PDF Studio - PDF Editor for Mac, Windows, Linux. For Evaluation. https://www.qoppa.com/pdfstudio

SGBD : BASES DE DONNÉES AVANCÉES [M3106C]

TD $N^0$2 - OPTIMISATION DE REQUÊTES SOUS POSTGRESQL

PostgreSQL propose un plan de requête pour chaque requête. Choisir le bon plan pour correspondre à la structure de requête et aux propriétés des données est absolument essentiel pour de bonnes performances, de sorte que le système comprend un planificateur complexe qui essaie de sélectionner de bons plans. Vous pouvez utiliser la commande EXPLAIN pour voir quelle requête planifie le planificateur pour toute requête.

OBJECTIFS

- Utilisation de EXPLAIN
- Notions de *tuning* des requêtes
- Méthodes d'accès

CORRIGÉS

Exercice I :

Question 1.1.

```
# EXPLAIN SELECT dy_company, date(dy_timestamp) FROM diary;
                            QUERY PLAN
-------------------------------------------------------------
Seq Scan on diary  (cost=0.00..22.50 rows=1000 width=12)
(1 ligne)
```

temps passé avant le démarrage du scan

temps total passé

nombre de lignes obtenues en sortie

scan séquentiel sur la table DIARY

taille en octets de la sortie

Question 1.2.

```
# select reltuples, relpages
         from pg_class where relname='diary';
 reltuples | relpages
-----------+----------
      1000 |       10
(1 ligne)

# show cpu_tuple_cost;
 cpu_tuple_cost
----------------
 0.01
(1 ligne)

# show cpu_operator_cost;
 cpu_operator_cost
-------------------
```

nombre de blocs de la relation

nombre de tuples de la relation

coût CPU de traitement d'un tuple

Date: 20 mars 2014.

Hocine ABIR - IUT Villetaneuse .

PDF Studio - PDF Editor for Mac, Windows, Linux. For Evaluation. https://www.qoppa.com/pdfstudio


```
 0.0025
(1 ligne)

# select 10+1000*0.01+1000*0.0025;
 ?column?
----------
  22.5000
(1 ligne)
```

coût CPU d'une opération de la clause WHERE

Question 1.3.

avant de lancer les deux requêtes, il faut lancer :
SET enable_bitmapscan to OFF;

```
# EXPLAIN SELECT dy_company, date(dy_timestamp)
              FROM diary WHERE dy_id>=900;
                        QUERY PLAN
-------------------------------------------------------------
 Seq Scan on diary  (cost=0.00..23.33 rows=333 width=12)
   Filter: (dy_id >= 900)
(2 lignes)

# EXPLAIN SELECT dy_company, date(dy_timestamp)
              FROM diary WHERE dy_id>=10;
                        QUERY PLAN
-------------------------------------------------------------
 Seq Scan on diary  (cost=0.00..23.33 rows=333 width=12)
   Filter: (dy_id >= 10)
(2 lignes)
```

Si ça ne suffit pas :
SET enable_indexscan to OFF;

dans les deux cas on a fait un scan séquentiel

De plus, il a estimé le même nombre de lignes en sortie

Commentaire :

- Les estimations de ces deux requêtes sont identiques !
- La première a un faible taux de sélectivité(>=900) mais l'accès séquentiel `Seq Scan` est préféré à l'accès par l'index `Index Scan` !
- le nombre de tuples estimé pour le résultat est identique : bizarre

Pour les deux plans, l'optimiseur a estimé le meme nombre de tuple pour le resultat ! ! (333) . Ceci est du au fait qu'il n'a pas de statistiques sur la table `diary` (et les autres).

Pour remédier à ce problème :

```
-------------------------------------------------------
# VACUUM ANALYZE diary;
VACUUM
-------------------------------------------------------
```

Question 1.4.

notre table n'est en fait pas optimisée. Il faut lancer la commande VACUUM :
VACUUM récupère le stockage occupé par des tuples morts et met à jour le profil des données pour assister le planner. Dans l'opération PostgreSQL normale, les tuples qui sont supprimés ou obsolètes par une mise à jour ne sont pas retirés physiquement de leur table; ils restent présents jusqu'à ce qu'un VACUUM soit terminé. Par conséquent, il est nécessaire de faire VACUUM périodiquement, en particulier sur des tableaux fréquemment mis à jour.

