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

Solution-select avg(population) from city where district='California';Graphical user interface, text, application

Description automatically generated

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

Solution-select round(avg(population)) from city;

Graphical user interface, application

Description automatically generated

3) Sol - select a.hacker\_id,a.name,count(b.hacker\_id)

from Hackers a, Challenges b

WHERE a.hacker\_id = b.hacker\_id

GROUP BY a.hacker\_id,a.name

HAVING count(b.hacker\_id) not in (select distinct count(hacker\_id) from Challenges

WHERE hacker\_id != a.hacker\_id

group by hacker\_id

having count(hacker\_id) < (select max(x.challenge\_count)

from (select count(b.challenge\_id) as challenge\_count from Challenges b GROUP BY b.hacker\_id) as x ))

ORDER BY count(b.hacker\_id) desc, a.hacker\_id Graphical user interface, application

Description automatically generated

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

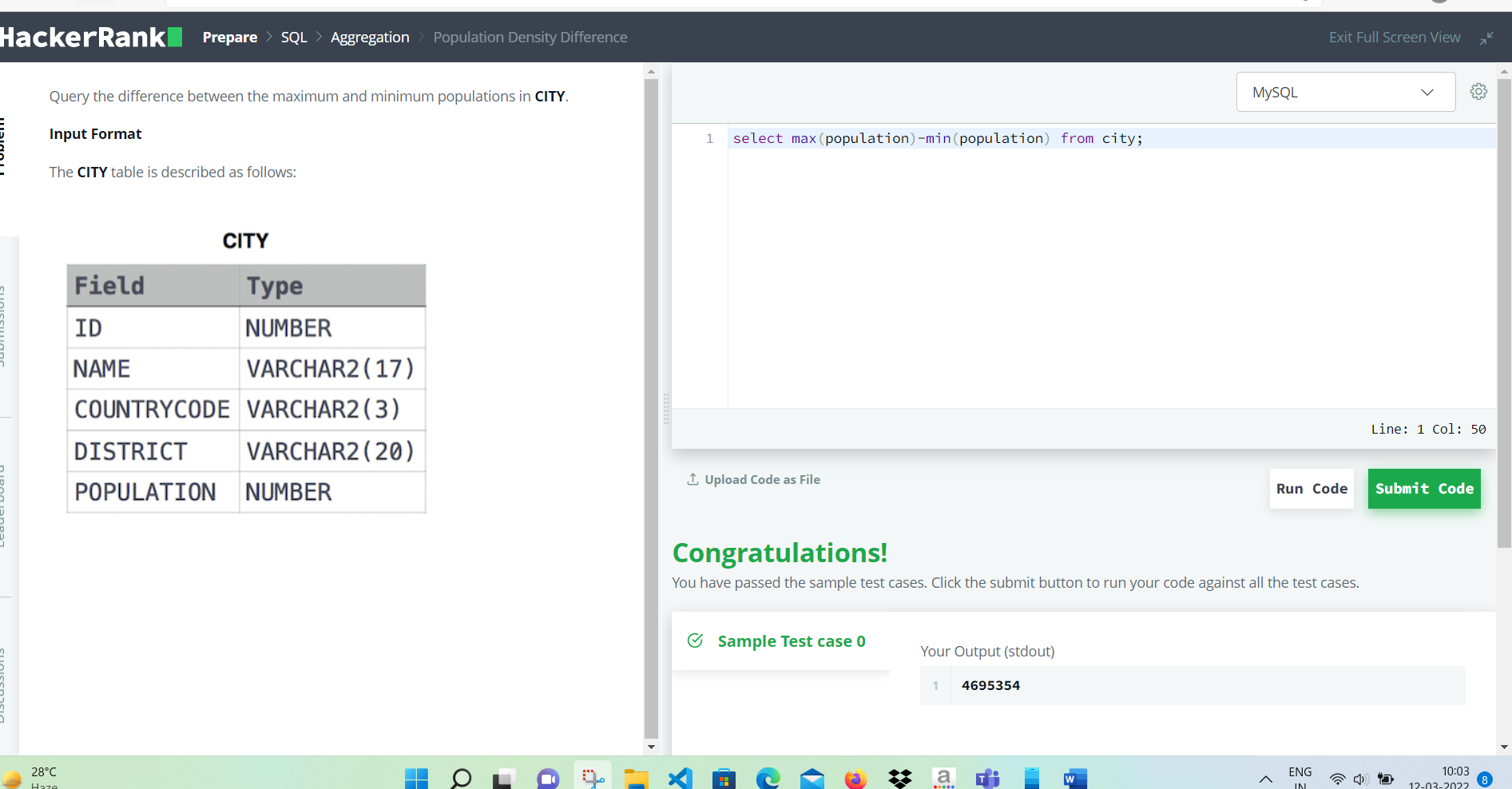
Solution-select sum(population) from city where countrycode='JPN';

Graphical user interface, text, application

Description automatically generated

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

Solution-select max(population)-min(population) from city;



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

Solution-SELECT round((AVG(Salary)-AVG(REPLACE(Salary,'0','')))+1 )FROM EMPLOYEES;Graphical user interface, text, application

Description automatically generated