

Operation Analytics and Investigating Metric Spike

A. **Number of jobs reviewed:** Amount of jobs reviewed over time.

Your task: Calculate the number of jobs reviewed per hour per day for November 2020?

```
select
ds,
count(job_id) as jobs_per_day,
sum(time_spent)/3600 as hours_spent
from nav.jobdata;
```

OUTPUT

```
'30-11-2020', '8', '0.0828'
```

B.Throughput: It is the no. of events happening per second.

Your task: Let's say the above metric is called throughput. Calculate 7 day rolling average of throughput? For throughput, do you prefer daily metric or 7-day rolling and why?

```
WITH CTE AS ( SELECT ds,
COUNT(job_id) AS num_jobs,
SUM(time_spent) AS total_time
FROM nav.jobdata;
WHERE
event IN('transfer','decision') AND ds BETWEEN '2020-11-01' AND '2020-11-30'
GROUP BY ds )
SELECT ds,
ROUND(1.0*SUM(num_jobs) OVER (ORDER BY ds ROWS BETWEEN 6
PRECEDING AND CURRENT ROW) / SUM(total_time) OVER (ORDER BY ds ROWS
BETWEEN 6 PRECEDING AND CURRENT ROW),2) AS throughput_7d FROM CTE
```

C.Percentage share of each language: Share of each language for different contents.

Your task: Calculate the percentage share of each language in the last 30 days?

```
SELECT language,
```

```
language * 100/(SELECT SUM(language) FROM nav.jobdata) as 'Percentage of Total'
```

```
From nav.jobdata
```

```
# language    Percentage of Total
```

```
'Persian', NULL
```

```
'Persian', NULL
```

```
'Persian', NULL
```

```
'Italian', NULL
```

```
'Hindi', NULL
```

```
'French', NULL
```

```
'English', NULL
```

```
'Arabic', NULL
```

D.Duplicate rows: Rows that have the same value present in them.

Your task: Let's say you see some duplicate rows in the data. How will you display duplicates from the table?

```
WITH CTE AS (  
  SELECT  
    *,  
    ROW_NUMBER() OVER (PARTITION BY ds, job_id, actor_id) AS rownum  
  FROM  
    job_data  
  WITH CTE AS (  
    SELECT  
      *,  
      ROW_NUMBER() OVER (PARTITION BY ds, job_id, actor_id) AS rownum  
    FROM  
      job_data
```

```

WITH CTE AS (
SELECT
*,
ROW_NUMBER() OVER (PARTITION BY ds, job_id, actor_id) AS rownum
FROM
job_data
WITH CTE AS (
SELECT
*,
ROW_NUMBER() OVER (PARTITION BY ds, job_id, actor_id) AS rownum
FROM
job_data

```

```

WITH CTE AS (
SELECT
*,
ROW_NUMBER() OVER (PARTITION BY ds, job_id, actor_id) AS rownum
FROM
job_data
SELECT

```

```

job_id, COUNT(job_id)

```

```

FROM nav.jobdata

```

```

GROUP BY job_id

```

```

HAVING COUNT(job_id) > 1

```

```

)

```

```

DELETE
FROM
CTE
WHERE
rownum > 1
# job_idCOUNT(job_id)

```

```

23      3

```

```

SELECT

actor_id, COUNT(actor_id)

FROM nav.jobdata

GROUP BY actor_id

HAVING COUNT(actor_id) > 1

```

```
# actor_id    COUNT(actor_id)
1003    2
```

A. **User Engagement:** To measure the activeness of a user. Measuring if the user finds quality in a product/service.

Your task: Calculate the weekly user engagement?

```
select
count(user_type),
event_type
from metrics.events
left join
metrics.users
on events.user_id = users.user_id
group by events.location
```

```
# count(user_type)  event_type
292    engagement
14     engagement
123    engagement
372    engagement
148    engagement
41     engagement
293    engagement
1155   engagement
```

21	engagement
222	signup_flow
148	signup_flow
109	signup_flow
64	signup_flow
140	signup_flow
59	signup_flow
16	signup_flow
22	signup_flow
20	signup_flow
184	signup_flow
127	signup_flow
93	signup_flow
71	signup_flow
8	signup_flow
105	signup_flow
81	signup_flow
31	signup_flow
23	signup_flow
50	signup_flow
80	signup_flow
5	signup_flow
28	signup_flow

43	signup_flow
18	signup_flow
12	engagement
21	signup_flow
60	signup_flow
21	signup_flow
16	signup_flow
33	signup_flow
57	signup_flow
17	signup_flow
38	signup_flow
60	signup_flow
14	signup_flow
5	signup_flow

B. **User Growth:** Amount of users growing over time for a product.

Your task: Calculate the user growth for product?

BEFORE 2014

select

count(user_id)

from metrics.users

where activated_at > 01-01-2014

```
# count(user_id)
```

```
19066
```

AFTER 2013

```
select
```

```
count(user_id)
```

```
from metrics.users
```

```
where activated_at < 2013-12-31
```

```
# count(user_id)
```

```
9685
```

C. **Weekly Retention:** Users getting retained weekly after signing-up for a product.

Your task: Calculate the weekly retention of users-sign up cohort?

```
SELECT count(user_id)
```

```
FROM metrics.events;
```

```
where occurred_at > 01-01-2014
```

```
# count(user_id)
```

```
4560
```

```
select
```

```
count(distinct user_id)
```

```
from metrics.users
```

```
where state = 'active'
```

```
# count(distinct user_id)
```

```
9381
```

Divide these 2 answer = $9381/4560$

= 2.057

D. **Weekly Engagement:** To measure the activeness of a user. Measuring if the user finds quality in a product/service weekly.

Your task: Calculate the weekly engagement per device?

```
use metrics ;
```

```
select
```

```
count(user_id),
```

```
user_type
```

```
from metrics.events
```

```
where event_type = 'engagement'
```

```
group by events.device
```

```
# count(user_id)    user_type
```

```
170    3
```

```
321    1
```

```
139    2
```

```
39     3
```

```
378    2
```

164 2

763 2

50 2

119 2

132 3

207 3

249 1

469 1

24 3

89 1

89 2

57 2

48 3

112 2

190 2

58 3

51 1

66 1

39 3

61 2

41 3

E. **Email Engagement:** Users engaging with the email service.

Your task: Calculate the email engagement metrics?

```
SELECT
```

```
sum(user_id),
```

```
user_type
```

```
FROM metrics.email_events
```

```
group by action
```

```
# sum(user_id) user_type
```

```
448731290    1
```

```
184717340    1
```

```
91437200     3
```

```
55673639     3
```