

## TD 5

**Situation** On travaille toujours sur la situation de l'agence de voyages des TD précédents.

*Personne*(*nper, nom, âge, solde*) *Séjour*(*nséj, dest, activité, prix, nbplaces*) *Voyage*(*nséj, nper, jour*)

### Partie sur table (1h)

On commence par résoudre les exercices sur papier en pseudo-langage (c'est-à-dire sans les détails de syntaxe), et les corriger au tableau, toujours en pseudo-langage (sans les détails de syntaxe).

*Remarque* Sortez vos transparents de tous les cours, ainsi que les exemples distribués sur papier. Avant chaque exercice, commencez par indiquer les transparents du cours et les exemples papier concernés, et relisez-les brièvement.

### 1 Confidentialité et encapsulation (15mn table/15mn machine)

On veut appliquer dans le cadre des procédures stockées PL/SQL le principe de confidentialité déjà vu pour les séquences pures d'ordres SQL dans l'exercice 1 du TD 2. On veut donc maintenant encapsuler les tables sujettes aux mises à jour dans la situation exemple des TD.

1. Dans cette question uniquement, on se replace artificiellement dans la situation de la fin du TD 2, avant l'introduction de PL/SQL. Vous êtes le programmeur de l'application (et en même temps pour simplifier dans cet exercice le gérant de l'agence). On suppose l'existence d'un compte SGBD *client* utilisé par tous les clients de l'agence. Rappelez les ordres SQL gérant la confidentialité dans cette situation.
2. Retirez les droits sur ces tables à tout le monde (rappel : il ne suffit pas de retirer *all* et *public*).
3. Faites en sorte que ce soit une procédure qui effectue ces mises à jour.
4. Donnez le droit d'exécuter cette procédure à tout le monde.
5. Demandez à votre voisin de tester l'accès direct et l'appel de cette procédure. Vérifiez le résultat en interrogeant les tables concernées. Votre table est-elle encapsulée ?

### 2 Indépendance des niveaux et procédures stockées (20/15)

On veut appliquer dans le cadre des procédures stockées PL/SQL le principe d'indépendance des niveaux déjà vu pour les séquences pures d'ordres SQL dans l'exercice 2.1 du TD 2.

1. On ne gèrera pas l'indépendance des niveaux pour les tables sujettes aux mises à jour (rappelez pourquoi), mais vous l'indiquerez en commentaire dans le script.
2. Appliquez le principe d'indépendance des niveaux pour les tables en lecture de votre application. Donnez explicitement un jeu de tests rigoureux et appliquez-le. L'indépendance des niveaux est-elle satisfaite ?

### 3 Triggers (20/20)

1. *Familiarisation. Uniquement sur machine.* Tapez et faites tourner l'exemple distribué sur papier en cours.
2. Le cahier des charges de l'application englobante de notre micro-application demande de maintenir dans *Voyage* uniquement certains voyages (non encore faits, ainsi que d'autres sur lesquels certains traitements de sont pas terminés). Divers programmes de l'application englobante, ou des experts BD à la main, détruisent donc de temps à autres des voyages.  
Pour des raisons juridiques, le cahier des charges demande aussi d'archiver chaque voyage, au moment où il est détruit dans *Voyage*, en l'insérant dans une table d'archivage *Archive\_Voyage*. Mais l'application est grosse, elle évolue, des programmes supprimants parfois des voyages sont créés, de nombreux experts BD interviennent en interactif sur la base... Il n'est donc pas possible de maîtriser toutes ces interventions (programmes, manuelles) et de les forcer à inclure dans leur code l'opération d'archivage.
  - (a) Concevez et codez une solution à ce problème. Justifiez. Testez avec un ordre *delete* supprimant deux ou trois voyages.
  - (b) *Uniquement sur table (mais vous pouvez le faire sur machine tout à la fin s'il vous reste du temps)*. Testez en effectuant les suppressions depuis une procédure stockée simple que vous écrirez. Le comportement est-il le même ?

### 4 Packages (5/10)

1. Placez les procédures de votre application dans un package. Appelez-la sous SQL\*Plus.
2. Donnez à votre voisin les droits sur ce package, mais pas sur la procédure. Qu'il appelle votre procédure par l'intermédiaire du package, pour tester.

### Partie sur machine (1h)

Créez un répertoire *td6*. Programmez sur machine les exercices de la partie sur table en respectant l'ordre.

### Suivi projet

Il n'est pas prévu de consacrer du temps au projet pendant le TD. L'enseignant discutera simplement quelques minutes avec les coordinateurs, en particulier concernant l'organisation du groupe de TD et de son planning.

### Travail personnel

Le même qu'aux TD précédents.