

M2 DOC TP Bases de données avancées

Plan d'exécution et optimisation

Source : Philippe Rigaux

Pratique de Mysql et Php, Dunod

Mise en route

1. Connectez-vous à la machine oracle10 : `ssh <votre login>@oracle10`
2. Copiez les fichiers de `/share/m2infopro/DOC/tp/tp2` dans votre compte.
3. Lancez le script de configuration pour Oracle (choisir dbtp comme ORACLE_SID) :
`source /usr/local/env/envora.sh`
4. Lancer l'interface avec Oracle : `sqlplus /`
5. Chargez la base de données Cinéma, en exécutant successivement les scripts `Films.sql`, `ComplFilms.sql`, `BaseFilms.sql` (Si avez besoin d'effacer les tables, vous pouvez utiliser `DropFilms.sql`.):
`@Films ; (puis @ComplFilms ; etc.)`

Plan d'exécution

Un plan d'exécution est un programme combinant des opérateurs physiques (chemins d'accès et traitements de données). Il a la forme d'un arbre : chaque nœud est un opérateur qui

- prend des données en entrée,
- applique un traitement,
- produit les données traitées en sortie.

La phase d'optimisation du SGBD fonctionne ainsi : pour une requête SQL donnée, le système a le choix entre plusieurs plans d'exécution. Ils diffèrent par l'ordre des opérations, les algorithmes, les chemins d'accès. Pour chaque plan on peut estimer :

- le coût de chaque opération,
- la taille du résultat.

L'objectif du SGBD est de diminuer le plus vite possible la taille des données manipulées.

Mettre SQLPLUS en mode AUTOTRACE, ce qui permet de visualiser le plan d'exécution exécuté pour chaque requête soumise. Pour bien visualiser le plan, agrandissez votre fenêtre et indiquez à SQLPLUS une plus grande taille de ligne :

```
SET AUTOTRACE ON
SET LINESIZE 200
```

6. Obtenez le plan d'exécution pour la requête obtenant les informations du film 'Vertigo'.

```
SELECT * FROM Film WHERE titre = 'Vertigo' ;
```

7. Si seul le plan d'exécution vous intéresse, vous pouvez utiliser le mode TRACEONLY : `SET AUTOTRACE TRACEONLY`

8. Que fait l'opérateur physique TABLE ACCESS FULL ?
(voir http://docs.oracle.com/cd/B19306_01/server.102/b14211/ex_plan.htm#i23461)
9. Expliquez la stratégie choisie pour évaluer la jointure suivante :

```
select nom,prenom from film f,artiste a where f.annee=a.annee_naissance and  
titre='Vertigo';
```

Y-a-t-il des index pour les attributs utilisés ?

10. Expliquez la stratégie choisie pour évaluer cette autre requête :
- ```
SELECT * FROM Film WHERE id=21;
```

Pour cela, identifiez la signification de UNIQUE SCAN et BY ROWID. Y-a-t-il présence d'un index sur l'attribut id ?

11. Comment fonctionne l'évaluation proposée pour la requête suivante ?

```
SELECT titre, nom, prenom FROM Film f, Artiste a WHERE f.id_realisateur = a.id;
```

Aurait-on pu utiliser l'index ?

**Vous êtes arrivé au checkpoint, bravo ! Prévenir votre chargé de TP !**

12. Obtenez le plan d'exécution de la requête

```
SELECT titre FROM Film f, Role r, Artiste a WHERE a.nom = 'Stewart' AND
a.prenom='James' AND f.id = r.id_film AND r.id_acteur = a.id AND f.annee = 1958 ;
```

13. La requête suivante est-elle équivalente (du point de vue sémantique) ?

```
SELECT titre FROM Film f, Role r WHERE f.id = r.id_film AND f.annee = 1958 AND
r.id_acteur IN (SELECT id FROM Artiste WHERE nom='Stewart' AND prenom= 'James') ;
```

14. Y-a-t-il une différence sur le plan d'exécution ?

15. Mêmes questions avec le requête suivante :

```
select titre from Film f, Role r where f.id = r.id_film and f.annee = 1958 and EXISTS
(SELECT 'x' FROM Artiste a WHERE nom= 'Stewart' AND prenom='James' AND
r.id_Acteur = a.id)
```

16. Encore les mêmes questions avec le requête suivante :

```
SELECT titre FROM Film WHERE annee = 1958 AND id IN (SELECT id_film FROM Role
WHERE idActeur IN (SELECT id FROM Artiste WHERE nom='Stewart' AND
prenom='James'))
```

17. Que pouvez-vous conclure de l'impact de la rédaction SQL sur l'optimiseur ?