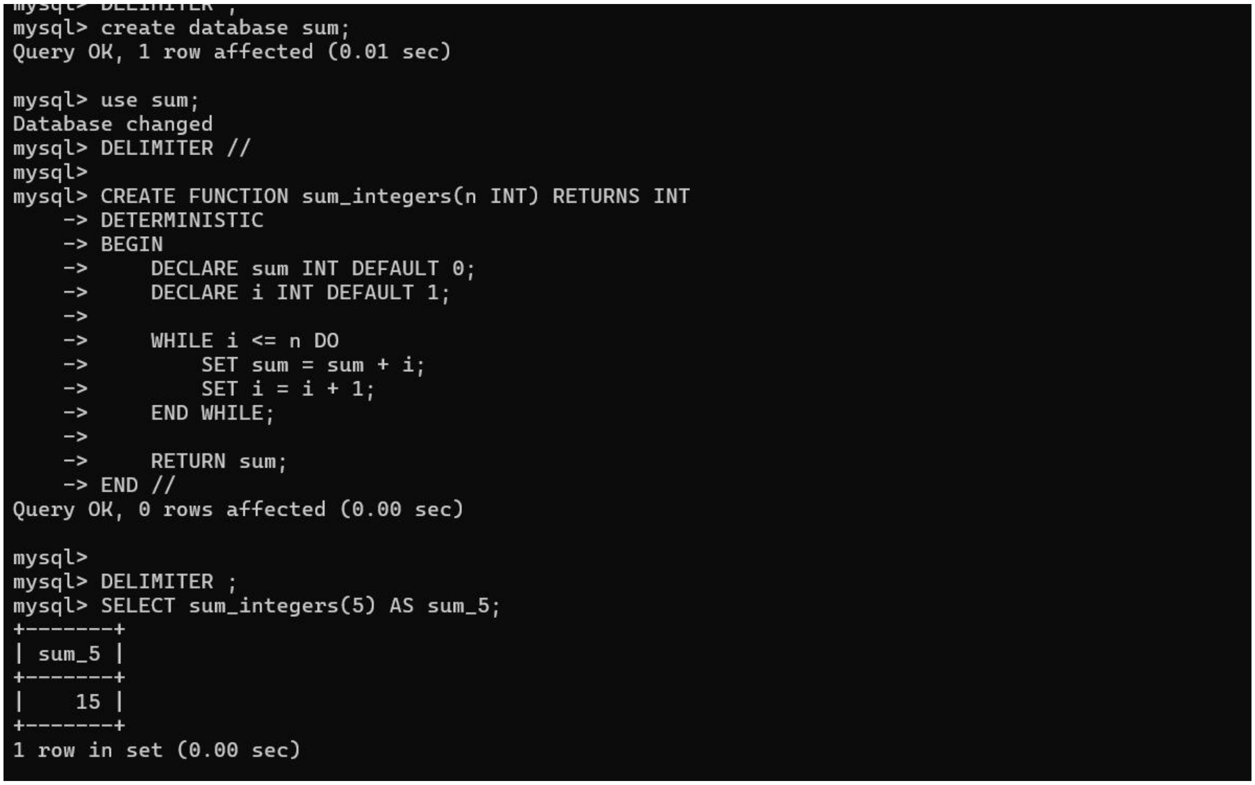
1. Write a recursive MySQL function to calculate the sum of all integers from 1 to a given number.

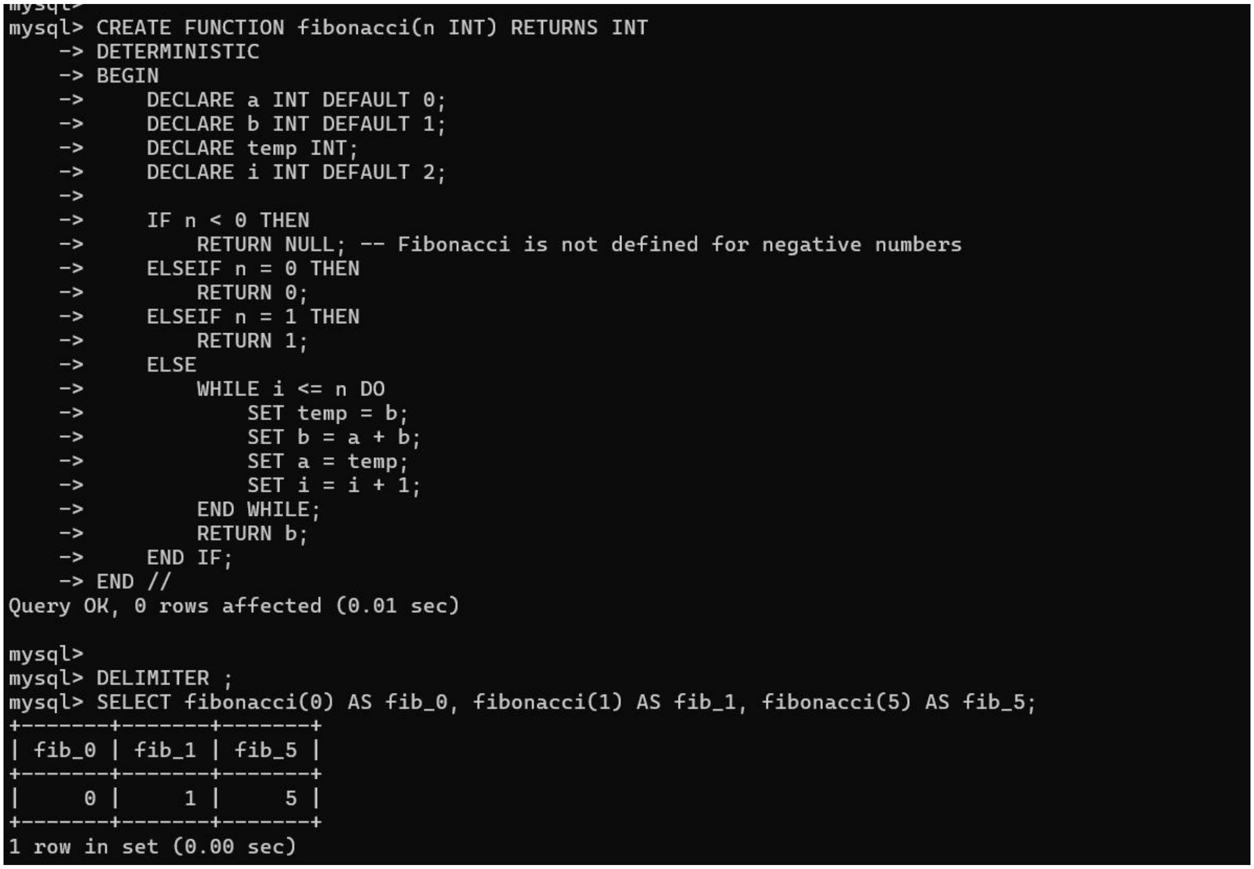


```
mysql> DELIMITER ;
mysql> create database sum;
Query OK, 1 row affected (0.01 sec)

mysql> use sum;
Database changed
mysql> DELIMITER //
mysql>
mysql> CREATE FUNCTION sum_integers(n INT) RETURNS INT
    -> DETERMINISTIC
    -> BEGIN
    ->     DECLARE sum INT DEFAULT 0;
    ->     DECLARE i INT DEFAULT 1;
    ->
    ->     WHILE i <= n DO
    ->         SET sum = sum + i;
    ->         SET i = i + 1;
    ->     END WHILE;
    ->
    ->     RETURN sum;
    -> END //
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> DELIMITER ;
mysql> SELECT sum_integers(5) AS sum_5;
+-------+
| sum_5 |
+-------+
|    15 |
+-------+
1 row in set (0.00 sec)
```

