

Tablespaces et segments d'une Base de données Oracle

les tablespaces

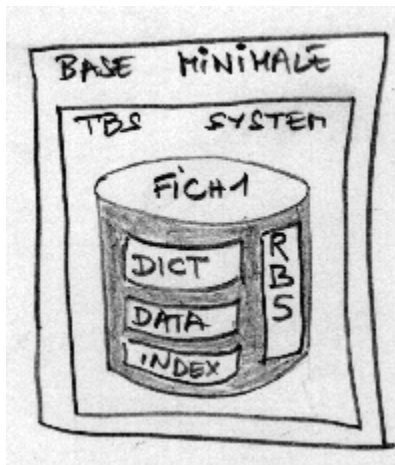
Tablespaces standards (ou PERMANENTs)

Une base peut être décomposée en tablespaces : partitions logiques contenant un ou plusieurs fichiers. Un fichier appartient à 1 et 1 seul tablespace.

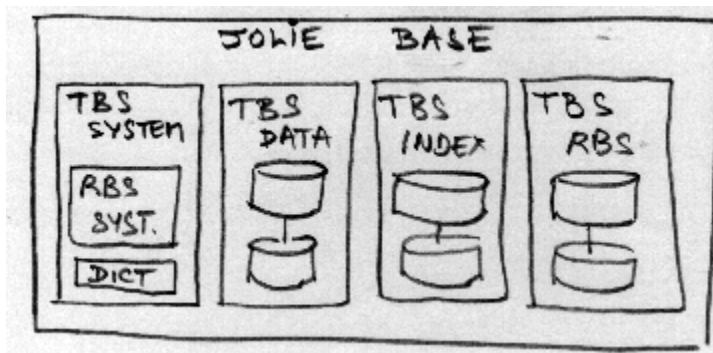
Un tablespace peut s'étendre soit par ajout (on-line) d'un fichier, soit par auto-extension DU fichier du tablespace.

Par défaut il existe toujours un tablespace baptisé SYSTEM qui contient le dictionnaire de données et le rollback segment SYSTEM (dans le cas où il n'existe pas d'UNDO tablespace).

On peut également stocker les datas et les index dans ce même tablespace, et obtenir ainsi une base minimale peu structurée, peu performante et peu sécurisée :



Au contraire on peut répartir les données, les index, mais aussi les images avant (rollback segments) sur un nombre maximum de disques. On y gagnera en performance, en souplesse, et en sécurité :



ordres SQL associés aux tablespaces :

```
SQL> CREATE TABLESPACE ...  
SQL> DROP TABLESPACE...  
SQL> ALTER TABLESPACE...
```

Par défaut un tablespace à la création est ON LINE (et donc accessible), il peut être mis OFFLINE pour en interdire l'accès ou pour certaines opérations de maintenance

Description des tablespaces de la base courante dans les vues **Dbf_tablespaces** et **Dbf_data_files** du dictionnaire.

Les fichiers du tablespace

Un tablespace contient AU MOINS un fichier. Celui ci est créé lors de la création du tablespace, de manière automatique par Oracle, en fonction des paramètres données par la commande CREATE ou ALTER tablespace (emplacement du fichier, nom et taille).

lors de la suppression du tablespace (DROP TABLESPACE...) les fichiers correspondant ne sont PAS SUPPRIMES par Oracle !! il faut le faire manuellement au niveau Unix ou Windows (rm, del...) !

quelques exemples SQL pour les tablespaces et les fichiers

→ création d'un tablespace nommé RBS contenant un fic de 10MO et des EXTENTS de 1MO

```
CREATE TABLESPACE RBS DATAFILE 'E:\orant\database\TEST\Rbs1TEST.ora' SIZE 10M
DEFAULT STORAGE ( INITIAL 1024K NEXT 1024K PCTINCREASE 0 );
```

```
ALTER TABLESPACE toto OFFLINE;
```

→ changement des paramètres d'un tablespace existant

```
ALTER TABLESPACE SYSTEM
DEFAULT STORAGE ( INITIAL 100K NEXT 100K MINEXTENTS 1 MAXEXTENTS 300 PCTINCREASE 1 );
```

→ Ajout d'un fichier auto extensible jusqu'à 100 MO

```
ALTER TABLESPACE toto ADD DATAFILE 'E:\orant\database\TEST\TEST.ora' SIZE 10M
AUTOEXTEND ON NEXT 5M MAXSIZE 100M;
```

→ passage en AUTO extension d'un fichier de tablespace existant

```
ALTER DATABASE DATAFILE 'E:\orant\database\TEST\Usr1TEST.ora' AUTOEXTEND ON;
```

extension du tablespace

La taille d'un tablespace est la taille de son (ses) fichier(s) d'origine.

Pour augmenter la taille d'un tablespace, il y a 2 solutions :

- Ajouter un fichier au tablespace, qui sera chaîné au premier (ALTER TABLESPACE toto ADD DATAFILE...)
- mettre le fichier du tablespace en AUTO extension (ALTER DATABASE DATAFILE toto.dbf AUTOEXTEND ON)

Une table, peut "s'étaler" sur plusieurs fichiers. Ainsi le fait qu'une table sature un tablespace n'est pas bloquant il suffit d'augmenter la taille du tablespace.

ATTENTION : la clause AUTOEXTEND spécifie la taille d'extension du fichier d'un tablespace. La clause STORAGE INITIAL, NEXT, MINEXTENTS ... spécifie la taille d'extension d'UN SEGMENT du tablespace par exemple une table. Ces 2 paramètres sont totalement indépendants. La preuve en est qu'une table

(un segment de données) est forcément en allocation dynamique alors qu'un fichier peut avoir une taille fixe (AUTOEXTEND OFF)

Les différents types de tablespaces spéciaux

Tablespaces en lecture seule (READ ONLY tablespaces)

Ces tablespaces sont utilisés en lecture seule. Ils permettent de stocker des données statiques sur des CDROMS, et ne rentrent pas en ligne de compte dans les sauvegardes / restaurations. Pour modifier les données d'un Tablespace READ ONLY il est évidemment obligatoire de modifier préalablement son statut.

```
SQL> ALTER TABLESPACE toto READ ONLY;
SQL> ALTER TABLESPACE toto READ WRITE;
```

Tablespaces temporaires (temporary tablespaces)

Ces tablespaces apparus avec la 9i remplacent les segments temporaires placés précédemment dans des tablespaces standards.

On peut créer un tablespace temporaire par défaut autre que SYSTEM, où seront stockées toutes les données temporaires (utilisées lors des tris, création d'index, jointures, etc). Ils sont définis lors de la création de la base :

```
SQL> CREATE DATABASE ma_base...
DEFAULT TEMPORARY TABLESPACE mon_temp;
```

ou a posteriori :

```
SQL> ALTER DATABASE DEFAULT TEMPORARY TABLESPACE tempts2;
```

En plus de ce tablespace temporaire par défaut, chaque utilisateur peut se voir assigner un tablespace temporaire particulier

```
SQL> CREATE TEMPORARY TABLESPACE mon_temp TEMPFILE '/oracle/data/temp01.dbf'
SIZE 20M
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 10M

SQL> CREATE USER toto
IDENTIFIED BY tutu
DEFAULT TABLESPACE data
QUOTA 100M ON data
TEMPORARY TABLESPACE temp_ts
```

Voir les vues dynamiques **V\$TEMP_EXTENT_MAP** et **V\$TEMP_SPACE_HEADER** pour des infos précises sur l'utilisation en temps réel de ces tablespaces.

Tablespace d'annulation (undo tablespaces)

Les UNDO tablespaces sont exclusivement réservés au stockage de segments d'images avant modification des données pour des annulations éventuelles (ROLLBACK).

Dans les versions précédentes d'Oracle, ces structures n'existaient pas et on utilisait des ROLLBACK SEGMENTS implantables dans n'importe quel tablespace. Oracle peut désormais fonctionner avec des

UNDO tablespaces (gestion Automatique, préconisée) ou avec des rollback segments (gestion manuelle)

Paramètres de l'INIT.ORA associés

paramètres	valeurs	but
UNDO MANAGEMENT	AUTO MANUAL	utiliser les UNDO tbs ou les rollback segments
UNDO TABLESPACE	nom tablespace	précise le tablespace d'UNDO a utiliser par la base
UNDO_SUPPRESS_ERRORS	TRUE FALSE	evite les erreurs lorsue l'on tente d'utiliser explicitement les Rollback Segments (ALTER ROLLBACK..., SET TRANSACTION USE ROLLBACK...) alors que la base utilise des UNDO tbs

SQL spécifique associé

```
CREATE UNDO TABLESPACE undo_1  
DATAFILE '/tmp/undo1.dbf' SIZE 10M AUTOEXTEND ON
```

ou dès la creation de la base

```
CREATE DATABASE test...  
UNDO TABLESPACE undo_1  
DATAFILE '/tmp/undo1.dbf' SIZE 10M AUTOEXTEND ON
```

Tablespace gérés localement (Locally managed tablespaces)

A la différence des tablespaces standards géré au niveau du dictionnaire de données, la gestion de l'espace physique (allocation / libération de blocs) se fait dans l'entête du fichier(s) du tablespace. Une table binaire d'allocation (bitmap) y est maintenue.

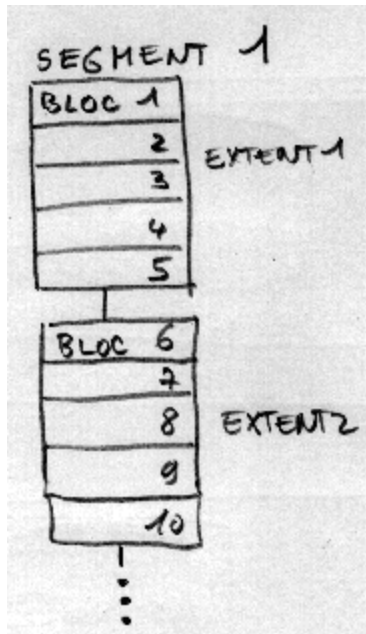
Avantages :

- pas de contention en mise à jour au niveau du dictionnaire
- et conséquemment pas d'utilisation de Rollback segment pour ces transactions
- pas de soucis de gestion de l'espace (calcul d'un storage adéquat)
- "coalesce" automatique (fusion des espaces libres contigus pour optimiser l'espace libre)

Evidemment la clause "DEFAULT STORAGE" est invalide pour les tablespaces gérés localement.

les segments

un segment est composé d'extents. Un **extent** est composé de blocs contigus dont la taille dépend de l'OS. Le segment s'étend dynamiquement au sein du tablespace (éventuellement sur plusieurs fichiers donc...).



Il existe plusieurs types de segments :

- segment de données = table
- segment d'index = index
- segment d'annulation = **rollback segment**, qui stocke l'image avant modification des données
- segment temporaire, utilisé en interne par Oracle, si la zone mémoire de tri est insuffisante

remarque : seuls les objets 'physiques' peuvent être des segments. Ainsi une vue ou un synonyme n'est pas un segment...

On peut forcer les segments de données et d'index à s'implanter dans un tablespace particulier :

- explicitement à la création du segment
- implicitement en affectant un tablespace par défaut à l'utilisateur qui va créer le segment.

```
SQL> create table credit (n number, ...)
tablespace TBS_COMPTA;
ou bien
SQL> create user Appli_comptable default tablespace COMPTA;
SQL> create table credit (n number, ...);
```

Description dans la vue **dba_segments** du dictionnaire.

principe d'allocation des tables spaces

Allocation des Tablespaces standards

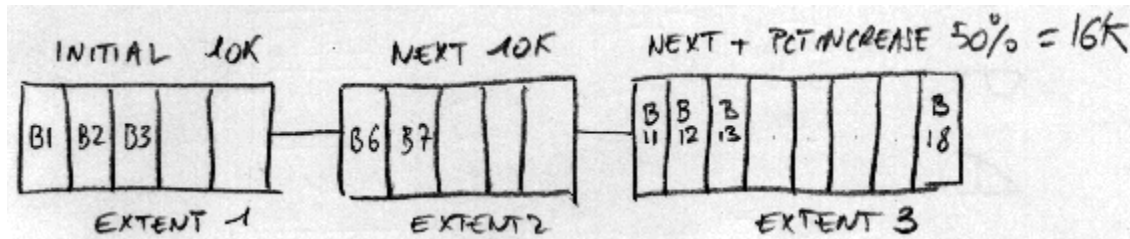
L'allocation obéit à des règles définies par une clause STORAGE. Oracle applique d'abord la clause STORAGE du segment (définie explicitement dans un create table par exemple), sinon il utilise la clause du tablespace (définie explicitement lors du create tablespace), sinon il utilise les valeurs par défaut (implicites) du tablespace.

Au départ, lors de la création du segment (table, index ou autre) Oracle alloue MINEXTENTS **extents** de taille INITIAL. Ensuite, lorsque le segment se remplit, quand le(s) premier(s) extent(s) est (sont) plein(s), il

alloue un extent de taille NEXT. Ensuite il augmente la taille des extents d'un pourcentage fixé par PCTINCREASE. La limite est définie par MAXEXTENTS.

```
SQL> create table credit (n number,...)
storage ( initial 10K next 10K pctincrease 50 maxextents 100) ;
```

voici ce que donne le segment après création, puis 2 allocations supplémentaires :



Voir dba extents dans le dictionnaire.

Allocation des Tablespaces gérés localement

Il y a deux types d'allocation pour ces tablespaces :

- "SYSTEM" : le DBA précise la taille de l'extent INITIAL puis Oracle détermine au mieux la taille des extents suivants (minimum 64K)
- "UNIFORM" : tous les extents auront la même taille, précisée par le DBA, ou la taille par défaut : 1MO

La gestion de l'espace libre

Lorsqu'un tablespace est créé (et le fichier minimum associé) Il est pratiquement vide hormis l'entete du fichier. L'espace libre (FREE SPACE) diminue au fur et a mesure de la creation ou de l'augmentation des segments dans ce tablespace et se fragmente lors de la libération de blocs (DELETE, TRUNCATE, etc). Une "carte des trous" est donnée dans DBA_FREE_SPACE. Cette table possède une ligne par trou et la colonne BYTES donne la taille de chaque trou en octets.

Pour avoir l'espace libre total d'un tablespace il faudra donc sommer tous les espaces libres de tous les fichier du tablespace.

```
SELECT SUM(BYTES)/1024 "Taille en KO" FROM DBA_FREE_SPACE
WHERE TABLESPACE_NAME= 'TBS_TOTO'
```

Pour récupérer l'espace libre d'un segment, on peut utiliser

```
TRUNCATE table_toto DROP STORAGE
```

ou

```
ALTER TABLE table_toto DEALLOCATE UNUSED
```

L'espace libre est défragmenté de temps en temps par Oracle. En clair les espaces libres contigus sont fusionnés (COALESCED).

voir DBA_FREE_SPACE_COALESCED dans le dictionnaire.

On peut faire un COALESCE manuel :

```
ALTER TABLESPACE toto COALESCE
```