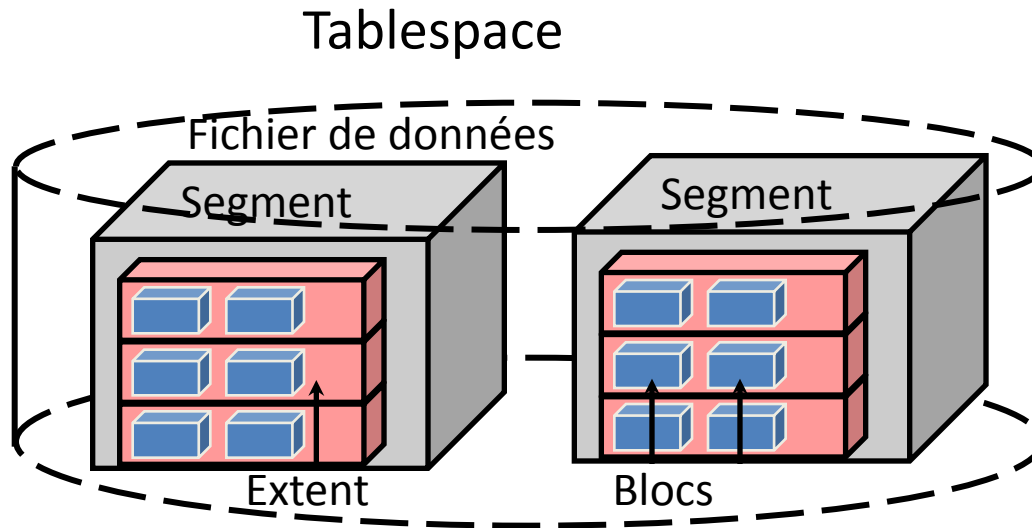


Structure logique

- La structure logique définit le mode d'utilisation de l'espace physique d'une base de données.
- Cette structure possède une hiérarchie composée de tablespaces, de segments, d'extents et de blocs.



Structure logique

Tablespaces

Les données d'une base de données Oracle sont stockées dans des tablespaces.

- Une base de données Oracle peut être divisée en plus petites zones logiques d'espace appelées tablespaces.
- Un tablespace ne peut appartenir qu'à une seule base de données à la fois.
- Chaque tablespace est constitué d'un ou de plusieurs fichiers du système d'exploitation, appelés fichiers de données.
- Un tablespace peut contenir un ou plusieurs segments.
- Les tablespaces peuvent être mis en ligne lorsque la base de données est active.
- A l'exception du tablespace `SYSTEM` ou des tablespaces contenant un segment d'annulation actif, les tablespaces peuvent être mis hors ligne sans interruption de la base de données.
- Les tablespaces peuvent être accessibles en lecture et écriture ou en lecture seule.

Structure logique

Fichiers de données (structure non logique)

- Chaque tablespace d'une base de données Oracle contient un ou plusieurs fichiers appelés fichiers de données. Ces fichiers sont des structures physiques conformes au système d'exploitation sur lequel s'exécute le serveur Oracle.
- Un fichier de données ne peut appartenir qu'à un seul tablespace.
- Un serveur Oracle crée un fichier de données dans un tablespace en allouant l'espace disque défini et un petit espace supplémentaire.
- L'administrateur de base de données peut modifier la taille des fichiers de données après leur création ou indiquer que leur taille peut augmenter de façon dynamique à mesure que celle des objets du tablespace s'accroît.

Structure logique

Segments

- Un segment est un espace alloué à une structure de stockage logique spécifique d'un tablespace.
- Un tablespace peut contenir un ou plusieurs segments.
- Un segment ne peut pas être réparti sur plusieurs tablespaces, mais peut s'étendre à plusieurs fichiers de données d'un même tablespace.
- Chaque segment est constitué d'un ou de plusieurs extents.