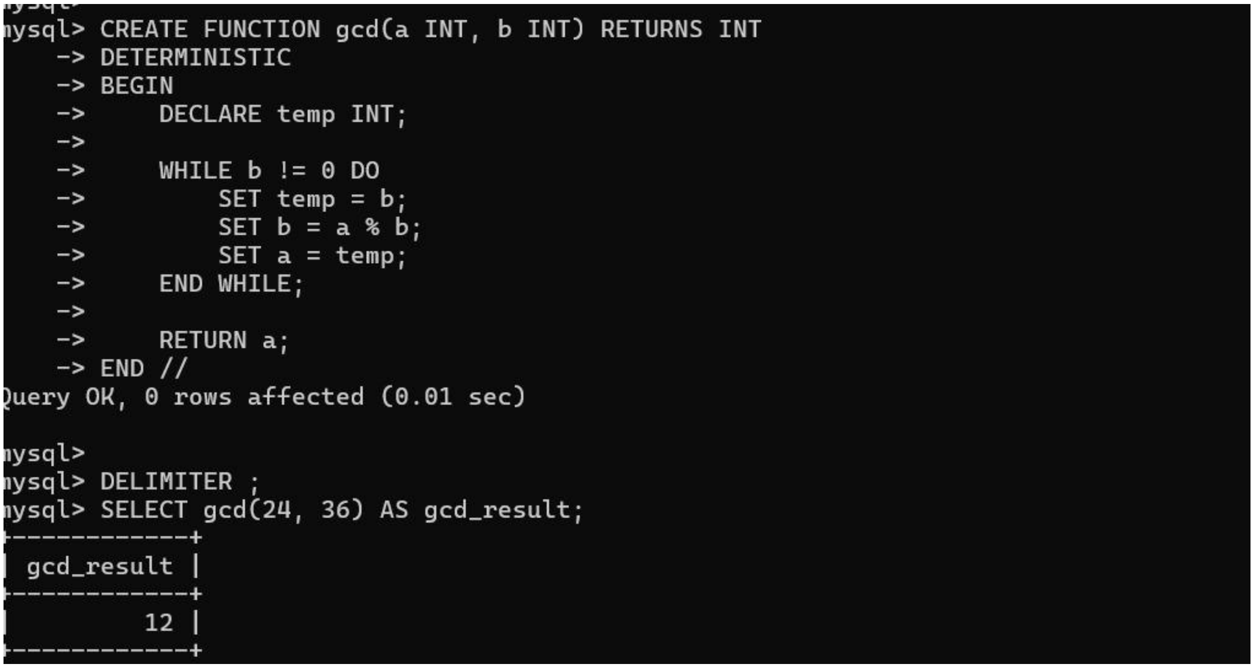
2. Create a recursive MySQL function to compute the nth Fibonacci number.



```
mysql>
mysql> CREATE FUNCTION fibonacci(n INT) RETURNS INT
    -> DETERMINISTIC
    -> BEGIN
    ->     DECLARE a INT DEFAULT 0;
    ->     DECLARE b INT DEFAULT 1;
    ->     DECLARE temp INT;
    ->     DECLARE i INT DEFAULT 2;
    ->
    ->     IF n < 0 THEN
    ->         RETURN NULL; -- Fibonacci is not defined for negative numbers
    ->     ELSEIF n = 0 THEN
    ->         RETURN 0;
    ->     ELSEIF n = 1 THEN
    ->         RETURN 1;
    ->     ELSE
    ->         WHILE i <= n DO
    ->             SET temp = b;
    ->             SET b = a + b;
    ->             SET a = temp;
    ->             SET i = i + 1;
    ->         END WHILE;
    ->         RETURN b;
    ->     END IF;
    -> END //
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
mysql> SELECT fibonacci(0) AS fib_0, fibonacci(1) AS fib_1, fibonacci(5) AS fib_5;
+-------+-------+-------+
| fib_0 | fib_1 | fib_5 |
+-------+-------+-------+
|     0 |     1 |     5 |
+-------+-------+-------+
1 row in set (0.00 sec)
```

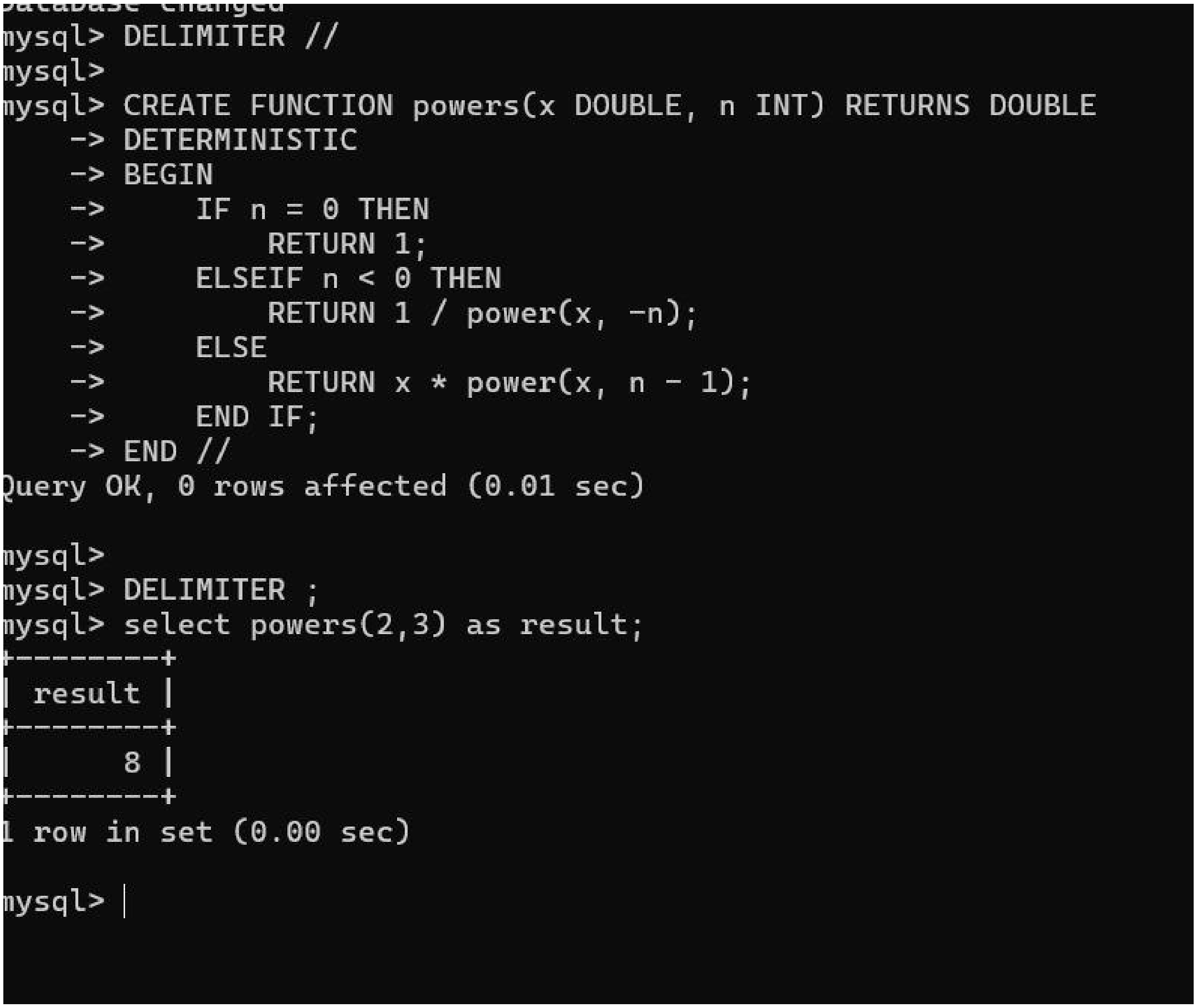
