

# Résumé Complet Oracle Database 19c

## Administration et Architecture

Cours Oracle Database

27 janvier 2026

### Table des matières

<b>1</b>	<b>Introduction à Oracle Database</b>	<b>2</b>
1.1	Définitions . . . . .	2
1.2	Rôle du DBA (Database Administrator) . . . . .	2
1.3	Éditions Oracle Database 11g/19c . . . . .	2
1.4	Types de connexion . . . . .	2
<b>2</b>	<b>Architecture du Serveur Oracle</b>	<b>2</b>
2.1	Composants principaux . . . . .	2
2.2	Instance Oracle . . . . .	3
<b>3</b>	<b>Structures Mémoire</b>	<b>3</b>
3.1	SGA (System Global Area) . . . . .	3
3.1.1	Composants de la SGA . . . . .	3
3.2	PGA (Program Global Area) . . . . .	4
3.2.1	Composants de la PGA . . . . .	4
<b>4</b>	<b>Structure des Processus</b>	<b>4</b>
4.1	Types de processus . . . . .	4
4.2	Processus en arrière-plan principaux . . . . .	4
<b>5</b>	<b>Structure de Stockage</b>	<b>5</b>
5.1	Fichiers principaux . . . . .	5
5.2	Organisation logique . . . . .	5
5.2.1	Hiérarchie . . . . .	5
5.2.2	Tablespaces spéciaux . . . . .	6
<b>6</b>	<b>Gestion des Fichiers de Contrôle (Control Files)</b>	<b>6</b>
6.1	Description . . . . .	6
6.1.1	Définition et rôle . . . . .	6
6.1.2	Contenu du fichier de contrôle . . . . .	6
6.2	Informations sur les fichiers de contrôle . . . . .	7
6.2.1	Vues dynamiques . . . . .	7
6.2.2	Autres vues utilisant le control file . . . . .	7

6.3	Multiplexage des fichiers de contrôle . . . . .	7
6.3.1	Principe . . . . .	7
6.3.2	Multiplexage via SPFILE . . . . .	7
6.3.3	Multiplexage via PFILE . . . . .	8
6.4	Sauvegarde des fichiers de contrôle . . . . .	8
6.4.1	Recommandations . . . . .	8
6.4.2	Deux méthodes de sauvegarde . . . . .	8
6.5	Création d'un fichier de contrôle . . . . .	8
6.5.1	Étapes de création . . . . .	8
<b>7</b>	<b>Architecture Multitenant (CDB/PDB)</b>	<b>9</b>
7.1	Introduction . . . . .	9
7.2	Container Database (CDB) . . . . .	9
7.3	Pluggable Database (PDB) . . . . .	9
7.4	Avantages . . . . .	10
7.5	Gestion des PDBs . . . . .	10
7.5.1	Commandes principales . . . . .	10
<b>8</b>	<b>Tablespaces</b>	<b>10</b>
8.1	Création de tablespaces . . . . .	10
8.1.1	Tablespace permanent standard . . . . .	10
8.1.2	Options importantes . . . . .	10
8.1.3	BigFile Tablespace . . . . .	11
8.2	Tablespace Undo . . . . .	11
8.3	Tablespace Temporaire . . . . .	11
8.4	Gestion des tablespaces . . . . .	11
8.4.1	ONLINE/OFFLINE . . . . .	11
8.4.2	READ ONLY / READ WRITE . . . . .	12
8.4.3	Redimensionnement . . . . .	12
8.4.4	Déplacer/Renommer un fichier de données . . . . .	12
8.4.5	Supprimer un tablespace . . . . .	12
8.5	Informations sur tablespaces et datafiles . . . . .	13
8.5.1	Vues utiles . . . . .	13
<b>9</b>	<b>Interaction Utilisateur / Mémoire / Processus / Stockage</b>	<b>13</b>
9.1	Flux d'exécution d'une requête . . . . .	13
<b>10</b>	<b>Conclusion</b>	<b>13</b>

# 1 Introduction à Oracle Database

## 1.1 Définitions

**Base de données (DB)** : Ensemble structuré d'informations organisées en tables, permettant leur exploitation avec le moins de redondance possible.

**Système de gestion de base de données (SGBD)** : Logiciel qui gère et contrôle l'accès à une base de données, assurant une interface normalisée entre les applications et les bases de données.

