

# Oracle et les index

## 1. Préalable

---

Le TP est consacré à une meilleure compréhension de l'utilisation d'index dans le périmètre du SGBD Oracle. Nous nous limitons pour l'instant aux index de type B+Tree.

## 2. 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 pour 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;
```

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 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 donné
ANALYZE INDEX <index_name> STRUCTURE;
-- pour la table COMMUNE
```

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

La documentation Oracle pour les vues `index_stats` et `user_indexes` est disponible sur [http://docs.oracle.com/cd/B28359\\_01/server.111/b28320/statviews\\_5119.htm](http://docs.oracle.com/cd/B28359_01/server.111/b28320/statviews_5119.htm) et [http://docs.oracle.com/cd/B28359\\_01/server.111/b28320/statviews\\_1096.htm](http://docs.oracle.com/cd/B28359_01/server.111/b28320/statviews_1096.htm).

### 2.1 Exercices

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

1. Que renvoie la requête suivante ?

---

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

```
select rowid, rownum, code_insee from commune;
```

2. quels sont les index déjà présents sur votre schéma utilisateur ? Expliquez leur présence
3. Indiquez si ces index sont uniques ? denses ?
4. Quelle est la hauteur de la taille de l'index de la table COMMUNE ?
5. Quels sont les nombres de blocs de branches et de feuilles, qui ont été réservés pour l'index de la table COMMUNE ?
6. Quelle est la taille de chaque tuple (clé, pointeurs, rowid) présent au niveau des blocs des feuilles (on considère que les blocs sont pleins) ?
7. Par comparaison, quelle est la taille moyenne de chaque tuple de la table COMMUNE et combien de tuples peuvent être stockés dans un bloc (tenir compte de l'espace libre et donc de la valeur de PCT\_FREE de la vue USER\_TABLES) ?
8. Calculer le nombre de blocs théorique nécessaire au stockage de l'ensemble des tuples de la table et comparez ce nombre avec la valeur de l'attribut BLOCKS de USER\_TABLES. Le nombre calculé vous semble-t'il réaliste ?
9. Dans quel espace de tables est organisé logiquement l'index COMMUNE\_PK ?
10. Quel est l'espace de stockage (en Moctets) nécessaire pour la table COMMUNE ? Quel espace de stockage supplémentaire faut'il rajouter pour stocker l'index COMMUNE\_PK ?

### 3. Construction d'index et évaluation du temps

---

#### 3.1 Construction et manipulation d'index

Vous créerez un index non unique (DEP\_IDX) sur l'attribut DEP de COMMUNE et un index unique (COM\_IDX) sur la fonction très simple LOWER(NOM\_COM). Consulter user\_indexes, index\_stats et user\_extents pour avoir quelques informations sur les index créés. Vous exploiterez les syntaxes suivantes pour soit désactiver, soit supprimer les index (DEP\_IDX, COM\_IDX, COMMUNE\_PK) :

```
alter index <nom_index> disable;
alter index <nom_index> enable;
drop index <nom_index>;
```

Que remarquez vous ?

#### 3.2 Evaluation du temps

Il existe différents manières d'évaluer le temps d'exécution d'une requête dans Oracle. La plus simple est d'exploiter la variable d'environnement TIMING (set timing on pour l'activation et set timing off pour la désactivation). Un exemple d'utilisation vous est donné que vous testerez sur une requête élémentaire sur COMMUNE. Le résultat est précis au centième de secondes, ainsi la requête de l'exemple a demandé 62 centièmes pour le processus (parse-fetch-execute) donnant accès aux résultats.

```
set timing on
select dep from commune_15;
... les résultats
Elapsed: 00:00:00.62
```

Vous pouvez manipuler aussi l'endormissement du processus, pour avoir une meilleure maîtrise du temps écoulé.

```
begin
    dbms_lock.sleep(10);
end;
/
```

Le paquetage dbms\_utility donne accès aux méthodes get\_time et set\_time pour manipuler le temps. Testez également le programme principal suivant. Pourquoi à votre avis le temps écoulé est bien plus long avec ce code ?

```
DECLARE
    l_start NUMBER DEFAULT dbms_utility.get_time;
BEGIN
    FOR i IN
        (SELECT dep
         FROM   commune_15)
    LOOP
        dbms_output.put_line ('name      : ' || i.dep);
    END LOOP;
    dbms_output.put_line(round( (dbms_utility.get_time-l_start)/100, 2 ) || ' seconds...');
END;
/
```

### 3.3 Premières évaluations de performance

Nous reviendrons dans un prochain TP sur la manière de tracer une requête et d'en évaluer la qualité d'exécution avec des mécanismes tels que EXPLAIN PLAN. Pour l'instant, nous allons nous contenter de porter une réflexion sur l'utilisation ou non d'une structure d'index par le système selon la nature de la requête. Une liste de requêtes vous est donnée et vous essaieriez de déterminer si l'index COMMUNE\_PK est exploité ou non (justifiez vos réponses). Exploitez le temps écoulé pour l'exécution pour tenter d'étayer vos réponses. Assurez vous d'avoir supprimé les index sur NOM\_COM et DEP au préalable (à défaut supprimez les).

1. select code\_insee from commune ?
2. select nom\_Com from commune ;
3. select nom\_Com, code\_insee from commune ;
4. select nom\_Com from commune where code\_insee='34192' ;
5. select nom\_Com from commune where code\_insee like '34%';
6. select nom\_Com from commune where code\_insee >= 34 ;
7. select nom\_Com from commune where code\_insee in ('09330','09331','09332','09334') ;

Une directive (ou hint) dans une requête SQL permet entre autres de faire le choix sur l'exploitation d'un index ou non dans une requête. Des exemples de manipulation de cette directive vous sont donnés. Vous les exploiterez pour là encore tenter de vérifier vos réponses.

```
select  /** NO_INDEX(Commune) */ *
        from    Commune;
```

```
select  /** NO_INDEX(Commune) */ code_insee, nom_com from commune
where code_insee = '09342' ;
```

### 3.4 Autres tests

L'index COMMUNE\_15\_PK a été construit automatiquement par le système lors de la définition de la contrainte de clé primaire COMMUNE\_15\_PK. Si la contrainte de clé primaire est désactivée, est ce que l'index peut continuer à être utilisé ? Pourquoi à votre avis ? Que se passe t'il à la réactivation de la contrainte ? De même, quel est le devenir de l'index si la contrainte de clé primaire est supprimée ?

## 4. Estimation de la taille d'un index

---

La construction d'un index entraîne un surcoût de stockage qui peut s'avérer non négligeable. Vous testerez les fonctionnalités d'estimation de la taille en blocs (alloués et utilisés) d'un index dont la création est envisagée (testez sur DEP\_IDX et COM\_IDX). Vous vérifierez ensuite la pertinence de ces estimations en recréant DEP\_IDX et COM\_IDX et en interrogeant la vue USER\_SEGMENTS. L'utilisation de la procédure CREATE\_INDEX\_COST se fait par le biais de variables d'environnement (bind variables) et donc l'utilisation du SQL Dynamique. Nous verrons dans le prochain TP que ces variables présentent un intérêt tout particulier dans tout ce qui relève de l'exécution de requêtes.

```
-- declaration des variables de session
variable used_bytes number
variable alloc_bytes number
exec dbms_space.create_index_cost( 'create index testCommune_idx on commune(nom_com)',
    :used_bytes, :alloc_bytes );
-- affichage contenu variables
print :used_bytes
print :alloc_bytes

-- valeur réelle de la taille

create index testCommune_idx on commune(nom_com);
select trim(to_char(bytes, '999,999,999')) actual_bytes
    from user_segments where segment_name = 'TESTCOMMUNE_IDX';
```