

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 principales exceptions pouvant survenir, 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, parmi les deux fonctions nommées *allocatedBytesInDataCache* et *ratioCachePerUser*. Les étudiants en binôme doivent obligatoirement construire ces deux fonctions.

Les signatures des fonctions sont données. Le code de la fonction *blocksNumberInDataCache* est entièrement donné en annexe, et doit servir de base pour construire la fonction *allocatedBytesInDataCache*. La fonction *blocksNumberPerTablespace* est à traiter par tous, et servira de base à la construction d'une procédure en question 2.

1. `function blocksNumberInDataCache return integer ;`
2. `function allocatedBytesInDataCache return integer ;`
3. `function ratioCachePerUser (username varchar) return float ;`
4. `function blocksNumberPerTablespace (tsname varchar) return integer ;`

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 *allocatedBytesInDataCache* exploite la vue `v$bh` (par le biais de la fonction *blocksNumberInDataCache*) ainsi que la vue `v$parameter` pour avoir la taille du blocs de données en octets, et renvoie le nombre d'octets, en cours d'exploitation, dans le cache de données

2. La fonction *ratioCachePerUser* exploite les vues *v\$bh* et *dba_objects* (jointure sur *objd = object_id*) et renvoie pour un schéma utilisateur dont le nom est passé comme argument d'entrée, le pourcentage d'usage du cache qui lui revient (rapport entre le nombre total de blocs exploités obtenu avec *blocksNumberInDataCache*, et le nombre de blocs exploités, concernant les objets de cet usager).
3. La fonction *blocksNumberPerTablespace* exploite les vues *v\$bh* et *v\$tablespace* (jointure sur l'attribut commun *ts#* ou encore numéro de tablespace) et renvoie pour un tablespace donné, le nombre de blocs de données exploités par ce tablespace en mémoire cache.

2.2 Question 2 : procédure *afficheBlocsParTableSpace*

Vous avez à définir la procédure *afficheBlocsParTableSpace* qui appelle la fonction *blocksNumberPerTablespace* et exploite *v\$tablespace* pour afficher pour tous les tablespaces définis au sein du serveur de données et le nombre de blocs de données en cours d'exploitation par ces tablespaces. Vous donnerez un exemple d'utilisation, ainsi que le résultat obtenu.

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, soit au sein de programmes principaux, soit encore au sein de nouvelles fonctions et procédures.

- Les étudiants composant seuls, doivent proposer des exemples pour la fonction choisie entre *allocatedBytesInDataCache* et *ratioCachePerUser*.
- Les étudiants composant en binôme, doivent proposer des exemples pour les deux fonctions *allocatedBytesInDataCache* et *ratioCachePerUser*.

3. Partie 2 : Library Cache et Data Buffer Cache 6 points

Les étudiants seuls n'ont à traiter qu'une question au choix sur les deux. Les étudiants en binôme doivent traiter 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. Vous pouvez également enrichir la requête pour la rendre encore plus informative.

```
set linesize 200
col osuser for a30

select s.sid, s.osuser, substr(a.sql_text,1,60), plan_hash_value from v$session s join
    v$sqlarea a
on a.hash_value = s.prev_hash_value ;
```

Listing 1 – Requête mobilisant *v\$sqlarea*

3.2 Question 2 : requête SQL portant sur *v\$bh*, *dba_users* et *dba_objects*

Vous donnerez dans le détail la signification de la requête ci-dessous, et illustrerez son intérêt avec quelques tuples résultats. Vous pouvez également enrichir la requête pour la rendre encore plus

informative.

```
col username for a16

select username, to_char(last_login,'DD-MM-YYYY HH-MI-SS'), to_char(created,'DD-MM-YYYY
      HH-MI-SS') from
dba_users where username in
((select username from dba_users
minus
select owner from v$bh join dba_objects on objd = object_id)
minus
(select username from dba_users
minus
select owner from dba_objects)) ;
```

Listing 2 – Requête mobilisant v\$bh

4. Annexe

```
function blocksNumberInDataCache return integer
is
blocks# integer ;
begin
SELECT count(*) INTO blocks# FROM v$bh;
return blocks#;
exception when others then return -1;
end;
```

Listing 3 – Fonction blocksNumberInDataCache