## 1.2 Rôle du DBA (Database Administrator)

L'administrateur de base de données est responsable de :

- Installer Oracle et les outils, créer les bases de données
- Assurer la maintenance et la disponibilité de la base de données
- Créer la structure logique de stockage (tablespaces, tables, vues, index)
- Manager la structure physique (datafiles, control files, redo logs)
- Allouer et prévoir l'espace disque nécessaire
- Maintenir, contrôler et monitorer les accès des utilisateurs
- Assurer la sécurité du système
- Monitorer les performances
- Créer un plan de sauvegarde et de récupération

## 1.3 Éditions Oracle Database 11g/19c

- **Enterprise Edition (EE)** : Pour OLTP, data warehouses, applications critiques
- **Standard Edition One (SE1)** : Pour PME, facile à utiliser
- **Personal Edition (PE)** : Pour un seul utilisateur, compatible avec les autres éditions
- **Express Edition (XE)** : Version gratuite et légère pour développement

## 1.4 Types de connexion

1. **Connexion locale** : Utilisateur directement sur le serveur Oracle
2. **Client/Serveur (2 tiers)** : Application distante se connectant directement au serveur
3. **Multi-tiers (3 tiers)** : Utilisateur → Serveur applicatif → Serveur Oracle

# 2 Architecture du Serveur Oracle

## 2.1 Composants principaux

L'architecture d'un serveur Oracle se base sur deux composants :

1. **Instance** : Programme chargé en mémoire (RAM)

## 2. Stockage de Base de Données : Fichiers physiques sur disque

Elle comprend trois types de structures :

- **Memory Structures** (Structures mémoire) : SGA, PGA
- **Process Structures** (Structures de processus) : User, Server, Background, Listener
- **Storage Structures** (Structures de stockage) : Datafiles, Control files, Redo Logs

## 2.2 Instance Oracle

**Définition :** L'instance Oracle est un programme chargé sur la RAM du serveur. Elle existe uniquement dans la mémoire et est créée à chaque démarrage de la base de données.

**Caractéristiques :**

- Non persistante : disparaît au redémarrage
- Contient des structures de mémoire partagées (SGA) et des processus Oracle
- Chaque instance est associée à une seule base de données
- Avec RAC (Real Application Cluster), une base peut être ouverte par plusieurs instances

# 3 Structures Mémoire

## 3.1 SGA (System Global Area)

La SGA est une zone de mémoire partagée contenant les données et informations de contrôle pour une instance Oracle.

### 3.1.1 Composants de la SGA

#### 1. Shared Pool (Zone de mémoire partagée)

- **Library Cache** : Contient les requêtes SQL récemment utilisées, leurs plans d'exécution
- **Data Dictionary Cache** : Contient les métadonnées de la base (objets, utilisateurs, privilèges)

#### 2. Database Buffer Cache (Cache de tampons)

- Contient les blocs de données lus depuis les fichiers de données
- Permet un accès rapide aux données fréquemment utilisées
- Types spéciaux : Keep Pool, Recycle Pool, nK Buffer Cache

#### 3. Redo Log Buffer (Tampon de journalisation)

- Mémorise les modifications apportées à la base
- Utilisé pour la récupération d'instance
- Écrit par LGWR dans les fichiers redo log

#### 4. Large Pool

- Mémoire pour sauvegarde/restauration
- Serveurs partagés
- Processus serveur d'E/S

### 5. Java Pool

- Code Java et données JVM

### 6. Streams Pool

- Pour Oracle Streams (partage de données entre bases)

### 7. Fixed SGA

- Informations générales sur l'état de la base et communication entre processus

## 3.2 PGA (Program Global Area)

La PGA est une zone de mémoire privée associée à chaque processus serveur.

### 3.2.1 Composants de la PGA

- **Stack Space** : Espace de pile
- **UGA (User Global Area)** : Dans un environnement serveur dédié
  - **Cursor State** : Informations sur les curseurs SQL
  - **User Session Data** : Données de session utilisateur
  - **SQL Work Areas** :
    - Sort Area : Pour ORDER BY, GROUP BY
    - Hash Area : Pour jointures de hachage
    - Create Bitmap Area : Pour création d'index bitmap
    - Bitmap Merge Area : Pour plans d'exécution avec index bitmap

