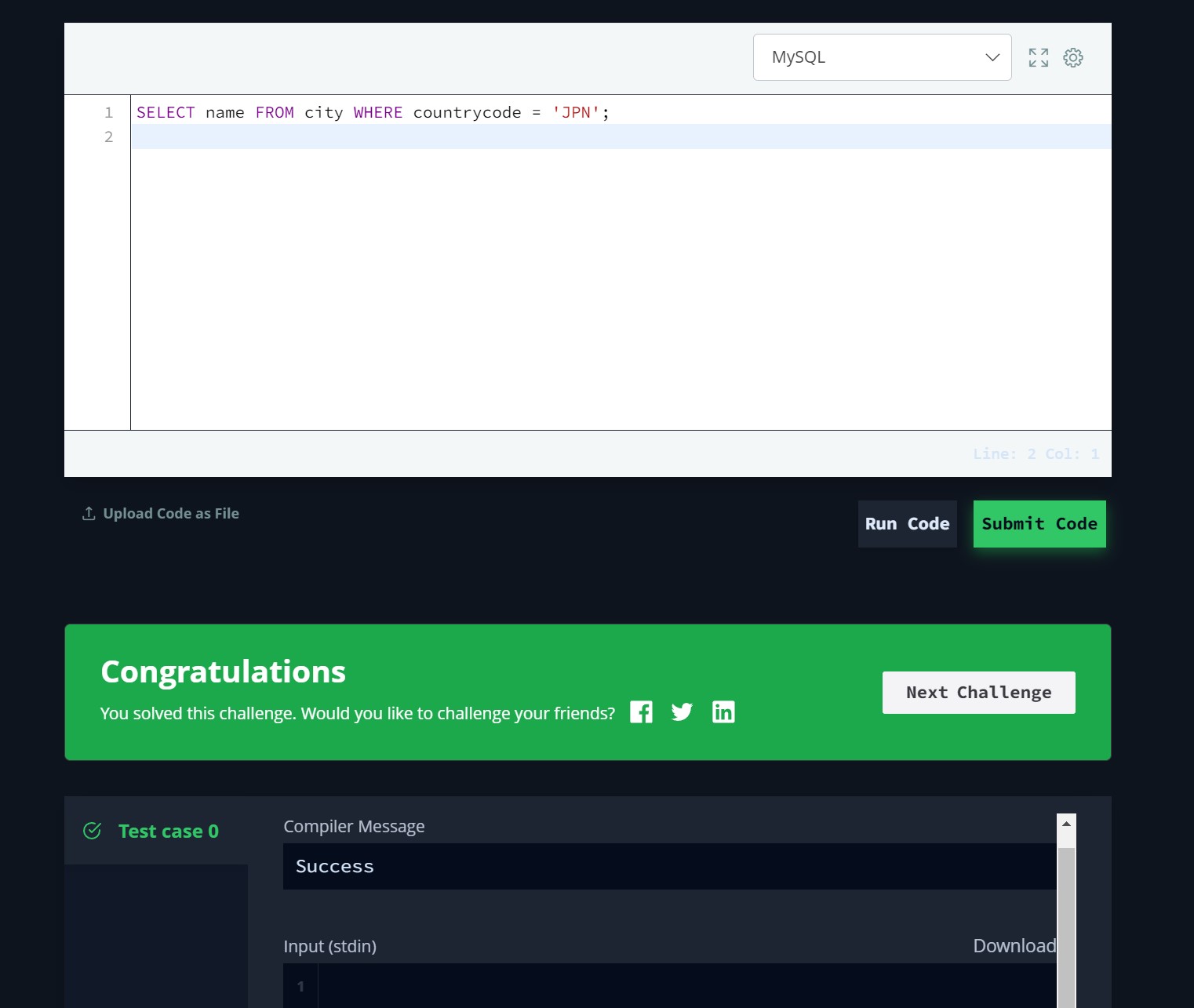
**Database Assignment 2**

1. [Japanese Cities’ Names](https://www.hackerrank.com/challenges/japanese-cities-name/problem)

SQL Script Solution:

SELECT name FROM city WHERE countrycode = 'JPN';

Output Screenshot:



1. [Weather Observation Station 3](https://www.hackerrank.com/challenges/weather-observation-station-3/problem?isFullScreen=true)

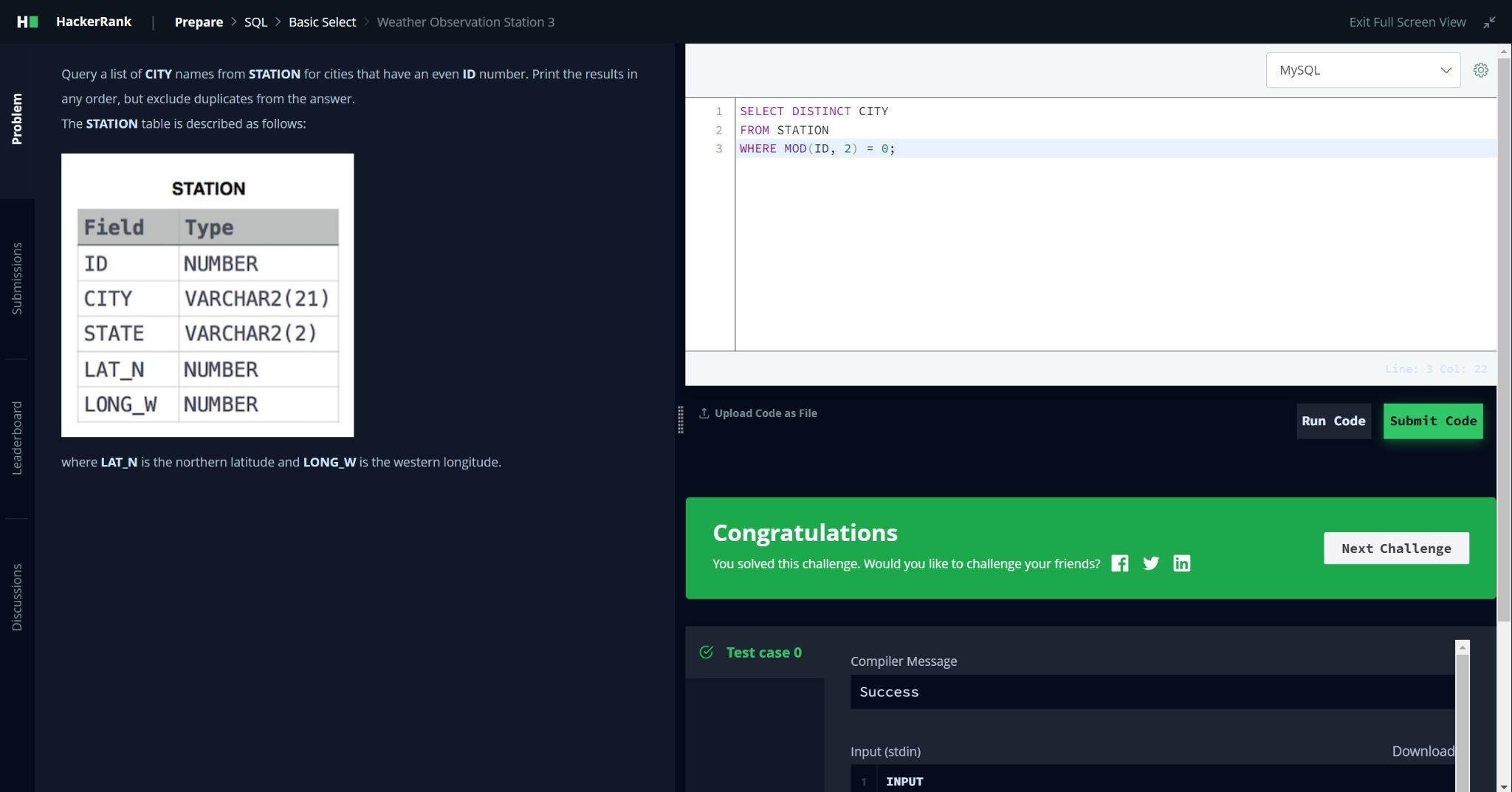
SQL Script Solution:

SELECT DISTINCT CITY

FROM STATION

WHERE MOD(ID, 2) = 0;

Output Screenshot:



1. [Weather Observation Station 5](https://www.hackerrank.com/challenges/weather-observation-station-5/problem)

SQL Script Solution:

// for shortest city

SELECT city, LENGTH(city) AS city\_length

FROM station

ORDER BY LENGTH(city) ASC, city ASC

LIMIT 1;

// for longest city

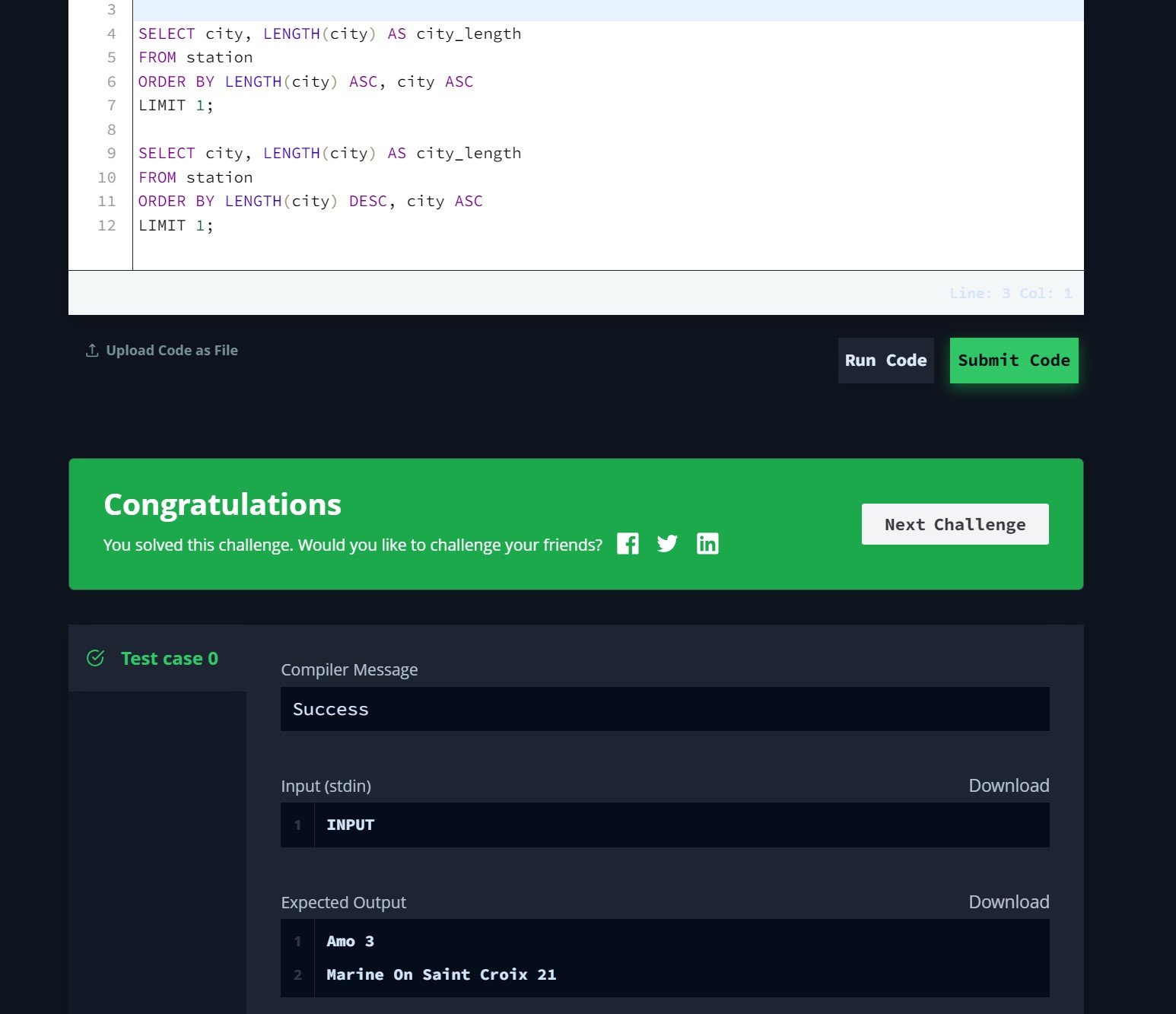
SELECT city, LENGTH(city) AS city\_length

FROM station

ORDER BY LENGTH(city) DESC, city ASC

LIMIT 1;

Output Screenshot:



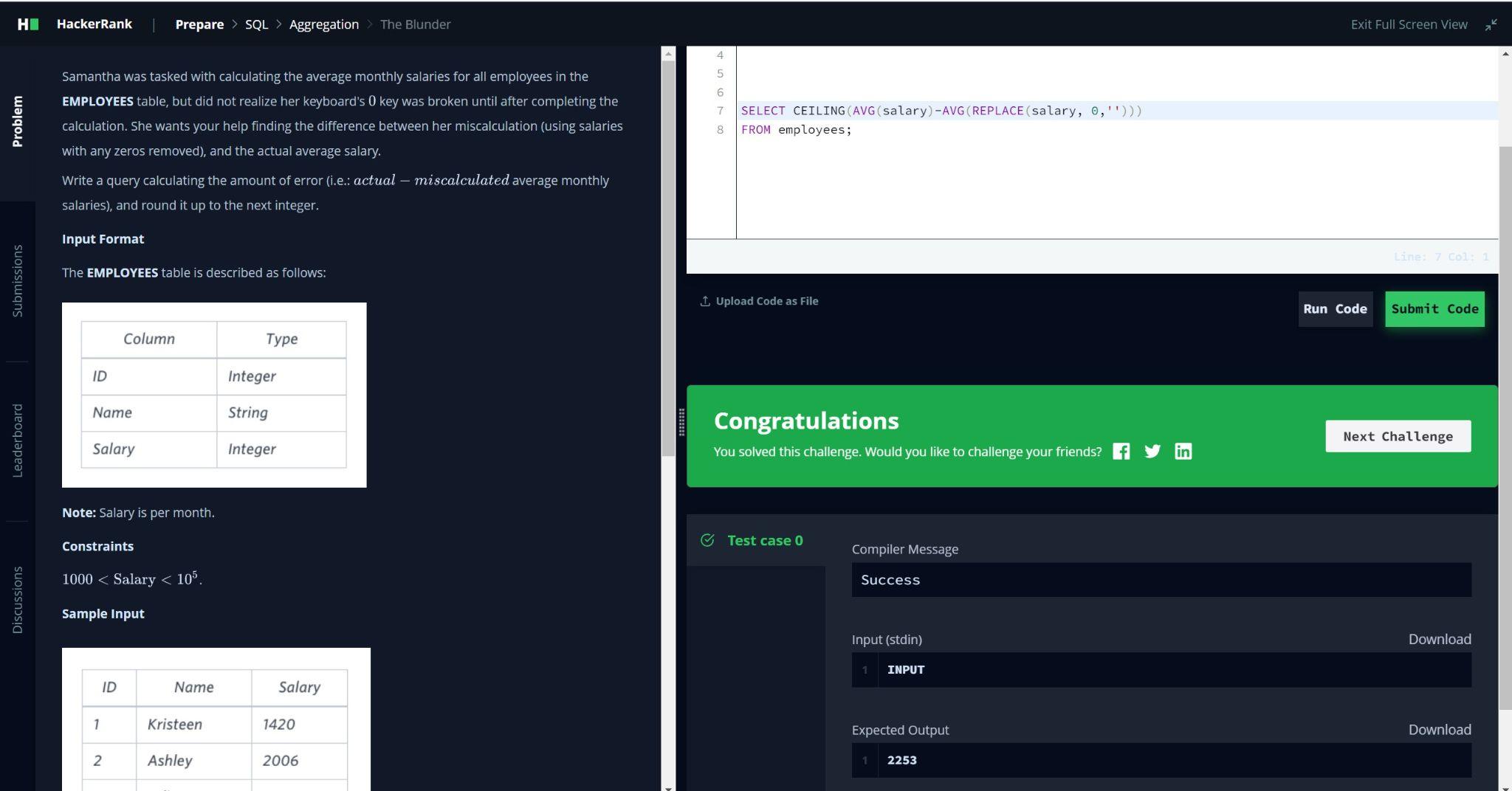
1. [The Blunder](https://www.hackerrank.com/challenges/the-blunder/problem?isFullScreen=true)

SQL Script Solution:

SELECT CEILING(AVG(salary)-AVG(REPLACE(salary, 0,'')))

FROM employees;

Output Screenshot:



1. [Weather Observation Station 18](https://www.hackerrank.com/challenges/weather-observation-station-18/problem)

SQL Script Solution:

WITH diff AS (

SELECT

MIN(LAT\_N) AS a\_latn\_min,

MAX(LAT\_N) AS c\_latn\_max,

MIN(LONG\_W) AS b\_longw\_min,

MAX(LONG\_W) AS d\_longw\_max

FROM station

)

SELECT

ROUND(

ABS(c\_latn\_max - a\_latn\_min)

+

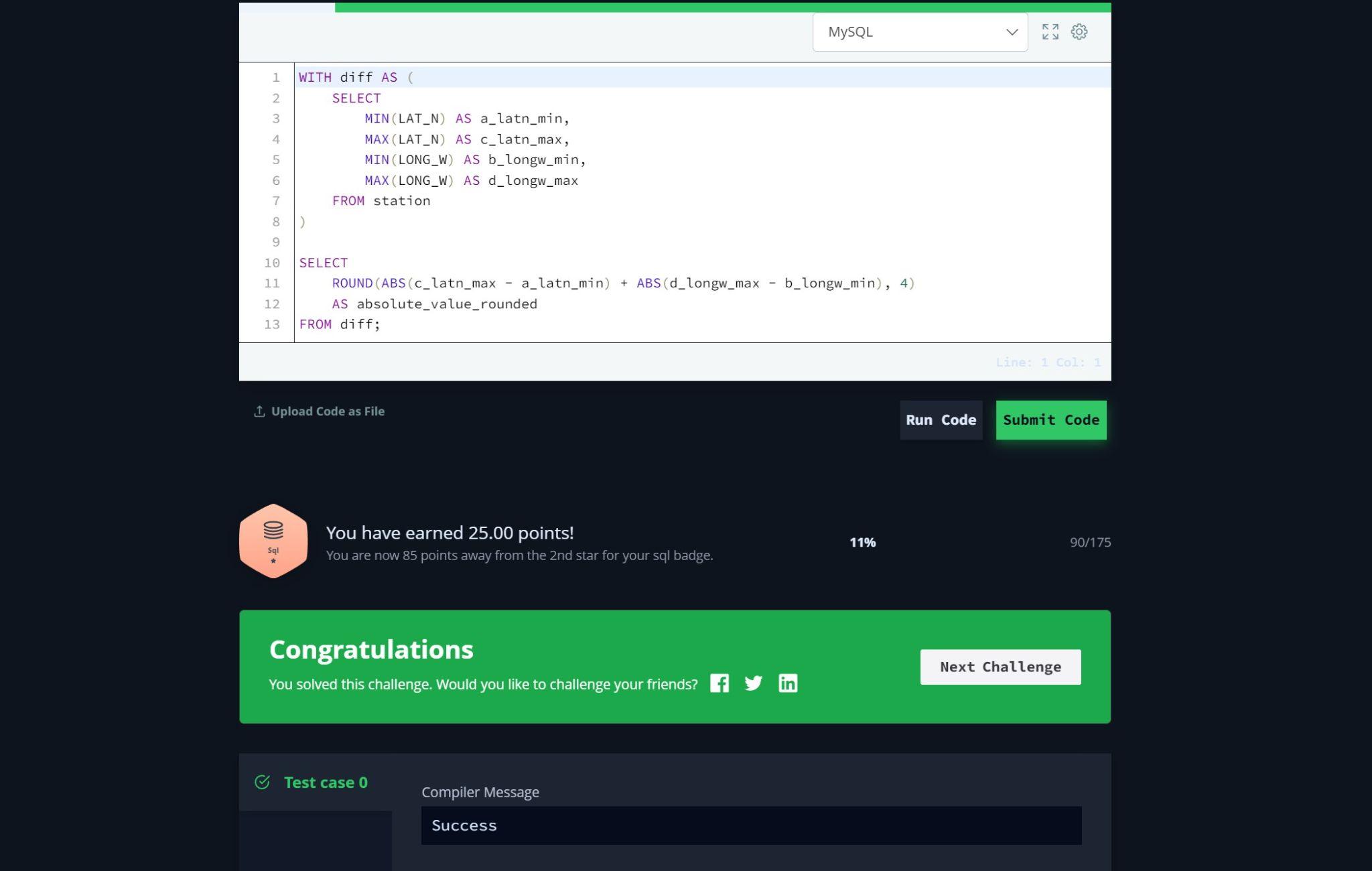
ABS(d\_longw\_max - b\_longw\_min),

4)

AS absolute\_value\_rounded

FROM diff;

Output Screenshot:



1. [Average Population of Each Continent](https://www.hackerrank.com/challenges/average-population-of-each-continent/problem?isFullScreen=true)

SQL Script Solution:

SELECT COUNTRY.Continent, FLOOR(AVG(CITY.Population))

