

TP Noté (3 heures) tous documents y compris numériques autorisés

1. Préalable

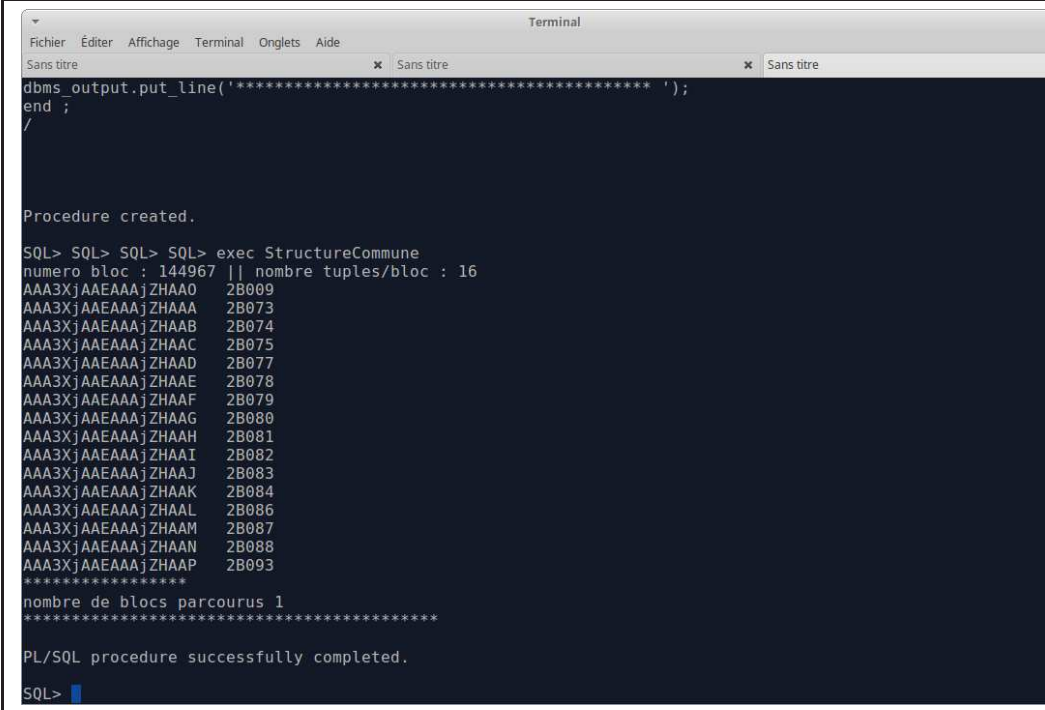
Les questions sont volontairement ouvertes. Vous pouvez mettre des commentaires concis en guise d'en-tête de vos scripts pour préciser vos choix. Les étudiants qui auront des scripts trop ressemblants, seront convoqués pour un oral complémentaire. Ce travail est en effet strictement individuel. Les scripts sont à déposer dans l'espace réservé à cet effet sur Moodle. Lisez les questions avant de commencer à composer.

Les étudiants qui n'ont pas de table Commune peuvent travailler depuis leur compte sur ma table (à préfixer par le schéma imougenot au sein de la base de données master). Un exemple d'appel à la table est donné :

```
select rowid, code_insee from imougenot.commune ;
```

2. Question 1

A l'aide de la table Commune et du paquetage dbms_rowid, vous construirez une procédure PL/SQL nommée BlocsMoinsRemplisCommune qui permettra l'affichage des tuples (rowid et code_insee) contenus dans le ou les blocs contenant le moins de tuples. Pour ce faire, construisez une première procédure nommée ListeTuples qui prend en entrée le numéro du bloc à parcourir et qui affiche le rowid et le code_insee des tuples enregistrés dans ce bloc. ListeTuples sera ensuite exploitée au sein de BlocsMoinsRemplisCommune. BlocsMoinsRemplisCommune est bâtie autour d'un curseur explicite qui exploite un partitionnement sur le numéro de blocs et une condition sur le partitionnement qui permet de retourner le ou les blocs les moins remplis.



```
dbms_output.put_line('*****');
end ;
/

Procedure created.

SQL> SQL> SQL> SQL> exec StructureCommune
numero bloc : 144967 || nombre tuples/bloc : 16
AAA3XjAAEAAAjZHA0 2B009
AAA3XjAAEAAAjZHA0 2B073
AAA3XjAAEAAAjZHA0 2B074
AAA3XjAAEAAAjZHA0 2B075
AAA3XjAAEAAAjZHA0 2B077
AAA3XjAAEAAAjZHA0 2B078
AAA3XjAAEAAAjZHA0 2B079
AAA3XjAAEAAAjZHA0 2B080
AAA3XjAAEAAAjZHA0 2B081
AAA3XjAAEAAAjZHA0 2B082
AAA3XjAAEAAAjZHA0 2B083
AAA3XjAAEAAAjZHA0 2B084
AAA3XjAAEAAAjZHA0 2B086
AAA3XjAAEAAAjZHA0 2B087
AAA3XjAAEAAAjZHA0 2B088
AAA3XjAAEAAAjZHA0 2B093
*****
nombre de blocs parcourus 1
*****
PL/SQL procedure successfully completed.

SQL>
```

FIGURE 1 – Exemple d'attendu possible

2.1 Question 2

Vous exploiterez la vue `dba_segments` qui est une vue complémentaire aux vues déjà abordées en cours et qui donne des informations supplémentaires sur les organisations logiques (segments) des objets (table, index, ...). Vous construirez une première procédure nommée `AllocationTous` qui affiche par utilisateur (owner) et par type de segment (table, index, ...) le nombre de blocs alloués à cet usager et l'espace mémoire occupé. Vous définirez une seconde procédure nommée `AllocationUsager` qui prend un usager en entrée (owner) et ne restitue les informations que pour cet usager.

