

Systèmes de gestion de bases de données

L'optimiseur ORACLE

P. Rigaux

Cnam, dépt. informatique

May 11, 2015

L'optimiseur

L'optimiseur ORACLE suit une approche classique :

- Génération de plusieurs plans d'exécution.
- Estimation du coût de chaque plan généré.
- Choix du meilleur et exécution.

Tout ceci est automatique, **mais il est possible d'influer, voire de forcer le plan d'exécution.**

Les chemins d'accès

Ceux que nous avons déjà vus.

- **Parcours séquentiel** (*FULL TABLE SCAN*).
- **Par adresse** (*ACCESS BY ROWID*).
- **Parcours d'index** (*INDEX SCAN*).

Opérations physiques

Voici les principales :

- *INTERSECTION* : intersection de deux ensembles de n-uplet.
- *CONCATENATION* : union de deux ensembles.
- *FILTER* : élimination de n-uplets (sélection).
- *PROJECTION* : opération de l'algèbre relationnelle.

D'autres opérations sont liées aux algorithmes de jointures.

Algorithmes de jointure sous ORACLE

ORACLE utilise trois algorithmes de jointures :

- **Boucles imbriquées** quand il y a au moins un index.
Opération NESTED LOOP.
- **Tri/fusion** quand il n'y a pas d'index.
Opération SORT et MERGE.
- **Jointure par hachage** quand il n'y a pas d'index.
Opération HASH JOIN

L'outil EXPLAIN

L'outil EXPLAIN donne le plan d'exécution d'une requête. La description comprend :

- Le chemin d'accès utilisé.
- Les opérations physiques (tri, fusion, intersection, ...).
- L'ordre des opérations.

Il est représentable par un arbre.

Rappel du schéma

- Film (**idFilm**, titre, année, genre, résumé, *idMES*, *codePays*)
- Artiste (**idArtiste**, nom, prénom, annéeNaissance)
- Role (**idActeur**, **idFilm**, nomRôle)
- Internaute (**email**, nom, prénom, région)
- Notation (**email**, **idFilm**, note)
- Pays (**code**, nom, langue)

Sélection sans index

La requête :

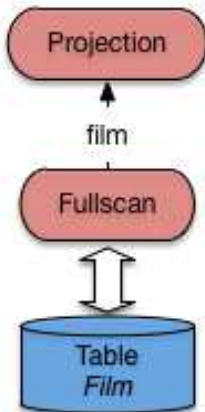
```
explain plan
set statement_id='SelSansInd' for
select *
from Film
where titre = 'Vertigo'
```

Le résultat de EXPLAIN :

```
0 SELECT STATEMENT
1 TABLE ACCESS FULL FILM
```


Plan d'exécution

On ne fait pas plus simple !



Sélection avec index

La requête :

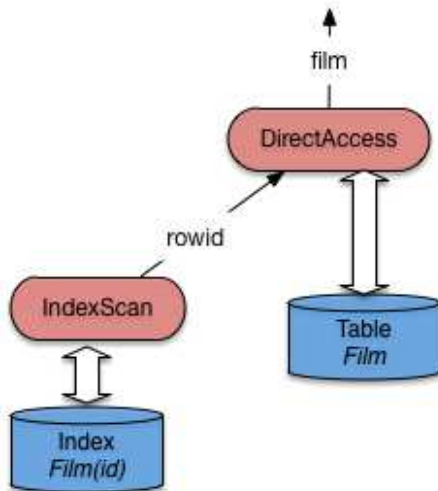
```
explain plan
set statement_id='SelInd' for
select *
from Film
where id=21;
```

Le résultat de EXPLAIN :

```
0 SELECT STATEMENT
1 TABLE ACCESS BY ROWID FILM
2 INDEX UNIQUE SCAN IDX-FILM-ID
```

Plan d'exécution

Accès à l'index, puis à la table.



Jointure avec index

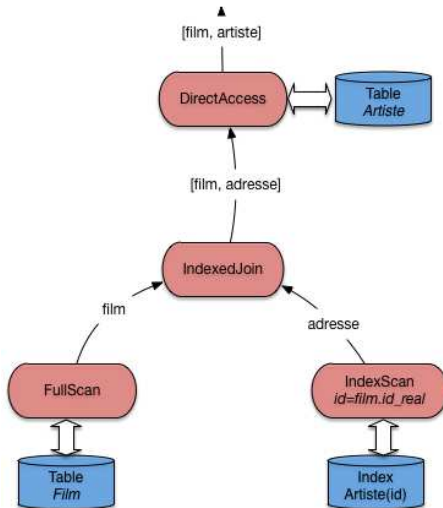
La requête :

```
explain plan
set statement_id='JoinIndex' for
select titre, nom, prenom
from   Film f, Artiste a
where  f.id_realisateur = a.id;
```

Le résultat de EXPLAIN :

```
0 SELECT STATEMENT
1 NESTED LOOPS
2 TABLE ACCESS FULL FILM
3 TABLE ACCESS BY ROWID ARTISTE
4 INDEX UNIQUE SCAN IDXARTISTE
```

Plan d'exécution (déjà vu)



Jointure avec index et sélection

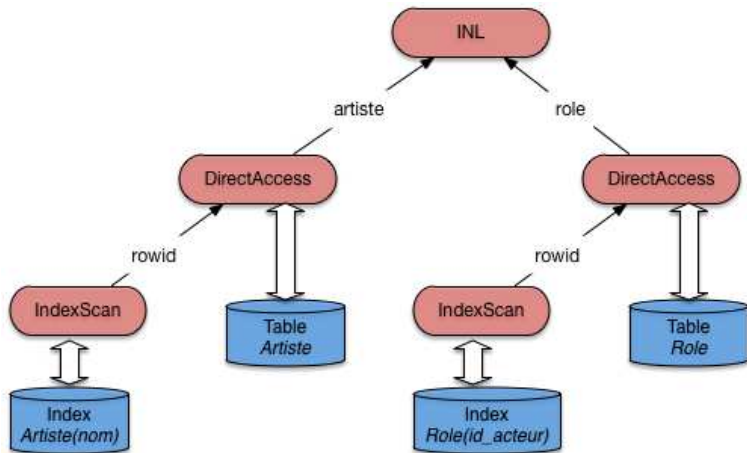
La requête (avec index sur le nom des artistes) :

```
explain plan
set statement_id='JoinSelIndex' for
select nom_role
from   Role r, Artiste a
where  r.id_acteur = a.id
and    nom = 'Pacino';
```

Le résultat de EXPLAIN :

```
0 SELECT STATEMENT
1 NESTED LOOPS
2 TABLE ACCESS BY ROWID ARTISTE
3 INDEX RANGE SCAN IDX-NOM
4 TABLE ACCESS BY ROWID ROLE
5 INDEX RANGE SCAN IDX-ROLE
```

Plan d'exécution



Jointure sans index

Requête:

```
explain plan set  
statement_id='JoinSansIndex' for  
select nom, prenom  
from Film f, Artiste a  
where f.annee = a.annee_naissance  
and titre = 'Vertigo';
```

Le résultat de EXPLAIN :

```
0 SELECT STATEMENT  
1 MERGE JOIN  
2 SORT JOIN  
3 TABLE ACCESS FULL ARTISTE  
4 SORT JOIN  
5 TABLE ACCESS FULL FILM
```


Plan d'exécution

