

# 9

## Les fichiers d'une base Oracle 10g

---

### Dans ce chapitre :

- les fichiers qui composent une base Oracle 10g ;
- les fichiers de données ;
- les fichiers redo-log ;
- les fichiers de contrôle ;
- les fichiers d'initialisation et de configuration ;
- l'architecture OFA proposée par Oracle.

Si la mémoire est largement utilisée pour accélérer le fonctionnement de la base, les fichiers inscrits sur des disques durs permettent le stockage persistant des données. Oracle a réservé à certains fichiers le rôle de conserver les informations, d'autres sont destinés uniquement au fonctionnement interne de la base ou à son paramétrage.

Dans ce chapitre, nous ne nous limitons pas aux fichiers qui composent la base de données et qui doivent être sauvegardés, mais nous étudions aussi les fichiers de paramétrage, trace, log, etc. Ils sont méconnus et pourtant d'une grande utilité pour rechercher l'origine de problèmes.

Nous décrirons l'OFA (*Optimal Flexible Architecture*), qui propose une méthode d'organisation des fichiers Oracle sur votre système. L'approche OFA, rigoureuse et structurée, simplifie énormément le travail de l'administrateur Oracle 10g.

## Les fichiers Oracle 10g et leur rôle

Installer Oracle sur un système signifie y placer un nombre impressionnant de fichiers. C'est pourquoi, en amont de toute installation, il faut pouvoir identifier les différentes catégories de fichiers. Du point de vue d'un administrateur Oracle, on peut classer ces fichiers comme suit :

- les logiciels Oracle ;
- les fichiers de données ;
- les fichiers redo-log ;
- les fichiers redo-log archivés ;
- les fichiers de contrôle ;
- le fichier d'initialisation ;
- les fichiers d'initialisation d'Oracle Net ;
- le fichier d'alerte ;
- le fichier trace des utilisateurs.

### Les logiciels Oracle

Ce sont tous les fichiers dont l'installation est nécessaire au fonctionnement d'une base de données. Leur nombre est impressionnant, mais vous n'avez pas à vous en soucier. Tout comme les fichiers qui composent le système Windows, ils sont installés suivant les options choisies. Seul l'ajout d'un nouveau logiciel ou la modification d'une option d'un composant Oracle vous obligerait à intervenir sur ces fichiers.

En jargon Oracle, l'ensemble de ces fichiers s'appelle une *distribution*. Tous les logiciels d'une même distribution sont placés dans une même Oracle Home. Ce point est détaillé plus loin dans ce chapitre.

Plusieurs distributions d'Oracle peuvent coexister sur la même machine, par exemple, une version 9.2 en production et une version 10.1 en cours de validation. Les règles OFA, abordées dans ce chapitre, vous aident à organiser vos distributions, les bases de données et les fichiers d'administration.

### Les fichiers de données

Une fois votre distribution Oracle installée, une ou plusieurs bases de données sont créées. Leur dimension dépend de la quantité d'informations à stocker. Pour répondre à ces besoins, le nombre, la taille et l'emplacement des fichiers de données seront adaptés.

Les fichiers de données sont les plus volumineux de votre base. Si celle-ci est démarrée, il suffit d'interroger son dictionnaire interne pour connaître les caractéristiques de l'ensemble des fichiers qui la composent. Si elle est arrêtée, mieux vaut avoir suivi des

règles d'installation rigoureuses, comme les règles OFA, pour savoir où se trouvent ces fichiers. Comme le volume des informations est susceptible d'évoluer, il est fréquent qu'un administrateur Oracle intervienne sur ces fichiers.

Oracle évolue : historiquement, l'emplacement, le nom et parfois la taille des fichiers doivent être précisés par le DBA. À partir d'Oracle9, vous aviez la possibilité d'indiquer juste les répertoires où placer les fichiers, Oracle se chargeait de les nommer, les créer et les gérer. Oracle 10g apporte la notion d'ASM (*Automatic Storage Management*). Nous franchissons une nouvelle étape : c'est un substitut au système de fichiers du système d'exploitation (Windows, Linux). Vous créez un « système de fichiers » ASM et Oracle s'occupe de tout ! ASM est présenté au chapitre 2, *Configurer les disques pour Oracle*. Comme ASM dépasse les objectifs de ce livre, seule la gestion « classique » des fichiers sera abordée.

Les fichiers de données contiennent toutes les informations de votre base dans un format spécifique à Oracle. Il n'est pas possible d'en visualiser le contenu avec un éditeur de texte. Même un éditeur binaire montre un format interne très complexe : mieux vaut renoncer !

Le seul et unique moyen pour accéder et manipuler des données stockées dans Oracle est d'utiliser le langage SQL. Vous ne pourrez jamais y accéder en manipulant les fichiers.

Pour ajouter de nouveaux fichiers de données, il faut utiliser des ordres SQL. Pour déplacer ou supprimer ces fichiers, il faut associer des ordres SQL (par exemple préciser à la base le déplacement) et des commandes Windows (pour le transfert physique du fichier). Une option SQL permet maintenant d'associer la suppression physique d'un fichier à la commande SQL. Assurez-vous de disposer d'une bonne sauvegarde avant d'utiliser cette option !

Lorsqu'un fichier de la base est plein, il peut automatiquement augmenter en taille. Le DBA précise un incrément et fixe une dimension maximale. Cette limite supérieure évite de nombreux désagréments, comme la saturation de votre espace disque, la création de fichiers d'une taille supérieure à la celle permise par l'espace disque disponible ou dépassant les capacités de votre outil de sauvegarde.

Les threads utilisateurs lisent les données dans les fichiers de données et le thread Database Writer (DBWR) les écrit dans ces fichiers.

Les fichiers de données contiennent deux types d'informations : d'une part, celles du dictionnaire de données et de travail, d'autre part celles des utilisateurs.

Les données des utilisateurs et le dictionnaire constituent le cœur de la base de données Oracle. Ces éléments sont tous composés de tables et d'index. Les informations de travail sont utilisées dans le fonctionnement interne d'Oracle. Elles permettent de gérer la notion de transaction, de trier et de manipuler de gros volumes de données.

Tous les fichiers de données ont un format spécifique à Oracle : le *bloc Oracle* est défini lors de la création de la base et il est difficile de le modifier. Tous les échanges entre les disques, la mémoire et les threads se feront par des multiples de ce bloc. Entre une base

de données conçue pour effectuer de très nombreuses lectures/écritures « petites » et un infocentre accédant de manière linéaire à de très gros volumes de données, on perçoit l'importance du bloc oracle sur les performances.

Un fichier de données appartient à un unique tablespace, mais un tablespace peut posséder plusieurs fichiers. La notion de tablespace est abordée plus loin dans ce chapitre.

### Les fichiers redo-log

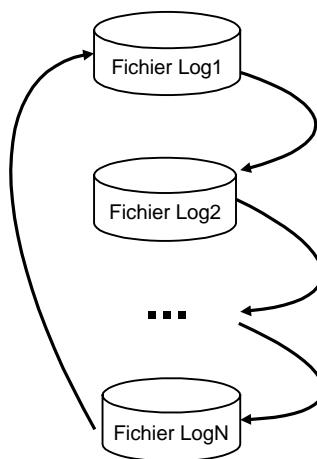
Les fichiers redo-log sont des fichiers de journalisation utilisés pour le fonctionnement interne d'Oracle, ils enregistrent et conservent toutes les modifications successives de votre base de données. Ils sont utiles lors d'une restauration à la suite d'un problème d'instance ou de disque. Cette restauration consiste à rejouer le contenu des fichiers redo-log dans la base.

Le thread LGWR (*Log Writer*) est chargé de l'écriture dans les fichiers redo-log. Il y enregistre les valeurs avant et après modification, ainsi que des renseignements internes. À l'instar des fichiers de données, il n'est pas possible d'accéder au contenu des fichiers redo-log avec un éditeur de texte. Un nouvel outil livré avec Oracle 10g, le Log Miner, permet d'analyser le contenu des fichiers log. Il s'appuie sur des packages PL/SQL (DBMS\_LOGMNR et DBMS\_LOGMNR\_D) et des vues V\$ de la base.

Lorsque le fichier redo-log courant est saturé, Oracle poursuit sur le suivant et ainsi de suite jusqu'au dernier. Quand celui-ci est plein, Oracle réutilise le premier, puis le second, etc. L'utilisation des fichiers est donc circulaire.

Figure 9-1

Chaînage circulaire  
des fichiers redo-log



Avant la réutilisation d'un fichier redo-log, Oracle le sauvegarde à un autre emplacement si la base fonctionne en mode archivage (ARCHIVELOG MODE). Dans les autres cas, Oracle le réemploie en supprimant son contenu sans sauvegarde préalable. Entre-temps, les données présentes en mémoire et copiées dans les fichiers redo-log à écraser ont été

érites dans les fichiers de données. Le mode d'archivage est optionnel : c'est l'administrateur Oracle qui décide de sa mise en service ou non.

Pour des raisons de sécurité, ces fichiers peuvent être multiplexés (c'est-à-dire dupliqués) dans des groupes (*Groups*). Si chaque groupe est situé sur un disque différent, cela évite une perte d'informations en cas de panne d'un disque.

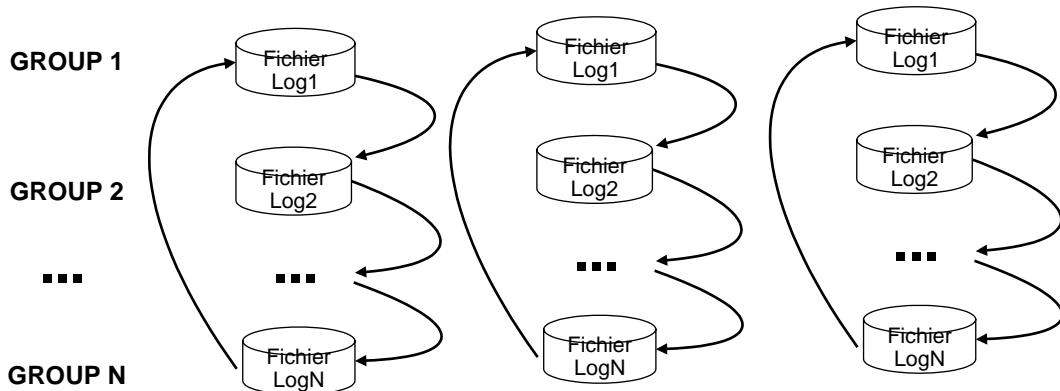


Figure 9-2

Fichiers redo-log groupés

Les fichiers redo-log doivent se trouver sur les disques les plus rapides, car chaque modification s'y trouve inscrite. Les disques rapides étant souvent les moins sécurisés, la fiabilité est alors assurée par le multiplexage des fichiers redo-log placés sur des disques différents.

Des vues du dictionnaire interne de votre base permettent de localiser vos fichiers redo-log. Un administrateur Oracle intervient sur ces fichiers uniquement pour les déplacer ou en modifier la taille afin, sous certaines conditions, d'améliorer la performance de la base. Oracle Enterprise Manager dispose d'une interface graphique pour réaliser ces actions.

## Les fichiers redo-log archivés

Lorsqu'une base est en mode ARCHIVELOG, les fichiers redo-log sont archivés à mesure de leur remplissage. Vous devez apporter la plus grande attention à cette opération. Si le répertoire de destination des archives est saturé, la base de données se bloque et un message s'inscrit dans le fichier d'alerte (voir plus loin). Oracle considère qu'il vaut mieux bloquer les utilisateurs que débrayer un mécanisme de sécurité majeur. L'emplacement et le nettoyage régulier des anciens fichiers archivés représentent les principales préoccupations de l'administrateur d'une base en mode ARCHIVELOG.

En cas de restauration, l'opération consiste à réaliser une sauvegarde complète de la base de données puis à redérouler les fichiers redo-log archivés les uns à la suite des autres.

## Les fichiers de contrôle

Les fichiers de contrôle sont créés en même temps que la base de données. Ils sont principalement utilisés à chaque démarrage de celle-ci, puis mis à jour automatiquement par Oracle. Pour des raisons de sécurité, on peut créer plusieurs fichiers de contrôle, mais ils sont tous identiques.

Le fichier d'initialisation utilisé pour lancer l'instance situe l'emplacement des fichiers de contrôle. Chacun d'eux précise la localisation de tous les autres fichiers (données et redo-log) qui composent la base.

Les fichiers de contrôle indiquent si la base de données a été correctement fermée et si une restauration est nécessaire. Il est impossible de les visualiser pour en exploiter le contenu.

