Administration de bases de données Oracle

Cycle Ingénieur Applications Web et Mobiles -Semestre 08

Ecole d'Ingénierie Digitale et d'Intelligence Artificielle
Université Euro-Méditerranéenne de Fès

Pr. Abderrahim El Qadi
Département Mathématique Appliquée et Génie Informatique
ENSAM, Université Mohammed V de Rabat

A.U. 2023/2024

Partie 2 :

7. Audit d'une base Oracle

- Types d'audits
- Autorisation de Mise en route de l'audit
- Déclenchement effectif de l'audit
- Désactivation de l'audit sur les Ordres SQL et les privilèges systèmes

8. Gestion des fichiers de contrôle

- Informations sur les fichiers de contrôle
- Sauvegarde du fichier de contrôle
- Multiplexage des fichiers de contrôle
- Création d'un fichier de contrôle

9. Gestion des fichiers de journalisation et Archivage

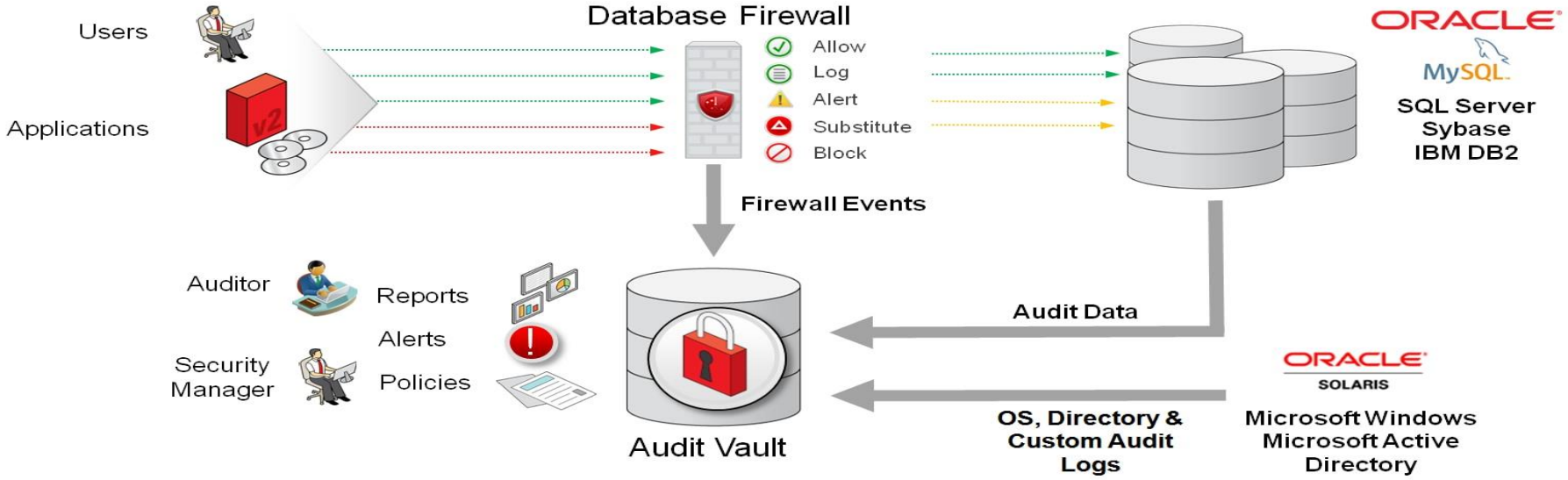
- Contenu du fichier redo
- Création des groupes et des membres redo
- Forcer les Logs Switchs
- Archivage des redo logs files

Références

- Oracle 10g Administration, Olivier Heurtel, 2005, 489p.
- Les bases de données Oracle8i, Développement, Administration et Optimisation, Roger Chapuis, Dunod, 2001, 382pp.
- Administration Oracle 10G, Partie I, G. Mopolo-Moké, MBDS / UNSA NICE 2005/ 2006
- Oracle11g Administration, Razvan Bizoï, Eyrolles, 2011, ISBN : 978-2-212-12898-7
- Oracle11g Sauvegarde et sécurité, Razvan Bizoï, Eyrolles, 2011, ISBN : 978-2-212-12899-4
- Oracle 11g New Features for Administrators Summary Sheets, Version 1.2 Editor: Ahmed Baraka

7. L'audit

- L'audit (Auditing) est un travail nécessaire et important de DBA.
- Il est toujours mis en œuvre en premier lieu dans toute initiative de sécurité d'**Oracle**.
- L'audit Oracle permet de surveiller l'activité de la base de données pour :
  - Contrôler les accès à la base, à des fins de sécurité,
  - Vérifier que tel ou tel objet est accédé en lecture ou en écriture,
  - Vérifier les tentatives d'accès infructueuses à des objets,
  - Contrôler l'audit éventuellement pirate !



