Write SQL queries in MySQL for the following.

a. Write an SQL Query to find the year from date.
query : mysql> select year(current_date);
Output :

```
+-------------------+
| year(current_date) |
+-------------------+
|               2024 |
+-------------------+
```

b. Check whether date passed to Query is the date of a given format or not.
query :SELECT DATE('07/07/2024');
output:

```
+--------------------+
| DATE('07/07/2024') |
+--------------------+
| NULL               |
+--------------------+
```

c. Find the size of the SCHEMA/USER.
query :

```
SELECT SUM(DATA_LENGTH + INDEX_LENGTH) AS size
FROM information_schema.TABLES
WHERE TABLE_SCHEMA = 'mysql';
```

Output :

```
+---------+
| size |
+---------+
| 2752512 |
+---------+
```

d. Display the current time.
query :SELECT CURRENT_TIME();
Output :

```
+----------------+
| CURRENT_TIME() |
+----------------+
| 15:31:08       |
+----------------+
```

e. Given a date, retrieve the next day’s date.
query :SELECT DATE_ADD(current_date, INTERVAL 1 DAY) AS next_day_date;
Output :

```
+---------------+
| next_day_date |
+---------------+
| 2024-07-26    |
+---------------+
```

f. Get database’s date.

query :select current_date() as database_date;
Output :

database_date
2024-07-25

g. Returns the default(current) database name.
query :select database();
Output :

database()
university

h. Retrieve the current MySQL user name and host name.
query :SELECT USER() AS current_user_host;
Output :

current_user_host
root@localhost

i. Find the string that tells the MySQL server version.
query :SELECT VERSION();
Output :

VERSION()
8.0.37-0ubuntu0.22.04.3

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

```
query :SELECT 3 | 4 AS bitwise_or_result,
       3 ^ 4 AS bitwise_xor_result,
       3 & 4 AS bitwise_and_result;
```

Output :

bitwise_or_result	bitwise_xor_result	bitwise_and_result
7	7	0

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

```
query :SELECT DATEDIFF('2024-07-25', '2004-03-29') AS
day_difference;
```

Output :

day_difference

```
|           7423 |
+----------------+
```

l. Add one day to the current date.

```
query :SELECT DATE_ADD(CURDATE(), INTERVAL 1 DAY) AS next_day;
Output :
+------------+
| next_day   |
+------------+
| 2024-07-26 |
+------------+
```

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

```
query :SELECT DATE_ADD(Curdate(), INTERVAL '2:5000' HOUR_MINUTE) AS
new_date;
Output :
+---------------------+
| new_date            |
+---------------------+
| 2024-07-28 13:20:00 |
+---------------------+
```

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(3.45) AS floored_value,
    CEIL(4.56) AS ceiled_value,
    POWER(2,5) AS power_value,
    LOG(2) AS natural_log_value,
    10 % 7 AS modulus_result,
    ROUND(4.24673556, 2) AS rounded_value,
    TRUNCATE(4.5643567, 3) AS truncated_value;

Output :
+---------------+-------------+------------+-------------------
+----------------+---------------+-----------------+
| floored_value | ceiled_value | power_value | natural_log_value  |
modulus_result | rounded_value | truncated_value |
+---------------+-------------+------------+-------------------
+----------------+---------------+-----------------+
|             3 |           5 |          32 | 0.6931471805599453 |
3 |          4.25 |           4.564 |
+---------------+-------------+------------+-------------------
+----------------+---------------+-----------------+
```

o. In the first name of the employee, match the following using regular expressions.

```
query :SELECT
     CASE
         WHEN 'Charan' REGEXP '^c' THEN 'Name starts with c'
         ELSE 'Name does not start with c'
     END AS result;
```

Output :

```
+--------------------+
| result             |
+--------------------+
| Name starts with c |
+--------------------+
```

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

query :

```
SELECT CASE WHEN 'string1' = 'string2' THEN 'yes' ELSE 'no' END AS comparison_result;
```

Output :

```
+-------------------+
| comparison_result |
+-------------------+
| no                |
+-------------------+
```

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

query :

```
SELECT
    75 AS marks,
    CASE
        WHEN 75 >= 90 AND 75 <= 100 THEN 'A'
        WHEN 75 >= 80 AND 75 < 90 THEN 'B'
        WHEN 75 >= 70 AND 75 < 80 THEN 'C'
        WHEN 75 >= 60 AND 75 < 70 THEN 'D'
        WHEN 75 >= 0 AND 75 < 60 THEN 'F'
        ELSE 'Invalid marks'
    END AS grade;
```

Output :

```
+-------+-------+
| marks | grade |
+-------+-------+
|    75 | B     |
+-------+-------+
```

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

query :

```
SELECT IFNULL(10 / 5, 'Result is NULL') AS result;
```

Output :

```
+--------+
| result |
+--------+
| 2.0000 |
+--------+
```