Introduction au modèle relationnel et à PostgreSQL

Pierre Senellart (pierre.senellart@ens.fr)

1er février 2017

Le but de ce TP est de se familiariser avec l'utilisation d'un système de gestion de base de données (en l'occurrence PostgreSQL) et de SQL. Ce TP n'est pas évalué, aucun rendu n'est attendu.

1 Modélisation

On veut représenter les données suivantes. On dispose d'une liste d'étudiants (avec leurs noms et prénoms) et d'une liste d'examens avec pour chacun leur type (p. ex., écrit, oral, devoir à la maison, projet), leur date, un titre éventuel, un coefficient, et une note maximale (par exemple, sur 20 ou sur 10). Chaque étudiant a, pour chaque examen, au plus une note (éventuellement manquante si l'examen n'a pas encore eu lieu ou si l'étudiant est absent).

Proposer un schéma relationnel (avec tables, attributs, contraintes d'intégrité) pour ces données. On veillera à éviter au maximum la redondance de l'information : une donnée élémentaire ne doit pas être stockée plusieurs fois dans la base.

Dans un cours ultérieur, nous formaliserons cette notion de redondance, et proposerons une stratégie générale pour concevoir des schémas de bases de données.

2 Construction de la base

On utilisera pour ce TP le serveur de bases de données PostgreSQL accessible avec les paramètres (serveur, identifiant, mot de passe, base) donnés pendant la séance. Il est également possible d'utiliser votre propre installation de PostgreSQL.

Ces paramètres peuvent être utilisés pour se connecter au serveur de bases de données : avec le programme de ligne de commande `psql` qui s'utilise comme suit :

```
psql base identifiant -h serveur
```

1. Tester la connexion à la base. La base est vide pour l'instant. « `\h` » fournit une aide sur les commandes SQL, tandis que « `\?` » fournit une aide sur les commandes propres à l'outil `psql`.
2. Créer en utilisant des commandes `CREATE TABLE` les tables et contraintes d'intégrité proposées dans l'étape précédente. On se reportera à la documentation de PostgreSQL pour sélectionner les types de données les plus appropriés pour chaque attribut.
3. Vérifier le schéma des tables créées avec la commande « `\d` » suivi du nom de table.
4. Remplir ces tables en y ajoutant quelques tuples d'exemple d'étudiants et d'examens.
5. À l'aide d'un tableur (par exemple LibreOffice Calc), créer un ensemble de tuples indiquant la note obtenue par chaque étudiant à chaque examen, dans un format approprié pour l'import à la base. Sauvegarder le résultat comme un fichier CSV, puis utiliser la commande `\COPY`, dont vous pouvez consulter la syntaxe à l'URL `https://www.postgresql.org/docs/current/static/app-psql.html#APP-PSQL-META-COMMANDS-COPY`

3 Algèbre relationnelle et requêtes SQL

Pour chacune des requêtes suivantes, écrire l'expression de l'algèbre relationnelle et la requête SQL correspondantes, et tester l'exécution de cette requête sur vos données.

1. L'ensemble des informations (titre, date, coefficient, note maximale) sur tous les examens.
2. L'ensemble des informations sur un examen d'un identifiant donné.
3. L'ensemble des notes de tous les étudiants à un examen donné (avec, pour chaque étudiant, son prénom et son nom).
4. La moyenne pondérée obtenu par tous les étudiants à l'ensemble des examens, normalisée en une note sur 20.