CHAR and VARCHAR

Firebird provides two basic data types to store text or character information: CHAR and VARCHAR (blobs also allow character storage using the subtype text).

CHAR and VARCHAR are data types which can store any text information. Numbers that are not calculated, such as zip codes, are traditionally stored in CHAR or VARCHAR columns. The length is defined as a parameter, and can be between 1 and 32,767 bytes. It is particularly useful for codes that typically have a fixed or predefined length, such a the zip code for a single country.

Compared to most other databases, Firebird only stores significant data. If a column is defined as CHAR(100), but only contains entries with 10 characters, the additionally defined bytes are not used, as Firebird stores CHAR and VARCHAR types similarly, and does not fill unused spaces with blanks. Both CHAR and VARCHAR are stored in memory buffer in their full, declared length; but the whole row is compressed prior to storing i.e. CHARs, VARCHARSs, INTEGERs, DATESs, etc. all together.

Indeed, VARCHAR columns require more storage than CHAR columns, because when storing a VARCHAR, Firebird adds two bytes that state just how big the VARCHAR actually is.

So a CHAR will in fact be stored in a smaller space. However, when a SELECT is performed on a VARCHAR column, Firebird strips the 2 byte padding and returns the stored value. When a SELECT is performed on a CHAR column, Firebird returns the value and the “empty spaces”. Thus the two bytes saved in storage of a CHAR must be balanced against the subsequent need to strip the spaces on the client side. These two bytes however are, with today's hardware, too negligible to have an influence upon the database performance. This can however be disadvantageous when defining short text fields.

In practical terms consider just this one rule: only use CHARs if strings of few characters are to be stored; the exception to the rule being when working with intermediate tables that are required to export data to fixed length prn files. Then the fixed length field will be a positive advantage.

This efficient storage in Firebird can lead to considerable confusion particularly when importing data, as Paradox or dBase databases save all blank spaces, and after importing a 10MB dBase file into InterBase, often only 3-6 MB remain, although all data sets were imported correctly.

For this reason columns can be defined generously in Firebird without a problem, whereas in other databases each defined byte influences the size of the database, regardless of whether data is stored in these fields or not.

Please note however that indexed CHAR fields should not be more than approx. 80 characters in length (with Firebird 1.5 the limit is somewhat higher).

Another factor to take into consideration when defining the length of CHAR and VARCHAR fields is the size of temporary files, as these also include the full space definition for long CHAR or VARCHAR columns. Many developers are very generous when defining CHARs and VARCHARs because they know that Firebird stores them in binary form with the string content plus the number of empty spaces (and not the empty spaces themselves). However, when these fields are loaded into the temp directory, they are written out in full. For this reason it often makes more sense to define a text BLOB rather than a huge VARCHAR. BLOB fields are only limited in size by the page size; for example, with a page size of 8 KB the maximum blob size is 32 Gigabytes.

The CHAR data type definition can be written in two ways:

```
CHAR
CHARACTER
```

The VARCHAR data type definition can be written as follows:

```
VARCHAR
CHARACTER VARYING
CHAR VARYING
```