

# 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.

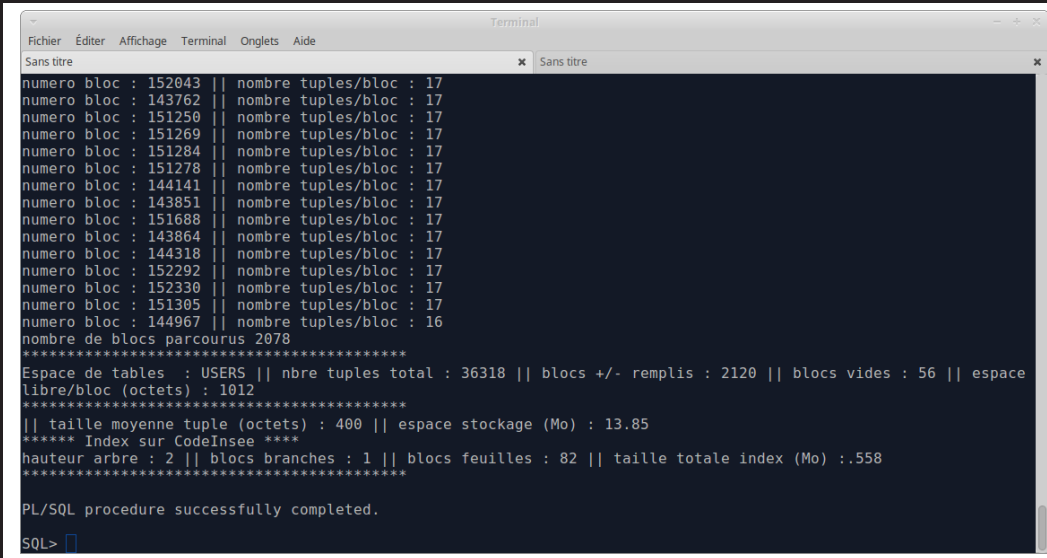
Les étudiants qui n'ont pas de table Commune peuvent travailler depuis leur compte sur ma table (à préfixer par le schéma imougenot au sein de la base de données master). l'index sur code\_insee se nomme com\_pk. Un exemple d'appel à la table est donné :

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

## 2. Question 1

---

A l'aide des vues user\_tables, index\_stats et user\_indexes et du paquetage dbms\_rowid, vous construirez une procédure PL/SQL nommée StructureCommune qui permettra l'affichage d'un grand nombre d'informations concernant l'organisation de la table Commune : espace de tables hébergeant la table, nombre de blocs alloués à la table, taille en octets de l'espace de stockage nécessaire pour l'ensemble des tuples de la table, taille moyenne du tuple, listing des blocs qui contiennent des tuples (nombre de tuples / bloc) pour donner quelques exemples. L'information précédente sera complétée par des informations portant sur l'index unique défini sur l'attribut codeInsee porteur de contrainte de clé primaire (hauteur de l'index B-arbre, nombre de blocs feuilles et branches, espace de stockage mémoire nécessaire pour l'index en particulier). Si vous exploitez index\_stats, pensez à analyser l'index au préalable (analyze index commune\_pk validate structure).



