

STRUCTURES DE STOCKAGE DE BASE DE DONNÉES

Plan

- Les concepts de stockage de données
- Contenu des blocs de base de données
- Informations de la structure de stockage
- Les tablespaces créés par défaut
- Création de tablespace
 - ❖ Le tablespace permanent
 - ❖ Le tablespace Temporaire
 - ❖ Le tablespace UNDO
- Gestion des tablespaces
- Consulter les informations sur les tablespaces
- Le package DBMS_SPACE

Les concepts de stockage de données

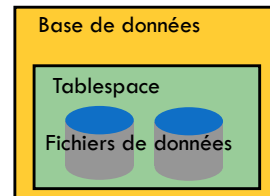
Oracle stocke les données logiquement dans les tablespaces et physiquement dans les fichiers de données.

■ Un tablespace :

- ne peut appartenir qu'à une seule base de données à la fois
- est composé d'un ou de plusieurs fichiers de données
- est divisé en unités logiques de stockage appelées segment

■ Un fichier de données :

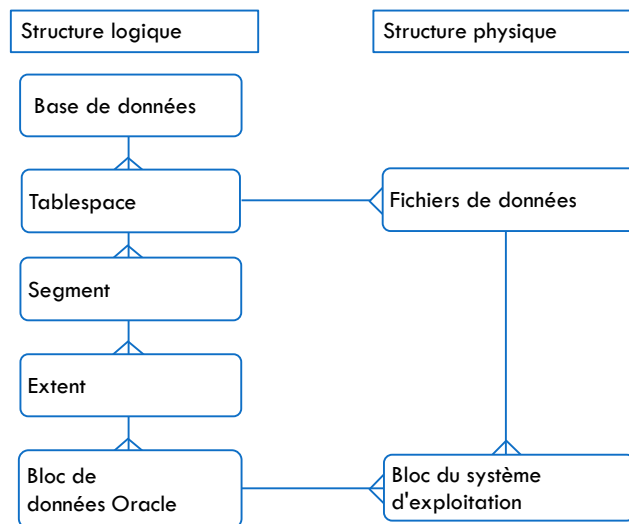
- ne peut appartenir qu'à un seul tablespace et une seule base de données,
- est un référentiel pour les données d'objet de schéma.



Les concepts de stockage de données

- Chaque tablespace comporte plusieurs segments
- Chaque segment est formé de plusieurs extents
- Un extent est une ensemble de blocs contigus
- Un blocs de données Oracle est la plus petite unité d'E/S logique
- Le paramètre DB_BLOCK_SIZE définit la taille d'un bloc logique
- La taille d'un bloc logique peut être comprise entre 2 ko et 32 ko.

Les concepts de stockage de données



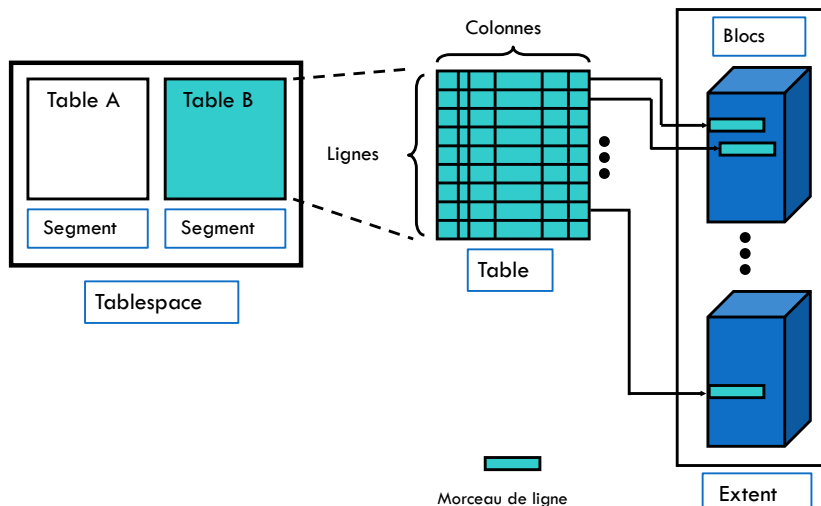
Les concepts de stockage de données

Segment

Il y a 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

Les concepts de stockage de données



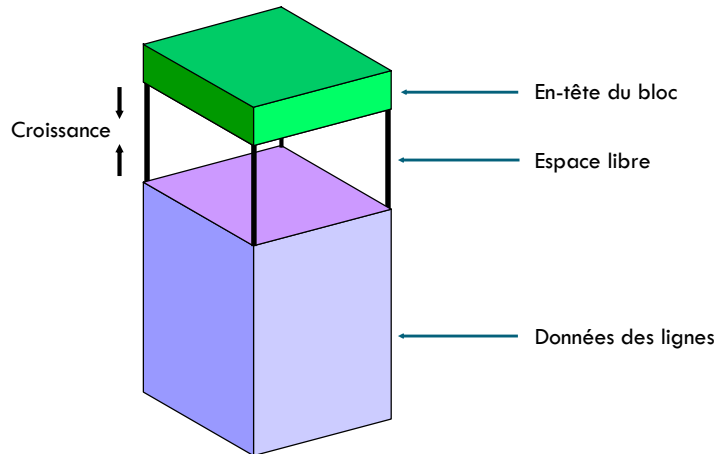
Contenu des blocs de base de données

Les blocs de données Oracle contiennent les éléments suivants :

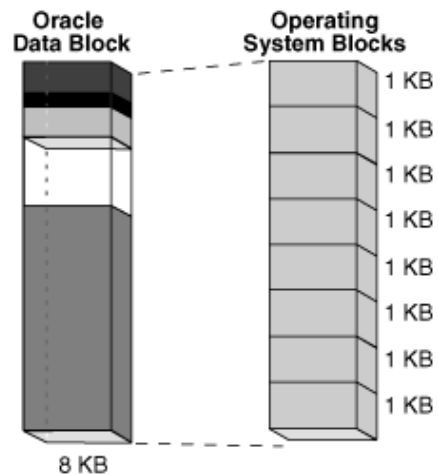
- **En-tête du bloc** : L'en-tête d'un bloc contient le type de segment (table ou index, par exemple), l'adresse du bloc, le répertoire des tables, le répertoire des lignes, et des espaces de transaction de 23 octets chacun, utilisés lors de la modification des lignes du bloc. L'en-tête de bloc croît du haut vers le bas
- **Données des lignes** : Il s'agit des données réelles des lignes contenues dans le bloc. L'espace réservé aux données des lignes augmente du bas vers le haut
- **Espace libre** : L'espace libre se trouve au centre du bloc. Il permet à l'espace d'en-tête et à l'espace des données des lignes d'augmenter en fonction des besoins

Contenu des blocs de base de données

Anatomie d'un bloc de base de données



Contenu des blocs de base de données



Informations de la structure de stockage

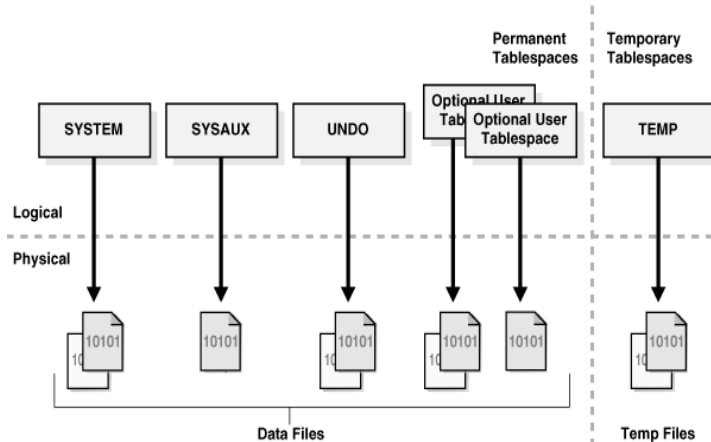
- ❑ La vue DBA_TABLESPACES permet de donner des informations sur les tablespaces de la base de données
- ❑ La vue DBA_SEGMENTS permet d'afficher les paramètres de stockage et l'utilisation de l'espace dans les segments
- ❑ La vue DBA_EXTENTS permet d'afficher le nombre de blocs, la taille de l'extent...

