

Administration des bases de données

Chapitre 1 : Architecture du serveur de base de données Oracle

Ines BAKLOUTI

ines.baklouti@esprit.tn

Ecole Supérieure Privée d'Ingénierie et de Technologies



Plan

1 Architecture du serveur oracle

- Serveur de base de données oracle
- Composantes du serveur oracle
 - Instance oracle
 - Base de donnée oracle

2 Démarrage de la base de données oracle

- Fichiers de paramètres d'initialisation
- Démarrage de la base de données oracle

3 Arrêt de la base de données oracle

Rôles d'un administrateur de base de données

1 Installer le serveur de base de données et démarrer et arrêter la base et manipuler les paramètres d'initialisation

2 Créer et configurer les espaces de stockage physiques et logiques

3 Créer des utilisateurs (compte Oracle), leurs affecter des priviléges, rôles et des profils.

7

Assurer la sauvegarde et la récupération en cas de panne de la BD

6

Assurer l'export et l'import des données inter-base et transposer des données externes vers une base de données

5

Maintenir les ressources et assurer la continuité de fonctionnement et contrôler les indices de performance

4

Auditer et sécuriser la base de données (mouchards, interdictions d'accès et d'exécution, limitation de consommation des ressources, etc.)



Plan

1 Architecture du serveur oracle

- Serveur de base de données oracle
- Composantes du serveur oracle
 - Instance oracle
 - Base de donnée oracle

2 Démarrage de la base de données oracle

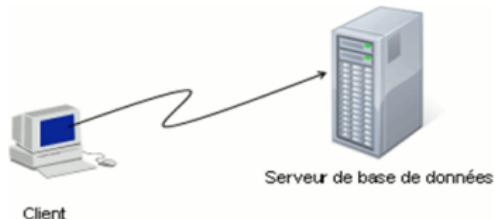
- Fichiers de paramètres d'initialisation
- Démarrage de la base de données oracle

3 Arrêt de la base de données oracle

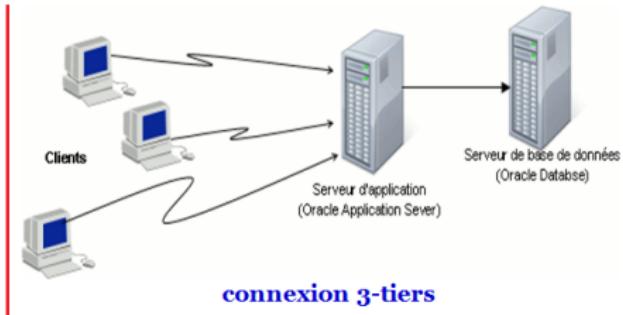
Serveur de base de données oracle

Serveur oracle

- est un système de gestion de base de données qui fournit une approche intégrée, complète et ouverte de la gestion des informations
- se compose d'une instance Oracle et des fichiers de base de données



connexion client/serveur



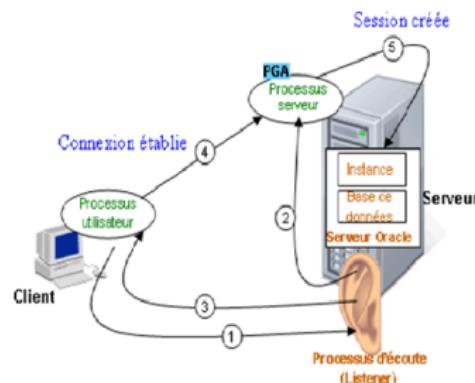
connexion 3-tiers

Connexion à la base de données oracle

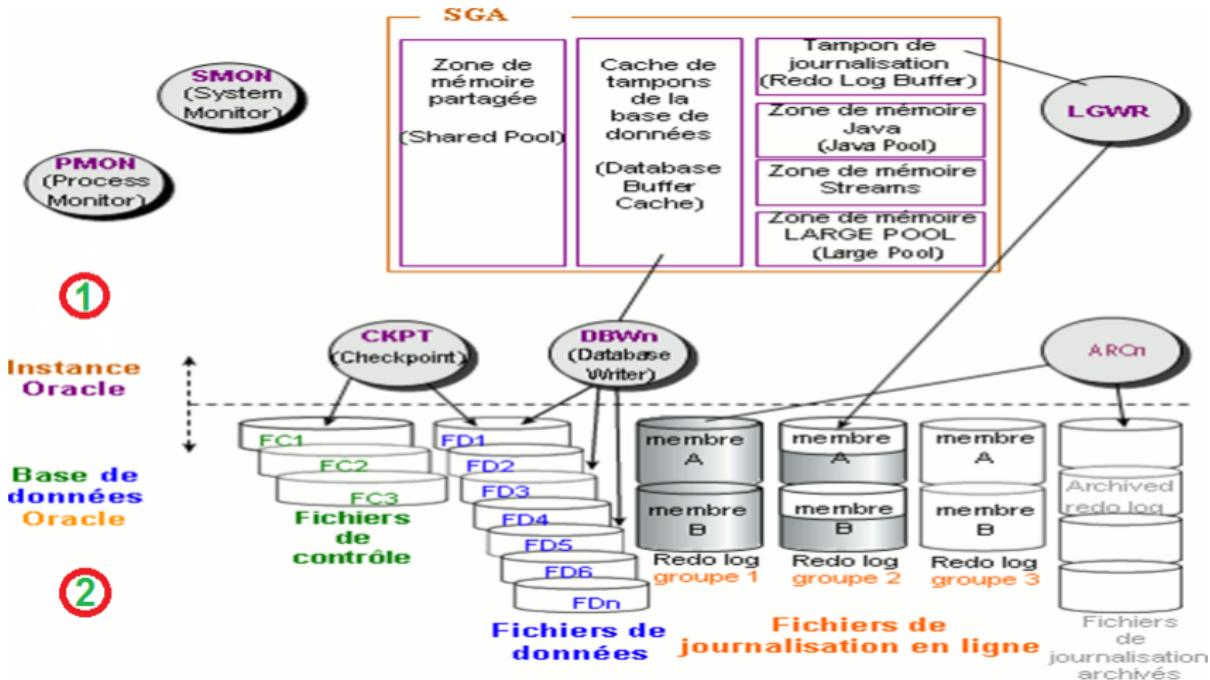
Scénario de connexion à une base de données oracle

- 1 Le client contacte le listener Oracle en choisissant l'instance à laquelle il souhaite se connecter (demande d'un nom de service).
- 2 Le listener démarre un processus dédié appelé processus serveur
- 3 Le listener envoie un accusé de réception au client avec l'adresse du processus serveur dédié
- 4 Le client établit une connexion avec le processus serveur dédié
- 5 Le processus serveur se connecte à l'instance Oracle pour le compte du processus utilisateur (création d'une session utilisateur)

⇒ Le processus utilisateur n'entre pas directement en interaction avec le serveur Oracle. C'est plutôt, le processus serveur qui interagit avec le serveur Oracle, répond aux demandes de l'utilisateur et lui renvoie les résultats.