Des vues du dictionnaire interne de la base permettent de les localiser. Si vous les perdez, il est possible de les créer à nouveau, à partir d'une base Oracle en fonctionnement. Si celle-ci est arrêtée, il faudra repartir d'une sauvegarde. Sauf à les déplacer et à les sauvegarder, un administrateur Oracle n'a pas à s'en préoccuper.

## Le fichier d'initialisation

Toute instance Oracle nécessite des paramètres d'initialisation utilisés à chaque démarrage. Ils peuvent être conservés sous forme d'un fichier texte ou enregistrés dans un fichier d'initialisation persistant.

Dans le premier cas, on parlera du fichier initSID.ora et du fichier persistant spfileSID.ora dans le second. Le SID correspond à l'identifiant de l'instance qu'il doit démarrer.

Le fichier d'initialisation persistant et de nouveaux paramètres d'initialisation dynamiques pouvant être modifiés « base en marche » sont des nouveautés apportées par Oracle 10g. Leur utilisation conjointe est très souple et nous vous engageons fortement à les utiliser. Ces points sont décrits au chapitre 10, *Démarrer et arrêter une base Oracle 10g*.

Les fichiers d'initialisation persistants spfileSID.ora sont placés par défaut en ORACLE\_HOME\database, soit C:\oracle\product\10.1.0\db\_1\database\spfileSID.ora. L'explorateur Windows doit être paramétré car ce sont des fichiers cachés.

Le fichier d'initialisation détermine entre autre la taille mémoire allouée à la SGA (System Global Area), zone mémoire commune à tous les utilisateurs de votre instance. Son rôle est déterminant dans l'optimisation des ressources d'une machine.

Pour modifier votre fichier d'initialisation initSID.ora, utilisez un éditeur de texte « propre », tel que le notepad. N'utilisez surtout pas un traitement de texte qui incorpore ses propres caractères de contrôle, comme Word ou Wordpad. Ne modifiez jamais le fichier d'initialisation persistant spfileSID.ora à l'aide d'un éditeur. Il risquerait d'être corrompu et inutilisable.

Lorsque vous installez Oracle 10g et qu'une base de données est créée par défaut, un fichier exemple documenté est fourni. Pour chaque paramètre important, des exemples de valeurs de configuration petite, moyenne ou importante sont fournies.

Il existe de nombreux paramètres, mais peu sont réellement utiles. Si vous ne modifiez pas la valeur de l'un d'entre eux, une valeur par défaut lui est affectée. Celle-ci dépend de votre système d'exploitation. Pour connaître la liste des paramètres du fichier d'initialisation, utilisez sous SQL\*Plus la commande suivante :

```
connect system/mot_de_passe
show parameters

# vous pouvez aussi accéder à la vue V$PARAMETER
select name, value from V$PARAMETER
/
```

La vue V\$PARAMETER vous permet aussi de retrouver les paramètres ayant une valeur différente de celle par défaut. Ce renseignement figure également dans le fichier d'alerte de votre base de données (voir plus loin).

Voici une liste des paramètres les plus courants rencontrés dans le fichier d'initialisation d'une instance Oracle. Vous les retrouverez dans pratiquement toutes les bases Oracle.

### Paramètres d'initialisation sans impact sur les performances

- **instance\_name** : nom de l'instance qui doit correspondre à la variable d'environnement ORACLE\_SID. Souvent nommé SID, il est codé sur 8 caractères maximum et ne peut plus être changé une fois la base créée ;
- **db\_name** : nom interne de la base de données. Nous vous conseillons très fortement de toujours attribuer à la base de données et au SID (qui identifie l'instance) le même nom (4 lettres majuscules). Ce nom est codé à l'intérieur des fichiers de contrôle ;
- **db\_domain** : chaîne alphanumérique identifiant le domaine réseau auquel appartient la base. Le nom global de la base de données sera la concaténation du db\_name avec le db\_domain. Par exemple, db\_name=TEST, db\_domain=gilles.fr, global\_dbname = TEST.gilles.fr. Ce paramètre est très important si la base doit communiquer avec d'autres bases Oracle par réplication ou autre ;
- **control\_files** : noms et emplacements des fichiers de contrôle de la base. Ils sont tous identiques et doivent être disponibles à chaque démarrage. Si l'un d'eux est perdu (destruction, défaillance disque...), il faut le supprimer de la liste des fichiers pour démarrer l'instance ;
- **open\_cursors** : nombre maximum de curseurs pouvant être ouverts simultanément par une session utilisateur ;
- **db\_files** : nombre maximum de fichiers de la base de données ;
- **remote\_login\_passwordfile** : indique le mode de gestion souhaité pour permettre à des utilisateurs distants d'arrêter/démarrer la base ;

- `max_dump_file_size` : limite la taille du fichier dump (erreur) lorsqu'un problème grave est détecté dans Oracle. Évite de saturer l'espace disque réservé aux fichiers dump ;
- `background_dump_dest` : précise l'emplacement des fichiers d'alerte et de log associés à l'instance ;
- `core_dump_dest` : précise l'emplacement de création des fichiers résultant d'erreurs internes Oracle ou « core » ;
- `user_dump_dest` : localise l'emplacement des fichiers d'alerte et de log associés à des problèmes concernant les threads des utilisateurs ;

### Paramètres d'initialisation importants pour les performances

- `compatibles` : assure un mode de compatibilité entre différentes versions de base Oracle. Ainsi, vous pouvez demander à ce qu'une base Oracle9.2 se comporte comme la version 7.x ;
- `db_block_size` : détermine la taille d'un bloc Oracle utilisé par le tablespace SYSTEM pour son format interne et les échanges entre les fichiers et la mémoire. Utilisé lors de l'étape initiale de création de la base, ce paramètre influe énormément sur les performances. Nouveauté apportée par Oracle 10g, certains tablespaces peuvent avoir un `db_block_size` différent de celui du tablespace SYSTEM ;
- `sga_target` : indique la taille totale de tous les composants de la SGA. Si `sga_target` est précisée, alors les buffers de données (`db_cache_size`), la zone réservée pour Java (`java_pool_size`), la Large Pool (`large_pool_size`) et la mémoire partagée (`shared_pool_size`) seront dimensionnés automatiquement en fonction de l'activité de la base ;
- `db_cache_size` : (anciennement `db_block_buffers`) taille affectée au buffer de données en mémoire. C'est l'un des paramètres les plus visibles sur la taille de la SGA en mémoire. Comme tous les paramètres qui gèrent la taille de la SGA, il est modifiable dynamiquement ;
- `db_block_buffers` : par souci de compatibilité, on peut toujours utiliser ce paramètre mais il n'est pas dynamique alors que `db_cache_size` l'est ;
- `shared_pool_size` : taille en octets de la mémoire partagée affectée à l'instance. Ce paramètre dynamique influe sur la taille de la SGA ;
- `large_pool_size` : taille en octets de la mémoire partagée affectée à l'instance. Ce paramètre dynamique influe sur la taille de la SGA ;
- `java_pool_size` : taille en octets de la mémoire partagée affectée au fonctionnement des options Java incorporées dans la base ;
- `pga_aggregate_target` : taille cible de la mémoire PGA (tris, fusions...) disponible pour l'ensemble des processus attachés à l'instance ;
- `sort_area_size` : mémoire PGA affectée à un seul processus. Maintenu pour compatibilité ascendante. Oracle recommande d'utiliser `pga_aggregate_target` ;

- processes : nombre maximal de threads associés à une instance Oracle. Permet de limiter le nombre de connexions à une instance, car chacune crée un thread dédié. Pour éviter un blocage, fixer une valeur supérieure au nombre maximum de connexions envisagées ;
- sessions : valeur déduite automatiquement par Oracle depuis la valeur processes ;
- undo\_management : détermine si l'instance est lancée en gestion automatique des annulations gérées dans un tablespace spécifique ou dans le mode manuel utilisant des rollback segments. Le premier mode correspond à la valeur AUTO et le second à MANUAL. Oracle recommande d'utiliser la gestion automatique ;
- undo\_tablespace : indique le nom du tablespace utilisé par la base pour gérer automatiquement les segments d'annulation ou *undo* ;
- rollback\_segment : mode de gestion des segments d'annulation conservé pour compatibilité avec les versions antérieures à Oracle 10g. Oracle recommande d'utiliser les tablespaces de type UNDO ;
- db\_file\_multiblock\_read\_count : fixe le nombre de blocs de données écrits simultanément ;
- log\_buffer : nombre de blocs (d'une taille unitaire de db\_block\_size octets) alloués au buffers redo-log en mémoire ;
- log\_checkpoint\_interval : durée en secondes entre deux points de contrôle qui forcent l'écriture dans les fichiers redo-log ;
- audit\_trail : TRUE ou FALSE, si vous souhaitez utiliser les capacités d'audit d'une base ;
- log\_archive\_start : détermine si le thread destiné à archiver les fichiers redo-log doit être démarré. Attention, pour activer l'archivage, il faut aussi mettre la base en mode ARCHIVELOG ;
- timed\_statistics : collecte les statistiques avec sql\_trace ;
- sql\_trace : autorise l'analyse de toutes les sessions sur une base Oracle (très consommateur de CPU et d'espace disque). Les statistiques sont activées par timed\_statistics.

Un administrateur Oracle intervient sur ce fichier lors de la création puis du tuning de la base. De plus en plus de paramètres sont modifiables base en marche par la commande ALTER SYSTEM.

## Les fichiers d'initialisation d'Oracle Net

Les fichiers tnsnames.ora et listener.ora sont les principaux fichiers de paramétrage d'Oracle Net. Ils sont détaillés au chapitre 15, *Oracle Net, le middleware Oracle*. Si vous créez une nouvelle base, il faut modifier le fichier tnsnames.ora pour en tenir compte. Ne les oubliez pas dans vos sauvegardes, car leur paramétrage est parfois délicat à mettre au point.

## Le fichier d'alerte de la base

Ce fichier d'alerte est ignoré par la majorité des administrateurs. Pourtant, son exploitation est précieuse car, en cas de problèmes, Oracle y laisse des messages explicites. Lorsqu'une difficulté survient, pensez-y !

Chaque base Oracle possède son fichier d'alerte. Il ajoute des informations durant la vie d'une base. Vous pouvez archiver régulièrement son contenu, puis le supprimer. Pendant le démarrage de la base, si le fichier d'alerte n'existe pas, Oracle en crée un pour y écrire des informations de type « DBWR process started ». L'exemple suivant fournit le contenu d'un fichier d'alerte obtenu après le lancement d'une base sous WIndows.

```
Starting up ORACLE RDBMS Version: 10.1.0.2.0.
System parameters with non-default values:
  processes          = 150
  shared_pool_size   = 83886080
  large_pool_size    = 8388608
  java_pool_size     = 50331648
  nls_language       = FRENCH
  nls_territory      = FRANCE
  control_files      = C:\ORACLE\PRODUCT\10.1.0\ORADATA\PROD\CONTROL01.CTL,
                      ➤C:\ORACLE\PRODUCT\10.1.0\ORADATA\PROD\CONTROL02.CTL,
                      ➤C:\ORACLE\PRODUCT\10.1.0\ORADATA\PROD\CONTROL03.CTL
  db_block_size       = 8192
  db_cache_size      = 25165824
  compatible          = 10.1.0.2.0
  db_file_multiblock_read_count= 16
  db_recovery_file_dest = C:\oracle\product\10.1.0\flash_recovery_area
  db_recovery_file_dest_size= 2147483648
  undo_management     = AUTO
  undo_tablespace     = UNDOTBS1
  remote_login_passwordfile= EXCLUSIVE
  db_domain           = gilles.fr
  dispatchers         = (PROTOCOL=TCP) (SERVICE=PRODXDB)
  job_queue_processes = 10
  background_dump_dest = C:\ORACLE\PRODUCT\10.1.0\ADMIN\PROD\BDUMP
  user_dump_dest      = C:\ORACLE\PRODUCT\10.1.0\ADMIN\PROD\UDUMP
  core_dump_dest      = C:\ORACLE\PRODUCT\10.1.0\ADMIN\PROD\CDUMP
  sort_area_size      = 65536
  db_name              = PROD
  open_cursors         = 300
  pga_aggregate_target = 25165824
CKPT started with pid=6, OS id=2572
LGWR started with pid=5, OS id=2536
PMON started with pid=2, OS id=2388
DBW0 started with pid=4, OS id=2500
RECO started with pid=8, OS id=2644
SMON started with pid=7, OS id=2608
MMAN started with pid=3, OS id=2464
Fri May 06 10:04:46 2005
starting up 1 dispatcher(s) for network address
  ➤'(ADDRESS=(PARTIAL=YES)(PROTOCOL=TCP))'...
```

```
CJQ0 started with pid=9, OS id=2680
Fri May 06 10:04:46 2005
starting up 1 shared server(s) ...
Fri May 06 10:04:53 2005
alter database mount exclusive
Fri May 06 10:04:53 2005
Controlfile identified with block size 16384
Fri May 06 10:04:57 2005
Setting recovery target incarnation to 1
Fri May 06 10:04:58 2005
Successful mount of redo thread 1, with mount id 4264330181
Fri May 06 10:04:58 2005
Database mounted in Exclusive Mode.
Completed: alter database mount exclusive
Fri May 06 10:04:58 2005
alter database open
```

L'exploitation des renseignements contenus dans le fichier d'alerte dépend de votre expérience Oracle. Beaucoup de ces messages ne sont que des informations sans lien avec une éventuelle erreur.

