

Requêtes récursives

Pierre Senellart (pierre.senellart@ens.fr)

22 février 2017

Le but de ce TP est de se familiariser avec les requêtes récursives en PostgreSQL, en les utilisant pour explorer un jeu de données réel, celui des transports en commun en Île-de-France.

Rendu

Le rendu de ce TP doit être fait par mél, avant 23:59 le 22 février, à pierre.senellart@ens.fr en fournissant, en attachement du mél, un fichier *nom.sql* (ou *nom* doit être remplacé par votre nom de famille) contenant les requêtes permettant de répondre aux questions 1 à 6, séparées par une ligne vide, sans autre contenu. Le TP étant évalué semi-automatiquement, il est important de respecter les consignes sur le nommage des tables et ce que doit renvoyer chacune des requêtes. Des pénalités seront comptées en cas de rendu tardif (0,5 point par heure de retard), ou de non-respect des consignes.

1 Récupération et chargement du jeu de données

Nous travaillerons avec les données fournis par le Syndicat des Transports en Île-de-France (STIF), au format GTFS, téléchargeables sous la forme d'un fichier ZIP depuis <https://opendata.stif.info/explore/dataset/offre-horaires-tc-gtfs-idf/table/>.

- Récupérer ces données et étudier leur contenu. Vous trouverez une description du format GTFS sur <https://developers.google.com/transit/gtfs/reference/>. Pour traiter ce TP, nous aurons besoins des fichiers *routes*, *stop_times*, *stops* et *trips* uniquement.
- Créer quatre tables au sein de PostgreSQL permettant d'héberger ces données. Ces tables doivent avoir pour nom le nom du fichier correspondant (sans le *.txt*), et comme noms d'attributs les noms de colonnes figurant en première ligne du fichier. Pensez à proposer un type adéquat pour chaque colonne, et à spécifier les clefs primaires, clefs étrangères et éventuelles colonnes uniques (cela aura un impact important sur la performance des requêtes).
- À l'aide de la commande `\copy` du client en ligne de commande de PostgreSQL, charger le contenu de chacun des quatre fichiers dans la table correspondante.

2 Requêtes sur la base

Pour chacune des questions suivantes, proposer une requête SQL qui renvoie les informations demandées; ne pas hésiter à construire les requêtes petit à petit pour vérifier qu'elles renvoient des informations cohérentes. Pour vous aider, il est indiqué après chaque question le nombre de résultats que cette requête est censée renvoyer. Des requêtes récursives sont demandées pour les trois dernières questions.

1. Quels sont les arrêts de transport en commun qui se trouvent au sein de la zone « GARE DE BAGNEUX »? Pour chaque arrêt, renvoyer son identifiant et son nom. Les résultats doivent être triés par nom. (*5 résultats*)

2. Quels sont les lignes de transport en commun accessibles depuis la zone « GARE DE BAGNEUX » ? Pour chaque ligne, renvoyer le nom long de la ligne. Les résultats doivent être triés. *(6 résultats)*
3. Quels sont les arrêts accessibles en prenant un transport en commun depuis la zone « GARE DE BAGNEUX », sans changement ? Renvoyer à la fois le nom de l'arrêt et le nom long de la ligne de transport permettant d'y accéder. Les résultats doivent être triés par nom de ligne, puis par nom d'arrêt. Ne pas oublier que certains arrêts ne sont accessibles que dans des sens de trajets incompatibles avec un départ GARE DE BAGNEUX. *(205 résultats)*
4. Quelles sont les gares de train (banlieue ou RER) accessibles en prenant au plus deux trains successifs depuis la GARE DE BAGNEUX ? On ne s'intéresse pas aux autres modes de transport. Renvoyer pour chaque résultat le nom de la gare (tel qu'indiqué comme StopArea, pas comme StopPoint) ainsi que le nombre minimum de trains successifs nécessaires pour accéder à cette gare. Les résultats doivent être triés par distance minimum, puis par nom de gare. *(268 résultats)*
5. Quelles sont les gares de train (banlieue ou RER) accessibles en prenant au plus deux trains successifs depuis la GARE DE BAGNEUX ? On ne s'intéresse pas aux autres modes de transport. Renvoyer pour chaque résultat le nom de la gare (tel qu'indiqué comme StopArea, pas comme StopPoint) et les séquences possibles de lignes de train utilisables pour y accéder. On pourra représenter une séquence de lignes de train par la concaténation des noms longs de la ligne, séparés par des espaces, et utiliser la fonction d'agrégation `array_agg` pour agréger les séquences possibles pour un même arrêt. On interdira les séquences du type « B B ». Les résultats doivent être triés par nom de gare. *(268 résultats)*
6. Quels sont les arrêts de transport en commun accessibles par le trajet d'un véhicule de transport en commun s'arrêtant au plus 3 fois entre le point de départ et la destination (incluse) depuis la zone de la GARE DE BAGNEUX ? Il est par exemple possible d'atteindre Denfert-Rochereau par un RER B qui ne s'arrête qu'à Arcueil-Cachan et à Cité Universitaire. On autorise les changements de ligne, et on peut utiliser différents modes de transport en commun. Renvoyer pour chaque résultat le nom d'arrêt (tel qu'indiqué comme StopArea) et le nombre d'arrêts sur le chemin (destination incluse, p. ex., 3 pour DENFERT ROCHEREAU). Les résultats doivent être triés par distance, puis par nom d'arrêt. *(76 résultats)*