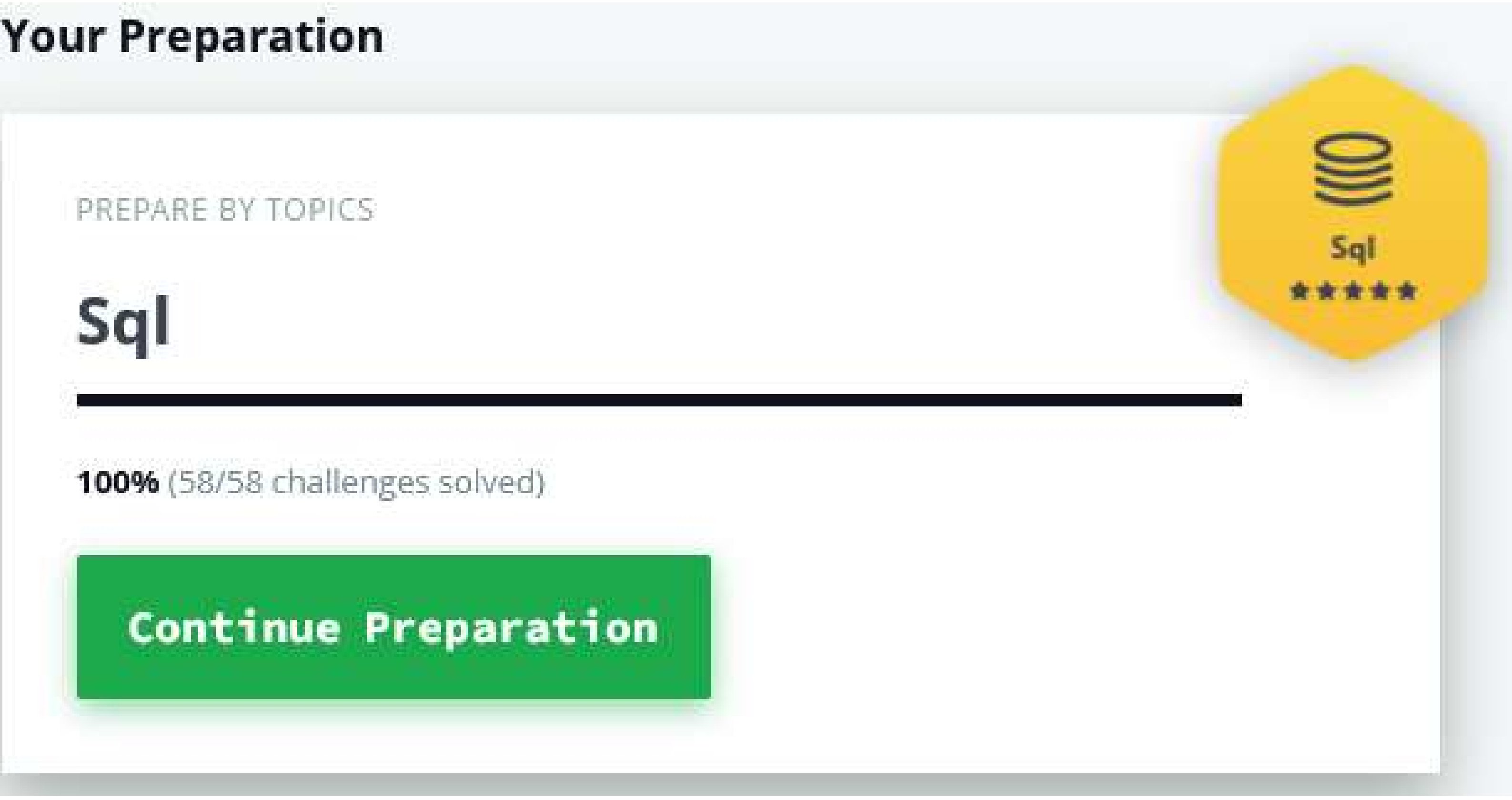
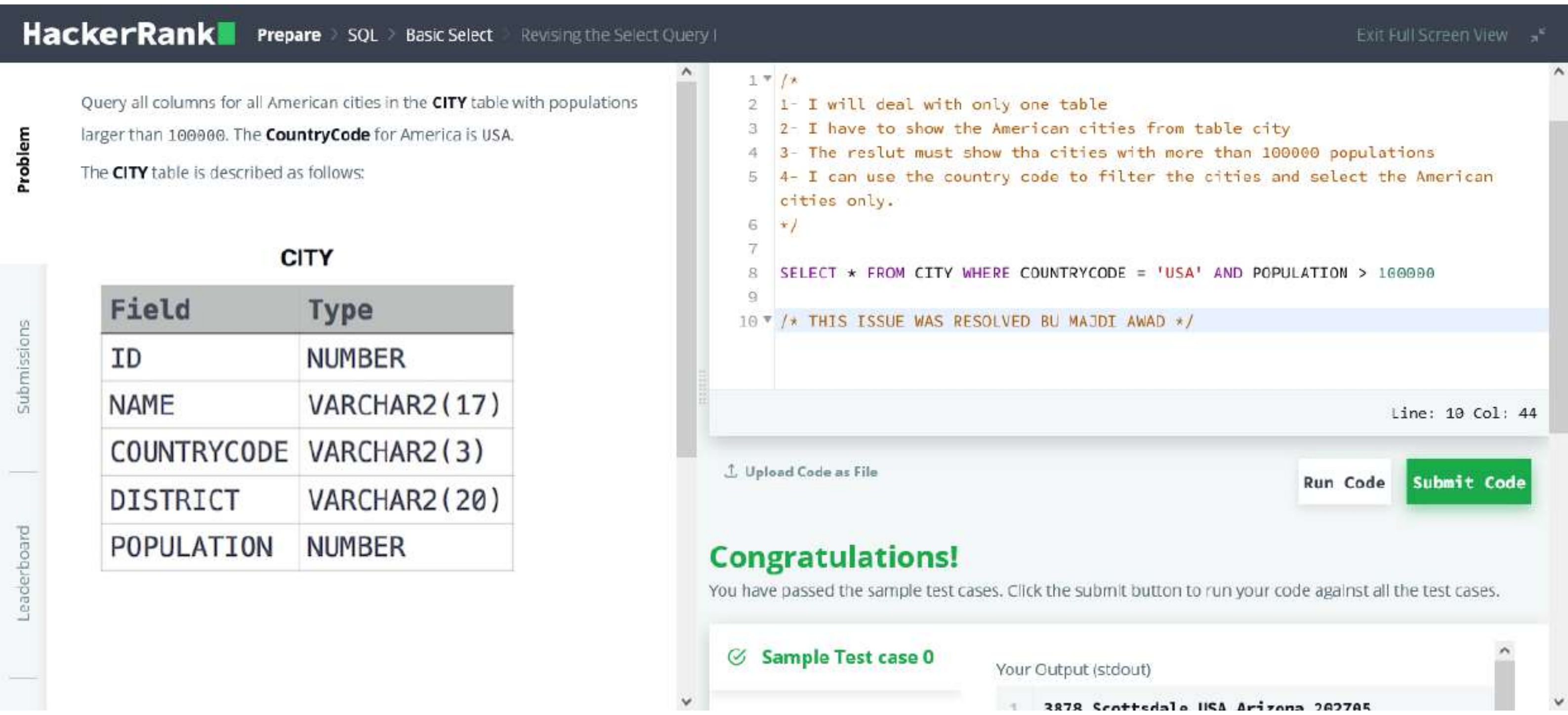


Your Preparation

PREPARE BY TOPICS

Sql

100% (58/58 challenges solved)



Query all columns for all American cities in the **CITY** table with populations larger than `100000`. The **CountryCode** for America is `USA`.

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

CITY

Field	Type
ID	NUMBER
NAME	VARCHAR2(17)
COUNTRYCODE	VARCHAR2(3)
DISTRICT	VARCHAR2(20)
POPULATION	NUMBER

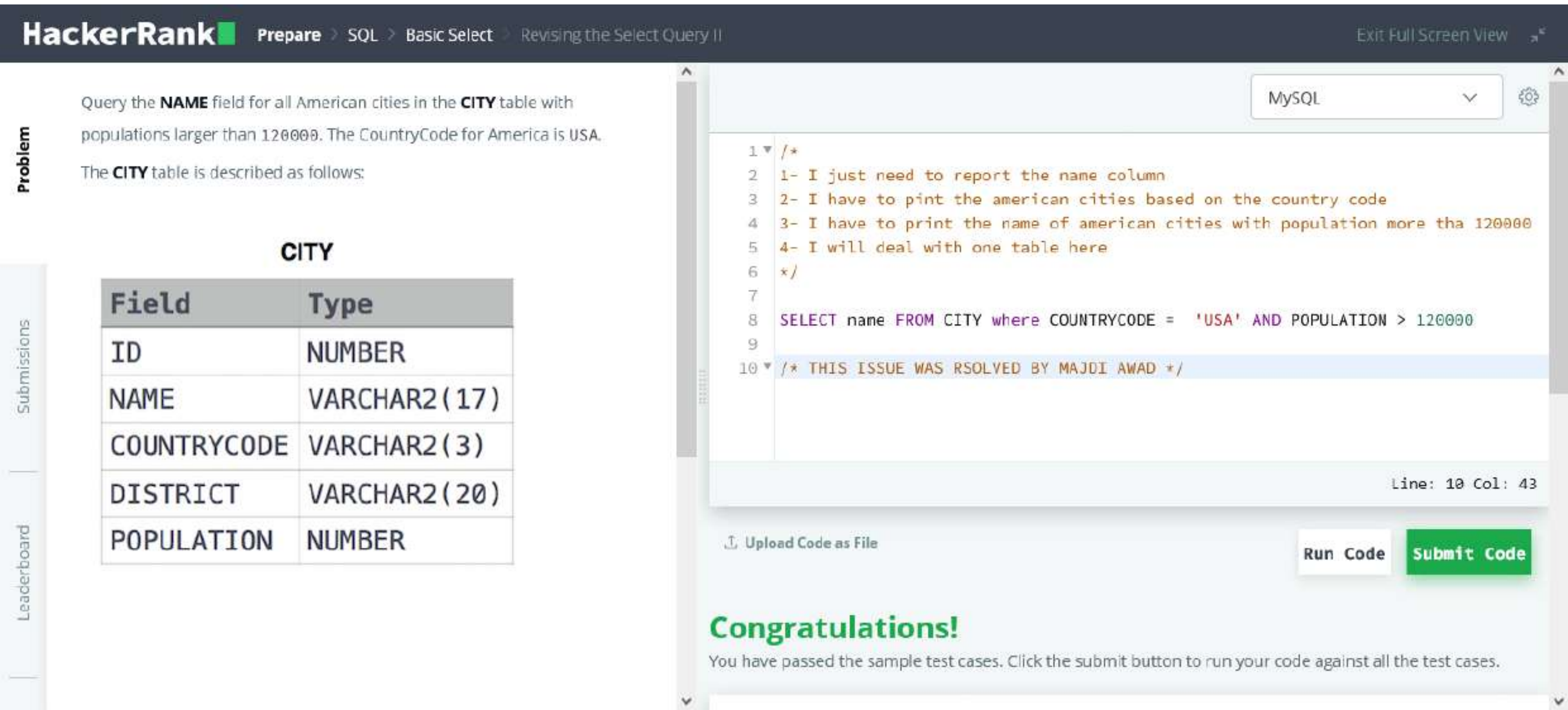
```
/*
1- I will deal with only one table
2- I have to show the American cities from table city
3- The reslut must show tha cities with more than 100000 populations
4- I can use the country code to filter the cities and select the American
cities only.
*/

SELECT * FROM CITY WHERE COUNTRYCODE = 'USA' AND POPULATION > 100000

/* THIS ISSUE WAS RESOLVED BU MAJDI AWAD */
```

Congratulations!

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

Your Output (stdout)

```
3878 Scottsdale USA Arizona 202705
```



Query the **NAME** field for all American cities in the **CITY** table with populations larger than `120000`. The CountryCode for America is `USA`.

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

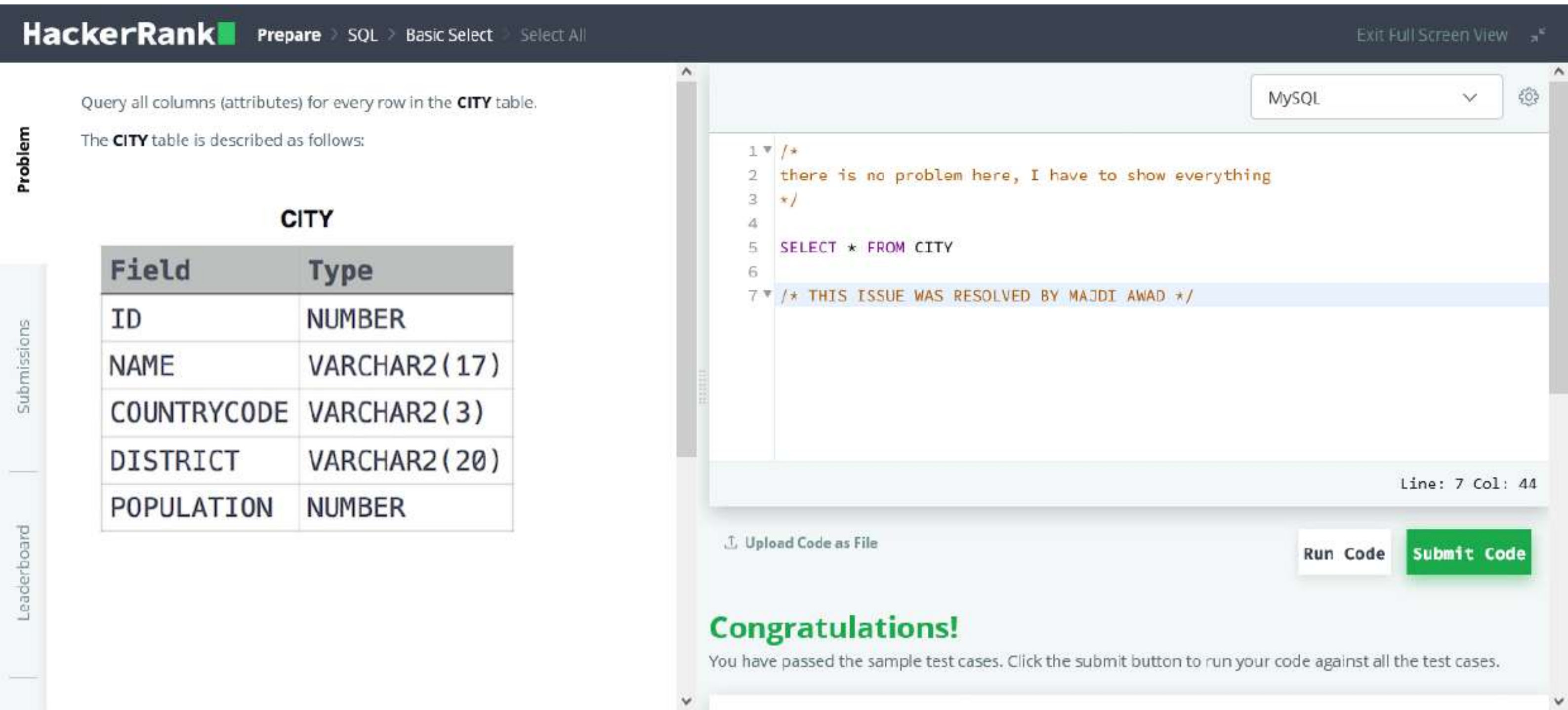
CITY

Field	Type
ID	NUMBER
NAME	VARCHAR2(17)
COUNTRYCODE	VARCHAR2(3)
DISTRICT	VARCHAR2(20)
POPULATION	NUMBER

```sql
/*
1- I just need to report the name column
2- I have to pint the american cities based on the country code
3- I have to print the name of american cities with population more tha 120000
4- I will deal with one table here
*/

SELECT name FROM CITY where COUNTRYCODE =  'USA' AND POPULATION > 120000

/* THIS ISSUE WAS RSOLVED BY MAJDI AWAD */
```

Congratulations!

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



Query all columns (attributes) for every row in the **CITY** table.

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

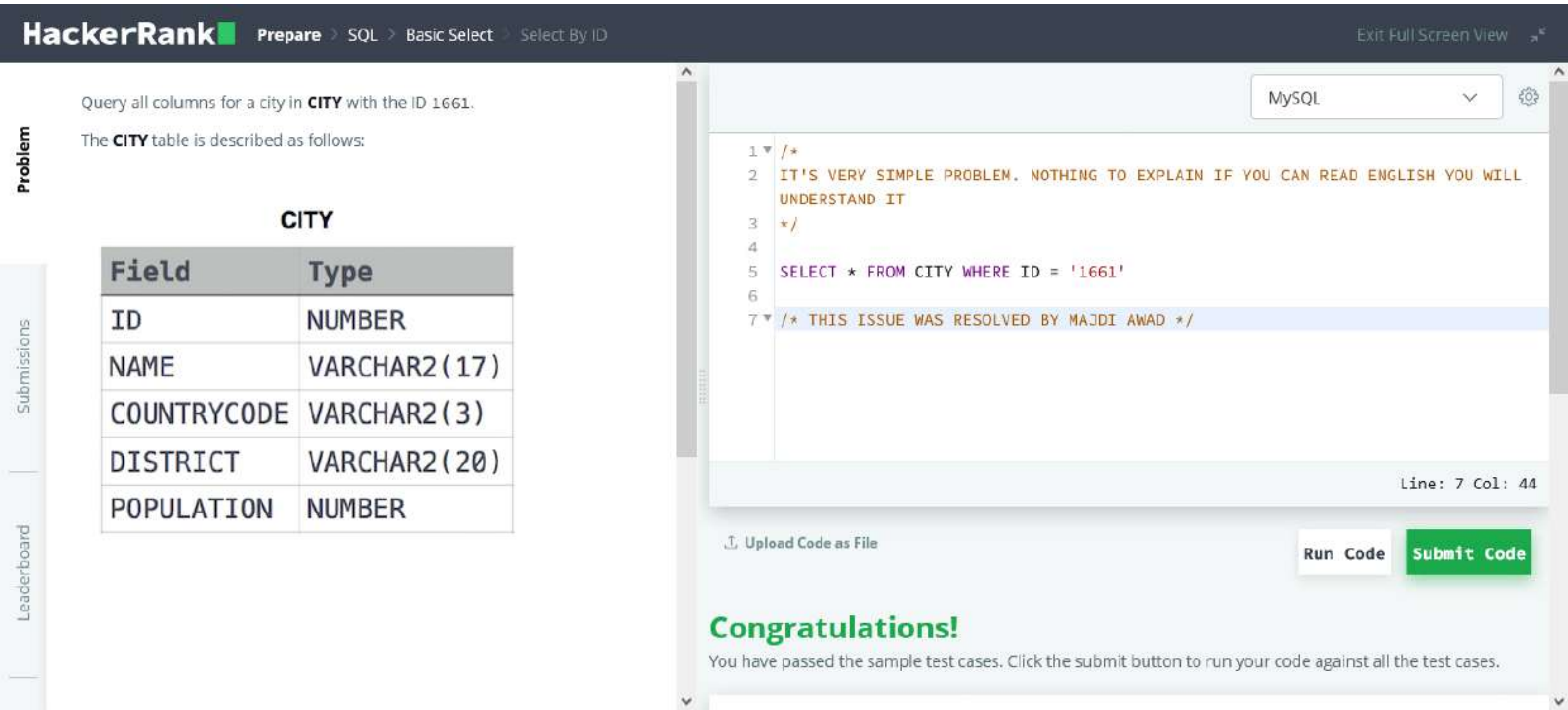
CITY

Field	Type
ID	NUMBER
NAME	VARCHAR2(17)
COUNTRYCODE	VARCHAR2(3)
DISTRICT	VARCHAR2(20)
POPULATION	NUMBER

```sql
/*
there is no problem here, I have to show everything
*/

SELECT * FROM CITY

/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

Congratulations!

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



Query all columns for a city in **CITY** with the ID 1661.

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

CITY

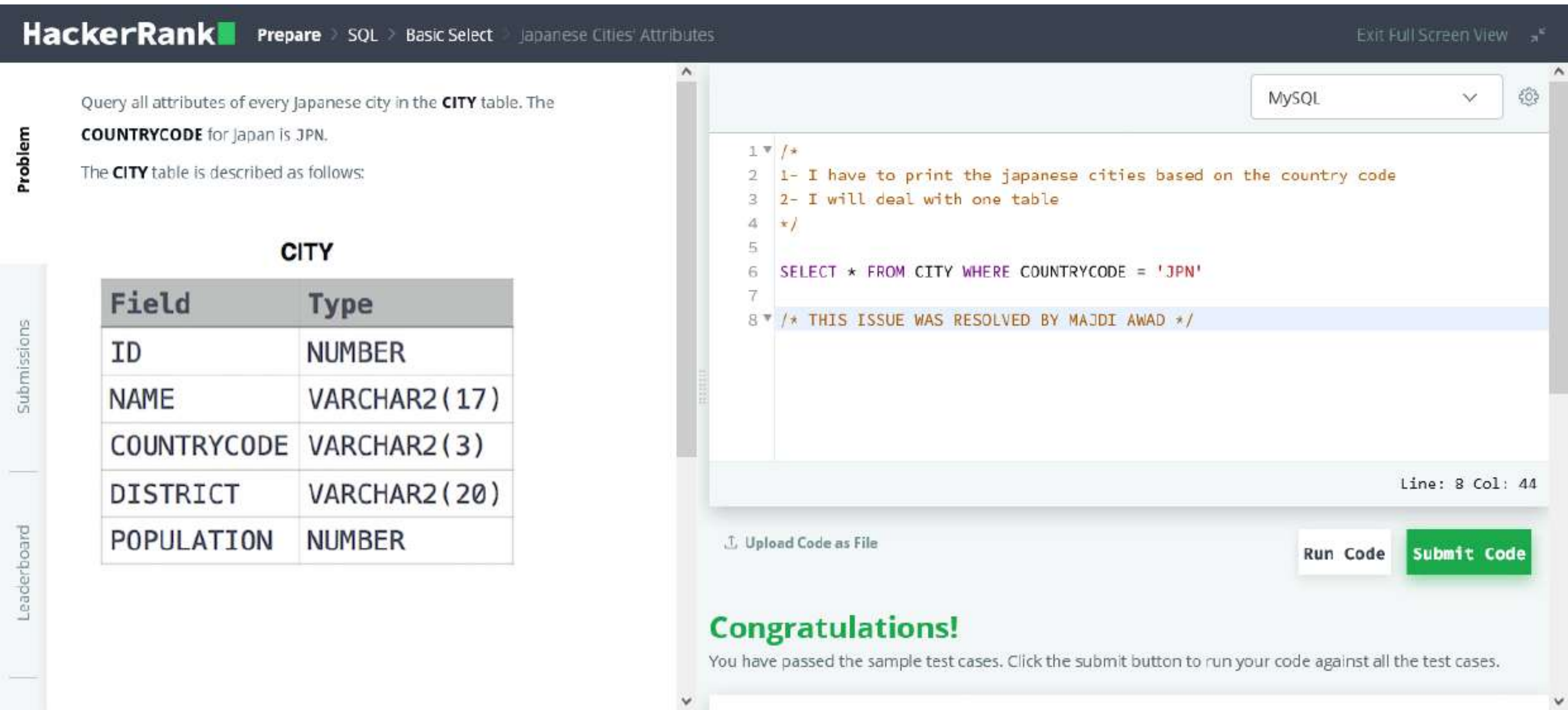
Field	Type
ID	NUMBER
NAME	VARCHAR2(17)
COUNTRYCODE	VARCHAR2(3)
DISTRICT	VARCHAR2(20)
POPULATION	NUMBER

```
/*
IT'S VERY SIMPLE PROBLEM. NOTHING TO EXPLAIN IF YOU CAN READ ENGLISH YOU WILL UNDERSTAND IT
*/

SELECT * FROM CITY WHERE ID = '1661'

/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

Line: 7 Col: 44

Congratulations!

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



Query all attributes of every Japanese city in the **CITY** table. The **COUNTRYCODE** for Japan is JPN.

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

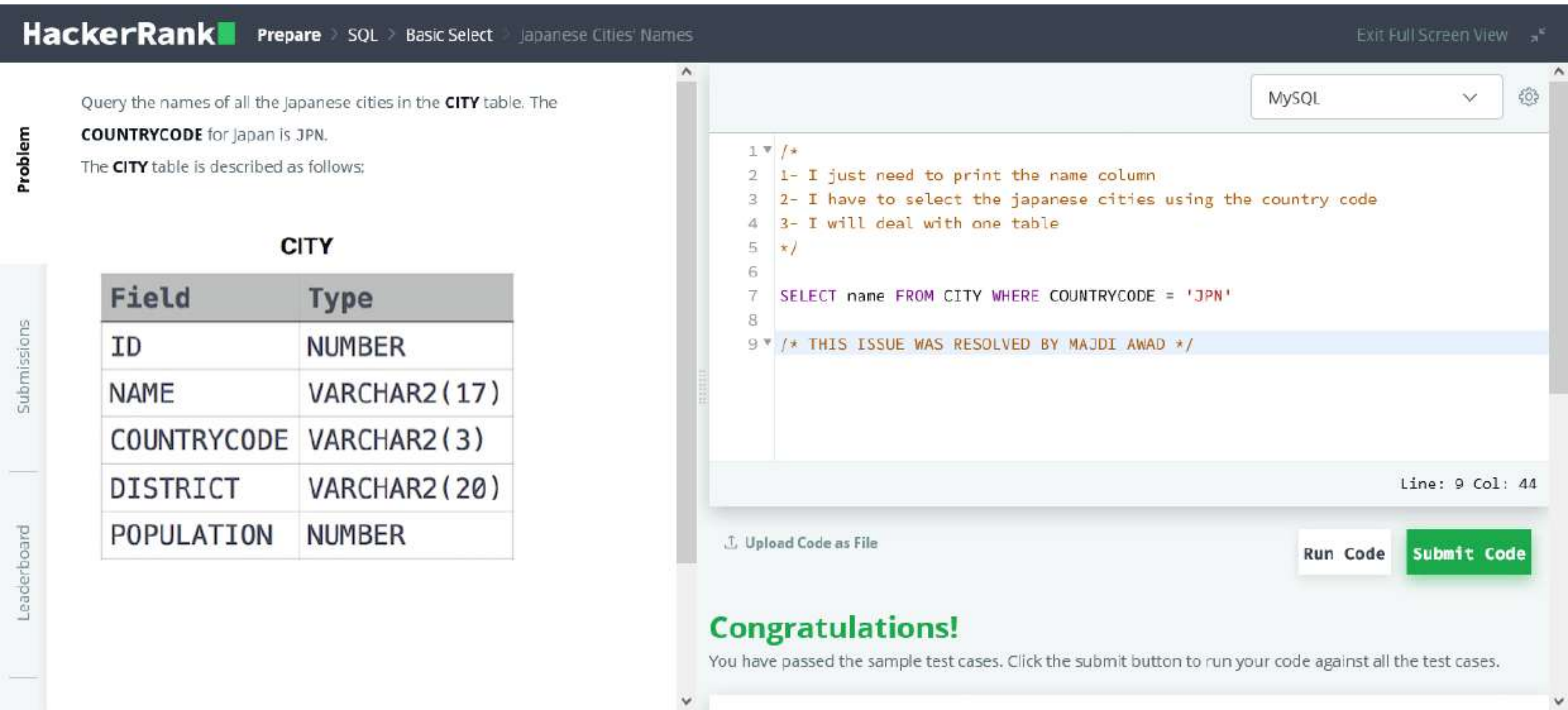
CITY

Field	Type
ID	NUMBER
NAME	VARCHAR2(17)
COUNTRYCODE	VARCHAR2(3)
DISTRICT	VARCHAR2(20)
POPULATION	NUMBER

```
/*
1- I have to print the japanese cities based on the country code
2- I will deal with one table
*/

SELECT * FROM CITY WHERE COUNTRYCODE = 'JPN'

/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

Congratulations!

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



Query the names of all the Japanese cities in the **CITY** table. The **COUNTRYCODE** for Japan is JPN.

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

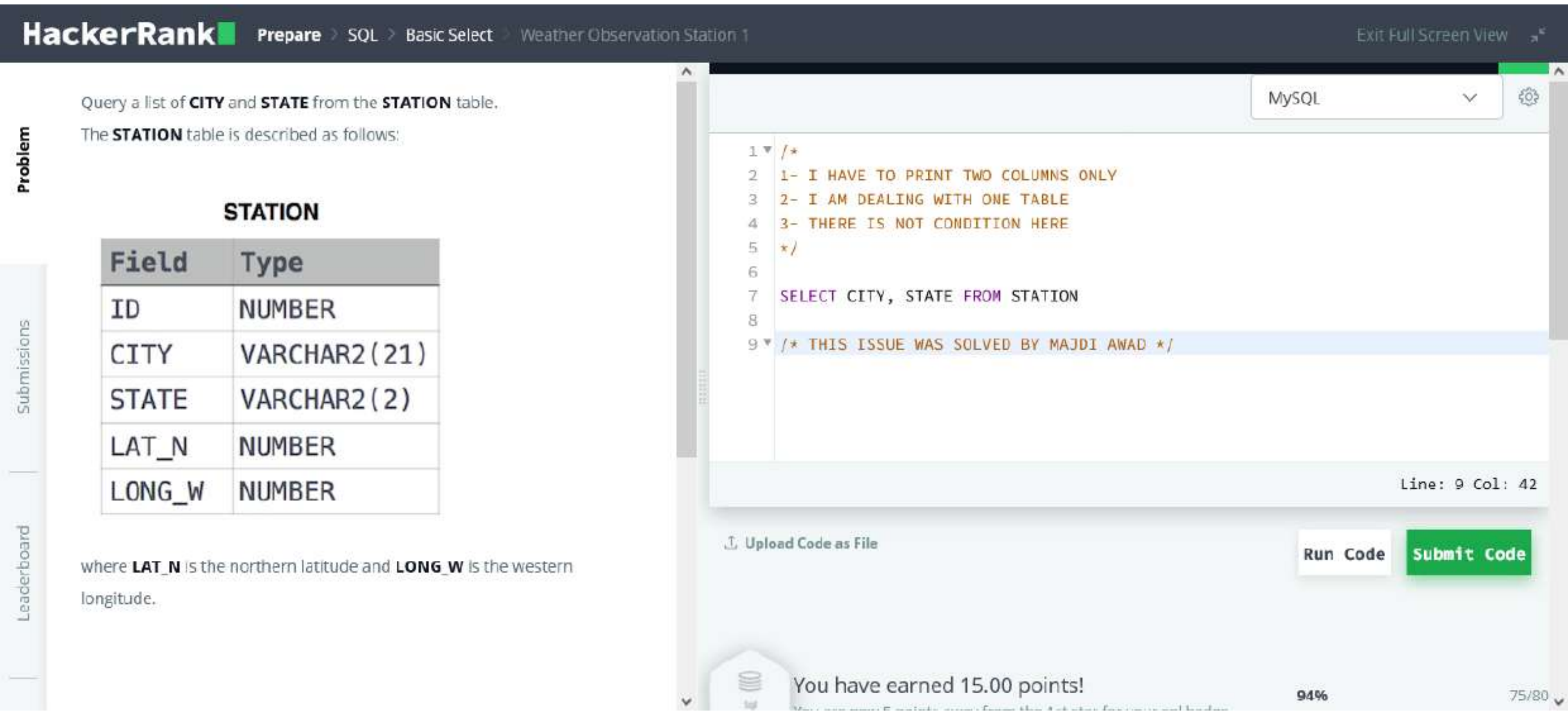
CITY

Field	Type
ID	NUMBER
NAME	VARCHAR2(17)
COUNTRYCODE	VARCHAR2(3)
DISTRICT	VARCHAR2(20)
POPULATION	NUMBER

```
/*
1- I just need to print the name column
2- I have to select the japanese cities using the country code
3- I will deal with one table
*/

SELECT name FROM CITY WHERE COUNTRYCODE = 'JPN'

/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

Congratulations!

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



Query a list of **CITY** and **STATE** from the **STATION** table.

