

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

## 1. Préalable

---

Le TP se compose de 4 exercices qui peuvent être traités, soit sous le forme de requêtes SQL, soit à l'aide de procédures PL/SQL élémentaires (avec ou sans arguments en entrée et juste de l'affichage). Les exercices sont indépendants. Un exercice est noté sur 4, si il est traité avec du requêtage SQL, et sur 5, si il est traité avec du PL/SQL.

## 2. Exercice 1

---

1. Question 1.1 à partir de `v$bh` et `dba_objects`

Pour les usagers qui sont propriétaires d'une table nommée EMP, vous donnerez le nom de ces usagers ainsi que la date de création de leur table EMP, quand cette table fait l'objet d'au moins un bloc de données en mémoire cache (data buffer cache)

2. Question 1.2 à partir de `v$bh` et `dba_objects`

Donner le nom de l'utilisateur (éventuellement plusieurs noms d'utilisateurs) qui est le plus consommateur de blocs en mémoire cache pour ce qui concerne sa table EMP.

## 3. Exercice 2

---

1. Question 2.1 à partir de `v$log_history`

Les fichiers journaux se voient exploités dans une rotation circulaire, pour l'écriture des transactions. La vue `v$log_history` garde une trace de toutes les rotations effectuées entre deux fichiers.

Donner le jour de la semaine (éventuellement plusieurs jours) pour lequel les rotations sont les plus nombreuses (donner également le nombre de rotations). Plus les rotations sont nombreuses, et plus les usagers mobilisent la base de données.

## 4. Exercice 3

---

1. Question 3.1 à partir de `dba_segments`

Donner pour chaque usager son nombre de segments par type de segments, le cumul des extents et des blocs pour chaque type de segment le concernant. Un exemple de résultat attendu est donné.

2. Question 3.2 à partir de `user_segments` et de `user_tables`

Donner la combinaison d'information qui vous semble la plus pertinente à partir de la jointure

OWNER	SEGMENT_TYPE	NBRSEGMENTS	NBREXTENTS	SOMMEBLOCKS	SOMMEKOCETTS
E20180008660	INDEX	24	44	352	2816
E20180008660	TABLE	24	47	496	3968
E20190012905	TABLE	24	43	464	3712
E20200007165	INDEX	24	65	2440	19520
E20200010900	TABLE	24	24	192	1536
E20210010123	TABLE	24	115	8000	64000
E20190005841	INDEX	25	66	2448	19584
E20200008323	TABLE	25	25	200	1600
E20170005405	INDEX	26	46	368	2944
E20180011220	TABLE	26	115	8000	64000
E20210012414	TABLE	27	27	216	1728
<hr/>					
OWNER	SEGMENT_TYPE	NBRSEGMENTS	NBREXTENTS	SOMMEBLOCKS	SOMMEKOCETTS
E20200007733	TABLE	28	121	8048	64384
E20200012678	INDEX	28	39	312	2496
E20200012678	TABLE	28	48	504	4032
E20210013456	INDEX	28	28	224	1792
P00000009432	TABLE	28	236	13408	107264
E20210010867	INDEX	29	29	232	1856
E20170004282	TABLE	30	97	1616	12928
E20200007165	TABLE	30	121	8048	64384
E20190005841	TABLE	31	123	8064	64512
E20200002764	TABLE	31	34	272	2176
E20170013076	TABLE	32	32	256	2048
<hr/>					
OWNER	SEGMENT_TYPE	NBRSEGMENTS	NBREXTENTS	SOMMEBLOCKS	SOMMEKOCETTS
E20200010501	INDEX	33	55	440	3520

FIGURE 1 – Exemple illustratif

entre `user_segments` et `user_tables` (pour vos tables donc)

## 5. Exercice 4

### 1. Question 4.1

Différentes vues systèmes ont été exploitées lors des TP transaction et architecture Oracle. Vous mettrez à profit les vues qui vous semblent d'intérêt pour consulter les transactions qui ont posé des verrous, et qui s'avèrent bloquantes pour d'autres transactions. Parmi les informations à retourner :

- identifiants de session des couples de transactions bloquante et bloquée
- nom des schémas utilisateurs à l'origine des sessions des couples de transactions bloquante et bloquée
- identifiants système des utilisateurs à l'origine des sessions des couples de transactions bloquante et bloquée
- identifiant et nom de l'objet sur lequel est posé le verrou
- type(s) des verrous bloquants
- type(s) des verrous en attente ...

Donner un exemple de résultat de vos requêtes/procédures (une transaction bloquante a été créée pour la durée du tp afin de vous aider à tester votre code).