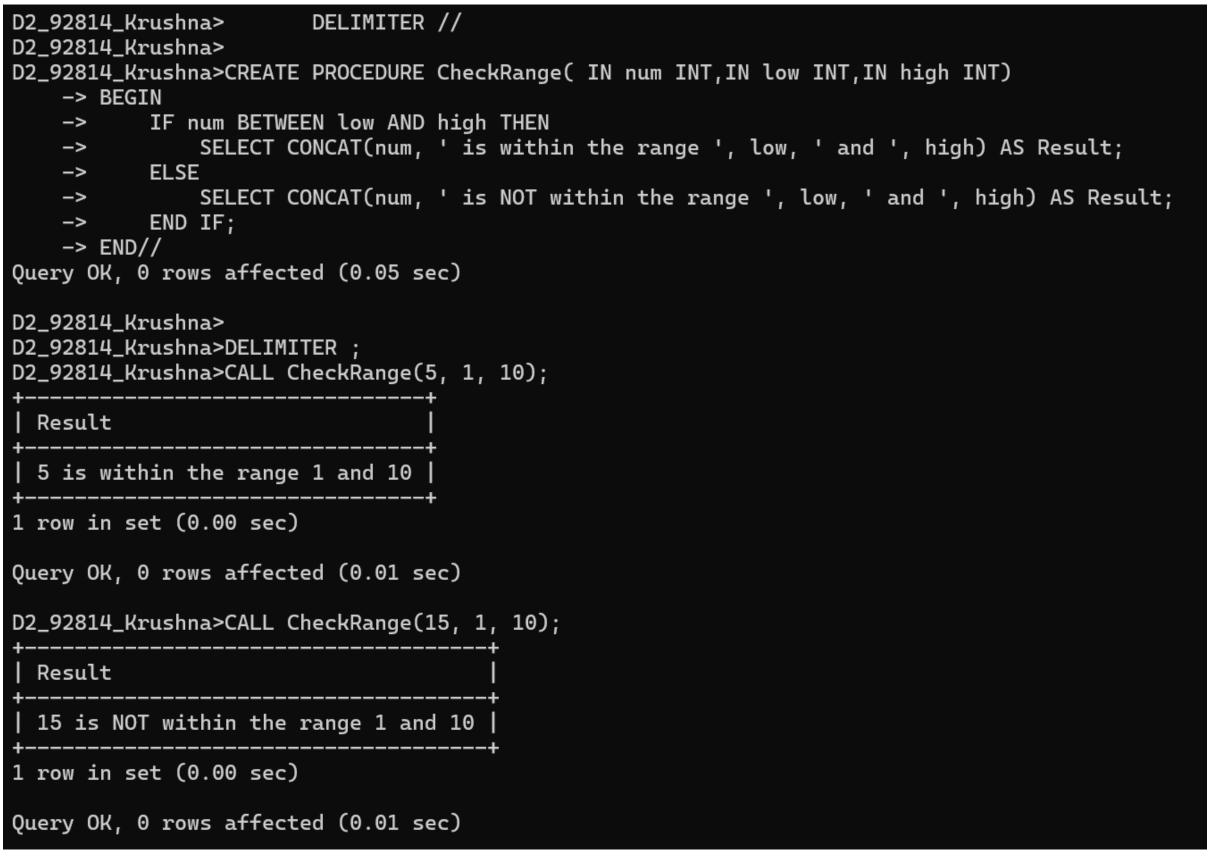
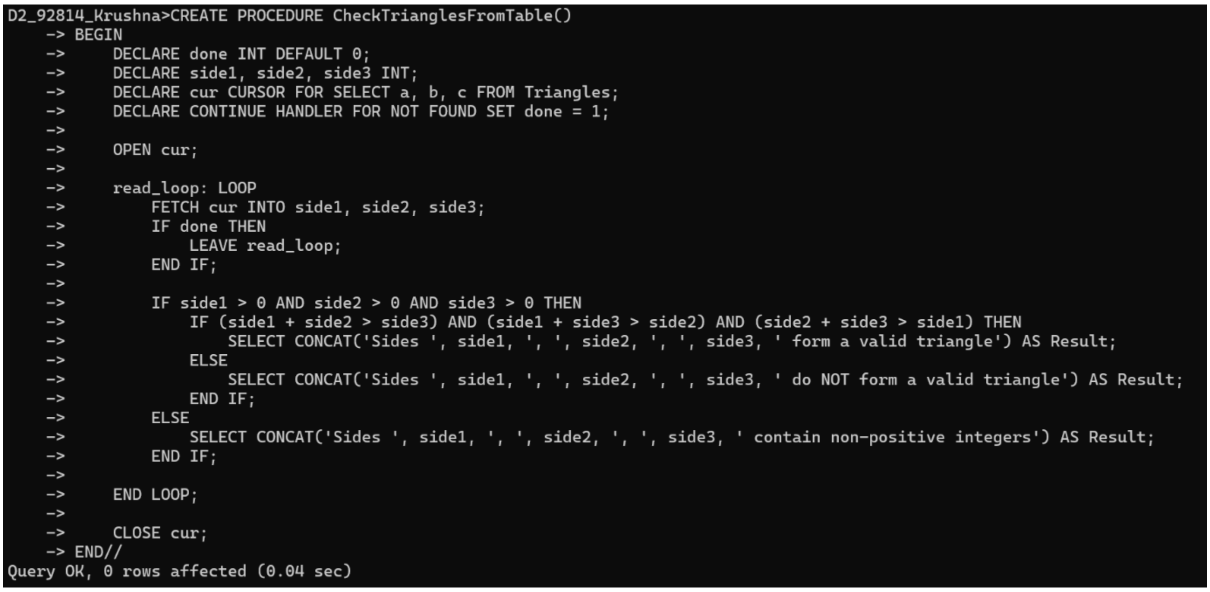
MySQL Exercise 2

1. Select from any table a number and determine whether it is within a given range (for example, between 1 and 10).



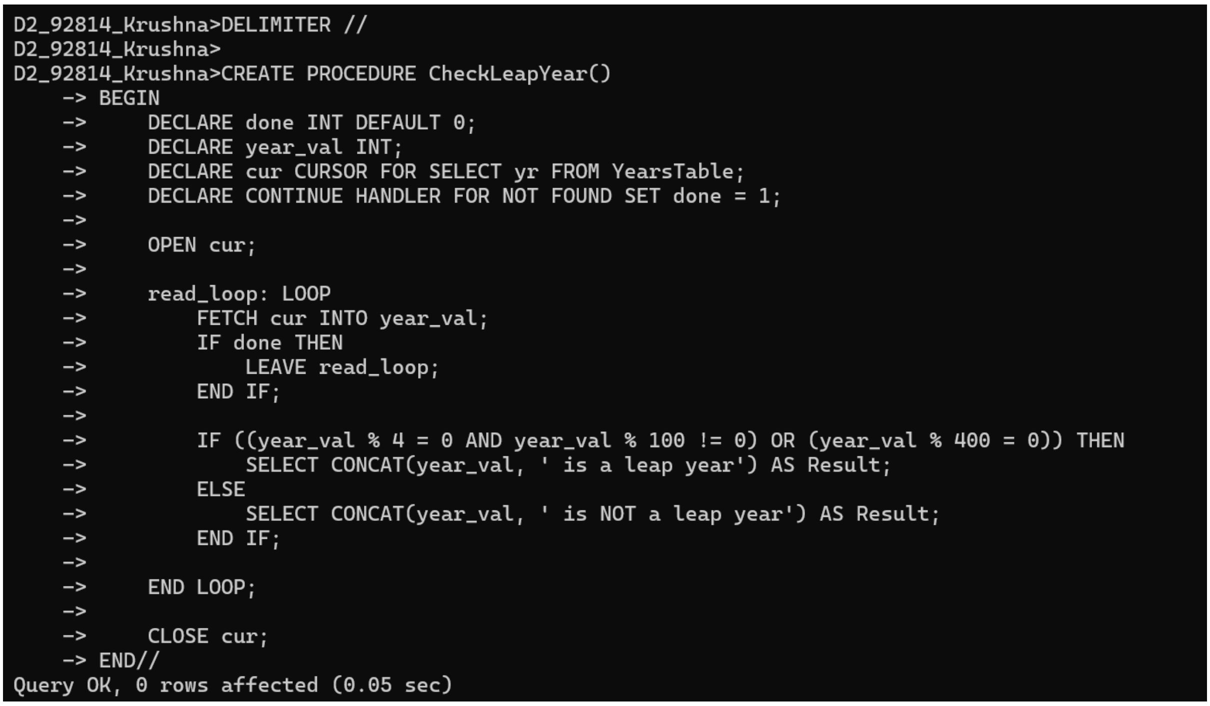
```
D2_92814_Krushna>       DELIMITER //
D2_92814_Krushna>
D2_92814_Krushna>CREATE PROCEDURE CheckRange( IN num INT,IN low INT,IN high INT)
    -> BEGIN
    ->     IF num BETWEEN low AND high THEN
    ->         SELECT CONCAT(num, ' is within the range ', low, ' and ', high) AS Result;
    ->     ELSE
    ->         SELECT CONCAT(num, ' is NOT within the range ', low, ' and ', high) AS Result;
    ->     END IF;
    -> END//
Query OK, 0 rows affected (0.05 sec)

D2_92814_Krushna>
D2_92814_Krushna>DELIMITER ;
D2_92814_Krushna>CALL CheckRange(5, 1, 10);
+--------------------------------+
| Result                         |
+--------------------------------+
| 5 is within the range 1 and 10 |
+--------------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

D2_92814_Krushna>CALL CheckRange(15, 1, 10);
+-------------------------------------+
