1. **String Function**(concat,lower,upper,replace, substr, length, char\_length)
2. **Math Function**(abs, ceil, floor, mod(n, m), round, exp, pow, sqrt,)
3. **Date Function**(curdate,now,sysdate,last\_day,date\_format, month,year,day , monthname , datediff, Date Time Function + Extract Clause)

Let me explain each section of the SQL code provided:

### **String Functions**

 SELECT CONCAT ('Hello', ' ', 'World') AS concatenated\_string;  
 SELECT LOWER ('HELLO') AS lower\_case;  
 SELECT UPPER ('hello') AS upper\_case;  
 SELECT REPLACE ('Hello World', 'World', 'Universe') AS replaced\_string;  
 SELECT   SUBSTR ('Hello World', 7) AS substring\_from\_position;  
 SELECT   LENGTH ('Hello World') AS length\_of\_string;  
 SELECT   CHAR\_LENGTH ('Hello World') AS character\_length\_of\_string;

* CONCAT: Concatenates two or more strings together.
* LOWER: Converts a string to lowercase.
* UPPER: Converts a string to uppercase.
* REPLACE: Replaces occurrences of a substring within a string with another substring.
* SUBSTR: Extracts a substring from a string starting from a specified position.
* LENGTH: Returns the length of a string in bytes.
* CHAR\_LENGTH: Returns the length of a string in characters. japanese\_string| length\_in\_bytes| length\_in\_characters |

japanese\_string| length\_in\_bytes| length\_in\_characters  |

+----------------+----------------+-----------------------+

| こんにちは    |             15 |                     5 |

<https://database.guide/mysql-length-vs-char_length/>

### **Math Functions**

SELECT  
   ABS(-10) AS absolute\_value,  
   CEIL(4.5) AS ceil\_value,  
   FLOOR(4.5) AS floor\_value,  
   MOD(10, 3) AS modulus,  
   ROUND(4.567, 2) AS rounded\_value  
   ,  
   EXP(1) AS exponential\_value,  
   POW(2, 3) AS power\_value,  
   SQRT(25) AS square\_root\_value;

* ABS: Returns the absolute value of a number.
* CEIL: Returns the smallest integer greater than or equal to a number.
* FLOOR: Returns the largest integer less than or equal to a number.
* MOD: Returns the remainder of a division operation.
* ROUND: Rounds a number to a specified number of decimal places.
* TRUNCATE: Truncates a number to a specified number of decimal places.
* EXP: Returns the value of e raised to the power of a number.
* POW: Returns a number raised to the power of another number.
* SQRT: Returns the square root of a number.

### **Date Time Functions with Extract Clause**

**SELECT  
    DATE\_ADD(NOW(), INTERVAL 1 HOUR) AS added\_one\_hour,  
    DATE\_SUB(NOW(), INTERVAL 1 DAY) AS subtracted\_one\_day,  
    EXTRACT(YEAR FROM NOW()) AS extracted\_year,  
    EXTRACT(MONTH FROM NOW()) AS extracted\_month,  
    EXTRACT(DAY FROM NOW()) AS extracted\_day,  
    EXTRACT(HOUR FROM NOW()) AS extracted\_hour,  
    EXTRACT(MINUTE FROM NOW()) AS extracted\_minute,  
    EXTRACT(SECOND FROM NOW()) AS extracted\_second,  
    DATEDIFF('2024-05-30', '2024-05-21') AS date\_difference;**

* DATE\_ADD: Adds a specified time interval to a date or datetime expression.
* DATE\_SUB: Subtracts a specified time interval from a date or datetime expression.
* EXTRACT: Extracts a specific component (year, month, day, hour, minute, second) from a date or datetime expression.
* DATEDIFF: Returns the difference in days between two dates.

=================================

Aggregate Function(avg, max,min,sum,count,)

GROUP BY clause with HAVING

concepts with examples:

### **Aggregate Functions (AVG, MAX, MIN, SUM, COUNT)**

Aggregate functions perform calculations on a set of values and return a single value as the result.

**SELECT  
    AVG(column\_name) AS average\_value,  
    MAX(column\_name) AS maximum\_value,  
    MIN(column\_name) AS minimum\_value,  
    SUM(column\_name) AS sum\_of\_values,  
    COUNT(column\_name) AS count\_of\_values  
FROM table\_name;**

### **GROUP BY Clause with HAVING**

The GROUP BY clause is used to group rows that have the same values into summary rows. The HAVING clause filters groups based on specified conditions.

SELECT  
    column1,  
    COUNT(\*) AS count\_per\_group  
FROM table\_name  
GROUP BY column1  
HAVING COUNT(\*) > 1; -- Example condition for HAVING clause

This query counts the occurrences of each value in column1, groups the rows by column1, and then filters out groups with a count greater than 1 using the HAVING clause.

These are fundamental SQL concepts that you can combine to perform various data manipulations and analysis tasks.