3. Develop a recursive MySQL function to compute the greatest common divisor (GCD) of two numbers.



```
mysql>
mysql> CREATE FUNCTION gcd(a INT, b INT) RETURNS INT
    -> DETERMINISTIC
    -> BEGIN
    ->     DECLARE temp INT;
    ->
    ->     WHILE b != 0 DO
    ->         SET temp = b;
    ->         SET b = a % b;
    ->         SET a = temp;
    ->     END WHILE;
    ->
    ->     RETURN a;
    -> END //
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
mysql> SELECT gcd(24, 36) AS gcd_result;
+------------+
| gcd_result |
+------------+
|         12 |
+------------+
```

4. Create a recursive MySQL function to determine the power of a number (base raised to exponent).

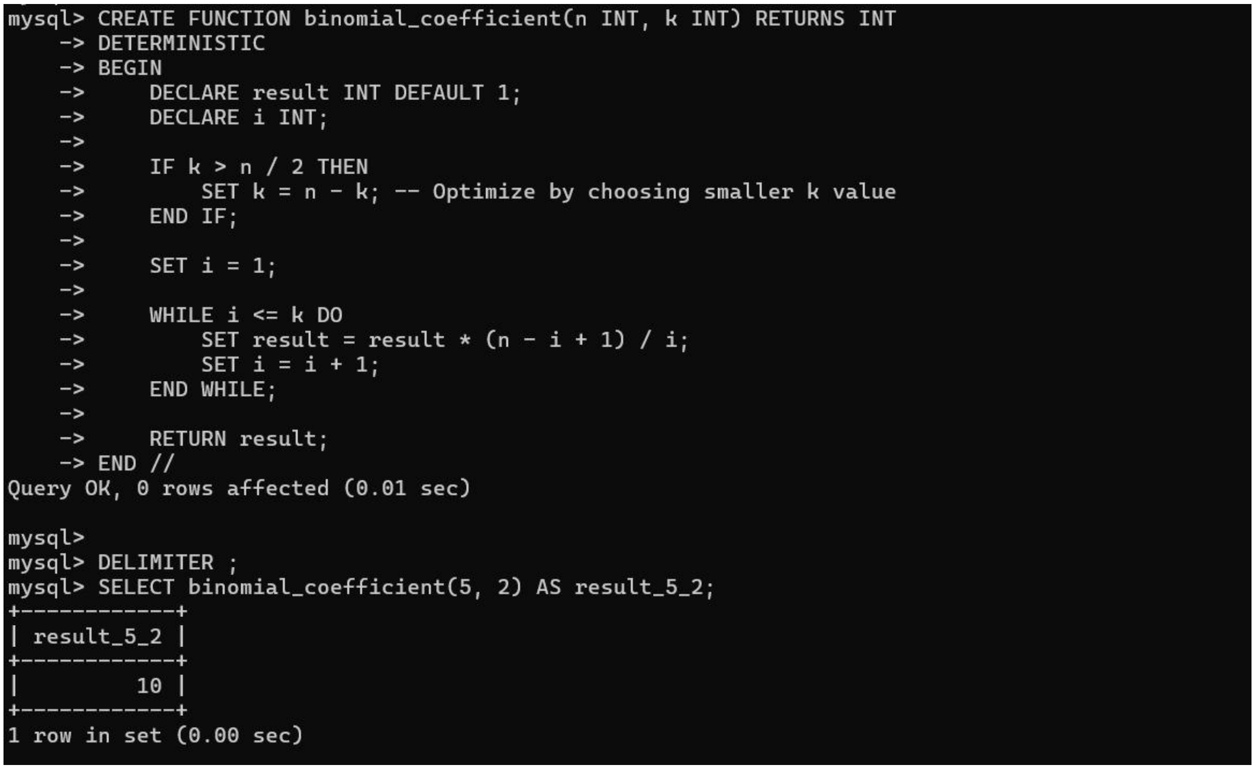


```
Database changed
mysql> DELIMITER //
mysql>
mysql> CREATE FUNCTION powers(x DOUBLE, n INT) RETURNS DOUBLE
    -> DETERMINISTIC
    -> BEGIN
    ->     IF n = 0 THEN
    ->         RETURN 1;
    ->     ELSEIF n < 0 THEN
    ->         RETURN 1 / power(x, -n);
    ->     ELSE
    ->         RETURN x * power(x, n - 1);
    ->     END IF;
    -> END //
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
mysql> select powers(2,3) as result;
+--------+
| result |
+--------+
|      8 |
+--------+
1 row in set (0.00 sec)

mysql> 
```

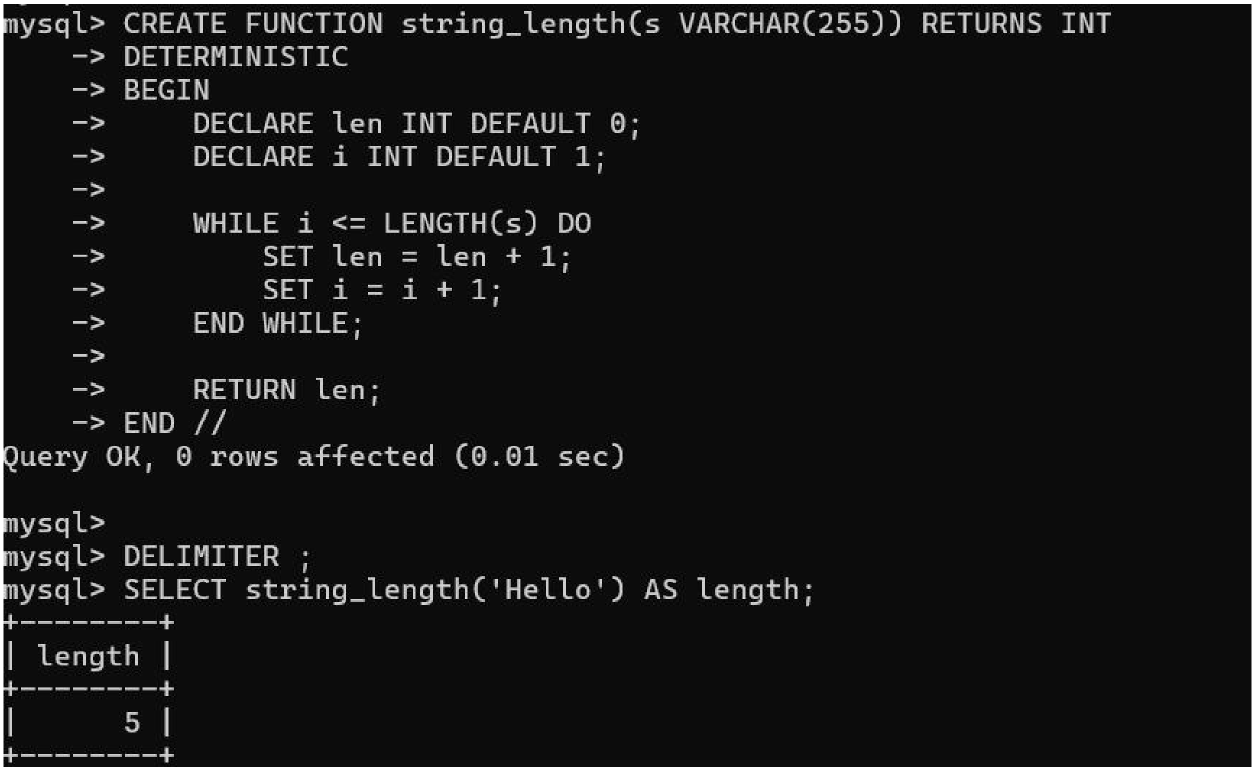
5. Write a recursive MySQL function to compute the binomial coefficient (n choose k).



```
mysql> CREATE FUNCTION binomial_coefficient(n INT, k INT) RETURNS INT
    -> DETERMINISTIC
    -> BEGIN
    ->     DECLARE result INT DEFAULT 1;
    ->     DECLARE i INT;
    ->
    ->     IF k > n / 2 THEN
    ->         SET k = n - k; -- Optimize by choosing smaller k value
    ->     END IF;
    ->
    ->     SET i = 1;
    ->
    ->     WHILE i <= k DO
    ->         SET result = result * (n - i + 1) / i;
    ->         SET i = i + 1;
    ->     END WHILE;
    ->
    ->     RETURN result;
    -> END //
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
mysql> SELECT binomial_coefficient(5, 2) AS result_5_2;
+------------+
| result_5_2 |
+------------+
|         10 |
+------------+
1 row in set (0.00 sec)
```

