

- `NO_ZERO_DATE`

The `NO_ZERO_DATE` mode affects whether the server permits `'0000-00-00'` as a valid date. Its effect also depends on whether strict SQL mode is enabled.

- If this mode is not enabled, `'0000-00-00'` is permitted and inserts produce no warning.
- If this mode is enabled, `'0000-00-00'` is permitted and inserts produce a warning.
- If this mode and strict mode are enabled, `'0000-00-00'` is not permitted and inserts produce an error, unless `IGNORE` is given as well. For `INSERT IGNORE` and `UPDATE IGNORE`, `'0000-00-00'` is permitted and inserts produce a warning.

`NO_ZERO_DATE` is deprecated. `NO_ZERO_DATE` is not part of strict mode, but should be used in conjunction with strict mode and is enabled by default. A warning occurs if `NO_ZERO_DATE` is enabled without also enabling strict mode or vice versa.

Because `NO_ZERO_DATE` is deprecated, you should expect it to be removed in a future MySQL release as a separate mode name and its effect included in the effects of strict SQL mode.

- `NO_ZERO_IN_DATE`

The `NO_ZERO_IN_DATE` mode affects whether the server permits dates in which the year part is nonzero but the month or day part is 0. (This mode affects dates such as `'2010-00-01'` or `'2010-01-00'`, but not `'0000-00-00'`. To control whether the server permits `'0000-00-00'`, use the `NO_ZERO_DATE` mode.) The effect of `NO_ZERO_IN_DATE` also depends on whether strict SQL mode is enabled.

- If this mode is not enabled, dates with zero parts are permitted and inserts produce no warning.
- If this mode is enabled, dates with zero parts are inserted as `'0000-00-00'` and produce a warning.
- If this mode and strict mode are enabled, dates with zero parts are not permitted and inserts produce an error, unless `IGNORE` is given as well. For `INSERT IGNORE` and `UPDATE IGNORE`, dates with zero parts are inserted as `'0000-00-00'` and produce a warning.

`NO_ZERO_IN_DATE` is deprecated. `NO_ZERO_IN_DATE` is not part of strict mode, but should be used in conjunction with strict mode and is enabled by default. A warning occurs if `NO_ZERO_IN_DATE` is enabled without also enabling strict mode or vice versa.

Because `NO_ZERO_IN_DATE` is deprecated, you should expect it to be removed in a future MySQL release as a separate mode name and its effect included in the effects of strict SQL mode.

- `ONLY_FULL_GROUP_BY`

Reject queries for which the select list, `HAVING` condition, or `ORDER BY` list refer to nonaggregated columns that are neither named in the `GROUP BY` clause nor are functionally dependent on (uniquely determined by) `GROUP BY` columns.

A MySQL extension to standard SQL permits references in the `HAVING` clause to aliased expressions in the select list. The `HAVING` clause can refer to aliases regardless of whether `ONLY_FULL_GROUP_BY` is enabled.

For additional discussion and examples, see [Section 12.20.3, “MySQL Handling of GROUP BY”](#).