

Université Mohamed Boudiaf M'sila



Administration des bases de données

Master 2 IDO

Année 2022/2023

Réalisé par Dr M.Sahraoui

Contenu

1. Le métier d'administration de base de données.
2. Composants de l'architecture d'Oracle.
3. Gestion d'une instance Oracle.
4. Administration physique des BDs: structures physiques de stockage, structures logiques de stockage, tuning logique des BDs.
5. Gestion de la sécurité.
6. Sauvegarde et restauration.
- 7. Optimisation des requêtes: Sauvegarde et restauration**

7. Optimisation des requêtes: Sauvegarde et restauration

- **Les buts d'optimisation**

- **Meilleur temps de réponse**

- ❖ Le but d'optimisation consistant à obtenir le meilleur temps de réponse permet de réduire le temps d'obtention de la première ligne du résultat.
 - ❖ Par exemple, une jointure par imbrication de boucles (NESTED LOOPS) favorise le temps de réponse.

- **Meilleur débit du résultat**

- ❖ Le but d'optimisation consistant à obtenir le meilleur débit du résultat permet de minimiser le temps global d'exécution (c.-à-d. pour obtenir la totalité) du résultat.
 - ❖ Par exemple, une jointure par tri et fusion (MERGE JOIN) favorise le débit du résultat.

7. Optimisation des requêtes: Sauvegarde et restauration

1. Les phases d'exécution d'une requête
2. Le plan d'exécution
3. L'accès aux données
4. Les méthodes d'optimisation
5. Paramétrage
6. Forcer les choix de l'optimiseur par programmation
7. Modification de la requête par l'optimiseur
8. Optimisation de la mémoire

1. Les phases d'exécution d'une requête

Les différentes phases d'exécution d'une requête sont l'analyse, l'exécution et la recherche. Toute requête suit le traitement

suivant :

- si l'analyse n'a pas déjà été effectuée

 - création d'un curseur

 - analyse (*parse*) de la requête

 - vérifications syntaxiques, vérifications d'existence et des droits sur les objets

 - verrouillages

 - planification de l'exécution, chargement en mémoire, répartition éventuelle

- si la requête est une requête d'interrogation

 - description des champs du résultat (notamment pour une requête interactive ou du SQL dynamique)

 - définition des variables en sortie devant accueillir le résultat

- substitution des variables (*bind*) par les valeurs courantes pour les paramètres en entrée

- exécution (*exec*) de la commande, avec verrouillage (suppression ou modification) ou sans verrouillage

 - (insertion ou

 - interrogation)

- si la requête est une requête d'interrogation

 - recherche (*fetch*), une à une, des lignes du résultat, avec tri éventuel

2. Le plan d'exécution

Le plan d'exécution d'une requête consiste pour Oracle à trouver le meilleur chemin pour accéder aux données, en minimisant le nombre d'opérations d'entrée/sortie et en minimisant le temps de traitement, et en optant pour une méthode statique ou statistique.

La commande `explain` permet d'expliquer le plan d'exécution d'une requête.

3. L'accès aux données

L'accès aux données (sur disque) peut s'effectuer :

- directement dans la table relativement à l'adresse physique (identification du fichier, bloc dans le fichier et ligne dans le bloc) de la ligne (ROWID)
- par un parcours d'un cluster indexé (*cluster scan*) où les lignes ayant même valeur pour la clé du cluster sont stockées dans le même bloc
- par un parcours d'un cluster par hachage (*hash scan*) où les lignes ayant même valeur de hachage sont stockées dans le même bloc
- par un parcours d'un index (*index scan*) pour retrouver les lignes grâce à une ou plusieurs colonnes de l'index ; l'index peut être unique (le parcours de la table d'index retourne une seule valeur ROWID) ou multiple (le parcours de la table d'index retourne une ou plusieurs valeurs ROWID)
- par un parcours séquentiel de toutes les données directement recherchées dans la table (*full table scan*)

4. Méthodes d'optimisation

- La méthode statique**
- La méthode statistique

• La méthode statique (1/2)

La méthode statique consiste à déterminer le plan d'exécution en fonction des différentes méthodes d'accès aux données disponibles en privilégiant celui de moindre coût compte-tenu de la classification suivante (en commençant par la plus efficace) :

- accès à une ligne par son adresse physique
- accès à une ligne par jointure clustérisée
- accès à une ligne par cluster par hachage avec clé unique
- accès à une ligne par clé unique
- jointure clustérisée (c.-à-d. les deux tables référencées par la jointure sont stockées dans le même cluster)
- clé de cluster par hachage
- clé de cluster indexé
- index composé (conjonction sur toutes ses colonnes qui testent l'égalité)

- **La méthode statique (2/2)**
- index unique
- recherche bornée sur des colonnes indexées
- recherche non bornée sur des colonnes indexées
- jointure avec tri et fusion
- fonctions d'agrégation minimum ou maximum de colonnes indexées
- tri sur une colonne indexée
- parcours complet de la table

