# **Database Assignment 1**

# **Name : Yujan Basnet**

# **Roll No: 12**

# **Que1:** [Japanese Cities' Names](https://www.hackerrank.com/challenges/japanese-cities-name/problem)

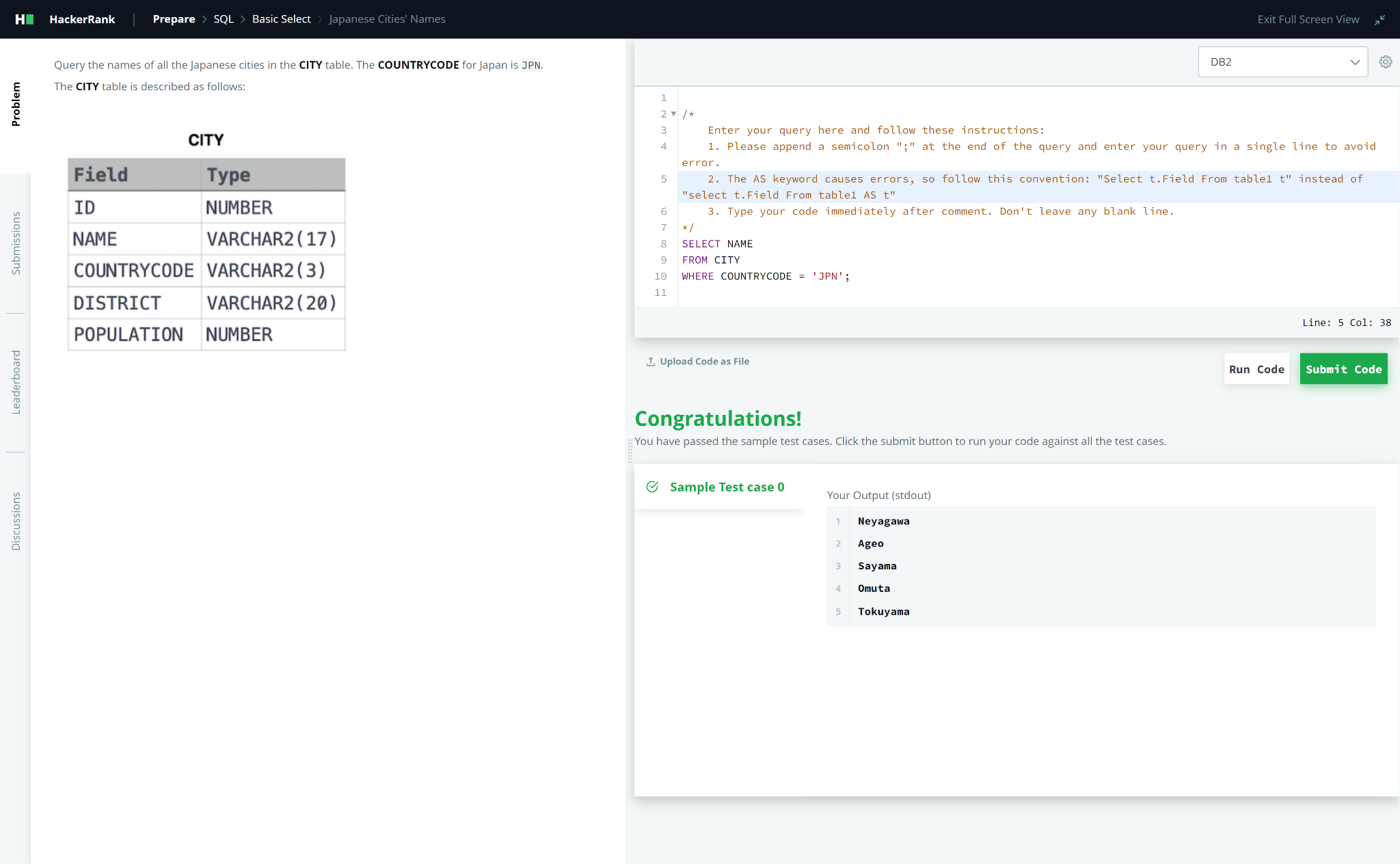
***SQL Script Solution:***

SELECT NAME

FROM CITY

WHERE COUNTRYCODE = 'JPN';

***Screenshot:***



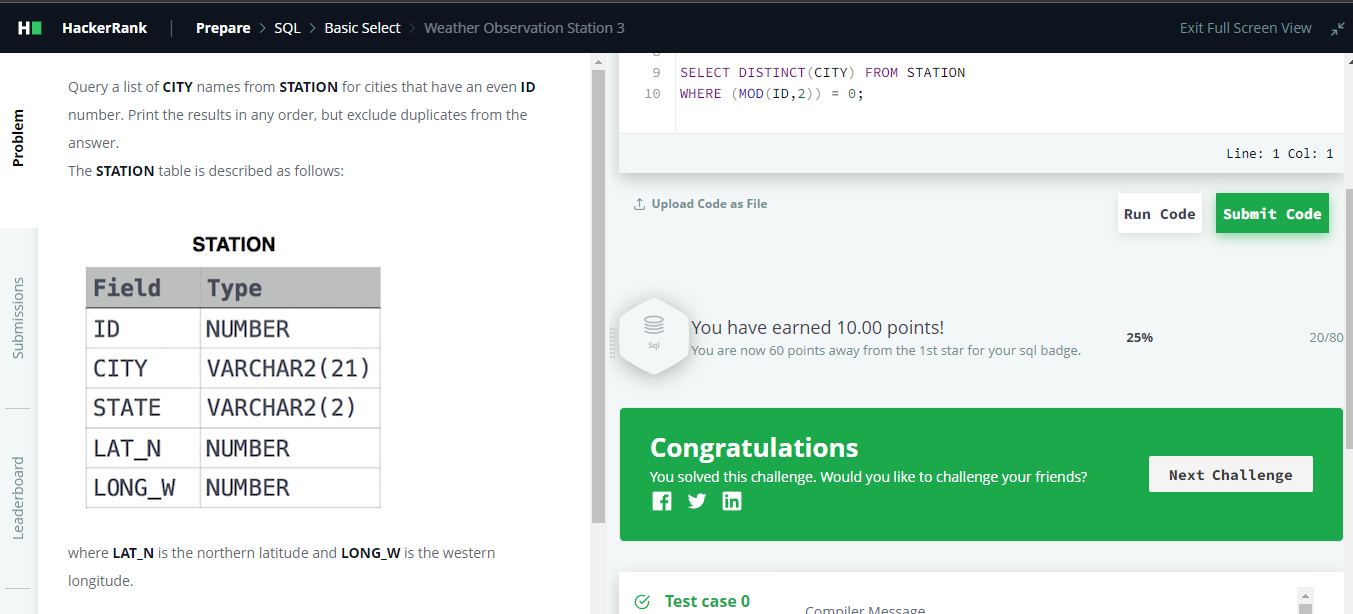
# **Que2:** [**Weather Observation Station 3**](https://www.hackerrank.com/challenges/weather-observation-station-3/problem?isFullScreen=true)

***SQL Script Solution:***

SELECT DISTINCT(CITY) FROM STATION

WHERE (MOD(ID,2)) = 0;

***Screenshot:***



# **Que3:** [**Weather Observation Station 5**](https://www.hackerrank.com/challenges/weather-observation-station-5/problem)

***SQL Script Solution:***

-- SMALLEST

SELECT CITY,LENGTH(CITY) FROM STATION

ORDER BY LENGTH(CITY),CITY

LIMIT 1;

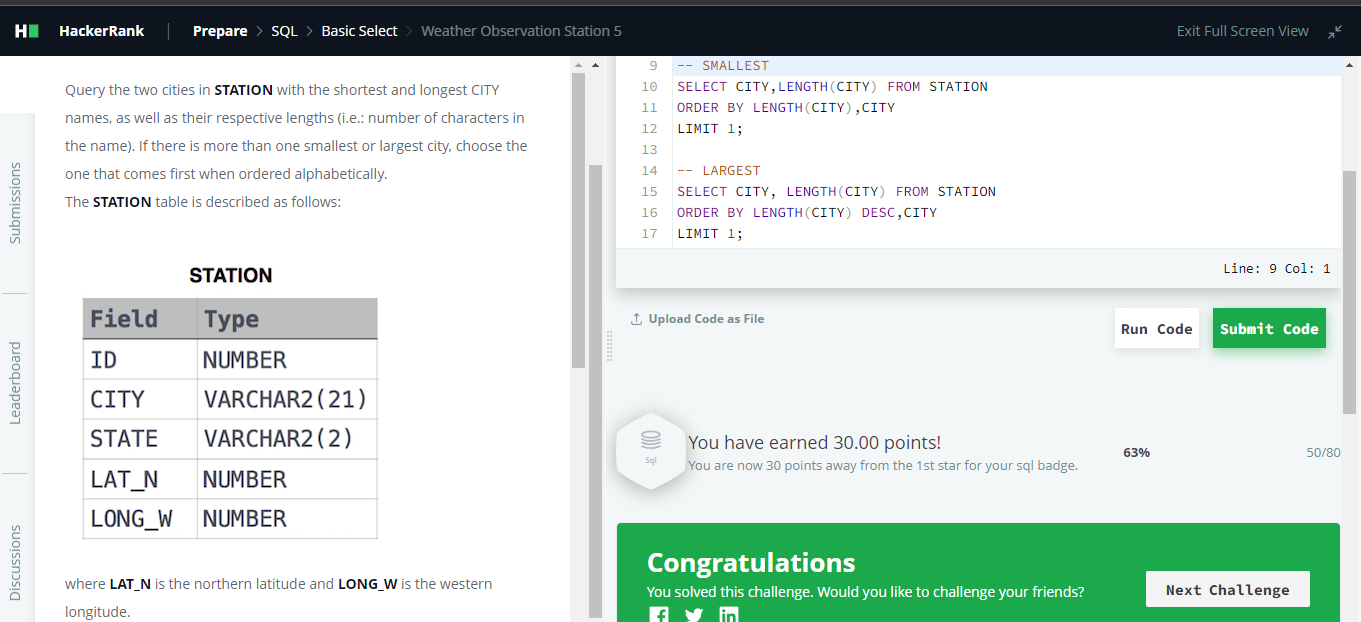
-- LARGEST

SELECT CITY, LENGTH(CITY) FROM STATION

ORDER BY LENGTH(CITY) DESC,CITY

LIMIT 1;

***Screenshot:***



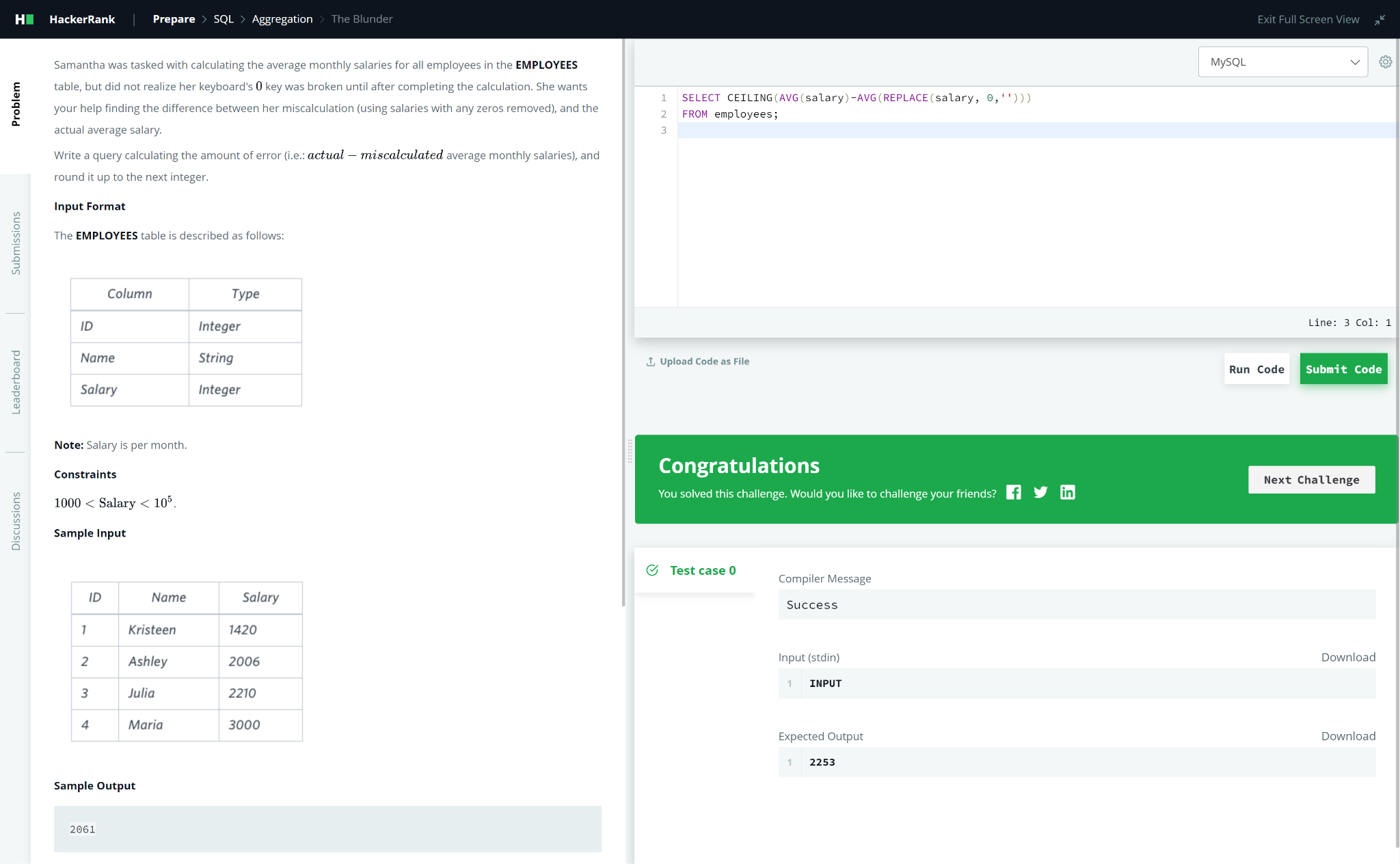
# **Que4:** [**The Blunder**](https://www.hackerrank.com/challenges/the-blunder/problem?isFullScreen=true)

