

Fiche de TD/TP N°8- Optimisation des bases de données :

Lab Sur L'optimisation

Dans ce TP vous allez devoir analyser des requêtes, pour cela, vous devrez afficher les plans d'exécution des différentes requêtes. Cela peut être fait simplement avec la commande **SET AUTOTRACE TRACEONLY EXPLAIN STATISTICS** qui permet d'afficher le plan d'exécution de chaque requête exécutée.

1. Chargement des données :

- Connecter avec SQL Developer avec user donné(ex « **malki** »)
- Lancer le script data_TP8_2025.sql
 - produit (idp,nom, type) commande(idp, qte);**
Aucune clé – primaire ou secondaire – n'a été définie.

2. Récupération des informations et des Statistique de la base :

- Connecter à SQL*Plus avec le user : **SYS AS SYSDBA**
- Régler l'affichage en exécutant la commande : **set linesize 9999 ;**
- Afficher la taille d'un block : **show PARAMETERS db_block_size;**

3. Activer l'affichage du plan d'exécution : **SET AUTOTRACE TRACEONLY EXPLAIN STATISTICS ;**

4. Exécuter les requêtes suivantes et noter les plans d'exécutions obtenus. Les résultats sont-ils cohérents ?

- SELECT Min(idp), Max(idp) FROM malki.produit;**
- SELECT * FROM malki.produit WHERE idp=1;**
- SELECT * FROM malki.produit WHERE idp!=200;**
- SELECT count(*) FROM malki.produit WHERE idp!=200;**

5. Analyse des plans d'exécution :

- Créer une clé primaire pour la table produit :
 - ALTER TABLE malki.produit ADD PRIMARY KEY(idp)**
- Créer un index BTree sur le « **type** » :
 - CREATE INDEX malki.idx_type ON malki.produit(type);**
- Exécuter les requêtes suivantes en utilisant les différents modes d'évaluation :
 - RULE-based (RBO)** Le mode sans statistiques
 - En utilisant un HINT, ex ➔ **select /*+rule*/ count(*) FROM malki.produit WHERE idp!=200 ;**
 - COST-based (CBO)** Le mode avec statistiques: **default**
 - Dans le mode **CBO**, vous indiquer le **Cost** de chaque requêtes (**CPU_COST (default)** ou **IO_COST**)



- 1) `SELECT Min(idp), Max(idp) FROM malki.produit;`
 - 2) `SELECT * FROM malki.produit WHERE idp=1;`
 - 3) `SELECT * FROM malki.produit WHERE idp!=200;`
 - 4) `SELECT count(*) FROM malki.produit WHERE idp!=200;`
 - 5) `Select * from malki.produit where type = 'g';`
 - 6) `Select count(*) from malki.produit where type = 'g';`
- d. Mettre l'index **BTree(type)** à l'état invisible :
- `Alter index malki.idx_type invisible ;`
- e. Et créer un index Btmap :
- `CREATE BITMAP INDEX malki.BITMAP_type ON malki.Produit (type);`
- f. Réexécuter les requêtes (5,6)
- g. Analyser le plan des requêtes (Join) suivantes et proposer une optimisation :
- `Select count(*) from malki.produit p, malki.commande c where nom like 'produit100%' and p.idp=c.idp ;`
 - `Select count(*) from malki.produit p, malki.commande c where qte=1 and p.idp=c.idp ;`