



BAUBRY

ORACLE EXPLOITATION

"Ce produit a été fabriqué selon une organisation qualité conforme à la norme ISO 9001 certifiée par l'AFAQ N°1993/1747".

"La loi du 11 mars 1957 complétée par la loi du 3 juillet 1985 interdit les copies ou reproductions destinées à une utilisation collective. Toute représentation ou reproduction intégrale ou partielle faite par quelque procédé que ce soit, sans le consentement de l'auteur ou de ses ayants causes, est illicite et constitue une contrefaçon sanctionnée par les articles 425 et suivants du code pénal".



DESCRIPTIF DU STAGE

02898E**ORACLE : EXPLOITATION****OBJECTIF :**

Acquérir les éléments de base nécessaires à l'accueil et à l'exploitation d'un progiciel ou d'une application fonctionnant avec ORACLE.

PARTICIPANTS ET PREREQUIS :

Chefs de projets, administrateurs de données, exploitant maîtrisant le SQL (stage 03210).

MÉTHODES PÉDAGOGIQUES :

Exposés et travaux pratiques sur machine UNIX(Escala) ou WINDOWS NT pour exploiter une base ORACLE.

PROGRAMME :

- Présentation générale, environnement, vocabulaire et architecture du serveur Oracle.
- Etude de l'implantation du schéma physique dans la base : diagnostic des structures physiques (tablespaces, rollback segments, segments de données), de leur dimensionnement et de leur localisation.
- Sécurité des accès à la base : privilèges-système, privilèges-objet rôles et profiles-utilisateur.
- Arrêt, démarrage de la base.
- Surveillance et prévention à l'aide de requêtes sur le dictionnaire et Enterprise Manager.
- Eléments et techniques d'optimisation du serveur Oracle.8 : paramètres de l'instance.
- Les outils d'administration : SQL*Plus, Server Manager, Enterprise Manager, Exp, Imp, SQL*Loader.
- Sauvegarde physique, archivage des données et restauration de la base.

DURÉE :

2 jours



REVISION

REV #	DATE
0	février 2001
1	février 2002
2	mai 2003



SOMMAIRE DU STAGE

TITRE DES MODULES	No
Présentation générale	1
Démarrage - arrêt	2
SQL*NET	3
Tablespaces et Segments	4
Sécurité et accès à la base	5
Tables et Index	6
Traces, Sessions et Transactions	7
Export/Import SQL*Loader	8
Sauvegardes et restaurations physiques	9
Solutions des travaux pratiques	s

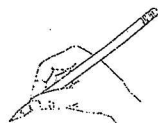


BASES DE DONNEES PRESENTATION GENERALE

Objectif :

A la fin de ce module, vous serez capable de connaître les principales fonctionnalités du système de gestion de base de données Oracle versions 7, 8i et 9i:

- Principes généraux des bases de données
- Intégrité transactionnelle
- Paramétrage et contrôle
- L'administrateur de la base de données
- Présentation d'Oracle



Principes Généraux des bases de données (rappels)

Un Système de Gestion de Base de Données est un outil pour gérer un ensemble d'informations.

Une base de données se compose principalement d'un ensemble de tables.

Parmi ces tables, il y a des tables systèmes (le dictionnaire) et les tables données applicatives créées par les utilisateurs de la base.

La façon d'accéder aux données se fait toujours par l'intermédiaire de requêtes SQL.

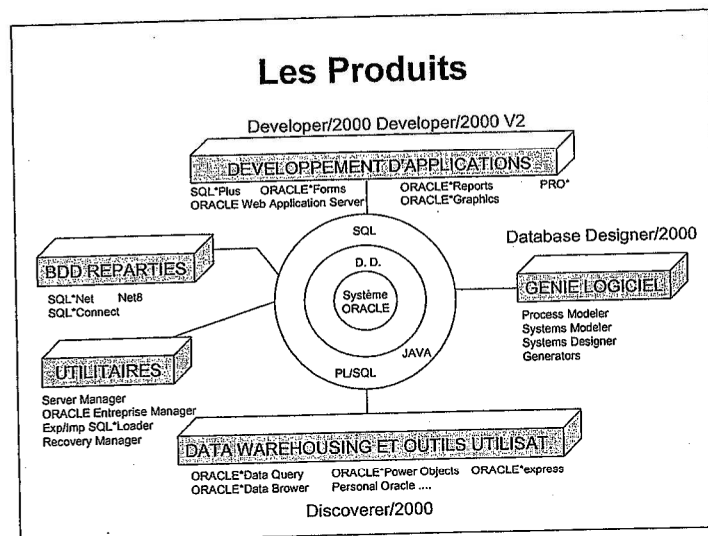
En Français on parle de :

S.G.B.D.R = Système de Gestion de Base de Données Relationnelle

En Anglais :

R.D.B.M.S = Relational DataBase Management System

ORACLE est disponible sur un grand nombre de plate-formes : IBM-MVS, BULL-GCOS7, DEC-VMS, NT, la plupart des UNIX et LINUX.



*** SQL*Plus**

Interface interactive.

Outil principal d'administration en Oracle9i.

Chargement de commandes SQL,

Exécution de commandes SQL, de procédures, fonctions ou ensembles de procédures PL/SQL,

Mise en page simplifiée d'états.

*** Oracle*Forms ou Oracle Forms Builder**

Générateur d'applications graphiques transactionnelles.

*** Oracle*Reports ou Oracle Reports Builder**

Générateur d'états graphiques.

*** PRO*...**

Interfaces pré compilateurs (langage C, Cobol, Fortran), avec accès à travers un langage hôte de 3^{ème} génération.

*** Import, Export**

Sauvegarde et restauration logique des données de la base.

*** BASES DE DONNEES REPARTIES (SQL*Net, Net8, Oracle Net, SQL*Connect)**

Gestion d'architecture distribuée : traitements distribués, bases de données réparties, bases de données répliquées.

*** SQL*Loader**

Utilitaire de chargement de la base à partir de fichiers de données O.S séquentiels.

*** Server Manager (svrmgrl) Plus supporté en 9.i (remplacé par SQL plus)**

Utilitaire principal d'administration de bases de données jusqu'à Oracle8i.

*** Oracle Enterprise Manager (OEM)**

Ensemble d'utilitaires graphiques d'administration, à installer sur un poste client WINDOWS NT/95/98/2000.

etc...

LE SQL

• SQL DML (Data Manipulation Language) Manipulation du contenu d'une table.

Interrogation	SELECT
Insertion	INSERT
Modification	UPDATE
Destruction	DELETE

• SQL DDL (Data Definition Language) Création, modification, destruction d'un objet.

Creation	CREATE
Modification	ALTER
Destruction	DROP

• SQL DCL (Data Control Language) Validation/invalidation de privilèges

Attribution	GRANT
Suppression	REVOKE

Quelque soit l'interface utilisée : terminal ASCII ou poste graphique, les programmes clients vont soumettre des requêtes SQL.

Le SQL DML est constitué de deux sous-ensembles

La sélection (SELECT)

La modification des données (DML)

Les ordres DML sont confirmés ou annulés explicitement avec les ordres SQL suivant :

confirmation	COMMIT
annulation	ROLLBACK

Implicitement lors d'une fin de session normale toutes les modifications seront validées. En cas contraire les modifications seront annulées.

Les Tables

	Col1	Col2	Col3	Col4

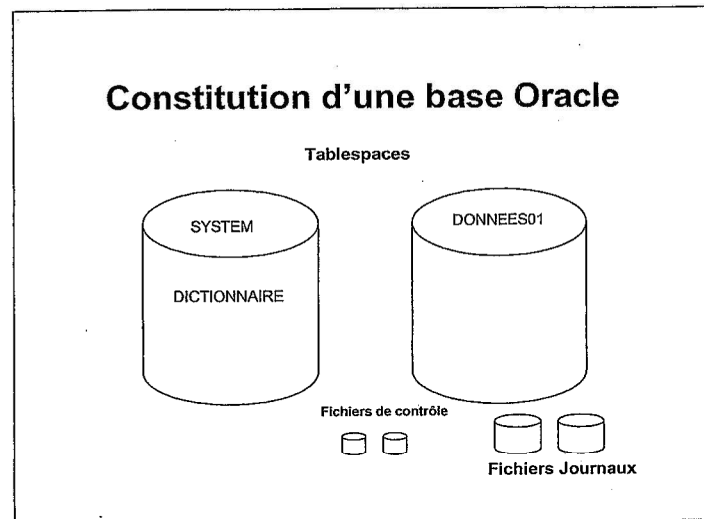
Ligne ou tuple →

↑ Colonne ou attribut ou rubrique

En SQL on utilise aussi le terme «relation» pour une table.

La définition des tables est dans le dictionnaire :

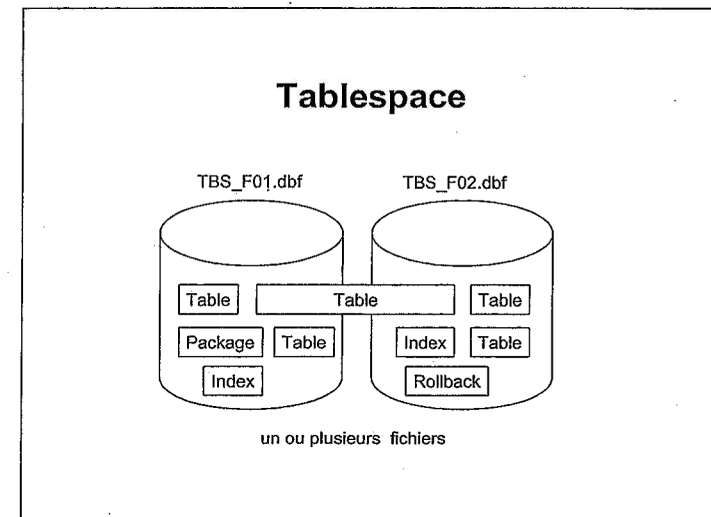
- Nombre et type des colonnes
- Contraintes par exemple colonne obligatoirement renseignée (NOT NULL)



Les espaces de table "**tablespaces**" sont des réservoirs de données qui sont physiquement constitués de fichiers ou de morceaux de disques (volumes logiques).

Les fichiers journaux "**redo log files**" servent à enregistrer toutes les modifications apportées à la base. Ils seront utilisés en cas de reprise sur incident.

Les **fichiers de contrôle** contiennent la description de la base : noms et état des fichiers la constituant, numéro de séquence du redo log courant, numéro du dernier point de contrôle...



Espace logique - réservoir de données

Les tables, les index sont dans des «tablespaces».

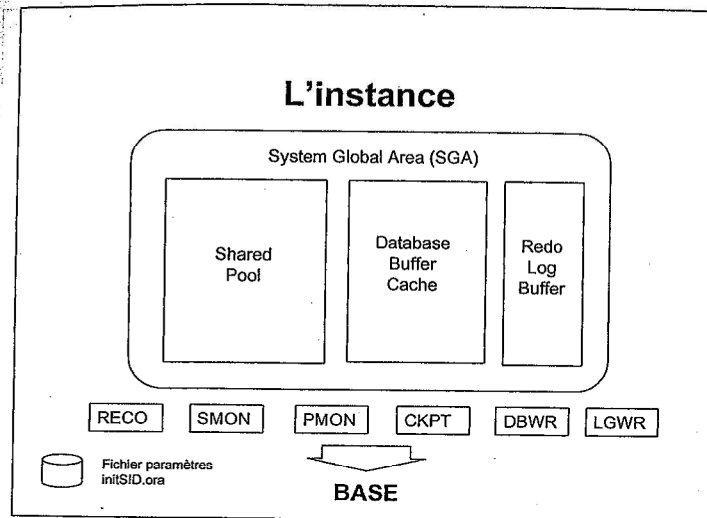
Les packages et les rollback occupent aussi de l'espace physique dans les tablespaces.

En Oracle 7:

Une table ou un index est dans un et un seul tablespace.

A partir d' Oracle 8:

Les tables et les index peuvent être partitionnés et donc dans plusieurs tablespaces. C'est au niveau de la partition que l'on va définir le tablespace d'accueil.



La SGA est une zone de mémoire partagée par les différents processus. Elle est identifiée par son SID.

Les programmes utilisateurs utilisent la variable ORACLE_SID pour connaître l'instance de travail.

- sous UNIX (Bourne ou Korn Shell)
\$ ORACLE_SID=T815; export ORACLE_SID
- sous WINDOWS
C:\> set ORACLE_SID=ORCL

Les processus et leur Rôle:

DBWn : **D**ata **B**ase **W**riter, écrit les blocs modifiés du cache de données sur les disques. Compte tenu de la journalisation (Redo log), les blocs ne sont pas forcément écrits à la validation des transactions (nombre de DBW de 1 à 10).

LGWR : **L**o**G** **W**riter, écrit dans les fichiers Redo Log online le contenu du cache Redo Log.

SMON : **S**ystem **M**onitor, assure la récupération à chaud d'une instance ou d'un fichier de la base de données.

PMON : **P**rocess **M**onitor, récupère un processus-client qui se termine anormalement, par nettoyage dans la SGA : mise au net des caches et libération des ressources encore détenues.

process d'archive **ARCn** : **ARC**hiver, archive les fichiers Redo Log online sur des fichiers offline de validité plus longue, lorsque la base est en ARCHIVELOG. Le nombre de processus peut varier de 1 à 10.

LCK 0 à n : assure la gestion de plusieurs instances en même temps sur la même base, uniquement en cas d'utilisation de l'option Oracle Parallel Server (OPS).

CKPT : surveille et prend le relais des processus LGWR et DBWR. Processus non nécessaire pour de petites bases de données (V7). Activé par défaut à partir de V8, fait des demandes d'écriture dans les fichiers de la base et écrit le checkpoint dans les fichiers de contrôle.

RECOverer : assure la cohérence des données dans les transactions distribuées sur plusieurs bases de données physiques réparties.

Et aussi

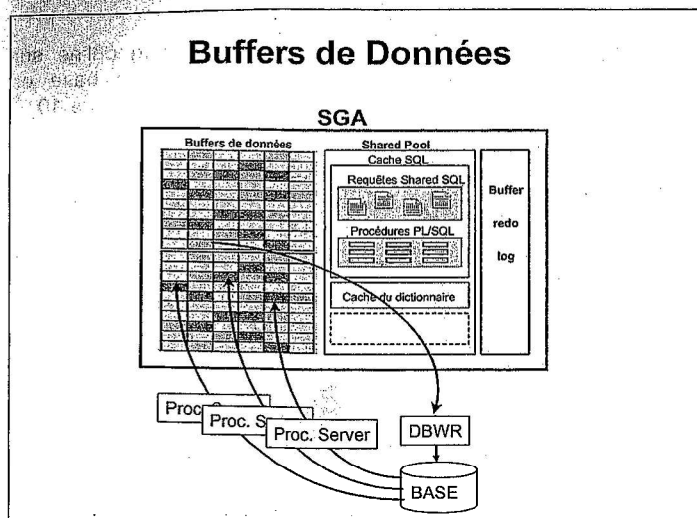
LISTENER : Relier un processus-client USER au processus DISPATCHER le plus adéquat.

DISPATCHER, D 000 à D nnn : assure le multiplexage de plusieurs processus-client USER avec la SGA en cas d'utilisation du MTS (Multi Thread System).

SHARED SERVER, S 000 à S nnn : exécutent les traitements des processus-client USER, par lecture du cache REQUEST QUEUE en cas d'utilisation du MTS.

DEDICATED SERVER : utilise un serveur particulier et dédié aux travaux d'administration (Server Manager) ou de batch, qui n'est pas ausculté par le LISTENER ou par l'un des DISPATCHERS mis en œuvre.

Buffers de Données

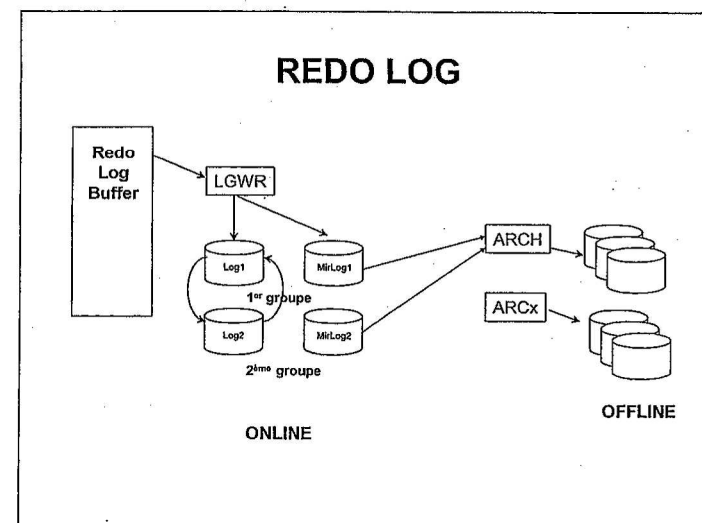


Le buffer de données joue le rôle d'un cache pour les informations contenues dans la base. Il est constitué de blocs de taille fixe jusqu'en Oracle8i.

Les process serveurs dédiés ou partagés chargent les blocs de données et éventuellement les modifient. Les DBWRn recopient dans la base ceux qui ont été modifiés.

En version 9i on peut gérer plusieurs tailles de blocs rassembler dans des zones différentes du cache.

REDO LOG



Les modifications de la base sont enregistrées dans le «redo log buffer».

Le processus **LGWR** sécurise les informations dans les fichiers redo log qui peuvent être en miroir, deux groupes au moins sont nécessaires.

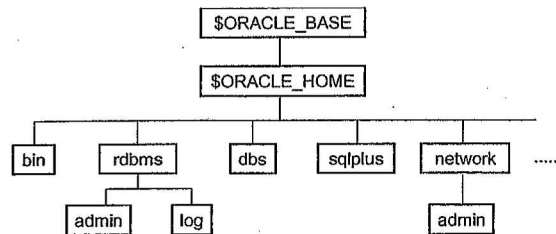
Les groupes sont écrits circulairement.

En option, ils peuvent être archivés par :

- le processus **ARCH** en version 7
- les processus **ARC1 ARC2 ...** à partir de la version 8i

dans des fichiers externes numérotés.

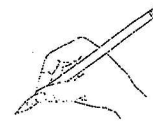
Répartition des fichiers ORACLE

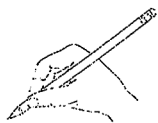


Cette architecture est commune aux différentes plates-formes (UNIX, WINDOWS...), elle est appelée **OFA** «Optimal Flexible Architecture».

ORACLE_BASE est le répertoire d'installation de tous les produits ORACLE toutes versions confondues.

Chaque version installée aura un ORACLE_HOME différent.





BASES DE DONNEES DEMARRAGE - ARRET

Objectif :

A la fin de ce module, vous serez capable de démarrer et arrêter une instance, visualiser les paramètres :

- Le fichier de paramètres
- L'utilitaire SVRMGRL
- Démarrer une instance
- Arrêter une instance
- Autorisation des connexions
- Visualiser les paramètres



Le Fichier de paramètres d'instance

Oracle fournit un exemple de fichier de paramètres:

`$ORACLE_HOME/dbs/init.ora`

A partir de cet exemple, on va créer un fichier `initSID.ora` qui contiendra les paramètres de l'instance de nom «SID».

Les paramètres de désignation à modifier initialement sont:

`db_name`
`control_files`

Les paramètres de capacité et de performance servent à adapter et optimiser la SGA.