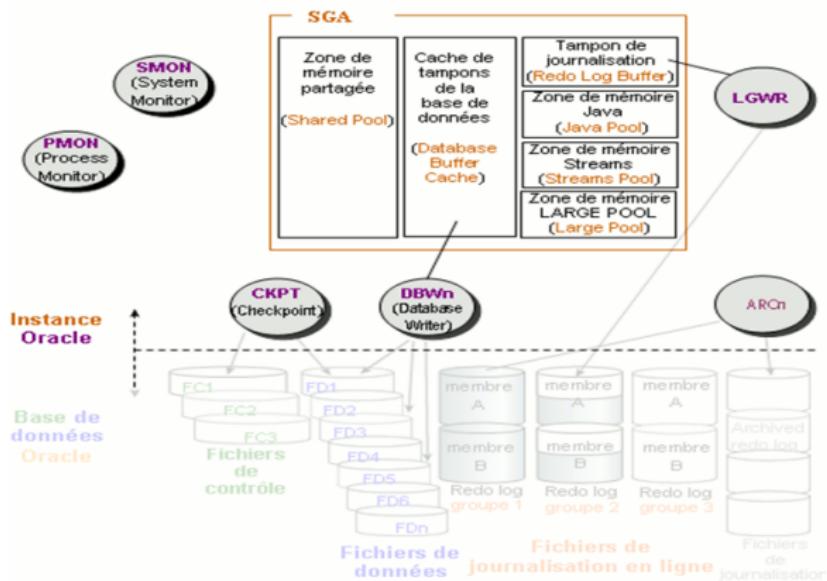


Composantes du serveur oracle



Un serveur Oracle se compose d'une **instance Oracle** et d'une **base de données Oracle**

Instance oracle



Une instance de la base de données oracle est composée d'une **structure de mémoire partagée SGA** (Système Global Area) et de plusieurs processus oracle en arrière-plan ayant chacun un rôle bien déterminé.

Instance oracle : La mémoire SGA

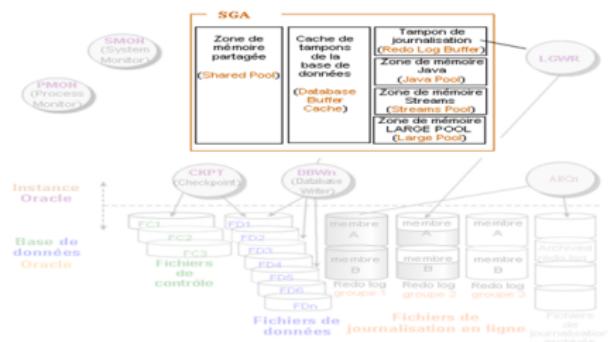
La mémoire SGA (System Global Area) est une mémoire partagée par tous les processus serveurs et les processus en arrière-plan.

■ Zones de mémoire obligatoires :

- Zone de mémoire partagée (Shared Pool)
- Cache de tampons de la base de données (database buffer cache)
- Tampon de journalisation (redo log buffer)

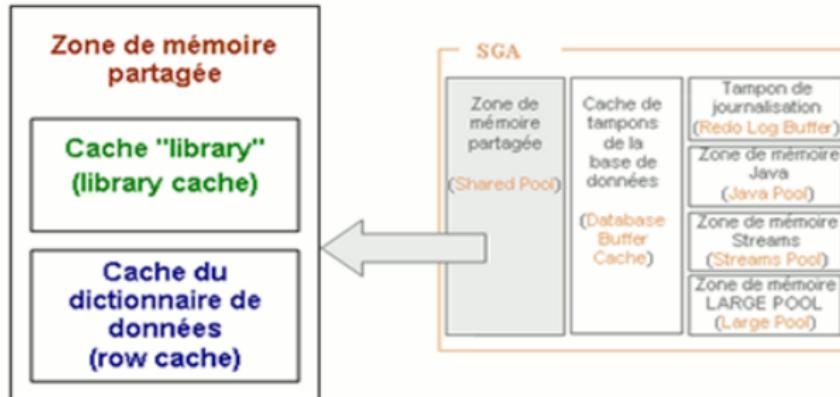
■ Zones mémoire non obligatoires :

- Zone de mémoire LARGE POOL
- Zone de mémoire Java (Java Pool)
- Zone de mémoire streams (streams pool)



Instance oracle : La mémoire SGA

Shared Pool



La zone de mémoire partagée Shared Pool est constituée de deux structures mémoire liées aux performances :

- Cache du dictionnaire de données (row cache)
- Cache "library"

Instance oracle : La mémoire SGA

Shared Pool : Cache du dictionnaire de données

- Lorsqu'un utilisateur soumet une requête SQL , le processus serveur extrait, au cours de l'analyse de la requête, des informations stockées dans les tables du dictionnaire (données du compte utilisateur, noms des fichiers de données, noms des segments de tables et index, emplacements d'extents, descriptions des tables et privilèges utilisateur, etc).
- Ces informations sont placées dans le cache du dictionnaire pour des besoins de réutilisation. Au cours des prochaines analyses, le processus serveur recherche les informations dans le cache du dictionnaire pour résoudre les noms d'objet et valider l'accès.
- La mise en mémoire cache des informations du dictionnaire de données réduit le temps de réponse aux instructions LID et LMD (SELECT, INSERT, UPDATE, DELETE). Une taille suffisante de ce cache contribue considérablement à l'amélioration des performances.

Instance oracle : La mémoire SGA

Shared Pool : Library Cache

- Le cache "library" conserve, à des fins de partage, des informations sur les commandes SQL et le code PL/SQL les plus récemment utilisés qui ont été soumis par des utilisateurs de la base de données.
- Le chargement d'une instruction SQL consomme beaucoup de ressources, c'est pourquoi le cache « library » partagé est utilisé pour optimiser et ne charger une instruction SQL qu'une seule fois pour de multiples exécutions.

Instance oracle : La mémoire SGA

Cache de tampons de la base de données

- Ce cache conserve des copies des blocs de données extraits des fichiers de données.
- Il permet des gains de performances considérables lors de l'obtention et de la mise à jour de données.
- Il est géré par un algorithme LRU.
- Le paramètre DB_BLOCK_SIZE détermine la taille du bloc principal.

