Keyword Based Queries in Mysql(18/07/2024)

1. Write SQL queries in MySQL for the following.

a. Write an SQL Query to find the year from date.

Query: SELECT YEAR('2023-12-19') AS Year;

```
+------+
| Year |
+------+
| 2023 |
+------+
```

b. Check whether date passed to Query is the date of a given format or not.

Query:
```
SELECT
    CASE
        WHEN STR_TO_DATE('2023-09-13','%Y-%m-%d') IS NOT NULL
            THEN 'Valid date format'
        ELSE 'Invalid date format'
    END AS Date_format_check;
```

```
+--------------------+
| Date_format_check  |
+--------------------+
| Valid date format  |
+--------------------+
```

c. Find the size of the SCHEMA/USER.

Query:
```
SELECT
    table_schema AS "DB Name",
     ROUND(SUM(data_length + index_length) / 1024 / 1024, 1) AS "DB
Size"
FROM
     information_schema.tables
GROUP BY
    table_schema;
```

```
+--------------------+---------+
| DB Name            | DB Size |
+--------------------+---------+
| information_schema |     0.0 |
+--------------------+---------+
```

d. Display the current time.

Query: SELECT CURRENT_TIME;

```
+--------------+
| CURRENT_TIME |
+--------------+
| 11:38:24     |
+--------------+
```

e. Given a date, retrieve the next day's date.

Query: SELECT DATE_ADD('2023-10-31',INTERVAL 1 DAY) AS Next_day;

```
+------------+
| Next_day   |
+------------+
| 2023-11-01 |
+------------+
```

f. Get database's date.

Query: SELECT CURRENT_DATE();

```
+----------------+
| CURRENT_DATE() |
+----------------+
| 2024-07-26     |
+----------------+
```

g. Returns the default(current) database name.

Query: SELECT DATABASE() AS Default_Database;

```
+------------------------+
| Default_Database       |
+------------------------+
| sandbox_db |
+------------------------+
```

h. Retrieve the current MySQL user name and host name.

Query: SELECT USER() AS User_and_Host;

```
+-------------------------------------+
| User_and_Host                       |
+-------------------------------------+
| user_42ktpxq56_42maafvp8@172.21.0.8 |
+-------------------------------------+
```

i. Find the string that tells the MySQL server version.

Query: SELECT VERSION() AS mysql_version;

```
+---------------+
| mysql_version |
+---------------+
| 8.0.27        |
+---------------+
```

j. Perform Bitwise OR, Bitwise XOR and Bitwise AND.

Query: SELECT 5|3 AS Bitwise_OR , 5&3 AS Bitwise_AND , 5^3 AS Bitwise_XOR;

```
+------------+-------------+-------------+
| Bitwise_OR | Bitwise_AND | Bitwise_XOR |
+------------+-------------+-------------+
|          7 |           1 |           6 |
+------------+-------------+-------------+
```

k. Find the difference between two dates and print in terms of the number of days.

Query: SELECT DATEDIFF('2023-10-05','2023-10-01') AS Days_Difference;

```
+-----------------+
| Days_Difference |
+-----------------+
|               4 |
+-----------------+
```

l. Add one day to the current date.

Query: SELECT DATE(DATE_ADD(NOW(), INTERVAL 1 DAY)) AS New_Date;

```
+------------+
| New_Date   |
+------------+
| 2024-07-27 |
+------------+
```

m. Add two hours and 50 minutes to the current date and print the new date.

Query: SELECT DATE(DATE_ADD(DATE_ADD(NOW(), INTERVAL 2 HOUR), INTERVAL 50 MINUTE)) AS New_Date;

```
+------------+
| New_Date   |
+------------+
| 2024-07-26 |
+------------+
```

n. Find the floor and ceil values of a floating point number. Also operate on the power, log,
modulus, round off and truncate functions.

Query: SELECT
FLOOR(5.75) AS Floor_Value,
CEIL(5.75) AS Ceil_Value,
POWER(5.75, 2) AS Power_Value,
LOG(5.75) AS Log_Value,
MOD(5.75, 3) AS Modulus_Value,
ROUND(5.75) AS Round_Off_Value,
TRUNCATE(5.75, 1) AS Truncate_Value;

Floor_Value	Ceil_Value	Power_Value	Log_Value	Modulus_Value	Round_Off_Value	Truncate_Value
5	6	33.0625	1.749199854809259	2.75	6	5.7

o. Compare two strings and print the value ‘yes’ if they are equal, else print ‘no’.

Query: SELECT
CASE
WHEN 'string1' REGEXP '^string2$' THEN 'yes'
ELSE 'no'
END AS Comparison_Result;

Comparison_Result
no

p. Simulate the “IF... ELSE” construct in MySQL for a mark and grade setup.

Query: SELECT
85 AS marks,
CASE
WHEN 85 >= 90 THEN 'S'
WHEN 85 >= 80 THEN 'A'
WHEN 85 >= 70 THEN 'B'
WHEN 85 >= 50 THEN 'C'
WHEN 85 >= 35 THEN 'D'
ELSE 'F'
END AS Grade;

marks	Grade
85	A

q. Use IFNULL to check whether a mathematical expression gives a NULL value or not.

Query: SELECT
IFNULL(10 / (2 - 2), 'Result is NULL') AS Result;

```
+-----------------+
| Result          |
+-----------------+
| Result is NULL |
+-----------------+
```