Fichier `init.ora`

```
#
# $Header: init.ora 05-jun-97.14:56:46 hpiao Exp $
#
# Copyright (c) 1991, 1997 by Oracle Corporation
# NAME
#   init.ora
# FUNCTION
# NOTES
# MODIFIED
#   hpiao      06/05/97 - fix for 803
#   .../...
#   Wijaya     02/05/91 - remove obsolete parameters
#
#####
# Example INIT.ORA file
#
# This file is provided by Oracle Corporation to help you
# customize
# your RDBMS installation for your site. Important system
# parameters
# are discussed, and example settings given.
#
```

```
# Some parameter settings are generic to any size installation.
# For parameters that require different values in different size
# installations, three scenarios have been provided: SMALL, MEDIUM
# and LARGE. Any parameter that needs to be tuned according to
# installation size will have three settings, each one commented
# according to installation size.
#
# Use the following table to approximate the SGA size needed for the
# three scenarios provided in this file:
#
# -----Installation/Database Size-----
#           SMALL           MEDIUM           LARGE
# Block      2K      4500K      6800K      17000K
# Size       4K      5500K      8800K      21000K
#
# To set up a database that multiple instances will be using, place
# all instance-specific parameters in one file, and then have all
# of these files point to a master file using the IFILE command.
# This way, when you change a public
# parameter, it will automatically change on all instances. This is
# necessary, since all instances must run with the same value for many
# parameters. For example, if you choose to use private rollback segments,
# these must be specified in different files, but since all gc_*
# parameters must be the same on all instances, they should be in one file.
#
# INSTRUCTIONS: Edit this file and the other INIT files it calls for
# your site, either by using the values provided here or by providing
# your own. Then place an IFILE= line into each instance-specific
# INIT file that points at this file.
#####
# replace DEFAULT with your database name
db_name=DEFAULT (nom de l'instance = $ORACLE_SID)

db_files = 80                                # SMALL
# db_files = 400                             # MEDIUM
# db_files = 1000                            # LARGE

db_file_multiblock_read_count = 8            # SMALL
# db_file_multiblock_read_count = 16         # MEDIUM
# db_file_multiblock_read_count = 32         # LARGE

db_block_buffers = 100                       # SMALL
# db_block_buffers = 550                     # MEDIUM
# db_block_buffers = 3200                    # LARGE
```

```
shared_pool_size = 3500000                  # SMALL
# shared_pool_size = 5000000                # MEDIUM
# shared_pool_size = 9000000                # LARGE

log_checkpoint_interval = 10000

processes = 50                              # SMALL
# processes = 100                          # MEDIUM
# processes = 200                          # LARGE

parallel_max_servers = 5                   # SMALL
# parallel_max_servers = 4 x (number of CPUs) # MEDIUM
# parallel_max_servers = 4 x (number of CPUs) # LARGE

log_buffer = 8192                           # SMALL
# log_buffer = 32768                       # MEDIUM
# log_buffer = 163840                      # LARGE

sequence_cache_entries = 10                # SMALL
# sequence_cache_entries = 30              # MEDIUM
# sequence_cache_entries = 100             # LARGE

sequence_cache_hash_buckets = 10           # SMALL
# sequence_cache_hash_buckets = 23        # MEDIUM
# sequence_cache_hash_buckets = 89        # LARGE

# audit_trail = true                        # if you want auditing
# timed_statistics = true                  # if you want timed statistics
max_dump_file_size = 10240                 # limit trace file size to 5 Meg each

# Uncommenting the line below will cause automatic archiving if archiving
# has been enabled using ALTER DATABASE ARCHIVELOG.
# log_archive_start = true
# log_archive_dest = disk$rdbs:[oracle.archive]
# log_archive_format = "T%TS%S.ARC"

# If using private rollback segments, place lines of the following
# form in each of your instance-specific init.ora files:
# rollback_segments = (name1, name2)

# If using public rollback segments, define how many
# rollback segments each instance will pick up, using the formula
# # of rollback segments= transactions/transactions_per_rollback_segment
# In this example each instance will grab 40/10 = 4:
# transactions = 40
# transactions_per_rollback_segment = 10
```



```
#Global Naming--enforce that a dblink has same name as the db it connects
to
global_names = TRUE

# Edit and uncomment the following line to provide the suffix that will be
# appended to the db_name parameter (separated with a dot) and stored as
the
# global database name when a database is created. If your site uses
# Internet Domain names for e-mail, then the part of your e-mail address
# after the '@' is a good candidate for this parameter value.

# db_domain = us.acme.com      # global database name is db_name.db_domain

# FOR DEVELOPMENT ONLY, DEFAULT TO SINGLE-PROCESS
# single_process = TRUE

# FOR DEVELOPMENT ONLY, ALWAYS TRY TO USE SYSTEM BACKING STORE
# vms_sga_use_gblpagfil = TRUE

# FOR BETA RELEASE ONLY. Enable debugging modes. Note that these can
# adversely affect performance. On some non-VMS ports the db_block_cache_*
# debugging modes have a severe effect on performance.

#_db_block_cache_protect = true      # memory protect buffers
#event ="10210 trace name context forever, level 2"# data block checking
#event ="10211 trace name context forever, level 2"# index block checking
#event ="10235 trace name context forever, level 1"# memory heap checking
#event ="10049 trace name context forever, level 2"# memory protect cursors

# define parallel server (multi-instance) parameters
#ifile = ora_system:itnps.ora

# define two control files by default
control_files = (ora_control1, ora_control2)

# Uncomment the following line if you wish to enable the Oracle Trace
# product to trace server activity. This enables scheduling of server
# collections from the Oracle Enterprise Manager Console.
# Also, if the oracle_trace_collection_name parameter is non-null,
# every session will write to the named collection, as well as enabling you
# to schedule future collections from the console.

# oracle_trace_enable = TRUE
```



SQLPLUS et SERVER MANAGER (SVRMGRL)

• Utilitaires d'administration mode ligne pour :

- Démarrer une instance
- Arrêter une instance
- Faire des opérations de maintenance (base hors service)
- Restaurer la base ou des fichiers de données
- Gérer l'archivage
- Visualiser les paramètres d'instance
- Et toutes les commandes SQL

• L'utilitaire Server Manager (svrmgrl) disparaît en Oracle9i.

Utilisation de SQLPLUS

```
(dba9i)/tpora/gr1>sqlplus /nolog

SQL*Plus: Release 9.0.1.0.0 - Production on Mon Mar 17 13:58:43 2003

(c) Copyright 2001 Oracle Corporation. All rights reserved.

SQL> connect / as sysdba
Connected.
SQL> show parameter db_name

NAME                                TYPE        VALUE
-----
db_name                             string      A901
SQL>
```



Connexion

```
(dba8i)8i/home>echo $ORACLE_HOME
/ora8i/app/oracle/product/8.1.7
(dba8i)8i/home>echo $ORACLE_SID
LT817
(dba8i)8i/home>svrmgrl
```

Oracle Server Manager Release 3.1.7.0.0 - Production
Copyright (c) 1997, 1999, Oracle Corporation. All Rights Reserved.

Oracle8i Enterprise Edition Release 8.1.7.0.0 - Production
With the Partitioning option
JServer Release 8.1.7.0.0 - Production

```
SVRMGR> connect internal
Connected.
SVRMGR>
```

Avant toute opération il faut se connecter
(même si la base est arrêtée)

Il faut dans son environnement :

La variable **ORACLE_HOME**

La variable **ORACLE_SID**

Le chemin d'accès **\$ORACLE_HOME/bin** dans «PATH»

Pour faire «connect internal»:

Être utilisateur **oracle** (propriétaire des fichiers) ou

Être membre du groupe propriétaire.

Mot de passe sur internal:

Il est possible de protéger l'accès par un mot de passe, qui sera enregistré dans un fichier externe à la base.

La commande orapwd sert à gérer ce fichier.

Sous WINDOWS, par défaut il y a un mot de passe qui est **oracle**.

ATTENTION: le « connect internal » disparaît en 9i au profit de la syntaxe de la page précédente.



Démarrer une instance

La commande startup met en service une base

```
SVRMGR> connect internal
SVRMGR> startup
ORACLE instance started.
Total System Global Area          4847776 bytes
Fixed Size                        47264 bytes
Variable Size                     4317184 bytes
Database Buffers                  409600 bytes
Redo Buffers                       73728 bytes
Database mounted.
Database opened.
SVRMGR>
```

La commande startup enchaîne 3 phases: création d'une instance, montage et ouverture de la base.

Options de la commande **startup**:

- **NOMOUNT** Création d'une instance avec une SGA et des processus d'arrière plan (phase 1).
- **MOUNT** Création d'une instance et montage d'une base (phases 1 et 2).
- **OPEN** Valeur par défaut: Création d'une instance, montage et ouverture de la base (phases 1, 2 et 3).
- **PFILE=** Nom du fichier de paramètre à utiliser. Par défaut \$ORACLE_HOME/dbs/initSID.ora.
- **RESTRICT** Base ouverte accessible uniquement aux dba et aux utilisateurs possédant le privilège restrict.

• ...

Remarque: en 9i, le fichier de paramètres peut être sous forme texte ou binaire (spfile). Si « **startup** » sans paramètres: Oracle recherche dans \$ORACLE_HOME/dbs en priorité un fichier de paramètres binaire de nom « spfileSID.ora », puis un fichier binaire « spfile.ora » et enfin un fichier de paramètres texte de nom « initSID.ora »

Arrêter une instance

La commande shutdown ferme et démonte la base puis détruit l'instance.

```
SVRMGR> connect internal
SVRMGR> shutdown
Database closed.
Database dismounted.
ORACLE instance shut down.
SVRMGR>
```

Attention : il suffit d'un utilisateur connecté pour que cette commande reste en attente, les nouvelles connexions seront interdites mais on attendra que tous les utilisateurs soient déconnectés pour fermer la base.

Options :

- NORMAL (par défaut)

Attend que les utilisateurs se déconnectent, interdit les nouvelles connexions, ferme et démonte la base, détruit l'instance.

- IMMEDIATE

Attend que les appels en cours se terminent, interdit les connexions, annule les transactions non validées, ferme et démonte la base, détruit l'instance.

- TRANSACTIONAL

Interdit la connexion, déconnecte les utilisateurs après la validation de la transaction en cours, ferme et démonte la base, détruit l'instance.

- ABORT (DÉCONSEILLÉ)

Détruit l'instance sans alignement des données de la base. Pour ouvrir à nouveau cette base, un réaligement sera nécessaire (**recover**).

Mise en service

Après un « startup » partiel, on utilisera la commande ALTER DATABASE pour monter et ouvrir la base.

```
SVRMGR> startup nomount
ORACLE instance started.
Total System Global Area          4847776 bytes
Fixed Size                        47264 bytes
Variable Size                     4317184 bytes
Database Buffers                  409600 bytes
Redo Buffers                       73728 bytes
SVRMGR> alter database mount;
Statement processed.
SVRMGR> alter database open;
Statement processed.
SVRMGR>
```

On fait un démarrage partiel pour effectuer des opérations d'administration particulières (passage en mode archive, changement de noms de fichiers, réparations...).

Si un incident est détecté pendant la phase de montage : fichier manquant ou non aligné, la base ne sera pas ouverte (elle reste montée. Il faudra réparer puis faire soit :

les commandes **shutdown** et **startup**

ou

alter database open;



Démarrage et arrêt automatique

Les bases de données ORACLE de la machine sont déclarées dans le fichier texte `/etc/oratab`, une ligne par base. 3 champs :

SID:ORACLE_HOME:y=yes ou n=no

LT817:/oracle8/app/oracle/product/8.1.7:Y

La commande **dbstart** démarre les bases et est mise en général dans `/etc/inittab`

La commande **dbshut** arrête les bases et est mise en général dans `/etc/rc.shutdown`



Passage du mode restreint au mode normal

Pour passer en mode normal, on utilisera la commande **ALTER SYSTEM**

```
SVRMGR> select instance_name,status,logins
from v$instance;
INSTANCE_NAME      STATUS  LOGINS
-----
AT805              OPEN    RESTRICTED
SVRMGR> alter system disable restricted session;
Statement processed.
SVRMGR> select instance_name,status,logins
from v$instance;
INSTANCE_NAME      STATUS  LOGINS
-----
AT805              OPEN    ALLOWED
```

On peut aussi arrêter l'instance et démarrer en mode normal.

Visualisation de paramètres

Utilisation de la commande show pour afficher :

- les paramètres d'initialisation
- la SGA
- l'instance

```
SVRMGR> show parameters
NAME                                TYPE      VALUE
-----
O7_DICTIONARY_ACCESSIBILITY        boolean    TRUE
always_anti_join                   string     NESTED_LOOPS
always_semi_join                   string     standard
aq_tm_processes                    integer    0
audit_file_dest                    string     ?/rdbms/audit
audit_trail                        string     NONE
.../...
```

Autres exemples d'utilisation :

```
SVRMGR> show parameter db_name
NAME                                TYPE      VALUE
-----
db_name                            string     EVT815

SVRMGR> show sga
Total System Global Area           5102724 bytes
Fixed Size                         64644 bytes
Variable Size                      4456448 bytes
Database Buffers                   409600 bytes
Redo Buffers                       172032 bytes

SVRMGR> show instance
Instance                           local
SVRMGR>
```

Les Vues du dictionnaire

- V\$SGA
tailles des différentes zones de la SGA
- V\$INSTANCE
état, date d'activation, status...
- V\$PARAMETER
nom, valeur...
- V\$DATABASE
nom, date de création, status...

*select * from v\$xxx;*

Exemples :

```
SQL> select name,value from v$parameter;
```

NAME	VALUE
processes	30
sessions	38
timed_statistics	TRUE
timed_os_statistics	0
resource_limit	FALSE
license_max_sessions	0
license_sessions_warning	0
.../...	

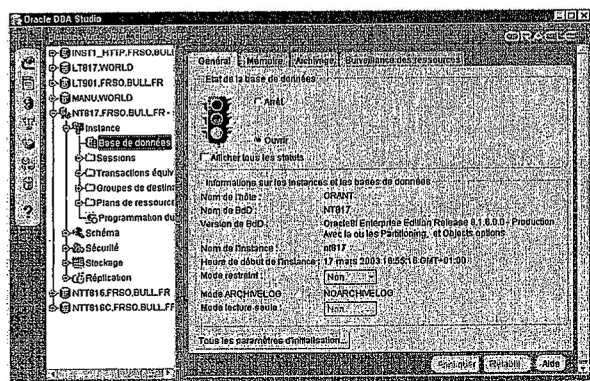
```
SQL> select instance_name,logins,shutdown_pending,startup_time
2. from v$instance;
```

INSTANCE_NAME	LOGINS	SHU	STARTUP_T
EVT815	ALLOWED	NO	04-JUL-00

```
SQL> select name,created from v$database;
```

NAME	CREATED
EVT815	18-OCT-99

ORACLE Enterprise Manager Instance manager



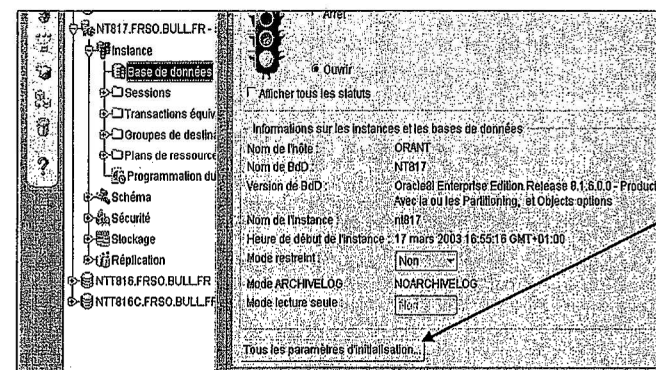
On peut arrêter et démarrer une instance à l'aide d'OEM - Instance Manager, Il faut pour cela :

- avoir un fichier de mot de passe, c'est le cas par défaut sous WINDOWS
- s'être connecté en mode SYSOPER ou SYSDBA sur le compte internal (ou son équivalent sous SQL*Plus)
- sauvegarder localement les paramètres d'instance pour pouvoir redémarrer

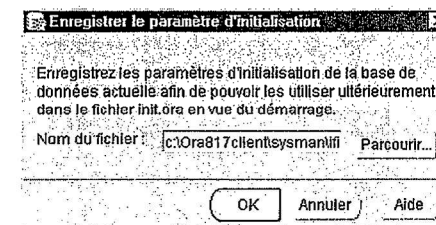
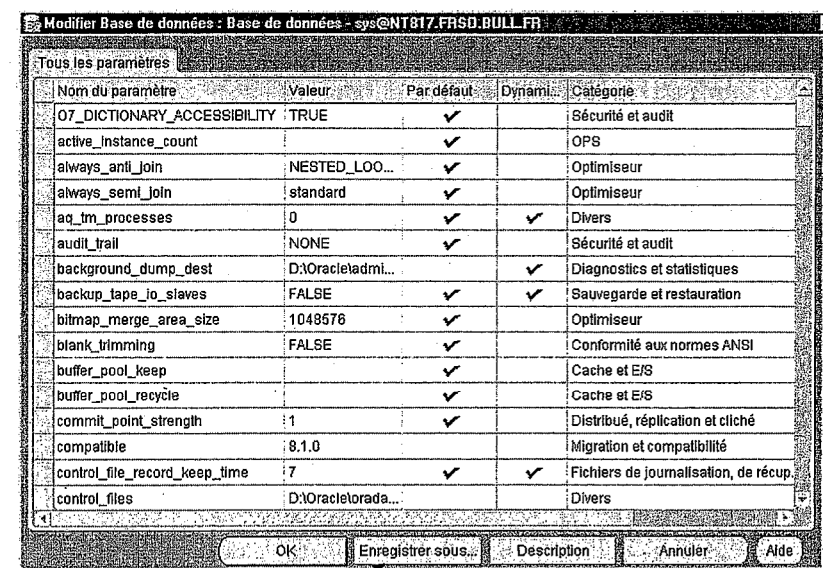
Mise en œuvre du fichier de mot de passe (sous UNIX) :

- 1 - initialiser le paramètre d'instance
remote_login_passwordfile = exclusive
- 2 - créer le fichier orapwSID dans le répertoire \$ORACLE_HOME/dbs avec la commande orapwd

\$ orapwd file=orapwORCL password=oracle
- 3 - Sauvegarde locale (c'est à dire sur le PC ou vous avez installé OEM pour démarrer à distance la base) du fichier initialisation:



Enregistrement local (PC contenant O.E.M) des paramètres d'instance de la base distante à gérer.