Le fichier trace de chaque base est placé dans le répertoire indiqué par la variable `background_dump_dest` du fichier d'initialisation `initSID.ora`.

Lorsque la base de données démarre, tous les paramètres du fichier `initSID.ora` qui ne correspondent pas aux valeurs par défaut sont écrits. Sont également indiqués : le lancement des différents threads et la restauration automatique de la base si elle n'a pas été fermée proprement. En général, le fichier d'alerte contient une trace de tous les démaragements et arrêts, de la création de tablespaces et de segment d'annulation, de quelques ordres de modification (ALTER) ainsi que des informations concernant la permutation des fichiers redo-log et des erreurs rencontrées.

Chaque renseignement est daté et un message précise souvent le lancement d'une action et un autre sa fin.

Pour un DBA Oracle, il est très important de consulter régulièrement le contenu du fichier trace de chaque base. C'est le « carnet de santé » de celle-ci. Les messages que vous y trouverez vous aideront à corriger les erreurs.

## ***L'observateur d'événements***

De nombreuses informations importantes concernant la vie d'une base Oracle sont inscrites dans le *journal d'application de l'observateur d'événements* : démarrage, arrêt, connexion d'un administrateur, actions importantes réalisées sur la base, etc.

On y retrouve nombre d'informations présentes dans le fichier d'alerte, mais il moins facile à exploiter en cas de problème.

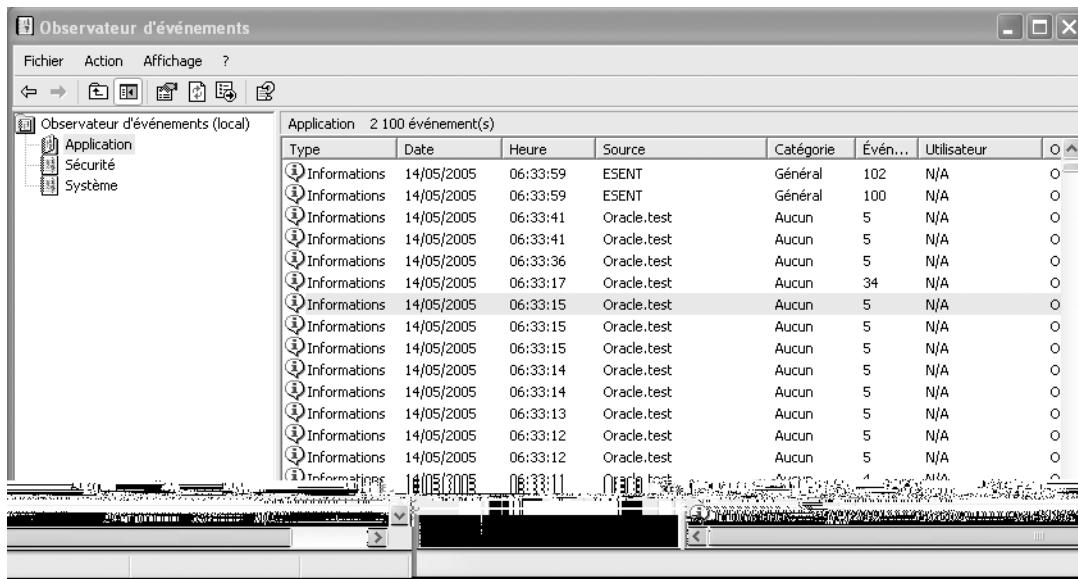


Figure 9-3

Observateur d'événements

## Les fichiers trace d'Oracle Net

Oracle Net, le middleware réseau Oracle utilisé en client-serveur, dispose lui aussi de fichiers trace. Côté serveur, le listener enregistre tous les démaragements, arrêts, connexions réussies ou ayant échoué dans un fichier « log ». Ce fichier, qui enregistre les connexions (et non leur contenu) est indispensable pour contrôler d'éventuelles tentatives de piratage de la base.

Côté poste client (client Windows, Linux ou autre...), il est possible de forcer la création d'un fichier trace pour enregistrer tout ce que votre application confie à Oracle Net dans une communication client-serveur. Ces points sont traités au chapitre 15, *Oracle Net, le middleware Oracle*.

## Les fichiers trace des utilisateurs

Il est possible de demander à Oracle de réaliser un fichier trace pour chaque connexion utilisateur à la base. L'utilisation de ces fichiers trace doit être exceptionnelle, car elle engendre sur le serveur une surconsommation de place disque et de temps processeur.

Le fichier trace des utilisateurs est placé dans le répertoire indiqué par la variable `user_dump_dest` du fichier d'initialisation `initSID.ora`. La trace est déclenchée/arrêtée par la commande `ALTER SESSION`.

## Faire coexister plusieurs versions Oracle

Il est possible de faire coexister plusieurs versions différentes d'Oracle sur un même serveur. Toutes ces versions peuvent fonctionner simultanément. Vous pouvez ainsi avoir une base en version 9.2 de production tout en testant l'installation de la version 10.0.1.

Pour cela, des variables d'environnement ainsi que des valeurs de la base de registre sont utilisées. Elles sont présentées dans ce paragraphe.

### Qu'est-ce qu'un Oracle Home ?

Pour chaque installation, Oracle Universal Installer vous demande de saisir un nom qui décrira un nouvel Oracle Home. Sous ce nom, Oracle regroupe des variables qui définissent l'environnement de fonctionnement d'une base. Ainsi, un Oracle Home détermine :

- le répertoire où seront installés les logiciels Oracle. Cette valeur correspond à la variable ORACLE\_HOME, par exemple C:\orant\ora92 ou C:\oracle\product\10.1.0\db\_1 ;
- la variable Windows PATH qui précise où sont les fichiers exécutables d'Oracle ;
- des valeurs de la base de Registre ;
- des services.

L'Oracle Home joue le rôle d'aiguillage qui permet de faire pointer un ensemble de variables vers un environnement ou vers un autre. Il est impératif d'installer chaque version d'Oracle dans un Oracle Home unique. Cela facilite la maintenance des différentes versions qui peuvent coexister sur un serveur.

Il est conseillé d'utiliser un nom d'Oracle Home qui soit en relation avec le logiciel à installer. Par exemple, OraHome92 si vous installez Oracle version 9.2 ou OraDb10g\_home1 si vous installez la version 10g.

### Quel est l'impact d'un Oracle Home sur le serveur ?

Oracle utilise l'Oracle Home pour personnaliser son installation. Par exemple, dans le menu Démarrer > Programmes, le menu « Oracle - *HomeName* » sera créé avec Home-Name correspondant au nom d'Oracle Home.

On trouvera ainsi les menus « Oracle – OraHome92 » et « Oracle – OraDb10g\_home1 » si les noms donnés aux Oracle Home sont respectivement OraHome92 et OraDb10g\_home1.

On retrouve aussi le nom d'Oracle Home dans l'identifiant de certains services. Par exemple, un listener installé dans un Oracle Home nommé OraDb10g\_home1 donnera naissance au service nommé « OracleOraDb10g\_home1TNSListener ».

### Quel est l'impact d'un Oracle Home sur le Registre ?

Comme les paramètres d'une installation Oracle sont conservés dans la base de registre, l'Oracle Home permet de structurer l'organisation des clés de registre.

- HKEY\_LOCAL\_MACHINE\SOFTWARE\ORACLE\KEY\_OraDb10g\_home1 permet de numérotter les différentes installations d'Oracle. Elle contient les informations sur

toutes les variables d'environnement utilisées par tous les logiciels Oracle liés à cet Oracle Home. La terminaison « homeN » indiquera le numéro d'installation. Cette clé contient les informations :

- ORACLE\_HOME\_NAME : le nom donné à l'Oracle Home ;
- ORACLE\_HOME : le répertoire pointant où est installé Oracle ;
- NLS\_LANG : le langage utilisé lors de cette installation ;
- ORACLE\_BUNDLE\_NAME : la version Oracle installée ;
- ORACLE\_SID\_XXX : les caractéristiques de démarrage et d'arrêt des instances *SID* (TEST et PROD sur la figure). Ne figurent à ce niveau que les instances relatives à la version Oracle installée dans le ORACLE\_HOME.

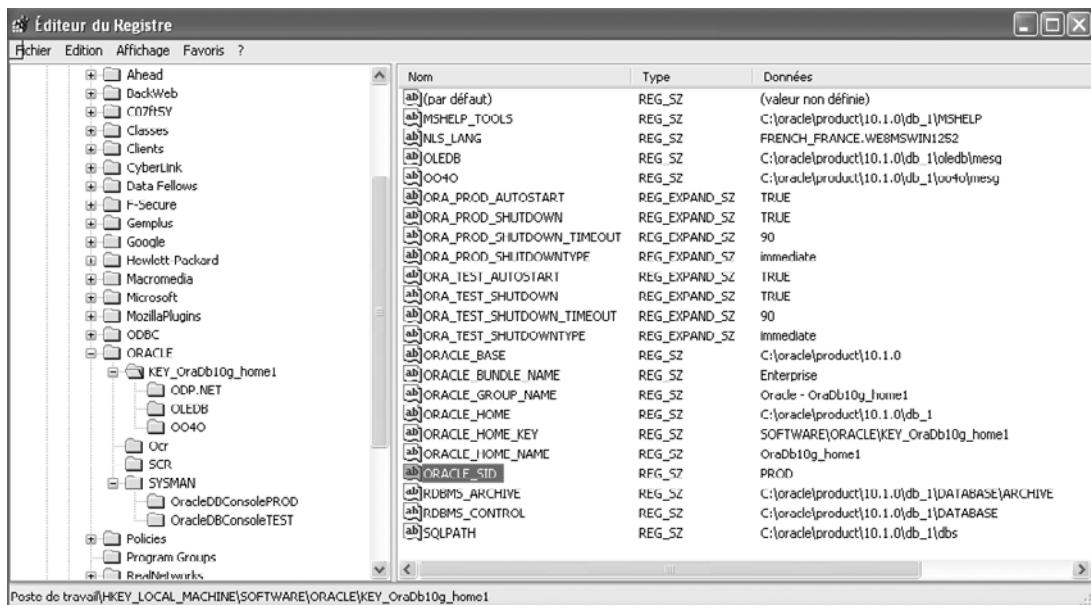


Figure 9-4

Clé de registre "ORACLE\_HOME"

Dans la figure précédente vous pouvez voir, par exemple, le paramétrage concernant le démarrage et l'arrêt de la base TEST et comment la base de données doit se comporter lors d'un arrêt/démarrage du serveur Windows.

Les paramètres de démarrage des instances TEST et PROD figurent dans la clé de registre relative à KEY\_OraDb10g\_home1. Cela indique que cette base a été créée avec Oracle version 10. Si une autre version d'Oracle était installée sur ce serveur, les bases qui seraient créées avec cette version figureraient dans la clé correspondante en HKLM\SOFTWARE\ORACLE\KEY\_OraDb10g\_home2.

Les valeurs de la base de registre peuvent être modifiées à l'aide de Regedit, mais aussi à l'aide de l'outil graphique *Oracle Administration Assistant for Windows*. On y retrouve ainsi les paramètres présents dans la base de registres.

