

Liaison Cours/TP Oracle Database 19c

Commandes Pratiques avec Explications Détaillées

Cours et Travaux Pratiques Oracle Database

27 janvier 2026

Table des matières

1	Introduction	2
2	TP1 : Formatage des Données et SQL*Plus	2
2.1	Lien avec le cours	2
2.2	Connexion et Gestion des Utilisateurs	2
2.2.1	Connexion à SQL*Plus	2
2.2.2	Consultation des Utilisateurs	2
2.2.3	Activation d'un Utilisateur	3
2.3	Paramètres d’Affichage SQL*Plus	4
2.3.1	Configuration de l’Affichage	4
2.4	Formatage des Colonnes	4
2.4.1	Formatage des En-têtes	4
2.4.2	Formatage des Données	5
2.4.3	Gestion du Retour à la Ligne (WRAP)	6
2.5	Commandes BREAK et COMPUTE	6
2.5.1	Commande BREAK	6
2.5.2	Commande COMPUTE	7
2.5.3	Titres de Rapport	8
2.6	Commande SPOOL	8
3	TP2 : Architecture et Instance	8
3.1	Lien avec le cours	8
3.2	Gestion des Sessions	9
3.2.1	Consultation des Sessions	9
3.2.2	Arrêt d’une Session	9
3.3	Gestion de l’Instance	10
3.3.1	Démarrage de l’Instance	10
3.3.2	Arrêt de l’Instance	11
3.4	Consultation de la Mémoire SGA	11

4	TP3 : Architecture Multitenant (CDB/PDB)	12
4.1	Lien avec le cours	12
4.2	Connexion aux CDB et PDB	12
4.2.1	Connexion à la CDB	12
4.2.2	Consultation des PDBs	12
4.2.3	Changement de Container	13
4.2.4	Connexion Directe à une PDB	13
4.3	Gestion des PDBs	14
4.3.1	Ouverture/Fermeture des PDBs	14
4.3.2	Démarrage de l'Instance CDB	15
5	TP4 : Paramètres d'Initialisation	15
5.1	Lien avec le cours	15
5.2	Processus en Arrière-Plan	15
5.3	Consultation des Paramètres	16
5.3.1	Affichage des Paramètres	16
5.3.2	Types de Paramètres	17
5.3.3	Paramètres Mémoire	17
5.4	Gestion des Fichiers de Paramètres	18
5.4.1	PFILE et SPFILE	18
5.5	Modification des Paramètres	18
5.5.1	Modification au Niveau Mémoire	18
5.5.2	Modification au Niveau SPFILE	19
5.5.3	Modification Immédiate	19
5.5.4	Correction d'Erreur de Paramètre	19
6	TP7 : Fichiers de Contrôle	20
6.1	Lien avec le cours	20
6.2	Consultation des Fichiers de Contrôle	20
6.2.1	Informations via V\$PARAMETER	20
6.2.2	Informations via V\$CONTROLFILE	20
6.2.3	Informations via V\$DATABASE	21
6.3	Importance des Fichiers de Contrôle	21
6.4	Multiplexage des Fichiers de Contrôle	22
6.4.1	Multiplexage via SPFILE	22
6.4.2	Multiplexage via PFILE	23
6.5	Sauvegarde des Fichiers de Contrôle	23
6.5.1	Sauvegarde Binaire	23
6.5.2	Sauvegarde en Fichier Trace	24
7	Conclusion	24
7.1	Points Clés à Retenir	24
7.2	Bonnes Pratiques	24

1 Introduction

Ce document fait le lien entre le cours théorique d'Oracle Database et les travaux pratiques. Pour chaque chapitre du cours, nous présentons les commandes pratiques correspondantes avec des explications détaillées.

2 TP1 : Formatage des Données et SQL*Plus

2.1 Lien avec le cours

Ce TP correspond à l'introduction à Oracle et à la manipulation de base de données via SQL*Plus. Il couvre :

- Connexion à Oracle Database
- Gestion des utilisateurs
- Formatage des données et rapports
- Commandes SQL*Plus essentielles

2.2 Connexion et Gestion des Utilisateurs

2.2.1 Connexion à SQL*Plus

Connexion sans utilisateur

```
1 sqlplus /nolog
```

Explication

Cette commande lance SQL*Plus sans se connecter à une base de données. Utile pour se connecter ensuite avec différents utilisateurs.