Les tablespaces créés par défaut

Lors de la création de la base de données plusieurs tablespaces sont créés par défaut :

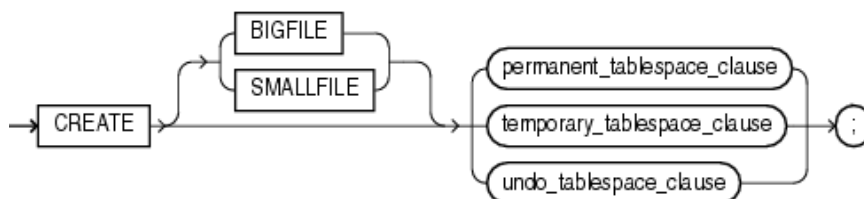
- ❑ SYSTEM contenant le dictionnaire de données
- ❑ SYSAUX utilisé pour les composants de Oracle
- ❑ UNDO tablespace utilisé pour les données d'annulation
- ❑ TEMP un tablespace temporaire pour les opérations de tri
- ❑ Il est possible de créer un tablespace par défaut pour les objets créés sans précision du tablespace en général nommé USERS
- ❑ La vue DATABASE_PROPERTIES permet de donner des infos sur les tablespaces par défaut de la BD

Les tablespaces créés par défaut



Création de tablespace

- Trois types de tablespaces peuvent être créés :
 - ▣ Tablespace permanent
 - ▣ Tablespace temporaire
 - ▣ Tablespace UNDO



Le tablespace Permanent

- ❑ La commande pour créer un tablespace permanent :

```
CREATE (SMALLFILE|BIGFILE) TABLESPACE NOMTABLESPACE  
[DATAFILE [NOM_FICHER] [SIZE INTEGER{K|M|G|T}]]  
  [AUTOEXTEND {OFF|ON[NEXT INTEGER]}]  
  [MAXSIZE{UNLIMITED|INTEGER}]  
[BLOCKSIZE INTEGER [K]]  
[LOGGING|NOLOGGING]  
[ONLINE|OFFLINE]  
[ENCRYPTION[USING ALGORITHM] [IDENTIFIED BY  
  PASS] DEFAULT STORAGE ENCRYPT]  
[DEFAULT {COMPRESS |NOCOMPRESS}]
```

Le tablespace Permanent

- ❑ **SMALLFILE** : tablespace sur plusieurs fichiers allant à 1022 fichiers (Total Max 128 TB)
- ❑ **BIGFILE** : tablespace sur un seul fichier de grande taille (Max 128 TB)
- ❑ **AUTOEXTEND** : active l'extension automatique du fichier
- ❑ **ONLINE** : Tablespace immédiatement utilisable
- ❑ **ENCRYPT** : crypter les données du tablespace
- ❑ **COMPRESS** : le stockage des segments dans le tablespace est compressé

Le tablespace Temporaire

- ❑ Le tablespace temporaire ne pourra pas être créé avec une taille du bloc différente de celle du bloc de la BD (DB_BLOCK_SIZE)

- ❑ Commande :

```
CREATE (SMALLFILE|BIGFILE) TEMPORARY  
TABLESPACE NOMTABLESPACE  
[TEMPFILE [NOM_FICHER] [SIZE  
  INTEGER{K|M|G|T}] [AUTOEXTEND  
  {OFF|ON[NEXT INTEGER]}]  
  [MAXSIZE{UNLIMITED|INTEGER}]  
[ONLINE|OFFLINE]  
TABLESPACE GROUP nomGroup;
```

Le tablespace Temporaire

- ❑ TABLESPACE GROUP : permet de créer et d'affecter le tablespace temporaire à un groupe de tablespaces temporaires
- ❑ Les avantages :
 - ▣ Permettre d'affecter la BD par un groupe de tablespaces temporaires et ainsi augmenter sa performance
 - ▣ Permettre à une application d'utiliser plusieurs tablespaces temporaires à la fois pour le tri
- ❑ Pour modifier le tablespace temporaire par défaut :
 - ▣ ALTER DATABASE DEFAULT TEMPORARY TABLESPACE
 NOM_GP_OU_TBS

