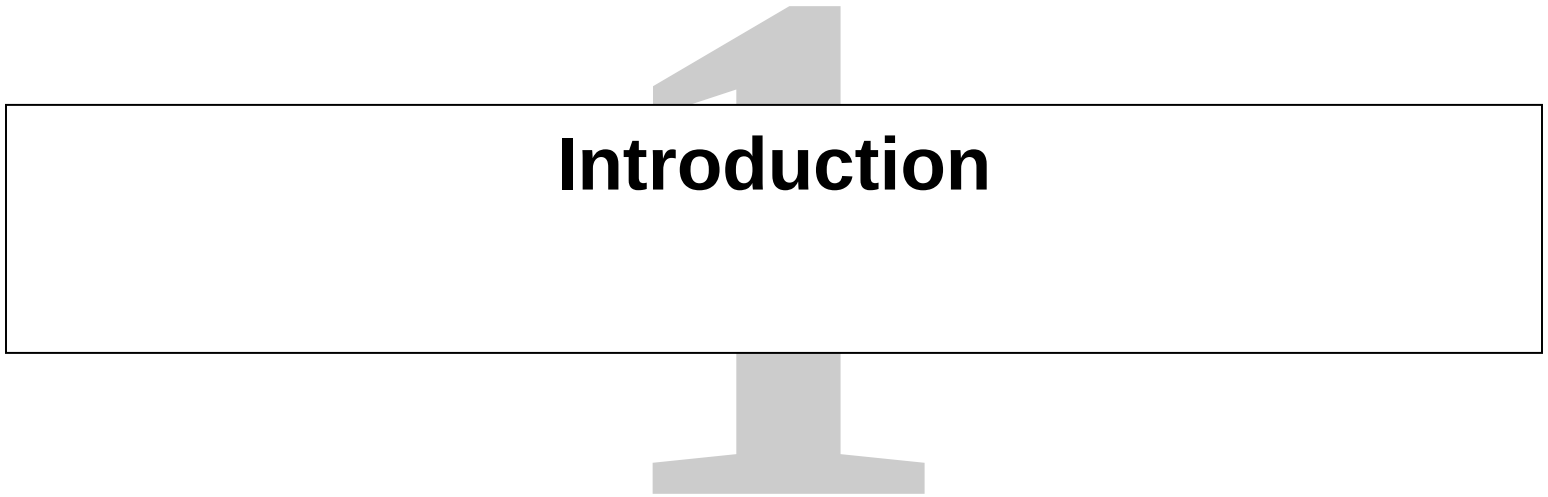


# Architecture du SGBD

**Un serveur Oracle :**

- **Est un système de gestion de base de données qui gère les informations de manière**
  - **Ouverte**
  - **Complète**
  - **Intégrée**
- **Consiste en une instance Oracle et une base de données Oracle**
- **Instance : Ensemble de structures mémoire de la base en cours d'exécution**
- **Base de données : Ensemble de fichiers**

# Objets de la base de données

**Table : Forme de stockage basique. Ue table a des colonnes et stocke des lignes de données**

**Vue : C'est une requête stockée. Elle ne nécessite pas d'espace de stockage**

**Index : Structure optionnelle utile pour trouver les données plus rapidement**

**Vue matérialisée : Comme les vues mais stockées sur disque pour accélerer les requêtes**

**Cluster : Groupe de tables qui partagent les mêmes blocs de stockage**

**Contrainte : Règle stockée pour renforcer les contraintes d'intégrité**

**Séquence : Mécanisme de génération de nombres de manière continue**

**Synonyme : Alias d'un schéma de la base**

**Trigger : Programme en PL/SQL exécuté quand un évènement a lieu**

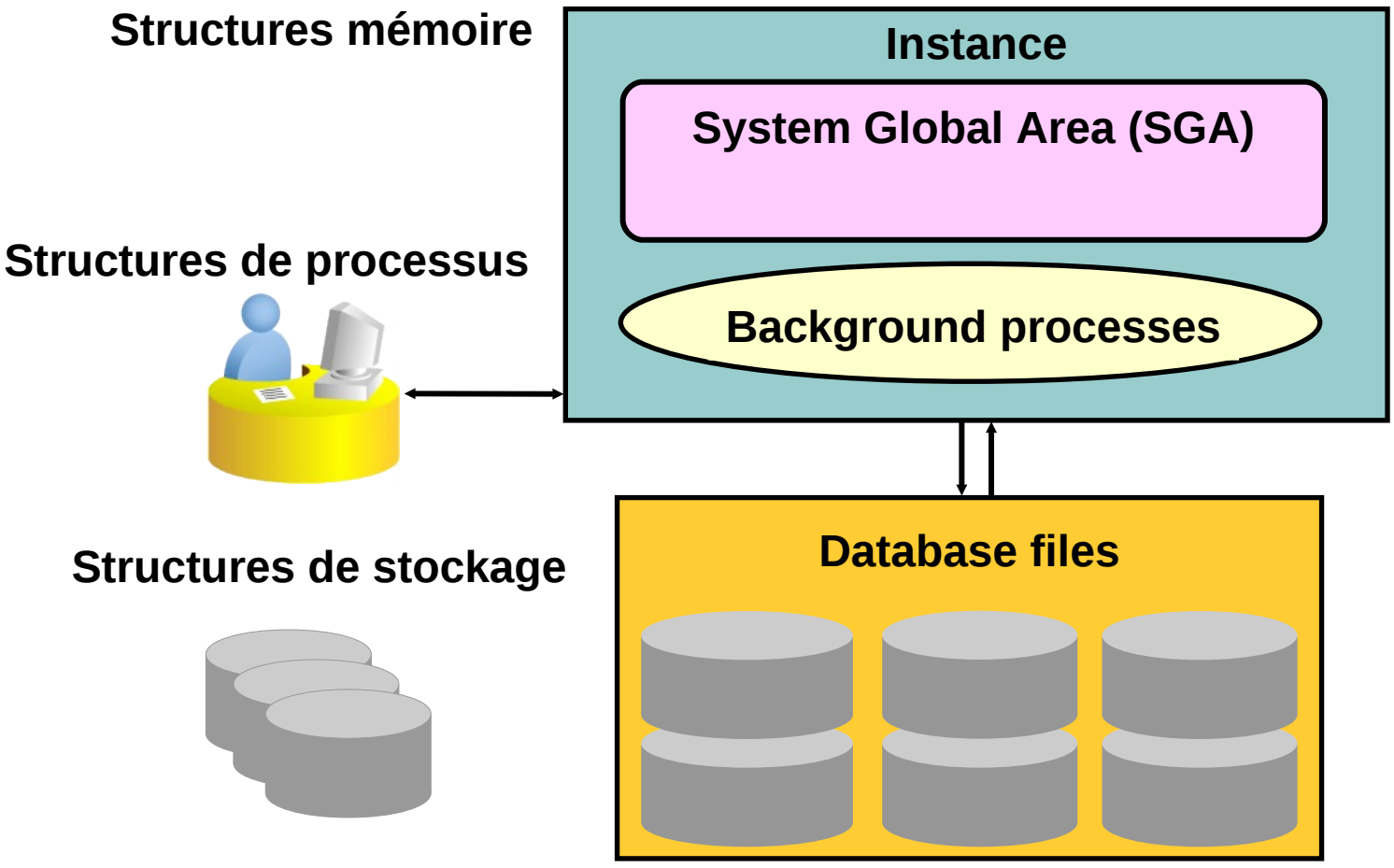
**Procédure stockée : Programme PL/SQL utilisé pour créer des fonctions personnalisées**

**Package : Ensemble de procédures, fonctions et programmes**

**Lien de base de données: utilisés pour communiquer entre les bases de données pour partager les données**

# Structures de la base de données

**Structures BD**
- Mémoire
- Processus
- Stockage

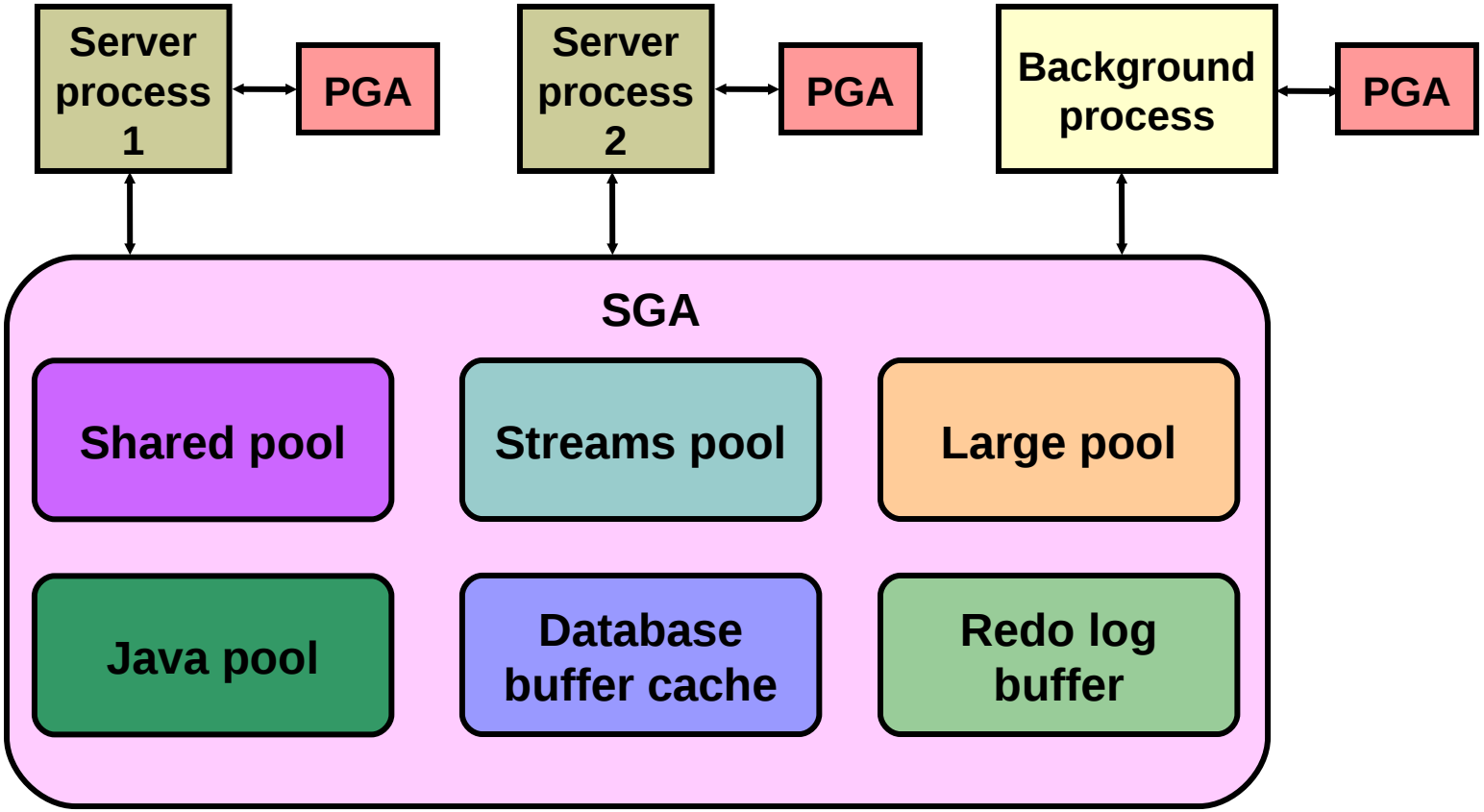


# Oracle Memory Structures

DB structures
>
Memory

Process

Storage

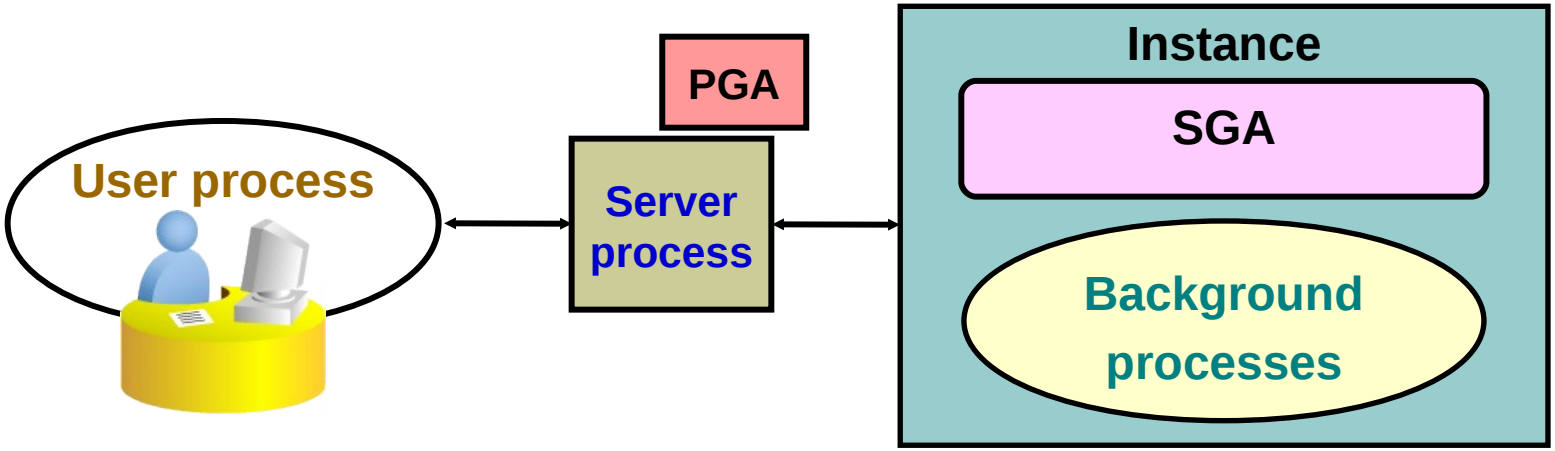


# Description des zones mémoire

- **Program Global Area (PGA):**
  - Zone privée pour chaque serveur et processus d'arrière-plan, Chaque processus a une PGA
- **System Global Area (SGA):**
  - Zone partagée par tous les serveurs et les processus en arrière-plan
  - Contient les données et les informations d contrôle de l'instance
- **La SGA contient les structures suivantes:**
  - **Database buffer cache:** Blocs de cache contentant les données de la base
  - **Redo log buffer:** Cache des information d'annulation (redo) utilisé pour la récupération de l'instance. Le buffer est ensuite écrit dans les fichiers redo sur le disque
  - **Shared pool:** cache les métadonnées et les dernières requêtes SQL des utilisateurs
  - **Large pool:** zone optionnell, Permet d'allouer de grandes zones mémoire pour les opérations de récupération et de sauvegarde
  - **Java pool:** Contient un cache des derniers objets JAVA exécutés dans la JVM de Oracle
  - **Streams pool:** Utilisée par Oracle Streams pour les opérations de file d'attente avancées

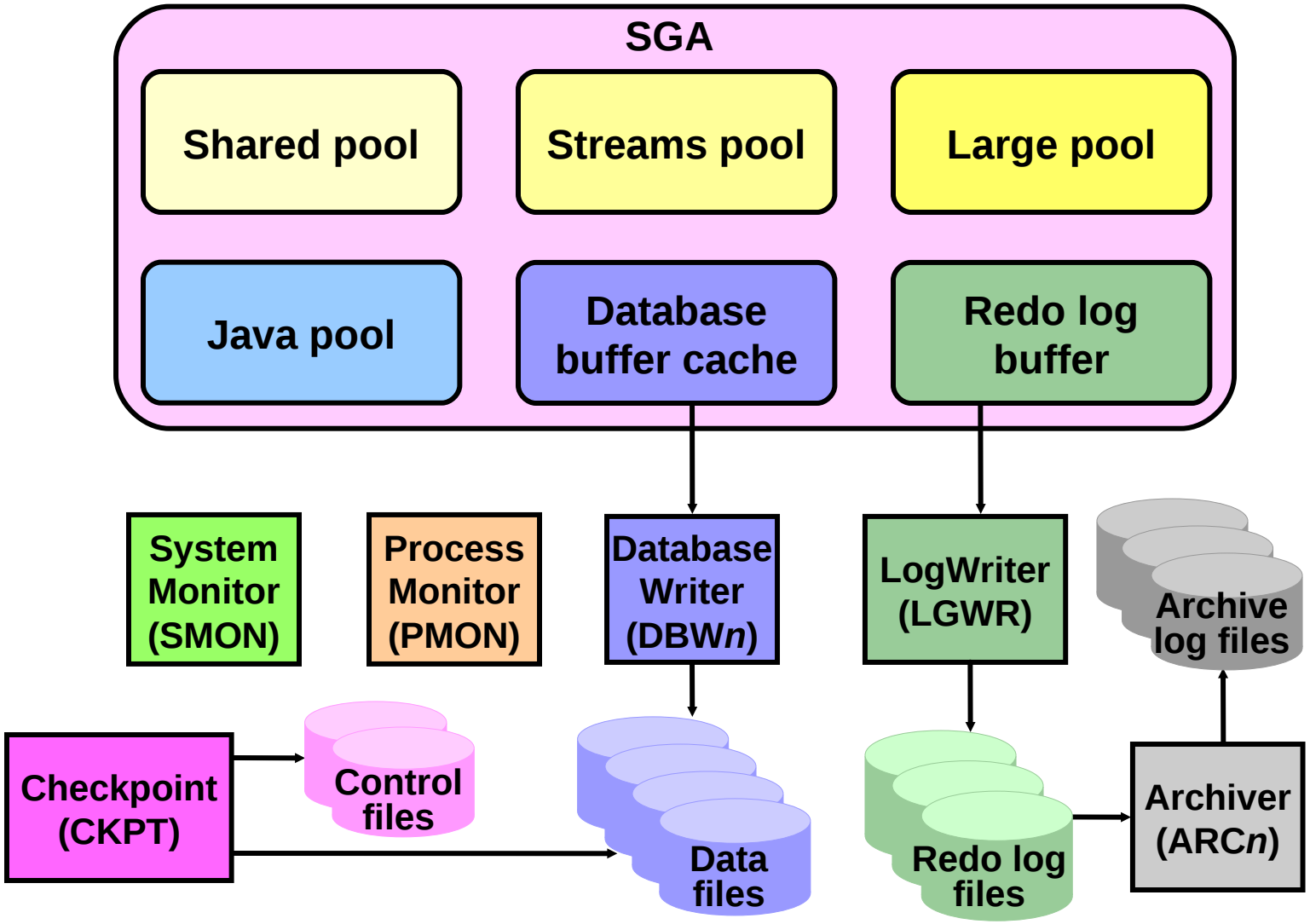
# Structures des processus

DB structures
Memory
> Process
Storage




- **User process:** Démarré quand un utilisateur demande une connexion au serveur Oracle
- **Server process:** Se connecte à l'instance Oracle et démarre quand un utilisteur établit une session
- **Background processes:** sont démarrés quand l'instance Oracle démarre

