



# Administration des Bases de données



Filière Ingénieur - GI

**Noreddine Gherabi**



# Plan

Gestion d'une instance Oracle

Création de la base de données

Structure d'une base de donnée Oracle.



# Architecture-Instance



## Démarrer une instance

Lire le fichier init.ora, à démarrer l'instance, allouer la mémoire, et préparer les processus d'arrière plan

**NOMOUNT**

Ouvrir le ou les fichiers CONTROL FILE.

**MOUNT**

Ouvrir tous les fichiers de données enregistrés dans les fichiers CONTROLFILE.

**OPEN**

```
STARTUP NOMOUNT PFILE="emplacement du init.ora"  
ALTER DATABASE MOUNT; -  
ALTER DATABASE OPEN;
```



# Architecture-Instance

## Arrêter une instance

La base de données est ni fermée, ni démontée.

- L'instance est arrêtée sans fermeture des fichiers.
- Les Redo Log Buffer ne sont pas écrits sur disques.
- Les transactions sont interrompues.
- Pas de Check Point.
- Les utilisateurs sont déconnectés.

**ABORT**

• les transactions non validées actives sont annulées (ROLLBACK).

- les Redo Log Buffer sont écrits sur disques.
- les utilisateurs sont déconnectés.
- la base de données est fermée.
- la base de données est démontée.
- l'instance est arrêtée.

**IMMEDIATE**

- attente de déconnexion des utilisateurs.
- les nouvelles transactions sont autorisées.
- les Redo Log Buffer sont écrits sur disques.
- processus d'arrière-plans sont arrêtés.
- mémoire SGA libérée.
- la base de données est fermée/démontée.
- l'instance est arrêtée.

**NORMAL**



# Architecture-Database

## Database

**La base de données est l'ensemble des fichiers qui permettent de gérer les données stockées dans la base de données.**

**Une base de données est constituée de :**

- Un fichier de contrôle( .ctl),** contenant les informations sur tous les autres fichiers de la base (nom, emplacement, taille).

**Le fichier de contrôle contient les informations suivantes :**

- Le nom et l'identifiant de la base de données**
- Le nom et l'emplacement des fichiers de données et des fichiers Redo Log**
- Le nom des tablespaces**
- La date et l'heure de création de la base de données**
- Des informations relatives au point de synchronisation**
- Les informations de sauvegarde de l'utilitaire Recovery Manager**



# Architecture-Database

## Database

- **Fichiers de Redo Log(.redo, .log)**, contenant l'activité des sessions connectées à la base. Ce sont des journaux de transactions de la base.

Les fichiers Redo-log contiennent l'historique des modifications apportées à la base de données Oracle

- **Fichiers de Redo Log archivés** contenant les archives d'anciens fichiers de Redo Log.



# Architecture-Database

## Database

- **Fichiers de données (.dbf)** : qui contiennent les données proprement dites, elle contient à la création de la base de données au minimum :
  - **Tablespace SYSTEM**, contenant le dictionnaire de données.
  - **Tablespace SYSAUX**, c'est le tablespace auxiliaire du tablespace SYSTEM contenant des fonctions Oracle.
  - **Tablespace Temporaire TEMP**, récupérant les segments temporaires utilisés par les requêtes SQL de la base de données.
  - **Tablespace UNDO**, récupérant la version précédente des données en cours de modification par les transactions se déroulant sur la base.
  - **Tablespace USERS**, tablespace de travail par défaut des utilisateurs.



# Architecture-Database

## Database

- **Un fichier de paramètres (admin):** binaire SPFILE<SID>.ORA, contenant les paramètres de démarrage de l'instance et d'autres valeurs qui déterminent l'environnement dans lequel elle s'exécute.
- **Un fichier de mots de passe,** contenant le mot de passe du privilège SYSDBA ( PWD <SID>)( Oracle.. / Product... / database)



# Architecture-Database

Database

## Liste des fichiers

```
Select NAME from v$datafile  
union all  
select NAME from v$controlfile  
union all  
select MEMBER from v$logfile;
```



## Installation et Crédit de base Oracle

- Installation du noyau et des outils Oracle
- Etats d'une base
- Crédit d'une base
- Le fichier d'initialisation *init.ora*
- Démarrage d'une base
- Arrêt d'une base



