

TP1 Architecture Oracle (1h30)

1. Les tables et les blocs (ou pages) Oracle

Le système alloue un certain nombre de blocs de données à chaque table en fonction des besoins de stockage. Les meilleurs compromis, pour accéder le plus rapidement possible aux enregistrements au sein des blocs, sont également recherchés. Nous explorons les allocations de blocs opérées en fonction des activités menées sur les tables. Nous regardons dans un second temps l'organisation logique (tablespace, segment, extent) qui va faciliter l'accès aux blocs de données au regard des besoins.

1.1 Exploiter les vues USER_TABLES et DBA_TABLES

Les informations concernant l'organisation logique et physique des tables de votre schéma utilisateur sont disponibles depuis les vues système user_tables et dba_tables. Les attributs suivants donnent une idée du nombre de blocs alloués à chaque table :

1. NUM_ROWS : nombre de lignes
2. BLOCKS : nombre de blocs utilisés
3. EMPTY_BLOCKS : nombre de blocs alloués, mais non utilisés
4. AVG_SPACE : espace libre moyen dans les blocs utilisés
5. AVG_ROW_LEN : longueur moyenne des lignes (tuples) en octets

1.1.1 Questions préalables

1. Espace de table cible

Vérifiez que toutes vos tables sont situées dans le même espace de tables (attribut TABLESPACE_NAME dans USER_TABLES et DBA_TABLES) et qu'il en va de même pour les autres schémas utilisateurs. Vous donnerez le nom de ce tablespace.

```
col tablespace_name for a30
col table_name for a30
select tablespace_name, table_name from dba_tables ;
-- donne le tablespace DATA_ETUD dans lequel donnees et index usagers sont stockees
```

2. Taille du bloc Oracle

Vous rechercherez également la taille en octets du bloc oracle (multiple de 1024 octets). Regardez pour ce faire les commandes sqlplus et sql :

```
show parameter
show parameter db_block_size

select name, value from v$parameter where name like 'db_block%';
```

Listing 1 – taille du bloc

la taille du bloc pour la majorite des tablespaces est de 8192 octets soit env. 8 Ko

3. Nombre de blocs Oracle pour une table d'un schéma

Combien de blocs de données sont utilisés pour contenir les données d'une de vos tables au choix ? Ces informations sont évaluées ou ré-évaluées par la commande

```
analyze table nom_table compute statistics
```

Vous pouvez évaluer toutes les tables de votre schéma utilisateur avec l'aide du paquetage DBMS_STATS.

```
-- semble ne pas collecter toutes les statistiques, preferer analyze table
exec dbms_stats.gather_schema_stats(USER)
```

Renvoyez le nombre de blocs alloués par table (BLOCKS + EMPTY_BLOCKS). Donnez le coût en octets.

```
analyze table emp compute statistics ;
-- possible aussi pour le schema
exec dbms_utility.analyze_schema(user,'COMPUTE')

select table_name, num_rows+avg_row_len as octetsRequis,
       blocks+empty_blocks as nombreBlocs, (blocks+empty_blocks)*8192 as octetsAlloues from user_tables where table_name = 'EMP' ;
```

1.2 Mécanismes d'allocation

Nous allons définir une table de test, puis l'alimenter à différents temps à l'aide d'une procédure PL/SQL qui vous est donnée (remplissage.sql).

Créer la table test(num char(3), commentaire char(97)), et appliquer sur num une contrainte de domaine : nombres entiers compris entre 0 et 999

1.2.1 Question 1

Est-ce que la longueur des tuples de cette table est variable ? Relever le nombre de blocs utilisés, alloués mais inutilisés, ...après :

1. la création de la table

```
-- analyse puis
select num_rows, blocks, empty_blocks, avg_space, avg_row_len from user_tables where table_name = 'TEST';
NUM_ROWS      BLOCKS EMPTY_BLOCKS  AVG_SPACE AVG_ROW_LEN
-----
0             0      0           0         0

--a la creation sur Oracle 10g, pas de tuples, 8 blocs vides alloues (1 segment d'un extent de 8 blocs)
a la creation sur Oracle 12c, pas de tuples, 0 blocs alloues
```

2. l'insertion de 50 lignes

```
-- analyse puis
select num_rows, blocks, empty_blocks, avg_space, avg_row_len from user_tables where table_name = 'TEST';
NUM_ROWS      BLOCKS EMPTY_BLOCKS  AVG_SPACE AVG_ROW_LEN
-----
51            5      3       6982       105

insertion de 51 tuples, la taille est fixe (recours à char, donc chaine fixe), 8 blocs alloues dont 3 vides,
taille libre dans les blocs en ecriture est importante 6982 octets, sachant que chaque bloc compte 8192 octets
```

