

Optimisation de requêtes : exercices de TD

1. Préambule

Le schéma comprend les relations Commune, Departement et Region qui peuvent être créées à partir du schéma P00000009432. Tous les exercices sont à faire à la fois sur papier et sur machine.

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 ou 34. Vous proposerez ces requêtes également dans le langage SQL. Vous indiquerez en le justifiant la forme qui vous semble la plus coûteuse.
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.
 - (c) Vous écrirez également la requête SQL via une double jointure mais en spécifiant ces jointures via des INNER JOIN dans la clause FROM. Vous regarderez si cette syntaxe légèrement différente modifie le plan d'exécution choisi par l'optimiseur.

- (d) Vous créerez une vue à partir de la requête exprimée avec des jointures et vous consulterez cette vue (renvoyez les noms de commune depuis la vue). Quelle est la stratégie suivie par le système ?

4. Exercice 4 :

- (a) Vous écrirez en algèbre relationnelle et deux différentes manières (avec une intersection pour l'une des manières), la requête qui renvoie le nom de la commune (en majuscules), et le code insee de cette commune, quand cette commune est à la fois chef-lieu de département et de région.

3. Exercices sur machine

Reprenez les exercices et testez tous les plans d'exécution à partir de l'interpréteur sqlplus d'Oracle. Exploitez à cet effet les commandes :

1. set autotrace traceonly
2. explain plan for select ...

Utilisez également les directives pour guider/forcer le plan d'exécution et ainsi évaluer le coût de différentes requêtes conduisant aux mêmes résultats. Des exemples d'utilisation de directives sont données dans le cours.