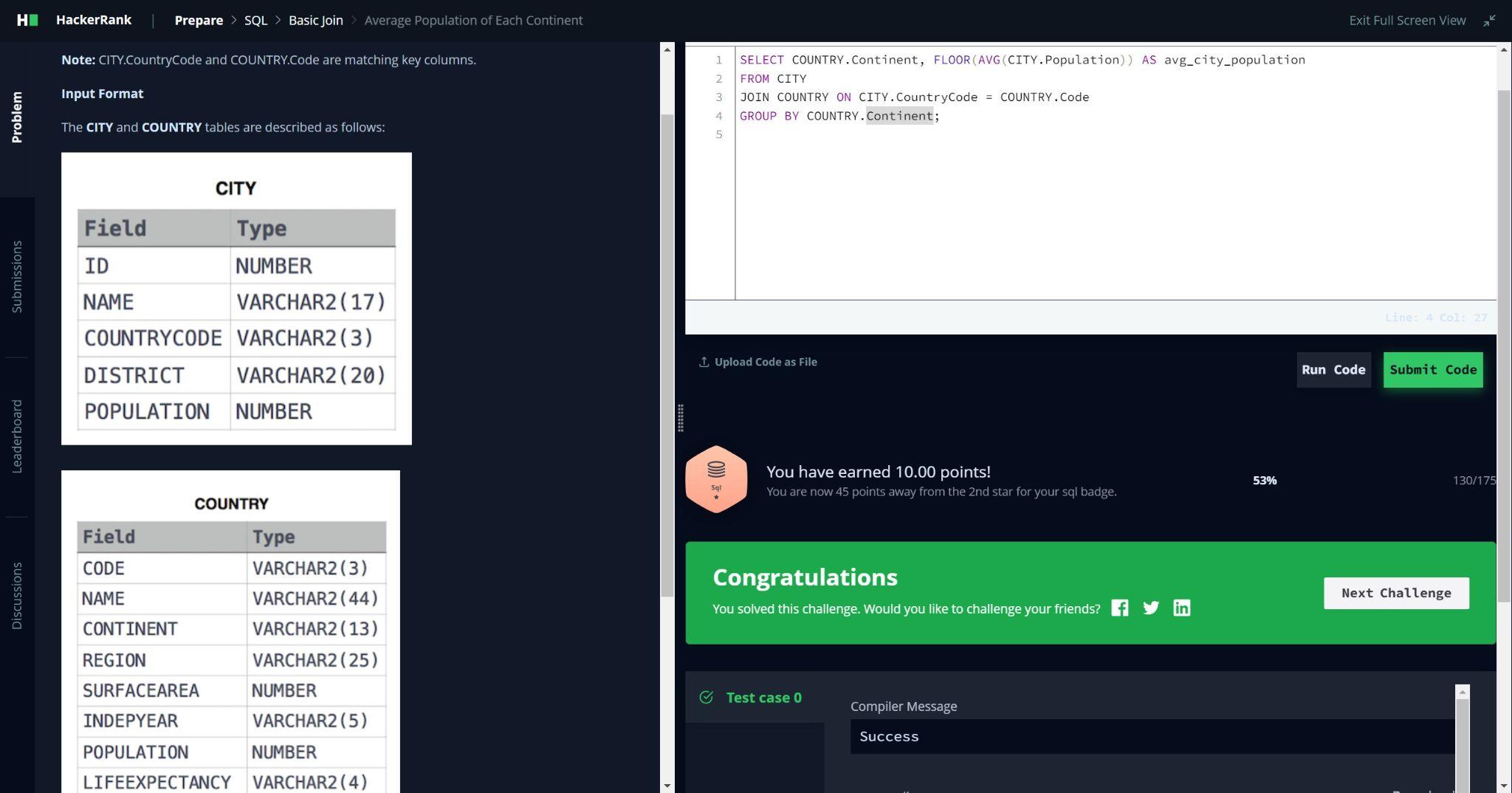
AS avg\_city\_population

FROM CITY

JOIN COUNTRY ON CITY.CountryCode = COUNTRY.Code

GROUP BY COUNTRY.Continent;

Output Screenshot:



1. [The PADS](https://www.hackerrank.com/challenges/the-pads/problem)

SQL Script Solution:

SELECT

CONCAT(name, '(', LEFT(occupation, 1), ')')

FROM occupations

ORDER BY name ASC;

SELECT

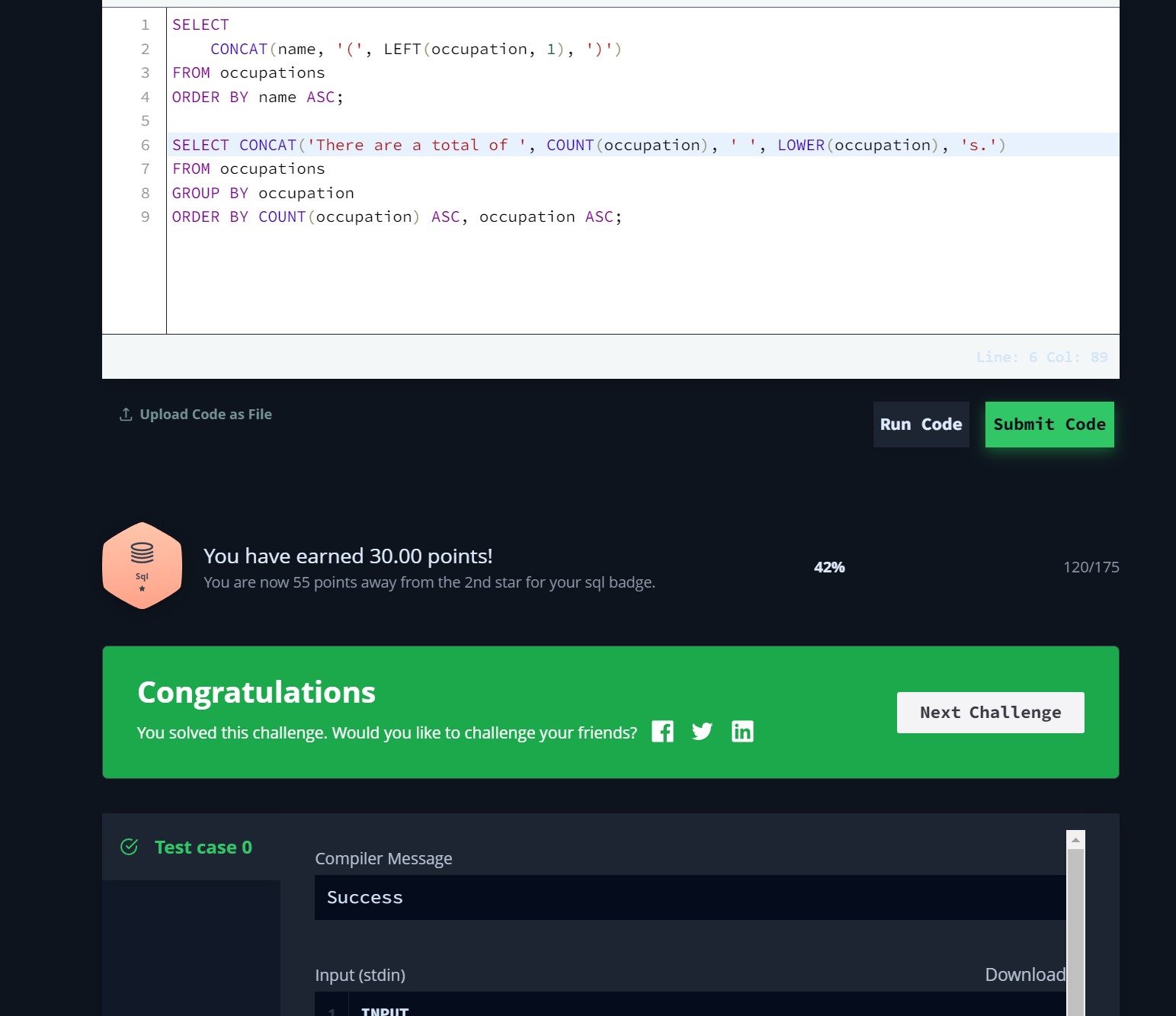
CONCAT('There are a total of ', COUNT(occupation), ' ', LOWER(occupation), 's.')

