

TP PostgreSQL n° 2

– Prise en main (suite) –

Il est conseillé de travailler le plus possible avec des fichiers sql plutôt qu'avec l'éditeur de PostgreSQL .

PostgreSQL et psql

Connectez-vous à la base de données (tp_intro ou autre bdd déjà existante).

Commandes de base. Les principales commandes de psql sont les suivantes :

- \? : Liste des commandes psql,
- \h : liste des instructions SQL,
- \h <une_instruction> : description de l'instruction SQL <une_instruction>.

Essayez avec l'instruction CREATE TABLE,

- \d : liste des tables (ou relations, cf. partie 2),
- \d <nom_table> : description de la table <nom_table>,
- \i <chemin/nom_fichier_script.sql> : exécution d'un fichier de script SQL,
- \o <chemin/nom_fichier_resultat.sql> : écriture des résultats des instructions dans le fichier passé en paramètre,
- \o : retour à un affichage à l'écran,.
- \q : quitter psql.

Retenez ces commandes, elles seront indispensables pour les TP.

1. Création de tables

Créez les tables correspondant aux relations suivantes. La détermination des types de données est laissée à votre appréciation, mais on considèrera que les heures de départ et d'arrivée sont sans les minutes). N'oubliez pas de créer les clés primaires (souligné) et étrangères (*italique*).

Ville (numville, *nomville*, *codepostal*)
Train (numtrain, *villedepart*, *villearrivee*, *heuredepart*, *heurearrivee*, *jour*)
Passager (numpassager, *nompas*, *age*, *reftrain*)

2. Création des données

Créez un ensemble de données correspondant au schéma précédent (une dizaine de villes, une quinzaine de trains, et une trentaine de passagers). Pour créer les valeurs de la colonne *jour* il faut utiliser la fonction `TO_DATE('val', 'fmt')` où *fmt* indique le format de la date tel que considéré dans *val*. Dans notre cas *fmt* vaudra : `'dd/mm/YYYY'`.

3. Modification de tables et de données

Videz les tables `Ville`, `Train`, `Passager` (instruction `DELETE`) et chargez le fichier
`/users/prof/fernandes/data/data01.sql`

On étend la relation `Passager` en lui rajoutant une colonne :

`Passager` (`numpassager`, `nompas`, `age`, `reftrain`, `categorie`)

Répercuter cette modification sur la table `Passager`.

Répercuter cette modification sur les données en considérant les contraintes suivantes :

- catégorie 2 pour les moins de 18 ans et les plus de 60 ans
- catégorie 1 pour tous les noms commençant par 'R' ou 'C'
- catégorie 3 pour les passagers restant (c'est-à-dire avec une catégorie valant NULL)

4. Ajout de tables

Ajoutez les tables correspondant aux relations suivantes : outre les contraintes de clés, certaines contraintes d'intégrités peuvent être écrites via la clause `CONSTRAINT`, par exemple le fait que la distance entre deux villes ne peut être négative. Spécifiez certaines contraintes sur ces tables et modifiez les tables précédentes pour leur ajouter de telles contraintes d'intégrités.

```
Tarif (reftrain, categorie, prix)  
Distance (ville1, ville2, nbkm)
```

Insérez dans la base de données quelques nouveaux tuples pour instancier ces deux tables et observez le comportement des tuples violant les contraintes d'intégrités. Effacez vos tuples de tests et utilisez les données du fichier `/users/prof/fernandes/data/data02.sql`.

Ajoutez dans la base de données la table correspondant à la relation :

```
Categorie (numcat, intitule)
```

ainsi que les tuples suivants :

```
(1, Privilege du nom)  
(2, Privilege Jeunes et Seniors)  
(3, Aucun privilege)
```

Comment répercuter intelligemment cet ajout de table sur les définitions des tables `Passager` et `Tarif` ?

5. Premières requêtes

Ecrivez les requêtes SQL répondant aux questions suivantes directement sous `sqlplus`.

1. Lister tous les trains.
2. Lister les noms des villes dont le code postal commence par 67.
3. Lister tous les trains partant avant midi.
4. Lister toutes les distances entre villes.
5. Lister toutes les distances entre villes de plus de 60km.
6. Lister le numéro et la durée de tous les trains qui roulent pendant plus de 5 heures.
7. Lister le jour de circulation des trains partant avant 10h.
8. Lister l'intitulé de toutes les catégories.