



Bases de données avancées

TP n° 2 : Révisions SQL

Rappel du psql

La description complète de toutes les commandes de psql se trouve à <https://docs.postgresql.fr/13/app-psql.html>

Voici un bref rappel du fonctionnement de psql :

- `\?` donne toutes les commandes
- `\h` ou `\h mot_cle` donne une aide sur les commandes SQL
- `\i commandes.sql` permet d'importer un fichier de commandes SQL. Nous vous encourageons à écrire toutes vos requêtes dans un fichier pour garder une trace de votre travail.
- `\q` pour quitter PostgreSQL
- `\d` pour obtenir de l'information sur vos tables
- `\d matable` pour obtenir de l'information sur une table particulière
- `\echo message` permet d'afficher un message, par exemple le numéro de la question à laquelle va répondre la requête ;
- `\e` édite la dernière commande (mode vi ou emacs)
- `\! ls` effectuer la commande `ls` (marche pour toute commande Unix)
- `\cd` change de répertoire
- `fleche haut` revenir à la commande précédente ;
- `CTRL-a` aller au début de la ligne ;
- `CTRL-e` aller à la fin de la ligne ;
- `CTRL-d` effacer le caractère suivant ;
- `CTRL-k` effacer tout jusqu'à la fin de la ligne.

Base sur les résultats scolaires

Les exercices suivants portent sur les résultats des étudiants aux contrôles de différentes matières. Vous pouvez rapatrier et importer le fichier `notes.sql` disponible sur Moodle qui contient les requêtes de création de tables et les requêtes INSERT pour les alimenter. Nous rappelons la description des tables, en soulignant les clés primaires, et en indiquant avec une ou plusieurs étoiles les champs composants des clés étrangères (la table de référence est implicite).

```
-- etudiant (num_etu, nom, prenom)
-- matiere (id_mat, nom, coeff)
-- controle (id_controle, id_mat*, date_controle, coeff)
-- inscription (num_etu*, id_mat**)
-- note (num_etu*, id_controle**, note, obs)
```

Remarquez que l'attribut `coeff` présent dans les tables `matiere` et `controle` désigne deux coefficients différents. De même l'attribut `nom` dans la table `matiere` n'a rien à voir avec `nom` dans la table `etudiant`. Ceci peut conduire à des résultats inattendus en cas de `NATURAL JOIN`.

Exercice 1 :

Que font les requêtes suivantes ?

```
1 SELECT AVG(note)
2   FROM note;
3
4 SELECT AVG(note)
5   FROM note
6  WHERE note > 10;
7
8 SELECT id_controle, AVG(note)
9   FROM note
10  WHERE note > 10
11  GROUP BY id_controle;
12
13 SELECT id_controle, AVG(note)
14   FROM note
15  WHERE note > 10
16  GROUP BY id_controle
17  HAVING AVG(note) > 18;
```

Exercice 2 :

Écrivez les requêtes qui répondent aux questions suivantes. Pour cette partie, ne pas tenir compte des coefficients des matières ou des contrôles.

1. Nombre total d'étudiants.
2. Nombre de matières qui ont au moins un contrôle.
3. Nombre de contrôles par matière (y compris les matières pour lesquelles il n'y a pas de contrôle). Afficher bien le nom de la matière.
4. Moyenne par contrôle. Pour chaque contrôle afficher : `id_controle`, `date_controle`, le `nom` de la matière et la moyenne.
5. Moyenne par matière. Pour chaque matière afficher `id_mat`, le `nom` de la matière et la moyenne.
6. Pour chaque matière, le nombre d'étudiants qui ont passé au moins un contrôle dans cette matière. Ne pas oublier les matières sans contrôle. Afficher : `id_mat`, `nom` de la matière, et le nombre d'étudiants. Le résultat trié par le `nom` de la matière.
7. Moyenne de chaque étudiant, par matière (moyenne simple de tous ses contrôles), présentée par ordre de `id_mat` croissant. Afficher le nom et prénom de l'étudiant, le `nom` de la matière et la moyenne.

8. Moyenne générale de chaque étudiant, ne prenant pas en compte les contrôles dont la note est inférieure à 5. Les étudiants pour lesquels on n'a pas de notes, ou dont les notes sont toutes inférieures à 5, doivent apparaître avec `NULL` comme moyenne. Présenter le résultat par l'ordre croissant de moyennes.
9. Moyenne de chaque étudiant (sans compter les contrôles dont la note est inférieure à 5) lorsque celle-ci est supérieure à 10.

Exercice 3 :

Les requêtes suivantes sont à faire avec les opérations ensemblistes `UNION`, `INTERSECT`, `EXCEPT`...

10. La liste des étudiants qui sont inscrits ou qui ont passé un contrôle dans la matière 'Algorithmique'. (La requête doit encore marcher si le code de la matière change.) Rappel : attention à l'attribut `nom` présent à la fois dans les tables `etudiant` et `matiere`.
11. La liste des étudiants qui ont passé un contrôle d'algorithmique mais qui ne sont pas inscrits dans cette matière.
12. La liste des étudiants qui sont inscrits en algorithmique et ont passé au moins un contrôle dans cette matière.

Exercice 4 :

Les trois requêtes suivantes peuvent nécessiter l'emploi de sous-requêtes. Les coefficients interviennent.

13. Nombre d'étudiants ayant une moyenne générale supérieure à 10.
14. Pour chaque matière, nombre d'étudiants ayant une moyenne supérieure à 12.
15. Pour chaque étudiant le nombre de matières réussies (c'est-à-dire où il a la moyenne supérieure ou égale à 10).