

Mysql-SQL&Function

[零 参考资料 3](#_Toc46763176)

[一 Functions and Operators 3](#_Toc46763177)

[1 概述 3](#_Toc46763178)

[2 Operators 3](#_Toc46763179)

[3 Control Flow Functions 4](#_Toc46763180)

[4 Numeric Functions and Operator 4](#_Toc46763181)

[5 Date and Time function 5](#_Toc46763182)

[6 String Function and operators 7](#_Toc46763183)

[7 Aggregate Functions 8](#_Toc46763184)

[8 Other Functions and Operators 9](#_Toc46763185)

[二 SQL Statements 12](#_Toc46763186)

[1 Data Definition Statements（DDL） 12](#_Toc46763187)

[1.1 ALTER DATABASE 12](#_Toc46763188)

[1.2 ALTER EVENT 12](#_Toc46763189)

[1.3 ALTER FUNCTION 12](#_Toc46763190)

[1.4 ALTER INSTANCE 12](#_Toc46763191)

[1.5 ALTER LOGFILE GROUP 12](#_Toc46763192)

[1.6 ALTER PROCEDURE 12](#_Toc46763193)

[1.7 ALTER SERVER 12](#_Toc46763194)

[1.8 ALTER TABLE 12](#_Toc46763195)

[1.9 ALTER TABLESPACE 12](#_Toc46763196)

[1.10 ALTER VIEW 12](#_Toc46763197)

[1.11 CREATE DATABASE 12](#_Toc46763198)

[1.12 CREATE EVENT 12](#_Toc46763199)

[1.13 CREATE FUNCTION 12](#_Toc46763200)

[1.14 CREATE INDEX 12](#_Toc46763201)

[1.15 CREATE LOGFILE GROUP 12](#_Toc46763202)

[1.16 CREATE PROCEDURE and CREATE FUNCTIONs 12](#_Toc46763203)

[1.17 CREATE SERVER 12](#_Toc46763204)

[1.18 CREATE TABLE 12](#_Toc46763205)

[1.19 CREATE TABLESPACE 12](#_Toc46763206)

[1.20 CREATE TRIGGER 12](#_Toc46763207)

[1.21 CREATE VIEW 12](#_Toc46763208)

[1.22 DROP DATABASE 12](#_Toc46763209)

[1.23 DROP EVENT 12](#_Toc46763210)

[1.24 DROP FUNCTION 12](#_Toc46763211)

[1.25 DROP INDEX 12](#_Toc46763212)

[1.26 DROP LOGFILE GROUP 12](#_Toc46763213)

[1.27 DROP PROCEDURE and DROP FUNCTIONs 12](#_Toc46763214)

[1.28 DROP SERVER 12](#_Toc46763215)

[1.29 DROP TABLE 12](#_Toc46763216)

[1.30 DROP TABLESPACE 12](#_Toc46763217)

[1.31 DROP TRIGGER 12](#_Toc46763218)

[1.32 DROP VIEW 13](#_Toc46763219)

[1.33 RENAME TABLE 13](#_Toc46763220)

[1.34 TRUNCATE TABLE 13](#_Toc46763221)

[2 Data Manipulation Statements（DML） 13](#_Toc46763222)

[2.1 INSERT 13](#_Toc46763223)

[2.2 UPDATE 13](#_Toc46763224)

[2.3 SELECT Statement 13](#_Toc46763225)

[2.4 Subqueries 13](#_Toc46763226)

[2.5 DELETE 13](#_Toc46763227)

[2.6 CALL 13](#_Toc46763228)

[2.7 DO 13](#_Toc46763229)

[2.8 HANDLER 13](#_Toc46763230)

[2.9 LOAD DATA 13](#_Toc46763231)

[2.10 LOAD XML 13](#_Toc46763232)

[2.11 REPLACE 13](#_Toc46763233)

[3 Transactional and Locking Statements 13](#_Toc46763234)

[3.1 Transctional 13](#_Toc46763235)

[3.2 XA Transactions 13](#_Toc46763236)

[3.3 LOCK TABLE 13](#_Toc46763237)

[4 Database Administration Statements 13](#_Toc46763238)

[4.1 SHOW 13](#_Toc46763239)

[4.2 ALTER USER 14](#_Toc46763240)

[4.3 CREATE USER 14](#_Toc46763241)

[4.4 DROP USER 14](#_Toc46763242)

[4.5 GRANT 14](#_Toc46763243)

