

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 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.

Un plus sera d'organiser les procédures à construire, au travers d'un packaging nommé TpCC. Proposer également des exemples d'utilisation des procédures. Vous pouvez également faire le choix (pour chaque question) de construire plusieurs procédures qui sont appelées par une procédure principale par souci de modularité. Il vous faudra aussi penser à gérer les exceptions au sein de chaque procédure.

2. Partie 1

2.1 Question 1 : coût de stockage d'une table

A l'aide de la vue `dba_segment`, vous construirez une procédure PL/SQL nommée `InformationsStorageTable` qui prendra deux arguments en entrée (nom d'un usager et nom d'une table) et qui affichera le plus d'informations possibles concernant le coût de stockage de cette table. Vous pouvez également faire appel aux informations rendues disponibles à partir de `user_tables`. Des exemples d'appels et de résultats attendus sont donnés ci-dessous.

```

variable usager varchar2(20);
variable maTable varchar2(20);
exec :usager := 'ISA'
exec :maTable := 'COMMUNE'

exec tpCC.InformationStockageTable(:usager, :maTable)

*****
nom Segment      type Segment      Espace table
COMMUNE          TABLE          USERS

Nbre extensions  Nbre blocs alloues  Taille en Mo
17               256               2
*****

exec InformationsStockageTable('TOTO', :maTable)

*****
usager ou table inexistants
*****

```

FIGURE 1 – Exemple illustratif

2.2 Question 2 : coût de stockage d'une table et de ses index

Il s'agit de reprendre la procédure précédente, et d'y ajouter les informations concernant le coût de stockage des index pouvant être définis à partir de la table dont le nom est passé en argument d'entrée (toujours en vous servant de `dba_segments` ou bien de `dba_indexes`). Un exemple du résultat à l'exécution de la procédure, est également donné.

```

exec tpCC.InformationStockageTableIndex(:usager, :maTable)

*****
nom Segment      type Segment      Espace table
COMMUNE          TABLE          USERS

Nbre extensions  Nbre blocs alloues  Taille en Mo
17               256               2
*****
nom Segment      type Segment      Espace table
COM_IDX          INDEX          USERS

Nbre extensions  Nbre blocs alloues  Taille en Mo
17               256               2
*****
nom Segment      type Segment      Espace table
COMMUNE_PK       INDEX          USERS

Nbre extensions  Nbre blocs alloues  Taille en Mo
12               96               .75
*****

```

FIGURE 2 – Exemple illustratif

2.3 Partie 2

2.4 Question 1 : Les blocs d'un usager dans le cache de données

Une nouvelle procédure nommée `blocsCache` est à construire à partir des vues `v$bh` et `dba_objects`. Elle renseigne sur les blocs de données de tables ou de données d'index trouvés dans le cache de données et appartenant à un usager donné. Les arguments à fournir en entrée sont le nom de l'utilisateur et le type d'objet (TABLE ou INDEX). Les informations à afficher vont concerner le numéro du bloc, sa classe et son statut ainsi que le nom de l'index ou de la table de provenance. Un exemple d'appel est fourni.

```
exec tpCC.BlocsCache('IMOUGENOT', 'INDEX')
```

Exemple retour

nom objet	type object	numero bloc	classe	statut
COMMUNE_PK	INDEX	3016	1	xcur

nom objet	type object	numero bloc	classe	statut
COMMUNE_PK	INDEX	3105	1	xcur

nom objet	type object	numero bloc	classe	statut
COMMUNE_PK	INDEX	3427	1	xcur

FIGURE 3 – Exemple illustratif