

LijianChen★ 53Last Edit: September 25, 2019 10:29 AM2.3K VIEWS

- 23
- The steps:
- (1) Get cumulative sum weight using `Join` with condition `q1.turn >= q2.turn` and `Group By q1.turn`
- (2) Filter the groups with cum sum `<=1000`
- (3) Order by cum sum with `Desc` order, select the 1st.

```
SELECT q1.person_name
FROM Queue q1 JOIN Queue q2 ON q1.turn >= q2.turn
GROUP BY q1.turn
HAVING SUM(q2.weight) <= 1000
ORDER BY SUM(q2.weight) DESC
LIMIT 1
```

Type comment here... (Markdown is supported)

Post

zbian1018★ 3October 22, 2019 7:50 AM

Can anyone explain what `q1.turn >= q2.turn` mean? Thanks

3

Show 2 replies

Reply

LVL99★ 148Last Edit: September 25, 2019 3:35 PM

same idea:

```
SELECT q1.person_name
FROM Queue q1, Queue q2
WHERE q1.turn >= q2.turn
GROUP BY q1.turn
HAVING SUM(q2.weight) <= 1000
ORDER BY SUM(q2.weight) DESC
LIMIT 1
```

3

Reply

dliu28★ 1March 15, 2020 11:05 PM

厉害厉害！

1

Show 1 reply

Reply

bhargavg★ 2November 18, 2019 8:09 AM

Same idea but different ORDER BY.

We can use `q1.turn`, instead of summing the weights again, which improved time by a little

```
# Write your MySQL query statement below
SELECT q1.person_name
FROM Queue q1 JOIN Queue q2 ON q1.turn >= q2.turn
GROUP BY q1.turn
having SUM(q2.weight) <=1000
order by q1.turn desc
limit 1
```

Read More

0

Reply

trilobite2019★ 12November 6, 2019 9:10 AM

Great!

0

Reply

Merciless★ 489September 25, 2019 7:57 PM

Is the time complexity of this solution $O(n^2)$?

0

Show 3 replies

Reply