Université de Lorraine
Département Informatique, Master 1

TD UE SGBD

Un compte-rendu pourra être demandé à chaque fin de séance.

TRES IMPORTANT : La plus extrême rigueur et le plus grand professionalisme sont exigés pour le bon déroulement des sessions de manipulation. **Lire très attentivement les recommandations ci-dessous.**

1 Avant Propos

1. Vous avez les droits super-utilisateur sur les machines de la salle de TP : ces droits ne doivent être utilisés que pour les manipulations nécessaires aux TPs d'administration Oracle. *Toute utilisation abusive de ces droits, hors de ce cadre, sera considérée comme une faute grave : il n'y aura pas de circonstances "atténuantes".*

2. Les postes de travail ne sont pas banalisés : chacun doit noter le nom ou le numéro du poste de travail qu'il utilise afin d'y effectuer toutes les séances.

3. La plupart des sessions peuvent se dérouler sous *SQL*Plus*. Elles constituent des applications des chapitres du cours.

4. Les sessions n'étant pas indépendantes, il est recommandé de conserver les commandes de chacune d'elles à des fins de ré-utilisation.

5. Chaque étudiant utilisera sa propre base de données (voir le paragraphe 5, page 5).

6. Au début de chaque séance, s'assurer que vous allez travailler sur la "bonne" base et la "bonne" instance en vous assurant que la variable d'environnement *ORACLE_SID* est positionnée à la valeur du nom de votre base (*echo $ORACLE_SID*). Si cela n'est pas le cas, lui donner la bonne valeur au niveau système d'exploitation (export *ORACLE_SID*=Nom de votre base.

7. Les binaires Oracle se trouvent sous /opt/oracle/product/11.2.0/db_1/bin : Modifier la variable d'environnement PATH dans le .bashrc en conséquence par echo 'export PATH=$PATH:/opt/oracle/.../bin' >> chemin vers le .bashrc.

8. Créer un répertoire par session et (1) y lancer sqlplus et (2) y inclure les résultats des manipulations car ils peuvent être demandés à tout instant.

2 Eléments d'information pour les connexions et les déconnexions

```
Compte linux : oracle ; Mot de passe : oracle
```

```
Compte oracle : sys ; Mot de passe : oracle
SID : Nom de votre base (voir ci-dessus).

###########  EN DEBUT de SEANCE  ###########
# Démarrer le processus ``d'écoute'' (Oracle Net Listener)
[oracle@localhost]$ lsnrctl start

# Positionnement du SID
[oracle@localhost]$ export ORACLE_SID=$\ldots$

# Démarrage  de l'instance Oracle (sous sqlplus):
[oracle@localhost]$ sqlplus /nolog
SQL> connect sys as sysdba
-- Le mot de passe de sys est alors requis
-- Une alternative:
SQL> connect sys/MotDePasse as sysdba
-- Lancement de l'instance Oracle
SQL> startup
     ...................

###########  EN FIN de SEANCE  ###########
-- Arrêt de l'instance Oracle
SQL> shutdown
-- Quitter sqlplus
SQL> exit
-- Arrêter le processus ``d'écoute'' (Oracle Net Listener)
[oracle@localhost]$ lsnrctl stop
```

1. A la fin de chaque séance, ne pas oublier
   (a) d'arrêter l'instance Oracle (*shutdown*[1]),
   (b) d'arrêter le *listener* (*lsnrctl stop*),
   (c) d'arrêter la machine après déconnexion de Linux.

3 Rappels et indications

1. *describe* nom d'objet : rend le schéma de l'objet indiqué
   - Exemple : *describe dictionary* : rend les noms et les types des colonnes de la table *dictionary*.

[1] Un *shutdown* pouvant parfois être très lent, on pourra utiliser un *shutdown "bulldozer" : shutdown immediate ou shutdown abort.*

```
SQL> describe dictionary;
Nom            NULL ?   Type
-------------- -------  ---------------
TABLE_NAME              VARCHAR2(30)
COMMENTS                VARCHAR2(4000)
```

2. Recherche à l'aide d'un modèle : Par exemple, trouver toutes les tables du dictionnaire qui peuvent avoir trait au *log* :

   select table_name, comments from dictionary
   where upper(table_name) like upper('%log%');

   Le dictionnaire comprenant plus de 1000 (mille) tables et vues (voir le nombre exact en faisant "select count(*) from dictionary"), il est préférable d'utiliser des expressions comme celle ci-dessus en lieu et place d'un *"select * from dictionary"*. Par ailleurs, la table *dict_columns* permet de connaître la signification des colonnes des tables du dictionnaire.

   Attention : Dans la comparaison de chaînes de caractères, Oracle est sensible aux majuscules/minuscules : d'où l'utilisation de la fonction *upper* ci-dessus.

3. Formattage des résultats :

```
 - SANS Formattage:
    SQL> select name, TS# from v$datafile;

    NAME
    ------------------------------------------------------------------
     TS#
    ----------
    D:\ORACLE\PRODUCT\10.2.0\ORADATA\MABASE\SYSTEM01.DBF
       0
    D:\ORACLE\PRODUCT\10.2.0\ORADATA\MABASE\UNDOTBS01.DBF
       1
                    ..............
- AVEC Formattage:
     SQL> column name format A55;
     /* Axx: chaîne de xx caractères alphanumériques    */
     SQL> column ts# format 9999;
     /* xx9: xx positions numériques         */

     SQL> select name, TS# from v$datafile
     NAME
     -----------------------------------------------------
```

```
D:\ORACLE\PRODUCT\10.2.0\ORADATA\MABASE\SYSTEM01.DBF         0
D:\ORACLE\PRODUCT\10.2.0\ORADATA\MABASE\UNDOTBS01.DBF      1
                .............
```

4 Rappel : Initiation à Sql*Plus

1. **Lancer SQL*Plus**: au niveau système, : commande sqlplus suivie du nom de l'utilisateur oracle; le mote de passe de l'utilisateur oracle est alors demandé.
2. **Entrer et exécuter 'directement' des requêtes** :
   - Entrer la requête,
   - Un retour à la ligne produit un nouveau numéro de ligne pour la requête,
   - Un point-virgule (;) suivi d'un retour à la ligne provoque l'exécution de la requête.
3. **Voir la dernière requête** : list
4. **Ré-exécuter la dernière requête** : / (Oracle ne conserve qu'une seule requête dans son tampon de requêtes).
5. **Sauvegarder le contenu du tampon dans un fichier** : SAVE Nom-Du-Fichier **CREATE**
6. **Ajouter le contenu du tampon** (dernière requête entrée) dans un fichier (ESSAI2.SQL, par exemple) : SAVE ESSAI2.SQL **APPEND**.
7. **Editer et exécuter des requêtes** :
   - **Choisir un éditeur de requêtes** : une fois connecté, positionner la variable _EDITOR à votre éditeur de texte préféré. Par exemple : DEFINE_EDITOR = emacs
   - **Editer une requête dans un fichier** (fichier essai.sql, par exemle) : EDIT essai.sql
   - **Sauvegarder le fichier contenant la requête** : CTRL-X CTRL-S, sous emacs
   - **Exécuter le contenu du fichier** : Après fermeture de l'éditeur (CTRL-X CTRL-C, sous emacs), retour à Sql*Plus puis START essai.sql
   - Si la requête est correcte, la liste des résultats s'affichera. Sinon, ré-éditer le fichier essai et modifier la requête puis ré-exécuter.
8. **Enregistrer unr requête et son résultat dans un fichier** (par exemple dans result.res) : SPOOL result.res, puis exécuter la requête.
9. **Retourner au seul affichage sur écran** : SPOOL OFF
10. **Quitter Sql*Plus** : exit.

5 Création d'une base de données

1. Après connexion à Linux, ouvrir un terminal : vérifier que vous êtes bien connecté comme utilisateur **oracle** (vérification au niveau Linux par la commande *whoami*)
2. Lancer le processus d'écoute Oracle (*listener*) : **lsnrctl start**
3. Lancer l'assistant de gestion de bases de données (Database Configuration Assistant) : au niveau Linux, **xhost +** puis **dbca**.
4. Créer votre propre bases de données (toujours sous dbca). Se laisser guider par dbca : créer une base de données *à usage général*
   (a) en lui attribuant un nom,
   (b) en choisissant oracle comme mot de passe, notamment pour les utilisateurs *sys* et *system*,
   (c) et en 'gardant au chaud' l'URL fournie par *dbca*.

6 Session : Espaces d'une base de données

L'objectif de cette session est de localiser les fichiers importants d'une base de données et de se familiariser avec la manipulation des différents espaces d'une base de données. Elle suppose des droits administrateur système sur la machine de résidence du serveur de données.

Après avoir lancer sqlplus, réaliser les opérations ci-dessous.

1. Se connecter comme administrateur (mot de passe de *sys* : *oracle*) ;
   Note : Si l'instance Oracle n'est pas active, l'activer en procédant comme suit :
   (a) Se connecter à une instance "inactive" : sqlplus /nolog (niveau Linux)
   (b) Se connecter comme administrateur : connect sys as sysdba (sous sqlplus)
   (c) Entrer le mot de passe : oracle
   (d) Démarrer l'instance : startup
2. Quel est le nom de l'instance Oracle, sa version, sa machine de résidence et l'état de la base (voir *v$instance*) ?
3. Quel est le nom de la base et son mode d'ouverture (voir *v$database*) ?
4. Localiser son fichier de paramètres d'initialisation (*init.ora.xxxxx*).
   Garder trace de son emplacement pour des besoins ultérieurs.
5. Ouvrir ce fichier et en examiner le contenu ; Voir aussi *v$parameters* ou encore exécuter *show parameter* ;
6. Quels sont les noms et les emplacements des fichiers de contrôle de la base (*control file*) ?

7. Quels sont les noms et les emplacements des fichiers du journal (*redo log*) des images après modification (voir *v$logfile*) ?

8. Quels sont les noms et les numéros des espaces de stockage (*tablespace*) ?

9. Quels sont les noms des fichiers de données (*datafile*) et la taille de leurs blocs (v$datafile) ?

10. Dans quel espace de stockage (*tablespace*) est localisé chaque fichier de données (*datafile*) ?

11. Créer un espace de stockage ayant un seul fichier de 2M et comme clause de stockage par défaut (*default storage*) pour tous ses objets, les valeurs suivantes :

    - premier *extent* : 128K,
    - *extents* suivants : 64K,
    - nombre initial d'extent à allouer : 1,
    - nombre maximum d'extents : 5.

    Vérifier le résultat.

12. Créer un espace de stockage temporaire (*create temporary tablespace ... tempfiles* ou *create tablespace ... temporary datafile*) ayant un seul fichier de 2M ;

13. Vérifier le résultat.

14. Se déconnecter (*exit* ou *disconnect*) et se reconnecter. Quels sont les espaces de stockage existants ? Expliquer.

15. Supprimer les objets créés dans les questions précédentes. A cet effet, voir notamment *drop tablespace ... including contents and datafiles*. Vérifier le résultat.

7 Session : utilisateurs, rôles et privilèges

L'objectif de cette session est de créer des utilisateurs, de leur affecter des droits et des ressources. **Il est fortement conseillé d'ouvrir une fenêtre pour les tâches d'administrateur et une autre pour l'utilisateur "lambda". Ne pas omettre de positionner la variable ORACLE_SID dans les deux fenêtres.**

1. Si l'instance Oracle n'est pas active, la lancer en ouvrant la base ;

2. Créer un utilisateur ; soit *U1* le nom de l'utilisateur créé ;

3. Quels sont les utilisateurs existants dans la base ?

4. Se connecter sous le nom de l'utilisateur *U1*. En cas d'échec de la connexion, expliquer, corriger et se connecter de nouveau ;