PDF Studio - PDF Editor for Mac, Windows, Linux. For Evaluation. https://www.qoppa.com/pdfstudio


```
EXPLAIN SELECT dy_company, date(dy_timestamp)
FROM diary WHERE dy_id>=900;
                                QUERY PLAN
-----------------------------------------------------------
 Index Scan using diary_pkey on diary  \
              (cost=0.00..5.60 rows=100 width=12)
   Index Cond: (dy_id >= 900)
(2 lignes)

abir=> EXPLAIN SELECT dy_company, date(dy_timestamp)
FROM diary WHERE dy_id>=10;
                        QUERY PLAN
-----------------------------------------------------------
 Seq Scan on diary  (cost=0.00..24.98 rows=991 width=12)
   Filter: (dy_id >= 10)
(2 lignes)
```

- Le nombre de tuples estimés dans chaque requête est plus réaliste.
- L'accès par index Index Scan pour la sélectivite faible est honoré.

Question 1.5.

```
# set enable_indexscan to off;
SET

# EXPLAIN SELECT dy_company, date(dy_timestamp)
FROM diary WHERE dy_id>=900;
                        QUERY PLAN
-----------------------------------------------------------
 Seq Scan on diary  (cost=0.00..22.75 rows=100 width=12)
   Filter: (dy_id >= 900)
(2 lignes)
```

on n'utilise plus l'index et on fait donc un scan séquentiel

d'où une grosse différence dans les temps d'exécution

le temps a plus que triplé : l'index `diary_pkey` est bénéfique.

Question 1.6.

"Donner pour chaque compagnie, les noms des produits et les quantités commandées (si ces dernières sont) supérieures à 60 ? "

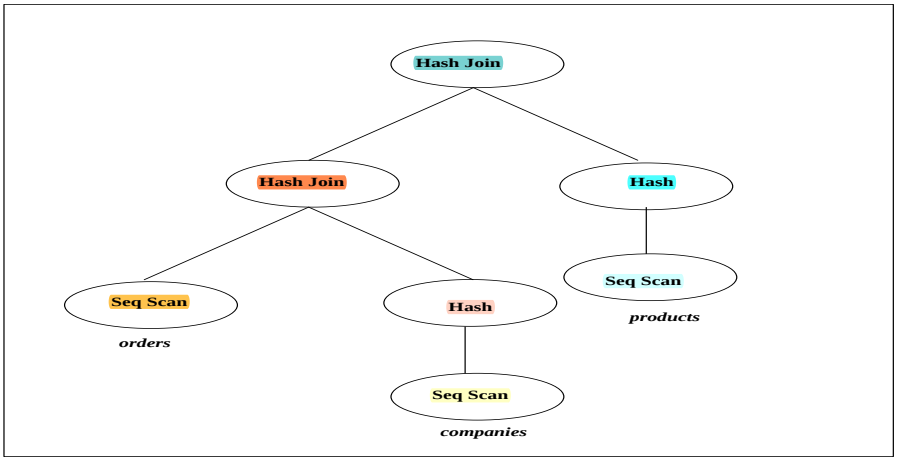
Question 1.7.

```
#  EXPLAIN SELECT co_name,pr_desc,ord_qty
              FROM companies,orders,products
              WHERE co_id=ord_company
```

PDF Studio - PDF Editor for Mac, Windows, Linux. For Evaluation. https://www.qoppa.com/pdfstudio


```
                AND pr_code=ord_product
                AND ord_qty>60;
                                        QUERY PLAN
-----------------------------------------------------------------------------
 Hash Join  (cost=4.38..36.16 rows=376 width=52)
   Hash Cond: (("outer".ord_product)::text = ("inner".pr_code)::text)
   ->  Hash Join  (cost=3.25..29.39 rows=376 width=39)
         Hash Cond: ("outer".ord_company = "inner".co_id)
         ->  Seq Scan on orders  (cost=0.00..20.50 rows=376 width=17)
               Filter: (ord_qty > 60)
         ->  Hash  (cost=3.00..3.00 rows=100 width=30)
               ->  Seq Scan on companies  (cost=0.00..3.00 rows=100 width=3
   ->  Hash  (cost=1.10..1.10 rows=10 width=31)
         ->  Seq Scan on products  (cost=0.00..1.10 rows=10 width=31)
(10 lignes)
```

Ici on colle la table companies sur la table orders, puis on y colle la table products

