

Atelier Pratique 1

Démarrage et arrêt d'une base oracle

Makhchoun Khadija

Exercice n 1 :

Mode d'arrêt	Abort	Immediate	Transactional	Normal
Permet de nouvelles connexions	Non	Non	Non	Non
Attend la fin des sessions en cours	Non	Non	Non	Oui
Attend la fin des transactions en cours	Non	Non	Oui	Oui
Applique un point de reprise et ferme les fichiers	Non	Oui	Oui	Oui

Exercice 2 :

- 1) j'ai créé une base de données oracle avec l'outil DBCA et je lui ai donné le nom : MAKHCHOUN.
- 2) j'ai lancé sql plus en tant qu'admin système et je me suis connectée à la base de données avec l'authentification système. Ensuite j'ai utilisé la commande `select name from v$database ;` pour afficher le nom de ma base de données.

```
SQL Plus
SQL*Plus: Release 19.0.0.0.0 - Production on Mer. Mai 19 16:09:25 2021
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Entrez le nom utilisateur : sys as sysdba
Entrez le mot de passe :

Connecté à :
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> connect sys as sysdba
Entrez le mot de passe :
Connecté.
SQL> select name from v$database;

NAME
-----
MAKHCHOU

SQL>
```

- 3) j'ai lancé la commande startup force

```
SQL Plus
SQL*Plus: Release 19.0.0.0.0 - Production on Mer. Mai 19 16:09:25 2021
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Entrez le nom utilisateur : sys as sysdba
Entrez le mot de passe :

Connecté à :
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> connect sys as sysdba
Entrez le mot de passe :
Connecté.
SQL> select name from v$database;

NAME
-----
MAKHCHOU

SQL> startup force
Instance ORACLE lancée.

Total System Global Area 654308168 bytes
Fixed Size 9270088 bytes
Variable Size 197132288 bytes
Database Buffers 440401920 bytes
Redo Buffers 7503872 bytes
Base de données montée.
Base de données ouverte.
SQL>
```

⇒ Cette commande permet de redémarrer la base Oracle c'est-à-dire qu'elle va Arrêter la base en mode ABORT (parce qu'on est en train de forcer le démarrage de la base) et ensuite elle va la démarrer en mode OPEN.

4) Pour fermer la base oracle on utilise la commande **alter database close immediate ;**

Cette commande permet de fermer les datafiles et les logfiles immédiatement.

```
SQL Plus
SQL*Plus: Release 19.0.0.0.0 - Production on Mer. Mai 19 16:09:25 2021
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Entrez le nom utilisateur : sys as sysdba
Entrez le mot de passe :

Connecté à :
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> connect sys as sysdba
Entrez le mot de passe :
Connecté.
SQL> select name from v$database;

NAME
-----
MAKHCHOU

SQL> startup force
Instance ORACLE lancée.

Total System Global Area 654308168 bytes
Fixed Size 9270088 bytes
Variable Size 197132288 bytes
Database Buffers 440401920 bytes
Redo Buffers 7503872 bytes
Base de données montée.
Base de données ouverte.
SQL> alter database close immediate;

Base de données modifiée.

SQL>
```

5) Pour arrêter la base oracle on utilise la commande **shutdown immediate**

Cette commande va fermer tous les fichiers de la base , elle va libérer la mémoire et arrêter les processus background.

```
SQL Plus
SQL*Plus: Release 19.0.0.0.0 - Production on Mer. Mai 19 16:09:25 2021
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Entrez le nom utilisateur : sys as sysdba
Entrez le mot de passe :

Connecté à :
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> connect sys as sysdba
Entrez le mot de passe :
Connecté.
SQL> select name from v$database;

NAME
-----
MAKHCHOU

SQL> startup force
Instance ORACLE lancée.

Total System Global Area 654308168 bytes
Fixed Size 9270088 bytes
Variable Size 197132288 bytes
Database Buffers 440401920 bytes
Redo Buffers 7503872 bytes
Base de données montée.
Base de données ouverte.
SQL> alter database close immediate;

Base de données modifiée.

SQL> shutdown immediate;
ORA-01109: base de données non ouverte

Base de données dūmontée.
Instance ORACLE arrêtée.

SQL>
```

Exercice 3 :

1) Pour afficher le statut de la base oracle on utilise la commande **select status from v\$instance ;**

a. Startup nomount

Donc on va lancer les 2 commandes :

> Startup nomount

> Select status from v\$instance;

```
SQL> startup nomount;
Instance ORACLE lanc  e.

Total System Global Area  654308168 bytes
Fixed Size                 9270088 bytes
Variable Size              197132288 bytes
Database Buffers           440401920 bytes
Redo Buffers               7503872 bytes
SQL> select status from v$instance;

STATUS
-----
STARTED

SQL>
```

⇒ Le statut de la base c'est : **started** c'est-à-dire que l'instance est lanc  e.

b. Startup mount

L'instance est d  j lanc  e donc il faut l'arr  ter pour qu'on puisse utiliser la commande startup mount.

> shutdown immediate

> startup mount

> select status from v\$instance;

```
SQL> shutdown immediate
ORA-01507: base de donnees non montee

Instance ORACLE arr  t  e.
SQL> startup mount
Instance ORACLE lanc  e.

Total System Global Area  654308168 bytes
Fixed Size                 9270088 bytes
Variable Size              197132288 bytes
Database Buffers           440401920 bytes
Redo Buffers               7503872 bytes
Base de donn  es mont  e.
SQL> select status from v$instance;

STATUS
-----
MOUNTED

SQL>
```

⇒ Le statut de la base c'est : **Mounted** c'est-  dire que la base est mont  e.

c. Startup open

Il faut arrêter la base pour qu'on puisse utiliser la commande startup open car l'instance est déjà lancée , le spfile et les controlfiles sont déjà ouverts.

> shutdown immediate

> startup open

> select status from v\$instance;

```
SQL> shutdown immediate
ORA-01109: base de données non ouverte

Base de données d'montée.
Instance ORACLE arrêtée.
SQL> startup open
Instance ORACLE lancée.

Total System Global Area  654308168 bytes
Fixed Size                  9270088 bytes
Variable Size              197132288 bytes
Database Buffers           440401920 bytes
Redo Buffers                7503872 bytes
Base de données montée.
Base de données ouverte.
SQL> select status from v$instance;

STATUS
-----
OPEN

SQL>
```

⇒ Statut de la base c'est : open.

2) L'équivalent de la commande startup c'est la commande : startup open parce que open c'est l'option par défaut de startup.

3) Après startup nomount la commande qui permet de monter la base est alter database mount ;

4) Après startup mount la commande qui permet d'ouvrir la base est alter database open ;

Atelier Pratique 2

Gestion de la mémoire utilisée par Oracle

Makhchoun Khadija

Exercice n 1 :

1)

Alter system set nom_parameter=valeur scope=?

Scope	Cas d'utilisation
Spfile	+ si le paramètre d'initialisation est statique ou dynamique. + la prise en compte de la modification du paramètre ne se fera qu'au prochain démarrage de la base. + la nouvelle valeur du paramètre sera inscrite dans le spfile.
Memory	+ si le paramètre d'initialisation est dynamique. + La prise en compte de la modification du paramètre est immédiate. + la nouvelle valeur du paramètre sera inscrite dans la mémoire SGA.
Both	+ si le paramètre d'initialisation est dynamique. + la prise en compte de la modification du paramètre est immédiate et la nouvelle valeur est valable au prochain démarrage de la base. + la nouvelle valeur du paramètre sera inscrite dans la mémoire SGA et aussi dans le spfile.

