NAAJI Dorian & ARMANET Nathan – 3A INFO GROUPE 2

POLYTECH LYON

Dorian.naaji@etu.univ-lyon1.fr

NATHAN.ARMANET@ETU.UNIV-LYON1.fr

Rapport TP03

BD1

# Partie 1 : Transactions

## Atomicité d’une transaction courante

### Question 1

« ROLLBACK » a supprimer tous les changements effectués à la table (insertion, modification et suppression de données) mais pas la table elle-même.

### Question 2

« COMMIT » a enregistré les modifications apportées à la table, ce qui à empêcher « ROLLBACK » de les supprimer.

### Question 3

« EXIT » ou « QUIT » ferme simplement la connexion à la base de données sans apporter de modification à celle-ci.

### Question 4

La fermeture brutale de la session est comme un « ROLLBACK » ; elle supprimer toutes les modifications de données non enregistrées.

### Question 5

Cette fois ci, le « ROLLBACK » ne supprime rien. Cela est dû au fait que l’on a modifier la table en elle-même (ajout, modification ou suppression de colonnes ou contraintes). Ces modifications enregistrent les données tel qu’elle sont avant de modifier la table.

### Question 6

Une transaction courante est un changement dans les données d'une table. Elles sont validées par « COMMIT » et annulé par « ROLLBACK ». Toutes modification de la table entraine un « COMMIT ».

## Plusieurs sessions sur un seul compte de BD et transactions concurrentes

### Question 1

On constate que le contenu de la base est le même quel que soit la fenêtre utilisée

### Question 2

On voit juste les transactions effectuées à partir de la même fenêtre

### Question 3

On voit les modifications de la table initiale apporté par l'utilisateur qui a créé la nouvelle table.

On peut voir la nouvelle table sur les deux fenêtres.

On voit seulement les transactions effectuer à partir de notre fenêtre

### Question 4

### Question 5

### Question 6

### Question 7

### Question 8

### Question 9

## Droits/privilèges entre deux compte d’une même base de données

### Question 1

### Question 2

### Question 3

### Question 4

### Question 5

# Partie 2 : PLSQL

## Copier dans votre compte les trois table Dept, Emp et Salgrade de Scott.

## Donner les requêtes SQL pour les questions suivantes :

### Donner le nom des employés dirigés directement par ‘King’.

### Donner le nom des employés qui dépendent (directement ou non) de Jones.

### Donner le nom des employés dont dépend (directement ou non) Jones.

### Donner le nom des employés dépendant de Blake, sauf Blake lui-même.

### Donner le nom des employés qui dépendent de King, sauf ceux qui dépendent de Blake.

## Ecrire une fonction PLSQL de paramètre un numéro de département et qui retourne le nombre d’employés de ce département. On pourra typer le numéro de département par *Emp.DeptNo%TYPE*.

## Ajouter la colonne NbEmps dans votre table Dept, qui contiendra le nombre d’employés de chaque département. On mettra à jour NbEmps de deux façons possibles:

### En utilisant la fonction stockée définie dans la question précédente

### Sans l’utiliser mais en utilisant un curseur associé à Dept suivant l’exemple suivant :

## Créer un déclencheur qui pour chaque ajout ou suppression d’un employé ou mise à jour du département d’un employé, modifiera le nombre d’employés dans la table Dept.