



ECOLE MAROCAINE DES
SCIENCES DE L'INGENIEUR

Membre de

HONORIS UNITED UNIVERSITIES

Administration Oracle 19c

C. EL ASRY

PLAN DU COURS



Chapitre 1: Introduction

Chapitre 2: Formatage des données

Chapitre 3: Architecture

Chapitre 4: Instance Oracle

Chapitre 5: Gestion des paramètres et des fichiers de paramètre

Chapitre 6: Gestion des Fichiers de journalisation

Chapitre 7: Archivelog

Chapitre 8: Gestion des fichiers de contrôle

Chapitre 9: Tablespaces

Chapitre 1: Introduction

Chapitre 2: Formatage des données

Chapitre 3: Architecture

Chapitre 4: Instance Oracle

Chapitre 5: Gestion des paramètres et des fichiers de paramètre

Chapitre 6: Gestion des Fichiers de journalisation

Chapitre 7: Archivelog

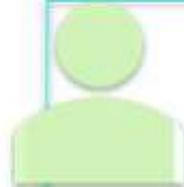
Chapitre 8: Gestion des fichiers de contrôle

Chapitre 9: Tablespaces

PLAN DE LA SEANCE



Introduction à Oracle Database



Rôle de l'administrateur de base de données

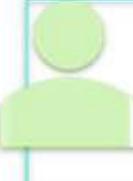


Outils d'administration d'une BD Oracle

PLAN DE LA SEANCE



Introduction à Oracle Database



Rôle de l'administrateur de base de données



Outils d'administration d'une BD Oracle

- 1- Qu'est-ce qu'une base de données (Rappel)**
- 2- Rôle des bases de données (Rappel)**
- 3- Modèle de données relationnel (Rappel)**
- 4- Qu'est-ce qu'un Système de Gestion de Bases de Données (SGBD) (Rappel)**
- 5-Oracle Database - Vue d'ensemble**

INTRODUCTION A ORACLE DATABASE

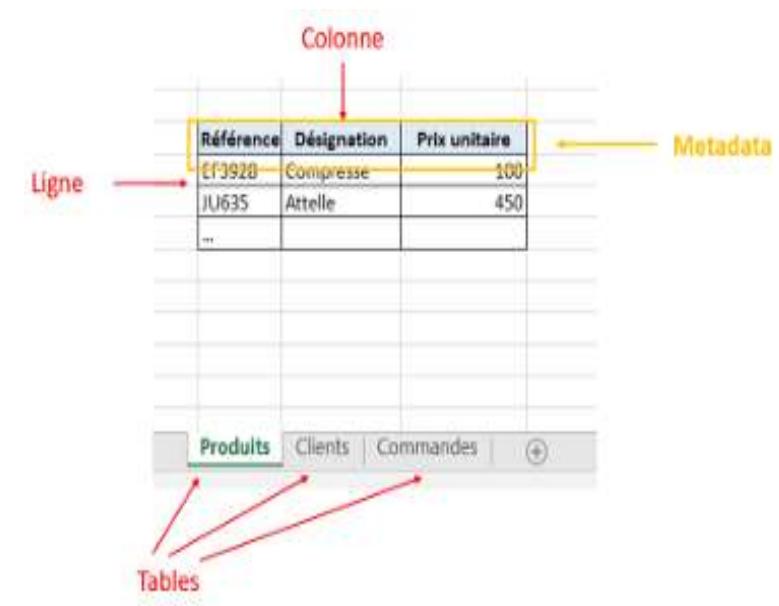
**QU'EST-CE QU'UNE BASE
DE DONNÉES ?**

INTRODUCTION A ORACLE DATABASE

Qu'est-ce qu'une base de données?

Une base de donnée est un ensemble de donnée stockées d'une manière structurée , organisée avec le moins de redondance possible .

Les données peuvent être de différents types, tels que des textes, des nombres, des images, etc



INTRODUCTION A ORACLE DATABASE

Les bases de données jouent un rôle essentiel dans le stockage et la gestion des données pour les organisations.

Elles permettent de stocker et de récupérer des informations de manière rapide et efficace.

Les bases de données facilitent, entre autres, la prise de décision, l'analyse des données, la gestion des transactions...

Exemples d'utilisation :

- Sites web (utilisateurs, produits, commandes)
- Systèmes bancaires (comptes, transactions)
- Applications scolaires (étudiants, cours, notes)

INTRODUCTION A ORACLE DATABASE

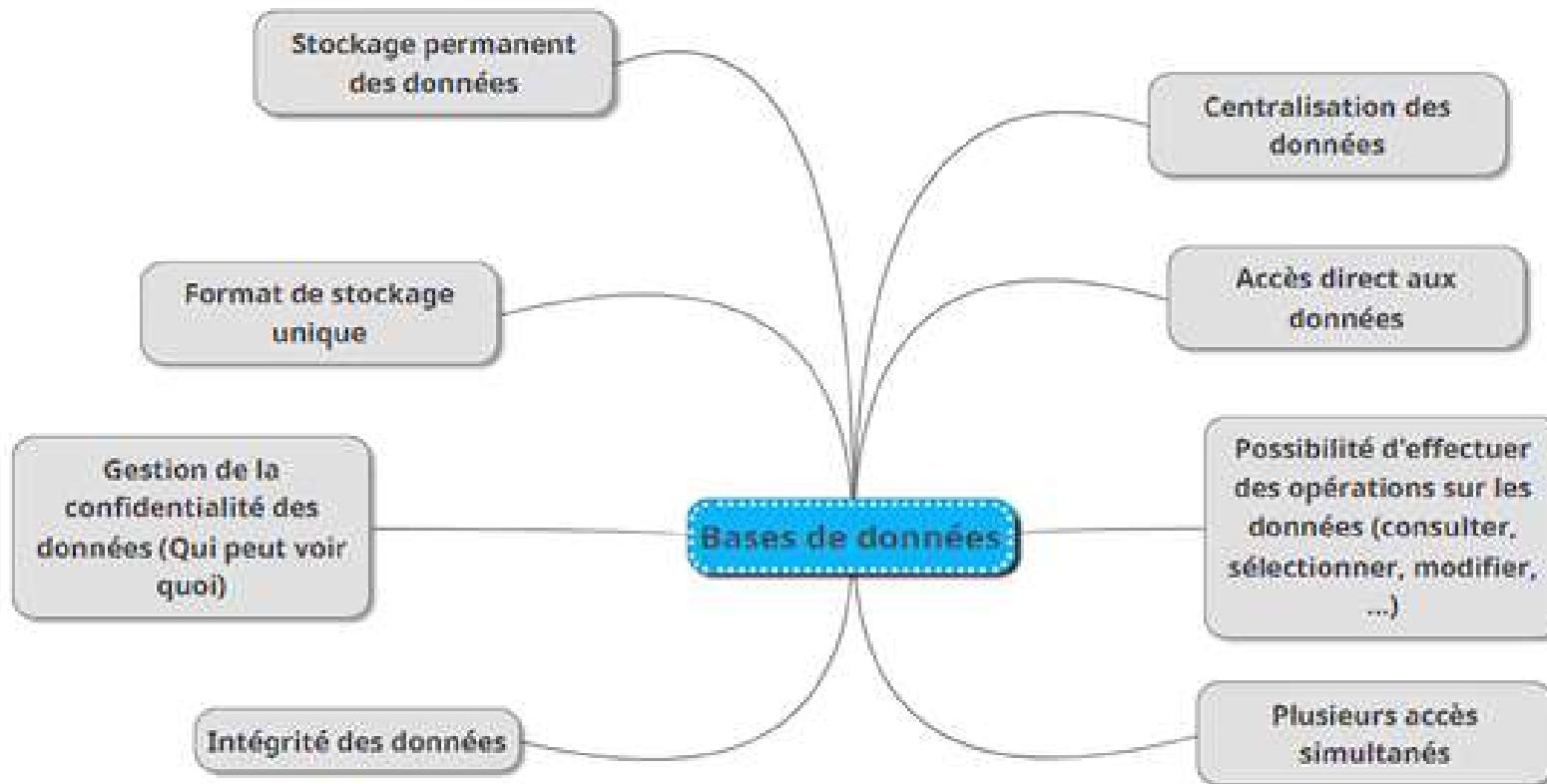
**RÔLE DES BASES DE
DONNÉES**

INTRODUCTION A ORACLE DATABASE

- Les bases de données jouent un rôle essentiel dans le stockage et la gestion des données pour les organisations.
- Elles permettent de stocker et de récupérer des informations de manière rapide et efficace.
- Les bases de données facilitent, entre autres, la prise de décision, l'analyse des données, la gestion des transactions..

INTRODUCTION A ORACLE DATABASE

- Rôle des bases de données



INTRODUCTION A ORACLE DATABASE

MODÈLE DE DONNÉES RELATIONNEL

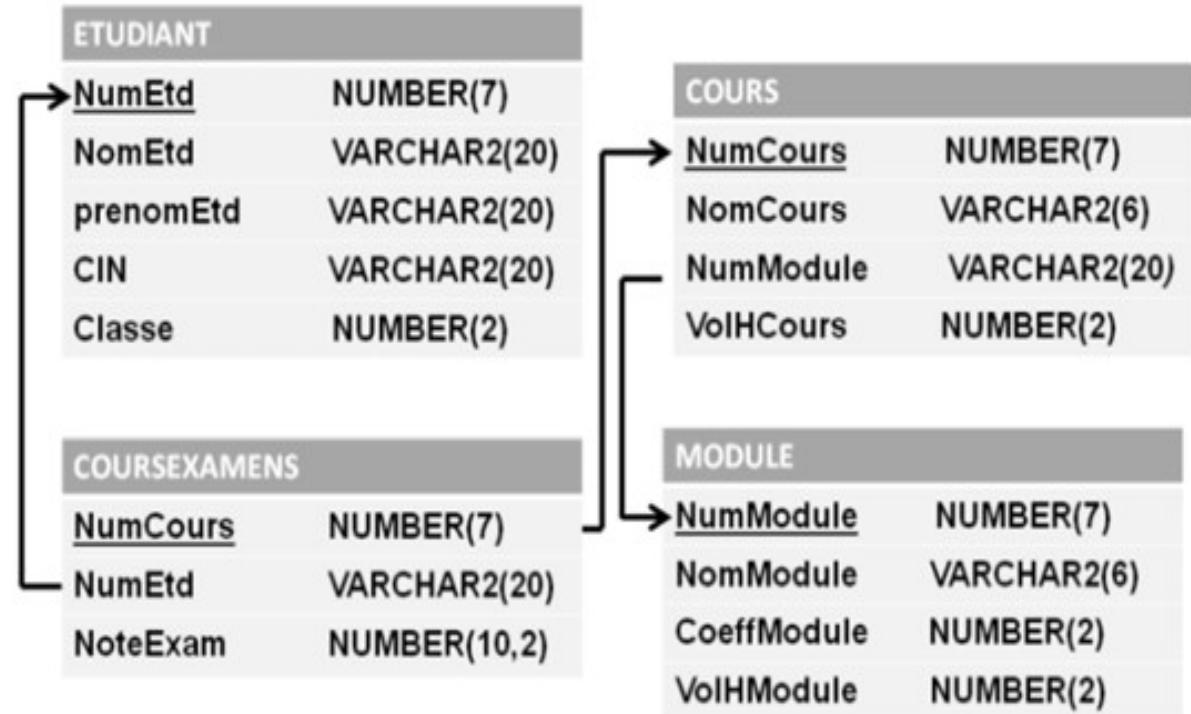
INTRODUCTION A ORACLE DATABASE

- Besoin
- Conception
- Implémentation
- Utilisation
- Evaluation
- Evolution

- **Identification des besoins :**
recensement des données
- **Conception :** modélisation des données
 - Modèle conceptuel
 - Modèle relationnel
- **Implémentation :** Réalisation à l'aide
d'un SGBD
- **Utilisation – Evaluation et Evolution :**
Avec les acteurs concernés

INTRODUCTION A ORACLE DATABASE

- Dans un modèle de données relationnel, les données sont organisées en tables avec des lignes et des colonnes.
- Les tables sont reliées les unes aux autres par des clés, ce qui permet de gérer des données structurées de manière cohérente.



INTRODUCTION A ORACLE DATABASE

• VOCABULAIRE

- **Table ou relation** : ensemble d'enregistrements correspondant à une description précise (liste des attributs, clé primaire, etc.)
- **Attribut ou colonne** : un élément de l'enregistrement qui possède un type et une longueur
- **Tuple ou Enregistrement** : un élément d'une table correspondant à un ensemble de valeur d'attributs.
- **clé primaire** : attribut choisi dans la description de la table comme étant celui dont la valeur identifie de manière unique chaque tuple de la relation.
- **clé étrangère** : attribut d'une relation dont la valeur est la clé primaire d'une autre relation (ex : le n° de client dans la table commande).

Nº	Nom	Prénom	Ville	C.A.
1024	Benharraf	Mohamed	Casablanca	10000

INTRODUCTION A ORACLE DATABASE

- **Méthodes de conception**

- MERISE (Méthode d'Etude et de Réalisation Informatique pour les Systèmes d'Entreprise) est une méthode d'analyse et de réalisation des systèmes d'information qui est élaborée en plusieurs étapes: schéma directeur, étude préalable, étude détaillée et la réalisation.
- UML (Unified Modeling Langage), est un langage de modélisation des systèmes standard, qui utilise des diagrammes pour représenter chaque aspect d'un système

INTRODUCTION A ORACLE DATABASE

**Qu'est-ce qu'un Système de Gestion de
Bases de Données (SGBD)**

INTRODUCTION A ORACLE DATABASE

La gestion d'une base de données nécessite:

- La gestion du stockage physique de façon transparente/utilisateurs
- La gestion des procédures d'accès et de traitement des données (optimisation des traitements, faciliter l'accès aux données...)
- La structuration des données
- La sécurité des données stockées

SGBD (Système de Gestion de Base de Données) : logiciel qui permet la **création**, la **modification**, la **recherche** et **la gestion efficace des bases de données**.

INTRODUCTION A ORACLE DATABASE

• FONCTIONNALITES

- Description des données : Un SGBD permet de décrire les données (Type, Longueur, Nature, etc.).
- Manipulation des données : Un SGBD permet l'exécution des opérations d'interrogation, d'ajout, suppression, modification des données.
- Gestion du stockage : le S.G.B.D organise les données en fichiers, qui reste transparente pour l'utilisateur.
- Partage des données Vu les multitâches et multiutilisateurs, le système gère le partage des données entre les différents utilisateurs (avec le principe du verrouillage des données).
- Confidentialité & Sécurité Définition des droits d'accès aux données : droit de lecture, modification, ajout, suppression...
- Sauvegarde et Restauration
- Contrôle d'intégrité : le S.G.B.D vérifie que les mises à jour effectuées sur les données respectent les règles de description des données.

INTRODUCTION A ORACLE DATABASE

Plusieurs SGBD relationnels existent sur le marché. Exemples:



INTRODUCTION A ORACLE DATABASE



ORACLE®

- Base de donnée importante
- Multiplateforme
- Fonctionnalité riche

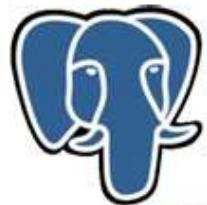
- Cout de la licence
- Consommation des ressources
- Complexité d'administration


Microsoft
SQL Server

- Base de donnée importante
- Windows
- Facilité d'administration

- Cout de la licence
- Fonctionnel sous Windows uniquement

INTRODUCTION A ORACLE DATABASE



PostgreSQL



- Base de donnée importante
- Multiplateforme
- Programmation avancée

- Complexité de déploiement



- Base de donnée importante
- Multiplateforme

- Pas adapté pour les volumes importants

INTRODUCTION A ORACLE DATABASE

**ORACLE DATABASE : VUE
D'ENSEMBLE**

INTRODUCTION A ORACLE DATABASE

Oracle Database est l'un des SGBD les plus populaires et les plus puissants au monde, édité par la société du même nom (Oracle Corporation, société Américaine-<http://www.oracle.com>)

- Compatible avec différents OS: Windows, Linux, IBM AIX...
- Offre une gamme complète de fonctionnalités pour la gestion de données, y compris la **sécurité**, la **haute disponibilité**, la **performance** et **la gestion des données volumineuses**.
- Oracle est utilisé dans divers secteurs (finance, santé, logistique, télécommunications...).

INTRODUCTION A ORACLE DATABASE

- Oracle Corporation est une entreprise américaine créée par Larry Ellison en 1977.
- Quelques produits et services Oracle:



Serveur d'application



Progiciel de gestion intégré
(ERP)



INTRODUCTION A ORACLE DATABASE

Historique des versions Oracle Database:

- 1979:Oracle Database est lancé pour la première fois par Larry Ellison, Bob Miner et Ed Oates sous le nom d'Oracle V2.
- 1983:Oracle Database 3 est publié, marquant le début de l'utilisation du modèle relationnel.
- 1992:Oracle 7 est lancé, introduisant des améliorations significatives telles que la gestion des transactions distribuées et la prise en charge des déclencheurs au niveau de la base de données.



INTRODUCTION A ORACLE DATABASE

Historique des versions Oracle Database:

- 1997: Oracle 8 apporte des fonctionnalités telles que les objets relationnels et les procédures stockées Java.
- 2001: Oracle 9i met l'accent sur l'auto-gestion et l'optimisation automatique des performances.
- 2005: Oracle Database 10g introduit l'architecture de grille et la gestion automatique des ressources.
- 2007: Oracle Database 11g se concentre sur la gestion automatisée et la consolidation des données.



INTRODUCTION A ORACLE DATABASE

Historique des versions Oracle Database:

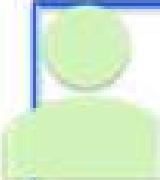
- 2013: Oracle Database 12c introduit le concept de base de données en tant que service(DBaaS) et met en œuvre des améliorations majeures telles que la prise en charge du multitenant et la gestion des données in-memory.
- 2019: Oracle continue à publier des versions majeures avec des améliorations de la sécurité, des performances et de la gestion des données, la dernière version connue étant Oracle Database 19c.
- 2023:Oracle continue à publier des versions majeures avec des améliorations de la sécurité, des performances et de la gestion des données, la dernière version connue étant Oracle Database19c



PLAN DE LA SEANCE



Introduction à Oracle
Database



Rôle de
l'administrateur de
base de données



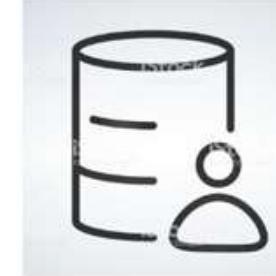
Outils d'administration
d'une BD Oracle

INTRODUCTION A ORACLE DATABASE

- **INTERVENANTS**

Administrateur
Base de Données

Analyste
Programmeur



UTILISATEUR BD

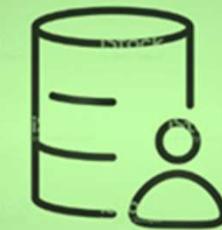
LES MÉTIERS AUTOUR DES BASES DE DONNÉES

INTERVENANTS

- Administrateurs
- Responsable de la sécurité
- Administrateurs réseaux
- Développeurs d'applications
- Utilisateurs: modifier les données, créer des rapports...

**Administrateur
Base de Données**

**Analyste
Programmeur**



UTILISATEUR BD

RÔLE DE L'ADMINISTRATEUR DE LA BASE DE DONNÉES

L'administrateur de la base de données est principalement chargé de:

1. Evaluer le matériel du serveur de base de données
2. Installer le logiciel Oracle
3. Elaborer la stratégie pour la base de données et la sécurité
4. Créer, migrer et ouvrir la base de données
5. Sauvegarder la base de données
6. Inscrire les utilisateurs système et définir leur accès
7. Implémenter la conception de la base de données
8. Récupérer la base de données suite à une panne
9. Surveiller les performances de la base de données

RÔLE DE L'ADMINISTRATEUR DE LA BASE DE DONNÉES



Création

- + Installer le serveur Oracle et les outils applicatifs
- + Créer les bases de données
- + Créer la structure logique (TS, indexes...)
- + Définir et gérer les priviléges et des droits d'accès
- + Assister l'équipe projet/dev pour le dimensionnement des serveurs

Admin, maintenance et optimisation

- + Monitorer les performances de la BD (taux d'occupation des TS, lenteur de requêtes,...)
- + Superviser le dimensionnement du serveur (++ volumétrie)
- + Superviser les bases de données (détection d'incidents)
- + Optimiser les requêtes

Support

- + Assister les développeurs et l'équipe projet
- + Assister les utilisateurs (BD non accessible, dégradation des performances, incidents applicatifs, ...)

Sécurité

- + Contrôler les accès (applications, utilisateurs, ...)
- + Gérer les échanges avec les applications externes

Intégrité des données

- + Mettre en place un plan de sauvegarde (backup) et de restauration
- + Vérifier le bon déroulement de la sauvegarde

PLAN DE LA SEANCE



**Introduction à Oracle
Database**



**Rôle de
l'administrateur de
base de données**



**Outils d'administration
d'une BD Oracle**

Les composants d'Oracle

On peut classer les outils d'Oracle selon diverses catégories :

- Les outils d'administration
- Les outils de développement
- Les outils de communication
- Les outils de génie logiciel
- Les outils d'aide à la décision

Outils d'administration de BD Oracle

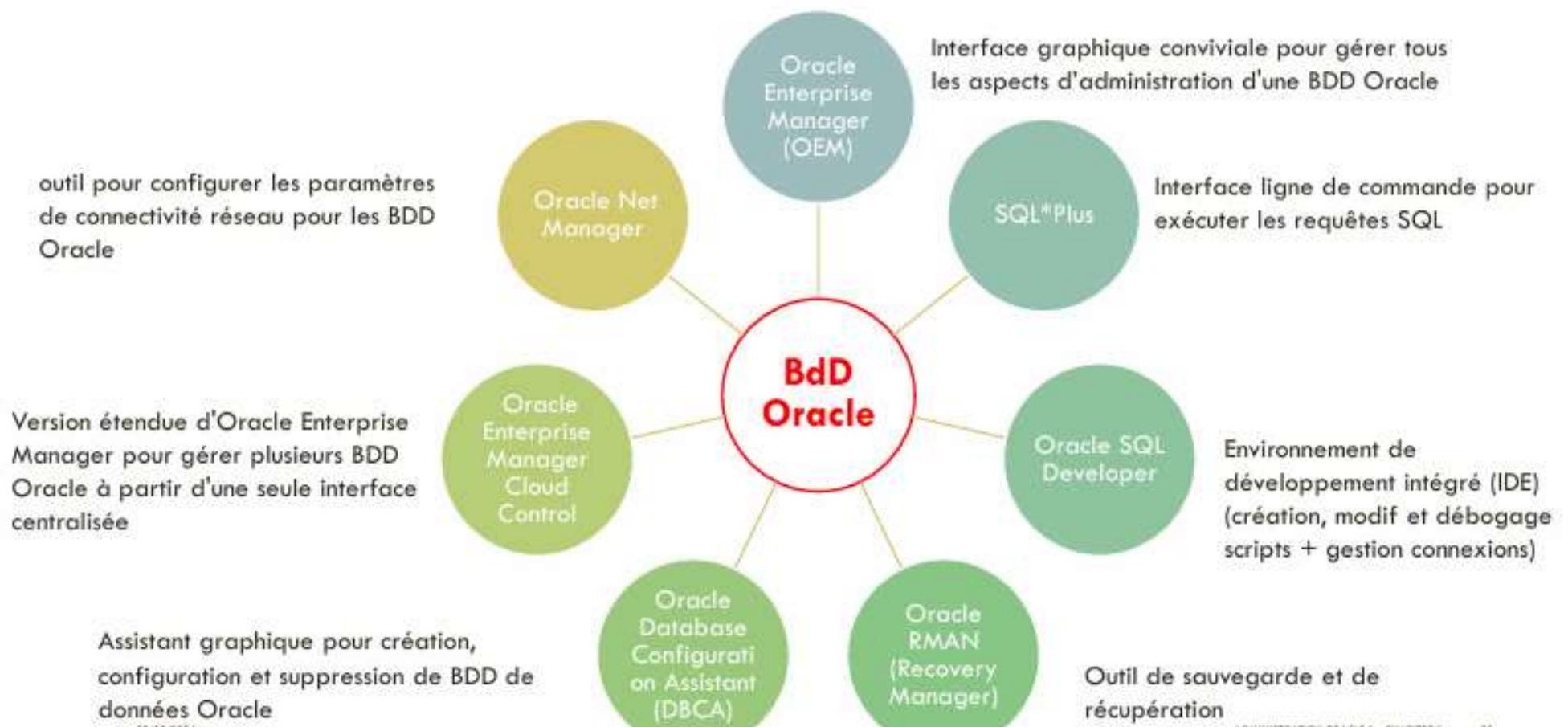
- **Oracle Universal Installer (OUI)** : permet d'installer le logiciel Oracle et ses options
- **Database Configuration Assistant (DBCA)** : permet de créer, configurer et supprimer les bases de données
- **Database Upgrade Assistant (DBUA)** : mise à niveau d'une base Oracle existante vers une nouvelle version.
- **Oracle Net Manager (NetMgr)** : permet de configurer la connectivité réseau des bases de données et applications Oracle
- **Oracle Net Configuration Assistant (NetCA)** : permet de configurer et de gérer des configurations Oracle Network.
- **Oracle Enterprise Manager (EM)** : interface web pour gérer la base de données
 - Enterprise Manager Database : pour administrer une seule base de données.
 - Enterprise Manager Grid Control : pour administrer simultanément plusieurs bases de données.
- **SQL*Plus** : est une interface de ligne de commande standard permettant de gérer la base de données.
- **Recovery Manager (RMAN)** est une solution Oracle complète pour les opérations de sauvegarde, de restauration et de récupération de l'ensemble de la base de données ou de fichiers spécifiques

Outils de développement d'Oracle

Oracle propose également de nombreux outils de développement permettant d'automatiser la création d'applications s'interfaisant avec la base de données. Ces outils de développement sont :

- Oracle Designer
- Oracle Developer
- SQL*Plus : une interface interactive permettant d'envoyer des requêtes SQL et PL/SQL à la base de données.

OUTILS D'ADMINISTRATION D'UNE BASE DE DONNÉES ORACLE



Chapitre 1: Introduction

Chapitre 2: Installation Oracle Database 19c et Formatage des données

Chapitre 3: Architecture

Chapitre 4: Instance Oracle

Chapitre 5: Gestion des paramètres et des fichiers de paramètre

Chapitre 6: Gestion des Fichiers de journalisation

Chapitre 7: Archivelog

Chapitre 8: Gestion des fichiers de contrôle

Chapitre 9: Tablespaces



Installation Oracle Database 11g

1. Modes d'installation Oracle 19c
2. Optimal flexible Architecture (OFA)
3. Installation d'Oracle Database 19c



Formatage des données et des rapports sous SQL Plus

INSTALLATION ORACLE DATABASE 19C

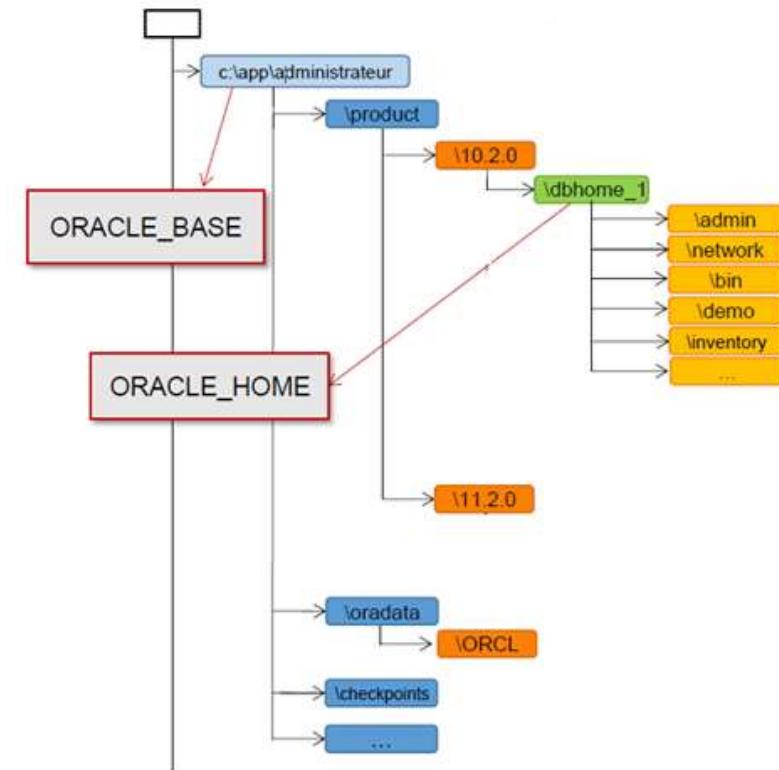
Quelques modes d'installation d'Oracle 19c:

- 1. Installation Standard (Le plus courant)** : Installe une base de données autonome avec tous les composants nécessaires (fichiers de données, fichiers de contrôle, fichiers journaux, et outils d'administration). Convient pour BdD sur une seule machine.
- 2. Installation Personnalisée** : Installation plus légère, adaptée aux besoins particuliers, choix des composants à installer.
- 3. Installation en Mode Silencieux** : Installation sans interface utilisateur graphique. Généralement utilisée pour automatiser l'installation sur plusieurs serveurs ou dans des environnements de script.
- 4. Installation en Mode Oracle RAC** : Permet d'exécuter une base de données Oracle sur plusieurs serveurs en cluster.

INSTALLATION ORACLE DATABASE 19C

Oracle Optimal Flexible Architecture (OFA) = Ensemble de règles de configuration permettant de garantir des installations Oracle bien organisées.

→ Simplification de l'administration, du support et de la maintenance.



ORACLE_HOME: Définit le chemin vers le répertoire d'installation du logiciel Oracle

ORACLE_BASE: C'est le répertoire racine où Oracle stocke les fichiers administratifs et de gestion.

ORACLE_SID: Identifie l'instance Oracle locale (le nom de la base/instance).

INSTALLATION ORACLE DATABASE 19C

3 phases à respecter pour une installation correcte d'Oracle Database:

- 1. Pré-installation:** préparation de l'environnement selon les prérequis d'installation d'Oracle.
- 1. Installation:** Déroulement des étapes d'installation
- 1. Post-installation:** Configuration des patchs

INSTALLATION ORACLE DATABASE 19C

Pré-installation sous Windows:

- Prérequis matériels et logiciels d'installation d'Oracle 11g sous Windows (source documentation Oracle)

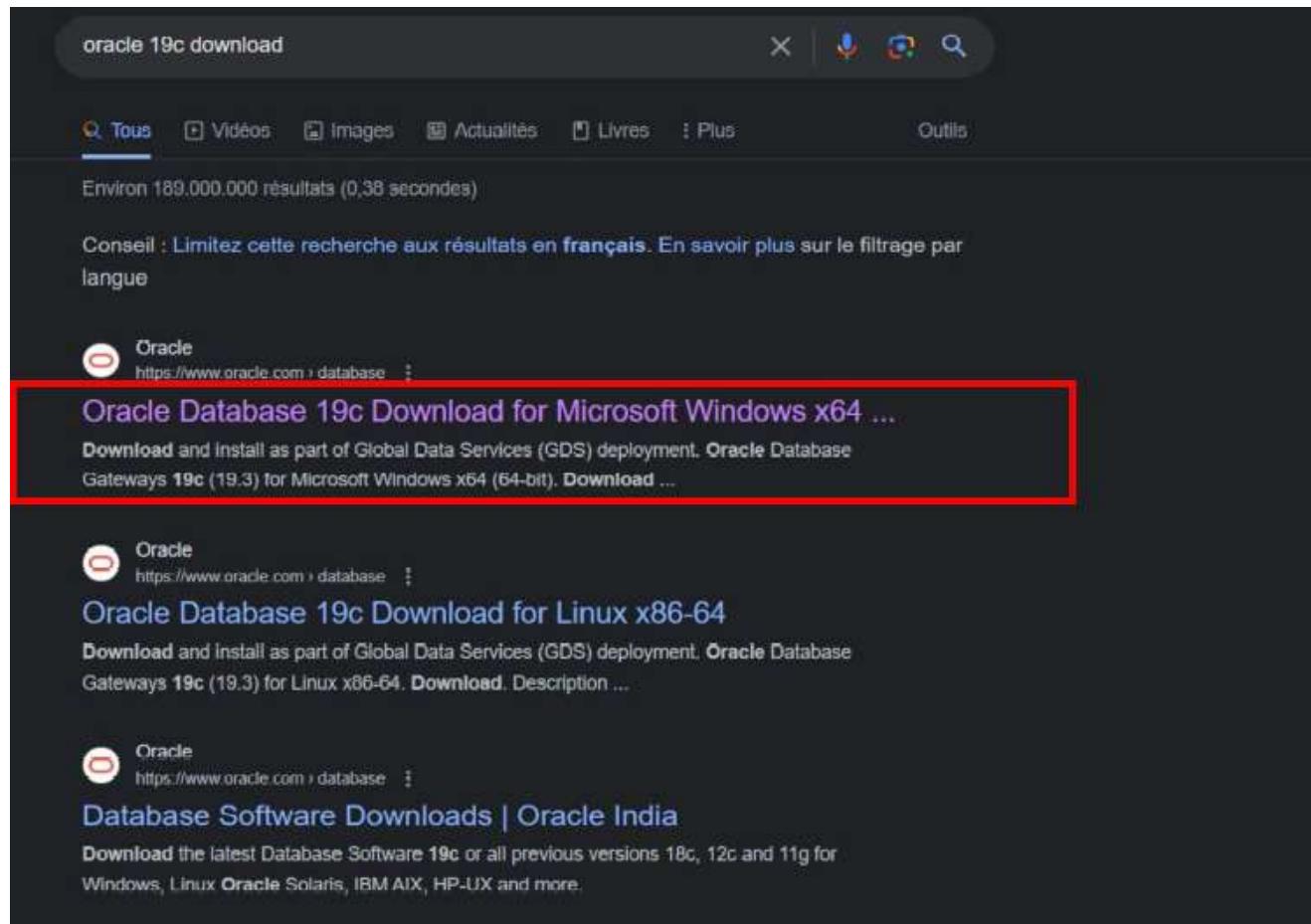
Processeur	Intel x86-64 compatible ou AMD64
RAM	Minimum 2 Go (recommandé 4 Go ou plus pour de meilleures performances)
Espace disque	Minimum 10 Go pour installation complète + espace pour bases de données
Résolution écran	1024x768 minimum
Système de fichiers	NTFS recommandé

INSTALLATION ORACLE DATABASE 19C

Delete D:\app\{user}\product\11.2.0\dbhome_1 and all Oracle's files from D:\app\

- If you can not delete these files, need to change the owner or add permissions
 - Properties>Security> Advanced> Change Owner> Advanced>Find now> Choose Users
 - Sometime you need to add permissions to Users: Properties>Security>Edit> Add Users> Allow ...
- Delete C:\Program Files (x86)\Oracle, C:\Oracle
- Delete C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Oracle - OraDb11g_home1
- Run regedit.exe>
 - delete HKEY_LOCAL_MACHINE\SOFTWARE\Oracle
 - delete HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\ORACLE
- Delete HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Ora*
- Start Run>Computer Management> System Tools>Local Users and Groups> Delete all Oracle's users and groups
- Restart windows.

INSTALLATION ORACLE DATABASE 19C



INSTALLATION ORACLE DATABASE 19C

The screenshot shows the Oracle Database 19c Download for Microsoft Windows x64 (64-bit) page. A modal dialog box is displayed, requiring the user to accept the Oracle License Agreement to proceed. The dialog contains the following text and form fields:

You must accept the [Oracle License Agreement](#) to download this software.

I reviewed and accept the Oracle License Agreement

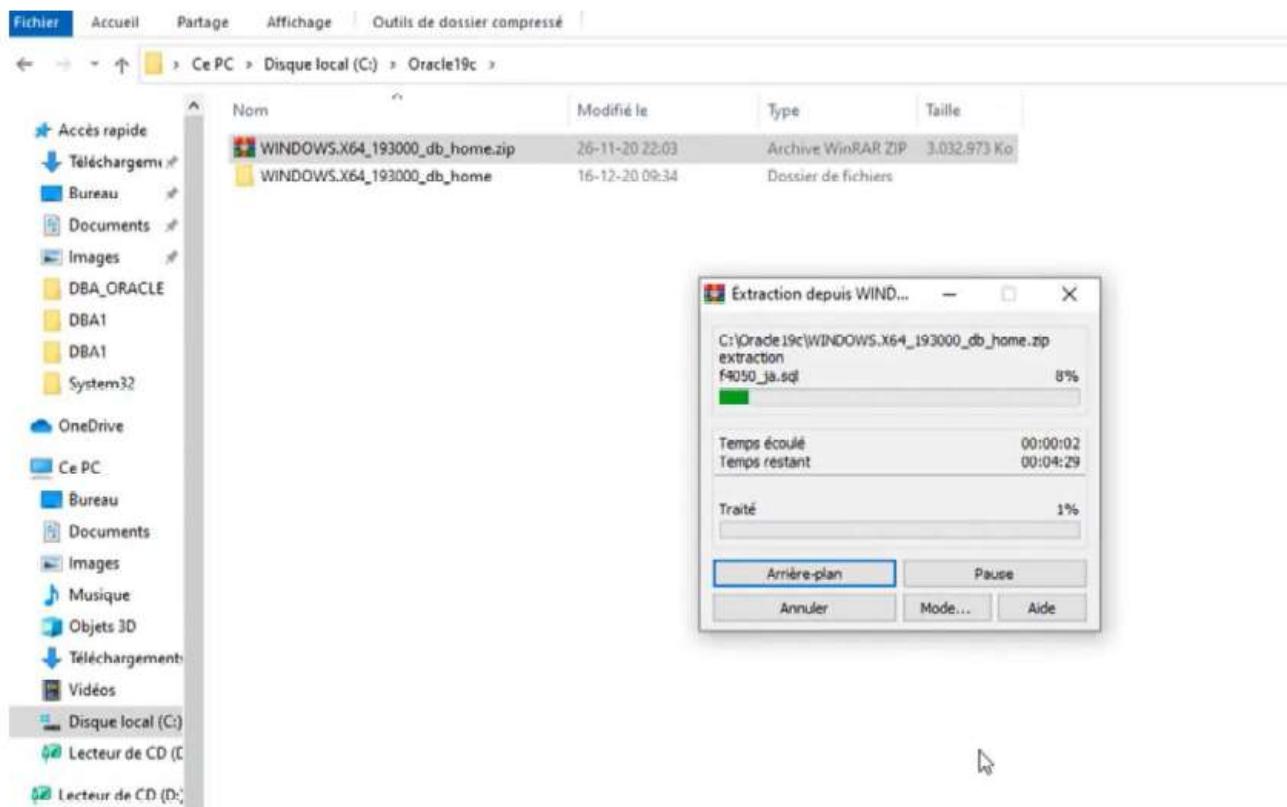
Required

You will be redirected to the login screen in order to download the file.

[Download WINDOWS.X64_193000_db_home.zip](#)

The 'Download' button and the download link are highlighted with a red box.

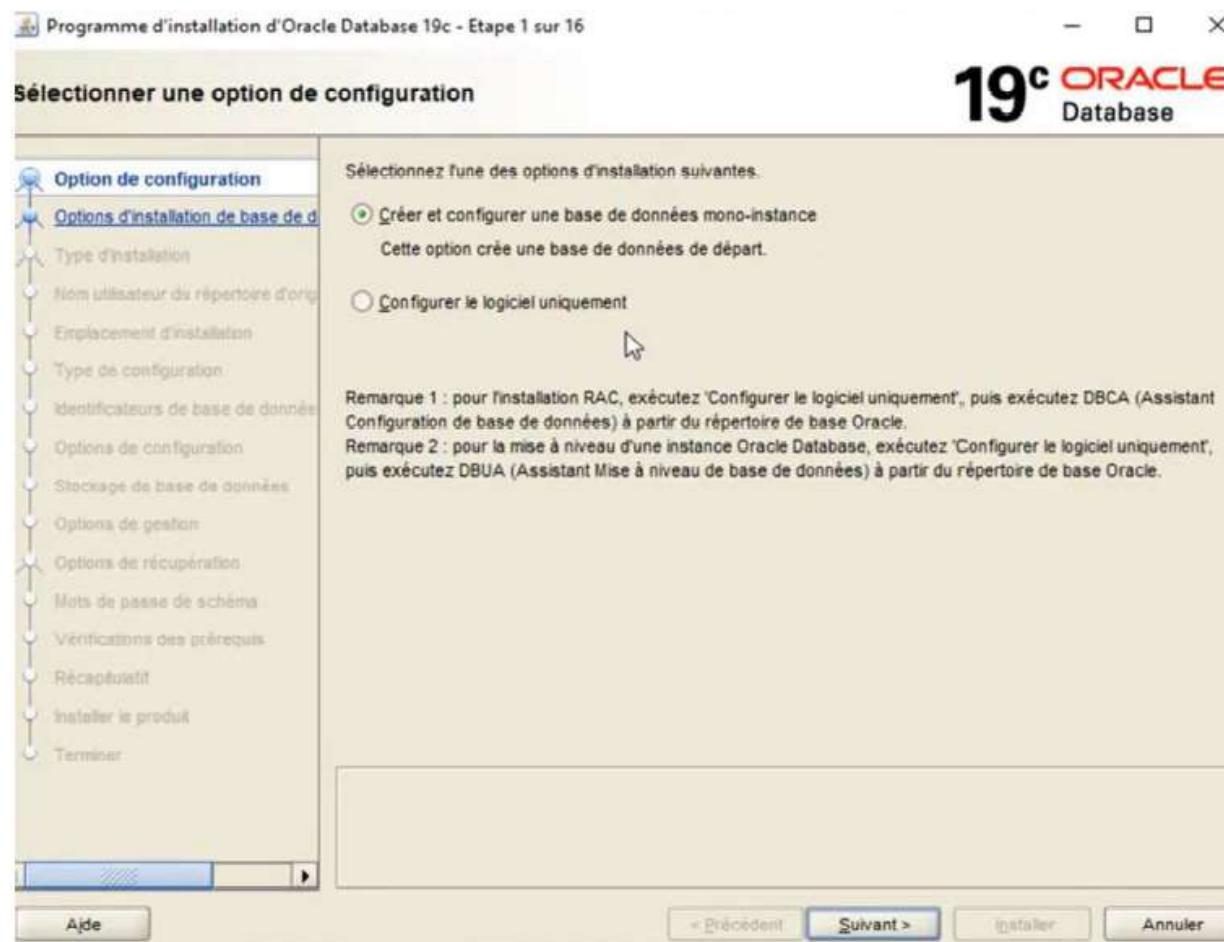
INSTALLATION ORACLE DATABASE 19C



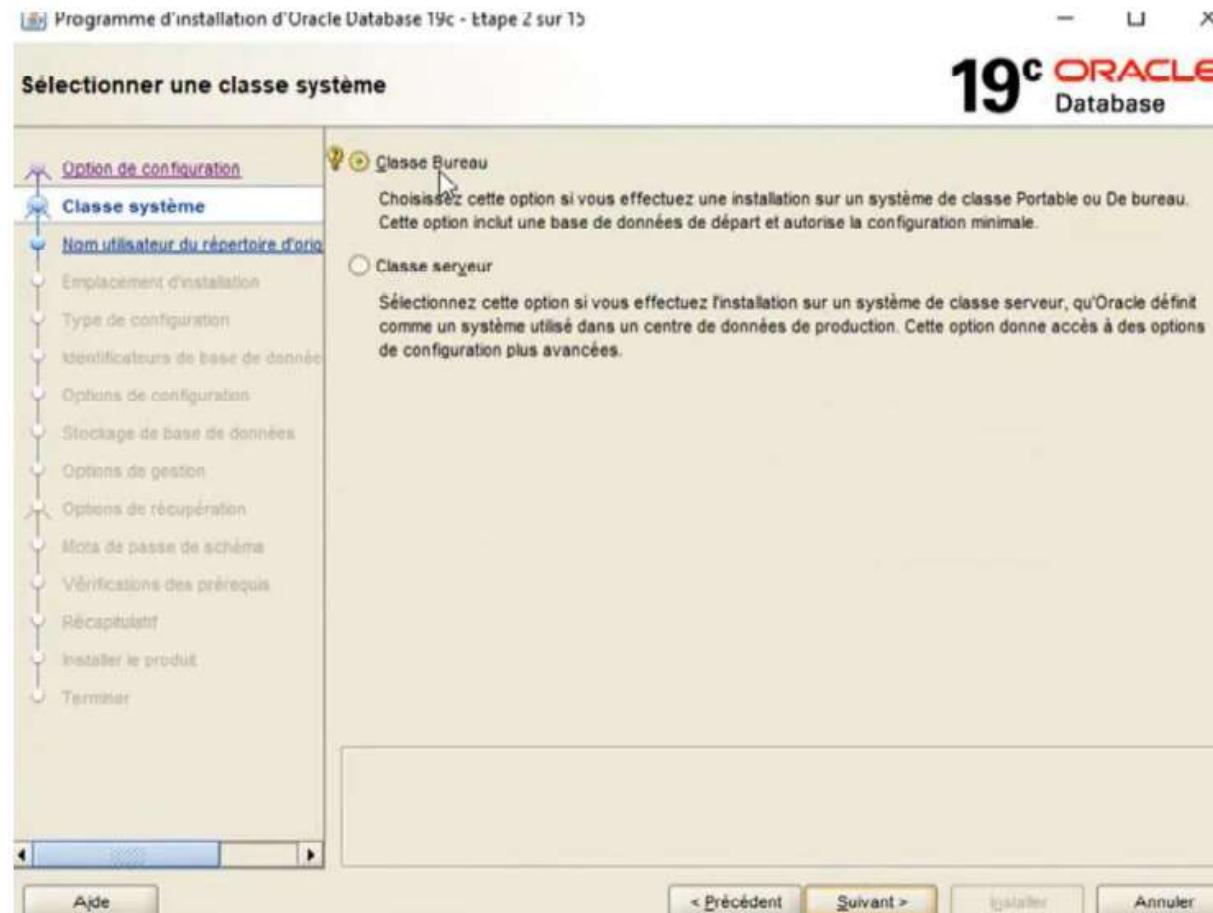
INSTALLATION ORACLE DATABASE 19C

	Nom	Modifié le
	ord	30-05-19 17:03
	ords	30-05-19 17:03
	oui	30-05-19 17:04
	owm	30-05-19 17:03
	perl	30-05-19 17:03
	plsql	30-05-19 17:03
	precomp	30-05-19 17:03
	QOpatch	30-05-19 17:03
	R	30-05-19 17:03
	racg	30-05-19 17:03
	rdbms	30-05-19 17:03
	relnotes	30-05-19 17:03
	slax	30-05-19 17:03
	sqldeveloper	30-05-19 17:04
	sqlj	30-05-19 17:03
	sqlpatch	30-05-19 17:05
	sqlplus	30-05-19 17:03
	srvm	30-05-19 17:03
	suptools	30-05-19 17:03
	ucp	30-05-19 17:03
	usm	30-05-19 17:03
	utl	30-05-19 17:04
	wwg	30-05-19 17:03
	xdk	30-05-19 17:03
	env.ora	26-08-15 04:26
	schagent.c	14-10-16 12:50
	setup.bat	18-09-18 20:05
	setup.exe	14-11-18 16:42

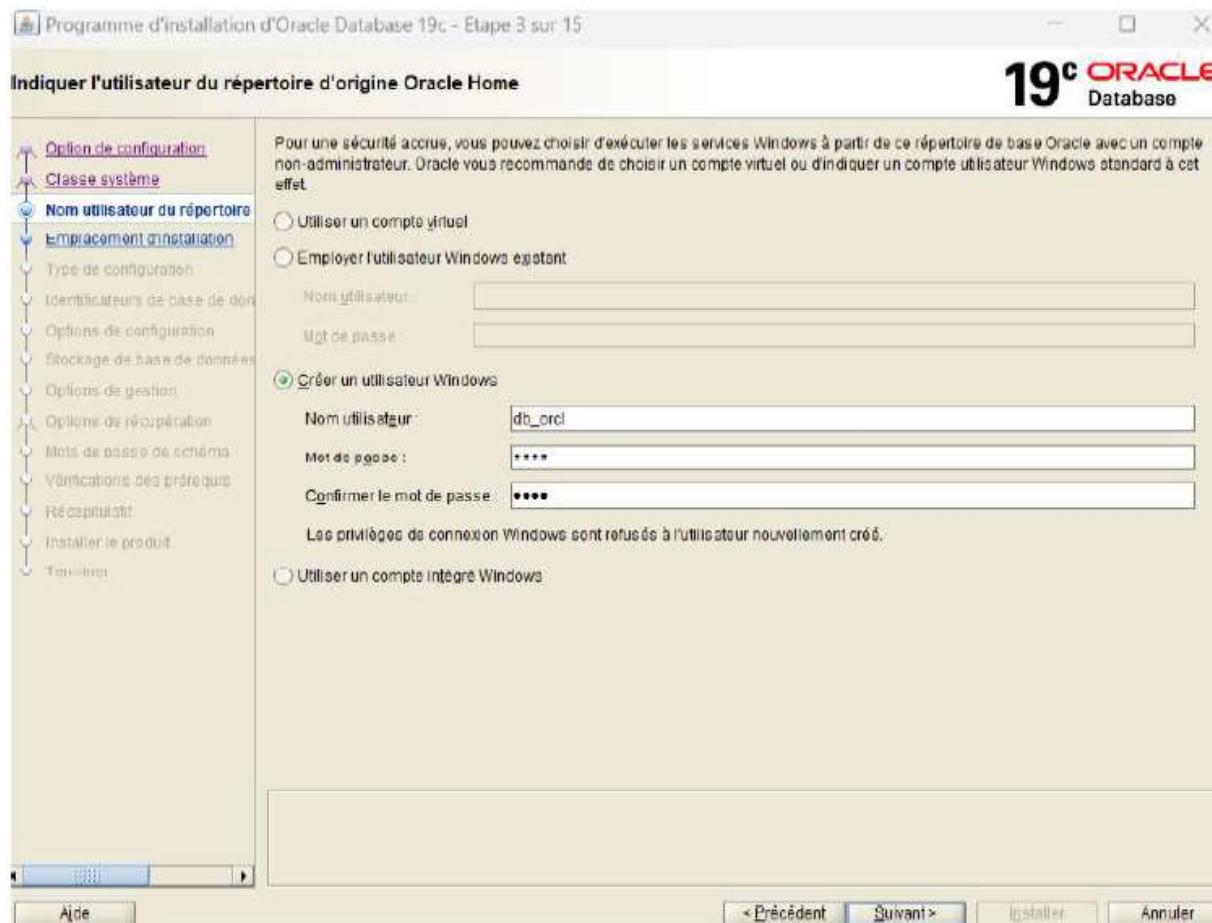
INSTALLATION ORACLE DATABASE 19C



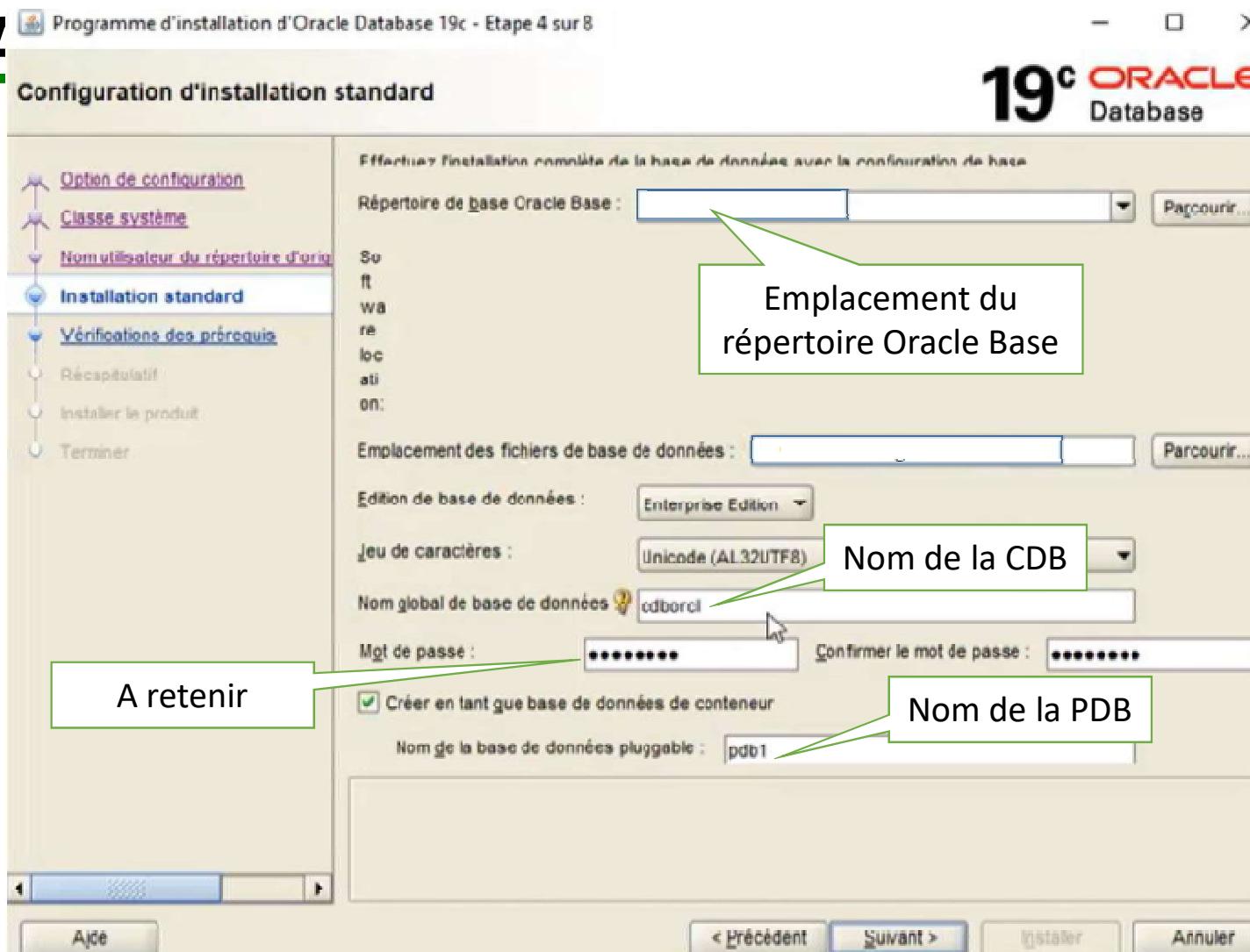
INSTALLATION ORACLE DATABASE 19C



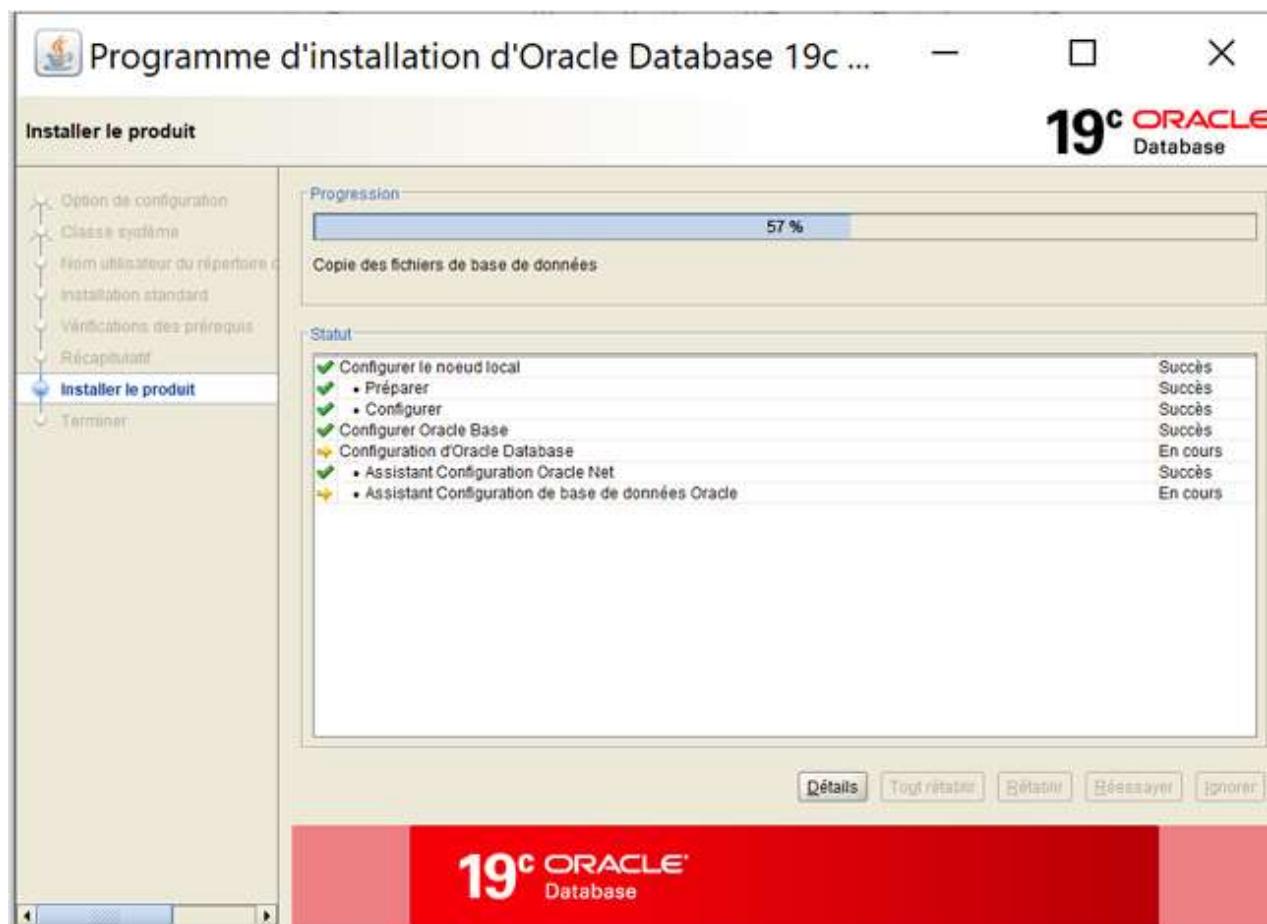
INSTALLATION ORACLE DATABASE 19C



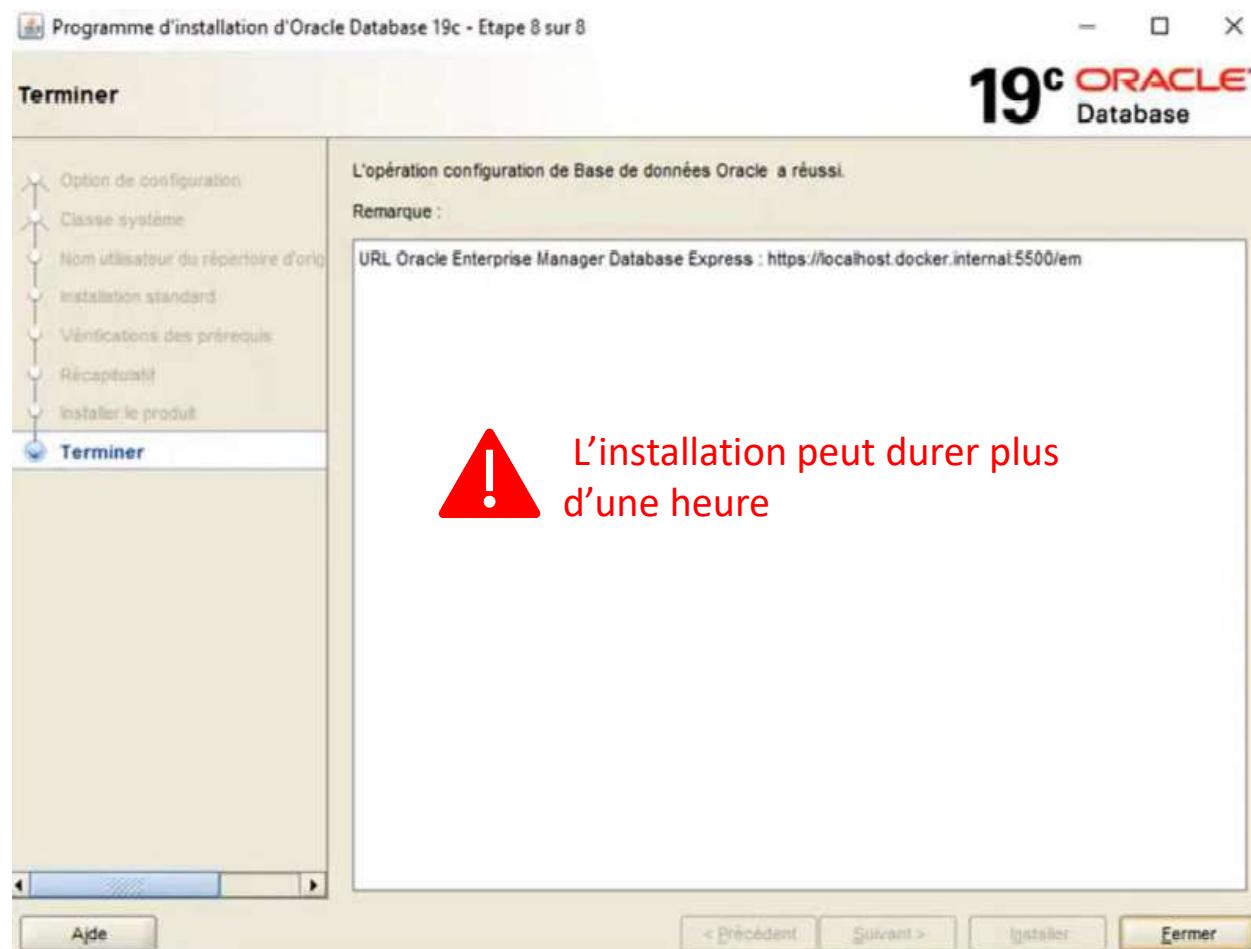
INSTALLATION



INSTALLATION ORACLE DATABASE 19C

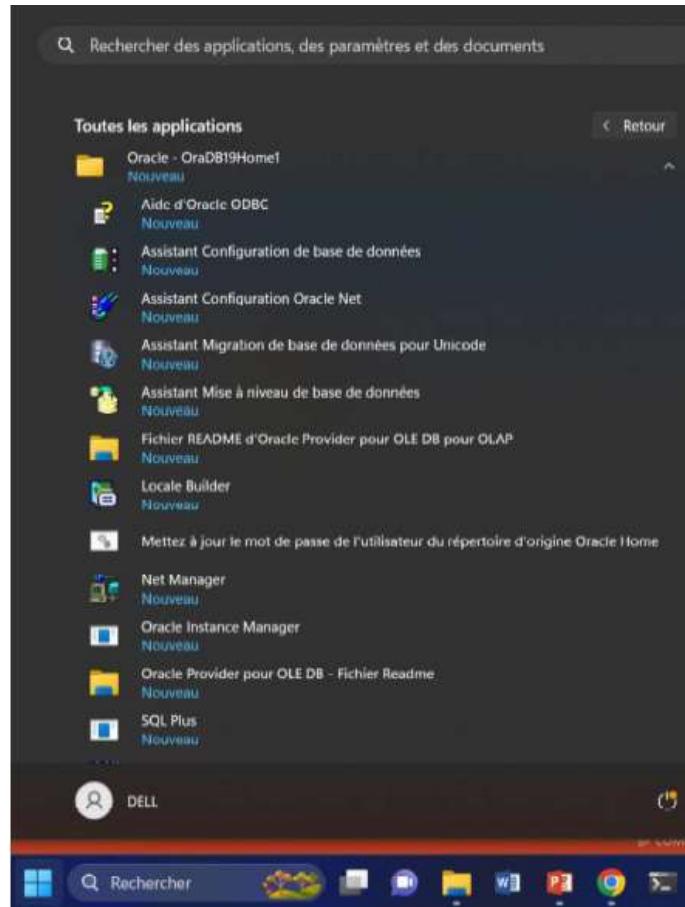


INSTALLATION ORACLE DATABASE 19C



INSTALLATION ORACLE DATABASE 19C

Après installation, les différents outils Oracle Database installés par défaut sont disponibles au niveau du menu de démarrage.