- Les résultats sont stockés dans une table du dictionnaire : SYS.AUD$
- **Type de résultat fourni :**
  - no session , nom du User , no/nom du Terminal , nom de l'objet accédé ,
  - type d'ordre SQL ou de commande , date d'occurrence…

Exemple:

SQL>SELECT SESSIONID, USERID, TERMINAL, ACTION# FROM **SYS.AUD$;**

```
SESSIONID USERID TERMINAL            ACTION#
------------------------------------------------------------
623050        USER1        DESKTOP-FPVVLB0      100
SQL> select * from AUDIT_ACTIONS;
  ACTION                    NAME
---------            ----------------------------
         0                 UNKNOWN
         1               CREATE TABLE
         2                  INSERT
         3                  SELECT
        ….
```

7.1 Autorisation de Mise en route de l'audit

– Il faut positionner le paramètre de démarrage AUDIT_TRAIL dans le fichier INIT.ORA de la base : 3 valeurs possibles :
  - NONE : invalide l'audit (valeur par défault)
  - DB : valide l'audit et stocke les résultats dans la table d'audit
  - OS : valide l'audit et stocke les résultats dans un fichier externe (un autre paramètre : AUDIT_FILE_DEST précise le répertoire de destination…)

```
SQL>show parameter audit_file_dest;
NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
audit_file_dest                      string      D:\APP\ADMIN\ORCL19\ADUMP
```

– Pour activer (ou désactiver) la piste d'audit, on modifie le paramètre AUDIT_TRAIL qui n'est pas dynamique :

```
SQL> ALTER SYSTEM SET AUDIT_TRAIL='DB' SCOPE=SPFILE;
SQL> SHUTDOWN IMMEDIATE;
SQL> STARTUP
```

7.2 Déclenchement effectif de l'audit (ciblé) par le DBA ou un user autorisé

- Il existe 4 niveaux d'audit :
  - Connexion / déconnexion : surveille les connexions
  - Ordre SQL : audit par type d'ordre SQL utilisé
  - Privilege: audit d'un privilege SYSTEM (SELECT ANY, DROP ANY, CREATE ANY, * ANY…)
  - Objet : un ordre SQL particulier sur un objet particulier.

- A chaque niveau d'audit, on peut de + surveiller aussi bien les **SUCCES** que les **ECHECS**, et avoir une entrée d'audit par commande utilisateur ou globalement pour la session.

- Le déclenchement de l'audit effectif se fait par **la commande AUDIT et une ou des option(s)**, qui précise(nt) :
  - Le type d'action à auditer,
  - Si l'on veut les tentatives réussies ou échouées,
  - Pour une session globale ou pour chaque ordre SQL

```
AUDIT { system_privilege | object_privilege [, ...] | ALL }
   [ BY { SESSION | ACCESS } ]
   [ WHENEVER [ NOT ] SUCCESSFUL ]
   [ BY session_user ]
   [ { { ON { table | view | synonym | DATABASE } }
      [ WHENEVER [ NOT ] SUCCESSFUL ]
      [ BY session_user ]
    | ON { DIRECTORY | LIBRARY }
      [ WHENEVER [ NOT ] SUCCESSFUL ]
    }
  ];
```

Exemples :

1. Audit connexion

- Surveille toutes les tentatives de connexions infructueuses

```
SQL>AUDIT SESSION WHENEVER NOT SUCCESSFUL;
SQL>CONNECT user1/user1
ERROR:
ORA-01017 : nom utilisateur/mot de passe non valide ; connexion refusé
SQL>SELECT CURRENT_USER, TERMINAL FROM SYS.AUD$;
SQL> connect / as sysdba
SQL> SELECT CURRENT_USER, TERMINAL FROM SYS.AUD$;
CURRENT_USER      TERMINAL
------------------------      -------------------
USER1                DESKTOP-FPVVLB0

SQL>SELECT OS_USERNAME, USERNAME, TERMINAL,
TO_CHAR(TIMESTAMP,'MM-DD-YYYY HH24:MI:SS') “Time”  FROM
DBA_AUDIT_TRAIL;
OS_USERNAME              USERNAME   TERMINAL                TIME
-------------------              ----------      --------------------          --------------------
DESKTOP-FPVVLB0\ADmiN     USER1      DESKTOP-FPVVLB0      04-16-2024 16:40:23
```

2. Audit Ordre SQL

- Surveille les select, insert, et delete infructueux sur n'importe quelle table, une entrée seulement par session

```
SQL> AUDIT SELECT TABLE, INSERT TABLE, DELETE TABLE
  BY ACCESS WHENEVER NOT SUCCESSFUL;
```

3. Audit objet

- Surveille les insertion (réussies ou échouées) sur la table DEPARTMENTS de HR :

```
SQL> AUDIT INSERT ON HR.DEPARTMENTS ;
```

- Auditer d'un tableau de HR.EMPLOYEES par **accès**.

```
SQL> AUDIT ALL ON HR.DEPARTMENTS BY ACCESS;
SQL> connect hr/hr
SQL> update hr.employees set salary=salary*0.95 where job_id='AC_MGR';
SQL> connect / as sysdba
SQL>select username, owner, obj_name, action_name, sql_text from dba_audit_object;
USERNAME  OWNER OBJ_NAME   ACTION_NAME SQL_TEXT
--------- ----- ---------- ----------- ------------------------------------
HR                      HR    EMPLOYEES  UPDATE
```

