

# TP Optimisation de requêtes dans le contexte d'Oracle

## 1. Préalable

---

Vous travaillerez sur les tables Commune, Departement, Region et éventuellement Population.

## 2. Exercice 1

---

Vous reprendrez les questions posées lors du dernier TP concernant le recours à l'index<sup>1</sup> dans une série de requêtes SQL sur la table Commune. Vous afficherez les plans proposés par l'optimiseur pour chacune de ces questions et vous analyserez les résultats obtenus. Vous pouvez aussi rapidement dessiner les plans physiques destinés à l'exécution des requêtes. Dans un second temps, soit vous désactivez la contrainte de clé primaire sur code\_insee qui conduit à la suppression de l'index, soit vous exploiterez la directive NO\_INDEX et vous réévaluerez les plans précédents.

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 like '%392' ;
7. select nom\_Com from commune where code\_insee >= 34 ;
8. 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' ;
```

Pour répondre au mieux à ces questions, construisez un tableau sur papier, contenant les informations :  
**numéro requête    recours index (o/n)    raison du choix de l'optimiseur    opérateurs en jeu**

---

1. vérifiez au préalable que l'index sur code\_insee est présent dans votre schéma

### 3. Exercice 2

---

Vous traiterez au travers d'un plan d'exécution les requêtes suivantes :

1. donnez toutes les informations concernant les communes dont le nom commence par 'MO'
2. donnez toutes les informations concernant les départements dont le numéro de département est compris entre 25 et 45.
3. donnez pour les communes, leur nom, et le nom de leur département d'appartenance et de leur région d'appartenance
4. donnez le nombre de communes par numéro et nom de département

Commentez les plans retournés. Vous pouvez également choisir de privilégier le temps de réponse.

Dans un deuxième temps, modifiez les tables pour poser des index secondaires sur les attributs impliqués dans les jointures (colonnes porteuses de contrainte de clé étrangère). Vous poserez également un index secondaire sur la colonne nom de la table Commune. Retraitez au travers d'un plan d'exécution les requêtes précédentes et précisez les différences de traitement par l'optimiseur.

Exploitez la table du méta-schéma *user\_indexes* pour visualiser l'ensemble des index posés sur les tables du schéma utilisateur.

### 4. Exercice 3

---

1. Exprimez la requête *donnez le nom, la latitude et la longitude des communes qui se situent dans les départements de l'Hérault et du Gard* sous différentes formes (expression d'une semi-jointure) : sous forme de jointure puis sous forme de requête imbriquée (test de vacuité (exists) et test d'appartenance (in)). Quelle est l'écriture qui vous semble la moins coûteuse (utilisez `set timing` et `explain`) ? Quels sont les opérateurs physiques exploités respectivement pour exprimer la jointure ? Exploiter les directives (hint) pour forcer le choix d'un opérateur (par exemple `use_nl`). Construisez sur papier un arbre algébrique puis les plans physiques correspondant à chaque plan d'exécution choisi.
2. Ecrivez la requête de différentes manières : *donnez le nom des communes, le nom de leur département et de leur région respectifs lorsque ces communes sont situées dans les régions Midi-Pyrénées, Languedoc-Roussillon et Provence-Alpes-Côte d'Azur*. Commentez les plans d'exécution obtenus.

### 5. Exercice 4

---

Vous construirez une procédure PL/SQL qui permettra de tirer parti de la vue *plan\_table* qui est la vue exploitée par le paquetage *dbms\_xplan* pour afficher le plan d'exécution choisi par l'optimiseur. L'idée de la procédure est de permettre d'afficher le plan d'exécution plutôt sous une forme d'arbre que sous une forme de tableau. Vous pouvez également proposer votre propre perception et renvoyer le résultat sous la forme d'une structure graphe visualisable avec des éditeurs appropriés. Des exemples de requêtes et une sortie susceptibles de vous aider sont donnés :

```
explain plan set statement_id = 'pe4'
for select nom_com, latitude, longitude, nom_dep from commune c, departement d
where d.dep in ('30','34') and c.dep = d.dep;

select id, parent_id, operation, object_name, object_type, cost, cardinality, bytes, time
```

```
from plan_table where statement_id = 'pe4';
```

```
0 (parent : ) == [ bytes 640 time 1 cost 6 ] ==  SELECT STATEMENT
1 (parent : 0) == [ bytes 200 time 1 cost 2 ] ==  TABLE ACCESS  COMMUNE
2 (parent : 1) ==== [ bytes 640 time 1 cost 6 ] ====  NESTED LOOPS
3 (parent : 2) ===== [ bytes  time  cost  ] =====  INLIST ITERATOR
4 (parent : 3) ===== [ bytes 30 time 1 cost 2 ] =====  TABLE ACCESS  DEPARTEMENT
5 (parent : 4) ===== [ bytes  time 1 cost 1 ] =====  INDEX DEPARTEMENT_PK
6 (parent : 2) ===== [ bytes  time 1 cost 1 ] =====  INDEX DEP_IDX
```