

CSLR 51
DBMS LAB - SESSION 1

1. Write SQL queries in MySQL for the following.

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

```
SELECT YEAR('2021-10-15');
```

Output:

```
+-----+
| YEAR('2021-10-15') |
+-----+
|                2021 |
+-----+
```

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

```
SELECT DATE('2021-15-15');
```

Output:

```
+-----+
| DATE('2021-15-15') |
+-----+
| NULL                |
+-----+
```

c. Find the size of the SCHEMA/USER.

```
SELECT SUM(ROUND(((DATA_LENGTH + INDEX_LENGTH) / 1024 / 1024), 2)) AS "SIZE IN MB"
FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_SCHEMA = "sys";
```

Output:

```
+-----+
| SIZE_IN_MB |
+-----+
|          0.02 |
+-----+
```

d. Display the current time.

```
SELECT CURRENT_TIME;
```

Output:

```
+-----+
| CURRENT_TIME |
+-----+
| 10:26:46      |
+-----+
```

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

```
SELECT DATE_ADD('2021-10-15', interval 1 day);
```

Output:

```
+-----+
| DATE_ADD('2021-10-15', interval 1 day) |
+-----+
| 2021-10-16                             |
+-----+
```

f. Get database's date.

```
SELECT CURRENT_DATE;
```

Output:

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

g. Returns the default(current) database name.

```
SELECT DATABASE();
```

Output:

```
+-----+
| DATABASE()   |
+-----+
| sandbox_db   |
+-----+
```

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

`SELECT user, host, authentication_string, plugin from mysql.user;`

```
mysql> SELECT user, authentication_string FROM mysql.user;
+-----+-----+
| user          | authentication_string |
+-----+-----+
| debian-sys-maint | $A$005$0qAQ19x] 1q5," :4VmuaHo8Ch5q7MsFve4qGnt7J3SbKclcjQGZX231fZH/ |
| mysql.infoschema | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED |
| mysql.session    | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED |
| mysql.sys        | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED |
| root            |                       |
+-----+-----+
5 rows in set (0.00 sec)
```

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

`SELECT VERSION();`

Output:

```
+-----+
| VERSION() |
+-----+
| 8.0.27    |
+-----+
```

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

`SELECT 5|6 as ANS;`

`SELECT 5&6 as ANS;`

`SELECT 5^6 as ANS;`

Output:

```
+-----+
|  ANS  |
+-----+
|    7   |
+-----+
```

Output:

```
+-----+
|  ANS  |
+-----+
|    4   |
+-----+
```

Output:

```
+-----+
|  ANS  |
+-----+
|    3   |
+-----+
```

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

```
SELECT DATEDIFF('2020-06-25', '2020-06-15');
```

Output:

```
+-----+
| DATEDIFF('2020-06-25', '2020-06-15') |
+-----+
|                                10 |
+-----+
```

l. Add one day to the current date.

```
SELECT DATE_ADD('2021-10-15', interval 1 day);
```

Output:

```
+-----+
| DATE_ADD('2021-10-15', interval 1 day) |
+-----+
| 2021-10-16                             |
+-----+
```

m. Add two hours and 5000 minutes to the current date and print the new date.

```
SELECT DATE_ADD(DATE_ADD(NOW(), INTERVAL 2 HOUR), INTERVAL 5000 MINUTE) AS
new_date;
```

Output:

```
+-----+
| new_date          |
+-----+
| 2024-07-30 00:02:52 |
+-----+
```

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

```
SELECT floor(2.8) as floor;
```

```
SELECT ceil(2.8) as ceil;
```

```
SELECT POWER(3,2) AS POW;
```

```
SELECT LOG(2.81) AS LOG_VAL;
```

```
SELECT MOD(10,5) AS MOD_VAL;
```

```
SELECT ROUND(5.4) AS round_off_value;
```

```
SELECT TRUNCATE(10.8461561, 2) AS truncate_value;
```

Output:

```
+-----+
| floor |
+-----+
|      2 |
+-----+
```

Output:

```
+-----+
| ceil  |
+-----+
|      3 |
+-----+
```

Output:

```
+-----+
| POW   |
+-----+
|      9 |
+-----+
```

Output:

```
+-----+
| LOG_VAL
+-----+
| 1.0331844833456545 |
+-----+
```

Output:

```
+-----+
| MOD_VAL |
+-----+
|        0 |
+-----+
```

Output:

```
+-----+
| round_off_value |
+-----+
|                5 |
+-----+
```

Output:

```
+-----+
| truncate_value |
+-----+
|          10.84 |
+-----+
```

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

```
SELECT IF('hello' = 'hello', 'yes', 'no') AS result;
```

Output:

```
+-----+
| result |
+-----+
| yes    |
+-----+
```

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

```
CREATE TABLE students (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100),
    marks INT
);
INSERT INTO students (name, marks) VALUES
('Alice', 95),
('Bob', 85),
('Charlie', 75),
('David', 65),
('Eve', 55);
SELECT
    name,
    marks,
    IF(marks >= 90, 'A',
        IF(marks >= 80, 'B',
            IF(marks >= 70, 'C',
                IF(marks >= 60, 'D', 'F')
            )
        )
    ) AS grade
FROM
```

students;

Output:

name	marks	grade
Alice	95	A
Bob	85	B
Charlie	75	C
David	65	D
Eve	55	F

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

SELECT IFNULL(1/NULL, 'Expression is NULL') AS result;

```
mysql> SELECT IFNULL(1 / NULL, 'Expression is NULL') AS result;
+-----+
| result |
+-----+
| Expression is NULL |
+-----+
1 row in set (0.00 sec)
```