4. Audit privilège

AUDIT <privilege_list>|ALL PRIVILEGES
[BY <username>|<proxyuser>]
[ON BEHALF OF <userlist>|ANY] Useful only if the Proxyuser is used.
[BY SESSION|ACCESS]
[WHENVER [NOT] SUCCESSFUL]

```
SQL> AUDIT CREATE TABLE BY ACCESS WHENEVER SUCCESSFUL;
SQL> AUDIT CREATE SESSION BY USER1 WHENEVER NOT SUCCESSFUL;
```

7.3 Vérification des options d'audit en cours

- Il suffit d'aller consulter les vues adéquates du dictionnaire de données (%AUDIT_OPTS) :

ALL_DEF_AUDIT_OPTS, DBA_STMT_AUDIT_OPTS, DBA_PRIV_AUDIT_OPTS, DBA_OBJ_AUDIT_OPTS, USER_OBJ_AUDIT_OPTS,

DBA_AUDIT_EXISTS, DBA_AUDIT_MGMT_CLEANUP_JOBS, DBA_AUDIT_MGMT_CLEAN_EVENTS, DBA_AUDIT_MGMT_CONFIG_PARAMS, DBA_AUDIT_MGMT_LAST_ARCH_TS, DBA_AUDIT_OBJECT, DBA_AUDIT_POLICIES, DBA_AUDIT_POLICY_COLUMNS, DBA_AUDIT_SESSION, DBA_AUDIT_STATEMENT, DBA_AUDIT_TRAIL, …

Sélection des résultats dans les vues d'audit

Faire un SELECT sur %_AUDIT_OBJECT ou %_AUDIT_SESSION ou %_AUDIT_STATEMENT

Exemples:

```
SQL>SELECT * FROM SYS.DBA_OBJ_AUDIT_OPTS
WHERE owner = 'HR' AND object_name LIKE 'EMPLOYEES%';
```

OWNER	OBJECT_NAME	OBJECT_TYPE	ALT	AUD	COM	DEL	GRA	IND	INS	LOC	REN	SEL	UPD	EXE	CRE	REA	WRI	FBK
HR	EMPLOYEES	TABLE	A/A	A/A	A/A	A/A	A/A	A/A	A/A	A/A	A/A	A/A	A/A	-/-	A/A	-/-	-/-	A/A

'-' : pas d'audit
'S' : par session
'A' : un par accès
à gauche du '/' : Successful
à droite du '/' : Not successful

- **Visualisation des options d'audit des ordres SQL actives** :

```
SQL>SELECT * FROM dba_stmt_audit_opts ;
AUDIT_OPTION                SUCCESS    FAILURE
------------------------------------------------------------------------------
CREATE SESSION              NOT SET    BY ACCESS
SELECT TABLE                NOT SET    BY ACCESS
INSERT TABLE                NOT SET    BY ACCESS
DELETE TABLE                NOT SET    BY ACCESS
EXECUTE PROCEDURE           NOT SET    BY ACCESS
```

- **Visualisation des options actives d'audit des privilèges** :

```
SQL>SELECT * FROM dba_priv_audit_opts ;
```

7.4 Désactivation de l'audit sur les Ordres SQL et les privilèges systèmes

Syntaxe:

```
NOAUDIT { statement_opt | system_priv }
[, { statement_opt | stystem_priv } ] ...
[ BY user [, user ] ...
[ WHENEVER [ NOT ] SUCCESSFUL ]
```

Mots clés et paramètres:

statement_opt : option des ordres sql à ne plus auditer

system_priv : privilège système à ne plus auditer

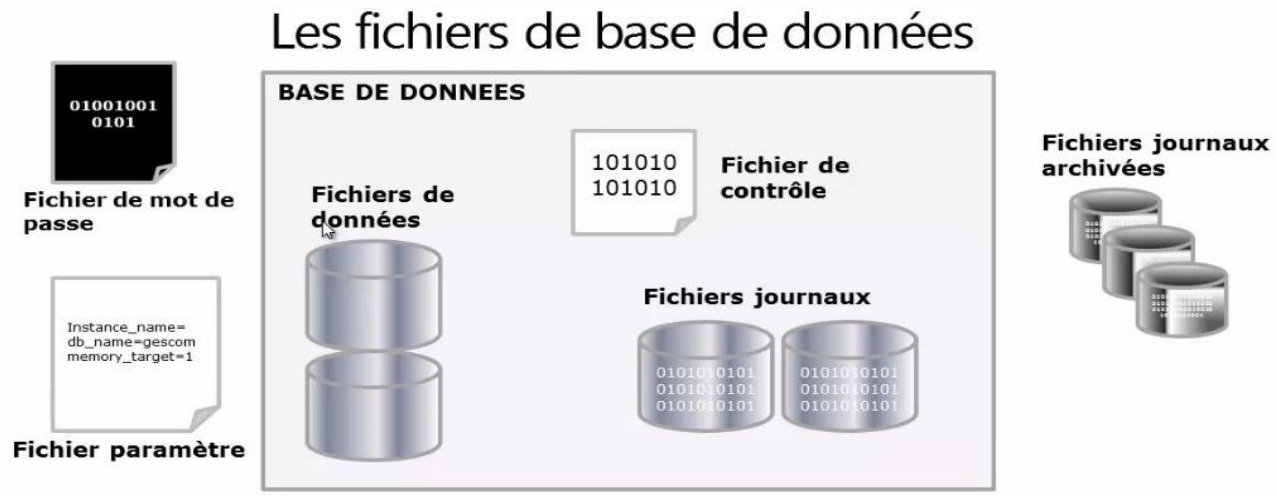
user : désactiver l'audit des ordres SQL sur un user