5. En tant qu'administrateur, donner le droit de créer des tables à *U1* ;
6. Se connecter en tant qu'utilisateur *U1* et tenter de créer une table ;
7. En tant qu'administrateur, affecter à *U1* l'espace *users* comme espace par défaut en limitant à 10K la taille de l'espace utilisable ; vérifier le résultat ;
8. Se connecter comme utilisateur *U1*, créer une table *T1* et donner les droits d'interrogation sur cette table à tout utilisateur. Si la création échoue, expliquer pourquoi et corriger ;
9. En tant qu'administrateur, créer une table *T2* et donner à *U1* le droit de mettre à jour *T2* ;
10. Quelles sont les tables du dictionnaire qui ont trait aux privilèges ?
11. Retrouver les droits détenus par *U1* et par qui ils lui ont été octroyés ?
12. Retrouver les droits octroyés par *U1* et à qui il les a octroyés ?
13. Quels sont les droits de *U1* pour la session en cours (*v$session_privs*) ?
14. Notion de rôle : exécuter le scénario ci-dessous en examinant les privilèges actifs après chaque instruction *set role* :
    (a) *U1* se déconnecte, se re-connecte : quels sont ses privilèges ?
    (b) Créer un rôle *LesIndex* ayant *create any index, alter any index, drop any index* comme privilèges ;
    (c) Allouer ce rôle à *U1* ;
    (d) Créer un rôle *Role1* ayant *grant any privilege* comme privilège ;
    (e) Allouer ce rôle à l'utilisateur *U1* ;
    (f) En tant qu'utilisateur *U1*,
        i. prendre le rôle *LesIndex* : quels sont les privilèges de l'utilisateur *U1* ?
        ii. prendre le rôle *Role1* : quels sont les privilèges de *U1* ?
        iii. prendre tous les rôles sauf *Role1* : quels sont les privilèges de *U1* ?
15. À titre d'exercice supplémentaire (facultatif) :
    (a) créer un rôle *AvecMdPass* doté d'un mot de passe, ayant le privilège *"alter any table"*, et l'attribuer à l'utilisateur *U1* ;
    (b) En tant qu'utilisateur *U1*, endosser ce rôle : consulter les privilèges de *U1* ;
    (c) *U1* se déconnecte, se re-connecte : quels sont ses privilèges ? Expliquer.
    (d) Rendre le rôle *AvecMdPass*, rôle par défaut de l'utilisateur *U1* ;
    (e) *U1* se déconnecte, se re-connecte : quels sont ses privilèges ? Expliquer.
16. Faire le ménage : supprimer tous les objets créés (utilisateurs, tables, rôles, etc.). Voir, entre autres, *drop user ... cascade*.

8 Session : utilisateurs, profils et ressources

Le but de cette session est de s'initier à la limitation des ressources en affectant des espaces à des utilisateurs et en utilisant des profils définis.

1. Quelles sont les tables du dictionnaire qui ont trait aux ressources et celles qui ont trait aux quotas ?
2. Créer un utilisateur *U1* ayant *users* comme espace par défaut et 1M de taille maximum autorisée sur cet espace ;
3. Vérifier les limites des ressources de *U1*.
4. Créer un espace de données (*tablespace*) *USERS02* avec 2 fichiers (*datafiles*) de 2 mégas ;
5. Affecter *USERS02* comme espace par défaut à *U1* en limitant son espace à 1 méga ;
6. Vérifier les limites des ressources de *U1*.
7. Quelles sont les tables du dictionnaire qui ont trait aux profils ?
8. Quels sont les profils existants ?
9. Créer un profil nommé *Petit* autorisant deux connections simultanées par le même utilisateur, chaque session ne devant pas excéder 2 minutes ; s'assurer que le profil est créé ;
10. Affecter le profil *Petit* à *U1* :
    (a) S'assurer que le profil est bien affecté à *U1* ;
    (b) Vérifier que le profil est effectif sur l'utilisateur *U1*, c'est-à-dire que les limites du profil s'appliquent à *U1* ;
11. Supprimer le profil *Petit* ; quelles sont les nouvelles limites de ressources de *U1* ?
12. Faire le ménage (utilisateur, profils, espaces, ...).

9 Session : Sécurité et reprise

Cette session permet de mettre en œuvre les différents mécanismes liés à la sécurité d'une base de données : *(i)* surveillance (*auditing*) du serveur de données, *(ii)* archivage de journaux, *(iii)* sauvegarde *(iv)* reprise après incident et enfin *(v)* duplication (ou multiplexage).