2) Pour modifier un paramètre d'initialisation dans le cas où la base est démarrée avec le pfile il faut :

- + arrêter la base.
- + ouvrir le pfile avec un éditeur de texte.
- + modifier le paramètre d'initialisation.
- + démarrer la base avec le pfile.

Exercice 2 :

1) La requête équivalente à `show parameter mot` est :

```
SELECT NAME,TYPE,VALUE FROM v$parameter WHERE name = 'mot';
```

2) Pour afficher la valeur du paramètre `db_cache_size` on utilise la commande :

```
Show parameter db_cache_size;
```

- ⇒ On remarque que la valeur de `db_cache_size=0`
- ⇒ La Gestion automatique de la mémoire partagée est activée. La mémoire est automatiquement allouée à la zone qui en a le plus besoin. Il n'y a pas de taille maximale pour chaque zone.

Pour afficher la valeur du paramètre `db_block_size` on utilise la commande :

Show parameter db_block_size;

- ⇒ On remarque que la valeur du paramètre db_block_size=8192 bytes.
- ⇒ Donc la taille du bloc de données est 8192 bytes.

```
SQL Plus
SQL*Plus: Release 19.0.0.0.0 - Production on Lun. Mai 31 17:59:50 2021
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Entrez le nom utilisateur : sys as sysdba
Entrez le mot de passe :

Connecté à :
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> show parameter db_cache_size;

NAME                                TYPE                                VALUE
-----                                -                                -
db_cache_size                        big integer                        0
SQL> show parameter db_block_size;

NAME                                TYPE                                VALUE
-----                                -                                -
db_block_size                        integer                            8192
SQL>
```

3) La relation entre eux :

Le **db_cache_size** désigne la taille du composant mémoire BUFFER CACHE.

Le **db_block_size** désigne la taille d'un seul bloc de données.

Le **db_block_buffers** désigne le nombre de blocs de données alloué au niveau du BUFFER CACHE. Ce dernier est organisé en un ensemble de blocs de données.

$$\Rightarrow db_cache_size = db_block_size * db_block_buffers$$

4)

La commande **show sga;** permet de visualiser la taille de la mémoire sga.

database buffers est la taille allouée au composant mémoire Buffer cache.

Donc le nombre de blocs de données alloués actuellement dans le database Buffers est :

$$\begin{aligned} \Rightarrow \text{Nb de blocs de données alloués} &= \text{database buffers} / \text{db_block_size} \\ &= 377487360 / 8192 = 46080 \text{ blocs de données} \end{aligned}$$

```
SQL> show sga;

Total System Global Area 654308168 bytes
Fixed Size                 9270088 bytes
Variable Size             260046848 bytes
Database Buffers          377487360 bytes
Redo Buffers               7503872 bytes
SQL>
```

Exercice 3 :

- 1) On va tenter de modifier la taille du composant mémoire Redo-log Buffer avec scope=memory.

```
SQL> alter system set log_buffer=30000 scope=memory;  
alter system set log_buffer=30000 scope=memory  
*  
ERREUR Ó la ligne 1 :  
ORA-02095: Le paramètre d'initialisation indiqu  ne peut pas  tre modifi   
  
SQL>
```

⇒ Ça affiche une erreur de type 'le param tre d'initialisation indiqu  ne peut pas  tre modifi '

- 2) D duisez pourquoi Redo-log buffer n'est pas concern  par la gestion automatique de la M moire ?

⇒ Le redo-log buffer n'est pas concern  par la gestion automatique de la m moire car c'est un param tre statique et donc on ne peut le modifier qu'au niveau du spfile et pas dans la m moire.

Atelier Pratique 3

Gestion des fichiers d'une base oracle

Makhchoun Khadija

Exercice 1 :

- 1) Interrogez la vue V\$logfile pour visualiser les fichiers redo-log.

```
SQL Plus
Entrez le nom utilisateur : sys as sysdba
Entrez le mot de passe :

Connecté à :
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> select status from v$instance;

STATUS
-----
OPEN

SQL> select * from v$logfile;

  GROUP# STATUS  TYPE
-----
MEMBER
-----
IS_     CON_ID
-----
      3      ONLINE
D:\APP\ORADATA\MAKHCHOUN\REDO03.LOG
NO      0
      2      ONLINE
D:\APP\ORADATA\MAKHCHOUN\REDO02.LOG
NO      0
  GROUP# STATUS  TYPE
-----
MEMBER
-----
IS_     CON_ID
-----
      1      ONLINE
D:\APP\ORADATA\MAKHCHOUN\REDO01.LOG
NO      0

SQL>
```

- ⇒ On constate qu'on a 3 fichiers différents.
- ⇒ On a 3 groupes, chaque groupe contient un seul fichier dans ce cas.

- 2) Interrogez la vue V\$log pour connaître le nombre de membres d'un groupe (Members), le fichier courant (status : inactive, current, unused ou active).

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

  GROUP#  THREAD#  SEQUENCE#  BYTES  BLOCKSIZE  MEMBERS ARC
-----
STATUS    FIRST_CHANGE# FIRST_TI NEXT_CHANGE# NEXT_TIM  CON_ID
-----
CURRENT   1             1         7  209715200      512      1 NO
2576154  24/05/21    1,8447E+19
INACTIVE  2             1         5  209715200      512      1 NO
2401988  21/05/21    2487515  23/05/21
INACTIVE  3             1         6  209715200      512      1 NO
2487515  23/05/21    2576154  24/05/21

SQL>
```

- ⇒ Dans ce cas on 2 fichiers inactifs : (c'est-à-dire qu'ils ne sont pas en cour d'utilisation) et un seul fichier curent (c'est-à-dire qu'il est utilisé par le processus LGWR).

- 3) Utilisez la commande SQL « **Alter system switch logfile ;** » pour forcer un changement de fichier de journalisation.


```
SQL> alter system switch logfile;

Système modifié.

SQL>
```

⇒ Cette commande force le changement du fichier de journalisation (utilisé par le LGWR).

4) Interrogez la vue V\$log pour visualiser le changement de sequence et status.

```
SQL> alter system switch logfile;

Système modifié.

SQL> select * from v$log;

  GROUP#  THREAD#  SEQUENCE#  BYTES  BLOCKSIZE  MEMBERS  ARC
-----  -
STATUS    FIRST_CHANGE# FIRST_TIME NEXT_CHANGE# NEXT_TIME  CON_ID
-----  -
ACTIVE    1            1          7  209715200      512      1 NO
          2576154  24/05/21      2613407  25/05/21
CURRENT   2            1          8  209715200      512      1 NO
          2613407  25/05/21  1,8447E+19
INACTIVE  3            1          6  209715200      512      1 NO
          2487515  23/05/21      2576154  24/05/21

SQL>
```

⇒ On constate que le groupe n 1 qui était curent est passé à active. Parce que le logfile courant c'est lui dans lequel on est en train d'écrire les commandes validées et la commande switch logfile permet de forcer le changement vers un autre logfile. Donc celui qui était utilisé par le LGWR va devenir active parce qu'on va le synchroniser avec le datafile.

⇒ Le groupe n 2 était inactive et il est passé à curent parceque c'est lui qui avait le min de séquences et son numéro de séquence va passer à : $\max(\text{sequence})+1=7+1=8$

5) Le tableau suivant est le résultat de la requête SQL « select ... from V\$log ; ». Donnez le nouveau tableau qui correspond à l'exécution de la commande SQL : Alter system switch logfile ;

```
select group#, sequence#, bytes, archived, status
from v$log ;