Instance oracle : La mémoire SGA

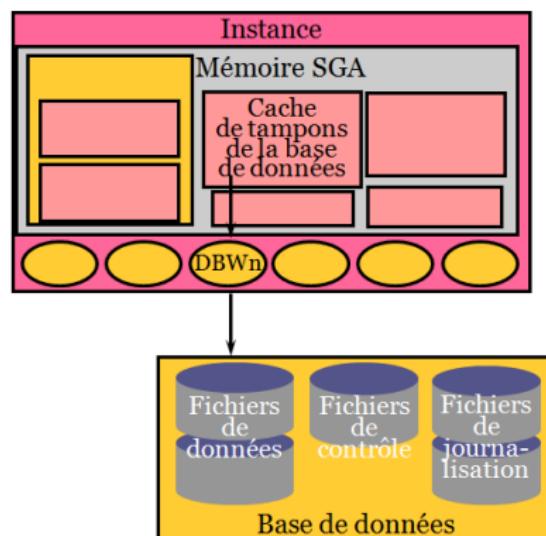
Tampon de journalisation

- Il enregistre toutes les modifications apportées aux blocs de données de la base.
- Sa principale fonction est la récupération de données.
- Les modifications enregistrées constituent des entrées de journalisation.
- La taille du tampon est définie par le paramètre LOG_BUFFER.

Instance oracle : Les processus en arrière plan

Le processus DBWn

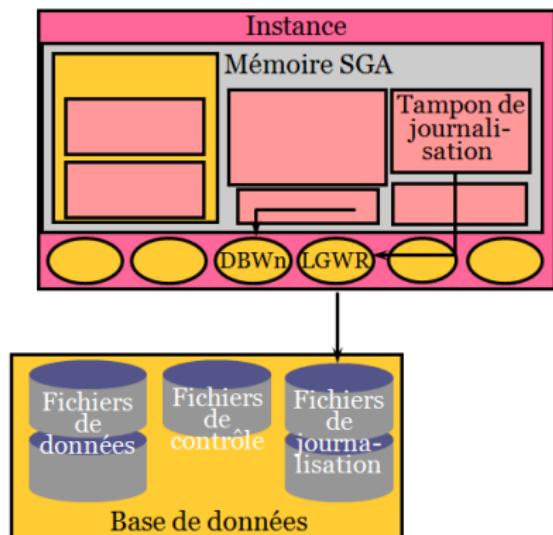
- DBWn (Database Writer) : écrit les blocs modifiés du cache tampon de base de données vers les fichiers de données.
- DBWn écrit dans les cas suivants :
 - point de reprise
 - seuil des tampons "dirty" atteint
 - aucune mémoire tampon disponible
 - temps imparti dépassé
 - demande de ping RAC
 - tablespace hors ligne
 - tablespace en lecture seule
 - DROP ou TRUNCATE sur une table
 - BEGIN BACKUP sur un tablespace



Instance oracle : Les processus en arrière plan

Le processus LGWR

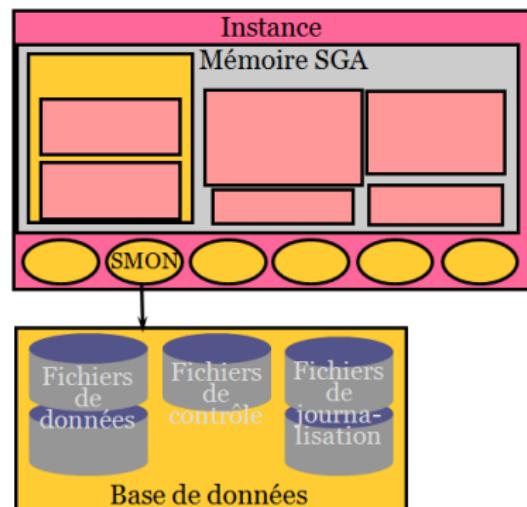
- LGWR (Log Writer) : écrit les entrées de journalisation sur le disque.
- LGWR écrit dans les cas suivants :
 - validation
 - un tiers du cache est occupé
 - la journalisation atteint 1 Mo
 - toutes les trois secondes avant que le processus DBWn ne procède à une opération d'écriture



Instance oracle : Les processus en arrière plan

Le processus SMON

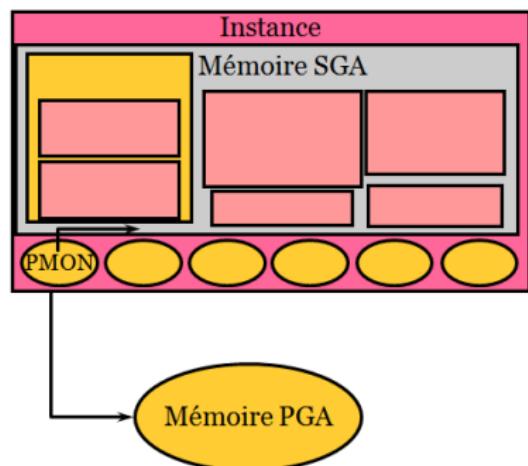
- SMON (System Monitor) : assure la récupération en cas de panne.
- Responsabilités :
 - Récupération de l'instance :
 - réimplémente des modifications dans les fichiers de journalisation,
 - ouvre la base de données pour permettre l'accès aux utilisateurs,
 - annule les transactions non validées.
 - Fusion de l'espace libre
 - Libération des segments temporaires



Instance oracle : Les processus en arrière plan

Le processus PMON

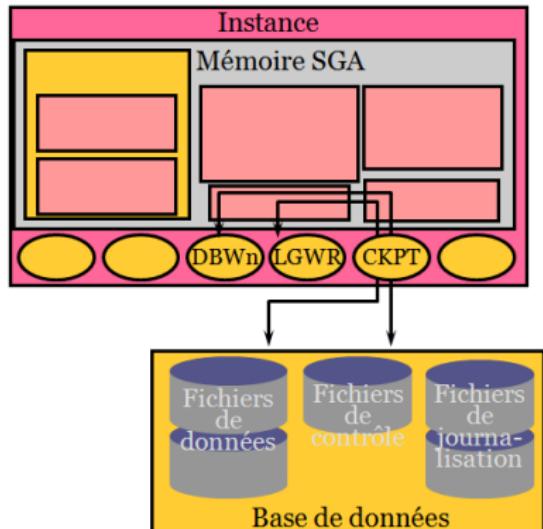
- PMON (Process Monitor) : assure le nettoyage des processus utilisateur en cas d'échec (session abandonnée) :
 - annule la transaction
 - libère des verrous
 - libère d'autres ressources
 - redémarre les répartiteurs interrompus