FROM occupations

GROUP BY occupation

ORDER BY COUNT(occupation) ASC, occupation ASC;

Output Screenshot:



1. [Type of Triangle](https://www.hackerrank.com/challenges/what-type-of-triangle/problem)

SQL Script Solution:

SELECT

CASE

WHEN A + B <= C OR A + C <= B OR B + C <= A THEN 'Not A Triangle'

WHEN A = B AND B = C THEN 'Equilateral'

WHEN A = B OR A = C OR B = C THEN 'Isosceles'

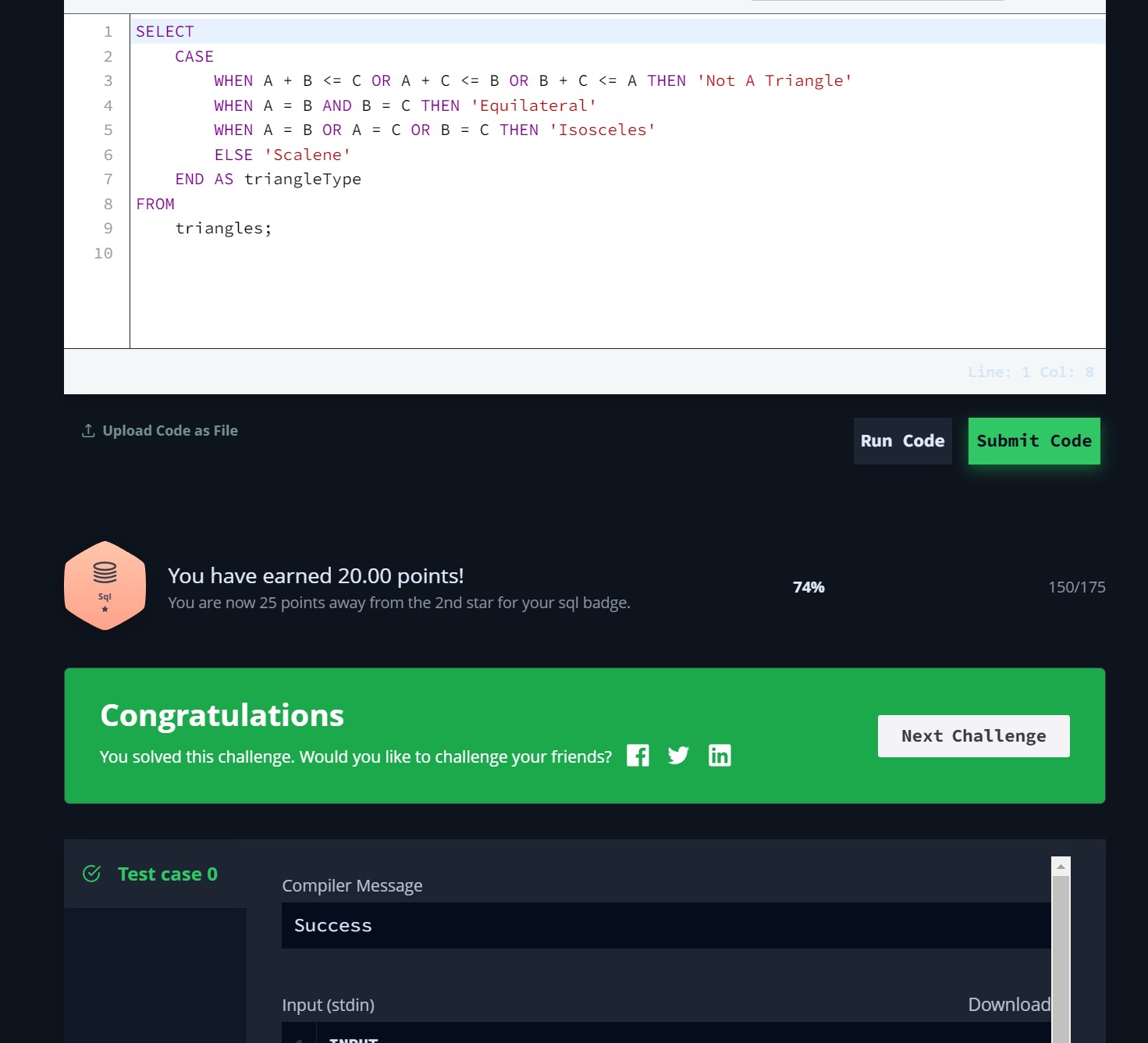
ELSE 'Scalene'