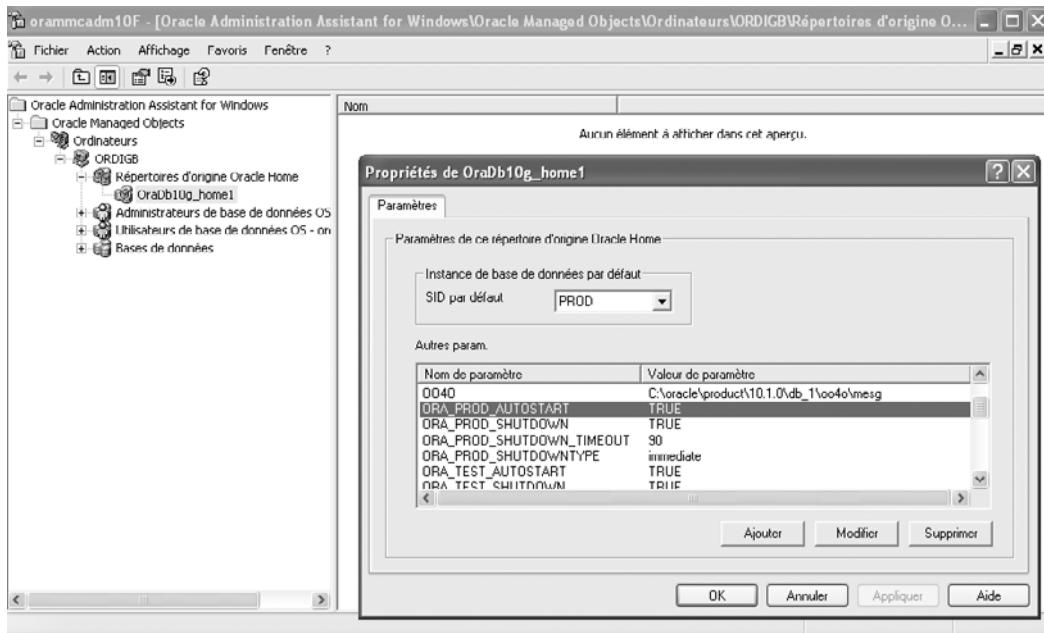


Figure 9-5

L'assistant Oracle pour Windows

## L'architecture OFA proposée par Oracle

Les répertoires d'installation de Windows sont bien structurés. Qu'en est-il de ceux d'Oracle ? Pour répondre à cette question, Oracle recommande d'utiliser la structure OFA (*Optimal Flexible Architecture*). C'est un ensemble de règles d'installation et de configuration qui vous donneront des bases Oracle rapides, fiables, faciles à installer et nécessitant peu de maintenance.

Les règles OFA sont le fruit de plus de dix années d'expérience de consultants d'Oracle sous Windows et Unix. Nous vous conseillons fortement de les consulter pour les utiliser avant toute installation Oracle 10g.

### Bénéfices procurés par OFA

OFA a été conçu pour :

- faciliter le travail des administrateurs Oracle en standardisant l'installation d'Oracle quel que soit le système d'exploitation utilisé : Windows, Unix, Linux... ;

- organiser un grand nombre de données et d'applications sur des disques en évitant les goulots d'étranglement nuisibles aux performances ;
- faciliter les tâches d'administration comme les sauvegardes des logiciels et des données. Ce sont des points essentiels ;
- installer plusieurs versions d'Oracle sur une même machine et les administrer facilement ;
- permettre le fonctionnement de plusieurs bases Oracle 10g sur une même machine, les gérer facilement ;
- suivre et modifier aisément l'augmentation de la taille des bases ;
- créer des espaces de travail pour les utilisateurs Oracle 10g.

## ***Caractéristiques d'une installation OFA***

Le système de fichiers est organisé pour vous offrir une administration simple :

- installer plusieurs versions d'Oracle sur un même serveur ;
- créer de nouvelles bases.

### **Les performances**

Les entrées/sorties sont réparties sur un nombre suffisant de disques pour éviter les problèmes d'engorgement.

En structurant l'installation, les ressources en matériel (CPU, disque) sont minimisées pour un même niveau de performance.

Une panne matérielle affecte le minimum possible d'éléments.

### **La sécurité**

Les sauvegardes sont facilitées, car l'emplacement de tous les composants sensibles est clairement identifié.

Dans chaque base de données, les fichiers qui la composent et ses fichiers d'administration ont tous des répertoires réservés. Ils comportent des noms significatifs qui permettent à un administrateur Windows de les identifier. L'association entre le tablespace et le fichier est contenue dans le nom du fichier.

### **Évolutivité**

Plusieurs versions d'Oracle peuvent fonctionner simultanément, ce qui permet d'installer et de tester une nouvelle version alors que l'ancienne est en production. Le transfert de l'ancienne version vers la nouvelle est simple pour l'administrateur et transparent pour les utilisateurs.

Il est très facile d'ajouter des composants à votre machine ou de changer une base Oracle de machine, car une installation OFA est bien séparée du reste des fichiers Windows.

OFA permet aisément de répartir les fichiers de vos bases de données sur un ou plusieurs disques.

## ***ORACLE\_BASE et ORACLE\_HOME***

Lors de l'installation, Oracle Universal Installer sépare les logiciels nécessaires au fonctionnement d'Oracle des fichiers membres des bases de données.

ORACLE\_BASE est le répertoire racine d'une installation Oracle. Si vous suivez les règles OFA, la valeur d'ORACLE\_BASE est X:\oracle (par exemple C:\oracle\product\10.1.0).

Oracle Home correspond aux variables nécessaires pour faire fonctionner les **logiciels Oracle**. Ces variables d'environnement regroupent : la variable ORACLE\_HOME qui indique l'emplacement où sont installés les logiciels (par exemple C:\oracle\product\10.1.0\db\_1). Partant de ce répertoire, Oracle sait que ses exécutables sont situés sous le répertoire BIN.

Si vous suivez les règles OFA, ORACLE\_HOME est situé sous ORACLE\_BASE. Par exemple :

```
ORACLE_BASE = C:\oracle\product\10.1.0
ORACLE_HOME1 = C:\oracle\product\10.1.0\db_1
ORACLE_HOME_NAME = OraDb10g_home1
Les logiciels Oracle seront alors installés en ORACLE_HOME.
```

Si plusieurs versions Oracle sont installées sur le serveur, il y aura un ensemble de variables Oracle Home par version d'Oracle :

```
ORACLE_BASE = C:\oracle\product\10.1.0
ORACLE_HOME1 = C:\oracle\product\10.1.0\db_1
ORACLE_HOME2 = C:\oracle\product\10.1.0\db_2
Les logiciels Oracle version 10.1 seront installés en ORACLE_HOME1
Les logiciels Oracle version 10.2 seront installés en ORACLE_HOME2
```

Pour chaque version d'Oracle, les autres variables attachées à chaque ORACLE\_HOME sont inscrites dans la base de registres.

## ***Où placer les fichiers des bases de données ?***

Nous avons vu dans ce chapitre que l'on peut interroger le dictionnaire de données Oracle pour connaître l'ensemble des fichiers qui compose la base. Ils peuvent être placés n'importe où sur votre système. Pour une meilleure organisation, OFA propose de ne pas mélanger les fichiers de vos bases de données avec ceux des logiciels Oracle. De même, il faut placer dans des répertoires différents les fichiers de bases séparées. Ce sont des erreurs répandues qui compliquent les sauvegardes et posent des problèmes lorsque vous souhaitez changer la version d'Oracle.

Pour bien séparer les fichiers de données des logiciels Oracle, les fichiers de données sont placés directement sous ORACLE\_BASE\oradata.

Comme plusieurs bases de données peuvent fonctionner simultanément, OFA conseille de créer un répertoire dédié pour contenir les fichiers de chaque base. Par exemple, si les bases TEST et PROD fonctionnent sur un serveur, on trouvera les répertoires :

Les fichiers sont placés sous ORACLE\_BASE\oradata  
C:\oracle\product\10.1.0\oradata\TEST : fichiers de la base TEST  
C:\oracle\product\10.1.0\oradata\PROD : fichiers de la base PROD

Comme les fichiers d'une base peuvent être répartis sur plusieurs disques,

→ on peut aussi trouver :

C:\oracle\product\10.1.0\oradata\TEST : fichiers de TEST sur C:\  
D:\oracle\product\10.1.0\oradata\TEST : fichiers de TEST sur D:\

## Séparer les segments d'utilisations différentes

Pour une meilleure organisation interne de la base de données, les règles OFA recommandent d'affecter un tablespace par type de segment à héberger :

- SYSTEM abritera le dictionnaire de données, procédures, fonctions, etc. ;
- SYSAUX, contiendra les tables, index de toutes les options Oracle (Java, Spatial, OEM...) ;
- TEMP accueillera les segments temporaires utilisés automatiquement par Oracle lors d'opérations de tri, de création d'index, etc. ;
- UNDOTBS réunira les segments d'annulation ou UNDO ;
- USERS rassemblera les segments (tables, index) de vos utilisateurs ;
- APPLI\_A\_DATA, APPLI\_A\_INDEX, APPLI\_B... serviront à l'hébergement des données des applications partagées entre plusieurs utilisateurs.

## Comment nommer les fichiers des bases de données ?

Pour identifier chaque type de fichiers d'une base, Oracle conseille de les repérer par des extensions spécifiques :

- .ctl pour les fichiers de contrôle ;
- .log pour les fichiers redo-log ;
- .dbf pour les fichiers de données (*Data Base Files*).

Comme les fichiers de contrôle, redo-log et de données peuvent être multiples, utilisez des noms significatifs pour les identifier :

Pour les fichiers de contrôle : controlnn.ctl. Par exemple :

Pour la base de SID=TEST  
control01.ctl : premier fichier de contrôle  
control02.ctl : second fichier de contrôle  
control03.ctl : troisième fichier de contrôle  
etc.

Pour les fichiers redo-log : *redonn.log*. Par exemple :

Pour la base de SID=TEST  
redo01.log : premier fichier redo-log  
redo02.log : second fichier redo-log  
redo03.log : troisième fichier redo-log  
etc.

Pour les fichiers de données : *tablespacenn.dbf*. Par exemple :

Pour les fichiers du tablespace SYSTEM  
system01.dbf : premier fichier  
system02.dbf : second fichier  
etc.

Pour les fichiers du tablespace DATA  
data01.dbf : premier fichier  
data02.dbf : second fichier  
etc.

Pour les fichiers du tablespace APPLI\_A  
appli\_a01.dbf : premier fichier  
appli\_a02.dbf : second fichier  
etc.

Conserver tous ces fichiers dans cette arborescence rend très facile la sauvegarde de vos données.

Par exemple, le premier fichier du tablespace SYSTEM de la base PROD sera : C:\oracle\product\10.1.0\oradata\PROD\system01.dbf ; son identification est simplifiée.

Pour des questions d'optimisation ou de place disque, vous pouvez répartir les fichiers d'une même base de données sur plusieurs disques. Dans ce cas, seuls le disque change (par exemple C :\, D :\, E :\), le reste de l'arborescence est identique.

Ainsi, le premier fichier de données du tablespace USER appartenant à la base PROD sera C:\oracle\product\10.1.0\oradata\PROD\user01.dbf. Le second fichier peut être placé sur un autre disque et se nomme D:\oracle\product\10.1.0\oradata\PROD\user02.dbf.

## Où placer les fichiers d'administration des bases ?

Chaque base de données possède ses propres fichiers de configuration, de trace, de log. Oracle conseille que tous les fichiers de création puis à d'exploitation d'une base soient placés dans des répertoires séparés. OFA vous propose de créer un jeu de répertoires pour chacune, avec pour chemin d'accès ORACLE\_BASE\admin. Par exemple, pour les deux bases de SID respectif TEST et PROD, leurs fichiers d'administration respectifs seront placés en :

Les fichiers sont placés sous ORACLE\_BASE\admin\SID  
C:\oracle\product\10.1.0\admin\TEST : contient les fichiers d'administration de TEST  
C:\oracle\product\10.1.0\admin\PROD : contient les fichiers d'administration de PROD

Pour bien structurer les fichiers d'administration de chaque base, Oracle crée sous **ORACLE\_BASE\admin\SID** les répertoires :

- *\bdump* : trace des threads de la base ;
- *\cdump* : fichier d'erreur du « noyau Oracle » ;
- *\create* : fichiers de création de la base ;
- *\exp* : fichiers d'export de la base (optionnel, à ajouter) ;
- *\pfile* : fichier d'initialisation de la base (*initSID.ora*) ;
- *\udump* : fichiers trace d'ordres SQL.

Par exemple, le répertoire qui hébergera le fichier d'alerte de la base PROD, alert-PROD.ora sera C:\oracle\product\10.1.0\admin\PROD\bdump.

## Avantages d'OFA

L'organisation des fichiers proposée par OFA rend votre administration très simple. Considérez les exemples ci-dessous qui utilisent le caractère « \* » pour manipuler les fichiers :

C :\	premier disque ;
D :\	second disque ;
*\oracle\*	liste de tous les répertoires d'installation des différentes versions Oracle (quel que soit leur disque d'installation) ;
*\oracle\admin\*	liste de toutes les bases de données Oracle dont les fichiers de configuration sont structurés suivant OFA (quel que soit leur disque d'installation) ;
*\*\oradata	liste de toutes les bases de données Oracle dont les fichiers de données sont structurés suivant OFA (quel que soit leur disque d'installation) ;
*\*\oradata\TEST\*	liste de tous les fichiers de la base de données TEST (quel que soit leur disque d'installation) ;
*\*\oradata\TEST\*.ctl	liste de tous les fichiers de contrôle de la base de données TEST (quel que soit leur disque d'installation).