  GROUP#  SEQUENCE#  BYTES  ARC  STATUS
-----  -
1         269    1048576 NO   INACTIVE
2         270    1048576 NO   INACTIVE
3         271    1048576 NO   CURRENT
```

⇒ le nouveau tableau :

GROUP#	SEQUENCE#	BYTES	ARC	STATUS
1	272	1048576	NO	CURRENT
2	270	1048576	NO	INACTIVE
3	271	1048576	NO	ACTIVE

- ⇒ le groupe 3 qui était curent va passer à active parce qu'on va le synchroniser avec le datafile.
- ⇒ le groupe 1 qui a le min de séquences (269) et qui était inactive va devenir curent et son nouveau nombre de séquence va passer au $\max(\text{séquence})+1=271+1=272$.
- ⇒ le groupe 2 va rester inactive ?

Exercice 2 :

- 1) Démarrez le processus ARC (via le paramètre d'initialisation log_archive_start).

On va d'abord vérifier si le processus ARC est déjà démarré. Donx on va afficher la valeur du paramètre `log_archive_start` via la commande : `show parameter log_archive_start`

```
SQL Plus
SQL*Plus: Release 19.0.0.0.0 - Production on Mar. Mai 25 21:28:09 2021
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Entrez le nom utilisateur : sys as sysdba
Entrez le mot de passe :

Connectû Ó :
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> show parameter log_archive_start

NAME                                TYPE        VALUE
-----
log_archive_start                    boolean     FALSE
SQL>
```

- ⇒ On remarque que le processus ARC n'est pas démarré car la valeur du paramètre `log_archive_start` égale à false.

* On va lui affecter la valeur true. Pour lancer le processus ARC on doit lancer la commande `alter system set log_archive_start=true scope= ?` Cette commande va affecter au paramètre `log_archive_start` la valeur true et grâce à cette opération on pourrait démarrer le processus ARC.

* Pour identifier la nature du paramètre d'initialisation `log_archive_start` on utilise la commande :

`select name, issys_modifiable from v$parameter where name='log_archive_start';`

Cette commande permet d'afficher le nom du paramètre ainsi qu'une 2^{ème} colonne qui admet 2 valeurs : false si le paramètre est statique et immediate si le paramètre est dynamique.

```
SQL> select name , issys_modifiable from v$parameter where name='log_archive_start';

NAME                                TYPE        VALUE
-----
ISSYS_MOD
-----
log_archive_start
FALSE
SQL>
```

- ⇒ On a false donc ce paramètre est statique d'où la valeur de scope est spfile.

*On va maintenant démarrer le processus ARC :

Alter system set log_archive_start=true scope=spfile;

```
SQL> alter system set log_archive_start=true scope=spfile;
Systeme modifié.
SQL>
```

⇒ La nouvelle valeur de ce paramètre est enregistrée dans le spfile. Donc on doit redémarrer la base pour que le processus oracle puisse lire à nouveau le contenu du spfile et prendre en considération les nouvelles valeurs.

> shutdown immediate

> startup

> show parameter log_archive_start (pour s'assurer que le processus ARC est démarré)
On constate que la valeur de log_archive_start est passée à true)

```
SQL Plus
SQL> shutdown immediate
ORA-01109: base de données non ouverte

Base de données démontée.
Instance ORACLE arrêtée.
SQL> startup
ORA-32004: obsolete or deprecated parameter(s) specified for RDBMS instance
Instance ORACLE lancée.

Total System Global Area  654308168 bytes
Fixed Size                  9270088 bytes
Variable Size              197132288 bytes
Database Buffers           440401920 bytes
Redo Buffers                7503872 bytes
Base de données montée.
Base de données ouverte.
SQL> show parameter log_archive_start

NAME                           TYPE        VALUE
-----
log_archive_start               boolean     TRUE
```

2) Démarrez la base en mode MOUNT et positionnez-la en mode ARCHIVELOG.

* Pour démarrer la base en mode Mount il faut d'abord l'arrêter.

> shutdown immediate

> startup mount

* ensuite on va vérifier le mode de fonctionnement de la base oracle est ce qu'il est archive log ou noarchive log.

> archive log list (permet d'identifier le mode de fonctionnement de la base oracle)

Ou bien on utilise > select log_mode from v\$database ;

```
SQL> shutdown immediate
Base de données fermée.
Base de données démontée.
Instance ORACLE arrêtée.
SQL> startup mount
ORA-32004: obsolete or deprecated parameter(s) specified for RDBMS instance
Instance ORACLE lancée.

Total System Global Area  654308168 bytes
Fixed Size                  9270088 bytes
Variable Size              197132288 bytes
Database Buffers           440401920 bytes
Redo Buffers                7503872 bytes
Base de données montée.
SQL> archive log list
mode Database log               mode No Archive
Archivage automatique           Désactivé
Destination de l'archive        USE_DB_RECOVERY_FILE_DEST
Séquence de journal en ligne la plus ancienne  6
Séquence de journal courante    8
SQL>
```

⇒ On constate que le mode de fonctionnement de la base est noarchive c'est-à-dire que l'archivage automatique est désactivé.

* Maintenant on va basculer vers le mode ARCHIVELOG via la commande

```
alter database archivelog ;
```

* Ensuite on affiche le nouveau mode de fonctionnement de la base avec la commande :

```
archive log list ou bien select log_mode from v$database ;
```

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

LOG_MODE
-----
NOARCHIVELOG

SQL> alter database archivelog;

Base de données modifiée.

SQL> archive log list
mode Database log          mode Archive
Archivage automatique      Activé
Destination de l'archive    USE_DB_RECOVERY_FILE_DEST
Séquence de journal en ligne la plus ancienne      6
Séquence de journal suivante à archiver           8
Séquence de journal courante                        8
SQL> select log_mode from v$database;

LOG_MODE
-----
ARCHIVELOG

SQL>
```

3) Ouvrez la base.

Pour ouvrir la base on utilise la commande : `alter database open` ; car la base est déjà montée il suffit d'ouvrir les datafiles et les logfiles.

```
SQL> alter database open;

Base de données modifiée.

SQL>
```

4) Vérifiez que la base est en mode ARCHIVELOG.

Soit on utilise la commande : `log archive start`

Soit on utilise la commande : `select log_mode from v$database ;`

```
SQL> alter database open;
Base de données modifiée.

SQL> archive log list
mode Database log          mode Archive
Archivage automatique      Activé
Destination de l'archive    USE_DB_RECOVERY_FILE_DEST
Séquence de journal en ligne la plus ancienne  6
Séquence de journal suivante à archiver      8
Séquence de journal courante                  8
SQL> select log_mode from v$log;

LOG_MODE
-----
ARCHIVELOG

SQL>
```

⇒ On constate que la base est en mode ARCHIVELOG.

5) Identifiez le fichier redo-log courant.

La commande `select * from v$log;` permet d'afficher le statut de tous les logfiles.

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

  GROUP#  THREAD#  SEQUENCE#  BYTES  BLOCKSIZE  MEMBERS ARC
-----
STATUS    FIRST_CHANGE# FIRST_TI NEXT_CHANGE# NEXT_TIM  CON_ID
-----
INACTIVE  1            1          7  209715200      512      1 YES
          2576154  24/05/21    2613407  25/05/21
CURRENT  2            1          8  209715200      512      1 NO
          2613407  25/05/21    1,8447E+19
INACTIVE  3            1          6  209715200      512      1 YES
          2487515  23/05/21    2576154  24/05/21

SQL>
```

⇒ Dans ce cas le groupe n°2 est le groupe courant ça veut dire que le processus LGWR permet d'écrire les commandes validées au niveau des fichiers de ce groupe.

6) Forcez la création d'un fichier redo-log archivé.

Pour switcher vers un autre groupe et forcer ainsi l'archivage du logfile qui était courant on utilise la commande `alter system switch logfile;`

```
SQL> alter system switch logfile;
Système modifié.