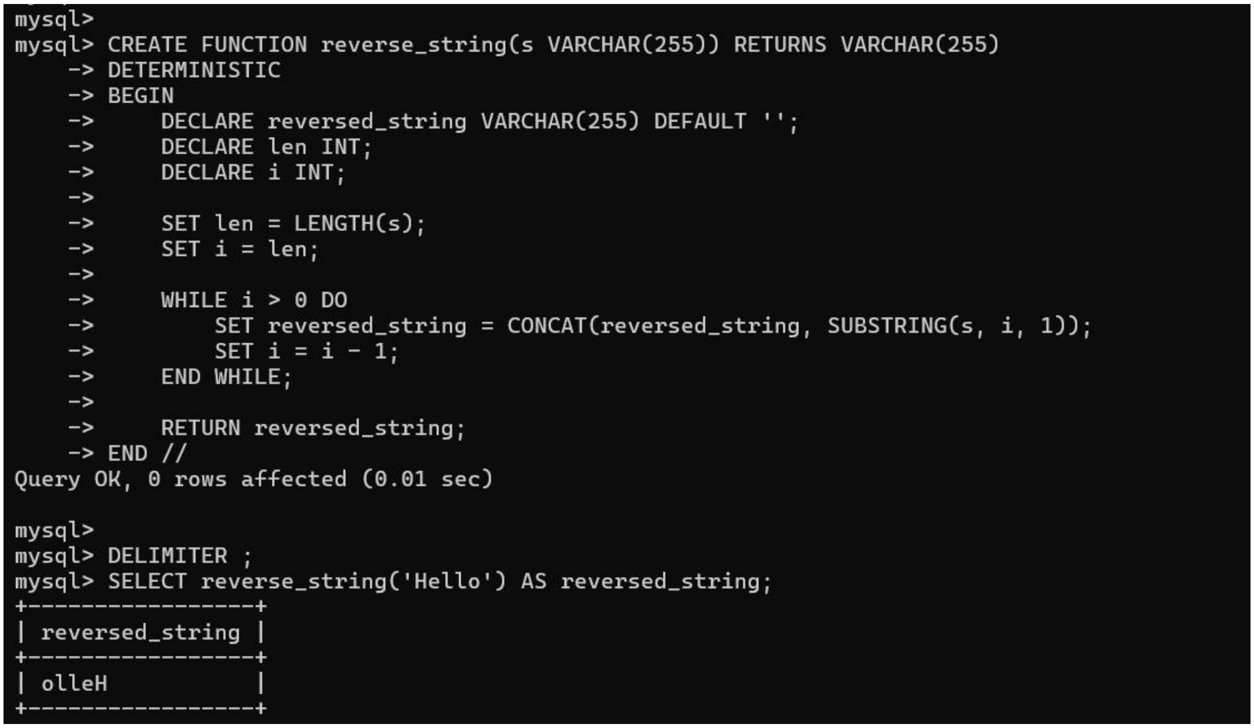
6. Develop a recursive MySQL function to find the length of a string.



```
mysql> CREATE FUNCTION string_length(s VARCHAR(255)) RETURNS INT
    -> DETERMINISTIC
    -> BEGIN
    ->     DECLARE len INT DEFAULT 0;
    ->     DECLARE i INT DEFAULT 1;
    ->
    ->     WHILE i <= LENGTH(s) DO
    ->         SET len = len + 1;
    ->         SET i = i + 1;
    ->     END WHILE;
    ->
    ->     RETURN len;
    -> END //
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
mysql> SELECT string_length('Hello') AS length;
+--------+
| length |
+--------+
|      5 |
+--------+
```

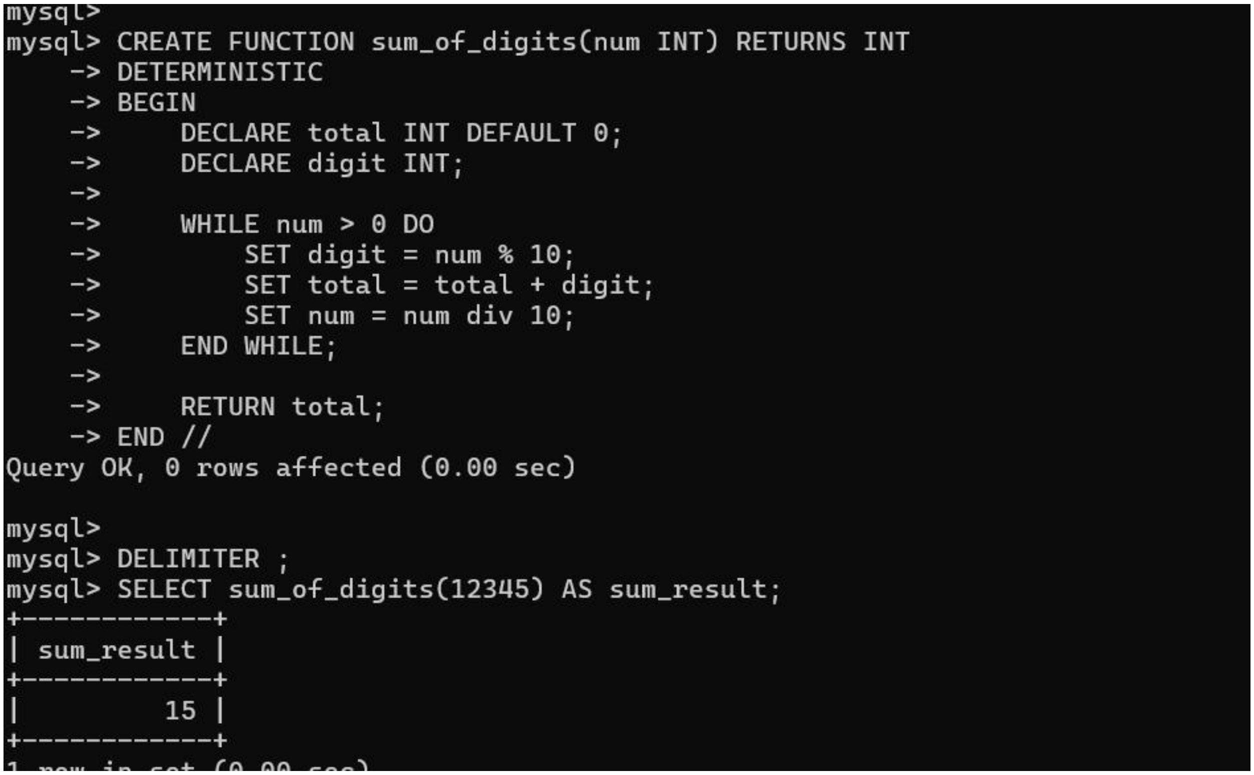
7. Create a recursive MySQL function to reverse a given string.



```
mysql>
mysql> CREATE FUNCTION reverse_string(s VARCHAR(255)) RETURNS VARCHAR(255)
    -> DETERMINISTIC
    -> BEGIN
    ->     DECLARE reversed_string VARCHAR(255) DEFAULT '';
    ->     DECLARE len INT;
    ->     DECLARE i INT;
    ->
    ->     SET len = LENGTH(s);
    ->     SET i = len;
    ->
    ->     WHILE i > 0 DO
    ->         SET reversed_string = CONCAT(reversed_string, SUBSTRING(s, i, 1));
