

Performances

Introductions aux Bases de Données
Nathanaël Martel

Performances

Problématique : comment accélérer notre application ?

L'optimisation peut se faire à deux niveaux :

- **L'application**
 - Algorithme et architecture n-tiers
 - Mise en cache des requêtes
- **La base de données**
 - Index

Optimisation de l'application

- **Objectif : limiter le nombre d'aller retour entre l'application et la base de données.**
- **Exemple : Liste de 25 citations avec leurs auteurs**

Optimisation de l'application

- **Solution 1 :**

```
$results = $pdo->query('SELECT * FROM `citation` LIMIT 25') ;  
  
foreach ($results as $line) {  
    $q = 'SELECT * FROM `author` WHERE id=' .  
    $line['author_id'] ;  
    $author = mysql_query($q) ;  
    Echo $line['quote'] . ' - ' . $author['name'];  
}
```

Optimisation de l'application

- **Solution 2 :**

```
$result = $pdo->query('SELECT * FROM `citation` LEFT  
OUTER JOIN `author` ON  
citation.author_id=citation.author_id LIMIT 25') ;  
  
foreach ($results as $line) {  
    Echo $line['quote']. ' - ' . $line['name'];  
}
```

Optimisation de l'application

Nombre de requête :

- **Solution 1 : 26 requêtes**
- **Solution 2 : 1 requête**

→ Écrire des requêtes les plus précise possible pour ne pas avoir à faire plusieurs requêtes.

Optimisation de l'application

De manière générale

→ Demander à la base de données de faire le plus de calcul possible :

- Trier les résultats**
- Calculer des sommes, moyennes...**

Optimisation de l'application

Si vous avez une requête qui est utilisée plusieurs fois, il peut être intéressant de la mettre en cache, au moins sa structure.

```
$sth = $pdo->prepare('SELECT nom, couleur, calories  
FROM fruit  
WHERE calories < ? AND couleur = ?');  
$sth->execute(array(150, 'rouge'));  
$red = $sth->fetchAll();  
$sth->execute(array(175, 'jaune'));  
$yellow = $sth->fetchAll();
```


Optimisation de la base de données

Certains champs peuvent avoir un :

- **« index »**

Optimisation de la base de données

Pour que la base de données trouve plus vite les enregistrements, elle peut « indexer » un champs en particulier.

Concrètement, elle va stocker dans un fichier la valeur de ce champs dans un ordre très précis (alphabétique) pour pouvoir retrouver très rapidement les enregistrements concerné.

→ indexer les champs sur lesquels vous faites des requêtes fréquente.

Conclusion

Deux axes d'optimisation :

 Limiter le nombre de requêtes

 (et rejeté à la base de données un maximum de calcul)

Mettre des index au champs de la table qui sont fréquemment filtré