Summary – Day 12

User Defined Functions:

- Functions are blocks of codes to perform specific tasks and return the result. However, it is not mandatory for a function return anything and also a function is not limited to performing one task only. User defined functions are basic building blocks of a program and can be found in the basic structure of C program.
- Functions can be classified into two categories, namely, system-defined functions and user-defined functions. The functions which are developed by user at the time of writing a program are called user defined functions. Thus, user defined functions are functions developed by user.

Syntax:

```
DELIMITER $$

CREATE FUNCTION Function_Name(
    Parameter_1 DataType,
    Parameter_2 DataType,
    Parameter_n DataType,
)
RETURNS Return_Datatype
[NOT] DETERMINISTIC
BEGIN
        Function Body
        Return Return_Value
END $$

DELIMITER ;
```

- In this syntax:
    - First, specify the name of the stored function that you want to create after CREATE FUNCTION keywords.
    - Second, list all parameters of the stored function inside the parentheses followed by the function name. By default, all parameters are the IN parameters. You cannot specify IN , OUT or INOUT modifiers to parameters.
    - Third, specify the data type of the return value in the RETURNS statement, which can be any valid MySQL data types.
    - Fourth, specify if a function is deterministic or not using the DETERMINISTIC keyword.
    - A deterministic function always returns the same result for the same input parameters whereas a non-deterministic function returns different results for the same input parameters.
    - If you don't use DETERMINISTIC or NOT DETERMINISTIC, MySQL uses the NOT DETERMINISTIC option by default.
    - Fifth, write the code in the body of the stored function in the BEGIN END block. Inside the body section, you need to specify at least one RETURN statement. The RETURN statement returns a value to the calling programs. Whenever the RETURN statement is reached, the execution of the stored function is terminated immediately.

Window Functions:

- Windows function in MySQL helps to solve a query problem. The operation is performed on a set of query rows while keeping the number of rows intact.
- The window functions allow you to solve query problems in new, easier ways and with better performance.
- Aggregate functions summarize the data from multiple rows into a single result row, but window functions, which also operate on a subset of rows, do not reduce the number of rows returned by the query.

Syntax:

window_function (expression) OVER (*[partition_definition] [order_defenition]*)

Where

- window_funtion -the name of the window function
- expression – field on which function is performed
- partition_definition – constraint for partition
- order_definition – order of result set

Types:

ROW_NUMBER(): Used to insert the row numbers.

RANK(): Provides every row with rank but it is not always a consecutive number like Row_number()

DENSE_RANK(): Will assign consecutive rank numbers for each group.

PERCENT_RANK(): Calculates the percentile of a row within the small result set and will return a value ranging from 0 to 1 for every row.

CUME_DIST(): Returns a value that represents the number of rows with values less than or equal to (<=)the current row's value divided by the total number of rows.

NTILE(): Helps to split the result set rows into a specified number of groups based upon the partition_by and order_by clauses provided.

LAG(): Returns the value to the previous rows in a sorted and partitioned result set.

LEAD(): Will return the values ahead, in the partitioned and sorted result set.

FIRST_VALUE(): Returns the first row among a partitioned sorted result set.

LAST_VALUE(): Returns the last row among a partitioned sorted result set.

NTH VALUE(): Returns the Nth row among a partitioned sorted result set.