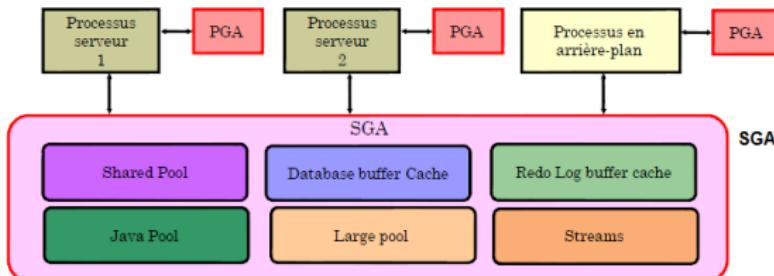
Instance oracle : Les processus en arrière plan

Le processus CKPT

- CKPT (Check Point) : informe le processus DBWn des points de reprise pour lancer la mise à jour.
- Ce processus est chargé de :
 - signaler DBWn aux points de reprise,
 - mettre à jour les en-têtes de fichiers de données avec les informations sur le point de reprise,
 - de mettre à jour les fichiers de contrôle avec les informations sur le point de reprise.

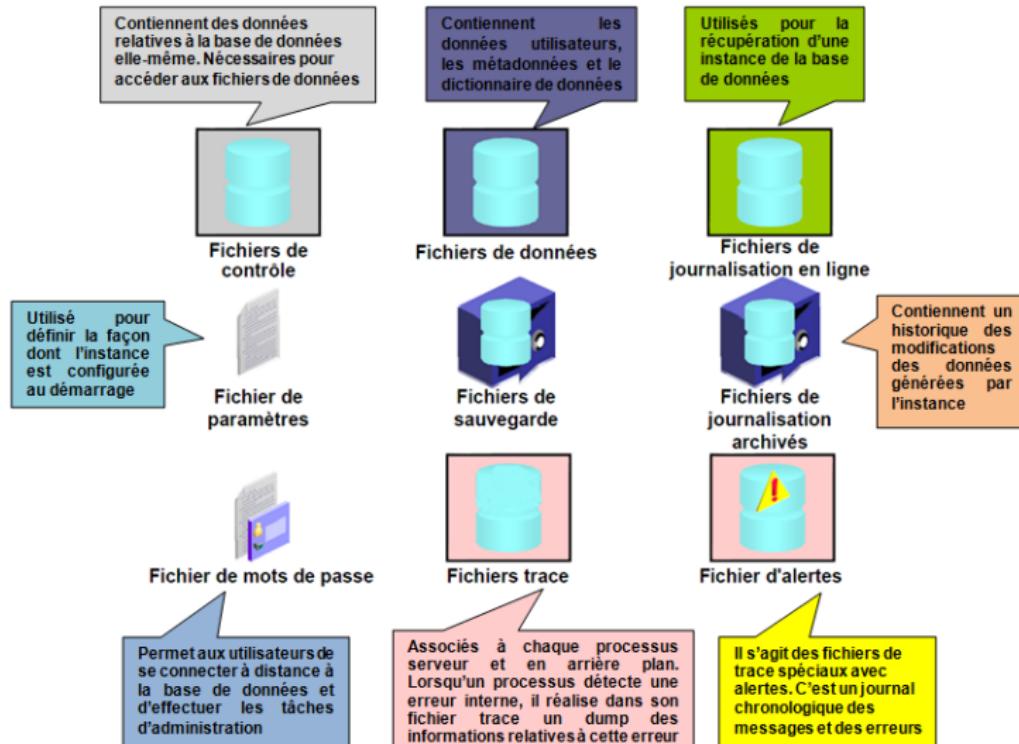


Instance oracle : La mémoire PGA



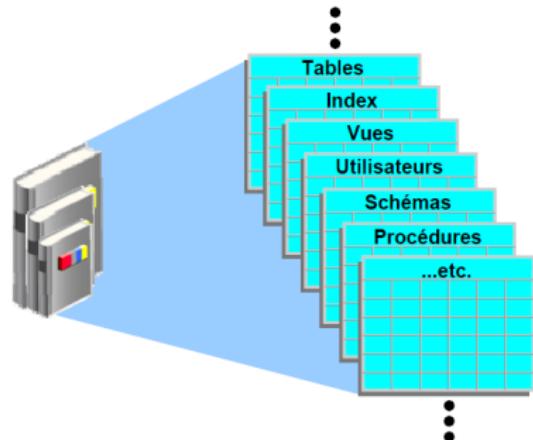
- Une structure de mémoire appelée PGA (Program Global Area) est créée pour chaque utilisateur connecté.
- La PGA stocke des informations de contrôle spécifiques à la session de l'utilisateur telles que les zones privées pour le traitement des curseurs, les variables attachées (bind), des informations sur la session, une zone de tri, etc.
- Chaque processus serveur et chaque processus en arrière plan dispose de sa propre mémoire PGA privée qui lui est exclusivement réservée.
- Lorsque le processus utilisateur se déconnecte (fin de session), le processus serveur associé prend fin et la mémoire PGA est libérée.

Base de donnée oracle



Base de donnée oracle : Le dictionnaire de données

- Le dictionnaire de données oracle est la description d'une base de données
- Il contient le nom et les attributs de tous les objets de la base
- Il est mis à jour à chaque création ou modification d'un objet
- Crée en même temps que la base de données
- Il appartient à l'utilisateur SYS
- Les informations du dictionnaire sont stockées dans des tables de base
- Vous n'avez pas un accès direct aux tables du dictionnaire, vous y accéder par le biais de vues prédéfinies



| Préfixe de la vue | Usage |
|-------------------|--|
| USER | Vue de l'utilisateur (ce que contient votre schéma, dont vous êtes propriétaire) |
| ALL | Vue étendue de l'utilisateur (ce à quoi vous pouvez accéder) |
| DBA | Vue de l'administrateur de base de données (ce que contient le schéma de chaque utilisateur) |
| V\$ | Données relatives aux performances |

Base de donnée oracle : Le dictionnaire de données

Vue dictionary

- Cette vue contient le nom et la description des tables et vues du dictionnaire.

