

1

Types de segments

Notions sur les segments

⌘ **Table** (ou *table non-clusterisée* ou *table non-partitionnée*)

- ☑ ressource la plus courante pour stocker les données
- ☑ données stockées sans ordre particulier
- ☑ on a peu de contrôle sur l'emplacement des *lignes* dans les *blocs d'une table*.
- ☑ Appartient à un seul tablespace

⌘ **Partition de table** (V8 + option Partitioning)

- ☑ Partition de la table suivant des clés
- ☑ stockage des partitions dans plusieurs tablespaces
- ☑ une partition est un segment
- ☑ *Intérêts* : objets utilisés souvent et de façon simultanée

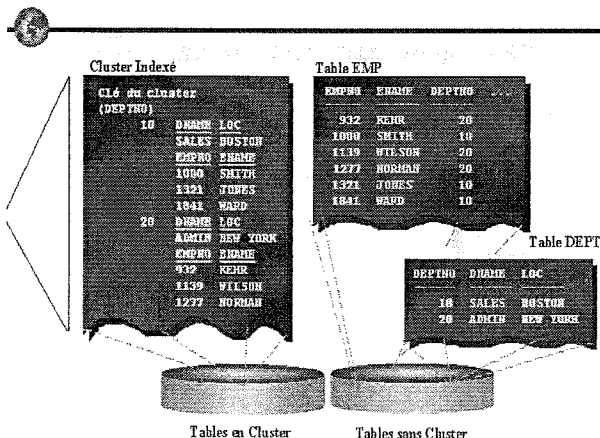
cluster : ensemble de Tables qui possèdent des colonnes commune accélérer les jointures

Types de segments (2)

⌘ **Cluster**

- ☑ les lignes d'un cluster sont stockées suivant une valeur clé de colonne
- ☑ un cluster contient une ou plusieurs tables
- ☑ le cluster et les tables qui y participent appartiennent au même segment
- ☑ *Intérêts* : accélère les jointures

Clusters Indexés



Types de segments (3)

⌘ **Index**

- ☑ Toutes les entrées d'un index sont stockées dans un segment d'index
- ☑ Si une table à 3 index, 3 segments d'index sont utilisés
- ☑ *Intérêts* : accélère les recherches, les jointures ...

⌘ **Partition d'index**

- ☑ même idée que *Partition de table*

⌘ **Rollback segment (segment UNDO)**

- ☑ utilisé par une transaction modifiant l'état de la base
- ☑ avant de changer les blocs de données ou d'index, l'ancienne valeur est stockée dans le rollback segment
- ☑ Sert pour restaurer la base en cas de panne

Types de segments (4)

⌘ Segment temporaire

- ☒ pour effectuer les tris engendrés par
CREATE INDEX, SELECT DISTINCT,
GROUP BY

⌘ Segment LOB

- ☒ pour stocker des colonnes d'objets volumineux (LOB), e.g. documents textes, images, vidéos

⌘ Index LOB

- ☒ implicitement créé lors de la création d'un segment LOB
- ☒ Pour accélérer les requêtes sur ces colonnes

⌘ Table imbriquée (V8 + Option Objects)

- ☒ table en non 1ere forme normal
- ☒ permet de faire de *l'orienté-objet*

Allocation et restitution d'extents

⌘ Alloué quand le segment est :

- ☒ *créé ou étendu ou modifié*

⌘ Restitué quand le segment est :

- ☒ *supprimé* ou
☐ *modifié* ou
☐ *vidé* ou
☐ *automatiquement redimensionné* (rollback
segment uniquement)

⌘ Manuellement :

- ☒ Avec l'ordre
 - ☒ ALTER TABLE table ALLOCATE EXTENT;
- ☒ Avec les outils d'import/d'export

Fusion d'espace libre

⌘ Libération d'espace contigu lorsque des extents sont libérés

- ```
☒ ALTER TABLESPACE data01 COALESCE;
```

```
SELECT tablespace_name, total_extents,
percent_extents_coalesced
FROM dba_free_space_coalesced
WHERE percent_extents_coalesced <> 100;
```