[4.6 RENAME USER 14](#_Toc46763244)

[4.7 REVOKE 14](#_Toc46763245)

[4.8 SET PASSWORD 14](#_Toc46763246)

[4.9 ANALYZE TABLE 14](#_Toc46763247)

[4.10 CHECK TABLE 14](#_Toc46763248)

[4.11 CHECKSUM TABLE 14](#_Toc46763249)

[4.12 OPTIMIZE TABLE 14](#_Toc46763250)

[4.13 REPAIR TABLE 14](#_Toc46763251)

[4.14 SET Variable 14](#_Toc46763252)

[4.15 Other 14](#_Toc46763253)

[5 Utility Statements 15](#_Toc46763254)

[5.1 EXPLAIN 15](#_Toc46763255)

[5.2 Other 15](#_Toc46763256)

[6 Replication Statements 15](#_Toc46763257)

[7 Compound Statements 15](#_Toc46763258)

[8 Prepared Statements 15](#_Toc46763259)

# 零 参考资料

《高性能Mysql 第三版》

<https://dev.mysql.com/doc/refman/5.7/en/functions.html>

<https://dev.mysql.com/doc/refman/5.7/en/sql-statements.html>

# Functions and Operators

## 概述

## 运算符

|  |  |
| --- | --- |
| 运算符 | 描述 |
| & | Bitwise AND |
| > | Greater than operator |
| >> | Right shift |
| >= | Greater than or equal operator |
| < | Less than operator |
| <>, != | Not equal operator |
| << | Left shift |
| <= | Less than or equal operator |
| <=> | NULL-safe equal to operator |
| %, MOD | Modulo operator |
| \* | Multiplication operator |
| + | Addition operator |
| - | Minus operator |
| - | Change the sign of the argument |
| -> | Return value from JSON column after evaluating path; equivalent to JSON\_EXTRACT(). |
| ->> (introduced 5.7.13) | Return value from JSON column after evaluating path and unquoting the result; equivalent to JSON\_UNQUOTE(JSON\_EXTRACT()). |
| / | Division operator |
| := | Assign a value |
| = | Assign a value (as part of a SET statement, or as part of the SET clause in an UPDATE statement) |
| = | Equal operator |
| ^ | Bitwise XOR |
| AND, && | Logical AND |
| BETWEEN ... AND ... | Whether a value is within a range of values |
| BINARY | Cast a string to a binary string |
| CASE | Case operator |
| DIV | Integer division |
| IN() | Whether a value is within a set of values |
| IS | Test a value against a boolean |
| IS NOT | Test a value against a boolean |
| IS NOT NULL | NOT NULL value test |
| IS NULL | NULL value test |
| LIKE | Simple pattern matching |
| NOT, ! | Negates value |
| NOT BETWEEN ... AND ... | Whether a value is not within a range of values |
| NOT IN() | Whether a value is not within a set of values |
| NOT LIKE | Negation of simple pattern matching |
| NOT REGEXP | Negation of REGEXP |
| OR, || | Logical OR |
| REGEXP | Whether string matches regular expression |
| RLIKE | Whether string matches regular expression |
| SOUNDS LIKE | Compare sounds |
| XOR | Logical XOR |
| | | Bitwise OR |
| ~ | Bitwise inversion |

## Control Flow Functions

|  |  |
| --- | --- |
| **Name** | **Description** |
| CASE | Case operator |
| IF() | If/else construct |
| IFNULL() | Null if/else construct |
| NULLIF() | Return NULL if expr1 = expr2 |

## Numeric Functions and Operator

|  |  |
| --- | --- |
| Name | **Description** |
| %, MOD | Modulo operator |
| \* | Multiplication operator |
| + | Addition operator |
| - | Minus operator |
| - | Change the sign of the argument |
| / | Division operator |
| ABS() | Return the absolute value |
| ACOS() | Return the arc cosine |
| ASIN() | Return the arc sine |
| ATAN() | Return the arc tangent |
| ATAN2(), ATAN() | Return the arc tangent of the two arguments |
| CEIL() | Return the smallest integer value not less than the argument |
| CEILING() | Return the smallest integer value not less than the argument |
| CONV() | Convert numbers between different number bases |
| COS() | Return the cosine |
| COT() | Return the cotangent |
| CRC32() | Compute a cyclic redundancy check value |
| DEGREES() | Convert radians to degrees |
| DIV | Integer division |
| EXP() | Raise to the power of |
| FLOOR() | Return the largest integer value not greater than the argument |
| LN() | Return the natural logarithm of the argument |
| LOG() | Return the natural logarithm of the first argument |
| LOG10() | Return the base-10 logarithm of the argument |
| LOG2() | Return the base-2 logarithm of the argument |
| MOD() | Return the remainder |
| PI() | Return the value of pi |
| POW() | Return the argument raised to the specified power |
| POWER() | Return the argument raised to the specified power |
| RADIANS() | Return argument converted to radians |
| RAND() | Return a random floating-point value |
| ROUND() | Round the argument |
| SIGN() | Return the sign of the argument |
| SIN() | Return the sine of the argument |
| SQRT() | Return the square root of the argument |
| TAN() | Return the tangent of the argument |
| TRUNCATE() | Truncate to specified number of decimal places |

## Date and Time function

|  |  |
| --- | --- |
| Name | **Description** |
| ADDDATE() | Add time values (intervals) to a date value |
| ADDTIME() | Add time |
| CONVERT\_TZ() | Convert from one time zone to another |
| CURDATE() | Return the current date |
| CURRENT\_DATE(), CURRENT\_DATE | Synonyms for CURDATE() |
| CURRENT\_TIME(), CURRENT\_TIME | Synonyms for CURTIME() |
| CURRENT\_TIMESTAMP(), CURRENT\_TIMESTAMP | Synonyms for NOW() |
| CURTIME() | Return the current time |
| DATE() | Extract the date part of a date or datetime expression |
| DATE\_ADD() | Add time values (intervals) to a date value |
| DATE\_FORMAT() | Format date as specified |
| DATE\_SUB() | Subtract a time value (interval) from a date |
| DATEDIFF() | Subtract two dates |
| DAY() | Synonym for DAYOFMONTH() |
| DAYNAME() | Return the name of the weekday |
| DAYOFMONTH() | Return the day of the month (0-31) |
| DAYOFWEEK() | Return the weekday index of the argument |
| DAYOFYEAR() | Return the day of the year (1-366) |
| EXTRACT() | Extract part of a date |
| FROM\_DAYS() | Convert a day number to a date |
| FROM\_UNIXTIME() | Format Unix timestamp as a date |
| GET\_FORMAT() | Return a date format string |
| HOUR() | Extract the hour |
| LAST\_DAY | Return the last day of the month for the argument |
| LOCALTIME(), LOCALTIME | Synonym for NOW() |
| LOCALTIMESTAMP, LOCALTIMESTAMP() | Synonym for NOW() |
| MAKEDATE() | Create a date from the year and day of year |
| MAKETIME() | Create time from hour, minute, second |
| MICROSECOND() | Return the microseconds from argument |
| MINUTE() | Return the minute from the argument |
| MONTH() | Return the month from the date passed |
| MONTHNAME() | Return the name of the month |
| NOW() | Return the current date and time |
| PERIOD\_ADD() | Add a period to a year-month |
| PERIOD\_DIFF() | Return the number of months between periods |
| QUARTER() | Return the quarter from a date argument |
| SEC\_TO\_TIME() | Converts seconds to 'hh:mm:ss' format |
| SECOND() | Return the second (0-59) |
| STR\_TO\_DATE() | Convert a string to a date |
| SUBDATE() | Synonym for DATE\_SUB() when invoked with three arguments |
| SUBTIME() | Subtract times |
| SYSDATE() | Return the time at which the function executes |
| TIME() | Extract the time portion of the expression passed |
| TIME\_FORMAT() | Format as time |
| TIME\_TO\_SEC() | Return the argument converted to seconds |
| TIMEDIFF() | Subtract time |
| TIMESTAMP() | With a single argument, this function returns the date or datetime expression; with two arguments, the sum of the arguments |
| TIMESTAMPADD() | Add an interval to a datetime expression |
| TIMESTAMPDIFF() | Subtract an interval from a datetime expression |
| TO\_DAYS() | Return the date argument converted to days |
| TO\_SECONDS() | Return the date or datetime argument converted to seconds since Year 0 |
| UNIX\_TIMESTAMP() | Return a Unix timestamp |
| UTC\_DATE() | Return the current UTC date |
| UTC\_TIME() | Return the current UTC time |
| UTC\_TIMESTAMP() | Return the current UTC date and time |
| WEEK() | Return the week number |
| WEEKDAY() | Return the weekday index |
| WEEKOFYEAR() | Return the calendar week of the date (1-53) |
| YEAR() | Return the year |
| YEARWEEK() | Return the year and week |