SUCCESSFUL : désactiver seulement en cas de succès

Exemples :

SQL>NOAUDIT session ;
SQL>NOAUDIT role;
SQL>NOAUDIT session BY hr;
SQL>NOAUDIT select any table, insert any table, delete any table, execute any procedure;
SQL>NOAUDIT select table, insert table, delete table;
SQL>NOAUDIT ALL;
SQL>NOAUDIT ALL PRIVILEGES;

8. Fichier de Contrôle : Administration



- Le fichier de contrôle est un fichier binaire ; créé pendant la création de la base et modifié en permanence.
- Ce fichier doit être toujours disponible car il est consulté ; et modifié fréquemment par le serveur oracle.
- Il est indispensable pour la restauration de la base.
- Perdre tous les fichiers de contrôle rend difficile la restauration de la base.

- Les informations contenues dans un fichier de contrôle sont :

  - Le nom de la base qui est pris avec le paramètre d'initialisation DB_NAME ou le nom utilisé dans la commande CREATE DATABASE.
  - L'identifiant de la base de données lorsque la base de données est créée.
  - Le timestamp de la création de la base de données.
  - Les noms et localisations des fichiers de données et des fichiers de redo log sont mis à jour dans un fichier de contrôle lorsqu'un fichier de données ou un fichier de redo log est ajouté ou supprimé.
  - Les informations sur les tablespaces sont mises à jour lorsqu'un tablespace est supprimé ou ajouté.
  - L'historique des switches de redo log.
  - Les localisations et statuts des logs archivés sont enregistrés lorsque l'archivage est activé.
  - Les localisations et statuts des backups sont enregistrés par l'utilitaire Recovery Manager.
  - Le numéro de séquence de log courant est enregistré lorsque des switchs de log se produisent.
  - Les informations de checkpoint sont enregistrées lorsque le checkpoint se produit.

-

8.1 Informations sur les fichiers de contrôle

- Les vues suivantes affichent des informations sur les fichiers de contrôle :

V$DATABASE, V$CONTROLFILE,

V$CONTROLFILE_RECORD_SECTION, V$PARAMETER, $BACKUP,

V$DATAFILE, V$TEMPFILE, V$TABLESPACE, V$ARCHIVE, V$LOG,

V$LOGFILE, V$LOGHIST, V$ARCHIVED_LOG

- Pour afficher les noms des fichiers de contrôle on utilise :

```
SQL>SHOW PARAMETER CONTROL_FILES

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
control_files                        string      D:\APP\ORADATA\ORCL19\CONTROL01.CTL,
                                                 D:\APP\ORADATA\ORCL19\CONTROL02.CTL
```

```
SQL>SELECT VALUE FROM V$PARAMETER WHERE NAME='control_files';

VALUE
--------------------------------------------------------------------------------
D:\APP\ORADATA\ORCL19\CONTROL01.CTL,
D:\APP\ORADATA\ORCL19\CONTROL02.CTL
```

- La vue v$controlfile_record_section fournit quant à elle des informations sur les sections dans les fichiers de contrôle.

```
select type, record_size, records_total, records_used
from v$controlfile_record_sectionTYPE;
RECORD_SIZE RECORDS_TOTAL RECORDS_USED
----        ----------- ------------ ------------
DATABASE          192          1            1
CKPT PROGRESS     4084         1            0
REDO THREAD       104          1            1
REDO LOG          72           32           3
DATAFILE          180          254          10
```

8.2.1 Sauvegarde du fichier de contrôle

- Oracle recommande de sauvegarder le fichier de contrôle à chaque modification de la structure de la base comme le fait d'ajouter, renommer ou supprimer un fichier de données ou un fichier journal.

