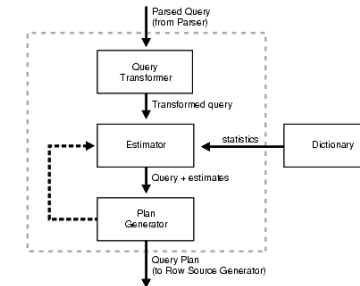


## 3I009 Licence d'informatique

### Mode Explain d'oracle

## Les composants de l'optimiseur



## Tuning

Le comportement du SGBD dépend de sa configuration

- Création, suppression d'index
  - create index nom\_index on nom\_relation(nom\_attribut);
  - drop index nom\_index;
- Taille des caches
- Taille du journal
- Etc.

Pour agir, il est nécessaire de pouvoir comprendre ce qui se passe, pourquoi une requête prend du temps.

**Explain** : demander au SGBD ce qu'il fait afin de lui demander éventuellement de faire autrement

## Explain : principe

- Permet d'observer (comprendre) comment Oracle optimise les requêtes (accès aux tables, algorithmes de jointure...)
- Permet d'agir sur l'optimiseur
  - Forcer l'utilisation (ou non) des index
  - Optimisation à base de coût ou de règles heuristiques
  - Utilisation ou non des statistiques (les statistiques doivent être mise à jour, le fameux compromis...)

## Explain : observation

explain stat / set autotrace off

- Permet de visualiser le plan d'exécution et d'avoir des informations quantitatives (métriques) sur les ressources utilisées

```
SELECT phone_number FROM employees
WHERE phone_number LIKE '650%';
```

Id	Operation	Name
0	SELECT STATEMENT	
1	TABLE ACCESS FULL	EMPLOYEES

```
SELECT last_name FROM employees
WHERE last_name LIKE 'Pe%';
```

Id	Operation	Name
0	SELECT STATEMENT	
1	INDEX RANGE SCAN	EMP_NAME_IX

UPMC - UFR 919 Ingénierie - Cours Bases de données I3009 Evaluation et optimisation - 5 5

## Explain : observation

```
SELECT employee_id, last_name
FROM employees
WHERE last_name LIKE :b1;
```

Identificateur de n-uplet interne

Id	Operation	Name	Rows	Bytes	Cost (%CPU)
0	SELECT STATEMENT		5	95	3 (34)
1	TABLE ACCESS BY INDEX ROWID	EMPLOYEES	5	95	3 (34)
* 2	INDEX RANGE SCAN	EMP_NAME_IX	2		3 (34)

Predicate Information (identified by operation id):

```
2 - access("EMPLOYEES"."LAST_NAME" LIKE :Z)
   filter("EMPLOYEES"."LAST_NAME" LIKE :Z)
```

UPMC - UFR 919 Ingénierie - Cours Bases de données I3009 Evaluation et optimisation - 6 6

## Opérateurs algébriques

nom	algorithme
nested loops	jointure par boucles imbriquées
merge join	jointure par fusion
hash join	jointure avec hachage temporaire des tuples
sort (join)	tri préliminaire avant jointure par fusion
table access full	lecture séquentielle d'une relation
table access by rowid	lecture non séquentielle d'une relation (un accès par tuple, rowid provient d'un parcours d'index)
index (range scan)	traversée d'un index (arbre B+)
bitmap index/minus/or/...	génère et manipule des index bitmaps
Hash group by etc.	

UPMC - UFR 919 Ingénierie - Cours Bases de données I3009 Evaluation et optimisation - 7

## Explain : métriques

- recursive calls
- db block gets : dernière version
- consistent gets : dernière versions validée
- physical reads : depuis le disque + le cache
- redo size : redo généré
- ....
- sorts (memory)
- sorts (disk)
- rows processed

UPMC - UFR 919 Ingénierie - Cours Bases de données I3009 Evaluation et optimisation - 8 8

## Actions : modes d'optimisation

alter session set optimizer\_mode =

- Permet de visualiser le plan d'exécution et d'avoir des informations quantitatives sur les ressources utilisées
- Modes :
  - ALL\_ROWS : par défaut
  - FIRST\_ROWS : Gets the first row faster (generally forces index use).
  - FIRST\_ROWS (1|10|100|1000) Gets the first n rows faster.
  - CHOOSE : optimisation à base de coût. Plus maintenu
  - RULE : optimisation à base de règles heuristiques. Plus maintenu