# Management d'une instance Oracle

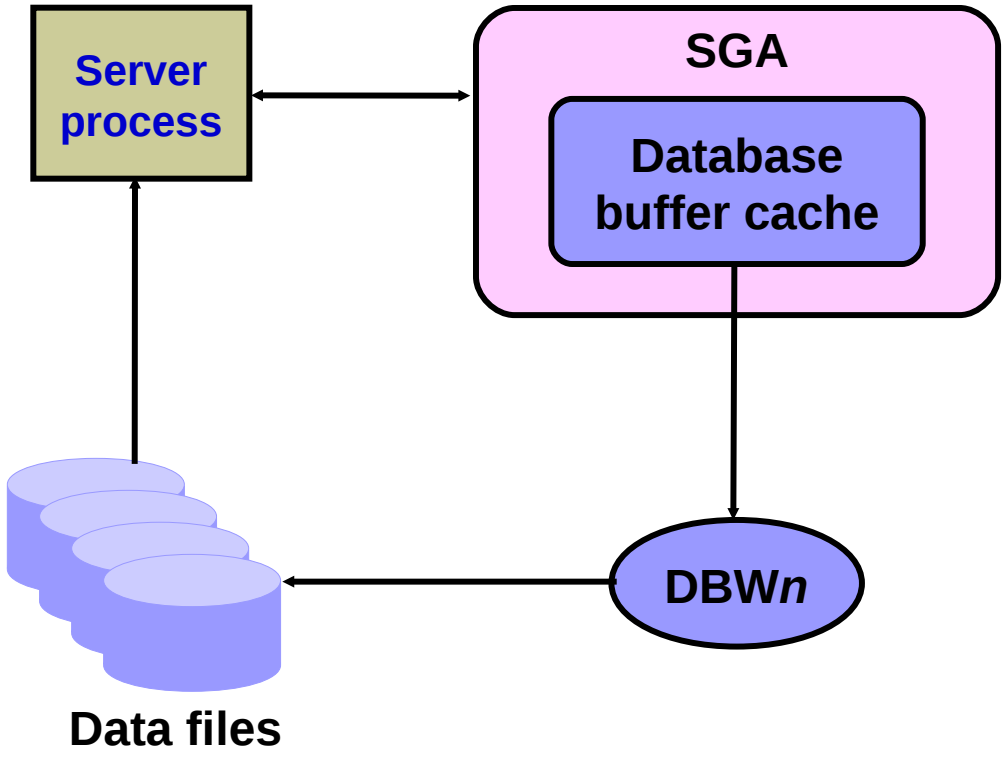


# Processus d’arrière-plan

- **System Monitor (SMON):** Effectue la récupération de pannes après un crash de la base
- **Process Monitor (PMON):** Effectue le nettoyage et la récupération des ressources quand un processus termine
- **Database Writer (DBW*n*):** Écrit les blocs de données modifiés du buffer cache vers les fichiers de données sur le disque
- **Checkpoint (CKPT):** Met à jour les fichiers de données et de contrôle pour indiquer le dernier checkpoint
- **LogWriter (LGWR):** Écrit les entrées redo log dans le disque
- **Archiver (ARC*n*):** Copie les fichiers redo log dans l’archive quand le fichier log est plein

- Beaucoup d’autres processus : RECO, CJQn, QMNn, DIAG, EMNC, MMAN, Snnn, etc.

# Server Process et Buffer Cache




**Etat du buffer:**

- Marqué
- Propre
- Libre
- Sale

# États des blocs du buffer

**Marqué (Pinned): Plusieurs sessions sont interdites d'écrire dans le même bloc simultanément. D'autres sessions attendent l'accès au bloc**

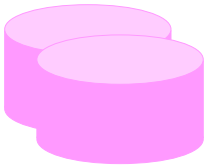
**Propre (Clean): Le buffer n'est pas marqué. Le contenu est synchronisé avec la version sur le disque ou bien l'accès est en lecture consistante**

**Libre (Free): Le buffer est libre parce que l'instance vient de démarrer.**

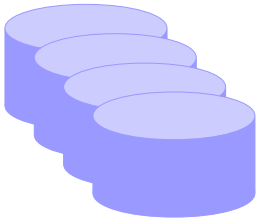
**Sale (Dirty): Le buffer n'est pas marqué mais les blocs ont changé et doivent être écrits sur le disque**

# Structure physique de la base

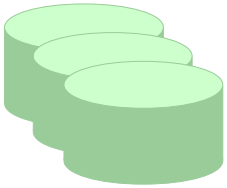
DB structures
Memory
Process
> Storage



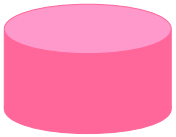
Control files



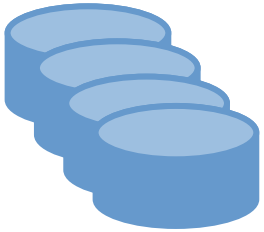
Data files



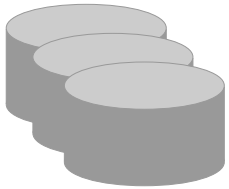
Online redo log files



Parameter file

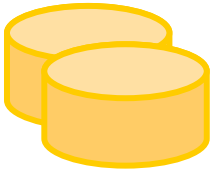


Backup files



Archive log files

Password file



Alert and trace log files

# Fichiers de la base de données

**Fichiers de contrôle:**

- **Contiennent des données sur la base (structure physique de la base)**
- **Sont des fichiers critiques. Sans les fichiers de contrôle l'accès à la base est impossible**

**Fichiers de données: Contiennent les données des utilisateurs et des applications**

**Fichiers redo log:**

- **Permettent la récupération de la base**
- **Après une panne, on peut récupérer les données non sauvegardées grâce à ces fichiers**

**Fichier Paramètre : Utilisé pour définir les paramètres de démarrage de la base**

**Fichier Password : Permet aux utilisateurs de se connecter à la base et faire les tâches administratives**

**Fichiers de sauvegarde: Utilisés pour la récupération de la base.**

**Fichiers d'archives: Contient tout l'historique des changements effectués sur la base (redo). Avec l'archive et la sauvegarde, on peu restaurer la base si un fichier de données est perdu.**

# … Fichiers de la base de données

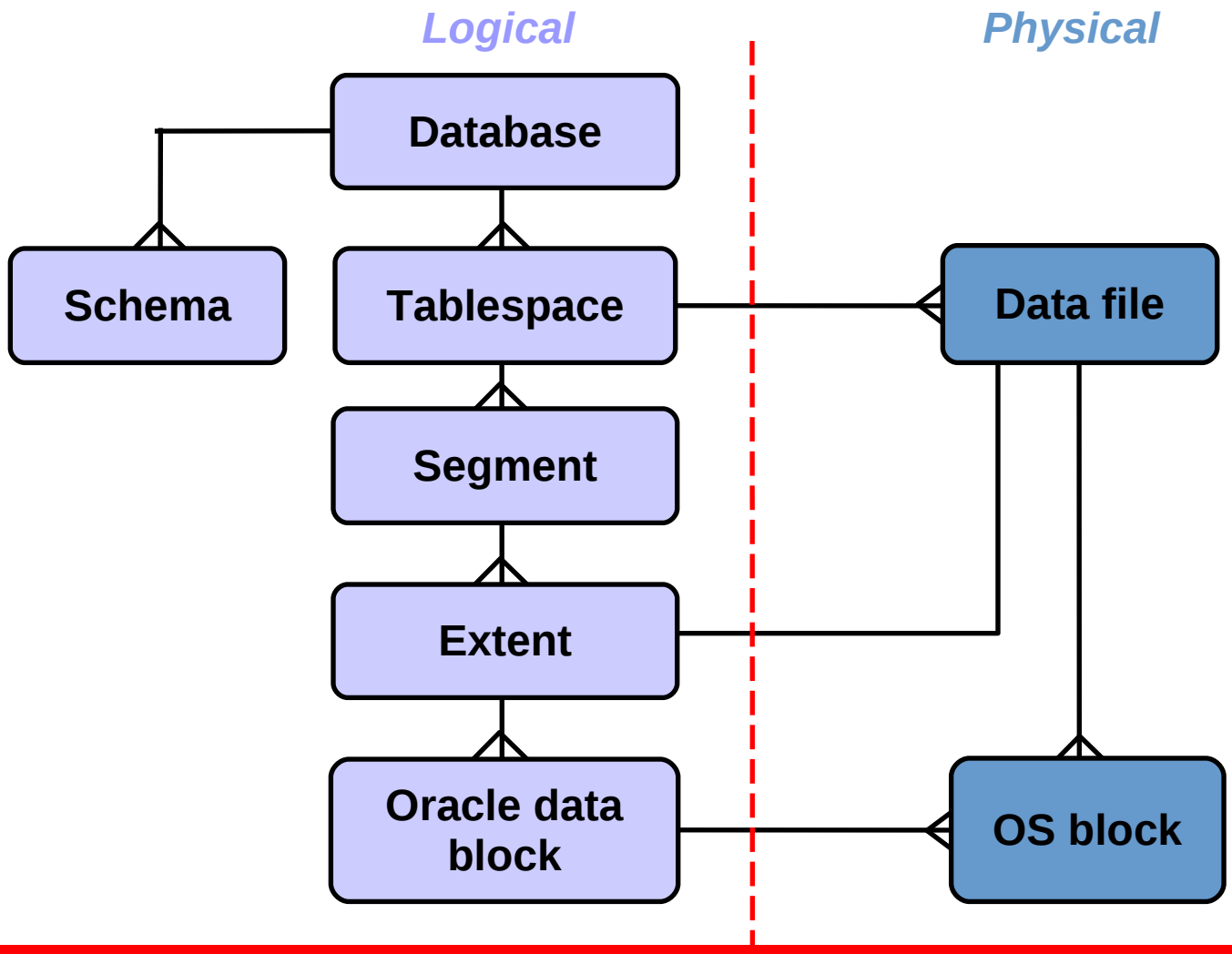
**Fichiers de trace:**

- **Chaque processus d'arrière-plan ou processus serveur peut écrire dans son fichier de trace.**
- **Quand un processus détecte une erreur interne, l'information sur l'erreur est écrite dans le fichier de trace pour l'administrateur de la base**

**Fichiers logs d'alerte:**

- **Ce sont des fichiers de trace spéciaux**
- **C'est un journal qui contient les messages et les erreurs de la base.**
- **L'administrateur doit consulter régulièrement ces fichiers.**

# Structures logiques et physiques de la base

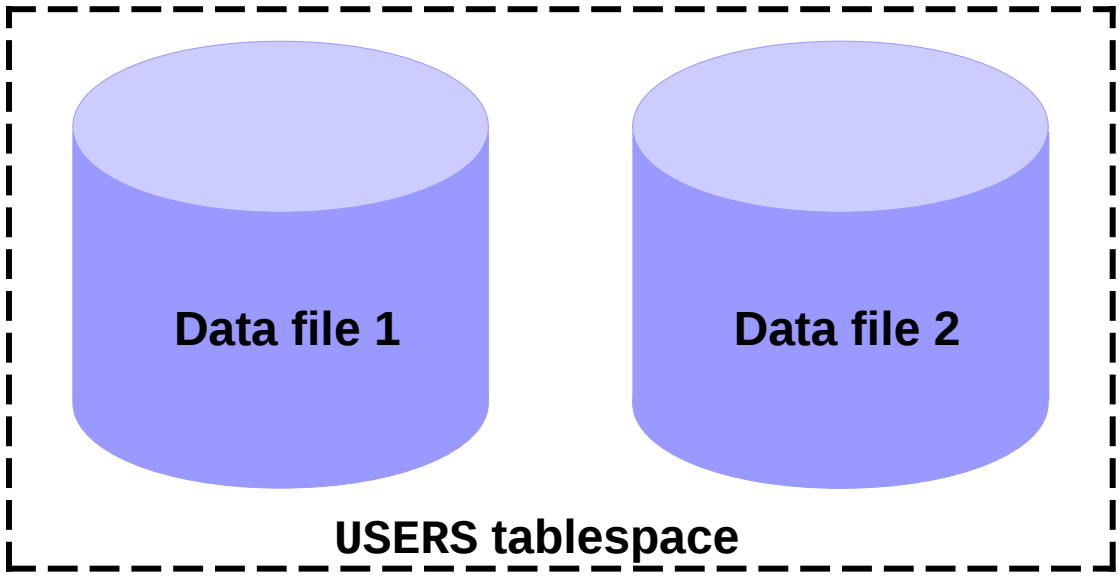


# Structure logique de la base de données

- La base de données est divisée en unités de stockage logiques appelées tablespaces, qui peuvent être utilisées pour regrouper des structures logiques liées.
- Exemples :
  - Tablespace de données est logiquement liée au stockage des objets(tables, index, vues…) créés par les utilisateurs.
  - Tablespace Undo est logiquement liée aux données de gestion d'annulation …
- Les objets de base de données (Users, information UNDO, information temporary, tables, index….)sont stockés dans les tablespaces. Chaque tablespace contiendra les objets lui correspondant.
- Un tablespace est constitué de segments.

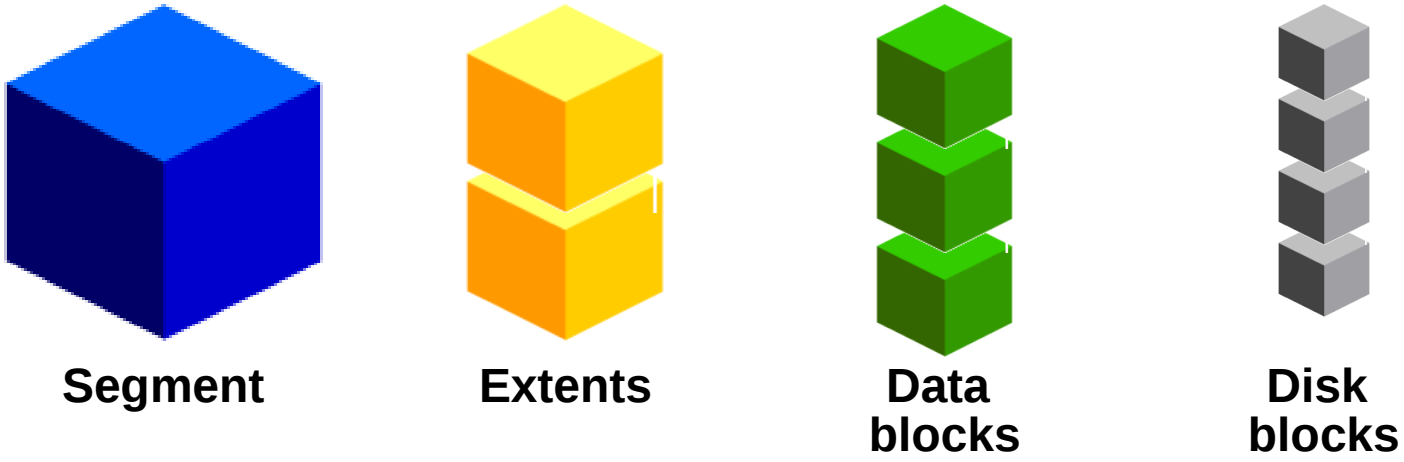
# Tablespaces et fichiers de données

- Un tablespace consiste en un ou plusieurs fichiers de données.
- Un fichier de données appartient à un seul tablespace.



# Segments, Extents, et Blocs

- Un tablespace contient des segments
- Les segments sont des collections d'extents.
- Les extents sont des collections de blocs de données.
- Les blocs de données sont mappés sur des blocs disques



# Les segments

- Un segment appartient à un et un seul tablespace
- Chaque objet de la base est sur un segment différent
- Un Tablespace de données contient les objets créés par les utilisateurs (tables, index, vues). Mais chaque objet se trouve sur un segment différent du tablespace.
- Le segment contient un ou plusieurs extents

# Les extents

- Un extent est un ensemble de blocs contigus du disque
- Un extent est alloué lorsque le segment est:

  Créé, Étendu ou Modifié

- Un extent est libéré lorsque le segment est:

  Supprimé, Modifié, Vidé ou Redimensionné automatiquement

- Mode d'allocation des extents :
  - AUTOALLOCATE: la taille des extents est calculée automatiquement par Oracle
  - UNIFORM: la taille des extents est uniforme

# Les blocs Oracle

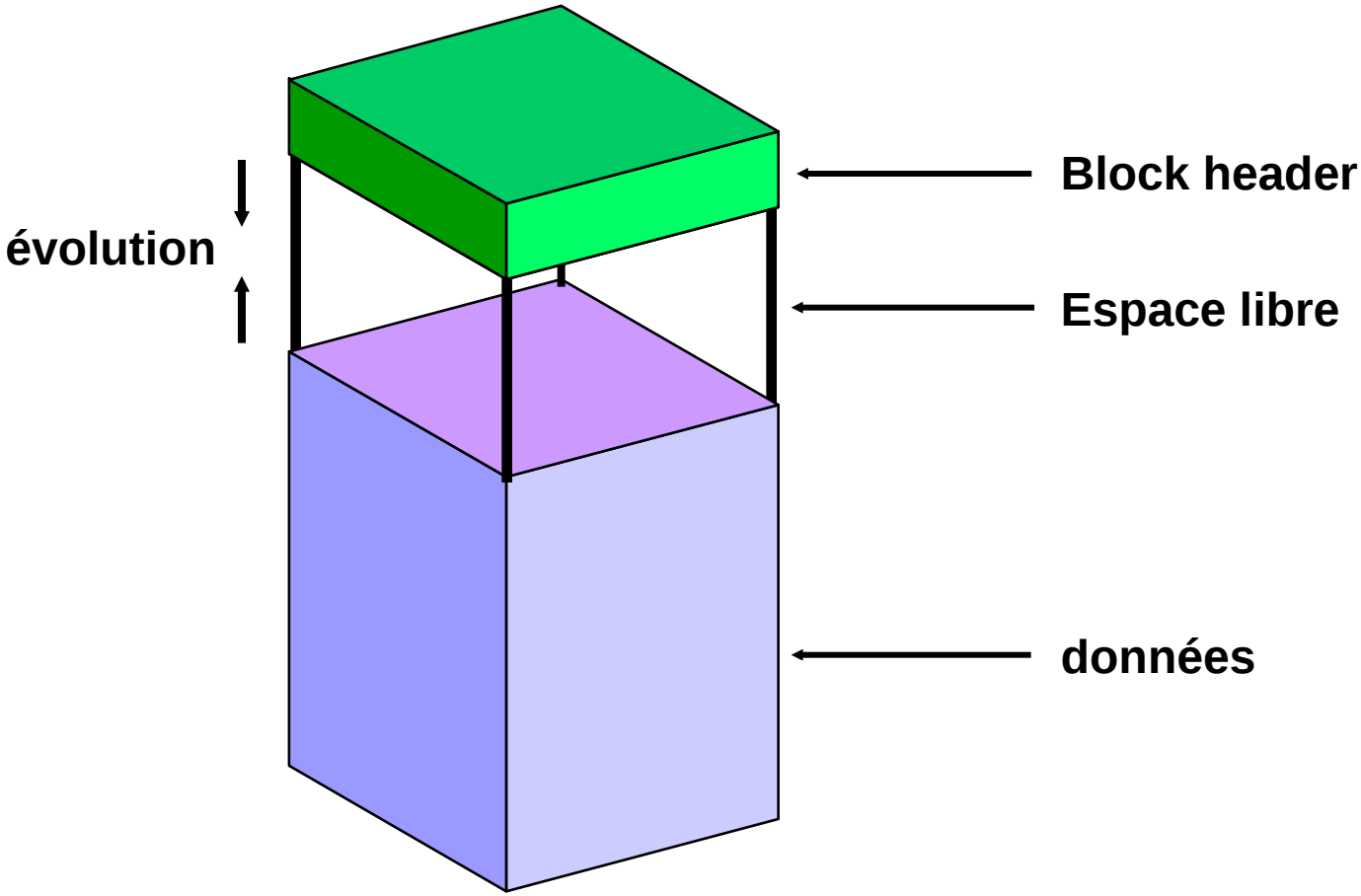
- Le bloc est l'unité d'échange entre données disque et la mémoire Oracle
- SHOW PARAMETER DB_BLOCK_SIZE

- Les tailles de bloc : 2KO, 4KO, 8KO (par défaut), 16kO, 32KO

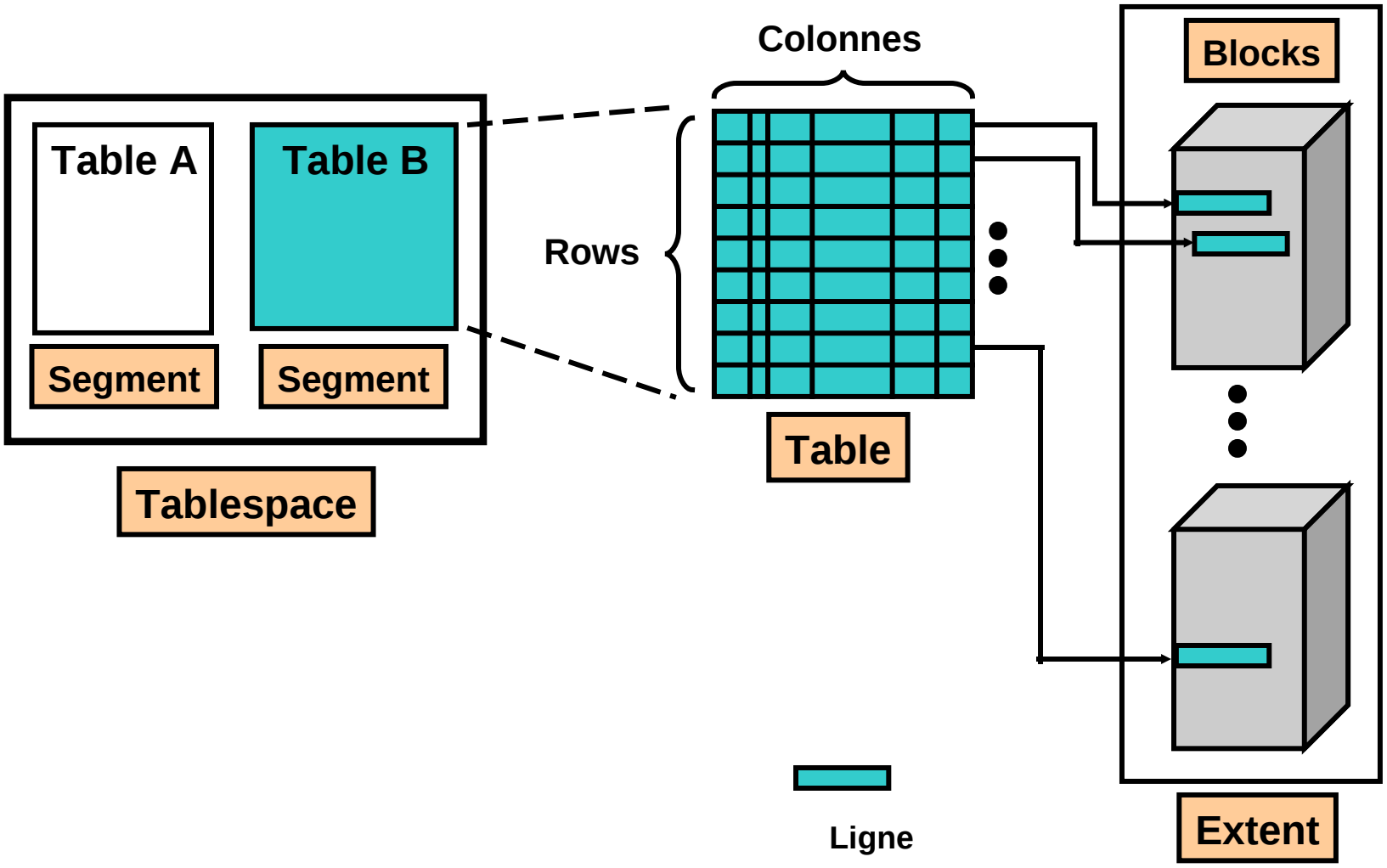
- On distingue entre le bloc logique Oracle et le bloc physique lié au support de stockage et au système d'exploitation (4K pour Windows pour les disques NTFS de moins de 16 TO)

- Discussion : Quelle est la taille de bloc idéale pour une application donnée ?

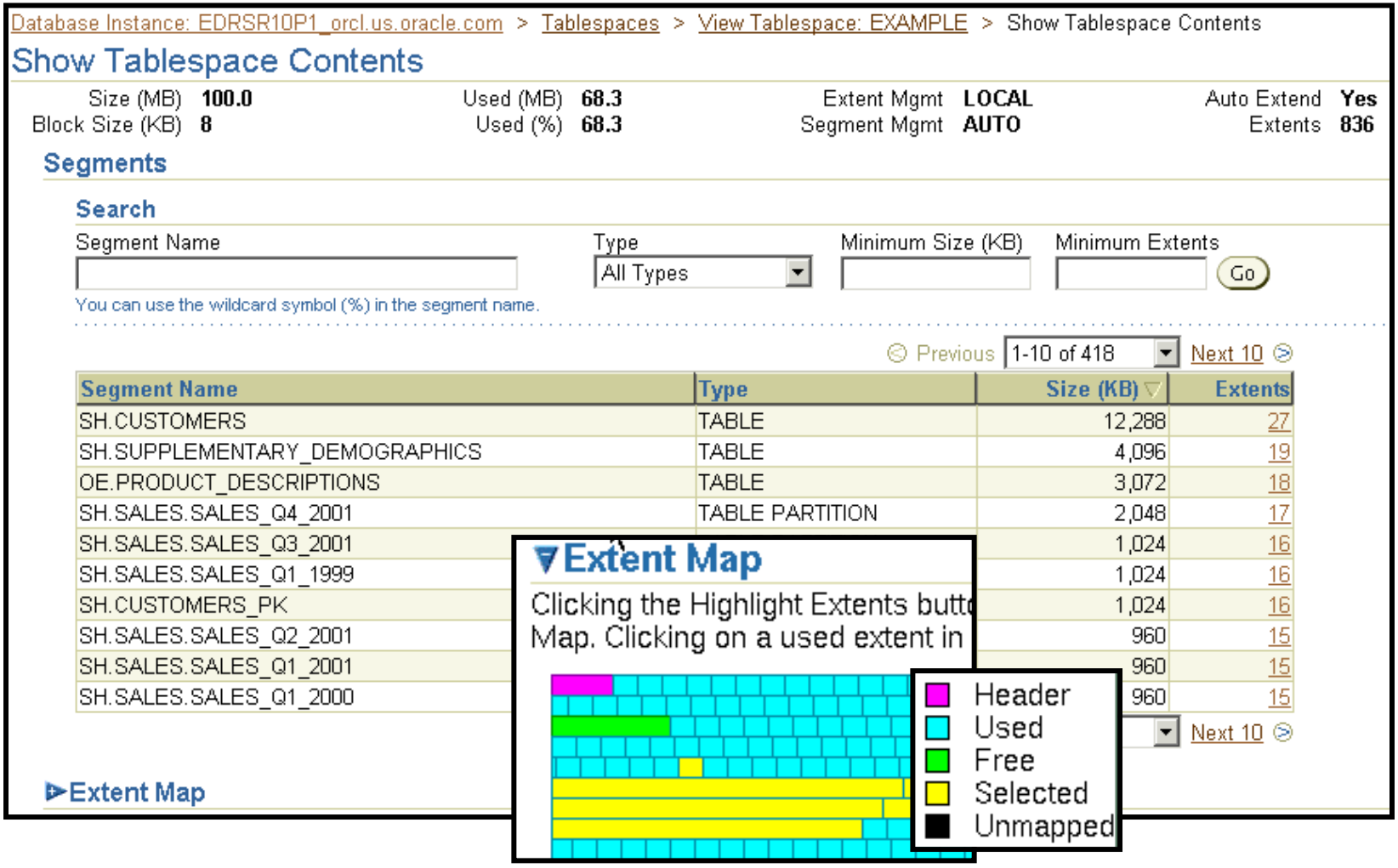
# Anatomie d’un Bloc Oracle



# Comment sont stockées les données



# Voir le contenu des tablespaces

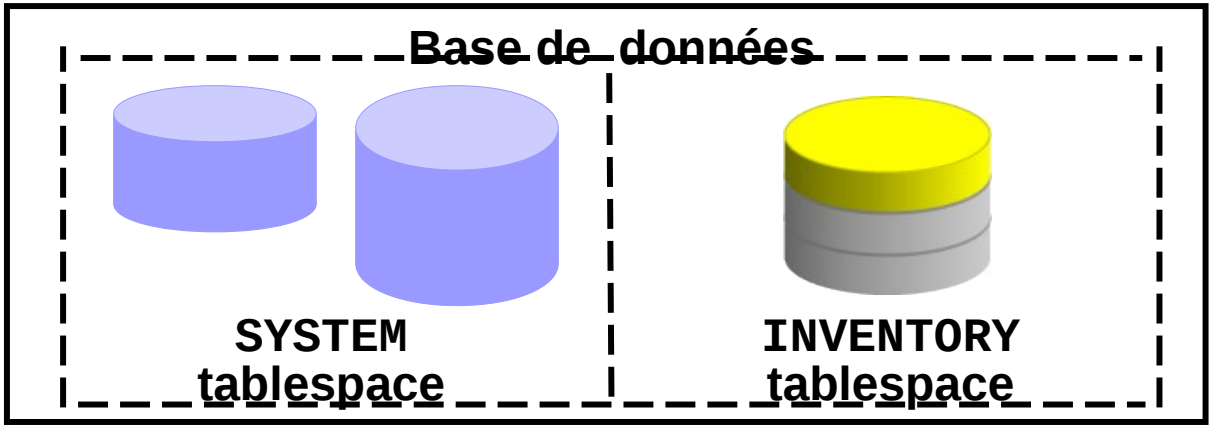


# Élargir la base de données

**On peut élargir la base de données de plusieurs façons :**

- **Créer un nouvveeau tablespace**
- **Ajouter un nouveau fichier à un tablspace**
- **Augmenter la taille du fichier**

**Prévoir à l'avance la croissance de la base de données**



# Questions pour résumer

**Une table peut-elle appartenir à différents tablespace ?**

**Une table peut-elle appartenir à différents segment ?**

**Une table peut-elle appartenir à différents fichiers ?**

**Une table peut-elle appartenir à différents extents ?**

**Une table peut-elle appartenir (ou contenir) différents blocs de données ?**

**Un extent peut-il appartenir à différentes tables ?**

**Un extent peut-il appartenir à différents fichiers ?**

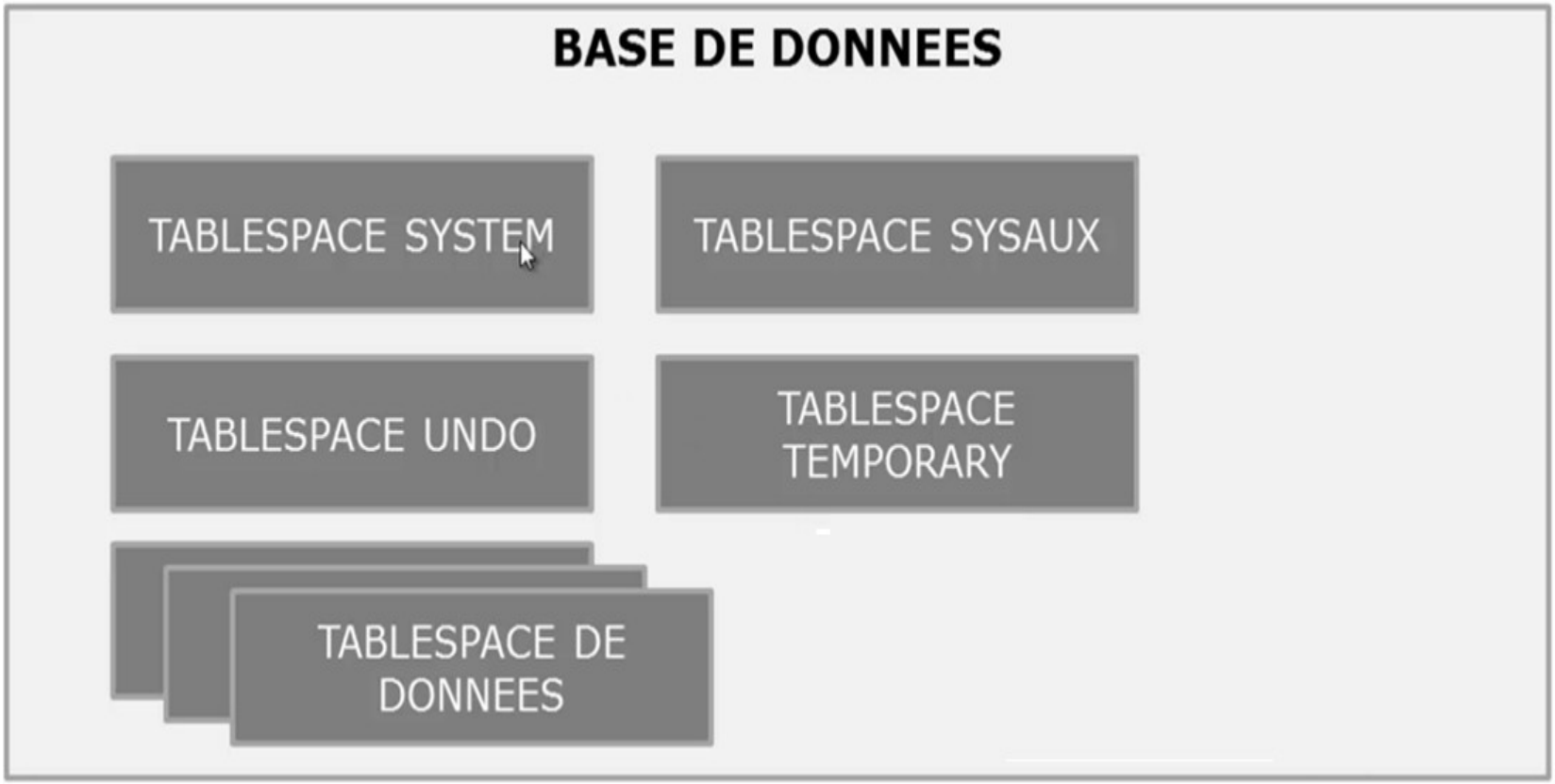
**Un bloc peut-il appartenir à différent extents?**

**Un fichier peut-il contenir les données d'une table et d'une vue et d'autre objet en même temps ?**

**Quand est ce qu'il y a ajout ou suppression d'extent ?**

# Types de tablespace

Les types de tablespace qu'on rencontre en général :



# Tablespaces SYSTEM et SYSAUX

- Les tablespaces SYSTEM et SYSAUX sont nécessaires pour le fonctionnement de la base.
- Créés lors de la création de la base et doivent être online.
- Le tablespace SYSTEM est utilisé pour stocker les données de base (tables du dictionnaire de données, ...).
- Le dictionnaire de données est l'ensemble de tables systèmes contenant les informations relatives à la structure de la base de données :
  - Utilisateurs de la base (ainsi que leurs privilèges et leur rôle)
  - Noms et caractéristiques des objets contenus dans la base (tables, vues, index, clusters, triggers, packages, …)
  - Contraintes d'intégrité
  - Les noms de tablespaces, des fichiers et les taux d'allocation
- Exemple de tables système du dictionnaire de données : dba_segments, dba_tablespace….

# Types de tablespace

- Le tablespace auxiliaire  SYSAUX est utilisé pour des composants additionnels de la base (Enterprise Manager par exemple).

- SYSAUX : Chaque instance de la base de donnée Oracle doit comporter un tablespace SYSAUX. Ce tablespace doit rester en ligne pour faire fonctionner ces composants

# Types de tablespace

- Le tablespace temporaire est défini pour héberger les opérations de tri de données.

Lors d'importantes opérations de tri (select distinct, union et create index), si la taille de la zone PGA du processus utilisateur ne suffit pas, Oracle va stocker dans les tablespaces TEMPORARY de la base de données des informations concernant le tri des enregistrements avant de retourner les données aux utilisateurs.

- Il est recommandé de définir un tablespace temporaire par défaut pour la base de données. Sauf indication contraire, ce tablespace est affecté à chaque utilisateur nouvellement créé.

