

## Travail pratique 4

### Optimisation de requêtes et analyse de la performance

---

## 1 Introduction

L'objectif de ce travail est de vous familiariser avec l'évaluation de performance des requêtes et l'optimisation du schéma interne d'une BD. Le travail à réaliser dans le cadre de ce TP est indépendant de celui accompli dans les précédents travaux. L'évaluation des requêtes et l'optimisation seront faites sur des données synthétiques, générées à l'aide d'un script fourni avec cet énoncé.

## 2 Travail à réaliser

### Tâche 1 : Évaluation et optimisation de requêtes

La première tâche de ce travail consiste à optimiser des opérations simples de sélection et de jointure portant sur des données synthétiques. Dans un premier temps, vous devrez générer ces données en exécutant le script **TP4-ScriptCreation.sql** fourni avec cet énoncé. Ensuite, pour chacune des requêtes décrites ci-dessous, vous devrez :

1. Modifier le schéma pour inclure les optimisations appropriées ;
2. Modifier la requête pour inclure les indices d'optimisation (HINTS) appropriés. Vous pouvez consulter le document suivant pour cette étape ;
3. Exécuter la requête en mode AUTOTRACE, soit avec la commande SET AUTOTRACE ON sous SQLPlus (mode console) ou avec le mode trace de SQLDeveloper et attendre que l'exécution de la requête se termine.
4. Copier dans votre rapport une capture d'écran du texte (affichage console) ou de la fenêtre (SQLDeveloper) correspondant au plan d'exécution utilisé par Oracle et ses statistiques de traçage des performances. Assurez-vous que tous les nœuds de l'arbre soient visibles et qu'on puisse voir les colonnes suivantes :

- OPERATION
- OBJECT\_NAME
- COST
- LAST\_ELAPSED\_TIME

Mettez également dans le rapport le temps d'exécution de la requête. À noter que cette statistique est moins fiable puisqu'elle dépend du taux d'utilisation du serveur BD.

## Opérations de sélection

### Sélectivité petite

Requête :

```
SELECT COUNT(*) FROM T1 WHERE T1.col2=1
```

Vous devrez évaluer cette requête en utilisant les configurations suivantes :

1. Sans optimisation avec l'indice FULL ;
2. Avec un index secondaire sur T1.col2 ainsi que l'indice INDEX ou INDEX\_FFS (au choix) ;
3. Avec HASH CLUSTER sur T1 (clé col2) et l'indice HASH.

Consigne 1 : même si l'analyse faite dans le rapport considère l'ensemble des résultats pour cette requête, ces configurations devront être évaluées séparément.

Consigne 2 : évitez de régénérer les données en créant des nouvelles tables pour les CLUSTER. Par exemple, supposons qu'on doit insérer T1 dans un CLUSTER Grappe1 avec la clé col1, on crée une nouvelle table T1\_cluster de la manière suivante :

```
CREATE TABLE T1_CLUSTER
CLUSTERR Grappe1(col1)
AS SELECT * FROM T1
```

Consigne 3 : choisissez le paramètre HASHKEYS du HASH CLUSTER de manière à éviter les débordements.

Note : les consignes 2 et 3 s'appliquent à tous les autres CLUSTER à créer pour le travail.

Pour plus d'informations sur les clusters, consultez la documentation Oracle :

[https://docs.oracle.com/cd/A97385\\_01/server.920/a96521/hash.htm](https://docs.oracle.com/cd/A97385_01/server.920/a96521/hash.htm)

## **Sélectivité grande**

Requête :

```
SELECT COUNT(*) FROM T1 WHERE T1.col3=1
```

Vous devrez évaluer cette requête en utilisant les configurations suivantes :

1. Sans optimisation avec l'indice FULL ;
2. Avec un index secondaire sur T1.col3 et l'indice INDEX ;
3. Avec un HASH CLUSTER sur T1 (clé col3) et l'indice HASH ;
4. Avec un BITMAP INDEX sur T1.col3 et l'indice INDEX. \*\*\*\*

Consigne 1 : même si l'analyse faite dans le rapport considère l'ensemble des résultats pour cette requête, ces configurations devront être évaluées séparément.

\*\*\*\* Consigne 2 : La création de BITMAP INDEX est impossible sur une version XE d'Oracle. Indiquez le code qui permettrait de créer votre index. Pour votre analyse, indiquez, selon ce que vous connaissez de cet index, si son utilisation améliorerait la performance ou non par rapport aux 3 autres méthodes indiquées ci-dessus.

## **Opérations de jointure**

### **Jointure simple**

Requête :

```
SELECT COUNT(*) FROM T1, T2 WHERE T1.col1 = T2.col1
```

Vous devrez évaluer cette requête en utilisant les configurations suivantes :

1. Sans optimisation avec l'indice FULL ;
2. Avec des index secondaires sur T1.col1 et T2.col1, de même qu'en spécifiant un des indices suivants (au choix) :
  - USE\_NL\_WITH\_INDEX;
  - USE\_HASH;
  - USE\_MERGE.
3. Avec un HASH CLUSTER hétérogène contenant T1 et T2 (clé col1), sans indice.
4. Avec deux INDEX CLUSTER homogènes contenant respectivement T1 et T2 (clé col1 pour les deux), sans indice.

Consigne 1 : même si l'analyse faite dans le rapport considère l'ensemble des résultats pour cette requête, ces configurations devront être évaluées séparément.

## Jointure double avec sélection

### Requête:

```
SELECT COUNT(*) FROM T1,T2,T3  
WHERE T1.col1=T3.col1 AND T2.col1 = T3.col2 AND T1.col2 = 1
```

Pour cette requête, vous devrez :

1. Proposer, dans un premier temps, une seule configuration (index, CLUSTER, etc.) permettant d'accélérer le plus possible cette requête ;
2. Analyser l'impact de modifier l'ordre des jointures en changeant l'ordre des tables dans le FROM et en utilisant l'indice ORDERED.

### Tâche 2 : Rédaction du rapport

Enfin, la dernière tâche sera de rédiger un rapport présentant pour chaque configuration de test :

1. Le code SQL employé pour créer le(s) index ou CLUSTER requis ;
2. La requête testée comprenant les indices et, possiblement, les nouvelles tables insérées dans les CLUSTER;
3. Le plan d'exécution et les statistiques obtenues lors de l'exécution de la requête. Cela peut être donné sous la forme d'un table ou une capture d'écran.

Le rapport devra également contenir, pour chaque catégorie d'opération (sélection avec facteur de sélectivité faible, sélection avec facteur de sélectivité élevé, jointure simple, et jointure double avec sélection), une courte analyse (10-15 lignes) expliquant les résultats obtenus à la lumière de la matière vue en classe.

Note : utilisez le gabarit prévu à cet effet.

## 3 Consignes de remise

- Le 24 décembre 2018 au plus tard, vous devrez remettre votre document **TP4-GabaritRapport.doc** (n'oubliez pas d'y inclure votre code SQL tel que demandé) sur le Moodle du cours.

## 4 Barème de correction

L'évaluation du travail sera faite sur un total de 100 points, distribués comme suit :

| COMPOSANTE                      | POINTS    |
|---------------------------------|-----------|
| Rapport :                       | (100 pts) |
| Code SQL pour le schéma interne | 15        |
| Reformulation des requêtes      | 15        |
| Méthodologie expérimentale      | 15        |
| Analyse des résultats           | 40        |
| Choix de la configuration       | 15        |
| TOTAL                           | 100       |

## 5 Pénalités pour retard

Une pénalité de 10% par jour, incluant les jours de fin de semaine, sera appliquée à tout travail remis en retard.