SQL>
```

- 7) Vérifiez que le fichier d'archive a été créé au moment du « switch logfile » en regardant, le répertoire pointé par le paramètre DB_RECOVERY_FILE_DEST (Zone de récupération rapide : emplacement par défaut des archives et des sauvegardes).

La commande `show parameter DB_RECOVERY_FILE_DEST` permet d'identifier l'emplacement par défaut des archives.

```
SQL> show parameter DB_RECOVERY_FILE_DEST
```

NAME	TYPE	VALUE
db_recovery_file_dest	string	D:\APP\fast_recovery_area
db_recovery_file_dest_size	big integer	12732M

```
SQL>
```

On va aller dans le disque dans le chemin indiqué :

The first screenshot shows the path `Ce PC > Nouveau nom (D:) > APP > fast_recovery_area > MAKHCHOUN`. It contains a table of files:

Nom	Modifié le	Type	Taille
ARCHIVELOG	26/05/2021 00:00	Dossier de fichiers	
ONLINELOG	18/05/2021 17:02	Dossier de fichiers	
CONTROL02.CTL	31/05/2021 20:03	Fichier CTL	18 288 Ko

The second screenshot shows the path `Ce PC > Nouveau nom (D:) > APP > fast_recovery_area > MAKHCHOUN > ARCHIVELOG`. It contains a table of files:

Nom	Modifié le	Type	Taille
2021_05_18	18/05/2021 16:57	Dossier de fichiers	
2021_05_19	19/05/2021 18:18	Dossier de fichiers	
2021_05_25	25/05/2021 23:02	Dossier de fichiers	
2021_05_26	26/05/2021 01:16	Dossier de fichiers	

The third screenshot shows the path `Ce PC > Nouveau nom (D:) > APP > fast_recovery_area > MAKHCHOUN > ARCHIVELOG > 2021_05_26`. It contains a table of files:

Nom	Modifié le	Type	Taille
O1_MF_1_8_JBV4TPN9_ARC	26/05/2021 01:16	Fichier ARC	67 078 Ko

⇒ On constate que l'archive est bien créée.

- 8) Basculez la base en mode NOARCHIVELOG.

Il faut d'abord arrêter la base et ensuite la démarrer en mode mount. Ensuite on bascule vers le mode noarchivelog et on ouvre la base.

`>shutdown immediate`

`>startup mount`

`>alter database noarchivelog ;`

`>alter database open ;`

```

SQL> shutdown immediate
Base de données fermée.
Base de données démontée.
Instance ORACLE arrêtée.
SQL> startup mount
ORA-32004: obsolete or deprecated parameter(s) specified for RDBMS instance
Instance ORACLE lancée.

Total System Global Area  654308168 bytes
Fixed Size                  9270088 bytes
Variable Size              197132288 bytes
Database Buffers           440401920 bytes
Redo Buffers                7503872 bytes
Base de données montée.
SQL> alter database noarchivelog;

Base de données modifiée.

SQL> alter database open;

Base de données modifiée.

SQL>

```

9) Ajoutez des fichiers redo-log

- A un groupe existant

La commande qui permet d'ajouter un fichier à un groupe existant est :

Alter database add logfile member 'D:\APP\ORADATA\MAKHCHOUN\REDO04.LOG' to group 1;

⇒ On ajoute un fichier au groupe 1.

Ensuite on utilise la commande **select * from v\$logfile ;** pour vérifier est ce que le fichier est ajouté.

```

SQL> alter database add logfile member 'D:\APP\ORADATA\MAKHCHOUN\REDO04.LOG' to group 1;
Base de données modifiée.

SQL> select * from v$logfile;

  GROUP# STATUS  TYPE
-----
MEMBER
-----
IS_     CON_ID
-----
      3      ONLINE
D:\APP\ORADATA\MAKHCHOUN\REDO03.LOG
NO      0
      2      ONLINE
D:\APP\ORADATA\MAKHCHOUN\REDO02.LOG
NO      0
  GROUP# STATUS  TYPE
-----
MEMBER
-----
IS_     CON_ID
-----
      1      ONLINE
D:\APP\ORADATA\MAKHCHOUN\REDO01.LOG
NO      0
      1 INVALID ONLINE
D:\APP\ORADATA\MAKHCHOUN\REDO04.LOG
  GROUP# STATUS  TYPE
-----
MEMBER
-----
IS_     CON_ID
-----
NO      0

```

- A un nouveau groupe

On doit créer un nouveau groupe et lui affecter un numéro qui n'existe pas et indiquer une taille pour les fichiers de ce nouveau groupe. La commande qui permet de faire ceci est :

Alter database add logfile group 4 'D:\APP\ORADATA\MAKHCHOUN\REDO05.LOG' size 5M;

```
SQL> alter database add logfile group 4 'D:\APP\ORADATA\MAKHCHOUN\REDO05.LOG' size 5M;
Base de données modifiée.
SQL>
```

10) Supprimez le groupe inactive de fichiers redo-log.

Pour supprimer un groupe on a 2 conditions :

- ✓ Il faut avoir plus de 2 groupes.
- ✓ Il ne faut pas supprimer un groupe courant ou active.

Dans notre cas on va supprimer le groupe 1 car il est inactive. La commande utilisée est :

Alter database drop logfile group 1 ;

```
SQL> SET LINESIZE 32000;
SQL> select * from v$log;

  GROUP#  THREAD#  SEQUENCE#    BYTES BLOCKSIZE  MEMBERS ARC STATUS           FIRST_CHANGE# FIRST_TI NEXT_CHANGE# NEXT_TIM    CON_ID
-----
       1         1         7 209715200      512         2 YES INACTIVE           2576154 24/05/21      2613407 25/05/21         0
       2         1         8 209715200      512         1 YES INACTIVE           2613407 25/05/21      2646990 26/05/21         0
       3         1         9 209715200      512         1 NO  CURRENT           2646990 26/05/21  1,8447E+19         0
       4         1         0 5242880      512         1 YES UNUSED              0         0         0         0

SQL> alter database drop logfile group 1;
Base de données modifiée.
SQL> select * from v$log;

  GROUP#  THREAD#  SEQUENCE#    BYTES BLOCKSIZE  MEMBERS ARC STATUS           FIRST_CHANGE# FIRST_TI NEXT_CHANGE# NEXT_TIM    CON_ID
-----
       2         1         8 209715200      512         1 YES INACTIVE           2613407 25/05/21      2646990 26/05/21         0
       3         1         9 209715200      512         1 NO  CURRENT           2646990 26/05/21  1,8447E+19         0
       4         1         0 5242880      512         1 YES UNUSED              0         0         0         0

SQL>
```

11) Tentez de supprimer un groupe current de fichiers redo-log.

⇒ On ne peut pas supprimer un groupe courant.

Exercice 3 :

1) Identifiez l'emplacement des fichiers redo-log.

Pour identifier l'emplacement des fichiers redo-log on utilise la commande :

Select member from v\$logfile ;

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

MEMBER
-----
-----
-----
-----
D:\APP\ORADATA\MAKHCHOUN\REDO03.LOG
D:\APP\ORADATA\MAKHCHOUN\REDO02.LOG
D:\APP\ORADATA\MAKHCHOUN\REDO05.LOG

SQL>
```


2) Arrêtez la base et déplacez un fichier redo-log.

