

TP noté : Requêtes SQL avancées

Ce TP doit être réalisé en équipe de 1 ou 2 élèves, et remis sur Campus. La date limite est indiquée dans l'espace de remise.

Sommaire

1	Sujet.....	1
2	Attention	1
2.1	Correction automatique	1
2.2	Exécution du script	2
2.3	Base de correction.....	2
2.4	SQL standard	3
3	Barème	3
4	Remise	3
4.1	Livrable	3
4.2	Dépôt.....	3

1 Sujet

Il s'agit de répondre aux questions listées dans le script `queries.sql` sous forme de commentaires SQL. Ces requêtes portent sur la base de données créée et peuplée par le script `database.sql`.

Pour faire le TP : (1) téléchargez et exécutez le script `database.sql`, (2) téléchargez et complétez le script `queries.sql`.

La population de la base fournie en exemple dans `database.sql` est volontairement simple. Il se peut que les questions posées dans `queries.sql` ne donnent aucune réponse avec cette population. Vous devez donc la compléter pour tester vos requêtes dans le plus de cas possibles.

L'archive `Example.zip` vous donne un exemple du fichier que vous devez remettre (répertoire « File to Submit ») en fonction des fichiers ressources qui vous sont donnés (répertoire « Resources »).

Si vous avez une question à propos de ce TP, postez-la svp sur le forum Questions et Réponses. Je ne répondrai pas aux courriels envoyés à mon adresse personnelle. Merci.

2 Attention

2.1 Correction automatique

Votre script sera corrigé de façon automatique. Les réponses qu'il affichera seront comparées, lignes à lignes et colonnes à colonnes, avec celles données par le script corrigé exécuté sur la même base de données. Vous devez donc veiller attentivement aux points suivants :

- Répondez précisément aux questions, comme en TP : une seule requête SQL par question, avec tous les attributs demandés et ceux-là seulement, dans l'ordre demandé. Si aucun attribut particulier n'est demandé, il faut les afficher tous, dans l'ordre où ils sont déclarés dans la table.
- Pour certaines questions, les attributs permettant de trouver ou trier les informations demandées ne doivent pas être affichés : c'est normal. Vous pouvez bien entendu afficher ces attributs quand vous testez vos requêtes, mais pensez à les retirer avant de remettre votre script.
- Dans vos requêtes, respectez la casse des noms de table et d'attributs définie dans le script `database.sql`. Attention, le serveur de base de données que je vais utiliser est sensible à la casse, alors que celui de votre installation WAMP/XAMPP/MAMP ne l'est pas nécessairement.
- Ne supprimez ou ne modifiez aucune des lignes du script, notamment les balises « `select 'Query xy' ...` » : elles servent à identifier, dans la sortie, le numéro de question traitée.
- Votre script ne doit pas afficher autre chose (tests, commentaires, etc.) que le résultat des requêtes ; il ne doit pas appeler ou contenir tout ou partie du script `database.sql`.

Le nom des colonnes des tables affichées par votre script est ignoré lors de la correction : dans la clause `select` de vos requêtes, vous pouvez renommer les attributs et expressions comme bon vous semble.

2.2 Exécution du script

Votre script sera exécuté sur mon installation MySQL locale en utilisant un outil similaire à l'onglet `Import` de phpMyAdmin. En conséquence, vous devez exécuter le même test pour vérifier que tout est correct avant de remettre votre script.

Notez svp que tester vos requêtes une par une dans l'onglet `SQL` de phpMyAdmin n'est pas suffisant. Exécuter votre script d'un coup via l'onglet `Import` vous permettra de détecter les erreurs les plus courantes : (a) oubli du point-virgule à la fin de chaque ordre SQL, (b) erreur sur la casse des noms de table et d'attribut, sachant que le serveur MySQL de l'ECE y est sensible, (c) non prise en compte de l'ordre `set session sql_mode...` car vous testez vos requêtes en les exécutant une par une.

2.3 Base de correction

Une requête ne doit pas faire d'hypothèse sur la population de la base : elle doit retourner un résultat correct quelle que soit cette population. Votre script sera testé sur une base de données ayant une population différente de celle de la base exemple. (Son schéma, en revanche, sera bien sûr le même.)

En conséquence, si une question demande d'afficher les produits commandés par Smith (par exemple), votre requête ne doit pas utiliser en dur l'ID 1234 parce que c'est celui donné pour Smith dans la base exemple. Votre requête doit au contraire reprendre les termes de la question (ici « Smith ») et ceux-là seulement.

Pensez également à tenir compte de tous les cas particuliers : valeurs null, doublons, entités ne participant pas aux associations, etc. S'il y a un doute sur le fait qu'un tuple réponde à une question, notamment à cause de valeurs null, votre requête doit faire comme la clause `where` : ne pas l'afficher.

2.4 SQL standard

MySQL autorise l'exécution de requêtes SQL incorrectes, notamment avec la clause `group by`. Pour les détecter, le script de requêtes contient la commande suivante :

```
set session sql_mode = 'ONLY_FULL_GROUP_BY';
```

Vous ne devez évidemment pas supprimer cette ligne.

3 Barème

Le barème indicatif est le suivant :

Rubrique	Points
Requêtes	18
Qualité du code	2
<i>Total</i>	<i>20</i>

Toutes les requêtes, simples ou complexes, valent les mêmes points. Pour la « Qualité du code », vos requêtes doivent : (1) être écrites en SQL standard, (2) utiliser les formes « natural join » ou « join on » de SQL92 pour les jointures, (3) être proprement formatées.

4 Remise

4.1 Livrable

Le livrable est constitué du seul script `queries.sql` que vous devez renommer `NOM1.Prenom1.NOM2.Prenom2.sql`, **sans aucun espace**, pour identifier les membres de l'équipe.

4.2 Dépôt

Déposez directement le fichier dans l'espace de remise, **sans constituer d'archive**. Vous pouvez le redéposer autant de fois que vous voulez jusqu'à la date limite, mais **toujours sous le même utilisateur Campus**.