  • Crée un fichier en format texte dans le répertoire USER_DUMP_DEST, qu'on peut modifier pour reconstruire un nouveau fichier de contrôle

```
SQL> show parameter USER_DUMP_DEST
NAME                 TYPE        VALUE
-------------------------------- ----------- ------------------------------
user_dump_dest tring      D:\MYDATA\LOGICIEL\ORACLE19C\RDBMS\TRACE
```

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

Cette commande génère un script SQL de création des fichiers de contrôle dans un fichier de trace créé dans le répertoire de trace udump

8.2.2 Multiplexage des fichiers de contrôle

- Pour multiplexer les fichiers de contrôle, il suffit de copier un des fichiers dans un autre emplacement et de l'indiquer dans le paramètre CONTROL_FILES du fichier d'initialisation initSID.ora

```
1. Arrêter la base : SHUTDOWN IMMEDIATE
2. Modifier le paramètre CONTROL_FILES dans le fichier initSID.ora
CONTROL_FILES = ('...\oradata\ORCL\control01.ctl',
                 '...\oradata\ORCL\conrol02.ctl',
                 '...\oradata\ORCL\control03.ctl')
3. Copier le fichier de contrôle en utilisant les commandes OS
    C:>copy ...\ORCL\control01.ctl ...\ORCL\control03.ctl
4. Démarrer la base : STARTUP
```

- Oracle met à jours les fichiers de contrôle en même temps, mais seul le premier fichier cité dans le paramètre CONTROL_FILES est consulté.

8.2.3 Multiplexage des fichiers de contrôle en utilisant le spfile

1. Ajouter le nouveau fichier de contrôle dans le paramètre CONTROL_FILES en utilisant

```
ALTER SYSTEM SET CONTROL_FILES=('...\oradata\ORCL\control01.ctl',
  '...\oradata\ORCL\conrol02.ctl',
 '...\oradata\ORCL\control03.ctl')
SCOPE=SPFILE;
```

2. Arrêter la base : SHUTDOWN IMMEDIATE
3. Copier le fichier de contrôle en utilisant les commandes OS.
   C :> copy ...\ORCL\control01.ctl ...\ORCL\control03.ctl
4. Démarrer la base : STARTUP

8.2 Création d'un fichier de contrôle

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

- Etapes de création :
  - Lister tous les fichiers de données et de journaux en ligne
  - Arrêter la base : SHUTDOWN IMMEDIATE
  - Sauvegarder tous les fichiers de données et journaux
  - Démarrer la base en mode NOMOUNT : STARTUP NOMOUNT
  - Créer le nouveau fichier de contrôle en utilisant CREATE CONTROLFILE.

```
CREATE CONTROLFILE
SET DATABASE mabase
LOGFILE GROUP 1    ('...\mabase\redo01_01.log',
          '...\mabase\redo01_02.log'),
  GROUP 2  ('...\mabase\redo02_01.log', '...\mabase\redo02_02.log'),
NORESETLOGS
DATAFILE  '...\mabase\system01.dbf' SIZE 3M,
                '...\mabase\rbs01.dbs' SIZE 5M,
                '...\mabase\users01.dbs' SIZE 5M,
                '...\mabase\temp01.dbs' SIZE 5M
MAXLOGFILES 50
MAXLOGMEMBERS 3
MAXLOGHISTORY 400
MAXDATAFILES 300
MAXINSTANCES 6
ARCHIVELOG ;
```

- L'option RESETLOGS est utilisée lorsqu'on perdre l'un des groupes de fichiers journaux ou lorsqu'on renomme la base.
- Si le nouveau fichier de contrôle est créé avec l'option NORESETLOGS on peut restaurer la base complètement,
- et s'il est créé avec l'option RESETLOGS il faut utiliser USING BACKUP CONTROL FILE et ouvrir la base avec ALTER DATABASE OPEN RESETLOGS ;
- Si le fichier de contrôle existe déjà, il faut utiliser CONTROLFILE REUSE au lieu de CONTROLFILE SET dans la clause CREATE DATABASE, et si la taille de l'ancien fichier est différente du nouveau on ne peut pas utiliser l'option REUSE.

- Si le paramètre REMOTE_LOGIN_PASSWORDFILE =EXCLUSIVE, oracle signale une erreur. Il faut donc affecter à ce paramètre la valeur SHARED ou créer un nouveau fichier de mot de passe.
- Il faut faire très attention, le fait d'oublier de mentionner un fichier de données pendant la création du fichier de contrôle peut causer la perte de ce fichier et même la perte de la base entière. Il faut encore faire attention si on utilise le mode FORCE LOGGING.

8.3 Suppression d'un fichier de contrôle

1. Arrêter la base normalement : SHUTDOWN IMMEDIATE
2. Modifier le paramètre CONTROL_FILES dans le fichier initSID.ora
   control_files = (...\mabase\control01.ctl, ...\mabase\control02.ctl)
3. Supprimer le fichier de contrôle : Exemple control03.ctl
4. Démarrer la base : STARTUP

8.4 Restauration d'un fichier de contrôle corrompu à partir d'un autre fichier valide

1. Supposons que le fichier control03.ctl est corrompu.
2. Arrêter la base normalement : SHUTDOWN IMMEDIATE
3. Copier le fichier de contrôle valide en écrasant le fichier corrompu :

