

User Engagement Drop Analysis

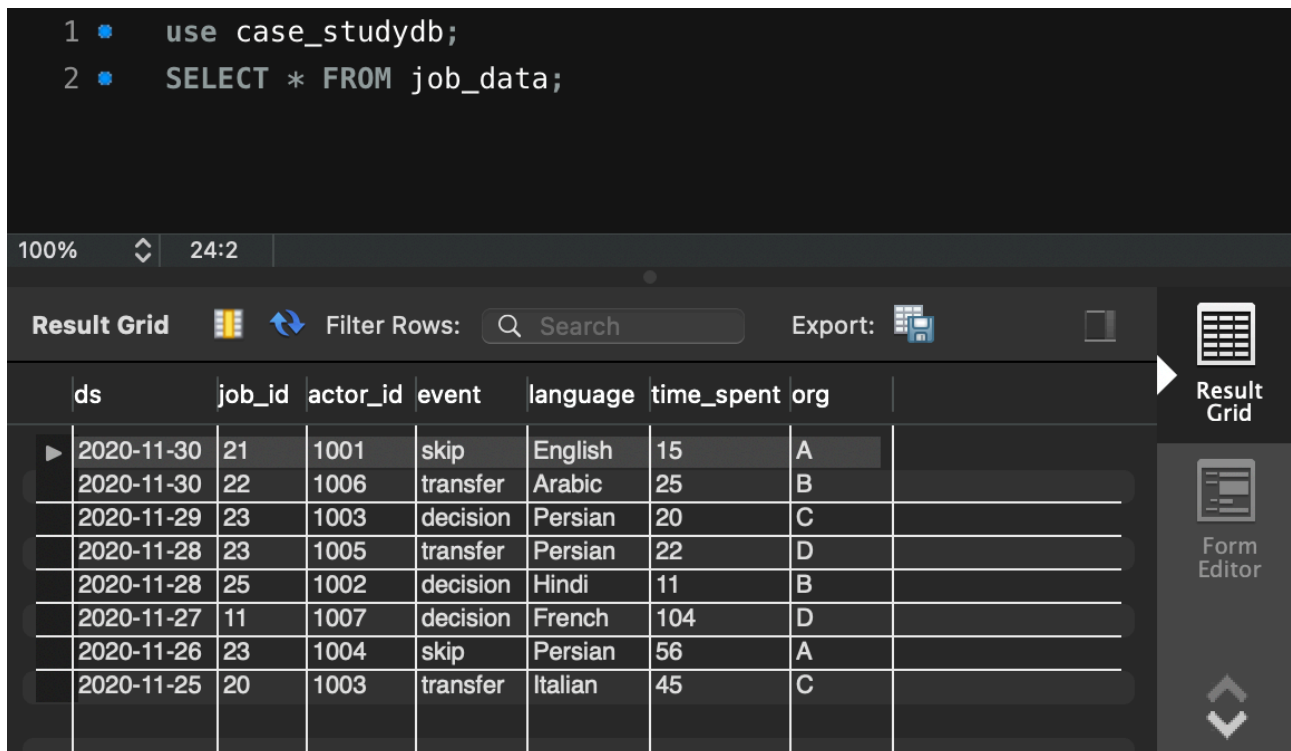
Project Description: Yammer is a social network for communicating with coworkers. Individuals share documents, updates, and ideas by posting them in groups. Yammer is free to use indefinitely, but companies must pay license fees if they want access to administrative controls, including integration with user management systems like ActiveDirectory. Yammer analysts are trained to constantly consider the value of each individual project; they seek to maximize the return on their time. Analysts typically opt for less precise solutions to problems if it means investing substantially less time as well. They are also taught to consider the impact of everything on the company at large. This includes high-level decision making like choosing which projects to prioritize. It also influences the way analysts think about metrics. Product decisions are always evaluated against core engagement, retention, and growth metrics in addition to product-specific usage metrics (like, for example, the number of times someone views another user's profile).

Goal: The goal of this project is to determine a dip caused in the number of engaged users (Yammer defines engagement as having made some type of server call by interacting with the product) in the last week of July.

Approach: I am going to perform analysis through SQL queries and after having a good understanding of the data.

Case Study -1: Operation Analytics

Table - job_data



The screenshot shows a database query interface. At the top, a SQL query is entered: `1 use case_studydb;` and `2 SELECT * FROM job_data;`. Below the query, a toolbar includes a zoom level of 100%, a refresh icon, a time indicator of 24:2, and buttons for 'Result Grid', 'Filter Rows', 'Search', and 'Export'. The main area displays a table with the following data:

	ds	job_id	actor_id	event	language	time_spent	org
▶	2020-11-30	21	1001	skip	English	15	A
■	2020-11-30	22	1006	transfer	Arabic	25	B
■	2020-11-29	23	1003	decision	Persian	20	C
■	2020-11-28	23	1005	transfer	Persian	22	D
■	2020-11-28	25	1002	decision	Hindi	11	B
■	2020-11-27	11	1007	decision	French	104	D
■	2020-11-26	23	1004	skip	Persian	56	A
■	2020-11-25	20	1003	transfer	Italian	45	C

On the right side of the interface, there are buttons for 'Result Grid' and 'Form Editor'.

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

```
SELECT ds , count(job_id) as jobs_Perday, sum(time_spent)/3600 as hours_spent
FROM job_data
WHERE ds >='2020-11-01' and ds <='2020-11-30'
AND event IN ('transfer','decision')
GROUP BY ds;
```

Output-



The screenshot shows the output of the SQL query. It is a table with the following data:

	ds	jobs_Perday	hours_spent
▶	2020-11-30	1	0.0069
■	2020-11-29	1	0.0056
■	2020-11-28	2	0.0092
■	2020-11-27	1	0.0289
■	2020-11-25	1	0.0125

Ans: Total no. of jobs - 8

Total no. of hours spent reviewing the jobs - 0.0631 hr

Total no. of days - 6

Jobs reviewed per hour per day = $(8 / 0.0631) / 6 = \underline{\underline{21.13}}$

(it means that if in total 1 hr was spent in reviewing jobs at avg 21 jobs would be reviewed)

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

```
SELECT language, (count(language)/6)*100 as language_perc
FROM job_data
GROUP BY language;
```

Output -

	language	language_perc	
▶	English	16.6667	
	Arabic	16.6667	
	Persian	50.0000	
	Hindi	16.6667	
	French	16.6667	
	Italian	16.6667	

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

```
SELECT *
FROM job_data
GROUP BY ds,job_id,language,actor_id,event,time_spent,org
HAVING COUNT(*) > 1;
```

(For an entire duplicate row)

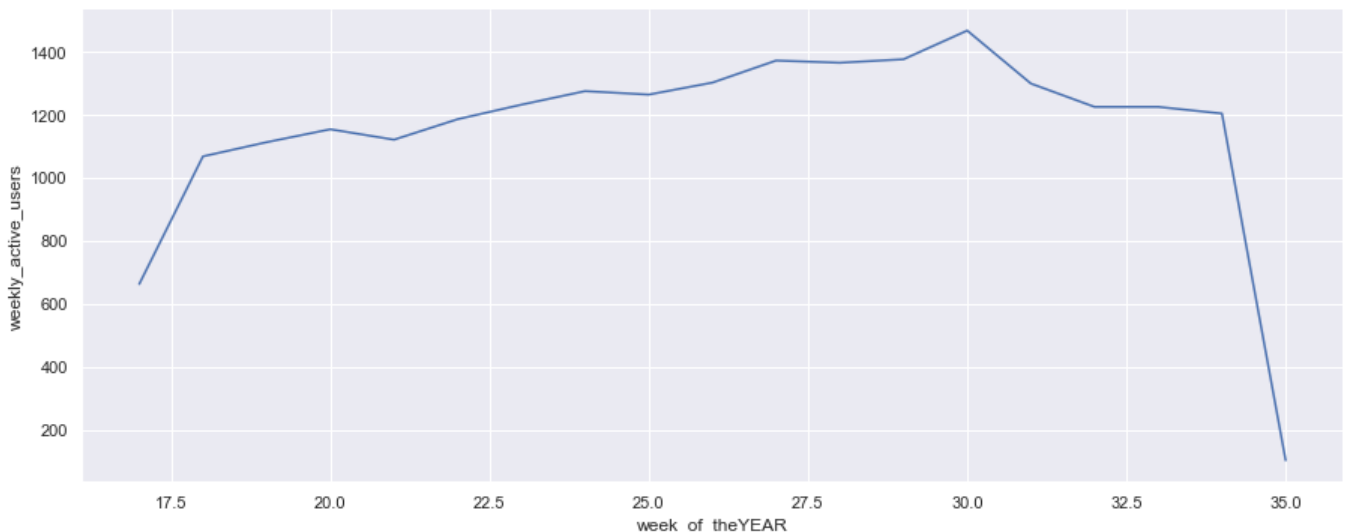
Case Study – 2: Investigating metric Spike

QA : Calculate the weekly user engagement?

```
SELECT WEEK(occurred_at) AS week_of_theYEAR,  
COUNT(DISTINCT user_id) AS weekly_active_users  
FROM events_table  
WHERE event_type = 'engagement'  
AND event_name = 'login'  
GROUP BY 1  
ORDER BY 1;
```

Output -

	week_of_theYEAR	weekly_active_users
▶	17	663
	18	1068
	19	1113
	20	1154
	21	1121
	22	1186
	23	1232
	24	1275
	25	1264
	26	1302
	27	1372
	28	1365
	29	1376
	30	1467
	31	1299
	32	1225
	33	1225
	34	1204
	35	104

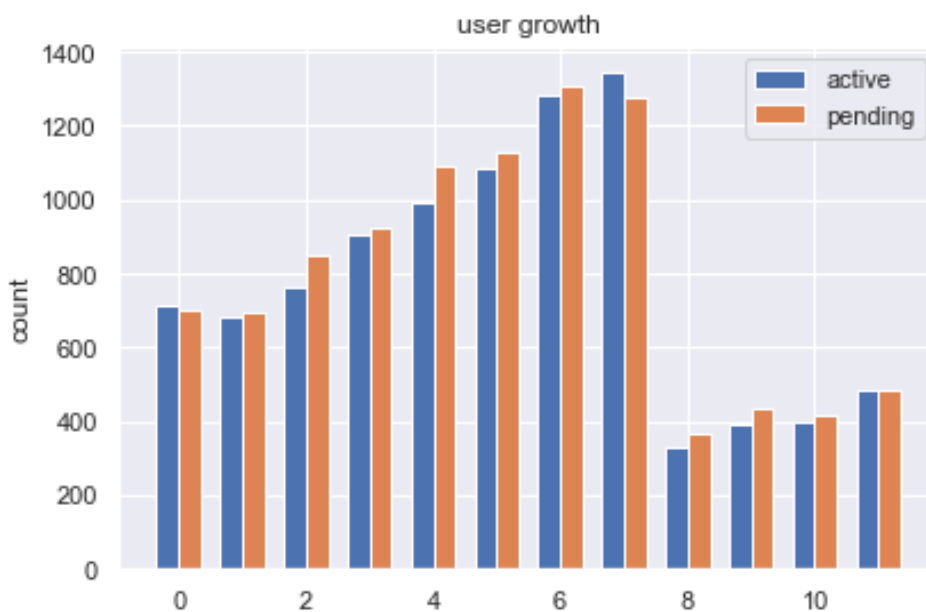


QB : Calculate the user growth for product?

```
SELECT state, count(state),MONTH(created_at) AS months
FROM users_table
GROUP BY MONTH(created_at) , state;
```

Output-

state	count(state)	months
active	712	1
pending	703	1
active	685	2
pending	697	2
active	765	3
pending	849	3
pending	922	4
active	907	4
active	993	5
pending	1090	5
active	1086	6
pending	1127	6
pending	1310	7
active	1281	7
active	1347	8
pending	1279	8
active	330	9
pending	369	9
pending	436	10
active	390	10
active	399	11
pending	417	11
active	486	12
pending	486	12



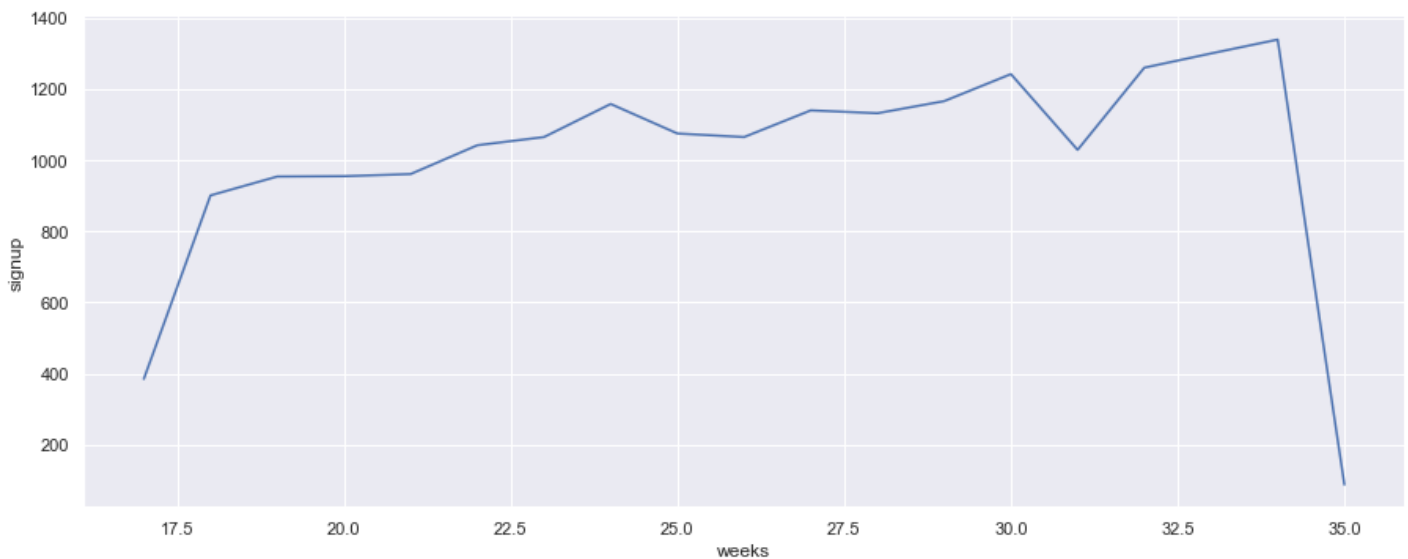
From the graph we can infer that the growth increased in the first 7 months especially in the 6th and 7th month and decreased drastically towards the end of the year

