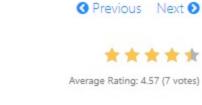
■ Articles > 612. Shortest Distance in a Plane ▼

612. Shortest Distance in a Plane

June 22, 2017 | 16K views



6 9 6

Table point_2d holds the coordinates (x,y) of some unique points (more than two) in a plane.

Write a query to find the shortest distance between these points rounded to 2 decimals.

```
| x | y |
|----|----|
| -1 | -1 |
| 0 | 0 |
| -1 | -2 |
```

shortest

The shortest distance is 1.00 from point (-1,-1) to (-1,2). So the output should be:

```
Note: The longest distance among all the points are less than 10000.
```

Approach 1: Using SQRT, POW() functions and math knowledge [Accepted] Intuition

Solution

Calculate the distances between each two points and then display the smallest one.

Algorithm

The euclidean distance between two points P1(x1,y1) and P2(x2, y2) in two dimensions is defined as

 $\sqrt{(x1-x2)^2+(y1-y2)^2}$. So in order to get the distances, we can join this table with itself, and then utilize the built-in function POW() and SQRT() like below.

dille the i

SELECT

p1.x, p1.y, p2.x, p2.y,

```
SQRT((POW(p1.x - p2.x, 2) + POW(p1.y - p2.y, 2))) AS distance

FROM

point_2d p1

JOIN

point_2d p2 ON p1.x != p2.x OR p1.y != p2.y

Note: - The condition 'p1.x != p2.x OR p2.y != p2.y' is to avoid calculating the distance of a point with itself. Otherwise, the minimum distance will be always zero. - The columns p1.x, p1.y, p2.x and p2.y are for demonstrating. They are not necessary for the final solution.
```

| -1 | -1 | -2 | 1 | 0 | 0 | -1 | -2 | 2.23606797749979 |

```
FROM

point_2d p1

JOIN

point_2d p2 ON p1.x != p2.x OR p1.y != p2.y
```

At last, choose the minimum distance and round it to 2 decimals as required.

So the output would be as below after running this code on the sample data.

x y x y distance

| -1 | -2 | -1 | -1 | 1

0 0 -1 -1 1.4142135623730951

| -1 | -1 | 0 | 0 | 1.4142135623730951 | -1 | -2 | 0 | 0 | 2.23606797749979

```
Note: To put the MIN() inside of SQRT() will slightly improve the performance.

Approach 2: Optimize to avoid reduplicate calculations [Accepted]

Intuition
```

It is unnecessary to calculate the distance between all points to all other points since some of them may

point in a certain rule such ponts with bigger x value. By following this rule, we can avoid quite a lot of

ROUND(SQRT(MIN((POW(p1.x - p2.x, 2) + POW(p1.y - p2.y, 2)))), 2) AS shortest

Algorithm

When join the table with itself, we can claim to only calculate the distance between one point to another

already be done. So how to avoid the reduplicate calculations?

point_2d t2 **ON** (t1.x <= t2.x **AND** t1.y < t2.y)

-2) and (-1, -1) appearing twice in the first and last line in the output.

Here comes the solution to select the shortest distance and round to two decimals.

point_2d p2 ON (p1.x <= p2.x AND p1.y < p2.y)

OR (p1.x <= p2.x AND p1.y > p2.y)
OR (p1.x < p2.x AND p1.y = p2.y)

Mr-Bin # 123 @ January 5, 2019 1:04 AM

from point_2d p1, point_2d p2

5 A V & Share Share

OR (t1.x <= t2.x AND t1.y > t2.y)
OR (t1.x < t2.x AND t1.y = t2.y)

SELECT

reduplicate calculations.

t1.x, t1.y, t2.x, t2.y,

MySQL

SELECT

SQRT((POW(t1.x - t2.x, 2) + POW(t1.y - t2.y, 2))) AS distance

FROM

point_2d t1

JOIN

SELECT ROUND(SQRT(MIN((POW(p1.x - p2.x, 2) + POW(p1.y - p2.y, 2)))),2) AS shortest FROM

```
Rate this article: * * * * *
```

O Previous

;

MySQL

point_2d p1
JOIN

```
Type comment here... (Markdown is supported)

Preview

Post

tyumneva ★ 83 ② July 17, 2018 12:12 AM
select min(round(sqrt(power(t1.x-t2.x,2) + power(t1.y-t2.y,2)),2)) shortest from point_2d t1
cross join point_2d t2
on !(t1.x = t2.x and t1.y=t2.y)

13 ↑ ✔ ☑ Share ♠ Reply
```

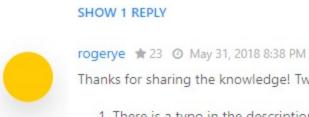
select round(min(sqrt(power(p1.x - p2.x, 2) + power(p1.y - p2.y, 2))), 2) as shor

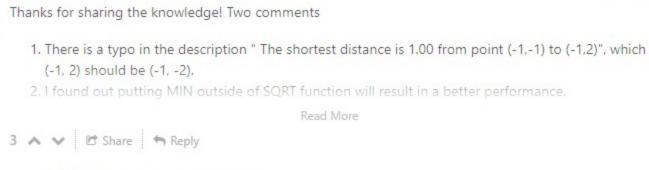
Next **1**

A Report

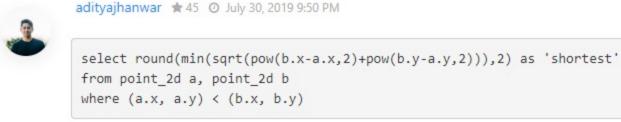
A Report

A Report





Read More



aaronmok * 0 O March 29, 2019 10:21 PM

noint id n1.

vasanthtillu91 🛊 5 🗿 June 1, 2018 9:32 AM

as A from point_2d p1 join point_2d p2

on p1.x!=p2.x or p1.y!=p2.y

1 A V C Share Reply

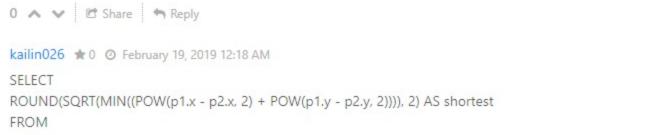
0 A V C Share Reply

SELECT

FROM

point_2d p1

select sqrt((select (power((p1.x-p2.x),2)+ power((p1.y-p2.y),2))



Read More

Read More

power(p2.x-p1.x, 2))), 2) shortest

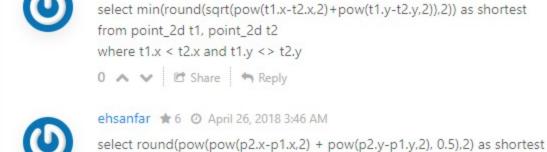
MIN(ROUND(SQRT(POWER(p1.x-p2.x,2)-POWER(p1.y-p2.y,2))),2)) shortest

Read More

©	SHOW 1 REPLY
	yuhui4 ★ 35 ② January 12, 2019 7:15 AM select round(min(sqrt(power(p2.y-p1.y, 2) + from point_2d p1, point_2d p2 where not (p1.x = p2.x and p1.y = p2.y);

ztj 🛊 15 🗿 August 4, 2018 2:36 AM

from point_2d as p1, point_2d as p2 where p1.x!= p2.x or p1.y!= p2.y



order by shortest