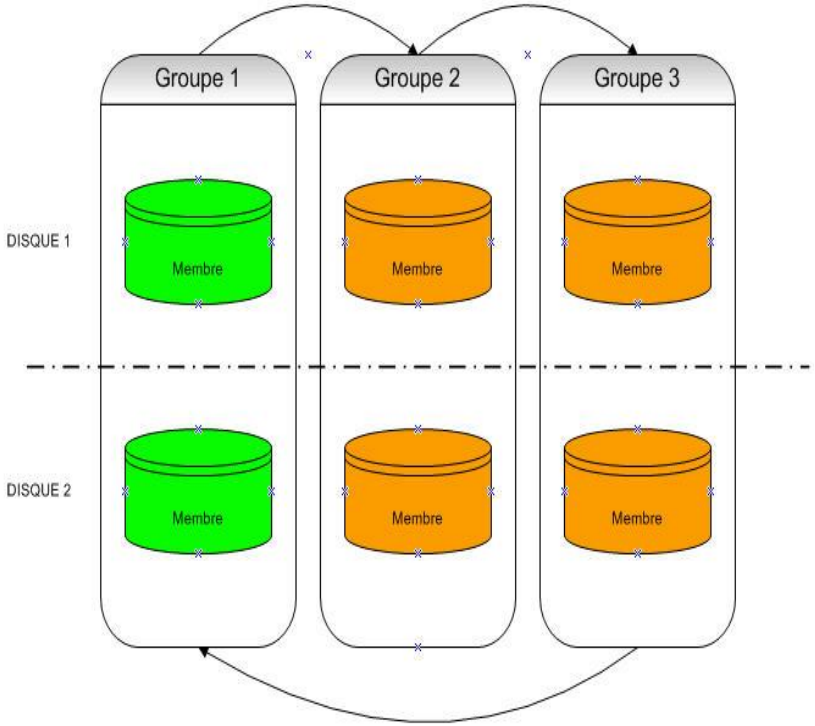
   copy ...\ORCL\control01.ctl ...\ORCL\control03.ctl

4. Démarrer la base : STARTUP

9. Les Redo logs: Administration

- Les fichiers redo sont essentiels pour les processus de restauration.

  - Les fichiers redo doivent être placés sur des axes différents et des disques très rapides.
  - Oracle dispose les journaux redo en groupes.
  - Chaque groupe a au moins un fichier redo. Le process LGWR écrit simultanément dans les 2 fichiers du groupe courant.
  - On doit avoir au minimum deux groupes distincts de fichiers redo (aussi appelés redo threads), chacun contient au minimum un seul membre. Car, si l'on n'a qu'un seul fichier redo, Oracle écrasera ce fichier redo et on perdra toutes les transactions.



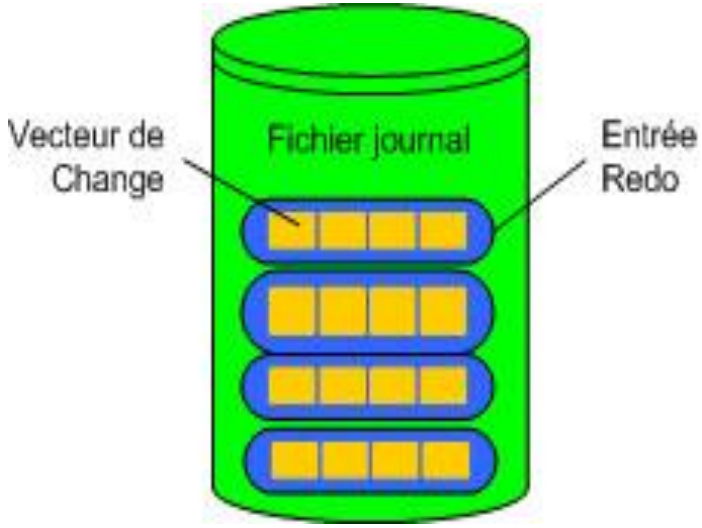
Groupe de fichiers de journalisations et les membres