# Types de tablespace

- On peut créer plusieurs tablespaces temporaires  pour distribuer les espaces de stockages des utilisateurs ayant des besoins semblables sur les mêmes tablespaces.
- Le tablespace « UNDO » : Toutes les données d'annulation (de rollback) sont stockées dans ce tablespace. Se sont toutes les traces de modification faites sur les objets de la base de données.(fichier redoLog…)
- Tablespace de données contient (les lignes des tables, les index, les vues….) Il en existe plusieurs car l'administrateur peut les distribuer sur les classes d'utilisateurs.

# Types de segments

Quatre types de segment sont possibles :

- **Le segment de données** : Dans un tablespace de données, on trouve des segments de données.
- Chaque table de la base porte sur un segment de ce tablespace. Et donc L'ensemble des données de cette table sont stockées dans les extents du segment de données de cette table.
- La commande *create table* crée un segment de données associé à la table dans le tablespace par défaut du compte user.

**Le segment d'index** : L'objet index sont gérés de manière différente que les autres objets.

- Chaque index créé se trouve sur un segment d'index.

# Types de segments

- **Les segments temporaires :** Dans un tablespace temporaire, on trouve les segments temporaires.
- Ils sont créés par la base de données (expl: l'exécution d'une instruction SQL qui requiert une zone de travail)

**Les segments undo**(d'annulation): L'administrateur de base de données crée un tablespace d'annulation (UNDO) pour la base de donnée afin de stocker de manière temporaire les informations d'annulation.

→Chaque type de segment sert à contenir un type de données (données, index, undo, temporary) et cette différence influe sur la manière avec laquelle Oracle va gérer ces segment

→Exemple : parmi ces types de segment quels sont ceux que oracle doit vider et ceux non

# Commandes SQL

1. Création de tablespace permanent ou de données:

```
CREATE [ SMALLFILE | BIGFILE ] TABLESPACE 'tablespace_name'
[ DATAFILE 'file_name' ]
[ SIZE 'value‘ [ K | M | G | T] ]
[ AUTOEXTEND { OFF | ON [ NEXT 'value'  [ K | M | G | T ] ] }
[ MAXSIZE { UNLIMITED | 'value‘ [ K | M | G | T ] } ],
[OTHER DATAFILE…..]
[ LOGGING | NOLOGGING ]
[ ONLINE | OFFLINE| READ ONLY]
[BLOCKSIZE 'value' [ K ]]
```

# Commandes SQL

## 1. Création de tablespace permanent ou de données:

```
CREATE [ SMALLFILE | BIGFILE ] TABLESPACE 'tablespace_name'
[ DATAFILE 'file_name' ]
[ SIZE 'value‘ [ K | M | G | T] ]
[ AUTOEXTEND { OFF | ON [ NEXT 'value'  [ K | M | G | T ] ] }
[ MAXSIZE { UNLIMITED | 'value‘ [ K | M | G | T ] } ],
[OTHER DATAFILE…..]
[ LOGGING | NOLOGGING ]
[ ONLINE | OFFLINE]
[BLOCKSIZE 'value' [ K ]]
```

- Par défaut SMALLFILE
- Le cas SMALLFILE peut contenir plusieurs fichiers.
- Par contre un BIGFILE ne peut contenir qu’un seul fichier.
- Le cas SMALLFILE chaque fichier peut aller dans sa taille jusqu’à(2^22)blocs
- Le cas BIGFILE le fichier peut avoir la taille (2^32)blocs

# Commandes SQL

## 1. Création de tablespace permanent ou de données:

```
CREATE [ SMALLFILE | BIGFILE ] TABLESPACE 'tablespace_name'
[ DATAFILE 'file_name' ]
[ SIZE 'value‘ [ K | M | G | T] ]
[ AUTOEXTEND { OFF | ON [ NEXT 'value'  [ K | M | G | T ] ] }
[ MAXSIZE { UNLIMITED | 'value‘ [ K | M | G | T ] } ],
[OTHER DATAFILE.....]
[ LOGGING | NOLOGGING ]
[ ONLINE | OFFLINE]
[BLOCKSIZE 'value' [ K ]]
```

- Ici  on précise les fichiers qui seront rattaché au tablespace

# Commandes SQL

1. Création de tablespace permanent ou de données:

```
CREATE [ SMALLFILE | BIGFILE ] TABLESPACE 'tablespace_name'
[ DATAFILE 'file_name' ]
[ SIZE 'value‘ [ K | M | G | T] ]
[ AUTOEXTEND { OFF | ON [ NEXT 'value'  [ K | M | G | T ] ] }
[ MAXSIZE { UNLIMITED | 'value‘ [ K | M | G | T ] } ],
[OTHER DATAFILE.....]
[ LOGGING | NOLOGGING ]
[ ONLINE | OFFLINE]
[BLOCKSIZE 'value' [ K ]]
```

- Ici  on précise les fichiers qui seront rattaché au tablespace

# Commandes SQL

1. Création de tablespace permanent ou de données:

```
CREATE [ SMALLFILE | BIGFILE ] TABLESPACE 'tablespace_name'
[ DATAFILE 'file_name' ]
[ SIZE 'value‘ [ K | M | G | T] ]
[ AUTOEXTEND { OFF | ON [ NEXT 'value'  [ K | M | G | T ] ] }
[ MAXSIZE { UNLIMITED | 'value‘ [ K | M | G | T ] } ],
[OTHER DATAFILE…..]
[ LOGGING | NOLOGGING ]
[ ONLINE | OFFLINE]
[BLOCKSIZE 'value' [ K ]]
```

- Ici  on précise si le fichier est en mode incrémentation automatique ou pas.
- Si on active l’option ON alors on doit préciser le pas d’incrémentation (NEXT value)

# Commandes SQL

1. Création de tablespace permanent ou de données:

```
CREATE [ SMALLFILE | BIGFILE ] TABLESPACE 'tablespace_name'
[ DATAFILE 'file_name' ]
[ SIZE 'value‘ [ K | M | G | T] ]
[ AUTOEXTEND { OFF | ON [ NEXT 'value'  [ K | M | G | T ] ] }
[ MAXSIZE { UNLIMITED | 'value‘ [ K | M | G | T ] } ],
[OTHER DATAFILE…..]
[ LOGGING | NOLOGGING ]
[ ONLINE | OFFLINE]
[BLOCKSIZE 'value' [ K ]]
```

- Ici  on peut limiter la taille du fichier (par défaut c’est UNLIMITED). Si on veut limiter la taille on donne une valeur

# Commandes SQL

1. Création de tablespace permanent ou de données:

```
CREATE [ SMALLFILE | BIGFILE ] TABLESPACE 'tablespace_name'
[ DATAFILE 'file_name' ]
[ SIZE 'value‘ [ K | M | G | T] ]
[ AUTOEXTEND { OFF | ON [ NEXT 'value'  [ K | M | G | T ] ] }
[ MAXSIZE { UNLIMITED | 'value‘ [ K | M | G | T ] } ],
[OTHER DATAFILE.....]
[ LOGGING | NOLOGGING ]
[ ONLINE | OFFLINE]
[BLOCKSIZE 'value' [ K ]]
```

- Par défaut LOGGING
- Précise si on veut que les transactions lancées sur les objets de ce tablespace seront enregistrées dans les journaux redolog ou pas

# Commandes SQL

1. Création de tablespace permanent ou de données:

```
CREATE [ SMALLFILE | BIGFILE ] TABLESPACE 'tablespace_name'
[ DATAFILE 'file_name' ]
[ SIZE 'value‘ [ K | M | G | T] ]
[ AUTOEXTEND { OFF | ON [ NEXT 'value'  [ K | M | G | T ] ] }
[ MAXSIZE { UNLIMITED | 'value‘ [ K | M | G | T ] } ],
[OTHER DATAFILE…..]
[ LOGGING | NOLOGGING ]
[ ONLINE | OFFLINE]
[BLOCKSIZE 'value' [ K ]]
```

- Par défaut ONLINE
- Cette option précise si on veut que ce tablespace soit en ligne ou pas. S'il l'est alors on peur créer des objets au sein de ce tablespace. Sinon il est hors ligne la création d'objets dans ce tablespace ne sera pas possible.

# Commandes SQL

## 2. Création de tablespace temporaire:

CREATE [ SMALLFILE | BIGFILE ] TEMPORARY TABLESPACE 'tablespace_name'
[ TEMPFILE 'file_name' ]
[ SIZE 'value‘ [ K | M | G | T] ]
[ AUTOEXTEND { OFF | ON [ NEXT 'value' [ K | M | G | T ] ] }
[ MAXSIZE { UNLIMITED | 'value‘ [ K | M | G | T ] } ],
[OTHER DATAFILE…..]

# Commandes SQL

## 3. Création de tablespace UNDO:

CREATE [ SMALLFILE | BIGFILE ] UNDO TABLESPACE 'tablespace_name'
[ DATAFILE 'file_name' ]
[ SIZE 'value‘ [ K | M | G | T] ]
[ AUTOEXTEND { OFF | ON [ NEXT 'value' [ K | M | G | T ] ] }
[ MAXSIZE { UNLIMITED | 'value‘ [ K | M | G | T ] } ],
[OTHER DATAFILE…..]

# Commandes SQL

## VUES UTILES POUR L'ADMINISTRATION DES TABLESPACE:

1. dba_data_files (FILE_NAME, #TABLESPACE_NAME, BLOCKS, AUTOEXTENSIBLE, MAXBLOCKS, INCREMENT_BY) Affiche tous les fichiers de la base de données

2. dba_tablespaces (TABLESPACE_NAME, STATUS(ONLINE,OFFLINE,READ ONLY), LOGGING(LOGGING, NOLOGGING), BIGFILE(YES, NO)) Affiche tous les tablespaces de la base de données

3. user_tablespaces contient les mêmes champs que la vue précédente mais affiche les tablespaces accessibles par l'utilisateur courant

# Commandes SQL

## MODIFICATION ET SUPRESSION DE TABLESPACE:

1. ALTER TABLESPACE ‘TABLESPACE_NAME’ [OFFLINE|ONLINE|READ ONLY]

2. DROP TABLESPACE ‘TABLESPACE_NAME’ INCLUDING CONTENTS AND DATAFILES
   Cette commande supprime le talespace avec tout ce qui est lié

3. ALTER DATABASE DATAFILE ‘FILE_NAME’ [ONLINE|OFFLINE]
   Cette commande rend le fichier en question accessible ou pas

4. ALTER DATABASE DATAFILE ‘FILE_NAME’ resize value
   cette commande donne une nouvelle valeur pour la taille d’un fichier

5. ALTER TABLESPACE ‘TABLESPACE_NAME’ ADD DATAFILE ‘FILE_NAME’ SIZE ‘VALUE’
   Cette commande ajoute un fichier au tablespace (ceci impose que ce dernier soit un SMALLFILE)

# Commandes SQL

## MODIFICATION  ET SUPRESSION DE TABLESPACE:

1. ALTER DATABASE DATAFILE ‘FILE_NAME’ AUTOEXTEND [OFF|ON NEXT VALUE MAXSIZE VALUE]
2. ALTER TABLESPACE ‘TABLESPACE_NAME’ RESIZE ‘VALUE’
   Cette commande redimensionne le fichier du tablespace
3. ATER TABLESPACE ‘TABLESPACE_NAME’ AUTOEXTEND ON NEXT ‘VALUE’
   Cette commande rend le fichier En autoextend on
4. ATER TABLESPACE ‘TABLESPACE_NAME’ AUTOEXTEND OFF
   Cette commande rend le fichier En autoextend off

# Commandes SQL

## AUGMENTER LA TAILLE DE TABLESPACE:

1. Ajouter un fichier au tablespace en question s'il est SMALLFILE

```
ALTER TABLESPACE tbs ADD DATAFILE '/oracle/oradata5/fich.dbf' SIZE 2G
```

2. Augmenter la taille du fichier déjà utilisé par le tablespace en question

```
ALTER DATABASE DATAFILE '/oracle/oradata2/fich.dbf' resize 5G;
```

3. Activer l'option autoextend ON

```
ALTER DATABASE DATAFILE '/oracle/oradata2/fich.dbf' AUTOEXTEND ON
    NEXT 512M MAXSIZE 5G
```

# SCENARIOS DE TEST

Affichage des tablespaces de cette base:

```
SQL> select tablespace_name, status, logging, bigfile from dba_tablespaces;

TABLESPACE_NAME                STATUS    LOGGING   BIG
------------------------------ --------- --------- ---
SYSTEM                         ONLINE    LOGGING   NO
SYSAUX                         ONLINE    LOGGING   NO
UNDOTBS1                       ONLINE    LOGGING   NO
TEMP                           ONLINE    NOLOGGING NO
USERS                          ONLINE    LOGGING   NO

SQL>
```

# SCENARIO DE TEST

Affichage des fichiers correspondant à ces tablespaces:

```
SQL> select tablespace_name, file_name from dba_data_files;

TABLESPACE_NAME
------------------------------
FILE_NAME
--------------------------------------------------------------------------------

USERS
C:\ORACLEXE\APP\ORACLE\ORADATA\XE\USERS.DBF

SYSAUX
C:\ORACLEXE\APP\ORACLE\ORADATA\XE\SYSAUX.DBF

UNDOTBS1
C:\ORACLEXE\APP\ORACLE\ORADATA\XE\UNDOTBS1.DBF

SYSTEM
C:\ORACLEXE\APP\ORACLE\ORADATA\XE\SYSTEM.DBF
```

# SCENARIO DE TEST

## Ajout de tablespaces :

```
SQL> create tablespace app_test datafile 'C:/ORACLEXE/APP/ORACLE/ORADATA/XE/app_
test.dbf' size 512M autoextend off;

Tablespace created.
```

## Vérification:

```
SQL> select tablespace_name, file_name,bytes/1024/1024 from dba_data_files;

TABLESPACE_NAME
------------------------------
FILE_NAME
--------------------------------------------------------------------------------

BYTES/1024/1024
---------------
USERS
C:\ORACLEXE\APP\ORACLE\ORADATA\XE\USERS.DBF
            100

SYSAUX
C:\ORACLEXE\APP\ORACLE\ORADATA\XE\SYSAUX.DBF
            660

TABLESPACE_NAME
------------------------------
FILE_NAME
--------------------------------------------------------------------------------

BYTES/1024/1024
---------------

UNDOTBS1
C:\ORACLEXE\APP\ORACLE\ORADATA\XE\UNDOTBS1.DBF
            380

SYSTEM
C:\ORACLEXE\APP\ORACLE\ORADATA\XE\SYSTEM.DBF

TABLESPACE_NAME
------------------------------
FILE_NAME
--------------------------------------------------------------------------------

BYTES/1024/1024
---------------
            360

APP_TEST
C:\ORACLEXE\APP\ORACLE\ORADATA\XE\APP_TEST.DBF
            512
```

# SCENARIO DE TEST

## Ajout de tablespaces BIGFILE :

```
SQL> create BIGFILE  tablespace app_big datafile 'C:/ORACLEXE/APP/ORACLE/ORADATA
/XE/app_big.dbf' size 512M autoextend off;

Tablespace created.
```

## Ajout de fichiers pour les deux tablespace:

```
SQL> alter tablespace app_test add datafile 'C:/ORACLEXE/APP/ORACLE/ORADATA/XE/A
PP_TEST2.DBF' size 128M;

Tablespace altered.

SQL> alter tablespace app_big add datafile 'C:/ORACLEXE/APP/ORACLE/ORADATA/XE/AP
P_BIG2.DBF' size 128M;
alter tablespace app_big add datafile 'C:/ORACLEXE/APP/ORACLE/ORADATA/XE/APP_BIG
2.DBF' size 128M
*
ERROR at line 1:
ORA-32771: cannot add file to bigfile tablespace
```

# SCENARIO DE TEST

## Vérifier l'état des tablespaces :

```
SQL> select tablespace_name,  bigfile from dba_tablespaces;

TABLESPACE_NAME                BIG
------------------------------ ---
SYSTEM                         NO
SYSAUX                         NO
UNDOTBS1                       NO
TEMP                           NO
USERS                          NO
APP_TEST                       NO
APP_BIG                        YES

7 rows selected.
```

## Redimensionner la taille d'un fichier:

```
SQL> alter database datafile 'C:/ORACLEXE/APP/ORACLE/ORADATA/XE/APP_TEST2.DBF' r
esize 800M;

Database altered.

SQL> select  file_name,bytes/1024/1024 from dba_data_files;

FILE_NAME
--------------------------------------------------------------------------------

BYTES/1024/1024
---------------
C:\ORACLEXE\APP\ORACLE\ORADATA\XE\USERS.DBF
             100

C:\ORACLEXE\APP\ORACLE\ORADATA\XE\SYSAUX.DBF
             660

C:\ORACLEXE\APP\ORACLE\ORADATA\XE\UNDOTBS1.DBF
             380


FILE_NAME
--------------------------------------------------------------------------------

BYTES/1024/1024
---------------
C:\ORACLEXE\APP\ORACLE\ORADATA\XE\SYSTEM.DBF
             360

C:\ORACLEXE\APP\ORACLE\ORADATA\XE\APP_TEST.DBF
             512

C:\ORACLEXE\APP\ORACLE\ORADATA\XE\APP_BIG.DBF
             512


FILE_NAME
--------------------------------------------------------------------------------

BYTES/1024/1024
---------------
C:\ORACLEXE\APP\ORACLE\ORADATA\XE\APP_TEST2.DBF
             800


7 rows selected.
```

# SCENARIO DE TEST

## Vérifier le statut des tablespaces :

```
SQL> select tablespace_name, status from dba_tablespaces;

TABLESPACE_NAME                STATUS
------------------------------ ---------
SYSTEM                         ONLINE
SYSAUX                         ONLINE
UNDOTBS1                       ONLINE
TEMP                           ONLINE
USERS                          ONLINE
APP_TEST                       ONLINE
APP_BIG                        ONLINE

7 rows selected.
```

## Changer le statut d'un tablespace à offline:

```
SQL> alter tablespace APP_TEST OFFLINE;

Tablespace altered.

SQL> create table test(chmp int) tablespace APP_TEST;
create table test(chmp int) tablespace APP_TEST
*
ERROR at line 1:
ORA-01542: tablespace 'APP_TEST' is offline, cannot allocate space in it
```

# SCENARIO DE TEST

## Effet du statut READ ONLY :

```
SQL> create table test(chmp int) tablespace APP_BIG;

Table created.

SQL> alter tablespace APP_BIG read only;

Tablespace altered.

SQL> insert into test values (15);
insert into test values (15)
            *
ERROR at line 1:
ORA-00372: file 6 cannot be modified at this time
ORA-01110: data file 6: 'C:\ORACLEXE\APP\ORACLE\ORADATA\XE\APP_BIG.DBF'
```