**Note** : Dans un environnement serveur partagé, la mémoire UGA est transférée dans la SGA (Shared Pool ou Large Pool).

## 4 Structure des Processus

### 4.1 Types de processus

1. **User Process (Processus utilisateur)** : Application ou outil qui se connecte à la base
2. **Server Process (Processus serveur)** : Traite les requêtes SQL, lit/écrit les fichiers de données
3. **Background Processes (Processus en arrière-plan)** : Démarrent avec l'instance
4. **Listener Process (Processus d'écoute)** : Gère les connexions réseau

### 4.2 Processus en arrière-plan principaux

**DBWn (Database Writer)** Écrit les blocs modifiés (dirty) du Buffer Cache sur disque. Déclenché quand :

- Le cache est plein
- Un checkpoint est atteint
- Pas de tampon "clean" disponible

**LGWR (Log Writer)** Écrit le Redo Log Buffer dans les fichiers redo log. Déclenché quand :

- Une transaction est validée (COMMIT)
- Un tiers du buffer est plein
- Avant qu'un DBWn écrive sur disque
- Toutes les 3 secondes

**CKPT (Checkpoint)** Met à jour le fichier de contrôle et les en-têtes des fichiers de données avec le SCN (System Change Number)

**SMON (System Monitor)** — Récupération des données au démarrage

- Nettoyage des segments temporaires inutilisés
- Récupération des transactions ignorées

**PMON (Process Monitor)** — Récupération des processus utilisateur échoués

- Nettoyage du cache de tampons
- Libération des ressources
- Surveillance des sessions inactives

**RECO (Recoverer)** Résout les transactions distribuées équivoques

**ARCn (Archiver)** Copie les fichiers redo log sur stockage pour archivage

**LREG (Listener Registration)** Enregistre l'instance auprès du listener

## 5 Structure de Stockage

### 5.1 Fichiers principaux

**Control Files** Fichiers binaires contenant des informations essentielles sur la structure physique de la base (noms et emplacements des datafiles, redo logs, tablespaces, checkpoints, SCN)

**Data Files** Contiennent les données utilisateurs, métadonnées et dictionnaire de données

**Online Redo Log Files** Enregistrent toutes les modifications pour récupération d'instance. Organisés en groupes, chaque groupe contient un ou plusieurs membres identiques

**Archived Redo Log Files** Contiennent l'historique complet des modifications. Permettent la récupération complète après perte de fichiers

**Backup Files** Copies de sauvegarde pour récupération

**Parameter Files** Configuration de l'instance au démarrage (PFILE ou SPFILE)

**Password Files** Accès distant pour administration (SYSDBA, SYSOPER)

**Trace Files & Alert Log** Diagnostic et suivi des erreurs

### 5.2 Organisation logique

#### 5.2.1 Hiérarchie

1. **Data Block** : Plus petite unité de stockage

- Taille définie lors de la création du tablespace : 2K, 4K, 8K (défaut), 16K, 32K

- Contient : En-tête, données des lignes, espace disponible
- 2. **Extent** : Ensemble de blocs contigus alloués pour un type d'information
  - Ne peut se trouver que dans un seul fichier de données
- 3. **Segment** : Ensemble d'extents alloués à une structure logique
  - Types : Segments de données, segments d'index, segments d'annulation (undo), segments temporaires
- 4. **Tablespace** : Regroupe les segments liés à une application
  - Composé d'un ou plusieurs fichiers de données
  - Ne peut appartenir qu'à une seule base de données
- 5. **Database** : Collection de tous les tablespaces

### 5.2.2 Tablespaces spéciaux

**SYSTEM** Métadonnées système et dictionnaire de données. Toujours en ligne

**SYSAUX** Composants supplémentaires (Enterprise Manager). Toujours en ligne

**UNDO** Segments d'annulation pour transactions non validées ou rollback

**TEMP** Stockage temporaire pour opérations de tri, segments temporaires. Utilise des tempfiles (toujours NOLOGGING)

**USERS** Stocke les objets et données des utilisateurs (par défaut)

## 6 Gestion des Fichiers de Contrôle (Control Files)

### 6.1 Description

#### 6.1.1 Définition et rôle

Le fichier de contrôle est un fichier binaire qui :

- Contient des informations sur la structure physique de la base
- Garantit la cohérence de toute la base
- Est commun pour CDB et les PDBs
- Est créé pendant la création de la base et modifié en permanence
- Est nécessaire au démarrage, au bon fonctionnement et à la restauration
- Est associé à une seule base de données Oracle
- Est lu avant l'ouverture de la base pour vérifier son intégrité
- Ne peut être modifié que par le serveur Oracle (pas par l'administrateur)