Structure logique

Extents

- Les extents permettent d'allouer de l'espace à un segment.
 - Un segment peut être constitué d'un ou de plusieurs extents.
 - Lorsque vous créez un segment, celui-ci contient au moins un extent.
 - A mesure que la taille du segment augmente, des extents lui sont ajoutés.
 - L'administrateur de base de données peut ajouter manuellement des extents à un segment.
 - Un extent est constitué de blocs Oracle contigus.
 - Un extent ne peut pas s'étendre sur plusieurs fichiers de données. Il doit donc appartenir à un seul fichier de ce type.

Structure logique

Blocs de données

- Le serveur Oracle gère l'espace de stockage des fichiers de données à l'aide d'unités appelées blocs de données ou blocs Oracle.
 - Lorsque le degré de détail maximum est atteint, les données d'une base Oracle sont stockées dans des blocs de données.
 - Les blocs de données Oracle représentent la plus petite unité de stockage que le serveur Oracle peut allouer, écrire ou lire.
 - Un bloc de données correspond à un ou plusieurs blocs du système d'exploitation alloués à partir d'un fichier de données existant.
 - Le paramètre d'initialisation `DB_BLOCK_SIZE` permet de définir la taille de bloc standard lors de la création d'une base de données Oracle.
 - La taille des blocs de données doit correspondre à un multiple de la taille des blocs du système d'exploitation afin d'éviter les opérations d'entrée/sortie inutiles.
 - La taille maximale d'un bloc de données dépend du système d'exploitation utilisé.

Types de tablespace

- L'administrateur de base de données crée des tablespaces pour améliorer le contrôle et faciliter la gestion de la base.
- Le serveur Oracle accepte deux types de tablespace :
 - le tablespace `SYSTEM`
 - et tous les autres tablespaces.

Types de tablespace

- **Le tablespace SYSTEM :**
 - est créé en même temps que la base de données,
 - doit exister dans toutes les bases de données,
 - contient le dictionnaire de données
 - contient le segment d'annulation SYSTEM,
 - ne doit pas contenir de données utilisateur, bien que cela soit possible.

Types de tablespace

- **Les tablespaces non SYSTEM :**
 - facilitent l'administration de la base de données,
 - séparent les segments d'annulation, les segments temporaires, les segments de données d'application et les segments d'index d'application,
 - séparent les données en fonction des besoins de sauvegarde,
 - séparent les données dynamiques des données statiques,
 - gèrent la quantité d'espace allouée aux objets utilisateur.

Créer des tablespaces

- La commande CREATE TABLESPACE permet de créer un tablespace :

```
CREATE TABLESPACE tablespace  
  [DATAFILE clause]  
  [MINIMUM EXTENT integer[K|M]]  
  [BLOCKSIZE integer [K]]  
  [LOGGING|NOLOGGING]  
  [DEFAULT storage_clause ]  
  [ONLINE|OFFLINE]  
  [PERMANENT|TEMPORARY]  
  [extent_management_clause]  
  [segment_management_clause]
```

Créer des tablespaces

- **Tablespace** correspond au nom du tablespace à créer.
- **DATAFILE** définit les fichiers de données constituant le tablespace.
- **MINIMUM EXTENT** garantit que la taille de chaque extent (ensemble de blocs contigus) du tablespace est un multiple de **integer**. Utilisez **K** ou **M** pour définir la taille en kilo-octets ou en méga-octets.

Créer des tablespaces

- **BLOCKSIZE** indique une taille de bloc non standard pour le tablespace.
- Pour définir cette clause, le paramètre **DB_CACHE_SIZE** et au moins un paramètre **DB_nK_CACHE_SIZE** doivent être définis et la valeur de **Integer** que vous définissez dans cette clause doit être identique à celle d'un paramètre **DB_nK_CACHE_SIZE**.

Créer des tablespaces

- **LOGGING** indique que toutes les modifications effectuées sur les tables, index et partitions du tablespace seront écrites par défaut dans le fichier de journalisation (LOGGING est la valeur par défaut).
- **NOLOGGING** indique que toutes les modifications effectuées ne seront pas écrites par défaut dans le fichier de journalisation

Créer des tablespaces

- **DEFAULT** définit les paramètres de stockage par défaut de tous les objets créés dans le tablespace.
- **OFFLINE** rend le tablespace indisponible dès sa création.
- **PERMANENT** indique que le tablespace peut être utilisé pour stocker des objets permanents.
- **TEMPORARY** indique que le tablespace ne contiendra que des objets temporaires, tels que les segments de tri implicite créés par la clause ORDER BY.
 - La définition des clauses EXTENT MANAGEMENT LOCAL et BLOCKSIZE est impossible.

