Name: Reethikka.S.S

Roll_no-106122100

CSE-B

Keyword based Queries in MySQL

1. Write SQL queries in MySQL for the following.

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

```
mysql> select year('12-2-2004');
+-------------------+
| year('12-2-2004') |
+-------------------+
|              NULL |
+-------------------+
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 STR_TO_DATE("August,5,2017",'%e,%M,%Y') AS VALID;
+-------+
| VALID |
+-------+
| NULL  |
+-------+
```

1 row in set, 1 warning (0.00 sec)

```
mysql> SELECT STR_TO_DATE("5,August,2017",'%e,%M,%Y') AS VALID;
+------------+
| VALID      |
+------------+
| 2017-08-05 |
+------------+
1 row in set (0.00 sec)
```

c. Find the size of the SCHEMA/USER.

```
mysql> SELECT table_schema "DB Name",
    ->        ROUND(SUM(data_length + index_length) / 1024 / 1024, 1) "DB Size in MB"
    -> FROM information_schema.tables
    -> GROUP BY table_schema;
+--------------------+---------------+
| DB Name            | DB Size in MB |
+--------------------+---------------+
| mysql              |         2.6 |
| information_schema |         0.0 |
| performance_schema |         0.0 |
| sys                |         0.0 |
| DBMSLAB1           |         0.1 |
+--------------------+---------------+
5 rows in set (0.10 sec)
```

d. Display the current time.

```
mysql> SELECT CURTIME();
+-----------+
| CURTIME() |
+-----------+
| 13:05:28  |
+-----------+
1 row in set (0.00 sec)
```

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

```
mysql> SELECT ADDDATE('2003-12-2',INTERVAL 1 DAY);
+-------------------------------------+
| ADDDATE('2003-12-2',INTERVAL 1 DAY) |
+-------------------------------------+
| 2003-12-03                          |
+-------------------------------------+
1 row in set (0.00 sec)
```

f. Get database's date.

```
mysql> select curdate();
+------------+
```

| curdate() |

+------------+

| 2024-07-18 |

+------------+

1 row in set (0.00 sec)

g. Returns the default(current) database name.

mysql> SELECT DATABASE();

+------------+

| DATABASE() |

+------------+

| NULL |

+------------+

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 CURRENT_USER();
+----------------+
| CURRENT_USER() |
+----------------+
| root@localhost |
+----------------+
1 row in set (0.00 sec)
```

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

```
mysql> SELECT VERSION();
+-----------+
| VERSION() |
+-----------+
| 8.0.38    |
+-----------+
1 row in set (0.00 sec)

mysql> SELECT @@VERSION;
+-----------+
| @@VERSION |
+-----------+
| 8.0.38    |
+-----------+
1 row in set (0.00 sec)
```

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

```
mysql> SELECT 2|4;
+-----+
| 2|4 |
+-----+
|   6 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT 2^4;
+-----+
| 2^4 |
+-----+
|   6 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT 2&4;
+-----+
| 2&4 |
+-----+
|   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('2003-12-5','2003-12-2')+1;
+-----------------------------------+
| datediff('2003-12-2','2003-12-5') |
+-----------------------------------+
|                                 4 |
+-----------------------------------+
1 row in set (0.00 sec)
```

l. Add one day to the current date.

```
mysql> SELECT ADDDATE(CURDATE(),1);
+----------------------+
| ADDDATE(CURDATE(),1) |
+----------------------+
| 2024-07-26           |
+----------------------+
1 row in set (0.00 sec)
```

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

```
mysql> SELECT ADDTIME(NOW(),'2:50:00');
+--------------------------+
| ADDTIME(NOW(),'2:50:00') |
+--------------------------+
```

| 2024-07-25 16:04:54 |

+---------------------+

1 row in set (0.00 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 CEIL(2.33);

+------------+

| CEIL(2.33) |

+------------+

| 3 |

+------------+

1 row in set (0.00 sec)

mysql> SELECT FLOOR(2.33);

+-------------+

| FLOOR(2.33) |

+-------------+

| 2 |

+-------------+

1 row in set (0.00 sec)

mysql> SELECT POW(2,3);

+----------+

| POW(2,3) |

+----------+

```
|      8 |
+----------+
1 row in set (0.01 sec)
mysql> SELECT LOG(10);
+-------------------+
| LOG(10)         |
+-------------------+
| 2.302585092994046 |
+-------------------+
1 row in set (0.00 sec)

mysql> SELECT LOG10(10);
+-----------+
| LOG10(10) |
+-----------+
|       1 |
+-----------+
1 row in set (0.00 sec)
mysql> SELECT MOD(24,7);
+-----------+
| MOD(24,7) |
+-----------+
|       3 |
+-----------+
1 row in set (0.00 sec)
mysql> SELECT ROUND(1.298,1);
```

```
+----------------+
| ROUND(1.298,1) |
+----------------+
|            1.3 |
+----------------+
1 row in set (0.00 sec)
mysql> SELECT TRUNCATE(1.298,1);
+-------------------+
| TRUNCATE(1.298,1) |
+-------------------+
|               1.2 |
+-------------------+
1 row in set (0.00 sec)
```

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

```
mysql> CREATE TABLE EMPLOYEE(first_name varchar(15),last_name varchar(15),city
char(15));
Query OK, 0 rows affected (0.10 sec)

mysql> INSERT INTO EMPLOYEE VALUES('Rahul','Anand','Bangaluru');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO EMPLOYEE VALUES('Ria','Suresh','Hosur');
Query OK, 1 row affected (0.01 sec)
```

```
mysql> INSERT INTO EMPLOYEE VALUES('Dia','Khana','Hosur');
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM EMPLOYEE WHERE first_name REGEXP '^R';
+------------+-----------+-----------+
| first_name | last_name | city      |
+------------+-----------+-----------+
| Rahul      | Anand     | Bangaluru |
| Ria        | Suresh    | Hosur     |
+------------+-----------+-----------+
2 rows in set (0.01 sec)
```

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

```
mysql> SELECT IF(STRCMP('test','test1'),'no','yes');
+---------------------------------------+
| IF(STRCMP('test','test1'),'no','yes') |
+---------------------------------------+
| no                                    |
+---------------------------------------+
1 row in set (0.00 sec)
```

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

```
mysql> USE home_work
Database changed
```

```
mysql> DELIMITER //
mysql> CREATE PROCEDURE check_grade(score INT)
   -> BEGIN
   -> DECLARE grade VARCHAR(10);
   -> IF score>=90 THEN
   -> SET grade='A';
   -> ELSEIF score>=80 THEN
   -> SET grade='B';
   -> ELSEIF score>=70 THEN
   -> SET grade='C';
   -> ELSE
   -> SET grade='F';
   -> END IF;
   -> SELECT grade;
   -> END//
Query OK, 0 rows affected (0.05 sec)
mysql> DELIMITER ;
mysql> CALL check_grade(85);
+-------+
| grade |
+-------+
| B     |
+-------+
1 row in set (0.00 sec)
```

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

```
mysql> SELECT IFNULL(NULL,67+NULL);
+----------------------+
| IFNULL(NULL,67+NULL) |
+----------------------+
|                 NULL |
+----------------------+
1 row in set (0.00 sec)

mysql> SELECT IFNULL(6+7,NULL);
+------------------+
| IFNULL(6+7,NULL) |
+------------------+
|               13 |
+------------------+
1 row in set (0.00 sec)
```