| Result                              |
+-------------------------------------+
| 15 is NOT within the range 1 and 10 |
+-------------------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
```

2. Select from any table three positive integers representing the sides of a triangle, and determine whether they form a valid triangle. Hint: In a triangle, the sum of any two sides must always be greater than the third side.

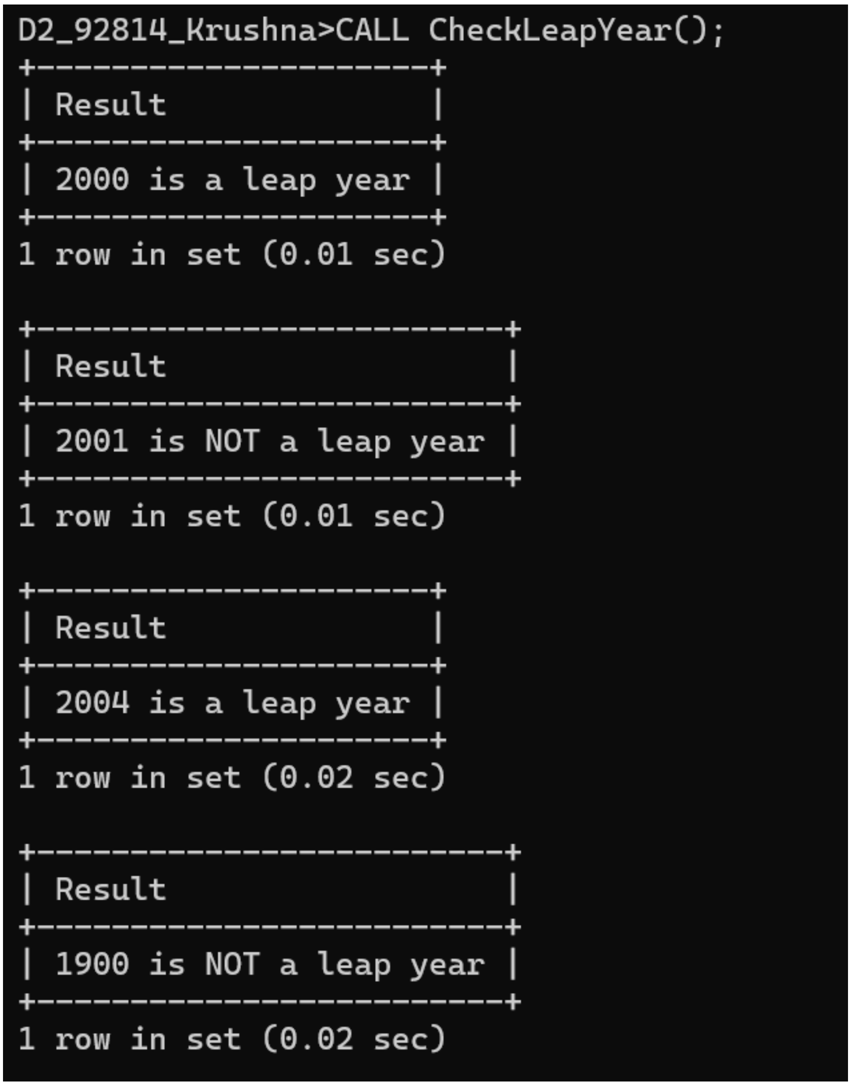


```
D2_92814_Krushna>CREATE PROCEDURE CheckTrianglesFromTable()
    -> BEGIN
    ->     DECLARE done INT DEFAULT 0;
    ->     DECLARE side1, side2, side3 INT;
    ->     DECLARE cur CURSOR FOR SELECT a, b, c FROM Triangles;
    ->     DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;
    ->
    ->     OPEN cur;
    ->
    ->     read_loop: LOOP
    ->         FETCH cur INTO side1, side2, side3;
    ->         IF done THEN
    ->             LEAVE read_loop;
    ->         END IF;
    ->
    ->         IF side1 > 0 AND side2 > 0 AND side3 > 0 THEN
    ->             IF (side1 + side2 > side3) AND (side1 + side3 > side2) AND (side2 + side3 > side1) THEN