***SQL Script Solution:***

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

FROM employees;

***Screenshot:***

******

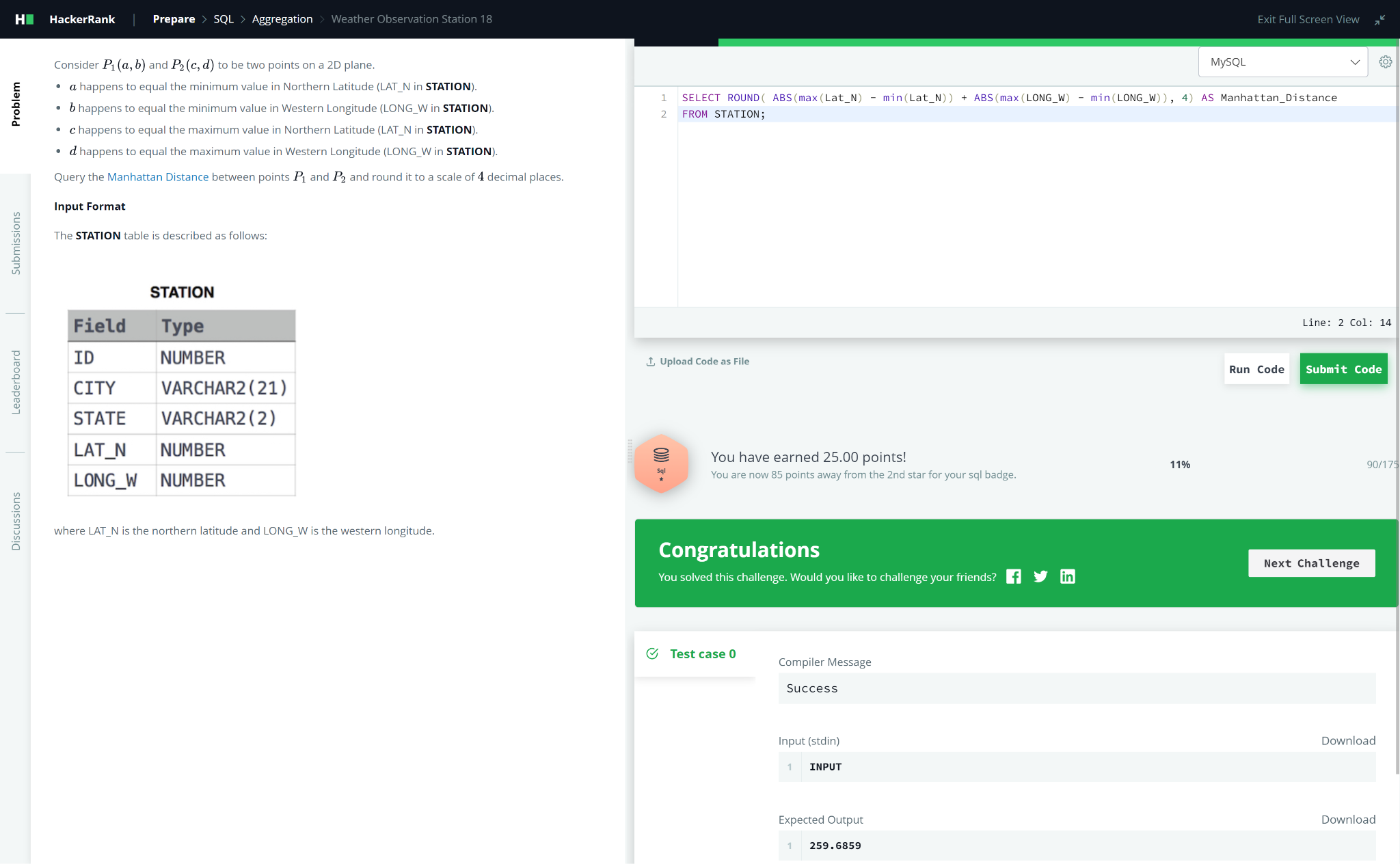
# **Que5:** [Weather Observation Station 18](https://www.hackerrank.com/challenges/weather-observation-station-18/problem)

***SQL Script Solution:***

SELECT ROUND( ABS(max(Lat\_N) - min(Lat\_N)) + ABS(max(LONG\_W) - min(LONG\_W)), 4) AS Manhattan\_Distance

FROM STATION;

***Screenshot:***



# **Que6:** [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

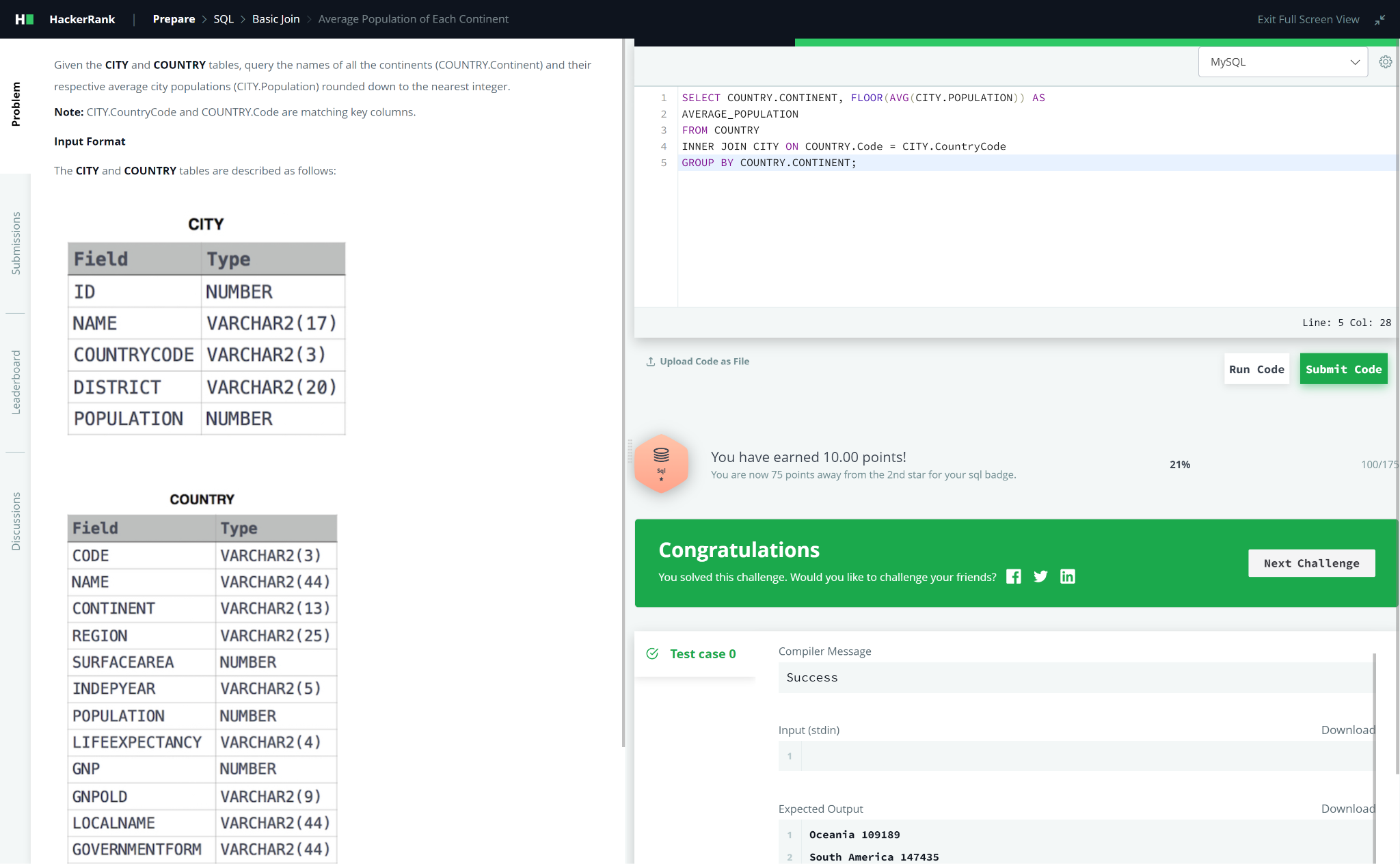
AVERAGE\_POPULATION

FROM COUNTRY

INNER JOIN CITY ON COUNTRY.Code = CITY.CountryCode

GROUP BY COUNTRY.CONTINENT;

***Screenshot:***



# **Que7:** [The PADS](https://www.hackerrank.com/challenges/the-pads/problem)

***SQL Script Solution:***

SELECT

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

FROM occupations

ORDER BY name;

SELECT

CONCAT(

"There are a total of ",

COUNT(occupation),

" ",

LOWER(occupation),

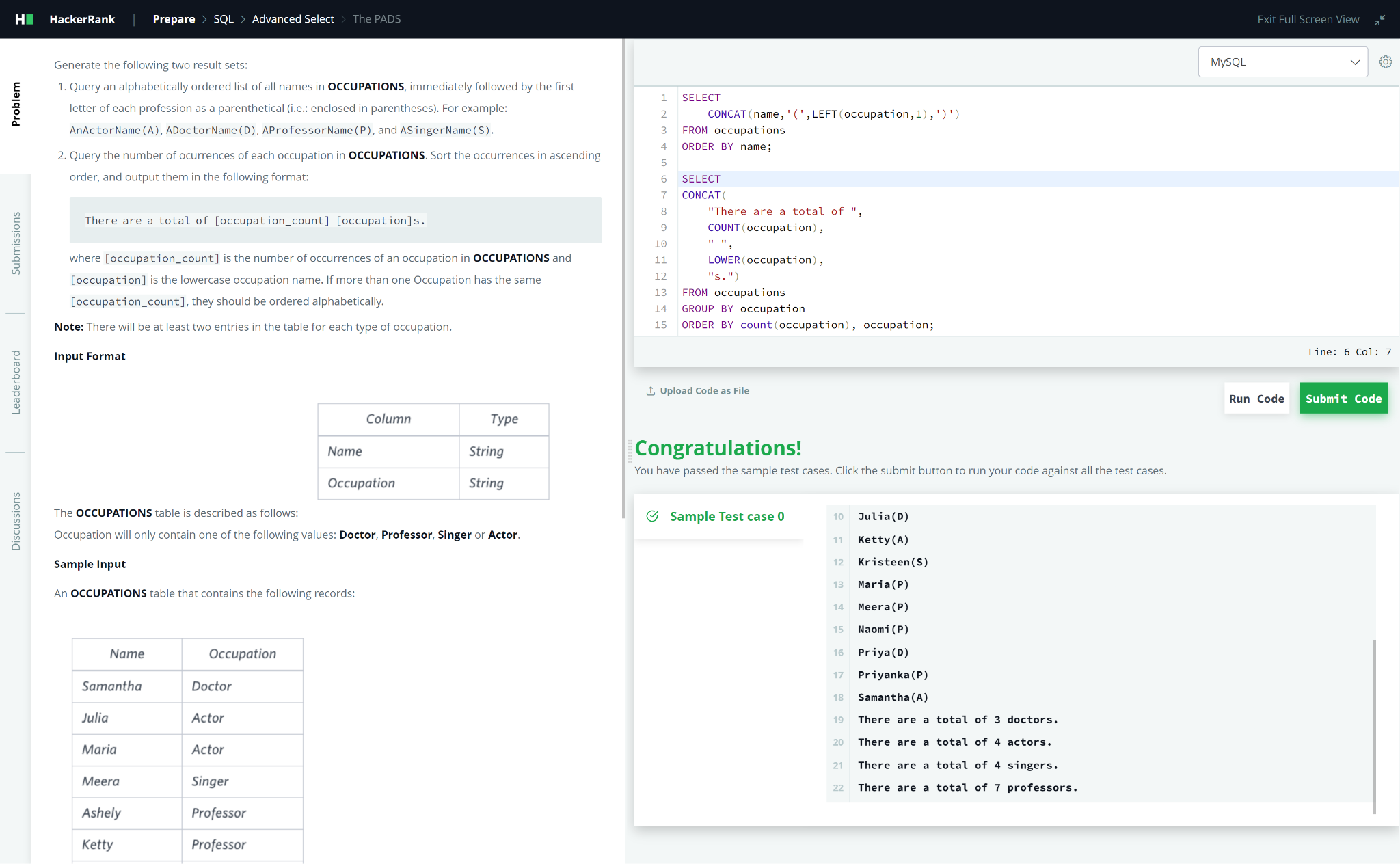
"s.")

FROM occupations

GROUP BY occupation

ORDER BY count(occupation), occupation;

***Screenshot:***



# **Que8:** [Type of Triangle](https://www.hackerrank.com/challenges/what-type-of-triangle/problem?isFullScreen=true)

***SQL Script Solution:***

SELECT

CASE

WHEN a + b <= c OR b + c <= a OR a + c <= b THEN 'Not A Triangle'

WHEN a=b AND b=c AND a=c THEN 'Equilateral'

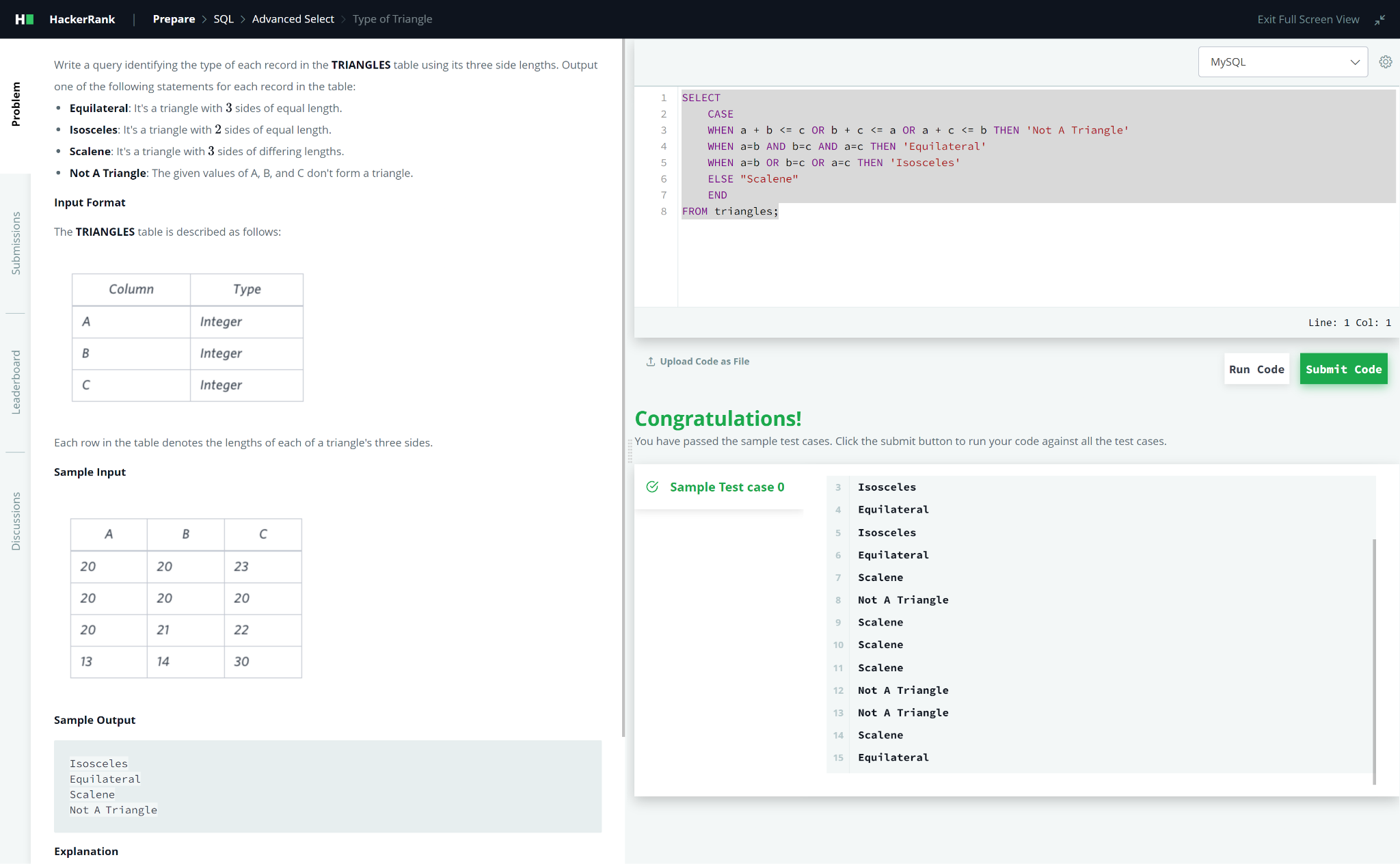
WHEN a=b OR b=c OR a=c THEN 'Isosceles'

ELSE "Scalene"

END

FROM triangles;

***Screenshot:***

******

# **Que9:** [Weather Observation Station 13](https://www.hackerrank.com/challenges/weather-observation-station-13/problem)