Le tablespace UNDO

- Lorsque l'utilisateur exécute une requête LMD, les blocs concernés sont modifiés mais avant d'écraser les blocs originaux, Oracle prend une copie de ces données et la place dans le tablespace UNDO
- Si l'utilisateur décidera d'annuler sa transaction (Rollback), Oracle cherche les blocs originaux dans le tablespace UNDO
- Le tablespace UNDO ne pourra pas être créé avec une taille du bloc différente de celle du bloc de la BD(DB_BLOCK_SIZE)

Le tablespace UNDO

- Commande de création :

```
CREATE (SMALLFILE|BIGFILE) UNDO TABLESPACE
    NONTABLESPACE
    [DATAFILE [NOM_FICHIER] [SIZE INTEGER{K|M|G|T}]
    [AUTOEXTEND {OFF|ON[NEXT INTEGER]}]
    [MAXSIZE{UNLIMITED| INTEGER}]
    [ONLINE|OFFLINE] ;
```

- Pour modifier le tablespace UNDO par défaut : `ALTER SYSTEM SET UNDO_TABLESPACE=NOMTBS_UNDO`

Gestion des tablespaces

La gestion des tablespaces englobe les opérations suivantes :

- ❑ Agrandissement du tablespace
- ❑ Changement du mode OFFLINE | ONLINE
- ❑ Déplacement du tablespace
- ❑ Tablespace en lecture seule
- ❑ Suppression du tablespace

Gestion des tablespaces : Agrandissement du tablespace

Pour agrandir un tablespace :

- ❑ Si le tablespace est SMALLFILE alors deux manières peuvent être utilisées :
 - Ajouter un fichier au tablespace
 - Augmenter la taille des fichiers du tablespace
- ❑ Si le tablespace est BIGFILE alors une seule possibilité : Augmenter la taille du fichier

Gestion des tablespaces : Agrandissement du tablespace

□ Ajouter un fichier au tablespace :

```
ALTER TABLESPACE nom_tbs ADD
{ DATAFILE |TEMPFILE } [nom_fichier]
[SIZE INTEGER{K|M|G|T}]
[AUTOEXTEND {OFF|ON[NEXT INTEGER]}]
[MAXSIZE{UNLIMITED|INTEGER}]
```

□ Supprimer un fichier du tablespace :

```
ALTER TABLESPACE nom_tbs DROP
DATAFILE|TEMPFILE nom_fichier
```

Gestion des tablespaces : Agrandissement du tablespace

□ Augmenter la taille ou activer l'auto-extension d'un fichier dans le tablespace:

```
ALTER DATABASE [nom_bd] {DATAFILE|TEMPFILE}
nom_fichier
{ RESIZE INTEGER |
AUTOEXTEND {OFF|ON[NEXT INTEGER]}]
[MAXSIZE{UNLIMITED|INTEGER}]
```

Gestion des tablespaces : Agrandissement du tablespace

- Le cas d'un tablespace BIGFILE:

```
ALTER tablespace nom_tbs  
{RESIZE INTEGER |  
[AUTOEXTEND {OFF|ON[NEXT INTEGER]}]  
[MAXSIZE{UNLIMITED|INTEGER}]}
```

Gestion des tablespaces : Changement du mode OFFLINE | ONLINE

- En cas de problème dans un tablespace, on peut le mettre OFFLINE
- Ceci permet d'éviter de fermer la BD et ainsi arrêter les applications n'utilisant pas le tablespace concerné
- Commande :

```
ALTER TABLESPACE nom_tbs {  
  ONLINE |  
  OFFLINE [NORMAL | TEMPORARY | IMMEDIATE]  
}
```

Gestion des tablespaces : Changement du mode OFFLINE | ONLINE

- NORMAL : Lance l'écriture des blocs modifiés dans le tablespace avant sa fermeture. Ceci ne nécessite pas la récupération des données avant de le rendre ONLINE
- TEMPORARY : lance un checkpoint sans s'assurer que les blocs sont écrites. Peut nécessiter la récupération
- IMMEDIATE : Récupération obligatoire lors du changement vers le mode ONLINE

