

le cnam

Ingénierie et Optimisation de Bases de Données

NFE106

Travaux Pratiques

CNAM Paris

nicolas.travers (at) cnam.fr

1 Etude de la base de données	3
1.1 Informations	3
2 Analyse de plan EXPLAIN	4
2.1 Instructions	4
2.1.1 Pour SQL*PLUS	4
2.1.2 Pour SQLDeveloper	4
2.1.3 Pour toute requête à exécuter	4
2.2 Étude de plans d'exécution	4
2.2.1 Requêtes Simples	5
2.2.2 Requêtes de jointures	5
3 Optimisation	7
3.1 HINT	7
3.2 Optimisation de requête	7
3.3 EXPLAIN vers SQL	8
3.4 Scénario	8

Avant de commencer, veuillez lire le document sur les **Informations Pratiques** qui vous permettront de mieux appréhender ce TP et de vous permettre de vous connecter à la base de données (SSH, TOAD ou SQL Developer). Il vous donne tous les éléments nécessaires au bon déroulement de ce TP.

Le but de ce TP est d'étudier le contenu de la base de données. Vous devrez en extraire les informations nécessaires à la bonne compréhension des plans d'exécutions et statistiques associées.

1.1 Informations

Remplir les tableaux suivants :

— **Tables et Indexes**

Nom	Nb tuples	Nb pages	Indexes (type + attribut)
ARTISTE			
FILM1			
FILM2			
FILM3			
FILM4			
FILM5			
FILM6			
INTERNAUTE			
NOTATION			
PAYS			
ROLE			

— **Attributs**

Nom	Cardinalité	Min	Max
FILM1.IDFILM			
FILM1.ANNEE			
FILM1.CODEPAYS			

Le but de ce TP est de comprendre le fonctionnement de l'optimiseur à travers l'outil EXPLAIN. Vous aurez un ensemble de requêtes à effectuer sur la base de données que vous avez étudié précédemment.

2.1 Instructions

2.1.1 Pour SQL*PLUS

- Nous ne souhaitons voir que le plan EXPLAIN et les statistiques, pas les résultats. Pour ce faire :

```
SET AUTOTRACE TRACEONLY EXPLAIN STATISTICS;
```

- Pour vous faciliter la tâche de recopie de requêtes, ouvrez un éditeur de texte¹ pour y écrire vos requêtes puis les copier dans SQLPLUS.

2.1.2 Pour SQLDeveloper

- Pour afficher le plan d'exécution et les statistiques, veuillez vous référer au guide.

2.1.3 Pour toute requête à exécuter

Voici les commandes qui devront être utilisées dans le cadre de l'exécution de requêtes :

RULE Le mode sans statistiques :

```
ALTER SESSION SET OPTIMIZER_MODE=RULE;
```

CHOOSE Le mode avec statistiques permet de voir un meilleur plan d'exécution² :

```
ALTER SESSION SET OPTIMIZER_MODE=CHOOSE;
```

CACHE Afin de vider le cache avant chaque exécution³ :

```
ALTER SYSTEM FLUSH BUFFER_CACHE;
```

FILMi Une requête sur la table FILM peut se faire sur chaque version (FILM1 à FILM6). Il vous sera demandé à chaque fois un sous-ensemble de requêtes à exécuter, ici *FILM1*.

```
select titre, genre from FILM1 f where IDFILM=50273 ;
```

2.2 Étude de plans d'exécution

Pour chaque requête à exécuter :

- Exécuter la requête sur les différentes versions demandées de la table FILM_i (de 1 à 6) ;
- Par défaut, nous utiliserons le mode CHOOSE. Si cela est précisé, vous aurez : RULE/CHOOSE ;
- N'oubliez pas de vider le cache ;

1. nedit par exemple

2. Toujours en fonction des statistiques qui lui sont fournies

3. Sélectionnez une requête, puis utilisez la copie avec le bouton centrale de la souris

— Pour le TP, veuillez remplir un tableau pour chaque requête, comme dans l'exemple suivant :

Table	MODE	Opérateurs	Consistent Gets	Physical Reads
FILM1	RULE	INDEX RANGE SCAN + ACCESS ROWIDS	14177	14167
FILM1	CHOOSE	TABLE ACCESS FULL	10165	10000
FILM2	CHOOSE	TABLE ACCESS FULL	10165	10000

— Expliquer pour chaque question les particularités obtenues ⁴.

Exemple avec le tableau précédent : Le mode RULE utilise l'index car il peut être utilisé. Toutefois, le mode avec CHOOSE constate que celui-ci coûtera trop cher et fait un parcours séquentiel. Idem pour FILM2.

Pour chaque question, un ensemble d'exécutions vous est demandé. Voici un exemple de séquence obtenue pour FILM1, FILM6, RULE, CHOOSE :

```
ALTER SESSION SET OPTIMIZER_MODE=RULE;
alter system flush buffer_cache;
select titre, genre from FILM1 f where IDFILM=50273 ;
alter system flush buffer_cache;
select titre, genre from FILM6 f where IDFILM=50273 ;

ALTER SESSION SET OPTIMIZER_MODE=CHOOSE;
alter system flush buffer_cache;
select titre, genre from FILM1 f where IDFILM=50273 ;
alter system flush buffer_cache;
select titre, genre from FILM6 f where IDFILM=50273 ;
```