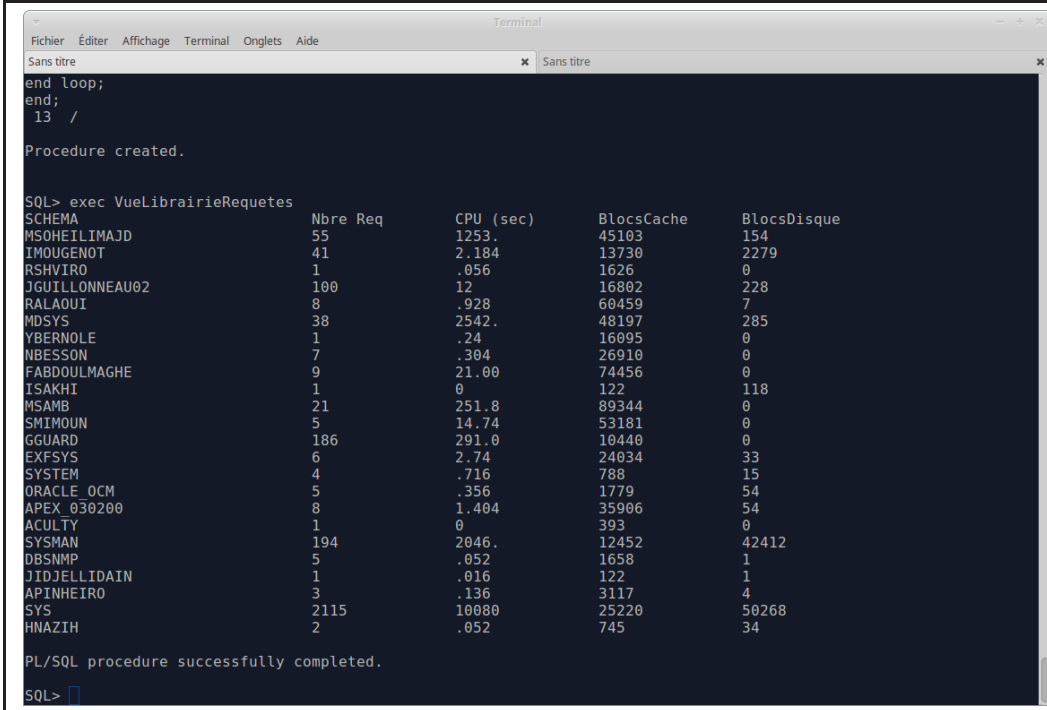
```

Fichier  Éditer  Affichage  Terminal  Onglets  Aide
Sans titre
numero bloc : 152043 || nombre tuples/bloc : 17
numero bloc : 143762 || nombre tuples/bloc : 17
numero bloc : 151250 || nombre tuples/bloc : 17
numero bloc : 151269 || nombre tuples/bloc : 17
numero bloc : 151284 || nombre tuples/bloc : 17
numero bloc : 151278 || nombre tuples/bloc : 17
numero bloc : 144141 || nombre tuples/bloc : 17
numero bloc : 143851 || nombre tuples/bloc : 17
numero bloc : 151688 || nombre tuples/bloc : 17
numero bloc : 143864 || nombre tuples/bloc : 17
numero bloc : 144318 || nombre tuples/bloc : 17
numero bloc : 152292 || nombre tuples/bloc : 17
numero bloc : 152330 || nombre tuples/bloc : 17
numero bloc : 151305 || nombre tuples/bloc : 17
numero bloc : 144967 || nombre tuples/bloc : 16
nombre de blocs parcourus 2078
*****
Espace de tables : USERS || nbre tuples total : 36318 || blocs +/- remplis : 2120 || blocs vides : 56 || espace
libre/bloc (octets) : 1012
*****
|| taille moyenne tuple (octets) : 400 || espace stockage (Mo) : 13.85
***** Index sur CodeInsee *****
hauteur arbre : 2 || blocs branches : 1 || blocs feuilles : 82 || taille totale index (Mo) :.558
*****
PL/SQL procedure successfully completed.
SQL>
```

FIGURE 1 – Exemple d’attendu possible

## 2.1 Question 2

Vous exploiterez les vues `v$sql` et `v$sqlarea` qui permettent de disposer d’informations concernant la mémoire ”library cache” pour construire une procédure nommée `VueLibrairieRequetes`. Cette procédure permettra d’afficher, pour les schémas utilisateurs (`parsing_schema_name`) qui ont passé les derniers ordres, le nombre d’ordres exécutés et la somme des temps nécessaires aux traitements de ces requêtes (`cpu_time` en microsecondes, diviser par 1 000 000 pour avoir ce temps en secondes). Vous pouvez également renvoyer le nombre total de blocs accédés au niveau du cache de données ou database buffer cache (`buffer_gets`) et de la mémoire périphérique (`disk_reads`).



```

end loop;
end;
13 /

Procedure created.

SQL> exec VueLibrairieRequetes
SCHEMA                Nbre Req      CPU (sec)      BlocsCache      BlocsDisque
-----
MSOHEILIMAJD          55           1253.         45103           154
IMOUGENOT             41           2.184         13730           2279
RSHVIRO               1             .056          1626             0
JGUILLONNEAU02       100           12           16802           228
RALAOUI              8             .928          60459             7
MDSYS                38           2542.         48197           285
YBERNOLE             1             .24           16095             0
NBESSON              7             .304          26910             0
FABDOULMAGHE         9           21.00         74456             0
ISAKHI               1             0             122             118
MSAMB               21           251.8         89344             0
SMIMOUN              5           14.74         53181             0
GGUARD             186           291.0         10440             0
EXFSYS              6           2.74          24034             33
SYSTEM              4           .716           788             15
ORACLE_OCM           5           .356           1779             54
APEX_030200          8           1.404         35906             54
ACULTY              1             0             393             0
SYSMAN             194           2046.         12452           42412
DBSNMP              5             .052           1658             1
JIDJELLIDAIN         1             .016           122             1
APINHEIRO            3             .136           3117             4
SYS                2115          10080         25220           50268
HNAZIH              2             .052           745             34

PL/SQL procedure successfully completed.

SQL>

```

FIGURE 2 – Exemple d’attendu possible (sur la base d’une requête sur v\$sqlarea)

## 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. Vous noterez que les arguments d’entrée comme de sortie des deux procédures ne sont pas imposés, et sont laissés à votre libre appréciation. Vous pouvez également faire le choix (pour chaque question) de construire plusieurs procédures qui sont appelées par une procédure principale par souci de modularité.