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

1. Préalable

Les questions sont à traiter de manière ouverte. Vous pouvez mettre des commentaires concis en guise d'en-tête de vos scripts pour préciser vos choix. Les étudiants (en monôme ou en binôme) qui auront des scripts trop ressemblants, seront convoqués pour un oral complémentaire. Les étudiants en binôme rendent un seul travail pour deux (portant bien les noms des deux étudiants). Les scripts sont à déposer dans l'espace réservé à cet effet sur Moodle. Lisez les questions avant de commencer à composer.
Un paquetage PL/SQL nommé DataCacheMetrics est à construire. Le paquetage DataCacheMetrics contient les fonctions et procédures concernant l'exploration du cache de données (blocs contenant les données accessibles en lecture et en écriture). Vous aurez à proposer également des exemples d'utilisation des fonctions et procédures. Il vous faudra aussi penser à gérer les exceptions au sein de chaque fonction/procédure.

2. Partie 1 : Paquetage DataCacheMetrics 14 points

Une pénalité, de 2 points, sera appliquée, si le paquetage n'est pas construit, et que seules, les fonctions et procédures le sont.

2.1 Question 1 : construction de fonctions

Les étudiants composant seuls, doivent construire une fonction au choix. Les étudiants en binôme doivent construire deux des trois fonctions listées.
Les signatures des fonctions et des procédures (telles qu'elles doivent être retrouvées dans la déclaration du paquetage) sont données. La procédure *mostSignificantUser* sera traitée dans la question 2.

1. function blocksNumberInDataCache return integer ;
2. function blocksNumberPerUser (username varchar) return integer ;
3. function blocksNumberPerObject (objectname varchar, username varchar) return integer
4. procedure mostSignificantUser (blocks# out integer, username out varchar) ;

Des explications sont données à la fois pour la/les vues du méta-schéma exploitée(s), et pour le comportement de chaque fonction.

1. La fonction *blocksNumberInDataCache* exploite la vue v$bh et renvoie le nombre de blocs de données, en cours d'exploitation, dans le cache de données
2. La fonction *blocksNumberPerUser* exploite les vues v$bh et dba_objects (jointure sur objd = object_id) et renvoie pour un schéma utilisateur donné, le nombre de blocs de données exploités par ce schéma en mémoire cache.

3. La fonction *blocksNumberPerObject* exploite les vues v$bh et dba_objects (jointure sur objd = object_id) et renvoie pour un objet (une table ou un index par exemple) dont le nom est passé comme argument d'entrée, le nombre de blocs de données exploités pour cet objet.

2.2 Question 2 : procédure *mostSignificantUser*

Le corps de la procédure *mostSignificantUser* vous est fourni. Vous décrirez le code correspondant à cette procédure, ainsi que ses fonctionnalités. Les exceptions ne sont pas prises en charge, vous enrichirez le code fourni par la prise en charge des exceptions qui vous semblent s'imposer.

```
procedure mostSignificantUser (blocks# out integer, username out varchar)
is
begin
select owner, count(*) into username, blocks# from v$bh bh join dba_objects on objd =
    object_id where owner not in ('SYS','XDB','WMSYS')
group by owner having count(*) >= all (select count(*) from v$bh bh join dba_objects on
    objd = object_id where owner not in ('SYS','XDB','WMSYS')
group by owner) ;
end;
```

Listing 1 – Procédure mostSignificantUser

2.3 Question 3 : Exemples de mise en œuvre

Cette question est volontairement plus ouverte. Vous avez à proposer des exemples de mise en œuvre, soit au sein de requêtes pour les fonctions, soit au sein de programmes principaux ou au travers d'ordre "exec" pour la procédure.

— Les étudiants composant seuls, doivent proposer des exemples pour la fonction choisie et la procédure mostSignificantUser

— Les étudiants composant en binôme, doivent proposer des exemples pour les deux fonctions choisies et la procédure mostSignificantUser

3. Partie 2 : Zone mémoire partagée : Library Cache 6 points

Les étudiants seuls n'ont à traiter qu'une question au choix sur les deux.

3.1 Question 1 : requête SQL portant sur v$sqlarea

Vous donnerez dans le détail la signification de la requête ci-dessous, et illustrerez son intérêt avec quelques tuples résultats.

```
set linesize 200
col parsing_schema_name for a20
select substr(sql_text,1,80), parsing_schema_name, cpu_time/1000000 cpu_sec, fetches,
    executions, disk_reads+buffer_gets from v$sqlarea where parsing_schema_name like
    'E%' order by cpu_time/1000000;
```

Listing 2 – Requête tirant partie de v$sqlarea

3.2 Question 1 : requête SQL à partir d'un énoncé

Vous écrirez en exploitant v$sqlarea et dba_users la requête qui permet de connaître les utilisateurs qui n'ont pas de requête récemment traitée par le système (pas de requête retrouvée dans la zone library cache)