9.1 Contenu du fichier redo

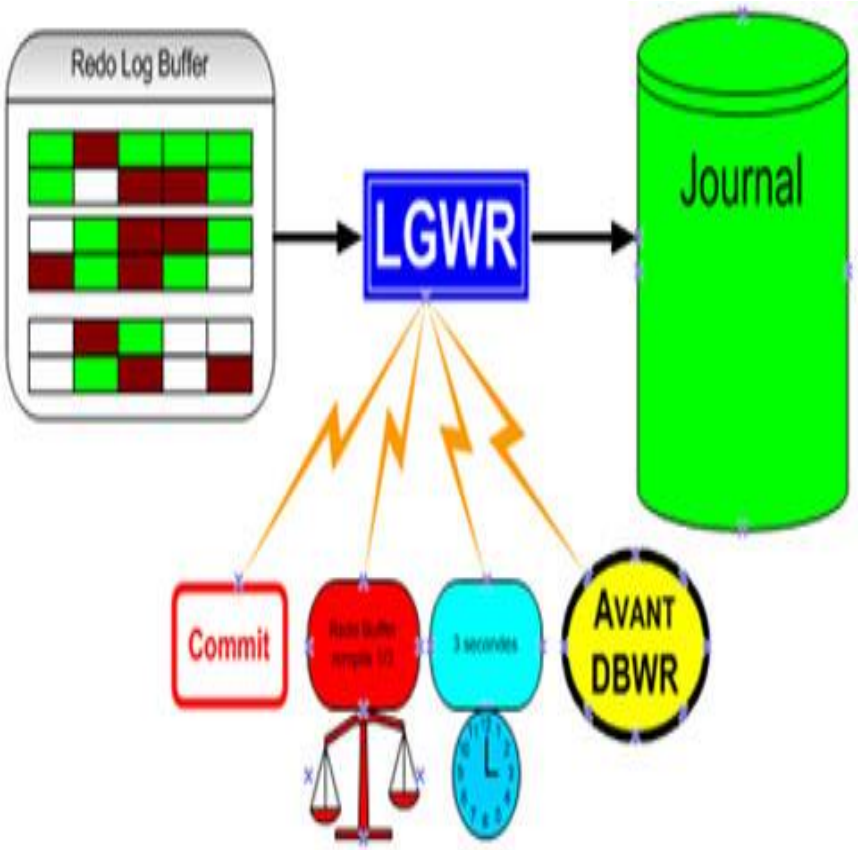
- Un fichier redo est essentiellement une séquence d'enregistrements de modifications (ou "redologs") qui enregistrent toutes les modifications apportées aux données d'une base de données Oracle.

Voici un exemple très simplifié de ce à quoi ressemblerait le contenu de ce fichier

```
-- Redo Group Header --
Groupe de Redo: 1
Numéro de Séquence: 12345
...
-- Redo Records --
INSERT INTO table1 VALUES (1, 'Valeur1', ...);
UPDATE table2 SET column1 = 'NouvelleValeur'
WHERE condition = 'something';
DELETE FROM table3 WHERE condition =
'another_condition';
...
```



- Les enregistrements redo enregistrent les données que l'on peut utiliser après pour reconstruire toutes les changements effectués sur la base, segments undo inclus.
- De plus, le fichier redo protège aussi les données d'annulation. Quand on restaure la base en utilisant les données redo, la base lit les vecteurs de changements dans les enregistrements redo et applique le changement aux blocs appropriés.
- Les enregistrements redo sont mis d'une façon circulaire dans le buffer redo log de la mémoire SGA. Et ils sont écrits dans un seul fichier redo par le processus LGWR.



9.2 Informations sur les fichiers redo

- Les vues V$THREAD, V$LOG, V$LOGFILE et V$LOG_HISTORY fournissent des informations sur les fichiers Redo.
  - La vue V$THREAD donne les informations sur le fichier redo en cours.
  - La vue V$LOG donne les informations en lisant dans le fichier de contrôle au lieu de lire dans le dictionnaire de données

9.3 Création des groupes et des membres redo

9.3.1. Création des groupes de redo

Exemple :

```
SQL> ALTER DATABASE
        ADD LOGFILE ('...\oradata\mabase\log1c.rdo',
        '...\oradata\mabase\log2c.rdo') SIZE 500K;
```

- On peut spécifier le numéro qui identifie le groupe en utilisant la clause GROUP :

```
SQL> ALTER DATABASE
ADD LOGFILE GROUP 5 ('...\oradata\mabase\log1c.rdo',
 '...\oradata\mabase\log2c.rdo') SIZE 500K;
```

- Le numéro de groupe doit être entre 1 et MAXLOGFILES. Surtout ne pas sauter les numéros de groupes (par exemple 10, 20,30), sinon de l'espace dans les fichiers de contrôle sera consommé inutilement.

9.3.2. Création des membres de fichiers redo

- La base peut avoir au maximum MAXLOGMEMBERS membres.

Exemple : on ajoute un nouveau membre au groupe de redo numéro 2 :

```
SQL> ALTER DATABASE ADD LOGFILE MEMBER
'...\oradata\mabase\log2b.rdo' TO GROUP 2;
```

- Le nom du fichier doit être indiqué, mais sa taille n'est pas obligatoire.
- La taille du nouveau membre est déterminée à partir de la taille des membres existants du groupe.

9.3.3. Remplacement et renomination des membres de fichiers redo

- Avant de déplacer les fichiers redo, ou tous autres changements de structures de la base, sauvegarder complètement la base.
- Par précaution, après la renommination ou le déplacement d'un ensemble de fichiers redo, effectuer immédiatement une sauvegarde du fichier de contrôle.

- Pour déplacer les fichiers redo, on utilise les méthodes suivantes :
  - Les fichiers redo sont situés dans deux disques : disque1 et disque2.
  - Les fichiers redo sont dupliqués : un groupe est constitué des membres \disque1\logs\log11.rdo et \disque2\logs\log12.rdo, et le second groupe est constitué des membres \disque1\logs\log21.rdo et \disque2\logs\log22.rdo.
  - Les fichiers redo situés dans le disque disque1 doivent être déplacés dans le disque3.
  - Le nouveau nom du fichier reflète le nouvel emplacement : \disque3\logs\log13.rdo et \disque3\logs\log23.rdo.