DESCRIBE DICTIONARY

| Name | Null? | Type |
|------------|-------|----------------|
| TABLE_NAME | | VARCHAR2(30) |
| COMMENTS | | VARCHAR2(4000) |

```
SELECT *
FROM   dictionary
WHERE  table_name = 'USER_OBJECTS';
```

| TABLE_NAME | COMMENTS |
|--------------|---------------------------|
| USER_OBJECTS | Objects owned by the user |

Base de donnée oracle : Le dictionnaire de données

Vue user_objects

- Décrit tous les objets dont vous êtes propriétaire
- Permet d'afficher facilement le nom et le type de tous les objets de votre schéma, ainsi que les informations suivantes :
 - Date de création
 - Date de dernière modification
 - Statut (valide ou non valide)
- Vous pouvez interroger la vue ALL_OBJECTS afin d'afficher tous les objets auxquels vous avez accès

Base de donnée oracle : Le dictionnaire de données

Vue user_objects

```
SELECT object_name, object_type, created, status
FROM   user_objects
ORDER BY object_type;
```

| OBJECT_NAME | OBJECT_TYPE | CREATED | STATUS |
|------------------|-------------|-----------|--------|
| REG_ID_PK | INDEX | 10-DEC-03 | VALID |
| ... | | | |
| DEPARTMENTS SEQ | SEQUENCE | 10-DEC-03 | VALID |
| REGIONS | TABLE | 10-DEC-03 | VALID |
| LOCATIONS | TABLE | 10-DEC-03 | VALID |
| DEPARTMENTS | TABLE | 10-DEC-03 | VALID |
| JOB_HISTORY | TABLE | 10-DEC-03 | VALID |
| JOB_GRADES | TABLE | 10-DEC-03 | VALID |
| EMPLOYEES | TABLE | 10-DEC-03 | VALID |
| JOBS | TABLE | 10-DEC-03 | VALID |
| COUNTRIES | TABLE | 10-DEC-03 | VALID |
| EMP_DETAILS_VIEW | VIEW | 10-DEC-03 | VALID |

Base de donnée oracle : Le dictionnaire de données

Vue user_tables

- Vue user_tables : contient les informations relatives aux tables dont vous êtes propriétaire.

```
DESCRIBE user_tables
```

| Name | Null? | Type |
|-----------------|----------|--------------|
| TABLE_NAME | NOT NULL | VARCHAR2(30) |
| TABLESPACE_NAME | | VARCHAR2(30) |
| CLUSTER_NAME | | VARCHAR2(30) |
| IOT_NAME | | VARCHAR2(30) |

```
SELECT table_name  
FROM user_tables;
```

| TABLE_NAME |
|-------------|
| JOB_GRADES |
| REGIONS |
| COUNTRIES |
| LOCATIONS |
| DEPARTMENTS |
| ... |

Base de donnée oracle : Le dictionnaire de données

Vue user_tab_columns

- Vue user_tab_columns : contient les informations relatives aux colonnes des tables dont vous êtes propriétaire.

`DESCRIBE user_tab_columns`

| Name | Null? | Type |
|-----------------|----------|---------------|
| TABLE_NAME | NOT NULL | VARCHAR2(30) |
| COLUMN_NAME | NOT NULL | VARCHAR2(30) |
| DATA_TYPE | | VARCHAR2(106) |
| DATA_TYPE_MOD | | VARCHAR2(3) |
| DATA_TYPE_OWNER | | VARCHAR2(30) |
| DATA_LENGTH | NOT NULL | NUMBER |
| DATA_PRECISION | | NUMBER |
| DATA_SCALE | | NUMBER |
| NULLABLE | | VARCHAR2(1) |
| COLUMN_ID | | NUMBER |
| DEFAULT_LENGTH | | NUMBER |
| DATA_DEFAULT | | LONG |
| ... | | |

Base de donnée oracle : Le dictionnaire de données

Vue user_tab_columns

```
SELECT column_name, data_type, data_length,  
       data_precision, data_scale, nullable  
FROM   user_tab_columns  
WHERE  table_name = EMPLOYEES ;
```

| COLUMN_NAME | DATA_TYPE | DATA_LENGTH | DATA_PRECISION | DATA_SCALE | NUL |
|----------------|-----------|-------------|----------------|------------|-----|
| EMPLOYEE_ID | NUMBER | 22 | 6 | 0 | N |
| FIRST_NAME | VARCHAR2 | 20 | | | Y |
| LAST_NAME | VARCHAR2 | 25 | | | N |
| EMAIL | VARCHAR2 | 25 | | | N |
| PHONE_NUMBER | VARCHAR2 | 20 | | | Y |
| HIRE_DATE | DATE | 7 | | | N |
| JOB_ID | VARCHAR2 | 10 | | | N |
| SALARY | NUMBER | 22 | 8 | 2 | Y |
| COMMISSION_PCT | NUMBER | 22 | 2 | 2 | Y |
| MANAGER_ID | NUMBER | 22 | 6 | 0 | Y |
| DEPARTMENT_ID | NUMBER | 22 | 4 | 0 | Y |

Base de donnée oracle : Le dictionnaire de données

Vue user_constraints

- Vue user_constraints : décrit les définitions de contraintes sur les tables.

```
DESCRIBE user_constraints
```

| Name | Null? | Type |
|-------------------|----------|--------------|
| OWNER | NOT NULL | VARCHAR2(30) |
| CONSTRAINT_NAME | NOT NULL | VARCHAR2(30) |
| CONSTRAINT_TYPE | | VARCHAR2(1) |
| TABLE_NAME | NOT NULL | VARCHAR2(30) |
| SEARCH_CONDITION | | LONG |
| R_OWNER | | VARCHAR2(30) |
| R_CONSTRAINT_NAME | | VARCHAR2(30) |
| DELETE_RULE | | VARCHAR2(9) |
| STATUS | | VARCHAR2(8) |
| ... | | |

Base de donnée oracle : Le dictionnaire de données

Vue user_constraints

```
SELECT constraint_name, constraint_type,  
       search_condition, r_constraint_name,  
       delete_rule, status  
FROM   user_constraints  
WHERE  table_name = EMPLOYEES ;
```

| CONSTRAINT_NAME | CON | SEARCH_CONDITION | R_CONSTRAINT_NAME | DELETE_RULE | STATUS |
|------------------|-----|-------------------------|-------------------|-------------|---------|
| EMP_LAST_NAME_NN | C | "LAST_NAME" IS NOT NULL | | | ENABLED |
| EMP_EMAIL_NN | C | "EMAIL" IS NOT NULL | | | ENABLED |
| EMP_HIRE_DATE_NN | C | "HIRE_DATE" IS NOT NULL | | | ENABLED |
| EMP_JOB_NN | C | "JOB_ID" IS NOT NULL | | | ENABLED |
| EMP_SALARY_MIN | C | salary > 0 | | | ENABLED |
| EMP_EMAIL_UK | U | | | | ENABLED |
| EMP_EMP_ID_PK | P | | | | ENABLED |
| EMP_DEPT_FK | R | | DEPT_ID_PK | NO ACTION | ENABLED |
| EMP_JOB_FK | R | | JOB_ID_PK | NO ACTION | ENABLED |
| EMP_MANAGER_FK | R | | EMP_EMP_ID_PK | NO ACTION | ENABLED |

Base de donnée oracle : Le dictionnaire de données

Vue user_cons_columns

- Vue user_cons_columns : décrit les colonnes dont vous êtes propriétaire et qui sont définies dans des contraintes.

```
DESCRIBE user_cons_columns
```

| Name | Null? | Type |
|-----------------|----------|----------------|
| OWNER | NOT NULL | VARCHAR2(30) |
| CONSTRAINT_NAME | NOT NULL | VARCHAR2(30) |
| TABLE_NAME | NOT NULL | VARCHAR2(30) |
| COLUMN_NAME | | VARCHAR2(4000) |
| POSITION | | NUMBER |

```
SELECT constraint_name, column_name
FROM   user_cons_columns
WHERE  table_name = 'EMPLOYEES' ;
```

| CONSTRAINT_NAME | COLUMN_NAME |
|------------------|-------------|
| EMP_EMAIL_UK | EMAIL |
| EMP_SALARY_MIN | SALARY |
| EMP_JOB_NN | JOB_ID |
| EMP_HIRE_DATE_NN | HIRE_DATE |

...

Base de donnée oracle : Le dictionnaire de données

Vue user_views

- Vue user_views : contient les informations relatives aux vues dont vous êtes propriétaire.

1

DESCRIBE user_views

| Name | Null? | Type |
|-------------|----------|--------------|
| VIEW_NAME | NOT NULL | VARCHAR2(30) |
| TEXT_LENGTH | | NUMBER |
| TEXT | | LONG |

2

SELECT DISTINCT view_name FROM user_views;

VIEW_NAME

EMP_DETAILS_VIEW

3

SELECT text FROM user_views
WHERE view_name = EMP_DETAILS_VIEW ;

TEXT

SELECT e.employee_id, e.job_id, e.manager_id, e.department_id, d.location_id, l.country_id,
e.first_name, e.last_name, e.salary, e.commission_pct, d.department_name, j.job_title, l.city,
l.state_province, c.country_name, r.region_name FROM employees e, departments d, jobs j, locations l,
countries c, regions r WHERE e.department_id = d.department_id AND d.location_id = l.location_id AND
l.country_id = c.country_id AND c.region_id = r.region_id AND j.job_id = e.job_id WITH READ ONLY

Base de donnée oracle : Le dictionnaire de données

Vue user_sequences

- Vue user_sequences : contient les informations relatives aux séquences dont vous êtes propriétaire.

```
DESCRIBE user_sequences
```

| Name | Null? | Type |
|---------------|----------|--------------|
| SEQUENCE_NAME | NOT NULL | VARCHAR2(30) |
| MIN_VALUE | | NUMBER |
| MAX_VALUE | | NUMBER |
| INCREMENT_BY | NOT NULL | NUMBER |
| CYCLE_FLAG | | VARCHAR2(1) |
| ORDER_FLAG | | VARCHAR2(1) |
| CACHE_SIZE | NOT NULL | NUMBER |
| LAST_NUMBER | NOT NULL | NUMBER |

Base de donnée oracle : Le dictionnaire de données

Vue user_sequences

```
SELECT sequence_name, min_value, max_value,  
       increment_by, last_number  
FROM   user_sequences;
```

| SEQUENCE_NAME | MIN_VALUE | MAX_VALUE | INCREMENT_BY | LAST_NUMBER |
|-----------------|-----------|------------|--------------|-------------|
| LOCATIONS_SEQ | 1 | 9900 | 100 | 3300 |
| DEPARTMENTS_SEQ | 1 | 9990 | 10 | 280 |
| EMPLOYEES_SEQ | 1 | 1.0000E+27 | 1 | 207 |

- La colonne LAST_NUMBER affiche le prochain numéro de séquence disponible.

Base de donnée oracle : Le dictionnaire de données

Vue user_synonyms

- Vue user_synonyms : contient les informations relatives aux synonymes dont vous êtes propriétaire.

```
DESCRIBE user_synonyms
```

| Name | Null? | Type |
|--------------|----------|---------------|
| SYNONYM_NAME | NOT NULL | VARCHAR2(30) |
| TABLE_OWNER | | VARCHAR2(30) |
| TABLE_NAME | NOT NULL | VARCHAR2(30) |
| DB_LINK | | VARCHAR2(128) |

```
SELECT *  
FROM user_synonyms;
```

| SYNONYM_NAME | TABLE_OWNER | TABLE_NAME | DB_LINK |
|--------------|-------------|------------|---------|
| EMP | ORA1 | EMPLOYEES | |

Base de donnée oracle : Le dictionnaire de données

Vues V\$

- Vues dynamiques :

- Enregistrent l'activité en cours de la base de données
- Constantement mises à jour lorsque la base de données est active
- Les informations sont lues à partir de la mémoire et du fichier de contrôle
- Commencent par le prefixe v\$
- Accessible par un dba

- Exemples :

- V\$session : affiche les sessions en cours
- V\$logfile : affiche la liste des fichiers journaux
- V\$log : affiche le statut des groupes de fichiers journaux

Plan

1 Architecture du serveur oracle

- Serveur de base de données oracle
- Composantes du serveur oracle
 - Instance oracle
 - Base de donnée oracle

2 Démarrage de la base de données oracle

- Fichiers de paramètres d'initialisation
- Démarrage de la base de données oracle

3 Arrêt de la base de données oracle

Fichiers de paramètres d'initialisation

- Une BD oracle n'est disponible à l'utilisateur que lorsque le DBA a démarré une instance et ouvert la BD
- Pour démarrer une instance, le serveur Oracle doit lire le fichier de paramètres d'initialisation qui contient les paramètres d'initialisation pour allouer la SGA et démarrer les processus d'arrière plan. Il existe deux types de fichier de paramètres d'initialisation :
 - Fichier de paramètres statique PFILE : (nommé initSID.ORA)
 - Fichier de paramètres persistant SPFILE : (nommé spfileSID.ORA)