## String Function and operators

|  |  |
| --- | --- |
| Name | **Description** |
| ASCII() | Return numeric value of left-most character |
| BIN() | Return a string containing binary representation of a number |
| BIT\_LENGTH() | Return length of argument in bits |
| CHAR() | Return the character for each integer passed |
| CHAR\_LENGTH() | Return number of characters in argument |
| CHARACTER\_LENGTH() | Synonym for CHAR\_LENGTH() |
| CONCAT() | Return concatenated string |
| CONCAT\_WS() | Return concatenate with separator |
| ELT() | Return string at index number |
| EXPORT\_SET() | Return a string such that for every bit set in the value bits, you get an on string and for every unset bit, you get an off string |
| FIELD() | Index (position) of first argument in subsequent arguments |
| FIND\_IN\_SET() | Index (position) of first argument within second argument |
| FORMAT() | Return a number formatted to specified number of decimal places |
| FROM\_BASE64() | Decode base64 encoded string and return result |
| HEX() | Hexadecimal representation of decimal or string value |
| INSERT() | Insert substring at specified position up to specified number of characters |
| INSTR() | Return the index of the first occurrence of substring |
| LCASE() | Synonym for LOWER() |
| LEFT() | Return the leftmost number of characters as specified |
| LENGTH() | Return the length of a string in bytes |
| LIKE | Simple pattern matching |
| LOAD\_FILE() | Load the named file |
| LOCATE() | Return the position of the first occurrence of substring |
| LOWER() | Return the argument in lowercase |
| LPAD() | Return the string argument, left-padded with the specified string |
| LTRIM() | Remove leading spaces |
| MAKE\_SET() | Return a set of comma-separated strings that have the corresponding bit in bits set |
| MATCH | Perform full-text search |
| MID() | Return a substring starting from the specified position |
| NOT LIKE | Negation of simple pattern matching |
| NOT REGEXP | Negation of REGEXP |
| OCT() | Return a string containing octal representation of a number |
| OCTET\_LENGTH() | Synonym for LENGTH() |
| ORD() | Return character code for leftmost character of the argument |
| POSITION() | Synonym for LOCATE() |
| QUOTE() | Escape the argument for use in an SQL statement |
| REGEXP | Whether string matches regular expression |
| REPEAT() | Repeat a string the specified number of times |
| REPLACE() | Replace occurrences of a specified string |
| REVERSE() | Reverse the characters in a string |
| RIGHT() | Return the specified rightmost number of characters |
| RLIKE | Whether string matches regular expression |
| RPAD() | Append string the specified number of times |
| RTRIM() | Remove trailing spaces |
| SOUNDEX() | Return a soundex string |
| SOUNDS LIKE | Compare sounds |
| SPACE() | Return a string of the specified number of spaces |
| STRCMP() | Compare two strings |
| SUBSTR() | Return the substring as specified |
| SUBSTRING() | Return the substring as specified |
| SUBSTRING\_INDEX() | Return a substring from a string before the specified number of occurrences of the delimiter |
| TO\_BASE64() | Return the argument converted to a base-64 string |
| TRIM() | Remove leading and trailing spaces |
| UCASE() | Synonym for UPPER() |
| UNHEX() | Return a string containing hex representation of a number |
| UPPER() | Convert to uppercase |
| WEIGHT\_STRING() | Return the weight string for a string |

## Aggregate Functions

|  |  |
| --- | --- |
| Name | Description |
| AVG() | Return the average value of the argument |
| BIT\_AND() | Return bitwise AND |
| BIT\_OR() | Return bitwise OR |
| BIT\_XOR() | Return bitwise XOR |
| COUNT() | Return a count of the number of rows returned |
| COUNT(DISTINCT) | Return the count of a number of different values |
| GROUP\_CONCAT() | Return a concatenated string |
| JSON\_ARRAYAGG() | Return result set as a single JSON array |
| JSON\_OBJECTAGG() | Return result set as a single JSON object |
| MAX() | Return the maximum value |
| MIN() | Return the minimum value |
| STD() | Return the population standard deviation |
| STDDEV() | Return the population standard deviation |
| STDDEV\_POP() | Return the population standard deviation |
| STDDEV\_SAMP() | Return the sample standard deviation |
| SUM() | Return the sum |
| VAR\_POP() | Return the population standard variance |
| VAR\_SAMP() | Return the sample variance |
| VARIANCE() | Return the population standard variance |

## Other Functions and Operators

|  |  |
| --- | --- |
| Name | **Description** |
| BINARY | Cast a string to a binary string |
| CAST() | Cast a value as a certain type |
| CONVERT() | Cast a value as a certain type |
| ExtractValue() | Extract a value from an XML string using XPath notation |
| UpdateXML() | Return replaced XML fragment |
| & | Bitwise AND |
| >> | Right shift |
| << | Left shift |
| ^ | Bitwise XOR |
| BIT\_COUNT() | Return the number of bits that are set |
| | | Bitwise OR |
| ~ | Bitwise inversion |
| AES\_DECRYPT() | Decrypt using AES |
| AES\_ENCRYPT() | Encrypt using AES |
| ASYMMETRIC\_DECRYPT() | Decrypt ciphertext using private or public key |
| ASYMMETRIC\_DERIVE() | Derive symmetric key from asymmetric keys |
| ASYMMETRIC\_ENCRYPT() | Encrypt cleartext using private or public key |
| ASYMMETRIC\_SIGN() | Generate signature from digest |
| ASYMMETRIC\_VERIFY() | Verify that signature matches digest |
| COMPRESS() | Return result as a binary string |
| CREATE\_ASYMMETRIC\_PRIV\_KEY() | Create private key |
| CREATE\_ASYMMETRIC\_PUB\_KEY() | Create public key |
| CREATE\_DH\_PARAMETERS() | Generate shared DH secret |
| CREATE\_DIGEST() | Generate digest from string |
| DECODE() (deprecated) | Decode a string encrypted using ENCODE() |
| DES\_DECRYPT() (deprecated) | Decrypt a string |
| DES\_ENCRYPT() (deprecated) | Encrypt a string |
| ENCODE() (deprecated) | Encode a string |
| ENCRYPT() (deprecated) | Encrypt a string |
| MD5() | Calculate MD5 checksum |
| PASSWORD() (deprecated) | Calculate and return a password string |
| RANDOM\_BYTES() | Return a random byte vector |
| SHA1(), SHA() | Calculate an SHA-1 160-bit checksum |
| SHA2() | Calculate an SHA-2 checksum |
| UNCOMPRESS() | Uncompress a string compressed |
| UNCOMPRESSED\_LENGTH() | Return the length of a string before compression |
| VALIDATE\_PASSWORD\_STRENGTH() | Determine strength of password |
| GET\_LOCK() | Get a named lock |
| IS\_FREE\_LOCK() | Whether the named lock is free |
| IS\_USED\_LOCK() | Whether the named lock is in use; return connection identifier if true |
| RELEASE\_ALL\_LOCKS() | Release all current named locks |
| RELEASE\_LOCK() | Release the named lock |
| BENCHMARK() | Repeatedly execute an expression |
| CHARSET() | Return the character set of the argument |
| COERCIBILITY() | Return the collation coercibility value of the string argument |
| COLLATION() | Return the collation of the string argument |
| CONNECTION\_ID() | Return the connection ID (thread ID) for the connection |
| CURRENT\_USER(), CURRENT\_USER | The authenticated user name and host name |
| DATABASE() | Return the default (current) database name |
| FOUND\_ROWS() | For a SELECT with a LIMIT clause, the number of rows that would be returned were there no LIMIT clause |
| LAST\_INSERT\_ID() | Value of the AUTOINCREMENT column for the last INSERT |
| ROW\_COUNT() | The number of rows updated |
| SCHEMA() | Synonym for DATABASE() |
| SESSION\_USER() | Synonym for USER() |
| SYSTEM\_USER() | Synonym for USER() |
| USER() | The user name and host name provided by the client |
| VERSION() | Return a string that indicates the MySQL server version |
| ANY\_VALUE() | Suppress ONLY\_FULL\_GROUP\_BY value rejection |
| DEFAULT() | Return the default value for a table column |
| INET\_ATON() | Return the numeric value of an IP address |
| INET\_NTOA() | Return the IP address from a numeric value |
| INET6\_ATON() | Return the numeric value of an IPv6 address |
| INET6\_NTOA() | Return the IPv6 address from a numeric value |
| IS\_IPV4() | Whether argument is an IPv4 address |
| IS\_IPV4\_COMPAT() | Whether argument is an IPv4-compatible address |
| IS\_IPV4\_MAPPED() | Whether argument is an IPv4-mapped address |
| IS\_IPV6() | Whether argument is an IPv6 address |
| MASTER\_POS\_WAIT() | Block until the slave has read and applied all updates up to the specified position |
| NAME\_CONST() | Cause the column to have the given name |
| SLEEP() | Sleep for a number of seconds |
| UUID() | Return a Universal Unique Identifier (UUID) |
| UUID\_SHORT() | Return an integer-valued universal identifier |
| VALUES() | Define the values to be used during an INSERT |

# SQL Statements

## Data Definition Statements（DDL）

### ALTER DATABASE

### ALTER EVENT

### ALTER FUNCTION

### ALTER INSTANCE

### ALTER LOGFILE GROUP

### ALTER PROCEDURE

### ALTER SERVER

### ALTER TABLE

### ALTER TABLESPACE

### ALTER VIEW

### CREATE DATABASE

### CREATE EVENT

### CREATE FUNCTION

### CREATE INDEX

### CREATE LOGFILE GROUP

### CREATE PROCEDURE and CREATE FUNCTIONs

### CREATE SERVER

### CREATE TABLE

### CREATE TABLESPACE

### CREATE TRIGGER

### CREATE VIEW

### DROP DATABASE

### DROP EVENT

### DROP FUNCTION

### DROP INDEX

### DROP LOGFILE GROUP

### DROP PROCEDURE and DROP FUNCTIONs

### DROP SERVER

### DROP TABLE

### DROP TABLESPACE

### DROP TRIGGER

### DROP VIEW

### RENAME TABLE

### TRUNCATE TABLE

## Data Manipulation Statements（DML）

### INSERT

### UPDATE

### SELECT Statement

### Subqueries

### DELETE

### CALL

### DO

### HANDLER

### LOAD DATA

### LOAD XML

### REPLACE

## Transactional and Locking Statements

### Transctional

### XA Transactions

### LOCK TABLE

## Database Administration Statements

### SHOW

|  |  |
| --- | --- |
| 语句 | 说明 |
| SHOW BINARY LOGS |  |
| SHOW BINLOG EVENTS |  |
| SHOW CHARACTER SET |  |
| SHOW COLLATION |  |
| SHOW COLUMNS |  |
| SHOW CREATE DATABASE |  |
| SHOW CREATE EVENT |  |
| SHOW CREATE FUNCTION |  |
| SHOW CREATE PROCEDURE |  |
| SHOW CREATE TABLE |  |
| SHOW CREATE TRIGGER |  |
| SHOW CREATE USER |  |
| SHOW CREATE VIEW |  |
| SHOW DATABASES |  |
| SHOW ENGINE |  |
| SHOW ENGINES |  |
| SHOW ERRORS |  |
| SHOW EVENTS |  |
| SHOW FUNCTION CODE |  |
| SHOW FUNCTION STATUS |  |
| SHOW GRANTS |  |
| SHOW INDEX |  |
| SHOW MASTER STATUS |  |
| SHOW OPEN TABLES |  |
| SHOW PLUGINS |  |
| SHOW PRIVILEGES |  |
| SHOW PROCEDURE CODE |  |
| SHOW PROCEDURE STATUS |  |
| SHOW PROCESSLIST |  |
| SHOW PROFILE |  |
| SHOW PROFILES |  |
| SHOW RELAYLOG EVENTS |  |
| SHOW SLAVE HOSTS |  |
| SHOW SLAVE STATUS |  |
| SHOW STATUS |  |
| SHOW TABLE STATUS |  |
| SHOW TABLES |  |
| SHOW TRIGGERS |  |
| SHOW VARIABLES |  |
| SHOW WARNINGS |  |

### ALTER USER

### CREATE USER

### DROP USER

### GRANT

### RENAME USER

### REVOKE

### SET PASSWORD

### ANALYZE TABLE

### CHECK TABLE

### CHECKSUM TABLE

### OPTIMIZE TABLE

### REPAIR TABLE

### SET Variable

### Other

|  |  |
| --- | --- |
| 语句 | 说明 |
| CREATE FUNCTION |  |
| DROP FUNCTION |  |
| INSTALL PLUGIN |  |
| UNINSTALL PLUGIN |  |
| BINLOG |  |
| CACHE INDEX |  |
| FLUSH |  |

## Utility Statements

### EXPLAIN

### Other

|  |  |
| --- | --- |
| 描述 | 说明 |
| HELP |  |
| USE |  |

## Replication Statements

## Compound Statements

## Prepared Statements