4. Méthodes d'optimisation

- La méthode statique
- La méthode statistique**

- **La méthode statistique (1/2)**

La méthode statistique consiste à :

- commencer par générer les plans d'exécution en fonction des différents chemins d'accès aux données disponibles,
- estimer ensuite le coût (nombre d'entrées/sorties, temps CPU, mémoire nécessaire) de chaque plan d'exécution à partir des statistiques existantes sur les tables/colonnes/clusters/index concernés en considérant les critères de sélectivité (pourcentage de lignes ramenées) et de facteur de blocage (nombre de lectures multi-blocs dépendant du paramètre

`DB_FILE_MULTIBLOCK_READ_COUNT`),

- choisir finalement l'un des plans d'exécution de coût minimal.

- **La méthode statistique (2/2)**

- La commande `analyze` permet de générer des statistiques sur un objet. Par exemple, l'ordre `analyze index « index » compute statistics` lance une analyse complète (plutôt pour les index ou de petites tables) d'un index
- Tandis que la commande `analyze table « table » estimate statistics sample 20 percent` lance une analyse partielle (plutôt pour des tables volumineuses) d'une table.

5. Paramétrage

Le paramètre de session `OPTIMIZER_GOAL` (modifiable par la commande `alter session`) indique la méthode et le but poursuivis :

- statique et temps de réponse (`FIRST_ROWS`) indépendamment de la disponibilité de valeurs statistiques
- statique et débit du résultat (`ALL_ROWS`) indépendamment de la disponibilité de valeurs statistiques
- statistique (`CHOOSE`) : en présence de valeurs statistiques relatives à au moins une table référencée dans la requête (le but d'optimisation est alors le meilleur débit du résultat) ; en l'absence de valeurs statistiques, la méthode utilisée est alors statique

Le paramètre d'initialisation `OPTIMIZER_MODE` indique la méthode utilisée :

- statique (`RULE`) indépendamment de la disponibilité de valeurs statistiques
- statistique (`COST`) en présence de valeurs statistiques relatives à au moins une table référencée dans la requête (le but d'optimisation est alors le meilleur débit du résultat) ; en l'absence de valeurs statistiques, la méthode utilisée est alors statique

Le paramètre `OPTIMIZER_GOAL` est prioritaire sur le paramètre `OPTIMIZER_MODE`.

6. Forcer les choix de l'optimiseur par programmation

Comme le programmeur a souvent une connaissance plus fine de l'architecture des données que l'optimiseur, il peut forcer le comportement d'Oracle pour exécuter une requête SQL en y insérant un mot-clé (*hint*), entouré par `/*+` et `*/`, pour le choix :

- de la méthode et du but : statique (RULE), statique et temps de réponse (FIRST_ROWS), statique et débit du résultat (ALL_ROWS)
- de la méthode d'accès : parcours séquentiel (FULL), directement avec l'adresse physique (ROWID), parcours d'un cluster indexé (CLUSTER), parcours d'un cluster par hachage (HASH), parcours d'un index (INDEX)
- de la jointure : dans l'ordre dans lequel les tables se présentent (ORDERED), par des boucles imbriquées (USE_NL), par un algorithme tri et fusion (USE_MERGE)

7. Modification de la requête par l'optimiseur

Oracle modifie la requête en une syntaxe équivalente lorsque cela est possible :

- en évaluant les expressions

Par exemple, le prédicat $X=12/4$ and $X>Y$ and Z in ('un', 'deux') devient $Y<3$ and ($Z='un'$ or $Z='deux'$).

- en transformant une requête ayant un prédicat contenant un `or` en une requête `union all` si cela améliore l'exécution (c.-à-d. si chaque composante de la condition génère un chemin d'accès basé sur un index ou si l'une des deux conditions impose le parcours complet de la table en l'absence d'index)
- en transformant une requête complexe en une jointure équivalente (pour une sous-requête par exemple)
- en prenant en compte les vues, soit en propageant la définition de la vue dans la requête, soit en propageant la définition de la requête dans la vue

8. Optimisation de la mémoire

Il s'agit d'obtenir une bonne distribution de la mémoire centrale pour les différentes structures internes d'Oracle de sorte à maximiser les données accessibles en mémoire (c.-à-d. minimiser les défauts mémoire).

Certains paramètres définissent la taille de la mémoire :

- du cache base de données (DB_BLOCK_BUFFERS)
- de la zone de reprise (LOG_BUFFER)
- du pool partagé (SHARED_POOL_SIZE).

La zone privée SQL et PL/SQL contient l'analyse et le plan d'exécution des commandes SQL partagées ;

- le paramètre OPEN_CURSORS définit le nombre maximum de zones par utilisateur.