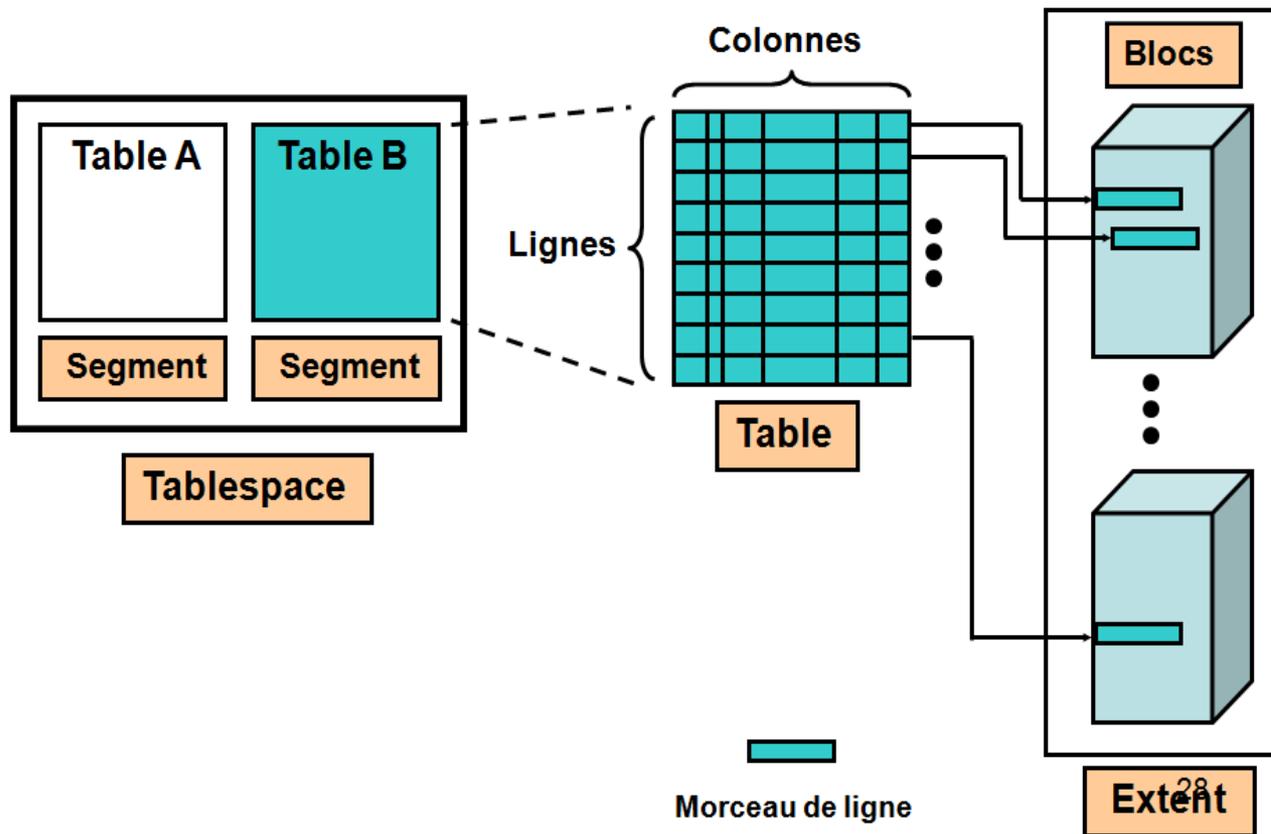




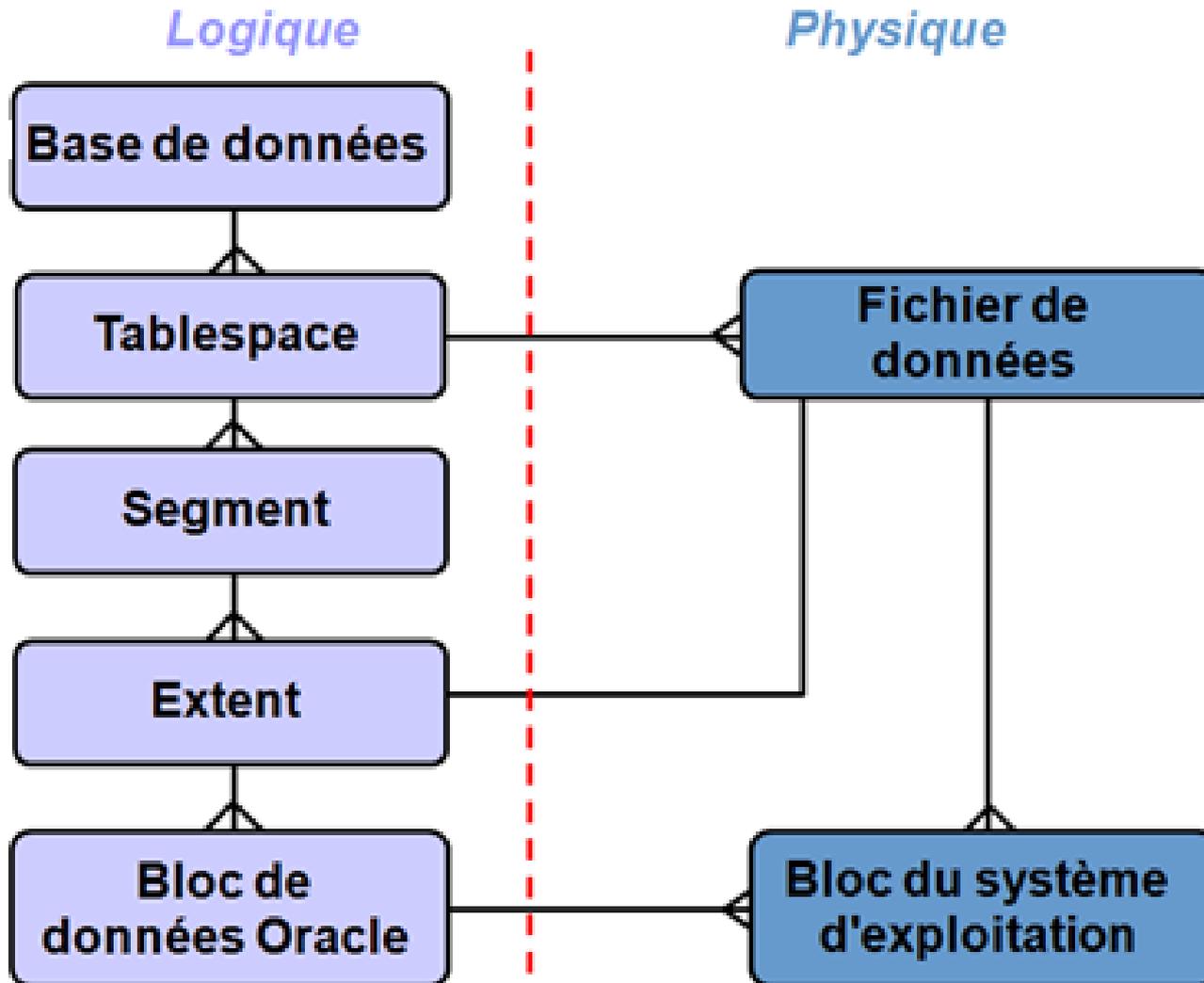
Cours Administration BD

Chapitre 3 : Gestion des tablespaces (Partie 2)

Mode de stockage des données d'une table



Modèle Logique/Physique



- Une base peut être décomposée en tablespaces : partitions logiques contenant un ou plusieurs fichiers.
- Le **tablespace** est une unité logique de stockage composée de **fichiers de données**.
- C'est à partir de cet objet que l'on va gérer le stockage principalement.
- Le stockage est organisé en **segments** et **extent**.

- Un tablespace est composé d'au moins un datafile, c'est à dire un fichier de données qui est physiquement présent sur le serveur à l'endroit spécifié lors de sa création.
- Un fichier appartient à 1 et 1 seul tablespace.
- Un tablespace peut s'étendre soit par ajout (on-line) d'un fichier, soit par auto-extension du fichier du tablespace.
- Chaque datafile est constitué de segments d'au moins un extent (ou page) lui-même constitué d'au moins 3 blocs : l'élément le plus petit d'une base de données.

L'extent n'a aucune signification particulière, c'est juste un groupe de blocs contigus pouvant accueillir des données,

- Les fichiers de données sont découpés en **blocs** d'une taille donnée (4 ko, 8 ko,...).
- L'espace occupé par un objet dans un tablespace est désigné par le terme générique de **segment**.
- Un segment appartient à un tablespace et est constitué **d'extensions** (extents).
- Une extension est un ensemble de **blocs** contigus dans un fichier de données.

■ Il y à quatre types principaux de segments:

- Les segments de table : espace occupé par les tables
- Les segments d'index : espace occupé par les index
- Les segments d'annulation: espace temporaire utilisé pour stocker les informations permettant d'annuler une transaction
- Les segments temporaire : espace temporaire utilisé notamment lors d'un tri

- Une base de données **Oracle** possède au minimum 4 **tablespace**:
 - **Tablespace SYSTEM** dictionnaire de données
 - **Tablespace SYSAUX** system auxiliaire composants oracle
 - **Tablespace UNDO** segments d'annulations
 - **Tablespace TEMPORAIRE** segments temporaires
- Puis des **tablespace** spécifiquement créés pour des données applicatives et utilisateurs.
- *Mais pourquoi autant de tablespaces?*

Simplement pour bien ranger les objets par type (application, index, tables, etc...) et ainsi permettre une administration plus fluide et plus souple, une meilleure optimisation, et performances.

Le tablespace temporaire

- Un tablespace temporaire est un tablespace spécifique aux opérations de tri pour lesquelles la SORT_AREA_SIZE ne serait pas suffisamment grande.
- Ce tablespace n'est pas destiné à accueillir des objets de la base de données et son usage est réservé au système.

Le tablespace UNDO

- Le tablespace UNDO, comme son nom l'indique, est réservé exclusivement à l'annulation des commandes DML (Data Manipulate Language) (UPDATE, INSERT, etc...).
- Lorsqu'on exécute l'ordre DELETE par exemple, Oracle commence par copier les lignes à supprimer dans le tablespace UNDO et ensuite indique que les blocs contenant les données dans le tablespace d'origine sont libres.
- Un ROLLBACK permet de revenir en arrière alors que le COMMIT supprimera les lignes du tablespace UNDO .

- **Gestion de l'allocation d'espace**
- Lors de la création d'un segment , Oracle crée un extent dans le tablespace cible de l'objet.
- Lorsqu'on remplit ce segment Oracle remplit les blocs de données qui constituent l'extent jusqu'à remplir l'extent entièrement et crée un nouvel extent si le précédent est plein.
- *Il existe 2 modes de gestion:*
 - 1** - Soit localement, les informations **extents** libres et alloués sont stockées dans l'entête des fichiers de données du **tablespace**.
 - 2** - Soit par le **dictionnaire de données**, les informations extents libres et alloués sont stockées dans les tables du **dictionnaire de données (Tablespace SYSTEM)**.

- Un Tablespace est une unité logique de stockage composée de fichiers physiques.
- Le stockage est organisé en Segments et Extents.
- Un Tablespace peut être géré dans le dictionnaire ou localement.
- On appelle Tablespace permanents, les Tablespace autres que TBS UNDO et TBS TEMPORARY.
- A partir de la version 10G, Oracle permet la création de Tablespace Bigfile (1 fichier unique volumineux), sinon il est appelé Tablespace Smallfile par défaut.
- Un Tablespace peut être ONLINE (accessible) ou OFFLINE (inaccessible).
- Un Tablespace peut être en READ WRITE (lecture/écriture) ou READ ONLY (lecture).

Gestion des tablespaces

- Création d'un tablespace
- Modification d'un tablespace
- Suppression d'un tablespace

- ordres SQL associés aux tablespaces :
- SQL> CREATE TABLESPACE ...
- SQL> DROP TABLESPACE...
- SQL> ALTER TABLESPACE...

Création d'un tablespace : Syntaxe

```
CREATE TABLESPACE nom_tablespace  
DATAFILE 'chemin d'accès du fichier' SIZE  
valeur [K|M]  
[MINIMUM EXTENT valeur [K|M] ]  
[MAXIMUM EXTENT valeur [K|M] ]  
DEFAULT STORAGE  
( [INITIAL valeur [K|M]]  
  [NEXT valeur [K|M]]  
  [MINEXTENTS nombre]  
  [MAXEXTENTS nombre]  
  [PCTINCREASE pourcentage] ) ;
```

Création d'un tablespace : Syntaxe

```
CREATE TABLESPACE nom_tablespace  
DATAFILE 'chemin d'accès du fichier' SIZE valeur [K|M]  
[MINIMUM EXTENT valeur [K|M] ]  
[MAXIMUM EXTENT valeur [K|M] ]  
[DEFAULT STORAGE  
( [INITIAL valeur [K|M]]  
[NEXT valeur [K|M]]  
[MINEXTENTS nombre]  
[MAXEXTENTS nombre]  
[PCTINCREASE pourcentage] ) ;
```

Taille du fichier (tablespace)

Chemin d'accès du **fichier** dans lequel est stocké le tablespace
Ex : 'F:\oracle\oradata**data_tbs.dbf**'

Création d'un tablespace : Syntaxe

CREATE TABLESPACE nom_tablespace

DATAFILE 'chemin d'accès du fichier' **SIZE** valeur [K|M]

[**MINIMUM EXTENT** valeur [K|M]]

Taille **minimale** d'une extension

[**MAXIMUM EXTENT** valeur [K|M]]

Taille **maximale** d'une extension

[**DEFAULT STORAGE**

([**INITIAL** valeur [K|M]]

[**NEXT** valeur [K|M]]

[**MINEXTENTS** nombre]

[**MAXEXTENTS** nombre]

[**PCTINCREASE** pourcentage]) ;

Création d'un tablespace : Syntaxe

CREATE TABLESPACE nom_tablespace

DATAFILE 'chemin d'accès du fichier' **SIZE** valeur [K|M]

[**MINIMUM EXTENT** valeur [K|M]]

[**MAXIMUM EXTENT** valeur [K|M]]

[**DEFAULT STORAGE**

([**INITIAL** valeur [K|M]]
[**NEXT** valeur [K|M]]
[**MINEXTENTS** nombre]
[**MAXEXTENTS** nombre]
[**PCTINCREASE** pourcentage]) ;

Caractéristiques du stockage

- **INITIAL** : Taille de la première extension allouée lors de la création d'un segment
- **NEXT** : Taille de la deuxième extension du segment
- **MINEXTENTS** : Nombre d'extensions allouées à la création du segment
- **MAXEXTENTS** : Nombre maximal d'extensions pouvant être allouées au segment
- **PCTINCREASE** : Pourcentage d'accroissement de la taille des extensions, appliqué à partir de la troisième extension

- *Syntax ordre sql CREATE TABLESPACE Permanent.*

```
CREATE [ BIGFILE | SMALLFILE ] TABLESPACE Name
DATAFILE file_specification SIZE integer [ K | M | G | T | P | E ] [REUSE]
AUTOEXTEND
  { OFF
  | ON [ NEXT integer [ K | M | G | T | P | E ] ]
      [ MAXSIZE { UNLIMITED | integer [ K | M | G | T | P | E ] } ]
  | DEFAULT [ { COMPRESS | NOCOMPRESS } ]
STORAGE
  ( { INITIAL integer [ K | M | G | T | P | E ]
    | NEXT integer [ K | M | G | T | P | E ]
    | MINEXTENTS integer
    | MAXEXTENTS { integer | UNLIMITED }
    | PCTINCREASE integer
    | FREELISTS integer
    | FREELIST GROUPS integer
    | OPTIMAL [ integer [ K | M | G | T | P | E ]
              | NULL
            ]
    | BUFFER_POOL { KEEP | RECYCLE | DEFAULT }
  } )
EXTENT MANAGEMENT
```

EXTENT MANAGEMENT

```
{ LOCAL  
  [ AUTOALLOCATE  
  | UNIFORM  
    [ SIZE integer [ K | M | G | T | P | E ] ]  
  ]  
  | DICTIONARY  
}
```

```
SEGMENT SPACE MANAGEMENT { AUTO | MANUAL }
```

```
| [ MINIMUM EXTENT integer [ K | M | G | T | P | E ]  
| BLOCKSIZE integer [ K ]  
| { LOGGING | NOLOGGING }  
| FORCE LOGGING  
| FLASHBACK { ON | OFF }  
| { ONLINE | OFFLINE };
```

- Création d'un Tablespace avec une gestion locale uniforme des extensions.

```
SQL> CREATE SMALLFILE TABLESPACE "DATA"  
DATAFILE 'C:\ORACLE\PRODUCT\10.2.0\ORADATA\DBTEST\DATA.DBF' SIZE 2G  
AUTOEXTEND ON NEXT 100M MAXSIZE 5000M  
LOGGING  
ONLINE  
PERMANENT  
BLOCKSIZE 8192  
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 10M  
SEGMENT SPACE MANAGEMENT AUTO;
```

- Création d'un Tablespace avec une gestion locale automatique des extensions.

```
SQL> CREATE SMALLFILE TABLESPACE "DATA"  
DATAFILE 'C:\ORACLE\PRODUCT\10.2.0\ORADATA\DBTEST\DATA.DBF' SIZE 2G  
AUTOEXTEND ON NEXT 100M MAXSIZE 5000M  
LOGGING  
ONLINE  
PERMANENT  
BLOCKSIZE 8192  
EXTENT MANAGEMENT LOCAL AUTOALLOCATE  
SEGMENT SPACE MANAGEMENT AUTO;
```

- Descriptions de quelques Options.

- **BIGFILE | SMALLFILE.**

Si cette clause est omise, Oracle prendra le type par défaut défini au niveau de la base de données.

- **Name.**

C'est le nom que vous donnerez à votre Tablespace.

- **DATAFILE** file_specification.

Permet de préciser l'emplacement du fichier de données pour le Tablespace.

- **AUTOEXTEND.**

Indique si le fichier pourra grossir une fois l'espace alloué est utilisé.

- **NEXT.**

Espace alloué lors de l'extension.

- **MAXSIZE.**

Taille maximale du fichier.

- **EXTENT MANAGEMENT.**

Mode de gestion des extensions du Tablespace.

- **SEGMENT SPACE MANAGEMENT.**

Mode de gestion de l'espace libre des segments dans le Tablespace.(clause valable si TBS géré localement uniquement).

- **MINIMUM EXTENT.**

Taille minimum des Extensions dans le Tablespace. (clause valable si TBS géré dans le Dictionnaire uniquement).

- **BLOCKSIZE.**

Taille du bloc utilisée par le Tablespace. (2k, 4K, 8K, 16K, 32K)

- **LOGGING | NOLOGGING.**

Définit le mode de journalisation des segments qui seront stockés dans le Tablespace. Clause ignorée si FORCE LOGGING est actif niveau Tablespace ou Base de données.

- **FORCE LOGGING.**

Permet de garantir que les modifications sont enregistrées dans les fichiers de journalisation.

- **FLASHBACK { ON | OFF.**

Indique si le Tablespace participe aux opérations de FLASHBACK Database.

- **ONLINE | OFFLINE.**

Indique si le Tablespace est crée Online ou Offline.

Modification d'un tablespace

- Agrandissement de l'espace de stockage
- Modification des caractéristiques de stockage
- Modification des caractéristiques des fichiers associés
- Mise hors service d'un tablespace
- Mise en service d'un tablespace
- Tablespace en lecture seule

Agrandissement de l'espace de stockage

```
ALTER TABLESPACE nom_tablespace  
ADD DATAFILE fichier [, fichier] ;
```

- Il est possible d'agrandir le tablespace en ajoutant de nouveaux fichiers

Modification des caractéristiques de stockage

```
ALTER TABLESPACE nom_tablespace  
[DEFAULT STORAGE  
( [INITIAL valeur [K|M]]  
  [NEXT valeur [K|M]]  
  [MINEXTENTS nombre]  
  [MAXEXTENTS nombre]  
  [PCTINCREASE pourcentage]) ;
```

Modification des caractéristiques des fichiers associés

ALTER TABLESPACE nom_tablespace

RENAME DATAFILE texte_existant [, texte_existant] ...

TO texte_nouveau [, texte_nouveau] ;

- Avec texte_nouveau de la forme '**chemin\nom_fichier**' **SIZE** taille.

Ce qui permet de :

- Changer le nom du fichier
- Changer le fichier du disque
- Changer la taille du fichier

Mise hors service d'un tablespace

```
ALTER TABLESPACE nom_tablespace  
OFFLINE [ NORMAL | IMMEDIATE ] ;
```

- **OFFLINE NORMAL** : le tablespace est **mis hors service** lorsque tous les utilisateurs du tablespace ont fini leurs transactions en cours
- **OFFLINE IMMEDIATE** : le tablespace est **mis hors service** même si les transactions en cours ne sont pas terminées

Mise en service d'un tablespace

```
ALTER TABLESPACE nom_tablespace  
ONLINE ;
```

Tablespace en lecture seule

```
ALTER TABLESPACE nom_tablespace  
READ ONLY ;
```

- Un tablespace peut être créé pour recevoir des tables de type historique dont les données ne doivent jamais être modifiées. Il est possible de protéger une telle structure en attribuant au tablespace le mode lecture seule

Suppression d'un tablespace : Syntaxe

DROP TABLESPACE nom_tablespace
[**INCLUDING CONTENT**] ;

- L'option **INCLUDING CONTENT** doit être utilisée si le tablespace contient des informations
- L'exécution de l'ordre **DROP** ne supprime pas les fichiers du disque ; il faut ensuite détruire ces fichiers par les commandes du système d'exploitation
- Avant de supprimer un tablespace, il faut le mettre **hors service**