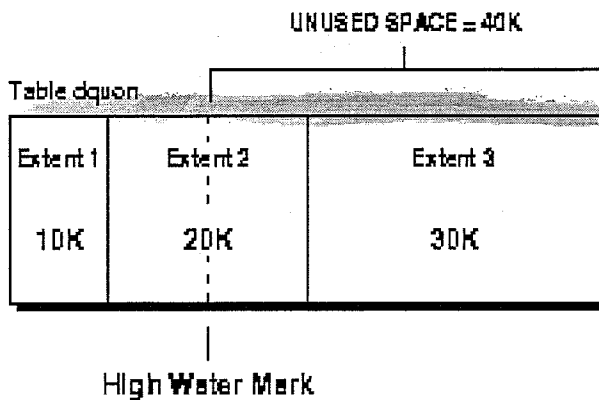
ou

- ☒ Utiliser *Oracle Storage Manager*
  - tools->Coalesce Free Extents

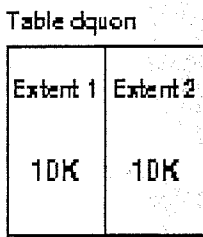
### ⌘ Sur une table (V8)

- ```
☒ALTER TABLE table
DEALLOCATE UNUSED
```

Affiche espace libre dans la base toto
 SELECT block-id, bytes, blocks FROM dba-free-space
 WHERE tablespace-name = 'toto';



	Before	After
- ALTER TABLE dept DEALLOCATE UNUSED;		



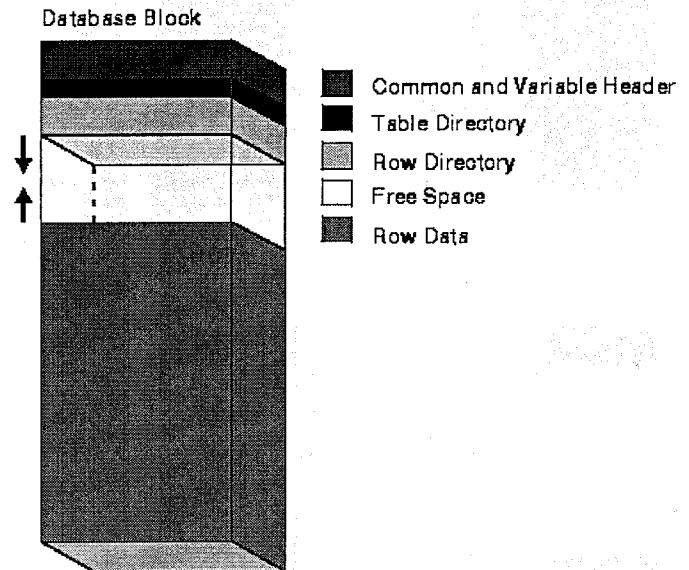
Blocs de base de données

⌘ Rappels :

- ☑ plus petite unité d'E/S
- ☑ consiste en 1 ou plusieurs blocs SE
- ☑ taille fixé par DB_BLOCK_SIZE : donné à la création de la base de données

⌘ Un bloc contient :

- ☑ *entête de bloc* : @ du bloc de données, répertoire de la table et de la ligne et entrées transactionnelles. Croit de façon *descendante*.
- ☑ *espace des données* : les données d'une ligne sont insérées de manière *ascendante*.
- ☑ *espace libre* : se trouve au milieu !



Paramètres d'un bloc

⌘ Pour les accès concurrents :

☑ INITRANS

- par défaut, vaut 1 si segment de données, 2 si segments d'index

☑ MAXTRANS

- par défaut, vaut 255

spécifie le nombre min/max d'entrées de transaction créées dans un bloc de données ou d'index

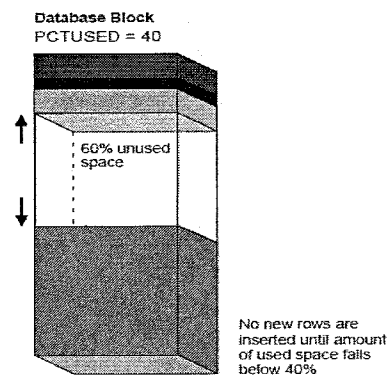
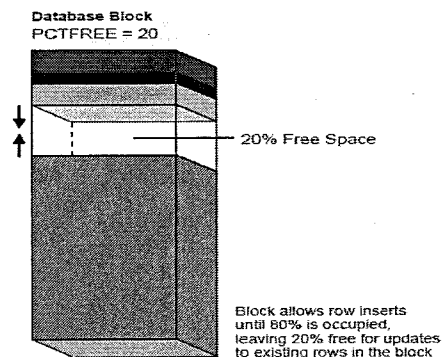
⌘ Pour la gestion de l'espace libre :