Connexion en tant que SYSDBA

```
1 connect / as sysdba
```

Explication

Se connecte en tant qu'administrateur système (SYSDBA). Permet d'effectuer toutes les opérations d'administration. Le / indique une authentification par le système d'exploitation (OS authentication).

2.2.2 Consultation des Utilisateurs

Afficher tous les utilisateurs

```
1 SELECT * FROM all_users;
```

Explication

Affiche tous les utilisateurs visibles par l'utilisateur courant. La vue ALL_USERS contient les utilisateurs accessibles.

Afficher les informations détaillées des utilisateurs

```
1 SELECT * FROM dba_users;
```

Explication

Affiche toutes les informations sur les utilisateurs de la base. Nécessite les privilèges DBA. Contient notamment : username, account_status, created, expiry_date, profile, etc.

Afficher le statut d'un utilisateur

```
1 SELECT account_status FROM dba_users
2 WHERE username='HR';
```

Explication

Le statut peut être : OPEN (actif), LOCKED (bloqué), EXPIRED (mot de passe expiré), EXPIRED & LOCKED.

2.2.3 Activation d'un Utilisateur**Débloquer un utilisateur**

```
1 ALTER USER HR ACCOUNT UNLOCK;
```

Explication

Débloque le compte utilisateur HR. Un compte bloqué ne peut pas se connecter même avec le bon mot de passe.

Définir le mot de passe d'un utilisateur

```
1 ALTER USER HR IDENTIFIED BY HR;
```

Explication

Définit ou modifie le mot de passe de l'utilisateur HR. Le mot de passe est sensible à la casse.

Attention

Après avoir débloqué et défini le mot de passe, l'utilisateur peut se connecter avec :

```
1 connect HR/HR
```

2.3 Paramètres d’Affichage SQL*Plus

2.3.1 Configuration de l’Affichage

Afficher les paramètres d’affichage

```
1 SHOW LINESIZE PAGESIZE
```

Explication

Affiche la largeur de ligne (LINESIZE) et le nombre de lignes par page (PAGESIZE).
Par défaut : LINESIZE=80, PAGESIZE=14.

Modifier les paramètres d’affichage

```
1 SET LINESIZE 200
2 SET PAGESIZE 1000
```

Explication

- LINESIZE : Largeur maximale d’une ligne (en caractères)
- PAGESIZE : Nombre de lignes avant un saut de page (0 = pas de saut)

Ces paramètres affectent uniquement la session courante.

2.4 Formatage des Colonnes

2.4.1 Formatage des En-têtes

Modifier l’en-tête d’une colonne

```
1 COLUMN FIRST_NAME HEADING 'FIRST NAME'
2 COLUMN LAST_NAME HEADING 'LAST NAME'
3 COLUMN HIRE_DATE HEADING 'HIRE DATE'
```

Explication

Modifie le titre affiché pour une colonne. Le caractère | dans l'en-tête crée un saut de ligne :

```
1 COLUMN REGION_ID HEADING 'IDENTIFIANT | REGION'
```

Modifier le caractère de soulignement

```
1 SET UNDERLINE +
```

Explication

Change le caractère utilisé pour séparer l'en-tête du corps du résultat (par défaut : -).

2.4.2 Formatage des Données**Formater une colonne de type caractère**

```
1 COLUMN HIRE_DATE FORMAT A10
```

Explication

A10 signifie : format Alphanumérique de 10 caractères. Si la valeur dépasse, elle est tronquée.

Formater une colonne numérique

```
1 COLUMN SALARY FORMAT \ $99,990
```

Explication

Format numérique avec :

- \$: Symbole dollar
- 9 : Chiffre (supprimé si 0)
- 0 : Chiffre (toujours affiché)
- , : Séparateur de milliers

Exemple : 50000 devient \$50,000

2.4.3 Gestion du Retour à la Ligne (WRAP)

Effacer les formats de colonnes

```
1 CLEAR COLUMNS
```

Explication

Supprime tous les formats de colonnes définis précédemment.

