14.7.5.39 SHOW VARIABLES Syntax

```
SHOW [GLOBAL | SESSION] VARIABLES
    [LIKE 'pattern' | WHERE expr]
```

> **Note**
>
> As of MySQL 5.7.6, the value of the show_compatibility_56 system variable affects the information available from and privileges required for the statement described here. For details, see the description of that variable in Section 6.1.4, “Server System Variables”.

SHOW VARIABLES shows the values of MySQL system variables (see Section 6.1.4, “Server System Variables”). This statement does not require any privilege. It requires only the ability to connect to the server.

System variable information is also available from these sources:

- Performance Schema tables. See Section 23.9.13, “Performance Schema System Variable Tables”.
- The GLOBAL_VARIABLES and SESSION_VARIABLES tables. See Section 22.10, “The INFORMATION_SCHEMA GLOBAL_VARIABLES and SESSION_VARIABLES Tables”.
- The **mysqladmin variables** command. See Section 5.5.2, “**mysqladmin** — Client for Administering a MySQL Server”.

For SHOW VARIABLES, a LIKE clause, if present, indicates which variable names to match. A WHERE clause can be given to select rows using more general conditions, as discussed in Section 22.32, “Extensions to SHOW Statements”.

SHOW VARIABLES accepts an optional GLOBAL or SESSION variable scope modifier:

- With a GLOBAL modifier, the statement displays global system variable values. These are the values used to initialize the corresponding session variables for new connections to MySQL. If a variable has no global value, no value is displayed.
- With a SESSION modifier, the statement displays the system varaible values that are in effect for the current connection. If a variable has no session value, the global value is displayed. LOCAL is a synonym for SESSION.

- If no modifier is present, the default is `SESSION`.

The scope for each system variable is listed at Section 6.1.4, “Server System Variables”.

`SHOW VARIABLES` is subject to a version-dependent display-width limit. For variables with very long values that are not completely displayed, use `SELECT` as a workaround. For example:

```
SELECT @@GLOBAL.innodb_data_file_path;
```

Most system variables can be set at server startup (read-only variables such as `version_comment` are exceptions). Many can be changed at runtime with the `SET` statement. See Section 6.1.5, “Using System Variables”, and Section 14.7.4, “SET Syntax”.

Partial output is shown here. The list of names and values may differ for your server. Section 6.1.4, “Server System Variables”, describes the meaning of each variable, and Section 9.12.2, “Tuning Server Parameters”, provides information about tuning them.

```
mysql> SHOW VARIABLES;
+-----------------------------------------+----------------------------+
| Variable_name                           | Value                      |
+-----------------------------------------+----------------------------+
| auto_increment_increment                | 1                          |
| auto_increment_offset                   | 1                          |
| autocommit                              | ON                         |
| automatic_sp_privileges                 | ON                         |
| back_log                                | 50                         |
| basedir                                 | /home/jon/bin/mysql-5.5    |
| big_tables                              | OFF                        |
| binlog_cache_size                       | 32768                      |
| binlog_direct_non_transactional_updates | OFF                        |
| binlog_format                           | STATEMENT                  |
| binlog_stmt_cache_size                  | 32768                      |
| bulk_insert_buffer_size                 | 8388608                    |
...
| max_allowed_packet                      | 4194304                    |
| max_binlog_cache_size                   | 18446744073709547520       |
| max_binlog_size                         | 1073741824                 |
| max_binlog_stmt_cache_size              | 18446744073709547520       |
| max_connect_errors                      | 100                        |
| max_connections                         | 151                        |
| max_delayed_threads                     | 20                         |
| max_error_count                         | 64                         |
| max_heap_table_size                     | 16777216                   |
| max_insert_delayed_threads              | 20                         |
| max_join_size                           | 18446744073709551615       |
...
```

```
| thread_handling                         | one-thread-per-connection |
| thread_stack                            | 262144                    |
| time_format                             | %H:%i:%s                  |
| time_zone                               | SYSTEM                    |
| timestamp                               | 1316689732                |
| tmp_table_size                          | 16777216                  |
| tmpdir                                  | /tmp                      |
| transaction_alloc_block_size            | 8192                      |
| transaction_prealloc_size               | 4096                      |
| tx_isolation                            | REPEATABLE-READ           |
| unique_checks                           | ON                        |
| updatable_views_with_limit              | YES                       |
| version                                 | 5.5.17-log                |
| version_comment                         | Source distribution       |
| version_compile_machine                 | x86_64                    |
| version_compile_os                      | Linux                     |
| wait_timeout                            | 28800                     |
| warning_count                           | 0                         |
+-----------------------------------------+---------------------------+
```

With a `LIKE` clause, the statement displays only rows for those variables with names that match the pattern. To obtain the row for a specific variable, use a `LIKE` clause as shown:

```
SHOW VARIABLES LIKE 'max_join_size';
SHOW SESSION VARIABLES LIKE 'max_join_size';
```

To get a list of variables whose name match a pattern, use the "`%`" wildcard character in a `LIKE` clause:

```
SHOW VARIABLES LIKE '%size%';
SHOW GLOBAL VARIABLES LIKE '%size%';
```

Wildcard characters can be used in any position within the pattern to be matched. Strictly speaking, because "_" is a wildcard that matches any single character, you should escape it as "`\_`" to match it literally. In practice, this is rarely necessary.