Gestion des tablespaces : Déplacement du tablespace

- Le déplacement du tablespace nécessite que la BD soit ouverte et le tablespace en mode OFFLINE
- Démarche :
 - Rendre OFFLINE le tablespace
 - Copier les fichiers à l'aide du système d'exploitation vers le nouveau emplacement
 - Renommer les fichier de données
 - Mettre le tablespace en ligne

Gestion des tablespaces : Déplacement du tablespace

La commande pour renommer les fichiers de données :

- `ALTER TABLESPACE nom_tbs RENAME DATAFILE
ancien_nom TO nouveau_nom`

Gestion des tablespaces : Tablespace en Lecture Seule

- On peut mettre un tablespace en lecture seule dans ce cas les opérations LMD sont interdites

- Commande :

- `Alter Tablespace nom_tbs READ { ONLY | WRITE }`

Gestion des tablespaces : Suppression du tablespace

- ❑ **Commande :**

```
DROP Tablespace nom_tbs  
[  
INCLUDING CONTENTS  
[ AND DATAFILES]          [CASCADE CONSTRAINTS]  
]
```
- ❑ **INCLUDING CONTENTS :** Supprime le tablespace s'il contient des objets
- ❑ **DATAFILES :** supprime les fichiers du tablespace (Inutile avec OMF)
- ❑ **CASCADE CONSTRAINTS :** supprime les contraintes d'intégrité référentielles qui font référence aux tables du tablespace a supprimé

Consulter les informations sur les tablespaces

- ❑ Pour avoir des informations sur les tablespaces, nous pouvons utiliser les vues suivantes :
 - **V\$TABLESPACE :** affiche les informations sur l'espace table depuis le fichier de contrôle.
 - **DBA_TABLESPACES :** décrit tous les tablespaces de la base de données
 - **USER_TABLESPACES :** descriptions de tous les tablespaces accessibles aux utilisateurs
 - **DBA_DATA_FILES :** infos sur les fichiers de données

Application

□ Lister les tablespaces:

```
select contents, TABLESPACE_NAME from  
dba_tablespaces order by TABLESPACE_NAME
```

CONTENTS	TABLESPACE_NAME

PERMANENT	SYSAUX
PERMANENT	SYSTEM
TEMPORARY	TEMP
UNDO	UNDOTBS1
PERMANENT	USERS

Application

□ Ajouter un tablespace

```
CREATE TABLESPACE IDX02 DATAFILE  
'C:\APP\ADMIN\ORADATA\orcl\idx02.dbf' size 10m  
EXTENT MANAGEMENT LOCAL AUTOALLOCATE SEGMENT  
SPACE MANAGEMENT AUTO;
```

CONTENTS	TABLESPACE_NAME

PERMANENT	IDX02
PERMANENT	SYSAUX
PERMANENT	SYSTEM
TEMPORARY	TEMP
UNDO	UNDOTBS1
PERMANENT	USERS

Application

□ Ajouter un tablespace UNDO

```
CREATE UNDO TABLESPACE UNDO02 DATAFILE  
'C:\APP\ADMIN\ORADATA\ORCL\undo02.dbf' size 10m  
AUTOEXTEND on next 5M extent MANAGEMENT LOCAL  
AUTOALLOCATE;
```

CONTENTS	TABLESPACE_NAME

PERMANENT	IDX02
PERMANENT	SYSAUX
PERMANENT	SYSTEM
TEMPORARY	TEMP
UNDO	UNDOTBS1
UNDO	UNDO02
PERMANENT	USERS

Application

□ Ajouter un tablespace Temporaire

```
CREATE TEMPORARY TABLESPACE TEMP02 TEMPFILE  
'C:\APP\ADMIN\ORADATA\ORCL\temp02.dbf' size 10m;
```

