

Re: what is the speed hit of "SELECT *" with MySql, as opposed to narrower database calls?

Re: what is the speed hit of "SELECT *" with MySql, as opposed to narrower database calls?

Source: <http://coding.derkeiler.com/Archive/PHP/comp.lang.php/2005-04/msg00640.html>

- *From:* "Tony Marston" <tony@xxxxxxxxxxxxxxxxxxxx>
 - *Date:* Sun, 10 Apr 2005 10:52:55 +0100
-

"Philip Nelson" <teamdba@xxxxxxxxxx> wrote in message
[news:weX5e.7695123\\$f47.1412387@xxxxxxxxxxxxxxxxxxxxxxxx](mailto:news:weX5e.7695123$f47.1412387@xxxxxxxxxxxxxxxxxxxxxxxx)

> lkrubner@xxxxxxxxxxxxxxxx wrote:

>

>>

>> Are there any benchmarks on how much an extra, unneeded VARCHAR, CHAR,
>> INT, BIGINT, TEXT or MEDIUMTEXT slows down a database call with MySql?
>> PostGre info would also be useful.

>>

>> I'm trying to explain to some friends the utility of making database
>> calls return only needed data.

>>

>> As an example of what I'm talking about, suppose we had a database
>> table sort of like this:

>>

>> table weblogs (
>> int id,
>> varchar 255 headline,
>> text mainContent,
>> int dateCreated,
>> varchar 255 author,
>> varchar 255 navigationText,
>> char 1 isPrivate
>>);

>>

>>

>> Suppose I'm doing a PHP command that get's the headlines and offers a
>> link to the actual page. What is the speed difference between

>>

>> SELECT * FROM weblogs

>>

>> as opposed to

>>

>> SELECT id, headline FROM weblogs

>>

>> And is there info out there that gives a general sense of how much each
>> extra, unneeded database field might slow down a script?

Re: what is the speed hit of "SELECT *" with MySql, as opposed to narrower database calls?

Re: what is the speed hit of "SELECT *" with MySql, as opposed to narrower database calls?

- >
- > I've not got any figures on performance, although depending on the number
- > of
- > columns involved and their size it could be quite a bit. I can tell you
- > of
- > one instance I came across with another DBMS using ODBC, which just loves
- > to read the system catalog for information about everything right down to
- > column level, that to return 5000 rows of user data from a 70 column table
- > took over 2 million page I/Os to the system catalog. Not exactly great
- > for performance.
- >
- > In my opinion there are other, more important, reasons for not using
- > "SELECT
- > *". These principally relate to what will happen to the application if
- > you change the database structure. For example, in the table you gave
- > lets
- > assume we decide to add a column publisher varchar(255) immediately after
- > the author column. Let's also assume that you access the columns in a
- > resultset by position, rather than by name. Now if you use a "select *"
- > where you displayed navigationText before you will now display publisher.

That's the primary reason for returning associative lists, so that you can reference items by name instead of their position. That way it does not matter in what order the columns are retrieved.

- > Similar problems occur with insert statements where the column list isn't
- > specified, but just the values list. Everything is OK until you add a
- > column. If the last column is "NOT NULL" the statement breaks. Worse
- > though you can end up putting data into the wrong places.

If you do what I do and construct all INSERT, UPDATE and DELETE statements programmatically then this problem will never appear.

--

Tony Marston

<http://www.tonymarston.net>

• **References:**

- ◆ **what is the speed hit of "SELECT *" with MySql, as opposed to narrower database calls?**
 ◇ From: lkrubner
- ◆ **Re: what is the speed hit of "SELECT *" with MySql, as opposed to narrower database calls?**
 ◇ From: Philip Nelson

Re: what is the speed hit of "SELECT *" with MySql, as opposed to narrower database calls?

- Prev by Date: [*"not a valid JPEG file"*](#)
- Next by Date: [*splitting lines in arrays?*](#)
- Previous by thread: [*Re: what is the speed hit of "SELECT *" with MySql, as opposed to narrower database calls?*](#)
- Next by thread: [*Re: what is the speed hit of "SELECT *" with MySql, as opposed to narrower database calls?*](#)
- Index(es):
 - ◆ [*Date*](#)
 - ◆ [*Thread*](#)