Tutorial: Window Function

> Original designed by ZHU Yueming in 2022. Apr. 11th

Experimental objective

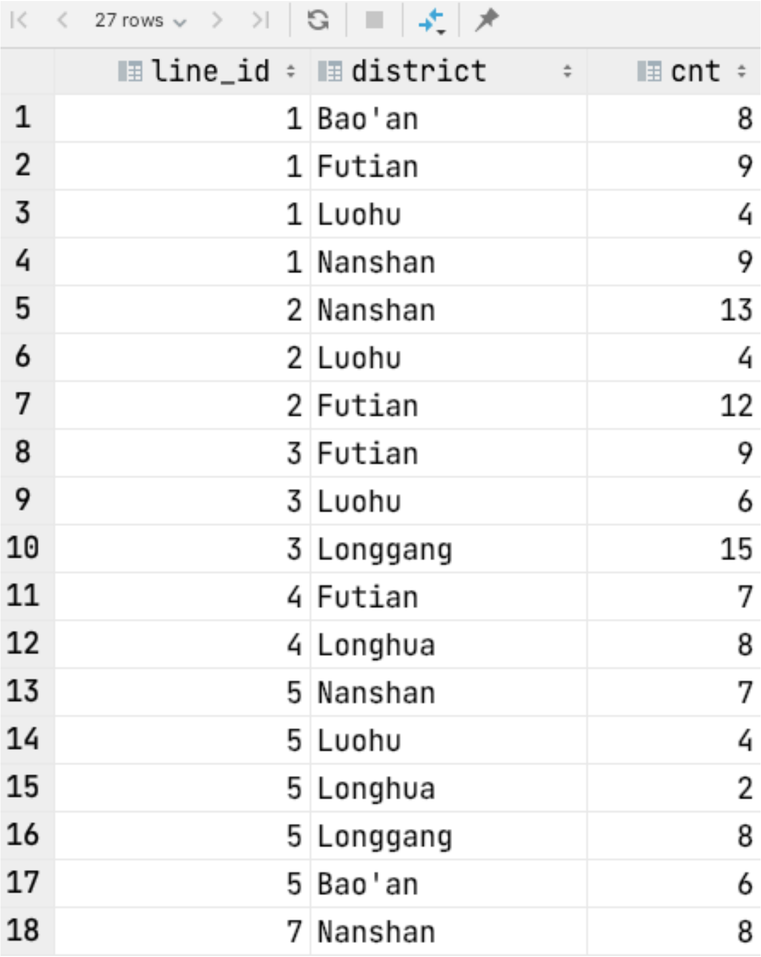
- Understand basic concept of window function. Keywords like over(), partition by
- Understand how to use rank(), dense_rank(), row_number(), lag()

Part 1. Basic use of window function

The query below descripts a requirement that how many subway stations in each line and district?

```
select line_id, district, count(*) cnt
         from line_detail l
                  join stations s on l.station_id = s.station_id
where district <> ''
         group by line_id, district
         order by line_id;
```

The result set would be:



	line_id	district	cnt
1	1	Bao'an	8
2	1	Futian	9
3	1	Luohu	4
4	1	Nanshan	9
5	2	Nanshan	13
6	2	Luohu	4
7	2	Futian	12
8	3	Futian	9
9	3	Luohu	6
10	3	Longgang	15
11	4	Futian	7
12	4	Longhua	8
13	5	Nanshan	7
14	5	Luohu	4
15	5	Longhua	2
16	5	Longgang	8
17	5	Bao'an	6
18	7	Nanshan	8

Based on the result set above, we have another requirement that calculate the percentage of the cnt in each row to the total number of the stations. To complete the requirements, we need a total number of stations in each row, and in this case window functions would be working.

