PL EXERSICE 5

1. Write a stored function to take three parameters, the sides of a triangle. The sides of the triangle should be accepted from the user. The function should return a Boolean value:- *true* if the triangle is valid, *false* otherwise. A triangle is valid if the length of each side is less than the sum of the lengths of the other two sides. Check if the dimensions entered can form a valid triangle.

```
mysql> DELIMITER //
mysql>
mysql> CREATE FUNCTION is_valid_triangle(a DOUBLE, b DOUBLE, c DOUBLE)
    -> RETURNS BOOLEAN
    -> DETERMINISTIC
    -> BEGIN
    ->     DECLARE is_valid BOOLEAN;
    ->
    ->     IF (a + b > c) AND (a + c > b) AND (b + c > a) THEN
    ->         SET is_valid = TRUE;
    ->     ELSE
    ->         SET is_valid = FALSE;
    ->     END IF;
    ->
    ->     RETURN is_valid;
    -> END;
    -> //
Query OK, 0 rows affected (0.11 sec)

mysql>
mysql> DELIMITER ;
mysql>
mysql>
mysql> SELECT is_valid_triangle(3, 4, 5);   -- Returns 1 (TRUE)
+----------------------------+
| is_valid_triangle(3, 4, 5) |
+----------------------------+
|                          1 |
+----------------------------+
1 row in set (0.10 sec)

mysql> SELECT is_valid_triangle(1, 2, 3);   -- Returns 0 (FALSE)
+----------------------------+
| is_valid_triangle(1, 2, 3) |
+----------------------------+
|                          0 |
+----------------------------+
1 row in set (0.00 sec)

mysql> SELECT is_valid_triangle(10, 10, 25); -- Returns 0 (FALSE)
+-----------------------------+
| is_valid_triangle(10, 10, 25) |
+-----------------------------+
|                           0 |
+-----------------------------+
1 row in set (0.00 sec)
```

2. Write a function that generates a random number between 1 and 10. Use any logic of your choice to achieve this.

```
mysql> DELIMITER //
mysql>
mysql> CREATE FUNCTION random_number_1_to_10()
    -> RETURNS INT
    -> DETERMINISTIC
    -> BEGIN
    ->     RETURN FLOOR(1 + (RAND() * 10));
    -> END;
    -> //
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
mysql>
mysql> SELECT random_number_1_to_10();
+-------------------------+
| random_number_1_to_10() |
+-------------------------+
|                       9 |
+-------------------------+
1 row in set (0.00 sec)

mysql> SELECT random_number_1_to_10();
+-------------------------+
| random_number_1_to_10() |
+-------------------------+
|                       8 |
+-------------------------+
1 row in set (0.00 sec)

mysql> SELECT random_number_1_to_10();
+-------------------------+
| random_number_1_to_10() |
+-------------------------+
|                       4 |
+-------------------------+
1 row in set (0.00 sec)
```

3. Create a function that accepts a string of *n* characters and exchanges the first character with the last, the second with the next – to – last, and so forth until *n* exchanges have been made. What will the final string look like? Write the function to verify your conclusion.

```
mysql> DELIMITER //
mysql>
mysql> CREATE FUNCTION reverse_string_swap(input_str VARCHAR(255))
   -> RETURNS VARCHAR(255)
   -> DETERMINISTIC
   -> BEGIN
   ->     DECLARE len INT;
   ->     DECLARE i INT DEFAULT 1;
   ->     DECLARE result VARCHAR(255) DEFAULT '';
   ->
   ->     SET len = CHAR_LENGTH(input_str);
   ->
   ->     WHILE i <= len DO
   ->         SET result = CONCAT(SUBSTRING(input_str, i, 1), result);
   ->         SET i = i + 1;
   ->     END WHILE;
   ->
   ->     RETURN result;
   -> END;
   -> //
Query OK, 0 rows affected (1.57 sec)

mysql>
mysql> DELIMITER ;
mysql>
mysql> SELECT reverse_string_swap('MYSQL');
+------------------------------+
| reverse_string_swap('MYSQL') |
+------------------------------+
| LQSYM                        |
+------------------------------+
1 row in set (0.35 sec)
```