# Installation d'Oracle

## Avant l'installation

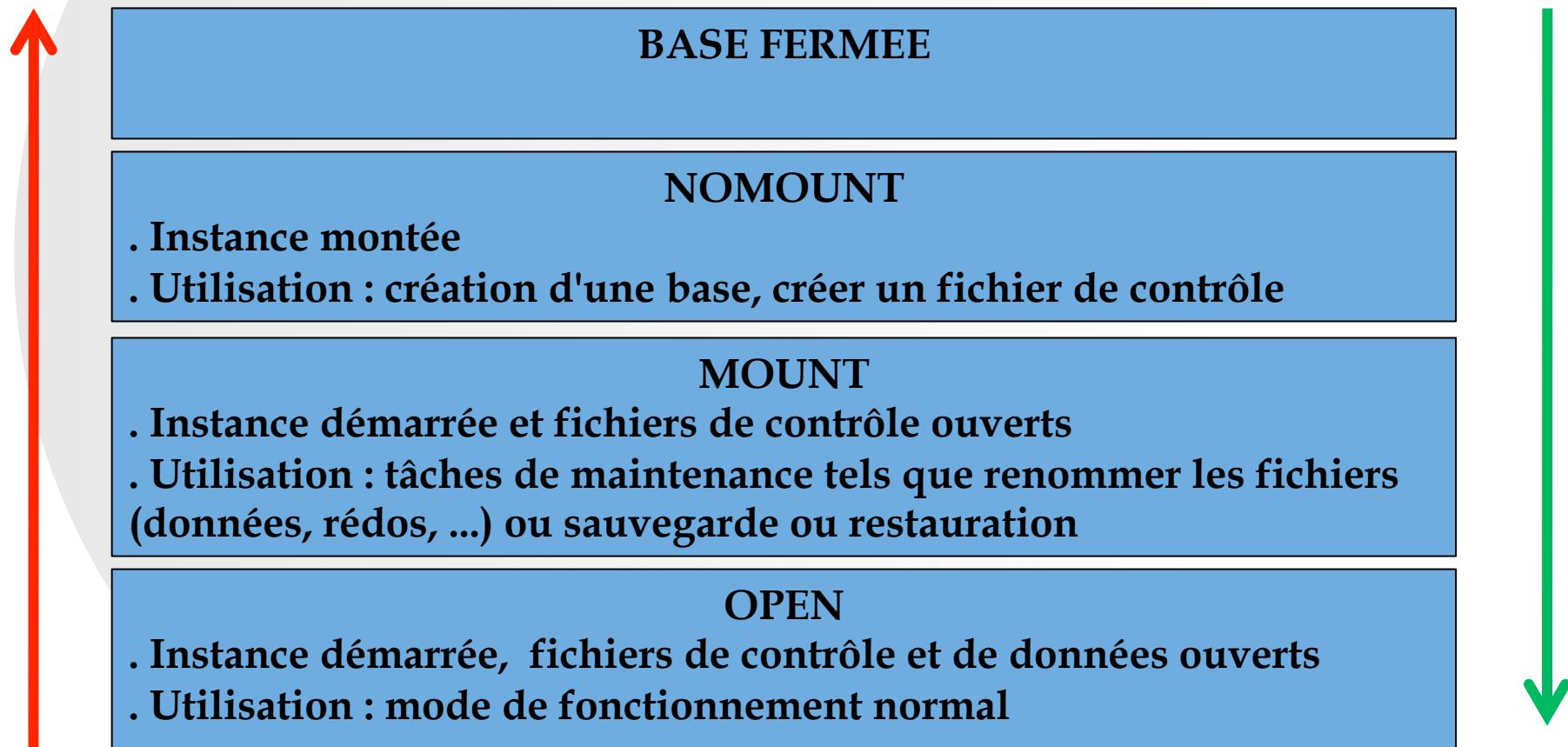
**Etudier le manuel d'installation d'Oracle ou du produit**

- **Préparer l'OS et la machine**
  - Cas NT : se connecter sur la machine comme Administrateur.
- **Estimer l'espace disque nécessaire pour les produits à installer et la base de départ**
- **Positionner les variables d'environnement ORACLE\_HOME et ORACLE\_SID, ...**



# Installation et Création de base Oracle

## Etats de la base de données



**Shutdown**

**Startup**



# Installation d'Oracle

## Création d'une base de données

L'ordre SQL CREATE DATABASE crée la nouvelle base de données :

- Création des fichiers de contrôle ;
- Création des fichiers de journalisation ;
- Création du tablespace SYSTEM et de son fichier de données ;
- Création du tablespace SYSAUX et de son fichier de données ;
- Création des comptes SYS et SYSTEM ;
- Création éventuelle d'un tablespace d'annulation, d'un tablespace temporaire par défaut et d'un tablespace permanent par défaut.
- Création du dictionnaire de données (dans le tablespace SYSTEM)



# Structure d'une Base de données Oracle

## Création d'une base de données

### Structure Physique d'une Base de Données Oracle

- Les fichiers de données
- Les fichiers Redo log
- Les fichiers d'Archives
- Les fichiers de Contrôle

### Structure logique d'une Base de données Oracle

- Les tablespaces
- Les segments et leurs composants
- Les segments de données de type table
- Les segments d'index
- Les segments temporaires
- Les segments rollback



# Installation d'Oracle

## Création d'une base de données

Les différentes étapes de la création de la base de données proprement dite sont :

1. Créer les répertoires sur les disques
2. Préparer un nouveau fichier de paramètres init<SID>.ora
3. Créer un fichier de paramètres serveur à partir du fichier init<SID>.ora
4. Positionner ORACLE\_SID , au nom de l'instance
5. Sous Windows uniquement, créer le service associé à l'instance en utilisant l'outil ORADIM (qui gère les services rattachés aux instances des bases oracle)
6. Démarrer l'instance en état NOMOUNT
7. Créer la base en exécutant l'ordre CREATE DATABASE
8. Installer le dictionnaire de données et packages destinés au bon fonctionnement de la base de données



# Installation d'Oracle

## Création d'une base de données

### 1- Crédit du chemin de l'installation

```
c:\>mkdir E:\oracle\ORA
```

### 2- Variables d'environnement à positionner

```
C:\>set ORACLE_HOME= %ORACLE_HOME%
```

```
C:\>set PATH= %ORACLE_HOME% \BIN
```

```
C:\>set ORACLE_SID=nom de SID
```



# Installation d'Oracle

## Création d'une base de données

### 3- Création de l'instance

```
C:\>oradim -NEW -SID ora -STARTMODE auto -INTPWD manager
```

### 4- Connection à l'instance :

```
SQL> sqlplus/nolog  
SQL> conn / as sysdba
```



# Installation d'Oracle

## Création d'une base de données

### 5- Crée le chier de paramètres PFILE (init.ora)

Spécification des paramètres nécessaires:

- db\_name = 'ora11'
- control\_files = 'E:\ORACLE\ORA\control1.ctl','E:\ORACLE\ORA\control2.ctl'
- diagnostic\_dest='E:\ORACLE\ORA'

### 6-Creation des paramètres serveur (SPFILE)

```
SQL> create SPFILE from PFILE='c:\init.TXT';
```

### 7. Démarrage de l'instance

```
SQL> startup nomount
```



# Installation d'Oracle

## Création d'une base de données

### 8- Création de la base

**CREATE DATABASE ora**

**USER SYS IDENTIFIED BY manager**

**USER SYSTEM IDENTIFIED BY manager**

**LOGFILE GROUP 1 ('E:\oracle\ora\redo01.log') SIZE 100M,**

**GROUP 2 ('E:\oracle\ora\redo02.log') SIZE 100M,**

**GROUP 3 ('E:\oracle\ora\redo03.log') SIZE 100M**

**MAXLOGFILES 5**

**MAXLOGMEMBERS 5**

**MAXDATAFILES 100**

**MAXINSTANCES 1**

**DATAFILE 'E:\oracle\ora\system01.dbf' SIZE 325M**

**SYSAUX DATAFILE 'E:\oracle\ora\sysaux01.dbf' SIZE 325M**

**UNDO TABLESPACE undotbs**

**DATAFILE 'E:\ORACLE\ORA\undotbs01.dbf'**

**SIZE 200M REUSE AUTOEXTEND ON MAXSIZE UNLIMITED;**



# Installation d'Oracle

## Création d'une base de données

### 9- Construction de dictionnaires de données

```
SQL> @%ORACLE_HOME%/rdbms/admin/catalog.sql  
SQL> @%ORACLE_HOME%/rdbms/admin/catproc.sql  
SQL> @%ORACLE_HOME%/sqlplus/admin/pupbld.sql
```



## Structure physique d'une Base de données

### Les fichiers de données

Ils contiennent toutes les données relatives à une base de données (dictionnaire Oracle, Tables, index, rollback segments, segment temporaires)

- Ces fichiers sont lus par les process Serveurs et modifiés par DBWn
- L'unité de découpage est le bloc (2K, 4K, 8K, 16K, 32K) selon l'OS
- Ils appartiennent à une et une seule base

Les fichiers de données contiennent des informations de deux types :

- Le dictionnaire de données
- Les données des utilisateurs

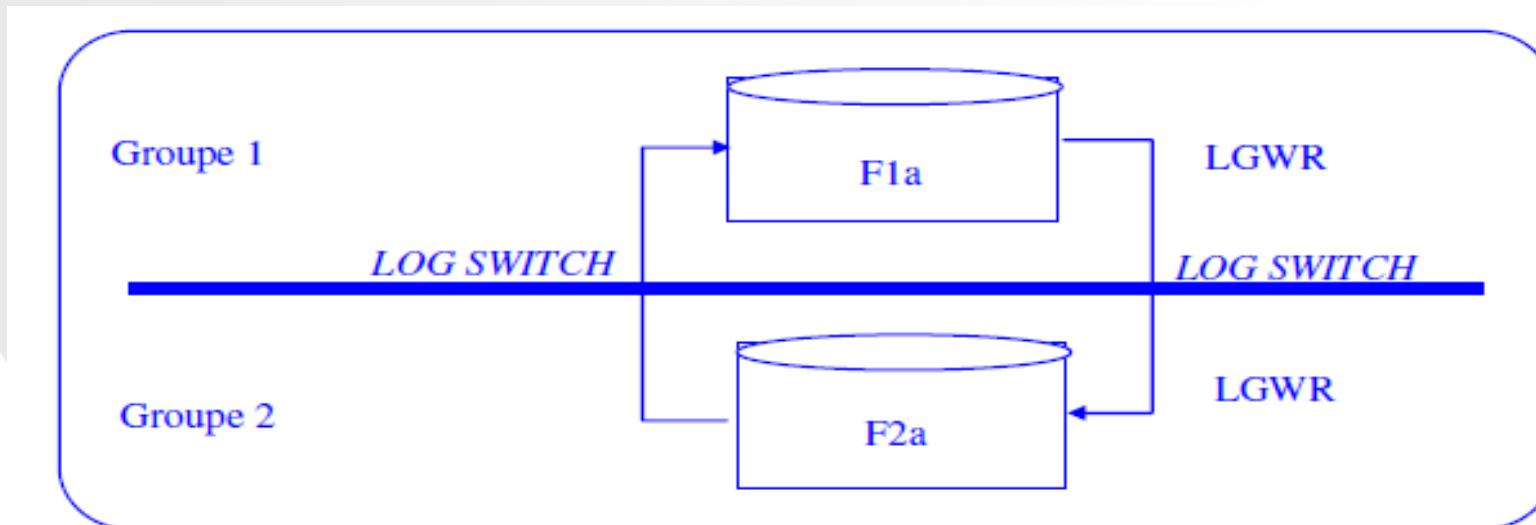


## Structure physique d'une Base de données

### Les fichiers REDO LOG

Les fichiers Redo-log contiennent l'historique des modifications apportées à la base de données

Ces fichiers peuvent être multiplexés (comprenez *dupliqués dans des répertoires de groupe*) afin de fournir un maximum de sécurité.

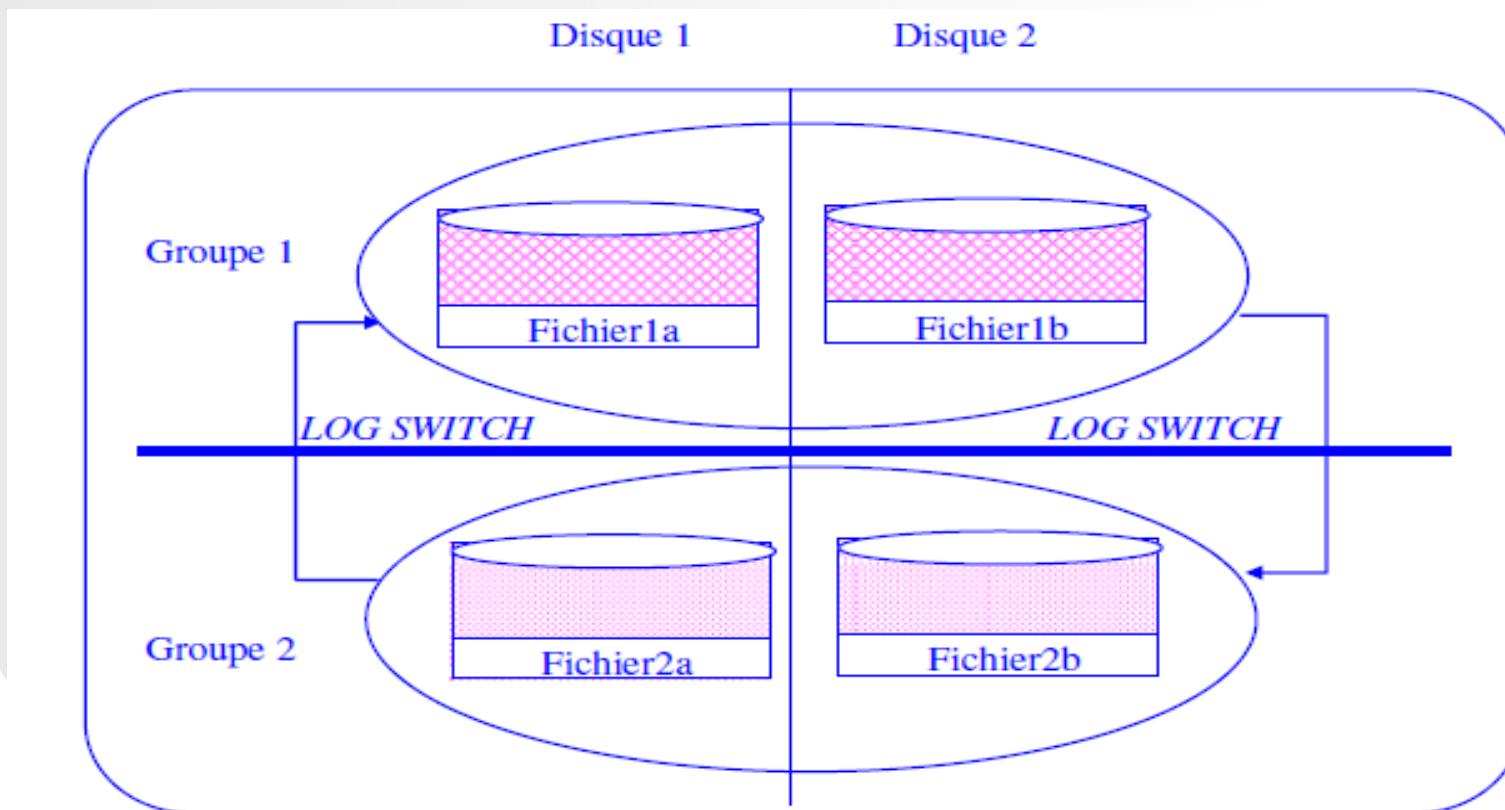


Configuration minimale



# Structure physique d'une Base de données

## Les fichiers REDO LOG



Configuration normale



## Structure physique d'une Base de données

### Les fichiers REDO LOG

- Ajout de fichiers REDO

- Ajout d'un groupe de fichiers

```
ALTER DATABASE [database]
    ADD LOGFILE
        [ GROUP integer ] filespec
```

**Exemple :**

```
ALTER DATABASE FST
ADD LOGFILE GROUP 4 ('I:\app\nour\oradata\fst\redo.log')
size 5000K;
```



## Structure physique d'une Base de données

### Les fichiers REDO LOG

- Suppression des fichiers REDO
- Pour supprimer tous les membres d'un groupe il faut supprimer le groupe

```
ALTER DATABASE [database ]
DROP LOGFILE
{ GROUP integer
```

```
DROP LOGFILE MEMBER
```



## Structure physique d'une Base de données

### Les fichiers REDO LOG

- **Forcer le basculement du groupe courant**
- Le basculement d'un groupe courant peut être utilisé lorsque l'on a besoin d'effectuer une suppression de groupe ou lorsque l'on veut générer un archive avant d'effectuer une sauvegarde

```
ALTER SYSTEM SWITCH LOGFILE ;
```



## Structure physique d'une Base de données

### Les fichiers REDO LOG

- **Les informations sur les fichiers Redo Log**

- **V\$LOG : informations sur les groupes**
- **V\$LOGFILE : informations sur les membres**



## Structure physique d'une Base de données

### Les fichiers REDO LOG

- Ajout de fichiers REDO
  - Ajout d'un membre dans un groupe

```
ALTER DATABASE [database ]
| ADD LOGFILE MEMBER
'filename' TO { GROUP integer
```

Exemple :

```
ALTER DATABASE Fst
ADD LOGFILE MEMBER 'I:\app\nour\oradata\fst \redo.log '
TO GROUP 4;
```

Ou

```
ALTER DATABASE Fst
ADD LOGFILE MEMBER 'I:\app\nour\oradata\fst \redo1.log '
TO 'I:\app\nour\oradata\fst \redo.log ';
```



## Structure physique d'une Base de données

### Les fichiers de contrôle

- Le contenu de: controlfile
  - Le nom et l'identifiant de la base de données
  - Le nom et l'emplacement des fichiers de données et des fichiers Redo Log
  - Le nom des tablespaces
  - La date et l'heure de création de la base de données
  - Des informations relatives au point de synchronisation
  - Les informations de sauvegarde de l'utilitaire Recovery Manager



## Structure physique d'une Base de données

### Les fichiers de contrôle

- Backup d'un fichier de contrôle

Sauvegarde en dupliquant un fichier de contrôle

```
ALTER DATABASE  
BACKUP CONTROLFILE TO arch_ctl.dbf;
```



## Structure physique d'une Base de données

### Les fichiers de contrôle

- **Changement de l'emplacement d'un fichier de contrôle**

1. Récupérer le fichier Spfile
2. Arrêter l'instance et la base de données
3. Changer le chemin des fichiers de contrôle
4. Recharger le fichier spfile
5. Démarrer la base

- **Liste des fichiers de controle**

```
SELECT status, name FROM v$controlfile;
```



## Structure physique d'une Base de données

### Les fichiers de contrôle

- *Voir le statut des fichiers de contrôle « controls files »*

- SELECT status, name FROM v\$controlfile;**
- Show parameter control\_files**
- Select name, value from v\$parameter where name ='control\_files'**

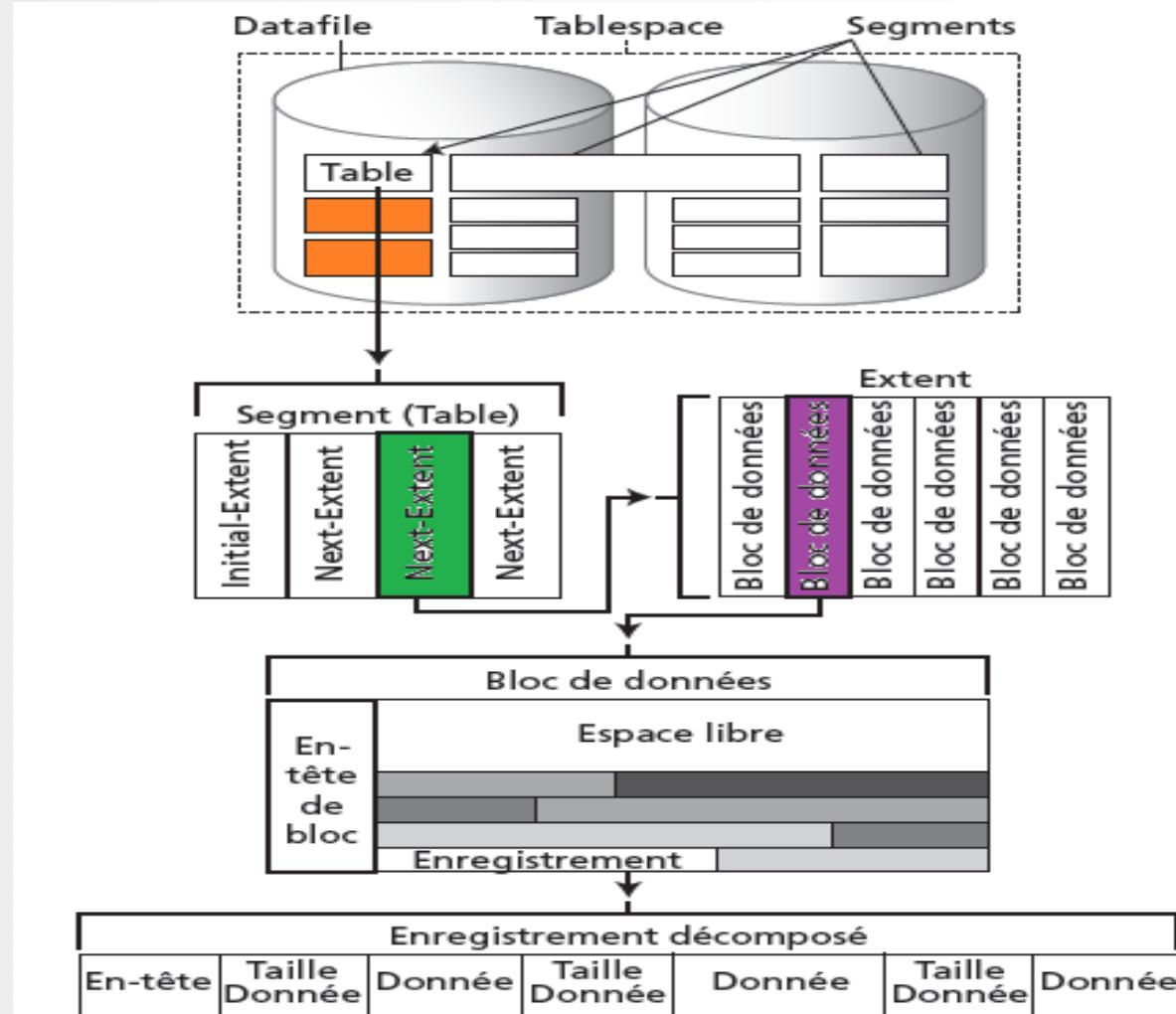


## Structure Logique d'une Base de données

- **Tablespace**
- **Gestion des Tablespace**
- **Segments**
- **Extension**
- **B loc**



# Gestion de stockage



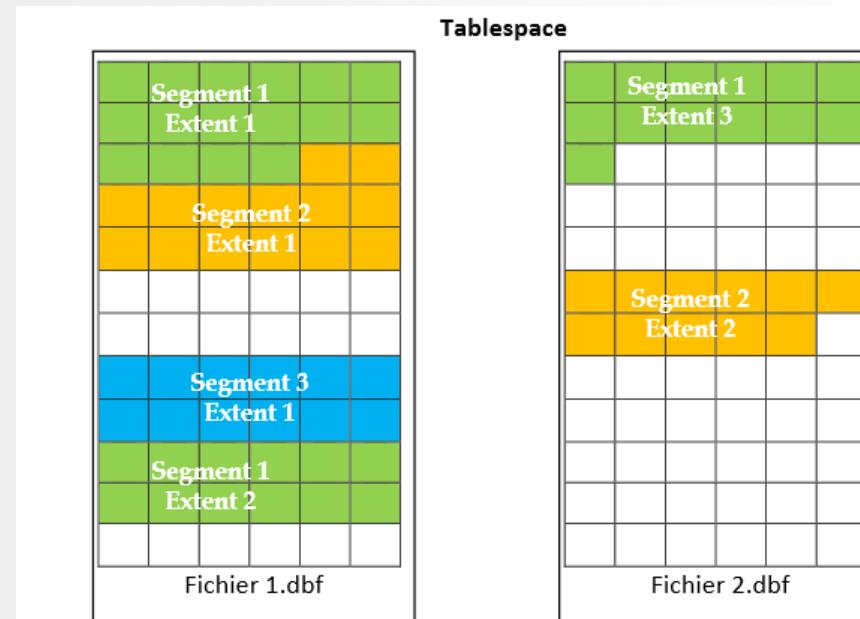


# Structure physique d'une Base de données

## Tablespace

À l'intérieur d'un tablespace, le stockage est organisé en segments contenant une ou plusieurs extensions (extents),

- Une extension étant un ensemble de blocs Oracle contigus.
- Lorsqu'un segment est créé dans un tablespace, Oracle lui alloue une (ou plusieurs) extension(s) dans un des fichiers de données du tablespace





## Structure physique d'une Base de données

### Tablespace

Un tablespace est une unité logique de stockage composée d'un ou de plusieurs fichiers physiques.

- Pour chaque base, il y a au moins un tablespace d'origine, appelé **SYSTEM**, qui contient le dictionnaire des données (toutes les informations sur la base)
- Un tablespace peut être actif (online) ou désactivé (offline)
- Un tablespace correspond à un ou plusieurs fichiers physiques;
- Une table peut être créée dans un tablespace , qui peut s'étendre sur plusieurs fichiers physiques..
- Un tablespace peut être TEMPORARY ou PERMANENT (mode par défaut).
- Un tablespace temporaire contient des données temporaires liés au TRI par exemple
- Un tablespace peut être réservé à contenir des données Rollback : UNDO TABLESPACE



## Structure physique d'une Base de données

### Pourquoi Tablespace ?

- Pour séparer les données de l'application des données du dictionnaire Oracle ;
- Pour séparer les données de plusieurs applications stockées dans la même base de données ;
- Pour séparer le stockage des différents types d'objets ;
- Pour répartir les entrées/sorties sur plusieurs disques ;
- Pour réaliser des sauvegardes/restaurations partielles ;
- Pour contrôler la disponibilité des données.
- Contrôler l'allocation des espaces disques aux utilisateurs via des quotas.
- Augmenter la disponibilité des données (arrêt partiel d'une base via l'arrêt d'un Tablespace)



## Structure physique d'une Base de données

### Etats d'un Tablespace

- Par défaut ONLINE à la création
- Peut être mis OFFLINE
- Peut être sauvegardé et restauré sans arrêter la base entière
- Il est créé explicitement par l'administrateur
- Peut être mis READ ONLY



## Structure physique d'une Base de données

### Création d'un tablespace permanent

```
CREATE [ BIGFILE | SMALLFILE ] TABLESPACE Ts_nom  
DATAFILE 'nom_fichier' [ SIZE valeur [K|M|G|T] ] [REUSE]  
AUTOEXTEND { OFF | ON [ NEXT valeur [K|M|G|T] ]  
[ MAXSIZE { UNLIMITED | valeur [K|M|G|T] } ] }  
EXTENT MANAGEMENT  
LOCAL { AUTOALLOCATE | UNIFORM [ SIZE valeur [K|M|G|T] ] }  
SEGMENT SPACE MANAGEMENT { MANUAL | AUTO }  
[ BLOCKSIZE valeur (K) ]  
[ LOGGING | NOLOGGING ]  
[ FORCE LOGGING ]  
[ ONLINE | OFFLINE ];
```



## Structure physique d'une Base de données

### Paramètres de création d'un tablespace permanent

- **DATAFILE** = spécification (emplacement, taille, ...) d'un (ou plusieurs) fichier(s) de données pour le tablespace.
- **AUTOEXTEND** : Indique si le fichier de données peut (ON) ou non (OFF) grossir une fois que tout l'espace initialement alloué est utilisé..
- **NEXT** : espace minimum alloué lors d'une extension
- **MAXSIZE** : taille maximum du fichier, éventuellement non limitée
- **EXTENT MANAGEMENT** : permet de définir le mode de gestion des extents à l'intérieur du tablespace.
- **SEGMENT SPACE MANAGEMENT** : permet de définir le mode de gestion de l'espace libre des segments à l'intérieur du tablespace.



## Structure physique d'une Base de données

### Paramètres de création d'un tablespace permanent

- **BLOCKSIZE valeur [K]**: Cette clause définit la taille de bloc utilisée par le tablespace.  
Les valeurs autorisées sont 2 Ko, 4 Ko, 8 Ko, 16 Ko et 32 Ko.
- **LOGGING | NOLOGGING**: enregistre les segments dans les Redo Log si LOGGING (par défaut) est défini.
- **FORCE LOGGING** : oblige l'enregistrement des segments dans les Redo Log même si la clause NOLOGGING est définie.
- **ONLINE | OFFLINE** : indique si le tablespace est accessible (ON) ou non (OFF).



## Structure physique d'une Base de données

### Tablespace par défaut

Lorsqu'un utilisateur crée un segment sans préciser de tablespace , Oracle stocke le segment dans le tablespace par défaut de l'utilisateur.

**Exemple :**

```
CREATE TABLESPACE data  
DATAFILE 'g:\oracle\ora\datadaf01.dbf' SIZE 10M  
AUTOEXTEND ON NEXT 10M MAXSIZE 500M  
EXTENT MANAGEMENT LOCAL AUTOALLOCATE;
```

**ALTER DATABASE DEFAULT TABLESPACE data;**  
« nom tablespace existant »



## Structure physique d'une Base de données

### Tablespace par défaut

Pour retrouver le nom du tablespace permanent par défaut, vous pouvez interroger

La vue DATABASE\_PROPERTIES pour la  
**DEFAULT\_PERMANENT\_TABLESPACE** :

**Syntaxe:**

```
SELECT property_value FROM database_properties
WHERE property_name = 'DEFAULT_PERMANENT_TABLESPACE';
```



## Structure physique d'une Base de données

### Modifier un tablespace permanent

#### Ajouter un fichier de données

```
ALTER TABLESPACE nom  
ADD DATAFILE spécification_fichier_data [...];
```



# Structure physique d'une Base de données

## Modifier un tablespace permanent

**Modifier la taille du fichier rattaché au tablespace**

**ALTER DATABASE**

**DATAFILE numéro\_fichier [...] RESIZE valeur [K|M|G|T];**

**ALTER TABLESPACE nom\_tablespace\_bigfile RESIZE valeur [K|M|G|T];**



## Structure physique d'une Base de données

**Modifier un tablespace permanent**

**Passer un tablespace OFFLINE / ONLINE**

**ALTER TABLESPACE nom ONLINE | OFFLINE;**



## Structure physique d'une Base de données

### Modifier un tablespace permanent

déplacer un fichier rattaché au tablespace

- Mettre le tablespace OFFLINE  
`ALTER TABLESPACE nom_ts OFFLINE ;`
- Déplacer les fichiers vers le nouvel emplacement  
`COPIER .. +.. COLLER`
- Indiquer à Oracle le nouvel emplacement  
`ALTER DATABASE RENAME FILE ancien_nom TO nouveau_nom ;`
- Mettre le tablespace ONLINE  
`ALTER TABLESPACE nom_ts ONLINE ;`



## Structure physique d'une Base de données

### Supprimer un tablespace permanent

```
DROP TABLESPACE nom [ INCLUDING CONTENTS [ AND  
DATAFILES ]  
[ CASCADE CONSTRAINTS ] ]
```



## Structure physique d'une Base de données

### Supprimer un fichier d'un tablespace

- Supprimer un fichier de données d'un tablespace s'effectue avec l'ordre SQL ALTER TABLESPACE.
- Le fichier de données doit être vide (ne doit contenir aucune extension) ;
- Le fichier de données ne peut pas être le premier fichier créé pour le tablespace
- Le fichier de données ne doit pas appartenir à un tablespace en lecture seule ;
- Le fichier de données doit être en ligne (ONLINE) ;
- Le fichier ne doit pas appartenir au tablespace SYSTEM.

```
ALTER TABLESPACE nom DROP DATAFILE 'nom_complet';
```



## Structure physique d'une Base de données

### Tablespace UNDO

Le tablespace UNDO, comme son nom l'indique, est réservé exclusivement à l'annulation des commandes DML (UPDATE, INSERT, etc...).

#### Création du tablespace UNDO

```
CREATE UNDO TABLESPACE undotbs  
DATAFILE 'g:\oracle\ora\undotbs.dbf' size 100M;
```

#### Modification du tablespace UNDO utilisé

```
ALTER SYSTEM  
SET UNDO_TABLESPACE=undotbs;
```



## Structure physique d'une Base de données

### Tablespace Temporaire

Le tablespace temporaire héberge les segments temporaires issus des requêtes ou des commandes SQL telles que :

- Select ... order by , Select ... group by , Select distinct ...
- Les requêtes ensemblistes (UNION, MINUS, INTERSECT)
- Les jointures ( JOIN)

#### Création du tablespace temporaire

```
CREATE [ BIGFILE | SMALLFILE ] TEMPORARY TABLESPACE Ts_nom  
TEMPFILE 'nom_fichier' [ SIZE valeur [K|M|G|T] ] [REUSE]  
AUTOEXTEND { OFF|ON [ NEXT valeur [K|M|M|T] ] [ MAXSIZE  
{ UNLIMITED | valeur [K|M|G|T] } ]  
[EXTENT MANAGEMENT LOCAL] [UNIFORM [SIZE valeur [K|M] ] ]
```



# Structure physique d'une Base de données

## Vues du dictionnaire de données

Plusieurs vues du dictionnaire permettent d'obtenir des informations sur les tablespaces et les fichiers de données :

- **DBA\_TABLESPACES** : informations sur les tablespaces
- **V\$TABLESPACE** : informations sur les tablespaces (à partir du fichier de contrôle)
- **DBA\_TEMP\_FILES**: informations sur les fichiers de données des tablespaces temporaires gérés localement.
- **V\$TEMPFILE**: informations sur les fichiers de données des tablespaces temporaires gérés localement.



## Structure physique d'une Base de données

### Allocation d'un tablespace à un objet

```
CREATE TABLE client ( Id_client NUMBER(6), nom VARCHAR2(20) )  
TABLESPACE data ;
```