    ->                 SELECT CONCAT('Sides ', side1, ', ', side2, ', ', side3, ' form a valid triangle') AS Result;
    ->             ELSE
    ->                 SELECT CONCAT('Sides ', side1, ', ', side2, ', ', side3, ' do NOT form a valid triangle') AS Result;
    ->             END IF;
    ->         ELSE
    ->             SELECT CONCAT('Sides ', side1, ', ', side2, ', ', side3, ' contain non-positive integers') AS Result;
    ->         END IF;
    ->
    ->     END LOOP;
    ->
    ->     CLOSE cur;
    -> END//
Query OK, 0 rows affected (0.04 sec)
```

3. Check if a given a year is a leap year. The condition is:- year should be (divisible by 4 and not divisible by 100) or (divisible by 4 and divisible by 400.). The year should be Selected from some table.

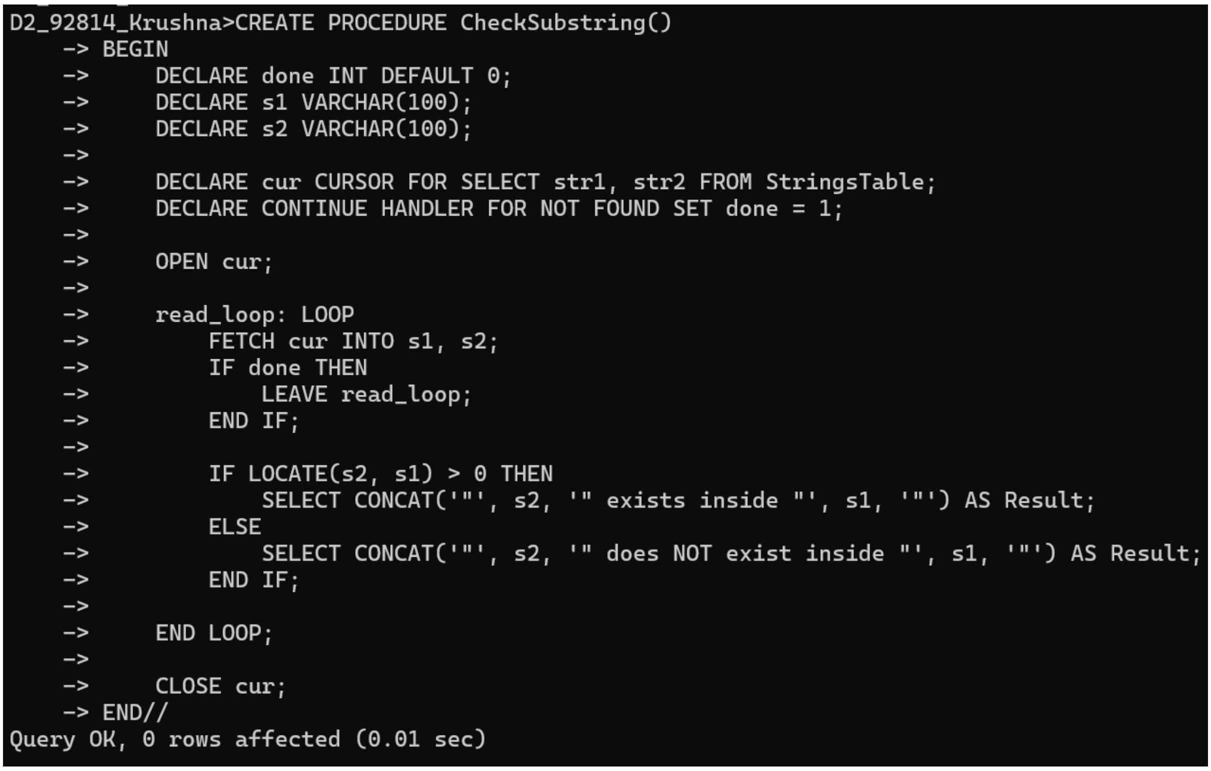


```
D2_92814_Krushna>DELIMITER //
D2_92814_Krushna>
D2_92814_Krushna>CREATE PROCEDURE CheckLeapYear()
    -> BEGIN
    ->     DECLARE done INT DEFAULT 0;
    ->     DECLARE year_val INT;
    ->     DECLARE cur CURSOR FOR SELECT yr FROM YearsTable;
    ->     DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;