    ->         SET i = i - 1;
    ->     END WHILE;
    ->
    ->     RETURN reversed_string;
    -> END //
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
mysql> SELECT reverse_string('Hello') AS reversed_string;
+-----------------+
| reversed_string |
+-----------------+
| olleH           |
+-----------------+
```

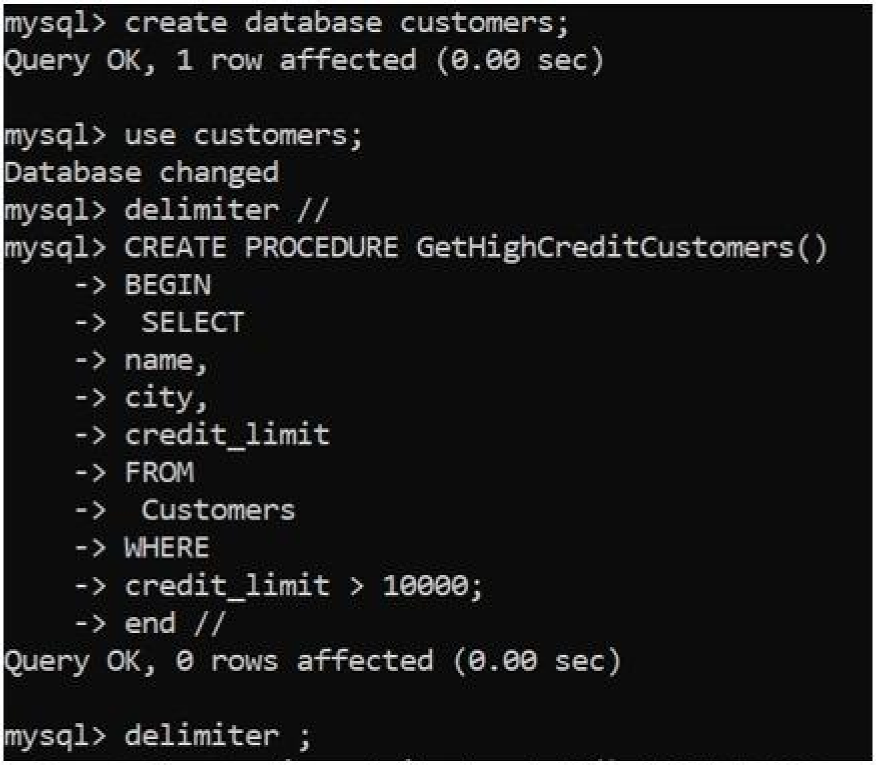
8. Develop a recursive MySQL function to calculate the sum of the digits of a given number.



```
mysql>
mysql> CREATE FUNCTION sum_of_digits(num INT) RETURNS INT
    -> DETERMINISTIC
    -> BEGIN
    ->     DECLARE total INT DEFAULT 0;
    ->     DECLARE digit INT;
    ->
    ->     WHILE num > 0 DO
    ->         SET digit = num % 10;
    ->         SET total = total + digit;
    ->         SET num = num div 10;
    ->     END WHILE;
    ->
    ->     RETURN total;
    -> END //
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> DELIMITER ;
mysql> SELECT sum_of_digits(12345) AS sum_result;
+------------+
| sum_result |
+------------+
|         15 |
+------------+
1 row in set (0.00 sec)
```

