Database Assignment 1

Name : Yujan Basnet

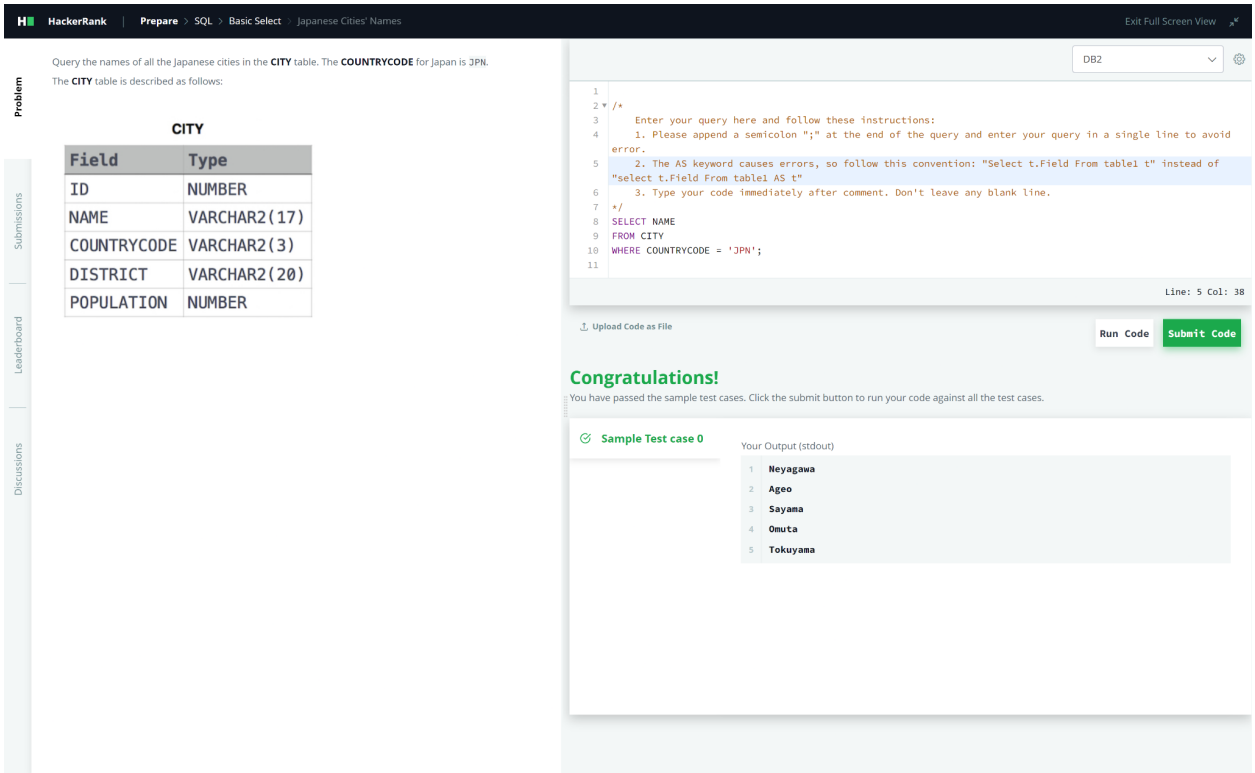
Roll No: 12

Que1: Japanese Cities' Names

SQL Script Solution:

```
SELECT NAME
FROM CITY
WHERE COUNTRYCODE = 'JPN';
```

Screenshot:

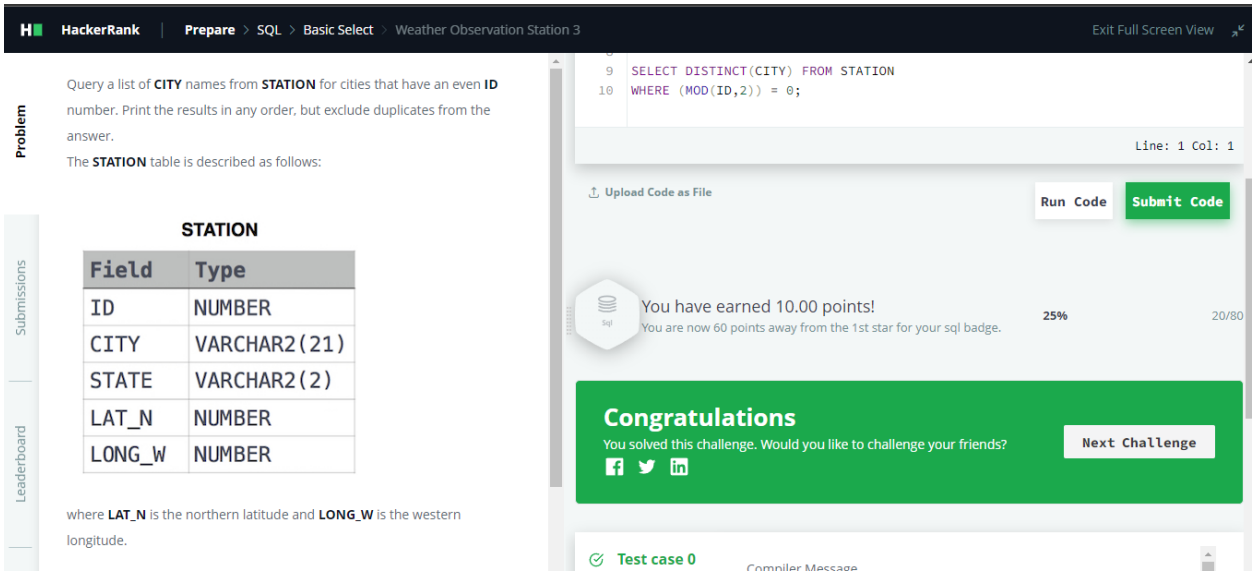


Que2: Weather Observation Station 3

SQL Script Solution:

```
SELECT DISTINCT(CITY) FROM STATION
WHERE (MOD(ID,2)) = 0;
```

Screenshot:



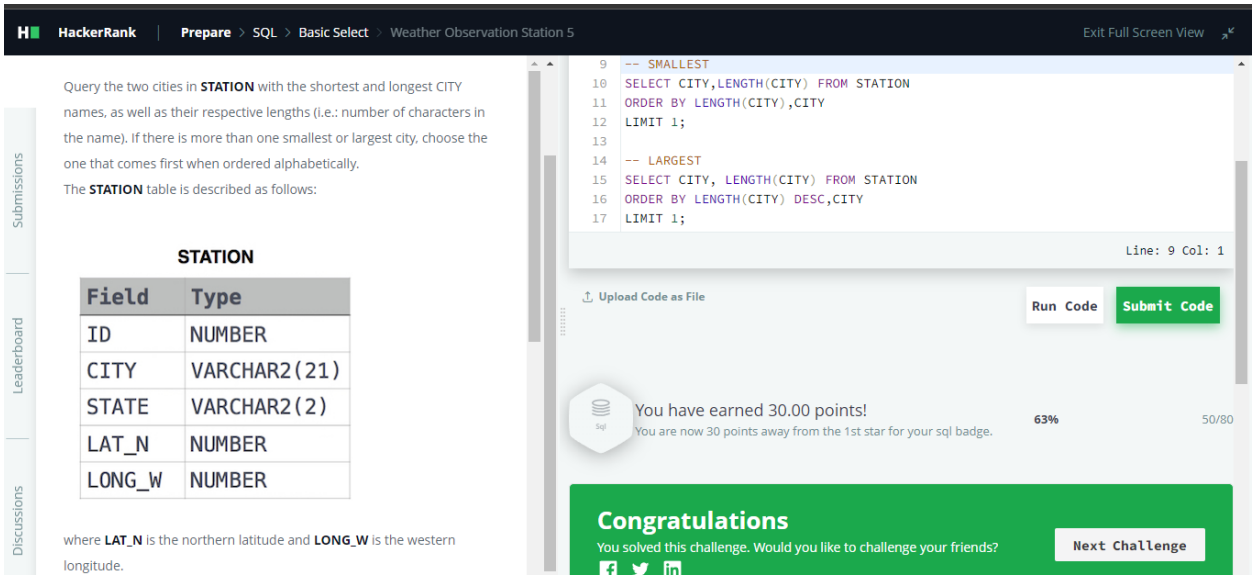
Que3: [Weather Observation Station 5]

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:

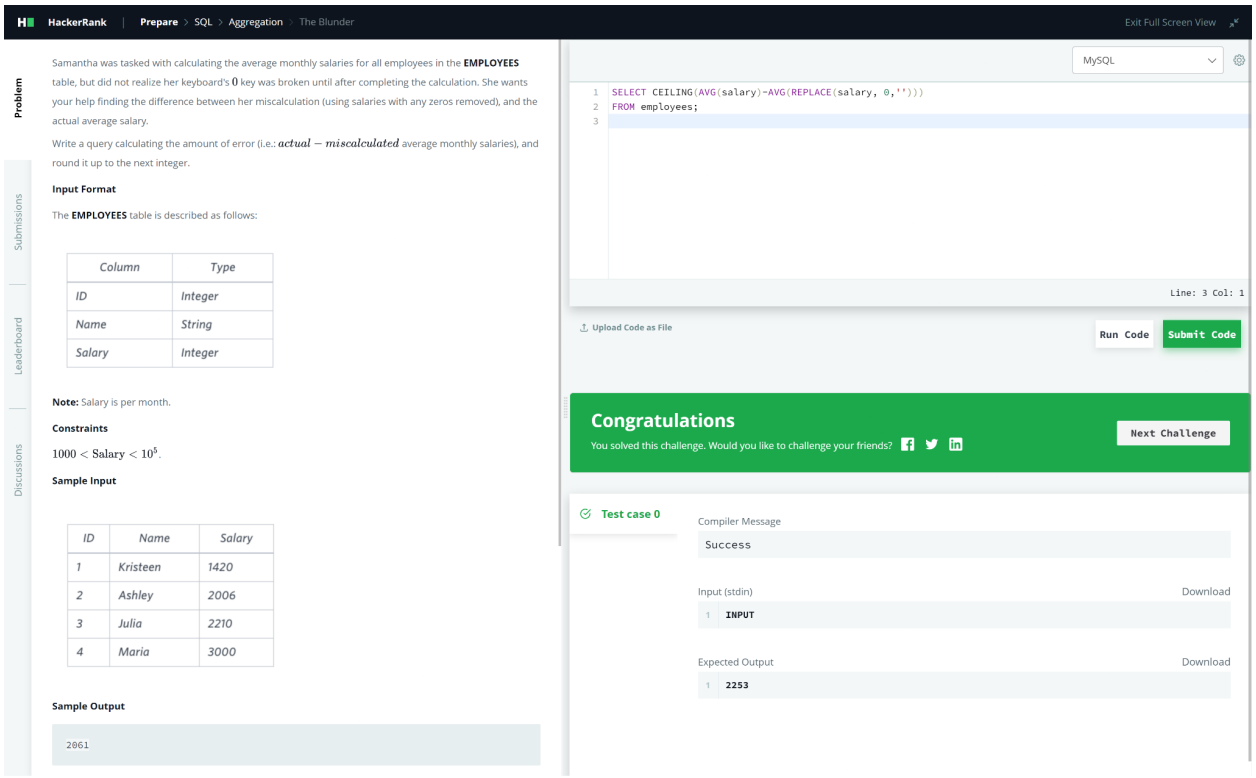


Que4: The Blunder

SQL Script Solution:

```
SELECT CEILING(AVG(salary)-AVG(REPLACE(salary, 0,'')))
FROM employees;
```

Screenshot:



HackerRank | Prepare > SQL > Aggregation > The Blunder — Exit Full Screen View

Problem | Submissions | Leaderboard | Discussions

Samantha was tasked with calculating the average monthly salaries for all employees in the **EMPLOYEES** table, but did not realize her keyboard's **0** key was broken until after completing the calculation. She wants your help finding the difference between her miscalculation (using salaries with any zeros removed), and the actual average salary.

Write a query calculating the amount of error (i.e.: $actual - miscalculated$ average monthly salaries), and round it up to the next integer.

Input Format

The **EMPLOYEES** table is described as follows:

Column	Type
ID	Integer
Name	String
Salary	Integer

Note: Salary is per month.

Constraints

$1000 < \text{Salary} < 10^5$.

Sample Input

ID	Name	Salary
1	Kristeen	1420
2	Ashley	2006
3	Julia	2210
4	Maria	3000

Sample Output

```
2061
```

MySQL

```
1 SELECT CEILING(AVG(salary)-AVG(REPLACE(salary, 0,'')))
2 FROM employees;
3
```

Line: 3 Col: 1

Upload Code as File | Run Code | Submit Code

Congratulations

You solved this challenge. Would you like to challenge your friends? Next Challenge

Test case 0

Compiler Message: Success

Input (stdin) — Download

```
1 INPUT
```

Expected Output — Download

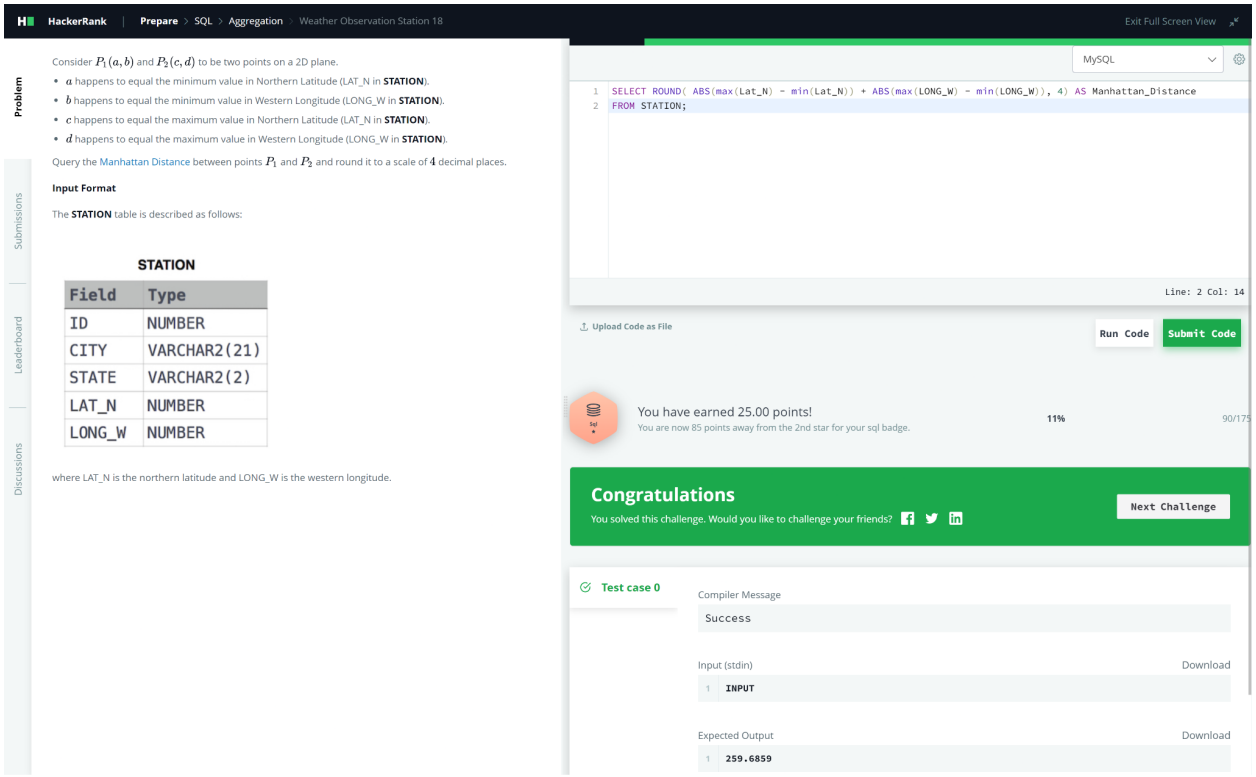
```
1 2253
```

Que5: Weather Observation Station 18

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:

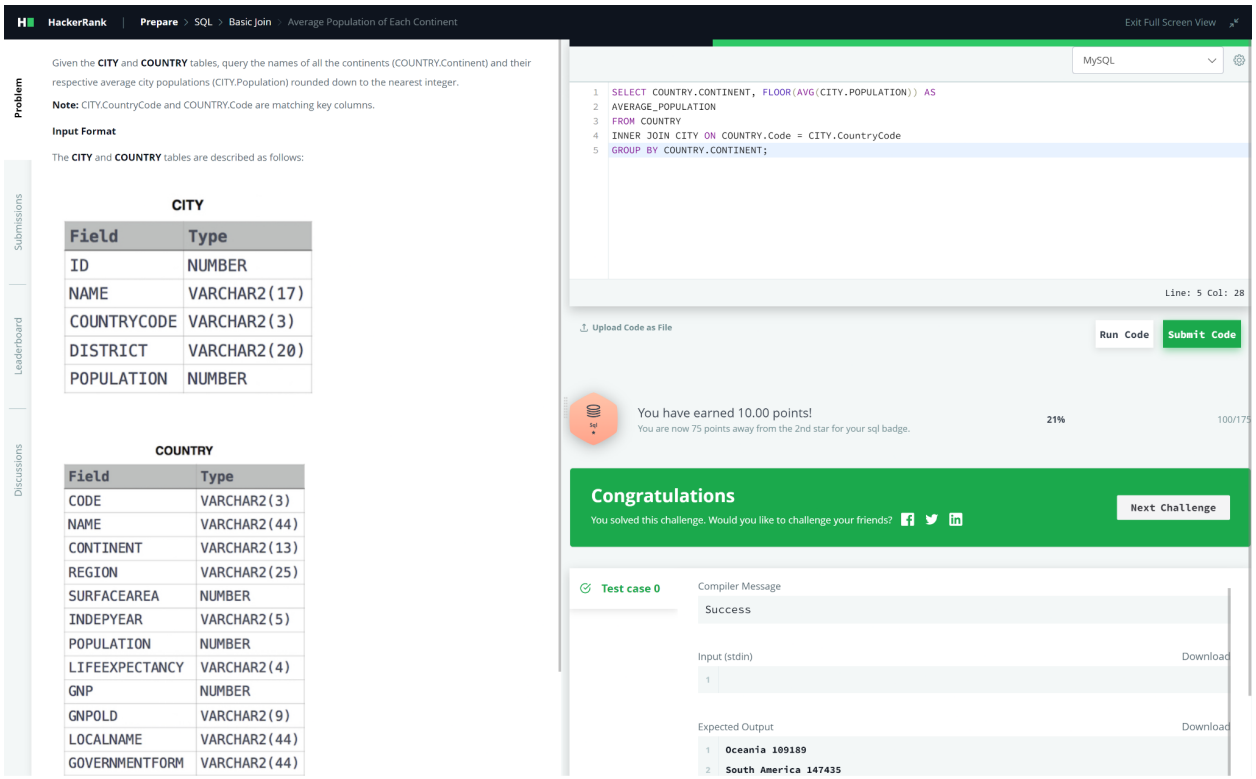


Que6: Average Population of Each Continent

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:



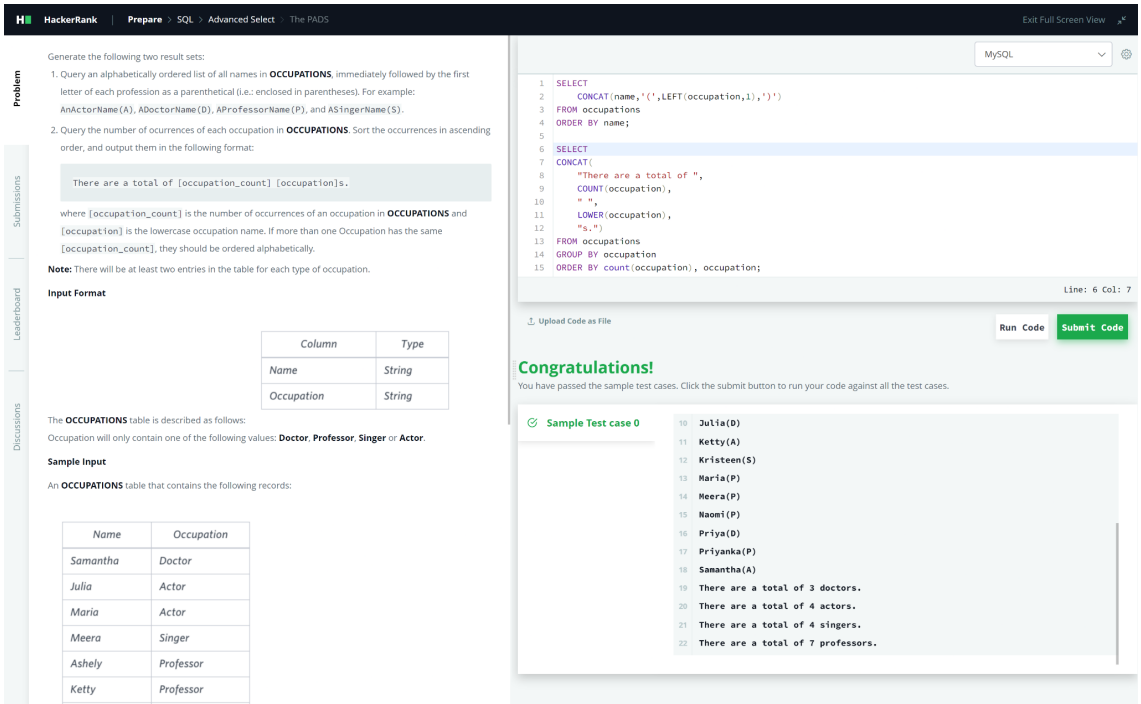
Que7: The PADS

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:

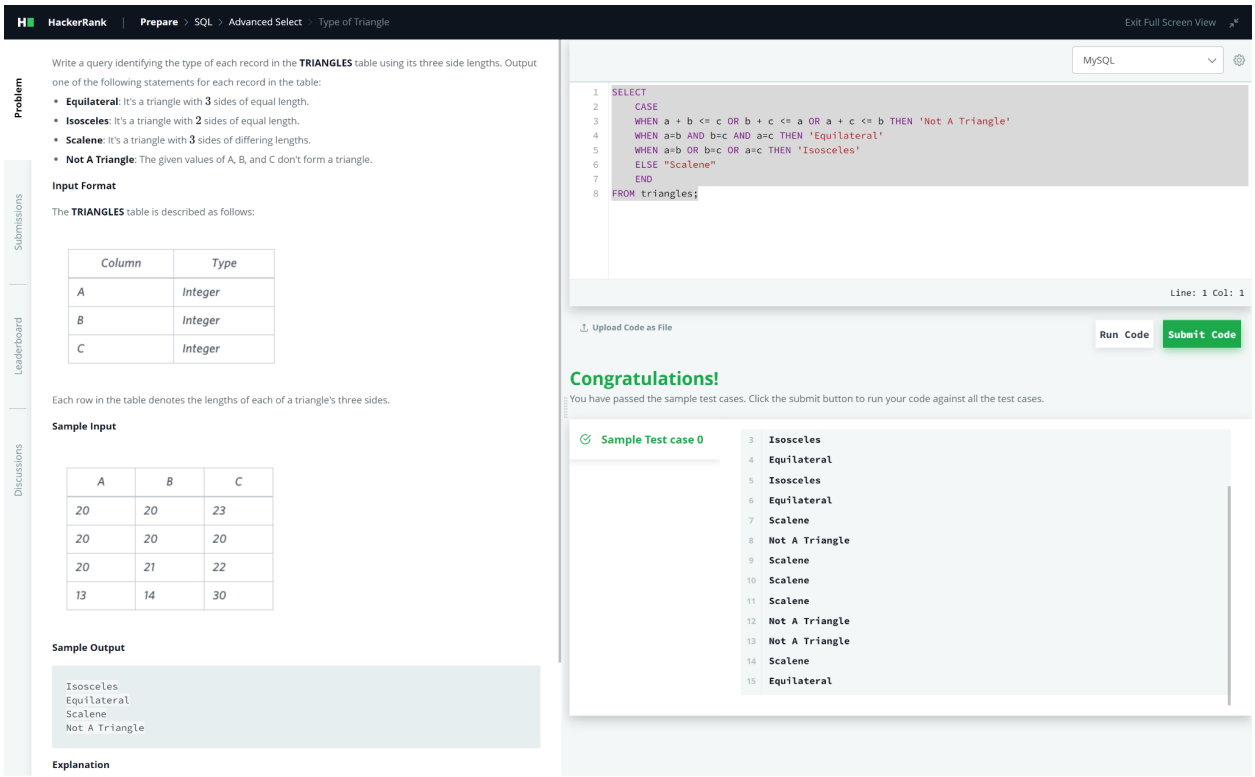


Que8: Type of Triangle

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:



HackerRank | Prepare > SQL > Advanced Select > Type of Triangle — Exit Full Screen View

Problem | Submissions | Leaderboard | Discussions

Write a query identifying the type of each record in the **TRIANGLES** table using its three side lengths. Output one of the following statements for each record in the table:

- **Equilateral**: It's a triangle with 3 sides of equal length.
- **Isosceles**: It's a triangle with 2 sides of equal length.
- **Scalene**: It's a triangle with 3 sides of differing lengths.
- **Not A Triangle**: The given values of A, B, and C don't form a triangle.