TRAVAUX PRATIQUES

- Contrôlez les variables d'environnement et la présence de vos processus «ORACLE». */ora3i/app/oracle/product/9.2.0*
F920
- Le fichier de paramètres :

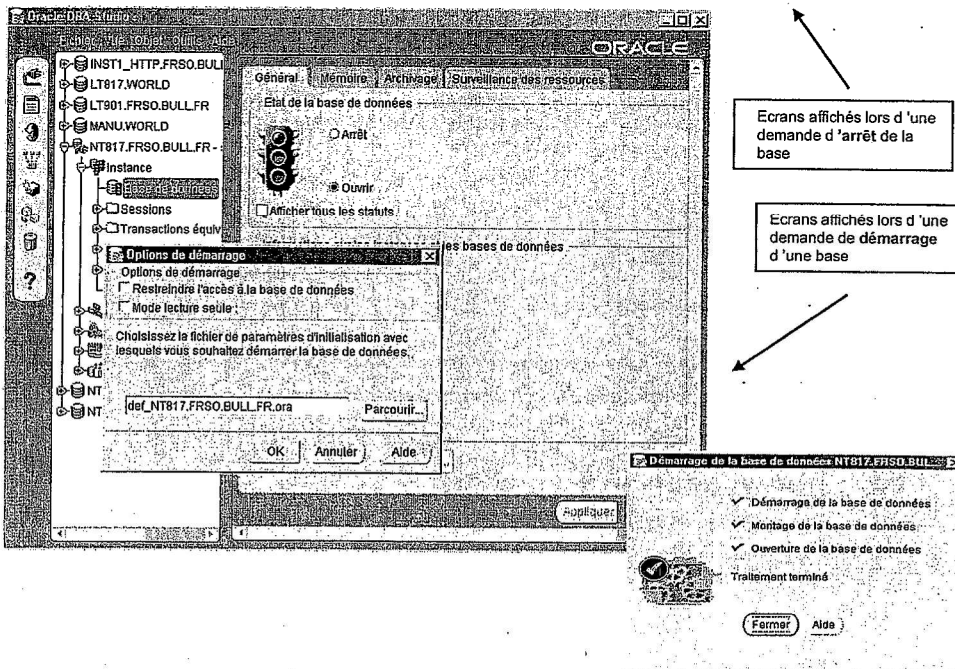
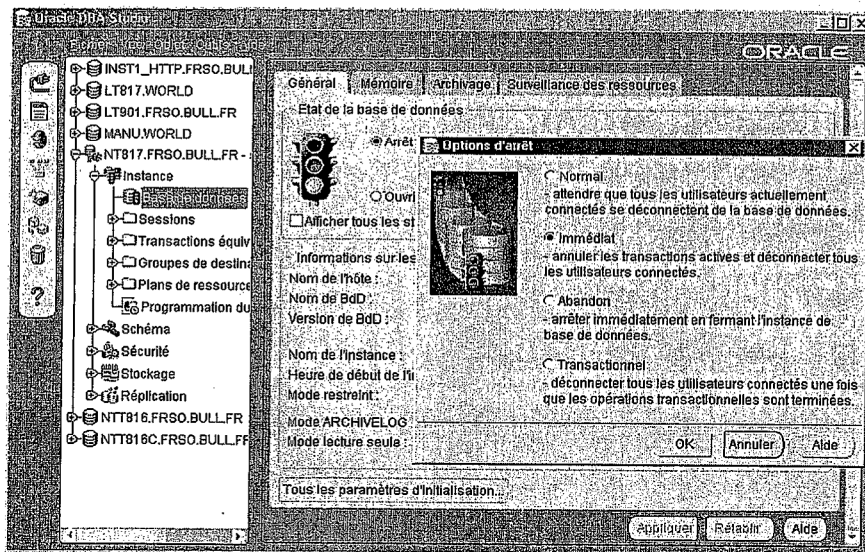
recherchez et visualisez le contenu de votre fichier de paramètres.

- Exécutez SVRMGRL et connectez vous en mode internal (ou son équivalent sous SQL*Plus)

- visualisez la SGA (show sga)
- si l'instance n'existe pas démarrez la (startup)
- visualisez les paramètres d'instance (show parameters)
- arrêtez l'instance rapidement (shutdown immediate)
- passiez les différentes phases avec des requêtes SQL

- Consultez les vues du dictionnaire.

SQL> select name, value from v\$parameter where name = 'db_name';
ou > show parameter db_name



Ecrans affichés lors d'une demande d'arrêt de la base

Ecrans affichés lors d'une demande de démarrage d'une base



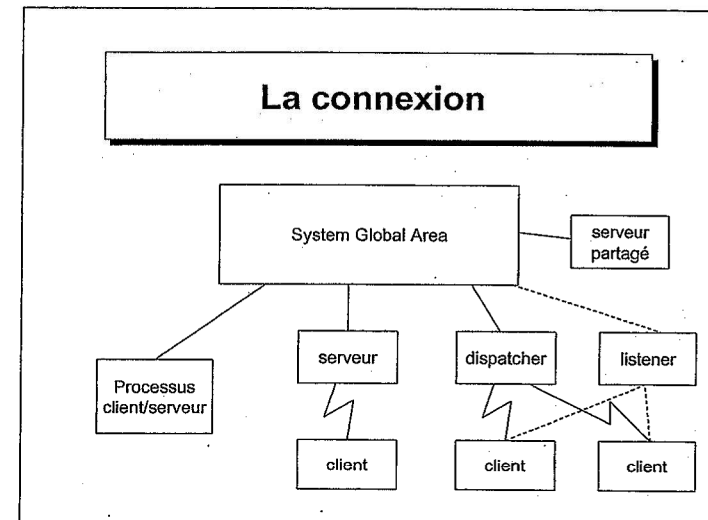
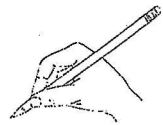
BASES DE DONNEES

SQL*NET

Objectif :

A la fin de ce module, vous serez capable de démarrer et arrêter le listener, visualiser les entrées du tnsnames, ajouter une entrée dans le fichier tnsnames à l'aide de l'outil graphique 'Net 8 Easy Config' :

- Le fichier listener.ora
- Le fichier tnsnames.ora
- La commande lsnrctl
- L'utilitaire tnsping
- L'utilitaire graphique 'Net8 Easy Config'



Différents mode de connexion sont possible :

- Le code de l'application est lié au serveur, un seul processus local est géré par le système.
- Le client et le serveur sont séparé, dans ce cas le client peut s'exécuter dans un autre système et la connexion va se faire par l'intermédiaire du **listener**. Si le client est local la connexion peut être directe ou par l'intermédiaire du listener.
- Connexion à un système multi-thread, dans ce cas la connexion se fait obligatoirement par l'intermédiaire du **listener**. Le client est indifféremment local ou distant.

La connexion à un service de manière explicite de fera ainsi :

\$ sqlplus nom/pass@service

La variable TWO_TASK sert à donner un nom de service qui sera utilisé par défaut.



Le fichier listener.ora

Plusieurs cas sont possibles :

- Un listener par machine écoutant un ou plusieurs sockets, dans ce cas on aura 1 seul fichier de configuration
- Un listener par version d'ORACLE, et un fichier de configuration par listener

Ce fichier texte se trouve dans le répertoire \$ORACLE_HOME/network/admin et contient les différents paramètres de configuration.

Il est lu au démarrage ou au rechargement.

Il peut être édité par des outils d'administration

exemple de fichier avec un seul listener s'appelant listener servant une instance LT817.

```
LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP) (HOST = lipari) (PORT = 1521))
      )
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC))
      )
    )
  )

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = PLSExtProc)
      (ORACLE_HOME = /ora81/app/oracle/product/8.1.7)
      (PROGRAM = extproc)
    )
    (SID_DESC =
      (GLOBAL_DBNAME = LT817.world)
      (ORACLE_HOME = /ora81/app/oracle/product/8.1.7)
      (SID_NAME = LT817)
    )
  )
```



L'utilitaire lsnrctl

Cet utilitaire permet en interactif ou en mode ligne de :

- démarrer un listener
- arrêter un listener
- afficher l'état d'un listener
- etc.

```
$ lsnrctl start
$ lsnrctl start listen1
$ lsnrctl stop
```

L'aide intégrée :

```
$ lsnrctl

LSNRCTL for IBM/AIX RISC System/6000: Version 8.1.7.0.0 - Production on 22-APR-2003 15:42:21

(c) Copyright 1998 Oracle Corporation. All rights reserved.

Welcome to LSNRCTL, type "help" for information.

LSNRCTL> help
The following operations are available
An asterisk (*) denotes a modifier or extended command:

start          stop          status
services       version       reload
save_config    trace         spawn
dbnmp_start    dbnmp_stop    dbnmp_status
change_password quit          exit
set*           show*

LSNRCTL> help.start
start [<listener_name>] : start listener

LSNRCTL>
```




Le fichier tnsnames.ora

C'est le fichier "client" qui fait la relation entre un nom logique de service et les paramètres de la connexion.

Chaque machine cliente aura un fichier de connexion.

```
LT817.WORLD =  
(DESCRIPTION =  
  (ADDRESS_LIST =  
    (ADDRESS = (PROTOCOL = TCP) (HOST = 129.181.176.111) (PORT = 1521))  
  )  
  (CONNECT_DATA =  
    (SERVICE_NAME = LT817.world)  
  )  
)
```

nom de l'instance

Ce fichier peut être généré par différents outils d'administration réseau.

Pour simplifier la gestion, un mécanisme de centralisation peut être mis en place (produit oracle names).



tnsping

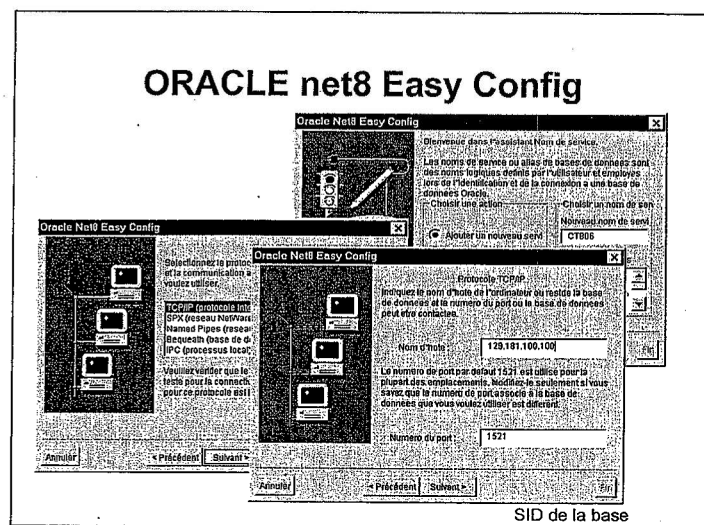
Cette utilitaire sert à tester la qualité du réseau et la disponibilité d'un nom réseau.

```
$tnsping LT817.world  
  
TNS Ping Utility for IBM/AIX RISC System/6000: Version 8.1.7.0.0 - Production on  
22-APR-2003 15:48:44  
  
(c) Copyright 1997 Oracle Corporation. All rights reserved.  
  
Attempting to contact (ADDRESS=(PROTOCOL=TCP) (HOST=129.181.176.111) (PORT=1521))  
OK (20 msec)
```

La réponse OK signifie que le listener distant est actif et qu'il sert bien cette instance.

Attention !

Si l'instance est hors service et le listener présent la réponse sera aussi OK.



Après avoir donné tous les renseignements de la connexion, l'outil propose de tester le service. Il faut avoir un nom d'utilisateur avec le mot de passe car l'outil se connecte à la base.

Il faut continuer jusqu'à la fin car la nouvelle entrée n'est enregistrée que lors du clic sur le bouton 'fin'.

TRAVAUX PRATIQUES

Visualisez le contenu des fichiers :

listener.ora
tnsnames.ora

Visualisez l'état du listener avec la commande **lsnrctl stat**.

Connectez vous en utilisant les services du listener :

\$ sqlplus nom/mot_de_passe@service



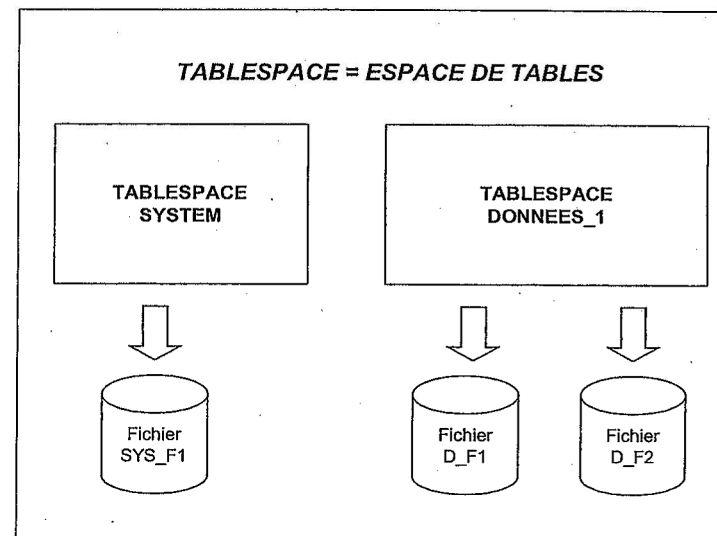
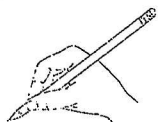
BASES DE DONNEES

Tablespaces et Segments

Objectif :

A la fin de ce module, vous serez capable de décrire les structures physiques de la base (tablespaces, segments).

- Créer, modifier supprimer des tablespaces
- Gérer, visualiser les différents segments
- Gérer les rollback segments
- visualiser l'espace libre de la base



Les tablespaces sont des espaces de données logiques contenant les objets (table, index ...) de la base. Ils sont constitués de 1 ou plusieurs fichiers de données physiques. Un objet stocké dans un tablespace, peut être éclaté sur un ou plusieurs fichiers de données composant ce tablespace.

Les tablespaces sont constitués de fichiers du système d'exploitation ou de morceaux de disque bruts.

A fin de la création de la base, seul le tablespace SYSTEM existe et contient le dictionnaire (objets systèmes).

Très rapidement, d'autres tablespaces sont créés pour « ranger » correctement les différentes familles d'objets d'une base:

- un tablespace dédiés aux espaces temporaires (TEMP),
- un tablespace dédiés aux rollback segments (RBS ou UNDOTBS en 9i),
- des tablespaces pour les objets applicatifs.

La « spécialisation » des tablespaces facilite l'administration et l'optimisation d'une base de données Oracle.

Création d'un tablespace

Extrait de la syntaxe du CREATE TABLESPACE

```
CREATE TABLESPACE nom
  DATAFILE 'file_name' SIZE integer K|M
  [ AUTOEXTEND { OFF
                | ON [ NEXT integer [ K | M ]
                [, filespec ...]
  [ DEFAULT STORAGE storage_clause ]
  [ ONLINE | OFFLINE ]
  [ PERMANENT | TEMPORARY ]    .. . ;
```

Il faut donner au minimum les caractéristiques du premier fichier associé au tablespace.

Par défaut le tablespace est online et permanent.

Les objets d'un tablespace offline sont inaccessibles.

L'état offline est maintenu lorsque que la base est arrêtée et redémarrée.

Exemples :

```
CREATE TABLESPACE data1
  DATAFILE '/oracle/base/data1f1.dbf' size 10M;
```

```
CREATE TABLESPACE data2
  DATAFILE '/oracle/base/data2f1.dbf' size 10M
  AUTOEXTEND ON NEXT 5M MAXSIZE 50M;
```

```
CREATE TEMPORARY TABLESPACE TEMP
  TEMPFILE '/oracle/base/tempf1.dbf' size 15M;
```

Modification d'un tablespace

Extrait de la syntaxe de l'ALTER TABLESPACE

```
ALTER TABLESPACE nom
  ADD DATAFILE 'file_name' SIZE integer K|M
  [ AUTOEXTEND { OFF
                | ON [ NEXT integer [ K | M ]
                [, filespec ...]
  [ RENAME FILE 'old_name' TO 'new_name']
  [ DEFAULT STORAGE storage_clause ]
  [ COALESCE]
  [ BEGIN | END BACKUP ]
  [ ONLINE | OFFLINE ]
  [ PERMANENT | TEMPORARY ]    .. . ;
```

Pour renommer un fichier de données d'un tablespace il faut :

- mettre le tablespace offline
- renommer le fichier avec une commande de l'OS (mv ou rename)
- prévenir Oracle avec la commande alter tablespace ... rename file ...
- repasser le tablespace online.

Exemples :

```
ALTER TABLESPACE data1
  ADD DATAFILE '/oracle/base/data1f2.dbf' size 10M;
```

```
ALTER TABLESPACE data2
  OFFLINE;
```

```
ALTER TABLESPACE data3
  RENAME FILE '/oracle/base/data3.dbf'
  TO '/oracle/base/data3f1.dbf';
```

DATAFILE



La suppression de tablespace

Tous les objets stockés dans le tablespace doivent être détruits au préalable sinon il faudra ajouter "INCLUDING CONTENTS".

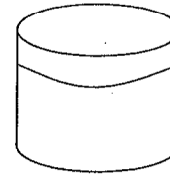
Avec "CASCADE CONSTRAINTS" les contraintes référentielles des tables dépendantes d'autres tablespaces seront supprimées.

```
DROP TABLESPACE nom_tablespace  
[ INCLUDING CONTENTS [ CASCADE CONSTRAINTS ] ]
```

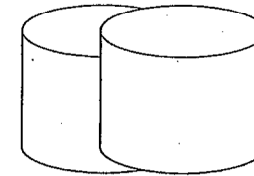
On ne peut pas détruire un tablespace avec des transactions actives ou qui contient des "rollback segments" online ou dans l'état "needs recovery".



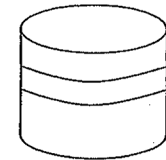
AGRANDISSEMENT



Agrandissement manuel



Ajout de fichier



Agrandissement auto

3 possibilités pour agrandir la taille d'un tablespace:

- modification de la taille des fichiers le constituant,
- ajout d'un nouveau fichier au tablespace,
- agrandissement automatique des fichiers le constituant.

Modification de la taille d'un fichier :

```
ALTER DATABASE DATAFILE 'nom_de_fichier' RESIZE entier [ K|M ]
```

Cette modification peut être faite en agrandissement ou en réduction (si l'espace libéré en fin de fichier n'est pas occupé).

Ajout d'un fichier au tablespace :

```
ALTER TABLESPACE TBS1  
ADD DATAFILE 'nom_de_fichier_n' SIZE entier [ K|M ]
```

Mise en auto-extension d'un fichier de la base :

```
ALTER DATABASE DATAFILE 'nom_du_fichier'  
AUTOEXTEND ON NEXT entier [ K|M ] MAXSIZE entier [ K|M ]
```

La mise en auto-extension peut être faite à la création du fichier ; lors du "CREATE" ou "ALTER" du tablespace.

La clause default storage

Chaque tablespace possède une clause "default storage" qui sera appliquée aux objets créés dans celui-ci si la clause "storage" n'a pas été spécifiée à la création de l'objet.

Cette clause peut être modifiée à la création ou à la modification du tablespace.

```
DEFAULT STORAGE ( INITIAL entier [ M|K ]
                  NEXT entier [ M|K ]
                  MINEXTENTS entier
                  MAXEXTENTS entier
                  PCTINCREASE entier
                  ...
                  )
```

A partir d'Oracle8i, on peut créer des tablespaces gérés localement avec possibilité de donner une taille fixe d'extent.

Exemple de syntaxe :

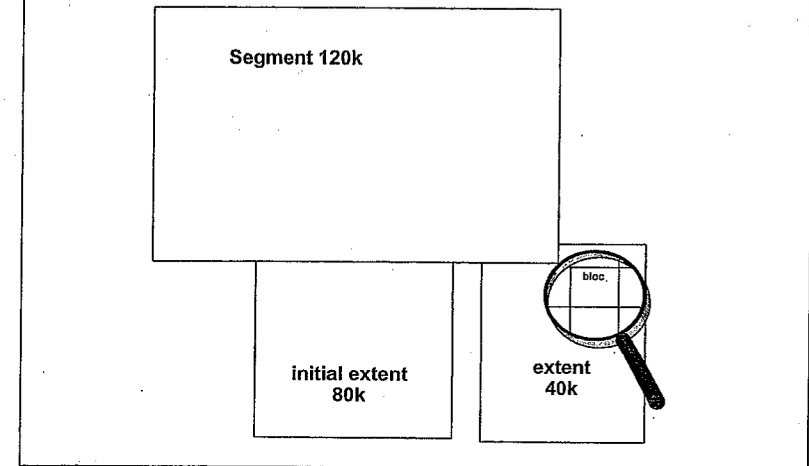
```
CREATE TABLESPACE tbspl
  DATAFILE 'nom_de_fichier' SIZE taille
  EXTENT MANAGEMENT LOCAL
  UNIFORM SIZE 1M;
```

uniform size # AUTOALLOCATE

Cette clause sera prioritaire par rapport aux autres clauses (default storage et storage).

La gestion de l'espace d'un tablespace géré localement est faite à travers une bitmap (et non plus à travers le tablespace SYSTEM). Cela diminue les accès au dictionnaire (tablespace SYSTEM) et améliore la gestion de l'espace. Cette possibilité, introduite en 8i, s'étend en 9i (un tablespace est géré localement par défaut).

LES SEGMENTS



Les segments sont attribués aux tables, index, rollback...

Ils sont constitués d'un ou plusieurs extents. Un extent est un ensemble de blocs contigus, la taille des blocs est fixe pour toute la base en Oracle7 et 8i. A partir d'Oracle9i, on peut avoir plusieurs tailles de blocs pour une base.

Les extents sont attribués automatiquement par Oracle lorsque le segment est plein.

Les attributs principaux d'un segment sont:

• le nom	segment_name
• le tablespace	tablespace_name
• le type	segment_type
• taille de l'initial	initial_extent
• taille de l'extent suivant	next_extent
• nombre minimum d'extents	min_extents
• nombre maximum d'extents	max_extents
• pourcentage d'incrément	pct_increase
• la taille du segment	bytes

Exemples de requêtes :

Affichage du nombre et type de segments par tablespace

```
SQL> SELECT TABLESPACE_NAME, SEGMENT_TYPE, COUNT(*)
2 FROM DBA_SEGMENTS GROUP BY TABLESPACE_NAME, SEGMENT_TYPE;
```

TABLESPACE_NAME	SEGMENT_TYPE	COUNT(*)
INDX	INDEX	1
RBS	ROLLBACK	4
SYSTEM	CACHE	1
SYSTEM	CLUSTER	8
SYSTEM	INDEX	153
SYSTEM	ROLLBACK	2
SYSTEM	TABLE	165
USERS	INDEX	57
USERS	TABLE	136

9 rows selected.

Affichage des segments ayant le plus d'extents

```
SQL> select tablespace_name, substr(owner,0,13) owner,
2 substr(segment_name,0,15) segment_name, segment_type, extents
3 from dba_segments
4 where owner not in ('SYSTEM', 'SYS') and extents > 1
5 order by extents desc
```

TABLESPACE_NAME	OWNER	SEGMENT_NAME	SEGMENT_TYPE	EXTENTS
USERS	COURS1	BEMP	TABLE	6
SYSTEM	OUTLN	OL\$	TABLE	2
SYSTEM	OUTLN	OL\$HINTS	TABLE	2
...				

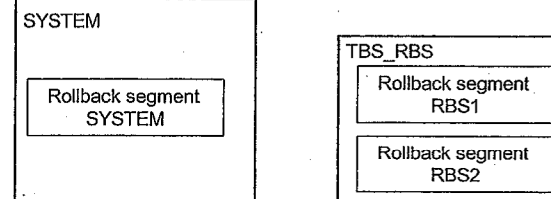
UNDO séparé depuis la version 9i

Les rollback segments

Les rollback segments enregistrent les images avant des transactions SQL en cours.

Après la création de la base, il existe un seul rollback segment nommé SYSTEM, stocké dans le tablespace SYSTEM et n'est utilisable que pour les objets du tablespace SYSTEM.

Très rapidement, on crée un tablespace dédiés aux rollback segments et des rollback segments dans ce tablespace.



Création d'un rollback segment:

```
CREATE [PUBLIC] ROLLBACK SEGMENT nom_rollback_segment
[ TABLESPACE nom_tablespace ]
[ STORAGE ( paramètres ) ]
```

Les rollback segments sont offline après la création et seront activés (online) avec la commande **alter rollback segment** (pour l'instance en cours) ou à l'ouverture de la base s'il ont été déclarés dans le fichier de paramètres initSID.ora. Exemple:

```
rollback_segments = ( rbs1 , rbs2 )
```

```
ALTER ROLLBACK SEGMENT nom_rollback_segment
[ TABLESPACE nom_tablespace ]
[ STORAGE ( paramètres ) ]
[ ONLINE | OFFLINE ]
[ SHRINK [ TO integer [ K | M ] ] ];
```

Les rollback segments s'agrandissent automatiquement en cas de besoin (grosses transactions) et restent à cette taille, "shrink" est l'action de réduction soit à la valeur optimale déclarée ou à une taille spécifiée.



Lors du passage d'un rollback segment de l'état ONLINE à OFFLINE si le rollback segment est en cours d'utilisation il va passer par un état "PENDING" (en cours de passage) avant que toutes les transactions soient closes.

Destruction d'un rollback segment:

```
DROP ROLLBACK SEGMENT nom_du_rollback_segment ;
```

Pour détruire un rollback segment, il faut qu'il soit offline.

Exemples: Création / Mise en service / Destruction

```
SQL> CREATE ROLLBACK SEGMENT rbs1
2 TABLESPACE tbs_rbs
3 STORAGE ( INITIAL 200 K NEXT 200 K OPTIMAL 1 M );
Rollback segment created.
SQL> ALTER ROLLBACK SEGMENT rbs1 ONLINE;
Rollback segment altered.
SQL> DROP ROLLBACK SEGMENT rbs1;
Rollback segment dropped.
```

Remarques:

Avec Oracle9i, vous pouvez gérer **automatiquement** ou **manuellement** les segments d'annulation. La gestion automatique **simplifie** la gestion des segments d'annulation. Le choix entre la gestion automatique ou manuelle se fera généralement à la création de la base avec le paramètre d'initialisation **UNDO_MANAGEMENT (AUTO)**: Oracle gère automatiquement les segments d'annulation (segments de type UNDO), **MANUAL** (valeur par défaut): gestion manuelle des segments d'annulation (segments de type rollback) comme en Oracle 7 ou 8i).

Avec la gestion automatique, toutes les images « avant » sont centralisées dans un **tablespace** spécifique de type **UNDO**. Au lieu de gérer un certain nombre de segments d'annulation de différentes tailles, il suffit d'allouer un tablespace de type UNDO pouvant contenir toutes les images « avant » de l'instance et Oracle gère automatiquement les segments d'annulation de type UNDO à l'intérieur de ce tablespace.

Pour démarrer en mode de gestion automatique des segments d'annulation, il faut positionner les paramètres init.ora:

UNDO_MANAGEMENT = AUTO

UNDO_TABLESPACE = nom_du_tbsp_de_type_undo

SQL> Show parameters undo



Rollback Segments : Les vues du dictionnaire

V\$ROLLNAME	Liste des rollback segments actifs
V\$ROLLSTAT	Etat des rollback segments actifs : numéro, taille, nombre de transactions actives...
DBA_ROLLBACK_SEGS	Tous les rollback segments : nom, tablespace, propriétaire, paramètres de stockage.

Exemples de requêtes :

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

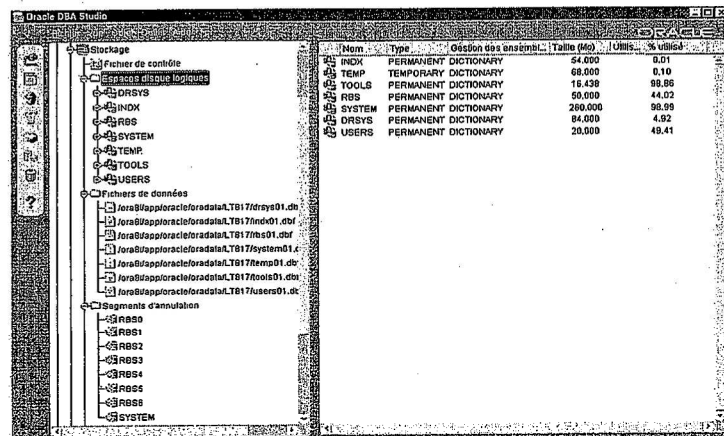
```
USN NAME
```

```
-----
0 SYSTEM
4 R01
5 R02
```

```
SQL> SELECT segment_name, tablespace_name, segment_id, status
2 FROM dba_rollback_segs;
```

SEGMENT_NAME	TABLESPACE_NAME	SEGMENT_ID	STATUS
SYSTEM	SYSTEM	0	ONLINE
RSB	SYSTEM	3	OFFLINE
R01	RBS	4	ONLINE
R02	RBS	5	ONLINE

Oracle Enterprise Manager Storage Manager



A l'aide d'Oracle Enterprise Manager (module Storage Manager ou son équivalent sous DBA Studio) vous pouvez effectuer les principales opérations d'administration sur les tablespaces, fichiers de données, et rollback segments de la base.

Les informations mises en forme par O.E.M sont extraites des vues du dictionnaire. Mais O.E.M n'indique pas les vues qu'il utilise. Vous avez la possibilité d'obtenir les mêmes informations en allant interroger les vues du dictionnaire avec un outil en mode ligne.

Principales vues du dictionnaire:

USER_TABLESPACES	DBA_TABLESPACES
USER_SEGMENTS	DBA_SEGMENTS
USER_EXTENTS	DBA_EXTENTS
USER_FREE_SPACE	DBA_FREE_SPACE
	DBA_DATA_FILES
V\$DBFILE	V\$DATAFILE
V\$CONTROLFILE	V\$LOGFILE

Exemple de requêtes:

Caractéristiques des tablespaces.

```
SQL> select tablespace_name,initial_extent,next_extent,
2 pct_increase,status,contents,extent_management
3 from dba_tablespaces;
```

TABSPACE_N	INITIAL_EXTENT	NEXT_EXTENT	PCT_INCREASE	STATUS	CONTENTS
SYSTEM	65536	65536	50	ONLINE	PERMANENT
DICTIONARY					
TOOLS	32768	32768	0	ONLINE	PERMANENT
DICTIONARY					
RBS	524288	524288	50	ONLINE	PERMANENT
DICTIONARY					
TEMP	65536	65536	0	ONLINE	TEMPORARY
DICTIONARY					
USERS	131072	131072	0	ONLINE	PERMANENT
DICTIONARY					
INDX	131072	131072	0	ONLINE	PERMANENT
DICTIONARY					

Informations sur les fichiers de données.

```
SQL> select file_name,file_id,tablespace_name,blocks,
2 status,autoextensible
3 from dba_data_files;
```

FILE_NAME	FILE_ID	TABSPACE	BLOCKS	STATUS	AUT
/ora81/app/oracle/oradata/	1	SYSTEM	33280	AVAILABLE	YES
LT817/system01.dbf					
/ora81/app/oracle/oradata/	2	TOOLS	2104	AVAILABLE	YES
LT817/tools01.dbf					
/ora81/app/oracle/oradata/	3	RBS	6400	AVAILABLE	YES
LT817/rbs01.dbf					
/ora81/app/oracle/oradata/	4	TEMP	8704	AVAILABLE	YES
LT817/temp01.dbf					
/ora81/app/oracle/oradata/	5	USERS	2560	AVAILABLE	YES
LT817/users01.dbf					
/ora81/app/oracle/oradata/	6	INDX	6912	AVAILABLE	YES
LT817/indx01.dbf					
/ora81/app/oracle/oradata/	7	DRSYS	10752	AVAILABLE	YES
LT817/drsys01.dbf					

Affichage de la liste des blocs libres

Nb blocs libres

SQL> select * from dba_free_space;

TABLESPACE_NAME	FILE_ID	BLOCK_ID	BYTES	BLOCKS
SYSTEM	1	4495	2564096	626
SYSTEM	1	3670	28672	7
SYSTEM	1	4359	36864	9
SYSTEM	1	3050	430080	105
SYSTEM	1	3181	77824	19
SYSTEM	1	2912	61440	15
SYSTEM	1	3027	61440	15
USERS	2	140	24576	6
...				
USERS	2	2	491520	120
USERS	2	203	24576	6
INDX	3	7	10461184	2554
RBS	4	232	102400	25
RBS	4	317	802816	196
RBS	4	47	122880	30
RBS	4	92	307200	75
TMP	5	2	5238784	1279

59 rows selected.

Si la liste est longue pour un tablespace donné on peut essayer de la réduire en exécutant la requête « ALTER TABLESPACE users COALESCE; ».

La requête suivante donne un aperçu des informations utiles à propos de l'espace libre dans la base. Attention il peut manquer des tablespaces.

```
SQL> SELECT tablespace_name, file_id, COUNT(*) "Nb Part",
1 MAX(bytes/1000) "MAXI (ko)", SUM(bytes/1000) "TOTAL (Ko)"
2 FROM dba_free_space
3 GROUP BY tablespace_name, file_id
4 ORDER BY 1
```

TABLESPACE_NAME	FILE_ID	Nb Part	MAXI (ko)	TOTAL (Ko)
INDX	7	1	2,093.1	2,093.1
RBS	4	1	1,683.5	1,683.5
SYSTEM	1	3	4,210.7	7,774.2
TMP	6	1	1,044.5	1,044.5
TMP	5	1	1,044.5	1,044.5
USERS	2	2	921.6	962.6
USERS	3	1	950.3	950.3

TRAVAUX PRATIQUES

TABLESPACE

- Listez les fichiers de la base avec les vues v\$dbfile, dba_data_files, v\$controlfile, v\$logfile. *v\$tempfile*
- Listez les tablespaces avec la vue dba_tablespaces.
- Créez un tablespace "essai" de 2 Mo sans renseigner la clause "default storage". *EXTEND MANAGEMENT DICTIONARY (9i)*
- Modifiez la taille du fichier à 3 Mo et contrôlez le résultat dans l'OS et dans la vue dba_data_files.
- Modifiez la clause "default storage" mettez la valeur 40 k à "initial extent".

ROLLBACK SEGMENT

- Créez un rollback segment dans votre tablespace.
- Affichez les informations à partir des vues v\$rollname, dba_rollback_segs.
- Passez votre rollback segment online et contrôlez le status à partir des vues du point précédent.

↳ dba_rollback_segs

initSID.ora

undo_management
undo_tablespace



BASES DE DONNEES

Sécurité et accès à la base

Objectif :

A la fin de ce module, vous serez capable de gérer la sécurité de vos applicatifs et données.

- Gestion des utilisateurs
- Gestion des rôles
- Gestion des profils
- Gestion des droits et privilèges
- Utilisation des vues et synonymes du dictionnaire
- Utilisation d ' O.E.M module "Oracle Security Manager"



La création / modification d'utilisateur

Après la création de la base au moins deux utilisateurs sont créés :

- SYS (propriétaire du dictionnaire)
mot de passe *change_on_install*
- SYSTEM (propriétaire des outils de développement)
mot de passe *manager*

Ils sont DBA "database administrator", ils ont donc le droit de créer d'autres utilisateurs en utilisant la requête CREATE USER

```
CREATE USER  nom_util.  IDENTIFIED BY mot_de_passe  
ALTER USER  EXTERNALLY
```



```
[ DEFAULT TABLESPACE nom_tablespace ]  
[ TEMPORARY TABLESPACE nom_tablespace ]  
[ QUOTA entier [ k|m ] ON nom_tablespace ]  
[ PROFILE nom_profile ] ... .. ;
```

Oracle a une gestion de la sécurité qui lui est propre avec des comptes utilisateurs et une gestion des privilèges (droits) systèmes et objets.

Exemples :

```
CREATE USER alice IDENTIFIED BY nyce  
DEFAULT TABLESPACE users  
TEMPORARY TABLESPACE temp  
QUOTA UNLIMITED ON users  
QUOTA 200 K ON system;
```



Modification d'un utilisateur

Changement du mot de passe :

```
ALTER USER sys IDENTIFIED BY secret;
```

Changement du profil de restrictions.

```
ALTER USER scott PROFILE achats;
```

Destruction d'un utilisateur

```
DROP USER user [CASCADE]
```

Si l'utilisateur est propriétaire d'objets dans la base, il faut ajouter le mot clé CASCADE pour pouvoir le détruire.

Options à partir d'Oracle8i

Lors de la création ou la modification d'un utilisateur on peut : faire expirer le mot de passe, verrouiller le compte.

Exemples :

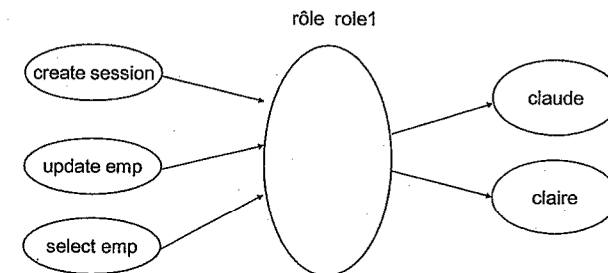
```
CREATE USER marcel IDENTIFIED BY pass  
PASSWORD EXPIRE;
```

```
CREATE USER stagiaire IDENTIFIED BY pass  
ACCOUNT LOCK;
```



Les rôles

Ils facilitent la gestion des privilèges (droits) en constituant des ensembles de privilèges qui pourront être attribués aux différents utilisateurs.



Il existe un ensemble de rôles prédéfinis qui regroupent différents privilèges systèmes.

```
SQL> select * from dba_roles;
```

ROLE	PASSWORD
CONNECT	NO
RESOURCE	NO
DBA	NO
...	
EXP_FULL_DATABASE	NO
IMP_FULL_DATABASE	NO
...	

Les trois premiers rôles sont les plus utilisés.

Les vues du dictionnaire concernant les privilèges obtenus à partir de rôles :

ROLE_SYS_PRIVS	Privilèges systèmes attribués au niveau rôle
ROLE_TAB_PRIVS	Privilèges objets attribués au niveau rôle
ROLE_ROLE_PRIVS	Rôles attribués à un rôle (A oublier)



Gestion des rôles

Création d'un rôle :

> CREATE ROLE nom ...

Modification d'un rôle (mot de passe) :

> ALTER ROLE nom ...

Suppression d'un rôle :

> DROP ROLE nom ...

Exemples :

```
SQL> CREATE ROLE role1;
```

```
SQL> CREATE ROLE role2 IDENTIFIED BY role2;
```

```
SQL> ALTER ROLE ROLE2 IDENTIFIED BY passecret;
```

```
SQL> DROP ROLE role2 CASCADE;
```



Les profils utilisateur

Les profils permettent de limiter, au niveau utilisateur, la consommation de certaines ressources systèmes.

Pour que les profils soient effectifs, il faut que le paramètre d'instance *resource_limit* soit mis à TRUE dans le fichier *initSID.ora*. Ce paramètre peut être modifié dynamiquement avec :

```
ALTER SYSTEM SET resource_limit=TRUE;
```



Syntaxe

```
CREATE PROFILE profile LIMIT
{ { SESSION_PER_USER                # 1 ...
  CPU_PER_SESSION                   # en 1/100 s
  CPU_PER_CALL                      # en 1/100 s
  CONNECT_TIME                      # en minutes
  IDLE_TIME                         # en minutes
  LOGICAL_READS_PER_SESSION         # en blocs
  LOGICAL_READS_PER_CALL           # en blocs
  COMPOSITE LIMIT }
{ integer | UNLIMITED | DEFAULT }
{ PRIVATE_SGA { integer [K | M] | UNLIMITED | DEFAULT } }
FAILED_LOGIN_ATTEMPTS
PASSWORD_LIFETIME
{ PASSWORD_REUSE_TIME | PASSWORD_REUSE_MAX }
ACCOUNT_LOCK_TIME
PASSWORD_GRACE_TIME
{ integer | UNLIMITED | DEFAULT }
PASSWORD_VERIFY_FUNCTION
{ function | NULL | DEFAULT } }...
```

Exemple :

```
CREATE PROFILE prof1 LIMIT
SESSION_PER_USER 1
PASSWORD_LIFE_TIME 60
PASSWORD_GRACE_TIME 30;
```

Durée de vie en jours.
Durée de grâce en jours.



Modification d'un profil

```
ALTER PROFILE profile LIMIT ...
```

Tous les paramètres peuvent être modifiés avec la requête "alter profile".