☑ PCTFREE : %tage d'espace réservé à la croissance due aux mises à jour

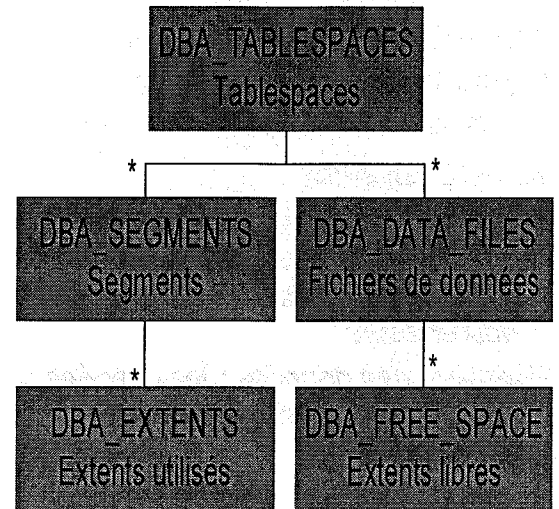
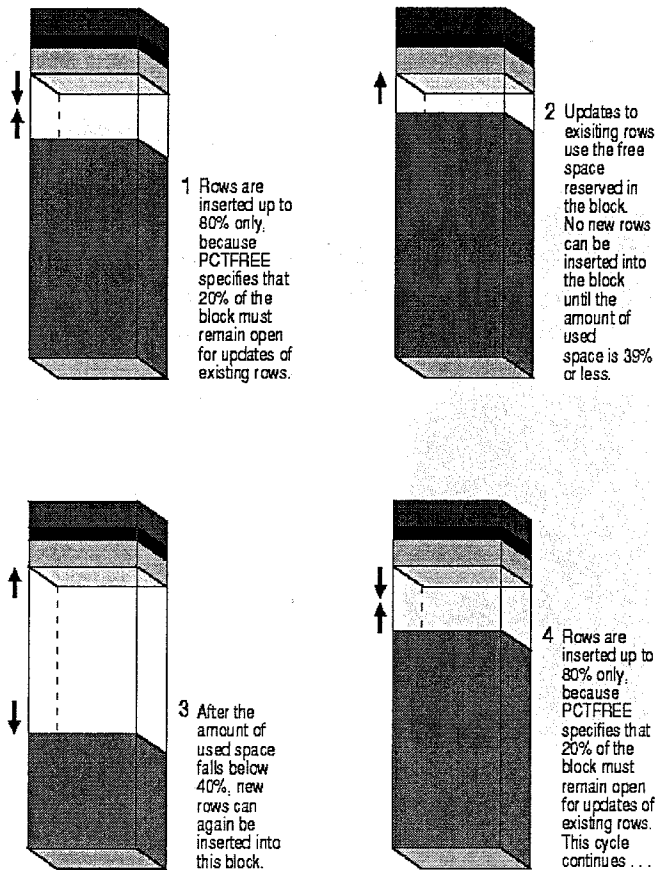
- par défaut, vaut 10%

☑ PCTUSED : %tage d'espace au delà duquel le bloc n'est pas considéré comme libre

- par défaut, vaut 40%



Les vues du dictionnaire



Information sur le Segment

⌘ Interroger la vue

☑ DBA_SEGMENTS

☑ information générale :

- owner, segment_name
- segment_type, tablespace_name

☑ taille :

- extents, blocs

☑ initialisation du stockage :

- initial_extent, next_extent
- min_extent, max_extent
- pct_increase

☑ Exemple :

```

☑ SELECT segment_name, tablespace_name,
extents, blocks
FROM DBA_SEGMENTS
WHERE owner = 'SCOTT'
  