OWNER	SEGMENT_TYPE	BLKS	TAILLE OCTETS
YBENABDERRAH	LOBINDEX	8	65536
YBENABDERRAH	LOBSEGMENT	8	65536
YBENABDERRAH	NESTED TABLE	8	65536
YBENABDERRAH	INDEX	208	1703936
YBENABDERRAH	TABLE	2584	21168128
YCOGNE	INDEX	80	655360
YCOGNE	TABLE	80	655360

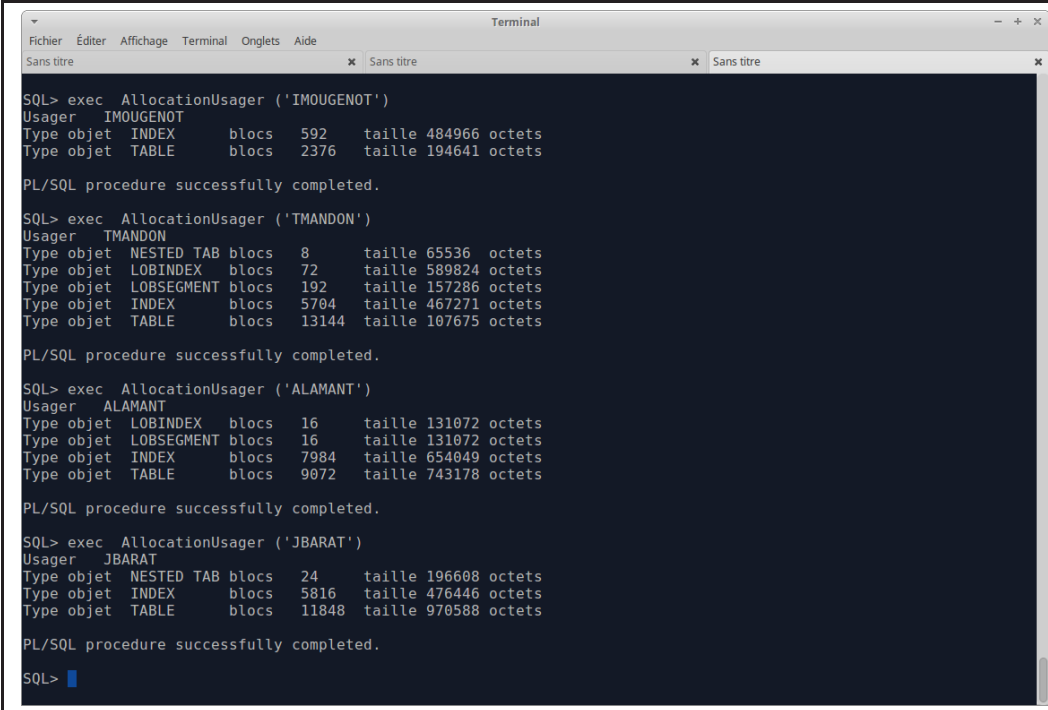
YDESMARAIS	INDEX	112	917504
YDESMARAIS	TABLE	112	917504
YDU	INDEX	8	65536
YDU	LOBINDEX	40	327680
YDU	LOBSEGMENT	40	327680
YDU	TABLE	136	1114112
YFREDERICK	INDEX	40	327680
YFREDERICK	TABLE	40	327680
YGUIGNARD	INDEX	80	655360
YGUIGNARD	TABLE	80	655360
YKERMOUCHE	INDEX	40	327680

YKERMOUCHE	TABLE	48	393216
YSHIH	INDEX	160	1310720
YSHIH	TABLE	4544	37224448
YTOGNIFODE	INDEX	96	786432
YTOGNIFODE	TABLE	96	786432
YZAHIDI	INDEX	32	262144
YZAHIDI	TABLE	72	589824
ZMOUSSA	INDEX	48	393216
ZMOUSSA	TABLE	64	524288

955 rows selected.

SQL>

FIGURE 2 – Exemple de requête illustrant la première procédure



```
SQL> exec AllocationUsager ('IMOUGENOT')
Usager IMOUGENOT
Type objet INDEX      blocs 592  taille 484966 octets
Type objet TABLE     blocs 2376 taille 194641 octets

PL/SQL procedure successfully completed.

SQL> exec AllocationUsager ('TMANDON')
Usager TMANDON
Type objet NESTED TAB blocs 8      taille 65536 octets
Type objet LOBINDEX  blocs 72     taille 589824 octets
Type objet LOBSEGMENT blocs 192   taille 157286 octets
Type objet INDEX     blocs 5704   taille 467271 octets
Type objet TABLE    blocs 13144  taille 107675 octets

PL/SQL procedure successfully completed.

SQL> exec AllocationUsager ('ALAMANT')
Usager ALAMANT
Type objet LOBINDEX  blocs 16     taille 131072 octets
Type objet LOBSEGMENT blocs 16     taille 131072 octets
Type objet INDEX     blocs 7984   taille 654049 octets
Type objet TABLE    blocs 9072   taille 743178 octets

PL/SQL procedure successfully completed.

SQL> exec AllocationUsager ('JBARAT')
Usager JBARAT
Type objet NESTED TAB blocs 24     taille 196608 octets
Type objet INDEX     blocs 5816   taille 476446 octets
Type objet TABLE    blocs 11848  taille 970588 octets

PL/SQL procedure successfully completed.

SQL>
```

FIGURE 3 – Exemple d’attendu possible pour AllocationUsager

2.2 Question 3

Un plus sera d’organiser les procédures définies au cours de votre travail, au travers d’un paquetage. Proposer également des exemples d’utilisation des procédures.