Pour arreter la base on utilise la commande **shutdown immediate**

```
SQL> shutdown immediate
Base de données fermée.
Base de données démontée.
Instance ORACLE arrêtée.
SQL>
```

On va aller sur le disque dans le chemin indiqué (D:\APP\ORADATA\MAKHCHOUN) et on va déplacer un fichier parmi les fichiers redo-log.

Ce PC > Nouveau nom (D:) > APP > oradata > makhchoun

	Nom	Modifié le	Type	Taille
id Files	pdb	18/05/2021 17:24	Dossier de fichiers	
	pdbseed	18/05/2021 17:03	Dossier de fichiers	
	CONTROL01.CTL	31/05/2021 21:43	Fichier CTL	18 288 Ko
	REDO01.LOG	26/05/2021 12:43	Document texte	204 801 Ko
	REDO02.LOG	31/05/2021 21:43	Document texte	204 801 Ko
	REDO03.LOG	31/05/2021 18:00	Document texte	204 801 Ko
	REDO04.LOG	26/05/2021 12:43	Document texte	204 801 Ko
	REDO05.LOG	31/05/2021 21:43	Document texte	5 121 Ko
	SYSAUX01.DBF	31/05/2021 21:43	Fichier DBF	768 008 Ko
	SYSTEM01.DBF	31/05/2021 21:43	Fichier DBF	931 848 Ko
nents	TEMP01.DBF	30/05/2021 19:11	Fichier DBF	32 776 Ko
	UNDOTBS01.DBF	31/05/2021 21:43	Fichier DBF	66 568 Ko
	USERS01.DBF	31/05/2021 21:43	Fichier DBF	5 128 Ko

J'ai choisi de déplacer le fichier REDO05 et je l'ai mis ici D:\APP\oradata

Ce PC > Nouveau nom (D:) > APP > oradata

	Nom	Modifié le	Type	Taille
e	makhchoun	26/05/2021 13:15	Dossier de fichiers	
	ORCL	31/03/2021 16:04	Dossier de fichiers	
	REDO05.LOG	31/05/2021 21:43	Document texte	5 121 Ko

3) Démarrez la base en mode MOUNT et indiquez à Oracle le nouvel emplacement du fichier redo-log.

- Pour démarrer la base en mode MOUNT on utilise la commande : **startup mount**
- Pour indiquer à oracle le nouvel emplacement du fichier redo-log qu'on a déplacé on utilise la commande :

```
alter database rename file 'D:\APP\ORADATA\MAKHCHOUN\REDO05.LOG' to 'D:\APP\ORADATA\REDO05.LOG';
```

```

SQL> startup mount
ORA-32004: obsolete or deprecated parameter(s) specified for RDBMS instance
Instance ORACLE lanc  e.

Total System Global Area  654308168 bytes
Fixed Size                 9270088 bytes
Variable Size             197132288 bytes
Database Buffers          440401920 bytes
Redo Buffers              7503872 bytes
Base de donn  es mont  e.
SQL> alter database rename file 'D:\APP\ORADATA\MAKHCHOUN\REDO05.LOG' to 'D:\APP\ORADATA\REDO05.LOG';

Base de donn  es modifi  e.

SQL>

```

- 4) Ouvrez la base et v  rifiez la prise en compte par Oracle de ce changement.
 - Pour ouvrir la base on utilise la commande : **alter database open ;**
 - Pour v  rifier la prise en compte par oracle de ce changement on utilise la commande : **select member from v\$logfile ;**

```

SQL> alter database open;

Base de donn  es modifi  e.

SQL> select member from v$logfile;

MEMBER
-----
-----
-----
-----
-----
D:\APP\ORADATA\MAKHCHOUN\REDO03.LOG
D:\APP\ORADATA\MAKHCHOUN\REDO02.LOG
D:\APP\ORADATA\REDO05.LOG

SQL>

```

⇒ On constate que le fichier REDO05 a   t   d  plac   au nouvel emplacement qu'on a indiqu      oracle.

Exercice 4 :

- 1) Interrogez la vue V\$controlfile pour identifier les fichiers de contr  le de votre base.

On utilise la commande : > **select * from v\$controlfile ;** ou bien > **show parameter control_files**

```
SQL*Plus: Release 19.0.0.0.0 - Production on Lun. Juin 7 21:30:38 2021
Version 19.3.0.0.0
```

```
Copyright (c) 1982, 2019, Oracle. All rights reserved.
```

```
Entrez le nom utilisateur : sys as sysdba
```

```
Entrez le mot de passe :
```

```
Connecté à une instance inactive.
```

```
SQL> startup
```

```
Instance ORACLE lancée.
```

```
Total System Global Area 520092992 bytes
```

```
Fixed Size 9268544 bytes
```

```
Variable Size 188743680 bytes
```

```
Database Buffers 314572800 bytes
```

```
Redo Buffers 7507968 bytes
```

```
Base de données montée.
```

```
Base de données ouverte.
```

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

```
NAME
```

```
-----
```

```
KHADIJAM
```

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

```
STATUS
```

```
-----
```

```
NAME
```

```
-----
```

```
IS_ BLOCK_SIZE FILE_SIZE_BKLS CON_ID
```

```
-----
```

```
C:\USERS\PC\DESKTOP\APP\DB_HOME\ORADATA\KHADIJAMAK\CONTROL01.CTL
```

```
NO 16384 1142 0
```

```
D:\APP\FAST_RECOVERY_AREA\KHADIJAMAK\CONTROL02.CTL
```

```
NO 16384 1142 0
```

```
STATUS
```

```
-----
```

```
NAME
```

```
-----
```

```
IS_ BLOCK_SIZE FILE_SIZE_BKLS CON_ID
```

```
-----
```

```
SQL> show parameter control_files
```

```
NAME
```

```
TYPE
```

```
VALUE
```

```
-----
```

```
control_files string
```

```
C:\USERS\PC\DESKTOP\APP\DB_HOME\ORADATA\KHADIJAMAK\CONTROL01
```

```
.CTL, D:\APP\FAST_RECOVERY_ARE
```

```
A\KHADIJAMAK\CONTROL02.CTL
```

```
SQL>
```

⇒ Dans ma base j'ai donc deux fichiers de contrôle.

2) Multiplexez le fichier de contrôle en utilisant le spfile.

➤ On va d'abord ajouter un nouveau fichier de contrôle (niveau logique) dans le paramètre CONTROL_FILES.

➤ Ensuite il faut arrêter la base via la commande : **shutdown immediate**

```
SQL> alter system set control_files='C:\USERS\PC\DESKTOP\APP\DB_HOME\ORADATA\KHADIJAMAK\CONTROL01.CTL', 'D:\APP\FAST_RECOVERY_AREA\KHADIJAMAK\CONTROL02.CTL',
2 'D:\APP\FAST_RECOVERY_AREA\KHADIJAMAK\CONTROL03.CTL' scope=spfile;

Système modifié.

SQL> shutdown immediate
Base de données fermée.
Base de données démontée.
Instance ORACLE arrêtée.
SQL>
```

- Ajouter le fichier de contrôle au niveau physique

```
cmd

C:\Users\pc
λ copy C:\USERS\PC\DESKTOP\APP\DB_HOME\ORADATA\KHADIJAMAK\CONTROL01.CTL D:\APP\FAST_RECOVERY_AREA\KHADIJAMAK\CONTROL03.CTL
1 fichier(s) copié(s).

C:\Users\pc
λ |
```

- Démarrer la base avec la commande : **Startup**
- Finalement on va interroger la vue v\$controlfile pour voir est ce que le nouveau controlfile est ajouté.

```
SQL> startup
Instance ORACLE lancée.

Total System Global Area  520092992 bytes
Fixed Size                 9268544 bytes
Variable Size             188743680 bytes
Database Buffers          314572800 bytes
Redo Buffers               7507968 bytes
Base de données montée.
Base de données ouverte.
SQL> select * from v$controlfile;

STATUS
-----
NAME
-----
IS_ BLOCK_SIZE FILE_SIZE_BLKs      CON_ID
---
C:\USERS\PC\DESKTOP\APP\DB_HOME\ORADATA\KHADIJAMAK\CONTROL01.CTL
NO      16384      1142      0

D:\APP\FAST_RECOVERY_AREA\KHADIJAMAK\CONTROL02.CTL
NO      16384      1142      0

STATUS
-----
NAME
-----
IS_ BLOCK_SIZE FILE_SIZE_BLKs      CON_ID
---
D:\APP\FAST_RECOVERY_AREA\KHADIJAMAK\CONTROL03.CTL
NO      16384      1142      0

SQL>
```

- 3) Multiplexez le fichier de contrôle en utilisant le pfile.

- Ensuite il faut arrêter la base via la commande : **shutdown immediate**

```
SQL> shutdown immediate
Base de données fermée.
Base de données démontée.
Instance ORACLE arrêtée.
SQL>
```

- Ensuite on va modifier le paramètre CONTROL_FILES dans le fichier pfile :

📁 > Ce PC > Nouveau nom (D:) > APP > admin > khadijamak > pfile

Id	Nom	Modifié le	Type	Taille
	init.ora.51202115337	01/06/2021 14:16	Fichier 51202115337	3 Ko

init.ora.51202115337 - Bloc-notes

Fichier Edition Format Affichage Aide

```
#####
# Copyright (c) 1991, 2013 by Oracle Corporation
#####

#####
# Cache and I/O
#####
db_block_size=8192

#####
# File Configuration
#####
control_files=('C:\Users\pc\Desktop\APP\db_home\oradata\khadijamak\control01.ct1', "D:\APP\fast_recovery_area\KHADIJAMAK\control02.ct1",
'D:\APP\fast_recovery_area\KHADIJAMAK\control03.ct1')
```

- Ajouter le fichier de contrôle au niveau physique

Le fichier control03 existe déjà sur le disque.

- Démarrer la base avec le pfile:
startup pfile='D:\APP\admin\khadijamak\pfile\init.ora.51202115337'

```
SQL> startup pfile='D:\APP\admin\khadijamak\pfile\init.ora.51202115337'
Instance ORACLE lanc e.
```

```
Total System Global Area  520092992 bytes
Fixed Size                  9268544 bytes
Variable Size              188743680 bytes
Database Buffers          314572800 bytes
Redo Buffers                7507968 bytes
```

Base de donn es mont e.

Base de donn es ouverte.

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

STATUS

NAME

IS_	BLOCK_SIZE	FILE_SIZE_BLKs	CON_ID
-----	------------	----------------	--------

C:\USERS\PC\DESKTOP\APP\DB_HOME\ORADATA\KHADIJAMAK\CONTROL01.CTL

NO	16384	1142	0
----	-------	------	---

D:\APP\FAST_RECOVERY_AREA\KHADIJAMAK\CONTROL02.CTL

NO	16384	1142	0
----	-------	------	---

STATUS

NAME

IS_	BLOCK_SIZE	FILE_SIZE_BLKs	CON_ID
-----	------------	----------------	--------

D:\APP\FAST_RECOVERY_AREA\KHADIJAMAK\CONTROL03.CTL

NO	16384	1142	0
----	-------	------	---

```
SQL>
```

Atelier Pratique 5

Gestion des tablespaces

MAKHCHOUN Khadija

Outil utilisé : SQL*Plus

Exercice 1 :

1. Créez un tablespace avec un datafile.

Pour créer un tablespace avec datafile on utilise la commande suivante :

```
create tablespace table1 datafile 'C:\Users\pc\Desktop\APP\db_home\oradata\khadijamak\table_1.dbf'
size 16M;
```

```
SQL> create tablespace table1 datafile 'C:\Users\pc\Desktop\APP\db_home\oradata\khadijamak\table_1.dbf' size 1M;
Tablespace créé.
SQL>
```

⇒ On remarque que le datafile TABLE_01.DBF est créé.

📁 > Bureau > APP > db_home > oradata > khadijamak

Nom	Modifié le	Type	Taille
pdb	01/06/2021 15:53	Dossier de fichiers	
pdbseed	01/06/2021 14:42	Dossier de fichiers	
CONTROL01.CTL	12/06/2021 16:00	Fichier CTL	18 288 Ko
REDO01.LOG	12/06/2021 14:15	Document texte	204 801 Ko
REDO02.LOG	12/06/2021 16:04	Document texte	204 801 Ko
REDO03.LOG	10/06/2021 15:00	Document texte	204 801 Ko
SYS_AUX01.DBF	12/06/2021 16:05	Fichier DBF	624 648 Ko
SYSTEM01.DBF	12/06/2021 16:00	Fichier DBF	921 608 Ko
TABLE_1.DBF	12/06/2021 16:42	Fichier DBF	1 032 Ko
TEMP01.DBF	12/06/2021 14:25	Fichier DBF	32 776 Ko
UNDOTBS01.DBF	12/06/2021 15:57	Fichier DBF	46 088 Ko
USERS01.DBF	12/06/2021 14:25	Fichier DBF	5 128 Ko

2. Définissez ce tablespace comme default pour l'utilisateur SYS.

Pour définir le tablespace qu'on a créé au niveau de la 1^{ère} question comme default pour l'utilisateur SYS on utilise la commande suivante :

```
alter user sys default tablespace table1;
```

```
SQL> alter user sys default tablespace table1;
Utilisateur modifié.
SQL>
```

3. Créez une table dans ce tablespace avec initial=16k.

Pour créer une table dans ce tablespace :

Create table Etudiant (
 Id int primary key,
 Nom varchar(30),
 Prenom varchar(30)
)
 Storage (initial 16k),
 Tablespace table1 ;

```
SQL> create table Etudiant (
  2 id int primary key,
  3 nom varchar(30),
  4 prenom varchar(30))
  5 storage (initial 16k)
  6 tablespace table1;
```

Table cr  e.

Consultez la vue dba_segments pour d  duire le nombre d'extensions (extents) initial de la table.

On utilise la commande :

```
select owner, segment_name, segment_type, tablespace_name, extents,max_extents, bytes
from dba_segments where segment_name='ETUDIANT' and owner='SYS';
```

```
SQL> select owner, segment_name, segment_type, tablespace_name, extents,max_extents, bytes from dba_segments where segment_name='ETUDIANT' and owner='SYS';
```

OWNER	SEGMENT_NAME	SEGMENT_TYPE	TABLESPACE_NAME	EXTENTS	MAX_EXTENTS	BYTES
SYS	ETUDIANT	TABLE	TABLE1	1	2147483645	65536

➤ Le nombre d'extensions   gale    1.

