

Optimize your Mysql base

### **Why to optimize a base?**

You must take care of your database in order it to be always effective. It means that the information contained in your base has to be sent the quickest as possible to the script that requests for it. To do so, your base must be structured and optimized. In this manual, we are going to show you how to optimize your base at the best.

### **On the base:**

- Index the base:

To increase the research speed at the time of a request, you must input an index on fields used in WHERE rule.

Example:

You make a research of people in a specific town. You must index 'town' field with the following request:  
`ALTER TABLE 'test' ADD INDEX ('town');`

- Purge the base:

Some data is consulted. Why don't you archive it? Your tables are less full and the researches will be quicker.

### **In the scripts:**

- Display limit:

Limit the display of the records to a restricted number (e.g. 10/page) with the LIMIT part of the request.

- Request consolidation:

Consolidate your request at the beginning of the script as follows:

```
connection_base  
request1  
request2  
...  
base disconnection
```

Display...  
Data processing  
Loops...  
Display...  
...

- Caching:

If you have information from the data base, which is not often modified, cache them. For example, generate the html page when you modify the information, it will reduce your access. For example a news page. If the html file (cache) exists, you use it (simple include). Otherwise, you generate it and at the time you post some news, you may include in the script that it deletes the html file. It will be thus regenerated the next time a visitor requests for it. You may also make a session cache. Put the results of the request in session variable, after the same request you won't execute it any more, you recover the session variables.

- Only take the necessary!

In your SQL requests, make sure that you only select what you need, and specially that you have not forgotten the links between the tables (where table1.field = table2.field2) otherwise it will take a lot of time (even if after, with the 'where' you get something correct).

- Avoid the very greedy options:

Avoid using 'HAVING', it is a gas turbine, id for GROUP BY. Of course, sometimes it is mandatory, but sometimes it isn't.