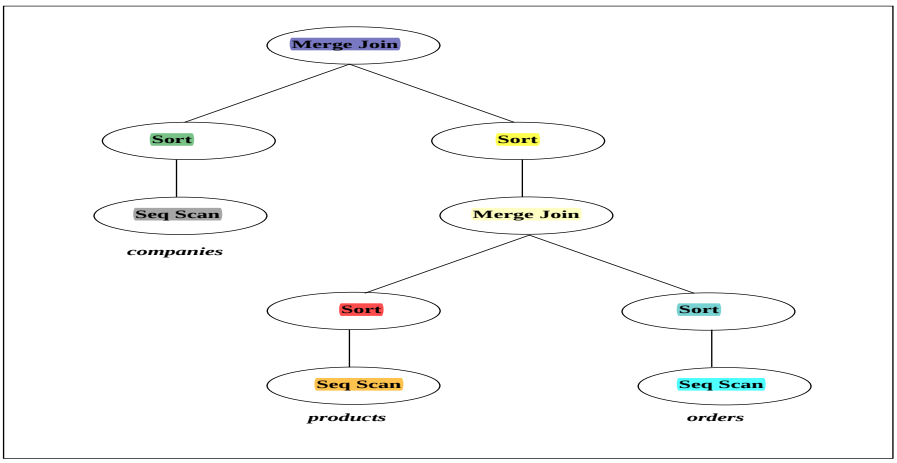



```
# set enable_hashjoin to off;
SET
# EXPLAIN SELECT co_name,pr_desc,ord_qty
                FROM companies,orders,products
                WHERE co_id=ord_company
                AND pr_code=ord_product
                AND ord_qty>60;
                                          QUERY PLAN
-----------------------------------------------------------------------------
 Merge Join  (cost=65.94..72.08 rows=376 width=52)
   Merge Cond: ("outer".co_id = "inner".ord_company)
   ->  Sort  (cost=6.32..6.57 rows=100 width=30)
         Sort Key: companies.co_id
         ->  Seq Scan on companies  (cost=0.00..3.00 rows=100 width=30)
```

PDF Studio - PDF Editor for Mac, Windows, Linux. For Evaluation. https://www.qoppa.com/pdfstudio


```
   ->  Sort  (cost=59.62..60.56 rows=376 width=30)
         Sort Key: orders.ord_company
         ->  Merge Join  (cost=37.85..43.54 rows=376 width=30)
               Merge Cond: ("outer"."?column3?" = "inner"."?column4?")
               ->  Sort  (cost=1.27..1.29 rows=10 width=31)
                     Sort Key: (products.pr_code)::text
                     ->  Seq Scan on products  (cost=0.00..1.10 rows=10 width
               ->  Sort  (cost=36.58..37.52 rows=376 width=17)
                     Sort Key: (orders.ord_product)::text
                     ->  Seq Scan on orders  (cost=0.00..20.50 rows=376 width
                           Filter: (ord_qty > 60)
(16 lignes)
```

ici on colle products et orders, on trie par ord_company et on colle le tout sur companies

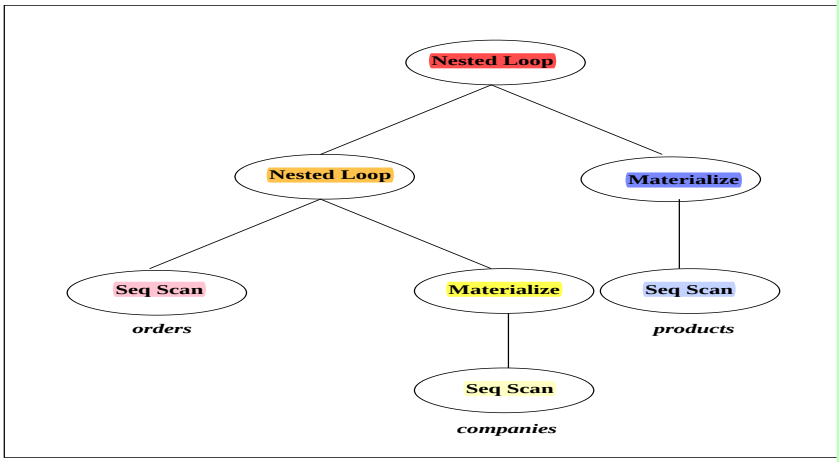



```
# set enable_mergejoin to off;
SET
 EXPLAIN SELECT co_name,pr_desc,ord_qty
              FROM companies,orders,products
              WHERE co_id=ord_company
              AND pr_code=ord_product
              AND ord_qty>60;
                                  QUERY PLAN
------------------------------------------------------------------------------
 Nested Loop  (cost=4.21..955.31 rows=376 width=52)
   Join Filter: (("inner".pr_code)::text = ("outer".ord_product)::text)
   ->  Nested Loop  (cost=3.10..869.60 rows=376 width=39)
         Join Filter: ("inner".co_id = "outer".ord_company)
         ->  Seq Scan on orders  (cost=0.00..20.50 rows=376 width=17)
               Filter: (ord_qty > 60)
         ->  Materialize  (cost=3.10..4.10 rows=100 width=30)
```

PDF Studio - PDF Editor for Mac, Windows, Linux. For Evaluation. https://www.qoppa.com/pdfstudio


```
                  ->  Seq Scan on companies  (cost=0.00..3.00 rows=100 width=3
   ->  Materialize  (cost=1.11..1.21 rows=10 width=31)
         ->  Seq Scan on products  (cost=0.00..1.10 rows=10 width=31)
(10 lignes)
```

ici on colle companies sur orders, et on colle products sur le tout



Un noeud MATERIALIZE signifie que la sortie de tout ce qui se trouve au-dessous de celle-ci dans l'arbre (qui peut être un balayage, ou un ensemble complet de jointures ou quelque chose comme ça) est mise en mémoire avant que le nœud supérieur ne soit exécuté. Cela se fait généralement lorsque le nœud externe a besoin d'une source qu'il peut réexaminer pour une raison ou une autre

Question 1.8.

```
# explain SELECT co_name,pr_desc,ord_qty
          FROM  orders join products on   (pr_code=ord_product ) ,
             companies
          where co_id=ord_company
              AND ord_qty>60;
                                  QUERY PLAN
-------------------------------------------------------------------------------
 Nested Loop  (cost=4.21..955.31 rows=376 width=52)
   Join Filter: ("inner".co_id = "outer".ord_company)
   ->  Nested Loop  (cost=1.11..106.21 rows=376 width=30)
         Join Filter: (("inner".pr_code)::text = ("outer".ord_product)::tex
         ->  Seq Scan on orders  (cost=0.00..20.50 rows=376 width=17)
               Filter: (ord_qty > 60)
         ->  Materialize  (cost=1.11..1.21 rows=10 width=31)
               ->  Seq Scan on products  (cost=0.00..1.10 rows=10 width=31)
   ->  Materialize  (cost=3.10..4.10 rows=100 width=30)
         ->  Seq Scan on companies  (cost=0.00..3.00 rows=100 width=30)
(10 rows)
```

ici on colle products sur orders, puis on colle companies sur le tout

Exercice II :

Question 2.1.

PDF Studio - PDF Editor for Mac, Windows, Linux. For Evaluation. https://www.qoppa.com/pdfstudio


```
select fou_nom,pie_nom from
   fournisseur join piece on fou_id=fou_id_fk;
```

Question 2.2.

```
select fou_nom,pie_nom
 from piece join fournisseur on fou_id=fou_id_fk;
```

on regarde les pièces, on prend le résultat, on regarde les fournisseurs et on colle pièce sur fournisseur. Ensuite on liste les fournisseurs

Question 2.3.

```
select fou_nom,pie_nom
 from  fournisseur join piece on fou_id=fou_id_fk;
 fou_nom | pie_nom
---------+---------
 Peugeot | Huile
 Renault | Pneu
(2 rows)
```

Question 2.4.

```
select fou_nom,pie_nom
 from   piece join fournisseur on fou_id=fou_id_fk;
 fou_nom | pie_nom
---------+---------
 Renault | Pneu
 Peugeot | Huile
(2 rows)
```

On regarde les fournisseurs, on prend le résultat, on regarde les pièces et on colle fournisseur sur pièce. Ensuite on liste les pièces

Question 2.5.

```
C1 : rows=1000   <--->   C2 : 2 rows
```

Rows = 1000 dans function scan mais il n'y a que 2 lignes en sortie

Question 2.6.

```
Select * from piece;
 pie_id |    pie_nom    | pie_stock | pie_prix | fou_id_fk
--------+---------------+-----------+----------+-----------
      1 | Pneu          |        11 |       59 |        20
      2 | Huile         |        20 |       18 |        10
      3 | Pneu Michelin |        10 |       90 |        20
      4 | Pneu Michelin |        10 |       90 |        20
      5 | Pneu Keber    |        10 |       90 |        10
      6 | Pneu Keber    |        10 |       90 |        10
(6 rows)
```

Question 2.7.

PDF Studio - PDF Editor for Mac, Windows, Linux. For Evaluation. https://www.qoppa.com/pdfstudio


```
# select reltuples from pg_class
   where relname='piece';
 reltuples
-----------
         2
(1 row)
```

Les statistiques ne sont pas mises à jour !

```
# vacuum piece;
VACUUM

# select reltuples from pg_class
   where relname='piece';
 reltuples
-----------
         6
(1 row)
```