Créer des tablespaces

extent_management_clause:

```
[ EXTENT MANAGEMENT [ DICTIONARY | LOCAL  
[ AUTOALLOCATE | UNIFORM [SIZE integer[K|M]] ] ] ]
```

Où :

- DICTIONNARY indique que le tablespace est géré à l'aide des tables du dictionnaire.
- LOCAL indique que le tablespace est géré localement à l'aide d'un bitmap. Si vous définissez LOCAL, vous ne pouvez pas définir DEFAULT storage_clause, MINIMUM EXTENT ou TEMPORARY.

Créer des tablespaces

extent_management_clause:

```
[ EXTENT MANAGEMENT [ DICTIONARY | LOCAL  
[ AUTOALLOCATE | UNIFORM [SIZE integer[K|M]] ] ] ]
```

Où :

- AUTOALLOCATE (option par défaut) indique que le tablespace est géré par le système (les utilisateurs ne peuvent pas définir de taille d'extent).
- UNIFORM indique que le tablespace est géré à l'aide d'extents d'une taille uniforme de SIZE octets. Utilisez K ou M pour définir la taille d'extent en kilo-octets ou en mégaoctets. La taille par défaut est de 1 Mo.

Créer des tablespaces

segment_management_clause

- ne s'applique qu'aux tablespaces permanents, gérés localement.
- Elle vous permet d'indiquer si Oracle doit procéder au suivi de l'espace libre et utilisé dans les segments du tablespace à l'aide des bitmaps ou des listes de blocs libres (free list).

datafile_clause ::= filename [SIZE integer[K|M]
[REUSE]

[autoextend_clause]

Créer des tablespaces

segment_management_clause

- **filename** correspond au nom d'un fichier de données du tablespace.
- **SIZE** définit la taille du fichier (utilisez K ou M pour définir la taille en kilo-octets ou en mégaoctets).
- **REUSE** permet au serveur Oracle de réutiliser un fichier existant.

Créer des tablespaces

segment_management_clause

autoextend_clause permet d'activer ou de désactiver l'extension automatique du fichier de données.

- **NEXT** indique la taille en octets de la prochaine incrémentation de l'espace disque devant être allouée automatiquement lorsque plusieurs extents sont requis.
- **MAXSIZE** indique l'espace disque maximum autorisé pour l'extension automatique d'un fichier de données.
- **UNLIMITED** indique l'espace disque pouvant être affecté

Créer des tablespaces

- Exemple :

```
CREATE TABLESPACE userdata  
    DATAFILE  
        '/u01/oradata/userdata01.dbf'  
    SIZE 100M  
    AUTOEXTEND ON NEXT 5M MAXSIZE  
        200M;
```

Gestion de l'espace dans les tablespaces

- Tablespace géré localement :
 - Extents libres gérés dans le tablespace
 - Un bitmap est utilisé pour enregistrer des extents libres
 - Chaque bit correspond à un bloc ou à un groupe de blocs
 - La valeur des bits indique si ceux-ci sont disponibles ou utilisés
- Tablespace géré au moyen du dictionnaire :
 - Les extents libres sont gérés par le dictionnaire de données.
 - Les tables appropriées sont mises à jour lorsque les extents sont alloués ou libérés.