## OFA : schéma d'installation des fichiers Oracle sous Windows

Dans ce premier exemple, seule la version 10 est installée et une seule base de données est en fonction, la base TEST. Nous indiquons aussi l'emplacement des principaux fichiers présentés dans ce chapitre : fichiers d'initialisation, d'administration, etc.

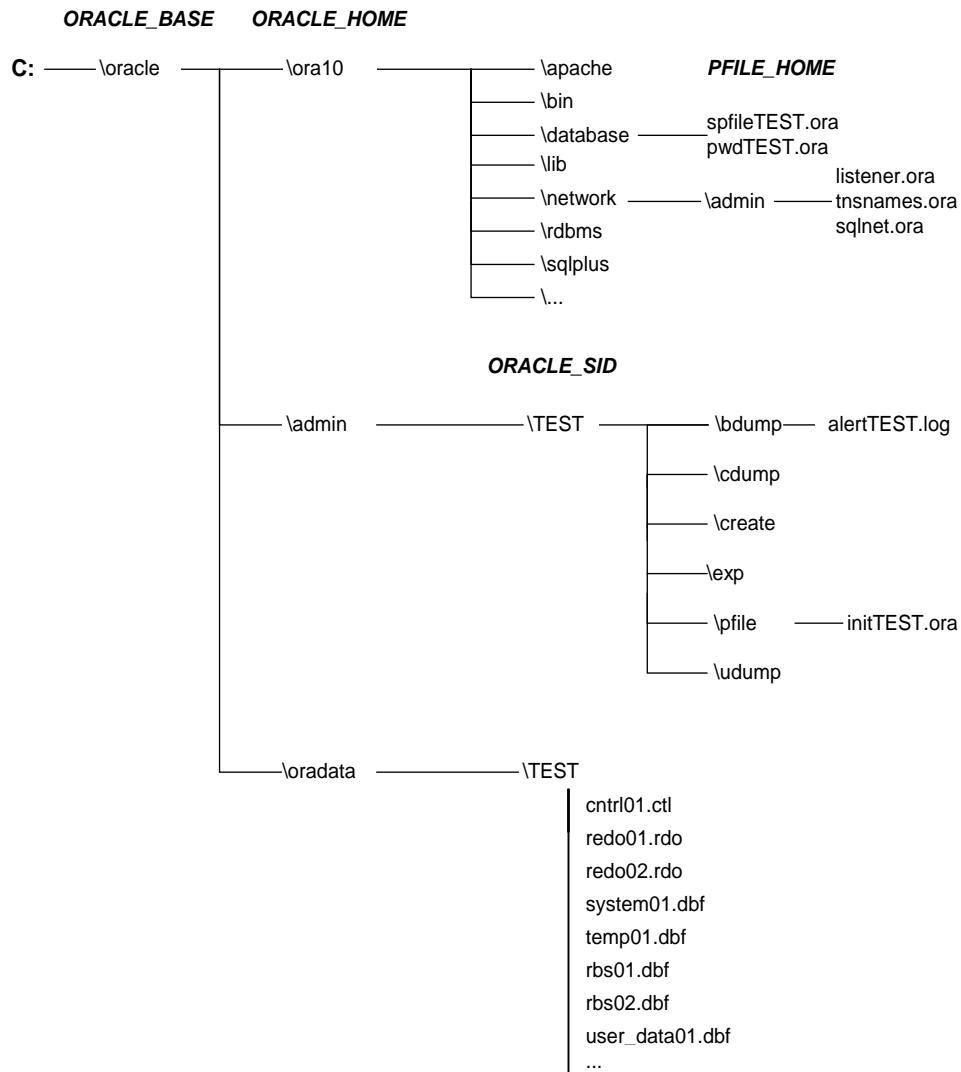


Figure 9-6

OFA : installation d'une base TEST sous Windows

Dans l'exemple suivant, seule la version 10 est installée et les fichiers de la base TEST sont répartis sur deux disques.

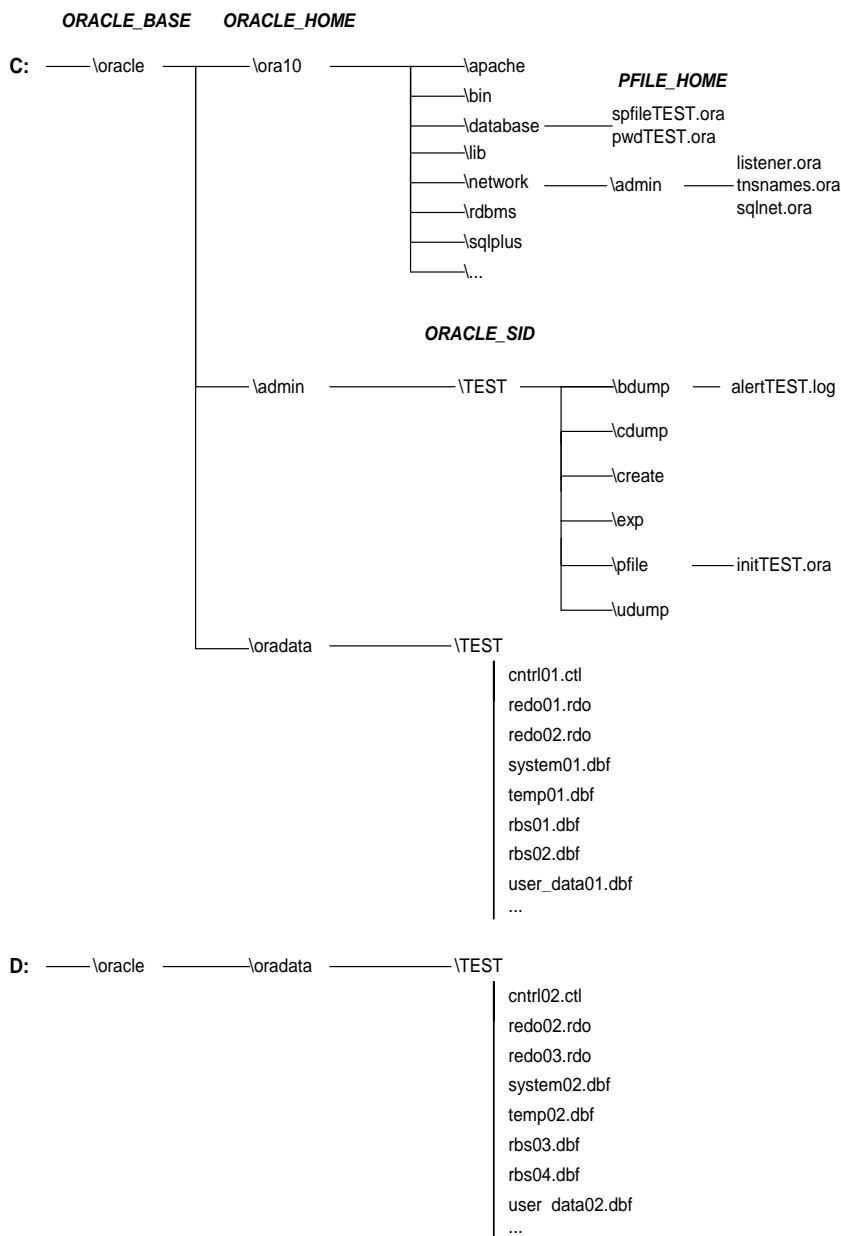


Figure 9-7

OFA : fichiers d'une base répartis sur deux disques

Dans l'exemple suivant, deux bases Oracle sont installées : TEST et PROD.

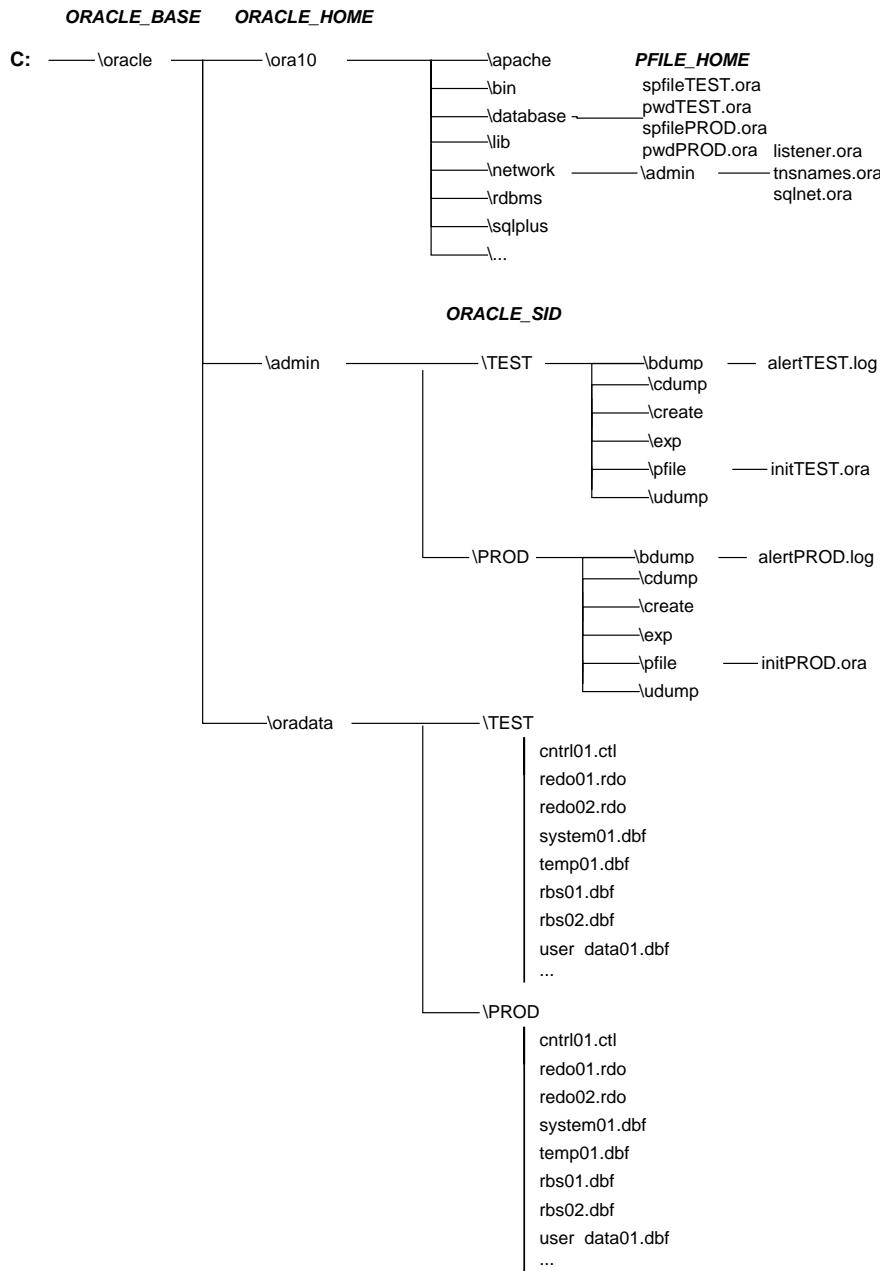


Figure 9-8

OFA : installation de deux bases Oracle sous Windows

Enfin, nous vous présentons comment coexistent deux versions différentes d'Oracle sur un même serveur. La base de production PROD utilisera la version 9.2 alors qu'une base de données TEST utilisera la version 10 avant de « basculer » la base de production de la version 9.2 vers la version 10.

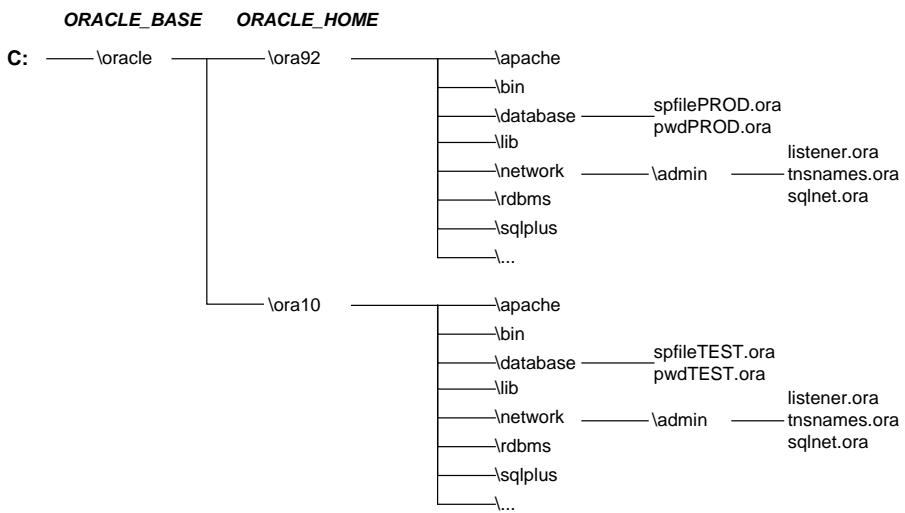


Figure 9-9

Coexistence de deux versions d'Oracle sur un serveur

Ces exemples (et d'autres) sont repris pour mémoire au chapitre 13, *L'installation d'Oracle 10g sous Windows*.