9.1 Surveillance (*auditing*)

1. Vérifier que l'*audit trail* et ses vues existent, sinon les créer ;
2. Quelles sont les options d'*audit* par défaut ?
3. Positionner les options d'interrogation et de modification de toute table par l'utilisateur *U1* ; vérifier que ces options sont bien positionnées ;
4. Vérifier que ces options sont effectives. Si tel n'est pas le cas, corriger et vérifier ;

   Note : *ses_actions* est un vecteur de 16 caractères, chaque position correspondant à un type d'instruction et pouvant avoir les valeurs *'S'* pour opération réussie, *'F'* pour échec et *'B'* pour les deux. Par exemple, dans ce vecteur, les positions 10 et 11 correspondent respectivement aux opérations *SQL* d'interrogation (*select*) et de modification (*update*)[2].

5. Quelle est la différence entre les résultats des deux instructions *select* ci-dessous (avant et après l'instruction *audit*) :

   *select * from all_def_audit_opts*
   audit alter, grant, insert, update, delete on default
   *select * from all_def_audit_opts*

6. Exercice supplémentaire (facultatif) : Exécuter et vérifier le résultat de l'exemple donné en cours et rappelé ci-dessous :

   1. *audit select on* U1.MaTable *whenever not successful*;
   2. *audit select table, update table by* U1, U2;
   3. *audit role whenever successful*;
   4. *audit all privileges by* U1;
   5. *audit insert, update, delete on sys.aud$ by access*;
   6. *noaudit select table by U1, U2*;

[2] Voir le manuel de référence pour la description détaillée des positions de ce vecteur.

9.2 Archivage des journaux

1. Quel est le mode d'archivage des journaux en vigueur dans la base ?
2. Si le mode d'archivage automatique n'est pas actif, l'activer au sein de la base active ;
3. Indiquer un répertoire comme destination de l'archivage ;
4. Vérifier le résultat des deux actions qui précèdent ;
5. Se déconnecter et se connecter de nouveau ;
6. Quel est l'état de la base ?
7. Quel est son mode d'archivage ?
8. Rendre permanent le mode d'archivage automatique ;
9. Forcer l'archivage des journaux ;
10. Quel est le *redo log* en cours d'utilisation (actif) ?
11. Archiver le seul journal actif ; vérifier le résultat ;
12. Créer une ou plusieurs tables, y faire un grand nombre d'insertions puis déterminer le nombre de fichiers *redo log* archivés ;
13. Quels sont les journaux dont aurait besoin le système pour réaliser une reprise ?
14. Quelles sont les numéros des modifications contenues dans chaque journal archivé ?

9.3 Sauvegardes

1. Créer un répertoire de sauvegarde ;
2. Y sauvegarder manuellement (i.e. par copie au niveau système d'exploitation) un espace de données (*tablespace*) ;
3. Sauvegarder le fichier de contrôle ;
4. Exécuter le scénario ci-dessous (donné en cours) en utilisant le gérant de reprise *rman* :

   a. Lancer *rman*, en tant qu'opérateur (*sysoper*) ou en tant qu'administrateur :
   b. Monter la base (sans l'ouvrir) ;
   c. S'informer sur les *tablespaces* (nom, taille, emplacement, etc.).
   d. Créér une copie du fichier de données numéro 1 ;
   e. Sauvegarder l'espace *system* ;
   f. Lister les sauvegardes et leur contenu ;
   g. Vérifier la consistance d'une sauvegarde identifiée par sa clé (1) ;

9.4 Reprise après incident

Ce paragraphe comporte trois cas : *(i)* reprise automatique après incident, *(ii)* reprise d'un fichier de données (*datafile*) sans disposer de sa sauvegarde et *(iii)* reprise à partir de sauvegardes. Ces différents cas supposent la base en mode automatique d'archivage des journaux.

9.4.1 Reprise automatique

Cette session simule un incident logiciel afin de tester la reprise automatique.

1. Quel est le mode de gestion du journal des images avant modification (*rollback segments* ou *undo tablespaces*) ?

2. En tant qu'utilisateur *U1*, créer une table, y faire quelques insertions puis vérifier et mémoriser le contenu de la table ;

3. En tant qu'utilisateur *U1*,

   (a) Créer la procédure ci-dessous :

   ```
   create or replace procedure Vilaine as
   i int;
   begin
       i:= 1;
       while true
       loop insert into T values(...); i:= i + 1; end loop;
   end;
   ```

   (b) Exécuter la procédure (*exec Vilaine*).

4. La procédure ne terminant pas, en tant qu'administrateur, arrêter l'instance Oracle (**pas en mode abort**) ;

5. Relancer l'instance et vérifier l'état de la table après la relance. Expliquer. On pourra également consulter les fichiers de trace et d'alerte (localisés, par défaut, dans . . . \admin\bdump sous le répertoire de résidence de l'instance Oracle).

9.4.2 Reprise d'un fichier de données sans sa sauvegarde

Cette suite d'opérations simule la perte ou la détérioration d'un fichier de données (*datafile*).

1. Créer un espace de stockage *tablespace*) *USERS03* comportant un fichier de données *users0301.dbf* ;

2. Créer un utilisateur (ou modifier un utilisateur existant) en lui affectant *USERS03* comme espace de stockage par défaut ; soit *U1* cet utilisateur ;

3. Se connecter en tant que *U1*, créer une ou plusieurs tables et y faire des insertions ;

4. Mettre la tablespace hors ligne et supprimer le fichier *users0301.dbf* au niveau du système d'exploitation ;

5. Créer de nouveau le fichier *users0301.dbf* (cf. *alter database create datafile*).

6. Mettre la tablespace en ligne et donner le contenu de la ou des tables créées ;

7. Mettre la tablespace hors ligne et effectuer la reprise du fichier *users0301.dbf* ;

8. Mettre la tablespace en ligne et donner le contenu de la ou des tables créées.

9.4.3 Restauration et reprise : Autres exercices (facultatifs)

La suite de manipulations ci-dessous fait référence au scénario donné en cours et rappelé ici :

La ligne notée *(1)* copie dans un nouvel espace les journaux archivés sauvegardés et les lignes *(2)* et *(3)* montrent deux façons de réaliser une reprise en utilisant cette copie de journaux.

(1) *cp* /disk1/LogBackup/*.arc /disk2/tmp/Archives/.
(2) *set logsource* /disk2/tmp/Archives; *recover;*
(3) *alter database recover from* /disk2/tmp/Archives *database;*

Suite de manipulations à exécuter :

1. Sauvegarder les fichiers *redo logs* archivés ;

2. Faire une suite d'insertions dans une table et forcer l'archivage du journal ;

3. Supprimer le fichier de données *users0301.dbf* et le créer de nouveau ;

4. Restaurer les fichiers *redo logs* sauvegardés ;

5. Effectuer le reprise sur l'espace de stockage *USERS03* en réinitialisant les journaux. Vérifier le résultat ;

9.5 Multiplexage

1. Quelles sont les localisations et les tailles des fichiers de journalisation des images après modification (*redo logs*) ?

2. Quel est le *redo log* en cours d'utilisation (voir *v$logfile.status*) et son numéro de groupe ?
   Note : Cas *v$logfile.status* =

   - INVALID : fichier inaccessible
   - stale : contenu incomplet
   - deleted : le fichier n'est plus utilisé
   - NULL : en cours d'utilisation

3. Créer un nouveau groupe de fichiers de journalisation ne comportant qu'un seul fichier de 64K ; vérifier le résultat ;
4. Ajouter un membre à chaque groupe existant et vérifier le résultat ;
5. Créer une table et y faire un nombre conséquent d'insertions (afin d'instancier un nombre significatif d'enregistrements dans le journal) ;
6. Quel est le *redo log* en cours d'utilisation (actif) et son numéro de groupe ?
7. Supprimer les membres ajoutés ; en cas de non fonctionnement, expliquer ;
8. Supprimer le groupe ajouté ; en cas de non fonctionnement, expliquer.

9.6 Exercices supplémentaires (facultatif) : Utilisation de *rman*

,

1. On pourra mettre en œuvre ce scénario sous le gérant de reprise *rman* comme présenté en cours :

   Restauration et reprise de la *tablespace TabSp1* dans une base ouverte : on notera que *rman* permet d'exécuter des commandes SQL en les introduisant par le mot-clé *sql*.

   run { *allocate channel* C1 *type disk*;
   sql "alter tablespace TabSp1 *offline immediate";*
   restore tablespace TabSp1;
   recover tablespace TabSp1;
   sql "alter tablespace TabSp1 *online";* }