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

STATION

Field	Type
ID	NUMBER
CITY	VARCHAR2(21)
STATE	VARCHAR2(2)
LAT_N	NUMBER
LONG_W	NUMBER

where **LAT_N** is the northern latitude and **LONG_W** is the western longitude.

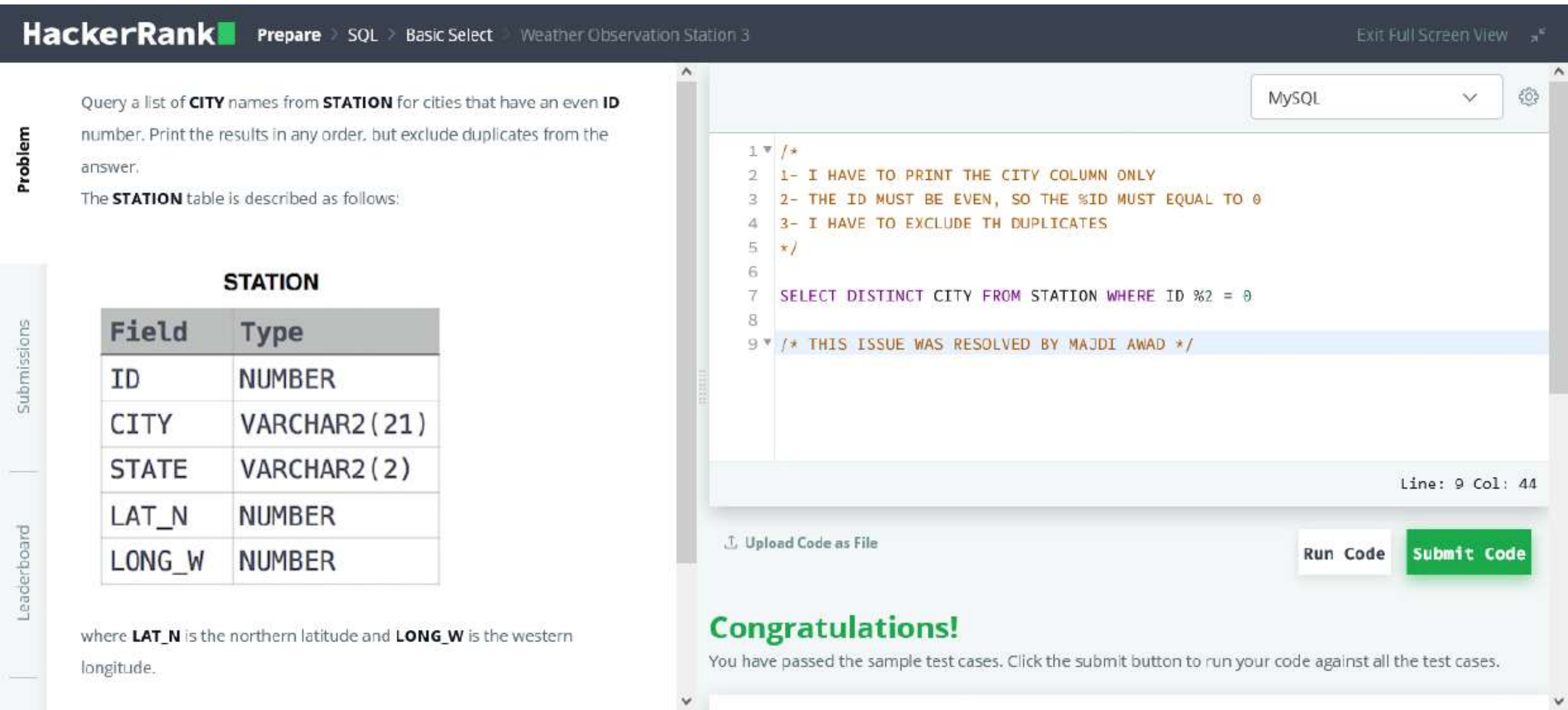
```sql
/*
1- I HAVE TO PRINT TWO COLUMNS ONLY
2- I AM DEALING WITH ONE TABLE
3- THERE IS NOT CONDITION HERE
*/

SELECT CITY, STATE FROM STATION

/* THIS ISSUE WAS SOLVED BY MAJDI AWAD */
```

Line: 9 Col: 42

You have earned 15.00 points!

94%

75/80



HackerRank
Prepare > SQL > Basic Select > Weather Observation Station 3
Exit Full Screen View
Problem
Submissions
Leaderboard

Query a list of **CITY** names from **STATION** for cities that have an even **ID** number. Print the results in any order, but exclude duplicates from the answer.

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

STATION

Field	Type
ID	NUMBER
CITY	VARCHAR2(21)
STATE	VARCHAR2(2)
LAT_N	NUMBER
LONG_W	NUMBER

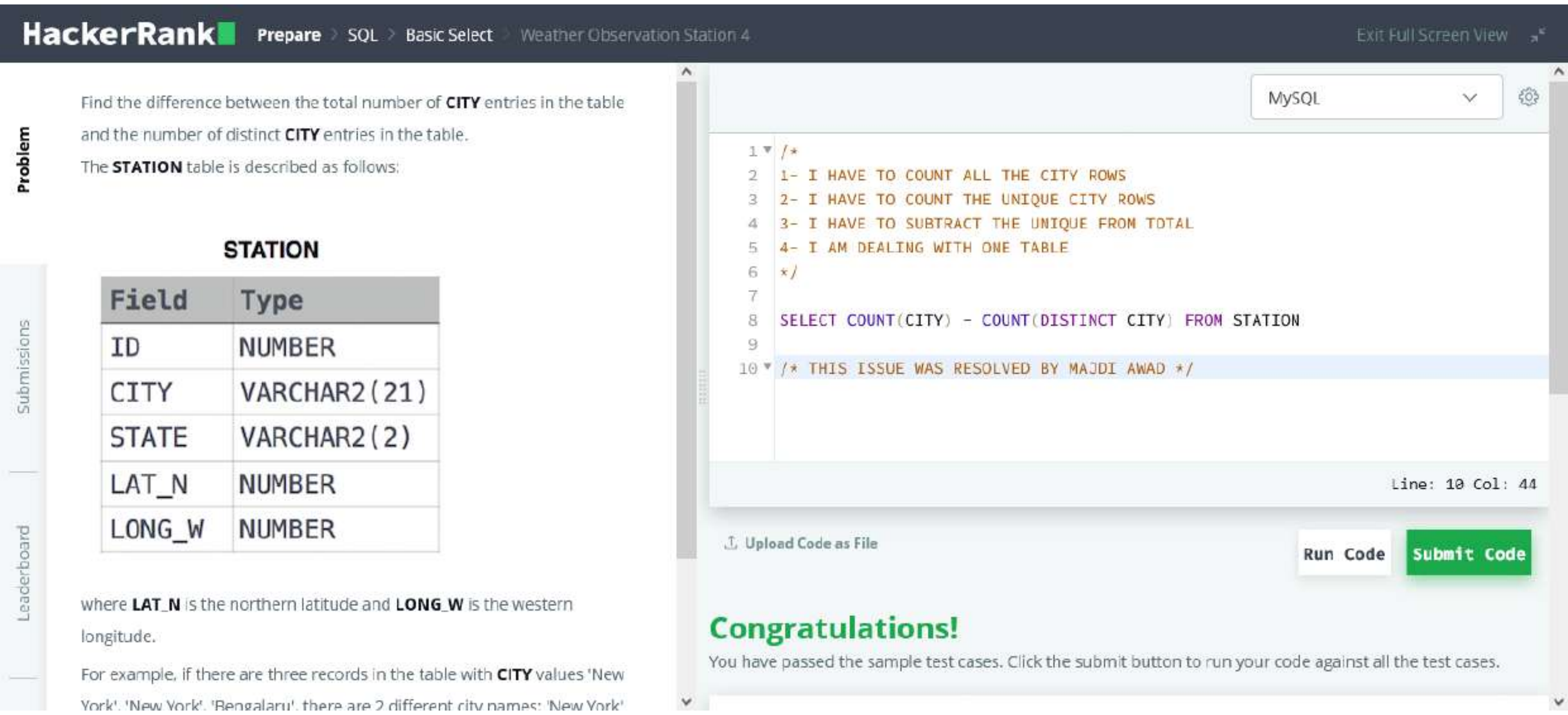
where **LAT_N** is the northern latitude and **LONG_W** is the western longitude.

```sql
/*
1- I HAVE TO PRINT THE CITY COLUMN ONLY
2- THE ID MUST BE EVEN, SO THE %ID MUST EQUAL TO 0
3- I HAVE TO EXCLUDE TH DUPLICATES
*/

SELECT DISTINCT CITY FROM STATION WHERE ID %2 = 0

/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

Line: 9 Col: 44

Congratulations!

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



Find the difference between the total number of **CITY** entries in the table and the number of distinct **CITY** entries in the table.

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

STATION

Field	Type
ID	NUMBER
CITY	VARCHAR2(21)
STATE	VARCHAR2(2)
LAT_N	NUMBER
LONG_W	NUMBER

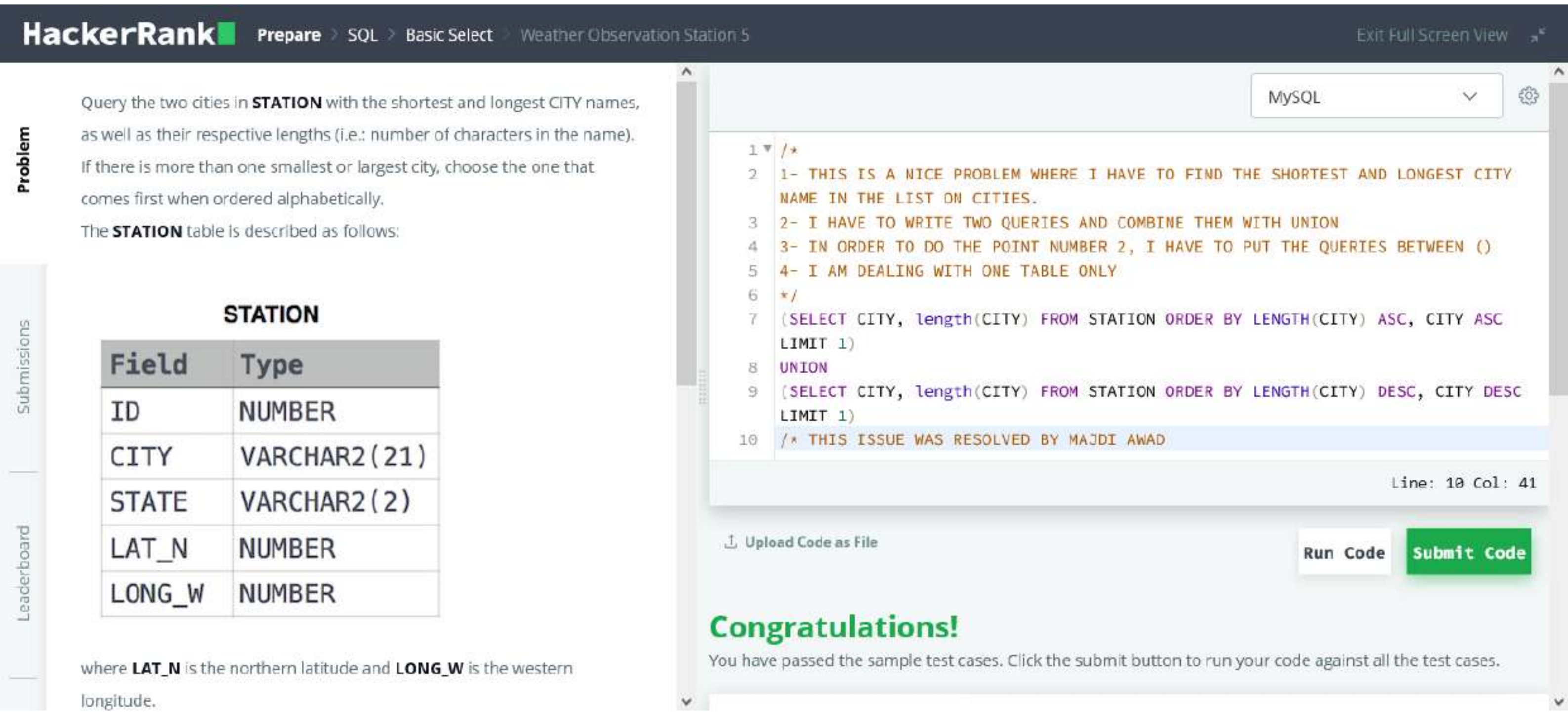
where **LAT_N** is the northern latitude and **LONG_W** is the western longitude.

For example, if there are three records in the table with **CITY** values 'New York', 'New York', 'Bengalaru', there are 2 different city names: 'New York'

```sql
/*
1- I HAVE TO COUNT ALL THE CITY ROWS
2- I HAVE TO COUNT THE UNIQUE CITY ROWS
3- I HAVE TO SUBTRACT THE UNIQUE FROM TOTAL
4- I AM DEALING WITH ONE TABLE
*/

SELECT COUNT(CITY) - COUNT(DISTINCT CITY) FROM STATION

/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

Congratulations!

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



Query the two cities in **STATION** with the shortest and longest CITY names, as well as their respective lengths (i.e.: number of characters in the name). If there is more than one smallest or largest city, choose the one that comes first when ordered alphabetically.

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

STATION

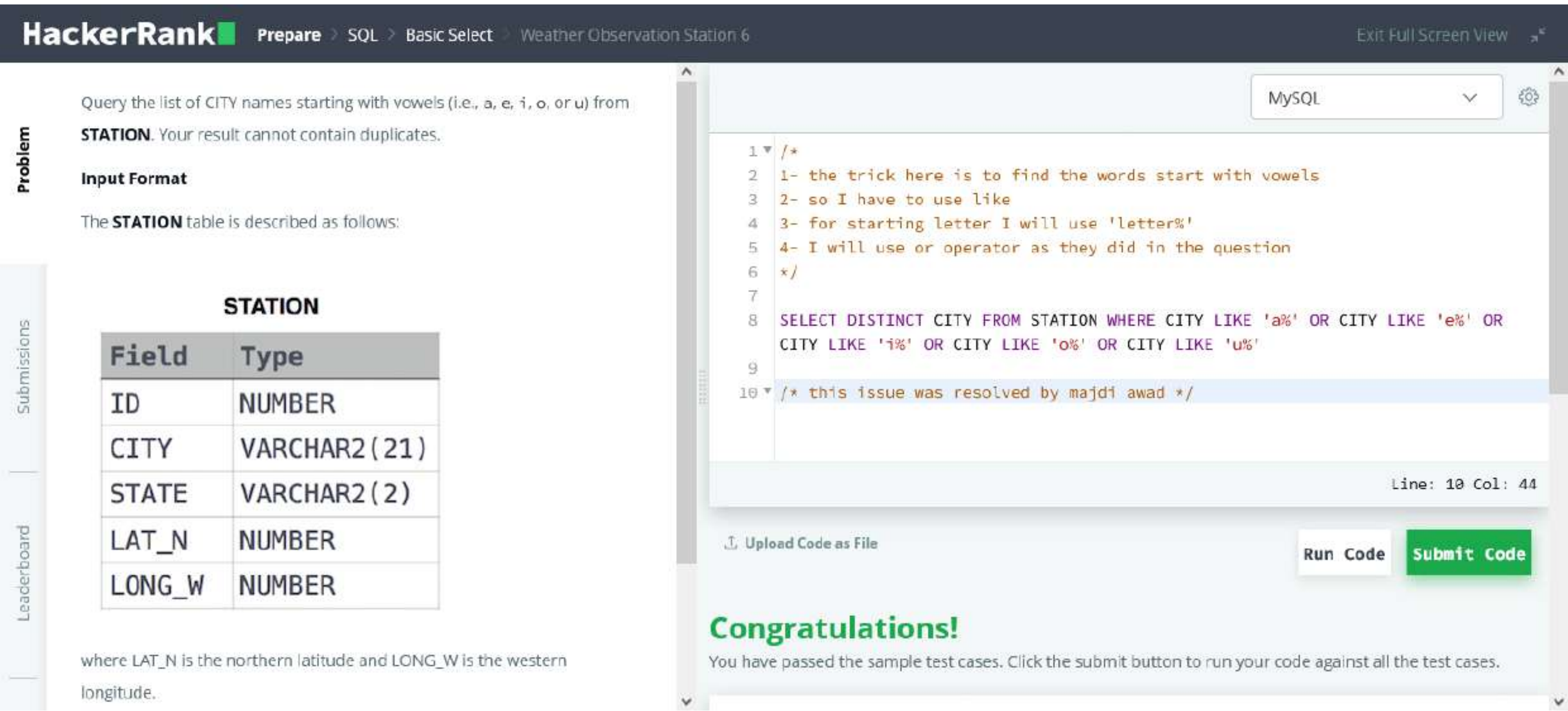
Field	Type
ID	NUMBER
CITY	VARCHAR2(21)
STATE	VARCHAR2(2)
LAT_N	NUMBER
LONG_W	NUMBER

where **LAT_N** is the northern latitude and **LONG_W** is the western longitude.

```
/*
1- THIS IS A NICE PROBLEM WHERE I HAVE TO FIND THE SHORTEST AND LONGEST CITY NAME IN THE LIST ON CITIES.
2- I HAVE TO WRITE TWO QUERIES AND COMBINE THEM WITH UNION
3- IN ORDER TO DO THE POINT NUMBER 2, I HAVE TO PUT THE QUERIES BETWEEN ()
4- I AM DEALING WITH ONE TABLE ONLY
*/
(SELECT CITY, length(CITY) FROM STATION ORDER BY LENGTH(CITY) ASC, CITY ASC LIMIT 1)
UNION
(SELECT CITY, length(CITY) FROM STATION ORDER BY LENGTH(CITY) DESC, CITY DESC LIMIT 1)
/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD
```

Congratulations!

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



Query the list of CITY names starting with vowels (i.e., a, e, i, o, or u) from **STATION**. Your result cannot contain duplicates.

Input Format

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

STATION

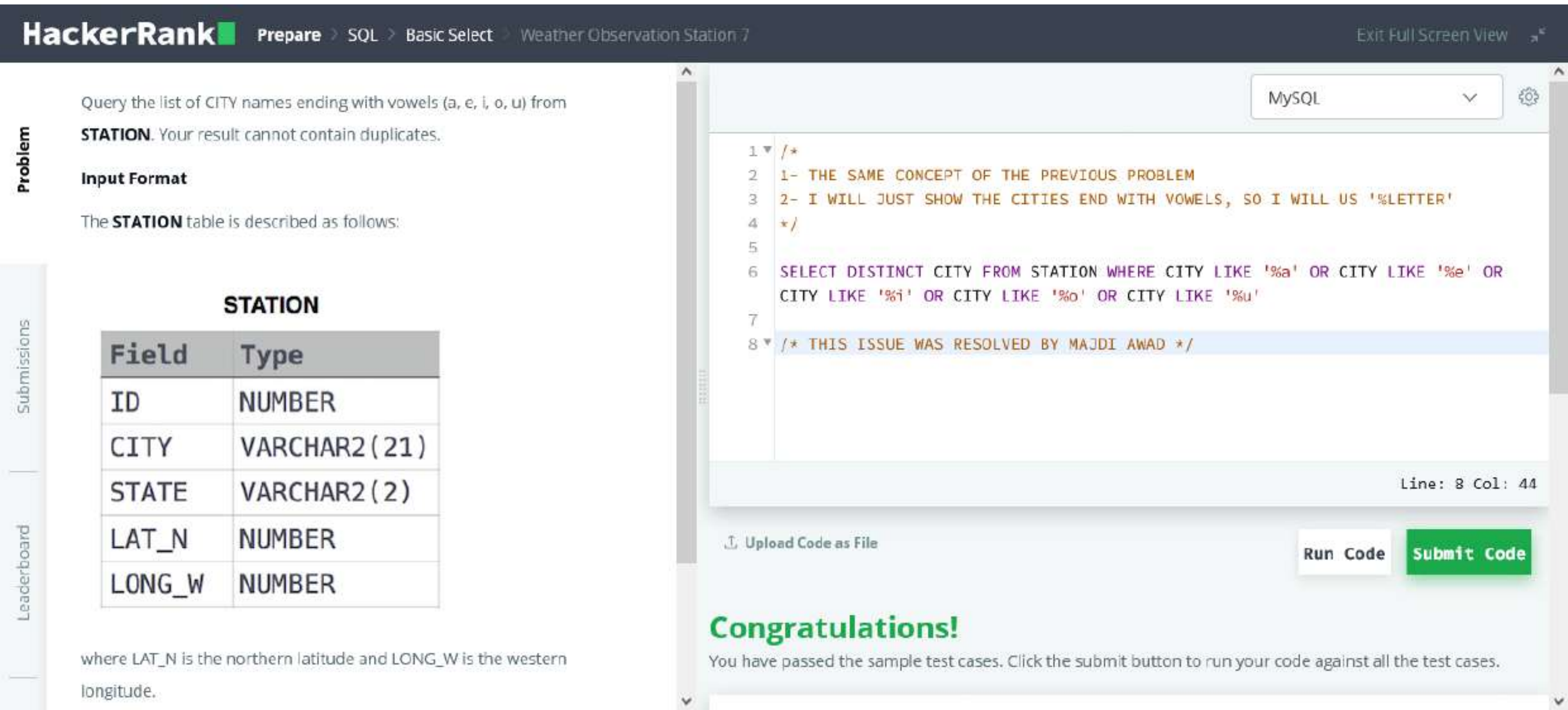
Field	Type
ID	NUMBER
CITY	VARCHAR2(21)
STATE	VARCHAR2(2)
LAT_N	NUMBER
LONG_W	NUMBER

where LAT_N is the northern latitude and LONG_W is the western longitude.

```
/*
1- the trick here is to find the words start with vowels
2- so I have to use like
3- for starting letter I will use 'letter%'
4- I will use or operator as they did in the question
*/

SELECT DISTINCT CITY FROM STATION WHERE CITY LIKE 'a%' OR CITY LIKE 'e%' OR
CITY LIKE 'i%' OR CITY LIKE 'o%' OR CITY LIKE 'u%'

/* this issue was resolved by majdi awad */
```

Congratulations!

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



Query the list of CITY names ending with vowels (a, e, i, o, u) from **STATION**. Your result cannot contain duplicates.

Input Format

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

STATION

Field	Type
ID	NUMBER
CITY	VARCHAR2(21)
STATE	VARCHAR2(2)
LAT_N	NUMBER
LONG_W	NUMBER

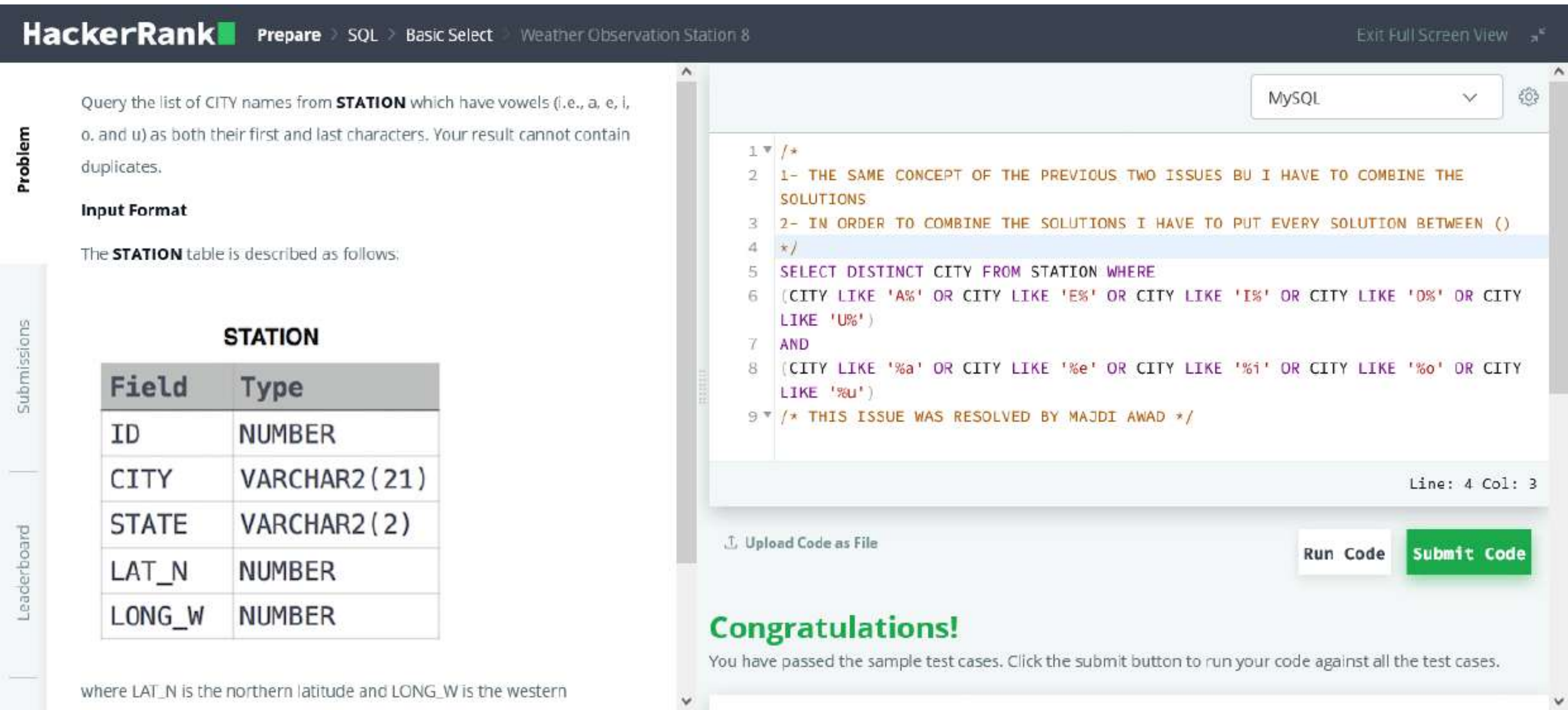
where LAT_N is the northern latitude and LONG_W is the western longitude.

```sql
/*
1- THE SAME CONCEPT OF THE PREVIOUS PROBLEM
2- I WILL JUST SHOW THE CITIES END WITH VOWELS, SO I WILL US '%LETTER'
*/

SELECT DISTINCT CITY FROM STATION WHERE CITY LIKE '%a' OR CITY LIKE '%e' OR
CITY LIKE '%i' OR CITY LIKE '%o' OR CITY LIKE '%u'

/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

Line: 8 Col: 44

Congratulations!

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



Query the list of CITY names from **STATION** which have vowels (i.e., a, e, i, o, and u) as both their first and last characters. Your result cannot contain duplicates.

Input Format

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

STATION

Field	Type
ID	NUMBER
CITY	VARCHAR2(21)
STATE	VARCHAR2(2)
LAT_N	NUMBER
LONG_W	NUMBER

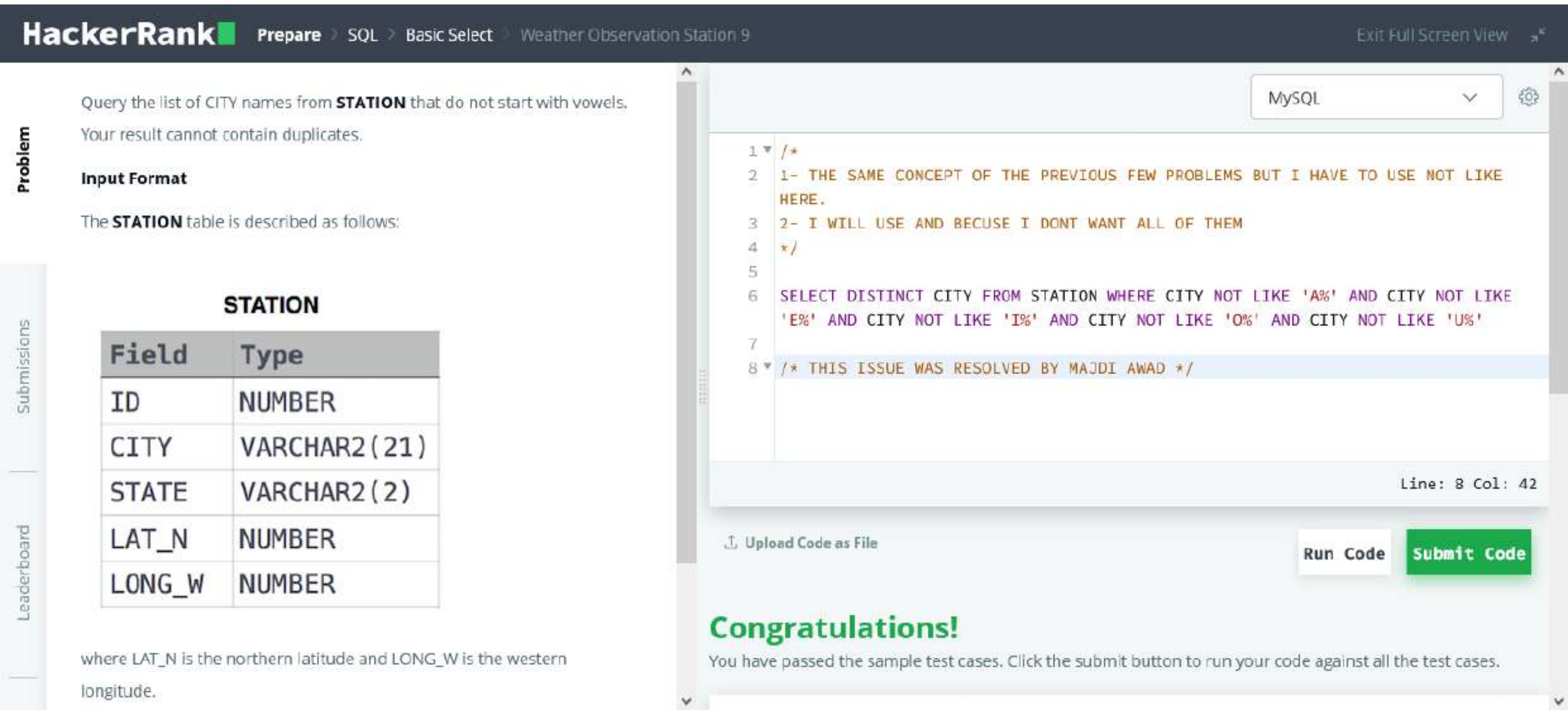
where LAT_N is the northern latitude and LONG_W is the western

```sql
/*
1- THE SAME CONCEPT OF THE PREVIOUS TWO ISSUES BU I HAVE TO COMBINE THE SOLUTIONS
2- IN ORDER TO COMBINE THE SOLUTIONS I HAVE TO PUT EVERY SOLUTION BETWEEN ()
*/
SELECT DISTINCT CITY FROM STATION WHERE
(CITY LIKE 'A%' OR CITY LIKE 'E%' OR CITY LIKE 'I%' OR CITY LIKE 'O%' OR CITY LIKE 'U%')
AND
(CITY LIKE '%a' OR CITY LIKE '%e' OR CITY LIKE '%i' OR CITY LIKE '%o' OR CITY LIKE '%u')
/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

Line: 4 Col: 3

Congratulations!

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



Query the list of CITY names from **STATION** that do not start with vowels. Your result cannot contain duplicates.

Input Format

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

STATION

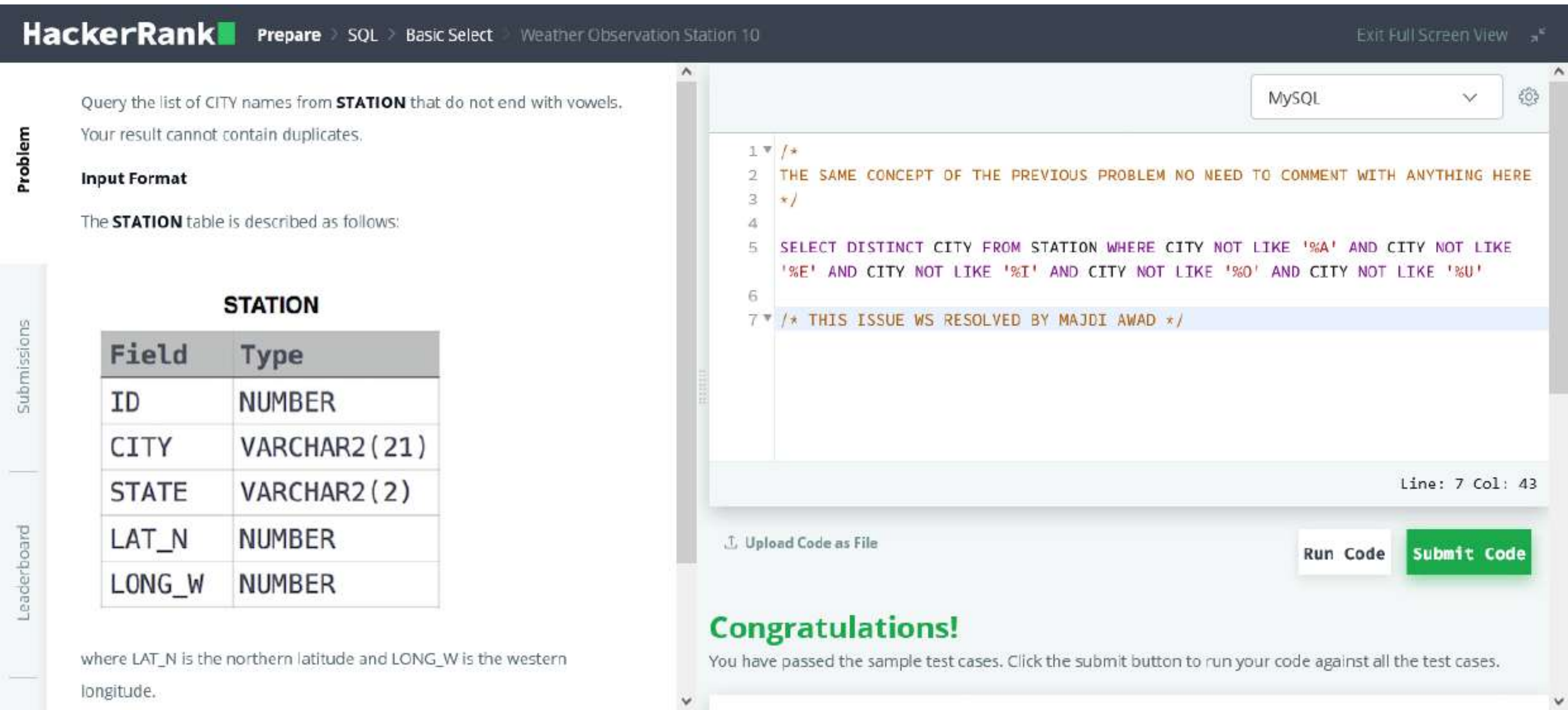
Field	Type
ID	NUMBER
CITY	VARCHAR2(21)
STATE	VARCHAR2(2)
LAT_N	NUMBER
LONG_W	NUMBER

where LAT_N is the northern latitude and LONG_W is the western longitude.

```
/*
1- THE SAME CONCEPT OF THE PREVIOUS FEW PROBLEMS BUT I HAVE TO USE NOT LIKE HERE.
2- I WILL USE AND BECUSE I DONT WANT ALL OF THEM
*/

SELECT DISTINCT CITY FROM STATION WHERE CITY NOT LIKE 'A%' AND CITY NOT LIKE 'E%' AND CITY NOT LIKE 'I%' AND CITY NOT LIKE 'O%' AND CITY NOT LIKE 'U%'

/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

Congratulations!

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



Query the list of CITY names from **STATION** that do not end with vowels. Your result cannot contain duplicates.

Input Format

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

STATION

Field	Type
ID	NUMBER
CITY	VARCHAR2(21)
STATE	VARCHAR2(2)
LAT_N	NUMBER
LONG_W	NUMBER

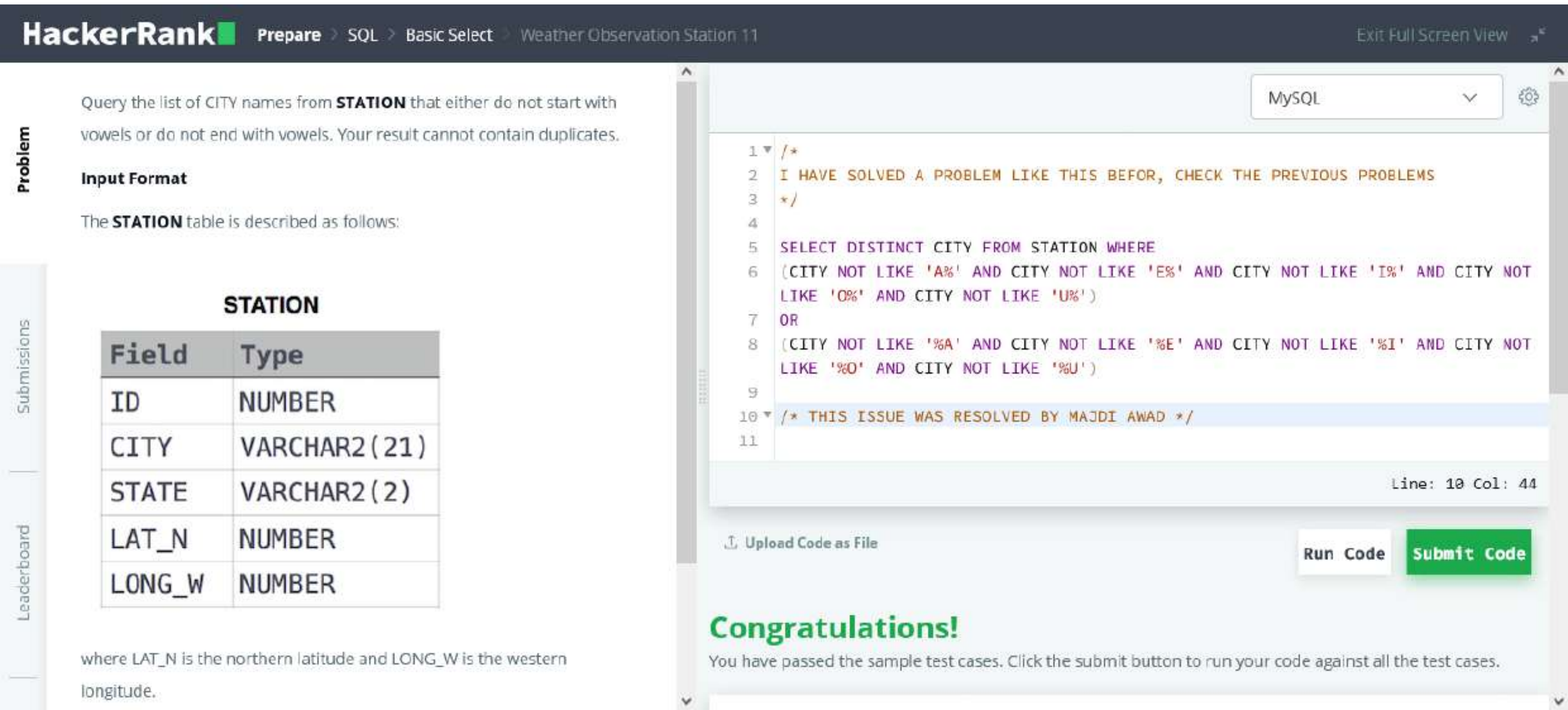
where LAT_N is the northern latitude and LONG_W is the western longitude.

```
/*
THE SAME CONCEPT OF THE PREVIOUS PROBLEM NO NEED TO COMMENT WITH ANYTHING HERE
*/

SELECT DISTINCT CITY FROM STATION WHERE CITY NOT LIKE '%A' AND CITY NOT LIKE
'%E' AND CITY NOT LIKE '%I' AND CITY NOT LIKE '%O' AND CITY NOT LIKE '%U'

/* THIS ISSUE WS RESOLVED BY MAJDI AWAD */
```

Line: 7 Col: 43

Congratulations!

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



Query the list of CITY names from **STATION** that either do not start with vowels or do not end with vowels. Your result cannot contain duplicates.

Input Format

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

STATION

Field	Type
ID	NUMBER
CITY	VARCHAR2(21)
STATE	VARCHAR2(2)
LAT_N	NUMBER
LONG_W	NUMBER

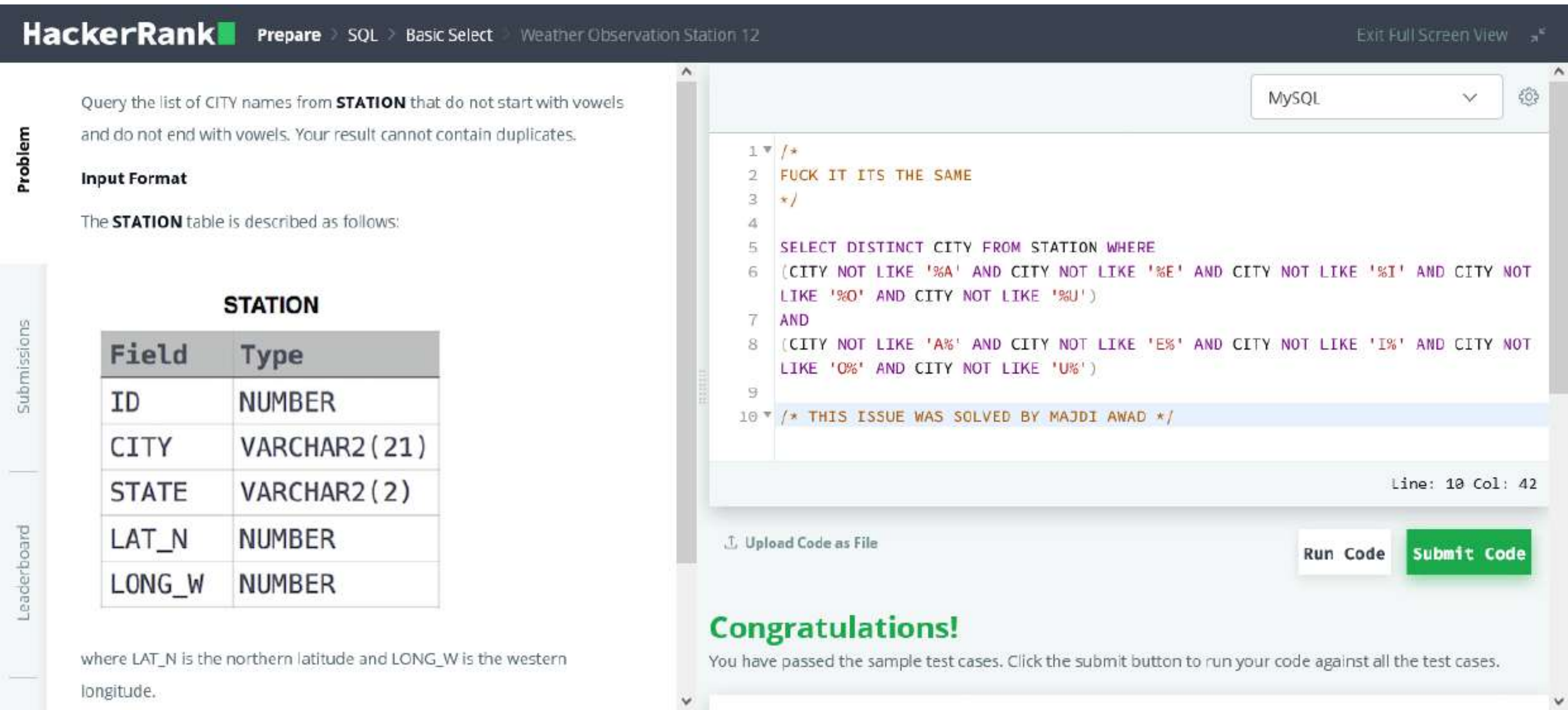
where LAT_N is the northern latitude and LONG_W is the western longitude.

```
/*
I HAVE SOLVED A PROBLEM LIKE THIS BEFOR, CHECK THE PREVIOUS PROBLEMS
*/

SELECT DISTINCT CITY FROM STATION WHERE
(CITY NOT LIKE 'A%' AND CITY NOT LIKE 'E%' AND CITY NOT LIKE 'I%' AND CITY NOT
LIKE 'O%' AND CITY NOT LIKE 'U%')
OR
(CITY NOT LIKE '%A' AND CITY NOT LIKE '%E' AND CITY NOT LIKE '%I' AND CITY NOT
LIKE '%O' AND CITY NOT LIKE '%U')

/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */

```

Line: 10 Col: 44

Congratulations!

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



Query the list of CITY names from **STATION** that do not start with vowels and do not end with vowels. Your result cannot contain duplicates.

Input Format

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

STATION

Field	Type
ID	NUMBER
CITY	VARCHAR2(21)
STATE	VARCHAR2(2)
LAT_N	NUMBER
LONG_W	NUMBER

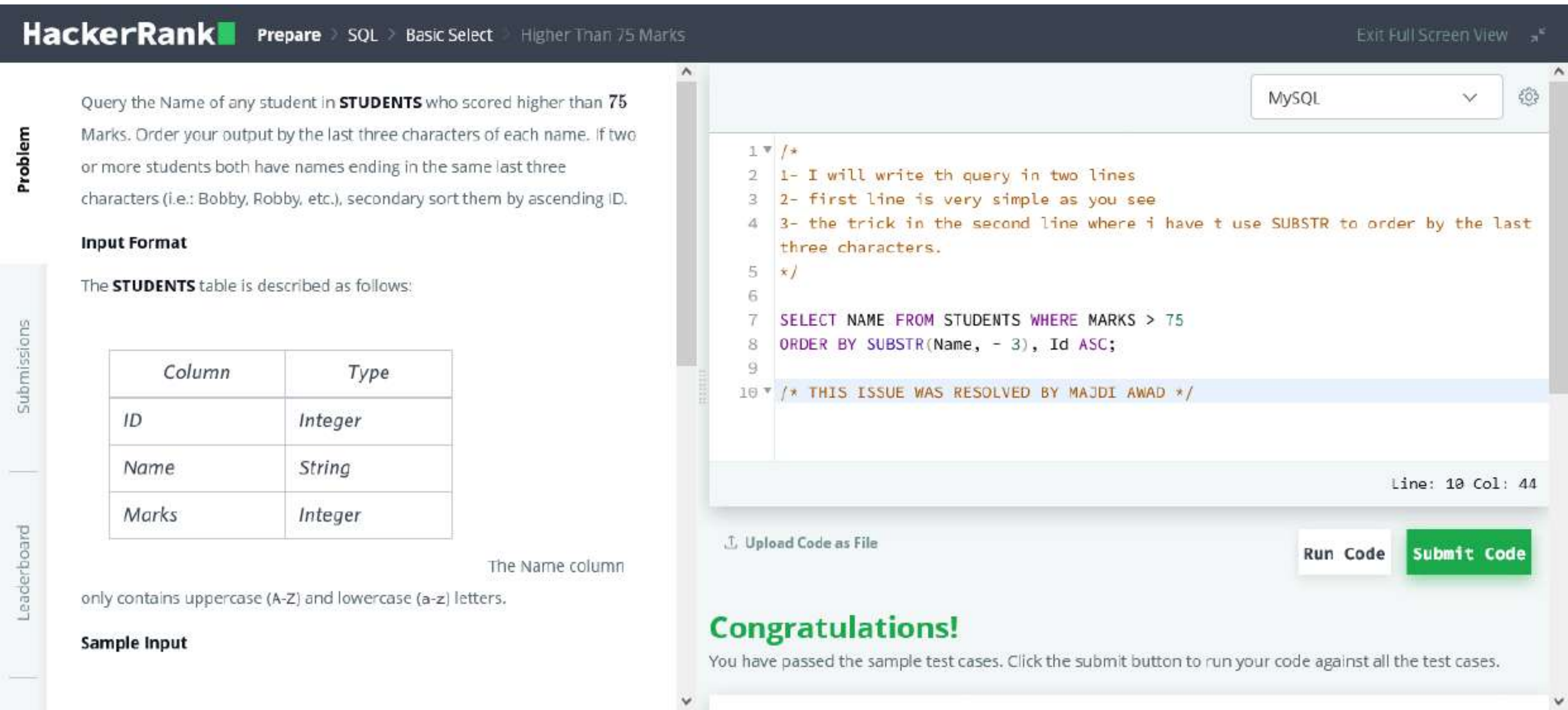
where LAT_N is the northern latitude and LONG_W is the western longitude.

```
/*
FUCK IT ITS THE SAME
*/

SELECT DISTINCT CITY FROM STATION WHERE
(CITY NOT LIKE '%A' AND CITY NOT LIKE '%E' AND CITY NOT LIKE '%I' AND CITY NOT LIKE '%O' AND CITY NOT LIKE '%U')
AND
(CITY NOT LIKE 'A%' AND CITY NOT LIKE 'E%' AND CITY NOT LIKE 'I%' AND CITY NOT LIKE 'O%' AND CITY NOT LIKE 'U%')

/* THIS ISSUE WAS SOLVED BY MAJDI AWAD */
```

Line: 10 Col: 42

Congratulations!

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



Query the Name of any student in **STUDENTS** who scored higher than 75 Marks. Order your output by the last three characters of each name. If two or more students both have names ending in the same last three characters (i.e.: Bobby, Robby, etc.), secondary sort them by ascending ID.

Input Format

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

Column	*Type*
ID	*Integer*
Name	*String*
Marks	*Integer*

The Name column only contains uppercase (A-Z) and lowercase (a-z) letters.

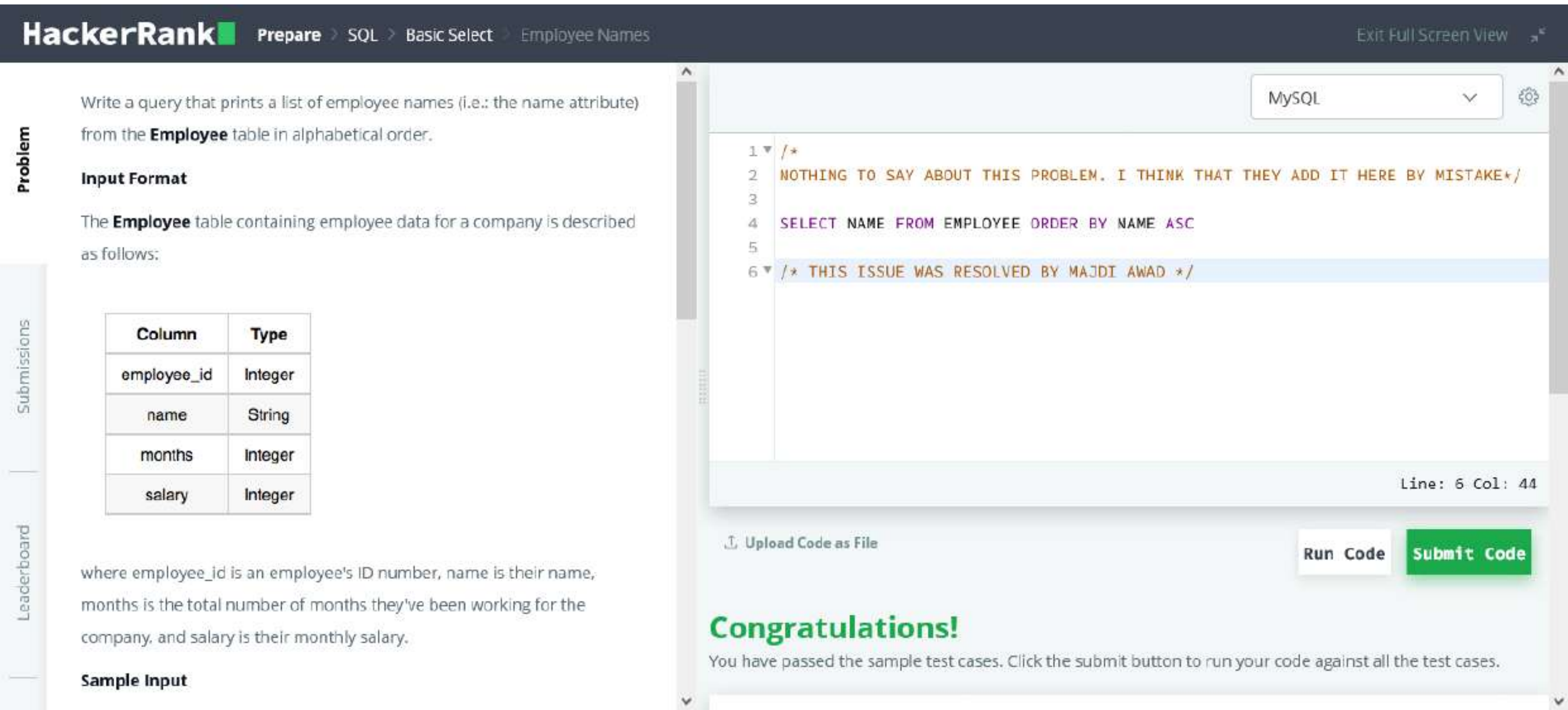
Sample Input

```
/*
1- I will write th query in two lines
2- first line is very simple as you see
3- the trick in the second line where i have t use SUBSTR to order by the last three characters.
*/

SELECT NAME FROM STUDENTS WHERE MARKS > 75
ORDER BY SUBSTR(Name, - 3), Id ASC;

/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

Line: 10 Col: 44

Congratulations!

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



Write a query that prints a list of employee names (i.e.: the name attribute) from the **Employee** table in alphabetical order.

Input Format

The **Employee** table containing employee data for a company is described as follows:

Column	Type
employee_id	Integer
name	String
months	Integer
salary	Integer

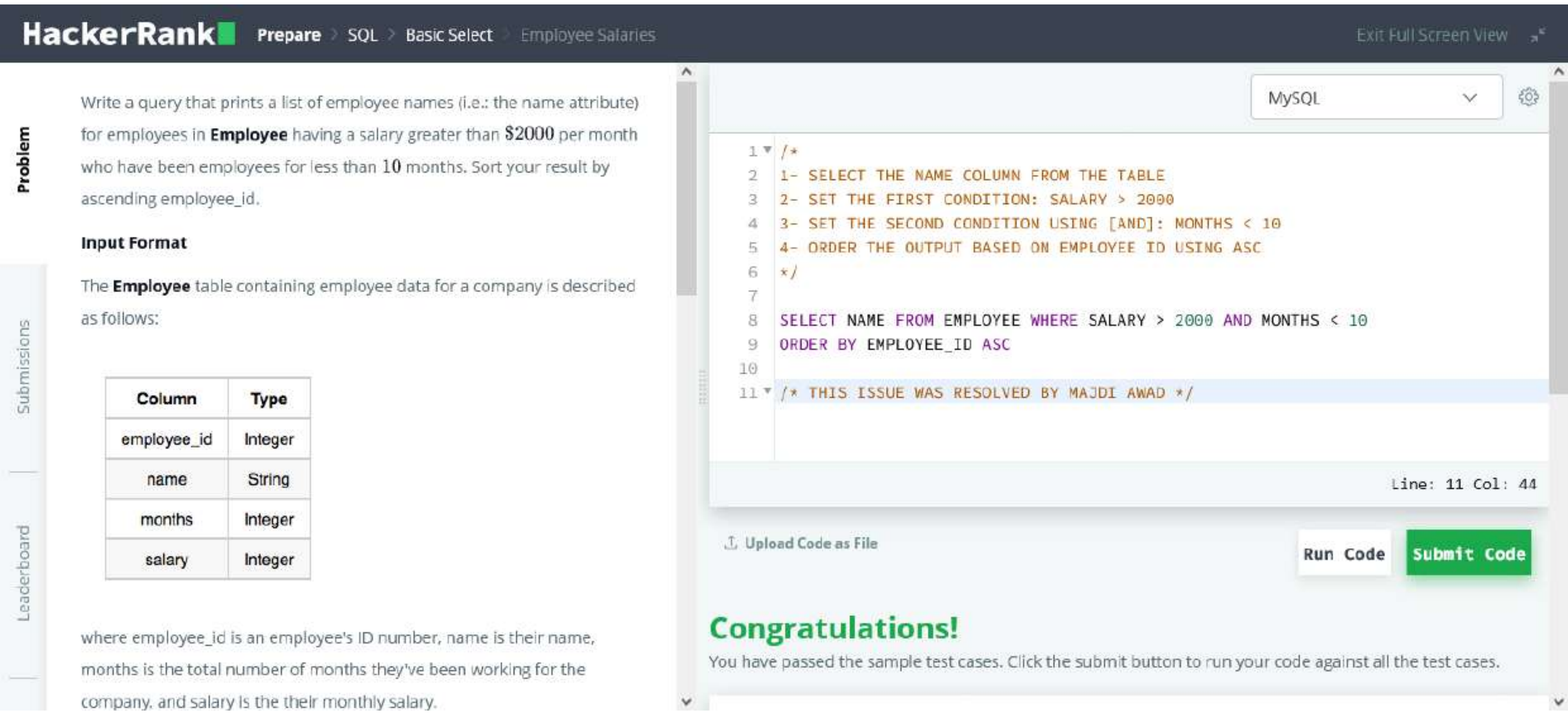
where employee_id is an employee's ID number, name is their name, months is the total number of months they've been working for the company, and salary is their monthly salary.

Sample Input

```
/*
NOTHING TO SAY ABOUT THIS PROBLEM. I THINK THAT THEY ADD IT HERE BY MISTAKE*/

SELECT NAME FROM EMPLOYEE ORDER BY NAME ASC

/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

Congratulations!

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



Write a query that prints a list of employee names (i.e.: the name attribute) for employees in **Employee** having a salary greater than $2000 per month who have been employees for less than 10 months. Sort your result by ascending employee_id.

Input Format

The **Employee** table containing employee data for a company is described as follows:

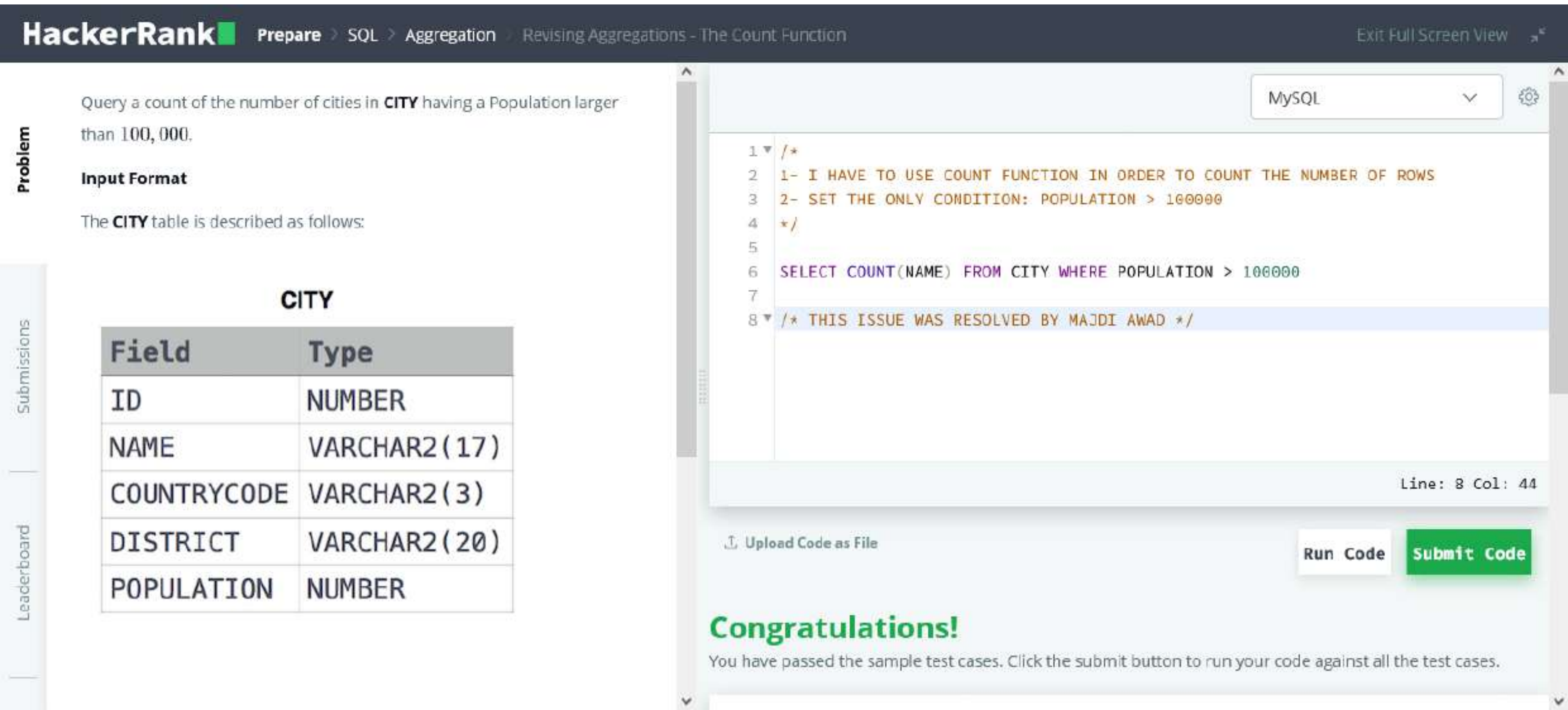
Column	Type
employee_id	Integer
name	String
months	Integer
salary	Integer

where employee_id is an employee's ID number, name is their name, months is the total number of months they've been working for the company, and salary is the their monthly salary.

```
/*
1- SELECT THE NAME COLUMN FROM THE TABLE
2- SET THE FIRST CONDITION: SALARY > 2000
3- SET THE SECOND CONDITION USING [AND]: MONTHS < 10
4- ORDER THE OUTPUT BASED ON EMPLOYEE ID USING ASC
*/

SELECT NAME FROM EMPLOYEE WHERE SALARY > 2000 AND MONTHS < 10
ORDER BY EMPLOYEE_ID ASC

/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

Congratulations!

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



Query a count of the number of cities in **CITY** having a Population larger than 100, 000.