#### 6.1.2 Contenu du fichier de contrôle

- Nom et identifiant de la base de données (SID)
- Horodatage de création
- Noms et emplacements des fichiers de données et redo logs
- Informations sur les tablespaces
- Historique de journalisation

- Statut et emplacement des journaux archivés
- Emplacement et statut des sauvegardes
- Numéro de séquence du journal en cours
- Points de reprise (checkpoints)
- Numéro de changement système (SCN) actuel

## 6.2 Informations sur les fichiers de contrôle

### 6.2.1 Vues dynamiques

- **V\$CONTROLFILE** : Liste des fichiers de contrôle de l'instance
- **V\$PARAMETER** : Paramètre `control_files` avec les chemins
- **V\$DATABASE** : Informations générales sur la base
- **V\$CONTROLFILE\_RECORD\_SECTION** : Sections du fichier de contrôle

Commandes utiles :

```
1 -- Afficher les control files
2 SHOW PARAMETER control_files;
3
4 -- Ou via la vue
5 SELECT name FROM v$controlfile;
6
7 -- Informations sur la base
8 SELECT * FROM v$database;
```

### 6.2.2 Autres vues utilisant le control file

- V\$BACKUP, V\$TEMPFILE, V\$ARCHIVE, V\$LOGFILE
- V\$ARCHIVED\_LOG, V\$DATAFILE, V\$TABLESPACE
- V\$LOG, V\$LOGHIST

## 6.3 Multiplexage des fichiers de contrôle

### 6.3.1 Principe

- Il est **fortement recommandé** de multiplexer les fichiers de contrôle
- Chaque copie doit être sur un disque différent
- Si un fichier est perdu, une copie multiplexée peut être utilisée pour redémarrer sans restaurer
- Les fichiers peuvent être multiplexés jusqu'à 8 fois

### 6.3.2 Multiplexage via SPFILE

Étapes :

1. Modifier le paramètre `control_files` dans SPFILE :



```
1 ALTER SYSTEM SET control_files =  
2 '/u01/app/oracle/oradata/orcl/Disk1/control01.ctl',  
3 '/u01/app/oracle/oradata/orcl/Disk2/control02.ctl'  
4 SCOPE=SPFILE;
```

2. Arrêter la base : `SHUTDOWN IMMEDIATE`
3. Copier physiquement le fichier :

```
1 cp '/u01/app/oracle/oradata/orcl/Disk1/control01.ctl' \  
2 '/u01/app/oracle/oradata/orcl/Disk2/control02.ctl'
```

4. Démarrer la base : `STARTUP`

### 6.3.3 Multiplexage via PFILE

Même principe mais modification directe du fichier `init.ora` :

```
1 control_files = (  
2     '/u01/app/oracle/oradata/orcl/Disk1/control01.ctl',  
3     '/u01/app/oracle/oradata/orcl/Disk2/control02.ctl'  
4 )
```

## 6.4 Sauvegarde des fichiers de contrôle

### 6.4.1 Recommandations

Oracle recommande de sauvegarder le fichier de contrôle à chaque modification de la structure de la base (ajout, renommage ou suppression d'un fichier de données ou redo log).

### 6.4.2 Deux méthodes de sauvegarde

1. Sauvegarde binaire :

```
1 ALTER DATABASE BACKUP CONTROLFILE TO  
2 '/backup/controlfile_backup.ctl';
```

2. Sauvegarde en fichier trace (script SQL de recreation) :

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

Le fichier trace est créé dans le répertoire `udump` ou `trace`.

## 6.5 Création d'un fichier de contrôle

### 6.5.1 Étapes de création

1. Sauvegarder le fichier de contrôle existant
2. Créer un script SQL à partir de la sauvegarde trace ou utiliser les informations de `V$DATABASE`, `V$DATAFILE`, `V$LOGFILE`
3. Arrêter la base : `SHUTDOWN IMMEDIATE`

4. Exécuter le script CREATE CONTROLFILE
5. Ouvrir la base : ALTER DATABASE OPEN
6. Si nécessaire, récupérer la base : RECOVER DATABASE