Configurer le retour à la ligne

```
1 COLUMN JOB_TITLE FORMAT A22
2 SET RECSEP WRAPPED
3 COLUMN JOB_TITLE WORD_WRAPPED
```

Explication

- RECSEP WRAPPED : Affiche un séparateur après chaque ligne qui a été coupée
- WORD_WRAPPED : Coupe aux espaces (mots entiers) plutôt qu'aux caractères
- RECSEP EACH : Affiche un séparateur après chaque ligne
- RECSEP OFF : Désactive les séparateurs

Modifier le caractère de séparation

```
1 SET RECSEPCHAR "*"
2 SET RECSEPCHAR ""
```

Explication

Change le caractère utilisé comme séparateur de ligne. Une chaîne vide supprime le séparateur.

2.5 Commandes BREAK et COMPUTE

2.5.1 Commande BREAK

Grouper les résultats par colonne

```
1 BREAK ON DEPARTMENT_ID
2 SELECT DEPARTMENT_ID, LAST_NAME, SALARY
3 FROM EMP_DETAILS_VIEW
4 WHERE SALARY > 12000
5 ORDER BY DEPARTMENT_ID;
```

Explication

La commande **BREAK** supprime les valeurs répétées dans une colonne. Utile pour les rapports groupés. La colonne doit être triée (**ORDER BY**).

BREAK avec saut de ligne

```
1 BREAK ON DEPARTMENT_ID SKIP 1 ON JOB_ID SKIP PAGE
```

Explication

- **SKIP 1** : Saut d'une ligne lors d'un changement de valeur
- **SKIP PAGE** : Saut de page lors d'un changement de valeur

Permet de créer des rapports structurés avec des sections.

Supprimer les BREAK

```
1 CLEAR BREAKS
```

2.5.2 Commande COMPUTE

Calculer des statistiques par groupe

```
1 BREAK ON JOB_ID SKIP 1
2 COMPUTE AVG LABEL 'Salaire moyen' OF SALARY ON JOB_ID
3 SELECT JOB_ID, LAST_NAME, SALARY
4 FROM EMP_DETAILS_VIEW
5 WHERE SALARY > 12000
6 ORDER BY JOB_ID;
```

Explication

COMPUTE calcule des statistiques (**AVG**, **SUM**, **COUNT**, **MIN**, **MAX**, **STD**, **VAR**) sur une colonne lors d'un changement de valeur dans la colonne du **BREAK**.

Calculer des statistiques globales

```
1 BREAK ON REPORT
2 COMPUTE AVG LABEL 'Salaire moyen' OF SALARY ON REPORT
```

Explication

ON REPORT calcule la statistique sur l'ensemble du résultat, à la fin du rapport.

2.5.3 Titres de Rapport

Ajouter un titre

```
1 TTITLE 'Liste des salariés par fonction et par département'
```

Explication

Ajoute un titre en haut de chaque page. Pour le supprimer : TTITLE OFF

2.6 Commande SPOOL

Sauvegarder les résultats dans un fichier

```
1 SPOOL fichier.txt
2 -- ou
3 SPOOL fichier.txt CREATE
4 SPOOL fichier.txt REPLACE
5 SPOOL fichier.txt APPEND
```

Explication

- CREATE : Crée un nouveau fichier (erreur si existe)
- REPLACE : Remplace le fichier s'il existe
- APPEND : Ajoute à la fin du fichier existant

Tout ce qui s'affiche dans SQL*Plus est enregistré dans le fichier.

Arrêter le SPOOL

```
1 SPOOL OFF
```

3 TP2 : Architecture et Instance

3.1 Lien avec le cours

Ce TP correspond au chapitre sur l'architecture Oracle et les instances. Il couvre :

- Gestion des sessions et connexions
- Démarrage/arrêt d'une instance
- Consultation de la mémoire SGA

3.2 Gestion des Sessions

3.2.1 Consultation des Sessions

Afficher toutes les sessions actives

```
1 SELECT sid, serial#, username, status, osuser, machine,
   program, paddr
2 FROM v$session;
```

Explication

La vue V\$SESSION contient toutes les sessions actives :

- SID : Identifiant de session
- SERIAL# : Numéro de série de la session
- USERNAME : Nom d'utilisateur
- STATUS : ACTIVE, INACTIVE, KILLED
- OSUSER : Utilisateur du système d'exploitation
- MACHINE : Machine cliente
- PROGRAM : Programme ayant initié la session
- PADDR : Adresse du processus serveur

Identifier les sessions inactives

```
1 SELECT sid, serial#, username, status, seconds_in_wait
2 FROM v$session
3 WHERE status = 'INACTIVE' AND seconds_in_wait > 180;
```

Explication

Trouve les sessions inactives depuis plus de 3 minutes (180 secondes). Utile pour identifier les sessions à nettoyer.

3.2.2 Arrêt d'une Session

Tuer une session

```
1 ALTER SYSTEM KILL SESSION '81,27';
```

Explication

Termine une session spécifique. Format : 'SID,SERIAL#'. La session est marquée comme KILLED et sera nettoyée par PMON.

Attention

Si la session a des transactions non validées, elles seront annulées (ROLLBACK). Vérifier avant de tuer une session importante.

3.3 Gestion de l'Instance

3.3.1 Démarrage de l'Instance

Démarrer l'instance

```
1 STARTUP
```

Explication

Démarre l'instance en trois étapes :

1. **NOMOUNT** : Démarre l'instance, lit le SPFILE/PFILE, alloue la SGA
2. **MOUNT** : Monte la base, ouvre les control files
3. **OPEN** : Ouvre les datafiles et redo logs, rend la base accessible

Démarrer en mode NOMOUNT

```
1 STARTUP NOMOUNT
```

Explication

Démarre uniquement l'instance sans monter la base. Utile pour certaines opérations d'administration (création de base, récupération).

Monter la base

```
1 ALTER DATABASE MOUNT;
```

Explication

Monte la base de données. Ouvre les control files mais pas les datafiles. La base n'est pas accessible aux utilisateurs.

Ouvrir la base

```
1 ALTER DATABASE OPEN;
```

Explication

Ouvre la base de données. Les datafiles et redo logs sont ouverts. La base est accessible aux utilisateurs.

3.3.2 Arrêt de l'Instance

Arrêter l'instance

```
1 SHUTDOWN
2 -- ou
3 SHUTDOWN IMMEDIATE
4 SHUTDOWN TRANSACTIONAL
5 SHUTDOWN ABORT
```

Explication

- SHUTDOWN (NORMAL) : Attend que tous les utilisateurs se déconnectent
- IMMEDIATE : Interrompt les transactions, fait un ROLLBACK, puis arrête
- TRANSACTIONAL : Attend la fin des transactions en cours
- ABORT : Arrêt immédiat sans vérification (à éviter sauf urgence)

3.4 Consultation de la Mémoire SGA

Afficher les composants de la SGA

```
1 SELECT * FROM v$sgainfo;
```

Explication

Affiche les informations sur les composants de la SGA :

- Taille de chaque composant
- Taille réservée
- Taille utilisée

Afficher l'utilisation de la SGA

```
1 SELECT * FROM v$sgastat;
```

Explication

Affiche l'utilisation détaillée de chaque composant de la SGA. Utile pour diagnostiquer les problèmes de mémoire.

4 TP3 : Architecture Multitenant (CDB/PDB)

4.1 Lien avec le cours

Ce TP correspond au chapitre sur l'architecture multitenant introduite dans Oracle 12c. Il couvre :

- Connexion aux CDB et PDB
- Gestion des PDBs
- Modes de démarrage

4.2 Connexion aux CDB et PDB

4.2.1 Connexion à la CDB

Connexion locale en SYSDBA

```
1 connect /as sysdba
```

Explication

Se connecte à la Container Database (CDB) en tant que SYSDBA. C'est la connexion principale pour l'administration.

Connexion via TNSNAMES

```
1 connect sys@cdb_name as sysdba
```

Explication

Se connecte à la CDB via le fichier `tnsnames.ora`. Nécessite que le service soit configuré dans TNSNAMES.