2. De même, on pourra adapter ce scénario pour réaliser une reprise avec choix automatique des journaux à appliquer :

   (a) Scénario d'application automatique de *redo logs* avec *recover* : on suppose que le répertoire */tsbackup* contient une sauvegarde de toute la base, que le répertoire */Oracle/Tbs2Mabase* contient les *tablespaces* de la base et que le nom de chaque *tablespace* commence par *tbs*. Seule la restauration (première instruction) est effectuée au niveau du système d'exploitation, les autres étapes étant exécutées sous Oracle (sous SQL*Plus, par exemple) :

      1) *cp* /tsbackup/tbs* /Oracle/Tbs2Mabase/. //- Restauration.
      2) *startup mount;* //- Montage (sans ouverture) de la base.
      3) *set autorecovery on;* //- Positionner le choix automatique des journaux.
      4) *recover database, tablespace* ou *datafile;* //- Faire la reprise.

   (b) Application automatique de *redo logs* avec *alter database recover* appliqué à l'espace de stockage nommé *users*. On suppose que cet espace a fait l'objet d'une restauration.

alter tablespace users *offline*;
alter database recover automatic tablespace users;
alter tablespace users *online*;

3. Idem pour une reprise avec choix manuel des journaux :
Application manuelle de *redo logs* avec *alter database recover* appliqué à l'espace de stockage *users*. Les étapes 3 et 4 peuvent être itérées, la remise en ligne de l'espace mettant fin à l'itération.

   1) *alter tablespace* users *offline*;
   2) *alter database recover tablespace* users;
   3) //– Oracle suggère un nom de fichier *log* (f_i.log, par exemple).
   4) *alter database recover logfile* "f_i.log";
   5) *alter tablespace* users *online*;

10 Session : Optimisation de requêtes

10.1 Rappels

1. Syntaxe de explain plan (ce qui est entre [ ] étant optionnel):

```
EXPLAIN PLAN  [SET STATEMENT_ID = 'Nom de Requête']
[INTO Nom2Table]  FOR instruction SQL ;

 Instruction  SQL:
         SELECT, INSERT, UPDATE, DELETE,  CREATE TABLE,
         CREATE INDEX, ALTER INDEX ... REBUILD
```

2. Par défaut, un plan est stocké dans la table *plan_table*. On peut néanmoins créer une table de même schéma que *plan_table* et y mettre le plan engendré. Dans ce dernier cas, on utilisera la clause *INTO Nom2Table* de EXPLAIN PLAN.

3. Les plans étant ajoutés à la *plan_table*, il faudra veiller :

   - soit à vider la plan_table à chaque génération de plan,
   - soit à nommer les requêtes (dans explain plan : explain plan set statement_id = 'Nom de la requête') et afficher le plan en ajoutant une clause where (comme dans l'exemple ci- dessous).

4. Exemple de génération de plan sans nommage de requête :

```
SQL> explain plan for
  2  select * from produit  where prod# = 700;

Explicité.
```

5. Exemple d'affichage des résultats sans nommage de requête :

```
SQL> column ID format 99;
SQL> column parent format 99;
SQL> column Objet format A10;
SQL> column operation format A20 ;
SQL> column options format A10;
SQL>
  1  select '|', id "ID", '|', parent_id "parent",  '|', operation,
  2         '|',  object_name "Objet", '|', options, '|'
  3* from plan_table;

'  ID ' parent ' OPERATION            ' Objet      ' OPTIONS   '
- --- - ------ - -------------------- - ---------- - -----------
|   0 |        | SELECT STATEMENT     |            |           |
|   1 |      0 | TABLE ACCESS         | PRODUIT    | FULL      |
```

6. Exemple de génération de plan avec nommage de requête :

```
SQL> explain plan set statement_id = 'Rqte1' for
  2  select * from produit  where prod# = 700;

Explicité.
```

7. Exemple d'affichage des résultats avec nommage de requête :

```
SQL>
  1  select '|', id "ID", '|', parent_id "parent",  '|', operation,
  2          object_name "Objet", '|', options, '|'
  3* from plan_table where statement_id = 'Rqte1';
```

On pourra évidemment afficher d'autres attributs, comme des attributs ayant trait aux coûts de la requête. Voir les attributs de *plan_table*.