    ->
    ->     OPEN cur;
    ->
    ->     read_loop: LOOP
    ->         FETCH cur INTO year_val;
    ->         IF done THEN
    ->             LEAVE read_loop;
    ->         END IF;
    ->
    ->         IF ((year_val % 4 = 0 AND year_val % 100 != 0) OR (year_val % 400 = 0)) THEN
    ->             SELECT CONCAT(year_val, ' is a leap year') AS Result;
    ->         ELSE
    ->             SELECT CONCAT(year_val, ' is NOT a leap year') AS Result;
    ->         END IF;
    ->
    ->     END LOOP;
    ->
    ->     CLOSE cur;
    -> END//
Query OK, 0 rows affected (0.05 sec)
```

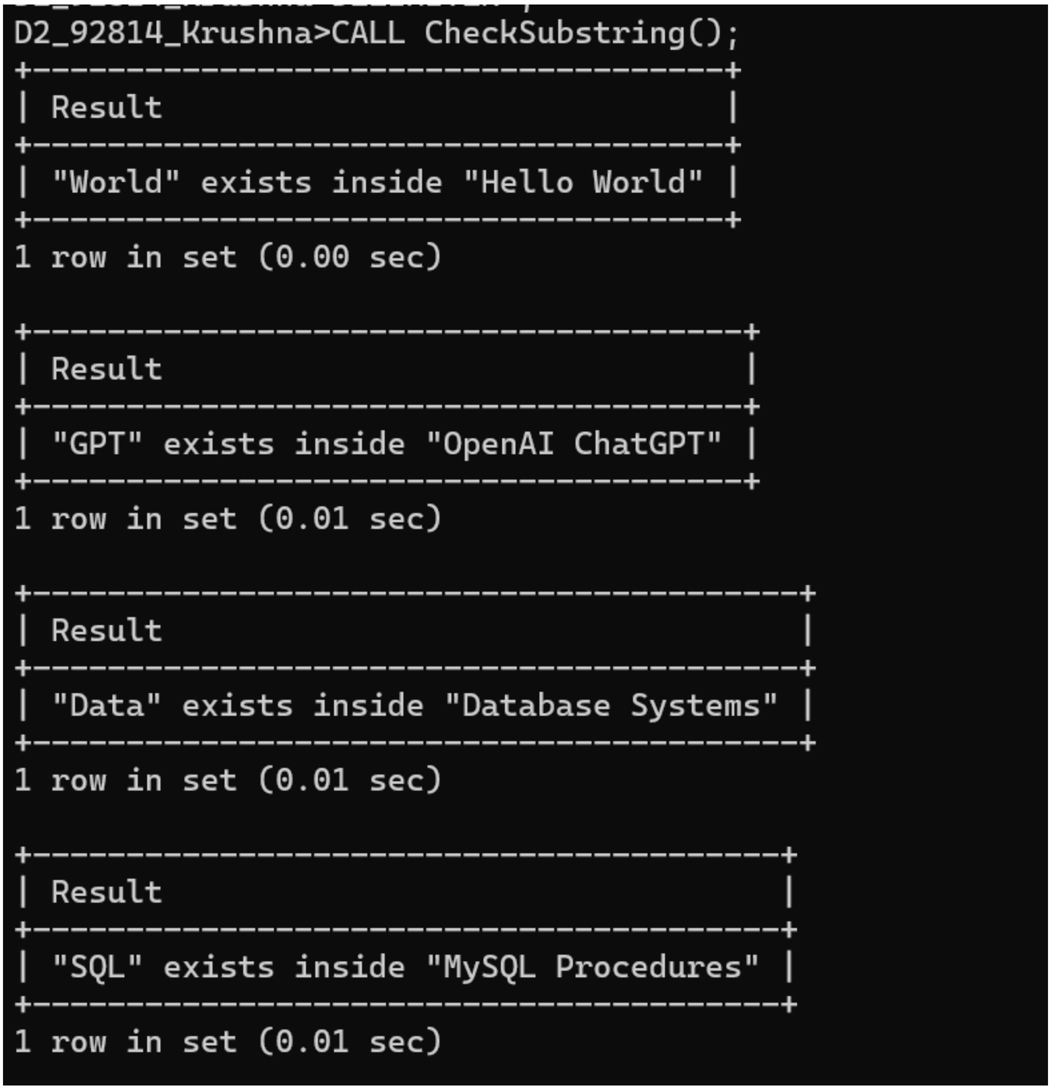


```
D2_92814_Krushna>CALL CheckLeapYear();
+---------------------+
| Result              |
+---------------------+
| 2000 is a leap year |
+---------------------+
1 row in set (0.01 sec)

+-------------------------+
| Result                  |
+-------------------------+
| 2001 is NOT a leap year |
+-------------------------+
1 row in set (0.01 sec)

+---------------------+
| Result              |
+---------------------+
| 2004 is a leap year |
+---------------------+
