

TP: Vues

Pierre Senellart (pierre.senellart@ens.fr)

26 avril 2017

Le but de ce TP est de manipuler les vues, et de mettre en place dans PostgreSQL des stratégies simples de maintenance de vues et de mises à jour au travers de vues.

Rendu

Le rendu de ce TP doit être fait par mél, avant 23:59 le vendredi 28 avril, à pierre.senellart@ens.fr en fournissant, en attachement du mél, un fichier texte ou PDF contenant un compte-rendu de vos réponses aux questions (y compris les ordres SQL utilisés). Deux jours supplémentaires sont accordés par rapport à d'habitude en raison d'un conflit d'emploi du temps affectant certains étudiants. Des pénalités seront comptées en cas de rendu tardif (0,5 point par heure de retard).

1 Vues virtuelles et matérialisées

- 1a. Récupérer sur le site du cours le corrigé du tout premier TP. Utiliser les ordres qui y figurent pour créer les trois tables du schéma.
- 1b. Remplir ces trois tables avec un nombre suffisant de valeurs, pour que la dernière requête du fichier (calculant l'ensemble des moyennes des étudiants) prenne un temps non négligeable (de l'ordre de la seconde). Ne pas le faire à la main, mais réfléchir à une stratégie de génération automatique ou semi-automatique !
- 1c. En utilisant `\timing`, relever le temps d'exécution de la requête calculant les moyennes.
- 1d. Créer une vue virtuelle `averagev` et une vue matérialisée `averagem` pour cette requête.
- 1e. Comparer le temps d'exécution d'un `SELECT *` sur ces deux vues.
- 1f. Créer une vue virtuelle `studentjv` et une vue matérialisée `studentjm` pour l'ensemble des identifiants, prénoms, noms d'étudiants dont le prénom commence par un « J ». (On pourra utiliser l'opérateur `LIKE` dont on consultera la documentation.)
- 1g. Ajouter un nouvel étudiant dont le prénom commence par un « J » et comparer le contenu des deux vues `studentjv` et `studentjm`. Utiliser `REFRESH` si nécessaire.
- 1h. Tenter d'ajouter un étudiant dont le prénom commence par un « J » au travers des deux vues `studentjv` et `studentjm`. Comparer.
- 1i. Tenter d'ajouter un étudiant dont le prénom ne commence pas par un « J » au travers des deux vues `studentjv` et `studentjm`. Comparer.
- 1j. Tenter de modifier le nom d'un étudiant dont le prénom commence par un « J » au travers des deux vues `studentjv` et `studentjm`. Comparer.
- 1k. Tenter de supprimer un étudiant dont le prénom commence par un « J » au travers des deux vues `studentjv` et `studentjm`. Comparer.
- 1l. Tenter de supprimer un étudiant au travers de la vue `averagev`. Commenter.

2 Maintenance de vue

On va maintenant faire à la main la maintenance de vues matérialisées, opération non supportée dans PostgreSQL.

- 2a. Remplacer la vue `studentjm` par une table `studentjm`. Définir des déclencheurs sur la table `student` pour que les insertions, suppressions, modifications d'étudiants soient correctement reflétés dans la table `studentjm`, en étant aussi économe en opérations que possible.
- 2b. (*Optionnel. À traiter après la dernière partie du TP, s'il vous reste du temps...*) Remplacer la vue `averagem` par une table `averagem`. Définir des déclencheurs sur les trois tables initiales pour que les insertions, suppressions, modifications soient correctement reflétées dans la table `averagem`, en étant aussi économe en opérations que possible.

3 Mise à jour au travers de vues

Définir des déclencheurs sur la vue `averagev` pour permettre :

- la suppression d'étudiants au travers de la vue ;
- la modification du prénom ou du nom d'un étudiant au travers de la vue.

Toutes les autres opérations doivent être rejetées.