2.2.1 Requêtes Simples

(1) FILM1, FILM4, FILM6 :

```
SELECT TITRE FROM FILMi WHERE IDFILM=50273 ;
```

(2) FILM1 et FILM6 :

```
SELECT COUNT(*) FROM FILMi WHERE IDFILM BETWEEN 50273 AND 60000 ;
```

(3) FILM2, FILM4, FILM5, RULE, CHOOSE :

```
SELECT TITRE FROM FILMi WHERE ANNEE=1999 ;
```

(4) FILM2, FILM3 :

```
SELECT COUNT(*) FROM FILMi WHERE ANNEE=1999 ;
```

(5) FILM2, FILM3, FILM5, RULE, CHOOSE :

```
SELECT TITRE FROM FILMi WHERE CODEPAYS='aaej' ;
```

2.2.2 Requêtes de jointures

(1) FILM2, FILM3, FILM5, RULE, CHOOSE :

```
SELECT TITRE, NOM FROM FILMi F, ARTISTE A
WHERE CODEPAYS='aaej' AND F.IDMES=A.IDARTISTE ;
```

(2) FILM3 et FILM4 :

4. En fonction de l'organisation des tables FILMs que vous avez étudié dans le premier chapitre

```
SELECT NOM, COUNT(*) FROM FILMi F, ARTISTE A
WHERE ANNEE=1999 AND F.IDMES=A.IDARTISTE
GROUP BY NOM ;
```

(3) FILM3 et FILM4 :

```
SELECT NOM, COUNT(*) FROM ARTISTE
WHERE IDARTISTE IN
  (SELECT DISTINCT IDMES FROM FILMi
   WHERE ANNEE=1999)
GROUP BY NOM ;
```

(4) FILM2, FILM3 et FILM5 :

```
SELECT COUNT(*) FROM FILMi F, ROLE R, ARTISTE A
WHERE F.IDFILM=R.IDFILM AND
  R.IDACTEUR=A.IDARTISTE AND F.IDMES=A.IDARTISTE AND
  CODEPAYS='aej' AND ANNEE=1999;
```

(5) FILM2, FILM6 ;

```
SELECT TITRE FROM FILMi F, NOTATION N
WHERE F.IDFILM = N.IDFILM
GROUP BY TITRE
HAVING AVG(NOTE) > 15;
```

(6) FILM2, FILM6 ;

```
SELECT TITRE FROM FILMi F,
  (SELECT IDFILM, AVG(NOTE) as AVG_NOTE FROM NOTATION N
   GROUP BY IDFILM) N
WHERE F.IDFILM = N.IDFILM AND AVG_NOTE > 15;
```

Pour aller plus loin, vous pouvez tester les requêtes mais sans vider le cache avant l'exécution (faites deux exécutions successives);

3.1 HINT

Dans cette partie, il vous est demandé d'utiliser le HINT (*Full, Index, et USE_NL*) dans les requêtes suivantes. Pour rappel l'utilisateur peut forcer l'optimiseur à faire certains choix, avec la commande `/*+ HINT un_conseil */` insérée juste après le SELECT.

(1)

```
SELECT TITRE FROM FILMi WHERE CODEPAYS= 'aaej'
```

(2)

```
SELECT NOM, COUNT(*) FROM ARTISTE
WHERE IDARTISTE IN
      (SELECT DISTINCT IDMESFROM FILMi
       WHERE ANNEE= 1999)
GROUP BY NOM ;
```

(3)

```
SELECT TITRE FROM FILMi F, NOTATION N
WHERE F.IDFILM = N. I DFILM
GROUP BY TITRE
HAVING AVG(NOTE) > 15 ;
```

3.2 Optimisation de requête

La requête suivante n'est pas optimisée. Étudier le plan EXPLAIN généré (sans cache), et proposer une nouvelle requête SQL produisant le même résultat mais avec moins d'entrées/sorties (consistent get/physical read).

```
SELECT titre FROM film1
WHERE exists (
  SELECT idfilm
  FROM ROLE R
  WHERE film1.idfilm = r.idfilm and
        nomrole like 't%'
  GROUP BY idfilm
  HAVING count(*) > 1);
```

3.3 EXPLAIN vers SQL

Le plan EXPLAIN ci-dessous fait des parties des fichiers de logs du DBA. Trouver la requête SQL ayant produit ce plan d'exécution.

```

0. SELECT STATEMENT
1*. HASH JOIN RIGHT SEMI
2*. TABLE ACCESS FULL. ROLE
3. TABLE ACCESS FULL. FILM1

1 - access("IDFILM"="IDFILM")
2 - filter("NOMROLE" LIKE 't%')

```

3.4 Scénario

En consultant les statistiques d'accès à la base de données, on constate que celui reçoit régulièrement les trois requêtes ci-dessous avec leur fréquence :

R1 1 200 fois par jour

```
SELECT TITRE FROM FILMi WHERE CODEPAYS= 'aej'
```

R2 300 fois par jour

```

SELECT NOM, COUNT(*)
FROM FILMi F , ARTISTE A
WHERE ANNEE= 1999 AND F.I DMES = A.I DARTISTE
GROUP BY NOM ;

```

R3 100 fois par jour

```

SELECT TITRE FROM FILMi F, NOTATION N
WHERE F.IDFILM = N. I DFI LM
GROUP BY TITRE
HAVING AVG(NOTE) > 15 ;

```

- (1) Donner un tableau de "Physical Reads" pour chaque requête sur chacune des tables FILM1 à FILM6.
- (2) Calculer le coût généré pour chaque Film_i par jour, en fonction de la fréquence de chaque requête (le cache n'est pas considéré). En déduire la table la plus optimisée pour notre cas.
- (3) Peut-on encore améliorer les performances du SGBD ? (index supplémentaire, organisation différente, extensions...)