CONTENTS	TABLESPACE_NAME

PERMANENT	IDX02
PERMANENT	SYSAUX
PERMANENT	SYSTEM
TEMPORARY	TEMP
TEMPORARY	TEMP02
UNDO	UNDOTBS1
UNDO	UNDO02
PERMANENT	USERS

Application

- Afficher le tablespace par défaut

```
SELECT PROPERTY_VALUE FROM DATABASE_PROPERTIES
WHERE PROPERTY_NAME=
'DEFAULT_PERMANENT_TABLESPACE';
```

PROPERTY_VALUE

USERS

Application

- Modifier le tablespace par défaut

```
Alter database default tablespace idx02;
```

PROPERTY_VALUE

idx2

Application

□ Afficher les états des différents tablespaces

```
Select contents, tablespace_name, status from  
dba_tablespaces;
```

CONTENTS	TABLESPACE_NAME	STATUS
-----	-----	-----
PERMANENT	IDX02	ONLINE
PERMANENT	SYSAUX	ONLINE
PERMANENT	SYSTEM	ONLINE
TEMPORARY	TEMP	ONLINE
TEMPORARY	TEMP02	ONLINE
UNDO	UNDOTBS1	ONLINE
UNDO	UNDO02	ONLINE
PERMANENT	USERS	ONLINE

Question

□ Créer un tablespace avec le minimum de configuration

```
CREATE TABLESPACE nom_TableSpace; !
```

La gestion automatique des fichiers

OMF (Oracle Managed Files)

- **OMF** a pour but de simplifier l'administration d'une base de données prenant en compte la gestion des fichiers physiques.
- Oracle utilise son interface avec le système de fichiers pour gérer la création, la modification, ou l'effacement des fichiers nécessaires pour les tablespaces, les groupes des fichiers journaux et les fichiers de contrôle.

La gestion automatique des fichiers

OMF (Oracle Managed Files)

- **OMF** a pour but de simplifier l'administration d'une base de données prenant en compte la gestion des fichiers physiques.
- Oracle utilise son interface avec le système de fichiers pour gérer la création, la modification, ou l'effacement des fichiers nécessaires pour les tablespaces, les groupes des fichiers journaux et les fichiers de contrôle.
- Les seuls fichiers concernés par cette fonctionnalité sont : les fichiers de données, les fichiers journaux et le fichier de contrôle.

La gestion automatique des fichiers

OMF (Oracle Managed Files)

La gestion **OMF** (Oracle Managed Files) présente les avantages suivants :

- ❑ Réduit l'espace perdu avec des fichiers obsolètes.
- ❑ Simplifie la création de base de données de tests et de développements.
- ❑ Réduit les erreurs d'administration des fichiers de données.
- ❑ Augmente la portabilité des scripts SQL de création.

La gestion automatique des fichiers

OMF (Oracle Managed Files)

Pour pouvoir utiliser la gestion OMF (Oracle Managed Files), il faut configurer les paramètres d'initialisation suivants :

- ❑ DB_CREATE_FILE_DEST
- et
- ❑ DB_CREATE_ONLINE_LOG_DEST_n

DB_CREATE_FILE_DEST : le nom d'un répertoire existant indiquant à Oracle où créer les fichiers de données et les fichiers temporaires.

DB_CREATE_ONLINE_LOG_DEST_n : le nom d'un répertoire existant indiquant à Oracle où créer les groupes des fichiers journaux et les fichiers de contrôle. La valeur « n » peut être un numéro entre un et cinq et représente le nombre de membres multiplexés que vous souhaitez avoir.

La gestion automatique des fichiers

□ OMF (Oracle Managed Files)

SQL> show parameter create

NAME	TYPE	VALUE
create_bitmap_area_size	integer	8388608
create_stored_outlines	string	
db_create_file_dest	string	
db_create_online_log_dest_1	string	
db_create_online_log_dest_2	string	
db_create_online_log_dest_3	string	
db_create_online_log_dest_4	string	
db_create_online_log_dest_5	string	

SQL> CREATE TABLESPACE test_OMF1;

CREATE TABLESPACE test_OMF1

*

ERREUR Ó la ligne 1 :

ORA-02199: clause DATAFILE/TEMPFILE absente

La gestion automatique des fichiers

□ OMF (Oracle Managed Files)

SQL> alter system set db_create_file_dest='c:/app/admin/oradata' scope=both;

Système modifié.

SQL> show parameter create

NAME	TYPE	VALUE
create_bitmap_area_size	integer	8388608
create_stored_outlines	string	
db_create_file_dest	string	c:/app/admin/oradata

...

La gestion automatique des fichiers

□ OMF (Oracle Managed Files)

```
SQL> CREATE TABLESPACE test_OMF1;
```

Tablespace créé.

```
SQL> SELECT TABLESPACE_NAME, FILE_NAME FROM DBA_DATA_FILES where  
tablespace_name='TEST_OMF1';
```

TABLESPACE	FILE_NAME
TEST_OMF1	C:\APP\ADMIN\ORADATA\MIP\DATAFILE\O1_MF_TEST_OMF_LZ1RFP4_.DBF

DONNÉES D'ANNULATION

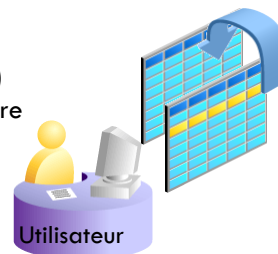
Introduction

- ❖ Le langage de manipulation de données (LMD) est composé des instructions SQL suivantes :
 - INSERT
 - UPDATE
 - DELETE
 - MERGE
- ❖ Une instruction LMD s'exécute toujours au sein d'une transaction, qui peut être :
 - annulée à l'aide de la commande `ROLLBACK`
 - validée à l'aide de la commande `COMMIT`

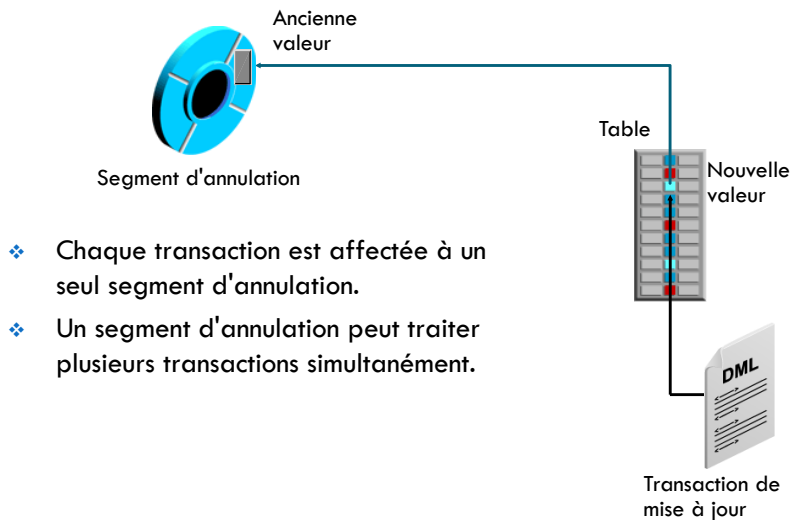
Introduction

Les données d'annulation (undo) :

- ❖ Représentent une copie des données d'origine avant une modification
- ❖ Sont capturées pour toute transaction qui modifie des données
- ❖ Sont conservées au minimum jusqu'à la fin de la transaction
- ❖ Permettent :
 - les opérations d'annulation (rollback)
 - les interrogations cohérentes en lecture et les interrogations flashback
 - la récupération suite à l'échec de transactions



Transactions et données d'annulation



Stocker les informations d'annulation

Les informations d'annulation sont stockées dans des segments d'annulation, lesquels sont à leur tour stockés dans un tablespace d'annulation.