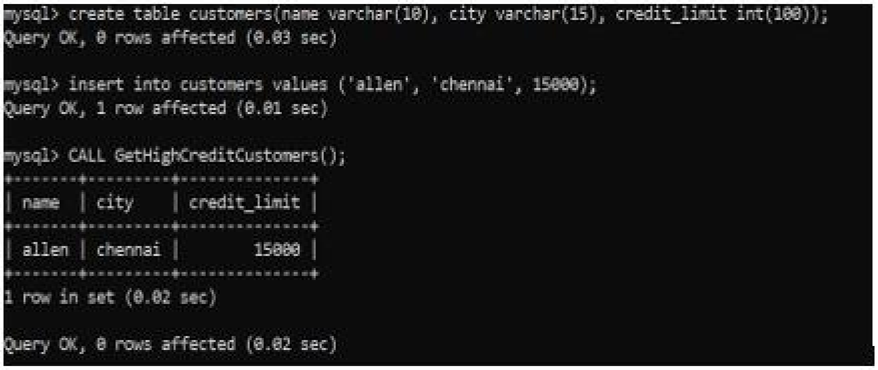
9. Create a procedure to get all the records from the table 'Customers' and list out the name, city, and credit limit where credit limit > 10000.



```
mysql> create database customers;
Query OK, 1 row affected (0.00 sec)

mysql> use customers;
Database changed
mysql> delimiter //
mysql> CREATE PROCEDURE GetHighCreditCustomers()
    -> BEGIN
    ->  SELECT
    -> name,
    -> city,
    -> credit_limit
    -> FROM
    ->  Customers
    -> WHERE
    -> credit_limit > 10000;
    -> end //
Query OK, 0 rows affected (0.00 sec)

mysql> delimiter ;
```



```
mysql> create table customers(name varchar(10), city varchar(15), credit_limit int(100));
Query OK, 0 rows affected (0.03 sec)

mysql> insert into customers values ('allen', 'chennai', 15000);
Query OK, 1 row affected (0.01 sec)

mysql> CALL GetHighCreditCustomers();
+-------+---------+--------------+
| name  | city    | credit_limit |
+-------+---------+--------------+
| allen | chennai |        15000 |
+-------+---------+--------------+
1 row in set (0.02 sec)

Query OK, 0 rows affected (0.02 sec)
```