

Cours de bases de données,
aspects systèmes,
<http://sys.bdpedia.fr>

Fichiers

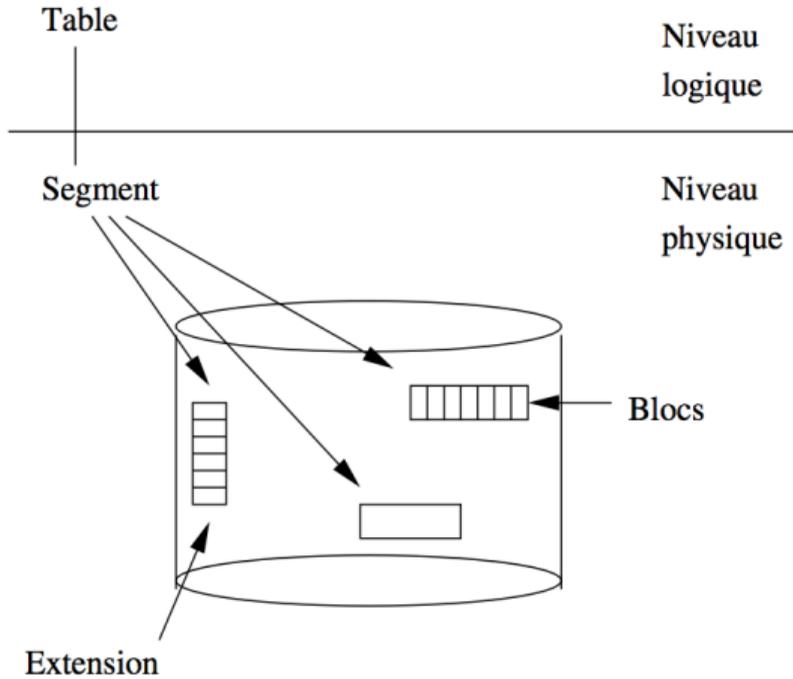
Oracle

Principales structures physiques dans ORACLE :

- **bloc** : unité physique d'E/S.
La taille d'un bloc ORACLE est un multiple de la taille des blocs du système sous-jacent.
- **Extension** : ensemble de blocs contigus contenant un même type d'information.
- **Segment** : ensemble d'extensions stockant un objet logique (une table, un index ...).

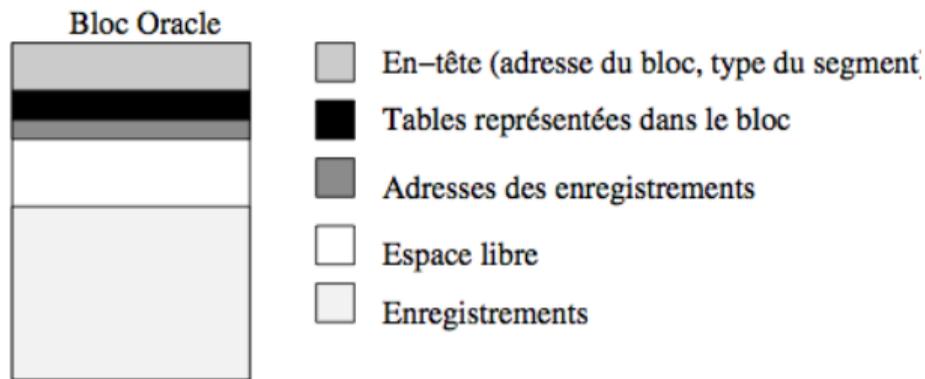
Le paramétrage du stockage des données (tables et index) est spécifié dans un *tablespace*

Tables, segments, extensions et blocs



Les blocs Oracle

La structure d'un bloc repose sur un adressage physique/logique, chaque enregistrement ayant une adresse interne.



Un chaînage est créé quand il faut déplacer un enregistrement.

Gestion de l'espace

- PCTFREE donne l'espace libre à préserver au moment de la création d'une table ou d'un index.
- PCTUSED indique le seuil en dessous duquel le bloc est disponible pour des insertions.

Oracle maintient un répertoire (?) des blocs disponibles pour insertions.