Installation Oracle Database 11g



Formatage des données et des rapports sous SQL Plus

Objectif: Savoir améliorer l'affichage et l'exploitation des résultats des requêtes sous SQL* Plus pour produire un rapport final

INTRODUCTION AU COMMENDES DE FORMATAGE

Objectifs pédagogiques

- Utiliser les **commandes de SQL*Plus** pour formater les résultats des requêtes.
- Améliorer la **lisibilité** des affichages (colonnes, lignes, nombres, titres, etc.).
- Enregistrer et exécuter des **scripts SQL** correctement formatés.

INTRODUCTION AU COMMANDES DE FORMATAGE

1. FORMATAGE DES ENTÊTES

La commande **COLUMN** sert à modifier les entêtes des colonnes et leur présentation dans l'output d'une requête SQL*PLUS. Elle permet de:

1. Changer les noms de colonnes dans l'entête
1. Afficher les noms des colonnes sur plusieurs lignes
1. Changer le caractère de soulignement

INTRODUCTION AU COMMANDES DE FORMATAGE

1. FORMATAGE DES ENTÊTES

Changer les noms de colonnes dans l'entête:

L'entête du résultat d'une requête SQL*Plus reprend les expressions ou les noms des colonnes telles que définies en base de données. La commande COLUMN permet de changer cet entête pour mettre des noms plus significatifs.

Syntaxe : **COLUMN** column_name **HEADING** column_heading

Exemple : TP1 Exercice 3.1.

INTRODUCTION AU COMMANDES DE FORMATAGE

1. FORMATAGE DES ENTÊTES

Afficher les noms des colonnes sur plusieurs lignes:

Il est possible d'afficher le nouvel entête d'une colonne sur deux ou plusieurs lignes

=> utiliser la commande COLUMN avec le séparateur « | » à l'emplacement où commence la nouvelle ligne.

Syntaxe : **COLUMN column_name HEADING 'column|heading'**

Exemple : TP1 Exercice 3.1.

INTRODUCTION AU COMMANDES DE FORMATAGE

1. FORMATAGE DES ENTÊTES

Changer le caractère de soulignement:

Le caractère de soulignement (-) utilisé par défaut pour la mise en page du résultat d'une requête SQL*Plus peut être modifié.

Syntaxe : **SET UNDERLINE underline**

Exemple : TP1 Exercice 3.1.

INTRODUCTION AU COMMENDES DE FORMATAGE

2. FORMATAGE DES COLONNES

La commande COLUMN permet également le formatage des données des colonnes, en utilisant un modèle de format.

Syntaxe : **COLUMN column_name FORMAT model**

INTRODUCTION AU COMMANDES DE FORMATAGE

FORMATAGE DES COLONNES TYPE CARACTÈRE

La largeur d'une colonne de type caractère peut être modifiée selon la syntaxe suivante.

Syntaxe : **COLUMN** column_name **FORMAT** An, avec n le nombre de caractères à afficher.

Exemple : TP1 Exercice 3.2.

INTRODUCTION AU COMMENDES DE FORMATAGE

FORMATAGE DES COLONNES TYPE NUMÉRIQUE

Par défaut dans Oracle, la largeur d'affichage d'une colonne numérique = Max(largeur entête; largeur FORMAT + 1 pour le signe).

La largeur et le format d'affichage d'une colonne de type numérique peuvent être modifiés par les commandes: **SET NUMWIDTH**, **SET NUMFORMAT** ou **COLUMN** accompagnée de l'option **FORMAT**.

Le format d'un numérique peut être changé par le paramétrage SET NUMFORMAT ou COLUMN en ajoutant des signes (\$; +; - ...), des séparateurs (point, virgule), des espaces...

INTRODUCTION AU COMMANDES DE FORMATAGE

FORMATAGE DES COLONNES TYPE NUMÉRIQUE

NUMWIDTH contrôle la largeur par défaut utilisée lors de l'affichage de valeurs numériques. Valeur par défaut = 10.

Set **NUMWIDTH** width

NUMFORMAT contrôle le format d'affichage par défaut commande de SELECT.

Set **NUMFORMAT** format (remise à 0: set numformat "")

COLUMN FORMAT permet de spécifier le format d'affichage colonne par colonne

COLUMN column_name **FORMAT** model

COLUMN

Prioritaire sur

NUMFORMAT

Prioritaire sur

NUMWIDTH

Exemple : TP1 Exercice 3.2.

INTRODUCTION AU COMMANDES DE FORMATAGE

FORMATAGE DES COLONNES TYPE NUMÉRIQUE

Exemple de modèle de format

(source documentation Oracle)

Plusieurs éléments peuvent être combinés dans un seul modèle. Exemple:

COLUMN SALARY FORMAT
\$99,990

Element	Examples	Description
,	9,999	Displays a comma in the specified position.
.	99,99	Displays a period (decimal point) to separate the integral and fractional parts of a number.
\$	\$9999	Displays a leading dollar sign.
0	0999 9990	Displays leading zeros Displays trailing zeros.
9	9999	Displays a value with the number of digits specified by the number of 9s. Value has a leading space if positive, a leading minus sign if negative. Blanks are displayed for leading zeroes. A zero (0) is displayed for a value of zero.

INTRODUCTION AU COMMANDES DE FORMATAGE

3. QUELQUES GÉNÉRALITÉS SUR LE FORMTAGE DES DONNÉES

Le formatage reste applicable tant qu'il n'a pas été explicitement changé ou supprimé, ou que la session SQL*Plus n'a pas été fermée.

- Afficher les attributs d'une colonne: **COLUMN** column_name
- Supprimer le formatage d'une colonne: **COLUMN** column_name **CLEAR**
- Supprimer le formatage de toutes les colonnes: **CLEAR COLUMNS**
- Revenir au format par défaut d'une colonne, sans supprimer les paramètres:
COLUMN column_name **OFF**
- Réappliquer les paramètres précédemment définis: **COLUMN** column_name
ON

INTRODUCTION AU COMMENDES DE FORMATAGE

4. COMMANDE WRAP

Par défaut, SQL*Plus opère un retour à la ligne lorsque la largeur de la colonne ne suffit pas pour afficher la valeur en entier.

Il est possible d'ajouter un “spérateur d'enregistrements”, soit après les enregistrements où le retour à la ligne a été opéré (paramètre **RECSEP WRAPPED**) ou après chaque enregistrement (paramètre **RECSEP EACH**)

WORD_WRAPPED: Pour faire un retour à la ligne sans tronquer les mots.

Commande	Rôle
SET RECSEP WRAPPED	Ajoute une ligne de séparation seulement après les lignes où un retour a été fait
SET RECSEP EACH	Ajoute une ligne de séparation après chaque enregistrement
SET RECSEPCHAR <caractère>	Définit le caractère utilisé pour séparer les lignes
SET RECSEP OFF	Désactive la ligne de séparation
COLUMN col_name WORD_WRAPPED	Retour à la ligne sans couper les mots

SET RECSEP OFF pour arrêter la séparation

INTRODUCTION AU COMMANDES DE FORMATAGE

4. COMMANDE BREAK

La commande BREAK permet de supprimer les valeurs en double sur la colonne spécifiée.

Syntaxe : **BREAK ON** column_name

BREAK (sans paramètre) Afficher les colonnes pour lesquelles un Break a été défini

CLEAR BREAKS Supprimer BREAK

Il est possible d'insérer des lignes vides ou une nouvelle page lorsque la valeur de la colonne sur laquelle le break a été opéré change.

Ligne(s) vide(s): **BREAK ON** break_column **SKIP** n

Nouvelle page: **BREAK ON** break_column **SKIP PAGE**

Toute nouvelle commande BREAK annule la précédente.

INTRODUCTION AU COMMENDES DE FORMATAGE

4. COMMANDE COMPUTE

La commande COMPUTE permet d'effectuer des calculs sur un groupe de données.

Elle ne peut être effectuée sans la commande BREAK.

S'exécute généralement sur des colonnes de type numérique, mais peut dans certains cas s'appliquer à d'autres types.

Syntaxe : **COMPUTE** fonction [LABEL] text **OF** column1 column2 ... **ON** **break_column** [ON REPORT]

Avec fonction une des fonction prédéfinies telle que: sum, avg, min, max...

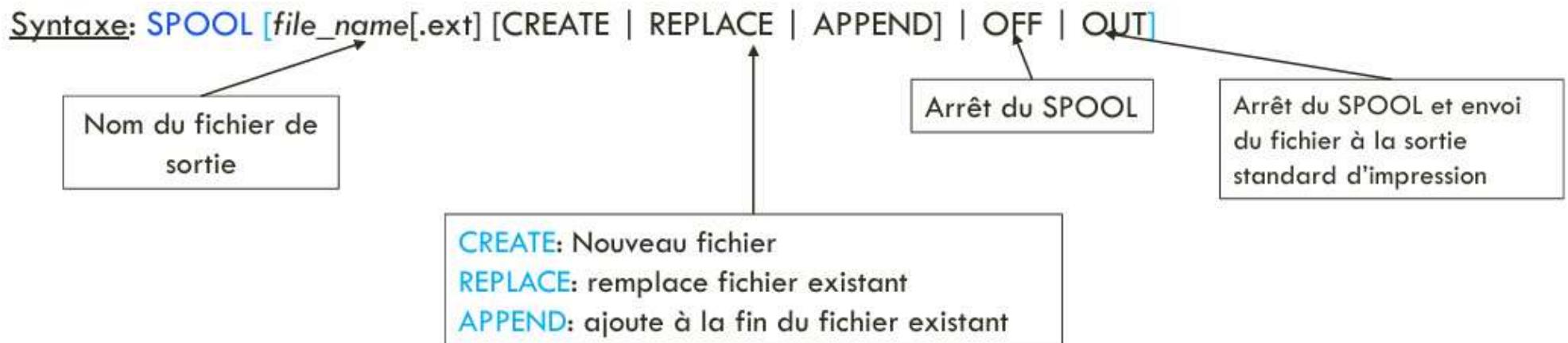
ONREPORT: pour appliquer la fonction à la fin du rapport.

CLEAR COMPUTES pour effacer.

INTRODUCTION AU COMMUNES DE FORMATAGE

4. COMMANDE SPOOL

La commande SPOOL permet de stocker le résultat de la requête dans un fichier.



Chapitre 1: Introduction

Chapitre 2: Installation Oracle Database 19c et Formatage des données

Chapitre 3: Architecture

Chapitre 4: Instance Oracle

Chapitre 5: Gestion des paramètres et des fichiers de paramètre

Chapitre 6: Gestion des Fichiers de journalisation

Chapitre 7: Archivelog

Chapitre 8: Gestion des fichiers de contrôle

Chapitre 9: Tablespaces

ARCHITECTURE GLOBALE

1.1. CONNEXION AU SERVEUR ORACLE

Il est possible de se connecter à un serveur Oracle suivant l'une des architectures ci-dessous:

1. Connexion locale: L'utilisateur est directement connecté sur la machine locale faisant office de Serveur Oracle.

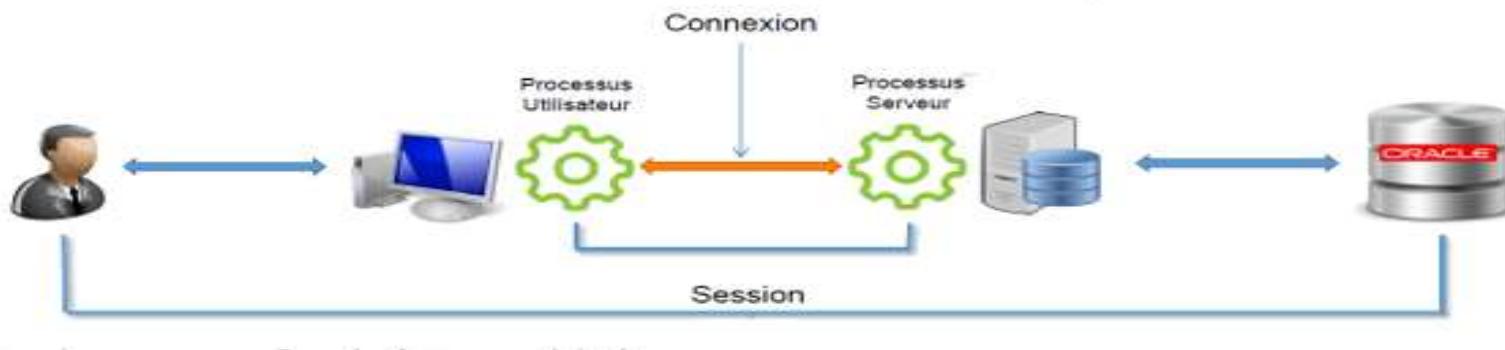
1. Connexion client/serveur: L'utilisateur se connecte à partir d'une machine directement connectée à un Serveur Oracle.

1. Connexion multi-tiers: La connexion entre la machine cliente et le serveur Oracle se fait à travers un autre serveur applicatif (Par exemple un Serveur Web).



ARCHITECTURE GLOBALE

1.1. CONNEXION AU SERVEUR ORACLE



La connexion à un serveur Oracle fait appel à deux processus:

1- Processus utilisateur (User process):

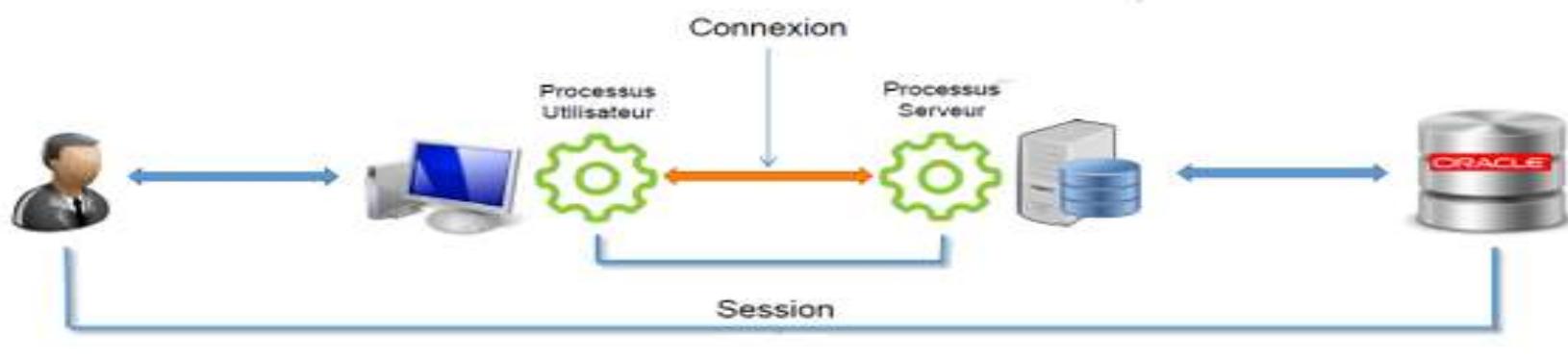
- Mécanisme exécutant le code d'un outil ou application qui se connecte à Oracle.
- Client local ou distant se connectant via réseau à la base.

2- Processus serveur (Server process):

- Permet l'interaction entre le processus utilisateur et le serveur Oracle.
- Effectue l'analyse (parse) et l'exécution des instructions SQL émises via l'application, et renvoie les résultats.

ARCHITECTURE GLOBALE

1.1. CONNEXION AU SERVEUR ORACLE



Connexion: Le lien entre le processus utilisateur et le processus serveur.

Session:

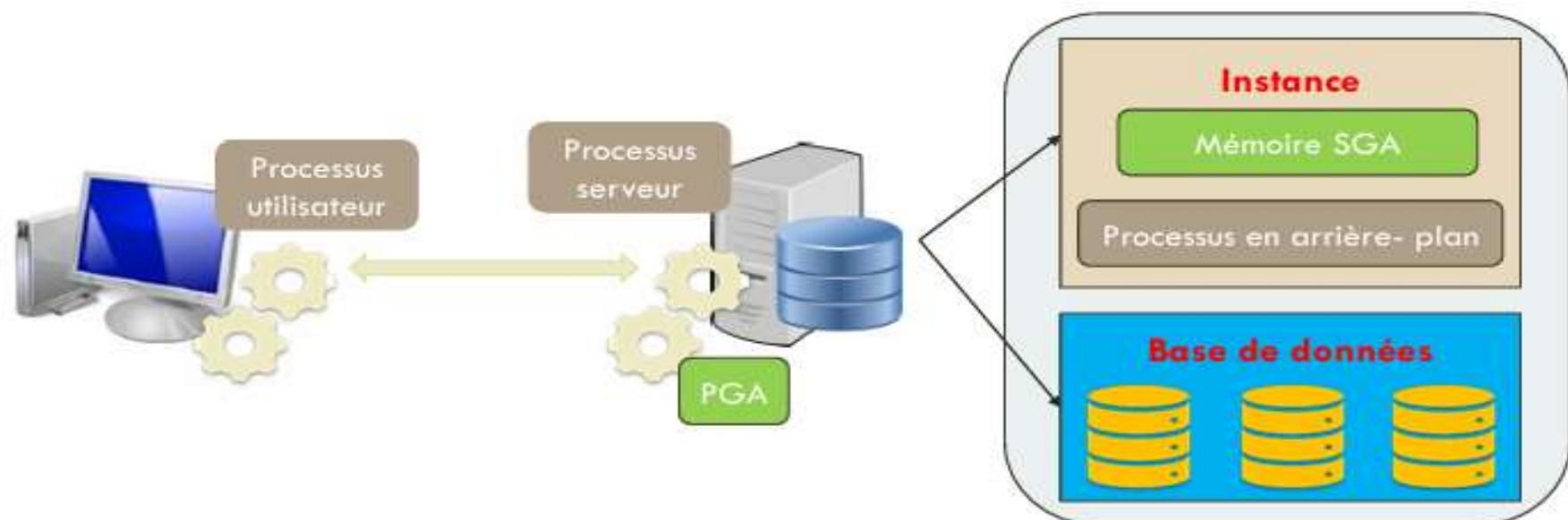
- o Connexion spécifique entre un utilisateur et le serveur Oracle.
- o Démarré lorsqu'une connexion utilisateur est validée par le serveur Oracle.
- o Se termine lorsqu'un utilisateur se déconnecte ou si fin prématurée de la connexion.

ARCHITECTURE GLOBALE

1.2. ARCHITECTURE DU SERVEUR DE BASE DE DONNÉS ORACLE

Le serveur de base de données Oracle comporte deux composants:

- o **Instance**: Se compose d'une mémoire SGA, et de processus d'arrière plan.
- o **Base de données**: un ensemble de fichiers, situés sur le disque, qui stockent des données.

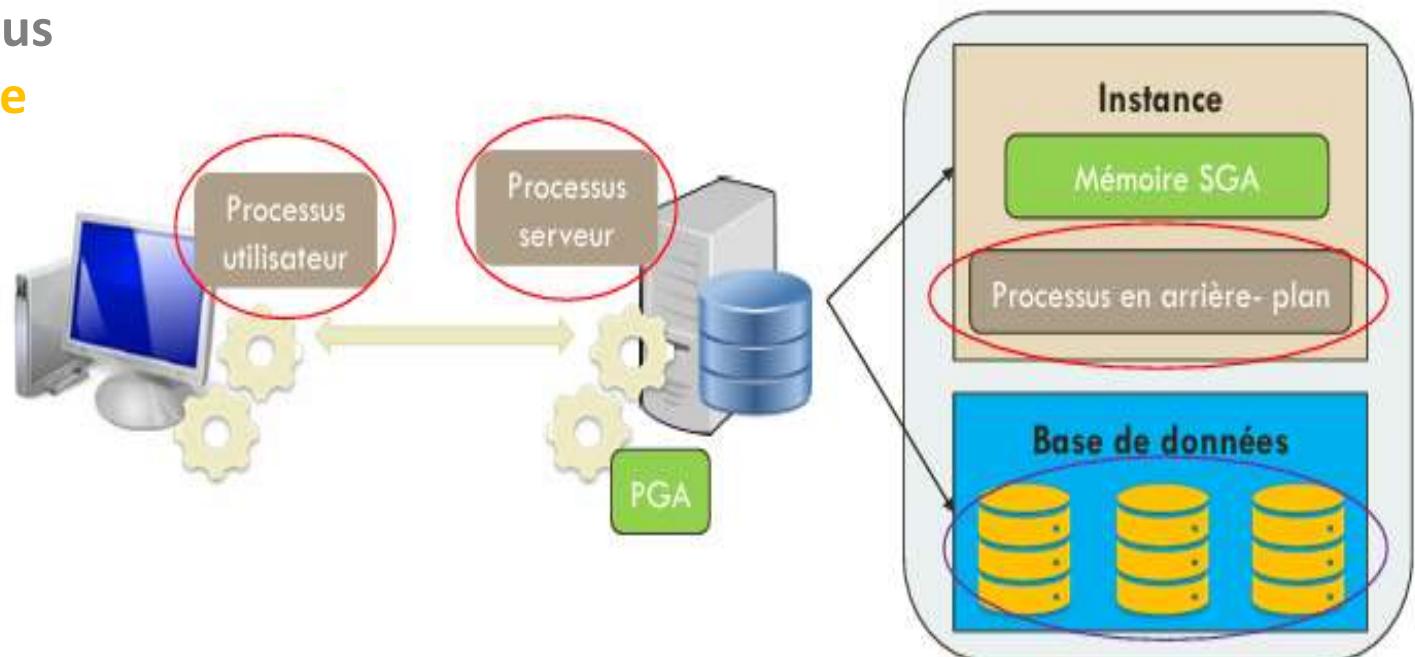


ARCHITECTURE GLOBALE

1.2. ARCHITECTURE DU SERVEUR DE BASE DE DONNÉS ORACLE

Le serveur de base de données Oracle comporte trois types de structure

- o Structures mémoire
- o Structures de processus
- o Structures de stockage



ARCHITECTURE GLOBALE

1.2. ARCHITECTURE DU SERVEUR DE BASE DE DONNÉS ORACLE INSTANCE ORACLE

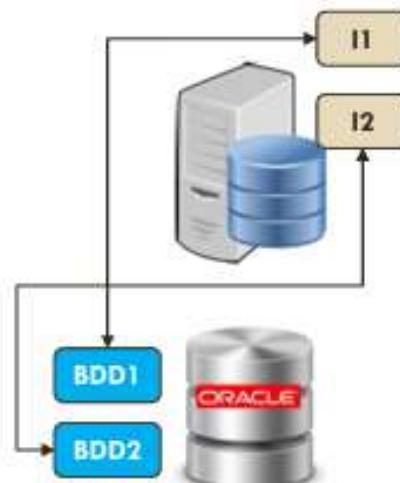
- Une **instance** Oracle est un programme **chargé sur la RAM du serveur**.
- Elle **existe uniquement dans la mémoire**.
- Elle est **créée à chaque fois que la base de donnée démarre**.
- L'instance **disparaît à chaque fois que la base de données redémarre**
- A chaque démarrage d'une instance, deux actions ont lieu :
 - Allocation d'une zone de mémoire partagée appelée **mémoire SGA**.
 - Lancement des **processus en arrière-plan**.



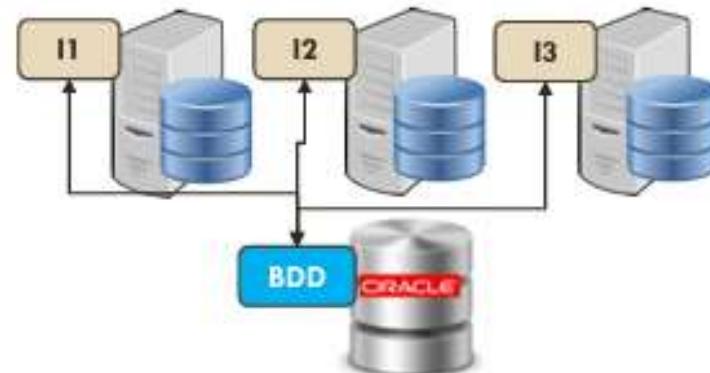
ARCHITECTURE GLOBALE

1.2. ARCHITECTURE DU SERVEUR DE BASE DE DONNÉS ORACLE INSTANCE ORACLE

- Chaque **instance** de base de données est associée à **une seule base**.
- S'il existe plusieurs bases sur le même serveur, il existe une instance distincte pour chacune.



- Une **instance** de base de données **ne peut pas être partagée**.
- Une **base de données** peut être ouverte par **plusieurs instances** situées dans des nœuds distincts (à l'aide de RAC (Real Application Cluster)).



ARCHITECTURE GLOBALE

1.2. ARCHITECTURE DU SERVEUR DE BASE DE DONNÉS ORACLE

STOCKAGE DE BASE DE DONNÉES

Stockage de la base de données Oracle = ensemble de fichiers utilisés pour préserver les données provenant de l'instance Oracle sur un disque physique.



ARCHITECTURE GLOBALE – ORACLE DATABASE 19C

Depuis la version Oracle 12C, la notion de conteneur (container) et d'architecture mutualisée (multitenant architecture) a été introduite.

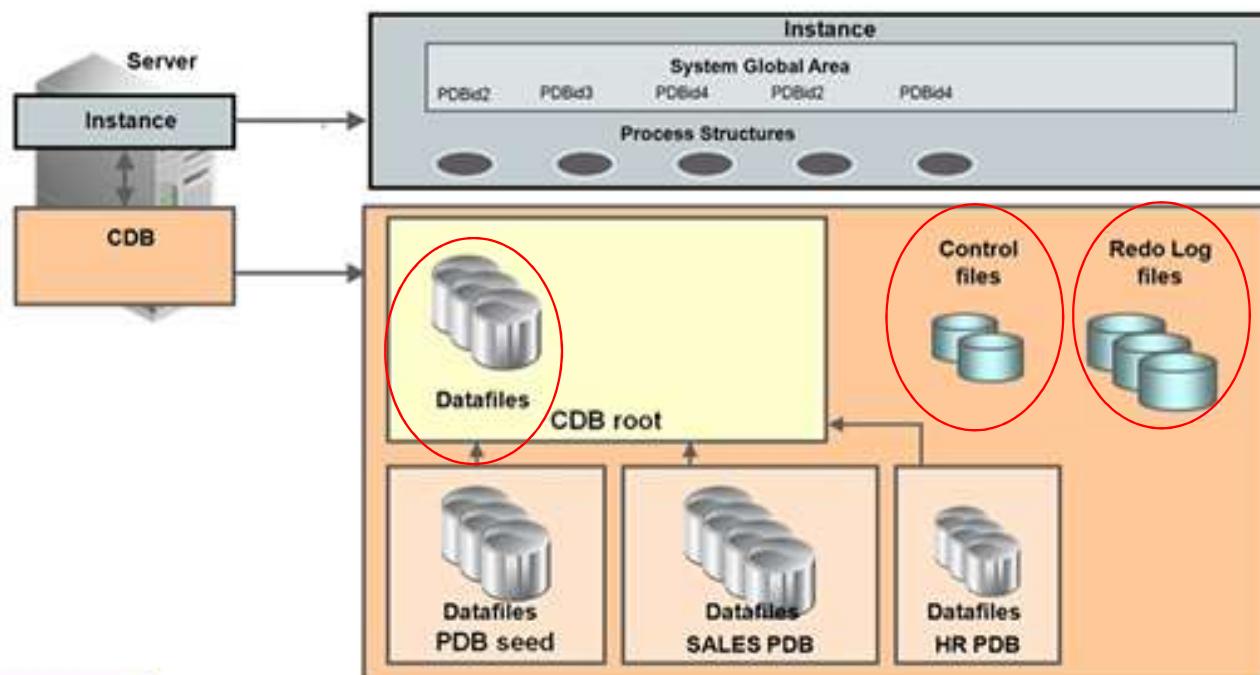
- **Conteneur (CDB):**
 - Collection logique de données et métadonnées dans une architecture mutualisée.
 - Un conteneur de bases de données (CDB) peut contenir 0, 1 ou plusieurs bases de données Pluggable (PDB).
 - Permet de regrouper plusieurs bases de données pour **partager les fonctionnalités** communes.
 - Comporte par défaut 2 bases de données:
 - **CDB\$Root (Root)**: Contient les metadata relatives aux utilisateurs en commun.
1 CDB contient exactement un Root.
 - **PDB\$Seed (Seed)**: Template pour la création d'autres bases de données.
1 CDB contient exactement un Root.