**Exemple de script CREATE CONTROLFILE :**

```
1 CREATE CONTROLFILE REUSE DATABASE "ORCL" NORESETLOGS
2 NOARCHIVELOG
3 MAXLOGFILES 16
4 MAXLOGMEMBERS 3
5 MAXDATAFILES 100
6 MAXINSTANCES 8
7 MAXLOGHISTORY 292
8 LOGFILE
9 GROUP 1 '/u01/oradata/orcl/redo01.log' SIZE 50M,
10 GROUP 2 '/u01/oradata/orcl/redo02.log' SIZE 50M
11 DATAFILE
12 '/u01/oradata/orcl/system01.dbf',
13 '/u01/oradata/orcl/sysaux01.dbf',
14 '/u01/oradata/orcl/users01.dbf'
15 CHARACTER SET AL32UTF8;
```

## 7 Architecture Multitenant (CDB/PDB)

### 7.1 Introduction

Depuis Oracle 12c, Oracle a introduit l'architecture multitenant avec les concepts de Container Database (CDB) et Pluggable Database (PDB).

### 7.2 Container Database (CDB)

- Collection logique de données et métadonnées
- Permet de regrouper plusieurs bases de données
- Contient au moins un conteneur Root et peut contenir plusieurs PDBs
- Contient par défaut :
  - **CDB\$ROOT (Root)** : Métadonnées communes aux utilisateurs et PDBs
  - **PDB\$SEED (Seed)** : Base template pour créer de nouvelles PDB

### 7.3 Pluggable Database (PDB)

- Base de données "insérable" dans un CDB
- Contient des objets, schémas, données
- Apparaît aux utilisateurs comme une base normale
- Peut être clonée à partir du Seed PDB\$SEED

## 7.4 Avantages

- Partage des ressources entre les bases
- Déplacement facile des données et du code
- Administration simplifiée (sauvegarde, restauration centralisée)

## 7.5 Gestion des PDBs

### 7.5.1 Commandes principales

```
1  -- Connexion en SYSDBA
2  sqlplus / as sysdba
3
4  -- Afficher les PDBs
5  SHOW PDBS;
6
7  -- Afficher le container courant
8  SHOW CON_NAME;
9
10 -- Changer de container
11 ALTER SESSION SET CONTAINER=pdb1;
12
13 -- Vérifier l'état des PDBs
14 SELECT name, open_mode FROM v$pdb;
15
16 -- Ouvrir une PDB
17 ALTER PLUGGABLE DATABASE pdb1 OPEN;
18
19 -- Fermer une PDB
20 ALTER PLUGGABLE DATABASE pdb1 CLOSE;
```

# 8 Tablespaces

## 8.1 Création de tablespaces

### 8.1.1 Tablespace permanent standard

```
1 CREATE TABLESPACE tbs_01
2 DATAFILE '/u01/oradata/tbs_01.dbf' SIZE 100M;
```

### 8.1.2 Options importantes

- `MINIMUM EXTENT integer[K|M]` : Taille minimale d'un extent
- `BLOCKSIZE integer[K]` : Taille des blocs
- `LOGGING|NOLOGGING` : Journalisation des modifications
- `ONLINE|OFFLINE` : Disponibilité du tablespace
- `PERMANENT|TEMPORARY` : Type de tablespace

- EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K : Gestion des extents
- SEGMENT SPACE MANAGEMENT AUTO|MANUAL : Gestion de l'espace dans les segments

### 8.1.3 BigFile Tablespace

Tablespace composé d'un seul très gros fichier (jusqu'à 1TB ou plus) :

```
1 CREATE BIGFILE TABLESPACE tbs_big
2 DATAFILE '/u01/oradata/tbs_big.dbf' SIZE 1T;
```

## 8.2 Tablespace Undo

Stocke les segments d'annulation pour transactions non validées :

```
1 CREATE UNDO TABLESPACE undo1
2 DATAFILE '/u01/oradata/undo01.dbf' SIZE 40M;
```

**Note** : Ne peut contenir aucun autre objet.

## 8.3 Tablespace Temporaire

Utilisé pour opérations de tri lorsque la mémoire est insuffisante :

```
1 CREATE TEMPORARY TABLESPACE temp
2 TEMPFILE '/u01/oradata/temp01.dbf' SIZE 500M
3 EXTENT MANAGEMENT LOCAL UNIFORM SIZE 4M;
4
5 -- Définir comme tablespace temporaire par défaut
6 ALTER DATABASE DEFAULT TEMPORARY TABLESPACE temp;
```

**Caractéristiques des tempfiles :**

- Toujours en mode NOLOGGING
- Ne peuvent pas être en lecture seule
- Ne peuvent pas être renommés
- Ne peuvent pas être restaurés physiquement

**Restrictions** : Les DEFAULT TEMPORARY TABLESPACES ne peuvent pas être supprimés, mis offline, ou transformés en tablespaces permanents tant qu'ils sont par défaut.