Tablespaces gérés localement

- La contention au niveau des tables du dictionnaire de données est réduite.
- Aucune annulation n'est générée lors de l'allocation ou de la libération d'espace.
- Aucune fusion n'est requise.

```
CREATE TABLESPACE userdata  
DATAFILE '/u01/oradata/userdata01.dbf' SIZE  
500M  
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K;
```

Tablespaces gérés au moyen du dictionnaire

- Les extents sont gérés dans le dictionnaire de données
- Chaque segment stocké dans le tablespace peut posséder une clause de stockage différente
- Une fusion est requise

```
CREATE TABLESPACE userdata  
DATAFILE '/u01/oradata/userdata01.dbf'  
SIZE 500M EXTENT MANAGEMENT DICTIONARY  
DEFAULT STORAGE  
(initial 1M NEXT 1M PCTINCREASE 0);
```

Tablespace d'annulation

- Il permet de stocker des segments d'annulation.
- Il ne peut contenir aucun autre objet.
- Les extents sont gérés localement.
- Il ne peut être utilisé qu'avec les clauses `DATAFILE` et `EXTENT`.

```
CREATE UNDO TABLESPACE undo1  
DATAFILE '/u01/oradata/undo01.dbf' SIZE 40M;
```


Tablespaces TEMPORARY

- Ils sont utilisés pour les opérations de tri
- Ils ne peuvent pas contenir d'objets permanents
- La gestion locale des extents est recommandée

```
CREATE TEMPORARY TABLESPACE temp  
TEMPFILE '/u01/oradata/temp01.dbf' SIZE 500M  
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 4M;
```

Tablespace TEMPORARY par défaut

- Définit un tablespace TEMPORARY par défaut correspondant à la base de données.
- Permet d'effectuer des suppressions à l'aide du tablespace SYSTEM pour stocker des données temporaires.
- Un tablespace TEMPORARY peut être créé à l'aide de :
 - CREATE DATABASE
 - Géré localement
 - ALTER DATABASE

```
ALTER DATABASE  
DEFAULT TEMPORARY TABLESPACE temp;
```

Créer un tablespace TEMPORARY par défaut

- Pendant la création de la base de données :

```
CREATE DATABASE DBA01
LOGFILE
GROUP 1 ('/$HOME/ORADATA/u01/redo01.log') SIZE 100M,
GROUP 2 ('/$HOME/ORADATA/u02/redo02.log') SIZE 100M,
...
DATAFILE '/$HOME/ORADATA/u01/system01.dbf' SIZE 325M
UNDO TABLESPACE undotbs
DATAFILE '/$HOME/ORADATA/u02/undotbs01.dbf' SIZE 200
DEFAULT TEMPORARY TABLESPACE temp
TEMPFILE '/$HOME/ORADATA/u03/temp01.dbf' SIZE 4M
CHARACTER SET US7ASCII
```

Créer un tablespace TEMPORARY par défaut

- Une fois la base de données créée :

```
ALTER DATABASE  
DEFAULT TEMPORARY TABLESPACE default_temp2;
```

- Pour trouver le tablespace TEMPORARY par défaut de la base de données, interrogez DATABASE_PROPERTIES.

```
SELECT * FROM DATABASE_PROPERTIES;
```

Restrictions relatives au tablespace TEMPORARY par défaut

- Les tablespaces TEMPORARY par défaut ne peuvent pas être :
 - supprimés tant qu'un nouveau tablespace par défaut n'est pas disponible,
 - mis hors ligne,
 - transformé en tablespace permanent.

Tablespaces accessibles en lecture seule

- Utilisez la commande suivante pour placer un tablespace en lecture seule.

```
ALTER TABLESPACE userdata READ ONLY;
```

- Crée un point de reprise.
- Ces données sont disponibles pour les opérations de lecture uniquement.
- Il est possible de supprimer des objets des tablespaces

Mettre un tablespace hors ligne

- Non disponible pour l'accès aux données
- Tablespaces ne pouvant pas être mis hors ligne :
 - Tablespace SYSTEM :
 - Tablespaces contenant des segments d'annulation actifs
 - Tablespace TEMPORARY par défaut
- Mettre un tablespace hors ligne :

```
ALTER TABLESPACE userdata OFFLINE;
```

- Mettre un tablespace en ligne :

```
ALTER TABLESPACE userdata ONLINE;
```

Modifier les paramètres de stockage

- Utiliser la commande ALTER TABLESPACE pour modifier les paramètres de stockage :

```
ALTER TABLESPACE userdata MINIMUM EXTENT 2M;
```

```
ALTER TABLESPACE userdata  
DEFAULT STORAGE (INITIAL 2M NEXT 2M  
MAXEXTENTS 999);
```

- Les paramètres de stockage des tablespaces gérés localement ne peuvent pas être modifiés.

