

Optimiser sa base Mysql

Pourquoi optimiser une base ?

Il faut entretenir sa base de données pour qu'elle soit toujours performante. Ce que l'on entend par performante, c'est le fait que les informations contenues dans la base soient le plus rapidement envoyées au script qui les demande. Pour cela, il faut une base bien structurée et optimisée. Dans ce guide, nous allons vous montrer comment optimiser au mieux votre base.

Sur la base

- Indexer la base :

Pour augmenter la rapidité des recherches lors d'une requête, il faut mettre un index sur les champs qui sont utilisés dans les clauses WHERE.

Exemple :

Vous faites une recherche de personne par rapport à la ville. Il faut indexer le champ "ville" avec la requête suivante :

```
ALTER TABLE `test` ADD INDEX ( `ville` );
```

- Purger la base :

Certaines de vos données ne sont plus consultées. Pourquoi ne pas les archiver ? Vos tables seront moins pleines et les recherches iront plus vite.

Dans les scripts

- Limitation d'affichage :

Limiter l'affichage des enregistrements à un nombre restreint (genre 10 par page) avec la partie LIMIT de la requête.

- Regroupement des requêtes :

Regrouper vos requêtes en debut de script comme cela :

```
connexion_base  
requete1  
requete2  
...  
deconnexion_base
```

Affichage ...
Traitement des donnees
Boucles ...
Affichage ...
...

- Faire du cache :

Si vous avez des infos qui sont tirées de la db et qui ne changent pas souvent, mettez-les en cache. Par exemple, générez la page html quand vous modifiez les infos, ce genre d'astuce diminuera drastiquement vos accès. Par exemple une page de news. Si le fichier (cache) html existe, vous l'utilisez (simple include) sinon, vous le générez. Et au moment où vous postez une nouvelle news, vous pouvez inclure dans le script qu'il supprime le fichier html : il sera ainsi régénéré la prochaine fois qu'un visiteur la demande. Vous pouvez aussi faire du cache de session. Mettre les résultats de requête en variable de session, après la même requete vous ne l'exécutez plus, vous récupérez les variables de session.

- Prendre uniquement le necessaire :

Dans vos requêtes SQL vérifiez que vous ne sélectionnez que ce dont vous avez besoin, et surtout que vous n'avez pas oublié les liaisons entre les tables (where table1.champs = table2.champs2) car sinon ça prend énormément de temps (même si après avec le where il vous sort un truc correct).

- Eviter les options très gourmandes :

Évitez d'utiliser HAVING c'est aussi une usine à gaz, idem pour GROUP BY. Bien sûr, parfois c'est obligatoire, mais parfois pas.