TP 3- BD1: Transactions / PLSQL

PARTIE 1 : Transactions

Atomicité d'une transaction courante

1° Créer une table et y insérer trois ou quatre lignes et les voir, modifier une ligne, en supprimer une autre, enfin **annuler** les mises à jour venant d'être effectuées (en écrivant « ROLLBACK ; »). Voir à nouveau le contenu de la table et sa structure.

2° Insérer à nouveau trois ou quatre lignes, les modifier et les détruire partiellement, puis **valider** (en écrivant « COMMIT ; ») ces mises à jour, puis déclencher un ROLLBACK. Que s'est-il passé ? Maintenant détruire les données de votre table et valider.

3° Insérer à nouveau dans votre table vide trois ou quatre lignes et clore la transaction par un EXIT ou un QUIT. Que s'est-il passé ?

4° Dans votre table, insérer à nouveau deux ou trois lignes et fermer brutalement votre session, puis rentrer à nouveau dans votre compte. Les données saisies ont-elles été préservées ?

5° Insérer à nouveau deux ou trois lignes dans la table, puis adjoindre une nouvelle colonne à sa table (ou plus généralement émettre n'importe quelle commande de description des données) et essayer d'annuler les dernières insertions..

6° En conclusion, qu'est-ce qu'une transaction courante et comment valider ou annuler une transaction ?

Plusieurs sessions sur un seul compte de BD et transactions concurrentes

1° Se connecter à son compte à partir d'une autre ***fenêtre*** (ou ***écran*** sur un autre poste de travail) et constater à travers cette nouvelle fenêtre le contenu du compte.

2° Insérer dans chacune des deux fenêtres deux ou trois lignes distinctes. Que voit-on de l'autre fenêtre ?

3° Créer dans l'une des deux fenêtres ouvertes une nouvelle table et y insérer par chacune des fenêtres deux ou trois lignes. Que voit-on de la table initiale et de la table nouvellement créée et des lignes insérées ?

4° Détruire la nouvelle table. Que se passe-t-il ? Comment la détruire ?

5° Adjoindre à votre table une ***clé***. Insérer dans cette table restructurée à partir des deux fenêtres une ligne ayant la même valeur pour chaque attribut de la clé. Qu'arrive-t-il ? Emettre un ROLLBACK. Que devient le blocage ?

6° Clore la session dans la fenêtre d'insertion de la ligne par un EXIT ou un QUIT. Que constate-t-on dans la fenêtre restante? Faire encore des mises à jour (des modifications et des destructions de ligne en plus des insertions) dans la table à partir de la fenêtre restante et sortir normalement.

7° Ouvrir une nouvelle session en utilisant une seule fenêtre. La dernière transaction a-t-elle été validée ?

8° Insérer encore une ligne, puis créer une nouvelle table, y insérer une ligne. Emettre un ROLLBACK. Que sont devenues les deux tables et les deux lignes insérées ?

9° Successivement insérer encore une ligne dans la première table, éliminer la dernière table créée et émettre un nouveau ROLLBACK. Qu'est devenue la dernière ligne insérée ?

Droits/privilèges entre deux comptes d'une même base de données

Les groupes de TP travaillent dans cette partie deux par deux.

1° Chaque groupe donne le droit à l'autre groupe de consulter l'une de ses tables en émettant un « GRANT SELECT ON *Table* TO *autreGroupe*; ». Vérifier que ce privilège a été donné à l'aide des vues ALL_OBJECTS ou ALL_TABLES, et USER_TAB_PRIVS et accéder à cette table que vous pouvez lire, mais qui ne vous appartient pas.

2° Quand l'autre groupe fait une mise à jour sur sa table, que voyez-vous ?

3° Essayer d'insérer une ligne dans la table de l'autre groupe.

4° Pour arriver à insérer, l'autre groupe doit vous donner le droit INSERT. Reprendre alors l'insertion.

5° Réaliser une jointure d'une de vos tables avec une table de vos camarades.

PARTIE 2 : PLSQL

Nous considérons la base de données d'Oracle stockée dans le compte Scott :

Emp(EmpNo, EName, Job, Mgr, Hiredate, Sal , Comm, DeptNo)

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17/12/80	800		20
7499	ALLEN	SALESMAN	7698	20/02/81	1600	300	30
7521	WARD	SALESMAN	7698	22/02/81	1250	500	30
7566	JONES	MANAGER	7839	02/04/81	2975		20
7654	MARTIN	SALESMAN	7698	28/09/81	1250	1400	30
7698	BLAKE	MANAGER	7839	01/05/81	2850		30
7782	CLARK	MANAGER	7839	09/06/81	2450		10
7788	SCOTT	ANALYST	7566	19/04/87	3000		20
7839	KING	PRESIDENT		17/11/81	5000		10
7844	TURNER	SALESMAN	7698	08/09/81	1500	0	30
7876	ADAMS	CLERK	7788	23/05/87	1100		20
7900	JAMES	CLERK	7698	03/12/81	950		30
7902	FORD	ANALYST	7566	03/12/81	3000		20
7934	MILLER	CLERK	7782	23/01/82	1300		10

Dept(DeptNo, DName, Loc)

DeptNo	DName	Loc
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

SalGrade(Grade, LoSal, HiSal)

GRADE	LOSAL	HISAL
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999

1. Copier dans votre compte les trois table Dept, Emp et Salgrade de Scott.
2. Donner les requêtes SQL pour les questions suivantes :
    a. Donner le nom des employés dirigés directement par 'King'.
    b. Donner le nom des employés qui dépendent (directement ou non) de Jones.
    c. Donner le nom des employés dont dépend (directement ou non) Jones.
    d. Donner le nom des employés dépendant de Blake, sauf Blake lui-même.
    e. Donner le nom des employés qui dépendent de King, sauf ceux qui dépendent de Blake.

3. 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*.

On utilisera la commande SQL:

CREATE [OR REPLACE] FUNCTION *nom_fonction*[(argument *typeDonnée,...)]* RETURN *typeDonnée* IS/AS *blocPL/SQLsansLeMotDECLARE;*

Puis appeler cette fonction stockée en utilisant un bloc PL/SQL.

4. 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:
    a. en utilisant la fonction stockée définie dans la question précédente
    b. sans l'utiliser mais en utilisant un curseur associé à Dept suivant l'exemple suivant :

```
DECLARE
CURSOR C1 IS SELECT * FROM Table
... ;
BEGIN
FOR C1_enr IN C1 LOOP ... ;
END LOOP;
END;
/
```

5. 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.

Créer une procédure pour mettre à jour le numéro de département de toute une équipe de travail depuis sa hiérarchie (l'ensemble des employés depuis la racine). Cette procédure prendra en paramètre le numéro d'un employé faisant parti de l'équipe (pas forcément la racine, mais toute l'équipe devra changer de département depuis sa racine).