Suppression d'un profil

```
DROP PROFILE nom_profil [CASCADE];
```

Les utilisateurs qui avaient ce profil reprendront le profil *default*.

Lors de la création d'un utilisateur si le profil n'a pas été spécifié, l'utilisateur a le profil *default*, donc supprimer les limites affectées à un utilisateur, c'est lui redonner le profil *default*.

Attribution d'un profil à un utilisateur

```
ALTER USER JEAN PROFILE CLERK;
```

Remarque :

Les limites sur les mots de passe ne sont possibles qu'à partir de la version 8.0

Les vues

USER_RESOURCE_LIMITS	pour les limites système
USER_PASSWORD_LIMITS	pour les limites relatives aux mots de passe



Les privilèges système

Ce sont des autorisations d'effectuer des actions (requêtes SQL) dans la base qui sont attribués par les administrateurs (DBA).

La vue SYSTEM_PRIVILEGE_MAP donne la liste des privilèges système.

Un utilisateur peut interroger les vues USER_SYS_PRIVS et ROLE_SYS_PRIVS pour connaître ses privilèges.

L'administrateur lui consultera la vue DBA_SYS_PRIVS

Exemples :

```
SQL> select * from user_sys_privs;
```

USERNAME	PRIVILEGE	ADM
ORAL	UNLIMITED TABLESPACE	NO

```
SQL> select * from dba_sys_privs;
```

GRANTEE	PRIVILEGE	ADM
CONNECT	ALTER SESSION	NO
CONNECT	CREATE CLUSTER	NO
CONNECT	CREATE DATABASE LINK	NO
...		
SYS	SELECT ANY TABLE	YES
SYS	UPDATE ANY TABLE	NO
SYSTEM	UNLIMITED TABLESPACE	YES
UTEST	UNLIMITED TABLESPACE	NO



Les privilèges objet

Ce sont des autorisations d'effectuer des actions sur des tables et procédures.

Ils sont attribués par les propriétaires des objets ou un utilisateur à qui on a donné le droit de transmettre un privilège reçu.

Un utilisateur peut connaître ses droits en interrogeant les vues USER_TAB_PRIVS et ROLE_TAB_PRIVS.

Les privilèges sont :

ALTER	EXECUTE	INDEX	<u>INSERT</u>
READ	REFERENCES	<u>SELECT</u>	<u>UPDATE</u>
<u>DELETE</u>			

Remarque :

Il faut d'abord faire attention aux privilèges système tel que SELECT ANY TABLE, UPDATE ANY TABLE qui peuvent aller à l'encontre des désirs du propriétaire de l'objet.



Attribution de privilèges

La commande GRANT permet de donner des privilèges systèmes :

```
GRANT
{ system_priv | role }
[, { system_priv | role } ] ...
TO
{ user | role | PUBLIC }
[, { user | role | PUBLIC } ] ...
[ WITH ADMIN OPTION]
```

et des privilèges objets :

```
GRANT
{ object_priv | ALL [PRIVILEGES] }
[ ( column [, column] ... ) ]
[, { object_priv | ALL [PRIVILEGES] }
  [ ( column [, column] ... ) ] ] ...
ON [ schema. | DIRECTORY ] object
TO { user | role | PUBLIC } ...
[ WITH GRANT OPTION]
```

Les privilèges sont attribués à des utilisateurs ou à des rôles de la même façon. Donner un privilège à PUBLIC c'est le donner à tous les utilisateurs de la base.

« WITH ADMIN OPTION » autorise celui qui a reçu le privilège de le transmettre à son tour à d'autres utilisateurs.

Exemples :

```
SQL> GRANT SELECT,UPDATE,INSERT ON resultats TO role1;
```

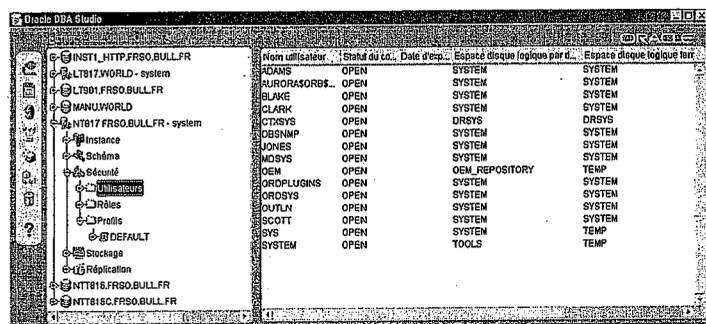
```
SQL> GRANT ALL ON dept TO util1;
```

```
SQL> GRANT CREATE TABLE,CREATE VIEW TO annie;
```

La commande REVOKE permet de retirer des privilèges systèmes et objets aux rôles et aux utilisateurs.



Oracle security manager



Ce module permet la gestion des utilisateurs, des rôles et des profils.

Vues utilisables par un dba

DBA_ROLES
DBA_USERS
DBA_SYS_PRIVS
DBA_TAB_PRIVS
DBA_PROFILES



TRAVAUX PRATIQUES

- 1 - Créez un utilisateur en spécifiant comme tablespace par défaut, le tablespace **USERS** et un quota de 300k sur ce tablespace.
- 2 - Créez un rôle avec les privilèges suivants (*create role et grant*):

CREATE SESSION, CREATE VIEW, CREATE SYNONYM
- 3 - Attribuer ce rôle à l'utilisateur que vous venez de créer (*grant*).
grant rôle to utilisateur
- 4 - Donner les droits de SELECT, UPDATE et d'INSERT sur votre table EMP à l'utilisateur que vous venez de créer (*grant*).
- 5 - Prenez l'identité de l'utilisateur créé en 1. Avec les tables du dictionnaire (user_role_privs, user_sys_privs, user_tab_privs, role_sys_privs et role_tab_privs) vérifiez s'il peut :
 - Créer une table ? *N*
 - Mettre à jour votre table EMP ? *O*
 - Créer un synonyme ? *O*
- 6 - Modifier son rôle pour qu'il puisse créer une table (*grant*).
- 7 - Modifier la définition de l'utilisateur pour que les tables temporaires soient créées dans le tablespace TEMP (*alter user*).
- 8 - Contrôler et vérifier en consultant les tables et vues du dictionnaire.



BASES DE DONNEES

Tables et Index

Objectif :

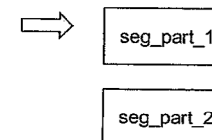
A la fin de ce module, vous serez capable de gérer et surveiller les tables et les index.

- Surveillance du segment
- Modification du stockage d'une table
- Surveillance des contraintes
- Validation/invalidation des contraintes
- Surveillance des index
- Administration de base des index



LES TABLES

- **ORACLE V7**
 - table mono-segment
 - tables en cluster
- **Nouveautés à partir ORACLE V8**
 - OIT table organisée en index
 - tables partitionnées (multi-segments)



En général les tables sont individuelles et mono segment : dans un seul tablespace.

Pour optimiser les accès en cas de jointures fréquentes on peut charger plusieurs tables dans un même segment de type cluster avec une clé de segment qui correspond au critère de jointure.

A Oracle V8, on peut créer des tables organisées en index (IOT), gérées triées en fonction de la clé primaire (obligatoirement déclarée à la création).

Il est possible de couper les très grandes tables en partitions, plusieurs possibilités de découpage sont possibles :

- modèle historisé : découpage en fonction de dates
- partitionnement par plages de valeurs
- ...



L'analyse de tables

La requête SQL ANALYZE TABLE permet de collecter des informations sur une table qui seront utilisées par l'optimiseur statistique et seront très utiles pour l'administrateur.

Ces informations sont disponibles dans les vues **user_tables** ou **dba_tables**.

Exemples de requêtes :

```
ANALYZE TABLE emp COMPUTE STATISTICS;
ANALYZE TABLE emp ESTIMATE STATISTICS;
ANALYZE TABLE emp DELETE STATISTICS;
```

Exemple d'informations produites par la commande ANALYZE:

NUM_ROWS	Nombre de lignes de la table
BLOCKS	Nombre de blocs utilisés par la table
EMPTY_BLOCKS	Nombre de blocs vides dans le segment
AVG_SPACE	Nombre moyen d'octets libres par bloc
CHAIN_CNT	Nombre de lignes chaînées

A partir de ces informations on pourra connaître :

- Le pourcentage d'espace libre dans le segment ce qui peut permettre de prévoir les extensions.
- Le taux moyen de remplissage des blocs et le nombre de lignes chaînées.

Modifications structurelles de table

La commande ALTER TABLE permet de modifier la structure d'une table :

- Les attributs d'un segment, tous sauf l'INITIAL
- Le PCTFREE (% d'espace libre laissé dans les blocs lors du chargement) ou le PCTUSED (seuil de réutilisation des blocs)
- Ajout ou suppression d'une colonne
- ...

Exemples de commandes :

```
ALTER TABLE dept STORAGE ( NEXT 1m );
ALTER TABLE dept PCTFREE 5 PCTUSED 60;
```

En 8i on peut déplacer une table d'un tablespace à un autre, cette facilité peut être aussi utilisée pour réorganiser le stockage.

Exemple :

```
ALTER TABLE bemp MOVE TABLESPACE users
STORAGE ( INITIAL 2 M NEXT 1m );
```



La gestion des contraintes

Les contraintes d'intégrité permettent de s'assurer que les informations stockées dans les tables respectent des règles préétablies.

- Colonnes non nulles (*NOT NULL*)
- Colonnes uniques (*UNIQUE*)
- Clés primaires de table (non nulles et uniques) (*PRIMARY KEY*)
- Plages de valeurs ou domaine de définition d'une colonne (*CHECK*)
- Contrôle d'intégrité référentielle (*FOREIGN KEY*)

L'exploitant doit savoir contrôler l'état, et éventuellement valider ou invalider les contraintes.

Exemples de requêtes :

Liste des contraintes d'intégrités

```
SELECT table_name, constraint_name, constraint_type,
       search_condition, status
FROM user_constraints;
```

TABLE_NAME	CONSTRAINT_N	C	SEARCH_CONDITION	STATUS
BEMP	SYS_C002090	P	→ Primary key	ENABLED
BEMP	SYS_C001202	C	EMPNO IS NOT NULL (<i>check</i>)	ENABLED
DEPT	SYS_C002091	P		ENABLED
DOCINDEX	PK_DOCINDEX	P		ENABLED
EMP	SYS_C001005	C	EMPNO IS NOT NULL	ENABLED
EMP	FK_DEPTNO	R		ENABLED

Invalidation d'une contrainte

```
ALTER TABLE emp DISABLE CONSTRAINTS fk_deptno;
Table altered.
```



Validation d'une contrainte

```
ALTER TABLE emp ENABLE CONSTRAINTS fk_deptno;
Table altered.
```

Attention !

Si la table est volumineuse cette requête peut prendre beaucoup de temps, et en cas de défaut peut échouer.

Dans ce cas on peut utiliser la table des exceptions pour retrouver les lignes en défaut et ensuite pouvoir réparer les données.

La table des exceptions peut être créée avec le script d'administration suivant : `$ORACLE_HOME/rdbms/admin/utlexcpt.sql`

Validation d'une contrainte en utilisant la table des exceptions :

```
ALTER TABLE emp ENABLE CONSTRAINTS fk_deptno
EXCEPTIONS INTO exceptions;
*
ERROR at line 1:
ORA-02298: cannot enable (SCOTT.FK_DEPTNO) - parent keys not found
```

```
SELECT * FROM emp WHERE rowid
IN ( SELECT row_id FROM exceptions WHERE table_name='EMP');
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		17-NOV-81	5000		5

Les Triggers

Déclenchement événementiel.

Lorsque les règles de gestion des données d'une entreprise sont trop complexes à gérer avec des contraintes d'intégrité, on peut mettre en œuvre des triggers base de données.

La validation / invalidation de tous les triggers d'une table se fait avec la requête ALTER TABLE.

Contrairement aux contraintes, lors de la validation des triggers les données existantes ne sont pas contrôlées.

Exemple de trigger simple qui transforme la saisie de colonnes en lettres capitales :

```
CREATE OR REPLACE TRIGGER majuscule
BEFORE INSERT OR UPDATE
ON EMP
FOR EACH ROW (autant de fois que de ligne)
BEGIN
:new.ename := upper(:new.ename);
:new.job := upper(:new.job);
:new.hiredate := upper(:new.hiredate);
END;
/
```

Vérification de l'état des triggers :

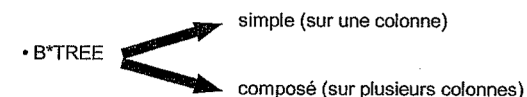
```
SELECT trigger_name, table_owner, table_name, status
FROM dba_triggers
```

TRIGGER_NAME	TABLE_OWNER	TABLE_NAME	STATUS
EMP_MAJUSCULE	SCOTT	EMP	ENABLED
...			

Les index

En règle générale, les données dans une table ne sont pas triées. Pour accélérer les traitements et éviter des balayages complets de tables volumineuses, on crée des index.

Les différentes catégories d'index sont :



BITMAP (sur plusieurs colonnes de faible cardinalité)

La visualisation des index existants et de leur état se fait à travers les vues USER_INDEXES ou DBA_INDEXES :

```
SELECT table_owner, index_name, table_name, status FROM dba_indexes
WHERE owner='SCOTT'
```

TABLE_OWNER	INDEX_NAME	TABLE_NAME	STATUS
SCOTT	I_ENAME_BEMP	BEMP	VALID
SCOTT	I_ENAME_EMP	EMP	VALID
SCOTT	PK_DOCINDEX	DOCINDEX	VALID
SCOTT	SYS_C002090	BEMP	VALID
SCOTT	SYS_C002091	DEPT	VALID
...			



Exemples :

création d'index {
CREATE INDEX i_emp_dptjob ON emp(deptno, job)
TABLESPACE indx
PCTFREE 20
STORAGE (INITIAL 500K NEXT 500K)
NOLOGGING;

CREATE BITMAP INDEX ib_emp_deptnojob ON emp(deptno, job)
TABLESPACE indx;

modif. { ALTER INDEX i_empdeptjob STORAGE(NEXT 1m);

suppression { DROP INDEX i_empdeptjob;

Analyse des Index

Comme les tables, les index peuvent être analysés. Les informations récupérées seront utilisées par l'optimiseur statistique pour déterminer l'utilisation ou pas des index.



TRAVAUX PRATIQUES

Visualisez les tables de votre schéma (tab ou user_tables).

Vérifiez que toutes les contraintes de table sont validées (user_constraints).

Vérifiez que tous les index sont valides (user_indexes).

user_tables → Table
user_indexes → Index
user_constraints → Contrainte



BASES DE DONNEES

Traces, Sessions et Transactions

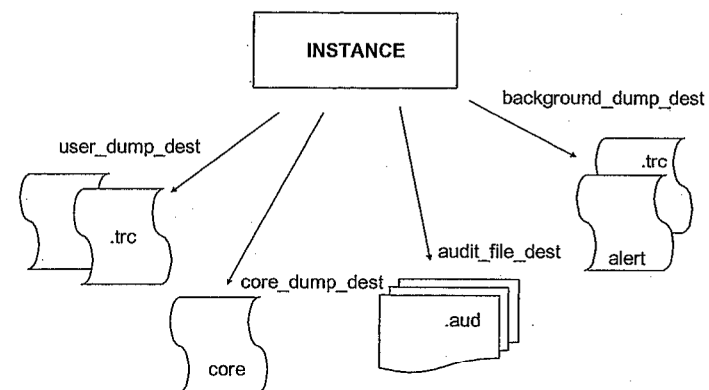
Objectif :

A la fin de ce module, vous serez capable de contrôler les sessions et transactions et surveiller les différents fichiers de trace.

- Les destinations de traces
- Le fichier d'alerte
- Les fichiers de trace des processus d'arrière plan
- Les traces SQL
- Les vues v\$session, v\$transaction, v\$lock
- L'ordre SQL-DML LOCK TABLE



Les destinations des traces et log



Il est conseillé de renseigner ces différents paramètres dans le fichier **"initSID.ora"** plutôt que laisser les valeurs par défaut.

nom du paramètre	valeur par défaut	remarque
core_dump_dest	?/dbs	V7 uniquement
user_dump_dest	?/rdbms/log	
background_dump_dest	?/rdbms/log	
audit_file_dest	?/rdbms/audit	

Les processus "background" (pmon,smon,dbwr...) créent au besoin les fichiers de trace des incidents dans le répertoire **background_dump_dest**.

L'instance y place aussi son fichier d'alerte (**alert_SID.log**) dans lequel sont consignés tous les événements majeurs de la vie d'une base : arrêt, démarrage, modifications de structures, bascules de log...

Dans le répertoire **user_dump_dest** seront placées les traces des processus serveur dédiés ou partagés.

Dans le répertoire de destination de l'audit sont tracées les sessions 'connect internal', un fichier par connexion.



Exemple de contenu du fichier d'alerte (alert_SID.log):

```

Tue Jan 30 15:22:39 2001
Completed: ALTER DATABASE CLOSE NORMAL
Tue Jan 30 15:22:39 2001
ALTER DATABASE DISMOUNT
Completed: ALTER DATABASE DISMOUNT
Tue Jan 30 15:24:19 2001
Starting ORACLE instance (normal)
LICENSE_MAX_SESSION = 0
LICENSE_SESSIONS_WARNING = 0
LICENSE_MAX_USERS = 0
Starting up ORACLE RDBMS Version: 8.0.5.0.0.
System parameters with non-default values:
  processes                = 50
  shared_pool_size          = 3500000
.../...
alter database mount
Thu Feb 1 10:32:15 2001
Successful mount of redo thread 1, with mount id 1408306927.
Thu Feb 1 10:32:15 2001
Database mounted in Exclusive Mode.
Completed: alter database mount
Thu Feb 1 10:32:15 2001
alter database open
Beginning crash recovery of 1 threads
Recovery of Online Redo Log: Thread 1 Group 1 Seq 107 Reading mem 0
  Mem# 0 errs 0: /tpora/gr1/log1A
Crash recovery completed successfully
Thu Feb 1 10:32:16 2001
Thread 1 advanced to log sequence 108
Thread 1 opened at log sequence 108
.../...
.../...
Fri Feb 2 10:14:32 2001
create tablespace tbs5 datafile '$PWD/tbs5.sql' size 1M
Completed: create tablespace tbs5 datafile '$PWD/tbs5.sql' si
Fri Feb 2 10:19:02 2001
Thread 1 advanced to log sequence 120
  Current log# 2 seq# 120 mem# 0: /tpora/gr1/log2A
  Current log# 2 seq# 120 mem# 1: /tpora/gr1/log4A
.../...
.../...

```

Passage du Redolog n° 108



Exemple de fichier de trace

```

$cat lgwr_33564_a.trc
Dump file /tpora/gr1/bdump/lgwr_33564_a.trc
Oracle8 Enterprise Edition Release 8.0.5.0.0 - Production
With the Partitioning and Objects options
PL/SQL Release 8.0.5.0.0 - Production
ORACLE_HOME = /oracle8/product/8.0.5
System name:      AIX
Node name:        aquitaine
Release:          3
Version:          4
Machine:          004C134A4C00
Instance name:    A
Redo thread mounted by this instance: 1
Oracle process number: 4
Unix process pid: 33564, image:

*** 2001.02.01.11.58.28.000
*** SESSION ID: (3.1) 2001.02.01.11.58.28.000
ORA-00313: open failed for members of log group 2 of thread 1
ORA-00312: online log 2 thread 1: '/tpora/gr1/log2A'
ORA-27037: unable to obtain file status
IBM AIX RISC System/6000 Error: 2: No such file or directory
Additional information: 3
ORA-00321: log 2 of thread 1, cannot update log file header
ORA-00312: online log 2 thread 1: '/tpora/gr1/log2A'
ORA-00320: cannot read file header from log 3 of thread 1

```

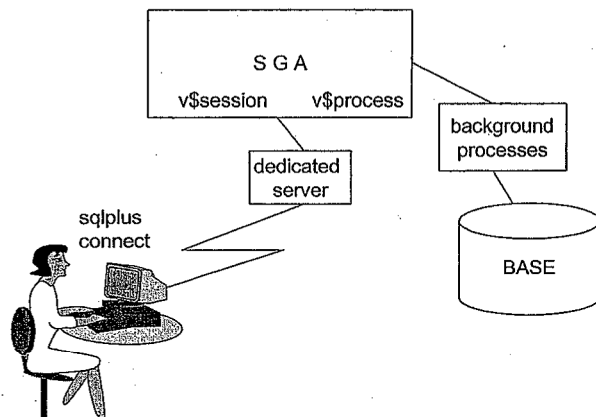
Tous ces répertoires sont à surveiller car oracle ne supprime pas les fichiers et ne surveille pas non plus leur taille.

Le paramètre max_dump_file_size = 10000 blocs par défaut permet de limiter les traces 'SQL' des processus serveur.

ATTENTION !

Le fichier alert_SID.log peut devenir rapidement très grand, en cas de problème récursif et bloquer un système de fichier.

Les sessions



Les vues dynamiques **v\$session** et **v\$process** donnent une grande quantité de renseignements.

Exemple de requête sur une base :

```
SQL> SELECT username, osuser, terminal, server, machine
2 FROM v$session
3 where type <> 'BACKGROUND'; (Supprimer les sessions Background)
```

USERNAME	OSUSER	TERMINAL	SERVER	MACHINE
SYSTEM	oracle8	pts/0	DEDICATED	aquitaine
PMAR	MAR-P	pelee	SHARED	pelee

L'utilisateur oracle SYSTEM est connecté en process serveur dédié, un process serveur par session, alors que l'utilisateur PMAR est connecté sur un dispatcher et utilise un process serveur partagé.

Les processus de 'background' établissent aussi des sessions qui sont visibles dans V\$SESSION.

Déconnexion d'un utilisateur

Pour déconnecter un utilisateur, connaissant le SID et le SERIAL# d'une session, un administrateur peut utiliser la requête SQL :

```
ALTER SYSTEM KILL SESSION 'sid,serial#'
```



A partir de la version 8.0 il peut aussi utiliser :

```
ALTER SYSTEM DISCONNECT SESSION
'sid,serial#' POST_TRANSACTION
```

select username, sid, serial# from v\$session;

La requête suivante fournit les informations qui éventuellement permettront de clore une session.

```
SQL> SELECT s.username, s.terminal, s.server, sid, s.serial#, p.spid
2 FROM v$session s, v$process p
3 WHERE s.paddr=p.addr AND s.username IS NOT NULL;
```

USERNAME	TERMINAL	SERVER	SID	SERIAL#	SPID
PMAR	pelee	NONE	8	2335	14400
SYSTEM	pts/0	DEDICATED	7	3934	29080

Si le serveur est du type dédié l'information SPID correspond au numéro de processus dans la table du système. Il pourra être utilisé dans une commande UNIX 'kill -9'.

Exemple de déconnexion (il faut être habilité) :

```
SQL> ALTER SYSTEM KILL SESSION '8,2335';
System altered.
```

Avec OEM Instance Manager on peut afficher et déconnecter les sessions.



Les transactions

Une transaction Oracle

Commence au premier ordre SQL-DML
(INSERT, UPDATE, DELETE)

Se termine sur COMMIT ou ROLLBACK

La vue V\$TRANSACTION donne les
informations sur les transactions en cours.

Exemple de requête :

```
col nom_rbs for a8
col nom_util for a15
SELECT s.username nom_util, s.terminal,
       p.spid, r.name nom_rbs, t.used_urec, t.used_ublk
FROM v$session s, v$process p, v$rollname r, v$transaction t
WHERE s.paddr=p.addr
      AND s.saddr=t.ses_addr
      AND r.usn=t.xidusn
```

NOM_UTIL	TERMINAL	SPID	NOM_RBS	USED_UREC	USED_UBLK
SYSTEM	pts/0	29080	RBS1	2	1

Cette requête donne :

- le nom de l'utilisateur
- le terminal
- le nom du rollback segment
- le nombre d'enregistrement d'annulation et de blocs d'annulation



Les verrous



Oracle gère des verrous de ligne et des verrous de table.

Les ordres INSERT, UPDATE, DELETE posent

- un verrou row exclusive sur la table modifiée
- un verrou exclusif sur les lignes modifiées

Les ordres DDL/DCL sont autocommittés.

```
SQL> SELECT s.username, s.serial#, s.sid,
2         o.object_name, o.owner, l.type, l.lmode
3 FROM v$session s, v$lock l, all_objects o
4 WHERE l.sid=s.sid
5 AND o.object_id=l.id1
6 AND username IS NOT NULL;
```

USERNAME	SERIAL#	SID	OBJECT_NAME	OWNER	TY	LMODE
SYSTEM	3934	7	DEPT	SYSTEM	TM	3
SYSTEM	3934	7	EMP	SYSTEM	TM	2
PMAR	2337	8	BEMP	PMAR	TM	6

Le type TM signifie verrou DML

Les modes de verrou

- 1 - null
- 2 - row share (SS), ordre SELECT ... FOR UPDATE
- 3 - row exclusive (SX), ordres INSERT UPDATE DELETE
- 4 - share (S)
- 5 - share row exclusive (SSX)
- 6 - exclusive (X), ordre LOCK TABLE

6 Type de verrou



La requête SQL LOCK

Pour effectuer des opérations de maintenance particulières sur une table, on peut prendre l'exclusivité de cette table avec l'ordre SQL

LOCK TABLE ...

Pour déverrouiller la table il suffira de faire

COMMIT

Exemple :

```
SQL> LOCK TABLE emp in EXCLUSIVE MODE;
```



TRAVAUX PRATIQUES

Visualisez les paramètres de destination de votre instance (user_dump_dest, background_dump_dest...) avec la commande « show parameter ».

Affichez le contenu de votre fichier d'alerte (alert_SID.log) et d'un éventuel fichier trace.

Visualisez les sessions actives.

Déconnectez un utilisateur avec ALTER SYSTEM.

Affichez les transactions (script /tpora/2898e/exch07-8.sql) et les verrous (/tpora/2898e/exch07-9.sql) en cours.



BASES DE DONNEES

Export / Import SQL*Loader

Objectif :

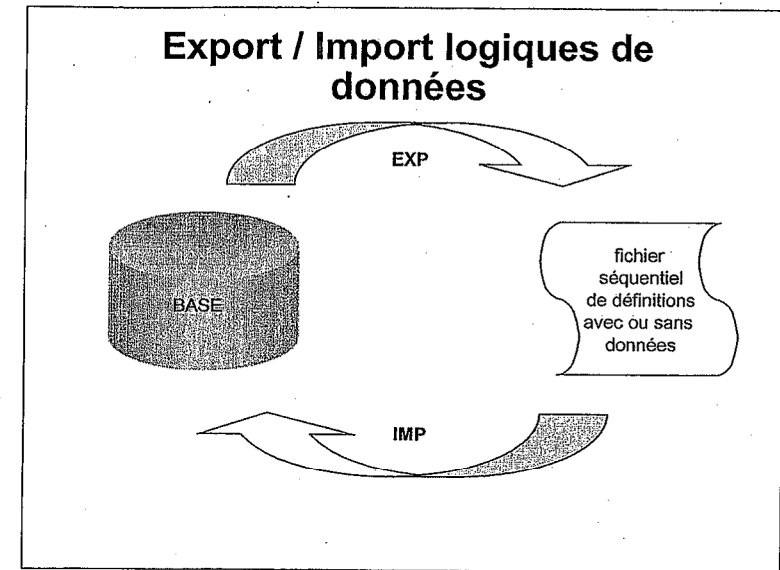
A la fin de ce module, vous posséderez les éléments nécessaires pour utiliser les outils EXPORT/IMPORT et SQL*LOADER.

- Sauvegardes et restaurations logiques
- Réorganisation de tables
- Chargement de données dans un format externe



on ne peut pas restaurer qu'à partir
d'un export import.

* → [+ → Backup online (7/semaine)
→ Export



EXP est un utilitaire logique qui peut être complémentaire aux sauvegardes physiques dans une stratégie de sauvegarde/restauration des données. Il permet de:

- remettre (importer) dans la base une table détruite par erreur,
- déplacer/dupliquer des données d'une base vers une autre,
- réorganiser le stockage des données,
- avoir un fichier source SQL de définition des structures,
- changer le propriétaire d'une table.

Le fichier d'export est dans un format séquentiel binaire, il ne doit pas être édité ou modifié. Ce fichier peut être transféré d'un O.S vers un autre pour être importé dans une autre base.

Par défaut l'initial extent des objets est recalculé à la taille du segment (compress=yes).

L'import peut être total ou partiel.

**EXP**

TABLES	UTILISATEUR SCHEMA	BASE
Définitions des Tables	Définitions des tables	Déf. des tablespaces
Données des tables	Données des tables	Déf. Des rollback seg.
Grants	Grants	Déf. Des profiles
Index	Index	Déf. Des utilisateurs
Contraintes	Contraintes	Privilèges
		Quotas
	Clusters	
	Définitions de liens	Clusters
	Vues	Définitions des tables
	Synonymes privés	Données des tables
	Séquences privés	Grants
	...	Index
		Contraintes
		Définitions de liens
		Vues
		Synonymes
		Séquences
		...

Pour utiliser la commande externe **exp** il faut que la base soit ouverte et que l'environnement de la commande soit correctement initialisé :
ORACLE_HOME, ORACLE_SID, NLS_LANG...

Pour exporter des utilisateurs ou toute la base il faut être DBA.

ATTENTION !

Les objets de l'utilisateur SYS ne sont jamais exportés.



Exemple d'export sous forme interactive:

```
(dba9i)9i/home>export NLS_LANG=french_france.we8iso8859p1  I M P O R T A N T
(dba9i)9i/home>exp system/manager

Export: Release 9.0.1.0.0 - Production on Je Dec 12 16:11:15 2002

(c) Copyright 2001 Oracle Corporation. All rights reserved.

Connecté à: Oracle9i Enterprise Edition Release 9.0.1.0.0 - 64bit Production
With the Partitioning option
JServer Release 9.0.1.0.0 - Production
Entrer taille mémoire tampon extract. par tableau : 4096 >

Fichier d'export : expdat.dmp > expfull.db.dmp

(1)B(dD Entière), (2)U(utilisat.) ou (3)T(ables): (2)U > 1

Export des privilèges (oui/non): oui >

Export des tables de données (oui/non): oui >

Comprimer les blocs contigus (oui/non): oui >

Export fait dans le jeu de car WE8ISO8859P1 et jeu de car NCHAR AL16UTF16

Prêt à exporter l'intégralité de la base de données ...
. export des définitions de tablespace
. export des profils
. export des définitions d'utilisateur
. export des rôles
. coût ressource export
. export des définitions de rollback segment
. export des liens de base de données (DATABASE LINKS)
. export des numéros de séquence
. export des alias de répertoire
. export des espaces de stockage des noms de contexte
. export des noms de bibliothèque de fonctions étrangères
. export des définitions de type d'objet
. export des actions et objets procéduraux du système
. export des actions et objets procéduraux de pré-schéma
. export des définitions de cluster
. Prêt à exporter les tables SYSTEM ... via le chemin classique...
. . export de la table      AQ$_INTERNET_AGENT_PRIVS      0 lignes exportées
. . export de la table      AQ$_INTERNET_AGENTS             0 lignes exportées
. . export de la table      CHAINED_ROWS                    0 lignes exportées
. .
. .
```



```

. Prêt à exporter les tables SCOTT ... via le chemin classique...
. export de la table          BONUS          0 lignes exportées
. export de la table          DEPT            4 lignes exportées
. export de la table          DUMMY           1 lignes exportées
. export de la table          EMP             57344 lignes exportées
. export de la table          SALGRADE        5 lignes exportées
. export de la table          T1              24576 lignes exportées
. export de la table          T2              0 lignes exportées
. Prêt à exporter les tables RACINE ... via le chemin classique...
. export de la table          BONUS          0 lignes exportées
. export de la table          CLIENTS         0 lignes exportées
. export de la table          COMMANDES       0 lignes exportées
. export de la table          DEPARTEMENT     0 lignes exportées
. export de la table          DEPT            4 lignes exportées
. export de la table          DUMMY           1 lignes exportées
. export de la table          EMP             14 lignes exportées
. export de la table          EVOL_OF_SAL     10 lignes exportées
. export de la table          EXCEPTIONS      0 lignes exportées
. export de la table          SALGRADE        5 lignes exportées
. export de la table          TRACE_SALAIRE   10 lignes exportées
. Prêt à exporter les tables JEAN ... via le chemin classique...
. Prêt à exporter les tables PIERRE ... via le chemin classique...
. export des synonymes
. export des vues
. export des contraintes d'intégrité référentielle
. export des procédures stockées
. export des opérateurs
. export des types d'index
. export des index bitmap, fonctionnels et extensibles
. export des actions post-tables
. export des déclencheurs
. export des vues matérialisées
. export des journaux de clichés
. export des files d'attente de travaux
. export des groupes de régénération et fils
. export des dimensions
. export des actions et objets procéduraux de post-schéma
. export de table historique utilisateur
. export des valeurs par défaut et options d'analyse du système
. export des statistiques
Procédure d'export terminée avec succès.
(dbas9i)9i/home>

```

L'aide de la commande EXP:

```

(dbas9i)9i/home>export NLS_LANG=french_france.we8iso8859p1
(dbas9i)9i/home>exp help=y

Export: Release 9.0.1.0.0 - Production on Je Dec 12 16:34:45 2002

(c) Copyright 2001 Oracle Corporation. All rights reserved.

Vous pouvez laisser l'export vous demander les param. en tapant la
commande EXP suivie de votre nom utilisateur/mot de passe :

Exemple : EXP SCOTT/TIGER

Pour contrôler l'exécution de l'export, entrer la commande EXP suivie de
divers arguments. Pour indiquer les paramètres, utiliser des mots-clés :

Format : EXP KEYWORD=valeur ou KEYWORD=(valeur1,valeur2,...,valeurN)
Exemple : EXP SCOTT/TIGER GRANTS=Y TABLES=(EMP,DEPT,MGR)
          ou TABLES=(T1:P1,T1:P2), si T1 est une table partitionnée

USERID doit être le premier paramètre de la ligne de commande.

```

Mot-clé	Description (Valeur par défaut)	Mot-clé	Description (Valeur par défaut)
USERID	nom utilisateur/mot de passe	FULL	export du fichier entier (N)
BUFFER	taille du tampon de données	OWNER	listes des noms utilisateur propriétaire
FILE	fichiers de sortie (EXPDAT.DMP)	TABLES	liste des noms de table
COMPRESS	import dans un extent (Y)	RECORDLENGTH	longueur de l'enregistrement d'E/S
GRANTS	export des autorisations d'accès (Y)	INCTYPE	type d'export incrémentiel
INDEXES	export des index (Y)	RECORD	suivi de l'export incr. (Y)
DIRECT	chemin direct (N)	TRIGGERS	export des déclencheurs (Y)
LOG	fichier journal de sortie d'écran	STATISTICS	analyse des objets (ESTIMATE)
ROWS	export des lignes de données (Y)	PARFILE	nom du fichier de paramètres
CONSISTENT	cohérence entre les tables	CONSTRAINTS	export des contraintes (Y)

```

FEEDBACK          afficher la progression toutes les x lignes (0)
FILESIZE          taille maximale de chaque fichier de vidage
FLASHBACK_SCN     SCN utilisé pour rétablir le cliché de session
FLASHBACK_TIME    temps d'obtention du SCN le plus près du temps indiqué
QUERY            clause select utilisée pour exporter un sous-ensemble d'une table
RESUMABLE         suspension lorsqu'une erreur associée à l'espace se produit (N)
RESUMABLE_NAME    chaîne de texte utilisée pour identifier l'instruction RESUMABLE
RESUMABLE_TIMEOUT temps d'attente pour RESUMABLE
TTS_FULL_CHECK    exécuter un contrôle de dépendance complet ou partiel pour TTS
VOLSIZE          nombre d'octets à écrire sur chaque volume de bande
TABLESPACES       liste des tablespaces à exporter
TRANSPORT_TABLESPACE - export des métadonnées des tablespaces transportables (N)
TEMPLATE nom de modèle appelant l'export du mode ias

Procédure d'export terminée avec succès.
(dbas9i)9i/home>

```



EXP sous forme ligne de commande

La ligne de commande peut être mise dans un script de sauvegarde.

Pour la confidentialité et faciliter la maintenance, les paramètres peuvent être placés dans un fichier (parfile=).

Sous cette forme il est possible de faire:

- des sauvegardes incrémentales
- des exports de tablespace (définitions des objets) 8i / 9i
- des exports consistants

Exemples de commandes :

Utilisation d'un fichier de paramètres

```
exp parfile=param_exp_quoti
```

Sauvegarde totale en mode consistant.

```
exp system/manager full=y consistent=y file=totale_`date +%j`.dmp
```

Sauvegarde incrémentale cumulative

- X totale
- C cumulative
- I incrémentale

```
exp system/manager full=y inctype=C record=y file=C_`date +%j`.dmp
```



IMP

Utilitaire de chargement de table à partir d'un fichier créé par EXP.

Utilisable en interactif pour les utilisations basiques :

- liste du contenu du fichier d'export
- importation de la totalité du fichier
- importation partielles.

En ligne de commandes pour :

- Changer le propriétaire de tables
- utilisations incrémentales
- les tablespaces transportables

Remarques:

Contrairement à l'export qui est assez rapide l'import est une opération qui prend beaucoup plus de temps.

Il faut prévoir un gros rollback segment car la validation (COMMIT) se fait table par table : pour une table occupant 100Mo il faudra 100Mo de rollback.

Par défaut, les objets importés ne doivent pas exister dans la base.

Toutes les contraintes déclaratives sont contrôlées lors de l'insertion des lignes.

Les index doivent, dans la mesure du possible, être créés après le chargement des données.

L'option indexfile permet de donner à **imp** le nom du fichier SQL qui servira à générer les index après le chargement des données.

Aide de la commande **IMP** :

```
(dba9i)9i/home>imp help=y
```

Import: Release 9.0.1.0.0 - Production on Mon Dec 16 13:35:31 2002
(c) Copyright 2001 Oracle Corporation. All rights reserved.
You can let Import prompt you for parameters by entering the IMP
command followed by your username/password:

Example: IMP SCOTT/TIGER

Or, you can control how Import runs by entering the IMP command followed
by various arguments. To specify parameters, you use keywords:

Format: IMP KEYWORD=value or KEYWORD=(value1,value2,...,valueN)
Example: IMP SCOTT/TIGER IGNORE=Y TABLES=(EMP,DEPT) FULL=N
or TABLES=(T1:P1,T1:P2), if T1 is partitioned table

USERID must be the first parameter on the command line.

Keyword	Description (Default)	Keyword	Description (Default)
USERID	username/password	FULL	import entire file (N)
BUFFER	size of data buffer	FROMUSER	list of owner usernames
FILE	input files (EXPDAT.DMP)	TOUSER	list of usernames
SHOW	just list file contents (N)	TABLES	list of table names
IGNORE	ignore create errors (N)	RECORDLENGTH	length of IO record
GRANTS	import grants (Y)	INCTYPE	incremental import type
INDEXES	import indexes (Y)	COMMIT	commit array insert (N)
ROWS	import data rows (Y)	PARFILE	parameter filename
LOG	log file of screen output	CONSTRAINTS	import constraints (Y)
DESTROY	overwrite tablespace data file (N)		
INDEXFILE	write table/index info to specified file		
SKIP_UNUSABLE_INDEXES	skip maintenance of unusable indexes (N)		
FEEDBACK	display progress every x rows(0)		
TOID_NOVALIDATE	skip validation of specified type ids		
FILESIZE	maximum size of each dump file		
STATISTICS	import precomputed statistics (always)		
RESUMABLE	suspend when a space related error is encountered(N)		
RESUMABLE_NAME	text string used to identify resumable statement		
RESUMABLE_TIMEOUT	wait time for RESUMABLE		
COMPILE	compile procedures, packages, and functions (Y)		
VOLSIZE	number of bytes in file on each volume of a file on tape		

The following keywords only apply to transportable tablespaces
TRANSPORT_TABLESPACE import transportable tablespace metadata (N)
TABLESPACES tablespaces to be transported into database
DATAFILES datafiles to be transported into database
TTS_OWNERS users that own data in the transportable tablespace set

Import terminated successfully without warnings.

Remarques sur les paramètres et leur utilisation :

IGNORE

Mis à YES, sert à charger des données dans une table existante, plusieurs cas d'utilisation sont possibles.

- Le CREATE TABLE ne passe pas car les paramètres de STORAGE ne sont pas compatible avec l'espace physique disponible.
- La table ne se charge pas car le rollback segment est trop petit, dans ce cas faire le chargement en plusieurs fois. La première fois, on met le paramètre COMMIT à YES et les fois suivantes on ajoute le paramètre IGNORE à YES.
- On veut volontairement ajouter des lignes supplémentaires à une table.

FROMUSER/TOUSER

Ces deux paramètres, utilisés conjointement, servent à changer le propriétaire d'objets. on a retrouvé le schéma

INCTYPE

Ce paramètre est utilisé en cas de sauvegarde incrémentale.

Opérations de restauration :

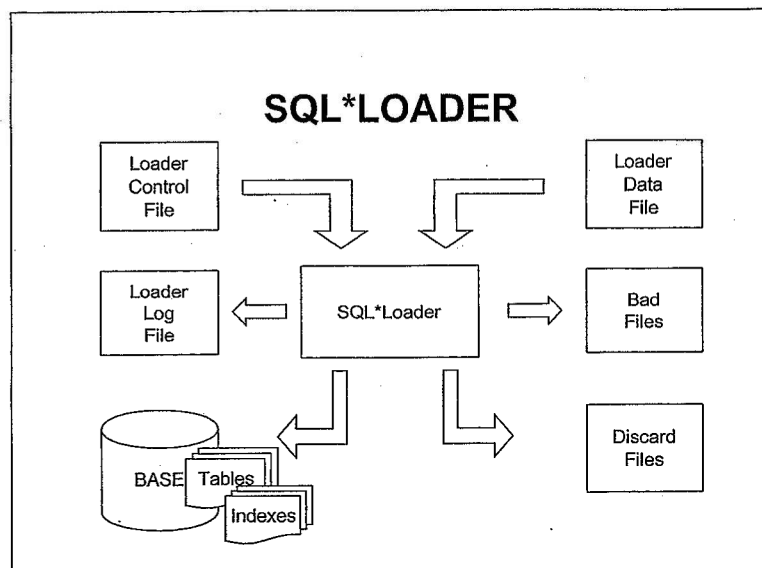
- passer la dernière sauvegarde générée avec INCTYPE=SYSTEM
- passer toutes les sauvegardes utiles dans l'ordre de création avec INCTYPE=RESTORE

Mode opératoire pour refaire une base détruite complètement.

Il faut exécuter la procédure 'CREATE DATABASE' avec le chargement du dictionnaire et un rollback segment supplémentaire actif dans le tablespace SYSTEM.

Ensuite on pourra faire l'import de la totalité de la base.

Si les noms des fichiers de la base sont différents dans ce cas il faut avant de faire l'importation créer les tablespaces et laisser DESTROY à NO.



Il faut que les tables existent : SQL*Loader ne crée pas d'objets dans la base.

Trois modes de chargement des données sont possibles :

[INSERT] Valeur par défaut : les tables doivent être vides

REPLACE La table est vidée, les lignes nouvelles sont ajoutées.

APPEND Seules les lignes nouvelles seront ajoutées, Les autres seront rejetées (contrôle de clé primaire).

Attention !

Il ne faut pas utiliser le mode **direct**, si on n'est pas sûr de ses données, car dans ce cas, les contrôles d'intégrité ne sont pas effectués.



TRAVAUX PRATIQUES

Exportez votre base ou les tables d'un utilisateur dans un fichier du répertoire courant.

Supprimez les tables DEPT et EMP.

Importez les tables DEPT et EMP.

*Supprimez scott/tiger
drop TABLE DEPT
EMP*



BASES DE DONNEES

Sauvegarde et restauration physiques

Objectif :

A la fin de ce module, vous posséderez les éléments nécessaires pour sauvegarder et restaurer physiquement les données de votre base, et mettre en œuvre l'archivage.

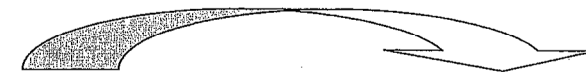
- Les sauvegardes physiques
- Mise en œuvre de l'archivage
- Les restaurations



ALTER SYSTEM SWITCH LOG FILE;
Permet de basculer de redo-log et d'archiver (Si ArchiveLog).



Sauvegardes physiques

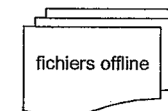


Sauvegarde physique
avec un utilitaire O.S
cp, backup, cpio, tar..



BASE

- fichiers de données
- fichiers de log
- fichier de contrôle
- fichier paramètres



fichiers offline

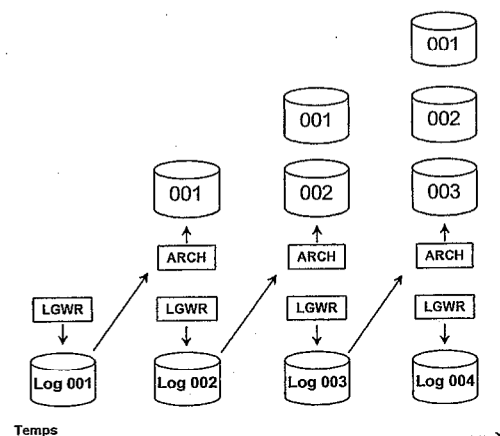
↓
 { init.ora { tnsnames.ora
 SPFILE(9i) TNSNAMES.ora
 SQLNET.ora

Pour faire une sauvegarde cohérente, il faut que la base soit **fermée** (sauvegarde à **froid**). Si la base doit fonctionner 24h sur 24h, alors il faudra être en mode **ARCHIVELOG** pour pouvoir effectuer des sauvegardes physiques base **ouverte** (sauvegarde à **chaud**).

Si la base est du type OLTP (On Line Transactionnal Processing) ou qu'il n'est pas acceptable de perdre des données en cas de défaillance de disque, il est fortement conseillé de faire fonctionner la base en mode **ARCHIVELOG** et de mettre les fichiers "redo-log" et de contrôle en miroir.



Qu'est ce que l'archivage



Si le processus ARCH est activé, dès qu'un fichier de log est plein, ARCH le recopie dans un fichier offline.

A partir de la version 8, on peut avoir plusieurs processus d'archivage et plusieurs destinations.

Lorsqu'une base est en mode **ARCHIVELOG**, LGWR ouvrira un fichier de log en écriture que si celui-ci a été archivé par le processus ARCH.



Mise en œuvre de l'archivage

Mise en œuvre du mode ARCHIVELOG

```
SVRMGR> SHUTDOWN
SVRMGR> STARTUP MOUNT
SVRMGR> ALTER DATABASE ARCHIVELOG;
SVRMGR> ALTER DATABASE OPEN;
```

Mise en œuvre du mode NOARCHIVELOG

```
...
SVRMGR> ALTER DATABASE NOARCHIVELOG;
...
```

La validation/invalidation de l'archivage se fait en phase "mount" du processus de démarrage de la base.

Pour que l'archivage soit géré automatiquement par le/les processus "ARCH" il faut mettre en place des paramètres d'instance dans le fichier **initSID.ora**

PARAMETRE	SIGNIFICATION	Ver.
log_archive_dest	Destination de base	
log_archive_dest_1	Destinations 1 à 5	8i
log_archive_dest_state_1	Etat des destinations 1 à 5	8i
log_archive_duplex_dest	Destination dupliquée	8
log_archive_max_processes	Nb. De processus maxi	8i
log_archive_min_succeed_dest	Nb. Mini de dest. OK	8
log_archive_start	True/false	
log_archive_format	Format des fichiers archives	



Exemples de paramètres d'instance :

Version 7

```
log_archive_dest = /oracle/oradata/PROD/arch
log_archive_format = log%s.arc
log_archive_start = true
```

Version 8i / 9i

```
log_archive_min_succeed_dest = 2
log_archive_dest_1 = "LOCATION=/ora8i/oradata/PROD/arch MANDATORY"
log_archive_dest_2 = "SERVICE=standby_db "
log_archive_dest_3 = "LOCATION=/archives2 OPTIONAL"
log_archive_max_processes = 3
log_archive_min_succeed_dest = 2
log_archive_dest_state_3 = DEFER
log_archive_format = "log%s.arc"
log_archive_start = true
```

La commande archive log

Sous SVRMGRL (ou SQL*Plus à partir de 8i)

```
ARCHIVE LOG [START] [STOP] [LIST] [NEXT] [<n>] [ALL] ['destination']
```

- **START** démarrage du ou des processus ARCH
- **STOP** arrêt du ou des processus ARCH
- **LIST** affichage des informations
- **NEXT** archivage manuellement du REDOLOG suivant non archivé
- **ALL** archivage de tous les REDOLOG non archivés
- **desti.** Fichier destination des archives
si non spécifié les paramètres d'instance sont utilisés

SQL> ARCHIVE LOG LIST] pour savoir si ArchiveLog ouvert



Sauvegarde Base arrêtée

Il faut sauvegarder

- Le fichier paramètres initSID.ora
- Le fichier de contrôle (v\$controlfile)
- Les fichiers des tablespaces (v\$dbfile)
- Les fichiers redo logs (v\$logfile)

En cas de panne

NOARCHIVELOG

Restaurer l'ensemble des fichiers sauvegardés, l'activité entre la sauvegarde et la panne est perdue

ARCHIVELOG

Restaurer le ou les tablespaces en défaut et appliquer un recouvrement des données (recover)

En cas de panne en mode **NOARCHIVELOG**, si les fichiers redo log online sont suffisamment gros ou qu'ils n'ont pas bouclé (fort peu probable), une reprise sans pertes peut être envisagée.

En mode **ARCHIVELOG**, Il ne faut surtout pas restaurer les fichiers REDOLOG sauvegardés et les fichiers de contrôle si on veut être en mesure de faire une reprise totale à partir des archives et fichiers REDOLOG ("recovery").

Sauvegarde Base ouverte

Il faut être en mode ARCHIVELOG

Sauvegarde du fichier de paramètre initSID.ora à chaque modification

Sauvegarde de fichier de contrôle à chaque changement de structure
alter database backup ...

Sauvegarde des fichiers de données des tablespaces
SQL> alter tablespace nom_tbs begin backup;
SQL> host 'copie des fichiers'
SQL> alter tablespace nom_tbs end backup;

En cas de panne :

Pour un tablespace autre que le tablespace SYSTEM on pourra :

- Passer le tablespace en panne offline
- Restaurer le ou les fichiers du tablespace à partir de la sauvegarde
- Sous SVRMGR faire : **recover tablespace nom_tbs**

Si le tablespace SYSTEM est panne ou que la base est arrêtée :

- Restaurer le ou les fichiers du tablespace à partir de la sauvegarde
- Sous SVRMGR faire : **startup mount**
- Sous SVRMGR faire : **recover database**
- Sous SVRMGR faire : **alter database open;**

RECOVER

Cette commande sert à réaligner:

- une base
- un tablespace
- un fichier de données

En cas de réaligement d'une base, il est possible une récupération incomplète et ouvrir la base en abandonnant des transactions.

SVRMGR> RECOVER DATABASE UNTIL CHANGE 54126
SVRMGR> RECOVER TABLESPACE *camrel* *nom_Archive*

En cas de reprise partielle, il faut évidemment que tous les fichiers soient dans un état antérieur ou égal au point de reprise.

Par défaut, la reprise est faite en mode interactif, avec confirmation à chaque fichier d'archive.

Le 'recover' peut aussi être fait sous SQL avec la requête alter database.

Exemples :

```
SQL> ALTER DATABASE
      RECOVER AUTOMATIC DATABASE;
```

```
SQL> ALTER DATABASE AUTOMATIC
      RECOVER UNTIL TIME '2001-02-25:14:00:00';
```

```
SQL> ALTER DATABASE
      RECOVER TABLESPACE users;
```



Exemple de sauvegarde d'un tablespace base ouverte:

```
SQL> select instance_name,host_name,version,status
2 from v$instance;
```

INSTANCE_NAME	HOST_NAME	VERSION	STATUS
MANU	lipari	9.0.1.0.0	OPEN

```
SQL> archive log list
Database log mode          Archive Mode
Automatic archival         Enabled
Archive destination        /ora9i/oradata/MANU/arch
Oldest online log sequence 103
Next log sequence to archive 105
Current log sequence       105
SQL> alter tablespace users begin backup;
Tablespace altered.

SQL> select tablespace_name,file_name
2 from dba_data_files
3 where tablespace_name='USERS';
```

TABLESPACE_NAME	FILE_NAME
USERS	/ora9i/oradata/MANU/users01.dbf
USERS	/ora9i/oradata/MANU/users02.dbf

```
SQL> !cp /ora9i/oradata/MANU/users* /ora9i/home/MANU/save
SQL> !ls -l /ora9i/home/MANU/save
total 34256
-rw-r----- 1 dba9i dba 15212544 21 jan 17:00 users01.dbf
-rw-r----- 1 dba9i dba 1056768 21 jan 17:00 users02.dbf
SQL> alter tablespace users end backup;
Tablespace altered.
```

Exemple de restauration du tablespace USERS base ouverte:

```
(dba9i)9i/home>sqlplus demo/demo
SQL*Plus: Release 9.0.1.0.0 - Production on Wed Jan 22 11:38:53 2003
SQL> select count(*) from emp;
select count(*) from emp
*
```

```
ERROR at line 1:
ORA-01116: error in opening database file 5
ORA-01110: data file 5: '/ora9i/oradata/MANU/users01.dbf'
ORA-27041: unable to open file
IBM AIX RISC System/6000 Error: 2: No such file or directory
Additional information: 3
```



```
(dba9i)9i/home>sqlplus /nolog

SQL*Plus: Release 9.0.1.0.0 - Production on Wed Jan 22 11:45:14 2003
(c) Copyright 2001 Oracle Corporation. All rights reserved.
SQL> connect / as sysdba
Connected.
SQL> select tablespace_name,file_name,file_id,status
2 from dba_data_files
3 where file_id = 5;
```

TABLESPACE_NAME	FILE_NAME	FILE_ID	STATUS
USERS	/ora9i/oradata/MANU/users01.dbf	5	AVAILABLE

```
SQL> alter tablespace users offline immediate;
Tablespace altered.

SQL> !cp /ora9i/home/MANU/save/users01.dbf /ora9i/oradata/MANU

SQL> archive log list;
Database log mode          Archive Mode
Automatic archival         Enabled
Archive destination        /ora9i/oradata/MANU/arch
Oldest online log sequence 108
Next log sequence to archive 110
Current log sequence       110

SQL> recover tablespace users
ORA-00279: change 271888 generated at 01/21/2003 16:51:24 needed for thread 1
ORA-00289: suggestion : /ora9i/oradata/MANU/arch/log0000000105.arc
ORA-00280: change 271888 for thread 1 is in sequence #105
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}

ORA-00279: change 271905 generated at 01/21/2003 17:01:11 needed for thread 1
ORA-00289: suggestion : /ora9i/oradata/MANU/arch/log0000000106.arc
ORA-00280: change 271905 for thread 1 is in sequence #106
ORA-00278: log file '/ora9i/oradata/MANU/arch/log0000000105.arc' no longer
needed for this recovery
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}

ORA-00279: change 272872 generated at 01/22/2003 11:29:58 needed for thread 1
ORA-00289: suggestion : /ora9i/oradata/MANU/arch/log0000000107.arc
ORA-00280: change 272872 for thread 1 is in sequence #107
ORA-00278: log file '/ora9i/oradata/MANU/arch/log0000000106.arc' no longer
needed for this recovery
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}

Log applied.
Media recovery complete.
SQL> alter tablespace users online;
Tablespace altered.
```



TRAVAUX PRATIQUES

Mettez en place l'archivage.

Arrêtez votre base et sauvegardez la (en copiant tous les fichiers dans un répertoire de sauvegarde).

*cp *.dbf*

Redémarrez votre base.

Effectuez quelques transactions sur les tables EMP ou DEPT d'un utilisateur de démonstration (SCOTT ou DEMO ou ...).

+ COMMIT

Faites des bascules de redolog manuelles (alter system switch logfile) ~~2~~ à 3 fois.

Arrêtez votre base.

Demandez à l'animateur de provoquer la panne.

Réparez le tablespace USERS en restaurant physiquement (utilitaire O.S) le ou les fichiers détruits du tablespace USERS.

Puis, effectuez le réaligement (récupération logique) du tablespace USERS avec la commande recover.

Contrôlez que vous n'avez pas perdu de transactions.

delete scott



BASES DE DONNEES

Solutions

des travaux pratiques

*no1 01
data 01
indx 01*

CHAPITRE 2

Vérification de son environnement de travail :

```
(dba8i)8i/home>env |grep ORA
ORACLE_BASE=/ora8i/app/oracle
ORACLE_SID=A817
ORA_NLS33=/ora8i/app/oracle/product/8.1.7/ocommon/nls/admin/data
ORACLE_HOME=/ora8i/app/oracle/product/8.1.7
A_z=! LOGNAME=$ ORACLE_SID
(dba8i)8i/home>
```

Le fichier de paramètres de l'instance.

```
(dba8i)8i/home>ls $ORACLE_HOME/dbs/init*
/ora8i/app/oracle/product/8.1.7/dbs/initA817.ora
/ora8i/app/oracle/product/8.1.7/dbs/initB817.ora
/ora8i/app/oracle/product/8.1.7/dbs/initC817.ora
/ora8i/app/oracle/product/8.1.7/dbs/initD817.ora
/ora8i/app/oracle/product/8.1.7/dbs/initLT817.ora
/ora8i/app/oracle/product/8.1.7/dbs/init.ora
(dba8i)8i/home> view $ORACLE_HOME/dbs/initA817.ora
# replace DEFAULT with your database name
db_name=A817
db_domain=world
db_files = 80
db_file_multiblock_read_count = 8
db_block_buffers = 200
shared_pool_size = 5000000
log_checkpoint_interval = 10000
processes = 50
parallel_max_servers = 5
:x
(dba8i)8i/home>
```

Exécution SVRMGR:

```
(dba8i)8i/home> svrmgrl (ou sqlplus /nolog)
Oracle Server Manager Release 3.1.7.0.0 - Production
Copyright (c) 1997, 1999, Oracle Corporation. All Rights Reserved.
Oracle8i Enterprise Edition Release 8.1.7.0.0 - Production
With the Partitioning option
JServer Release 8.1.7.0.0 - Production

SVRMGR> connect internal (ou connect / as sysdba)
Connected.
SVRMGR> show sga
Total System Global Area 26943380 bytes
Fixed Size 73620 bytes
Variable Size 25972736 bytes
Database Buffers 819200 bytes
Redo Buffers 77824 bytes
SVRMGR>
```

Suite SVRMGR

```
SVRMGR> show parameters
NAME                                TYPE      VALUE
-----
O7_DICTIONARY_ACCESSIBILITY         boolean    TRUE
active_instance_count               integer
always_anti_join                   string     NESTED_LOOPS
always_semi_join                    string     standard
aq_tm_processes                     integer    0
audit_file_dest                     string     /tpora/grl/audit
.../...
user_dump_dest                      string     /tpora/grl/udump
utl_file_dir                        string
SVRMGR> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SVRMGR> startup nomount
ORACLE instance started.
Total System Global Area          26943380 bytes
Fixed Size                         73620 bytes
Variable Size                     25972736 bytes
Database Buffers                   819200 bytes
Redo Buffers                       77824 bytes
SVRMGR> alter database mount;
Statement processed.
SVRMGR> alter database open;
Statement processed.
```

Consultation des vues du dictionnaire

```
SVRMGR> select instance_name, startup_time, status, logins, shutdown_pending
2> from v$instance;
INSTANCE_NAME  STARTUP_T STATUS LOGINS SHU
-----
A817          22-JAN-02 OPEN  ALLOWED NO
1 row selected.
SVRMGR> select name, created from v$databases;
NAME          CREATED
-----
A817          15-JAN-02
1 row selected.
```

CHAPITRE 3

```
$ cd $ORACLE_HOME/network/admin
$ view listener.ora
# Generated by Oracle configuration tools.

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP) (HOST = lipari) (PORT = 1521))
      )
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC))
      )
    )
  )

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = PLSExtProc)
      (ORACLE_HOME = /ora8i/app/oracle/product/8.1.7)
      (PROGRAM = extproc)
    )
  )

:x
$ view tnsnames.ora
.../...

B817 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = lipari) (PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = B817.world)
    )
  )
.../...
:x
$ lsnrctl stat

.../...
.../...

$ sqlplus scott/tiger@b817
```

CHAPITRE 4

TABLESPACE

```
SQL> col name for a40
SQL> select * from v$dbfile;
```

```
FILE# NAME
```

```
-----
1 /tpora/gr1/systemf1_A817.dbf
2 /tpora/gr1/rbsf1_A817.dbf
3 /tpora/gr1/tempf1_A817.dbf
4 /tpora/gr1/toolsf1_A817.dbf
5 /tpora/gr1/usersf1_A817.dbf
```

```
SQL> col file_name for a40
SQL> select file_name,tablespace_name,bytes,autoextensible
       from dba_data_files;
```

FILE_NAME	TABLESPACE_N	BYTES	AUT
/tpora/gr1/systemf1_A817.dbf	SYSTEM	52428800	YES
/tpora/gr1/rbsf1_A817.dbf	RBS	3145728	NO
/tpora/gr1/tempf1_A817.dbf	TEMP	4194304	NO
/tpora/gr1/toolsf1_A817.dbf	TOOLS	2097152	NO
/tpora/gr1/usersf1_A817.dbf	USERS	2097152	NO

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

```
STATUS NAME
```

```
-----
/tpora/gr1/ctl1_A.ora
```

```
.../...
```

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

```
GROUP# STATUS MEMBER
```

```
-----
1 /tpora/gr1/log1_A817.dbf
```

```
.../...
```

```
SQL> select * from dba_tablespaces;
```

TABLESPACE_N	INITIAL_EXTENT	NEXT_EXTENT	MIN_EXTENTS	MAX_EXTENTS	PCT_INCREASE
SYSTEM	12288	12288	1	249	50
RBS	131072	131072	2	249	

```
SQL> create tablespace essai
       datafile '/tpora/gr1/essaif1_A817.dbf' size 2M;
Tablespace created
```

```
SQL> alter database datafile '/tpora/gr1/essaif1.dbf' resize 3M;
Database altered
```

```
SQL> select name,status,create_bytes,bytes from v$datafile;
NAME                                STATUS  CREATE_BYTES  BYTES
-----
/tpora/gr1/systemf1_A817.dbf        SYSTEM      52428800  52428800
.../...
/tpora/gr1/essaif1_A817.dbf         ONLINE      2097152   3145728
```

```
SQL> alter tablespace essai default storage (initial 40k);
Tablespace altered
```

```
SQL> select * from dba_tablespaces where tablespace_name='ESSAI';
```

TABLESPACE_N	INITIAL_EXTENT	NEXT_EXTENT	MIN_EXTENTS	MAX_EXTENTS	PCT_INCREASE
ESSAI	40960	20480	1	249	50

Rollback segments

```
SQL> create rollback segment rbs_essai
       tablespace essai;
Rollback segment created
```

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

```
SQL> select * from dba_rollback_segs;
```

```
-- On a enfin trouvé ce rollback segment
```

```
SQL> alter rollback segment rbs_essai online;
Rollback segment altered
```

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

```
USN NAME
```

```
-----
0 SYSTEM
```

```
.../...
```

```
6 RBS_ESSAI
```

```
SQL> select * from dba_rollback_segs;
```

CHAPITRE 5

```

SQL> create user util1 identified by util1
      default tablespace users
      quota 300k on users;
User created.

SQL> create role role1;
Role created.

SQL> grant create session,create view, create synonym to role1;
Grant succeeded.

SQL> grant role1 to util1;
Grant succeeded.

SQL> connect scott/tiger
Connected.

SQL> grant select,insert,update on emp to util1;
Grant succeeded.

SQL> connect util1/util1
Connected.

SQL> create table tab1 (
  2 col1 number,
  3 col2 varchar2(15));
create table tab1 (
*
ERROR at line 1:
ORA-01031: insufficient privileges

SQL> update scott.emp set sal=6000 where ename='KING';
1 row updated.

SQL> create synonym semp for scott.emp;
Synonym created.

SQL> connect system/manager;
Connected.

SQL> grant create table to role1;
Grant succeeded.

SQL> alter user util1 temporary tablespace temp;
User altered.

```

```

SQL> connect util1/util1;
Connected.

SQL> create table uemp as select * from semp;
Table created.

SQL> select * from tab;

TNAME                                TABTYPE  CLUSTERID
-----
SEMP                                  SYNONYM
UEMP                                  TABLE

SQL> select * from user_users;

USERNAME                                USER_ID ACCOUNT_STATUS
-----
LOCK_DATE EXPIRY_DA DEFAULT_TABLESPACE
-----
TEMPORARY_TABLESPACE    CREATED    INITIAL_RSRC_CONSUMER_GROUP
-----
EXTERNAL_NAME
-----
UTIL1                                20 OPEN
USERS
TEMP                                23-JAN-02 DEFAULT_CONSUMER_GROUP

SQL> select * from user_role_privs;

USERNAME                                GRANTED_ROLE    ADM DEF OS_
-----
UTIL1                                ROLE1            NO YES NO

SQL> select * from role_sys_privs;

ROLE                                PRIVILEGE                                ADM
-----
ROLE1                                CREATE SESSION                                NO
ROLE1                                CREATE SYNONYM                                NO
ROLE1                                CREATE TABLE                                NO
ROLE1                                CREATE VIEW                                NO

SQL> select * from user_tab_privs_recd;

OWNER                                TABLE_NAME
-----
GRANTOR                                PRIVILEGE                                GRA
-----
SCOTT                                EMP
SCOTT                                INSERT                                NO

SCOTT                                EMP
SCOTT                                SELECT                                NO

SCOTT                                EMP
SCOTT                                UPDATE                                NO

```

CHAPITRE 6

```
SQL> connect scott/tiger
```

```
Connected.
```

```
SQL> desc user_tables
```

Name	Null?	Type
TABLE_NAME	NOT NULL	VARCHAR2(30)
TABLESPACE_NAME		VARCHAR2(30)
CLUSTER_NAME		VARCHAR2(30)
IOT_NAME		VARCHAR2(30)
PCT_FREE		NUMBER
PCT_USED		NUMBER
...		

```
SQL> select * from cat;
```

TABLE_NAME	TABLE_TYPE
BONUS	TABLE
DEPT	TABLE
DUMMY	TABLE
EMP	TABLE
SALGRADE	TABLE

```
SQL> select table_name,constraint_name,constraint_type,status
2 from user_constraints;
```

TABLE_NAME	CONSTRAINT_NAME	C	STATUS
EMP	SYS_C00772	C	ENABLED

```
SQL> alter table emp add primary key (empno);
Table altered.
```

```
SQL> select table_name,index_name,status from user_indexes;
```

TABLE_NAME	INDEX_NAME	STATUS
EMP	SYS_C00818	VALID

CHAPITRE 7

```
SQL> connect system/manager
```

```
Connected.
```

```
SQL> show parameter dump_dest
```

NAME	TYPE	VALUE
background_dump_dest	string	/tpora/gr1/bdump
core_dump_dest	string	/tpora/gr1/cdump
user_dump_dest	string	/tpora/gr1/udump

```
SQL> show parameter audit_file
```

NAME	TYPE	VALUE
audit_file_dest	string	/tpora/gr1/audit

```
SQL> host
```

```
(dba8i)/tpora/gr1>cd bdump
(dba8i)/tpora/gr1/bdump>ls -l
total 58
-rw-r--r-- 1 dba8i dba 28699 23 jan 10:22 alert_A817.log
(dba8i)/tpora/gr1/bdump>pg alert_A817.log
<RET>
<RET>
(dba8i)/tpora/gr1/bdump>cd ../udump
(dba8i)/tpora/gr1/udump>ls -l
(dba8i)/tpora/gr1/udump>cd ../cdump
(dba8i)/tpora/gr1/cdump>ls -l
(dba8i)/tpora/gr1/cdump>cd ../audit
(dba8i)/tpora/gr1/audit>ls -l
total 4
-rw-r----- 1 dba8i dba 704 22 jan 14:15 ora_21456.aud
-rw-r----- 1 dba8i dba 529 15 jan 16:20 ora_22150.aud
(dba8i)/tpora/gr1/audit>pg ora_21456.aud
<RET>
<RET>
(dba8i)/tpora/gr1/audit>exit
```

```
SQL> select username,osuser,terminal from v$session
```

USERNAME	OSUSER	TERMINAL
	dba8i	UNKNOWN
	dba8i	UNKNOWN
	dba8i	UNKNOWN
	dba8i	UNKNOWN
	dba8i	UNKNOWN
	dba8i	UNKNOWN
SYSTEM	dba8i	pts/1

```
7 rows selected.
```



```
SQL> select username,osuser,sid,serial#,terminal from v$session
       where type <>'BACKGROUND';
```

USERNAME	OSUSER	SID	SERIAL#	TERMINAL
SYSTEM	dba8i	7	318	pts/1
SCOTT	dba8i	8	221	pts/3

```
SQL> alter system kill session '8,221';
System altered.
```

Pour les transactions et verrous sur une fenêtre terminal ou sqlplus un utilisateur modifie une table et ne fait pas 'commit'.

Sur un autre terminal connecté en mode dba.

```
SQL> @/tpora/2898e/exch07-8
```

NOM_UTIL	TERMINAL	SPID	NOM_RBS	USED_UREC	USED_UBLK
SCOTT	pts/1	27304	R02	1	1

```
SQL> 1
1 select substr(s.username,1,15) nom_util,substr(s.terminal,1,8) terminal,
2      p.spid,substr(r.name,1,8) nom_rbs,t.used_urec, t.used_ublk
3 from v$session s,v$process p,v$rollname r,v$transaction t
4 where s.paddr=p.addr
5      and s.saddr=t.ses_addr
6*      and r.usn=t.xidusn
```

```
SQL> @/tpora/2898e/exch07-9
```

USERNAME	SERIAL#	SID	OBJECT_NAME	OWNER	TY	LMODE
SCOTT	605	7	EMP	SCOTT	TM	3

```
SQL> 1
1 select s.username,s.serial#,s.sid,substr(o.object_name,1,15) object_name,
2      substr(o.owner,1,10) owner,l.type,l.lmode
3 from v$session s,v$lock l,all_objects o
4 where l.sid=s.sid
5      and o.object_id=l.id1
6*      and username is not null
```

CHAPITRE 8

```
(dba8i)/tpora/grl>exp scott/tiger file=exp_scott owner=scott
```

```
Export: Release 8.1.7.0.0 - Production on Thu Jan 24 14:29:15 2002
```

```
(c) Copyright 2000 Oracle Corporation. All rights reserved.
```

```
Connected to: Oracle8i Enterprise Edition Release 8.1.7.0.0 - Production
With the Partitioning option
JServer Release 8.1.7.0.0 - Production
Export done in US7ASCII character set and WE8ISO8859P1 NCHAR character set
server uses WE8ISO8859P1 character set (possible charset conversion)
. exporting pre-schema procedural objects and actions
. exporting foreign function library names for user SCOTT
. exporting object type definitions for user SCOTT
About to export SCOTT's objects ...
. exporting database links
. exporting sequence numbers
. exporting cluster definitions
. about to export SCOTT's tables via Conventional Path ...
. . exporting table BONUS 0 rows exported
. . exporting table DEPT 4 rows exported
. . exporting table DUMMY 1 rows exported
. . exporting table EMP 14 rows exported
. . exporting table SALGRADE 5 rows exported
. exporting statistics
Export terminated successfully without warnings.
```

```
.../...
```

```
(dba8i)/tpora/grl>sqlplus scott/tiger
```

```
SQL> drop table emp;
```

```
Table dropped.
```

```
SQL>exit
```

```
(dba8i)/tpora/grl>imp scott/tiger file=exp_scott tables=emp
```

```
Import: Release 8.1.7.0.0 - Production on Thu Jan 24 14:37:46 2002
```

```
(c) Copyright 2000 Oracle Corporation. All rights reserved.
```

```
Connected to: Oracle8i Enterprise Edition Release 8.1.7.0.0 - Production
With the Partitioning option.
JServer Release 8.1.7.0.0 - Production
```

```
Export file created by EXPORT:V08.01.07 via conventional path
import done in US7ASCII character set and WE8ISO8859P1 NCHAR character set
import server uses WE8ISO8859P1 character set (possible charset conversion)
. importing SCOTT's objects into SCOTT
. . importing table "EMP" 14 rows imported
Import terminated successfully without warnings.
```

```
(dba8i)/tpora/grl>sqlplus scott/tiger
```

```
SQL> select * from emp;
```

CHAPITRE 9

Mise en place de l'archivage.

```
(dba8i)/tpora/gr1>vi initA817.ora
```

Chercher les lignes suivantes et enlever le commentaire de début (#)

```
log_archive_start = true
log_archive_dest = /tpora/gr1/arch
log_archive_format = "t%ts%s.arc"
```

```
(dba8i)/tpora/gr1>svrmgrl
```

Oracle Server Manager Release 3.1.7.0.0 - Production

Copyright (c) 1997, 1999, Oracle Corporation. All Rights Reserved.

Oracle8i Enterprise Edition Release 8.1.7.0.0 - Production
With the Partitioning option
JServer Release 8.1.7.0.0 - Production

```
SVRMGR> connect internal
Connected.
```

```
SVRMGR> shutdown
Database closed.
```

```
Database dismounted.
```

```
ORACLE instance shut down.
```

```
SVRMGR> startup mount
```

```
ORACLE instance started.
```

```
Total System Global Area
```

```
26943380 bytes
```

```
Fixed Size
```

```
73620 bytes
```

```
Variable Size
```

```
25972736 bytes
```

```
Database Buffers
```

```
819200 bytes
```

```
Redo Buffers
```

```
77824 bytes
```

```
Database mounted.
```

```
SVRMGR> alter database archivelog;
```

```
Statement processed.
```

```
SVRMGR> alter database open;
```

```
Statement processed.
```

```
SVRMGR> alter system switch logfile;
```

```
Statement processed.
```

```
SVRMGR> shutdown immediate
```

```
Database closed.
```

```
Database dismounted.
```

```
ORACLE instance shut down.
```

```
SVRMGR> exit
```

```
Server Manager complete.
```

```
(dba8i)/tpora/gr1>ls -l arch
```

```
total 1128
```

```
-rw-r----- 1 dba8i dba 574976 24 jan 16:02 t1s211.arc
```

```
(dba8i)/tpora/gr1>cp *.dbf initA817.ora ctl* sauve
```

Redémarrage de la base et travail

```
(dba8i)/tpora/gr1>sqlplus /nolog
```

SQL*Plus: Release 8.1.7.0.0 - Production on Thu Jan 24 16:10:01 2002

(c) Copyright 2000 Oracle Corporation. All rights reserved.

```
SQL> connect / as sysdba
```

```
Connected to an idle instance.
```

```
SQL> startup
```

```
ORACLE instance started.
```

```
Total System Global Area 26943380 bytes
```

```
Fixed Size 73620 bytes
```

```
Variable Size 25972736 bytes
```

```
Database Buffers 819200 bytes
```

```
Redo Buffers 77824 bytes
```

```
Database mounted.
```

```
Database opened.
```

```
SQL> connect scott/tiger
```

```
Connected.
```

```
SQL> create table emp2 as select * from emp;
```

```
Table created.
```

```
SQL> update emp set sal=2000;
```

```
14 rows updated.
```

```
SQL> connect system/manager
```

```
Connected.
```

```
SQL> alter system switch logfile;
```

```
System altered.
```

```
SQL> /
```

```
System altered.
```

```
SQL> /
```

```
System altered.
```

```
SQL> connect / as sysdba
```

```
Connected.
```

```
SQL> shutdown immediate;
```

```
Database closed.
```

```
Database dismounted.
```

```
ORACLE instance shut down.
```

```
SQL> exit
```

Disconnected from Oracle8i Enterprise Edition Release 8.1.7.0.0 - Production

With the Partitioning option

JServer Release 8.1.7.0.0 - Production

```
(dba8i)/tpora/gr1>rm usersf1_A817.dbf (Simulation de la destruction d'un
```

```
fichier de données)
```

```
(dba8i)/tpora/gr1>cp sauve/usersf1_A817.dbf . (Restauration physique)
```

```
(dbas1)/tpora/gr1>sqlplus /nolog
```

```
SQL*Plus: Release 8.1.7.0.0 - Production on Thu Jan 24 16:17:05 2002
```

```
(c) Copyright 2000 Oracle Corporation. All rights reserved.
```

```
SQL> connect / as sysdba
```

```
Connected to an idle instance.
```

```
SQL> startup
```

```
ORACLE instance started.
```

```
Total System Global Area 26943380 bytes
```

```
Fixed Size 73620 bytes
```

```
Variable Size 25972736 bytes
```

```
Database Buffers 819200 bytes
```

```
Redo Buffers 77824 bytes
```

```
Database mounted.
```

```
ORA-01113: file 5 needs media recovery
```

```
ORA-01110: data file 5: '/tpora/gr1/usersf1_A817.dbf'
```

```
SQL> recover database
```

```
ORA-00279: change 39513 generated at 01/24/2002 16:02:24 needed for thread 1
```

```
ORA-00289: suggestion : /tpora/gr1/arch/tls212.arc
```

```
ORA-00280: change 39513 for thread 1 is in sequence #212
```

```
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
```

```
Log applied.
```

```
Media recovery complete.
```

```
SQL> alter database open;
```

```
Database altered.
```

```
SQL> connect scott/tiger
```

```
Connected.
```

```
SQL> select * from emp;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	2000		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	2000	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	2000	500	30

```
.../...
```

```
14 rows selected.
```

```
SQL> select * from emp2;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30

```
.../...
```

```
14 rows selected.
```

