

1. Write SQL queries in MySQL for the following.

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

```
mysql> select year('2024/07/18') as year;
+-----+
| year |
+-----+
| 2024 |
+-----+
1 row in set, 1 warning (0.00 sec)
```

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

```
mysql> select year('18/07/2024') as year;
+-----+
| year |
+-----+
| NULL |
+-----+
1 row in set, 1 warning (0.17 sec)
```

c. Find the size of the SCHEMA/USER.

```
mysql> SELECT table_schema AS `Database`, SUM(data_length + index_length) / 1024 / 1024 AS `Size
(MB)` FROM information_schema.tables WHERE table_schema = 'dev' GROUP BY table_schema;
+-----+-----+
| Database | Size (MB) |
+-----+-----+
| dev      | 0.06250000 |
+-----+-----+
1 row in set (0.15 sec)
```

d. Display the current time.

```
mysql> select time(now()) as time;
+-----+
| time   |
+-----+
| 23:10:11 |
+-----+
1 row in set (0.01 sec)
```

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

```
mysql> select date('2024/07/25')+1 as newdate;
+-----+
| newdate |
+-----+
| 20240726 |
+-----+
1 row in set, 1 warning (0.00 sec)
```

f. Get database's date.

```
mysql> select date(now()) as databasedate;
```

```
+-----+
| databasedate |
+-----+
| 2024-07-25   |
+-----+
1 row in set (0.00 sec)
```

g. Returns the default(current) database name.

```
mysql> select database();
+-----+
| database() |
+-----+
| dev        |
+-----+
1 row in set (0.00 sec)
```

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

```
mysql> select user();
+-----+
| user()      |
+-----+
| root@localhost |
+-----+
1 row in set (0.00 sec)
```

```
mysql> select @@hostname;
+-----+
| @@hostname |
+-----+
| ubuntulinux |
+-----+
1 row in set (0.01 sec)
```

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

```
mysql> select version();
+-----+
| version()      |
+-----+
| 8.0.37-0ubuntu0.22.04.3 |
+-----+
1 row in set (0.00 sec)
```

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

```
mysql> select 1^0 as value;
+-----+
| value |
+-----+
| 1     |
+-----+
1 row in set (0.00 sec)
```

```
mysql> select 1|0 as value;
+-----+
```

```
| value |
+-----+
| 1 |
+-----+
1 row in set (0.02 sec)
```

mysql> select 1&0 as value;

```
+-----+
| value |
+-----+
| 0 |
+-----+
1 row in set (0.00 sec)
```

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

mysql> select datediff('2024/07/25','2024/07/18') as difference;

```
+-----+
| difference |
+-----+
| 7 |
+-----+
1 row in set, 2 warnings (0.00 sec)
```

l. Add one day to the current date.

mysql> select curdate()+1 as date;

```
+-----+
| date |
+-----+
| 20240726 |
+-----+
1 row in set (0.00 sec)
```

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

mysql> select date(now())+ interval '2' hour+interval '5000' minute as newdate;

```
+-----+
| newdate |
+-----+
| 2024-07-28 13:20:00 |
+-----+
1 row in set (0.01 sec)
```

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

mysql> select floor(6.5) as value;

```
+-----+
| value |
+-----+
| 6 |
+-----+
1 row in set (0.00 sec)
```

mysql> select ceil(6.5) as value;

```
+-----+
```

```
| value |
+-----+
| 7 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> select power(4,2) as value;
+-----+
| value |
+-----+
| 16 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> select log(4,2) as value;
+-----+
| value |
+-----+
| 0.5 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> select mod(4,3) as value;
+-----+
| value |
+-----+
| 1 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> select round(5.44449,2) as value;
+-----+
| value |
+-----+
| 5.44 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> select truncate(33.22,1) as value;
+-----+
| value |
+-----+
| 33.2 |
+-----+
1 row in set (0.00 sec)
```

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

```
mysql> SELECT * FROM employee WHERE first_name REGEXP 'a';
+-----+-----+-----+
| id | first_name | last_name |
+-----+-----+-----+
| 1 | Alice | Smith |
| 3 | Charlie | Williams |
| 4 | David | Jones |
| 6 | Frank | Davis |
```

```
+----+-----+-----+
4 rows in set (0.00 sec)
```

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

```
mysql> select case when 'string1'='string2' then 'yes' else 'No' end as result;
```

```
+-----+
| result |
+-----+
| No     |
+-----+
```

```
1 row in set (0.01 sec)
```

```
mysql> select case when 'string1'='string1' then 'yes' else 'No' end as result;
```

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

```
1 row in set (0.01 sec)
```

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

```
mysql> SELECT id, name, marks, CASE WHEN marks > 90 THEN 'S' WHEN marks > 80 THEN 'A'
WHEN marks > 70 THEN 'B' WHEN marks > 60 THEN 'C' ELSE 'D' END AS grade FROM student;
```

```
+----+-----+-----+-----+
| id | name  | marks | grade |
+----+-----+-----+-----+
| 1  | Alice | 95    | S     |
| 2  | Bob   | 85    | A     |
| 3  | Charlie | 75    | B     |
| 4  | David | 65    | C     |
| 5  | Eve   | 55    | D     |
+----+-----+-----+-----+
```

```
5 rows in set (0.01 sec)
```

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

```
mysql> SELECT id, score1, score2, IFNULL((score1 + score2) / 2, 0) AS average_score FROM results;
```

```
+----+-----+-----+-----+
| id | score1 | score2 | average_score |
+----+-----+-----+-----+
| 1  | 85.50  | 90.00  | 87.750000    |
| 2  | 78.25  | NULL   | 0.000000     |
| 3  | NULL   | 88.75  | 0.000000     |
| 4  | NULL   | NULL   | 0.000000     |
+----+-----+-----+-----+
```

```
4 rows in set (0.00 sec)
```