

Mise à jour des statistiques

- On doit mettre à jour les statistiques
 - A chaque mise à jour d'une table (insert, delete, update)
 - A chaque modification de schema (en particulier creation d'index, modification d'index qui deviendrait groupant etc.)
- ➔ En utilisant la commande **ANALYSE** *nomTable*

Plans de requêtes

- Commande SQL pour afficher le plan de requêtes calculé par le SGBD avec estimations des coûts
- **EXPLAIN** *req*: fournit uniquement le plan, la requête n'est pas évaluée.
- **EXPLAIN ANALYSE** *req*: évaluation de la requête et affichage des coûts réels sont affichés.

NB: si les coûts réels divergent des coûts estimés, l'optimiseur s'est trompé (statistiques pas à jour etc.)

Tables du TP

```
CREATE TABLE PEOPLE (pid INTEGER PRIMARY key,  
firstname VARCHAR(30),  
lastname VARCHAR(30));
```

```
CREATE TABLE MOVIE (mid INTEGER PRIMARY KEY,  
title VARCHAR(90) NOT NULL,  
year INTEGER NOT NULL,  
runtime INTEGER NOT NULL,  
rank INTEGER NOT NULL);
```

```
CREATE TABLE ROLE (mid INTEGER REFERENCES MOVIE,  
pid INTEGER REFERENCES PEOPLE,  
name VARCHAR(70),  
PRIMARY KEY(mid, pid, name));
```

```
CREATE TABLE DIRECTOR (mid INTEGER REFERENCES MOVIE,  
pid INTEGER REFERENCES PEOPLE,  
PRIMARY KEY (mid, pid));
```

Exemple

- EXPLAIN ANALYSE SELECT * FROM ROLE,PEOPLE
WHERE ROLE.pid = PEOPLE.pid;

Hash Join (cost=312.07..822.95 rows=14535 width=37)

(actual time=14.799..44.691 rows=14535 loops=1)

Hash Cond: (role.pid = people.pid)

-> Seq Scan on role (cost=0.00..238.35 rows=14535 width=20)

(actual time=0.019..7.570 rows=14535 loops=1)

-> Hash (cost=175.92..175.92 rows=10892 width=17)

(actual time=14.711..14.711 rows=10892 loops=1)

-> Seq Scan on people (cost=0.00..175.92 rows=10892 width=17)

(actual time=0.015..5.944 rows=10892
loops=1)

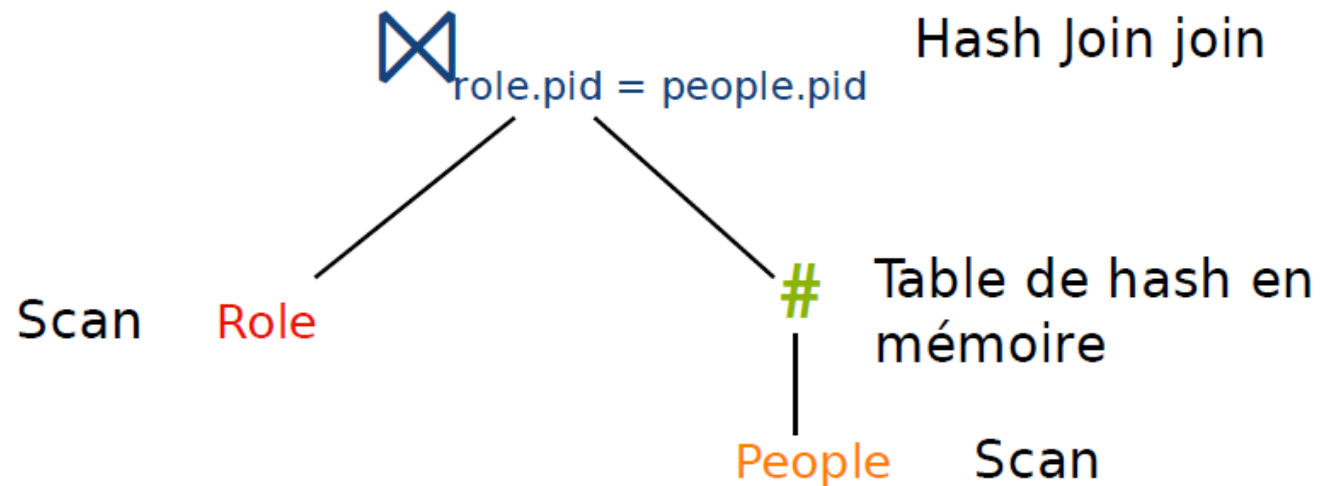
Evaluation des coûts

- Seq Scan on people (*cost=0.00..175.92 rows=10892 width=17*)
(*actual time=0.015..5.944 rows=10892 loops=1*)
- **Estimation du coût** (première parenthèse)
 - Unité : temps que met une lecture de bloc de 8ko (pour être indépendant du hardware)
 - Premier nombre = le temps estimé pour avoir le premier résultat. [0.00]
 - Deuxième le temps estimé pour avoir l'ensemble. [175.92]
 - Estimation du nombre de lignes dans le résultat [10892]
 - Taille des lignes en octets [17]
- **Coût réel** (deuxième parenthèse)
 - Unité : ms
 - Temps pour avoir le premier résultat [0.015]
 - Temps pour avoir l'ensemble [5.944]
 - Nombre de lignes dans le résultat [10892]
 - looks=x l'opérateur a été appelé x fois [1]

Lecture du plan de requête

```

Hash Join (cost=...) (actual time=...)
  Hash Cond: (role.pid = people.pid)
    -> Seq Scan on role (cost=...) (actual time=...)
    -> Hash (cost=...) (actual time=...)
      -> Seq Scan on people (cost=...) (actual time=...)
  
```



Opérateurs et implémentations

- **Seq Scan** : Scan séquentiel
- **Nested loop** : Jointure itérative page à page
- **Merge sort join** : Jointure par tri fusion
- **Hash join** : jointure par hashage (jointure sur index)
- **Sort** : Tri (le noeud indique l'algo de tri et la fonction de comparaison)
- **Index scan** : scan d'un index (précise l'index et la condition)
- **Hash** : génération d'une table de hash à la volée
- **Bitmap Index scan/Bitmap heap scan**: génération et utilisation d'un index
- bitmap à la volée (voire suite)
- **Materialize** : écriture de résultats intermédiaires sur le disque

Optimisation des index non groupants

- **Creation d'un bitmap**
- (1) Scanner l'index et créer un bitmap de taille N
 - N est le nombre de pages de la relation.
 - Pour chaque entrée d'index satisfaisant le résultat, mettre le bit de la page pointée par l'entrée d'index à 1 (phase Bitmap Index Scan)
- (2) Parcourir la relation dans l'ordre du disque, page à page
 - Si la page est à 1 dans le bitmap, on la charge, sinon on l'ignore.
 - Une fois la page chargée il faut réévaluer la condition car on a oublié quels étaient les résultats de l'index (phase Bitmap Heap Scan)
- Intérêt : Un bitmap est petit (relation de 10 000 pages -> bitmap 10 000 bits = 1 250 octets (< 1 page !))
- Technique très efficace pour les conditions complexes