Input Format

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

CITY

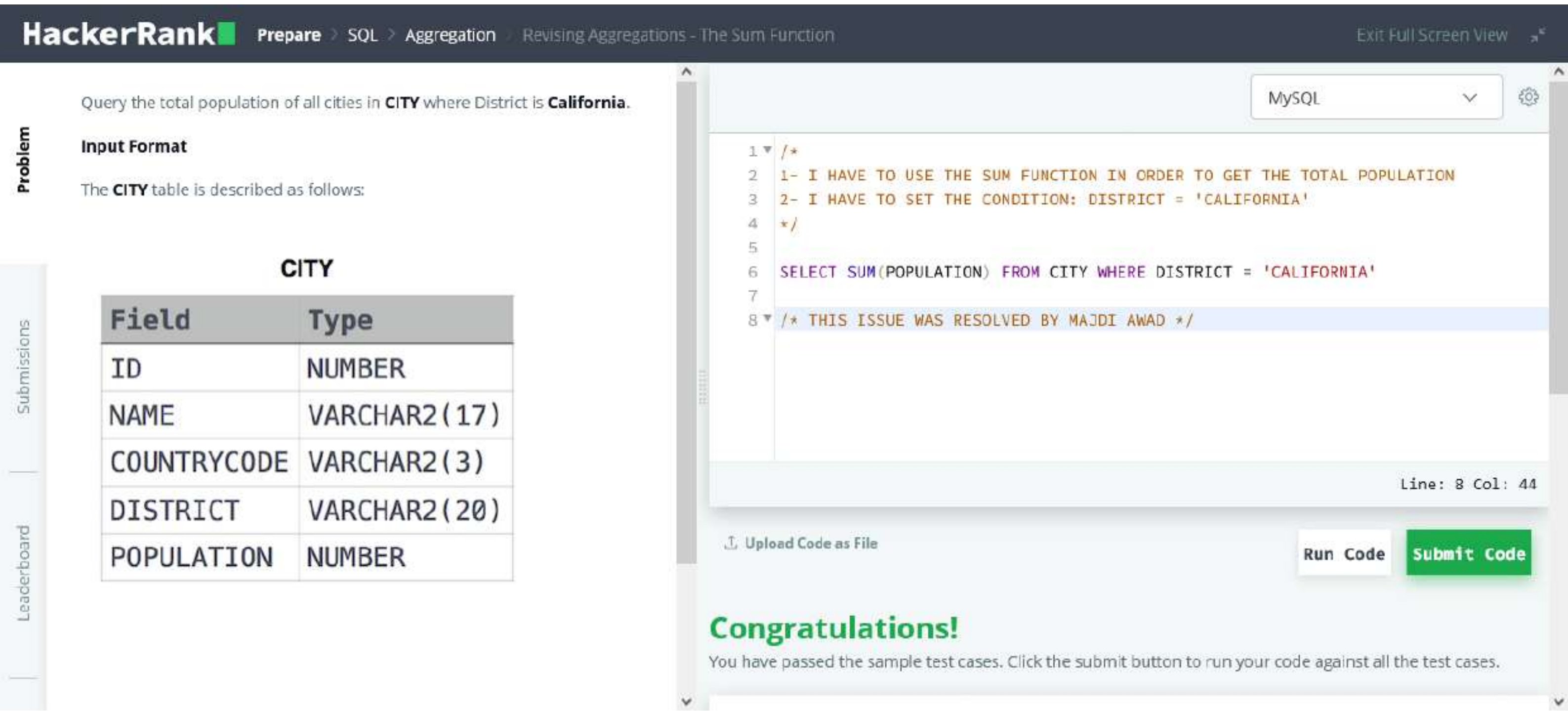
Field	Type
ID	NUMBER
NAME	VARCHAR2(17)
COUNTRYCODE	VARCHAR2(3)
DISTRICT	VARCHAR2(20)
POPULATION	NUMBER

```
/*
1- I HAVE TO USE COUNT FUNCTION IN ORDER TO COUNT THE NUMBER OF ROWS
2- SET THE ONLY CONDITION: POPULATION > 100000
*/

SELECT COUNT(NAME) FROM CITY WHERE POPULATION > 100000

/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

Line: 8 Col: 44

Congratulations!

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



Query the total population of all cities in **CITY** where District is **California**.

Input Format

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

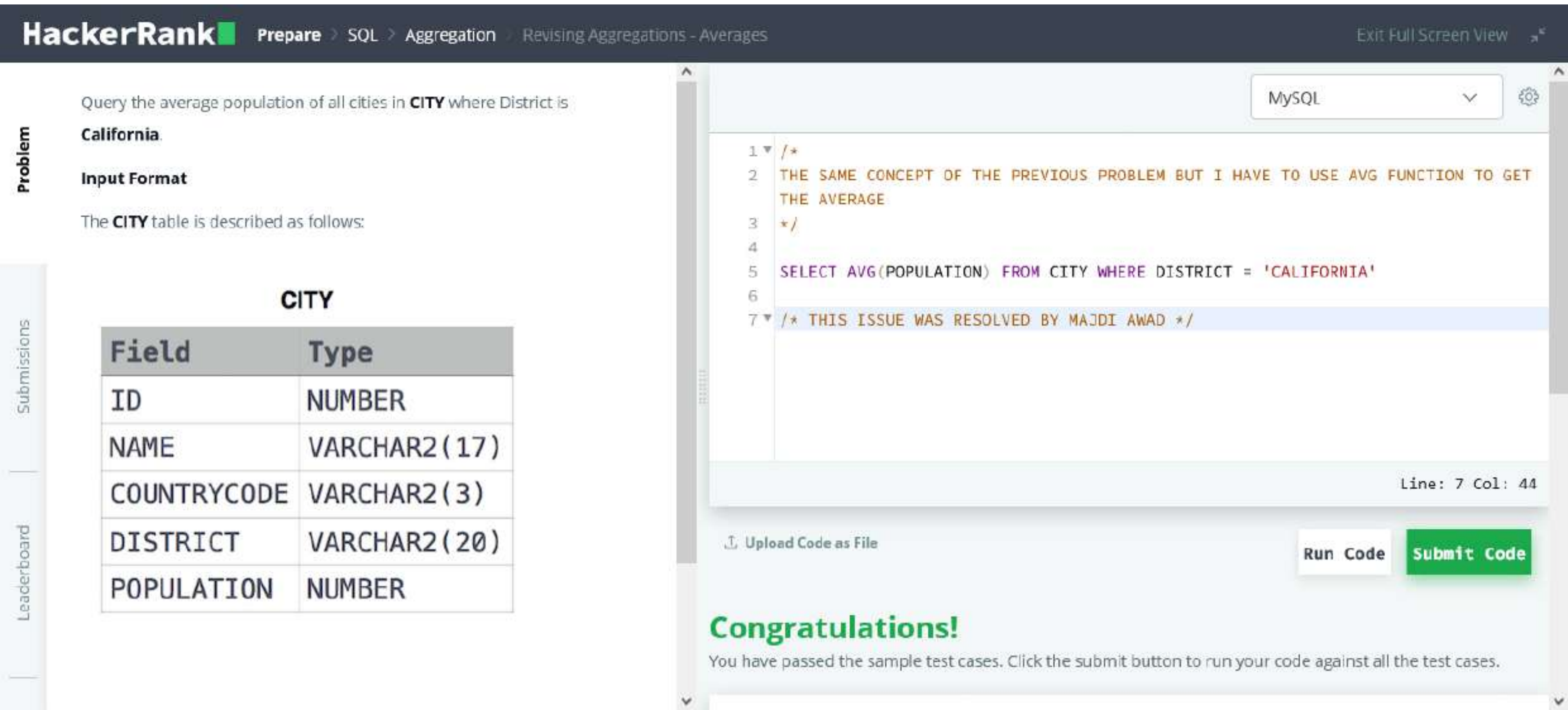
CITY

Field	Type
ID	NUMBER
NAME	VARCHAR2(17)
COUNTRYCODE	VARCHAR2(3)
DISTRICT	VARCHAR2(20)
POPULATION	NUMBER

```
/*
1- I HAVE TO USE THE SUM FUNCTION IN ORDER TO GET THE TOTAL POPULATION
2- I HAVE TO SET THE CONDITION: DISTRICT = 'CALIFORNIA'
*/

SELECT SUM(POPULATION) FROM CITY WHERE DISTRICT = 'CALIFORNIA'

/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

Congratulations!

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



Query the average population of all cities in **CITY** where District is **California**.

Input Format

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

CITY

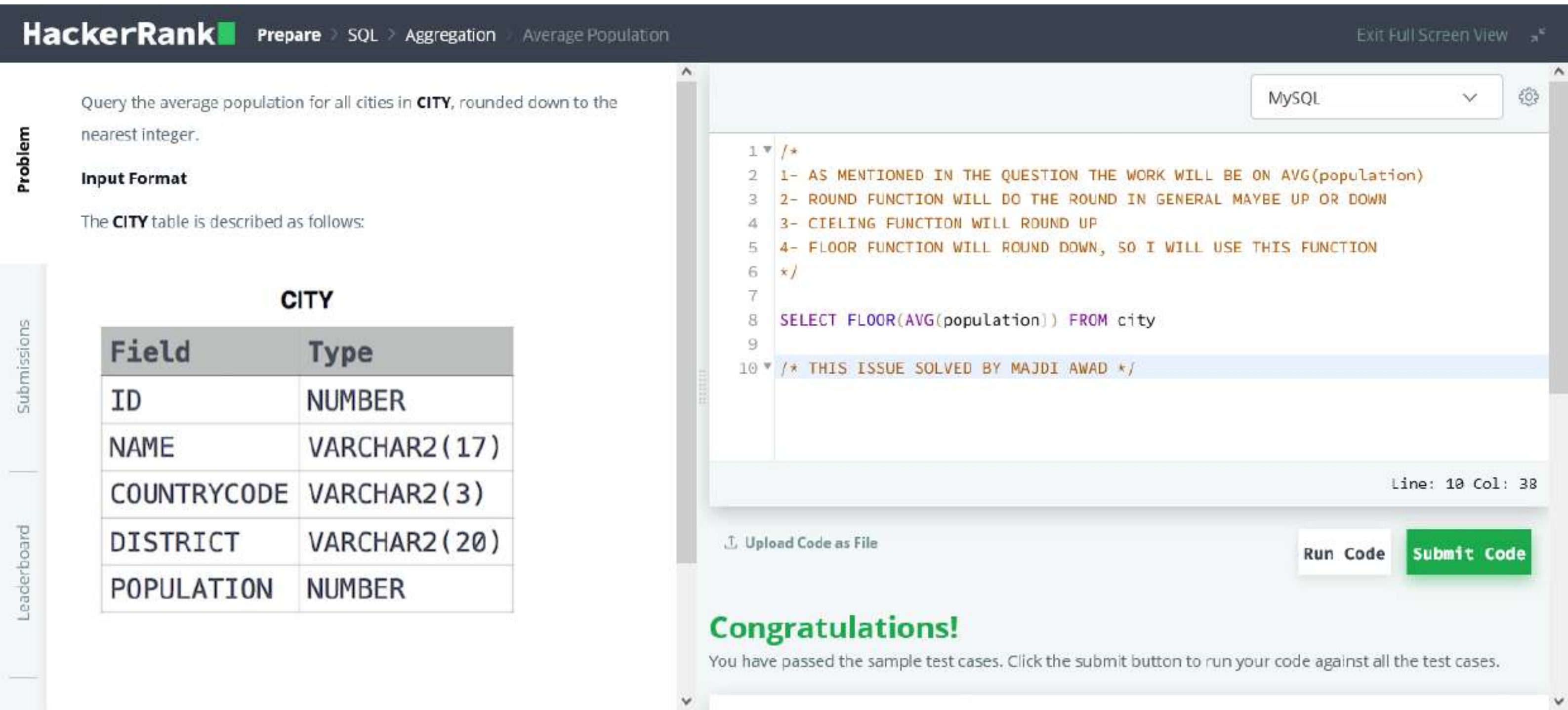
Field	Type
ID	NUMBER
NAME	VARCHAR2(17)
COUNTRYCODE	VARCHAR2(3)
DISTRICT	VARCHAR2(20)
POPULATION	NUMBER

```
/*
THE SAME CONCEPT OF THE PREVIOUS PROBLEM BUT I HAVE TO USE AVG FUNCTION TO GET
THE AVERAGE
*/

SELECT AVG(POPULATION) FROM CITY WHERE DISTRICT = 'CALIFORNIA'

/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

Line: 7 Col: 44

Congratulations!

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



Query the average population for all cities in **CITY**, rounded down to the nearest integer.

Input Format

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

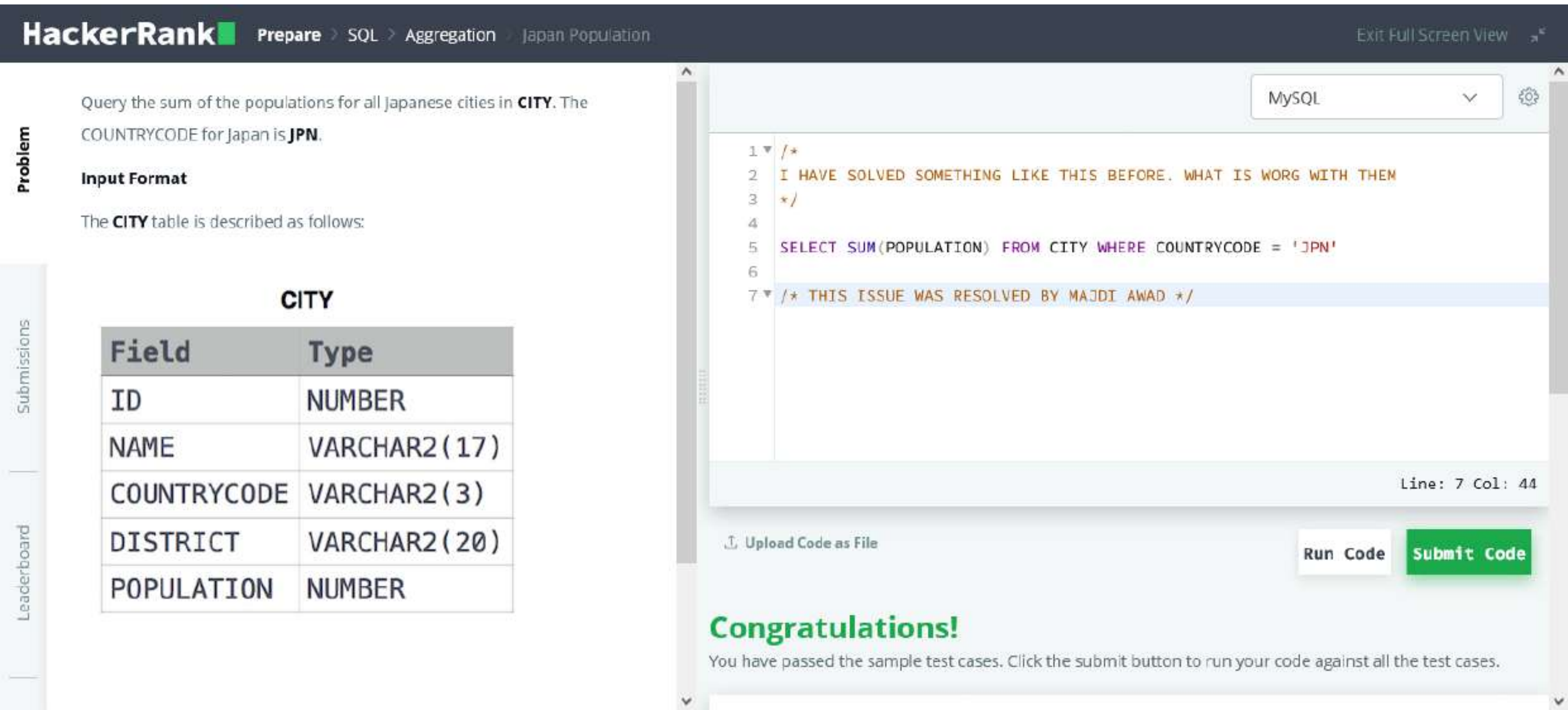
CITY

Field	Type
ID	NUMBER
NAME	VARCHAR2(17)
COUNTRYCODE	VARCHAR2(3)
DISTRICT	VARCHAR2(20)
POPULATION	NUMBER

```
/*
1- AS MENTIONED IN THE QUESTION THE WORK WILL BE ON AVG(population)
2- ROUND FUNCTION WILL DO THE ROUND IN GENERAL MAYBE UP OR DOWN
3- CIELING FUNCTION WILL ROUND UP
4- FLOOR FUNCTION WILL ROUND DOWN, SO I WILL USE THIS FUNCTION
*/

SELECT FLOOR(AVG(population)) FROM city

/* THIS ISSUE SOLVED BY MAJDI AWAD */
```

Congratulations!

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



Query the sum of the populations for all Japanese cities in **CITY**. The COUNTRYCODE for Japan is **JPN**.

Input Format

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

CITY

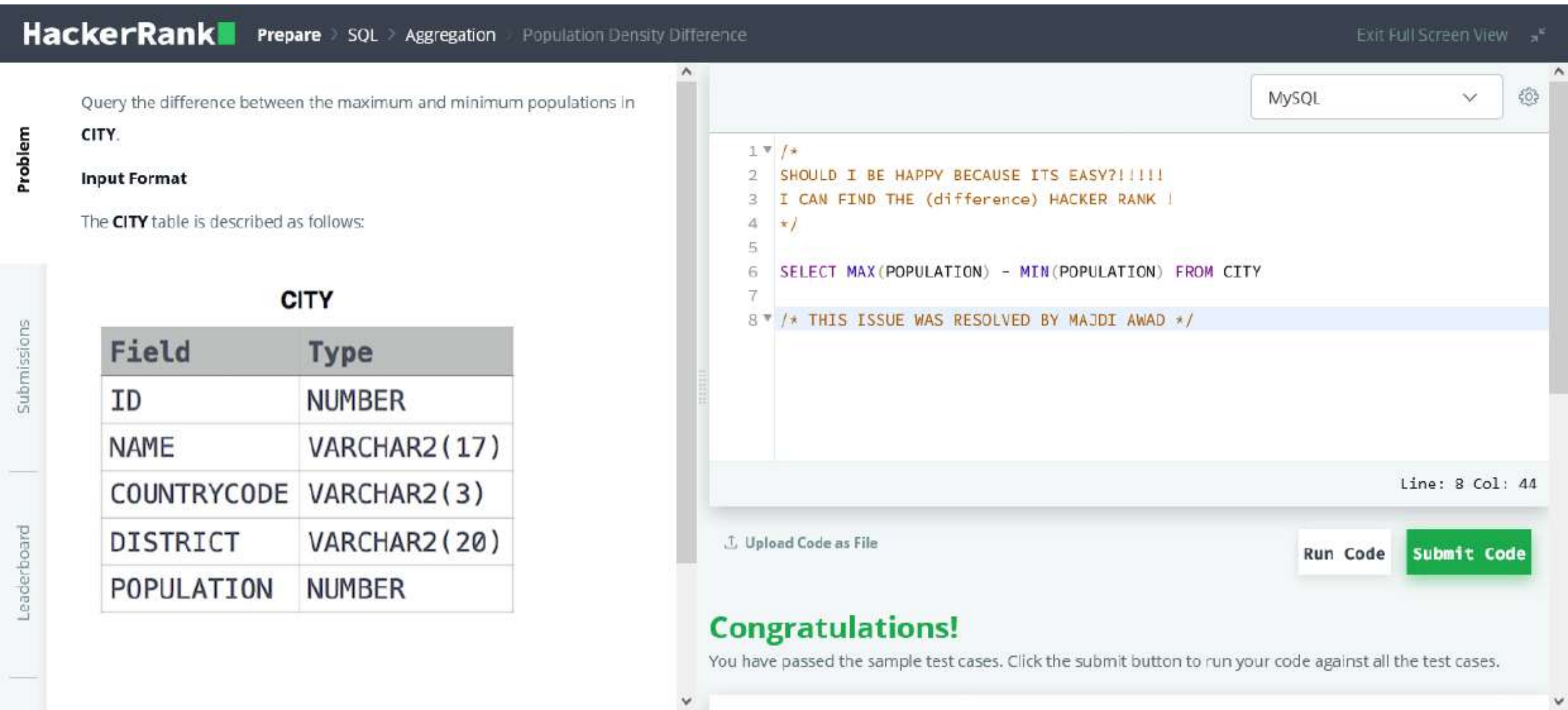
Field	Type
ID	NUMBER
NAME	VARCHAR2(17)
COUNTRYCODE	VARCHAR2(3)
DISTRICT	VARCHAR2(20)
POPULATION	NUMBER

```sql
/*
I HAVE SOLVED SOMETHING LIKE THIS BEFORE. WHAT IS WORG WITH THEM
*/

SELECT SUM(POPULATION) FROM CITY WHERE COUNTRYCODE = 'JPN'

/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

Line: 7 Col: 44

Congratulations!

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



Query the difference between the maximum and minimum populations in **CITY**.

Input Format

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

CITY

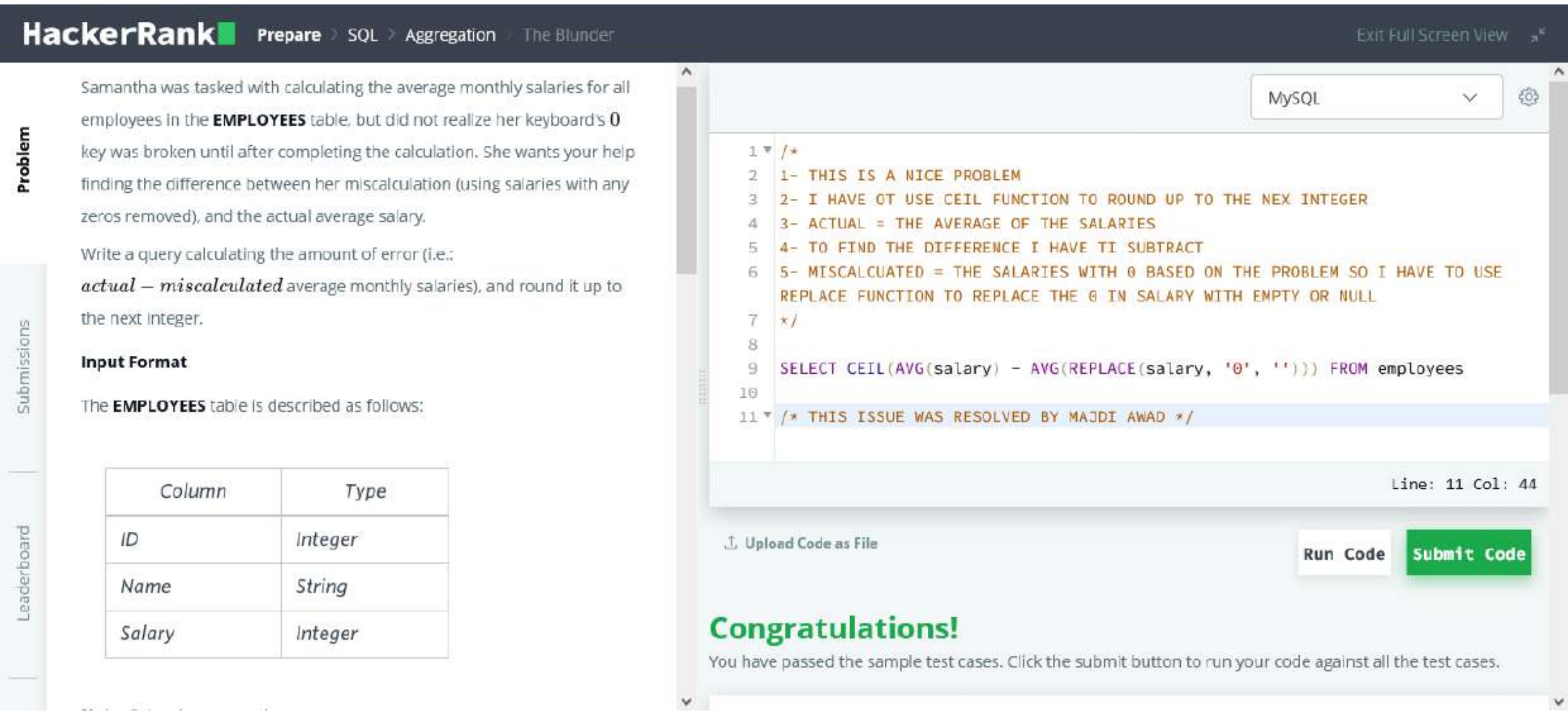
Field	Type
ID	NUMBER
NAME	VARCHAR2(17)
COUNTRYCODE	VARCHAR2(3)
DISTRICT	VARCHAR2(20)
POPULATION	NUMBER

```sql
/*
SHOULD I BE HAPPY BECAUSE ITS EASY?!!!!!
I CAN FIND THE (difference) HACKER RANK !
*/

SELECT MAX(POPULATION) - MIN(POPULATION) FROM CITY

/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

Line: 8 Col: 44

Congratulations!

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



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

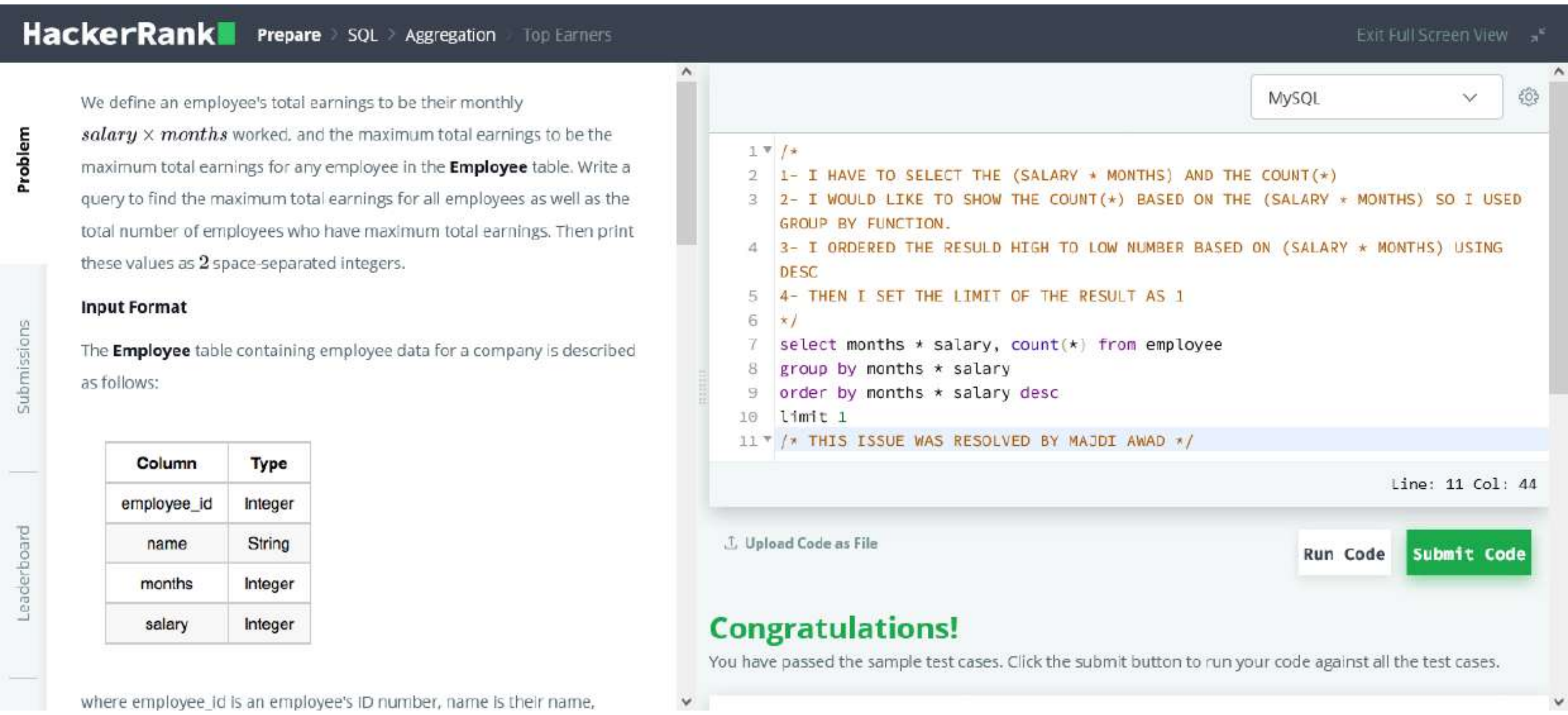
The **EMPLOYEES** table is described as follows:

Column	*Type*
ID	*Integer*
Name	*String*
Salary	*Integer*

```
/*
1- THIS IS A NICE PROBLEM
2- I HAVE OT USE CEIL FUNCTION TO ROUND UP TO THE NEX INTEGER
3- ACTUAL = THE AVERAGE OF THE SALARIES
4- TO FIND THE DIFFERENCE I HAVE TI SUBTRACT
5- MISCALCUATED = THE SALARIES WITH 0 BASED ON THE PROBLEM SO I HAVE TO USE REPLACE FUNCTION TO REPLACE THE 0 IN SALARY WITH EMPTY OR NULL
*/

SELECT CEIL(AVG(salary) - AVG(REPLACE(salary, '0', ''))) FROM employees

/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

Congratulations!

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



We define an employee's total earnings to be their monthly $salary \times months$ worked, and the maximum total earnings to be the maximum total earnings for any employee in the **Employee** table. Write a query to find the maximum total earnings for all employees as well as the total number of employees who have maximum total earnings. Then print these values as 2 space-separated integers.

Input Format

The **Employee** table containing employee data for a company is described as follows:

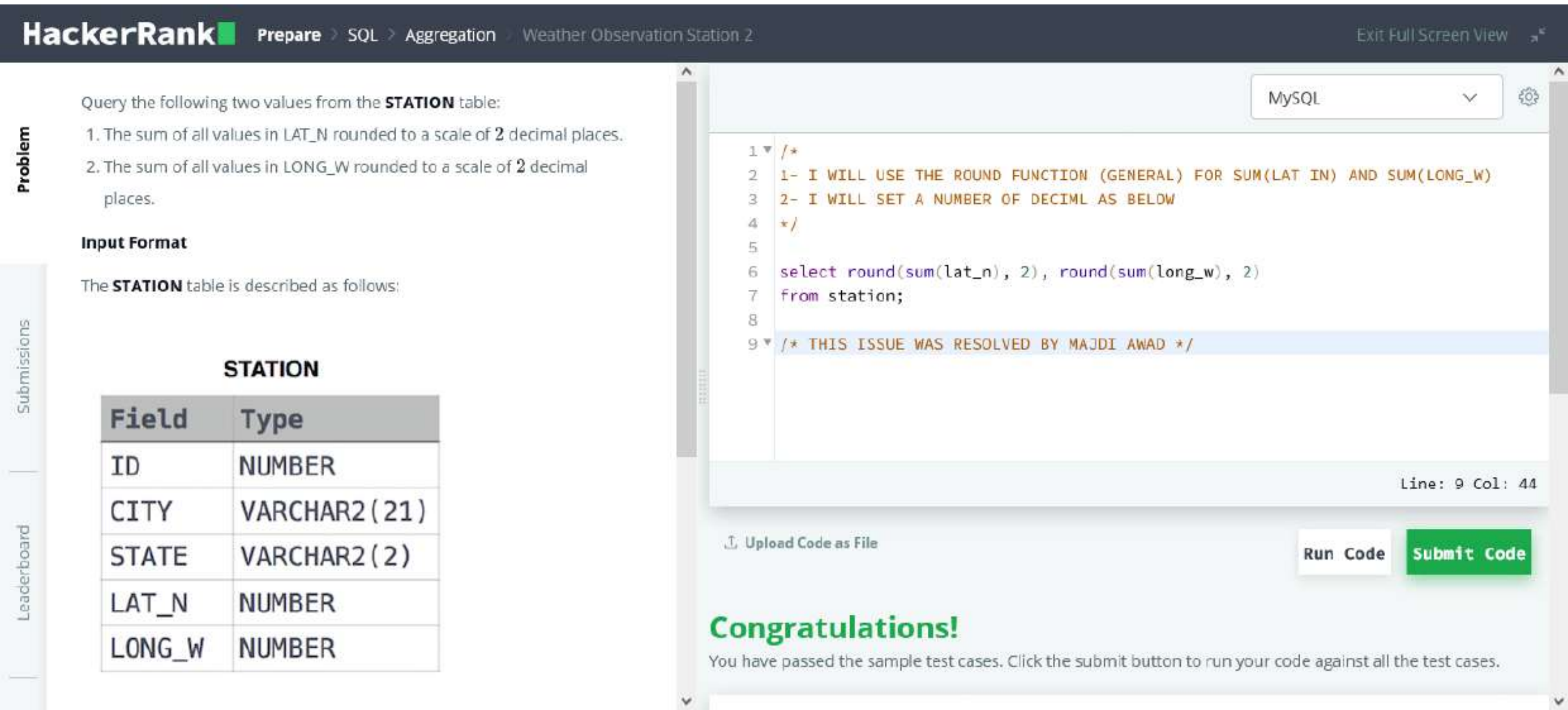
Column	Type
employee_id	Integer
name	String
months	Integer
salary	Integer

where employee_id is an employee's ID number, name is their name,

```sql
/*
1- I HAVE TO SELECT THE (SALARY * MONTHS) AND THE COUNT(*)
2- I WOULD LIKE TO SHOW THE COUNT(*) BASED ON THE (SALARY * MONTHS) SO I USED GROUP BY FUNCTION.
3- I ORDERED THE RESULD HIGH TO LOW NUMBER BASED ON (SALARY * MONTHS) USING DESC
4- THEN I SET THE LIMIT OF THE RESULT AS 1
*/
select months * salary, count(*) from employee
group by months * salary
order by months * salary desc
limit 1
/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

Congratulations!

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



Query the following two values from the **STATION** table:

1. The sum of all values in LAT_N rounded to a scale of 2 decimal places.
2. The sum of all values in LONG_W rounded to a scale of 2 decimal places.

Input Format

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

STATION

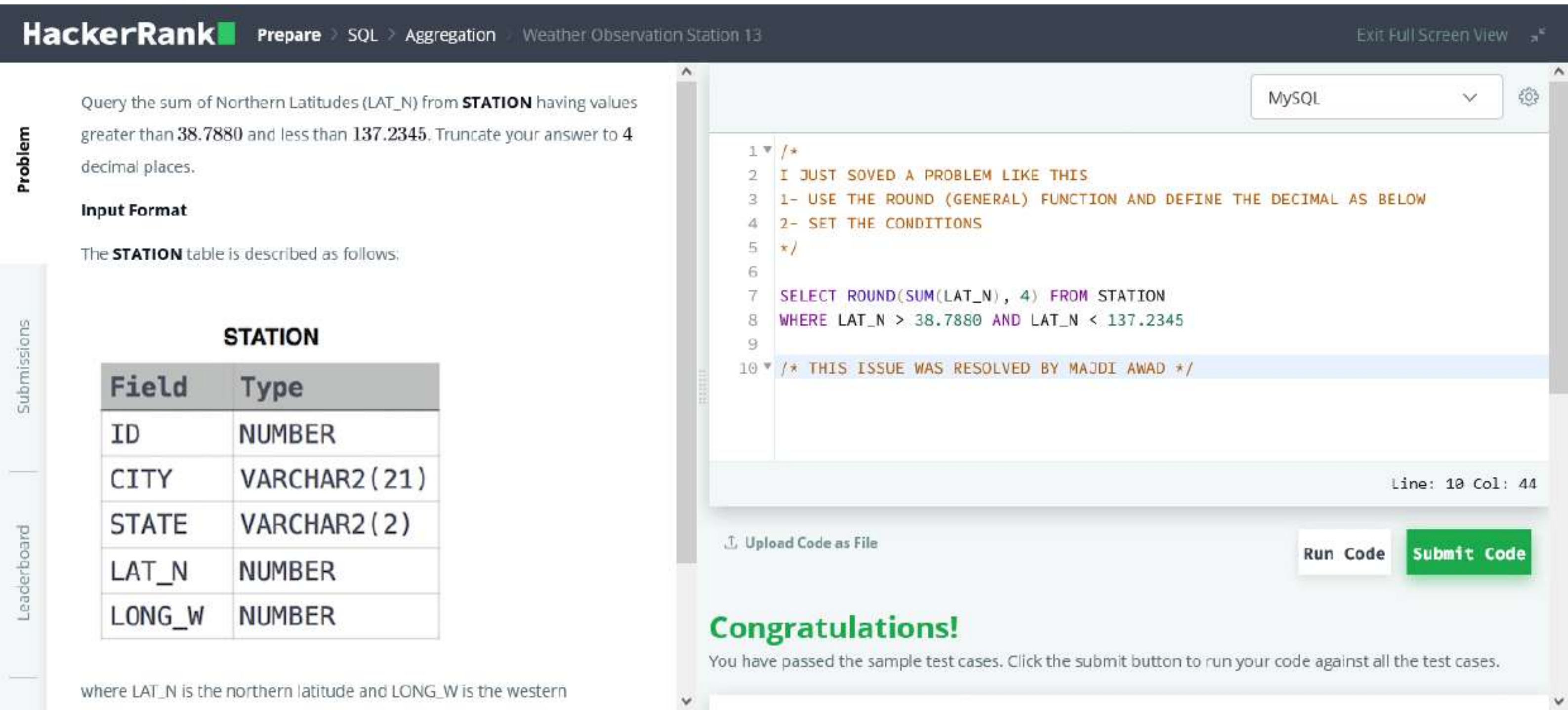
Field	Type
ID	NUMBER
CITY	VARCHAR2(21)
STATE	VARCHAR2(2)
LAT_N	NUMBER
LONG_W	NUMBER

```
/*
1- I WILL USE THE ROUND FUNCTION (GENERAL) FOR SUM(LAT IN) AND SUM(LONG_W)
2- I WILL SET A NUMBER OF DECIML AS BELOW
*/

select round(sum(lat_n), 2), round(sum(long_w), 2)
from station;

/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

Line: 9 Col: 44

Congratulations!