4.2.2 Consultation des PDBs

Afficher le container courant

```
1 SHOW CON_NAME
```

Explication

Affiche le nom du container dans lequel on se trouve (CDB\$ROOT ou nom d'une PDB).

Lister les PDBs

```
1 SHOW PDBS
```

Explication

Affiche toutes les PDBs avec leur CON_ID, nom et mode d'ouverture (MOUNTED, READ WRITE, READ ONLY).

Afficher les services actifs d'une PDB

```
1 SELECT name FROM v$active_services
2 WHERE con_id=3;
```

Explication

Affiche le nom du service pour une PDB spécifique. Ce nom est utilisé dans `tnsnames.ora` pour se connecter à la PDB.

4.2.3 Changement de Container**Changer de container**

```
1 ALTER SESSION SET CONTAINER=pdb1;
```

Explication

Change le container courant vers une PDB spécifique. Toutes les commandes suivantes s'exécutent dans ce container.

Revenir au CDB\$ROOT

```
1 ALTER SESSION SET CONTAINER=cdb$root;
```

Explication

Reviens au container root (CDB\$ROOT). Nécessaire pour certaines opérations d'administration au niveau CDB.

4.2.4 Connexion Directe à une PDB**Connexion via TNSNAMES à une PDB**

```
1 connect user/pwd@pdb_name
```

Explication

Se connecte directement à une PDB via TNSNAMES. Nécessite que la PDB soit configurée dans `tnsnames.ora` et que le listener soit actif.

4.3 Gestion des PDBs

4.3.1 Ouverture/Fermeture des PDBs

Ouvrir une PDB

```
1 ALTER PLUGGABLE DATABASE pdb1 OPEN;
```

Explication

Ouvre une PDB en mode READ WRITE. La PDB devient accessible aux utilisateurs.

Ouvrir plusieurs PDBs

```
1 ALTER PLUGGABLE DATABASE pdb1 , pdb2 OPEN;  
2 -- ou toutes  
3 ALTER PLUGGABLE DATABASE ALL OPEN;
```

Explication

Ouvre plusieurs PDBs en une seule commande. ALL ouvre toutes les PDBs.

Fermer une PDB

```
1 ALTER PLUGGABLE DATABASE pdb1 CLOSE;
```

Explication

Ferme une PDB. Elle passe en mode MOUNTED et n'est plus accessible aux utilisateurs.

Vérifier l'état des PDBs

```
1 SELECT name , open_mode FROM v$pdb;
```

Explication

Affiche l'état de toutes les PDBs : MOUNTED, READ ONLY, READ WRITE.

4.3.2 Démarrage de l'Instance CDB

Démarrage par étapes

```
1 STARTUP NOMOUNT;
2 SELECT instance_name, status FROM v$instance;
3 SELECT name, open_mode FROM v$pdb;
4
5 ALTER DATABASE MOUNT;
6 SELECT instance_name, status FROM v$instance;
7 SELECT name, open_mode FROM v$pdb;
8
9 ALTER DATABASE OPEN;
10 SELECT instance_name, status FROM v$instance;
11 SELECT name, open_mode FROM v$pdb;
```

Explication

Le démarrage d'une CDB suit les mêmes étapes qu'une base non-CDB :

1. **NOMOUNT** : Instance démarrée, PDBs en MOUNTED
2. **MOUNT** : CDB montée, PDBs toujours en MOUNTED
3. **OPEN** : CDB ouverte, PDB\$SEED en READ ONLY, autres PDBs en MOUNTED (à ouvrir manuellement)

5 TP4 : Paramètres d'Initialisation

5.1 Lien avec le cours

Ce TP correspond à la gestion des paramètres d'initialisation et des fichiers de paramètres (PFILE/SPFILE). Il couvre :

- Consultation des paramètres
- Modification des paramètres
- Gestion des processus en arrière-plan

5.2 Processus en Arrière-Plan

Lister les processus en arrière-plan

```
1 SELECT paddr, name, description
2 FROM v$bgprocess
3 WHERE paddr <> HEXTORAW('00');
```


Explication

Affiche tous les processus en arrière-plan actifs. `paddr <> HEXTORAW('00')` filtre les processus inactifs. Contient : DBWn, LGWR, CKPT, SMON, PMON, etc.

5.3 Consultation des Paramètres

5.3.1 Affichage des Paramètres

Afficher tous les paramètres

```
1 SHOW PARAMETERS
```

Explication

Affiche tous les paramètres d'initialisation. Équivalent à `SELECT * FROM v$parameter`.

Afficher un paramètre spécifique

```
1 SHOW PARAMETER control_files
2 SHOW PARAMETER memory
```

Explication

Affiche les paramètres contenant le mot-clé spécifié (recherche partielle).

Afficher les paramètres basiques

```
1 SELECT num, name, type, value, ISDEFAULT,
2         ISSES_MODIFIABLE, ISSYS_MODIFIABLE,
3         ISINSTANCE_MODIFIABLE, ISBASIC
4 FROM v$parameter
5 WHERE ISBASIC = 'TRUE';
```

Explication

Affiche uniquement les paramètres basiques (les plus importants). Ces paramètres sont essentiels au fonctionnement de la base.

5.3.2 Types de Paramètres

Afficher les paramètres statiques

```
1 SELECT name, value
2 FROM v$parameter
3 WHERE ISSYS_MODIFIABLE = 'FALSE';
```

Explication

Les paramètres statiques nécessitent un redémarrage de l'instance pour être modifiés. Exemples : `db_block_size`, `processes`.

Afficher les paramètres dynamiques

```
1 SELECT name, value
2 FROM v$parameter
3 WHERE ISSYS_MODIFIABLE = 'IMMEDIATE' OR ISSYS_MODIFIABLE = '
  DEFERRED';
```

Explication

- IMMEDIATE : Modification immédiate, affecte toutes les sessions
- DEFERRED : Modification différée, affecte seulement les nouvelles sessions

Ces paramètres ne nécessitent pas de redémarrage.

5.3.3 Paramètres Mémoire

Afficher les paramètres mémoire

```
1 SELECT name, value
2 FROM v$parameter
3 WHERE name IN ('sga_target', 'pga_aggregate_target',
4               'memory_target', 'memory_max_target');
```

Explication

Affiche les paramètres de gestion automatique de la mémoire :

- `memory_target` : Taille cible de la mémoire totale (SGA + PGA)
- `memory_max_target` : Taille maximale de la mémoire
- `sga_target` : Taille cible de la SGA
- `pga_aggregate_target` : Taille cible de la PGA

5.4 Gestion des Fichiers de Paramètres

5.4.1 PFILE et SPFILE

Afficher l'emplacement du SPFILE

```
1 SHOW PARAMETER spfile
```

Explication

Affiche le chemin du SPFILE (Server Parameter File). Le SPFILE est un fichier binaire modifiable uniquement via SQL.

Créer un PFILE à partir du SPFILE

```
1 CREATE PFILE FROM SPFILE;  
2 -- ou avec chemin spécifique  
3 CREATE PFILE='C:\oracle\dfs\initiorcl.ora'  
4 FROM SPFILE;
```

Explication

Crée un PFILE (fichier texte) à partir du SPFILE. Le PFILE peut être édité manuellement avec un éditeur de texte.

Créer un SPFILE à partir du PFILE

```
1 CREATE SPFILE FROM PFILE;  
2 -- ou avec chemins spécifiques  
3 CREATE SPFILE='C:\oracle\database\spfileorcl.ora'  
4 FROM PFILE='C:\oracle\dfs\initiorcl.ora';
```

Explication

Crée un SPFILE à partir d'un PFILE. Utile après modification manuelle du PFILE.

5.5 Modification des Paramètres

5.5.1 Modification au Niveau Mémoire

Modifier un paramètre au niveau mémoire

```
1 ALTER SYSTEM SET memory_target=600M SCOPE=MEMORY;
```

Explication

SCOPE=MEMORY : Modification uniquement en mémoire. La modification est perdue au redémarrage. Utile pour tester.

5.5.2 Modification au Niveau SPFILE

Modifier un paramètre au niveau SPFILE

```
1 ALTER SYSTEM SET memory_target=600M SCOPE=SPFILE;
```

Explication

SCOPE=SPFILE : Modification dans le SPFILE uniquement. Nécessite un redémarrage pour être effective. Persiste après redémarrage.

5.5.3 Modification Immédiate

Modifier un paramètre dynamique

```
1 ALTER SYSTEM SET memory_target=600M SCOPE=BOTH;
```

Explication

SCOPE=BOTH : Modification immédiate (mémoire) + persistance (SPFILE). Pour les paramètres dynamiques.

Attention

Si un paramètre nécessite un redémarrage (statique), utiliser SCOPE=SPFILE et redémarrer l'instance :

```
1 ALTER SYSTEM SET processes=300 SCOPE=SPFILE;  
2 SHUTDOWN IMMEDIATE;  
3 STARTUP;
```

5.5.4 Correction d'Erreur de Paramètre

Démarrer avec un PFILE en cas d'erreur

```
1 STARTUP PFILE='C:\oracle\dfs\initiorcl.ora';
```

Explication

Si le SPFILE contient une erreur empêchant le démarrage, on peut démarrer avec un PFILE corrigé, puis recréer le SPFILE.

6 TP7 : Fichiers de Contrôle

6.1 Lien avec le cours

Ce TP correspond au chapitre sur la gestion des fichiers de contrôle (Control Files). Il couvre :

- Consultation des fichiers de contrôle
- Multiplexage des fichiers de contrôle
- Sauvegarde des fichiers de contrôle

6.2 Consultation des Fichiers de Contrôle

6.2.1 Informations via V\$PARAMETER

Afficher les chemins des control files

```
1 COL name FORMAT A20
2 COL value FORMAT A40
3 SELECT name, value
4 FROM v$parameter
5 WHERE name = 'control_files';
```

Explication

Affiche les chemins de tous les fichiers de contrôle configurés. Ces chemins sont définis dans le SPFILE/PFILE.

6.2.2 Informations via V\$CONTROLFILE

Afficher les fichiers de contrôle

```
1 SELECT * FROM v$controlfile;
```

Explication

Affiche tous les fichiers de contrôle de l'instance. Contient : nom, statut, taille, timestamp. Accessible uniquement si la base est en mode MOUNT ou OPEN.

Attention

En mode NOMOUNT, la vue V\$CONTROLFILE est vide car les control files ne sont pas encore ouverts.

6.2.3 Informations via V\$DATABASE

Afficher les informations de la base

```
1 SELECT * FROM v$database;
```

Explication

Affiche les informations générales sur la base de données, lues depuis le control file : nom, ID, mode d'archivage, etc.

Afficher le type de control file

```
1 COL controlfile_type FORMAT A20
2 COL controlfile_created FORMAT A20
3 SELECT name, controlfile_type, controlfile_created
4 FROM v$database;
```

Explication

Affiche le type de control file (CURRENT, BACKUP) et la date de création. Accessible uniquement en mode MOUNT ou OPEN.

6.3 Importance des Fichiers de Contrôle

Test de l'importance

```
1 -- 1. Arrêter la base
2 SHUTDOWN IMMEDIATE;
3
4 -- 2. Renommer le control file (dans CMD)
5 -- ren control01.ctl control01.ctl1
6
7 -- 3. Essayer de démarrer
8 STARTUP;
```

Explication

Si le control file est manquant ou inaccessible, Oracle ne peut pas démarrer la base. C'est pourquoi le multiplexage est essentiel.

6.4 Multiplexage des Fichiers de Contrôle

6.4.1 Multiplexage via SPFILE

Étapes de multiplexage avec SPFILE

```
1  -- 1. Modifier le paramètre dans SPFILE
2  ALTER SYSTEM SET control_files=
3  'C:\oracle\oradata\orcl\DiskBackup\control01.ctl',
4  'C:\oracle\oradata\orcl\DiskBackup2\control02.ctl'
5  SCOPE=SPFILE;
6
7  -- 2. Arrêter la base
8  SHUTDOWN IMMEDIATE;
9
10 -- 3. Dans CMD : Créer les dossiers et copier les fichiers
11 -- mkdir DiskBackup
12 -- copy control01.ctl DiskBackup\
13 -- mkdir DiskBackup2
14 -- copy DiskBackup\control01.ctl DiskBackup2\control02.ctl
15
16 -- 4. Redémarrer la base
17 STARTUP;
18
19 -- 5. Vérifier
20 SHOW PARAMETER control_files;
```

Explication

Le multiplexage via SPFILE :

1. Modifie le paramètre `control_files` dans le SPFILE
2. Arrête la base pour copier physiquement les fichiers
3. Copie le control file existant vers les nouveaux emplacements
4. Redémarre la base qui utilisera maintenant tous les control files

Oracle écrit simultanément dans tous les control files pour garantir la cohérence.

6.4.2 Multiplexage via PFILE

Multiplexage avec PFILE

```
1  -- 1. Créer un PFILE depuis SPFILE
2  CREATE PFILE='C:\oracle\dfs\initiorcl.ora'
3  FROM SPFILE;
4
5  -- 2. Arrêter la base
6  SHUTDOWN IMMEDIATE;
7
8  -- 3. Éditer manuellement initiorcl.ora :
9  -- *.control_files=
10 -- 'C:\oracle\oradata\orcl\DiskBackup\control01.ctl',
11 -- 'C:\oracle\oradata\orcl\DiskBackup2\control02.ctl'
12
13 -- 4. Copier les fichiers (dans CMD)
14
15 -- 5. Démarrer avec le PFILE
16 STARTUP PFILE='C:\oracle\dfs\initiorcl.ora';
```

Explication

Avec PFILE, on modifie directement le fichier texte. Plus simple mais moins flexible que SPFILE. Après modification, on peut recréer le SPFILE.

Recommandation

Oracle recommande d'avoir au moins 2 copies de control files sur des disques physiques différents pour la redondance.

6.5 Sauvegarde des Fichiers de Contrôle

6.5.1 Sauvegarde Binaire

Sauvegarder le control file en binaire

```
1  ALTER DATABASE BACKUP CONTROLFILE TO
2  'C:\oracle\oradata\orcl\save\bkp.ctl';
```

Explication

Crée une copie binaire exacte du control file. Cette copie peut être utilisée pour restaurer le control file en cas de perte. Format identique à l'original.

6.5.2 Sauvegarde en Fichier Trace

Sauvegarder le control file en script SQL

```
1 ALTER DATABASE BACKUP CONTROLFILE TO TRACE;
```

Explication

Crée un fichier trace contenant un script SQL pour recréer le control file. Utile si tous les control files sont perdus.

Trouver le fichier trace

```
1 SELECT name, value
2 FROM v$diag_info
3 WHERE LOWER(name) LIKE '%trace%';
```

Explication

Affiche l'emplacement du répertoire de trace. Le fichier généré contient un script CREATE CONTROLFILE avec tous les chemins des datafiles et redo logs.

Recommandation

Oracle recommande de sauvegarder le control file à chaque modification de la structure de la base (ajout/suppression de datafiles ou redo logs).

7 Conclusion

Ce document a présenté les principales commandes Oracle Database en les liant aux concepts théoriques du cours. Chaque commande est accompagnée d'une explication détaillée de son fonctionnement et de son utilité.

7.1 Points Clés à Retenir

- **SQL*Plus** : Outil essentiel pour l'administration, avec de nombreuses commandes de formatage
- **Sessions** : Gestion via V\$SESSION, arrêt avec ALTER SYSTEM KILL SESSION
- **Instance** : Démarrage par étapes (NOMOUNT → MOUNT → OPEN)
- **CDB/PDB** : Architecture multitenant, gestion via ALTER SESSION SET CONTAINER
- **Paramètres** : Modification avec ALTER SYSTEM SET et différents SCOPE
- **Control Files** : Multiplexage essentiel, sauvegarde régulière recommandée

7.2 Bonnes Pratiques

1. Toujours multiplexer les control files (au moins 2 copies sur disques différents)

2. Sauvegarder le control file après chaque modification de structure
3. Tester les modifications de paramètres avec **SCOPE=MEMORY** avant de les rendre permanentes
4. Vérifier l'état des sessions avant de les tuer
5. Utiliser **SHUTDOWN IMMEDIATE** plutôt que **ABORT** sauf urgence
6. Documenter les modifications importantes dans le SPFILE