Exemple :

- PCTFREE = 30% et PCTUSED=70%
- PCTFREE = 10% et PCTUSED=80%

Le second choix est plus efficace, mais plus risqué et plus coûteux.

Extensions et segments

L'**extension** est une suite de blocs contigus.

Le **segment** est un ensemble d'extensions contenant un objet logique.

Il existe quatre types de segments :

- Le segment de données.
- Le segment d'index.
- Le *rollback segment* utilisé pour les transactions.
- Le segment temporaire (utilisé pour les tris).

Moins il y a d'extensions dans un segment, plus il est efficace.

Stockage et adressage des enregistrements

En règle générale un enregistrement est stocké dans un seul bloc.

L'adresse physique d'un enregistrement est le *ROWID*. Il contient :

- Le numéro du bloc dans le **fichier**.
- Le numéro du n-uplet dans le bloc.
- le numéro du segment.
- Le numéro du fichier.

Exemple : 00000DD5.000.2.001 est l'adresse du premier n-uplet du bloc DD5 dans le second segment du premier fichier.

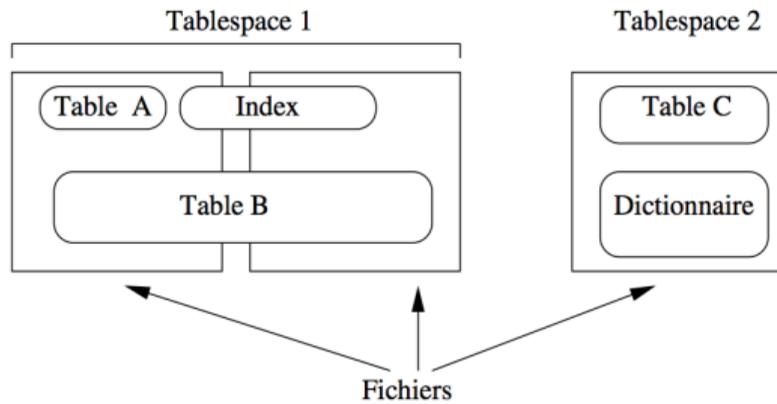
Les *tablespaces*

Une base est divisée par l'administrateur en *tablespace*. Chaque *tablespace* consiste en un (au moins) ou plusieurs fichiers.

La notion de *tablespace* permet :

- De contrôler l'emplacement physique des données. (par ex. : le dictionnaire sur un disque, les données utilisateur sur un autre).
- de régler l'allocation de l'espace (extensions).
- de faciliter la gestion (sauvegarde, protection, etc).

Exemple de *tablespaces*



Création de *tablespaces*

```
CREATE TABLESPACE TB1
  DATAFILE 'fichierTB1.dat' SIZE 50M
  DEFAULT STORAGE (
    INITIAL 100K, NEXT 40K, MAXEXTENTS 20,
    PCTINCREASE 20);
```

```
CREATE TABLESPACE TB2
  DATAFILE 'fichierTB2.dat' SIZE 2M
  AUTOEXTEND ON NEXT 5M MAXSIZE 500M
  DEFAULT STORAGE (INITIAL 128K
    NEXT 128K MAXEXTENTS UNLIMITED);
```

Maintenance d'un *tablespace*

Quelques actions disponibles sur un *tablespace* :

- On peut mettre un *tablespace* hors-service.
`ALTER TABLESPACE TB1 OFFLINE;`
- On peut mettre un *tablespace* en lecture seule.
`ALTER TABLESPACE TB1 READ ONLY;`
- On peut ajouter un nouveau fichier.
`ALTER TABLESPACE ADD DATAFILE
'fichierTB1-2.dat' SIZE 300 M;`

Affectation de tables à un *tablespace*

On peut placer une table dans un *tablespace*. Elle prend alors les paramètres de stockage de ce dernier.

On peut aussi remplacer certaines valeurs.

```
create table Film (...)  
pctfree 10  
pctused 40  
tablespace TB1  
storage ( initial 50K  
          next 50K  
          maxextents 10  
pctincrease 25 );
```