1. Over()

将聚合函数放在每一行里

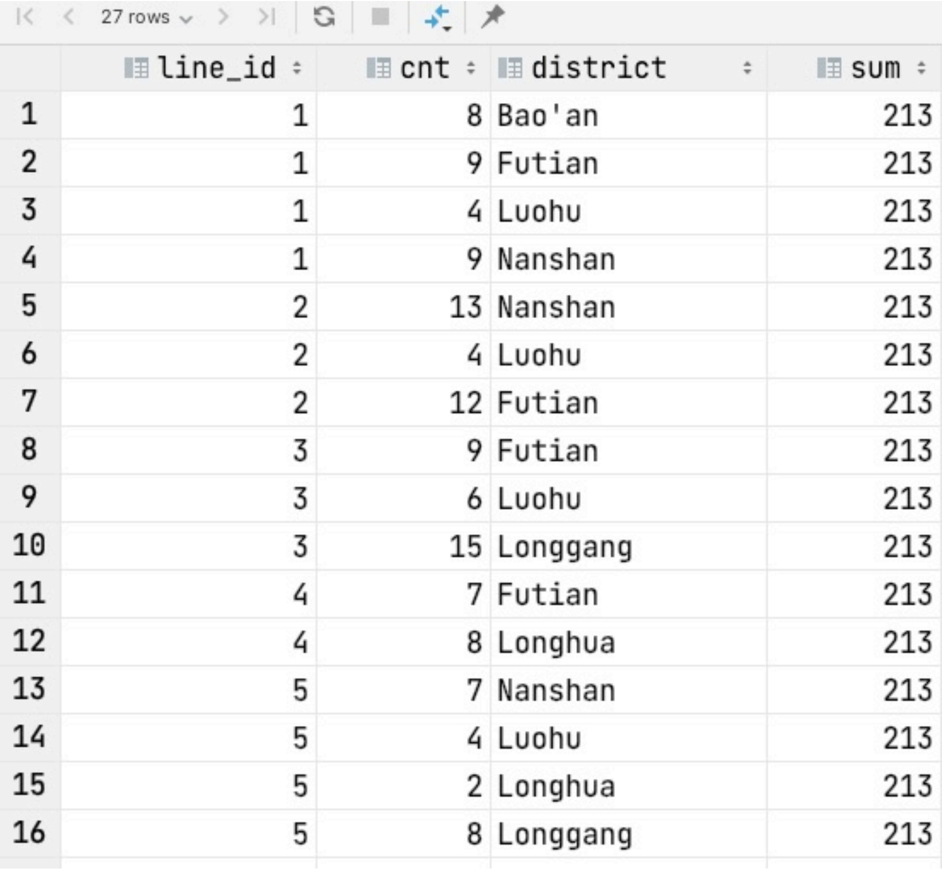
How many stations in each line and in each district? Return them in one row with total number of stations.

Return the aggregate result in each row.

For example:

```
select line_id, cnt, district, sum(cnt) over () as sum
from (
         select line_id, district, count(*) cnt
         from line_detail l
                  join stations s on l.station_id = s.station_id
         where district <> ''
         group by line_id, district
         order by line_id) sub;
```

Result:

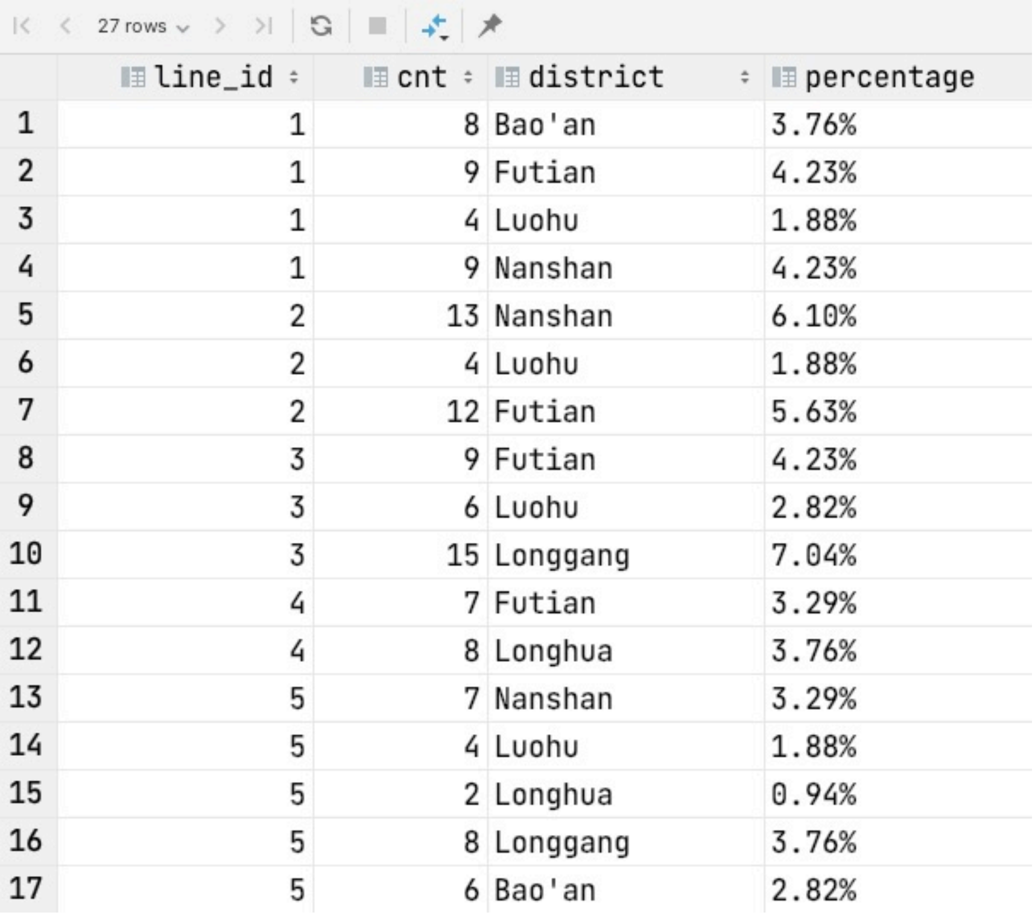


	line_id	cnt	district	sum
1	1	8	Bao'an	213
2	1	9	Futian	213
3	1	4	Luohu	213
4	1	9	Nanshan	213
5	2	13	Nanshan	213
6	2	4	Luohu	213
7	2	12	Futian	213
8	3	9	Futian	213
9	3	6	Luohu	213
10	3	15	Longgang	213
11	4	7	Futian	213
12	4	8	Longhua	213
13	5	7	Nanshan	213
14	5	4	Luohu	213
15	5	2	Longhua	213
16	5	8	Longgang	213

After that, if we want **display the percentage**, we need wrapper another select query.

```
select line_id, cnt, district, round(cnt * 100.0 / sum,2) ||'%' percentage
from (
         select line_id, cnt, district, sum(cnt) over () as sum
         from (
                  select line_id, district, count(*) cnt
                  from line_detail l
                           join stations s on l.station_id = s.station_id
                  where district <> ''
                  group by line_id, district
                  order by line_id) sub) sub2;
```

Result:



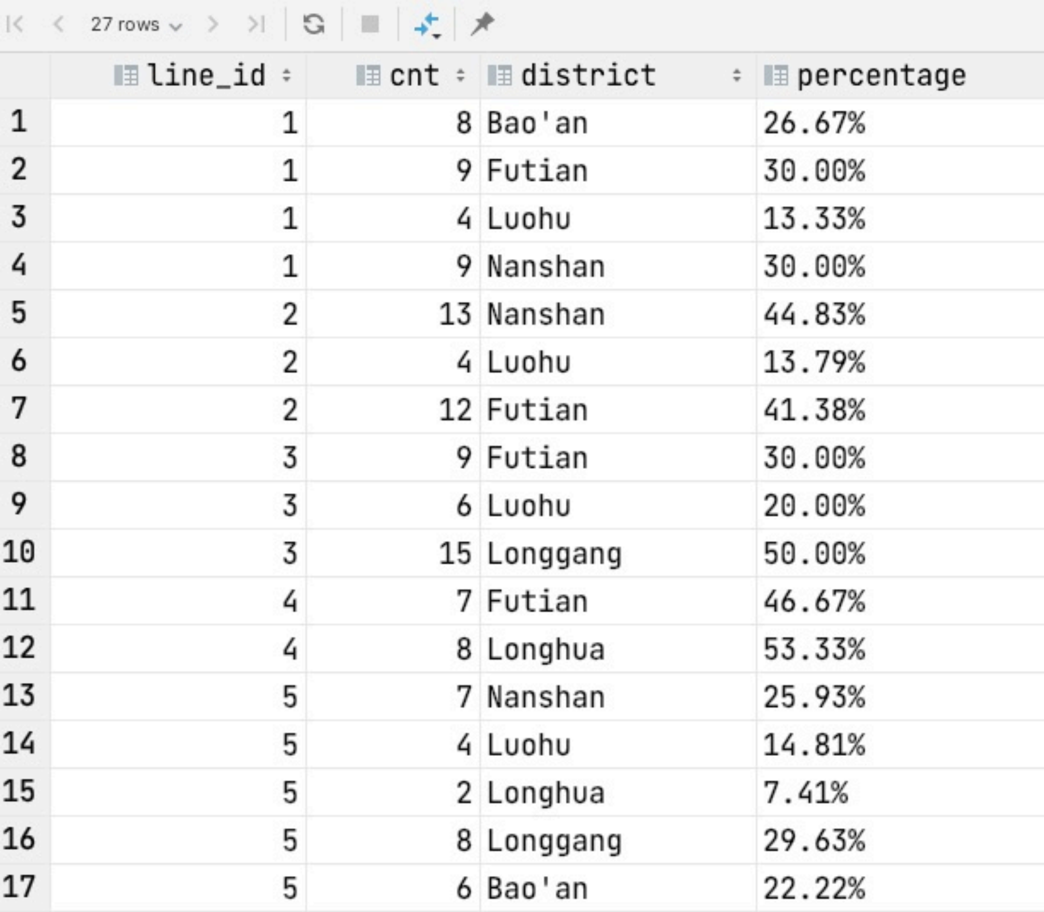
	line_id	cnt	district	percentage
1	1	8	Bao'an	3.76%
2	1	9	Futian	4.23%
3	1	4	Luohu	1.88%
4	1	9	Nanshan	4.23%
5	2	13	Nanshan	6.10%
6	2	4	Luohu	1.88%
7	2	12	Futian	5.63%
8	3	9	Futian	4.23%
9	3	6	Luohu	2.82%
10	3	15	Longgang	7.04%
11	4	7	Futian	3.29%
12	4	8	Longhua	3.76%
13	5	7	Nanshan	3.29%
14	5	4	Luohu	1.88%
15	5	2	Longhua	0.94%
16	5	8	Longgang	3.76%
17	5	6	Bao'an	2.82%

2. Partition by

Based on the result of pervious part, if the percentage we want is **the total number of stations in current line and in current district but not the total number of the whole stations**, we need add condition in `over()`

```
select line_id, cnt, district, round(cnt * 100.0 / sum,2) ||'%' percentage
from (
        select line_id, cnt, district, sum(cnt) over (partition by line_id) as sum
        from (
                select line_id, district, count(*) cnt
                from line_detail l
                        join stations s on l.station_id = s.station_id
                where district <> ''
                group by line_id, district
                order by line_id) sub) sub2;
```

Result:



	line_id	cnt	district	percentage
1	1	8	Bao'an	26.67%
2	1	9	Futian	30.00%
3	1	4	Luohu	13.33%
4	1	9	Nanshan	30.00%
5	2	13	Nanshan	44.83%
6	2	4	Luohu	13.79%
7	2	12	Futian	41.38%
8	3	9	Futian	30.00%
9	3	6	Luohu	20.00%
10	3	15	Longgang	50.00%
11	4	7	Futian	46.67%
12	4	8	Longhua	53.33%
13	5	7	Nanshan	25.93%
14	5	4	Luohu	14.81%
15	5	2	Longhua	7.41%
16	5	8	Longgang	29.63%
17	5	6	Bao'an	22.22%

