1. Write SQL queries in MySQL for the following.

I created a database school to execute some of the queries given in the questions

```
 SELECT * FROM students;
+------------+------------+-----------+------------+---------------------
---+
| student_id | first_name | last_name | dob        | email
|
+------------+------------+-----------+------------+---------------------
---+
|          1 | John       | Doe       | 2000-05-15 | john.doe@example.com
|
|          2 | Jane       | Smith     | 1999-08-21 |
jane.smith@example.com |
+------------+------------+-----------+------------+---------------------
---+

SELECT * FROM courses;
+-----------+-----------------------------+-------------+---------+
| course_id | course_name                 | course_code | credits |
+-----------+-----------------------------+-------------+---------+
|         1 | Introduction to Programming | CS101       |       3 |
|         2 | Database Management         | CS201       |       4 |
+-----------+-----------------------------+-------------+---------+

SELECT * FROM enrollment;
+---------------+------------+-----------+-----------------+
| enrollment_id | student_id | course_id | enrollment_date |
+---------------+------------+-----------+-----------------+
|             1 |          1 |         1 | 2024-07-25      |
|             2 |          1 |         2 | 2024-07-26      |
|             3 |          2 |         1 | 2024-07-27      |
+---------------+------------+-----------+-----------------+
```

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

```
 SELECT YEAR('2017/08/25') AS Year;
 +------+
| Year |
+------+
| 2017 |
+------+
```

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

```
>        SELECT
    ->   CASE
    ->     WHEN STR_TO_DATE('2023-01-04', '%Y-%m-%d') IS NOT NULL THEN
'Valid date'
    ->     ELSE 'Invalid date'
    ->   END AS result;
+------------+
| result     |
+------------+
| Valid date |
+------------+
SELECT
    ->   CASE
    ->     WHEN STR_TO_DATE('2023-04-32', '%Y-%m-%d') IS NOT NULL THEN
'Valid date'
    ->     ELSE 'Invalid date'
    ->   END AS result;
+--------------+
| result       |
+--------------+
| Invalid date |
+--------------+
```

c. Find the size of the SCHEMA/USER.

```
SELECT SUM(DATA_LENGTH + INDEX_LENGTH) AS size
       FROM information_schema.TABLES
       WHERE TABLE_SCHEMA = 'mysql';
+---------+
| size    |
+---------+
| 2752512 |
+---------+
```

d. Display the current time.

```
SELECT NOW();
+---------------------+
| NOW()               |
+---------------------+
| 2024-07-25 20:20:54 |
+---------------------+
```

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

```
SELECT DATE_ADD('2022-07-25', INTERVAL 1 DAY) AS next_day;
+------------+
| next_day   |
+------------+
| 2022-07-26 |
+------------+
```

f. Get database’s date.

```
SELECT CURDATE() AS database_date;
+---------------+
| database_date |
+---------------+
| 2024-07-25    |
+---------------+
```

g. Returns the default(current) database name.

```
SELECT DATABASE() AS current_database;
+------------------+
| current_database |
+--------------------+
| school             |
+--------------------+
```

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

```
SELECT USER();
+----------------+
| USER()         |
+----------------+
| root@localhost |
+----------------+
```

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

```
SELECT VERSION() AS mysql_version;
+-------------------------+
| mysql_version           |
+-------------------------+
| 8.0.37-0ubuntu0.22.04.3 |
+-------------------------+
```

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

```
 SELECT
    ->   (5 | 3) AS bitwise_or,
    ->   (5 ^ 3) AS bitwise_xor,
    ->   (5 & 3) AS bitwise_and;
+------------+-------------+-------------+
| bitwise_or | bitwise_xor | bitwise_and |
+------------+-------------+-------------+
|          7 |           6 |           1 |
+------------+-------------+-------------+
```

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

```
SELECT DATEDIFF('2022-07-25', '2022-07-20') AS days_difference;
+-----------------+
| days_difference |
+-----------------+
|               5 |
+-----------------+
```

l. Add one day to the current date.

```
SELECT DATE_ADD(CURDATE(), INTERVAL 1 DAY) AS tomorrow;
+------------+
| tomorrow   |
+------------+
| 2024-07-26 |
+------------+
```

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

```
SELECT DATE_ADD(NOW(), INTERVAL '2:5000' HOUR_MINUTE) AS new_date;
+---------------------+
| new_date            |
+---------------------+
| 2024-07-29 10:22:16 |
+---------------------+
```

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(3.7) AS floor_value,
    ->   CEIL(3.7) AS ceil_value,
    ->   POWER(2, 3) AS power_value,
    ->   LOG(10) AS log_value,
    ->   MOD(17, 5) AS modulus_value,
    ->   ROUND(3.7) AS round_value,
    ->   TRUNCATE(3.7, 1) AS truncate_value;
+-------------+------------+-------------+--------------------+----------
----+-------------+----------------+
| floor_value | ceil_value | power_value | log_value          |
modulus_value | round_value | truncate_value |
+-------------+------------+-------------+--------------------+----------
----+-------------+----------------+
|           3 |          4 |           8 | 2.302585092994046 |
2 |           4 |            3.7 |
+-------------+------------+-------------+--------------------+----------
----+-------------+----------------+
```

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

```
SELECT *
    -> FROM students
    -> WHERE first_name REGEXP '^J';
+------------+------------+-----------+------------+---------------------
---+
| student_id | first_name | last_name | dob        | email
|
+------------+------------+-----------+------------+---------------------
---+
|          1 | John       | Doe       | 2000-05-15 | john.doe@example.com
|
|          2 | Jane       | Smith     | 1999-08-21 |
jane.smith@example.com |
+------------+------------+-----------+------------+---------------------
---+
```

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

```
 SELECT IF('apple' = 'apple', 'yes', 'no') AS comparison_result;
+-------------------+
| comparison_result |
+-------------------+
| yes               |
+-------------------+
```

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

```
> SELECT
    ->     CASE
    ->         WHEN marks >= 90 THEN 'A'
    ->         WHEN marks >= 80 THEN 'B'
    ->         WHEN marks >= 70 THEN 'C'
    ->         WHEN marks >= 60 THEN 'D'
    ->         ELSE 'F'
    ->     END AS grade
    -> FROM marks;
```

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

```
SELECT
    ->   IFNULL((10 / NULL), 'Expression is NULL') AS result,
    ->   IFNULL((10 / 2), 'Expression is NULL') AS result2;
+--------------------+---------+
| result             | result2 |
+--------------------+---------+
| Expression is NULL | 5.0000  |
+--------------------+---------+
```