```

Information sur l'extent utilisé

⌘ Interroger la vue

☑ DBA_EXTENTS

☑ identification :

- owner
- segment_name
- extent_id

☑ Emplacement et taille

- tablespace_name
- relative_fno
- file_id
- block_id
- blocks

☑ Exemple :

```

☑ SELECT extent_id, file_id, block_id, blocks
FROM DBA_EXTENTS
WHERE owner = 'SCOTT'
AND segment_name = 'EMP'
  
```

Information sur l'extent libre

⌘ Sur la vue

☑ DBA_FREE_SPACE

☑ emplacements et taille

- tablespace_name
- relative_fno
- file_id
- block_id
- blocks

☑ Exemple :

```
☑ SELECT tablespace_name, count(*),  
       max(blocks), sum(blocks)  
FROM dba_free_space  
GROUP BY tablespace_name;
```

Gestion des rollback segments/UNDO tablespace

Intérêts

⌘ 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)

⌘ Transaction rollback :

☑ l'ancienne valeur sauvegardée dans le rollback segment est réécrite dans la ligne

⌘ Lecture cohérente

☑ les autres transactions voient l'image avant des données grâce au rollback segment.

⌘ Restauration de transactions :

☑ En cas de panne, relecture du rollback segment dans le fichier de données.

UNDO tablespaces

⌘ Paramètres de l'INIT.ORA associés

☑ UNDO MANAGEMENT AUTO| MANUAL =>
utilise les UNDO tbs ou les rollback segments

☑ UNDO TABLESPACE nom tablespace =>
précise le tablespace d'UNDO à utiliser par la base

⌘ SQL spécifique associé

```
☑ CREATE UNDO TABLESPACE undo_1  
  DATAFILE '...' SIZE 10M AUTOEXTEND  
  ON
```

ou dès la création de la base

```
☑ CREATE DATABASE test...  
  UNDO TABLESPACE undo_1 DATAFILE  
  '...' SIZE 10M AUTOEXTEND ON
```

```
☑ Alter system set undo_tablespace = undo_2;
```

```
☑ Show parameter undo_tablespace
```

Type et allocation de rollback segments

⌘Type :

☒SYSTEM

- ☒dans le tablespace du même nom
- ☒utilisé pour les modifications des objets de ce tablespace

☒Non-SYSTEM

- ☒créé par l'administrateur
- ☒deux cas :
 - Private
 - utilisation normale
 - Public
 - serveur // d'Oracle

⌘Allocation d'un rollback segment pour une transaction donnée :

- ☒Oracle choisit le rbs "le plus" disponible et l'affecte à la transaction.
- ☒Ou l'utilisateur le spécifie :
 - SET TRANSACTION USE ROLLBACK SEGMENT rbs012;

Création des rollback segments

⌘Ordre :

☒CREATE ROLLBACK SEGMENT

☒Exemple :

```
CREATE ROLLBACK SEGMENT rbs01  
TABLESPACE rbs;
```

créé OFFLINE par défaut

☒PCTINCREASE ne peut pas être spécifié

⌘Sous OEM, utiliser

☒Oracle Storage Manager

Gestion des rollback segments

⌘Modification

☒ALTER ROLLBACK SEGMENT

☒Mise ONLINE

- ☒ALTER ROLLBACK SEGMENT name ONLINE;
- ☒ou ajouter dans init<sid>.ora :
 - ROLLBACK_SEGMENTS = (rbs01)

☒Libération d'espace

- ☒ALTER ROLLBACK SEGMENT name SHRINK TO 2M;

☒Mise OFFLINE

- ☒ALTER ROLLBACK SEGMENT name OFFLINE

⌘Suppression

- ☒DROP ROLLBACK SEGMENT name;

Informations sur les rollback segments

⌘Interroger la vue

☒DBA_ROLLBACK_SEGS

☒identification

- SEGMENT_ID
- SEGMENT_NAME

☒emplacement, type, status

- TABLESPACE_NAME
- OWNER
- STATUS

⌘Statistiques

☒V\$ROLLNAME

☒V\$ROLLSTAT

⌘Activités courantes

☒V\$SESSION

☒V\$TRANSACTION