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



Query the sum of Northern Latitudes (LAT_N) from **STATION** having values greater than 38.7880 and less than 137.2345. Truncate your answer to 4 decimal places.

Input Format

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

STATION

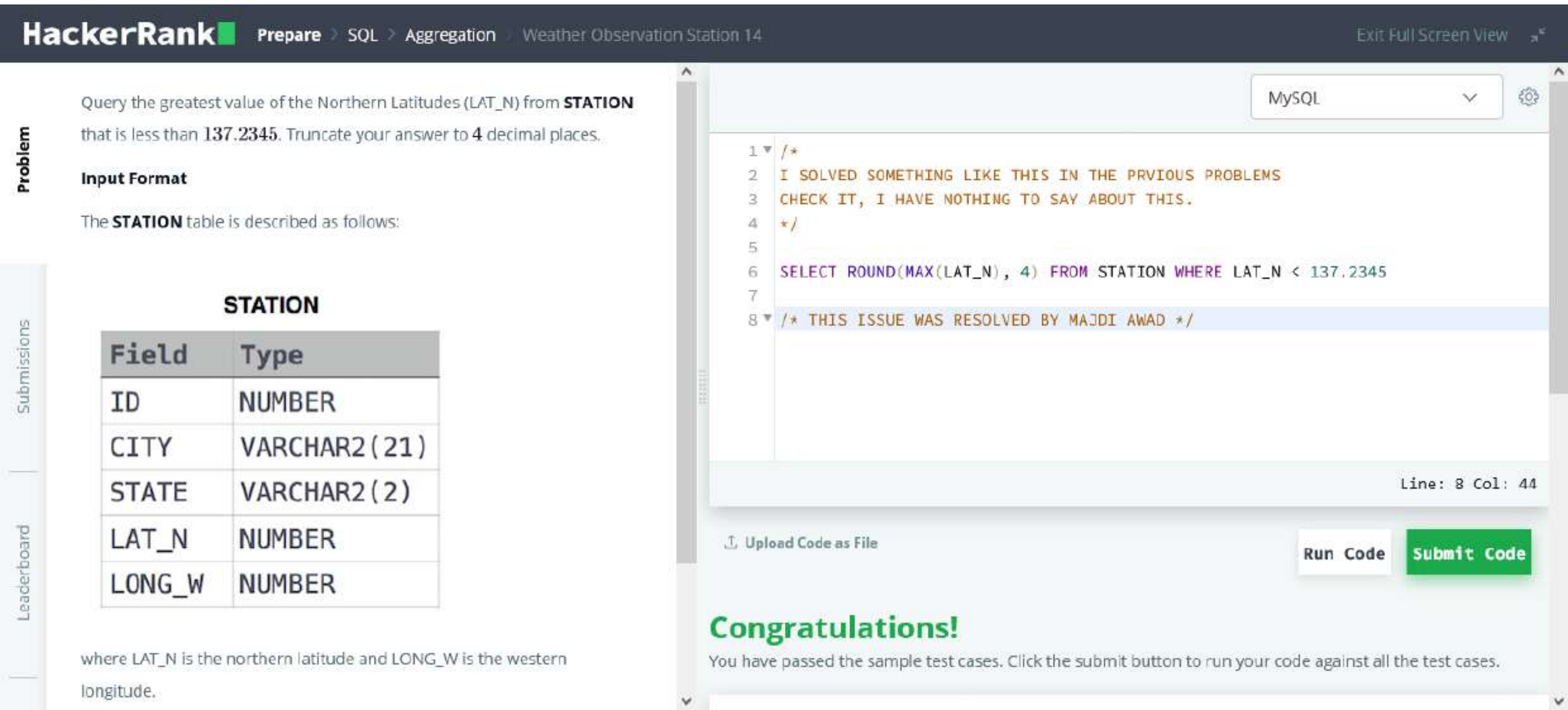
Field	Type
ID	NUMBER
CITY	VARCHAR2(21)
STATE	VARCHAR2(2)
LAT_N	NUMBER
LONG_W	NUMBER

where LAT_N is the northern latitude and LONG_W is the western

```
/*
I JUST SOVED A PROBLEM LIKE THIS
1- USE THE ROUND (GENERAL) FUNCTION AND DEFINE THE DECIMAL AS BELOW
2- SET THE CONDITIONS
*/

SELECT ROUND(SUM(LAT_N), 4) FROM STATION
WHERE LAT_N > 38.7880 AND LAT_N < 137.2345

/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

Congratulations!

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



Query the greatest value of the Northern Latitudes (LAT_N) from **STATION** that is less than 137.2345. Truncate your answer to 4 decimal places.

Input Format

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

STATION

Field	Type
ID	NUMBER
CITY	VARCHAR2(21)
STATE	VARCHAR2(2)
LAT_N	NUMBER
LONG_W	NUMBER

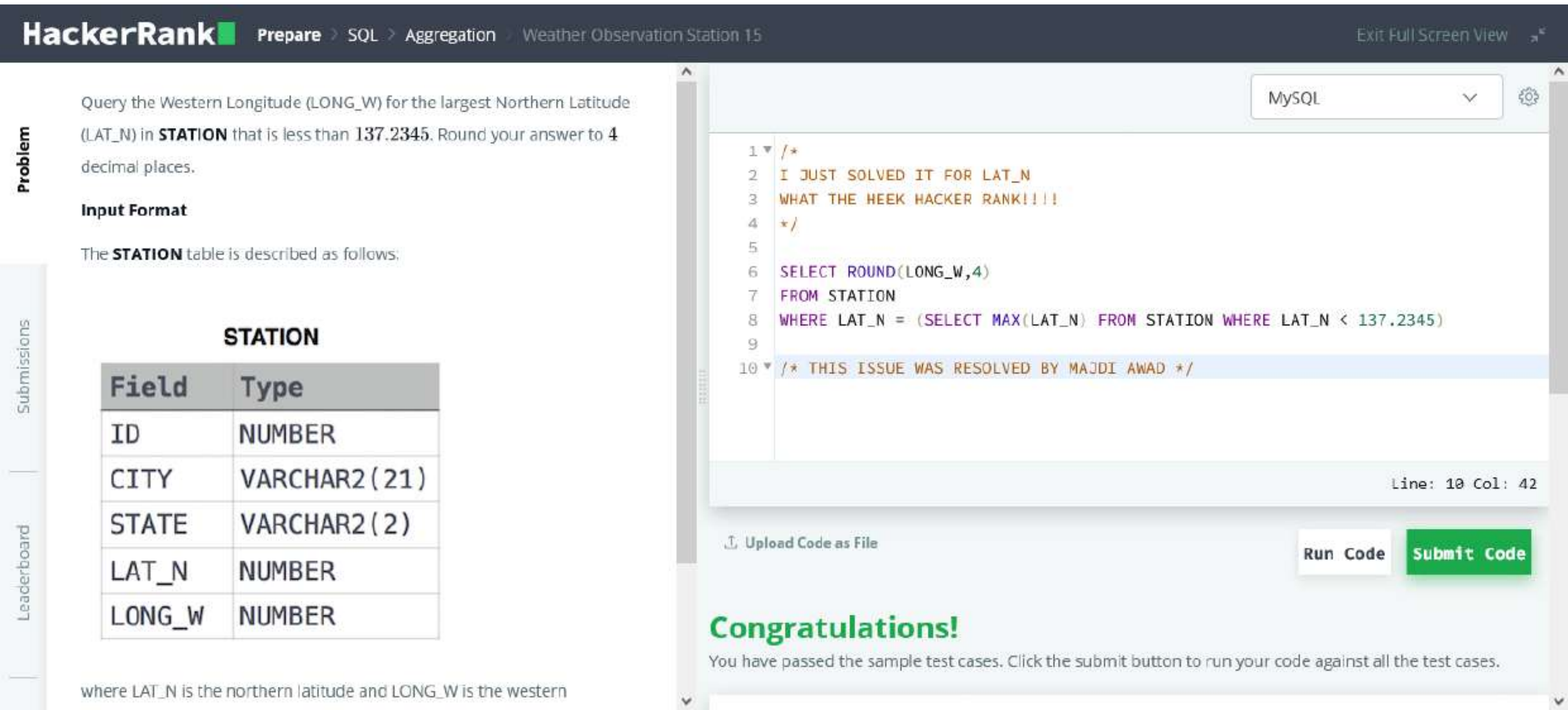
where LAT_N is the northern latitude and LONG_W is the western longitude.

```sql
/*
I SOLVED SOMETHING LIKE THIS IN THE PRVIOUS PROBLEMS
CHECK IT, I HAVE NOTHING TO SAY ABOUT THIS.
*/

SELECT ROUND(MAX(LAT_N), 4) FROM STATION WHERE LAT_N < 137.2345

/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

Line: 8 Col: 44

Congratulations!

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



Query the Western Longitude (LONG_W) for the largest Northern Latitude (LAT_N) in **STATION** that is less than 137.2345. Round your answer to 4 decimal places.

Input Format

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

STATION

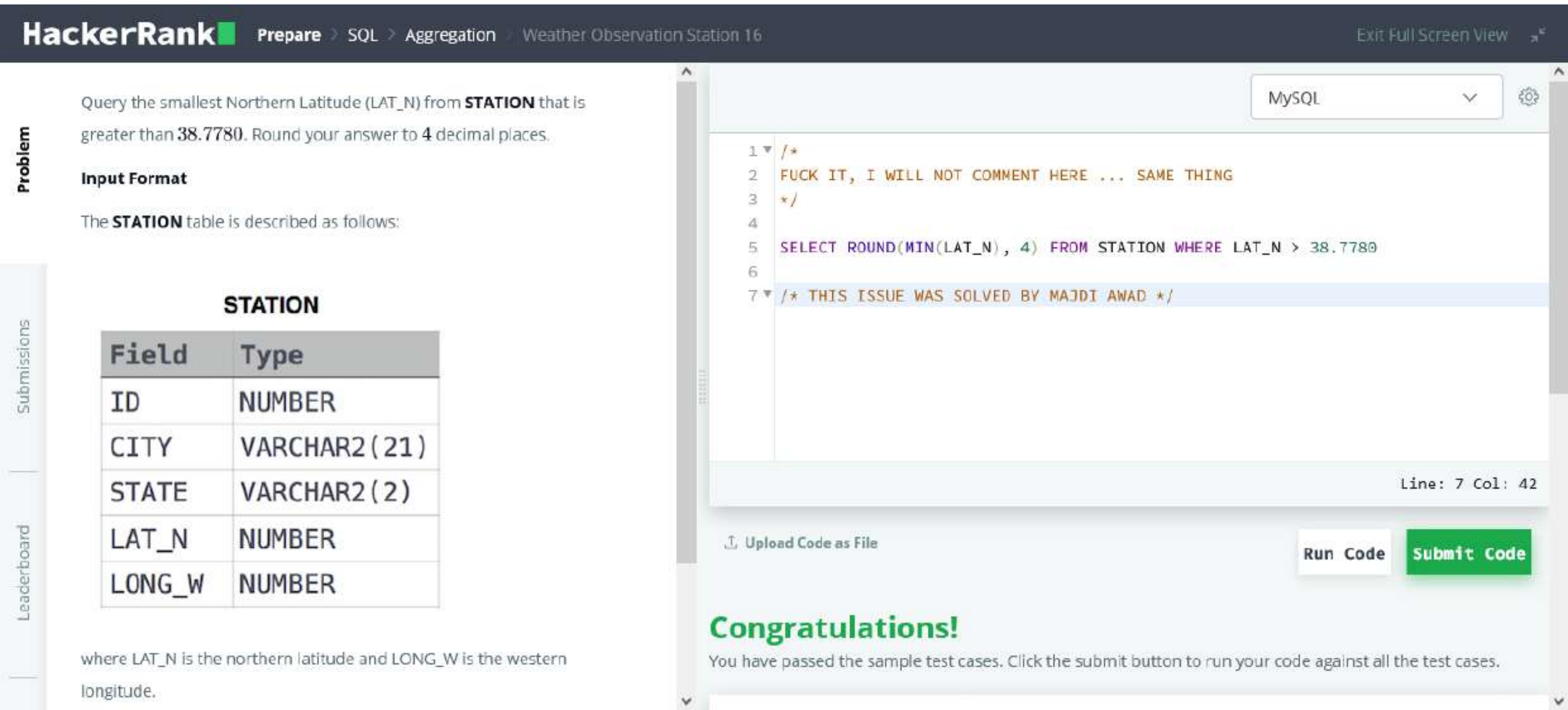
Field	Type
ID	NUMBER
CITY	VARCHAR2(21)
STATE	VARCHAR2(2)
LAT_N	NUMBER
LONG_W	NUMBER

where LAT_N is the northern latitude and LONG_W is the western

```
/*
I JUST SOLVED IT FOR LAT_N
WHAT THE HEEK HACKER RANK!!!!
*/

SELECT ROUND(LONG_W,4)
FROM STATION
WHERE LAT_N = (SELECT MAX(LAT_N) FROM STATION WHERE LAT_N < 137.2345)

/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

Congratulations!

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



Query the smallest Northern Latitude (LAT_N) from **STATION** that is greater than 38.7780. Round your answer to 4 decimal places.

Input Format

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

STATION

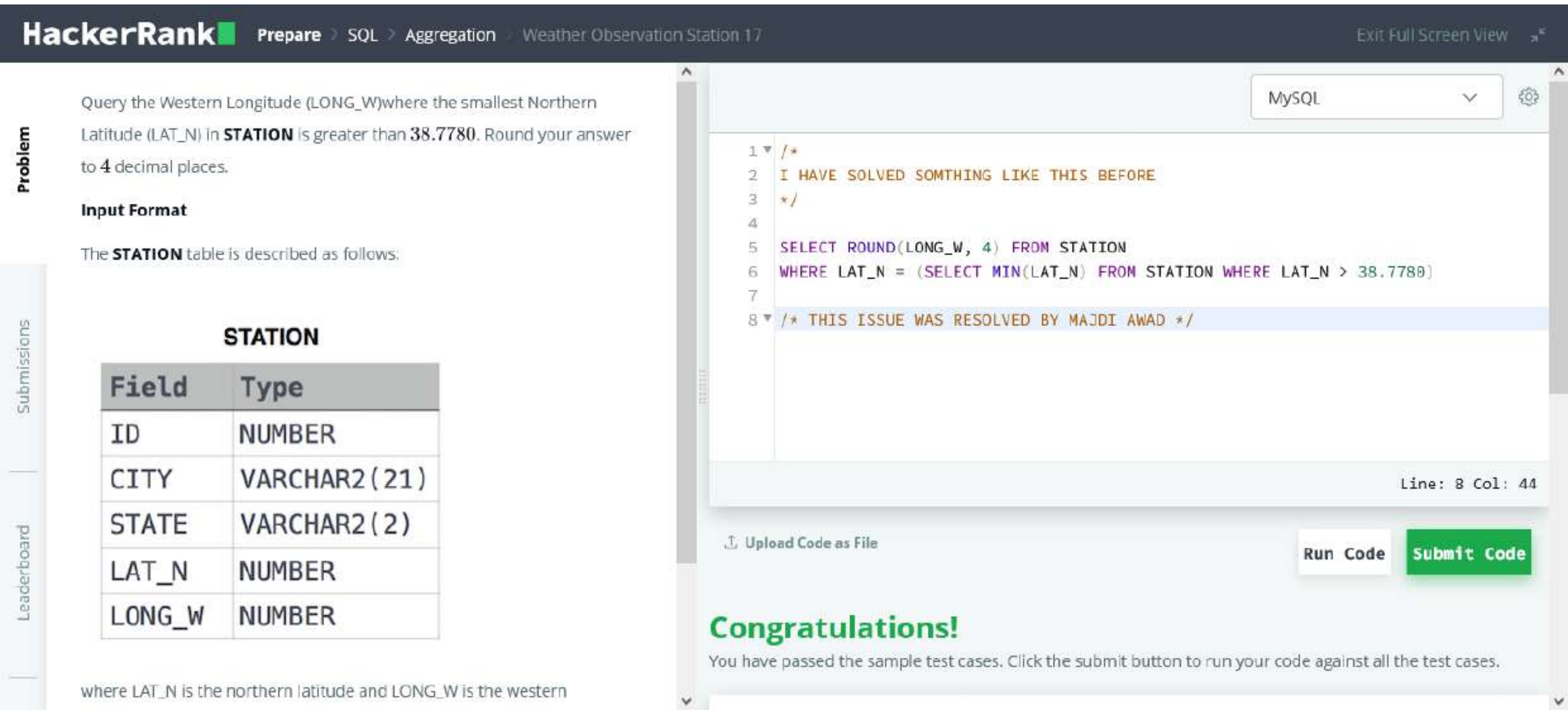
Field	Type
ID	NUMBER
CITY	VARCHAR2(21)
STATE	VARCHAR2(2)
LAT_N	NUMBER
LONG_W	NUMBER

where LAT_N is the northern latitude and LONG_W is the western longitude.

```sql
/*
FUCK IT, I WILL NOT COMMENT HERE ... SAME THING
*/

SELECT ROUND(MIN(LAT_N), 4) FROM STATION WHERE LAT_N > 38.7780

/* THIS ISSUE WAS SOLVED BY MAJDI AWAD */
```

Congratulations!

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



Query the Western Longitude (LONG_W)where the smallest Northern Latitude (LAT_N) in **STATION** is greater than 38.7780. Round your answer to 4 decimal places.

Input Format

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

STATION

Field	Type
ID	NUMBER
CITY	VARCHAR2(21)
STATE	VARCHAR2(2)
LAT_N	NUMBER
LONG_W	NUMBER

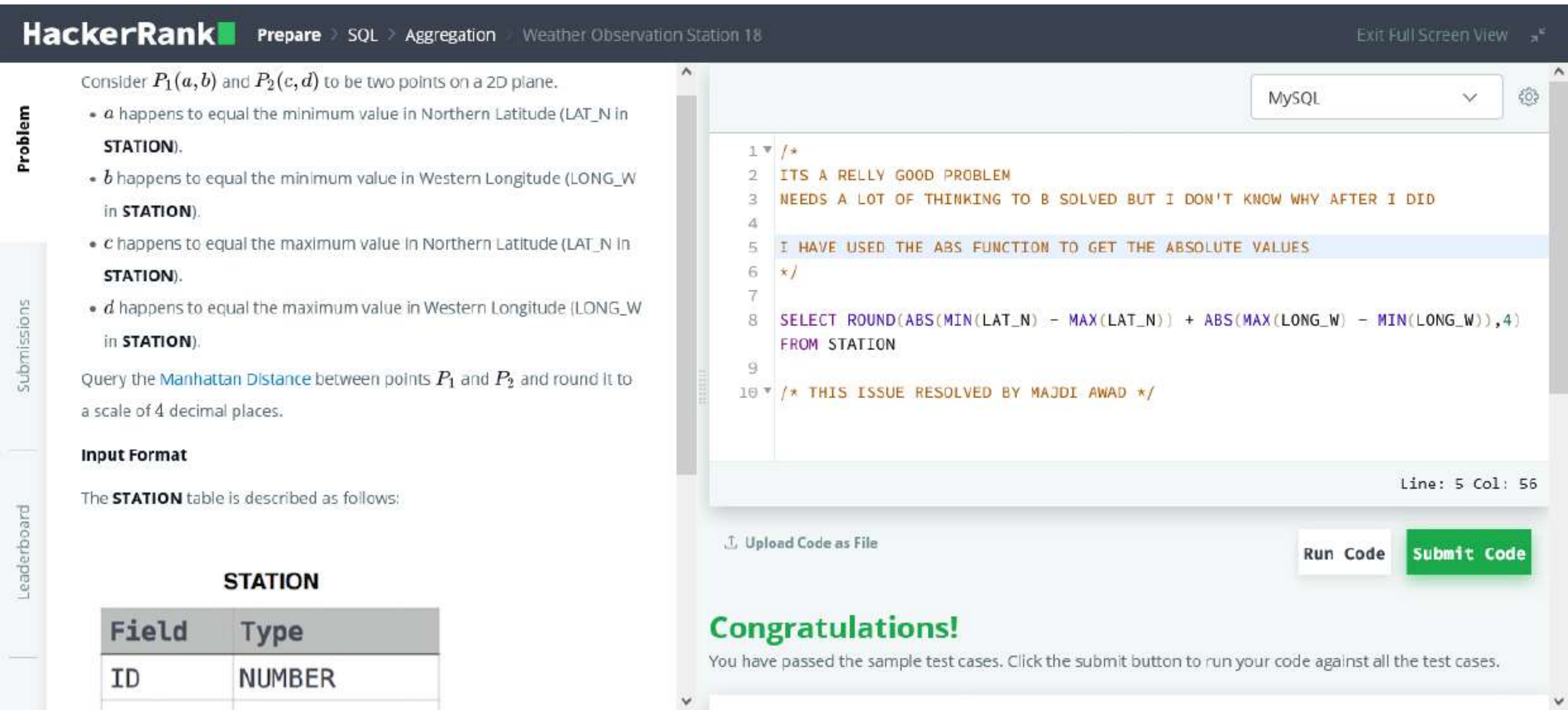
where LAT_N is the northern latitude and LONG_W is the western

```sql
/*
I HAVE SOLVED SOMTHING LIKE THIS BEFORE
*/

SELECT ROUND(LONG_W, 4) FROM STATION
WHERE LAT_N = (SELECT MIN(LAT_N) FROM STATION WHERE LAT_N > 38.7780)

/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

Line: 8 Col: 44

Congratulations!

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



Consider $P_1(a,b)$ and $P_2(c,d)$ to be two points on a 2D plane.

- a happens to equal the minimum value in Northern Latitude (LAT_N in **STATION**).
- b happens to equal the minimum value in Western Longitude (LONG_W in **STATION**).
- c happens to equal the maximum value in Northern Latitude (LAT_N in **STATION**).
- d happens to equal the maximum value in Western Longitude (LONG_W in **STATION**).

Query the Manhattan Distance between points P_1 and P_2 and round it to a scale of 4 decimal places.

Input Format

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

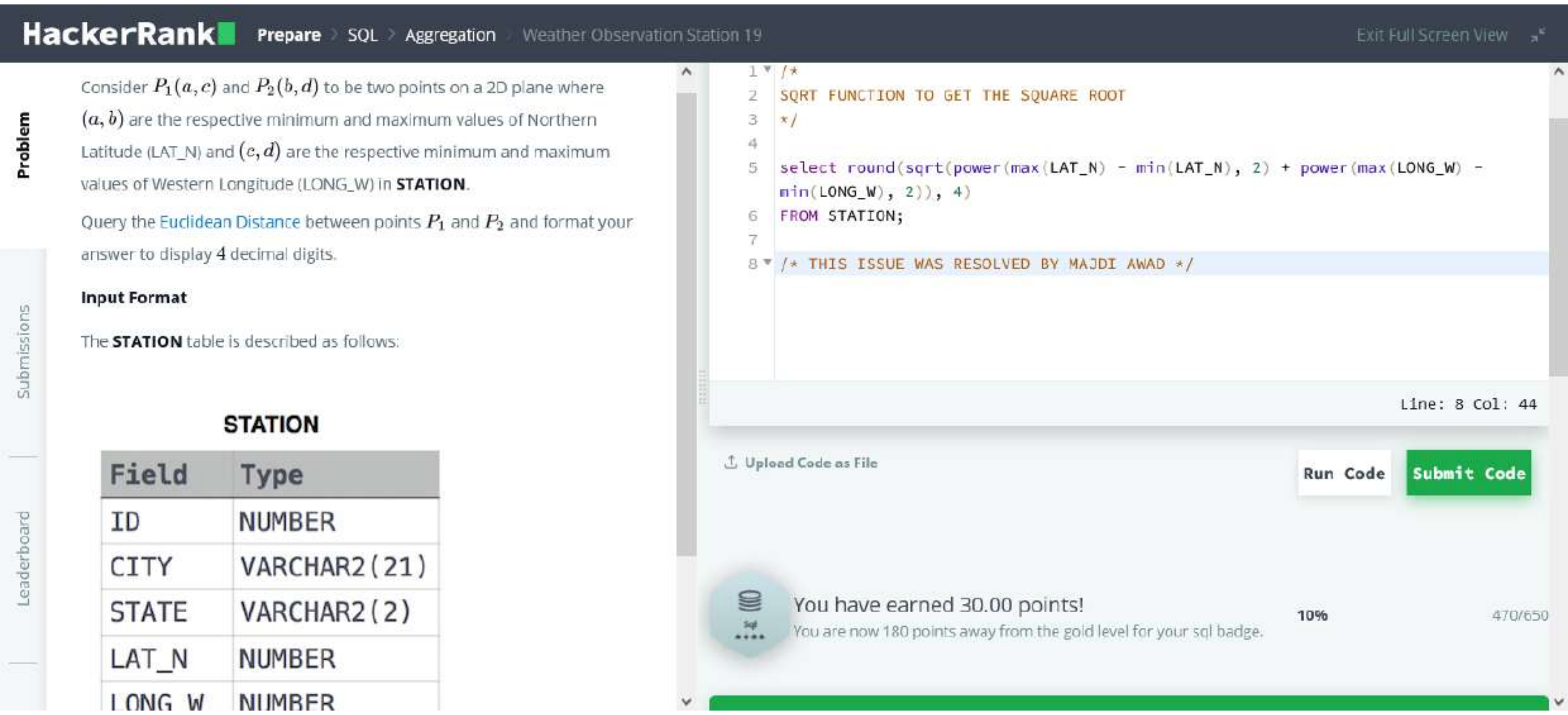
STATION

Field	Type
ID	NUMBER

```
/*
ITS A RELLY GOOD PROBLEM
NEEDS A LOT OF THINKING TO B SOLVED BUT I DON'T KNOW WHY AFTER I DID

I HAVE USED THE ABS FUNCTION TO GET THE ABSOLUTE VALUES
*/

SELECT ROUND(ABS(MIN(LAT_N) - MAX(LAT_N)) + ABS(MAX(LONG_W) - MIN(LONG_W)),4) FROM STATION

/* THIS ISSUE RESOLVED BY MAJDI AWAD */
```

Congratulations!

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



Consider $P_1(a, c)$ and $P_2(b, d)$ to be two points on a 2D plane where (a, b) are the respective minimum and maximum values of Northern Latitude (LAT_N) and (c, d) are the respective minimum and maximum values of Western Longitude (LONG_W) in **STATION**.

Query the Euclidean Distance between points P_1 and P_2 and format your answer to display 4 decimal digits.

