Craig S. Mullins Articles

Database Performance Management

Return to Home Page

October 1998

DB2 update

Character Versus Numeric Data Types

By Craig S. Mullins

Most DBAs have faced the situation where one of their applications requires a four-byte code that is used to identify products, accounts, or some other business object, and all of the codes are numeric and will stay that way. But, for reporting purposes, users or developers wish the codes to print out with leading zeroes. So, the users request that the column be defined as CHAR(4) to ensure that leading zeroes are always shown. But what are the drawbacks, if any, to doing this?

Well, there are drawbacks! Without proper edit checks, inserts and updates could place invalid alphabetic characters into the product code. This can be a very valid concern if ad hoc data modifications are permitted. This is rare in production databases, but data problems can still occur if the proper edit checks are not coded into every program that can modify the data. But let's assume that proper edit checks are coded and will never be bypassed. This removes the data integrity question.

There is another problem that is related to filter factors. Consider the possible number of values that a CHAR(4) column and a SMALLINT column can assume. Even if edit checks are coded for each, DB2 is not aware of these and assumes that all combinations of characters are permitted. DB2 uses base 37 math when it determines access paths for character columns, under the assumption that 26 alphabetic letters, 10 numeric digits, and a space will be used. This adds up to 37 possible characters. For a four-byte character column there are 374 or 1,874,161 possible values.

A SMALLINT column can range from -32,768 to 32,767 producing 65,536 possible small integer values. The drawback here is that negative or 5 digit product codes could be entered. However, if we adhere to our proper edit check assumption, the data integrity problems will be avoided here, as well.

DB2 will use the HIGH2KEY and LOW2KEY values to calculate filter factors. For character columns, the range between HIGH2KEY and LOW2KEY is larger than numeric columns because there are more total values. The filter factor will be larger for the numeric data type than for the character data type which may influence DB2 to choose a different access path. For this reason, favor the SMALLINT over the CHAR(4) definition.

The leading zeroes problem might be able to be solved using other methods. When using QMF, you can ensure that leading zeroes are shown by using the "J" edit code. Report programs can be coded to display leading zeroes easily enough by moving the host variables to appropriate display fields. Ad hoc access through other reporting tools typically provide a parameter that can enable leading zeroes to be displayed.

In general, it is wise to choose a data type which is closest to the domain for the column. IF the column is to store numeric data, favor choosing a numeric data type: SMALLINT, INTEGER, DECIMAL, or floating point. In addition, always be sure to code appropriate edit checks to ensure data integrity.

From DB2 Update (Xephon), October 1998.