3. Rank(), dense_rank(), row_number()

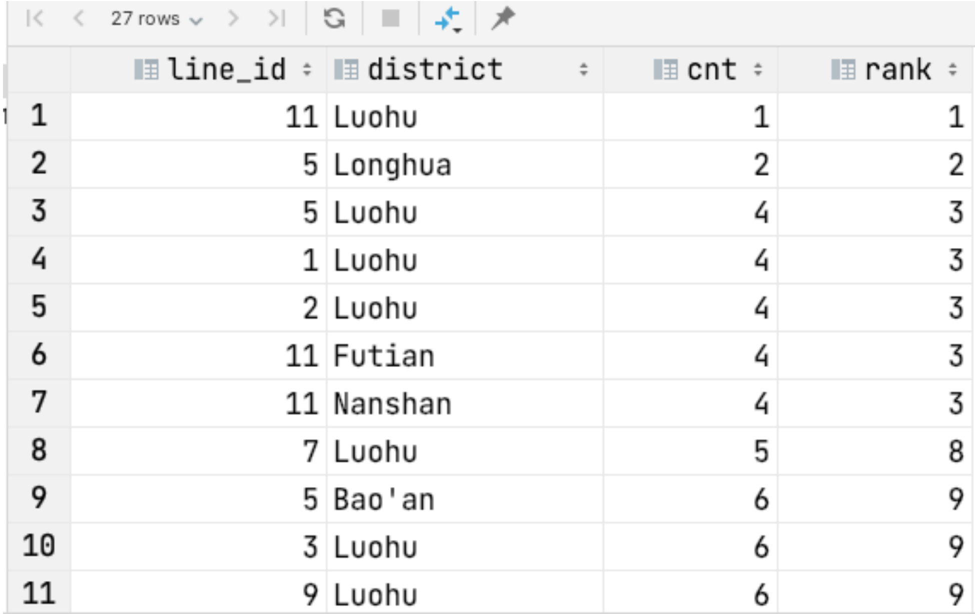
The following example can help to distinguish the usage of `rank()`, `dense_rank()` and `row_number()`

3.1 Rank()

Given a rank number of the ascending order about the number of stations in each line and in each district.

```
select *, rank() over(order by cnt)
from (
        select line_id, district, count(*) cnt
        from line_detail l
                join stations s on l.station_id = s.station_id
        where district <> ''
        group by line_id, district
        order by line_id) sub;
```

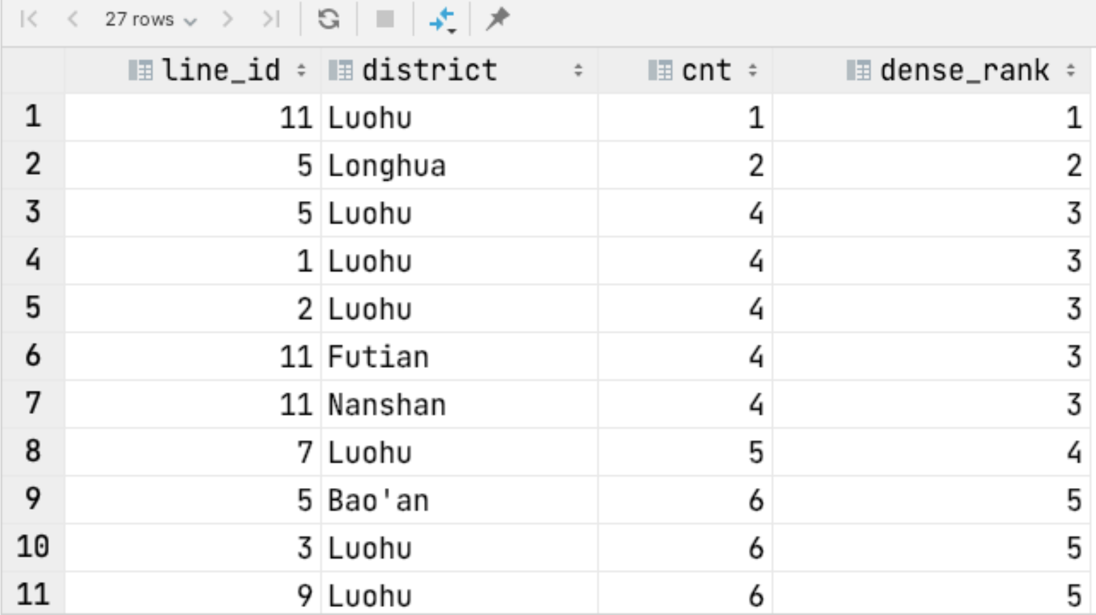
Result:



	line_id	district	cnt	rank
1	11	Luohu	1	1
2	5	Longhua	2	2
3	5	Luohu	4	3
4	1	Luohu	4	3
5	2	Luohu	4	3
6	11	Futian	4	3
7	11	Nanshan	4	3
8	7	Luohu	5	8
9	5	Bao'an	6	9
10	3	Luohu	6	9
11	9	Luohu	6	9

3.2 Dense_rank()

```
select *, dense_rank() over(order by cnt)
from (
         select line_id, district, count(*) cnt
         from line_detail l
                  join stations s on l.station_id = s.station_id
         where district <> ''
         group by line_id, district
         order by line_id) sub;
```

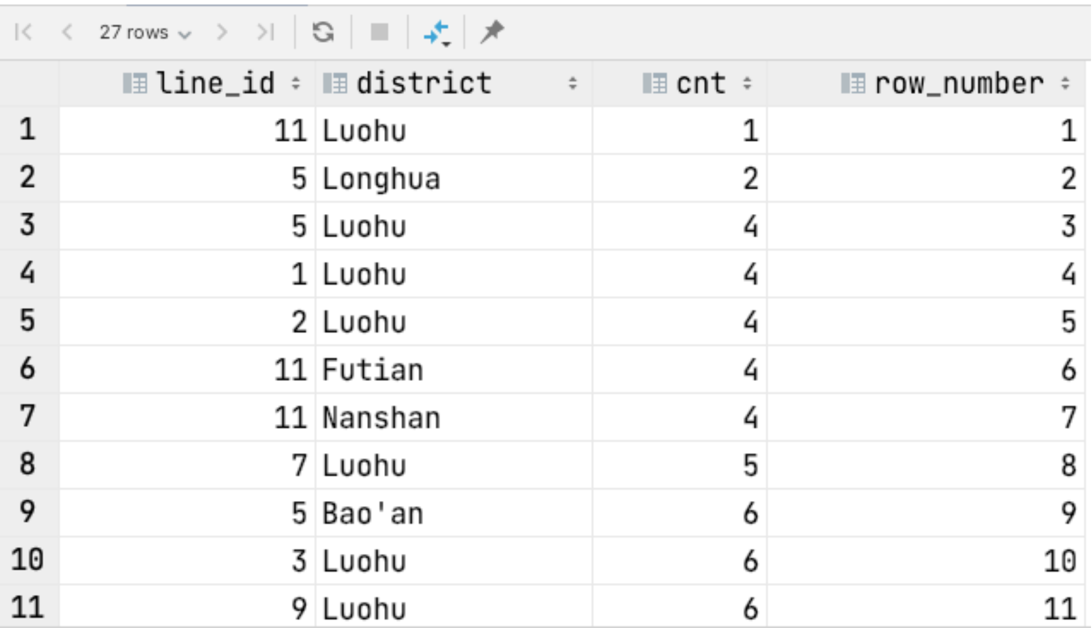


	line_id	district	cnt	dense_rank
1	11	Luohu	1	1
2	5	Longhua	2	2
3	5	Luohu	4	3
4	1	Luohu	4	3
5	2	Luohu	4	3
6	11	Futian	4	3
7	11	Nanshan	4	3
8	7	Luohu	5	4
9	5	Bao'an	6	5
10	3	Luohu	6	5
11	9	Luohu	6	5

3.3 Row_number()

```
select *, row_number()  over(order by cnt)
from (
         select line_id, district, count(*) cnt
         from line_detail l
                  join stations s on l.station_id = s.station_id
         where district <> ''
         group by line_id, district
         order by line_id) sub;
```