3. l'insertion de 100 lignes supplémentaires

```
-- analyse puis
select num_rows, blocks, empty_blocks, avg_space, avg_row_len from user_tables where table_name = 'TEST';
NUM_ROWS      BLOCKS EMPTY_BLOCKS  AVG_SPACE AVG_ROW_LEN
-----
151           5      3       4840       105

insertion de 100 tuples suppl'ementaire, toujours 8 blocs alloues dont 3 vides,
taille libre dans les blocs en ecriture diminue 4840 octets
```

4. l'insertion de 100 lignes supplémentaires

```
-- analyse puis
select num_rows, blocks, empty_blocks, avg_space, avg_row_len from user_tables where table_name = 'TEST';
TABLE_NAME  NUM_ROWS AVG_ROW_LEN  AVG_SPACE  BLOCKS EMPTY_BLOCKS
-----
TEST        250     105      2721      5        3
```

insertion de 100 tuples supplémentaires, toujours 8 blocs alloués dont 3 vides, taille libre dans les blocs en écriture diminue encore 2721 octets

5. l'insertion de 100 lignes supplémentaires

```
-- analyse puis
select num_rows, blocks, empty_blocks, avg_space, avg_row_len from user_tables where table_name = 'TEST';
TABLE_NAME  NUM_ROWS AVG_ROW_LEN  AVG_SPACE  BLOCKS EMPTY_BLOCKS
-----
TEST        350     105      5191     13        3
```

100 tuples supplémentaires, il y a toujours un espace réservé dans les blocs (gestion avec pct_used, pct_free) du coup ajout de 8 blocs supplémentaires, il s'agit d'un second extent qui chacun contient 8 blocs, donc ici 13 blocs en écriture avec un espace libre moyen de 5191 octets

Comparer à chaque étape l'espace réellement occupé par la table et l'espace théoriquement nécessaire à sa mémorisation. Comment expliquer les différences ?

Il faut calculer l'espace de stockage pour 350 tuples de 105 octets chacun et le nombre de blocs de 8192 octets qui pourraient être suffisants pour stocker cette volumétrie. Là, nous avons 16 blocs alloués, ce qui induit un espace de stockage beaucoup plus important. L'objectif du système est de réorganiser le moins possible et donc d'allouer d'emblée bien plus d'espace qu'il ne le faut

1.2.2 Question 2

Supprimer un tiers des tuples de la table (par exemple, ceux dont le num est divisible par 3).

```
select num from test where mod(num,3) = 0;
```

Listing 2 – exemple pour suppression

Réévaluer les statistiques sur cette table à l'issue de l'opération. Validez la suppression et tester à nouveau. Supprimer l'intégralité des tuples, avec avec l'ordre

```
delete from test;
```

Listing 3 – delete

valider (commit) et régénérez les statistiques. Que constatez-vous ?

les blocs alloués restent, on voit que les tuples n'occupent plus rien 0 en nombre et en taille. Les 16 blocs persistent cependant même après un commit.

Passer par l'ordre suivant

```
truncate table test;
```

Listing 4 – truncate

et ré-évaluer les statistiques. Que constatez vous ?

truncate désalloue les blocs en plus de faire un commit implicite. Il subsiste cependant le segment et un extent

1.3 Vues système associées à l'organisation logique

Vous réalimenterez la table TEST en insérant 600 tuples. Puis vous testerez les requêtes suivantes qui exploitent l'organisation logique des tables :

```
select tablespace_name, segment_name, blocks, bytes/1024 as Koctets, extents from
  user_segments order by segment_name ;

select segment_type, extent_id, bytes/1024 enKo, blocks from user_extents where
  segment_name = 'TEST';
```

Listing 5 – organisation logique

Vous répondrez aux questions :

- Quel est le nombre de segment(s) associé(s) à la table TEST ?
un seul segment qui a le meme nom que la table
- Est ce que le segment associé à la table TEST organise logiquement l'information d'une autre table de la base ?
dans le cas le plus habituel : dependance forte entre 1 segment et 1 table, donc le segment ne contient que des blocs de la table en question
- Quel est le nom du segment associé à la table la table TEST ?
TEST
- Quel est le type du segment associé à la table TEST ?
TABLE
- Quel est le nom de l'espace de tables dans lequel se range ce segment ?
DATA_ETUD
- De combien d'extensions (extents) se compose le segment ? Est ce nombre d'extensions peut évoluer avec l'ajout de nouveaux tuples dans la table ?
select segment_name, segment_type, extent_id, blocks from dba_extents where segment_name = 'TEST' and owner = user;
2 extents de 8 blocs chacun

oui, le nombre d'extents est en fonction des besoins. Un nouvel extent avec un nombre de blocs d'un multiple de 8 peut etre ajoute