

# TP2 index (noté) 3 heures

## 1. Schéma exploité

---

Vous créerez une table COMMUNE à partir de la table COMMUNE du schéma utilisateur P00000009432. Vous définirez une contrainte de clé primaire nommée COMMUNE\_PK s'appliquant à l'attribut CODEINSEE de votre table COMMUNE.

Vous construirez également la table DEPARTEMENT à partir du même schéma utilisateur et y poserez une contrainte de clé primaire. Vous définirez alors une contrainte de clé étrangère sur l'attribut NUMDEP de COMMUNE qui référence l'attribut NUMDEP de DEPARTEMENT.

### 1.1 Question 1 (2 points)

Donner les ordres associés aux créations des tables et contraintes ainsi définies.

### 1.2 Question 2 (1 point)

Vous définirez un index non unique (de nom numdep\_idx) sur l'attribut NUMDEP de COMMUNE qui pourra être utile lors de jointures entre COMMUNE et DEPARTEMENT.

Vous donnerez l'ordre de création de l'index numdep\_idx

## 2. Rappel sur la consultation des vues du méta-schéma relatives aux index

---

Les vues index\_stats et user\_indexes aident à la compréhension des structures d'index manipulées par un serveur de base de données. L'ordre SQL donné ci-dessous exploite user\_indexes et permet par exemple de consulter l'ensemble des index définis sur le schéma utilisateur, le nom de l'index, le nom de la table impactée par l'index ainsi que la hauteur de l'arbre (sans le niveau des feuilles)<sup>1</sup>.

```
SELECT index_name, blevel, table_name FROM USER_INDEXES;
```

Listing 1 – Vue index

La vue index\_stats donne des informations complémentaires (parfois chevauchantes) à la vue user\_indexes. Il est ainsi possible de disposer d'informations sur la place mémoire occupée par l'index, le nombre de blocs occupés par les nœuds branches (BR\_BLKs) et les nœuds feuilles de l'arbre (LF\_BLKs). Il est cependant nécessaire de collecter les statistiques sur les index avant de consulter cette vue. La consultation ci-dessous retourne respectivement le nom de l'index, l'espace occupé en octets, le nombre de répétitions pour la valeur de clé la plus répétée, le nombre de tuples (clé, rowid, pointeur tuple

---

1. La valeur 0 indique que l'index n'est constitué que d'un niveau racine

gauche, pointeur tuple droit) au niveau feuille, le nombre de tuples (clé, pointeur) au niveau nœud des branches et la hauteur de l'arbre (avec le niveau feuille et donc égal à `blevel+1`).

```
-- mettre a jour les statistiques pour un index cible
ANALYZE INDEX <index_name> VALIDATE STRUCTURE;

SELECT name, btree_space, most_repeated_key, lf_rows, br_rows, height FROM
  INDEX_STATS;
```

Listing 2 – Vue statistiques sur index

## 2.1 Question 3 (6 points)

Vous exploiterez les vues du méta-schéma et notamment les vues `index_stats` et `user_indexes` pour répondre aux questions suivantes.

Penser également à mettre à jour au préalable, les statistiques de la table d'une des deux manières suivantes :

```
--
ANALYZE TABLE COMMUNE COMPUTE STATISTICS;
-- possible aussi pour le schema entier
EXEC DBMS_UTILITY.ANALYZE_SCHEMA(user,'COMPUTE')
```

Listing 3 – collecter statistiques

Vous donnerez les requêtes SQL suivantes ainsi que leurs résultats

1. Quelle est la hauteur de la taille de l'index `COMMUNE_PK` de la table `COMMUNE` ?
2. Quels sont les nombres de blocs de branches et de feuilles, qui ont été réservés pour l'index `COMMUNE_PK` de la table `COMMUNE` ?
3. Pour cet index, quelle est la taille de chaque tuple (clé, pointeurs, rowid) présent au niveau des blocs des feuilles ?
4. Par comparaison, quelle est la taille moyenne de chaque tuple de la table `COMMUNE` et combien de tuples peuvent être stockés dans un bloc (calcul du facteur de blocage de l'espace qui tient compte de l'espace toujours laissé libre, et donc de la valeur de `PCT_FREE` de la vue `USER_TABLES`) ?
5. Quelle est la hauteur de la taille de l'index `NUMDEP_IDX` de la table `COMMUNE` ?
6. Expliquer ce que renvoient les valeurs des attributs `DISTINCT_KEYS` et `MOST_REPEATED_KEY` de la vue `INDEX_STATS` pour l'index `NUMDEP_IDX`

## 3. Manipulation du paquetage DBMS\_ROWID

L'adresse de chaque enregistrement sur le disque (identifiant de ligne ou rowid) renferme différentes informations à l'exemple du numéro de l'enregistrement, du numéro de l'objet associé (table ou index), du bloc de données qui contient cet enregistrement, ou de l'adresse relative du fichier qui contient les blocs de données. Le paquetage `DBMS_ROWID` permet d'exploiter l'ensemble de cette information. Des exemples vous sont donnés :

```
DECLARE
  object_no integer;
```

```

row_no integer;
row_id ROWID;

BEGIN
  SELECT ROWID INTO row_id FROM commune
    WHERE codeInsee = '34172';
  object_no := DBMS_ROWID.ROWID_OBJECT(row_id);
  row_no := DBMS_ROWID.ROWID_ROW_NUMBER(row_id);
  DBMS_OUTPUT.PUT_LINE('The obj. # is '||object_no||' '||row_no);
END;
/

SELECT DBMS_ROWID.ROWID_BLOCK_NUMBER(rowid), DBMS_ROWID.ROWID_OBJECT(rowid),
       nomcommaj
  FROM Commune where codeInsee = '34172';

-- nom de la structure de table ou d'index
select owner, object_name from dba_objects where data_object_id = ...

```

Listing 4 – collecte des statistiques

### 3.1 Question 4 (5 points)

- Que renvoie la requête suivante ?  
`select rowid, rownum, codeinsee from commune;`
- Vous construirez une procédure PL/SQL (nommée MEMEBLOCQUE) qui exploite le paquetage DBMS\_ROWID pour afficher tous les enregistrements (codeInsee, nom\_com) contenus dans le même bloc de données qu'un enregistrement donné de la table COMMUNE qui sera passé en paramètre d'entrée (par exemple l'enregistrement dont le code INSEE est 34172).
- Vous construirez une seconde procédure PL/SQL (nommée NBRETUPLESPARBLOC) qui renvoie le numéro des blocs (id) et le nombre de tuples contenus dans chacun des blocs, pour tous les blocs de la table COMMUNE. Le nombre de tuples listés par bloc est t'il en accord avec le facteur de blocage calculé précédemment, et les informations collectées dans la vue user\_tables ?

```
select blocks, avg_row_len from user_tables where table_name ='COMMUNE';
```

Listing 5 – consultation exemple

### 3.2 Question 5 (6 points)

- Vous construirez une nouvelle procédure PL/SQL (nommée BLOCSDUDEPARTEMENT) qui renvoie le numéro des différents blocs (id) contenant les tuples de COMMUNE d'un département donné.
- Vous construirez une dernière procédure PL/SQL (nommée DANSCACHE) qui renvoie le numéro des différents blocs (id) de COMMUNE, ainsi que leurs enregistrements (juste codeInsee, nom\_com) quand des copies de ces blocs sont présents dans le cache de données (il vous faudra faire appel à la vue v\$bh).

Les questions 4 et 5 peuvent faire l'objet de la construction d'un paquetage (package commun). Pensez également à la gestion des exceptions les plus courantes.