## Actions : directives (hints) d'optimisation

```
select /*+ use_nl(c,j,f) ordered */ c.nom, f.budget
from J, C, F
where J.cnum = C.cnum and C.cnum = F.cnum
and c.division=1 and J.salaire > 59000
and j.sport = 'sport1';
```

/\*+ use\_nl(c,j,f) \*/ : force l'utilisation de l'algo boucles imbriquées pour les table C,J,F  
 /\*+ ordered \*/ : fixe l'ordre de traitement des jointures : l'ordre imposé est celui indiqué dans la clause FROM de la requête.

/\*+ index(nom\_relation nom\_index) \*/ : pour forcer un accès par index  
 /\*+ no\_index(nom\_relation nom\_index) \*/ : pour interdire l'utilisation d'un index

Et aussi : NO\_USE\_NL, USE\_NL\_WITH\_INDEX, USE\_MERGE, NO\_USE\_MERGE, USE\_HASH, NO\_USE\_HASH, ...

## Action : analyse statistique des tables

analyze table .. compute statistics / analyze table ...delete statistics

Maintient des statistiques (nb nuplets, nb valeurs distinctes, etc..) utilisées pour estimer le coût plus précisément.

Le gain n'est pas forcément valable par rapport aux pertes sur les mises à jour

Lorsque l'optimiseur ne dispose pas de statistiques, il tente d'estimer la taille des résultats intermédiaires à partir d'un échantillon de données (technique appelée dynamic sampling). Désactiver le dynamic sampling en ajoutant la directive suivante.

```
select /*+ use_nl(c,j,f) dynamic_sampling(0) */ c.nom, f.budget
```

## Exemple

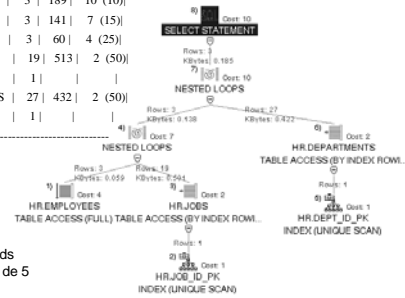
```
SELECT e.employee_id, j.job_title, e.salary, d.department_name
FROM employees e, jobs j, departments d
WHERE e.employee_id < 103 AND e.job_id = j.job_id AND e.department_id = d.department_id;
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)
0	SELECT STATEMENT		3	189	10 (10)
1	NESTED LOOPS		3	189	10 (10)
2	NESTED LOOPS		3	141	7 (15)
* 3	TABLE ACCESS FULL	EMPLOYEES	3	60	4 (25)
4	TABLE ACCESS BY INDEX ROWID	JOBS	19	513	2 (50)
* 5	INDEX UNIQUE SCAN	JOB_ID_PK	1		
6	TABLE ACCESS BY INDEX ROWID	DEPARTMENTS	27	432	2 (50)
* 7	INDEX UNIQUE SCAN	DEPT_ID_PK	1		

```
3 - filter("E"."EMPLOYEE_ID"<103)
5 - access("E"."JOB_ID"="J"."JOB_ID")
7 - access("E"."DEPARTMENT_ID"="D"."DEPARTMENT_ID")
```

## Exemple

Id	Operation	Name	Rows	Bytes	Cost (%CPU)
0	SELECT STATEMENT		3	189	10 (10)
1	NESTED LOOPS		3	189	10 (10)
2	NESTED LOOPS		3	141	7 (15)
3	TABLE ACCESS FULL	EMPLOYEES	3	60	4 (25)
4	TABLE ACCESS BY INDEX ROWID	JOBS	19	513	2 (50)
5	INDEX UNIQUE SCAN	JOB_ID_PK	1		
6	TABLE ACCESS BY INDEX ROWID	DEPARTMENTS	27	432	2 (50)
7	INDEX UNIQUE SCAN	DEPT_ID_PK	1		



- 3: lit tout Employees
- 5 : cherche les rowids correspondant à tous les jobids
- 4 : retrouve les n-uplets correspondants aux rowids de 5
- 7 & 6 : idem que 4&5 mais pour Departments
- 2 : boucles imbriquées Employees et Jobs
- 1 : boucles imbriquées 2 et Departments