Les tablespaces d'annulation :

- ❖ sont utilisés uniquement pour les segments d'annulation
- ❖ sont associés à une instance unique
- ❖ nécessitent qu'un seul tablespace d'annulation soit accessible en écriture pour une instance donnée à un instant précis

Les données d'annulation et les données de journalisation

	Données d'annulation	Données de journalisation
Enregistrement de	Mode d'annulation d'une modification	Mode de reproduction d'une modification
Utilisation pour	Annulation (rollback), cohérence en lecture	Réimplémentation des modifications de la base de données
Stockage dans	Segments d'annulation	Fichiers de journalisation
Protection contre	Lectures incohérentes dans les systèmes multi-utilisateurs	Pertes de données

Mise en œuvre de la gestion automatique

Pour mettre en œuvre la gestion automatique des informations d'annulation, il faut :

- Positionner le paramètre UNDO_MANAGEMENT à AUTO et éventuellement affecter des valeurs aux paramètres UNDO_TABLESPACE et UNDO_RETENTION ;
- Créer au moins un tablespace d'annulation lors de la définition de la base de données ou ultérieurement.

Les paramètres d'initialisation

□ UNDO_MANAGEMENT

- ❖ Indique le mode de gestion souhaité pour les informations d'annulation. Les valeurs possibles sont AUTO (valeur par défaut) ou MANUAL.
- ❖ Ce paramètre est statique ; il faut redémarrer la base de données pour changer le mode de gestion des informations d'annulation.

□ UNDO_TABLESPACE

- ❖ Spécifie le nom du tablespace d'annulation à utiliser lors du démarrage de l'instance.
- ❖ Par défaut, ce paramètre est vide et le premier tablespace d'annulation trouvé dans la base de données est utilisé;
- ❖ le paramètre UNDO_TABLESPACE est alors renseigné par Oracle. Ce paramètre est dynamique.

Gérer les informations d'annulation

La formule suivante permet de calculer la taille minimale du tablespace de type UNDO :

□ Taille (en blocs) = Undo_R * SpaceUPS

Avec :

- Undo_R : valeur du paramètre UNDO_RETENTION (en secondes)
- SpaceUPS : espace sollicité par secondes (en octets)

Configurer la conservation des informations d'annulation

- ❑ Le paramètre `UNDO_RETENTION` indique (en secondes) le temps de conservation des informations d'annulation dans le segment UNDO
- ❑ Ce paramètre dépend du mode de conservation du bloc (`RETENTION GUARANTEE` ou `NOGUARANTEE`)
- ❑ En mode `GUARANTEE`, la conservation des blocs est prioritaire par rapport aux transactions

Gérer les informations d'annulation

Exercice :

Si le facteur de rétention est de 15 mn (900 secondes) avec un taux de transaction de 10 blocs par seconde.

Quelle est la taille approximative pour le UNDO?

- ❑ Taille = $900 * 10 * 8(db_block_size)$

Gérer les informations d'annulation

```
show parameter UNDO_RETENTION
```

NAME	TYPE	VALUE
undo_retention	integer	900

Garantir la conservation des informations d'annulation

- ❖ Pour modifier un tablespace d'annulation afin de garantir la période de conservation des informations d'annulation est la suivante :

- SQL> ALTER TABLESPACE undotbs1 RETENTION GUARANTEE;

- ❖ Pour rétablir le comportement par défaut, utilisez la commande suivante :

- SQL> ALTER TABLESPACE undotbs1 RETENTION NOGUARANTEE;

Paramètre UNDO_TEMP_ENABLED (12c)

- ❑ Depuis la version 12c, le paramètre **TEMP_UNDO_ENABLED** a été proposé pour simplifier la gestion UNDO des tables temporaires
- ❑ Si vous activez ce paramètre, Oracle placera les blocs UNDO des tables temporaires dans le tablespace temporaire
- ❑ Ce paramètre est activé dans la session ou dans l'instance via la commande :

```
ALTER SYSTEM/SESSION SET TEMP_UNDO_ENABLED=TRUE
```

Exercice

1. Créer un Tablespace avec deux fichiers de données
2. Vérification que deux fichiers ont bien été créés pour le tablespace
3. Créer un tablespace de taille infinie
4. Afficher la taille du tablespace créé
5. Créer un tablespace avec le minimum de paramètres afin d'utiliser les paramètres par défaut
6. Créer un tablespace de taille fixe
7. Créer un tablespace avec une taille de bloc non par défaut (différent de DB_BLOCK_SIZE)