ARCHITECTURE GLOBALE – ORACLE DATABASE 19C

- **Pluggable database (PDB):**
 - o Collection d'objets, schémas, données...
 - o Apparaît aux utilisateurs et applications comme base de données non-CDB.
 - o Peut être « pluggable » dans différents conteneurs.
 - o Peut être générée à partir du Seed.
- **Avantages** de l'architecture mutualisée:
 - o Partage des ressources
 - o Faciliter le déplacement des données et codes d'un conteneur à un autre
 - o Faciliter l'administration de la base de données

ARCHITECTURE GLOBALE – ORACLE DATABASE 19C

Oracle Multitenant Container Database: Architecture



CDB comporte 3 fichiers:

- Les fichiers de données (Datafiles)

- Les fichiers de contrôle (Control Files)

- Les fichiers journaux de reprise (Redo Log Files)

ARCHITECTURE GLOBALE – ORACLE DATABASE 19C

SCHÉMA HR

1. Se connecter à SQL*Plus via invite de commande: >**sqlplus /as sysdba**
2. Afficher les PDBs: **SQL> show pdbs**
3. Afficher le nom du container: **SQL> show con_name**
4. Exécuter la commande suivante. Qu'est-ce que vous remarquez?:
SQL> alter user HR identified by HR account unlock;

ARCHITECTURE GLOBALE – ORACLE DATABASE 19C

SCHÉMA HR

5. Afficher le nom de service **SQL>Select name as "servicename" From v\$active_services Where con_id=3;**
6. Chercher le fichier tnsnames.ora dans %ORACLE_HOME%\network\admin
7. Ajouter/modifier dans le tnsnames.ora

```
PDB1=
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST = localhost) (PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = pdb1)
    )
  )
```

7. Recharger le listener: **SQL> lsnrctl reload**
8. Changer la session: **SQL>alter session set container=pdb1;**
9. Exécuter la commande: **SQL> select name, open_mode from v\$pdbs;**
10. Exécuter la commande: **SQL> alter pluggable database open;**
11. Exécuter la commande suivante. Qu'est-ce que vous remarquez?:
12. **SQL> alter user HR identified by HR account unlock;**

Chapitre 1: Introduction

Chapitre 2: Installation Oracle Database 19c et Formatage des données

Chapitre 3: Architecture

Chapitre 4: Instance Oracle

Chapitre 5: Gestion des paramètres et des fichiers de paramètre

Chapitre 6: Gestion des Fichiers de journalisation

Chapitre 7: Archivelog

Chapitre 8: Gestion des fichiers de contrôle

Chapitre 9: Tablespaces

ARCHITECTURE GLOBALE

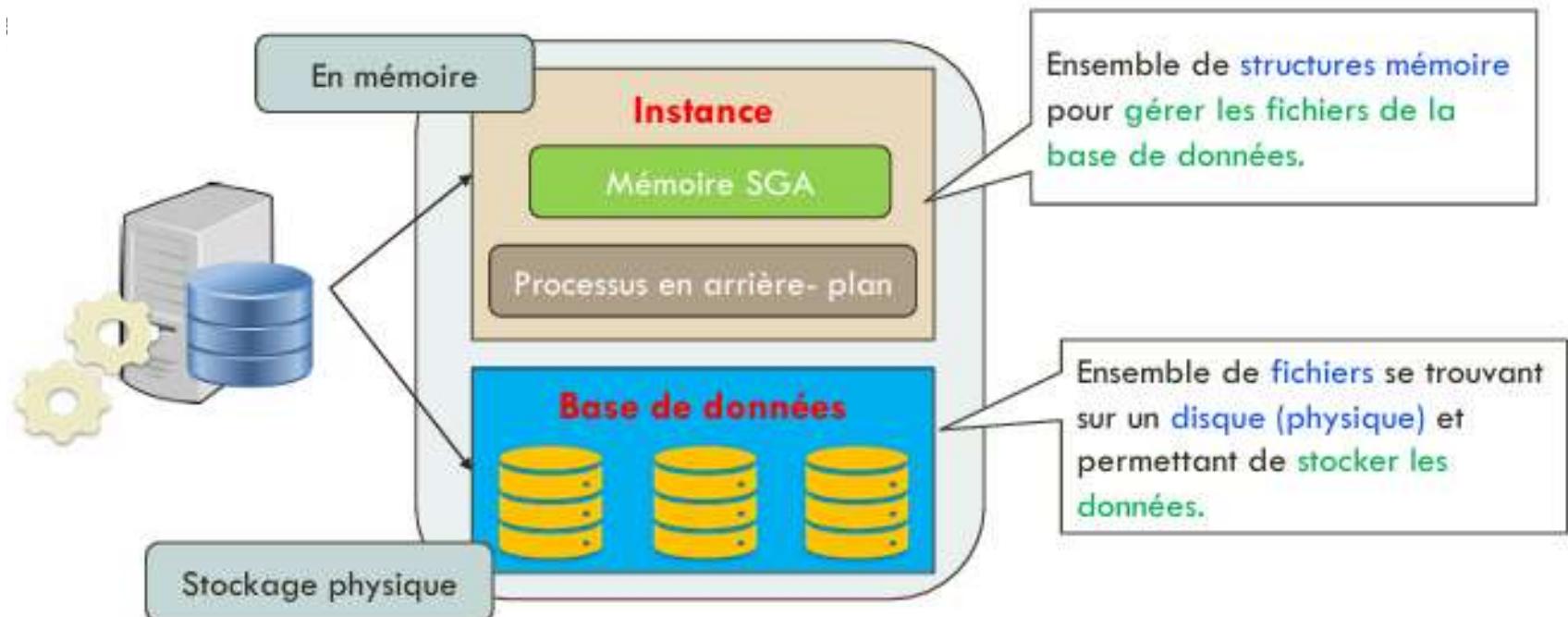
PLAN DE LA SÉANCE

- Les composants de la SGA
- Processus d'arrière-plan
 - Processus Obligatoires
 - Processus Optionnels
- Démarrage et arrêt d'une instance

ARCHITECTURE GLOBALE - RAPPEL

1.2. ARCHITECTURE DU SERVEUR DE BASE DE DONNÉS ORACLE

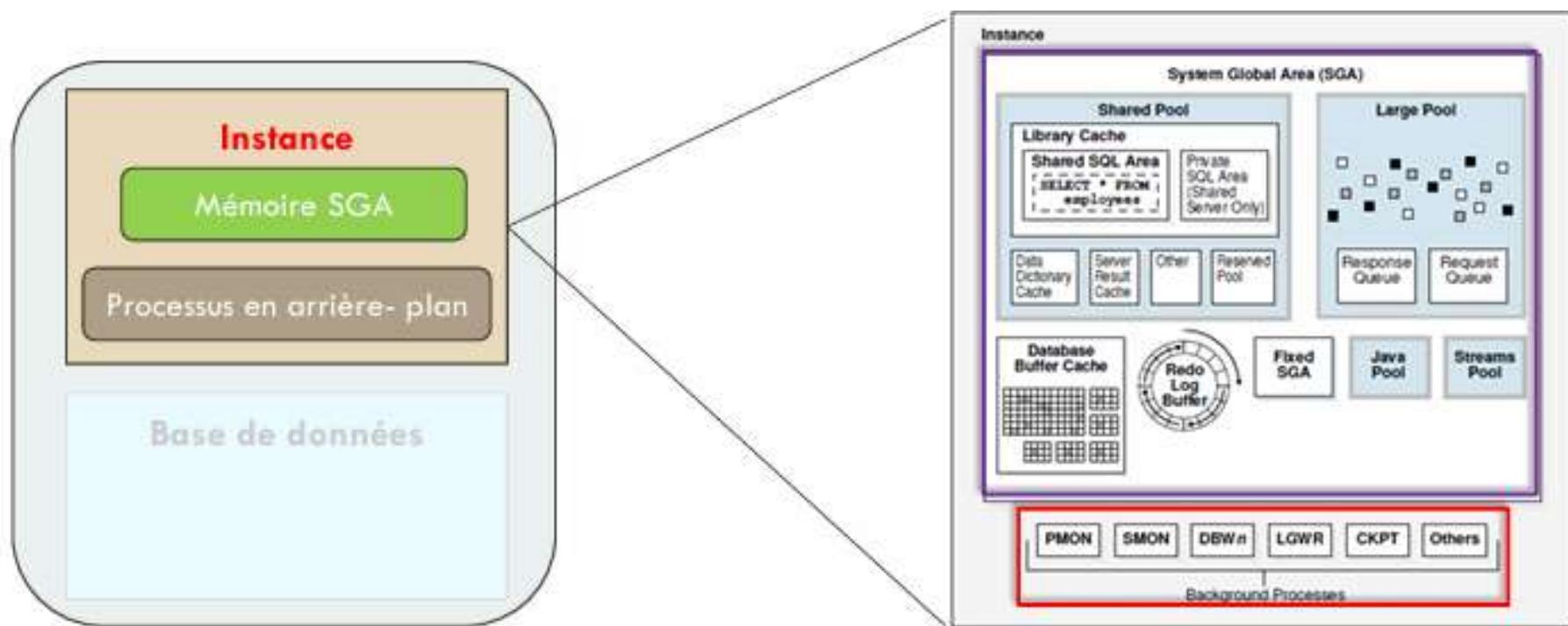
- Le serveur de base de données Oracle est constitué de deux composants:
 - Instance
 - Base de données



ARCHITECTURE GLOBALE- RAPPEL

INSTANCE ORACLE

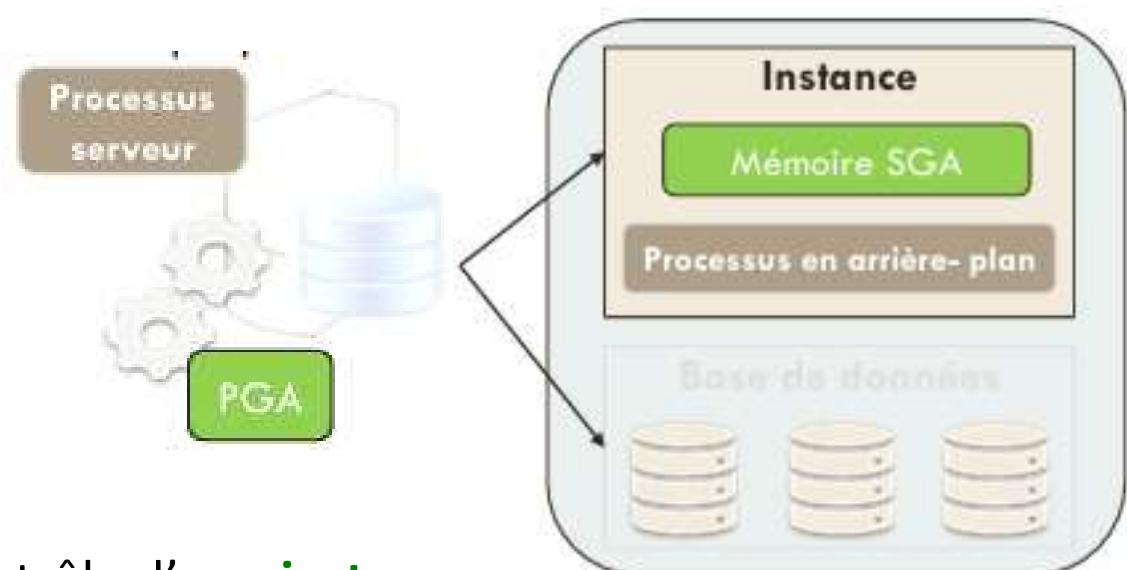
Instance Oracle = Mémoire partagée (SGA) + Processus en arrière plan



INSTANCE ORACLE – MÉMOIRES SGA ET PGA

Mémoire **SGA (System Global Area)** vs mémoire **PGA (Program Global Area)**:

- **SGA**: mémoire **partagée** par les processus serveur et les processus en arrière-plan
- **PGA**: **non partagée**, chaque processus serveur a sa propre mémoire PGA.



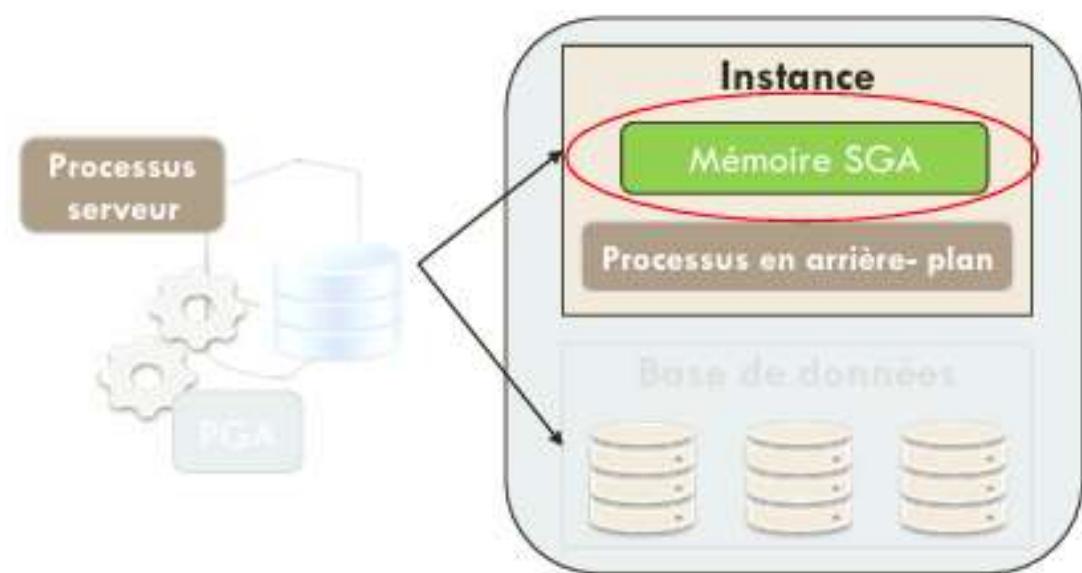
SGA: données et les informations de contrôle d'**une instance**.

PGA: données et informations de contrôle d'un **processus serveur**.

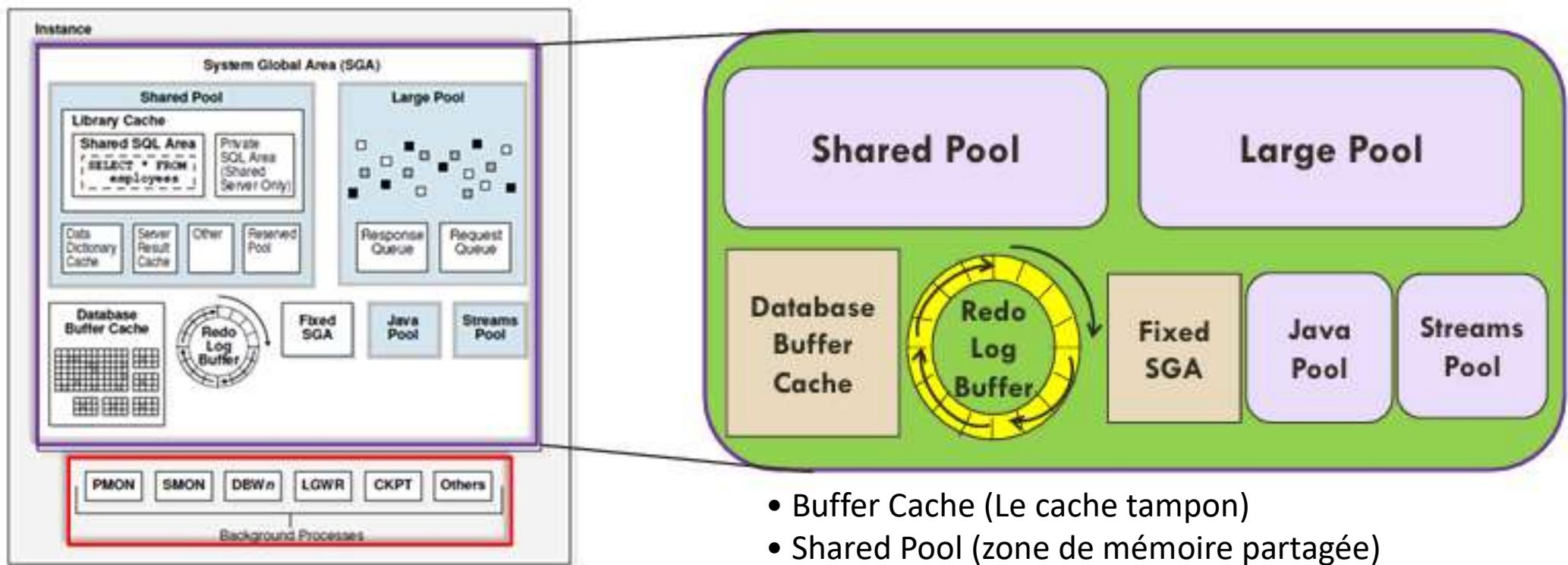
INSTANCE ORACLE – MÉMOIRES SGA

Mémoire SGA (System Global Area):

- Groupe de structures mémoire partagées (appelées composants SGA)
- contiennent les données et les informations de contrôle correspondant à une instance Oracle Database.
- Elle est partagée par les processus serveur et par les processus en arrière-plan.



INSTANCE ORACLE – MÉMOIRES SGA



Les principaux composants de la SGA

- Buffer Cache (Le cache tampon)
- Shared Pool (zone de mémoire partagée)
- Redo Log Buffer (Tampon de journalisation)
- Large Pool (zone de mémoire large)
- Java Pool (zone de mémoire Java)
- Streams Pool (zone de mémoire Streams)

INSTANCE ORACLE – MÉMOIRES SGA

Shared Pool (zone de mémoire partagée)

- Met en cache différents types de données de programme (code SQL, PL/SQL analysé), paramètres système, dictionnaire de données,...
- Impliquée dans (presque) toutes les opérations sur la base de données



Quelques composants:

Library cache (Cache librairie): informations sur les requêtes SQL récemment utilisées, le plan d'exécution de ces requêtes ainsi que l'arbre de parcours et le texte des requêtes (réutilisation des requêtes récemment exécutées, même par d'autres utilisateurs).

Data dictionary cache (Cache du dictionnaire de données): informations sur les objets de la base: structure des colonnes, priviléges utilisateurs,

INSTANCE ORACLE – MÉMOIRES SGA

Database Buffer Cache

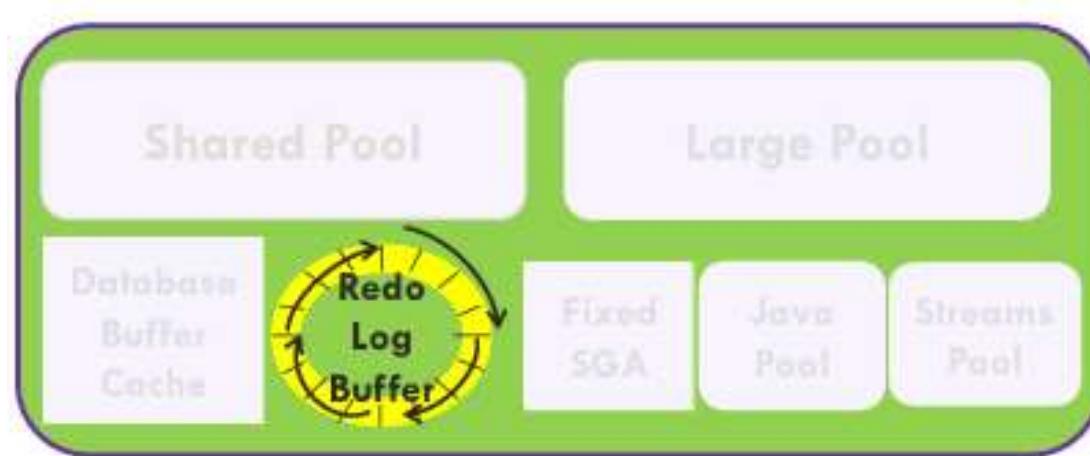
- Stocke des copies des blocs de données lus à partir des fichiers de données.
- Permet d'optimiser les opérations I/O en gardant au niveau du buffer les requêtes les plus fréquentes et au niveau du disque les moins fréquentes.
- La lecture des blocs de données se fait à partir du buffer.



INSTANCE ORACLE – MÉMOIRES SGA

Redo log buffer (tampon de journalisation):

- Mémoire tampon réutilisable
- Comporte les modifications opérées sur la BD par des opérations de manipulation des données (LMD), de définition des données (LD) ou internes.

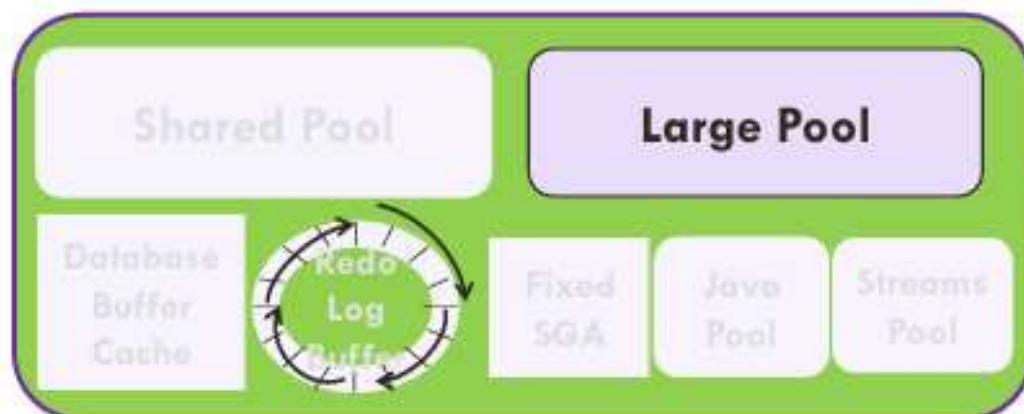


- Les entrées stockées permettent l'annulation des modifications apportées à la base.
- Si modifications dans le **Buffer Cache** par le processus serveur => des entrées de journalisation sont générées et écrites dans le Redo Log Buffer.

INSTANCE ORACLE – MÉMOIRES SGA

Large Pool (zone de mémoire Large Pool)

- Zone mémoire **optionnelle**.
- Permet une allocation mémoire plus large à certains composants tels que:
 - Mémoire UGA pour le serveur partagé et l'interface Oracle
 - Processus serveur d'E/S
 - Opérations de sauvegarde et restauration Oracle



Pour des raisons de performance, Oracle alloue de l'espace mémoire de session dans la mémoire Large Pool pour réserver la zone de mémoire partagée (Shared Pool) à la mise en cache des interrogations SQL partagées.

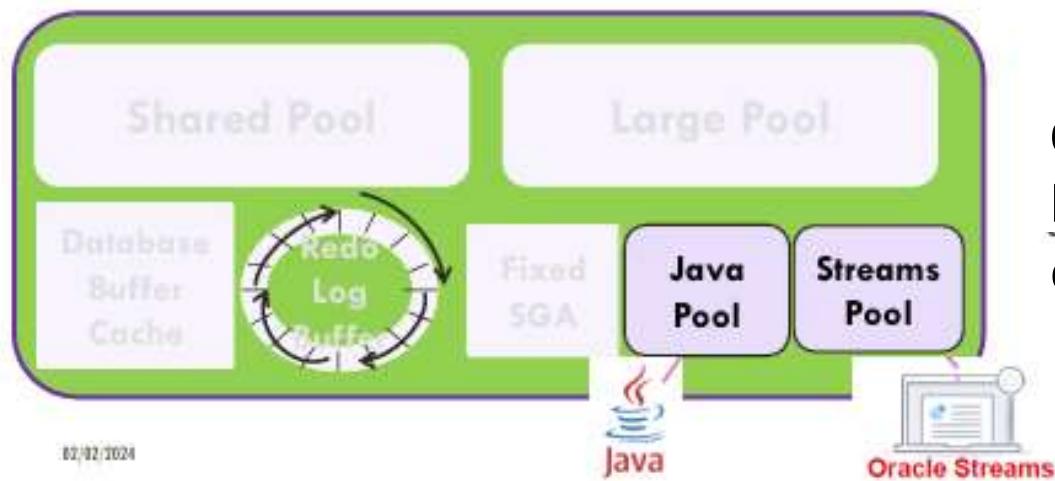
INSTANCE ORACLE – MÉMOIRES SGA

Java Pool (zone de mémoire Java)

Utilisée pour l'ensemble du **code Java** et des données propres à la session dans la JVM (Java Virtual Machine).

Streams Pool (zone de mémoire Streams)

Fournit la mémoire nécessaire aux processus Oracle Streams de capture et d'application des modifications.

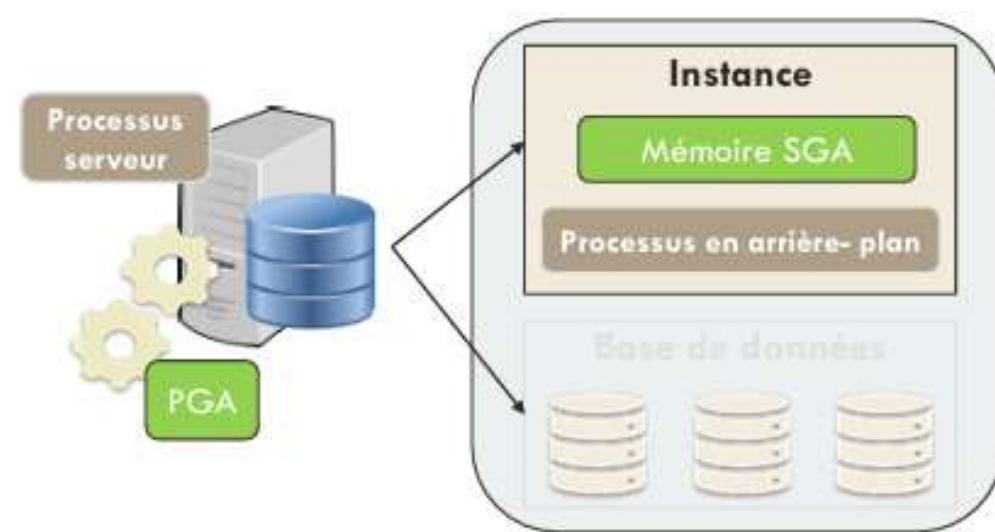


Oracle Streams : Outil qui permet de partager l'information (données) entre les bases de données.

INSTANCE ORACLE – MÉMOIRES PGA

Mémoire PGA (Program Global Area):

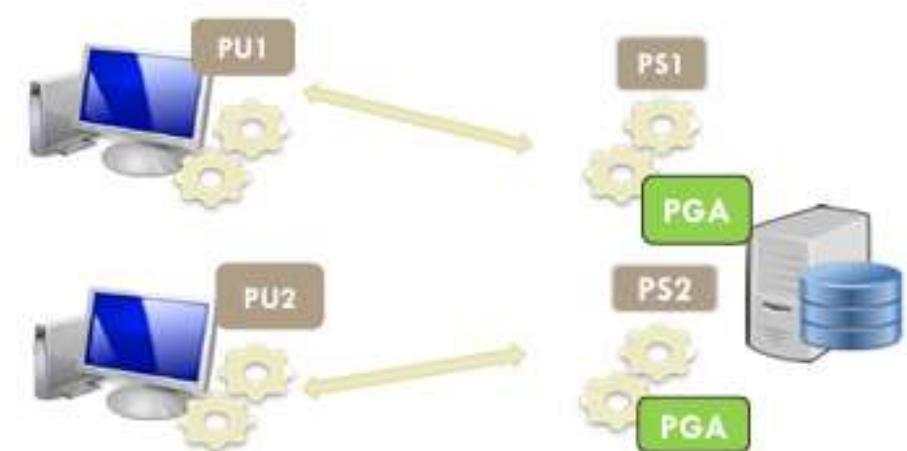
- Zone mémoire contenant les **données et les informations de contrôle** correspondant à **un processus serveur**.
- Chaque processus serveur dispose de sa mémoire PGA privée.



INSTANCE ORACLE – MÉMOIRES PGA

Mode serveur dédié:

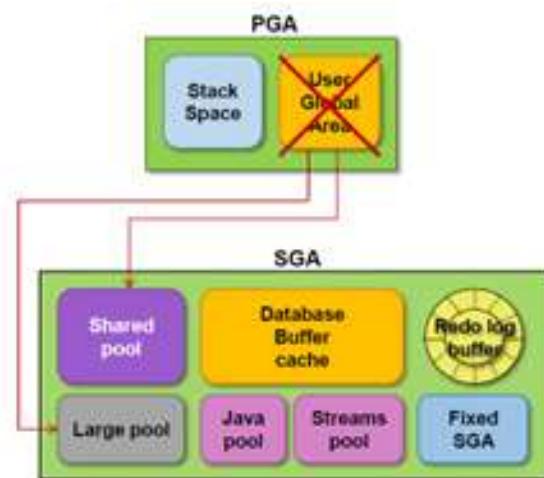
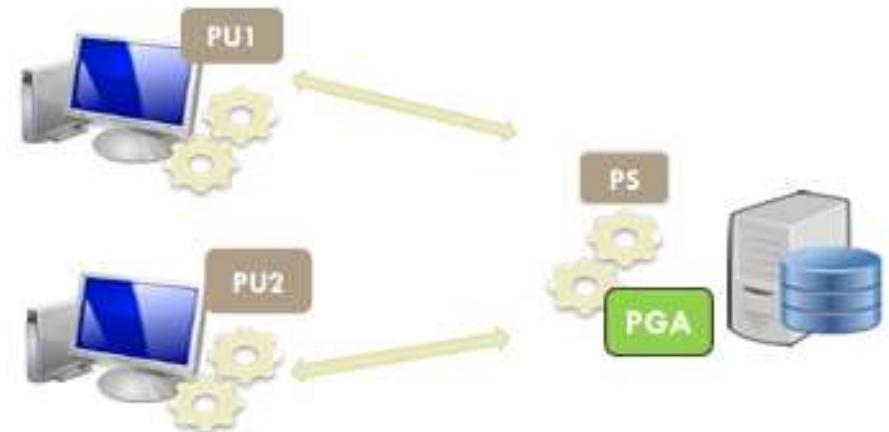
- Chaque utilisateur se connecte à l'instance de base de données via un processus serveur distinct.
- La mémoire PGA est composée d'un **Stack Space** (espace de pile) et d'une **mémoire UGA** (User Global Area)



INSTANCE ORACLE – MÉMOIRES PGA

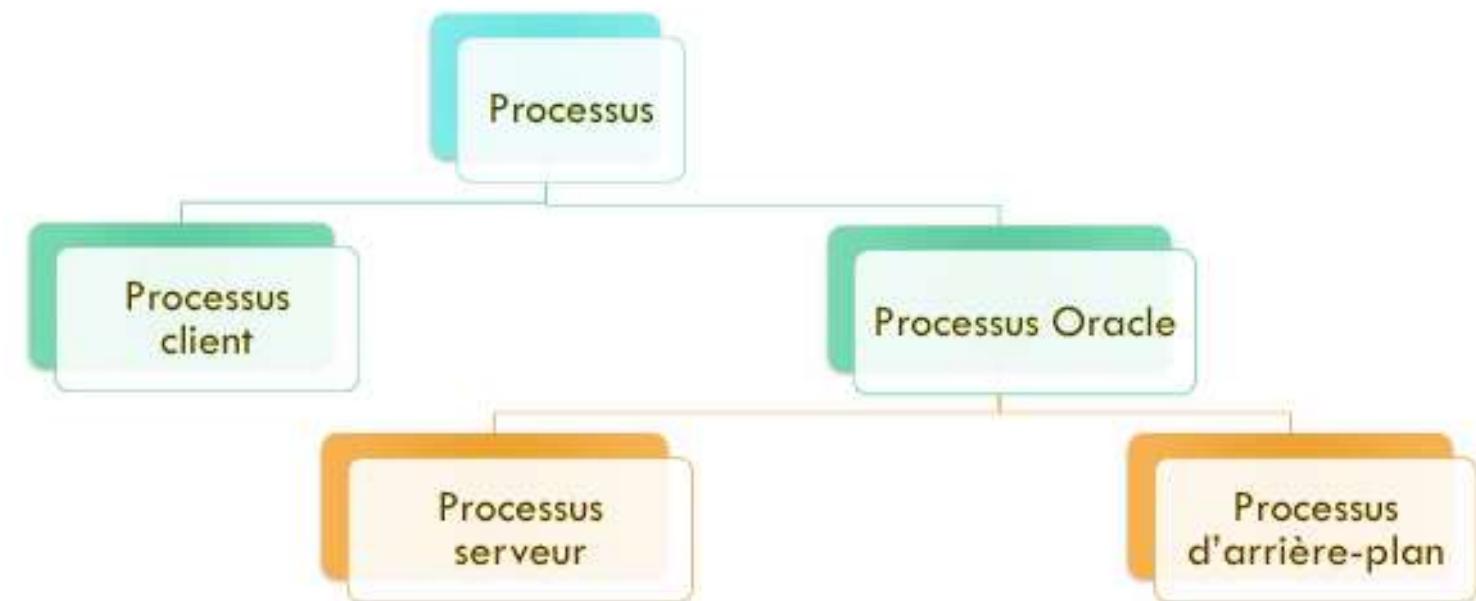
Mode serveur partagé:

- Plusieurs clients **partagent** un même processus serveur.
- la mémoire **UGA** est **transférée** dans la mémoire **SGA** (dans la Shared Pool ou dans la LARGE POOL si elle est configurée)
- La mémoire PGA contient **uniquement** l'espace de pile (StackSpace)



INSTANCE ORACLE – PROCESSUS ORACLE

- Au niveau de la machine client
- Relatif à l'application ou l'outil qui lance les requêtes Oracle (exemple: SQL*Plus).



- Exécute tâches relatives à la requête du client.
- Analyse (parse) les requêtes SQL.
- Place résultat dans Shared Pool.
- Crée et exécute plan de requête.
- Lit à partir du buffer cache ou disque
- Démarrant au lancement de l'instance.
- Exécutent des tâches de maintenance:
 - Nettoyage des process
 - Reprise de l'instance...