5. Cr  ez une proc  dure PL/SQL pour ins  rer 100000 lignes dans la table.

J'ai eu l'erreur suivante : impossible d'  tendre la table SYS.ETUDIANT de 8 dans le tablespace lorsque j'ai ex  cut   la commande execute sinscrire ; donc j'ai chang   le nombre de lignes    100 dans la proc  dure pour que l'espace soit suffisant.

```
create or replace procedure sinscrire is
Begin
For i in 1..100 LOOP
Insert into Etudiant (id, nom, prenom)
Values (i, 'nom', 'prenom') ;
End LOOP ;
End ;
/
```

```
SQL> create or replace procedure sinscrire is
  2 Begin
  3   For i in 1..100 LOOP
  4   Insert into ETUDIANT(id, nom, prenom)
  5   Values(i, 'nom', 'prenom');
  6   End LOOP;
  7   End;
  8   /
```

Proc  dure cr  e.

Ensuite on doit ex  cuter la commande : SQL> execute sinscrire;


```
SQL> execute inscrire;

Procédure PL/SQL terminée avec succès.

SQL>
```

6. Interprétez l'évolution de la table en termes de nombre d'extensions ajoutées.

```
select owner, segment_name, segment_type, tablespace_name, extents,max_extents, bytes
from dba_segments where segment_name='ETUDIANT' and owner='SYS';
```

```
SQL> select owner, segment_name, segment_type, tablespace_name, extents,max_extents, bytes from dba_segments where segment_name='ETUDIANT' and owner='SYS';
```

OWNER	SEGMENT_NAME	SEGMENT_TYPE	TABLESPACE_NAME	EXTENTS	MAX_EXTENTS	BYTES
SYS	ETUDIANT	TABLE	TABLE1	8	2147483645	524288

- Après avoir insérer 100 lignes dans la table, le nombre d'extensions a augmenté à 8.

7. Déduisez le nombre de blocs (blocks) nécessaires pour stocker 100000 lignes.
 Dans mon cas j'ai inséré que 100 lignes parceque l'espace n'est pas suffisant pour insérer 100000 ligne.

```
SQL> select BLOCKS from user_segments where segment_name='ETUDIANT';
```

BLOCKS
64

```
SQL>
```

- Pour stocker 100 lignes on a besoin donc de 64 blocs.

Exercice 2 :

1. Tentez de déplacer le fichier du tablespace USERS vers une autre destination.

On doit d'abord identifier l'emplacement de ce fichier, donc on va utiliser la commande suivante :

```
Select file_name from DBA_DATA_FILES where tablespace_name='USERS';
```

```
SQL> Select file_name from DBA_DATA_FILES where tablespace_name='USERS';
```

FILE_NAME
C:\USERS\PC\DESKTOP\APP\DB_HOME\ORADATA\KHADIJAMAK\USERS01.DBF

```
SQL>
```

db_home > oradata > khadijamak

Nom	Modifié le	Type	Taille
pdb	01/06/2021 15:53	Dossier de fichiers	
pdbseed	01/06/2021 14:42	Dossier de fichiers	
CONTROL01.CTL	12/06/2021 18:35	Fichier CTL	18 288 Ko
REDO01.LOG	12/06/2021 14:15	Document texte	204 801 Ko
REDO02.LOG	12/06/2021 18:16	Document texte	204 801 Ko
REDO03.LOG	10/06/2021 15:00	Document texte	204 801 Ko
SYSAUX01.DBF	12/06/2021 17:56	Fichier DBF	624 648 Ko
SYSTEM01.DBF	12/06/2021 18:06	Fichier DBF	921 608 Ko
TABLE_1.DBF	12/06/2021 18:35	Fichier DBF	1 032 Ko
TEMP01.DBF	12/06/2021 18:27	Fichier DBF	32 776 Ko
UNDOTBS01.DBF	12/06/2021 17:58	Fichier DBF	46 088 Ko
USERS01.DBF	12/06/2021 14:25	Fichier DBF	5 128 Ko

Maintenant, on va tenter de le déplacer vers une autre destination.

⇒ J'ai tenté de le déplacer manuellement et ça marche. Ensuite je l'ai remis au premier endroit.

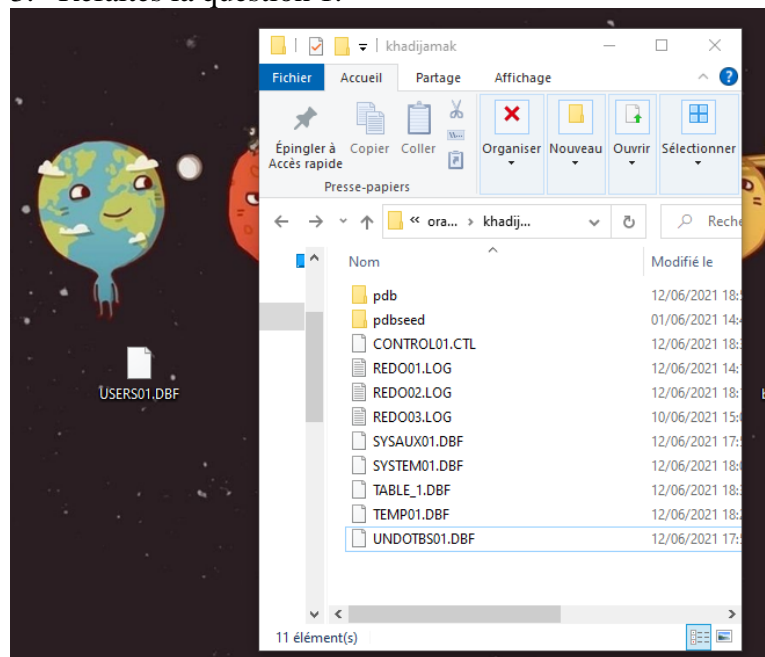
2. Mettez le statut du tablespace en OFFLINE.

La commande qui permet de basculer le tablespace vers le mode OFFLINE est la suivante :

Alter tablespace USERS offline ;

```
SQL> alter tablespace USERS offline;
Tablespace modifié.
SQL>
```

3. Refaites la question 1.



➤ On remarque que le fichier est bien déplacé.

4. Mettez le tablespace en ONLINE.

On utilise donc la commande suivante :

Alter tablespace USERS online ;

➤ On m'affiche l'erreur suivante :

```
SQL> alter tablespace USERS online;
alter tablespace USERS online
*
ERREUR Ó la ligne 1 :
ORA-01157: impossible d'identifier ou de verrouiller le fichier de données 7 -
voir le fichier trace DBWR
ORA-01110: fichier de données 7 :
'C:\USERS\PC\DESKTOP\APP\DB_HOME\ORADATA\KHADIJAMAK\USERS01.DBF'

SQL>
```

5. Indiquez au tablespace le nouveau datafile.

Pour indiquer au tablespace le nouveau datafile on utilise la commande :

Alter tablespace Users rename datafile

'C:\USERS\PC\DESKTOP\APP\DB_HOME\ORADATA\KHADIJAMAK\USERS01.DBF' to
'C:\Users\pc\Desktop\USERS01.DBF';

```
SQL> Alter tablespace Users rename datafile 'C:\USERS\PC\DESKTOP\APP\DB_HOME\ORADATA\KHADIJAMAK\USERS01.DBF' to 'C:\Users\pc\Desktop\USERS01.DBF';
Tablespace modifiÓ.

SQL>
```

6. Refaites la question 4.

```
SQL> alter tablespace USERS online;

Tablespace modifiÓ.

SQL>
```

- Maintenant on a pu basculer vers le mode online.
- On va afficher le nouvel emplacement du fichier du tablespace USERS :

```
SQL> Select file_name from DBA_DATA_FILES where tablespace_name='USERS';

FILE_NAME
-----
C:\USERS\PC\DESKTOP\USERS01.DBF

SQL>
```

- On remarque que l'emplacement est modifiÓ.