Input Format

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

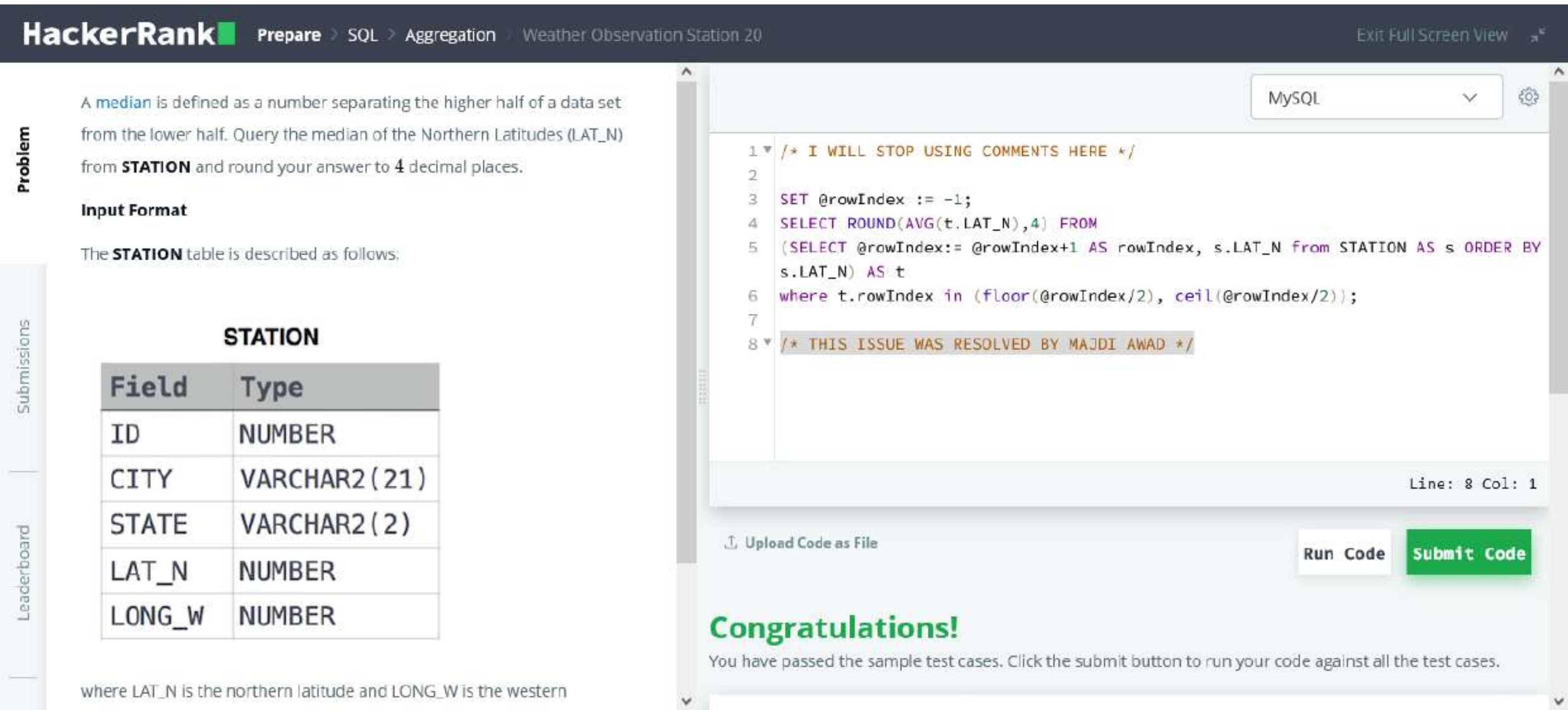
STATION

Field	Type
ID	NUMBER
CITY	VARCHAR2(21)
STATE	VARCHAR2(2)
LAT_N	NUMBER
LONG_W	NUMBER

```
/*
SQRT FUNCTION TO GET THE SQUARE ROOT
*/

select round(sqrt(power(max(LAT_N) - min(LAT_N), 2) + power(max(LONG_W) - min(LONG_W), 2)), 4)
FROM STATION;

/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

You have earned 30.00 points!

You are now 180 points away from the gold level for your sql badge.



A median is defined as a number separating the higher half of a data set from the lower half. Query the median of the Northern Latitudes (LAT_N) from **STATION** and round your answer to 4 decimal places.

Input Format

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

STATION

Field	Type
ID	NUMBER
CITY	VARCHAR2(21)
STATE	VARCHAR2(2)
LAT_N	NUMBER
LONG_W	NUMBER

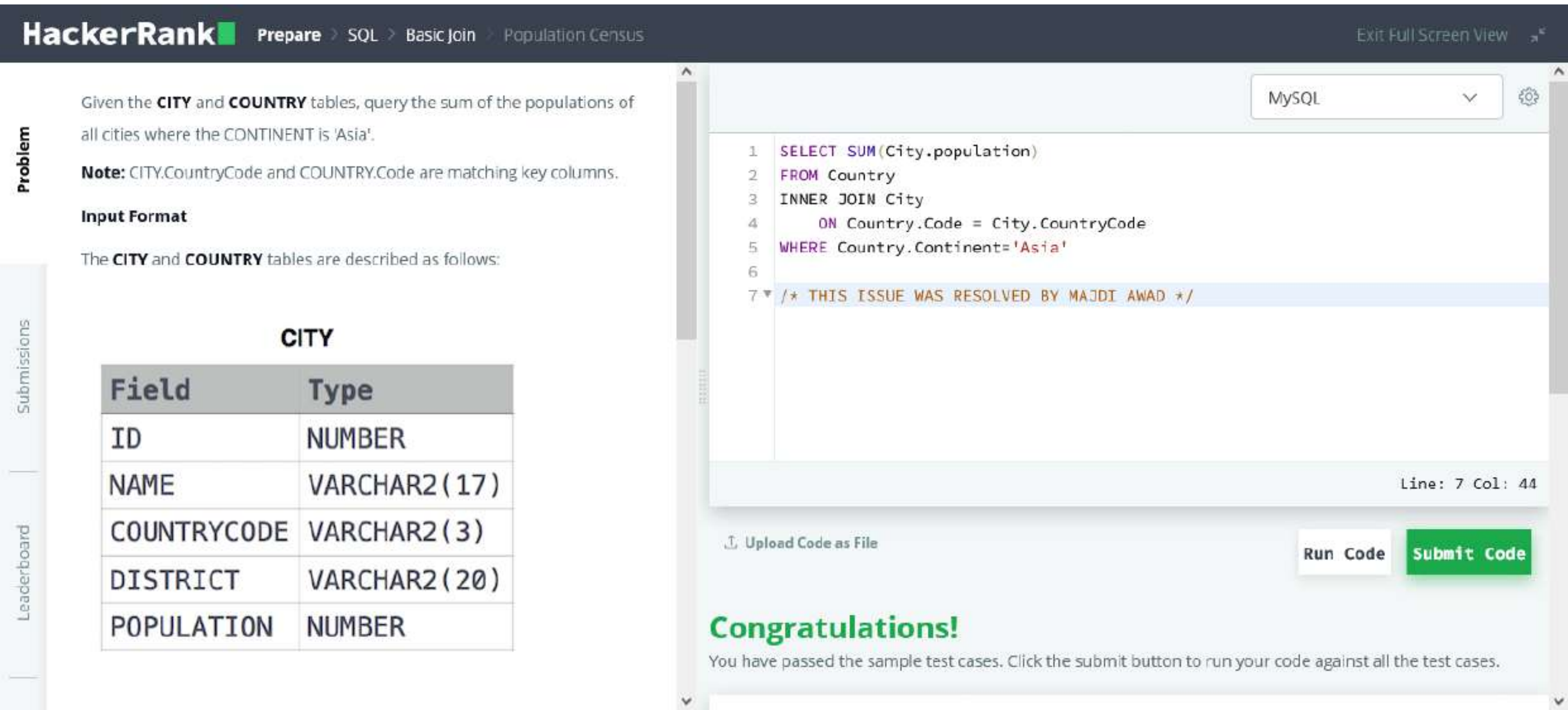
where LAT_N is the northern latitude and LONG_W is the western

```sql
/* I WILL STOP USING COMMENTS HERE */

SET @rowIndex := -1;
SELECT ROUND(AVG(t.LAT_N),4) FROM
(SELECT @rowIndex:= @rowIndex+1 AS rowIndex, s.LAT_N from STATION AS s ORDER BY s.LAT_N) AS t
where t.rowIndex in (floor(@rowIndex/2), ceil(@rowIndex/2));

/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

Line: 8 Col: 1

Congratulations!

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



Given the **CITY** and **COUNTRY** tables, query the sum of the populations of all cities where the CONTINENT is 'Asia'.

Note: CITY.CountryCode and COUNTRY.Code are matching key columns.

Input Format

The **CITY** and **COUNTRY** tables are described as follows:

CITY

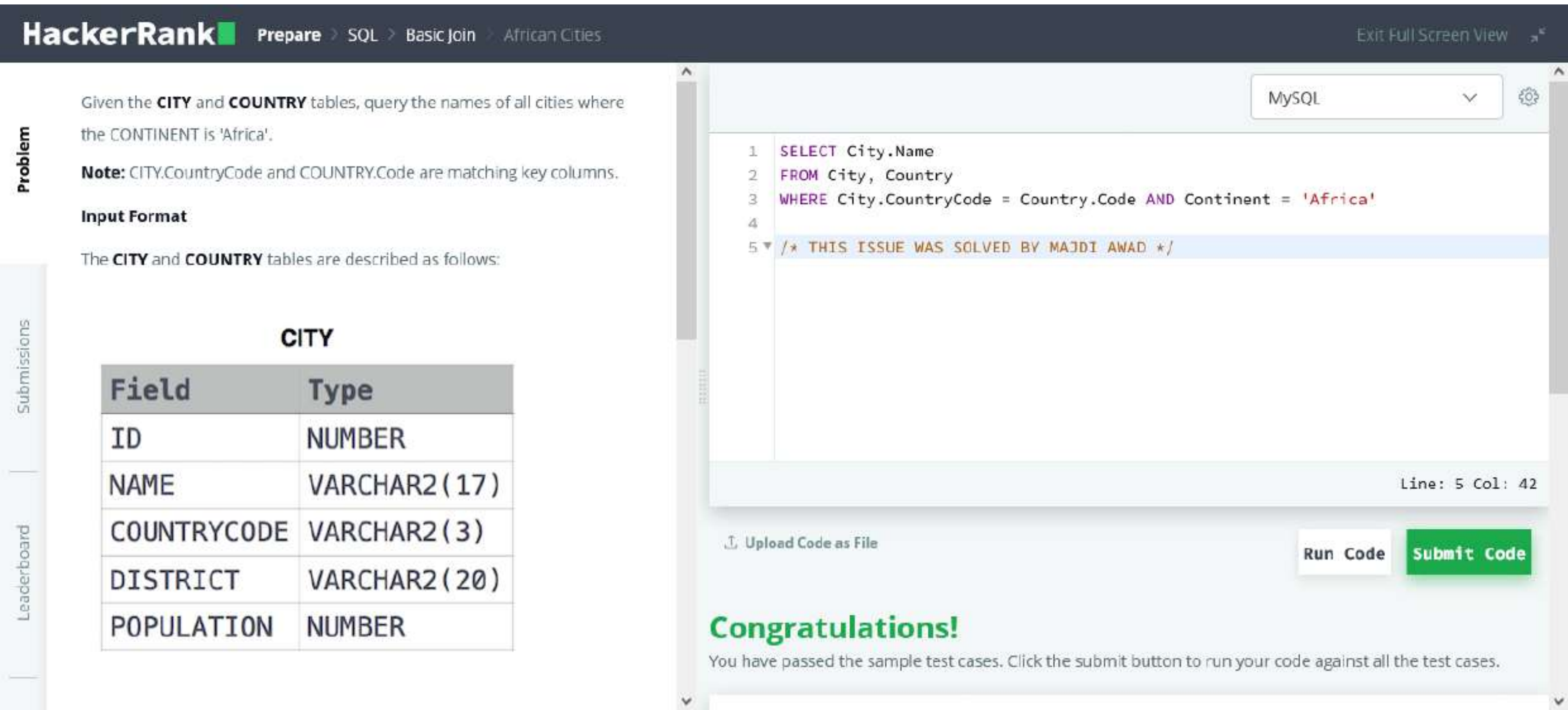
Field	Type
ID	NUMBER
NAME	VARCHAR2(17)
COUNTRYCODE	VARCHAR2(3)
DISTRICT	VARCHAR2(20)
POPULATION	NUMBER

```sql
SELECT SUM(City.population)
FROM Country
INNER JOIN City
    ON Country.Code = City.CountryCode
WHERE Country.Continent='Asia'

/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

Line: 7 Col: 44

Congratulations!

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



Given the **CITY** and **COUNTRY** tables, query the names of all cities where the CONTINENT is 'Africa'.

Note: CITY.CountryCode and COUNTRY.Code are matching key columns.

Input Format

The **CITY** and **COUNTRY** tables are described as follows:

CITY

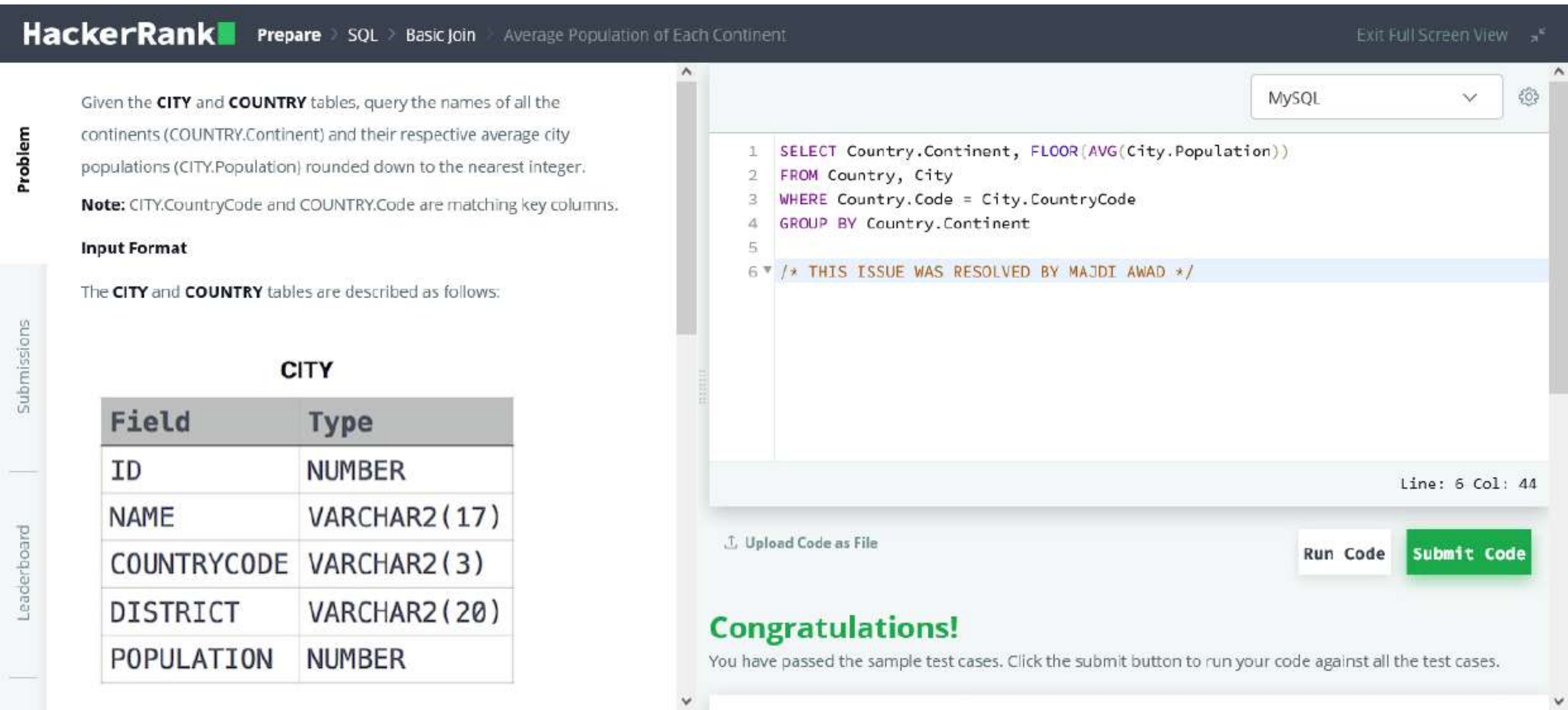
Field	Type
ID	NUMBER
NAME	VARCHAR2(17)
COUNTRYCODE	VARCHAR2(3)
DISTRICT	VARCHAR2(20)
POPULATION	NUMBER

```sql
SELECT City.Name
FROM City, Country
WHERE City.CountryCode = Country.Code AND Continent = 'Africa'

/* THIS ISSUE WAS SOLVED BY MAJDI AWAD */
```

Line: 5 Col: 42

Congratulations!

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



Given the **CITY** and **COUNTRY** tables, query the names of all the continents (COUNTRY.Continent) and their respective average city populations (CITY.Population) rounded down to the nearest integer.

Note: CITY.CountryCode and COUNTRY.Code are matching key columns.

Input Format

The **CITY** and **COUNTRY** tables are described as follows:

CITY

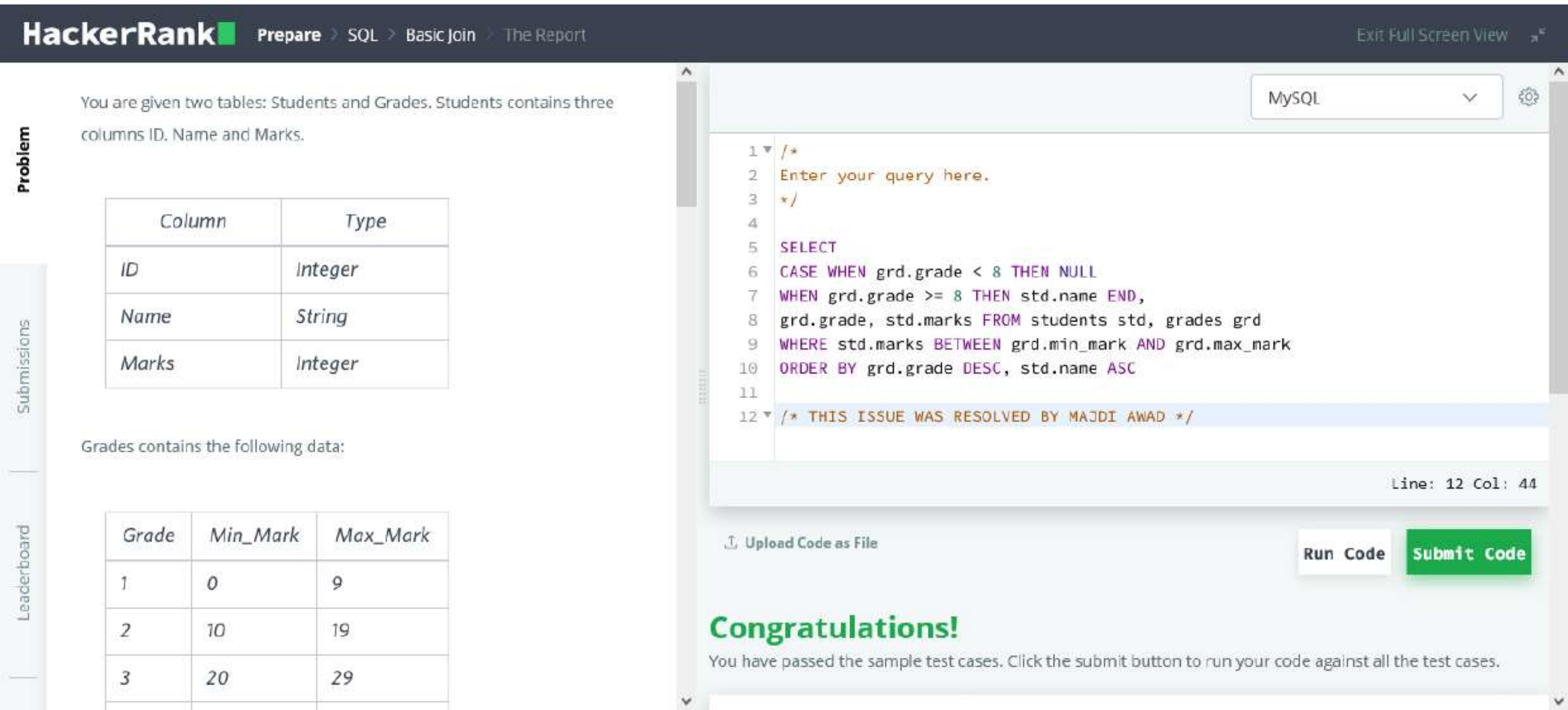
Field	Type
ID	NUMBER
NAME	VARCHAR2(17)
COUNTRYCODE	VARCHAR2(3)
DISTRICT	VARCHAR2(20)
POPULATION	NUMBER

```sql
SELECT Country.Continent, FLOOR(AVG(City.Population))
FROM Country, City
WHERE Country.Code = City.CountryCode
GROUP BY Country.Continent

/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

Line: 6 Col: 44

Congratulations!

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



You are given two tables: Students and Grades. Students contains three columns ID, Name and Marks.

Column	Type
ID	Integer
Name	String
Marks	Integer

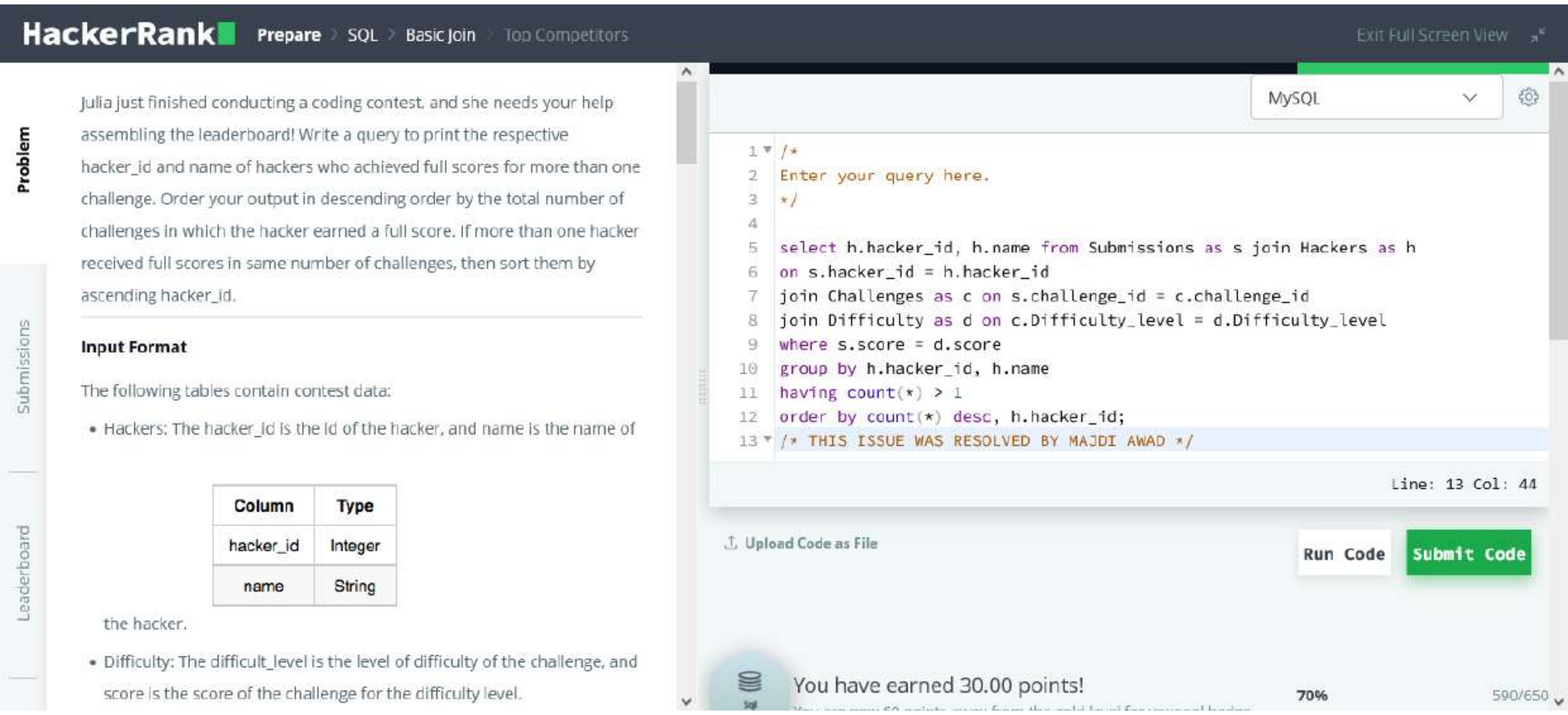
Grades contains the following data:

Grade	Min_Mark	Max_Mark
1	0	9
2	10	19
3	20	29

```sql
/*
Enter your query here.
*/

SELECT
CASE WHEN grd.grade < 8 THEN NULL
WHEN grd.grade >= 8 THEN std.name END,
grd.grade, std.marks FROM students std, grades grd
WHERE std.marks BETWEEN grd.min_mark AND grd.max_mark
ORDER BY grd.grade DESC, std.name ASC

/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

Congratulations!

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



Julia just finished conducting a coding contest, and she needs your help assembling the leaderboard! Write a query to print the respective hacker_id and name of hackers who achieved full scores for more than one challenge. Order your output in descending order by the total number of challenges in which the hacker earned a full score. If more than one hacker received full scores in same number of challenges, then sort them by ascending hacker_id.

Input Format

The following tables contain contest data:

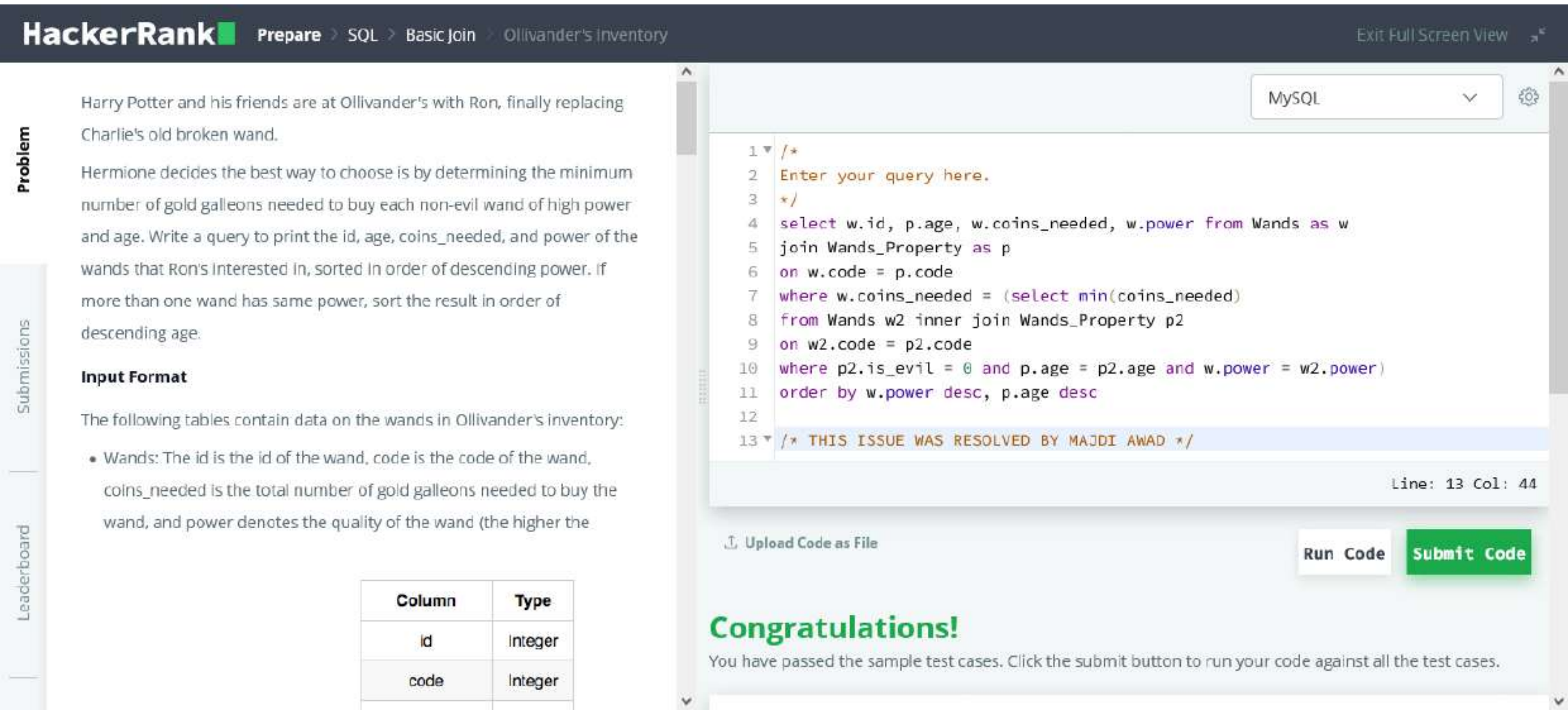
- Hackers: The hacker_id is the id of the hacker, and name is the name of the hacker.

Column	Type
hacker_id	Integer
name	String

- Difficulty: The difficult_level is the level of difficulty of the challenge, and score is the score of the challenge for the difficulty level.

```sql
/*
Enter your query here.
*/

select h.hacker_id, h.name from Submissions as s join Hackers as h
on s.hacker_id = h.hacker_id
join Challenges as c on s.challenge_id = c.challenge_id
join Difficulty as d on c.Difficulty_level = d.Difficulty_level
where s.score = d.score
group by h.hacker_id, h.name
having count(*) > 1
order by count(*) desc, h.hacker_id;
/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

You have earned 30.00 points!



Harry Potter and his friends are at Ollivander's with Ron, finally replacing Charlie's old broken wand.

Hermione decides the best way to choose is by determining the minimum number of gold galleons needed to buy each non-evil wand of high power and age. Write a query to print the id, age, coins_needed, and power of the wands that Ron's interested in, sorted in order of descending power. If more than one wand has same power, sort the result in order of descending age.

Input Format

The following tables contain data on the wands in Ollivander's inventory:

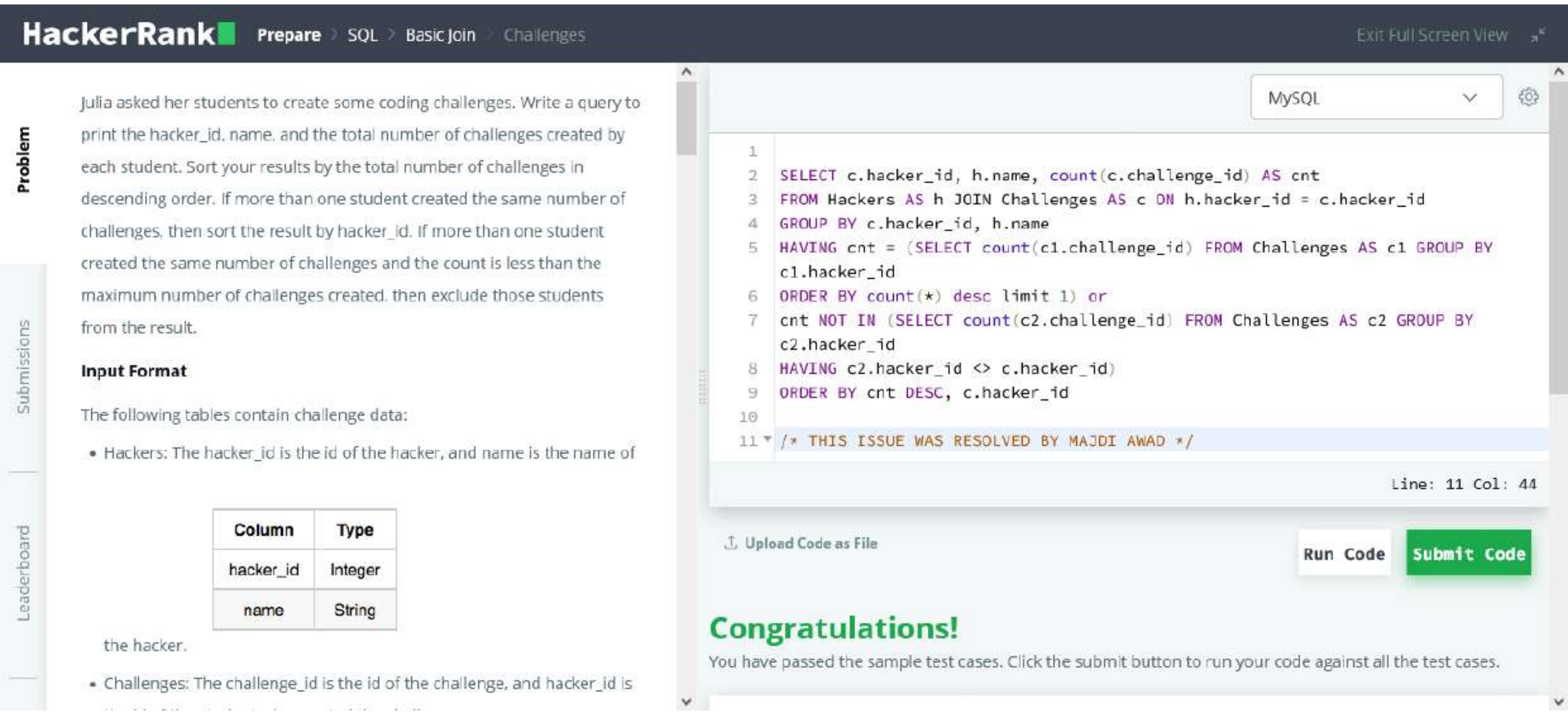
- Wands: The id is the id of the wand, code is the code of the wand, coins_needed is the total number of gold galleons needed to buy the wand, and power denotes the quality of the wand (the higher the

Column	Type
id	Integer
code	Integer

```
/*
Enter your query here.
*/
select w.id, p.age, w.coins_needed, w.power from Wands as w
join Wands_Property as p
on w.code = p.code
where w.coins_needed = (select min(coins_needed)
from Wands w2 inner join Wands_Property p2
on w2.code = p2.code
where p2.is_evil = 0 and p.age = p2.age and w.power = w2.power)
order by w.power desc, p.age desc

/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

Congratulations!

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



Julia asked her students to create some coding challenges. Write a query to print the hacker_id, name, and the total number of challenges created by each student. Sort your results by the total number of challenges in descending order. If more than one student created the same number of challenges, then sort the result by hacker_id. If more than one student created the same number of challenges and the count is less than the maximum number of challenges created, then exclude those students from the result.

Input Format

The following tables contain challenge data:

- Hackers: The hacker_id is the id of the hacker, and name is the name of the hacker.

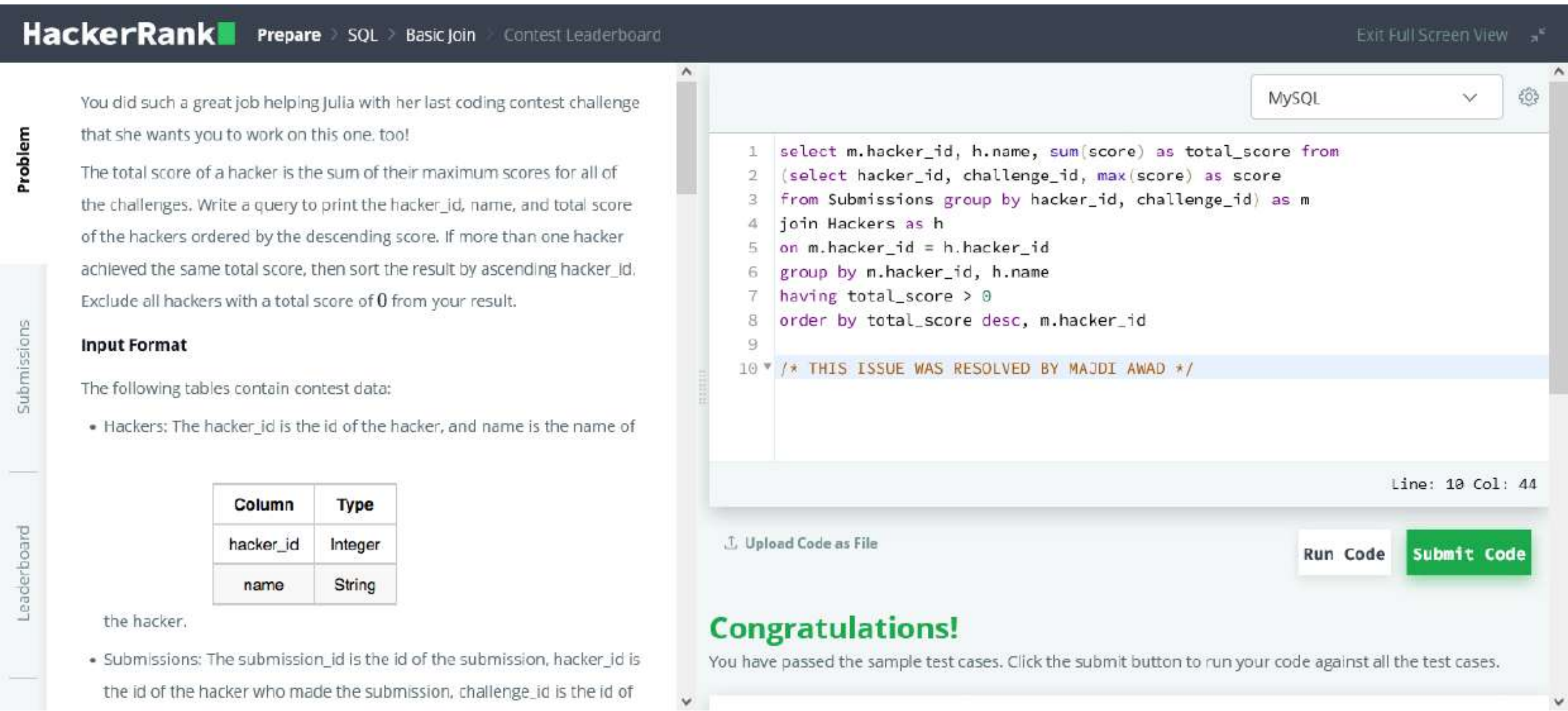
Column	Type
hacker_id	Integer
name	String

- Challenges: The challenge_id is the id of the challenge, and hacker_id is

```

SELECT c.hacker_id, h.name, count(c.challenge_id) AS cnt
FROM Hackers AS h JOIN Challenges AS c ON h.hacker_id = c.hacker_id
GROUP BY c.hacker_id, h.name
HAVING cnt = (SELECT count(c1.challenge_id) FROM Challenges AS c1 GROUP BY c1.hacker_id
ORDER BY count(*) desc limit 1) or
cnt NOT IN (SELECT count(c2.challenge_id) FROM Challenges AS c2 GROUP BY c2.hacker_id
HAVING c2.hacker_id <> c.hacker_id)
ORDER BY cnt DESC, c.hacker_id

/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

Congratulations!

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



You did such a great job helping Julia with her last coding contest challenge that she wants you to work on this one, too!

The total score of a hacker is the sum of their maximum scores for all of the challenges. Write a query to print the hacker_id, name, and total score of the hackers ordered by the descending score. If more than one hacker achieved the same total score, then sort the result by ascending hacker_id. Exclude all hackers with a total score of 0 from your result.

Input Format

The following tables contain contest data:

- Hackers: The hacker_id is the id of the hacker, and name is the name of the hacker.

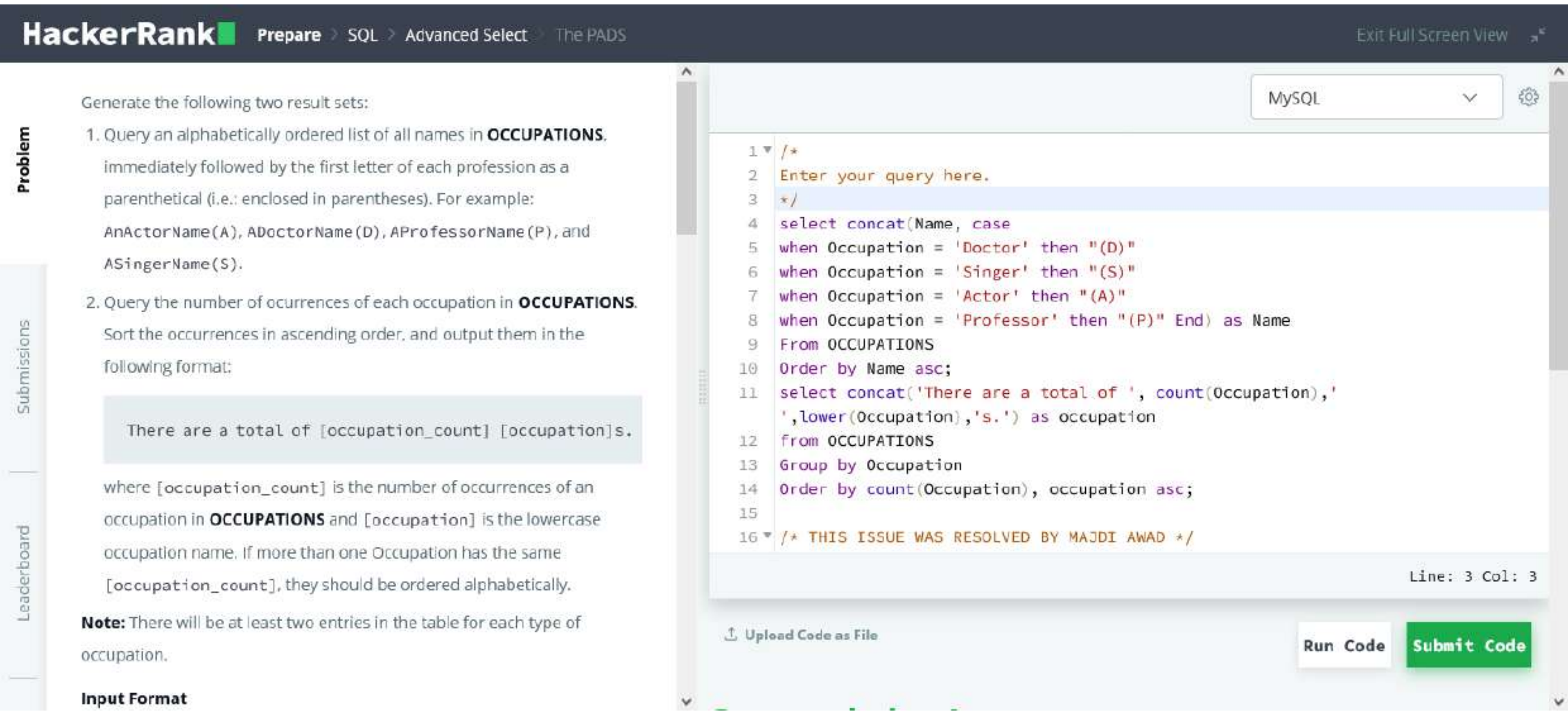
Column	Type
hacker_id	Integer
name	String

- Submissions: The submission_id is the id of the submission, hacker_id is the id of the hacker who made the submission, challenge_id is the id of

```sql
select m.hacker_id, h.name, sum(score) as total_score from
(select hacker_id, challenge_id, max(score) as score
from Submissions group by hacker_id, challenge_id) as m
join Hackers as h
on m.hacker_id = h.hacker_id
group by m.hacker_id, h.name
having total_score > 0
order by total_score desc, m.hacker_id

/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

Congratulations!

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



Generate the following two result sets:

1. Query an alphabetically ordered list of all names in **OCCUPATIONS**, immediately followed by the first letter of each profession as a parenthetical (i.e.: enclosed in parentheses). For example: `AnActorName(A)`, `ADoctorName(D)`, `AProfessorName(P)`, and `ASingerName(S)`.
2. Query the number of ocurrences of each occupation in **OCCUPATIONS**. Sort the occurrences in ascending order, and output them in the following format:

```
There are a total of [occupation_count] [occupation]s.
```

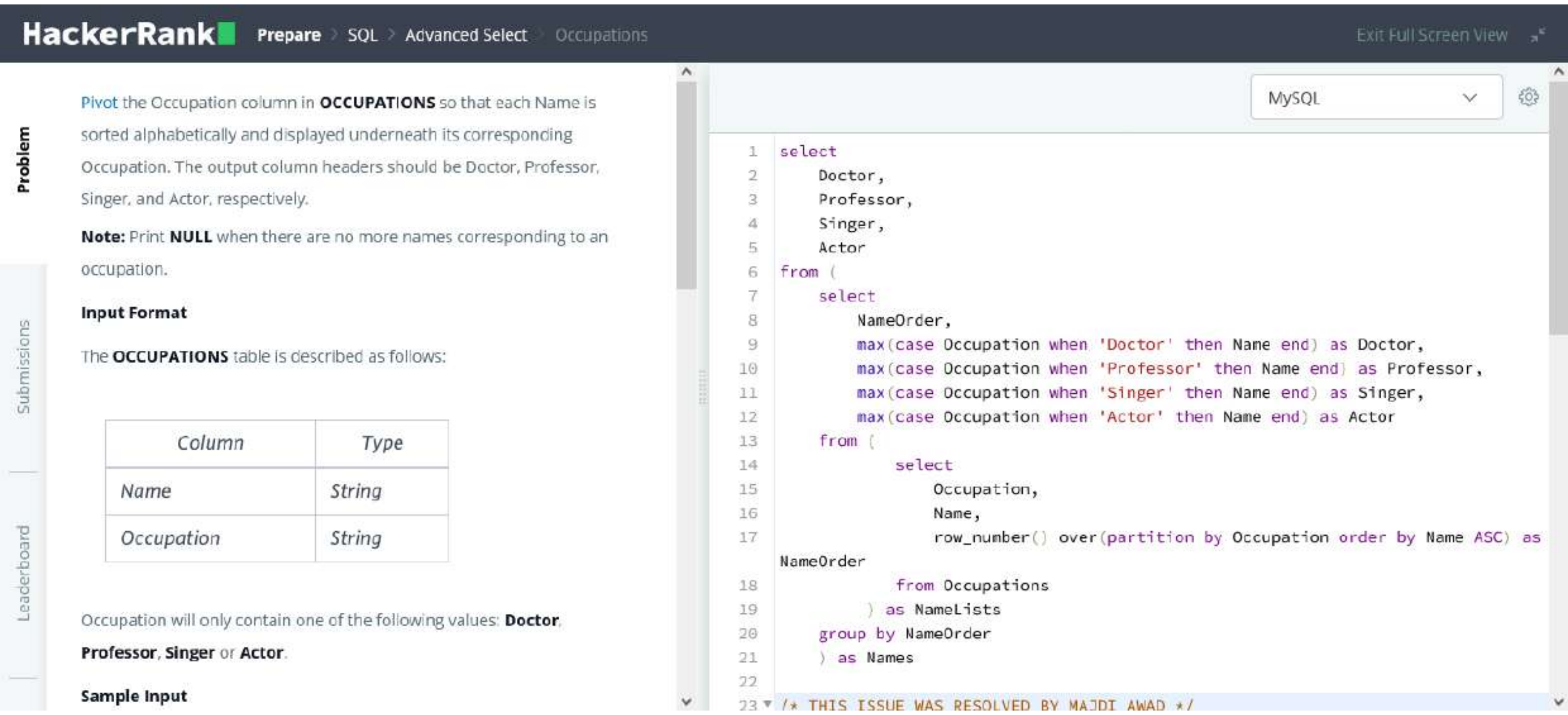
where `[occupation_count]` is the number of occurrences of an occupation in **OCCUPATIONS** and `[occupation]` is the lowercase occupation name. If more than one Occupation has the same `[occupation_count]`, they should be ordered alphabetically.

Note: There will be at least two entries in the table for each type of occupation.

Input Format

```
/*
Enter your query here.
*/
select concat(Name, case
when Occupation = 'Doctor' then "(D)"
when Occupation = 'Singer' then "(S)"
when Occupation = 'Actor' then "(A)"
when Occupation = 'Professor' then "(P)" End) as Name
From OCCUPATIONS
Order by Name asc;
select concat('There are a total of ', count(Occupation),'
',lower(Occupation),'s.') as occupation
from OCCUPATIONS
Group by Occupation
Order by count(Occupation), occupation asc;

/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

Line: 3 Col: 3



[Pivot](#) the Occupation column in **OCCUPATIONS** so that each Name is sorted alphabetically and displayed underneath its corresponding Occupation. The output column headers should be Doctor, Professor, Singer, and Actor, respectively.

Note: Print **NULL** when there are no more names corresponding to an occupation.

Input Format

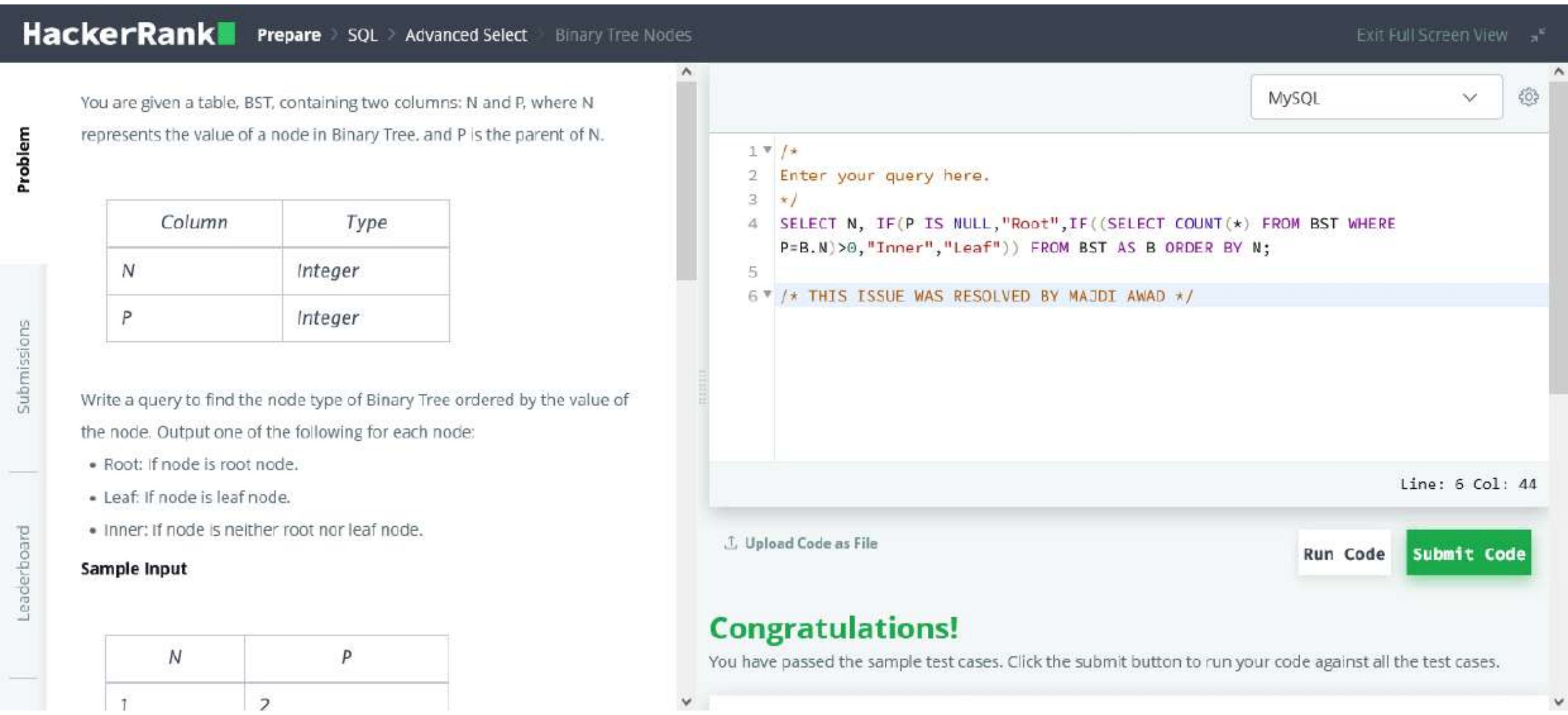
The **OCCUPATIONS** table is described as follows:

Column	*Type*
Name	*String*
Occupation	*String*

Occupation will only contain one of the following values: **Doctor**, **Professor**, **Singer** or **Actor**.

Sample Input

```
select
    Doctor,
    Professor,
    Singer,
    Actor
from (
    select
        NameOrder,
        max(case Occupation when 'Doctor' then Name end) as Doctor,
        max(case Occupation when 'Professor' then Name end) as Professor,
        max(case Occupation when 'Singer' then Name end) as Singer,
        max(case Occupation when 'Actor' then Name end) as Actor
    from (
            select
                Occupation,
                Name,
                row_number() over(partition by Occupation order by Name ASC) as
NameOrder
            from Occupations
         ) as NameLists
    group by NameOrder
    ) as Names

/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```



You are given a table, BST, containing two columns: N and P, where N represents the value of a node in Binary Tree, and P is the parent of N.

Column	*Type*
N	*Integer*
P	*Integer*

Write a query to find the node type of Binary Tree ordered by the value of the node. Output one of the following for each node:

- Root: If node is root node.
- Leaf: If node is leaf node.
- Inner: If node is neither root nor leaf node.

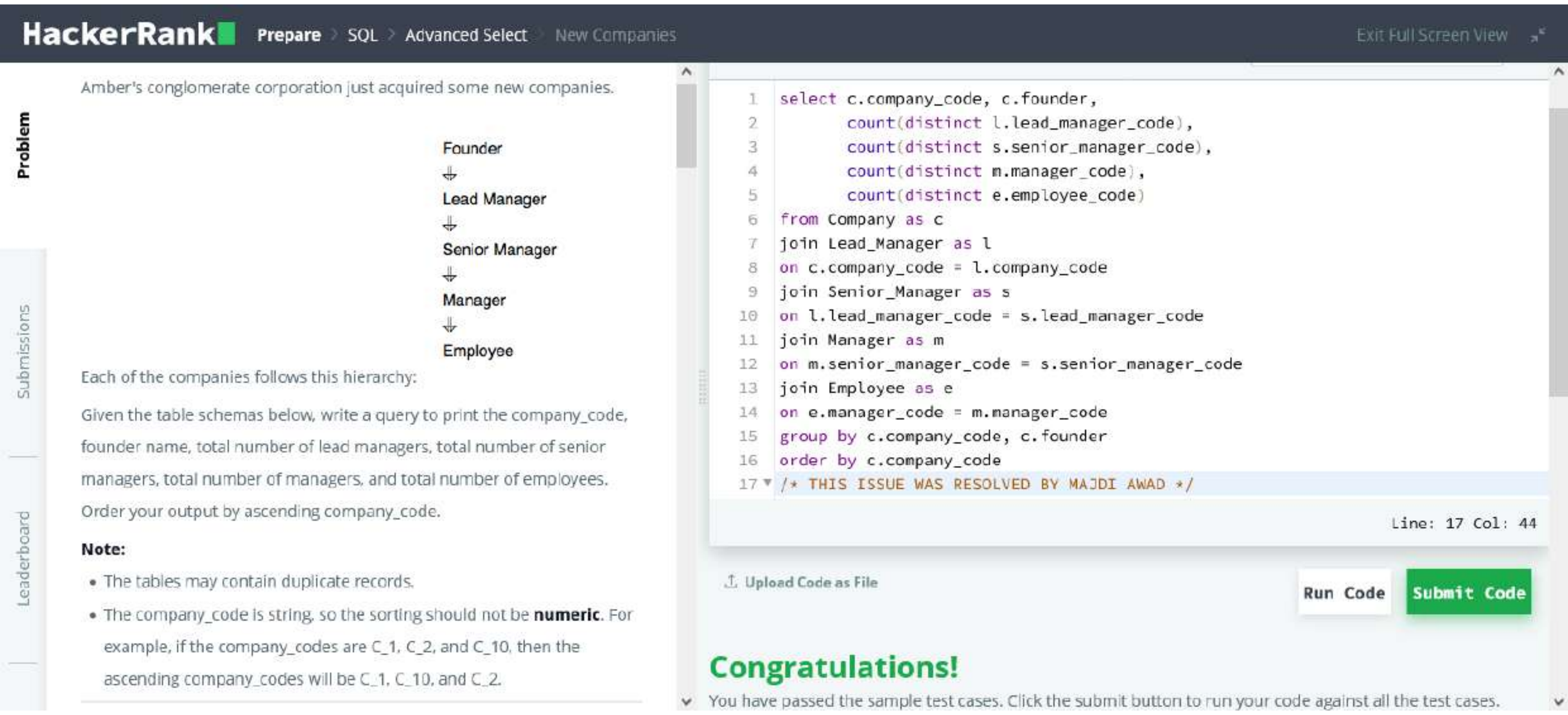
Sample Input

N	*P*
1	2

```
/*
Enter your query here.
*/
SELECT N, IF(P IS NULL,"Root",IF((SELECT COUNT(*) FROM BST WHERE
P=B.N)>0,"Inner","Leaf")) FROM BST AS B ORDER BY N;

/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

Congratulations!

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



Amber's conglomerate corporation just acquired some new companies.

Founder
↓
Lead Manager
↓
Senior Manager
↓
Manager
↓
Employee

Each of the companies follows this hierarchy:

Given the table schemas below, write a query to print the company_code, founder name, total number of lead managers, total number of senior managers, total number of managers, and total number of employees. Order your output by ascending company_code.

Note:

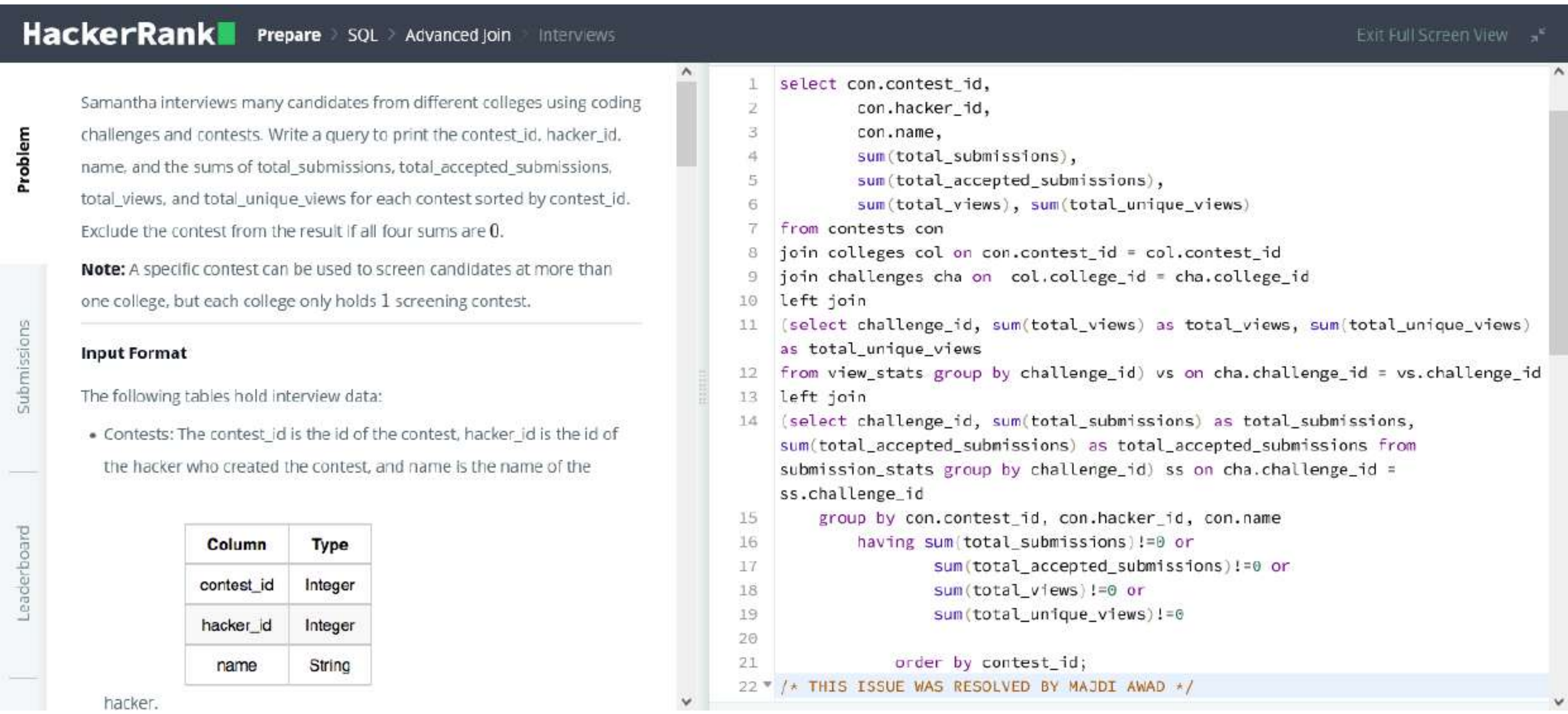
- The tables may contain duplicate records.
- The company_code is string, so the sorting should not be **numeric**. For example, if the company_codes are C_1, C_2, and C_10, then the ascending company_codes will be C_1, C_10, and C_2.

```sql
select c.company_code, c.founder,
       count(distinct l.lead_manager_code),
       count(distinct s.senior_manager_code),
       count(distinct m.manager_code),
       count(distinct e.employee_code)
from Company as c
join Lead_Manager as l
on c.company_code = l.company_code
join Senior_Manager as s
on l.lead_manager_code = s.lead_manager_code
join Manager as m
on m.senior_manager_code = s.senior_manager_code
join Employee as e
on e.manager_code = m.manager_code
group by c.company_code, c.founder
order by c.company_code
/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

Line: 17 Col: 44

Congratulations!

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



Samantha interviews many candidates from different colleges using coding challenges and contests. Write a query to print the contest_id, hacker_id, name, and the sums of total_submissions, total_accepted_submissions, total_views, and total_unique_views for each contest sorted by contest_id. Exclude the contest from the result if all four sums are 0.

Note: A specific contest can be used to screen candidates at more than one college, but each college only holds 1 screening contest.

Input Format

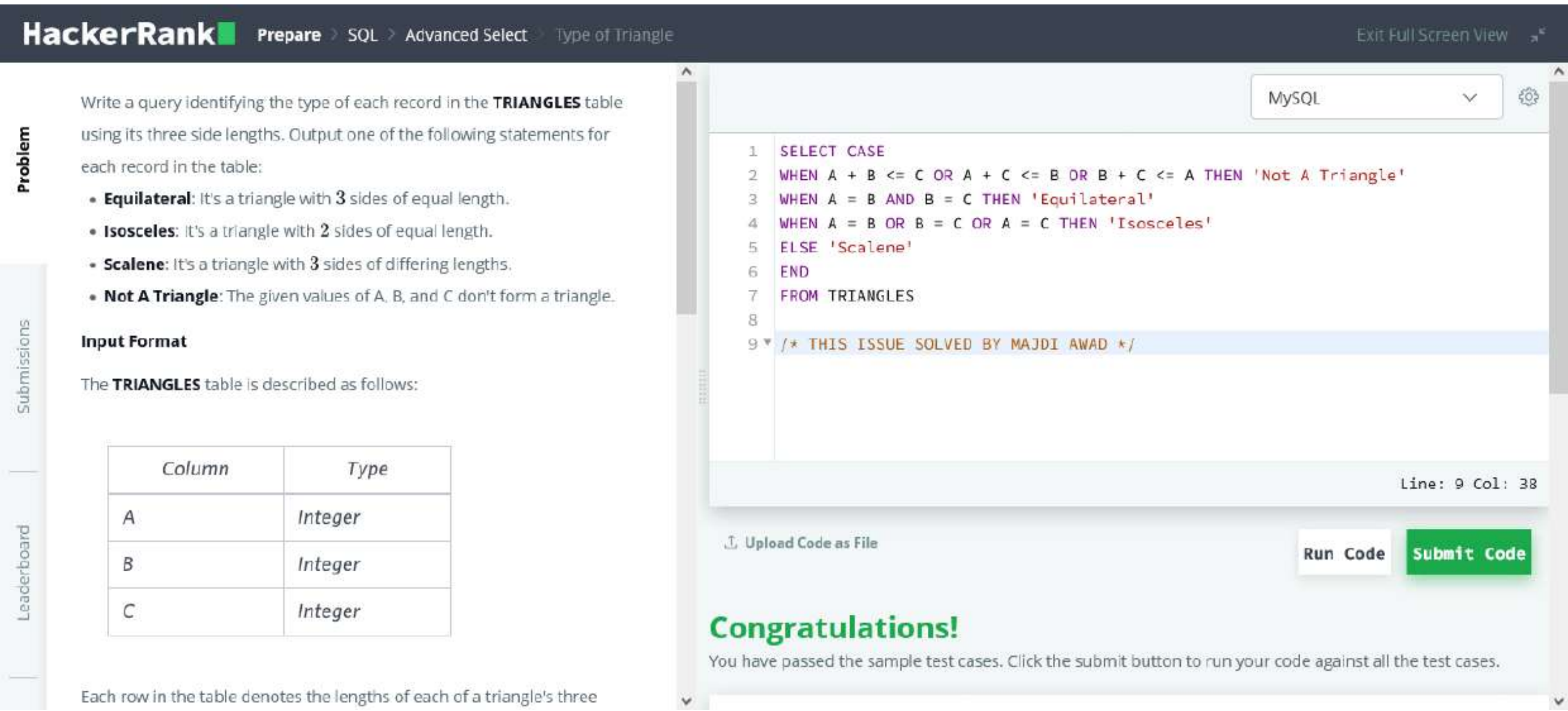
The following tables hold interview data:

- Contests: The contest_id is the id of the contest, hacker_id is the id of the hacker who created the contest, and name is the name of the

Column	Type
contest_id	Integer
hacker_id	Integer
name	String

hacker.

```sql
select con.contest_id,
        con.hacker_id,
        con.name,
        sum(total_submissions),
        sum(total_accepted_submissions),
        sum(total_views), sum(total_unique_views)
from contests con
join colleges col on con.contest_id = col.contest_id
join challenges cha on  col.college_id = cha.college_id
left join
(select challenge_id, sum(total_views) as total_views, sum(total_unique_views)
as total_unique_views
from view_stats group by challenge_id) vs on cha.challenge_id = vs.challenge_id
left join
(select challenge_id, sum(total_submissions) as total_submissions,
sum(total_accepted_submissions) as total_accepted_submissions from
submission_stats group by challenge_id) ss on cha.challenge_id =
ss.challenge_id
    group by con.contest_id, con.hacker_id, con.name
        having sum(total_submissions)!=0 or
                sum(total_accepted_submissions)!=0 or
                sum(total_views)!=0 or
                sum(total_unique_views)!=0

            order by contest_id;
/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```



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*

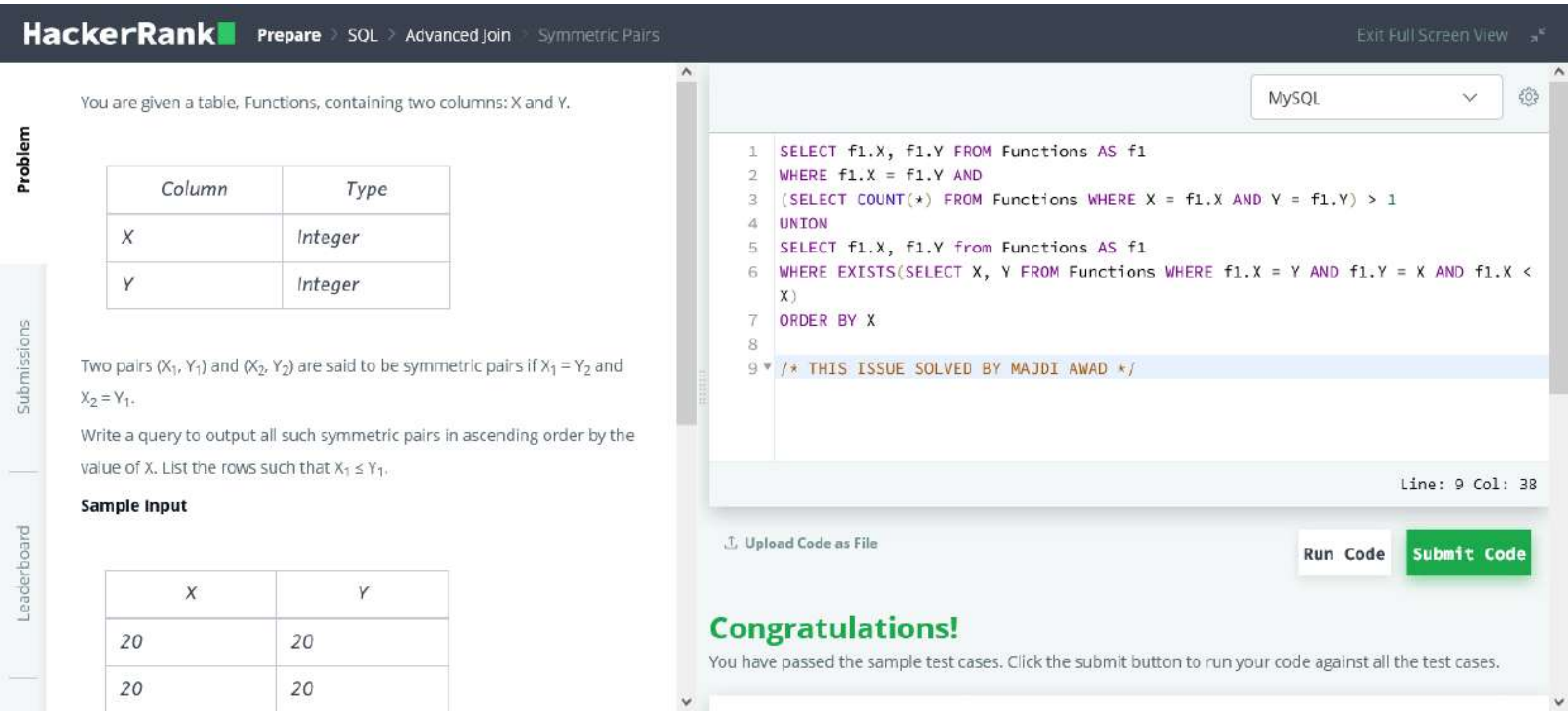
Each row in the table denotes the lengths of each of a triangle's three

```
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 B = C OR A = C THEN 'Isosceles'
ELSE 'Scalene'
END
FROM TRIANGLES

/* THIS ISSUE SOLVED BY MAJDI AWAD */
```

Line: 9 Col: 38

Congratulations!

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



You are given a table, Functions, containing two columns: X and Y.

Column	Type
X	Integer
Y	Integer

Two pairs (X_1, Y_1) and (X_2, Y_2) are said to be symmetric pairs if $X_1 = Y_2$ and $X_2 = Y_1$.

Write a query to output all such symmetric pairs in ascending order by the value of X. List the rows such that $X_1 \leq Y_1$.

Sample Input

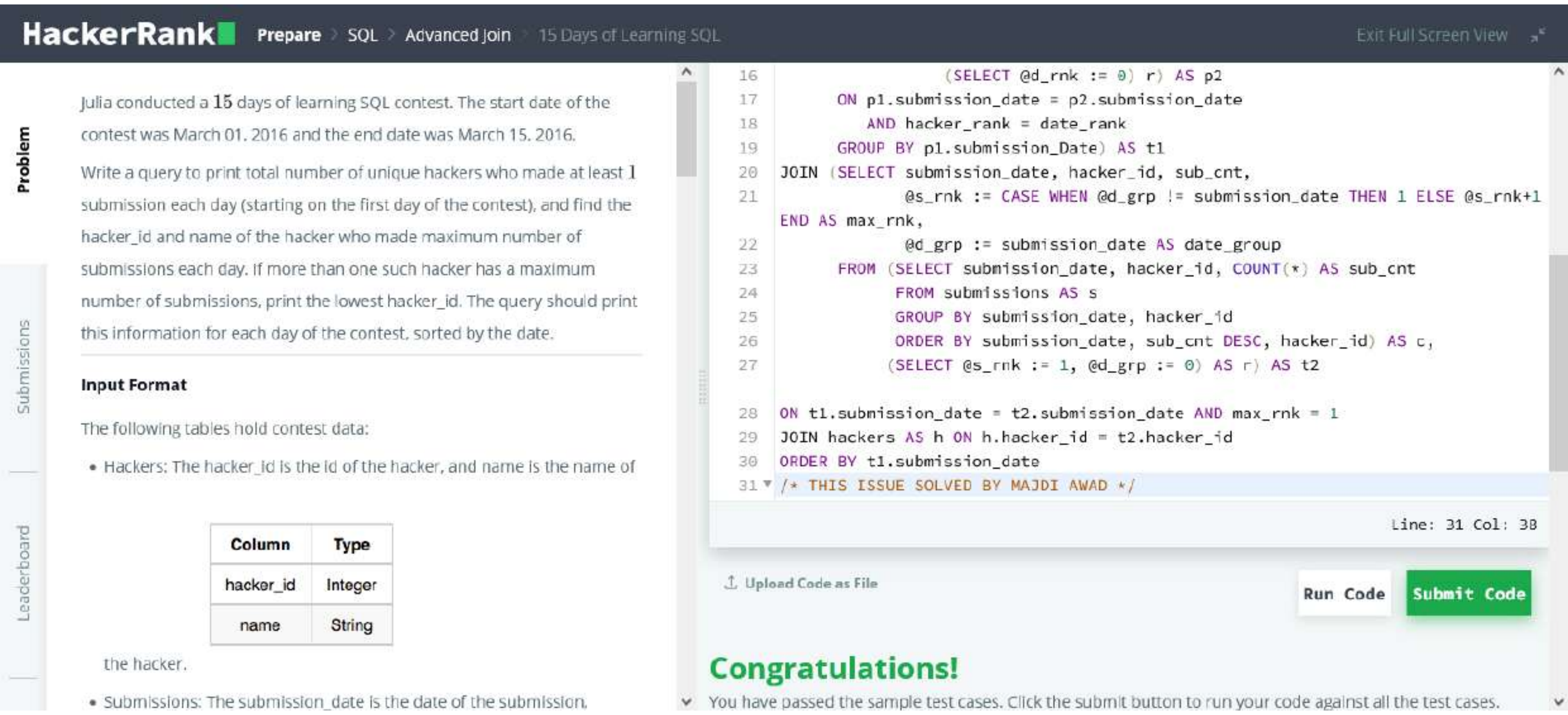
X	Y
20	20
20	20

```sql
SELECT f1.X, f1.Y FROM Functions AS f1
WHERE f1.X = f1.Y AND
(SELECT COUNT(*) FROM Functions WHERE X = f1.X AND Y = f1.Y) > 1
UNION
SELECT f1.X, f1.Y from Functions AS f1
WHERE EXISTS(SELECT X, Y FROM Functions WHERE f1.X = Y AND f1.Y = X AND f1.X < X)
ORDER BY X

/* THIS ISSUE SOLVED BY MAJDI AWAD */
```

Line: 9 Col: 38

Congratulations!

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



Julia conducted a 15 days of learning SQL contest. The start date of the contest was March 01, 2016 and the end date was March 15, 2016.

Write a query to print total number of unique hackers who made at least 1 submission each day (starting on the first day of the contest), and find the hacker_id and name of the hacker who made maximum number of submissions each day. If more than one such hacker has a maximum number of submissions, print the lowest hacker_id. The query should print this information for each day of the contest, sorted by the date.

Input Format

The following tables hold contest data:

- Hackers: The hacker_id is the id of the hacker, and name is the name of the hacker.

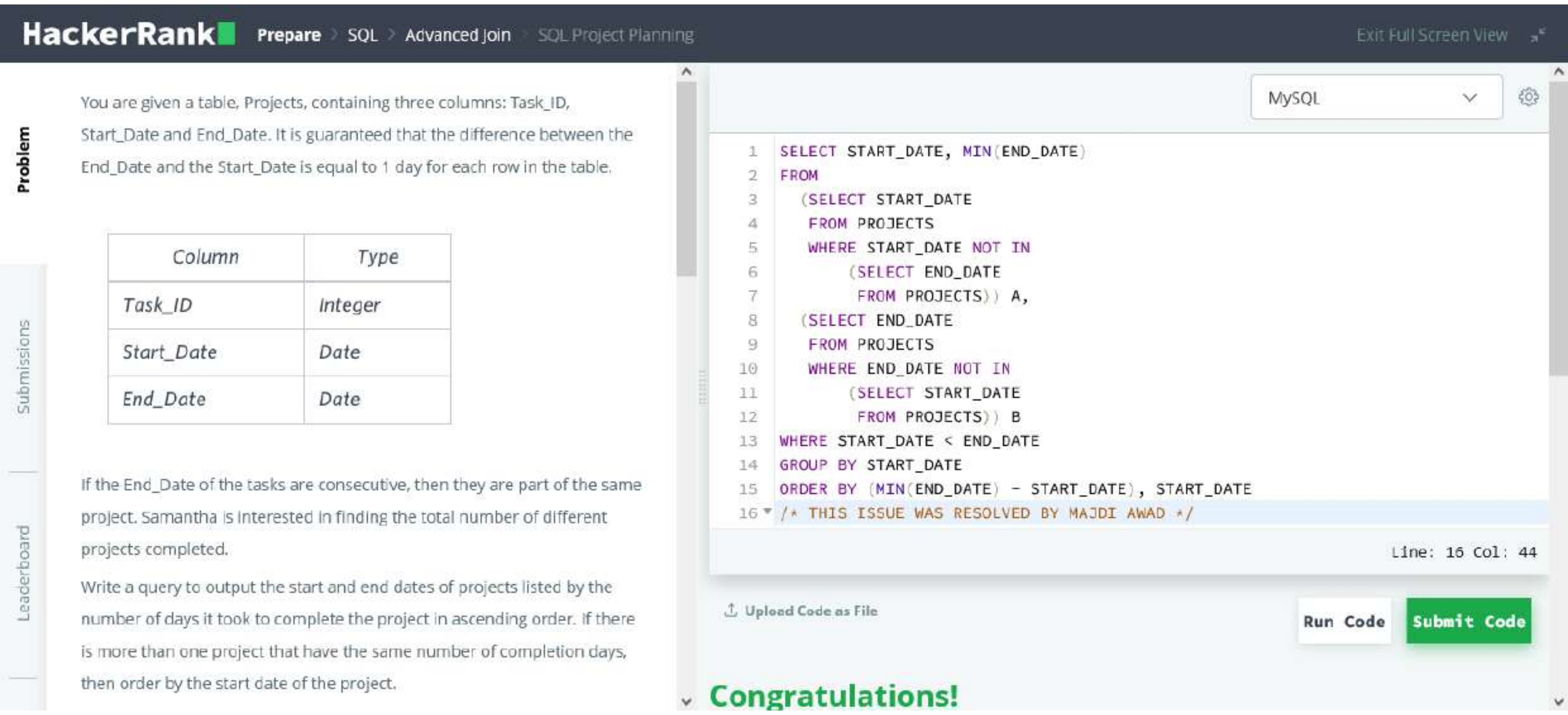
Column	Type
hacker_id	Integer
name	String

- Submissions: The submission_date is the date of the submission,

```sql
                    (SELECT @d_rnk := 0) r) AS p2
        ON p1.submission_date = p2.submission_date
           AND hacker_rank = date_rank
        GROUP BY p1.submission_Date) AS t1
JOIN (SELECT submission_date, hacker_id, sub_cnt,
                @s_rnk := CASE WHEN @d_grp != submission_date THEN 1 ELSE @s_rnk+1 END AS max_rnk,
                @d_grp := submission_date AS date_group
        FROM (SELECT submission_date, hacker_id, COUNT(*) AS sub_cnt
              FROM submissions AS s
              GROUP BY submission_date, hacker_id
              ORDER BY submission_date, sub_cnt DESC, hacker_id) AS c,
             (SELECT @s_rnk := 1, @d_grp := 0) AS r) AS t2

ON t1.submission_date = t2.submission_date AND max_rnk = 1
JOIN hackers AS h ON h.hacker_id = t2.hacker_id
ORDER BY t1.submission_date
/* THIS ISSUE SOLVED BY MAJDI AWAD */
```

Congratulations!

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



You are given a table, Projects, containing three columns: Task_ID, Start_Date and End_Date. It is guaranteed that the difference between the End_Date and the Start_Date is equal to 1 day for each row in the table.

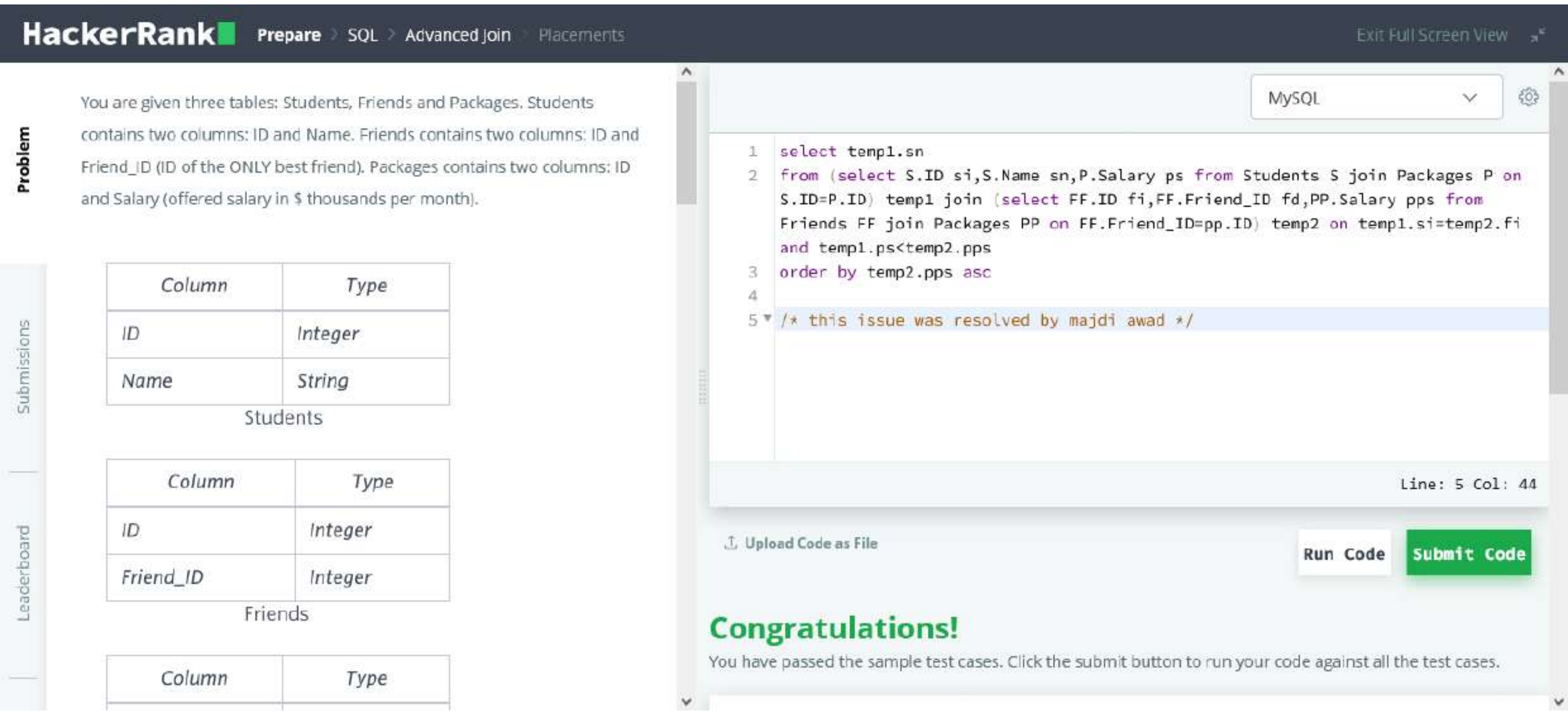
Column	*Type*
Task_ID	*Integer*
Start_Date	*Date*
End_Date	*Date*

If the End_Date of the tasks are consecutive, then they are part of the same project. Samantha is interested in finding the total number of different projects completed.

Write a query to output the start and end dates of projects listed by the number of days it took to complete the project in ascending order. If there is more than one project that have the same number of completion days, then order by the start date of the project.

```
SELECT START_DATE, MIN(END_DATE)
FROM
  (SELECT START_DATE
   FROM PROJECTS
   WHERE START_DATE NOT IN
        (SELECT END_DATE
         FROM PROJECTS)) A,
  (SELECT END_DATE
   FROM PROJECTS
   WHERE END_DATE NOT IN
        (SELECT START_DATE
         FROM PROJECTS)) B
WHERE START_DATE < END_DATE
GROUP BY START_DATE
ORDER BY (MIN(END_DATE) - START_DATE), START_DATE
/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

Congratulations!



You are given three tables: Students, Friends and Packages. Students contains two columns: ID and Name. Friends contains two columns: ID and Friend_ID (ID of the ONLY best friend). Packages contains two columns: ID and Salary (offered salary in $ thousands per month).

Column	Type
ID	Integer
Name	String

Students

Column	Type
ID	Integer
Friend_ID	Integer

Friends

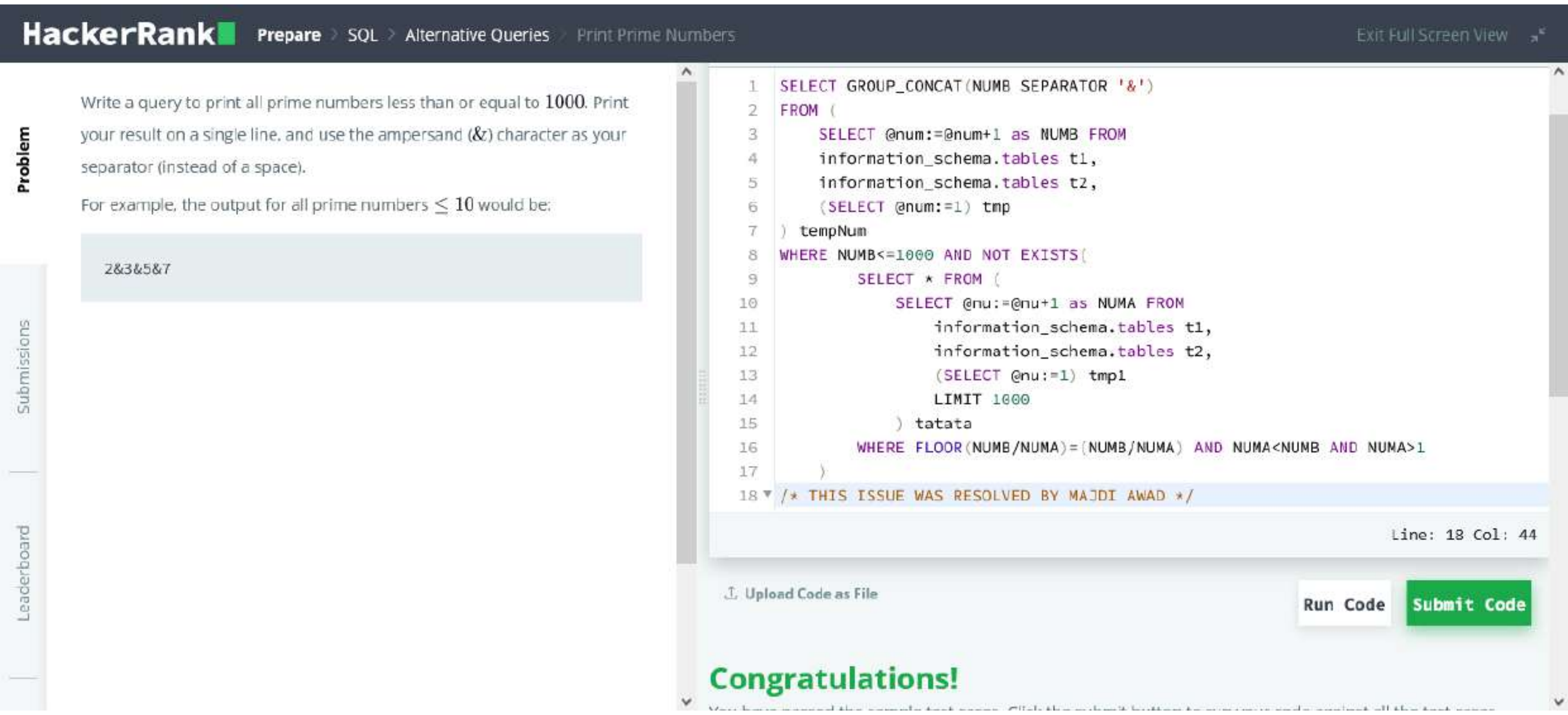
Column	Type

```sql
select temp1.sn
from (select S.ID si,S.Name sn,P.Salary ps from Students S join Packages P on S.ID=P.ID) temp1 join (select FF.ID fi,FF.Friend_ID fd,PP.Salary pps from Friends FF join Packages PP on FF.Friend_ID=pp.ID) temp2 on temp1.si=temp2.fi and temp1.ps<temp2.pps
order by temp2.pps asc

/* this issue was resolved by majdi awad */
```

Line: 5 Col: 44

Congratulations!

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



Write a query to print all prime numbers less than or equal to 1000. Print your result on a single line, and use the ampersand ($\&$) character as your separator (instead of a space).

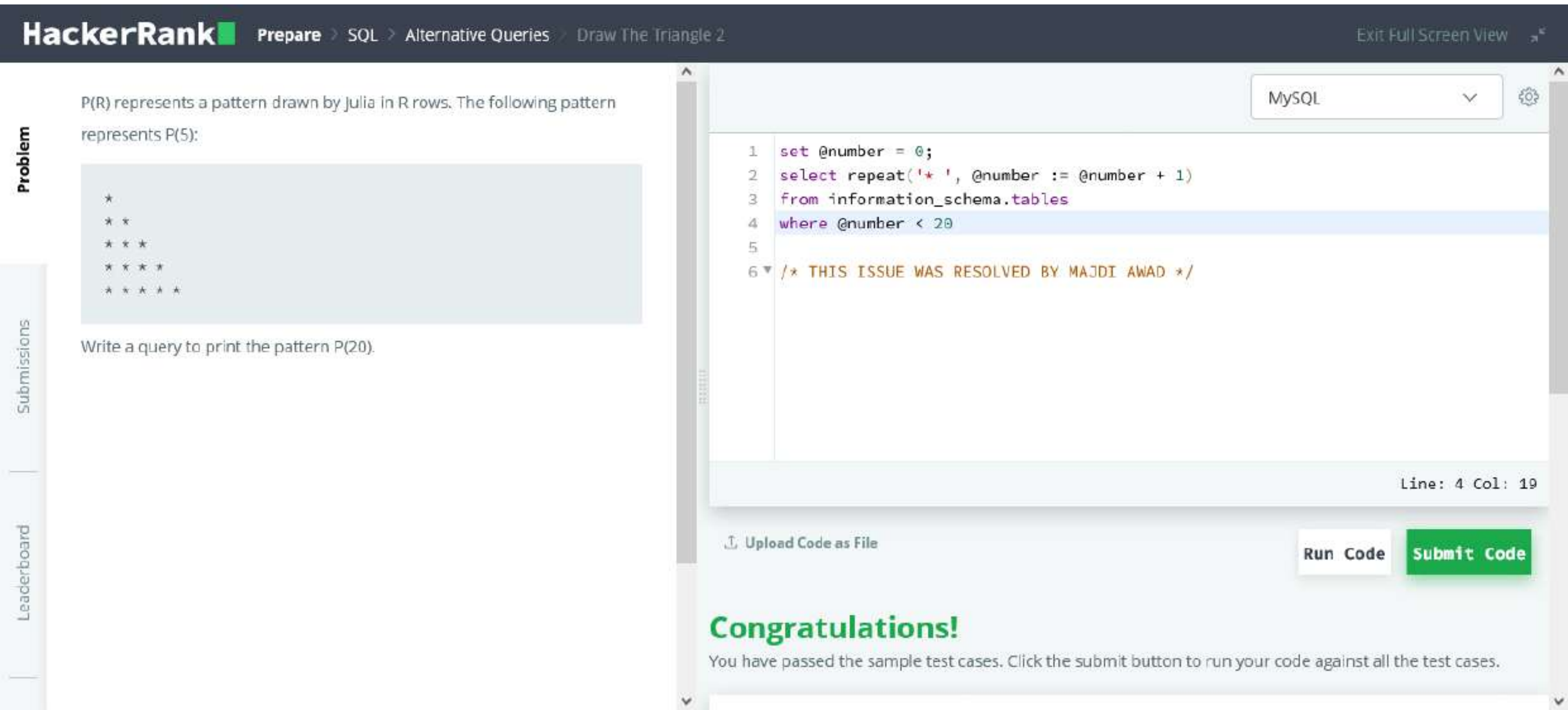
For example, the output for all prime numbers ≤ 10 would be:

```
2&3&5&7
```

```sql
SELECT GROUP_CONCAT(NUMB SEPARATOR '&')
FROM (
    SELECT @num:=@num+1 as NUMB FROM
    information_schema.tables t1,
    information_schema.tables t2,
    (SELECT @num:=1) tmp
) tempNum
WHERE NUMB<=1000 AND NOT EXISTS(
        SELECT * FROM (
            SELECT @nu:=@nu+1 as NUMA FROM
                information_schema.tables t1,
                information_schema.tables t2,
                (SELECT @nu:=1) tmp1
                LIMIT 1000
            ) tatata
        WHERE FLOOR(NUMB/NUMA)=(NUMB/NUMA) AND NUMA<NUMB AND NUMA>1
    )
/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

Congratulations!

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



P(R) represents a pattern drawn by Julia in R rows. The following pattern represents P(5):

```
*
* *
* * *
* * * *
* * * * *
```

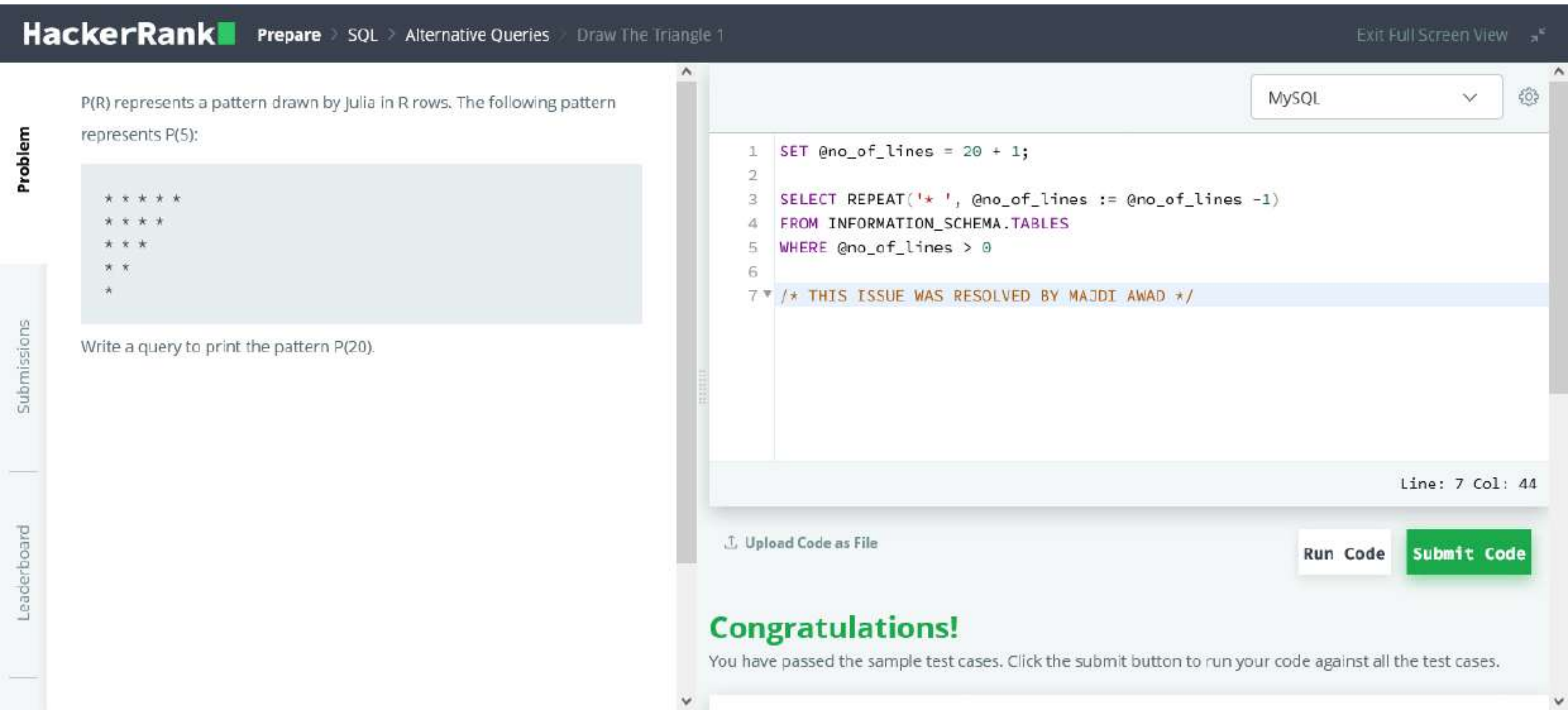
Write a query to print the pattern P(20).

```
set @number = 0;
select repeat('* ', @number := @number + 1)
from information_schema.tables
where @number < 20

/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

Line: 4 Col: 19

Congratulations!

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



P(R) represents a pattern drawn by Julia in R rows. The following pattern represents P(5):

```
* * * * *
* * * *
* * *
* *
*
```

Write a query to print the pattern P(20).

```sql
SET @no_of_lines = 20 + 1;

SELECT REPEAT('* ', @no_of_lines := @no_of_lines -1)
FROM INFORMATION_SCHEMA.TABLES
WHERE @no_of_lines > 0

/* THIS ISSUE WAS RESOLVED BY MAJDI AWAD */
```

Congratulations!

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