consultez la table à nouveau. Que se passe – il ?
- 5) Débutez une transaction. Créez un département. Consultez la table `Département`. Effectuez un `Commit`. Consultez la table à nouveau. Effectuez un `rollback`. Consultez la table à nouveau. Que se passe – il ?
- 6) Débutez une transaction. Insérez un nouveau département et quitter votre session. Ouvrez une nouvelle session. La dernière Transaction a-t-elle été validée ?
- 7) Insérez un nouveau département et quitter votre session. Ouvrez une nouvelle session. La dernière Transaction a-t-elle été validée ?
- 8) Commencez une transaction. Vider le contenu de la table `Département` et effectuez un `rollback`. Consultez la table.
- 9) Commencez une transaction. Vider le contenu de la table `Département` et effectuez un `Commit`. Consultez la table.

Comprendre les mises à jour et la transaction (vider les 2 tables)

- 10) Créez un département <'C1', 'Info'>. Créez un employé <01, 'Quentin', 'C1'> appartenant au département C1.
- 11) Créez un département <'C1', 'Informatique'>. Que se passe-t-il ?
- 12) Créez un employé <02, 'Grégoire', 'C2'>. Que se passe-t-il ?
- 13) Supprimez le département C1. Que se passe-t-il ?
- 14) Modifier la table Employé en supprimant la clé étrangère (ALTER TABLEDROP CONSTRAINT ...). Supprimez le département C1. Interrogez les deux tables. Expliquez les résultats.
- 15) Vider les deux tables.
- 16) Modifier la table Employé : Recréez la clé étrangère en mode ON DELETE CASCADE (ALTER TABLEADD CONSTRAINT ...). Recréer le département C1 et l'employé 01.
- 17) Supprimez le département C1. Interrogez les deux tables. Expliquez les résultats.
- 18) Recréer le département C1 et l'employé 01.

ALTER TABLE ... ADD CONSTRAINT ... NOT DEFERRABLE => n'est jamais déférable

ALTER TABLE ... ADD CONSTRAINT ... DEFERRABLE => est déférable

SET CONSTRAINTS <...> DEFERRED Impose que les contraintes mentionnées dans <...> et déférables soient déférées

Le déferrement n'intervient que dans une transaction et si contraintes déférée il y a, alors la vérification de la contrainte sera effectuée au moment du COMMIT et non à chaque ordre SQL.

A noter : c'est le seul endroit dans SQL où lançant un COMMIT on peut obtenir un ROLLBACK. En effet si une seule des contraintes déférée est violée alors la transaction est annulée...

Le terme déferé : le mot provient du vocabulaire juridique et indique un transfert de responsabilité... L'ordre SQL INSERT, UPDATE ou DELETE, délaisse l'application des contraintes dont il doit habituellement s'occuper pour en confier la vérification à la transaction.

- 19) Supprimez à nouveau la clé étrangère. Recréez la clé étrangère sans le mode ON DELETE CASCADE mais avec le mode DEFERRABLE.
- 20) Activer le mode DEFERRED avec la commande SET CONSTRAINTS <nom_de_votre_contrainte> DEFERRED;
- 21) Supprimez le département C1. Expliquez le résultat.

B. Contrôle de la Concurrency

Pour préparer la partie B, supprimer la clé étrangère pour la recréer sans mode DEFERRABLE ni On DELETE CASCADE. Videz la table Employé, puis la table Département.

- 1) Connectez-vous à votre base à partir de deux sessions différentes, simultanément.
- 2) Dans chacune des deux fenêtres, insérez un département (code C1 pour transaction 1, code C2 pour transaction 2). Consulter la table depuis chaque fenêtre. Que constate-t-on ?
- 3) Dans chacune des deux fenêtres, commencez une transaction.

Insérez un département dans la session 1. Consultez la table. N'effectuez pas encore de commit.

Insérez un autre dpt dans la session2. Commitez dans la session2.

Consulter la table dans les 2 sessions. Que constate-t-on ?

Commitez dans la session 1.

Consulter la table dans les 2 sessions. Que constate-t-on ?

- 4) Dans chacune des deux fenêtres, commencez une transaction.

Insérez un département dans la session 1. N'effectuez pas encore de commit.

Insérez **le même** département dans la session 2. Que constatez vous ?

Commitez dans la session 1. Que constatez-vous dans la session 2 ?

Terminez la session 2.

Consulter la table dans les 2 sessions. Que constate-t-on ?

- 5) Dans chacune des deux fenêtres, commencez une transaction.

Insérez un département dans la session 1. N'effectuez pas encore de commit.

Insérez **le même** département dans la session 2. Que constatez vous ?

Annulez l'insertion dans la session 1. Que constatez-vous dans la session 2 ?

Terminez la session 2.

Consulter la table dans les 2 sessions. Que constate-t-on ?

- 6) Dans l'un des 2 sessions, vider le contenu de la table Département (c'est un autocommit).
- 7) Dans la session 1, commencez une transaction et insérez le département <'C1', 'Info'>. Effectuer un Commit. Consultez la table Département.

- 8) Depuis la session 2, commencez une transaction et modifiez le libellé de ce département en 'Informatique' et validez
- 9) Depuis la session 1, consultez la table Departement à nouveau.
- 10) Depuis la session 1 commencez une transaction. modifiez le libellé de ce département en 'Info' sans validez encore ; faites la même chose dans la session 2, mais pour modifier ce libelle en 'Biologie'. Que se passe-t-il ?
- 11) Annulez la modification dans la première session. Consultez les tables dans les deux sessions. Que se produit-il?
- 12) Validez dans la première session. Consultez les tables dans les deux sessions. Que se produit-il?
- 13) Dans l'un des 2 sessions, vider le contenu de la table Departement (c'est un autocommit).
- 14) Depuis la session 1, insérez le département <'C1', 'Info'> et depuis la session 2 Insérez un employé <01, 'Henri', 'C1'>. Que se passe-t-il ?
- 15) Annulez l'insertion dans la session 1. Que se passe-t-il dans la session 2?
- 16) Depuis la session 1, insérez le département <'C1', 'Info'> **puis** depuis la session 2 Insérez un employé <01, 'Henri', 'C1'>. Consultez les deux tables depuis les deux sessions.
- 17) Depuis la session 1 supprimez le département C1. Que se passe-t-il ?
- 18) Ajoutez depuis la session 1 l'employé <02, 'Grégoire', 'C1'>. Validez la transaction.
- 19) Depuis la session 1, modifiez le nom de l'employé 01 en Coleen, ne validez pas encore. Depuis la fenêtre 2, modifiez le nom de l'employé 02 en Camille, ne validez pas encore. Consultez les tables. Que se passe-t-il ?
- 20) Validez la session1, consultez la table. Validez la session 2, consultez la table. Consultez à nouveau la table dans la session 1.