Exercice 3

Considérons le tablespace Tbs qui possède un fichier 'c:\oracle\tbs01.dbf' avec une taille de 50M. Quelles sont les différentes commandes qui permettront d'ajouter 10M à Tbs ?

- On va d'abord créer le tablespace Tbs avec le fichier de données tbs01.dbf, donc on va utiliser la commande suivante :

SQL> Create tablespace TBS datafile

'C:\Users\pc\Desktop\APP\db_home\oradata\khadijamak\tbs01.dbf' SIZE 50M;

```
SQL> create tablespace TBS datafile 'C:\Users\pc\Desktop\APP\db_home\oradata\khadijamak\tbs01.dbf' SIZE 50M;
Tablespace cr   .
SQL>
```

- On remarque que le fichier tbs01 est cr    dans l'emplacement indiqu    dans la commande.

eau > APP > db_home > oradata > khadijamak

Nom	Modifi���	Type	Taille
pdb	12/06/2021 18:54	Dossier de fichiers	
pdbseed	01/06/2021 14:42	Dossier de fichiers	
CONTROL01.CTL	12/06/2021 22:01	Fichier CTL	18 288 Ko
REDO01.LOG	12/06/2021 21:55	Document texte	204 801 Ko
REDO02.LOG	12/06/2021 21:53	Document texte	204 801 Ko
REDO03.LOG	12/06/2021 22:01	Document texte	204 801 Ko
SYS_AUX01.DBF	12/06/2021 22:01	Fichier DBF	624 648 Ko
SYSTEM01.DBF	12/06/2021 22:01	Fichier DBF	921 608 Ko
TABLE_1.DBF	12/06/2021 22:01	Fichier DBF	1 032 Ko
TBS01.DBF	12/06/2021 22:17	Fichier DBF	51 208 Ko
TEMP01.DBF	12/06/2021 18:27	Fichier DBF	32 776 Ko
UNDOTBS01.DBF	12/06/2021 22:01	Fichier DBF	46 088 Ko

- Pour ajouter 10M au tablespace TBS on a 3 m        :

- 1) La premi   re m        consiste     cr   er un nouveau datafile tbs012 de 10M et l'associer au tablespace TBS qu'on a d        cr       . Pour ce faire on utilise la commande :

```
alter tablespace TBS add datafile
```

```
'C:\Users\pc\Desktop\APP\db_home\oradata\khadijamak\tbs012.dbf' size 10M ;
```

```
SQL> alter tablespace TBS add datafile 'C:\Users\pc\Desktop\APP\db_home\oradata\khadijamak\tbs012.dbf' size 10M;
Tablespace modifi   .
SQL>
```

- 2) La deuxi   me m        est de modifier la taille du datafile tbs01 avec l'option resize en utilisant la commande suivante :

```
alter database datafile
```

```
'C:\Users\pc\Desktop\APP\db_home\oradata\khadijamak\tbs01.dbf' resize 60M ;
```

```
SQL> alter database datafile 'C:\Users\pc\Desktop\APP\db_home\oradata\khadijamak\tbs01.dbf' resize 60M;
Base de donn      modifi   .
SQL>
```

TBS01.DBF	12/06/2021 22:32	Fichier DBF	61 448 Ko
-----------	------------------	-------------	-----------

- 3) La troisi   me m        est d'augmenter automatiquement la taille du datafile tbs01 en ajoutant une extension automatique via la commande suivante :

```
alter database datafile
```

```
'C:\Users\pc\Desktop\APP\db_home\oradata\khadijamak\tbs01.dbf' autoextend
on next 10M maxsize 60M ;
```

```
SQL> alter database datafile 'C:\Users\pc\Desktop\APP\db_home\oradata\khadijamak\tbs01.dbf' autoextend
2 on next 10M maxsize 60M ;
Base de donn      modifi   .
SQL>
```

Exercice 4 :

1. Créez un tablespace pour la gestion des données d'annulation.

Pour créer un tablespace d'annulation 'annulation1' on utilise la commande :

Create undo tablespace annulation1 datafile

'C:\Users\pc\Desktop\APP\db_home\oradata\khadijamak\annulation01.dbf' size 50M
autoextend on next 5120K maxsize 1000M ;

```
SQL> Create undo tablespace annulation1 datafile 'C:\Users\pc\Desktop\APP\db_home\oradata\khadijamak\annulation01.dbf'
2 size 50M
3 autoextend on
4 next 5120K
5 maxsize 1000M
6 ;

Tablespace cr   .

SQL>
```

➤ On remarque que le fichier annulation01.dbf est cr   .

reau > APP > db_home > oradata > khadijamak

Nom	Modifi�� le	Type	Taille
pdb	12/06/2021 18:54	Dossier de fichiers	
pdbseed	01/06/2021 14:42	Dossier de fichiers	
ANNULATION01.DBF	12/06/2021 22:51	Fichier DBF	51 208 Ko
CONTROL01.CTL	12/06/2021 22:01	Fichier CTL	18 288 Ko
REDO01.LOG	12/06/2021 21:55	Document texte	204 801 Ko
REDO02.LOG	12/06/2021 21:53	Document texte	204 801 Ko
REDO03.LOG	12/06/2021 22:01	Document texte	204 801 Ko
SYS_AUX01.DBF	12/06/2021 22:01	Fichier DBF	624 648 Ko
SYSTEM01.DBF	12/06/2021 22:01	Fichier DBF	921 608 Ko
TABLE_1.DBF	12/06/2021 22:01	Fichier DBF	1 032 Ko
TBS01.DBF	12/06/2021 22:32	Fichier DBF	61 448 Ko
TBS012.DBF	12/06/2021 22:26	Fichier DBF	10 248 Ko
TEMP01.DBF	12/06/2021 18:27	Fichier DBF	32 776 Ko
UNDOTBS01.DBF	12/06/2021 22:01	Fichier DBF	46 088 Ko

2. Indiquez    Oracle qu'il faudra g  rer les donn  es d'annulation dans ce tablespace.

On va d'abord identifier la valeur du param  tre undo_management avec la commande :

>show parameter undo_management

```
SQL> show parameter undo_management

NAME                                TYPE        VALUE
-----
undo_management                     string      AUTO
SQL>
```

La valeur de ce param  tre undo_management est AUTO, ce qui signifie que le mode de gestion des donn  es d'annulation est sur les tablespace UNDO .

Avant d'indiquer le tablespace undo, on va identifier la valeur du param  tre undo_tablespace pour voir si on doit changer sa valeur ou pas :

>show parameter undo_tablespace

```
SQL> show parameter undo_tablespace
```

NAME	TYPE	VALUE
undo_tablespace	string	UNDOTBS1

```
SQL>
```

La valeur du paramètre d'initialisation est UNDOTBS1, donc on doit maintenant modifier cette valeur et indiquer à Oracle que le tablespace annulation1 qui doit gérer les données d'annulation, en utilisant la commande :

```
> alter system set undo_tablespace=annulation1 scope= both;
```

```
SQL> alter system set undo_tablespace=annulation1 scope= both;
```

```
Système modifié.
```

```
SQL>
```

2. Tentez de créer un Rollback segment dans le tablespace System.

La création du rollback segment revient à la création d'un segment d'annulation dans le tablespace system.

La création du rollback segment se fait à l'aide de la commande :

```
> create rollback segment test storage (initial 200K next 200K) tablespace system;
```

```
SQL> create rollback segment test storage (initial 200K next 200K) tablespace system;
```

```
Segment d'annulation (rollback segment) créé.
```

```
SQL>
```