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

```
SELECT WEEK(created_at) AS weeks,  
COUNT(CASE WHEN event_type = "signup_flow" THEN user_id ELSE NULL END)  
AS signup  
FROM events_table  
GROUP BY 1  
ORDER BY 1;
```

Output-

	weeks	signup	
▶	17	385	
	18	901	
	19	954	
	20	955	
	21	961	
	22	1042	
	23	1065	
	24	1158	
	25	1075	
	26	1065	
	27	1140	
	28	1132	
	29	1166	
	30	1242	
	31	1029	
	32	1260	
	33	1300	
	34	1339	
	35	88	



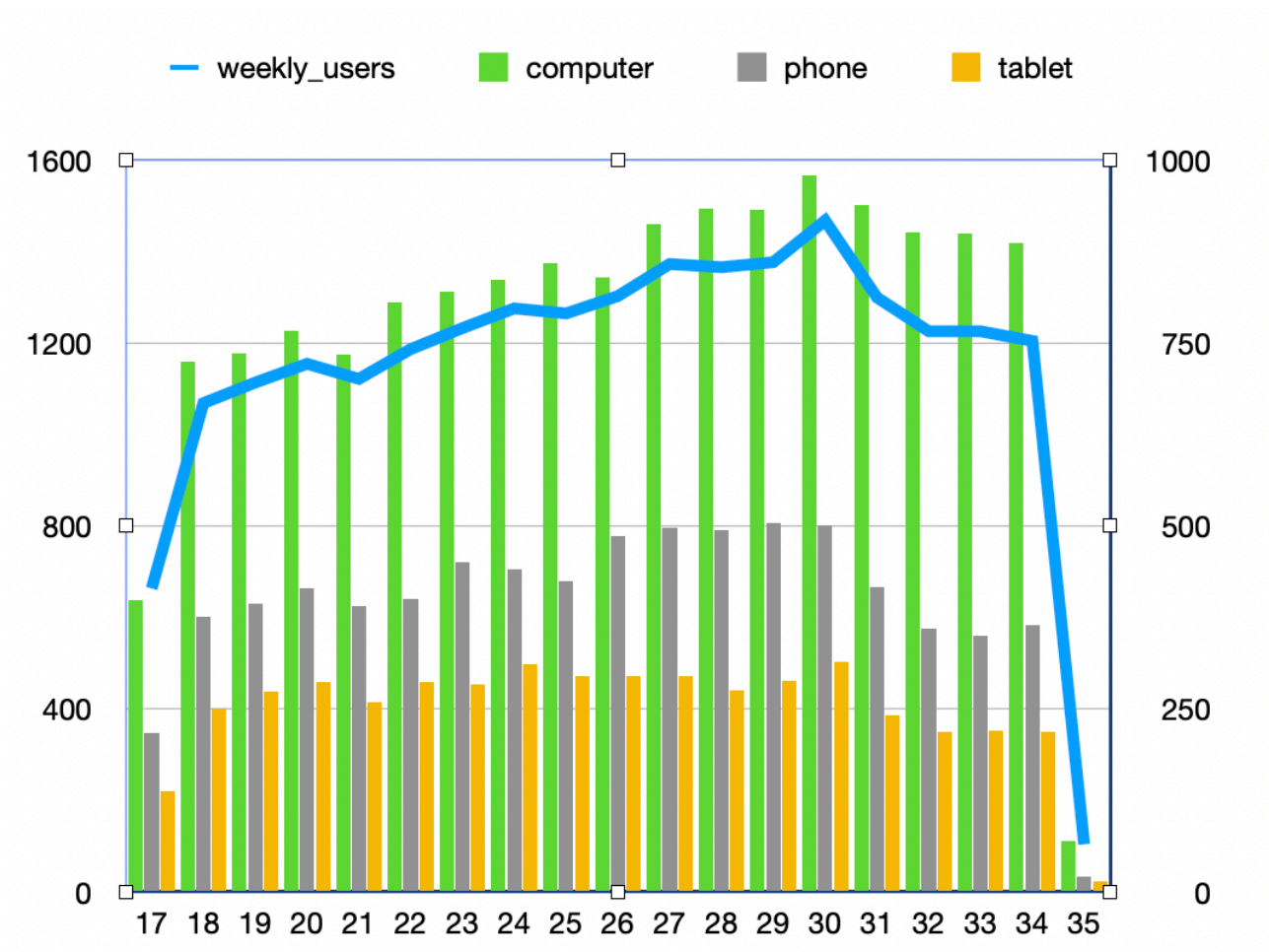
QD : Calculate the weekly engagement per device?

/*getting the list of devices first for grouping them as computes, mobiles etc*/
SELECT DISTINCT device FROM events_table;

```
SELECT WEEK(occurred_at) AS week_of_theYEAR,  
COUNT(DISTINCT user_id) AS weekly_users,  
COUNT(DISTINCT CASE WHEN device IN ('macbook pro', 'acer aspire  
notebook','acer aspire desktop','lenovo thinkpad', 'mac mini', 'dell inspiron  
desktop','dell inspiron notebook','windows surface','macbook air','asus  
chromebook','hp pavilion desktop') THEN user_id ELSE NULL END) AS computer,  
COUNT(DISTINCT CASE WHEN device IN ('iphone 5s','nokia lumia 635','amazon  
fire phone','iphone 4s','htc one','iphone 5','samsung galaxy s4') THEN user_id  
ELSE NULL END) AS phone,  
COUNT(DISTINCT CASE WHEN device IN ('kindle fire','samsung galaxy  
note','ipad mini','nexus 7','nexus 10','samsung galaxy tablet','nexus 5','ipad air')  
THEN user_id ELSE NULL END) AS tablet  
FROM events_table  
WHERE event_type = 'engagement'  
AND event_name = 'login'  
GROUP BY 1  
ORDER BY 1;
```

Output-

	week_of_theYEAR	weekly_users	computer	phone	tablet	
	17	663	399	217	137	
	18	1068	724	375	250	
	19	1113	736	394	274	
	20	1154	767	414	286	
	21	1121	734	391	259	
	22	1186	805	400	286	
	23	1232	819	450	283	
	24	1275	836	441	311	
	25	1264	858	425	294	
	26	1302	840	486	294	
	27	1372	912	497	294	
	28	1365	934	494	275	
	29	1376	932	503	288	
	30	1467	979	500	314	
	31	1299	938	417	242	
	32	1225	900	360	219	
	33	1225	899	350	220	
	34	1204	887	364	219	
	35	104	69	21	14	



It can be seen that the pattern of user engagement per device is
Computer > phone > tablet

QE : Calculate the email engagement metrics?

```
SELECT WEEK(occurred_at) AS weeks,
COUNT(CASE WHEN action = 'sent_weekly_digest' THEN user_id ELSE NULL
END) AS sent_weekly_digest,
COUNT(CASE WHEN action = 'email_open' THEN user_id ELSE NULL END) AS
email_open,
COUNT(CASE WHEN action = 'email_clickthrough' THEN user_id ELSE NULL
END) AS email_clickthrough,
COUNT(CASE WHEN action = 'sent_reengagement_email' THEN user_id ELSE
NULL END) AS sent_reengagement_email
FROM email_events_table
GROUP BY 1
ORDER BY 1;
```


Output -

	weeks	sent_weekly_digest	email_open	email_clickthrough	sent_reengagement_email	
	17	908	310	166	73	
	18	2602	912	430	157	
	19	2665	972	477	173	
	20	2733	1004	507	191	
	21	2822	1014	443	164	
	22	2911	987	488	192	
	23	3003	1075	538	197	
	24	3105	1155	554	226	
	25	3207	1096	530	196	
	26	3302	1165	556	219	
	27	3399	1228	621	213	
	28	3499	1250	599	213	
	29	3592	1219	590	213	
	30	3706	1383	630	231	
	31	3793	1351	445	222	
	32	3897	1337	418	200	
	33	4012	1432	490	264	
	34	4111	1528	490	261	
	35	0	41	38	48	