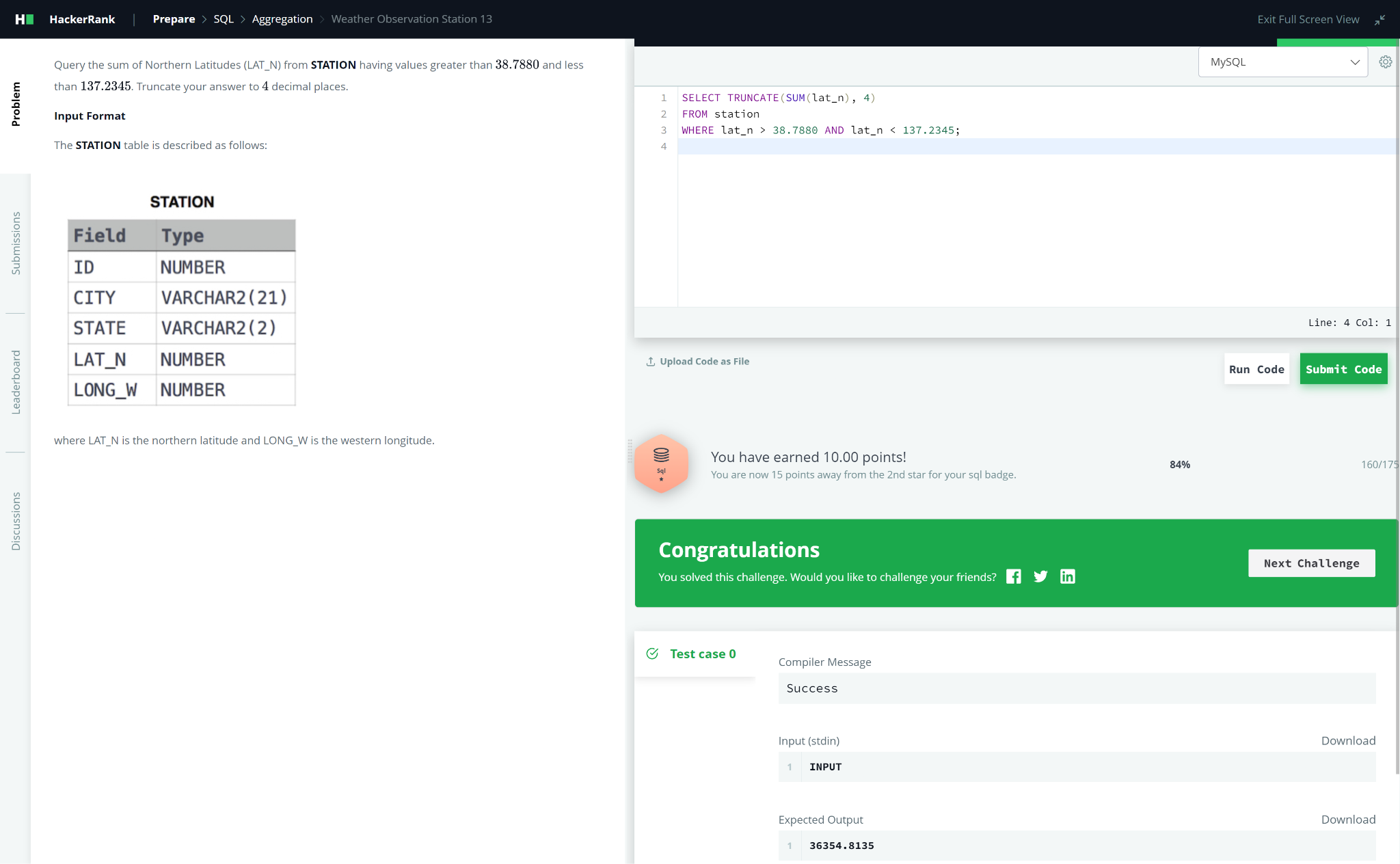
***SQL Script Solution:***

SELECT TRUNCATE(SUM(lat\_n), 4)

FROM station

WHERE lat\_n > 38.7880 AND lat\_n < 137.2345;

***Screenshot:***



# **Que10:** [The Report](https://www.hackerrank.com/challenges/the-report/problem)

***SQL Script Solution:***

SELECT

CASE

WHEN s.marks <70 THEN null

ELSE s.name

END,

(SELECT

grade

FROM grades WHERE s.marks >= min\_mark AND s.marks <= max\_mark) AS grade,

s.marks

FROM students s

ORDER BY grade DESC, s.name ASC, s.marks ASC;

***Screenshot:***

