1. String Functions

String functions manipulate or retrieve information about string data.

* Basic Operations:
  + ASCII(str) – Returns the ASCII value of the first character.
  + BIN(N) – Converts a number to binary.
  + CHAR(N,...) – Returns the character for each ASCII code.
  + CHAR\_LENGTH(str) / LENGTH(str) – Returns the string length.
  + CONCAT(str1, str2,...) – Concatenates strings.
  + CONCAT\_WS(separator, str1,...) – Concatenates with a separator.
  + FIELD(str, str1,...) – Returns the index of a string.
  + FIND\_IN\_SET(str, str\_list) – Finds the index of a string in a list.
  + FORMAT(X, D) – Formats a number with commas and decimals.
  + INSERT(str, pos, len, new\_str) – Replaces part of a string.
* Substrings and Searching:
  + LOCATE(substr, str, pos) – Finds the position of a substring.
  + POSITION(substr IN str) – Alias for LOCATE.
  + SUBSTRING(str, pos, len) / SUBSTR() – Extracts a substring.
  + LEFT(str, len) / RIGHT(str, len) – Extracts from left/right.
* Modification:
  + LOWER(str) / UPPER(str) – Changes case.
  + TRIM([remstr FROM] str) – Removes leading/trailing spaces.
  + REPLACE(str, from\_str, to\_str) – Replaces all instances.
  + REVERSE(str) – Reverses a string.
* Padding and Spaces:
  + RPAD(str, len, padstr) / LPAD() – Pads a string.
  + SPACE(N) – Returns a string of N spaces.
* Other:
  + HEX(str) / UNHEX() – Converts to/from hexadecimal.
  + QUOTE(str) – Escapes special characters.
  + ELT(N, str1,...) – Returns the N-th string.
  + MAKE\_SET(bits, str1,...) – Returns a comma-separated list of strings.
  + EXPORT\_SET(bits, on, off[, sep[, number\_of\_bits]]) – Converts bits to a string representation.

2. Numeric Functions

Numeric functions perform mathematical calculations.

* Basic Arithmetic:
  + ABS(X) – Absolute value.
  + CEIL(X) / CEILING() – Rounds up.
  + FLOOR(X) – Rounds down.
  + ROUND(X, D) – Rounds to D decimals.
  + TRUNCATE(X, D) – Truncates to D decimals.
* Exponents and Roots:
  + EXP(X) – Exponential of X.
  + LOG(X) – Natural logarithm.
  + LOG10(X) – Base-10 logarithm.
  + POWER(X, Y) / POW() – X raised to the power Y.
  + SQRT(X) – Square root.
* Random Numbers:
  + RAND([seed]) – Random number.
  + PI() – Returns π.
* Sign and Comparison:
  + SIGN(X) – Returns -1, 0, or 1 based on sign.
  + GREATEST(X1, X2,...) – Returns the largest value.
  + LEAST(X1, X2,...) – Returns the smallest value.
* Trigonometry:
  + SIN(X) / COS(X) / TAN(X) – Trigonometric functions.
  + ASIN(X) / ACOS(X) / ATAN(X) – Inverse trigonometric functions.
  + DEGREES(X) / RADIANS(X) – Converts between degrees and radians.

3. Date and Time Functions

Manipulate and retrieve date/time values.

* Current Date/Time:
  + NOW() / CURRENT\_TIMESTAMP() – Current date and time.
  + CURDATE() / CURRENT\_DATE() – Current date.
  + CURTIME() / CURRENT\_TIME() – Current time.
  + UTC\_TIMESTAMP() / UTC\_DATE() / UTC\_TIME() – UTC equivalents.
* Extracting Components:
  + YEAR(date) / MONTH() / DAY() – Extracts date parts.
  + HOUR(time) / MINUTE() / SECOND() – Extracts time parts.
  + DAYOFWEEK(date) – Returns the weekday (1=Sunday).
  + DAYOFYEAR(date) – Day of the year.
  + WEEK(date) / WEEKOFYEAR() – Week number.
* Date Arithmetic:
  + ADDDATE(date, interval) / SUBDATE() – Adds or subtracts intervals.
  + DATE\_ADD(date, interval) / DATE\_SUB() – Similar to above.
  + DATEDIFF(date1, date2) – Difference between two dates.
  + TIMESTAMPDIFF(unit, datetime1, datetime2) – Difference in specified units.
* Conversion and Formatting:
  + DATE\_FORMAT(date, format) – Formats a date.
  + STR\_TO\_DATE(str, format) – Converts a string to a date.
  + UNIX\_TIMESTAMP(date) – Converts date to timestamp.
  + FROM\_UNIXTIME(ts) – Converts timestamp to date.
  + SEC\_TO\_TIME(seconds) / TIME\_TO\_SEC(time) – Converts between time and seconds.

4. Aggregate Functions

Aggregate functions operate on groups of rows.

* AVG(column) – Average value.
* SUM(column) – Total value.
* COUNT(column) – Number of rows.
* MAX(column) / MIN(column) – Maximum and minimum values.
* GROUP\_CONCAT(column) – Concatenates values.

5. JSON Functions

Manipulate JSON data.

* JSON\_OBJECT(key, value,...) – Creates a JSON object.
* JSON\_ARRAY(value,...) – Creates a JSON array.
* JSON\_EXTRACT(json, path) – Extracts data from JSON.
* JSON\_SET(json, path, value) – Updates JSON.
* JSON\_REMOVE(json, path) – Removes JSON elements.
* JSON\_CONTAINS(json, value) – Checks if JSON contains a value.
* JSON\_ARRAYAGG(column) – Aggregates column values into a JSON array.

6. Control Flow Functions

Control the flow of query logic.

* IF(expr, true\_value, false\_value) – Conditional logic.
* CASE – Multi-condition branching.
* IFNULL(expr, value) – Returns value if expression is NULL.
* NULLIF(expr1, expr2) – Returns NULL if two values are equal.

7. Window Functions (MySQL 8.0+)

Perform calculations over rows in a result set.

* ROW\_NUMBER() – Sequential row number.
* RANK() – Rank of rows with gaps.
* DENSE\_RANK() – Rank of rows without gaps.
* NTILE(N) – Divides rows into N buckets.
* LEAD() / LAG() – Access rows before/after current row.

**10. Bitwise Functions**

Operate on bits in numbers.

* BIT\_COUNT(N) – Returns the number of bits set to 1 in the binary representation.
* BIT\_AND(expr) – Performs bitwise AND for all rows.
* BIT\_OR(expr) – Performs bitwise OR for all rows.
* BIT\_XOR(expr) – Performs bitwise XOR for all rows.

**11. Spatial (GIS) Functions**

MySQL supports spatial data types like GEOMETRY, POINT, LINESTRING, and provides functions to work with GIS data.

* **Geometry Constructors**:
  + ST\_GEOMFROMTEXT(wkt) – Creates a geometry from Well-Known Text (WKT).
  + ST\_POINT(x, y) – Creates a point geometry.
  + ST\_POLYGONFROMTEXT(wkt) – Creates a polygon from WKT.
  + ST\_LINESTRINGFROMTEXT(wkt) – Creates a line string from WKT.
* **Spatial Calculations**:
  + ST\_DISTANCE(g1, g2) – Computes the minimum distance between two geometries.
  + ST\_AREA(geometry) – Calculates the area of a polygon.
  + ST\_LENGTH(geometry) – Calculates the length of a geometry.
  + ST\_INTERSECTS(g1, g2) – Checks if two geometries intersect.
  + ST\_CONTAINS(g1, g2) – Checks if one geometry contains another.
* **Spatial Metadata**:
  + ST\_ASWKT(geometry) – Converts a geometry to WKT.
  + ST\_ASGEOJSON(geometry) – Converts a geometry to GeoJSON.

**12. Privilege and Security Functions**

Handle user authentication, privileges, and security.

* **Privilege Checks**:
  + HAS\_ROLE(role\_name) – Checks if the current user has a specific role.
  + IS\_ROLE\_ACTIVE(role\_name) – Checks if a role is active.
* **SSL and Authentication**:
  + SSL\_SESSION\_ID() – Returns the current SSL session ID.
  + USER() – Returns the authenticated user.

**13. Performance Schema Functions**

Retrieve performance metrics about server execution.

* **Query Profiling**:
  + FORMAT\_BYTES(N) – Formats byte values as human-readable strings.
  + FORMAT\_PICO\_TIME(N) – Formats time in a human-readable format.
  + TIMER\_WAIT() – Returns the time a thread waits.
* **Thread Functions**:
  + THREAD\_ID() – Returns the ID of the current thread.
  + SLEEP(seconds) – Pauses the session for a specified time.

**14. Miscellaneous Functions**

Some functions are specific to uncommon use cases or debugging.

* UUID() – Generates a globally unique identifier.
* UUID\_SHORT() – Generates a short, unique identifier.
* NAME\_CONST(name, value) – Returns a constant with the given name and value.
* BENCHMARK(count, expr) – Repeats an expression a specified number of times (used for testing performance).
* INET\_ATON(ip) / INET\_NTOA(num) – Converts IP addresses to/from numeric values.
* INET6\_ATON(ip) / INET6\_NTOA(num) – Converts IPv6 addresses to/from numeric values.
* IS\_FREE\_LOCK(lock\_name) – Checks if a named lock is free.
* MASTER\_POS\_WAIT(log\_name, log\_pos[, timeout]) – Waits until the replication master reaches a certain position.

**15. Replication Functions**

Used in master-slave or group replication configurations.

* GTID\_SUBSET(gtid\_set1, gtid\_set2) – Checks if one GTID set is a subset of another.
* GTID\_SUBTRACT(gtid\_set1, gtid\_set2) – Subtracts one GTID set from another.
* WAIT\_FOR\_EXECUTED\_GTID\_SET(gtid\_set, timeout) – Waits until the given GTID set is executed.
* WAIT\_UNTIL\_SQL\_THREAD\_AFTER\_GTIDS(gtid\_set) – Waits for SQL thread to execute specific GTIDs.

**16. Debugging Functions**

Help in debugging and testing queries.

* GET\_LOCK(name, timeout) – Acquires a named lock.
* RELEASE\_LOCK(name) – Releases a named lock.
* IS\_USED\_LOCK(name) – Checks if a named lock is in use.
* LAST\_INSERT\_ID() – Returns the last inserted auto-increment ID.

**17. Advanced JSON Functions**

Expanding JSON functionality further:

* JSON\_KEYS(json[, path]) – Returns the keys of a JSON object.
* JSON\_LENGTH(json[, path]) – Returns the length of a JSON array or object.
* JSON\_MERGE(json1, json2) – Merges JSON documents.
* JSON\_UNQUOTE(json\_path) – Unquotes a JSON string.
* JSON\_PRETTY(json) – Formats JSON in a readable way.

**18. Regular Expression Functions (MySQL 8.0+)**

Work with patterns in strings.

* REGEXP\_LIKE(str, pattern) – Checks if a string matches a pattern.
* REGEXP\_INSTR(str, pattern) – Returns the position of the first match.
* REGEXP\_SUBSTR(str, pattern) – Extracts the substring that matches a pattern.
* REGEXP\_REPLACE(str, pattern, replacement) – Replaces occurrences of a pattern.

**19. Window Functions (Analytical Functions) (MySQL 8.0+)**

Perform calculations across a window of rows.

* CUME\_DIST() – Cumulative distribution of a value.
* PERCENT\_RANK() – Returns the percentile rank.
* NTH\_VALUE(expr, N) – Returns the N-th value in a result set.