

12.18 JSON Functions	2344
12.18.1 JSON Function Reference	2344
12.18.2 Functions That Create JSON Values	2346
12.18.3 Functions That Search JSON Values	2347
12.18.4 Functions That Modify JSON Values	2363
12.18.5 Functions That Return JSON Value Attributes	2373
12.18.6 JSON Table Functions	2376
12.18.7 JSON Schema Validation Functions	2381
12.18.8 JSON Utility Functions	2387
12.19 Functions Used with Global Transaction Identifiers (GTIDs)	2392
12.20 Aggregate Functions	2394
12.20.1 Aggregate Function Descriptions	2394
12.20.2 GROUP BY Modifiers	2405
12.20.3 MySQL Handling of GROUP BY	2411
12.20.4 Detection of Functional Dependence	2414
12.21 Window Functions	2417
12.21.1 Window Function Descriptions	2418
12.21.2 Window Function Concepts and Syntax	2424
12.21.3 Window Function Frame Specification	2428
12.21.4 Named Windows	2432
12.21.5 Window Function Restrictions	2433
12.22 Performance Schema Functions	2433
12.23 Internal Functions	2436
12.24 Miscellaneous Functions	2438
12.25 Precision Math	2454
12.25.1 Types of Numeric Values	2454
12.25.2 DECIMAL Data Type Characteristics	2455
12.25.3 Expression Handling	2456
12.25.4 Rounding Behavior	2458
12.25.5 Precision Math Examples	2459

Expressions can be used at several points in [SQL](#) statements, such as in the [ORDER BY](#) or [HAVING](#) clauses of [SELECT](#) statements, in the [WHERE](#) clause of a [SELECT](#), [DELETE](#), or [UPDATE](#) statement, or in [SET](#) statements. Expressions can be written using values from several sources, such as literal values, column values, [NULL](#), variables, built-in functions and operators, loadable functions, and stored functions (a type of stored object).

This chapter describes the built-in functions and operators that are permitted for writing expressions in MySQL. For information about loadable functions and stored functions, see [Section 5.7, “MySQL Server Loadable Functions”](#), and [Section 25.2, “Using Stored Routines”](#). For the rules describing how the server interprets references to different kinds of functions, see [Section 9.2.5, “Function Name Parsing and Resolution”](#).

An expression that contains [NULL](#) always produces a [NULL](#) value unless otherwise indicated in the documentation for a particular function or operator.



Note

By default, there must be no whitespace between a function name and the parenthesis following it. This helps the MySQL parser distinguish between function calls and references to tables or columns that happen to have the same name as a function. However, spaces around function arguments are permitted.

To tell the MySQL server to accept spaces after function names by starting it with the `--sql-mode=IGNORE_SPACE` option. (See [Section 5.1.11, “Server](#)