

Author : Christophe Garion <christophe.garion@isae-superaero.fr>
Public : SUPAERO 3A SD/ARO
Date :

Résumé

L'objectif de ce TP est de manipuler le langage SQL au travers du SDGB PostgreSQL.

1. Connexion au serveur PostgreSQL

Le serveur PostgreSQL [2] que nous allons utiliser dans le cours est sur la machine db-cours. Vous pouvez vous y connecter en utilisant via pgAdmin3 [1] en utilisant comme login votre login et votre mot de passe habituels (cf. annexe du polycopié de cours).

Vous disposerez alors de plusieurs bases de données :

- une base db-login où login est votre login pour votre travail ;
- des bases correspondant aux bases de données des exercices suivants. Vous ne pouvez pas modifier ces bases mais vous pouvez les interroger.

2. La coupe du monde des vieux

La base de données db-mexico86 contient les résultats de la coupe du monde de football 1986. Elle est composée de trois tables :

Pays (nom, poule)
Typematch (type)
Match (paysl, paysv, butsl, butsv, type, date)

- ouvrir la base de données avec pgAdmin3 et explorer les données.
- écrire une requête qui liste les pays participant à la coupe du monde.
- écrire une requête qui liste les rencontres de la coupe du monde (couples des deux pays par rencontre).
- écrire une requête qui liste les rencontres s'étant déroulées le 5 juin 1986.
- écrire une requête qui liste les pays que la France a rencontré.
- écrire une requête qui donne le vainqueur de la coupe du monde.

3. Brune, blonde ou ambrée

On considère une base de données de buveurs de bière constituée de trois relations :

Frequente (buveur, bar)
Sert (bar, biere)
Aime (buveur, biere)

Écrire les requêtes SQL permettant de répondre aux questions suivantes :

- quelle est la liste des bars qui servent une bière que Martin aime ?
- quelle est la liste des buveurs qui fréquentent au moins un bar qui sert une bière qu'ils aiment ?
- quelle est la liste des buveurs qui fréquentent tous les bars qui servent uniquement de la bière qu'ils aiment (on suppose que chaque buveur aime au moins une bière et fréquente au moins un bar) ?
- quelle est la liste des buveurs qui ne fréquentent aucun bar qui sert une bière qu'ils aiment ?

4. La coupe du monde des vieux (avec des agrégats et des fonctions)

Reprenons le travail sur la base de données db-mexico86.

- dans un premier temps, utiliser les scripts disponibles en téléchargement sur le site du cours pour recréer les tables et tuples de la base de données dans votre base de données personnelle (il faut d'abord se connecter sur cette base).
- écrire une requête qui liste le nombre de buts marqués pour chaque rencontre (on aura également comme attributs la date et le type du match). Comme cette requête servira dans la suite, on créera une vue matchbutsglobal à partir de cette requête.

- (c) écrire une requête qui calcule le nombre de buts marqués en moyenne dans les matchs joués par l'équipe de France.
- (d) écrire une requête qui calcule le nombre total de buts marqués par l'équipe de France.
- (e) écrire une requête donnant le nombre total de buts marqués dans chaque poule. On ordonnera le résultat par groupe.
- (f) créer une fonction retournant le vainqueur d'un match (identifié par les deux pays et le type du match).
- (g) créer une fonction permettant de trouver le nombre de buts total et la moyenne de buts marqués par une équipe (on renverra une table avec deux attributs, le premier correspondant au total et le deuxième à la moyenne). Comment faire pour ajouter le nom de l'équipe comme premier attribut de la table renvoyée par la fonction ?
- (h) en utilisant la fonction précédente, écrire une requête qui liste les pays participant à la coupe du monde et le nombre de buts qu'ils ont inscrit. En déduire le pays qui a marqué le plus de buts

5. Pull the trigger

Dans cet exercice, nous allons nous intéresser à la création d'un *trigger* sur une base de données très simple. Rappelons qu'un *trigger* est un mécanisme permettant d'exécuter des actions lors de l'arrivée d'un événement particulier (mise-à-jour d'une relation, insertion d'un nouveau tuple etc.).

- (a) créer une table `rel(nom, valeur)` où `nom` est une chaîne de caractères et `valeur` un entier. On choisira `nom` comme clé primaire.
- (b) insérer 5 tuples dans la table.
- (c) on souhaite maintenant que lors de l'insertion de nouveaux tuples la moyenne actuelle des valeurs contenues dans la relation ne puisse pas décroître.

Références




- [1] THE PGADMIN DEVELOPMENT TEAM. *pgAdmin*. 2015. URL : <http://pgadmin.org>.
- [2] THE POSTGRESQL GLOBAL DEVELOPMENT GROUP. *PostgreSQL*. 2015. URL : <http://www.postgresql.org>.

License CC BY-NC-SA 3.0



This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported license (CC BY-NC-SA 3.0)

You are free to Share (copy, distribute and transmute) and to Remix (adapt) this work under the following conditions:

-  **Attribution** – You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
-  **Noncommercial** – You may not use this work for commercial purposes.
-  **Share Alike** – If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

See <http://creativecommons.org/licenses/by-nc-sa/3.0/> for more details.