

Re: Char and Varchar

Source:

<http://www.tech-archive.net/Archive/SQL-Server/microsoft.public.sqlserver.programming/2004-05/2818.html>

From: Hugo Kornelis (*hugo_at_pe_NO_rFact.in_SPAM_fo*)

Date: 05/12/04

Date: Wed, 12 May 2004 23:37:58 +0200

On Wed, 12 May 2004 09:36:31 -0400, Steve Z wrote:

>Hey guys, he was asking how it was stored – it's always good to know what
>goes on behind the scenes.

I agree. That's why I replied to your post, acknowledging that you were almost completely right, correcting a small error and adding an important consideration. I have seen too many examples of people using only fixed-length character data in order to avoid the overhead of having to find the start and end of the data. Much as I'd like to believe that people will realise by themselves that this overhead is nothing compared to the increased I/O, I can't.

>Would you ever create a table with lets say 100 single character VARCHAR
>fields? Would there be value to that, if the columns were guaranteed to
>have a character in them (ie. not a space).

Of course not. If the maximum length is short (≤ 10 characters), I always use CHAR. If it's longer, but most values use almost the full maximum length, I also use CHAR. I use VARCHAR if long and short strings are interspersed.

>How would a primary clustered key behave if it was built on a VARCHAR field
>vs. a CHAR field?

It depends. If the column has a maximum length of 250 characters, but most values are a lot shorter, using VARCHAR would reduce the number of index pages on the intermediate levels. The leaf level would of course not be affected (apart from the effect of any CHAR vs VARCHAR column decision on the data pages). For an unindexed key, the leaf level would need less pages as well. If the table holds many rows, I think (I haven't tested it!) that the number of I/O's would be less, hence the queries would return results faster. All this at the price of the CPU having less idle time.

>And, please explain to me how the 100th VARCHAR fields byte position can be
>determined in a buffer of data without reading each of the prior 99 VARCHAR

>control fields? Without knowing the length of each of the prior 99 data
>elements in that row, how could the starting byte of the 100th element be
>determined?

Easy. Take the end position of the 99th element (stored as a fixed position in the row, after the fixed length and before the varying length data) and add 1.

> Since the length of the VARCHAR is limited to 8000 byte – that
>takes 14 of the 16 bits in the 2–byte control field. So, I can assume from
>that fact that the 2–byte control field is not storing the "position" of the
>element, but simply the length of each element.

You seem to forget that the length of the complete row may not exceed 8,060 bytes (check BOL, Maximum Capacity Specifications). The 2–byte control field will hold the ending position of each element just fine.

>Inside MS SQL Server 2000, by Kalen Delany, on page 230 states that VARCHAR
>are most appropriate when you expect significant differences in the lengths
>of the data in the column and when the data length in the column wont
>frequently be changed.

Apart from the page number, this has apparently not changed from the previous edition (Inside MS SQL Server 7.0, by Ron Soukup and Kalen Delany).

>Changing the length frequently would required
>rebuilding the data page – right?
>
>If changing an element to full size causes it to no longer fit on the data
>page, then page splits occur – that's alot of work for the engine to handle.

It's been a while since I read this part, but I think that's not entirely correct. The row's data itself will have to be rebuilt, as there can be no gaps between the contents of the varying length columns. If the row is shortened, it will stay where it is. If the row's size increases, it is replaced by a forwarding pointer and the new row's content is stored somewhere else; the other rows on the same page remain intact. If the row has to move again later, the original forwarding pointer is changed (to prevent long chains of pointers).

>Page 233 indicates that "no one answer is right" and "If you understand the
>tradeoffs; you'll be able to make the best choice".

Yes – a terrific piece of advice that I support wholeheartedly!

Best, Hugo

--

(Remove `_NO_` and `_SPAM_` to get my e-mail address)