IUT2 Université Grenoble Alpes
Département INFO

R3.07

SQL dans les langages de programmation

Optimisation des requêtes

Dans ce TP, nous travaillons sur la base de données des villes, régions et départements de France (avant la fusion des régions). Cette base de données comporte un grand nombre de données : 36684 villes, 26 régions et 100 départements !

L'objectif de ce TP est d'analyser les plans d'exécution des requêtes générés par l'optimiseur de Postgres et de mettre en place des mécanismes permettant d'améliorer le coût d'exécution des requêtes dans des bases de données de grades dimensions.
Le schéma de cette base de données est le suivant :

TOWNS (id, code, article, name, department)
REGIONS(id, code, capital, name)
DEPARTMENTS(id, code, capital, region, name)

Préambule : Création de la base de données et insertion des données

- Lancez un terminal (*Terminal ou Konsole*).
- Dans votre répertoire R3.07, créez un répertoire FrenchTowns.
- Dans ce sous-répertoire FrenchTowns, copier le fichier ***create.sql*** disponible dans */users/info/pub/2a/R3.07/FrenchTowns.* Examinez ce fichier un peu différent des create.sql habituels…
- Dans le terminal, connectez-vous sur Postgres par la commande
  psql –h postgres-info –U user base
  (où user et base sont vos logins)
- Supprimez toutes les tables qui ont été créées lors des TPs précédents afin d'obtenir une base vierge.
- Exécutez la commande ***\i create.sql*** pour créer les tables de la base de données et y insérer les données.
- Vérifiez que les tables TOWNS, REGIONS et DEPARTMENTS ont été créées par la commande ***\d.*** Vous remarquerez que des tables de type « SEQUENCE » ont également été créées. Une séquence est une table contenant une seule ligne appelée générateur de séquence, utilisée pour générer des identifiants uniques de lignes de tables (ici les identifiants des villes, régions et départements). Ces tables séquence sont des tables utilitaires. **Vous remarquerez également que des index (cf cours) ont été créés automatiquement par Postgres pour tous les attributs UNIQUE, ceci afin d'optimiser le temps d'exécution des requêtes.**

- Pour répondre à certaines questions de ce TP, vous devrez effectuer certaines recherches sur la doc en ligne de Postgres : http://docs.postgresqlfr.org/

Exercice 1 : Etude des statistiques du schéma

Tapez la commande suivante :

```
SELECT relname, relpages, reltuples
FROM pg_class
ORDER BY relname ;
```

Question 1 : Que fait cette commande ?

Consultez la doc en ligne de postgres pour trouver ce que représente :

- pg_class ?
- relpages ?
- reltuples ?

Notez les résultats obtenus pour les relnames suivants : *departments, departments_capital_key, departments_code_key, departments_id_key, departments_id_seq, departments_name_key, towns, towns_id_key, towns_id_seq, towns_code_department_key, regions, regions_code_key, regions_id_key, regions_id_seq, regions_name_key*.

Question 2 :

Afin que l'estimation du temps d'exécution d'une requête par Postgres soit le plus exact possible, il est nécessaire de mettre à jour les statistiques des tables.
Pour cela, exécutez la commande **ANALYZE** (que fait cette commande ?)

Exercice 2 : Etude des plans d'exécution de différents types de requêtes

Dans cet exercice, nous utiliserons la commande EXPLAIN qui donne le plan d'exécution et calcule les coûts d'exécution prévus par l'optimiseur de Postgres dans le cadre d'une requête.

Pour mieux comprendre :

- https://www.bortzmeyer.org/explain-postgresql.html
- https://www.postgresql.org/docs/current/sql-explain.html

Question 1 : Tapez la commande suivante :

```
EXPLAIN SELECT * FROM TOWNS ;
```

Qu'indiquent les différents coûts générés par cette commande ?
Quel est le plan d'exécution généré dans ce cas ?

Tapez ensuite EXPLAIN ANALYZE SELECT * FROM TOWNS ; et comparez …

Question 2 :

Examinez les plans d'exécution des requêtes suivantes (vous devrez préalablement écrire la requête SQL…). **Notez vos observations**.

1. Code et noms des villes de France

2. Noms, départements et régions de toutes les villes de France. Effectuer une recherche sur internet pour comprendre ce que représentent « hash » (aussi appelé « hash join ») et « hash cond »

3. Nombre de villes de France. Comparer avec la question 1.

4. Nombre de villes par département. Effectuer une recherche sur internet pour comprendre ce que représente « hashAggregate »

5. Nombre de départements par région

6. Nombre de départements par région, pour les régions comportant plus de 5 départements.

Exercice 3 : Index et plan d'exécution

On rappelle la commande de création d'un index :

```
CREATE [UNIQUE] INDEX nom_index ON nom_table(liste_attributs) ;
```

Question 1 :

Examinez le plan d'exécution de la requête

```
SELECT * FROM TOWNS where name = 'Grenoble' ;
```

(noter les résultats)

Un index a-t-il été créé par Posgres sur l'attribut name de la relation TOWNS ? Pourquoi ?

Créez un index sur l'attribut name de TOWNS, puis ré-examinez le plan d'exécution de la requête ci-dessus.

Que notez-vous ?

Question 2 :

Cette question est basée sur la requête suivante :

```
SELECT towns.code, towns.name
FROM   towns, departments
WHERE  towns.department = departments.code
  AND  departments.name = 'Isère' ;
```

Exécutez cette requête et vérifiez les résultats.

Comparez les plans d'exécution générés par l'optimiseur de Postgres dans les cas suivants :

- Cas par défaut, avec les index générés par postgres

- Sans index (vous devrez préalablement supprimer tous les index générés par Postgres par la commande DROP INDEX nomIndex ou par la commande ALTER TABLE nomtable DROP CONSTRAINT nomcontrainte).
- Index sur departments.name (vous devrez donc re-créer un nouvel index)
- Index sur departments.code (vous devrez donc re-créer un nouvel index)
- Index sur departments.code et towns.department (vous devrez donc re-créer deux nouveaux index)
- Index sur departments.code, towns.department et departments.name vous devrez donc re-créer trois nouveaux index)

Exercice 4 : Etude des plans d'exécution et des temps d'exécution de requêtes avec jointure

Soit une requête permettant d'obtenir les noms et départements des villes de la région Rhône-Alpes.

1. Proposez une version de cette requête avec uniquement des égalités entre attributs de tables. Regardez le plan d'exécution de la requête (utilisez EXPLAIN). Exécutez la requête plusieurs fois et notez les temps d'exécution (utilisez ANALYZE).

2. Ecrivez une requête similaire à la requête de la question 1 avec des requêtes imbriquées (IN). Regardez le plan d'exécution de la requête suivante. Exécutez la requête plusieurs fois et notez les temps d'exécution.

3. Ecrivez une requête similaire à la requête de la question 1 avec des jointures join. Regardez le plan d'exécution de la requête suivante. Exécutez la requête plusieurs fois et notez les temps d'exécution.

4. Comparez les 3 exécutions. Laquelle est la plus efficace ? Laquelle est la moins efficace ?