**6. Checking the top-grossing search terms**

with lagging\_rank as

(

select

search\_term,

month,

search\_frequency\_rank,

lag(search\_frequency\_rank, 1) over (partition by search\_term order by search\_term, month) as Pre\_value,

from analysttest-project.analysttest\_dataset.dw\_fact\_amz\_searchterms

where month in ('2021-11-01', '2021-10-01', '2021-09-01')

group by search\_term, month, search\_frequency\_rank

order by search\_term, month

),

rank\_diff as

(

select

search\_term,

month,

search\_frequency\_rank,

Pre\_value,

cast(Pre\_value as numeric) - cast(search\_frequency\_rank as numeric) as search\_rank\_diff

from lagging\_rank

order by search\_term, month

),

summing\_diff as

(

select

search\_term,

round(sum(search\_rank\_diff)) as diff\_sum,

from

rank\_diff

group by search\_term

),

filter\_percent as

(

select

search\_term,

diff\_sum,

percentile\_cont(cast(diff\_sum as numeric), 0.99) OVER() as percentile\_99,

IF(percentile\_cont(cast(diff\_sum as numeric), 0.99) OVER() <= diff\_sum, true, false) as top\_grossing

from summing\_diff

group by search\_term, diff\_sum

)

select

distinct filter\_percent.search\_term,

diff\_sum

from filter\_percent left join rank\_diff on rank\_diff.search\_term=filter\_percent.search\_term

where top\_grossing=TRUE and filter\_percent.search\_term not like '%christmas%'

order by diff\_sum desc

limit 100

