

Optimisation de requêtes : exercices de TD

1. Préambule

Le schéma comprend les tables Commune, Departement et Region. Le schéma relationnel est indiqué de manière succincte.

COMMUNE (CODEINSEE, CODEPOSTAL, NUMDEP, NOMCOMMAJ, NOMCOMMIN, LATITUDE, LONGITUDE)
DEPARTEMENT (NUMDEP, CHEFLIEU, NUMREG, NOMDEPMAJ, NOMDEPMIN)
REGION (NUMREG, CHEFLIEU, NOMREGMAJ)

2. Exercices sur papier

Vous répondrez aux questions suivantes :

1. Exercice 1 : Vous écrirez en algèbre relationnelle, et de différentes manières, la requête qui renvoie le nom de la commune (en majuscules), ses latitude, longitude et numéro de département, quand le nom de la commune commence par un M et qu'elle se situe dans les départements 30 (Gard) ou 34 (Hérault). Vous proposerez ces requêtes également dans le langage SQL. Vous indiquerez en le justifiant la forme qui vous semble la plus coûteuse (notez que seule la table COMMUNE est exploitée dans le requêtage).
2. Exercice 2 :
 - (a) Vous écrirez en algèbre relationnelle et de différentes manières, la requête qui renvoie le nom de la commune (en majuscules), et le numéro de département, quand cette commune se situe dans le département de l'AVEYRON. Vous indiquerez en le justifiant la forme algébrique qui vous semble la moins coûteuse.
 - (b) Cette requête peut être traitée sous la forme d'une semi-jointure. Donnez l'écriture algébrique de cette semi-jointure et proposez deux façons de l'exprimer en SQL (prédicats EXISTS et IN)
3. Exercice 3 :
 - (a) Vous définirez plusieurs arbres algébriques pour la requête qui renvoie le nom d'une commune, sa latitude et sa longitude dès lors que cette commune voit son nom commencer par un M et que cette commune est une commune de la région OCCITANIE. Vous justifierez le choix d'un plan d'exécution qui vous semble proche de celui qui pourrait être choisi par l'optimiseur.
 - (b) La requête proposée est une requête qui peut s'exprimer en SQL au travers de jointures naturelles entre Commune, Departement et Region. Vous écrirez donc cette requête en SQL. Elle peut également être traitée comme une semi-jointure (seuls des attributs de Commune sont retournés). Vous écrirez en conséquence cette même requête sous une nouvelle forme algébrique et de deux nouvelles façons en SQL : recours au prédicat EXISTS (test de vacuité) et au prédicat IN (test d'appartenance) avec du "select" imbriqué.

4. Exercice 4 : un plan d'exécution est donné. Il a été obtenu après définition d'un index non unique sur l'attribut numdep de la table Commune. Les attributs projetés sont codeinsee, nomcommaj, numdep (de la table Departement), nomdepmaj, numreg.

```
Plan d'exécution
-----
Plan hash value: 3625588267

-----
| Id | Operation | Name | Rows | Bytes | Cost (%CPU) | Time |
-----+-----+-----+-----+-----+-----+-----+
| 0 | SELECT STATEMENT | | 503 | 20623 | 100 (1) | 00:00:01 |
| 1 | SORT ORDER BY | | 503 | 20623 | 100 (1) | 00:00:01 |
| 2 | NESTED LOOPS | | 503 | 20623 | 99 (0) | 00:00:01 |
| 3 | NESTED LOOPS | | 503 | 20623 | 99 (0) | 00:00:01 |
| 4 | TABLE ACCESS BY INDEX ROWID BATCHED | DEPARTEMENT | 12 | 216 | 2 (0) | 00:00:01 |
|* 5 | INDEX RANGE SCAN | DEPARTEMENTPK | 12 | | 1 (0) | 00:00:01 |
|* 6 | INDEX RANGE SCAN | IDX_DEP | 41 | | 1 (0) | 00:00:01 |
| 7 | TABLE ACCESS BY INDEX ROWID | COMMUNE | 41 | 943 | 8 (0) | 00:00:01 |
-----

Predicate Information (identified by operation id):
-----

5 - access("D"."NUMDEP" LIKE '3%')
   filter("D"."NUMDEP" LIKE '3%')
6 - access("D"."NUMDEP"="C"."NUMDEP")
   filter("C"."NUMDEP" LIKE '3%')

Note
-----
- this is an adaptive plan

Statistiques
-----
0 recursive calls
0 db block gets
1256 consistent gets
0 physical reads
0 redo size
164360 bytes sent via SQL*Net to client
3537 bytes received via SQL*Net from client
280 SQL*Net roundtrips to/from client
1 sorts (memory)
0 sorts (disk)
4171 rows processed
```

FIGURE 1 – Plan d'exécution

```
create index idx_dep on commune (numdep);
```

- Vous dessinerez le plan algébrique associé.
- Quels sont les index mobilisés ?
- Quel est l'opérateur physique de jointure choisi par l'optimiseur ? Discutez le choix de cet opérateur. Est ce que les estimations sur le nombre de tuples retournés sont correctes ?
- Donner la sémantique naturelle de la requête SQL qui a donné lieu à ce plan
- Proposer la requête SQL qui vous semble à l'origine de ce plan.
- Proposer une écriture de la requête intégrant des directives permettant de changer d'opérateur, et de table guidant la jointure.