## 8.4 Gestion des tablespaces

### 8.4.1 ONLINE/OFFLINE

```
1 -- Mettre offline
2 ALTER TABLESPACE userdata OFFLINE;
3
4 -- Mettre online
5 ALTER TABLESPACE userdata ONLINE;
```

**Restrictions** : SYSTEM, SYSAUX, TEMPORARY par défaut et tablespaces avec undo actifs ne peuvent pas être mis offline.

### 8.4.2 READ ONLY / READ WRITE

```
1 -- Lecture seule
2 ALTER TABLESPACE userdata READ ONLY;
3
4 -- Lecture/écriture
5 ALTER TABLESPACE userdata READ WRITE;
```

### 8.4.3 Redimensionnement

#### 1. Autoextend :

```
1 CREATE TABLESPACE user_data
2 DATAFILE '/u01/oradata/userdata01.dbf' SIZE 200M
3 AUTOEXTEND ON NEXT 10M MAXSIZE 500M;
```

#### 2. Redimensionnement manuel :

```
1 ALTER DATABASE DATAFILE '/u01/oradata/userdata01.dbf'
2 RESIZE 600M;
```

#### 3. Ajouter des fichiers :

```
1 ALTER TABLESPACE user_data
2 ADD DATAFILE '/u01/oradata/userdata03.dbf' SIZE 200M;
```

### 8.4.4 Déplacer/Renommer un fichier de données

#### Étapes :

1. Mettre le tablespace offline
2. Déplacer/copier le fichier avec le système d'exploitation
3. Renommer dans Oracle :

```
1 ALTER TABLESPACE userdata RENAME DATAFILE
2 '/u01/oradata/userdata01.dbf' TO '/u02/oradata/userdata01.dbf'
   ;
```

4. Mettre le tablespace online
5. Supprimer l'ancien fichier si nécessaire

### 8.4.5 Supprimer un tablespace

```
1 DROP TABLESPACE userdata
2 INCLUDING CONTENTS AND DATAFILES;
```

#### Options :

- INCLUDING CONTENTS : Supprime les segments
- INCLUDING CONTENTS AND DATAFILES : Supprime aussi les fichiers physiques
- CASCADE CONSTRAINTS : Supprime les contraintes d'intégrité référentielle

**Restrictions :** Ne peut pas supprimer SYSTEM, SYSAUX ou tablespace avec segments actifs.

## 8.5 Informations sur tablespaces et datafiles

### 8.5.1 Vues utiles

- `DBA_TABLESPACES`, `V$TABLESPACE` : Informations sur les tablespaces
- `DBA_DATA_FILES`, `V$DATAFILE` : Informations sur les datafiles
- `DBA_TEMP_FILES`, `V$TEMPFILE` : Informations sur les fichiers temporaires

## 9 Interaction Utilisateur / Mémoire / Processus / Stockage

### 9.1 Flux d'exécution d'une requête

1. L'instance Oracle démarre sur le serveur
2. L'utilisateur lance un processus utilisateur via une application
3. Le listener détecte la demande et crée un processus serveur dédié
4. L'utilisateur exécute une requête SQL :
  - Le serveur cherche la requête dans la Shared Pool
  - Vérifie les privilèges dans le Data Dictionary Cache
  - Exécute la requête avec les données du Database Buffer Cache ou du fichier de données
5. Les modifications sont enregistrées :
  - Dans le Redo Log Buffer
  - LGWR écrit dans les fichiers redo log
  - DBWn écrit les blocs modifiés sur disque au moment opportun
6. Le serveur renvoie le résultat à l'utilisateur
7. Les processus en arrière-plan surveillent et maintiennent la cohérence

## 10 Conclusion

Ce résumé couvre les aspects fondamentaux de l'administration Oracle Database 19c, de l'architecture générale jusqu'à la gestion détaillée des fichiers de contrôle. La compréhension de ces concepts est essentielle pour l'administration efficace d'une base de données Oracle.

### Points clés à retenir :

- L'instance (mémoire + processus) est distincte du stockage (fichiers)
- La SGA est partagée, la PGA est privée à chaque processus serveur
- Les fichiers de contrôle sont critiques et doivent être multiplexés
- Les tablespaces organisent logiquement le stockage
- L'architecture multitenant (CDB/PDB) permet la consolidation