Fichiers de paramètres d'initialisation

Le fichier de paramètres Init<SID>.ORA

- Fichier texte
- Il peut être modifié manuellement à l'aide d'un éditeur du système d'exploitation
- Les modifications sont effectives au démarrage suivant.
- Il ne peut être ouvert que lors du démarrage de l'instance. S'il est modifié en cours, l'instance doit être interrompue et redémarrée pour que les nouvelles valeurs des paramètres soit effectives.
- Indispensable dans les postes clients

Fichiers de paramètres d'initialisation

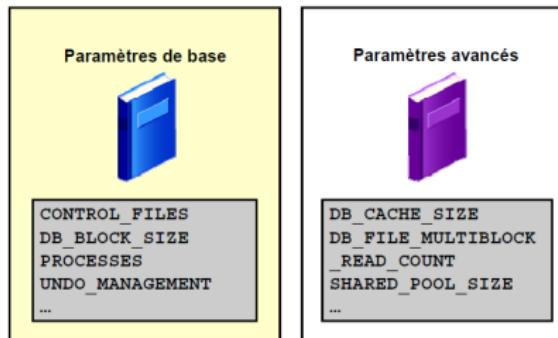
Le fichier de paramètres Spfile<SID>.ORA

- Fichier binaire
- Accessible par le serveur de BD
- Ne doit pas être modifié manuellement
- Moyen dynamique de gérer les paramètres
- Modifications apportées par le serveur et ne nécessite pas une installation en copie dans les postes clients en cas d'une connexion à distance

Fichiers de paramètres d'initialisation

- On peut créer un fichier de paramètre au format texte «pfile» à partir d'un fichier de paramètre au format binaire «spfile» avec la syntaxe suivante :
 - `create pfile [=’nom_pfile’] From Spfile [=’nom_spfile’];`
- On peut créer un fichier de paramètre binaire «spfile» à partir d'un fichier de paramètre texte «pfile» avec la syntaxe suivante :
 - `create spfile [=’nom_spfile’] From pfile [=’nom_pfile’];`

Paramètres d'initialisation



Les paramètres d'initialisation permettent entre autres de spécifier :

- le nom de la base de données à laquelle l'instance est associée,
- l'emplacement physique des fichiers de contrôle,
- le répertoire de destination, par défaut, des fichiers de données (datafiles),
- la destination, par défaut, des fichiers de journalisation (redo log files),
- les valeurs affectées aux structures mémoire SGA,
- le tablespace d'annulation,
- etc.

Paramètres d'initialisation

Exemples de paramètres d'initialisation :

| Paramètre | Description |
|----------------------|--|
| CONTROL_FILES | Un ou plusieurs noms de fichier de contrôle |
| DB_FILES | Nombre maximal de fichiers de base de données |
| PROCESSES | Nombre maximal de processus utilisateur du système d'exploitation pouvant se connecter simultanément |
| DB_BLOCK_SIZE | Taille de bloc de base de données standard utilisée par tous les tablespaces |
| DB_CACHE_SIZE | Taille du cache de tampons de bloc standard |
| SGA_TARGET | Taille totale de tous les composants SGA |
| MEMORY_TARGET | Mémoire utilisable à l'échelle du système Oracle |
| PGA_AGGREGATE_TARGET | Quantité de mémoire PGA allouée à tous les processus |
| SHARED_POOL_SIZE | Taille de la zone de mémoire partagée (en octets) |
| UNDO_MANAGEMENT | Mode de gestion du volume d'annulation à utiliser |

Paramètres d'initialisation

Il existe deux types de paramètres d'initialisation :

1 Paramètres statiques :

- Ils ne peuvent être modifiés que dans le fichier de paramètres
- Les modifications sont effectuées uniquement à l'aide des commandes ALTER SYSTEM avec l'option SCOPE='SPFILE'
- Un redémarrage de l'instance est nécessaire pour que les modifications prennent effet

2 Paramètres dynamiques :

- Ils peuvent être modifiés tant que la base de données est en ligne
- Les modifications peuvent être effectuées à deux niveaux :
 - Niveau session
 - Niveau système
- Les modifications sont effectuées à l'aide des commandes ALTER SESSION et ALTER SYSTEM
- Ils sont valides pour la durée de la session ou dans les limites établies par le paramètre SCOPE

Paramètres d'initialisation

Les commandes ALTER SYSTEM / ALTER SESSION

- 1 ALTER SYSTEM SET parameter=value COMMENT='...' [SCOPE = MEMORY| SPFILE | BOTH];**
 - COMMENT : associe une chaîne de commentaire à ce changement
 - SCOPE : Spécifie à quel moment le changement doit prendre effet
 - MEMORY : Indique que le changement est effectué en mémoire (instance courante), prend effet immédiatement et dure jusqu'à la base soit arrêtée.
 - SPFILE : Indique que le changement est effectué dans le SPFILE et prend effet lors du redémarrage de la base. A utiliser lorsque vous changez la valeur d'un paramètre statique.
 - BOTH : Indique que le changement est effectué à la fois en mémoire et dans le SPFILE ; prend effet immédiatement et perdure après que la base ait été redémarrée
 - BOTH est la valeur par défaut si la BD a été démarré en utilisant un SPFILE
 - MEMORY est la valeur par défaut et la seule portée si la BD a été démarré en utilisant un PFILE
- 2 ALTER SESSION SET parameter=value;**

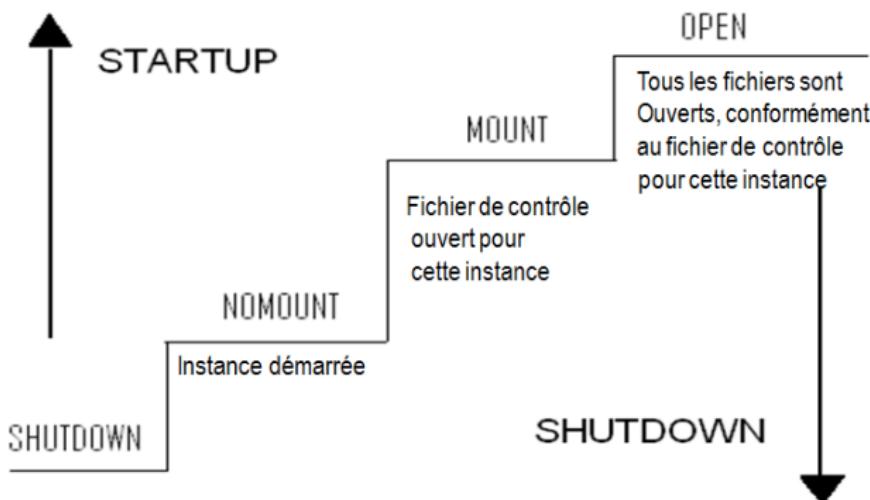
Paramètres d'initialisation

La vue V\$PARAMETER

- Pour savoir si un paramètre dynamique est modifiable immédiatement ou non (c-à-d modifiable uniquement en différé), il suffit de consulter la vue V\$PARAMETER et plus précisément la colonne ISSYS_MODIFIABLE
- ISSYS_MODIFIABLE :
 - FALSE si le paramètre est statique
 - IMMEDIATE si le paramètre peut être modifié immédiatement (c-à-d pour la session courante)
 - DEFERRED si le paramètre ne peut pas être modifié immédiatement (modification effectuée au niveau spfile)
- La vue V\$parameter permet d'obtenir les valeurs des paramètres utilisés dans l'instance ORACLE en cours. Les colonnes de v\$parameter
 - NAME : nom du paramètre
 - TYPE : type du paramètre
 - VALUE : la valeur
 - ISDEFAULT : FALSE : s'il est initialisé par le fichier de paramètres TRUE :sinon (valeur par défaut)
 - Isxxx_MODIFIABLE : indique si le paramètre peut être modifié par session (ISSES_MODIFIABLE) ou par instance (ISSYS_MODIFIABLE)
- Pour visualiser la valeur d'un paramètre on peut utiliser la commande : SHOW PARAMETER nom_paramètre (exemple show parameter resource_limit)

Démarrage de la base de données oracle

- Quand vous démarrez une base de données, les structures mémoires et les processus sont initialisés et démarrés
- Différents modes de démarrages sont possibles :



Démarrage de la base de données oracle

La commande STARTUP

- La commande STARTUP démarre l'instance et ouvre la BD
 - **STARTUP [OPEN [READ {ONLY | WRITE }] | MOUNT | NOMOUNT | FORCE | RESTRICT]**
 - OPEN : permet aux users d'accéder à la BD :
 - Ouvrir les fichiers de données en ligne et les fichiers de journalisation en ligne
 - READ ONLY : ouverture de la BD en mode lecture seule
 - READ WRITE : ouverture de la BD en mode lecture écriture
 - MOUNT : monte la BD mais ne permet pas l'accès à la BD (association d'une base de données à une instance démarrée)
 - NOMOUNT : crée la SGA et démarre les processus en arrière plan mais ne permet pas l'accès à la BD (ne s'effectue qu'à la création d'une base de données ou à la récréation des fichiers de contrôle)
 - FORCE : peut être utilisée quand le démarrage d'une BD avec l'option classique ne marche pas. Cette option effectue un shutdown abort et un restart de la base.
 - RESTRICT : démarre la base de données et la place en mode ouvert, mais donne accès seulement aux utilisateurs qui ont les priviléges RESTRICTED SESSION (pour effectuer des opérations de tuning, import/export, etc sans que personne ne se connecte à BD)
 - Annuler la restriction de session avec la commande : ALTER SYSTEM DISABLE RESTRICTED SESSION

Démarrage de la base de données oracle

La commande ALTER

- Permet de faire passer la base de données du statut NOMOUNT à MOUNT ou du statut MOUNT à OPEN :
 - ALTER DATABASE { NOMOUNT | MOUNT }
 - ALTER DATABASE { MOUNT | OPEN }

- Pour que les données ne soient pas modifiées, nous pouvons ouvrir la base en mode read-only :
 - ALTER DATABASE OPEN [READ WRITE] READ ONLY]

Plan

1 Architecture du serveur oracle

- Serveur de base de données oracle
- Composantes du serveur oracle
 - Instance oracle
 - Base de donnée oracle

2 Démarrage de la base de données oracle

- Fichiers de paramètres d'initialisation
- Démarrage de la base de données oracle

3 Arrêt de la base de données oracle

Arrêt de la base de données oracle

Les étapes d'arrêt de la base de données oracle

- Fermer la BD
 - Quand la base se ferme, Oracle écrit les changements du database buffer cache et les entrées du buffer redo log dans les data files et redo log files
 - Oracle ferme tous les data files et redo log files
 - Les control files restent ouverts
- DéMonter la BD d'une instance
 - Fermeture des control files
- Arrêt de l'instance
 - Libération de la SGA
 - Arrêt des background processes
 - Fermeture des fichiers trace et ALERT (journaux chronologiques des messages d'erreur)

Arrêt de la base de données oracle

- La commande SHUTDOWN arrête une instance
 - **SHUTDOWN [NORMAL | TRANSACTIONNEL | IMMEDIATE | ABORT]**
- Arrêt normal
 - Est l'arrêt par défaut
 - De nouvelles connexions ne sont pas permises
 - Le serveur oracle attend la déconnexion de tous les utilisateurs avant de terminer l'arrêt (temps d'attente peut être important sans délai prévisible de fermeture)
 - Oracle ferme et démonte la base avant d'arrêter l'instance
 - Pas de restauration de l'instance lors du démarrage suivant (car toutes les informations modifiées encore présentes dans la SGA sont écrites dans les data files et redo log files)
- Arrêt transactionnel
 - De nouvelles connexions ne sont pas permises
 - Le client ne sera plus connecté dès la fin de la transaction en cours
 - Pas de restauration de l'instance lors du démarrage suivant

Arrêt de la base de données oracle

■ Arrêt immédiat

- De nouvelles connexions ne sont pas permises
- Le serveur oracle n'attend pas la déconnexion des utilisateurs avant de terminer l'arrêt
- Les ordres SQL en cours ne seront pas traités
- Oracle ferme et démonte la base avant d'arrêter l'instance
- Pas de restauration de l'instance lors du démarrage suivant

■ Arrêt abort

- De nouvelles connexions ne sont pas permises
- Le serveur oracle n'attend pas la déconnexion des utilisateurs avant de terminer l'arrêt
- Les transactions non commitées ne seront pas effectuées (rollback)
- L'instance sera fermée sans la fermeture des fichiers
- Restauration de l'instance lors du démarrage suivant

Arrêt de la base de données oracle

Exemple

L'étape 1 jusqu'à l'étape 5 décrivent un transfert d'argent d'une banque à une autre

- 1 Interrogation du compte afin de récupérer le solde
- 2 Exécution de la commande INSERT pour transférer les fonds au nouveau compte
- 3 Exécution de la commande UPDATE afin de supprimer les fonds du solde courant
- 4 Exécution d'un COMMIT pour finir correctement la transaction
- 5 Déconnexion du Serveur Oracle