Ces exemples détaillés sont présentés pour insister sur l'importance des règles OFA. Elles sont simples, de bon sens et facilitent le travail des administrateurs Oracle et Windows. Utilisez-les !

## Résumé

Ce chapitre a abordé l'utilisation des différents fichiers auxquels recourt Oracle 10g : ceux destinés à conserver les données et ceux réservés au fonctionnement interne de la base ou à son paramétrage.

Pour mieux comprendre l'environnement d'une base Oracle 10g, nous ne nous sommes pas limités aux fichiers qui composent la base de données et qui doivent être sauvegardés, mais nous nous sommes intéressés aux fichiers de paramétrage, trace, log, etc. Ils sont méconnus et pourtant d'une grande utilité pour retrouver l'origine des problèmes.

Nous avons décrit l'OFA (*Optimal Flexible Architecture*) qui propose une méthode d'organisation des fichiers Oracle sur votre système. L'approche OFA, rigoureuse et structurée, simplifie considérablement le travail de l'administrateur Oracle 10g.

# 10

## Démarrer et arrêter une base Oracle 10g

---

### Dans ce chapitre :

- principes de l'installation d'Oracle ;
- étapes d'une création de base Oracle 10g ;
- démarrage d'une base Oracle 10g ;
- arrêt d'une base Oracle 10g.

Ce chapitre aborde les différentes étapes de la vie d'une base Oracle 10g ; il présente tout d'abord les grandes lignes de son installation, puis les étapes en amont et en aval qui encadrent sa création. Ces points ne constituent que des généralités, ils vous fournissent le fil conducteur d'une installation Oracle 10g. Enfin, nous étudions dans le détail les étapes relatives à son démarrage et à son arrêt.

Avant toute installation du logiciel Oracle 10g ou toute création de bases, nous vous recommandons de consulter les règles OFA (*Optimum Flexible Architecture*) préconisées par Oracle et abordées au chapitre 9, *Les fichiers d'une base Oracle 10g*. Les chapitres 13, *L'installation d'Oracle 10g sous Windows*, et 14, *Création d'une base Oracle 10g*, s'intéressent dans le détail aux étapes de sa création.

## Étapes d'installation d'Oracle 10g

Toute installation Oracle 10g doit être planifiée. On peut effectuer facilement et rapidement une installation, mais si les choix initiaux ne sont pas judicieux, leur correction risque de s'avérer longue et délicate. Ce point est particulièrement vrai sous Windows où l'installation s'effectue très simplement. Les paragraphes suivants décrivent succinctement les différentes étapes à suivre pour installer Oracle 10g et créer une ou plusieurs bases de données.

### S'assurer des prérequis techniques

Les prérequis techniques sous Windows se limitent généralement à la vérification du Service Pack Windows requis par Oracle, de l'espace disque disponible et de la vérification de la mémoire installée.

À ce moment-là, il est important de concevoir l'arborescence des répertoires qui respectera l'architecture OFA. Cela vous permettra d'installer Oracle et de créer vos bases sans aucun souci ultérieur d'espace disque.

### L'installation

L'installation s'effectue sous le compte administrateur de Windows, car Oracle a besoin d'accéder à la base de registres ainsi qu'aux services. Comme Oracle 10g dispose d'une « boîte à outils » très importante, vous devez choisir les options à installer.

### Après l'installation

Comme pour la vérification des prérequis, seules quelques actions techniques comme l'intégration des bases Oracle dans vos sauvegardes sont nécessaires.

## Étapes d'une création de base

Une fois le logiciel Oracle 10g mis en place, on peut créer une ou plusieurs bases au moyen d'outils Oracle 10g, ou « manuellement », par l'ordre CREATE DATABASE. Dans les deux cas, les logiciels graphiques qui permettent de créer une base ne font « qu'habiller » un ordre CREATE DATABASE.

Quelle que soit l'option choisie, certaines étapes sont à réaliser en amont de la création de la base, d'autres après.

### Le rôle du service OracleServiceSID

Sous Windows, une base Oracle nécessite que son service OracleServiceSID (SID correspondant à l'identifiant de la base) soit créé et démarré avant la création ou le démarrage de la base. Ce point est détaillé au chapitre 5, *Oracle 10g sous Windows*.

## Étapes préliminaires à la création d'une base

Avant toute création de base, il faut planifier son installation et déterminer ses caractéristiques, dont son nom. Certaines options ne sont plus modifiables une fois la base créée. Elles sont déterminées lors de la création dans le fichier d'initialisation initSID.ora.

D'autres options, comme la détermination des dimensions des fichiers ou le jeu de caractères utilisé pour construire la base, sont spécifiées dans l'ordre SQL de création de la base.

## Étapes postérieures à la création d'une base

Une fois la base créée, lancez des scripts SQL pour construire le dictionnaire de la base de données. Ces scripts sont fournis par Oracle. Ils sont nombreux, car chacun est spécialisé et assure une option technique précise. N'installez que les options techniques nécessaires.

Ensuite, il faut créer les tablespaces dédiés aux applications et aux espaces de tri.

Vient ensuite tout ce qui a trait à vos applications : les utilisateurs, leurs droits, les schémas de données, les procédures stockées, les données, etc.

N'oubliez pas d'intégrer chaque nouvelle base de données dans vos procédures de sauvegarde.

## Base de données et instance

Chaque *base de données* Oracle 10g « en marche » est associée à une *instance*. Tout d'abord, le service OracleServiceSID doit être démarré. Il lance l'exécutable oracle.exe, qui attend l'ordre de démarrage de l'instance. Lors du démarrage de la base de données, de nombreux threads sont créés et la mémoire SGA (*System Global Area*) est allouée au sein du processus. Cet ensemble SGA et threads est appelé une *instance* Oracle 10g.

Une instance est une base de données « en marche » associant mémoire et threads. On parlera de base de données pour l'ensemble des fichiers qui la composent.

Une base de données Oracle 10g se compose des éléments suivants :

- fichiers : base de données, redo-log, contrôle ;
- threads : pour faire fonctionner la base ;
- mémoire réservée : la SGA.

Un serveur Windows peut accueillir une ou plusieurs instances (bases de données fonctionnant simultanément). Chaque instance possède alors son propre jeu de fichiers, son processus oracle.exe, ses threads et sa SGA. Ces bases de données et instances auront des vies et des administrations indépendantes les unes des autres.

Nous aborderons les avantages et inconvénients à disposer d'une ou plusieurs bases sur un seul serveur au chapitre 14, *Création d'une base Oracle 10g*.

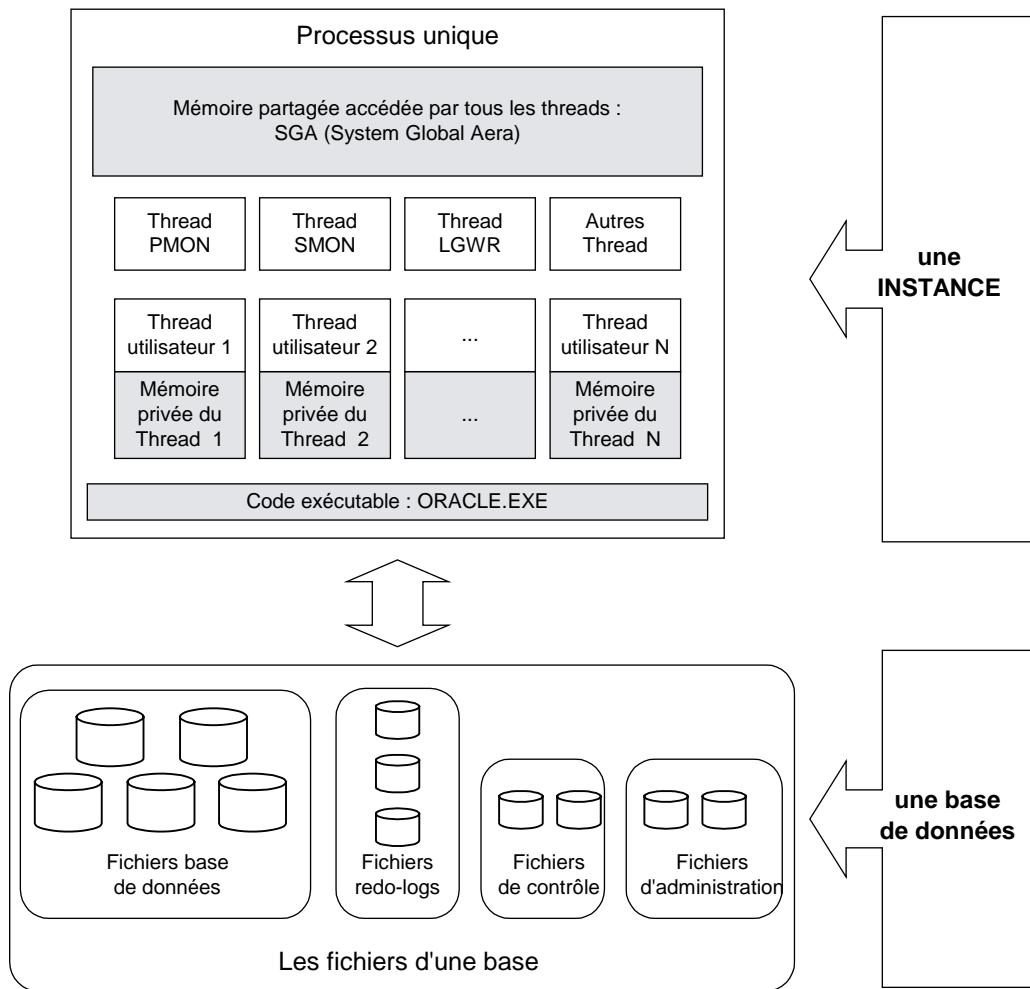


Figure 10-1

Instance et base de données

### Les priviléges administrateur

Le démarrage, l'arrêt et la restauration d'une base sont des options très importantes. Par souci de sécurité, elles nécessitent que les utilisateurs qui les effectuent possèdent des priviléges spéciaux. Comme le démarrage s'effectue base fermée, il n'est pas possible de conserver ces priviléges dans la base ; c'est pourquoi ils reposent sur des droits ou des mots de passe vérifiés et stockés hors de la base de données.

Il existe deux méthodes pour attribuer ce privilège à un utilisateur Windows (elles sont décrites au chapitre 24, *Stratégie de sécurité sous Windows* :

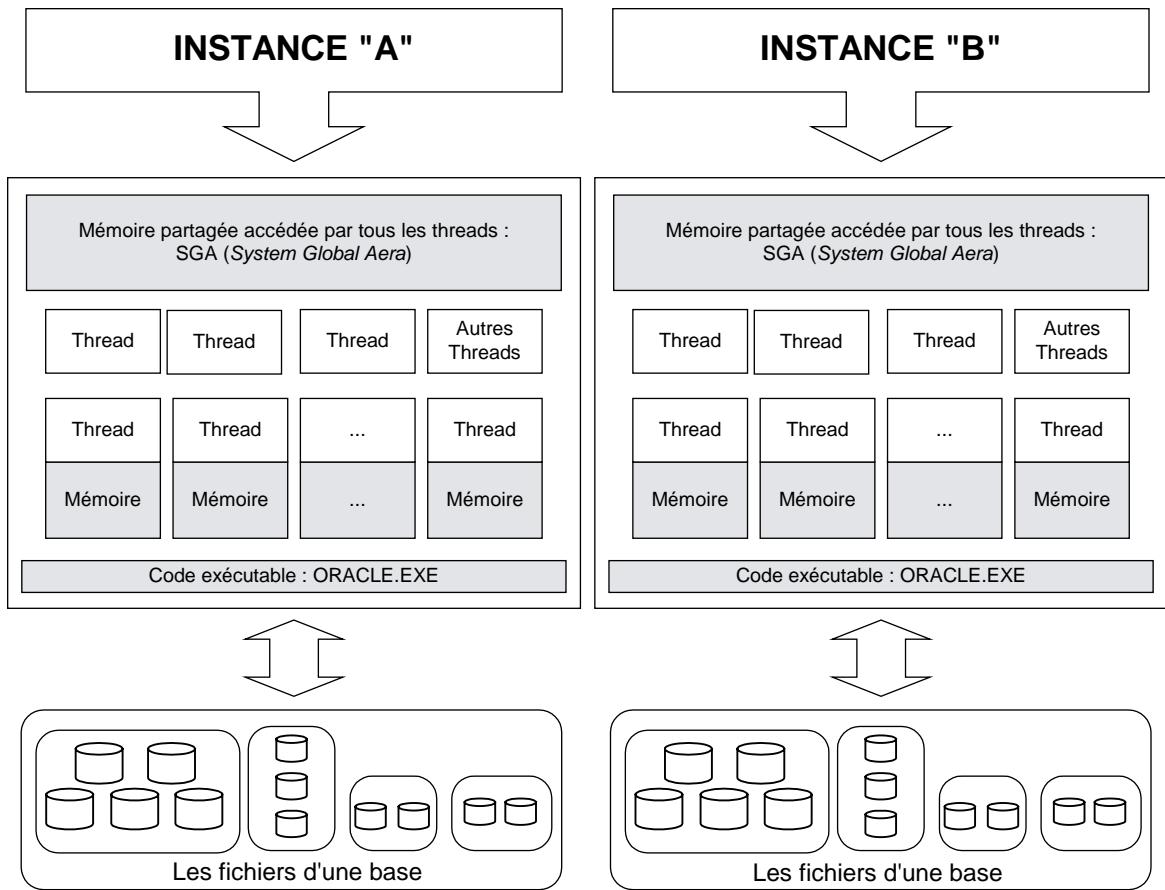


Figure 10-2

Deux instances sur un même serveur

- Un privilège au niveau du système d'exploitation fournit à un utilisateur Windows les droits nécessaires. C'est le cas de l'appartenance au groupe Windows *ORA\_DBA*.
- Un utilisateur déclaré dans la base, disposant du privilège Oracle *SYSDBA* ou *SYSOPER*, peut démarrer ou arrêter la base, si sa sécurité d'accès repose sur des fichiers mots de passe stockés dans des fichiers extérieurs à la base.

Nous vous conseillons fortement d'utiliser la première méthode : elle est facile à mettre en œuvre et sa sécurité est optimale.

Le groupe Windows *ORA\_DBA* est créé automatiquement lors de l'installation d'Oracle 10g. Lorsque vous êtes connecté à la base avec ce privilège, vous êtes placé dans le schéma propriété de *SYS*. Cela vous donne accès à tous les objets propriété de l'utilisateur *SYS*, dont le dictionnaire de données.

## Le fichier d'initialisation

Pour démarrer une instance, Oracle 10g a besoin de lire un fichier d'initialisation comprenant une liste de paramètres. Les valeurs sont relatives à cette instance et cette base de données. Son contenu est décrit au chapitre 9, *Les fichiers d'une base Oracle 10g*.

Le fichier d'initialisation recherché par défaut par Oracle 10g se nomme *initSID.ora*, *SID* étant l'identifiant de votre instance. Par exemple, pour l'instance TEST, on aura le fichier *initTEST.ora*. Rien n'empêche de disposer de plusieurs fichiers d'initialisation pour une même instance, il suffit d'indiquer celui utilisé au démarrage.

Certains paramètres figurant dans le fichier d'initialisation peuvent être modifiés une fois l'instance lancée. Si tous ne sont pas présents dans le fichier d'initialisation, des valeurs par défaut sont attribuées automatiquement.

Le fichier d'initialisation peut avoir la forme d'un fichier texte persistant, dans un format géré par Oracle, le fichier *pfileSID.ora*. Ce point est traité au chapitre 23, *Gestion de l'espace disque et des fichiers*.

## Démarrer une base Oracle 10g

Afin de la rendre disponible à l'ensemble des utilisateurs, le démarrage d'une base Oracle 10g s'effectue en quatre étapes :

- démarrage du service OracleServiceSID ;
- démarrage de l'instance ;
- montage de la base de données ;
- ouverture de la base de données.

Dans les précédentes versions d'Oracle, l'outil privilégié pour démarrer et arrêter une base était Server Manager. Cet outil a disparu car il est maintenant intégré dans SQL\*Plus.

### Démarrage du service OracleServiceSID

Le démarrage du service OracleServiceSID s'effectue à partir du gestionnaire de services Windows ou par la commande :

 C:\> net start OracleServiceSID

Le démarrage du service peut entraîner celui de l'instance. Ce point est évoqué plus loin dans ce chapitre.

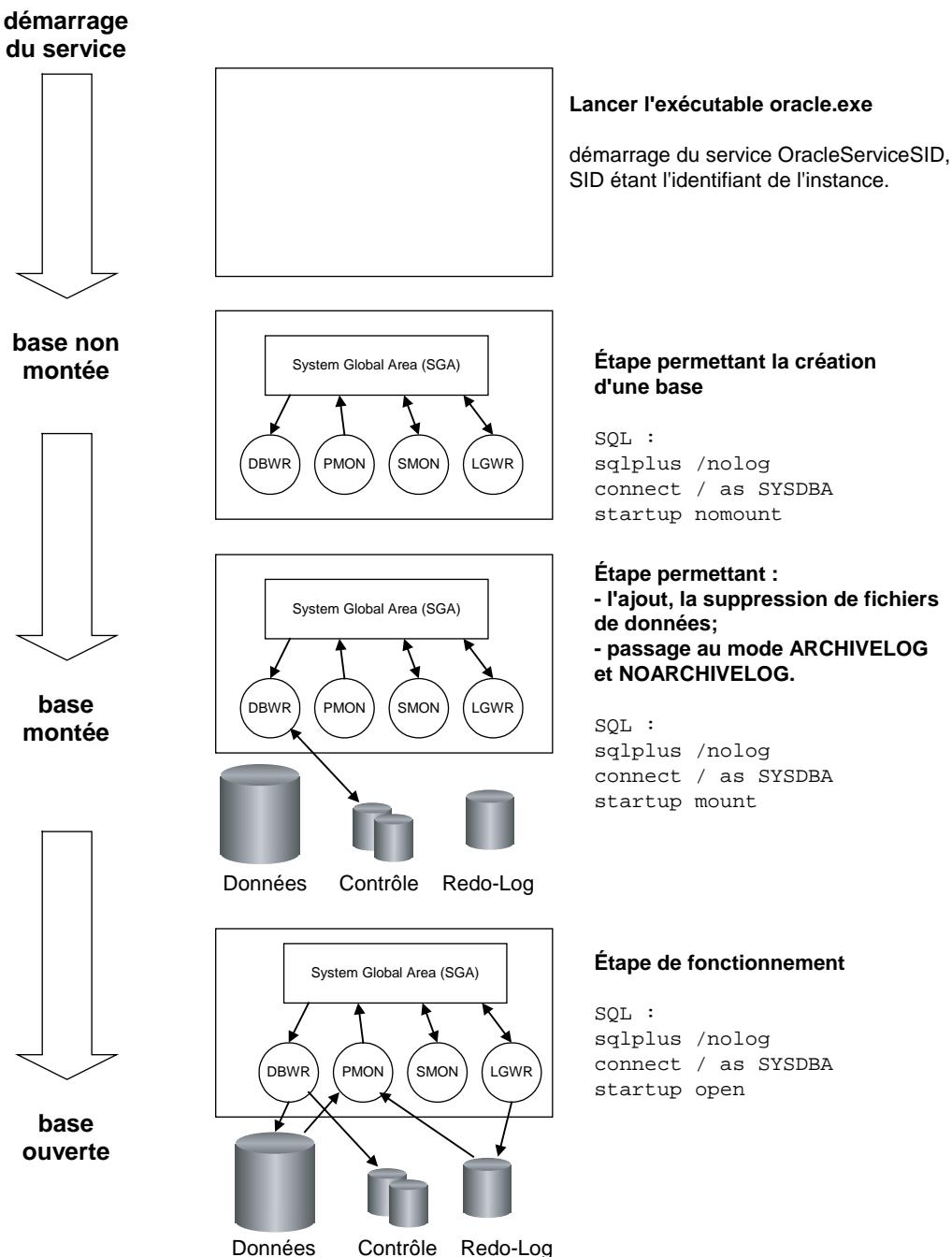


Figure 10-3

*Étapes du démarrage d'une instance*

## Démarrage d'une instance

Pour démarrer une instance, Oracle 10g lit les valeurs présentes dans le fichier d'initialisation (pfileID.ora s'il existe, puis initSID.ora) puis alloue la SGA (*System Global Area*) en mémoire et lance les processus. À ce moment, aucune base de données, c'est-à-dire aucun fichier, n'est associée à cette instance.

```
C:\> sqlplus /nolog
connect / as sysdba
startup nomount pfile=C:\oracle\admin\TEST\scripts\initTEST.ora

Instance ORACLE démarrée
Zone globale système (SGA) totale de 10006476 octets
  Fixed Size           65484 octets
  Variable Size        8740864 octets
  Database Buffers     1126400 octets
  Redo Buffers          73728 octets
```

## Montage de la base de données

Le montage associe les fichiers de contrôle à l'instance créée. Pour cela, le fichier d'initialisation contient l'emplacement des fichiers de contrôle (précisé par le paramètre **CONTROL\_FILES** du fichier initSID.ora). Ceux-ci contiennent la liste de tous les autres fichiers de données et redo-log qui composent la base.

```
-- suite de l'exemple précédent
alter database mount;

Instruction traitée
```

## Ouverture de la base de données

L'ouverture de la base de données la rend disponible à l'ensemble des utilisateurs. Tout utilisateur peut s'y connecter et travailler.

```
-- suite de l'exemple précédent
alter database open;

Instruction traitée
```

## Démarrage en un seul ordre

Toutes ces étapes peuvent être réalisées par un seul ordre SQL :

```
$ sqlplus /nolog
connect / as sysdba
startup pfile=C:\oracle\admin\TEST\scripts\initTEST.ora
```

```

Instance ORACLE démarrée
Zone globale système (SGA) totale de 10006476 octets
  Fixed Size           65484 octets
  Variable Size       8740864 octets
  Database Buffers   1126400 octets
  Redo Buffers        73728 octets
  Base de données montée
  Base de données ouverte

```

On retrouve les trois actions : le démarrage de l'instance, le montage de la base puis son ouverture.

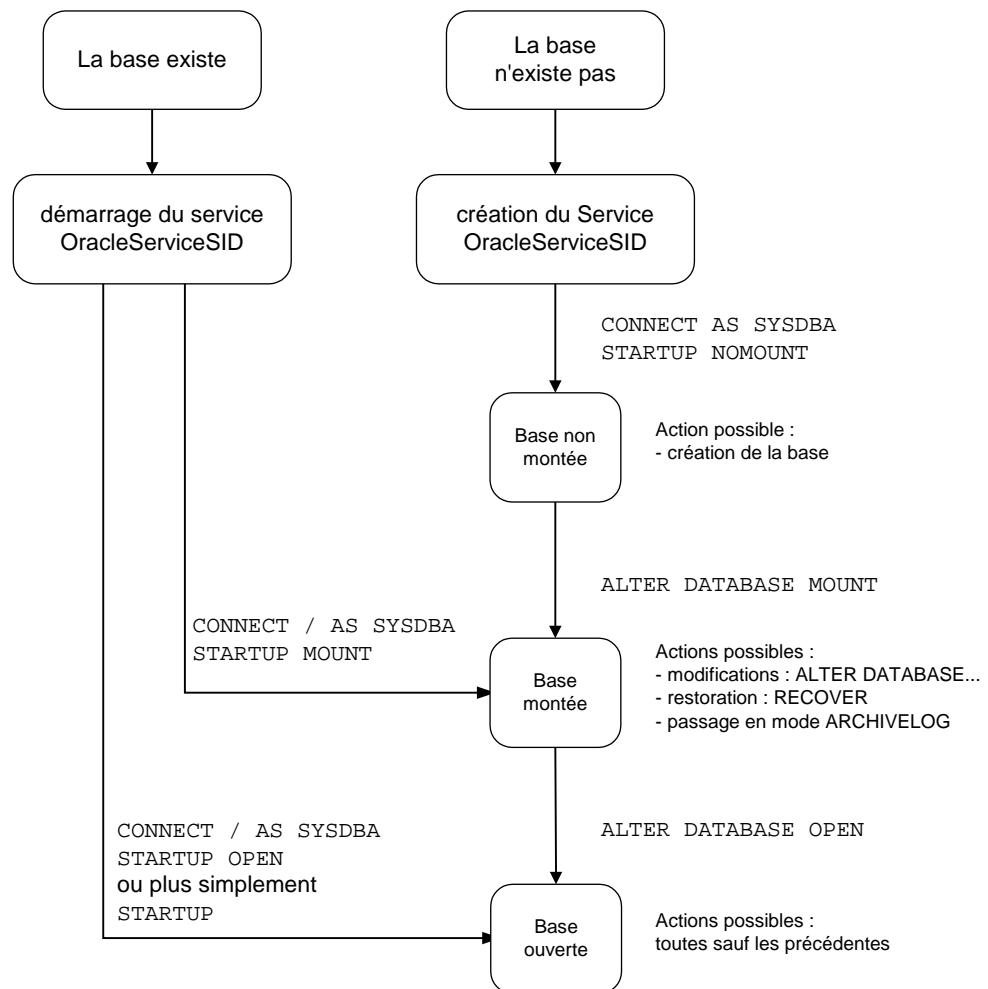


Figure 10-4

Ordres SQL lors du démarrage d'une instance

## Arrêter une base Oracle 10g

Lors de l'arrêt d'une base Oracle 10g, les étapes sont identiques à celles rencontrées au démarrage :

- fermeture de la base de données ;
- démontage de la base de données ;
- arrêt de l'instance ;
- arrêt du service OracleServiceSID.