Redimensionner un tablespace

- Un tablespace peut être redimensionné en :
 - modifiant la taille d'un fichier de données :
 - automatiquement à l'aide de `AUTOEXTEND`
 - manuellement à l'aide de `ALTER TABLESPACE`
 - ajoutant un fichier de données à l'aide de `ALTER TABLESPACE`

Activer l'extension automatique des fichiers de données

- Les fichiers peuvent être redimensionnés automatiquement à l'aide des commandes suivantes :
 - CREATE DATABASE
 - CREATE TABLESPACE
 - ALTER TABLESPACE ... ADD DATAFILE
- Exemple :

```
CREATE TABLESPACE user_data
DATAFILE
  '/u01/oradata/userdata01.dbf' SIZE 200M
  AUTOEXTEND ON NEXT 10M MAXSIZE 500M;
```

- Interrogez la vue DBA_DATA_FILES pour déterminer si AUTOEXTEND est activé.

Redimensionner manuellement un fichier de données

- Augmentez ou réduisez manuellement la taille d'un fichier de données à l'aide de `ALTER DATABASE`.
- Redimensionner un fichier de données ajoute de l'espace sans ajouter de fichier de données.
- Le redimensionnement manuel d'un fichier de données requiert l'utilisation de l'espace libre d'une base de données.
- Exemple :

```
ALTER DATABASE  
DATAFILE '/u03/oradata/userdata02.dbf'  
RESIZE 200M;
```

Ajouter des fichiers de données à un tablespace

- Augmente l'espace alloué à un tablespace en ajoutant des fichiers de données.
- La clause `ADD DATAFILE` permet d'ajouter un fichier de données.
- Exemple :

```
ALTER TABLESPACE user_data  
ADD DATAFILE '/u01/oradata/userdata03.dbf'  
SIZE 200M;
```

Méthodes de déplacement des fichiers de données

– ALTER TABLESPACE

- Le tablespace doit être hors ligne.
- Les fichiers de données cible doivent exister.

```
ALTER TABLESPACE userdata RENAME  
DATAFILE '/u01/oradata/userdata01.dbf'  
TO '/u02/oradata/userdata01.dbf';
```

– Etapes permettant de renommer un fichier

- Mettez le tablespace hors ligne.
- Utilisez la commande appropriée du système d'exploitation pour déplacer ou copier les fichiers.
- Exécutez la commande ALTER TABLESPACE RENAME DATAFILE.
- Mettez le tablespace en ligne.
- Au besoin, utilisez la commande appropriée du système d'exploitation pour supprimer le fichier.

Méthodes de déplacement des fichiers de données

– ALTER DATABASE

- La base de données doit être montée.
- Le fichier de données cible doit exister.

```
ALTER DATABASE RENAME  
FILE '/u01/oradata/system01.dbf'  
TO '/u03/oradata/system01.dbf';
```

Supprimer des tablespaces

- Un tablespace ne peut pas être supprimé :
 - s'il s'agit du tablespace `SYSTEM`,
 - s'il possède des segments actifs.
- `INCLUDING CONTENTS` supprime les segments
- `INCLUDING CONTENTS AND DATAFILES` supprime les fichiers de données
- `CASCADE CONSTRAINTS` supprime les contraintes d'intégrité référentielle

```
DROP TABLESPACE userdata  
INCLUDING CONTENTS AND DATAFILES;
```

Obtenir des informations sur les tablespaces

- Vous pouvez obtenir des informations sur les tablespaces et les fichiers de données en interrogeant les éléments suivants :
 - Tablespaces :
 - `DBA_TABLESPACES`
 - `V$TABLESPACE`
 - Informations sur le fichier de données :
 - `DBA_DATA_FILES`
 - `V$DATAFILE`
 - Informations sur les fichiers temporaires :
 - `DBA_TEMP_FILES`
 - `V$TEMPFILE`