INSTANCE ORACLE – PROCESSUS EN ARRIÈRE PLAN

Intérêt Processus en arrière-plan (background processes):

- **Consolider** des fonctionnalités, **remplacer plusieurs programmes** sur le serveur de base de données.
- Effectuer les **opérations de I/O** mémoire disque.
- Permettre le **parallélisme**, **l'amélioration des performances** et assurer la **consistance** des données.



PROCESSUS EN ARRIÈRE PLAN OBLIGATOIRES

Processus en arrière-plan **obligatoires**:

- Process Monitor Process (PMON) Group (**PMON**, CLMN, CLnn)
- Process Manager (PMAN)
- Listener Registration Process (**LREG**)
- System Monitor Process (**SMON**)
- Database Writer Process (**DBW**)
- Log Writer Process (**LGWR**)
- Checkpoint Process (**CKPT**)
- Manageability Monitor Processes (MMON and MMNL)
- RecovererProcess (**RECO**)



PROCESSUS EN ARRIÈRE PLAN OBLIGATOIRES -PMON

Le PMON Group comprend des processus qui interviennent dans le monitoring et le nettoyage des autres processus. Il supervise le nettoyage du buffer cache et la libération des ressources (des processus terminés) pour réutilisation par d'autres processus.

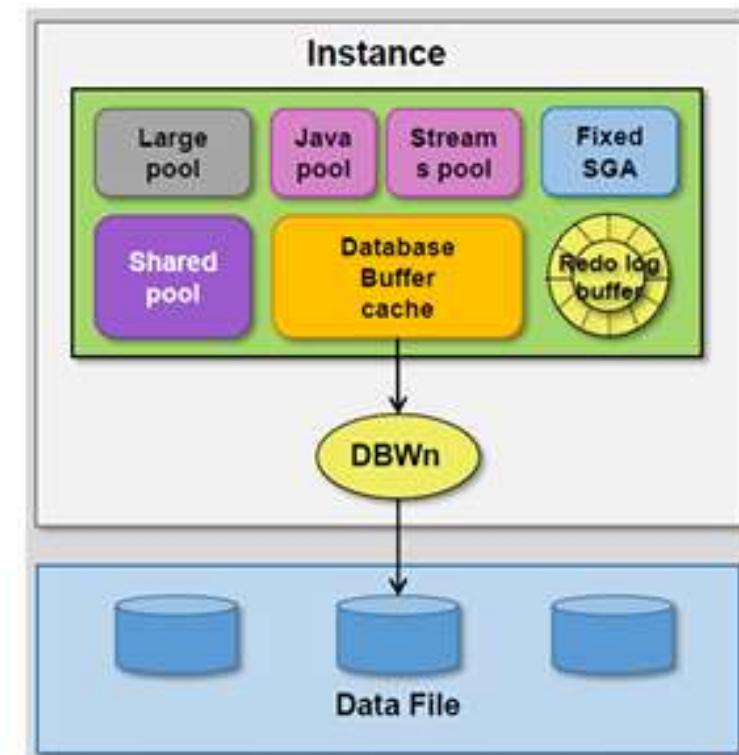
- **PMON (Process Monitor)**: détecte les processus en arrière plan terminés, ou anormalement terminés (exemple de la commande ALTER SYSTEM KILL SESSION).
- **CLMN (Cleanup Main Process)**: effectue le travail de nettoyage (processus, sessions, transactions ou connexions réseau terminés, sessions inactives, ...)
- **CLnn (Cleanup Helper Processes)**: peuvent participer dans la tâche de nettoyage, en fonction de la charge de travail de nettoyage (le nombre de ces processus est proportionnel à la charge) => interviennent si processus de nettoyage bloqué, pour raccourcir le temps de nettoyage...

PROCESSUS EN ARRIÈRE PLAN OBLIGATOIRES -SMON

- SMON (System monitor process) se charge d'un certain nombre d'opérations au niveau système:
 - ✓ Reprise de l'instance au moment du redémarrage (cas de redémarrage suite à une défaillance).
 - ✓ Libération des segments temporaires qui ne sont plus utilisés.
 - ✓ Récupération des transactions terminées mais ignorées pendant une défaillance du système et la récupération de l'instance.

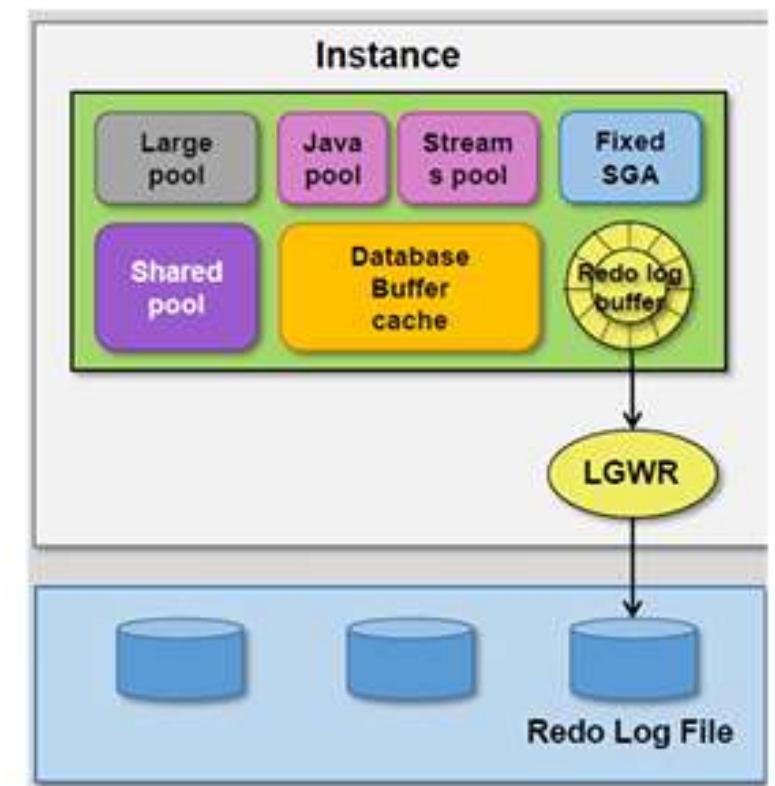
PROCESSUS EN ARRIÈRE PLAN OBLIGATOIRES -DBW

- DBW (Database writer) écrit le bloc de données modifiés (dirty buffer) depuis le database buffer cache vers les fichiers de données dans la base de données.
- DBW écrit dans les cas suivants:
 - ✓ Le processus serveur ne trouve plus de buffer « clean » pour écrire (après avoir passé en revue un nombre seuil de tampons) => demande au DBW d'écrire sur le disque pour libérer le buffer.
 - ✓ Le check point indique qu'il faut écrire les blocs « dirty » dans la base.
- Oracle Database autorise un maximum of 100 database writer processes. Ce nombre est défini par le paramètre d'initialisation DB_WRITER_PROCESSES.



PROCESSUS EN ARRIÈRE PLAN OBLIGATOIRES -LGWR

- LGWR (Log writer process) gère le redo log buffer en écrivant ses entrées dans le fichier de journalisation sur le disque (redo log files).
- Ecrit toutes les entrées de journalisation qui ont été copiées dans le tampon depuis la dernière opération d'écriture.
- Le redolog buffer est réutilisable => peut recevoir de nouvelles données provenant du processus serveur dès que le LGWR aura écrit dans les redo log files (sur disque).
- LGWR effectue une opération d'écriture:
 - ✓ Quand un processus utilisateur valide une transaction.
 - ✓ Quand un tiers du redo log buffer est plein.
 - ✓ Avant qu'un processus DBW n'écrive des tampons modifiés sur le disque (si nécessaire).
 - ✓ Toutes les 3 secondes.



PROCESSUS EN ARRIÈRE PLAN OBLIGATOIRES -CKPT

- CKPT (checkpoint process) met à jour le fichier de contrôle et l'entête des fichiers de données avec des informations notamment sur le checkpoint + SCN (System Change Number).
- Le CKPT indique au DBW d'écrire les blocs vers le disque.

PROCESSUS EN ARRIÈRE PLAN OBLIGATOIRES -RECO

- RECO (Recoverer process): Intervient dans une configuration de base de données distribuées.
- Se connecte automatiquement aux autres bases impliquées dans des transactions distribuées équivoques (présentant des doutes).
- Résout automatiquement toutes les transactions équivoques.
- Supprime, dans les différentes bases de données impliquées, toutes les lignes correspondant aux transactions équivoques résolues.

PROCESSUS EN ARRIÈRE PLAN FACULTATIFS

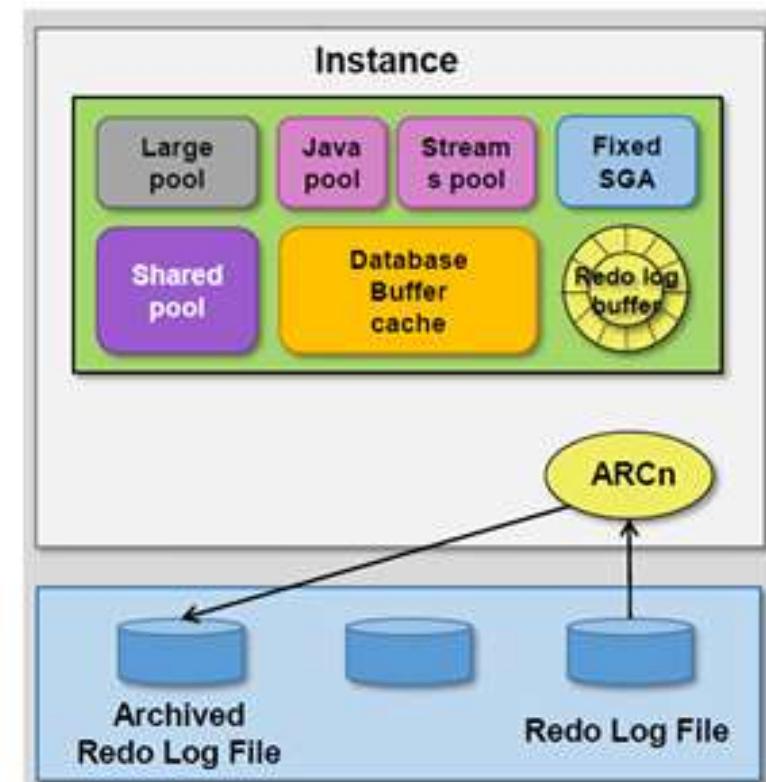
Processus en arrière-plan **facultatifs**:

- Archiver Processes (ARCn)
- Job Queue Processes (CJQ0 and Jnnn)
- Flashback Data Archive Process (FBDA)
- Space Management Coordinator Process (SMCO)



PROCESSUS EN ARRIÈRE PLAN FACULTATIFS

- **ARCn (Archiver Processes)**: Copie les fichiers de journalisation (redo log files) sur le périphérique d'archivage désigné après un changement de fichier de journalisation.
- Peut collecter des données de journalisation sur les transactions et les transmettre à des destinations de secours.



CAS D'EXÉCUTION D'UNE REQUÊTE SQL: INTERACTION INSTANCE/BASE DE DONNÉES

- Lister les étapes et les éléments qui interviennent dans l'exécution de la requête:

```
UPDATE customers SET city = 'Rabat' WHERE customer_id = 123;  
COMMIT;
```

PLAN DE LA SÉANCE

- ❑ Les composants de la SGA
- ❑ Processus d'arrière-plan
 - Processus Obligatoires
 - Processus Optionnels
- ❑ **Démarrage et arrêt d'une instance**

DÉMARRAGE/ARRÊT D'UNE INSTANCE

Parcours 19c Oracle DBA Workshop

Lesson 5: Starting UP and Shutting Down A Database Instance

DÉMARRAGE/ARRÊT D'UNE INSTANCE- LES MODES D'ARRÊT

Le démarrage suivant nécessite au préalable la récupération de l'instance.

Shutting Down an Oracle Database Instance

- Sometimes you need to shut down the database instance (for example, to change a static parameter or patch the database server).

Use the SHUTDOWN command to shut down the database instance in various modes: ABORT, IMMEDIATE, NORMAL, and TRANSACTIONAL.

	ABORT	IMMEDIATE	NORMAL	TRANSACTIONAL
Allows new connections	No	No	No	No
Waits until current sessions end	No	No	Yes	No
Waits until current transactions end	No	No	Yes	Yes
Forces a checkpoint and closes files	No	Yes	Yes	Yes

Le démarrage suivant ne nécessite pas au préalable la récupération de l'instance.

ORACLE

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

5-4

SQL > SHUTDOWN [NORMAL | TRANSACTIONAL | IMMEDIATE | ABORT]

DÉMARRAGE/ARRÊT D'UNE INSTANCE- LES MODES D'ARRÊT

Le démarrage suivant nécessite au préalable la récupération de l'instance.

Shutting Down an Oracle Database Instance

- Sometimes you need to shut down the database instance (for example, to change a static parameter or patch the database server).

Use the SHUTDOWN command to shut down the database instance in various modes: ABORT, IMMEDIATE, NORMAL, and TRANSACTIONAL.

	ABORT	IMMEDIATE	NORMAL	TRANSACTIONAL
Allows new connections	No	No	No	No
Waits until current sessions end	No	No	Yes	No
Waits until current transactions end	No	No	Yes	Yes
Forces a checkpoint and closes files	No	Yes	Yes	Yes

Le démarrage suivant ne nécessite pas au préalable la récupération de l'instance.

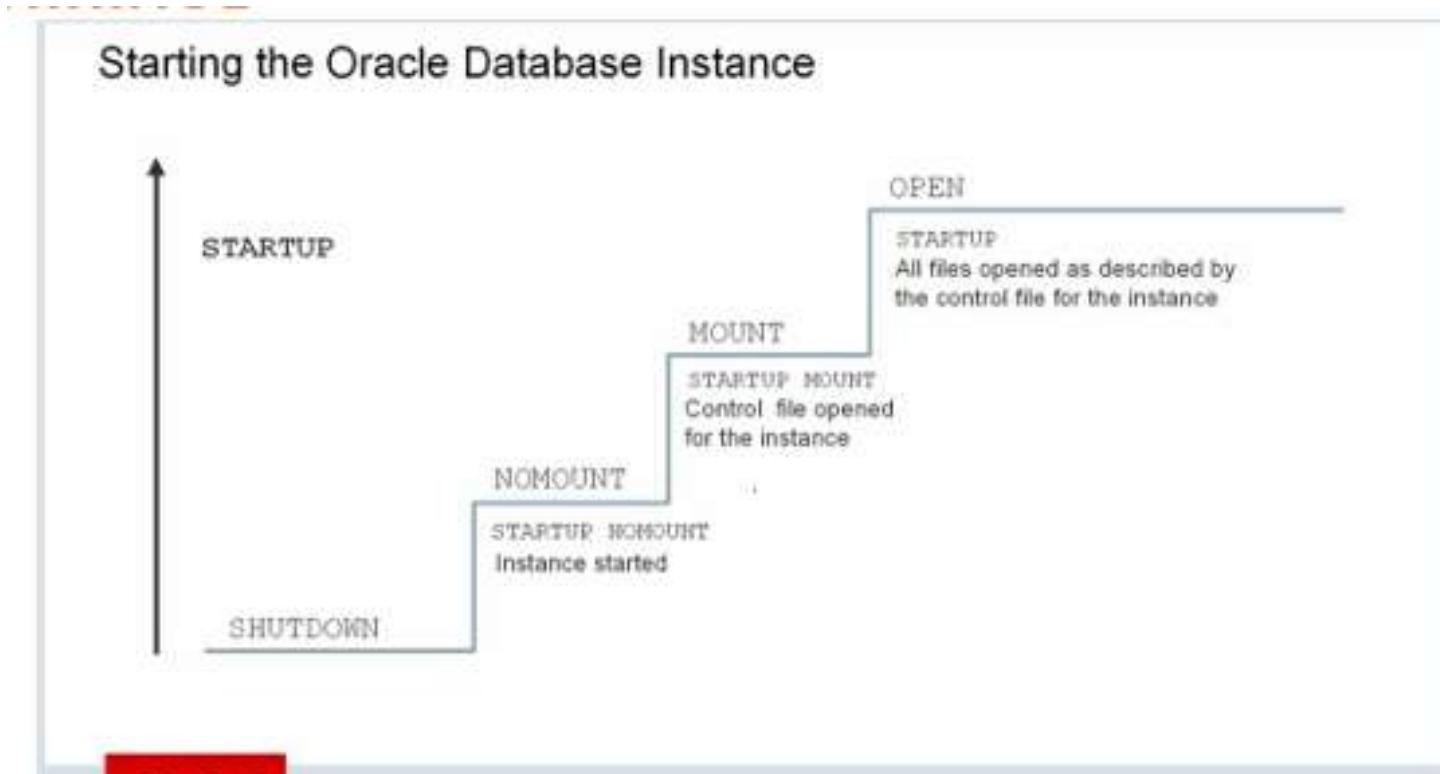
ORACLE

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

5-4

SQL > SHUTDOWN [NORMAL | TRANSACTIONAL | IMMEDIATE | ABORT]

DÉMARRAGE/ARRÊT D'UNE INSTANCE- LES MODES DE DÉMARRAGE



SQL > **STARTUP [FORCE] [RESTRICT] [MOUNT | OPEN | NOMOUNT]**

DÉMARRAGE/ARRÊT D'UNE INSTANCE- EXERCICE

Exercices :

Dérouler et commenter les commandes ci-dessous:

SQL> SHUTDOWN IMMEDIATE;

SQL> STARTUP NOMOUNT;

SQL> ALTER DATABASE MOUNT;

SQL> ALTER DATABASE OPEN;

Chapitre 1: Introduction

Chapitre 2: Installation Oracle Database 19c et Formatage des données

Chapitre 3: Architecture

Chapitre 4: Instance Oracle

Chapitre 5: Gestion des paramètres et des fichiers de paramètre

Chapitre 6: Gestion des Fichiers de journalisation

Chapitre 7: Archivelog

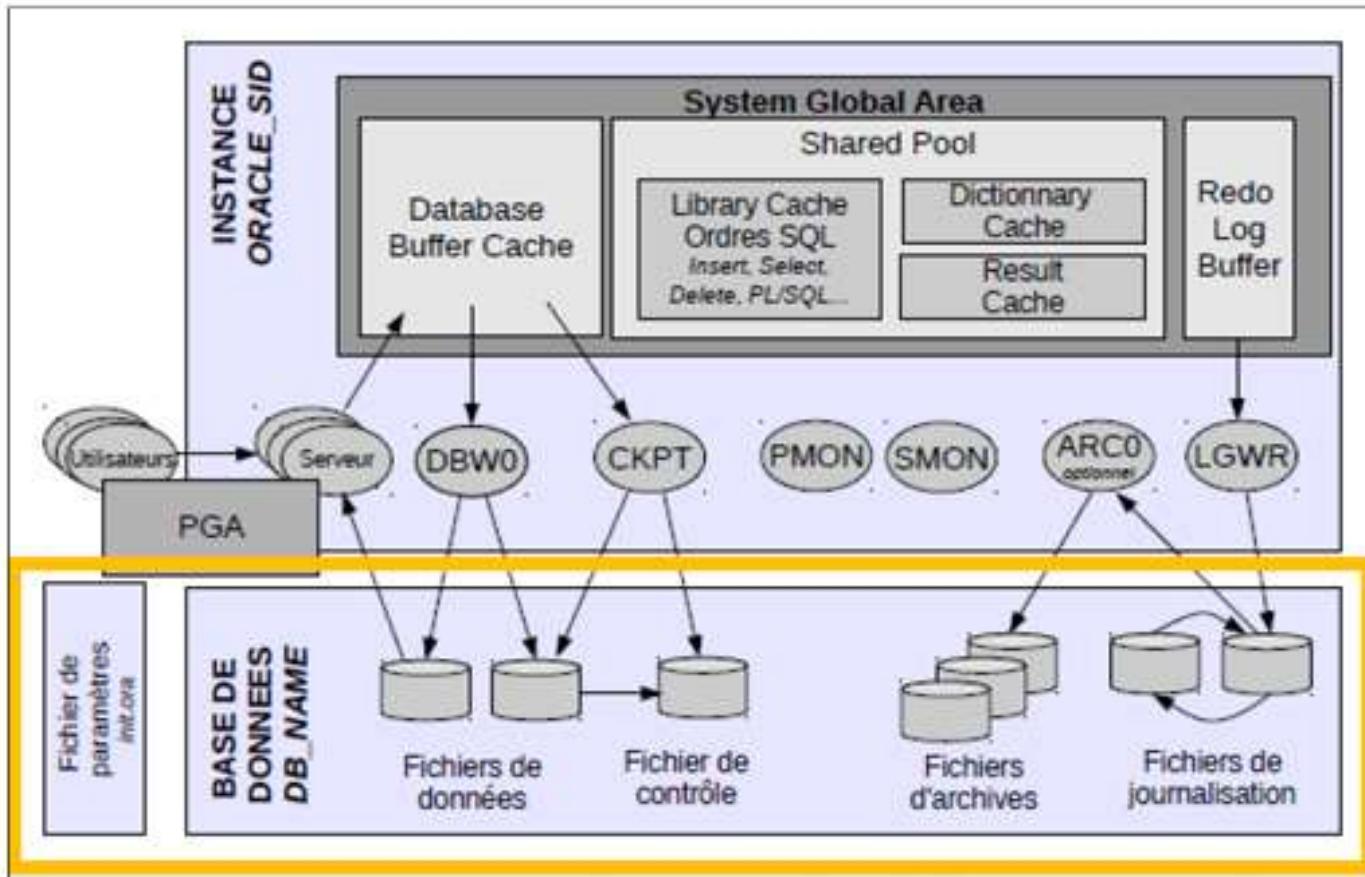
Chapitre 8: Gestion des fichiers de contrôle

Chapitre 9: Tablespaces

PLAN DE LA SÉANCE

- Gestion des paramètres d'initialisation et des fichiers de paramètres:
 - Fichiers de paramètres
 - Paramètres d'initialisation– Types de paramètres
 - Paramètres d'initialisation– Modification des paramètres
 - Manipulation des paramètres d'initialisation sous SQL*Plus

ARCHITECTURE DE L'INSTANCE -RAPPEL



FICHIERS DE PARAMÈTRES

Au démarrage, l'instance lit les paramètres de configuration depuis un fichier de paramètre.

Existe 2 types de fichiers de paramètres:

- Fichier **binnaire**
- Accessible par la base de données en **lecture et en écriture**
- Ne peut **pas** être modifié **manuellement**
- Commande **ALTER SYSTEM** pour modifier les paramètres d'initialisation
- Les **paramètres persistent** même après arrêt/redémarrage de l'instance
- Crée automatiquement par DBCA au moment de la création de la base (ou CDB)
- Nom par défaut: **SPFILE<SID>.ora**

SPFILE



PFILE



- Fichier **texte**
- Accessible par la base de données en **lecture uniquement**.
- Doit être modifié avec un éditeur de texte puis l'instance redémarrée pour prise en compte des modifications de façon **persistante**.
- Utilisé automatiquement par la base de données si SPFILE introuvé.
- Nom par défaut: **init<SID>.ora**

Ordre de priorité entre les fichiers de paramètres:

SPFILE<SID>.ora



SPFILE.ora



init<SID>.ora

FICHIERS DE PARAMÈTRES

Au démarrage, l'instance lit :

Existe 2 types de fichiers de paramètres :

- Fichier binaire
- Accessible par la base de données en lecture et en écriture

db_name='sample': Nom de ta base de données

control_files='/disk1/oradata/sample_cf.dbf':

Localisation du fichier de contrôle.

db_block_size=8192: Taille du bloc Oracle (8 Ko).

open_cursors=52: Nombre maximal de curseurs ouverts par session.

undo_management='AUTO': Indique qu'Oracle gère automatiquement les segments UNDO.

shared_pool_size=280M: Mémoire allouée au **Shared Pool** (parsing SQL, dictionnaire, PL/SQL, etc.)

pga_aggregate_target=29M: Quantité de mémoire allouée globalement au PGA (tri, hash, opérations privées par session)

Exemple de fichier PFILE qui décrit les paramètres nécessaires pour démarrer une instance

- Fichier texte

- Accessible par la base de données en lecture uniquement.

soit être modifié avec un autre contexte puis l'instance redémarre en compte des modifications persistante. Il est utilisé automatiquement si une base de données si SPFILE n'a pas été nommée par défaut: init<SID>.ora

de paramètre.

Ordre de priorité entre les fichiers de

```
db_name='sample'  
control_files='/disk1/oradata/sample_  
cf.dbf'  
db_block_size=8192  
open_cursors=52  
undo_management='AUTO'  
shared_pool_size=280M  
pga_aggregate_target=29M
```

init<SID>.ora

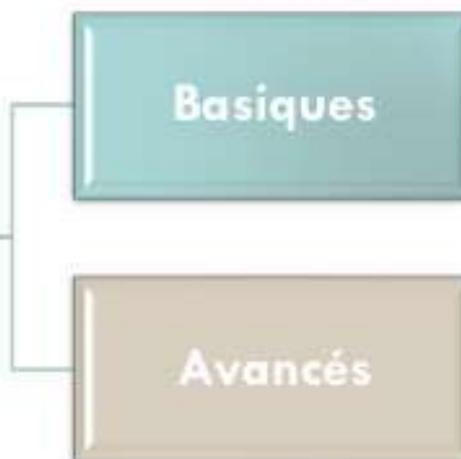
PARAMÈTRES D'INITIALISATION – TYPES DE PARAMÈTRES

Les paramètres d'initialisation permettent de:

- Spécifier des limites pour les ressources, les processus, la base de données...
- Spécifier l'emplacement de fichiers et de répertoires
- Manipuler les performances

- Le fichier de paramètres doit contenir au minimum la valeur du paramètre DB_NAME.
- Les autres paramètres ont des valeurs par défaut.

**Types de paramètres
d'initialisation**



- Une trentaine.
- Leur paramétrage est suffisant pour assurer des performances raisonnables de la base de données.
- Quelques 300 paramètres.
- Leur paramétrage est facultatif, rarement nécessaire pour atteindre un fonctionnement optimale de la base de données.

PARAMÈTRES D'INITIALISATION – MODIFICATION DES PARAMÈTRES

La vue v\$parameter renseigne sur comment modifier les paramètres:

- Au niveau session
- Au niveau système
- Au niveau PDB

Outils de modification des paramètres:

- Entreprise Manager
- SQL*Plus



PARAMÈTRES D'INITIALISATION – MODIFICATION DES PARAMÈTRES

ISSYS_MODIFIABLE

Clause SCOPE

- Trois valeurs possibles:
 - **IMMEDIATE**: Changement **immédiat** et applicable à **toutes les sessions courantes**.
 - **DEFERRED**: Changement applicable à partir des **prochaines sessions**.
 - **FALSE** : Changement applicable pour les **prochaines instances (paramètres statiques)**. Arrêt/redémarrage nécessaire avec option SPFILE.
 - Commande **ALTER SYSTEM**.
-
- Permet de définir la portée de la commande ALTER SYSTEM.
 - Trois valeurs possibles:
 - **MEMORY**: modifications applicables au niveau mémoire uniquement. Non applicables pour les prochaines sessions. Ce scope **n'est pas permis pour les paramètres statiques**.
 - **SPFILE**: Changement au niveau SPFILE. Changement immédiat et persiste après redémarrage. **Seul scope permis pour les paramètres statiques**.
 - **BOTH**: Changement au niveau mémoire et SPFILE. Changement immédiat et persiste après redémarrage.

PARAMÈTRES D'INITIALISATION – AFFICHAGE DES PARAMÈTRES VIA SQL*PLUS

Affichage de tous les paramètres:

```
SQL> SHOW PARAMETER
```

Affichage du type et de la valeur par défaut du paramètre

```
SQL> SHOW PARAMETER nom_paramètre
```

Affichage de certaines caractéristiques des paramètres via la requête v\$parameter

```
SQL> SELECT * FROM v$parameter;
```

Modification d'un paramètre

```
SQL > ALTER <SYSTEM | SESSION> SET parameter_name=value SCOPE=<SPFILE | MEMORY | BOTH>
```

PARAMÈTRES D'INITIALISATION – AFFICHAGE DES PARAMÈTRES VIA SQL*PLUS

1-Afficher le contenu de spfile

```
Select value from V$spparameter where name ='processes' ;
```

2-Modifier le contenu de spfile



```
Alter system set processes=200 scope =spfile ;
```

```
Alter system set name=valeur scope =spfile ;
```

```
Alter system set name=valeur scope =memory ;
```

```
Alter system set name=valeur scope =both ;
```

PARAMÈTRES D'INITIALISATION – AFFICHAGE DES PARAMÈTRES VIA SQL*PLUS

▪ Le fichier SPFILE

- Création du fichier SPFILE à partir du fichier PFILE
 - CREATE SPFILE[='nom_fichier_spfile'] FROM PFILE [='nom_fichier_pfile']
- Création du fichier PFILE à partir du fichier SPFILE
 - CREATE PFILE[='nom_fichier_spfile'] FROM SPFILE [='nom_fichier_pfile']



PLAN DU COURS



Chapitre 1: Introduction

Chapitre 2: Formatage des données

Chapitre 3: Architecture

Chapitre 4: Instance Oracle

Chapitre 5: Gestion des paramètres et des fichiers de paramètre

Chapitre 6: Gestion des Fichiers de journalisation

Chapitre 7: Archivelog

Chapitre 8: Gestion des fichiers de contrôle

Chapitre 9: Tablespaces

PLAN DE LA SÉANCE

- ❑ Description des fichiers de journalisation
 - Définition
 - Organisation
 - Mise à jour
- ❑ Obtenir des informations sur les fichiers de journalisation
- ❑ Ajouter des groupes et des membres de fichiers de journalisation
- ❑ Supprimer des groupes et des membres de fichiers de journalisation

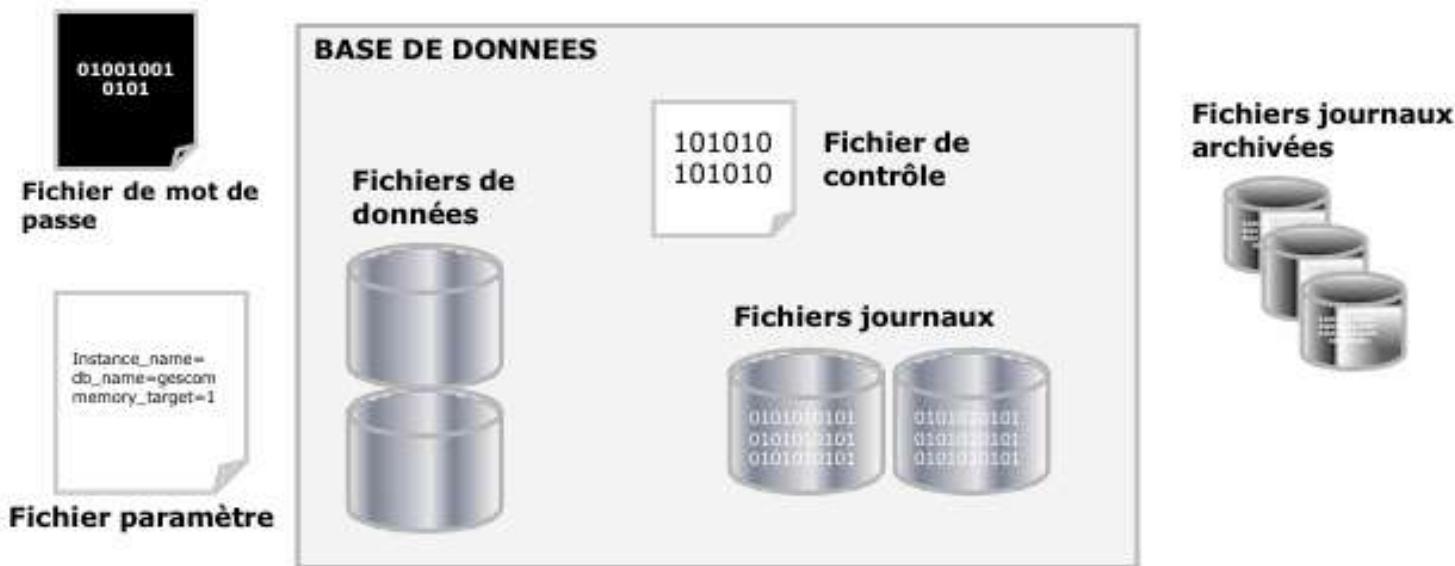
DESCRIPTION DES FICHIERS DE JOURNALISATION

DÉFINITION

- ❑ Les fichiers de journalisation en ligne (Online Redolog files) enregistrent toutes les modifications apportées à la base.
- ❑ Ils offrent :
 - Un **mécanisme de restauration** de l'instance après un **arrêt anormal**
 - Un **média** si un fichier de données est **perdu** ou **endommagé**
- ❑ Les Online Redolog Files sont organisés en **groupes** et les informations sont périodiquement écrasées.
- ❑ Lorsqu'un groupe est plein, l'instance oracle passe au groupe suivant

DESCRIPTION DES FICHIERS DE JOURNALISATION

DÉFINITION

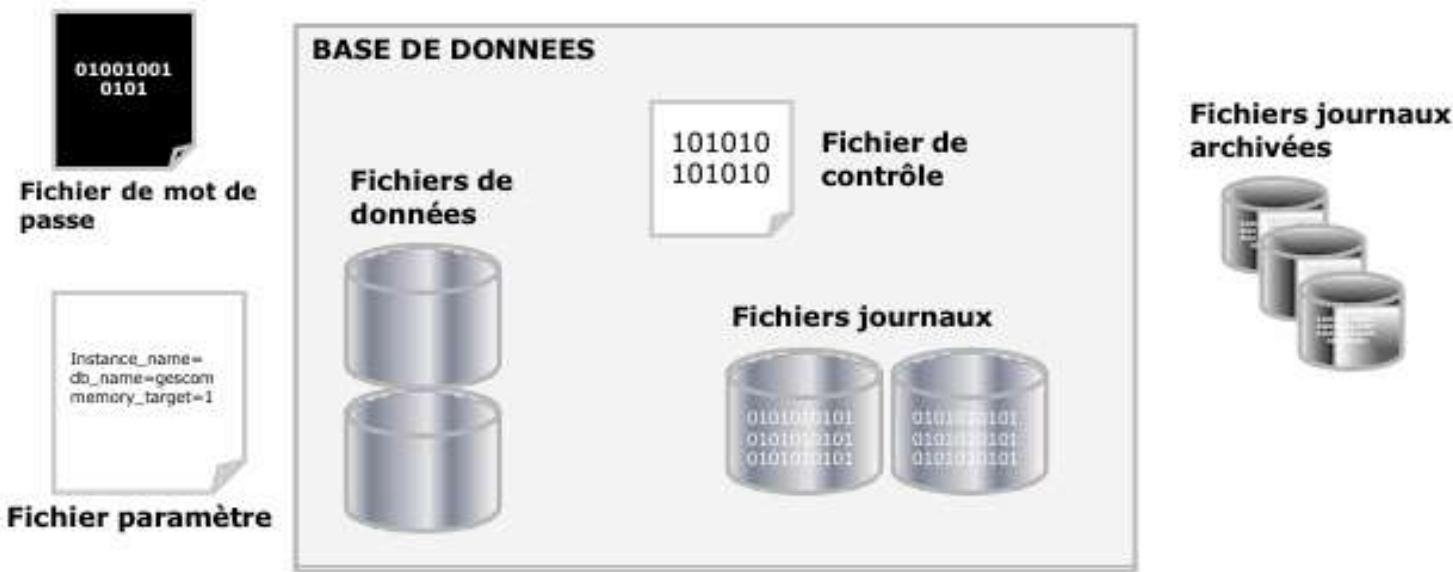


Les fichiers de journalisation présentent les caractéristiques suivantes :

- Ils enregistrent toutes les modifications apportées aux données.
- Ils offrent un mécanisme de récupération.
- Ils peuvent être organisés en groupes.
- Deux groupes au moins sont requis.

DESCRIPTION DES FICHIERS DE JOURNALISATION

DÉFINITION



Pourquoi des fichiers journaux

- Enregistre toutes les modifications (INSERT, UPDATE, DELETE,...) de la base de données
- Utilisés en cas de perte de fichiers de données

Mode de fonctionnement des Fichiers de journalisation

- Le serveur Oracle enregistre de manière séquentielle toutes les modifications apportées à la base de données dans le tampon de journalisation.
- Les entrées de journalisation sont écrites par le processus LGWR dans l'un des groupes de fichiers de journalisation, dans les cas suivants :
 - lorsqu'une transaction est validée,
 - lorsqu'un tiers du tampon de journalisation est occupé,
 - lorsque le tampon de journalisation contient plus d'un mégaoctet d'enregistrements modifiés,
 - avant que le processus DBWn n'écrive les blocs modifiés du cache de 5 tampons (buffer cache) de la base de données dans les fichiers de données.

Mode de fonctionnement des Fichiers de journalisation

- Les fichiers de journalisation sont utilisés de façon cyclique.
- Lorsqu'un fichier de ce type est rempli, le processus LGWR passe au groupe de fichiers de journalisation suivant.
 - On parle alors de changement de fichier de journalisation.
 - Une opération de point de reprise se produit également.
 - Les informations sont écrites dans le fichier de contrôle.

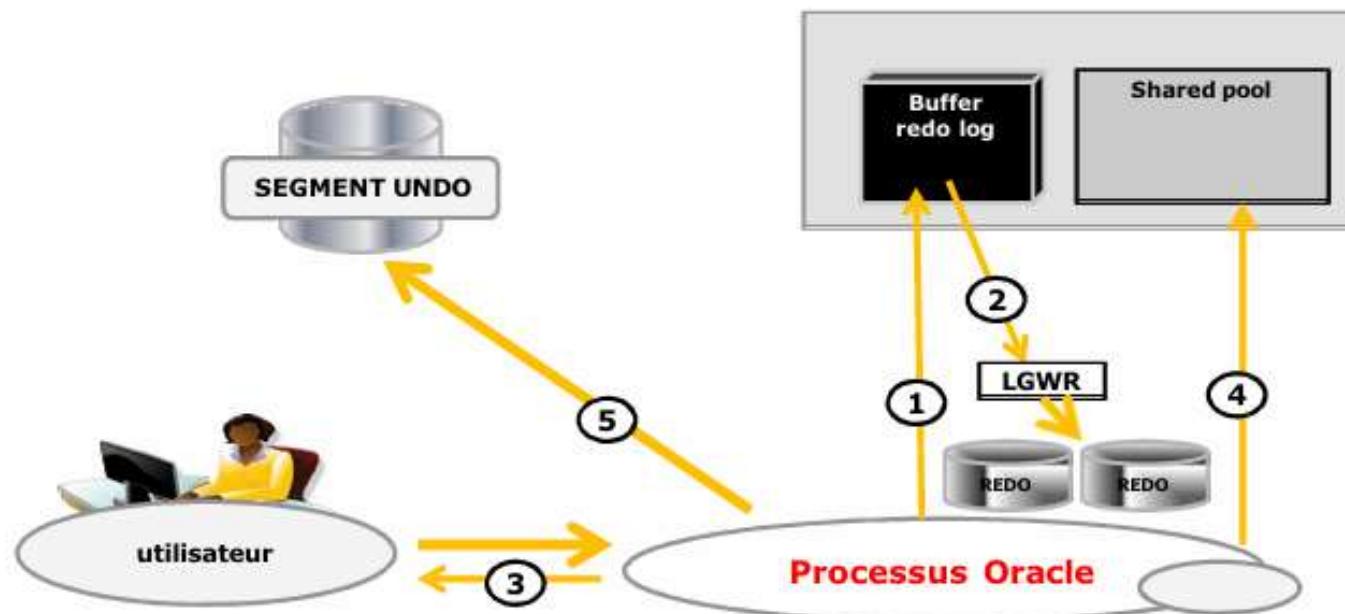
DESCRIPTION DES FICHIERS DE JOURNALISATION

Leur contenu

- L'identification de la transaction (N° process user)
- La date et l'heure de la transaction (TIMESTAMP, SCN: System Change Number)
- L'adresse physique de la données modifiée (ROWID)
- Le type d'opération effectuée : UPDATE, INSERT, DELETE, COMMIT, ROLLBACK
- Les données AVANT modification
- Les données APRES modification
- L'état de la transaction (en cours, validée, invalidée)
- L'emplacement des archives

DESCRIPTION DES FICHIERS DE JOURNALISATION

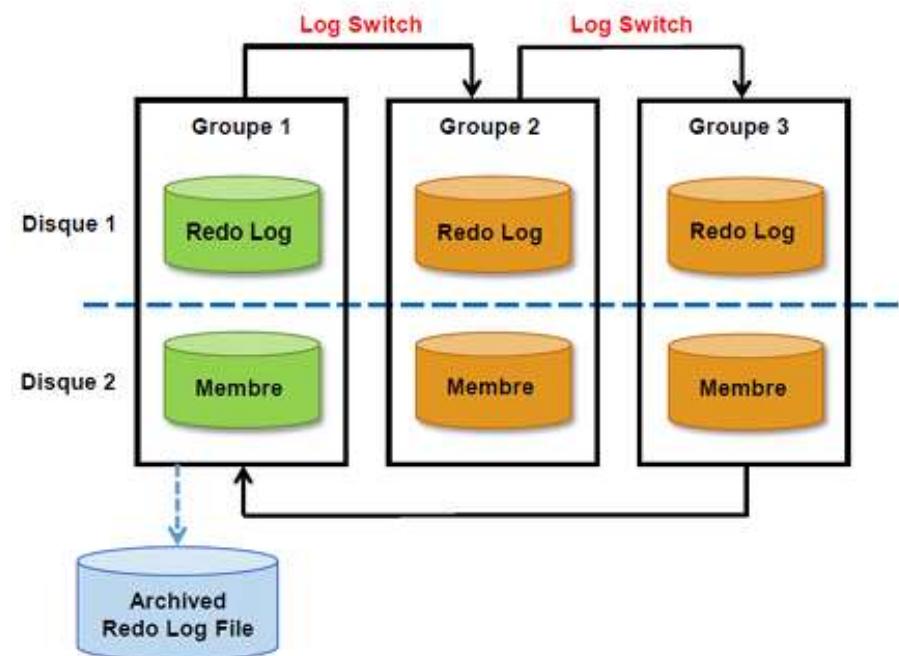
Etapes d'enregistrement



DESCRIPTION DES FICHIERS DE JOURNALISATION

ORGANISATION

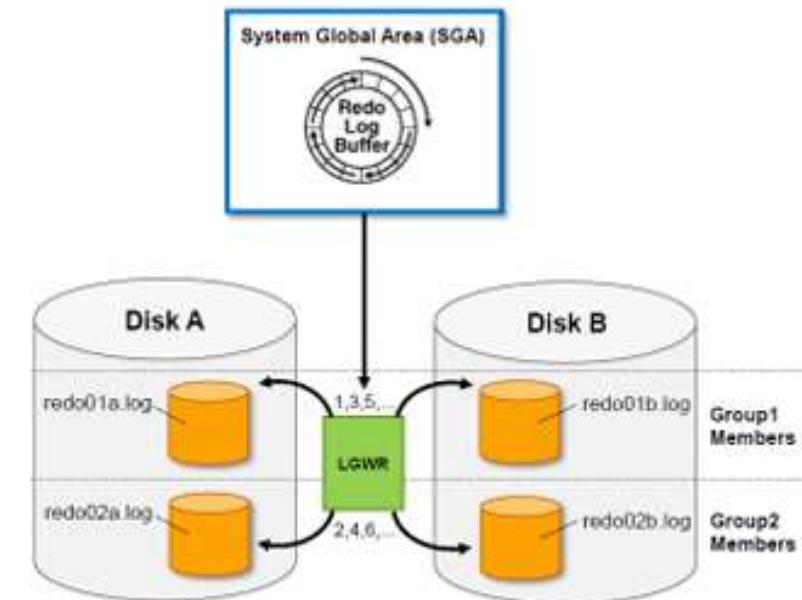
- Chaque groupe d'Online Redolog Files est composé de membres.
- Les membres d'un même groupe portent tous le même numéro de séquence (SCN : System Change Number) et ont tous la même taille.
 - SCN permet d'identifier de manière unique chaque fichier de journalisation
 - Le numéro en cours est stocké dans le fichier de contrôle et dans l'en-tête de tous les fichiers de données.
- Un groupe peut être archivé dans le fichier Archived Redo log File.



DESCRIPTION DES FICHIERS DE JOURNALISATION

MISE À JOUR

- Le serveur Oracle enregistre de manière **séquentielle** toutes les modifications apportées à la base de données dans le tampon de Redo log buffer.
- Les entrées de journalisation sont écrites par le processus **LGWR** dans l'un des groupes de fichiers de journalisation
- LGWR écrit **simultanément** les **mêmes informations** dans tous les fichiers de journalisation en ligne d'un groupe.
- Le serveur Oracle nécessite **au moins deux groupes** de fichiers de journalisation en ligne pour garantir un fonctionnement correct de la base de données.



INFORMATIONS SUR LES FICHIERS DE JOURNALISATION

Pour obtenir des informations sur les fichiers de journalisation, oracle fournit plusieurs vues dynamiques de performance, à savoir

- V\$THREAD** : obtenir des informations sur le fichier de journalisation en cours.

```
SQL> select thread#, status, groups, sequence# from v$thread;
      THREAD# STATUS      GROUPS SEQUENCE#
      -----  -----
          1 OPEN           3        14
```

INFORMATIONS SUR LES FICHIERS DE JOURNALISATION

Pour obtenir des informations sur les fichiers de journalisation, oracle fournit plusieurs vues dynamiques de performance, à savoir

- V\$LOG** : obtenir des informations sur les groupes

```
SQL> select group#, sequence#, bytes, members, status from v$log;
```

GROUP#	SEQUENCE#	BYTES	MEMBERS	STATUS
1	13	52428800	1	INACTIVE
2	14	52428800	1	CURRENT
3	12	52428800	1	INACTIVE

INFORMATIONS SUR LES FICHIERS DE JOURNALISATION

Pour obtenir des informations sur les fichiers de journalisation, oracle fournit plusieurs vues dynamiques de performance, à savoir

- V\$LOGFILE** : obtenir des informations sur les membres des groupes

```
SQL> select * from v$logfile;
```

GROUP#	STATUS	TYPE	MEMBER
3	ONLINE		E:\APP\POSTE\ORADATA\ORCL1\RED003.LOG
2	ONLINE		E:\APP\POSTE\ORADATA\ORCL1\RED002.LOG
1	ONLINE		E:\APP\POSTE\ORADATA\ORCL1\RED001.LOG

INFORMATIONS SUR LES FICHIERS DE JOURNALISATION

Pour obtenir des informations sur les fichiers de journalisation, oracle fournit plusieurs vues dynamiques de performance, à savoir

- V\$LOG_HISTORY** : obtenir des informations sur l'historique de fichier de journalisation

V\$LOG_HISTORY: informations sur les groupes

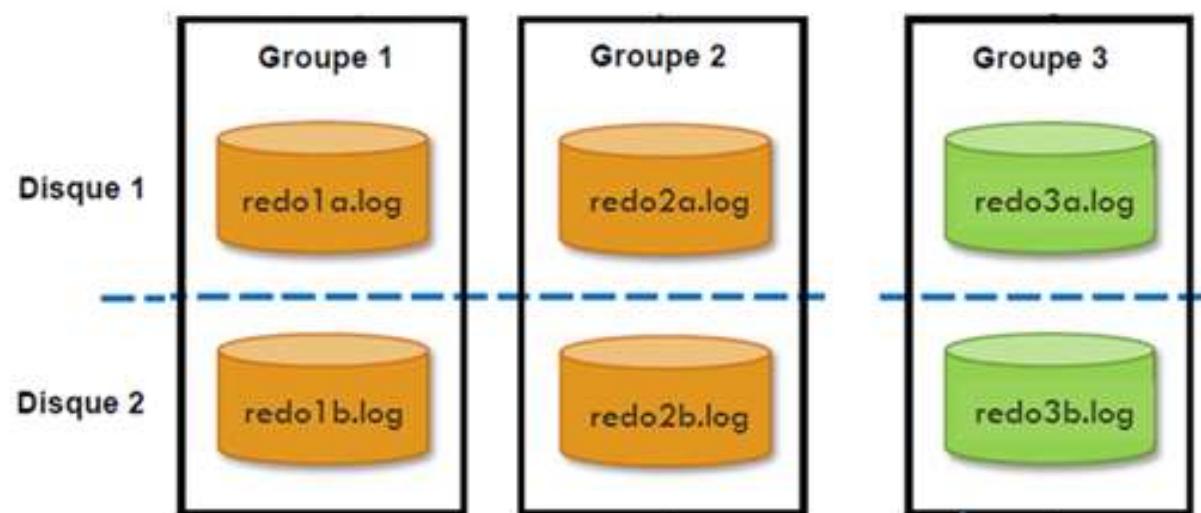
- SEQUENCE# : numéro de séquence de groupe (s'incrémente à chaque basculement)
- FIRST_CHANGE# : plus petit numéro SCN (numéro de transaction) écrit dans le groupe
- NEXT_CHANGE# : plus grand numéro SCN écrit dans le groupe
- FIRST_TIME : date et heure du plus petit numéro SCN

SQL> desc V\$LOG_HISTORY	Null?	Type
Name		

RECID		NUMBER
STAMP		NUMBER
THREAD#		NUMBER
SEQUENCE#		NUMBER
FIRST_CHANGE#		NUMBER
FIRST_TIME		DATE
NEXT_CHANGE#		NUMBER
RESETLOGS_CHANGE#		NUMBER
RESETLOGS_TIME		DATE

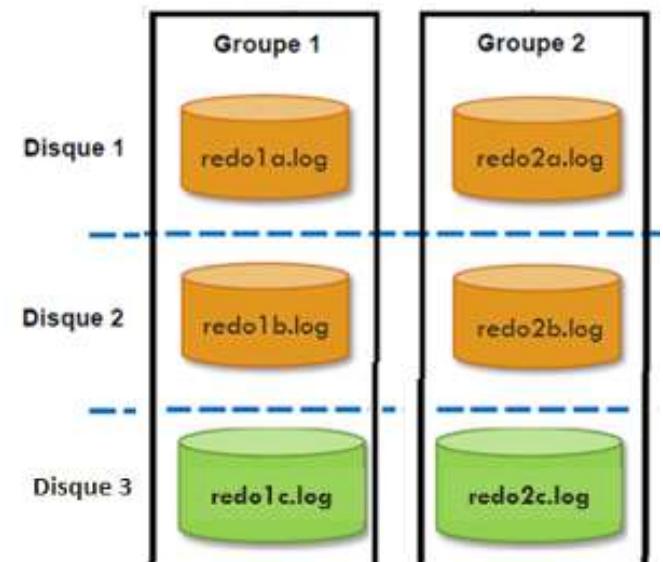
AJOUT DE GROUPES ET DE MEMBRES – AJOUT DE GROUPE

```
SQL > ALTER DATABASE ADD LOGFILE GROUP 3  
('C:/app/poste/oradata/Disk1/redo3a.log',  
 'C:/app/poste/oradata/Disk2/redo3b.log')  
SIZE 10M;
```



AJOUT DE GROUPES ET DE MEMBRES – AJOUT DE MEMBRES

```
SQL > ALTER DATABASE ADD LOGFILE MEMBER  
'C:/app/poste/oradata/Disk3/redo1c.log' TO GROUP 1,  
'C:/app/poste/oradata/Disk3/redo2c.log' TO GROUP 2;
```

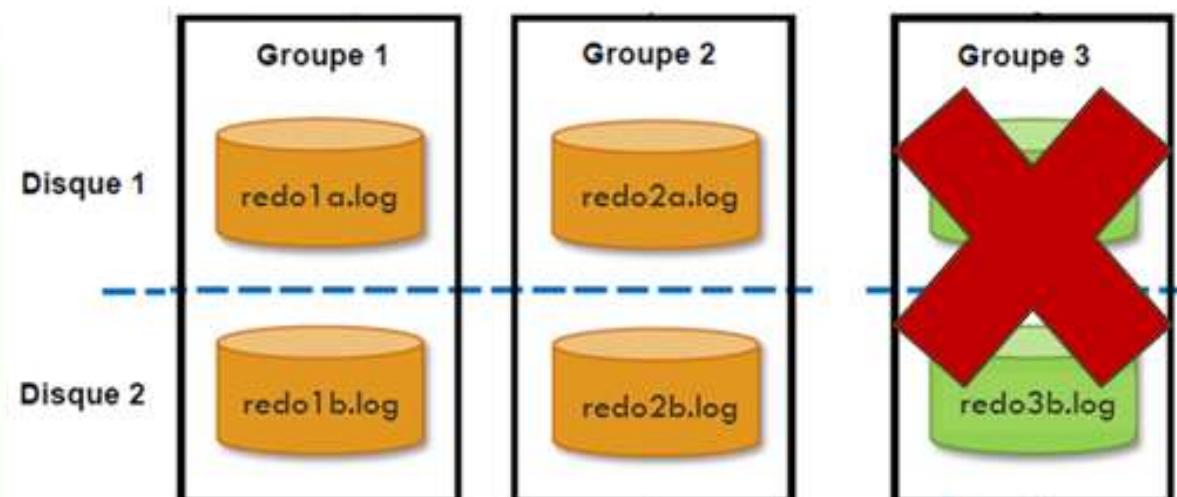


SUPPRESSION DE GROUPES ET DE MEMBRES – SUPPRESSION D'UN GROUPE

```
SQL > ALTER DATABASE DROP LOGFILE GROUP 3;
```



- L'instance nécessite au moins deux groupes de fichiers de journalisation en ligne.
- Il n'est pas possible de supprimer un groupe en cours (CURRENT).
- Lorsqu'un groupe est supprimé, les fichiers du système d'exploitation sont conservés



SUPPRESSION DE GROUPES ET DE MEMBRES – SUPPRESSION D'UN MEMBRE

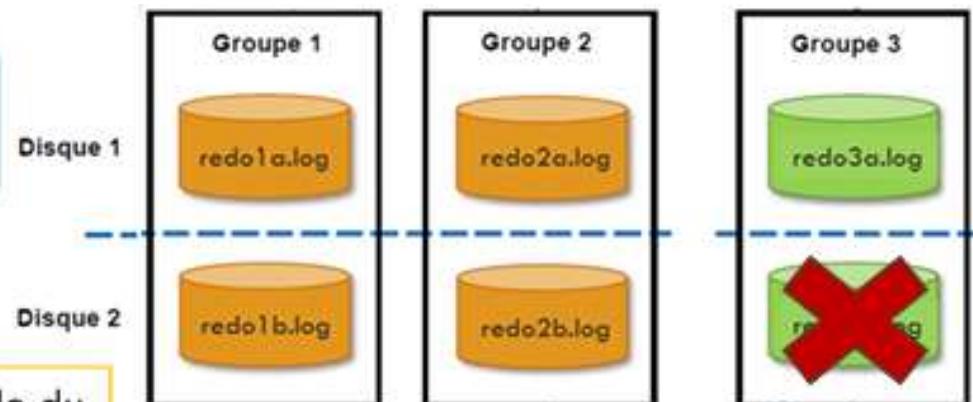
```
SQL > ALTER DATABASE DROP LOGFILE MEMBER  
'C:/app/poste/oradata/Disk3/redo3b.log';
```



- Il n'est pas possible de supprimer le dernier membre valide du groupe.
- Si la base de données fonctionne en mode **ARCHIVELOG** et que le groupe de fichiers de journalisation du membre n'est pas archivé, il n'est pas possible de supprimer le membre.

S'il s'agit du groupe en cours, il faut imposer un changement (Log switch) de fichier de journalisation pour supprimer le membre.

01/12/2023



```
SQL > ALTER SYSTEM SWITCH LOGFILE;
```

SUPPRESSION DE GROUPES ET DE MEMBRES – SUPPRESSION D'UN MEMBRE

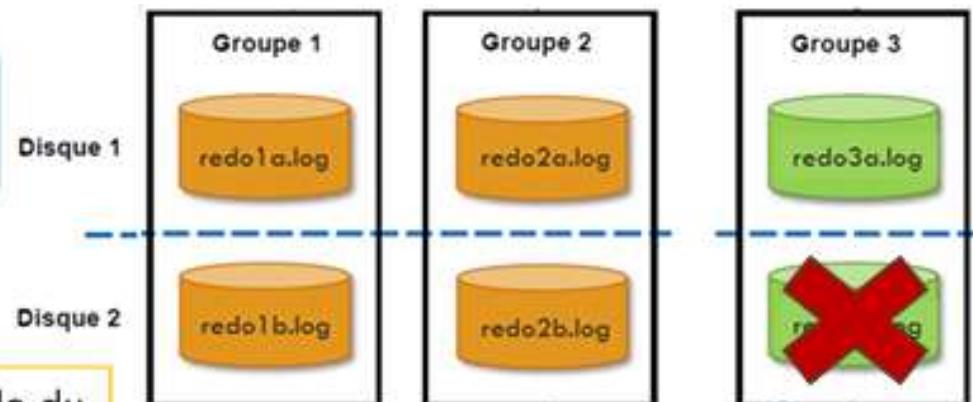
```
SQL > ALTER DATABASE DROP LOGFILE MEMBER  
'C:/app/poste/oradata/Disk3/redo3b.log';
```



- Il n'est pas possible de supprimer le dernier membre valide du groupe.
- Si la base de données fonctionne en mode **ARCHIVELOG** et que le groupe de fichiers de journalisation du membre n'est pas archivé, il n'est pas possible de supprimer le membre.

S'il s'agit du groupe en cours, il faut imposer un changement (Log switch) de fichier de journalisation pour supprimer le membre.

01/12/2023



```
SQL > ALTER SYSTEM SWITCH LOGFILE;
```

MANIPULATION DES GROUPES ET DES MEMBRES DE GROUPES

Changement de la taille d'un groupe:

Pour **augmenter** ou **réduire** la taille d'un groupe de fichiers de journalisation en ligne, il **faut ajouter un nouveau groupe** (ayant la nouvelle taille), puis **supprimer l'ancien**.

Déplacement ou renommage de membres:

Il est possible de **changer l'emplacement** des fichiers de journalisation en ligne **en renommant les fichiers**.

Avant de renommer les fichiers, il faut s'assurer que les **nouveaux fichiers existent**.

Le serveur Oracle **modifie uniquement** les pointeurs dans les fichiers de contrôle, il **ne renomme pas physiquement** les fichiers de système d'exploitation et **ne crée pas** de fichiers.

```
SQL > ALTER DATABASE RENAME FILE  
'C:/app/poste/oradata/Disk4/redo2a.log' TO  
'C:/app/poste/oradata/Disk5/redo4b.log';
```

PLAN DU COURS



Chapitre 1: Introduction

Chapitre 2: Formatage des données

Chapitre 3: Architecture

Chapitre 4: Instance Oracle

Chapitre 5: Gestion des paramètres et des fichiers de paramètre

Chapitre 6: Gestion des Fichiers de journalisation

Chapitre 7: Archivelog

Chapitre 8: Gestion des fichiers de contrôle

Chapitre 9: Tablespaces

PLAN DE LA SÉANCE

- ❑ Définition et avantages des fichiers de journalisation archivés
- ❑ Informations sur l'archivage
- ❑ Les modes d'archivage:
 - Mode NOARCHIVELOG
 - Mode ARCHIVELOG
 - Changement du mode d'archivage
 - Destination des archivelogs
 - Archivage manuel

FICHIERS DE JOURNALISATION ARCHIVÉS

DÉFINITION

- Les fichiers de journalisation archivés (Archived Redo log files) archivent les groupes des fichiers de journalisation en ligne qui sont **inactifs**.
- Un fichier de journalisation archivé est **une copie d'un membre** d'un groupe de fichiers de journalisation en ligne.
- L'archivage est assuré par le(s) processus en arrière plan ARCn.
- Une base de données peut être en mode **NOARCHIVELOG** ou **ARCHIVELOG**.
- Par défaut, une base de données est créée en mode NOARCHIVELOG.
- Les fichiers archivés peuvent être **multiplexés**.

INFORMATIONS SUR L'ARCHIVAGE

Afficher le mode d'archivage:

```
SQL> Select log_mode from  
v$database;
```

```
SQL> select log_mode from v$database;  
  
LOG_MODE  
-----  
NOARCHIVELOG
```

Trois valeurs possibles:

- NOARCHIVELOG
- ARCHIVELOG
- MANUAL

```
SQL> archive log list;
```

```
SQL> archive log list;  
Database log mode           No Archive Mode  
Automatic archival          Disabled  
Archive destination          USE_DB_RECOVERY_FILE_DEST  
Oldest online log sequence  25  
Current log sequence        26
```

```
SQL> archive log list  
Database log mode           Archive Mode  
Automatic archival          Enabled  
Archive destination          USE_DB_RECOVERY_FILE_DEST  
Oldest online log sequence  25  
Next log sequence to archive 26  
Current log sequence        26
```

INFORMATIONS SUR L'ARCHIVAGE

Afficher le statut d'archivage d'un Redo Log:

```
SQL > SELECT MEMBERS, ARCHIVED, STATUS FROM V$LOG;
```

```
SQL> SELECT MEMBERS, ARCHIVED, STATUS FROM V$LOG;
```

MEMBERS	ARC	STATUS
---------	-----	--------

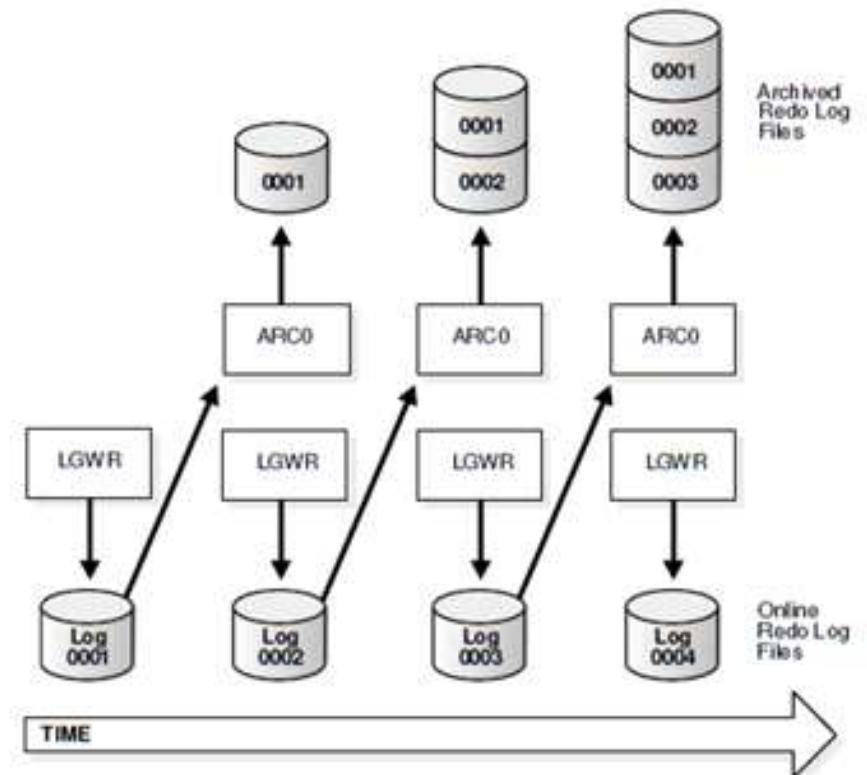
```
-----  
1 YES INACTIVE  
1 NO CURRENT
```

LES MODES D'ARCHIVAGE - NOARCHIVELOG

- En mode **NOARCHIVELOG**, l'archivage des fichiers redo est **désactivé**.
- Dès qu'un groupe de fichiers Onlive Redo log est inactifs après un log switch, le groupe sera disponible pour une réutilisation par LGWR.
- Seuls les modifications récentes dans la base, stockées dans les groupes de fichiers redo en ligne seront disponibles pour la restauration de la base.
- Pour restaurer une base en mode NOARCHIVELOG, obligatoirement faire une **sauvegarde complète quand la base est fermée**.
- Il faut **faire des sauvegardes régulièrement**.