L'arrêt du service OracleServiceSID entraîne plus ou moins brutalement l'arrêt des étapes précédentes, selon le paramétrage choisi.

### Les différents types d'arrêts

Lorsque vous fermez une base Oracle 10g, des utilisateurs peuvent être connectés. Comment sont alors traitées les transactions en cours ? Pour cela, il existe différentes options d'arrêts :

- SHUTDOWN NORMAL ;
- SHUTDOWN IMMEDIATE ;
- SHUTDOWN TRANSACTIONAL ;
- SHUTDOWN ABORT.

Le SHUTDOWN NORMAL attend que l'ensemble des utilisateurs soit déconnecté pour fermer la base. L'inconvénient, c'est que l'on peut attendre longtemps si, par exemple, un utilisateur s'est absenté sans fermer son application : sa session est toujours active. C'est pourquoi le SHUTDOWN IMMEDIATE déconnecte automatiquement tous les utilisateurs, puis ferme correctement votre base de données.

Le SHUTDOWN TRANSACTIONAL offre une alternative à ces méthodes : il attend la fin de toutes les transactions en cours en interdisant toute nouvelle transaction.

Pour arrêter votre base avant une sauvegarde, utilisez toujours un SHUTDOWN IMMEDIATE car un SHUTDOWN NORMAL risque de vous mettre en attente sans délai prévisible de fermeture. Même si aucun utilisateur n'est plus connecté à Oracle, il reste souvent des connexions liées au fonctionnement d'agents de surveillance, d'un « pool » de connexion PHP, etc.

Lors d'un SHUTDOWN NORMAL ou IMMEDIATE, Oracle 10g écrit toutes les informations modifiées encore présentes dans la SGA, dans les fichiers de données et redo-log. Ensuite, l'ensemble des fichiers *online* est fermé. (En effet, un fichier *offline* n'a plus de lien avec la base de données. Lorsque vous démarrez la base, un fichier ou un tablespace *offline* reste *offline*). À cet instant, la base est fermée et inaccessible pour des opérations normales. Si elle reste néanmoins montée, les fichiers de contrôle demeurent ouverts.

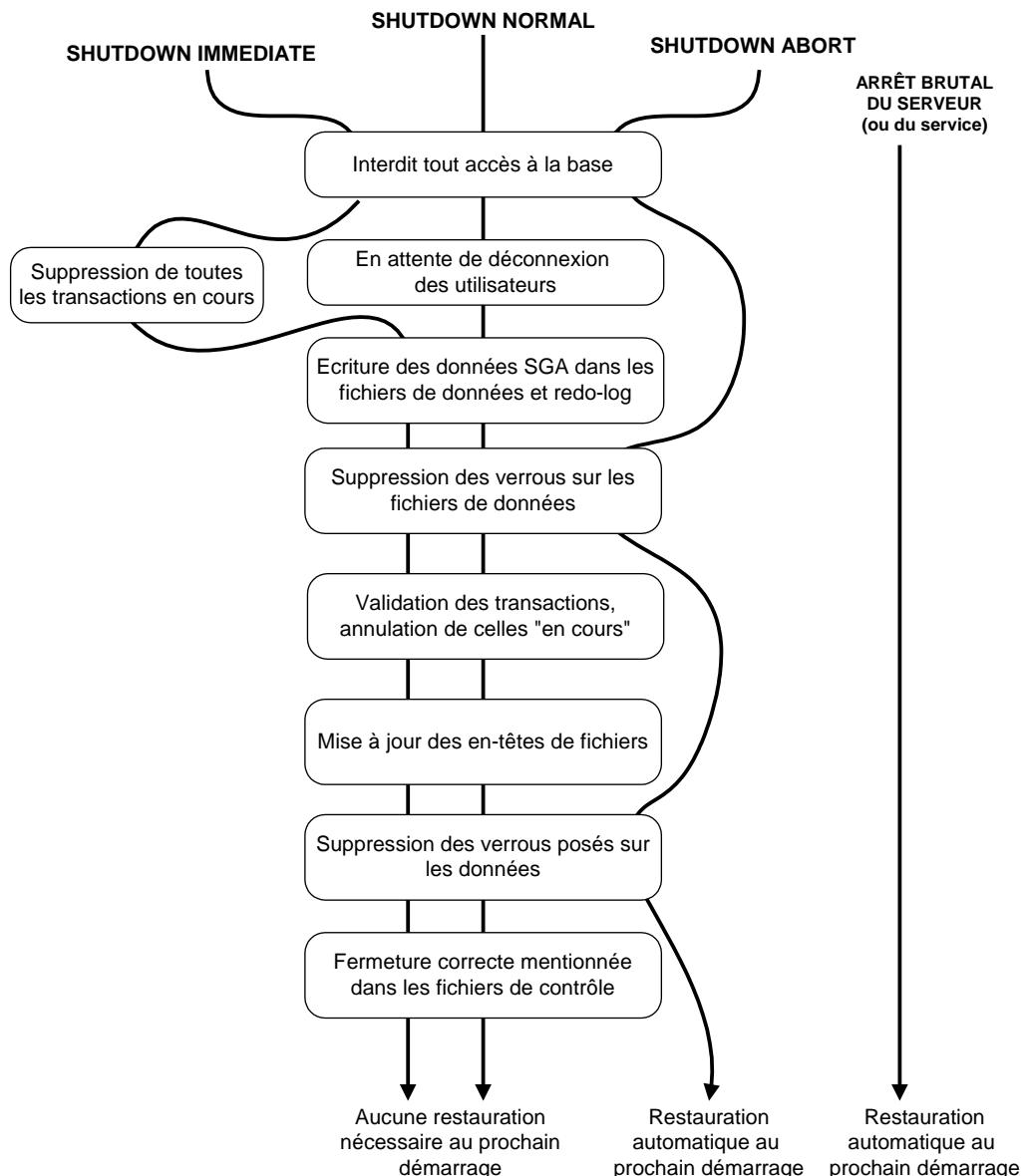


Figure 10-5

Les différents types de fermetures d'une base

Le SHUTDOWN ABORT est un arrêt violent qui correspond à supprimer brutalement le processus oracle.exe, ses threads, la SGA et libérer tous les verrous sur les fichiers. Cette option nécessite une restauration automatique au démarrage suivant. Cette méthode très rapide s'apparente à un arrêt électrique brutal de votre machine. Dans ce cas, il n'y a

aucune écriture des données de la SGA dans les fichiers de données ou redo-log. En conséquence, la prochaine ouverture de base nécessitera une opération de restauration, qu'Oracle 10g effectue automatiquement.

La restauration automatique de l'instance en cas d'arrêt brutal est l'un des points forts d'Oracle.

## Le paramétrage du service *OracleServiceSID*

Des clés de registre coordonnent l'arrêt du service *OracleServiceSID* avec celui de l'instance. Elles sont situées en `HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\KEY_OraDb10g_home1` et concernent aussi bien le démarrage que l'arrêt de l'instance.

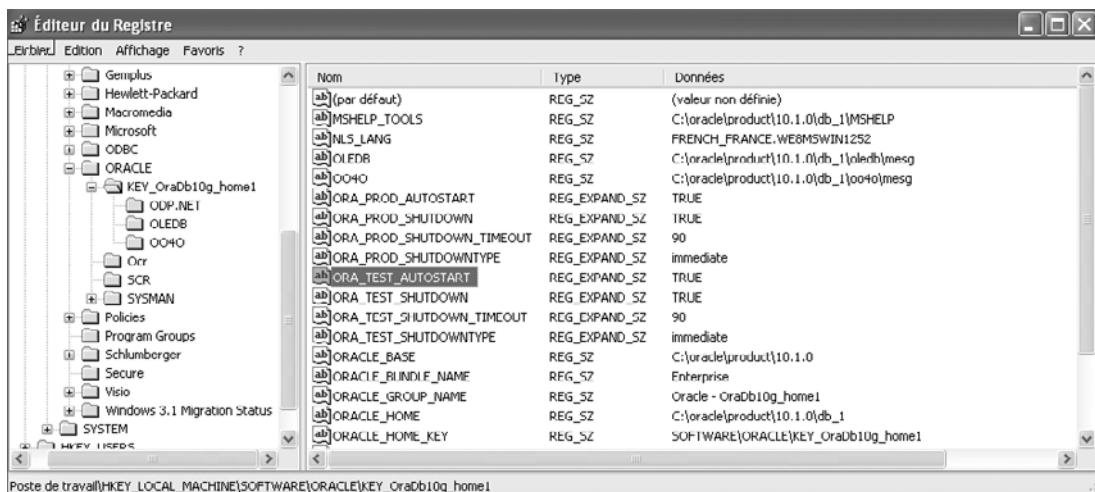


Figure 10-6

Clés du registre liées au service *OracleServiceSID*

Dans la figure précédente, on retrouve des clés identiques pour deux instances différentes : TEST et PROD.

ORA_SID_AUTOSTART	Indique si la base doit être démarrée en même temps que le service OracleServiceSID. Si le service est paramétré en démarrage automatique, il sera lancé en même temps que l'ordinateur et la base Oracle démarra aussi.
ORA_SID_SHUTDOWN	Indique si la base doit être arrêtée en même temps que l'arrêt du service OracleServiceSID.
ORA_SID_SHUTDOWN_TIMEOUT	Période de temps accordée par le service pour que la base s'arrête. Au-delà, le service s'arrête, quelle que soit l'étape d'arrêt de la base.
ORA_SID_SHUTDOWNTYPE	Immediate, normal ou abort

Pour éviter l'accès direct à la base de registre, l'utilitaire Oracle Administration Assistant for Windows offre un accès graphique à ces clés de registre. Il est décrit au chapitre 5, *Fonctionnement d'Oracle 10g sous Windows*.

## Fermeture de la base de données

La fermeture de la base de données la rend indisponible à l'ensemble des utilisateurs. Dans le cas suivant, la base est fermée, mais reste montée :

```
alter database close immediate;
```

Instruction traitée

## Démontage de la base de données

Le démontage de la base de données supprime l'association entre les fichiers de la base et l'instance. Dans le cas suivant, la base fermée est démontée, mais l'instance existe encore :

```
-- suite de l'exemple précédent
```

```
alter database dismount ;
```

Instruction traitée

## Arrêt de l'instance

Pour arrêter l'instance, effectuez une fermeture « normale » de la base ; elle enchaîne les étapes déjà effectuées puis termine l'instance. Les processus sont supprimés, la SGA libérée.

```
-- suite de l'exemple précédent
```

```
shutdown ;
```

ORA-01507: base de données non montée

Instance Oracle arrêtée

## Fermeture en un seul ordre

Toutes ces étapes peuvent être réalisées par un seul ordre SQL :

```
$ sqlplus /nolog
```

```
connect as sysdba
```

```
shutdown immediate
```

Base de données fermée

Base de données démontée

Instance ORACLE arrêtée

On retrouve les trois actions : la fermeture, le démontage de la base puis l'arrêt de l'instance.

## Résumé

Ce chapitre a abordé succinctement l'installation d'Oracle 10g ainsi que la création d'une base de données. Puis, nous avons abordé dans le détail les différentes options de démarrage et d'arrêt d'une instance Oracle 10g.