END AS triangleType

FROM

triangles;

Output Screenshot:



1. [Weather Observation Station 13](https://www.hackerrank.com/challenges/weather-observation-station-13/problem)

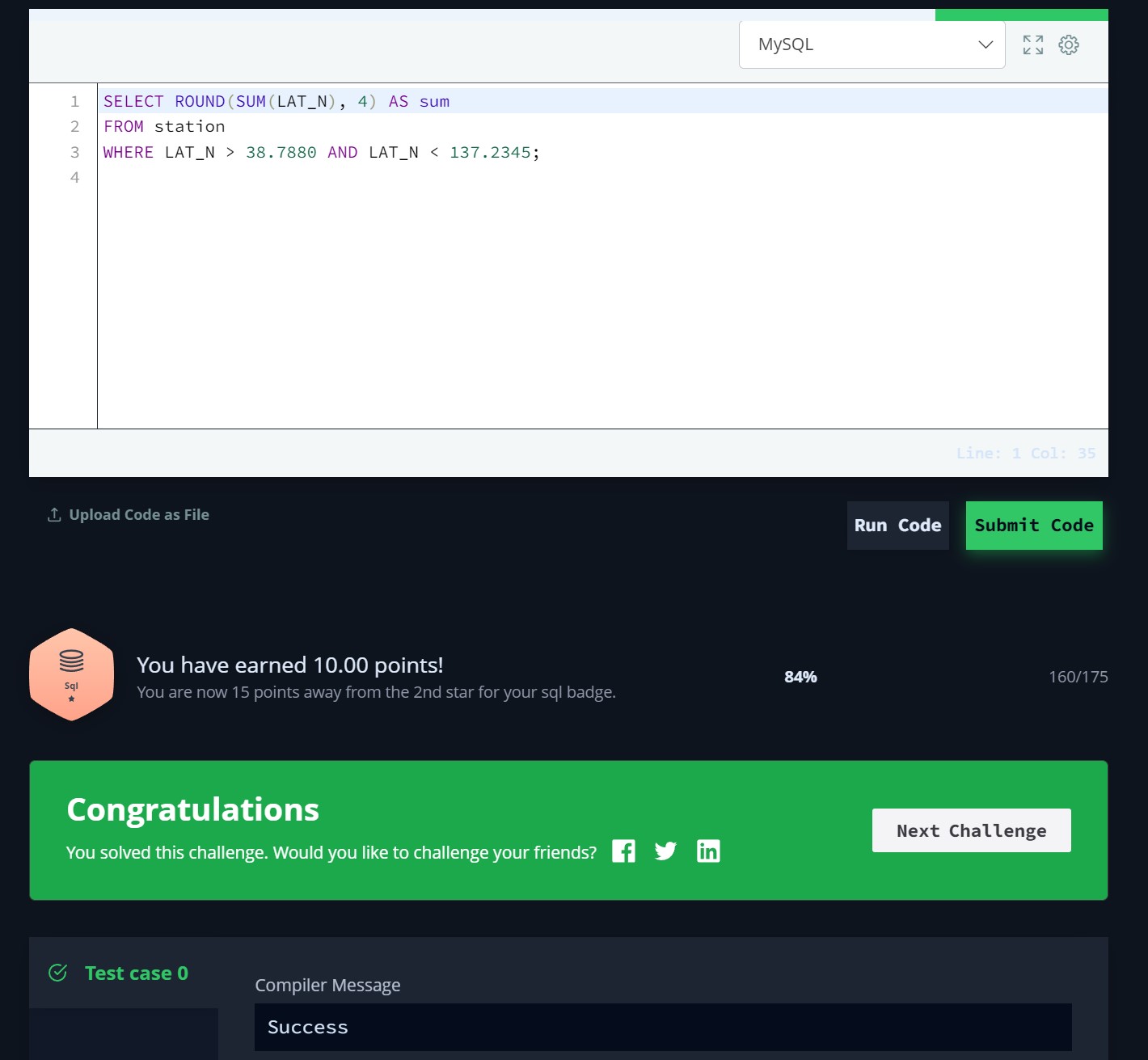
SQL Script Solution:

SELECT ROUND(SUM(LAT\_N), 4) AS sum

FROM station

WHERE LAT\_N > 38.7880 AND LAT\_N < 137.2345;

Output Screenshot:



1. [The Report](https://www.hackerrank.com/challenges/the-report/problem)

SQL Script Solution:

SELECT

CASE

WHEN grade >= 8 THEN name

ELSE 'NULL'

END AS name, grade, marks

FROM Students s

JOIN Grades g ON s.marks BETWEEN g.min\_mark AND g.max\_mark

ORDER BY grade desc, name asc, marks asc;

Output Screenshot:

