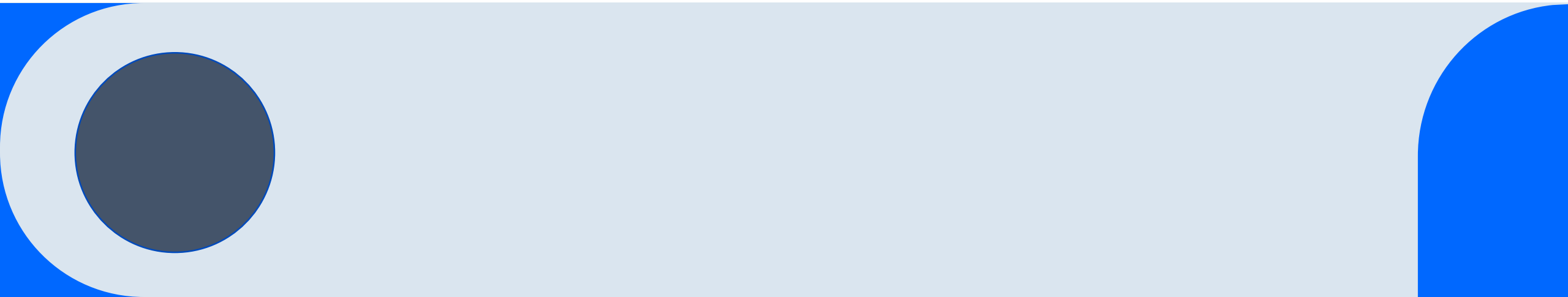
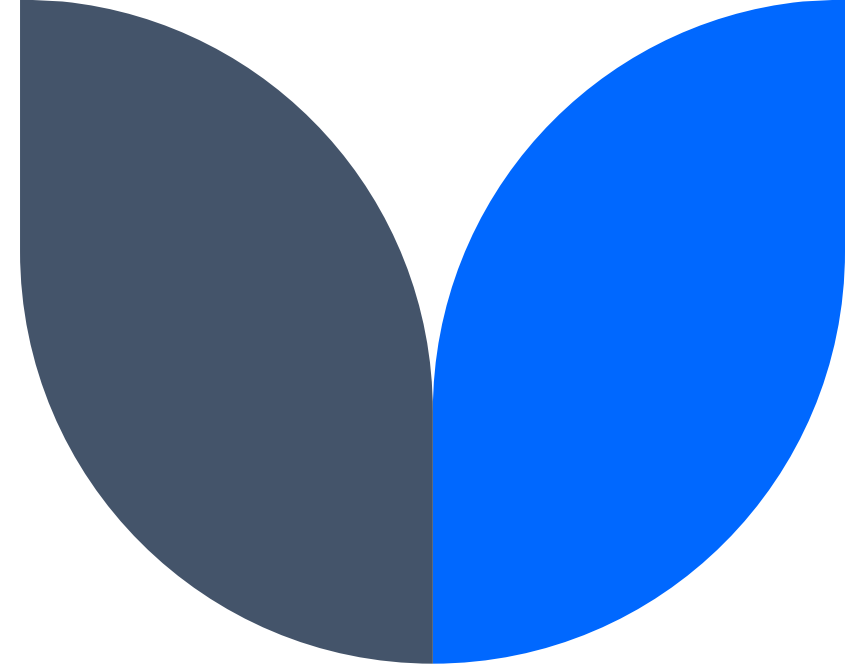




Operation Analytics and Investigating Metric Spike

SANJANA YADAV



Agenda

Case Study 1

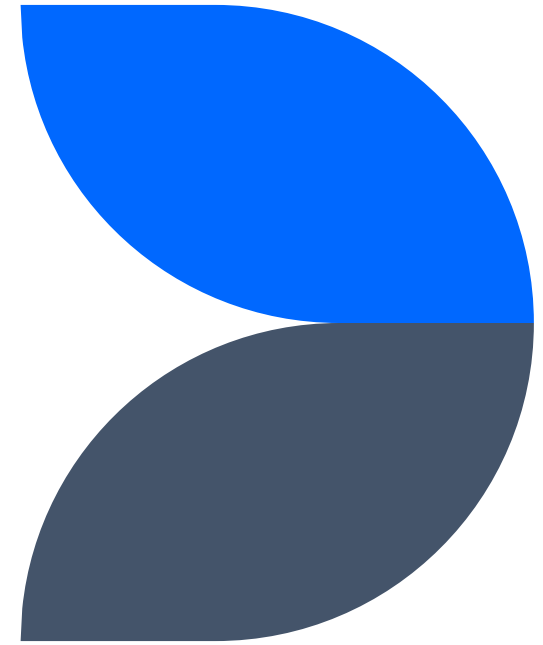
Case Study 2

Introduction

Operation Analytics is a crucial analysis performed on the complete end-to-end operations of a company. Its primary objective is to identify areas that require improvement. This type of analysis holds immense significance for a company as it aids in predicting overall growth or decline in the company's fortunes. It facilitates better automation, enhances understanding and collaboration between cross-functional teams, and improves workflow effectiveness.

Case Study 1

JOB_DATA



Case Study 1

The database given in this case study should be converted into SQL language by creating tables. With the help that database we can find the answer of the following question.

- Number Of job
- Throughput
- Percentage share of each language
- Duplicate

Number of job reviewed

Using the Statement :

```
SELECT count(job_id)/(30*24) AS  
number_of_jobs_reviewed FROM job_data;
```

Result :

number_of_jobs
0.0111

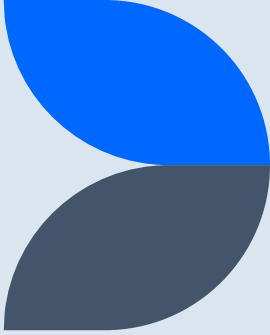
Throughput

Using the Statement :

```
SELECT ds as date,  
jobs_reviewed,  
AVG(jobs_reviewed)  
OVER(ORDER BY ds ROWS  
BETWEEN 6 PRECEDING AND  
CURRENT ROW) AS  
average_job_id FROM ( SELECT  
ds, COUNT(job_id) AS  
jobs_reviewed FROM job_data  
GROUP BY ds ORDER BY ds ) a;
```

	date	jobs_reviewed	average_job_id
	2020-11-25	1	1.0000
▶	2020-11-26	1	1.0000
	2020-11-27	1	1.0000
	2020-11-29	1	1.2000
	2020-11-28	2	1.2500
	2020-11-30	2	1.3333

Percentage share of each language



Using the Statement :

```
select job_data.job_id,  
job_data.language, count(distinct  
job_data.language) as total_ ,  
((count(job_data.language)/(select  
count(*) from job_data))*100) as  
percentage_ from job_data group  
by job_data.language;
```

Result :

job_id	language	total_	Percentage_
22	Arabic	1	12.5
21	English	1	12.5
11	French	1	12.5
25	Hindi	1	12.5
20	Italian	1	12.5
23	Persian	1	37.5

Duplicate Rows

Using the Statement :

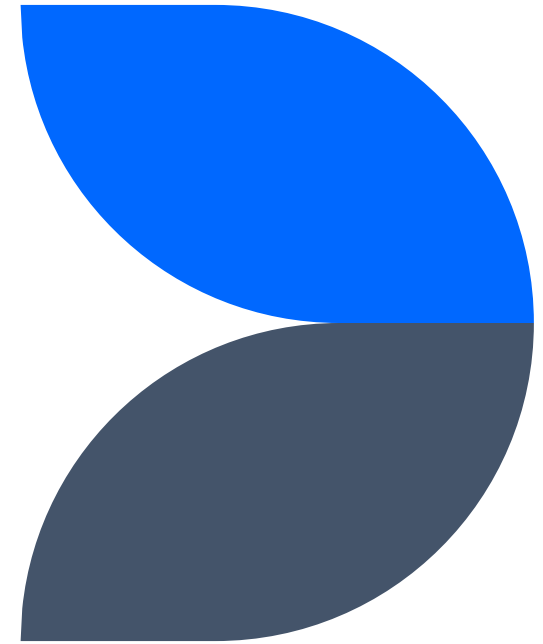
```
SELECT * FROM ( SELECT *,  
ROW_NUMBER()OVER(PARTITION BY job_id) AS row_num  
FROM job_data ) a WHERE row_num>1;
```

Result :

ds	job_id	actor_id	event	language	time_spent	org	row_num
28-11-2020	23	1005	transfer	Persian	22	D	2
26-11-2020	23	1004	skip	Persian	56	A	3

Case Study 2

Investigating metric spike

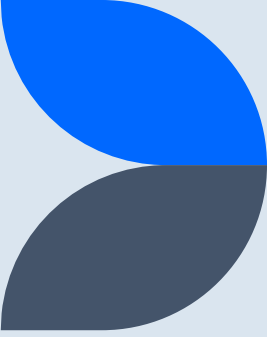


Case Study 2

The database given in this case study should be converted into SQL language by creating tables. With the help that database we can find the answer of the following question.

- User Engagement
- User Growth
- Weekly Retention
- Weekly Engagement
- Email Engagement

User Engagement



Your task: Calculate the weekly user engagement.

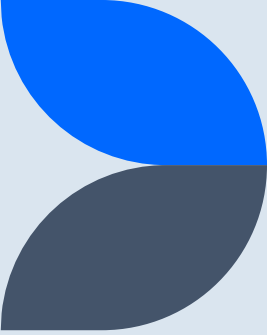
Query :

```
SELECT extract (week from occurred_at) as week_number,  
count(distinct user_id) as number_of_users FROM  
tutorial.yammer_events group by week_number;
```

RESULT

week_number	number_of_users
18	791
19	1244
20	1270
21	1341
22	1293
23	1366
24	1434
25	1464
26	1443
27	1477
28	1556
29	1556
30	1593
31	1685
32	1483
33	1438
34	1412
35	1442

User Growth



Your task: Calculate the user growth for product.

Query :

```
select year_num, week_num, num_active_users,  
SUM(num_active_users)OVER(ORDER BY year_num,  
week_num ROWS BETWEEN UNBOUNDED PRECEDING AND  
CURRENT ROW) AS cum_active_users from ( select extract  
(year from a.activated_at) as year_num, extract (week from  
a.activated_at) as week_num, count(distinct user_id) as  
num_active_users from tutorial.yammer_users a WHERE state =  
'active' group by year_num,week_num order by  
year_num,week_num ) a;
```

RESULTS

year_num	week_num	num_active_users	cum_active_users
2013	1	67	67
2013	2	29	96
2013	3	47	143
2013	4	36	179
2013	5	30	209
2013	6	48	257
2013	7	41	298
2013	8	39	337
2013	9	33	370
2013	10	43	413
2013	11	33	446
2013	12	32	478
2013	13	33	511
2013	14	40	551
2013	15	35	586
2013	16	42	628
2013	17	48	676
2013	18	48	724
2013	19	45	769
2013	20	55	824
2013	21	41	865
2013	22	49	914
2013	23	51	965
2013	24	51	1016
2013	25	46	1062
2013	26	57	1119
2013	27	57	1176
2013	28	52	1228
2013	29	71	1299
2013	30	66	1365
2013	31	69	1434
2013	32	66	1500
2013	33	73	1573
2013	34	70	1643
2013	35	80	1723
2013	36	65	1788
2013	37	71	1859
2013	38	84	1943
2013	39	92	2035
2013	40	81	2116
2013	41	88	2204
2013	42	74	2278
2013	43	97	2375
2013	44	92	2467

year_num	week_num	num_active_users	cum_active_users
2013		45	97
2013		46	94
2013		47	82
2013		48	103
2013		49	96
2013		50	117
2013		51	123
2013		52	104
2014	1	91	3374
2014	2	122	3496
2014	3	112	3608
2014	4	113	3721
2014	5	130	3851
2014	6	132	3983
2014	7	135	4118
2014	8	127	4245
2014	9	127	4372
2014	10	135	4507
2014	11	152	4659
2014	12	132	4791
2014	13	151	4942
2014	14	161	5103
2014	15	166	5269
2014	16	165	5434
2014	17	176	5610
2014	18	172	5782
2014	19	160	5942
2014	20	186	6128
2014	21	177	6305
2014	22	186	6491
2014	23	197	6688
2014	24	198	6886
2014	25	222	7108
2014	26	210	7318
2014	27	199	7517
2014	28	223	7740
2014	29	215	7955
2014	30	228	8183
2014	31	234	8417
2014	32	189	8606
2014	33	250	8856
2014	34	259	9115
2014	35	266	9381

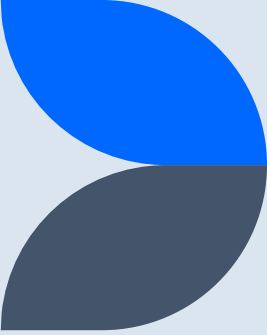
RESULT

Query :

```
select count(*) from tutorial.yammer_users where state =  
'active';
```

Hence there are total : 9381

WEEKLY RENTATION



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

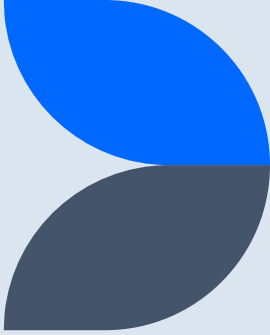
Query :
`SELECT distinct user_id, COUNT(user_id), SUM(CASE WHEN retention_week = 1 Then 1 Else 0 END) as per_week_retention FROM (SELECT a.user_id, a.signup_week, b.engagement_week, b.engagement_week - a.signup_week as retention_week FROM ((SELECT distinct user_id, extract(week from occurred_at) as signup_week from tutorial.yammer_events WHERE event_type = 'signup_flow' and event_name = 'complete_signup')a LEFT JOIN (SELECT distinct user_id, extract (week from occurred_at) as engagement_week FROM tutorial.yammer_events where event_type = 'engagement')b on a.user_id = b.user_id))d group by user_id order by user_id`

R E S U L T

Here is the link of the result:

https://drive.google.com/file/d/1ds_oQpU6eojYLDIZMRvhw07Br2g2BAuO/view?usp=sharing

WEEKLY ENGAGEMENT



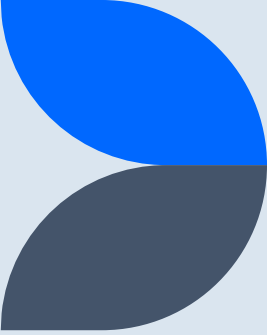
Your task: Calculate the weekly engagement per device?

Query :

```
SELECT extract(year from occurred_at) as year_num,  
extract(week from occurred_at) as week_num, device,  
COUNT(distinct user_id) as no_of_users FROM  
tutorial.yammer_events where event_type = 'engagement'  
GROUP by 1,2,3 order by 1,2,3;
```

RESULT : https://drive.google.com/file/d/1xta_GsUjTJO4SkCG-pkxQ3DPV1GN8UC4/view?usp=drive_link

EMAIL ENGAGEMENT



Your task: Calculate the email engagement metric.

Query :

```
SELECT 100.0*SUM(CASE when email_cat = 'email_opened' then 1
else 0 end)/SUM(CASE when email_cat = 'email_sent' then 1 else 0
end) as email_opening_rate, 100.0*SUM(CASE when email_cat =
'email_clicked' then 1 else 0 end)/SUM(CASE when email_cat =
'email_sent' then 1 else 0 end) as email_clicking_rate FROM ( SELECT
*, CASE WHEN action in
('sent_weekly_digest','sent_reengagement_email') then 'email_sent'
WHEN action in ('email_open') then 'email_opened' WHEN action in
('email_clickthrough') then 'email_clicked' end as email_cat from
tutorial.yammer_emails ) a;
```

RESULT

email_opening_rate	email_clicking_rate
33.58338805	14.78988838



Thank you

Sanjana Kumari Yadav