

```
FIELDS TERMINATED BY ' '
LINES TERMINATED BY '\n%%\n' (joke);
```

FIELDS [OPTIONALLY] ENCLOSED BY controls quoting of fields. For output (**SELECT ... INTO OUTFILE**), if you omit the word **OPTIONALLY**, all fields are enclosed by the **ENCLOSED BY** character. An example of such output (using a comma as the field delimiter) is shown here:

```
"1","a string","100.20"
"2","a string containing a , comma","102.20"
"3","a string containing a \" quote","102.20"
"4","a string containing a \", quote and comma","102.20"
```

If you specify **OPTIONALLY**, the **ENCLOSED BY** character is used only to enclose values from columns that have a string data type (such as **CHAR**, **BINARY**, **TEXT**, or **ENUM**):

```
1,"a string",100.20
2,"a string containing a , comma",102.20
3,"a string containing a \" quote",102.20
4,"a string containing a \", quote and comma",102.20
```

Occurrences of the **ENCLOSED BY** character within a field value are escaped by prefixing them with the **ESCAPED BY** character. Also, if you specify an empty **ESCAPED BY** value, it is possible to inadvertently generate output that cannot be read properly by **LOAD DATA**. For example, the preceding output just shown would appear as follows if the escape character is empty. Observe that the second field in the fourth line contains a comma following the quote, which (erroneously) appears to terminate the field:

```
1,"a string",100.20
2,"a string containing a , comma",102.20
3,"a string containing a " quote",102.20
4,"a string containing a ", quote and comma",102.20
```

For input, the **ENCLOSED BY** character, if present, is stripped from the ends of field values. (This is true regardless of whether **OPTIONALLY** is specified; **OPTIONALLY** has no effect on input interpretation.) Occurrences of the **ENCLOSED BY** character preceded by the **ESCAPED BY** character are interpreted as part of the current field value.

If the field begins with the **ENCLOSED BY** character, instances of that character are recognized as terminating a field value only if followed by the field or line **TERMINATED BY** sequence. To avoid ambiguity, occurrences of the **ENCLOSED BY** character within a field value can be doubled and are interpreted as a single instance of the character. For example, if **ENCLOSED BY ' '** is specified, quotation marks are handled as shown here:

```
"The ""BIG"" boss" -> The "BIG" boss
The "BIG" boss     -> The "BIG" boss
The ""BIG"" boss   -> The ""BIG"" boss
```

FIELDS ESCAPED BY controls how to read or write special characters:

- For input, if the **FIELDS ESCAPED BY** character is not empty, occurrences of that character are stripped and the following character is taken literally as part of a field value. Some two-character sequences that are exceptions, where the first character is the escape character. These sequences are shown in the following table (using **** for the escape character). The rules for **NULL** handling are described later in this section.

Character	Escape Sequence
\0	An ASCII NUL (x'00') character
\b	A backspace character
\n	A newline (linefeed) character