1 row in set (0.02 sec)

+-------------------------+
| Result                  |
+-------------------------+
| 1900 is NOT a leap year |
+-------------------------+
1 row in set (0.02 sec)
```

4. Write a program that Selects from any table two character strings. Your program should then determine if one character string exists inside another character string.



```
D2_92814_Krushna>CREATE PROCEDURE CheckSubstring()
    -> BEGIN
    ->     DECLARE done INT DEFAULT 0;
    ->     DECLARE s1 VARCHAR(100);
    ->     DECLARE s2 VARCHAR(100);
    ->
    ->     DECLARE cur CURSOR FOR SELECT str1, str2 FROM StringsTable;
    ->     DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;
    ->
    ->     OPEN cur;
    ->
    ->     read_loop: LOOP
    ->         FETCH cur INTO s1, s2;
    ->         IF done THEN
    ->             LEAVE read_loop;
    ->         END IF;
    ->
    ->         IF LOCATE(s2, s1) > 0 THEN
    ->             SELECT CONCAT('"', s2, '" exists inside "', s1, '"') AS Result;
    ->         ELSE
    ->             SELECT CONCAT('"', s2, '" does NOT exist inside "', s1, '"') AS Result;
    ->         END IF;
    ->
    ->     END LOOP;
    ->
    ->     CLOSE cur;
    -> END//
Query OK, 0 rows affected (0.01 sec)
```



```
D2_92814_Krushna>CALL CheckSubstring();
+-------------------------------------+
| Result                              |
+-------------------------------------+
| "World" exists inside "Hello World" |
+-------------------------------------+
1 row in set (0.00 sec)

+--------------------------------------+
| Result                               |
+--------------------------------------+
| "GPT" exists inside "OpenAI ChatGPT" |
+--------------------------------------+
1 row in set (0.01 sec)

+-----------------------------------------+
| Result                                  |
+-----------------------------------------+
| "Data" exists inside "Database Systems" |
+-----------------------------------------+
1 row in set (0.01 sec)

+----------------------------------------+
| Result                                 |
+----------------------------------------+
| "SQL" exists inside "MySQL Procedures" |
+----------------------------------------+
1 row in set (0.01 sec)
```