LES MODES D'ARCHIVAGE - ARCHIVELOG

- En mode **ARCHIVELOG**, l'archivage des fichiers redo est **activé**.
- Le processus en arrière plan LGWR ne peut pas réutiliser un groupe de fichiers Online Redo log s'il n'est pas archivé.

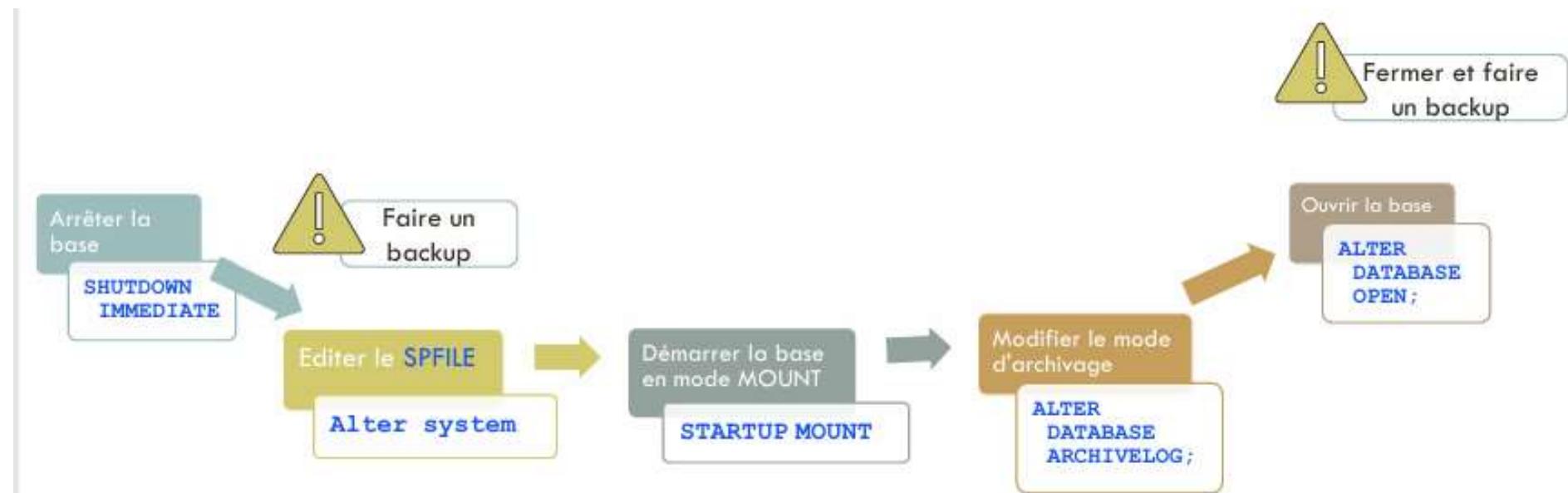


LES MODES D'ARCHIVAGE - ARCHIVELOG

- L'archivage est effectué immédiatement après un redo log switch.
- La sauvegarde de la base de données + des fichiers de journalisation en ligne + des fichiers de journalisation archivés permet de garantir la récupération de toutes les transactions validées.
- La **sauvegarde** peut s'effectuer lorsque la base de données est **ouverte**.
- L'archivage peut être automatique ou manuel.

LES MODES D'ARCHIVAGE - CHANGEMENT DU MODE D'ARCHIVAGE

- Commande ALTER DATABASE avec la clause **ARCHIVELOG** ou **NOARCHIVELOG**.
- Il faut être connecté avec des privilèges d'administrateur (**AS SYSDBA**).
- Activation du mode ARCHIVELOG:



LES MODES D'ARCHIVAGE - DESTINATION DES ARCHIVELOGS

- Il est possible de choisir une ou plusieurs destinations d'archivage (fichiers multiplexés)
- La destination d'archivage peut être locale ou distante (cas standby database)
- La destination peut être spécifiée/changée via les paramètres d'initialisation **LOG_ARCHIVE_DEST** et **LOG_ARCHIVE_DEST n** et **LOG_ARCHIVE DUPLEX DEST**

Archivage local ou distant

- Une seule destination >>> LOG_ARCHIVE_DEST
- Plusieurs destinations >>> LOG_ARCHIVE_DEST_n
 - Avec n allant de 1 à 10 pour les destinations locales et distantes et de 11 à 31 pour destinations distantes seulement.
- Exemple: LOG_ARCHIVE_DEST_1 = 'LOCATION=/disk1/arc' et
LOG_ARCHIVE_DEST_2='SERVICE=standby1'

Archivage local uniquement

- LOG_ARCHIVE_DEST et LOG_ARCHIVE_DUPLEX_DEST
- Exemple: LOG_ARCHIVE_DEST = '/disk1/arc' et LOG_ARCHIVE_DUPLEX_DEST = '/disk2/arc'

LES MODES D'ARCHIVAGE - ARCHIVAGE MANUEL

- ❑ Pour des questions de convenance et d'efficacité, **l'archivage automatique** reste le **meilleur** choix.

Si ARCHIVELOG manuel => archiver manuellement les fichiers redo logs remplis, autrement les opérations sur la base de données seront temporairement **suspendues**.

- ❑ Pour activer le mode d'archivage manuel

ALTER DATABASE ARCHIVELOG MANUAL;

- Se fait avec privilèges admin.
- Base de données doit être MOUNTED.

- ❑ Pour lancer l'archivage manuel des redo log files:

ALTER SYSTEM ARCHIVE LOG ALL;

- (archive tous les redo logs non archivés).
- L'archivage manuel peut être lancé même si le mode actif est ARCHIVELOG automatique (exemple archivage ponctuel sur un autre emplacement). Attention: LGWR peut écraser le contenu du redo log file avant la fin de cet archivage manuel.

PLAN DU COURS



Chapitre 1: Introduction

Chapitre 2: Formatage des données

Chapitre 3: Architecture

Chapitre 4: Instance Oracle

Chapitre 5: Gestion des paramètres et des fichiers de paramètre

Chapitre 6: Gestion des Fichiers de journalisation

Chapitre 7: Archivelog

Chapitre 8: Gestion des fichiers de contrôle

Chapitre 9: Tablespaces

PLAN DE LA SÉANCE

- Description du fichier de contrôle
- Informations sur les fichiers de contrôle
- Multiplexage d'un fichier de contrôle
- Sauvegarde d'un fichier de contrôle
- Crédit : Création d'un fichier de contrôle

DESCRIPTION DU FICHIER DE CONTRÔLE – INFORMATIONS GÉNÉRALES

- ❑ Le fichier de contrôle est un fichier **binnaire** qui contient des informations sur la **structure physique** de la base.
- ❑ Il garantit la **cohérence** de toute la base.
- ❑ Il est **commun** pour CDB et les PDBs.
- ❑ Il est créé **pendant la création de la base** et il est modifié en permanence.
- ❑ Il est **nécessaire au démarrage**, au **bon fonctionnement** et à la **restauration de la base**.
- ❑ Chaque fichier de contrôle est associé à **une seule base de données Oracle**.
- ❑ **Avant** l'ouverture d'une base, le fichier de contrôle est lu afin de déterminer si la base est opérationnelle
- ❑ **Seul le serveur Oracle** peut modifier le contenu du fichier de contrôle
- ❑ L'administrateur et l'utilisateur final ne peuvent pas modifier ce fichier manuellement

DESCRIPTION DU FICHIER DE CONTRÔLE – CONTENU

- Le nom de la base de données.
- L'identifiant de la base de données (SID).
- L'horodatage (TIMESTAMP) de la création de la base de données.
- Les noms et localisations des fichiers de données et des fichiers de redo log.
- Les informations sur les tablespaces.
- L'historique de journalisation.
- Les localisations et statuts des logs archivés.
- Les localisations et statuts des backups.
- Le numéro de séquence du log courant.
- Les informations des points de reprise (checkpoint).
- Le numéro de changement système (SCN) actuel



Le contenu du fichier de contrôle ne peut être affiché que si la base est dans l'état **Mount**

PLAN DE LA SÉANCE

- ❑ Description du fichier de contrôle
- ❑ **Informations sur les fichiers de contrôle**
- ❑ Multiplexage d'un fichier de contrôle
- ❑ Sauvegarde d'un fichier de contrôle
- ❑ Cration d'un fichier de controle

INFORMATIONS SUR LES FICHIERS DE CONTRÔLE

Les vues suivantes affichent des informations à partir du fichier de contrôle :

- V\$DATABASE
- V\$CONTROLFILE
- V\$CONTROLFILE_RECORD_SECTION
- V\$PARAMETER

INFORMATIONS SUR LES FICHIERS DE CONTRÔLE- LA VUE V\$CONTROLFILE

La vue V\$CONTROLFILE affiche tous les noms des fichiers de contrôle et leurs statuts qui peut être NULL ou INVALID:

```
SQL > SELECT * FROM V$CONTROLFILE;
```

```
SQL> select status, name from v$controlfile;
STATUS    NAME
-----
C:\APP\ORACLE\ORADATA\ORCL\DISKBACKUP\CONTROL01.CTL
C:\APP\ORACLE\ORADATA\ORCL\DISKBACKUP2\CONTROL02.CTL
```

```
SQL > SHOW PARAMETER CONTROL_FILES;
```

```
SQL> show parameter control_files
NAME                      TYPE        VALUE
control_files              string     C:\APP\ORACLE\ORADATA\ORCL\DIS
                            KBACKUP\CONTROL01.CTL, C:\APP\
                            ORACLE\ORADATA\ORCL\DISKBACKUP
                            2\CONTROL02.CTL
```

INFORMATIONS SUR LES FICHIERS DE CONTRÔLE- LA VUE V\$PARAMETER

La vue V\$PARAMETER retourne également ces informations:

```
SQL > SELECT name, value FROM V$PARAMETER where name='control_files';
```

```
SQL> select name, value from v$parameter where name='control_files';

NAME                  VALUE
-----
control_files          C:\APP\ORACLE\ORADATA\ORCL\DISKBACKUP\CO
                        NTROL01.CTL, C:\APP\ORACLE\ORADATA\ORCL\
                        DISKBACKUP2\CONTROL02.CTL
```

INFORMATIONS SUR LES FICHIERS DE CONTRÔLE- LA VUE V\$DATABASE

La vue V\$DATABASE contient d'autres informations sur les fichiers de contrôle.

```
SQL > select name, controlfile_type, controlfile_created from v$database;
```

```
SQL> select NAME, CONTROLFILE_TYPE, CONTROLFILE_CREATED from v$database;
```

NAME	CONTROLFILE_TYPE	CONTROLFILE_CREATED
ORCL	CURRENT	17-OCT-21

INFORMATIONS SUR LES FICHIERS DE CONTRÔLE

LA VUE V\$CONTROLFILE_RECORD_SECTION

La vue V\$CONTROLFILE_RECORD_SECTION fournit des informations sur les sections dans les fichiers de contrôle.

```
SQL > select type, record_size, records_total, records_used from v$controlfile_record_section;
```

TYPE	RECORD_SIZE	RECORDS_TOTAL	RECORDS_USED
DATABASE	316	1	1
CKPT PROGRESS	8180	11	0
REDO THREAD	256	8	1
REDO LOG	72	16	3
DATAFILE	520	100	5
FILENAME	524	2298	9
TABLESPACE	68	100	6
TEMPORARY FILENAME	56	100	1
RMAN CONFIGURATION	1108	50	0
LOG HISTORY	56	292	105
OFFLINE RANGE	200	163	0
ARCHIVED LOG	584	112	104
BACKUP SET	48	489	0
BACKUP PIECE	736	200	0
BACKUP DATAFILE	200	245	0
BACKUP REDOLOG	76	215	0
DATAFILE COPY	736	200	1

MAXLOGFILES : Nombre maximal de groupes de fichiers de redo log

INFORMATIONS SUR LES FICHIERS DE CONTRÔLE

LA VUE V\$CONTROLFILE_RECORD_SECTION

La vue V\$CONTROLFILE_RECORD_SECTION fournit des informations sur les sections dans les fichiers de contrôle.

```
SQL > select type, record_size, records_total, records_used from v$controlfile_record_section;
```

TYPE	RECORD_SIZE	RECORDS_TOTAL	RECORDS_USED
DATABASE	316	1	1
CKPT PROGRESS	8188	11	0
REDO THREAD	256	8	1
REDO LOG	72	16	3
DATAFILE	520	100	5
FILENAME	524	2298	9
TABLESPACE	68	100	6
TEMPORARY FILENAME	56	108	1
RMAN CONFIGURATION	1188	58	0
LOG HISTORY	56	292	186
OFFLINE RANGE	200	163	0
ARCHIVED LOG	584	112	104
BACKUP_SET	40	409	0
BACKUP_PIECE	736	200	0
BACKUP_DATAFILE	200	245	0
BACKUP_REDOLOG	76	215	0
DATAFILE_COPY	736	200	1

MAXDATAFILES :
Nombre maximal de fichiers de données

INFORMATIONS SUR LES FICHIERS DE CONTRÔLE

LA VUE V\$CONTROLFILE_RECORD_SECTION

La vue V\$CONTROLFILE_RECORD_SECTION fournit des informations sur les sections dans les fichiers de contrôle.

```
SQL > select type, record_size, records_total, records_used from v$controlfile_record_section;
```

```
SQL> select type, record_size, records_total, records_used from v$controlfile_record_section;
```

TYPE	RECORD_SIZE	RECORDS_TOTAL	RECORDS_USED
DATABASE	316	1	1
CKPT PROGRESS	8188	11	0
REDO THREAD	256	8	1
REDO LOG	72	16	3
DATAFILE	528	100	5
FILENAME	524	2298	9
TABLESPACE	68	100	6
TEMPORARY FILENAME	56	100	1
RMAN CONFIGURATION	1108	50	0
LOG HISTORY	56	292	106
OFFLINE RANGE	200	163	0
ARCHIVED LOG	584	112	104
BACKUP SET	48	409	0
BACKUP PIECE	736	200	0
BACKUP DATAFILE	200	245	0
BACKUP REDOLOG	76	215	0
DATAFILE COPY	736	200	1

MAXLOGHISTORY :
Nombre maximal
d'informations sur les
switchs de log

INFORMATIONS SUR LES FICHIERS DE CONTRÔLE

Les vues dynamiques ci-dessous retournent des informations à partir des fichiers de contrôle :

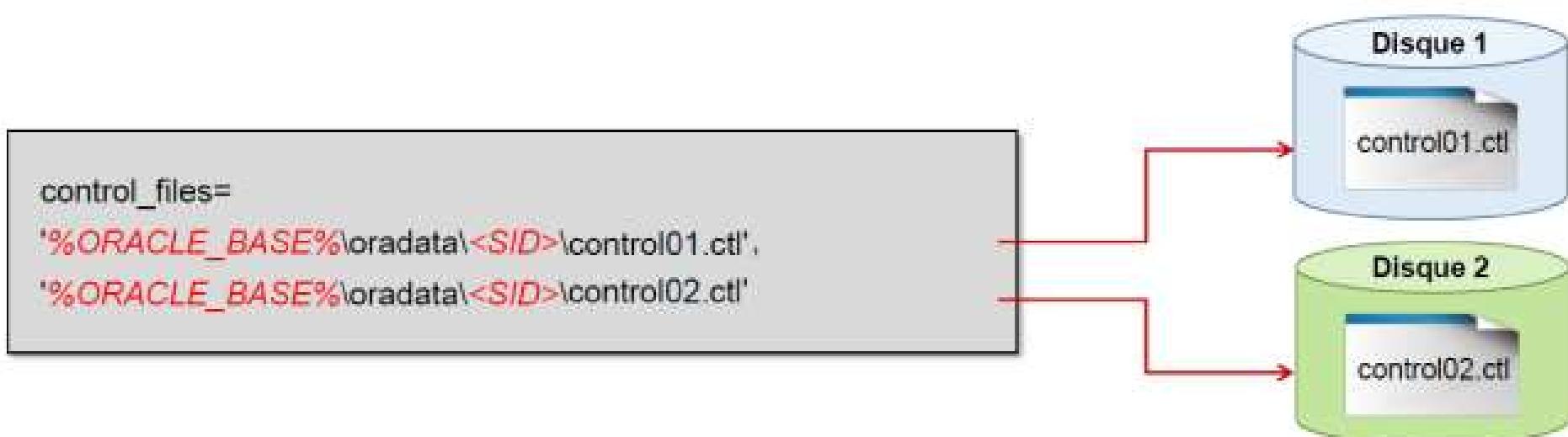
- V\$BACKUP
- V\$TEMPFILE
- V\$ARCHIVE
- V\$LOGFILE
- V\$ARCHIVED_LOG
- V\$DATAFILE
- V\$TABLESPACE
- V\$LOG
- V\$LOGHIST
- V\$DATABASE

PLAN DE LA SÉANCE

- ❑ Description du fichier de contrôle
- ❑ Informations sur les fichiers de contrôle
- ❑ Multiplexage d'un fichier de contrôle**
- ❑ Sauvegarde d'un fichier de contrôle
- ❑ Crédit d'un fichier de contrôle

MULTIPLEXAGE D'UN FICHIER DE CONTRÔLE

- Il est fortement recommandé de multiplexer les fichiers de contrôle en mettant chaque copie sur un disque différent
- Si un fichier de contrôle est perdu, une copie multiplexée du fichier de contrôle peut être utilisée pour redémarrer l'instance sans avoir à restaurer la base de données.
- Les fichiers de contrôles peuvent être multiplexés jusqu'à 8 fois.



MULTIPLEXAGE D'UN FICHIER DE CONTRÔLE

EN UTILISANT SPFILE

1. Modifier la valeur du paramètre control_files dans le fichier spfile.

```
SQL > ALTER SYSTEM set control_files=
  '<%ORACLE_BASE%>/oradata/<SID>/control01.ctl' ,
  '<%ORACLE_BASE%>/oradata/<SID>/control02.ctl' SCOPE = spfile;
```

2. Arrêter la base

```
SQL > shutdown immediate
```



Il faut faire le shutdown avant de créer la copy du fichier de contrôle

3. Copier le fichier de contrôle en utilisant les commandes du système d'exploitation

```
SQL > host copy <%ORACLE_BASE%>\oradata\<SID>\control01.ctl
  '<%ORACLE_BASE%>\oradata\<SID>\control02.ctl'
```

4. Démarrer la base

```
SQL > startup
```

MULTIPLEXAGE D'UN FICHIER DE CONTRÔLE

EN UTILISANT PFILE

1. Arrêter la base

```
SQL > shutdown immediate
```

2. Modifier le paramètre CONTROL_FILES dans le fichier init<SID>.ora

```
*.control_files=
'%ORACLE_BASE%\oradata\<SID>\control01.ctl',
'%ORACLE_BASE%\oradata\<SID>\control02.ctl'
```

3. Copier le fichier de contrôle en utilisant les commandes du système d'exploitation

```
SQL > host copy %ORACLE_BASE%\oradata\<SID>\control01.ctl
      '%ORACLE_BASE%\oradata\<SID>\control02.ctl'
```

4. Démarrer la base

```
SQL > startup pfile = %ORACLE_HOME%\database\init<SID>.ora
```

PLAN DE LA SÉANCE

- ❑ Description du fichier de contrôle
- ❑ Informations sur les fichiers de contrôle
- ❑ Multiplexage d'un fichier de contrôle
- ❑ Sauvegarde d'un fichier de contrôle**
- ❑ Crédit d'un fichier de contrôle

SAUVEGARDE D'UN FICHIER DE CONTRÔLE

- Oracle recommande de **sauvegarder** le fichier de contrôle **à chaque modification de la structure de la base** (ajouter, renommer ou supprimer un fichier de données ou un fichier journal).
- Il y a **deux façons** de sauvegarder le fichier de contrôle :
 1. Sauvegarder le fichier de contrôle dans un **fichier binaire**
 2. Sauvegarder le fichier de contrôle dans un **fichier trace**

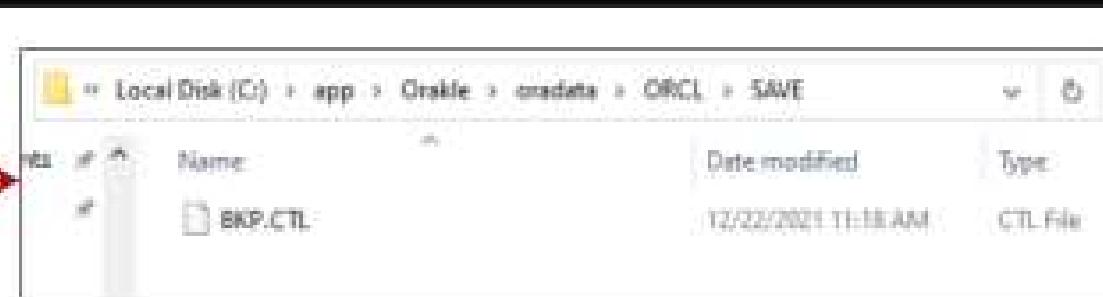
SAUVEGARDE D'UN FICHIER DE CONTRÔLE

DANS UN FICHIER BINAIRE

- Sauvegarde du fichier de contrôle dans un fichier binaire:

```
SQL > ALTER DATABASE BACKUP CONTROLFILE TO 'chemin\nom_du_fichier.ctl';
```

```
SQL> alter database backup controlfile to 'C:/app/Orakle/oradata/ORCL/SAVE/bkp.ctl';
Database altered.
```



SAUVEGARDE D'UN FICHIER DE CONTRÔLE

DANS UN FICHIER TRACE

- Sauvegarde du fichier de contrôle dans un fichier trace:

```
SQL > ALTER DATABASE BACKUP CONTROLFILE TO TRACE;
```

- C'est fichier texte.
- Permet de récupérer le script SQL pour la création d'un fichier de contrôle.
- Pour le retrouver, il faut récupérer son emplacement depuis la vue v\$diag_info:

```
SQL > SELECT name, value from v$diag_info where lower(name) like '%trace%';
```

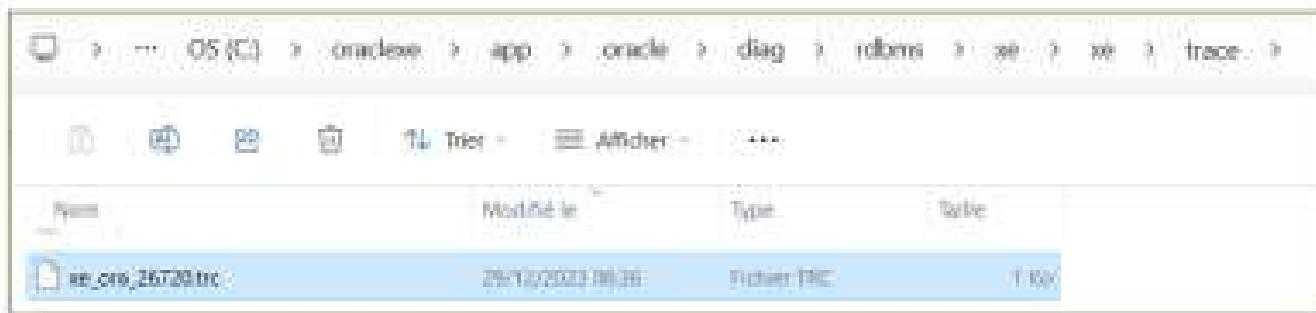
SAUVEGARDE D'UN FICHIER DE CONTRÔLE

DANS UN FICHIER TRACE

- Pour le retrouver, il faut récupérer son emplacement depuis la vue v\$diag_info:

```
SQL > SELECT name, value from v$diag_info where lower(name) like '%trace%';
```

NAME	VALUE
Diag Trace	C:\ORACLEXE\APP\ORACLE\diag\rdbms\xe\xe\trace
Default Trace File	C:\ORACLEXE\APP\ORACLE\diag\rdbms\xe\xe\trace\xe_ora_26728.trc



SAUVEGARDE D'UN FICHIER DE CONTRÔLE

DANS UN FICHIER TRACE

□ Contenu fichier trace:

```
CREATE CONTROLFILE REUSE DATABASE "XE" NORESETLOGS ARCHIVELOG
  MAXLOGFILES 16
  MAXLOGMEMBERS 3
  MAXDATAFILES 100
  MAXINSTANCES 8
  MAXLOGHISTORY 292
LOGFILE
  GROUP 1 'C:\ORACLEXE\APP\ORACLE\FAST_RECOVERY_AREA\XE\ONLINELOG\01_MF_1_llf8kqgr_log' SIZE 50M BLOCKSIZE 512,
  GROUP 2 'C:\ORACLEXE\APP\ORACLE\FAST_RECOVERY_AREA\XE\ONLINELOG\01_MF_2_llf8kqxf_log' SIZE 50M BLOCKSIZE 512
  - STANDBY LOGFILE
DATAFILE
  'C:\ORACLEXE\APP\ORACLE\ORADATA\XE\SYSTEM.DBF',
  'C:\ORACLEXE\APP\ORACLE\ORADATA\XE\SYSAUX.DBF',
  'C:\ORACLEXE\APP\ORACLE\ORADATA\XE\UNDOTBS1.DBF',
  'C:\ORACLEXE\APP\ORACLE\ORADATA\XE\USERS.DBF'
CHARACTER SET AL32UTF8
```

PLAN DE LA SÉANCE

- ❑ Description du fichier de contrôle
- ❑ Informations sur les fichiers de contrôle
- ❑ Multiplexage d'un fichier de contrôle
- ❑ Sauvegarde d'un fichier de contrôle
- ❑ Cration d'un fichier de controle**

CRÉATION D'UN FICHIER DE CONTRÔLE

- Il faut créer un fichier de contrôle dans les cas suivants:
 - Si tous les fichiers de contrôle sont perdus ou corrompus.
 - En cas de changement du nom de la base.
 - En cas de modification de certains paramètres en dur comme MAXDATAFILES, MAXLOGFILES, MAXLOGHISTROY...
 - Si la base est déplacée sur une autre machine et que l'emplacement des fichiers de données et des fichiers journaux est différent des emplacements originaux.



Il faut faire très attention pendant la création du fichier de contrôle, car il peut endommager les fichiers de données et les fichiers journaux.

Le fait d'oublier de mentionner un fichier de données pendant la création du fichier de contrôle peut causer la perte de ce fichier et même la perte de la base entière.

CRÉATION D'UN FICHIER DE CONTRÔLE

ÉTAPES DE CRÉATION

1. Shutdown immediate
2. Startup nomount
3. Exécuter le script de création récupéré depuis le fichier **trace**. Le paramétrier comme souhaité.
4. Alter database open;

```
CREATE CONTROLFILE REUSE DATABASE "ORCL" NORESETLOGS NOARCHIVELOG
MAXLOGFILES 10
MAXLOGMEMBERS 3
MAXDATAFILES 100
MAXINSTANCES 8
MAXLOGHISTORY 292

LOGFILE
  GROUP 1 ('C:\APP\ORACLE\ORADATA\ORCL\REDO01.LOG' SIZE 1M,
  GROUP 2 ('C:\APP\ORACLE\ORADATA\ORCL\REDO02.LOG' SIZE 1M,
  GROUP 3 ('C:\APP\ORACLE\ORADATA\ORCL\REDO03_01.LOG',
            'C:\APP\ORACLE\ORADATA\ORCL\REDO03_02.LOG') SIZE 1M

DATAFILE
  'C:\APP\ORACLE\ORADATA\ORCL\SYSTEM01.DBF',
  'C:\APP\ORACLE\ORADATA\ORCL\SYAUDY01.DBF',
  'C:\APP\ORACLE\ORADATA\ORCL\UNDOTBS01.DBF',
  'C:\APP\ORACLE\ORADATA\ORCL\USER01.DBF',
  'C:\APP\ORACLE\ORADATA\ORCL\EXAMPLE01.DBF',
CHARACTER SET WE8ISO8859P1
```

PLAN DU COURS



Chapitre 1: Introduction

Chapitre 2: Formatage des données

Chapitre 3: Architecture

Chapitre 4: Instance Oracle

Chapitre 5: Gestion des paramètres et des fichiers de paramètre

Chapitre 6: Gestion des Fichiers de journalisation

Chapitre 7: Archivelog

Chapitre 8: Gestion des fichiers de contrôle

Chapitre 9: Tablespaces

PLAN DE LA SÉANCE

□ Structures logiques et physiques de la base de données

Oracle

□ Informations sur les tablespaces et datafiles

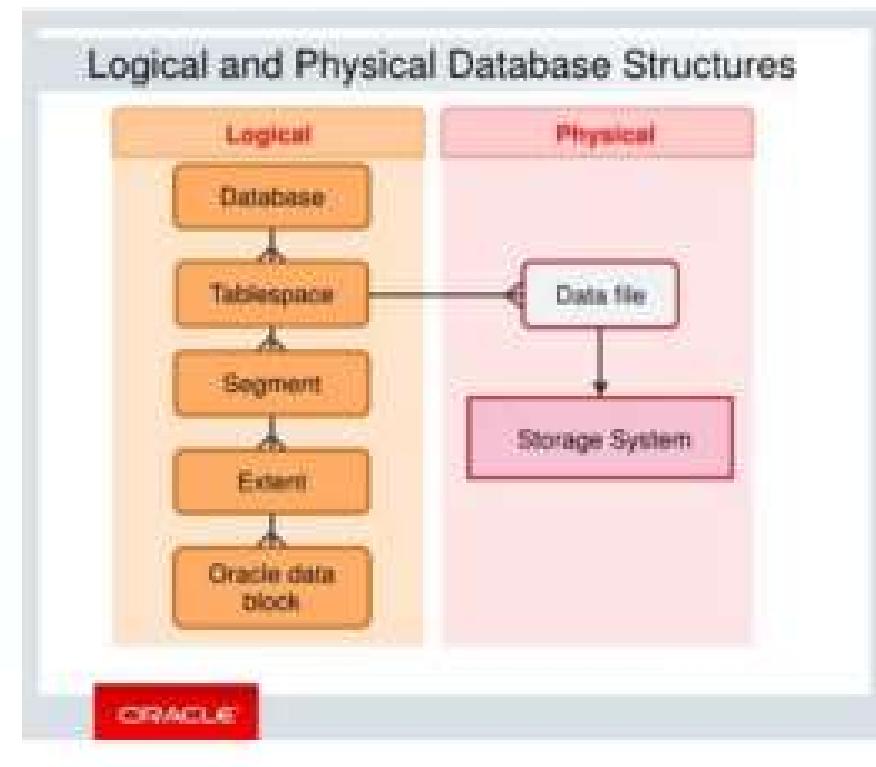
□ Crédit des tablespaces

□ Gestion des tablespaces

TABLESPACES ET FICHIERS DE DONNÉES

STRUCTURES LOGIQUE ET PHYSIQUE

- Oracle **stocke** les données du système et des utilisateurs :
 - Logiquement -> dans les tablespaces
 - Physiquement -> dans les fichiers de données
- **Une base de données:**
 - Est composée **logiquement** de deux tablespaces ou plus.
- **Un tablespace :**
 - Est une unité **logique** de stockage.
 - Regroupe les objets ayant un comportement similaire (tables, indexes, données d'une application...).
 - Est composé d'**un ou de plusieurs fichiers de données**.
 - Ne peut appartenir qu'à **une seule base de données à la fois**

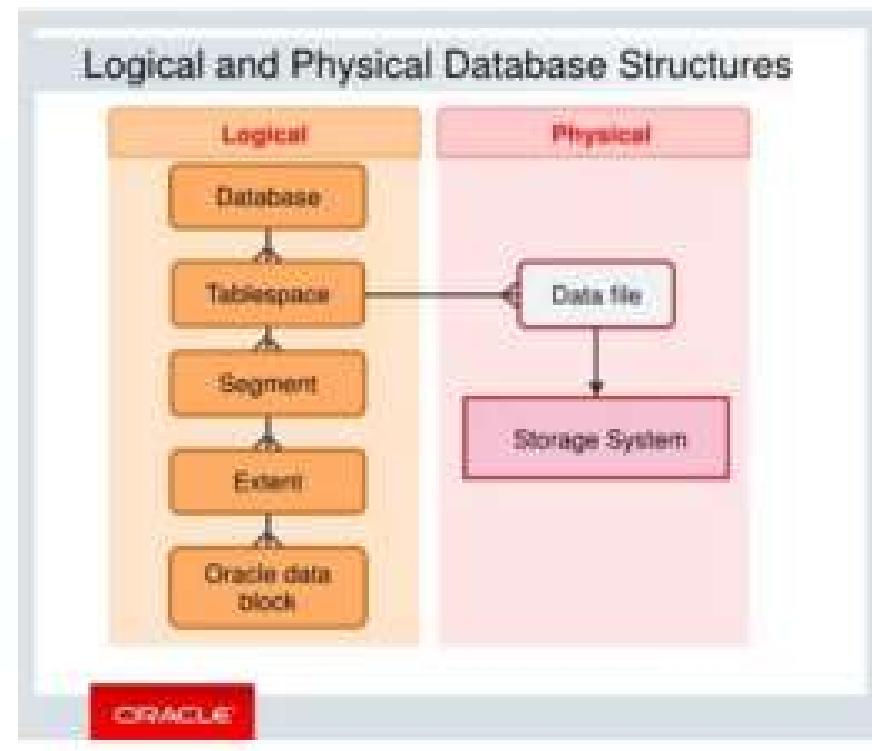


TABLESPACES ET FICHIERS DE DONNÉES

STRUCTURES LOGIQUE ET PHYSIQUE

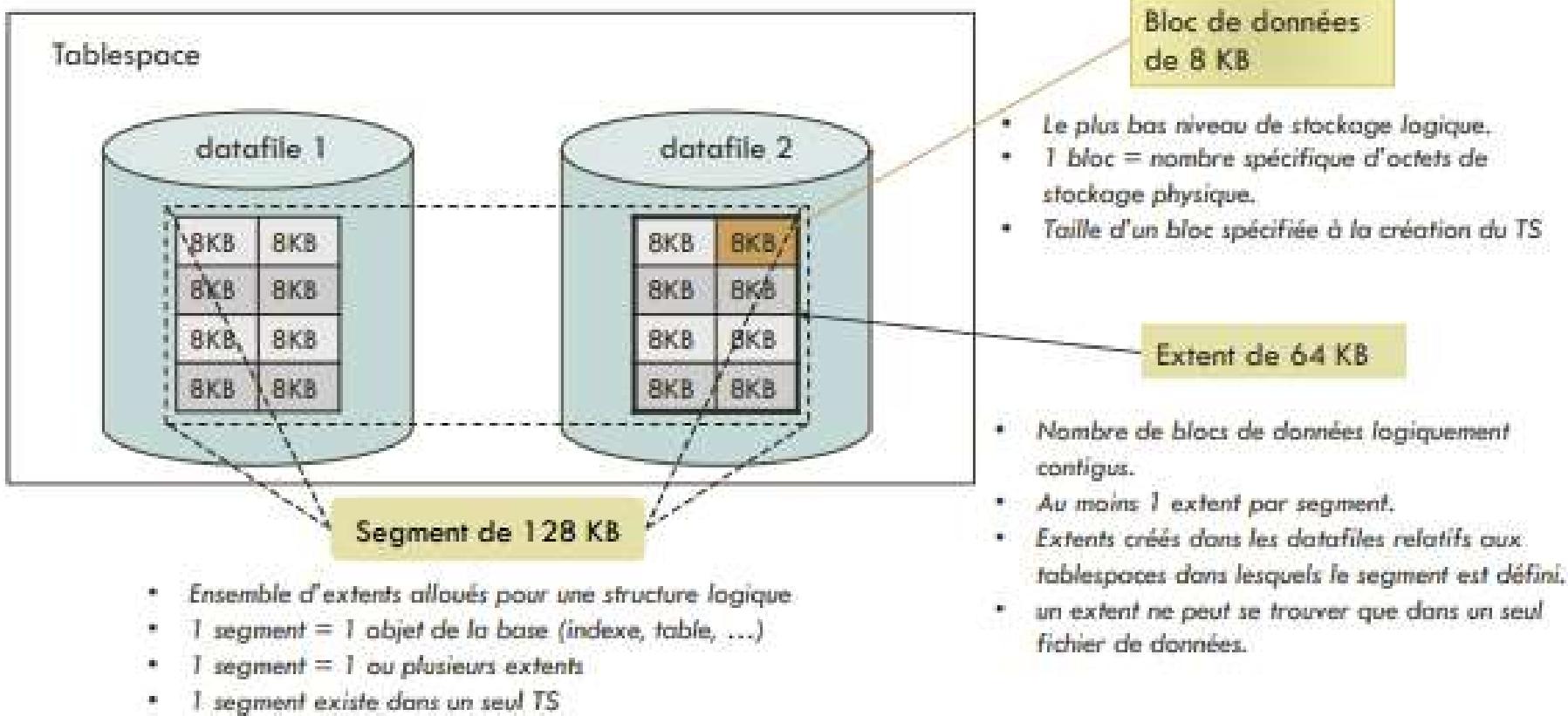
Un fichier de données (datafiles):

- Ne peut appartenir qu'à un tablespace
- Et qu'à une seule base de données
- Est créé pour stocker physiquement les données des segments (objets) d'un tablespace.
- Peut être physiquement stocké dans tout type de stockage supporté par Oracle



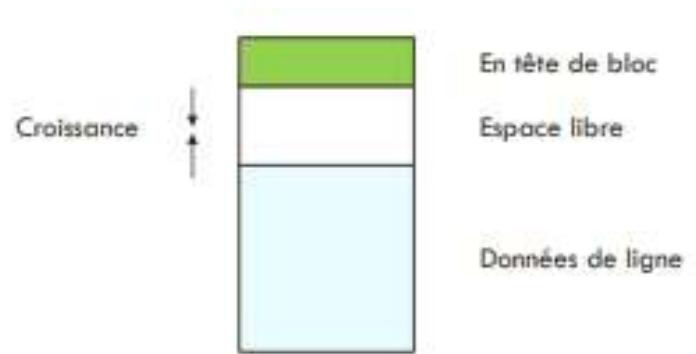
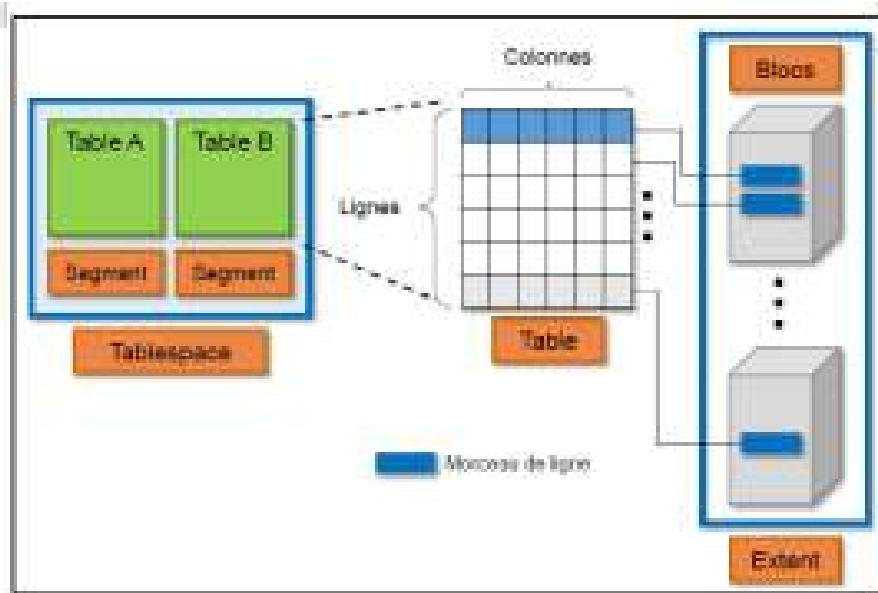
TABLESPACES ET FICHIERS DE DONNÉES

STRUCTURES LOGIQUE ET PHYSIQUE



TABLESPACES ET FICHIERS DE DONNÉES

STOCKAGE DES DONNÉES D'UNE TABLE DANS UN BLOC



La création d'une table implique la création d'un segment.

Une table est composée de lignes.

Chaque morceau de ligne est stocké dans un bloc de données (la ligne peut être coupée si sa taille > taille du bloc de données ou si une mise à jour des données augmente sa taille initiale).

L'entête du bloc contient, entre autres, le type du segment, l'adresse du bloc, le répertoire de la table et le répertoire de la ligne.

Les données de ligne représentent le stockage réel des lignes contenues dans le bloc.

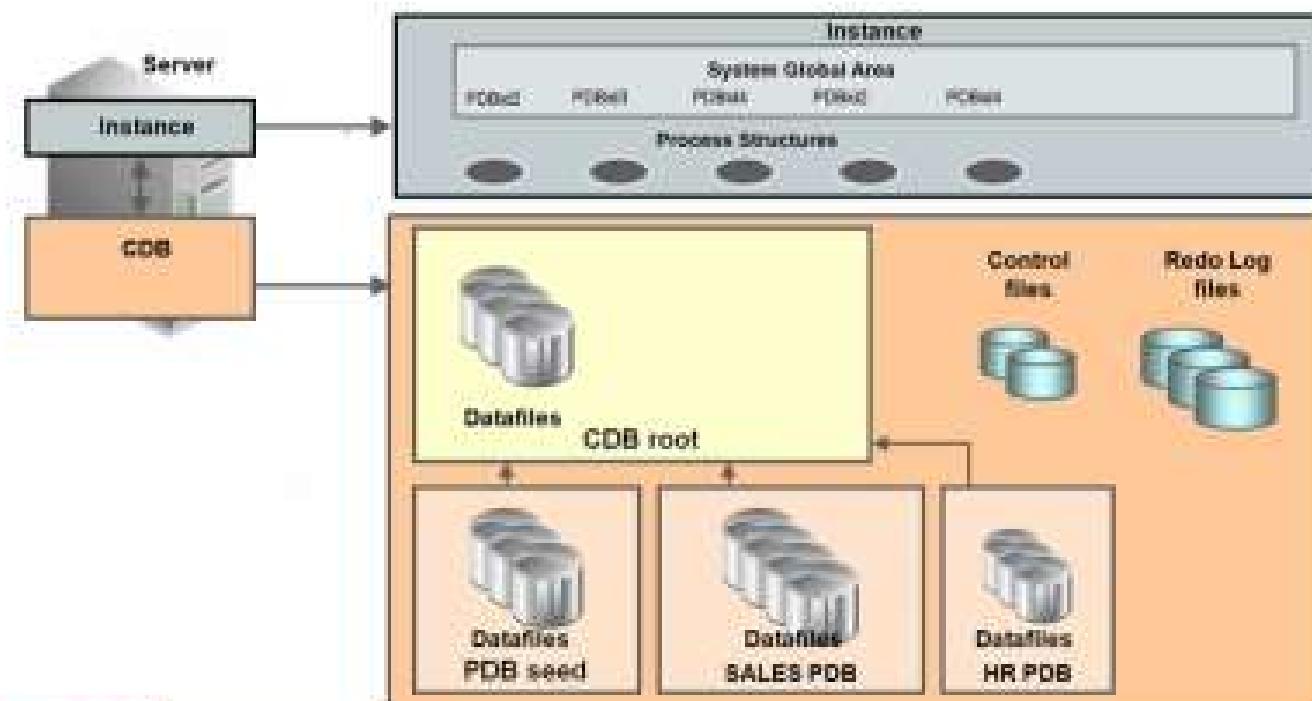
Espace libre : Il se situe au milieu du bloc et permet l'extension des espaces d'entête et de données si nécessaire.

PLAN DE LA SÉANCE

- Structures logiques et physiques de la base de données Oracle
- **Informations sur les tablespaces et datafiles**
- Création des tablespaces
- Gestion des tablespaces

RAPPEL ARCHITECTURE GLOBALE ORACLE DATABASE 19C

Oracle Multitenant Container Database: Architecture



ORACLE

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

ORACLE
UNIVERSITY

TABLESPACES PAR DÉFAUT DANS UN MULTITENANT CDB

Tablespace	Description
SYSTEM	<p>Créé en même temps que la base de données.</p> <p>Il reste toujours en ligne (accessible) lorsque la BD est ouverte.</p> <p>Au niveau du root, stocke les métadonnées du système, au niveau PDB stocke les métadonnées de l'utilisateur.</p>
SYSAUX	<p>Créé en même temps que la base de données.</p> <p>Il reste toujours en ligne lorsque la BD est ouverte.</p> <p>Il stocke des métadonnées auxiliaires et non cruciale pour le fonctionnement de la BD pour soulager le tablespace SYSTEM.</p> <p>Existe au niveau du root et des PDBs.</p>
UNDO	<p>Il stocke d'une manière temporaire les informations d'annulation (rollback).</p> <p>Il permet lors d'une récupération de base de données, d'annuler des transactions non validées par les utilisateurs.</p> <p>Existe au niveau du root.</p> <p>Il est recommandé d'avoir un UNDO local au niveau des PDBs.</p>
TEMP	<p>Les objets dans le TS TEMP sont stockés dans des fichiers temporaires.</p> <p>Il est utilisé lorsque l'exécution d'une instruction SQL requiert une zone de travail temporaire (exemple: tri, création d'indexe, ...)</p> <p>Un TS temporaire par utilisateur.</p>
USERS	<p>Stocke les objets et les données relatifs aux utilisateurs.</p> <p>Utilisé par défaut pour stocker les objets créés par l'utilisateur si pas de TS spécifique désigné.</p>

INFORMATIONS SUR LES TABLESPACES ET DATAFILES

INFORMATIONS SUR LES TABLESPACES

dba_tablespaces: (vue statique du dictionnaire des données) décrit tous les TS de la base de données (nom du TS, taille du bloc, status online/offline/Read only, bigfile/smallfile...)

```
SQL > select tablespace_name, block_size, contents, logging from dba_tablespaces;
```

cdb_tablespaces

v\$Tablespace: (vue dynamique de performance) affiche, à partir du [fichier de contrôle](#), les informations sur les tablespaces existant (numéro du TS, nom, le TS est un bigfile ou smallfile, ID du conteneur auquel le tablespace appartient...)

```
SQL > select * from v$Tablespace;
```

INFORMATIONS SUR LES TABLESPACES ET DATAFILES

INFORMATIONS SUR LES DATAFILES

dba_data_files: (vue statique du dictionnaire des données) décrit les fichiers de la base de données (nom, TS d'appartenance, taille en octet, taille en blocs Oracle, ...)

```
SQL > select file_name, tablespace_name, blocks from dba_data_files;
```

FILE_NAME	TABLESPACE_NAME	BLOCKS
C:\APP\ORACLE\ORADATA\ORCL\USERS01.DBF	USERS	6400
C:\APP\ORACLE\ORADATA\ORCL\UNDOTBS01.DBF	UNDOTBS1	7680
C:\APP\ORACLE\ORADATA\ORCL\SYSAUX01.DBF	SYSAUX	65280
C:\APP\ORACLE\ORADATA\ORCL\SYSTEM01.DBF	SYSTEM	109600
C:\APP\ORACLE\ORADATA\ORCL\EXAMPLE01.DBF	EXAMPLE	409600

cdb_data_files

v\$logfile: (vue dynamique de performance) affiche, à partir du **fichier de contrôle**, les informations sur les datafiles

INFORMATIONS SUR LES TABLESPACES ET DATAFILES

INFORMATIONS SUR LES FICHIERS TEMPORAIRES

dba_temp_files: (vue statique du dictionnaire des données) décrit les fichiers temporaires de la base de données

```
SQL> select file_name, tablespace_name, blocks from dba_temp_files;
```

```
SQL> select file_name, tablespace_name, blocks from dba_temp_files;
FILE_NAME          TABLESPACE_NAME      BLOCKS
C:\APP\ORACLE\ORADATA\ORCL\TEMP01.DBF    TEMP          2560
```

cdb_temp_files

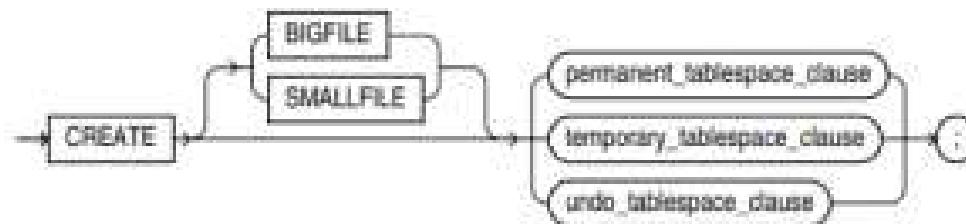
v\$tempfile: (vue dynamique de performance) affiche , à partir du **fichier de contrôle**, les informations sur les fichiers temporaires.

PLAN DE LA SÉANCE

- Structures logiques et physiques de la base de données Oracle
- Informations sur les tablespaces et datafiles
- **Création des tablespaces**
- Gestion des tablespaces

CRÉATION DES TABLESPACES

BIGFILE & SMALLFILE



Tablespace SMALLFILE:

Peut contenir plusieurs datafiles ou fichiers temporaires.
Fichiers moins volumineux que dans un tablespace bigfile.
C'est le type de tablespace par défaut.

Création d'un tablespace SMALLFILE

```
CREATE TABLESPACE tbs_01 DATAFILE  
'E:/app/poste/oradata/tbs_01.dbf'  
SIZE 100M
```

Tablespace BIGFILE:

Contient un seul grand datafile

Création d'un tablespace BIGFILE

```
CREATE BIGFILE TABLESPACE bigtbs_01  
DATAFILE  
'E:/app/poste/oradata/bigtbs_01.dbf'  
SIZE 1T
```

CRÉATION DES TABLESPACES

TABLESPACES PERMANENTS

```
CREATE [BIGFILE/SMALLFILE]
TABLESPACE tablespace
[DATAFILE clause]
[MINIMUM EXTENT integer [K|M]]
[BLOCKSIZE integer [K]]
[logging_clause]
[FORCE LOGGING]
[table_encryption_clause]
[default_table_space_params]
[ONLINE|OFFLINE]
[extent_management_clause]
[segment_management_clause]
[flashback_mode_clause]
[lost_write_protection]
```

MINIMUM EXTENT: Utilisé uniquement dans le cas des dictionary-managed tablespaces. Garantit que la taille de chaque extent du tablespace est un multiple de « integer ». Avec K/M permet de définir la taille en kilo-octets ou en méga-octets.

BLOCKSIZE: pour spécifier une taille de block non standard.

Logging_clause: pour spécifier si certaines opérations LMD seront écrite dans les redo log files (logging) ou non (nologging). Valeur par défaut: logging.

FORCE LOGGING: Pour forcer l'écriture de tous les changements sur les objets du tablespace au niveau des logs.

DEFAULT définit les paramètres par défaut du tablespace.

OFFLINE rend le tablespace indisponible dès sa création

CRÉATION DES TABLESPACES

TABLESPACES PERMANENTS/GESTION DES EXTENTS

```
CREATE [BIGFILE/SMALLFILE]
TABLESPACE tablespace
[DATAFILE clause]
[MINIMUM EXTENT integer [K|M]]
[BLOCKSIZE integer [K]]
[logging_clause]
[FORCE LOGGING]
[table_encryption_clause]
[default_table_space_params]
[ONLINE|OFFLINE]
[extent_management_clause]
[segment_management_clause]
[flashback_mode_clause]
[lost_write_protection]
```

Deux façons de gérer les extents dans le cas d'un locally-managed tablespace :

- Mode **UNIFORM SIZE**: impose à Oracle de créer des extents de taille identique.

```
CREATE TABLESPACE tbs_01 DATAFILE
'E:/app/poste/oradata/tbs_01.dbf' SIZE 100M
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K;
```

- Mode **AUTOALLOCATE**: crée des extents dont la taille augmente avec le nombre d'extents créés (les 16 premiers extents font 64k, les 64 suivants 1024k, etc.)

```
CREATE TABLESPACE tbs_01 DATAFILE
'E:/app/poste/oradata/tbs_01.dbf' SIZE 100M
EXTENT MANAGEMENT LOCAL AUTOALLOCATE;
```

CRÉATION DES TABLESPACES

TABLESPACES PERMANENTS/GESTION DES SEGMENTS

```
*CREATE [BIGFILE/SMALLFILE]
TABLESPACE tablespace
[DATAFILE clause]
[MINIMUM EXTENT integer [K|M]]
[BLOCKSIZE integer [K]]
[logging_clause]
[FORCE LOGGING]
[table_encryption_clause]
[default_table_space_params]
[ONLINE|OFFLINE]
[extent_management_clause]
[segment_management_clause]
[flashback_mode_clause]
[lost_write_protection]
```

Deux façons de gérer les segments dans le cas d'un locally-managed tablespace :

- Mode **AUTO**: utilise la méthode bitmaps pour gérer efficacement l'espace vide dans les segments

```
CREATE TABLESPACE tbs_01 DATAFILE
'E:/app/poste/oradata/tbs_01.dbf' SIZE 100M
EXTENT MANAGEMENT LOCAL
SEGMENT SPACE MANAGEMENT AUTO;
```

- Mode **MANUAL**: linked lists qui est une méthode moins performante pour gérer l'espace des segments

```
CREATE TABLESPACE tbs_01 DATAFILE
'E:/app/poste/oradata/tbs_01.dbf' SIZE 100M
EXTENT MANAGEMENT LOCAL
SEGMENT SPACE MANAGEMENT MANUAL;
```

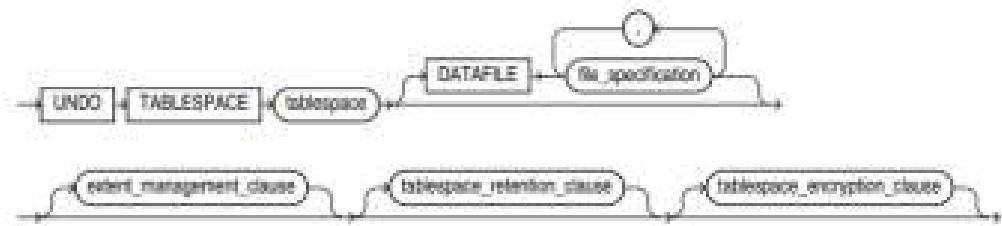
CRÉATION DES TABLESPACES

TABLESPACES D'ANNULATION (UNDO)

- ❑ Oracle stocke temporairement les données en cours de modifications dans des segments d'annulations. Ces segments d'annulations sont en attente de validation ou d'annulation (COMMIT ou ROLLBACK).

- ❑ Un tablespace d'annulation :

- Permet de stocker des segments d'annulation
- Ne peut contenir aucun autre objet

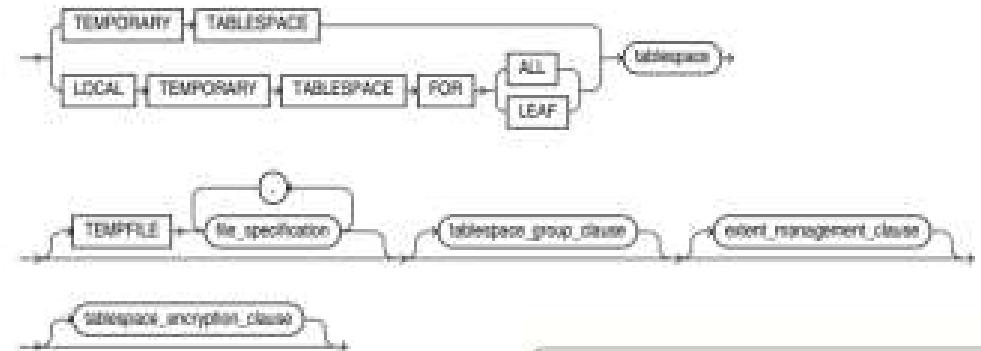


```
CREATE UNDO TABLESPACE undo01 DATAFILE  
'E:/app/poste/oradata/undo01.dbf' SIZE 40M
```

CRÉATION DES TABLESPACES

TABLESPACES TEMPORAIRES

- Un tablespace temporaire contient des **données temporaires** qui durent le **temps d'une session**.
- Aucun objet de schéma permanent ne peut résider dans un tablespace temporaire.
- Les tablespaces temporaires permettent d'**améliorer les performances** lorsque plusieurs **tris simultanés** ne peuvent tenir dans la mémoire.
- La taille du **segment de tri** augmente par **allocation d'extents** jusqu'à ce qu'elle soit égale ou supérieure au nombre **total des demandes de stockage** de tous les tris en cours d'exécution dans l'instance.



ATTENTION: pour un TS temporaire, il faut mettre **TEMPFILE** et non DATAFILE

```
CREATE TEMPORARY TABLESPACE temp01 TEMPFILE  
'E:/app/poste/oradata/temp01.dbf' SIZE 40M  
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 4M;
```

CRÉATION DES TABLESPACES

TABLESPACES TEMPORAIRES

- Les tablespaces temporaires (Temporary) contiennent des fichiers de données temporaires (tempfiles) semblables aux fichiers de données standard, sauf que :
 - Les fichiers temporaires sont toujours en mode NOLOGGING
 - Les fichiers temporaires ne peuvent pas être en lecture seule
 - Il n'est pas possible de renommer un fichier temporaire
 - Il n'est pas possible de créer un fichier temporaire à l'aide de la commande ALTER DATABASE
 - La restauration physique ne permet pas de restaurer les fichiers temporaires

Tablespace permanent → DATAFILES

Tablespace temporaire → TEMPFILES

CRÉATION DES TABLESPACES

TABLESPACES TEMPORAIRES

- Si la base de données ne contient pas un Tablespace temporaire par défaut, alors le tablespace SYSTEM va jouer le rôle d'un tablespace temporaire.
- Pour définir le tablespace par défaut:

```
ALTER DATABASE DEFAULT TEMPORARY TABLESPACE temp;
```

- Les DEFAULT TEMPORARY TABLESPACES ne peuvent pas être :
 - Supprimés tant qu'un nouveau tablespace par défaut n'est pas disponible
 - Mis offline (inaccessibles)
 - Transformés en tablespaces permanents

PLAN DE LA SÉANCE

- Structures logiques et physiques de la base de données Oracle
- Informations sur les tablespaces et datafiles
- Création des tablespaces
- Gestion des tablespaces**

GESTION DES TABLESPACES

ONLINE/OFFLINE

- Un tablespace est généralement en ligne, ce qui permet aux utilisateurs de la base d'accéder aux données qu'il contient.
- Le tablespace **SYSTEM** et les tablespaces **temporaires ne peuvent pas être mis OFFLINE**.
- Un tablespace peut être mis offline **manuellement** par le DBA, par exemple lors de certaines opérations de maintenance ou de sauvegarde/restauration, ou **automatiquement**, par exemple lorsque le process en arrière plan DBW n'arrive pas à écrire au niveau d'un datafile.
- Pour mettre un tablespace **offline**:

```
ALTER TABLESPACE userdata OFFLINE;
```

GESTION DES TABLESPACES

REDIMENSIONNEMENT

- Un tablespace peut être redimensionné en :
 1. Modifiant la taille d'un fichier de données automatiquement à l'aide de la clause **AUTOEXTEND**
 2. Modifiant la taille d'un fichier de données manuellement à l'aide de la commande **ALTER DATABASE**
 3. Ajoutant un fichier de données à l'aide de la commande **ALTER TABLESPACE**

GESTION DES TABLESPACES

REDIMENSIONNEMENT

1. En modifiant la taille d'un fichier de données automatiquement à l'aide de la clause **AUTOEXTEND**:

- La clause AUTOEXTEND permet d'activer ou de désactiver l'extension automatique des fichiers de données.
- La taille des fichiers est incrémentée jusqu'à la taille maximale fixée.
- Avantages de l'utilisation de la clause AUTOEXTEND :
 - Le besoin d'intervention immédiate est réduit lorsque l'espace disponible est insuffisant dans le tablespace
 - Les applications ne s'arrêtent plus à cause des pannes d'allocation des extents

```
CREATE TABLESPACE userdata01  
DATAFILE 'E:/app/poste/oradata/userdata01.dbf' SIZE 100M  
AUTOEXTEND ON NEXT 10M MAXSIZE 500M;
```

Le tablespace userdata01 a une taille initiale de 100M. Il sera incrémenté de 10M à chaque fois jusqu'à atteindre 500M maximum.

GESTION DES TABLESPACES

REDIMENSIONNEMENT

1. En modifiant la taille d'un fichier de données manuellement à l'aide de la commande **ALTER DATABASE**:

- Un fichier de données peut être redimensionné au lieu d'ajouter des fichiers de données
- La clause **RESIZE** permet d'augmenter ou de diminuer manuellement la taille d'un fichier de données.

```
ALTER DATABASE DATAFILE 'E:/app/poste/oradata/userdata01.dbf'
RESIZE 600M;
```

GESTION DES TABLESPACES

REDIMENSIONNEMENT

1. En ajoutant un fichier de données à l'aide de la commande **ALTER TABLESPACE**:

- Il est possible d'augmenter l'espace alloué à un tablespace en ajoutant des fichiers de données.
- La clause **ADD DATAFILE** permet d'ajouter un fichier de données.

```
ALTER TABLESPACE userdata ADD DATAFILE  
'E:/app/poste/oradata/userdata03.dbf' SIZE 200M;
```



Ceci est valable uniquement pour les tablespaces **SMALLFILE** (un tablespace **BIGFILE** ne peut contenir qu'un seul datafile)

GESTION DES TABLESPACES

DÉPLACER/RENOMMER DES DATAFILES

Les étapes permettant de déplacer/renommer un fichier de données:

1. Mettre le tablespace en mode **hors ligne**
2. Utiliser la commande appropriée du **système d'exploitation** pour **déplacer ou copier** les fichiers
3. S'assurer que les noms des fichiers correspondent à ceux définis dans le **fichier de contrôle**
4. Exécuter la commande **ALTER TABLESPACE RENAME DATAFILE**

```
ALTER TABLESPACE userdata RENAME DATAFILE  
'E:/app/poste/oradata/userdata01.dbf' TO  
'E:/app/poste/oradata/DiskTable/userdata01.dbf';
```

5. Mettre le tablespace **en ligne**
6. Au besoin, utiliser la commande appropriée du **système d'exploitation** pour **supprimer le fichier**

GESTION DES TABLESPACES

SUPPRIMER UN TABLESPACE

- Le tablespace SYSTEM ne peut pas être supprimé.
- L'option **INCLUDING CONTENTS** supprime les segments. Il faut l'ajouter lorsque le tablespace contient des données.
- L'option **INCLUDING CONTENTS AND DATAFILES** supprime les fichiers de données.

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

- L'option **CASCADE CONSTRAINTS** supprime les contraintes d'intégrité référentielle.
- Il est recommandé de mettre le tablespace **hors ligne avant de le supprimer**, afin d'empêcher d'éventuelles transactions d'accéder à ses segments