

Case Study 1 : JOB DATA

1. Calculate the number of jobs reviewed per hour per day for November 2020?

```
select ds as dates ,  
round((count(job_id)/sum(time_spent))*3600) as "jobs_reviewed"  
from job_data  
where ds between '2020-11-01' and '2020-11-30'  
group by ds ;
```

dates	jobs_Reviewed
2020-11-30	180
2020-11-29	180
2020-11-28	218
2020-11-27	35
2020-11-26	64
2020-11-25	80

2. Calculate 7 day rolling average of throughput? For throughput, do you prefer daily metric or 7-day rolling and why?

Weekly throughput:

```
select round(count(event)/sum(time_spent),2) as 'weekly_throughput'  
from job_data ;
```

weekly_Throughput
0.03

Daily Throughput:

```
select ds as dates,  
round(count(event)/sum(time_spent),2) as 'daily_throughput'  
from job_data  
group by dates  
order by dates desc ;
```

date	daily_Throughput
2020-11-30	0.05
2020-11-29	0.05
2020-11-28	0.06
2020-11-27	0.01
2020-11-26	0.02
2020-11-25	0.02

3. Calculate the percentage share of each language in the last 30 days?

```
select language as languages ,  
round((count(language)/(select count(*) from job_data)),2)*100  
as percentage_share  
from job_data  
group by language
```

languages	percentage_share
English	13.00
Arabic	13.00
Persian	38.00
Hindi	13.00
French	13.00
Italian	13.00

4. How will you display duplicates from the table?

```
select actor_id,  
count(*) as duplicates  
from job_data  
group by actor_id  
having duplicates > 1 ;
```

actor_id	duplicates
1003	2

Case Study 2 : Investigating metric spike

1. Calculate the weekly user engagement?

```
SELECT week(occurred_at) as Week,  
count(DISTINCT user_id) as Weekly_User_engagement  
FROM events  
GROUP BY week(occurred_at)  
ORDER BY week(occurred_at);
```

Week	Weekly_User_engagement
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

2. Calculate the user growth for product?

```

select year,
week,
num_active_users,
sum(num_active_users)over(order by year,week rows between
unbounded preceding and current row) as cum_active_users
from
(select extract(year from activated_at) as year,
extract(week from activated_at) as week,
count(distinct user_id) as num_active_users
from users
where state = 'active'
group by year,week
order by year,week) a ;

```

year	week	num_active_users	cum_active_users
2013	0	23	23
2013	1	30	53
2013	2	48	101
2013	3	36	137
2013	4	30	167
2013	5	48	215
2013	6	38	253
2013	7	42	295
2013	8	34	329
2013	9	43	372
2013	10	32	404
2013	11	31	435
2013	12	33	468
2013	13	39	507
2013	14	35	542
2013	15	43	585
2013	16	46	631
2013	17	49	680
2013	18	44	724
2013	19	57	781
2013	20	39	820
2013	21	49	869
2013	22	54	923
2013	23	50	973
2013	24	45	1018
2013	25	57	1075
2013	26	56	1131
2013	27	52	1183
2013	28	72	1255
2013	29	67	1322
2013	30	67	1389
2013	31	67	1456
2013	32	71	1527
2013	33	73	1600
2013	34	78	1678
2013	35	63	1741
2013	36	72	1813
2013	37	85	1898
2013	38	90	1988
2013	39	84	2072
2013	40	87	2159
2013	41	73	2232
2013	42	99	2331
2013	43	89	2420
2013	44	96	2516
2013	45	91	2607
2013	46	88	2695
2013	47	102	2797
2013	48	97	2894

2013	49	116	3010
2013	50	124	3134
2013	51	102	3236
2013	52	47	3283
2014	0	83	3366
2014	1	126	3492
2014	2	109	3601
2014	3	113	3714
2014	4	130	3844
2014	5	133	3977
2014	6	135	4112
2014	7	125	4237
2014	8	129	4366
2014	9	133	4499
2014	10	154	4653
2014	11	130	4783
2014	12	148	4931
2014	13	167	5098
2014	14	162	5260
2014	15	164	5424
2014	16	179	5603
2014	17	170	5773
2014	18	163	5936
2014	19	185	6121
2014	20	176	6297
2014	21	183	6480
2014	22	196	6676
2014	23	196	6872
2014	24	229	7101
2014	25	207	7308
2014	26	201	7509
2014	27	222	7731
2014	28	215	7946
2014	29	221	8167
2014	30	238	8405
2014	31	193	8598
2014	32	245	8843
2014	33	261	9104
2014	34	259	9363
2014	35	18	9381

3.Calculate the weekly retention of users-sign up cohort?

```

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 user_id,extract(week from occurred_at) as signup_week
from 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 events
where event_type = 'engagement')b
on a.user_id = b.user_id
)d
group by user_id
order by user_id ;

```

user_id	count(user_id)	per_week_retention
11768	1	0
11770	1	0
11775	2	1
11778	3	0
11779	5	1
11780	2	1
11785	1	0
11787	3	1
11791	2	1
11793	6	1
11795	2	1
11798	6	1
11799	10	1
11801	2	1
11804	2	1
11806	1	0
11809	1	0
11811	2	1
11813	6	0
11816	3	0
11818	2	1
11820	4	1

AND More.....

4.Calculate the weekly engagement per device?

```

SELECT week(occurred_at) as Weeks,
device,
count(distinct user_id)as User_engagement
FROM events
GROUP BY device,
week(occurred_at)
ORDER BY week(occurred_at);

```

Weeks	device	User_engagement
17	acer aspire desktop	9
17	acer aspire notebook	20
17	amazon fire phone	4
17	asus chromebook	21
17	dell inspiron desktop	18
17	dell inspiron notebook	46
17	hp pavilion desktop	14
17	htc one	16
17	ipad air	27
17	ipad mini	19
17	iphone 4s	21
17	iphone 5	65
17	iphone 5s	42
17	kindle fire	6
17	lenovo thinkpad	86
17	mac mini	6
17	macbook air	54
17	macbook pro	143
17	nexus 10	16
17	nexus 5	40

And more.....

5.Calculate the email engagement metrics?

```

SELECT week(occurred_at) as Week,
count( DISTINCT ( CASE WHEN action = "sent_weekly_digest"
THEN user_id end )) as weekly_digest,
count( distinct ( CASE WHEN action = "sent_reengagement_email"
THEN user_id end )) as reengagement_mail,
count( distinct ( CASE WHEN action = "email_open"
THEN user_id end )) as opened_email,
count( distinct ( CASE WHEN action = "email_clickthrough"
THEN user_id end )) as email_clickthrough
FROM email_events
GROUP BY week(occurred_at)
ORDER BY week(occurred_at);

```

Week	weekly_digest	reengagement_mail	opened_email	email_clickthrough
17	908	73	310	166
18	2602	157	900	425
19	2665	173	961	476
20	2733	191	989	501
21	2822	164	996	436
22	2911	192	965	478
23	3003	197	1057	529
24	3105	226	1136	549
25	3207	196	1084	524
26	3302	219	1149	550
27	3399	213	1207	613
28	3499	213	1228	594
29	3592	213	1201	583
30	3706	231	1363	625
31	3793	222	1338	444
32	3897	200	1318	416
33	4012	264	1417	490
34	4111	261	1502	481
35	0	48	41	38