Result:



	line_id	district	cnt	row_number
1	11	Luohu	1	1
2	5	Longhua	2	2
3	5	Luohu	4	3
4	1	Luohu	4	4
5	2	Luohu	4	5
6	11	Futian	4	6
7	11	Nanshan	4	7
8	7	Luohu	5	8
9	5	Bao'an	6	9
10	3	Luohu	6	10
11	9	Luohu	6	11

4. Lag()

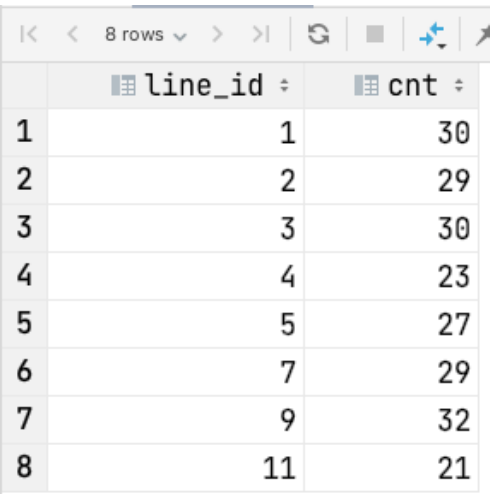
It can return the value of the nth previous line.

```
lag(column_name, previous_n_row)
```

Original query about the count of stations in each line :

```
select line_id, count(*) cnt
         from line_detail
         group by line_id
         order by line_id
```

Result:

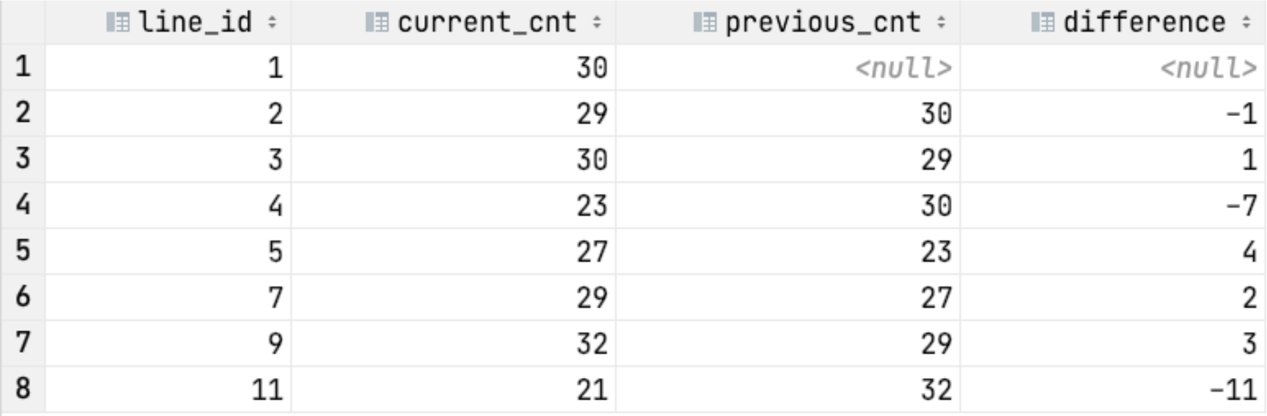


	line_id	cnt
1	1	30
2	2	29
3	3	30
4	4	23
5	5	27
6	7	29
7	9	32
8	11	21

Example requirement: **find the difference of the count of stations in current line and the count of stations in previous line**.

```
select *, current_cnt - previous_cnt as difference
from (
         select *, lag(current_cnt, 1) over () as previous_cnt
         from (
                  select line_id, count(*) current_cnt
                  from line_detail
                  group by line_id
                  order by line_id) sub) sub2;
```

Result:



	line_id	current_cnt	previous_cnt	difference
1	1	30	<null>	<null>
2	2	29	30	-1
3	3	30	29	1
4	4	23	30	-7
5	5	27	23	4
6	7	29	27	2
7	9	32	29	3
8	11	21	32	-11