- Les étapes à suivre pour renommer les membres des fichiers redo :

1. Arrêter la base : SHUTDOWN IMMEDIAT
2. Copier les fichiers redo dans le nouveau emplacement :

   \disque1\logs/log11.rdo --> \disque3\logs\log13.rdo

   \disque1\logs\log21.rdo --> \disque3\logs\log23.rdo

3. Démarrer la base avec un MOUNT: STARTUP MOUNT
4. Renommer le membre du fichier redo.

```
ALTER DATABASE
RENAME FILE '\disque1\logs\log11.rdo', '\disque1\logs\log21.rdo'
TO '\disque3\logs\log13.rdo', '\disque3\logs\log23.rdo'
```

5. Ouvrir la base normalement : ALTER DATABASE OPEN

La modification du fichier redo prend effet à l'ouverture de la base.

9.3.4. Suppression du groupe de fichiers redo

- On supprime un groupe en entier lorsqu'on veut réduire le nombre de groupes.
- On doit supprimer un ou plusieurs membres. Par exemple, si certains membres se trouvent dans un disque défaillant.

9.3.5. Suppression d'un groupe

- Avant de supprimer un groupe de fichiers redo, il faut prendre en considération les restrictions et les précautions suivantes :
  - Une instance réclame au minimum deux groupes de fichiers redo, sans se soucier du nombre de membres dans le groupe. (Un groupe contient un ou plusieurs membres.)
  - On peut supprimer un groupe de fichiers redo, seulement s'il est inactif. Si on a besoin de supprimer le groupe courant, en premier, on force un switch log.
  - S'assurer que le groupe de fichiers redo est bien archivé (si l'archivage est activé) avant de le supprimer.

- Pour voir ce qui se passe, on utilise la vue V$LOG.

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

 GROUP# ARC STATUS
 ---------- --- ----------------
 1          YES     ACTIVE
 2          NO      CURRENT
 3          YES     INACTIVE
 4          YES     INACTIVE
```

Exemple : Suppression le groupe numéro 3 :

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

9.3.6. Suppression des membres de fichiers redo

Exemple : Suppression un membre inactive d'un fichier redo

ALTER DATABASE DROP LOGFILE MEMBER '\...\log3c.rdo';

- Quand un membre d'un journal est supprimé, le fichier OS n'est pas supprimé du disque.
- Pour supprimer un membre d'un groupe actif, on doit forcer en premier le log switch.

9.4 Forcer les Logs Switchs

- Le log switch se produit quand LGWR s'arrête d'écrire dans un groupe de journaux et commence à écrire dans un autre.
- Par défaut, un log switch se produit automatiquement quand le groupe du fichier redo en cours est rempli.
- On peut forcer un log switch pour que le groupe courant soit inactif et disponible pour des opérations de maintenance sur les fichiers redo.
- La commande suivante force un log switch :

```
ALTER SYSTEM SWITCH LOGFILE ;
```

9.5 Archivage des redo logs files

- L'archivage permet de garder tout l'historique des fichiers redo logs, qui sont recopier sur le répertoire d'archivage dès qu'ils sont pleins (Switch).
- Si on n'archive pas les redo logs sont écrasés cycliquement, puisque rappelons le, ils sont utilisés de manière séquentielle et circulaire !
- Le processus d’archivage (process ARCH ou "ARCHIVER") est optionnel et permet de faire des restaurations les plus à jour possible.
- En l'absence d'archivage, on ne pourra récupérer les données que de la dernière sauvegarde.

9.5.1. Mise en place de l'archivage

- On définit le répertoire et le format des fichiers d'archivage avec les deux paramètres LOG_ARCHIVE_DEST_n (où n est un entier de 1 à 10), et LOG_ARCHIVE_FORMAT = arch%t_%s.arc

```
log_archive_start = true
log_archive_dest_1 = "location=C:\Oracle\oradata\BTEST\archive"
log_archive_format = %%ORACLE_SID%%T%TS%S.ARC
```

- Puis arrêt / redémarrage de la base.

- Il est possible d'autoriser l'archivage automatique de manière dynamique (sans arrêter la base) :

```
SQL> ALTER SYSTEM ARCHIVE LOG START;
```

9.5.2. Déclenchement du mode archivage

```
SQL> SHUTDOWN IMMEDIATE
SQL> STARTUP MOUNT
SQL> ALTER DATABASE ARCHIVELOG
SQL> ALTER DATABASE OPEN
```

- Pour forcer l'archivage du redo log courant et d'activer le redo log suivant.

```
SQL> ALTER SYSTEM SWITCH LOGFILE
ou SQL> ALTER SYSTEM CHECKPOINT;    pour archiver le redo log courant.
```

- Vérification du statut de l'archivage:

```
SQL> ARCHIVE LOG LIST
Ou  SQL>SELECT name, log_mode FROM v$database;
Ou SQL>SELECT archiver FROM v$instance;
```