

TP OPTIMISATION

Pour ce TP, privilège l'utilisation du terminal pour visualiser le plan d'exécution.

Le rôle PLUSTRACE est nécessaire pour ce TP.

Vérifiez que vous avez le rôle "PLUSTRACE

```
SQL> select * from session_roles;
```

ROLE

```
-----  
CONNECT  
RESOURCE  
PLUSTRACE    <= !!!
```

Les tables de travail : utiliser les mêmes tables que celles utilisées durant le S3

Exercice 1 :

Collectez les statistiques sur vos tables de travail (CLIENT...). Ces statistiques permettent à l'optimisateur Oracle 'Cost Based Optimizer' de choisir le bon plan d'optimisation.

```
EXEC DBMS_STATS.GATHER_TABLE_STATS(USER,Nom_Table);
```

Remplacer Nom_Table par le nom de la table.

Vérifiez que les statistiques ont bien été collectées en exécutant la requête ci-dessous.

```
select TABLE_NAME, BLOCKS, EMPTY_BLOCKS, AVG_SPACE, CHAIN_CNT ,  
AVG_ROW_LEN, SAMPLE_SIZE, LAST_ANALYZED FROM user_tables;
```

Chercher la signification de chaque colonne dans la documentation.

IMPORTANT : il est important de faire la mise à jour des statistiques régulièrement notamment lorsqu'un index est créé ou lorsque la mise à jour de la table est faite.

Exercice 2 :

Pour récupérer le plan d'exécution et les statistiques de consommation, il faut exécuter la commande suivante : *SET AUTOTRACE ON* ;

Ouvrez un terminal, connectez-vous et exécutez la commande ci-dessus.

1. Ecrire une requête SQL pour récupérer les informations de la commande dont la référence est 008084.
2. Vérifiez le plan d'exécution ; interprétez-le.
3. Commentez les statistiques.
4. Réexécuter une seconde fois la même requête afin de générer les statistiques à nouveau.
5. Comparer les statistiques de la première et la seconde exécution. Que remarquez-vous ? comment justifiez-vous ce changement ?

Exercice 3 :

Exécuter la commande suivante :

```
SQL> SELECT ID,  
        ROWID,  
        DBMS_ROWID.ROWID_RELATIVE_FNO(rowid) FICHER,  
        DBMS_ROWID.ROWID_BLOCK_NUMBER(rowid) BLOC,  
        DBMS_ROWID.ROWID_ROW_NUMBER(rowid) NUMLIGNE  
FROM COMMANDE;
```

1. Quelles informations sont retournées ? Que représente la colonne ROWID ?
2. Récupérer l'adresse physique de la ligne correspondant à la commande 10 en ajoutant un filtre sur l'ID de la commande.
3. Utiliser la colonne rowid dans la clause WHERE ; *select * from commande where rowid='xxx'*.
4. Vérifier le plan d'exécution, les statistiques de consommation et commentez.
5. Combien de blocs sont lus ? Peut-on faire mieux ?

Exercice 4 :

1. Créer un index sur la colonne REFERENCE : *CREATE INDEX ix_reference ON COMMANDE(REFERENCE)* ;
2. Collectez les statistiques de l'index :

```
SQL> EXEC  
DBMS_STATS.GATHER_INDEX_STATS('ton_nom_utilisateur','nom_index');
```
3. Exécutez à nouveau la requête de l'exercice 2 qui vous permet de récupérer les informations de la commande dont la référence est 008084.
4. Que remarquez-vous ? y'a-t-il une différence entre le plan d'exécution/statistique de l'exercice 2 et celui obtenu ?
5. Supprimer l'index *ix_reference*. Créer un autre *index CREATE INDEX ix_id_reference ON COMMANDE (DATE_ACHAT, REFERENCE)* ; collectez les statistiques
6. Exécuter la requête précédente afin de récupérer les informations de la commande ayant la référence 008084. Que remarquez-vous ? commentez.
7. Modifier la requête précédente pour ne récupérer que la date d'achat : *SELECT DATE_ACHAT FROM COMMANDE WHERE REFERENCE='008084'*
8. Que remarquez-vous ? commentez.
9. Supprimer les index créés.

Exercice 5 :

- Créer un index sur la colonne *prix_total* de la table *ligne_commande* : *SQL> create index ix_prix_total on ligne_commande(PRIX_TOTAL)* ;
- Exécuter la requête suivante : *SELECT commande_id FROM LIGNE_COMMANDE WHERE PRIX_TOTAL>1000* ;
- Regarder les étapes du plan d'exécution.
- Exécuter la requête suivante : *SELECT commande_id FROM LIGNE_COMMANDE WHERE PRIX_TOTAL/2>500* ;
- Regarder les étapes du plan d'exécution.
- Quelle différence remarquez-vous ? Comment expliquez-vous cela ?
- Supprimer l'index créé.

Exercice 6 :

- Ecrire une requête retourne les informations des clients ainsi que les commandes passées (jointure entre les tables client et commande) .
- Vérifier le plan d'exécution. Quel algorithme de jointure est utilisé ?

Oracle utilise 3 algorithmes de tri : **Nested Loop**, **Sort Merge Join** et **Hash Join**. Il est possible de forcer l'utilisation d'un algorithme à l'aide de la commande HINTS.

USE_NL pour Nested Loop; **USE_MERGE** pour Merge Join et **USE_HASH** pour Hash Join.

Modifier la commande ci-dessous afin de tester les 3 algorithmes de tri :

```
SQL> SELECT /*+USE_NL(CLIENT COMMANDE) */ * FROM CLIENT cl INNER JOIN  
COMMANDE co ON cl.id = co.client_id.
```

Comparer les statistiques. Quelle est la meilleure option ?

- Créer un index sur la colonne id de la table CLIENT : *CREATE INDEX ix_client_id ON COMMANDE(client_id) ;*
- Réexécuter la commande précédente.
- Vérifier le plan d'exécution et les statistiques. Que remarquez-vous ?