10.2 Session

1. Vérifier l'existence de la table *plan_table* et la créer si elle n'existe pas ;
2. Créer les relations suivantes sans clés (**utiliser les scripts fournis**) :
   create table produit (prod# int not null, libelle varchar(20), pu float not null)
   create table depot (dep# int not null, adr varchar(20), capacite int)
   create table stock (prod# int not null, dep# int not null, qte int not null)

3. Procédures d'instanciation (**utiliser les scripts fournis**) :

   (a) Une procédure qui permet d'instancier la relation produit. Cette procédure aura comme argument NbProd, le nombre de produits à insérer. Les tuples de la relation sont de la forme <i, 'Produit_i', i>, $i \in [1..NbProd]$ (Exemple : <1, Produit_1, 1>, <2, Produit_2, 2>, etc.).

   (b) Une procédure qui permet d'instancier la relation depot. De la même façon, cette procédure aura un argument NbDep désignant le nombre de dépôts à insérer. Les tuples de la relation sont de la forme <j, 'Depot Numero j', x >, $j \in [1..NbDep]$.

   (c) Une procédure qui permet d'instancier la relation stock Cette procédure aura comme argument NbrStock désignant le nombre moyen de dépôts dans lesquels est stocké un même produit. Les tuples de la relation sont de la forme <prod#, dep#, 3000>.

4. Instancier les relations avec un nombre conséquent de produits (NbProd $>$ 1500), de dépôts (NbDep $>=$ 20) en considérant qu'un produit est stocké en moyenne dans 10 dépôts.

5. Exécuter les requêtes ci-dessous et examiner leur plan d'exécution. Inclure des éléments de coût de la requête dans l'examen du plan d'exécution. Comparer les plans des couples de requêtes (3, 4), (5, 6), (7, 8) et (9, 10) :

```
Rqte3: select * from produit where prod# = 700 or prod# = 800

Rqte4: select * from produit where prod# in (700, 800)

Rqte5: select p.libelle from produit p, stock s
       where s.dep# = 4 and p.prod# = s.prod#

Rqte6: select p.libelle from produit p
       where p.prod# in (select s.prod# from stock s
                                 where s.dep# = 4)

Rqte7: select d.adr from depot d, stock s
       where d.dep# = 4 and d.dep# = s.dep#
       and s.prod# > 100 and s.prod# < 500

Rqte8: select d.adr from depot d where d.dep# = 4
        and d.dep# in (select s.dep# from stock s
                        where s.prod# > 100 and s.prod# < 500)

Rqte9: select prod#, sum(qte) "Total en stock" from stock
       where  dep# > 10 and dep# < 100  group by prod#

Rqte10: select prod#, sum(qte) "Total en stock" from stock
```

```
where  dep# > 10 and dep# < 100  group by prod#
order by prod#
```

6. **Variante 1** :
   (a) Changer l'ordre des relations dans la clause FROM des requêtes 5 et 7.
   (b) Exécuter ces requêtes et comparer leurs plans d'exécution avec ceux obtenus dans la question 4.
7. **Variante 2** :
   (a) Quel est l'espace occupé par chacune des relations (dans entre autres, *user_tables*, nombre de blocs, nombre de blocs libres, nombre de tuples, etc.) ?
   (b) Collecter des statistiques sur les tables.
   (c) Quel est l'espace occupé par chacune des relations ? Expliquer.
   (d) Exécuter quelques requêtes de la liste ci-avant et comparer les plans d'exécution avec ceux obtenus dans les questions 4 et 5.
8. **Variante 3** :
   (a) Créer un index sur la relation produit ayant prod# comme clé.
   (b) Créer un index sur la relation stock ayant (prod#, dep#) comme clé.
   (c) Exécuter quelques requêtes de la liste ci-avant et comparer les plans d'exécution avec ceux obtenus dans les questions 4, 5 et 6.
9. **Variante 4** :
   (a) Collecter les statistiques sur les index.
   (b) Examiner l'encombrement des index créés (Nombre de blocs, nombre de valeurs par blocs, nombre de niveaux, etc.).
   (c) Exécuter quelques requêtes de la liste ci-avant et comparer les plans d'exécution avec ceux obtenus dans les questions 4, 5, 6 et 7.
10. Adapter les exemples donnés en cours pour forcer le choix de l'optimiseur.
11. **Ménage** : Supprimer tous les objets (tables, index) créés.