Input Format

The **TRIANGLES** table is described as follows:

Column	Type
A	Integer
B	Integer
C	Integer

Each row in the table denotes the lengths of each of a triangle's three sides.

Sample Input

A	B	C
20	20	23
20	20	20
20	21	22
13	14	30

Sample Output

```
Isosceles
Equilateral
Scalene
Not A Triangle
```

Explanation

MySQL

```
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;
```

Line: 1 Col: 1

Upload Code as File | Run Code | Submit Code

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Sample Test case 0

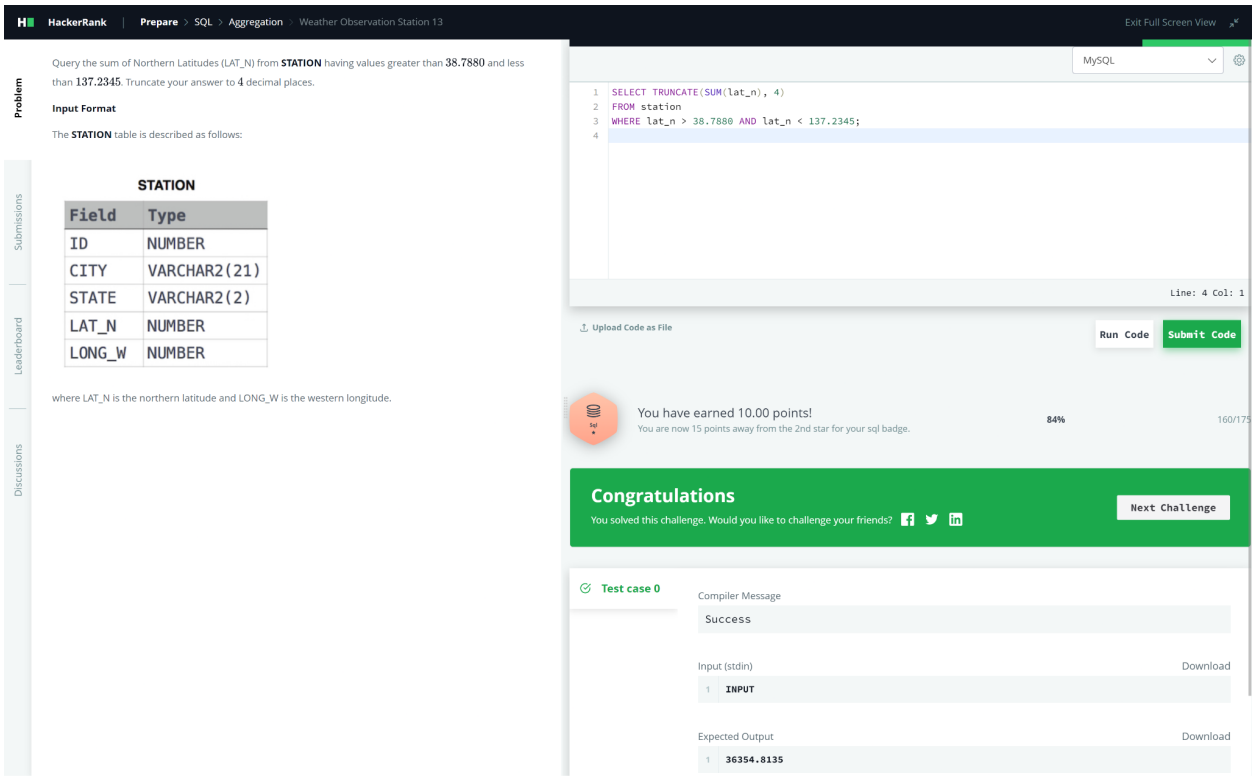
```
Isosceles
Equilateral
Isosceles
Equilateral
Scalene
Not A Triangle
Scalene
Scalene
Scalene
Not A Triangle
Not A Triangle
Scalene
Equilateral
```

Que9: Weather Observation Station 13

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

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: