

- Statements are ignored and not logged if they match any pattern in the “ignore” list. This list is described later.
- `mysql` logs each nonignored, nonempty statement line individually.
- If a nonignored statement spans multiple lines (not including the terminating delimiter), `mysql` concatenates the lines to form the complete statement, maps newlines to spaces, and logs the result, plus a delimiter.

Consequently, an input statement that spans multiple lines can be logged twice. Consider this input:

```
mysql> SELECT
-> 'Today is'
-> ,
-> CURDATE()
-> ;
```

In this case, `mysql` logs the “SELECT”, “Today is”, “,”, “CURDATE()”, and “;” lines as it reads them. It also logs the complete statement, after mapping `SELECT\n'Today is'\n,\nCURREDATE()` to `SELECT 'Today is' , CURDATE()`, plus a delimiter. Thus, these lines appear in logged output:

```
SELECT
'Today is'
,
CURDATE()
;
SELECT 'Today is' , CURDATE();
```

`mysql` ignores for logging purposes statements that match any pattern in the “ignore” list. By default, the pattern list is “`*IDENTIFIED*:*PASSWORD*`”, to ignore statements that refer to passwords. Pattern matching is not case-sensitive. Within patterns, two characters are special:

- `?` matches any single character.
- `*` matches any sequence of zero or more characters.

To specify additional patterns, use the `--histignore` option or set the `MYSQL_HISTIGNORE` environment variable. (If both are specified, the option value takes precedence.) The value should be a list of one or more colon-separated patterns, which are appended to the default pattern list.

Patterns specified on the command line might need to be quoted or escaped to prevent your command interpreter from treating them specially. For example, to suppress logging for `UPDATE` and `DELETE` statements in addition to statements that refer to passwords, invoke `mysql` like this:

```
mysql --histignore="*UPDATE*:*DELETE*"
```

Controlling the History File

The `.mysql_history` file should be protected with a restrictive access mode because sensitive information might be written to it, such as the text of SQL statements that contain passwords. See [Section 6.1.2.1, “End-User Guidelines for Password Security”](#). Statements in the file are accessible from the `mysql` client when the **up-arrow** key is used to recall the history. See [Disabling Interactive History](#).

If you do not want to maintain a history file, first remove `.mysql_history` if it exists. Then use either of the following techniques to prevent it from being created again:

- Set the `MYSQL_HISTFILE` environment variable to `/dev/null`. To cause this setting to take effect each time you log in, put it in one of your shell's startup files.
- Create `.mysql_history` as a symbolic link to `/dev/null`; this need be done only once: