

Operational Analytics and Investigating Metric Spike

Case Studies and Insights

trainity

Operation Analytics & Investigating metric spike case study



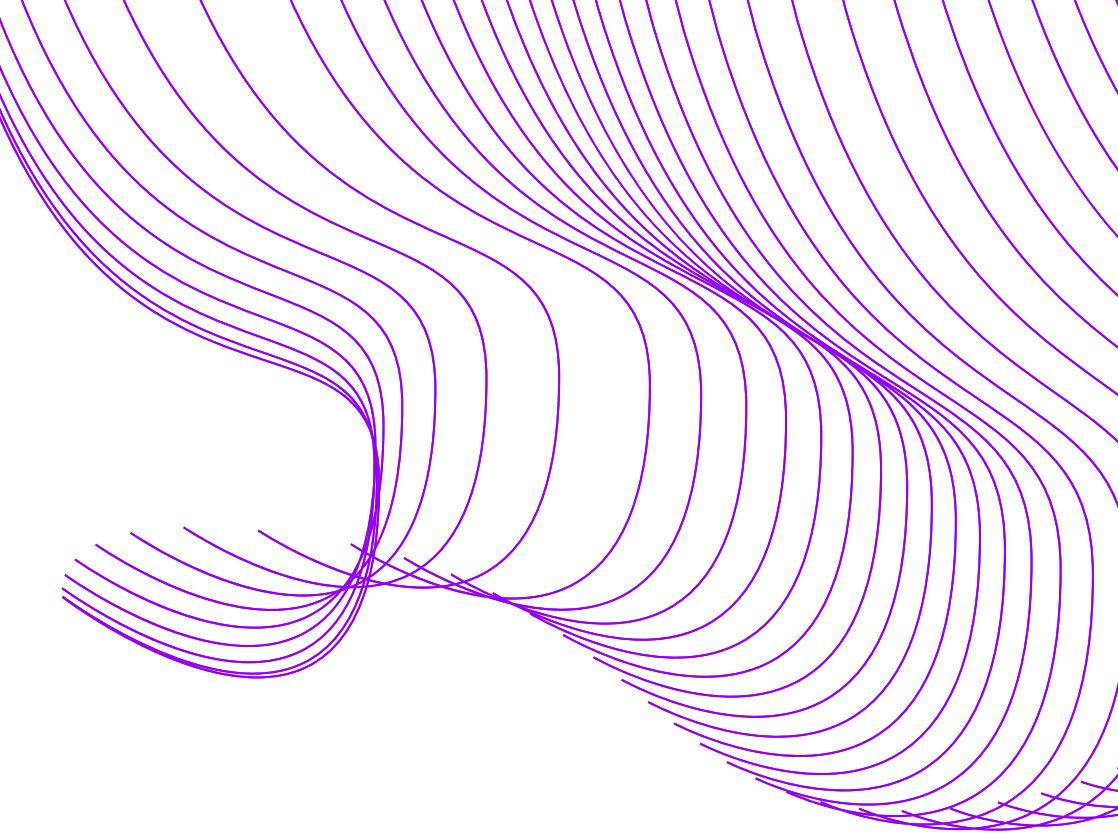
Srisatya

Project Description

- **Operational Analytics** focuses on analyzing a company's end-to-end operations to identify areas for improvement.
- The project involves investigating sudden changes in key metrics and providing insights based on data analysis.



Objectives



- Analyze job data for trends and spikes.
- Investigate user engagement and growth metrics.
- Provide actionable insights for operational improvement.

Methodology:

Database Creation

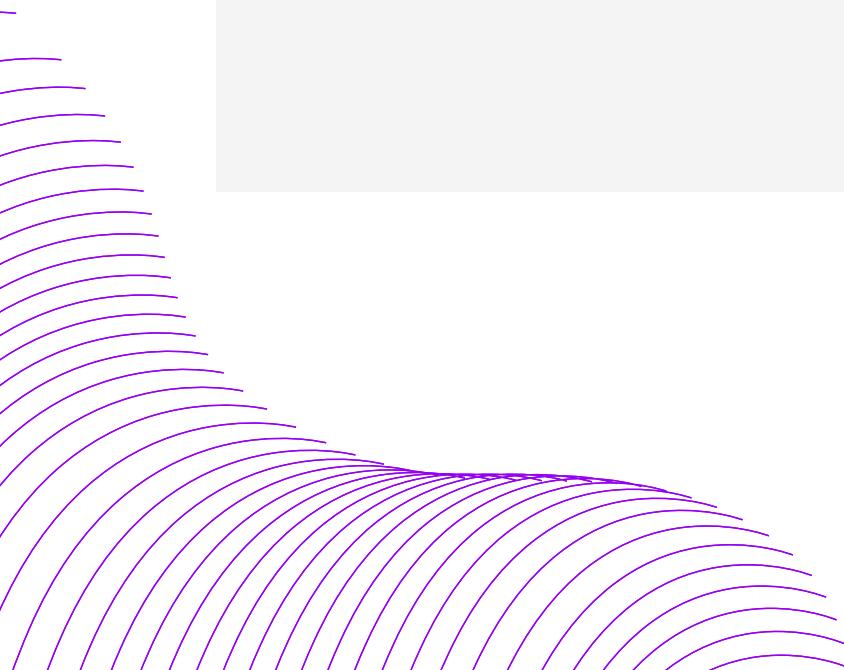
Created necessary tables and imported CSV files into MySQL Workbench.

Data Analysis

Utilized SQL for analysis based on the provided case studies.

Insights Generation

Focused on interpreting results to derive meaningful insights.



Tech Stack Used

MySQL Workbench

- For database management and SQL queries.
-

Reporting Tools

- PowerPoint and/or PDF for reporting findings.

Case Study 1: Job Data Analysis

1. Jobs Reviewed Over Time

Task : Calculate the number of jobs reviewed per hour for each day in November 2020

SQL Query :

#1. Jobs Reviewed Over Time:

```
SELECT
    ds,
    sum(time_spent)/3600 AS hour_of_day,
    COUNT(job_id) AS jobs_reviewed
FROM
    job_data
WHERE
    ds BETWEEN '2020-11-01' AND '2020-11-30'
GROUP BY ds;
```

Output :

	ds	hour_of_day	jobs_reviewed
▶	2020-11-30	0.0111	2
	2020-11-29	0.0056	1
	2020-11-28	0.0092	2
	2020-11-27	0.0289	1
	2020-11-26	0.0156	1
	2020-11-25	0.0125	1



Insights

- The analysis shows low engagement in job reviews during November 2020, with only 1-2 jobs reviewed per day and minimal time spent on each.
- This suggests a need for improvements to encourage more user participation and increase job review activity.

2. Throughput Analysis

Task : Calculate the 7-day rolling average of throughput.

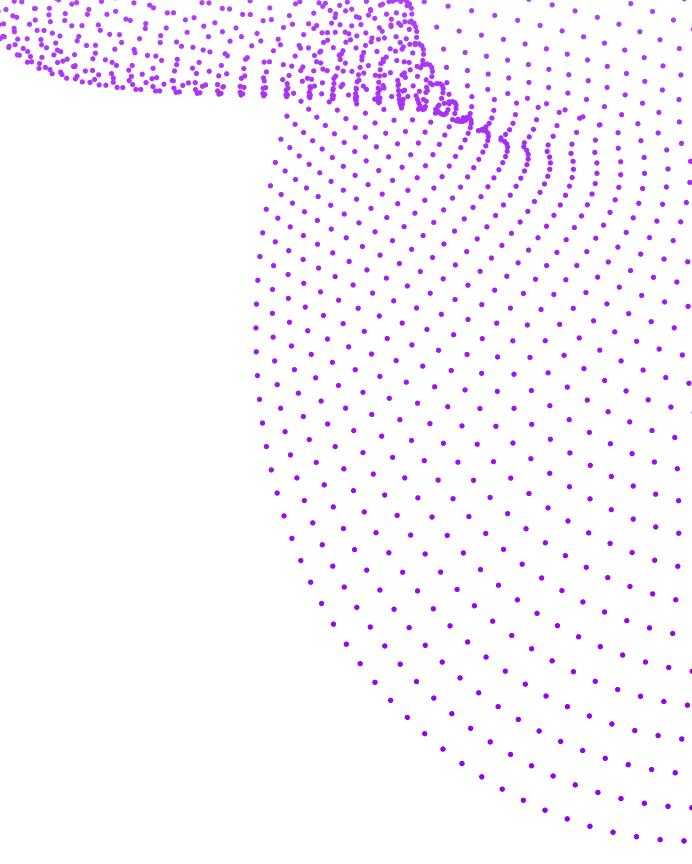
SQL Query :

```
#2.Throughput Analysis

WITH daily_throughput AS (
    SELECT
        ds,
        COUNT(job_id) AS daily_events,
        SUM(time_spent) AS total_time_spent
    FROM
        job_data
    WHERE event IN('transfer','decision')
    AND ds BETWEEN '2020-11-01' AND '2020-11-30'
    GROUP BY
        ds
)
SELECT
    ds,
    ROUND(1.0*SUM(daily_events) OVER (ORDER BY ds ROWS BETWEEN 6 PRECEDING AND CURRENT ROW) / SUM(total_time_spent) OVER (ORDER BY ds ROWS BETI
FROM
    daily_throughput;
```

Output :

	ds	rolling_throughput
▶	2020-11-25	0.02
	2020-11-27	0.01
	2020-11-28	0.02
	2020-11-29	0.02
	2020-11-30	0.03



Insights

- The throughput analysis reveals a low average of processed events, ranging from 0.01 to 0.03, indicating limited activity during November 2020.
- This suggests there may be inefficiencies in the job review process that need to be addressed to improve performance.

3. Language Share Analysis

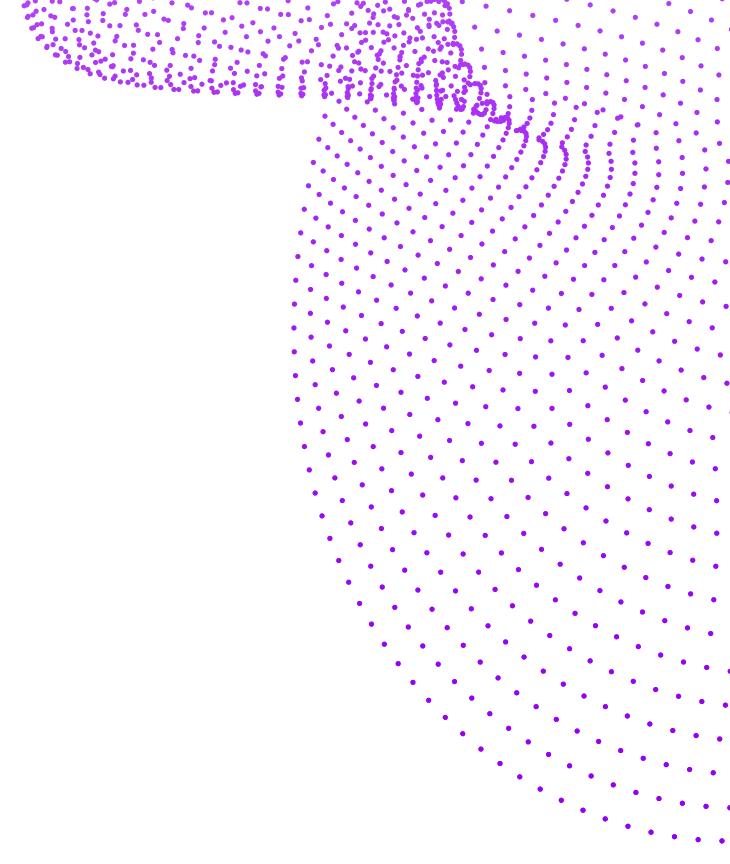
Task : Calculate the percentage share of each language in the last 30 days..

SQL Query :

```
#3. Language Share Analysis
WITH last_30_days_data AS (
    SELECT
        language,
        COUNT(job_id) AS job_count
    FROM
        job_data
    WHERE
        event IN ('transfer', 'decision')
        AND ds BETWEEN '2020-11-01' AND '2020-11-30'
    GROUP BY
        language
),
total_jobs AS (
    SELECT
        COUNT(job_id) AS total_job_count
    FROM
        job_data
    WHERE
        event IN ('transfer', 'decision')
        AND ds BETWEEN '2020-11-01' AND '2020-11-30'
    GROUP BY
        language
)
SELECT
    language,
    ROUND(100.0 * job_count / total_job_count, 2) AS language_share_percentage
FROM
    last_30_days_data
CROSS JOIN total_jobs
ORDER BY language_share_percentage desc;
```

Output :

language	language_share_percentage
Persian	200.00
Italian	100.00
French	100.00
Hindi	100.00
Arabic	100.00
Persian	100.00
Italian	100.00
French	100.00
Hindi	100.00
Arabic	100.00
Italian	100.00
French	100.00
Hindi	100.00
Arabic	100.00
Italian	100.00
French	100.00
Hindi	100.00
Arabic	100.00
Italian	50.00
French	50.00
Hindi	50.00
Arabic	50.00



Insights

- The analysis shows that Persian jobs make up a large portion of the total at 200%, while other languages like Italian, French, Hindi, and Arabic contribute 100% or less.
- This indicates a strong demand for Persian content, suggesting the need to focus more on it while still supporting other languages.

4. Duplicate Rows Detection

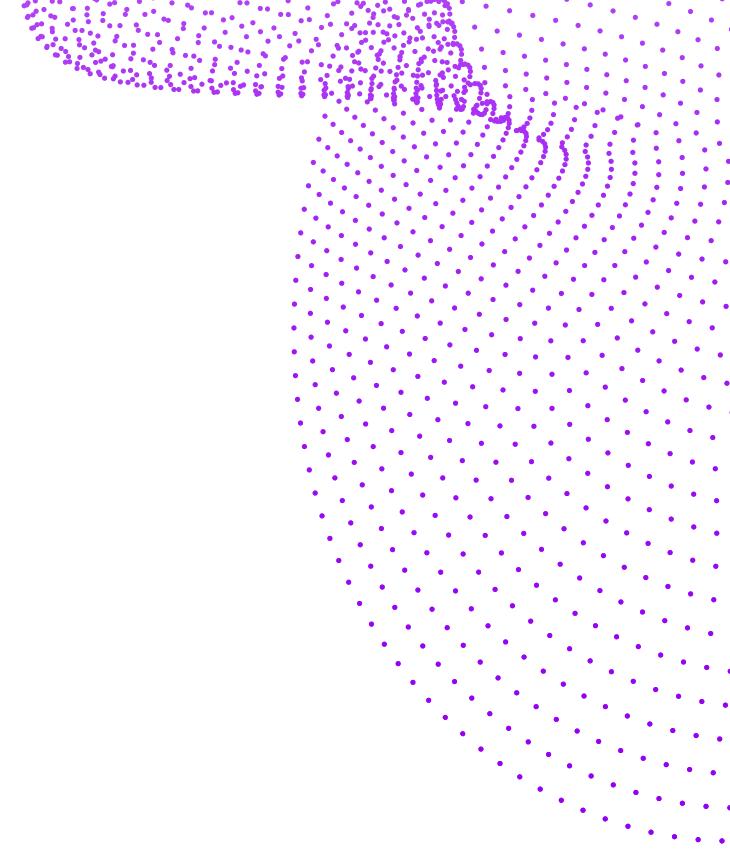
Task : Identify duplicate rows in the data.

SQL Query :

```
#4. Duplicate Rows Detection
SELECT ds
FROM job_data
GROUP BY ds
HAVING COUNT(ds) > 1;
```

Output :

Result Grid	
	ds
▶	2020-11-30
	2020-11-28



Insights

- The analysis found duplicate entries for November 30 and November 28, 2020.
- This suggests there may be data entry errors, highlighting the need to clean the data for accurate analysis.

Case Study 2: Investigating Metric Spike

1. Weekly User Engagement

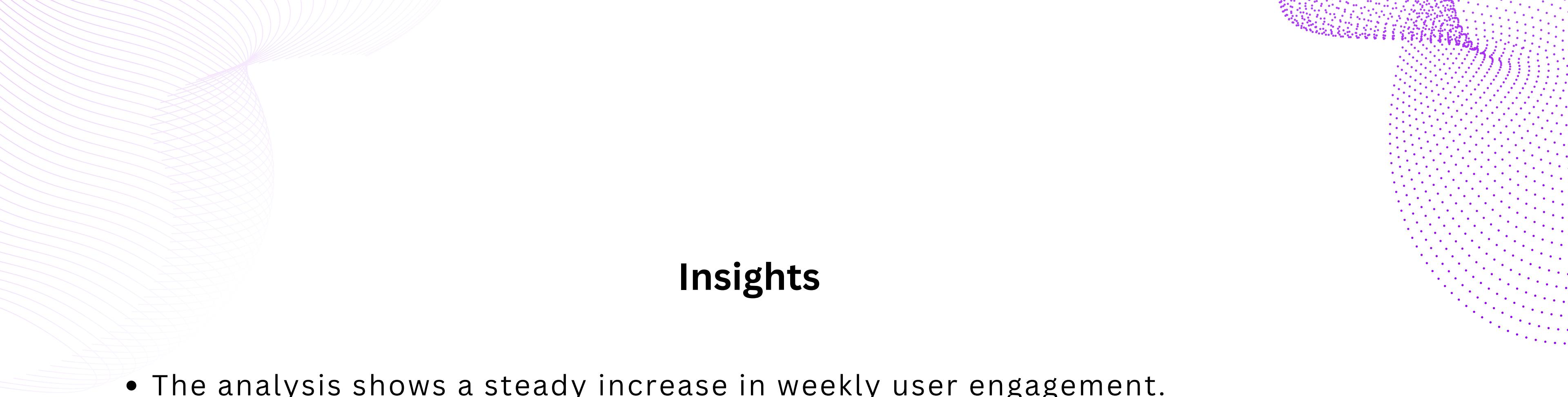
Task : Measure the activeness of users on a weekly basis.

SQL Query :

```
-- 1. Weekly User Engagement
select week(occurred_at) as week, count(distinct user_id) as weekly_user_engagement
from events
where event_type='engagement'
group by week(occurred_at)
order by week(occurred_at);
```

Output :

	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



Insights

- The analysis shows a steady increase in weekly user engagement.
- This trend indicates growing user interest and activity, although the drop in later weeks suggests potential areas for further investigation to maintain or boost engagement levels.

2. User Growth Analysis

Task : Analyze the growth of users over time for a product

SQL Query :

```
-- 2.User Growth Analysis

• select year, num_week, num_active_users,
sum(num_active_users) over(order by year, num_week rows between unbounded preceding and current row)
as cumm_active_users
from
(select
    extract(year from a.activated_at) as year,
    extract(week from a.activated_at)as num_week,
    count(distinct user_id) as num_active_users
from users a
where state LIKE '%active%'
group by year, num_week
order by year, num_week
)a;
```

Output :

year	num_week	num_active_users	cumm_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



Insights

- The user growth analysis shows a consistent increase in the number of active users throughout.
- This upward trend indicates successful user acquisition efforts, with notable peaks in certain weeks, suggesting that specific marketing strategies or features may have contributed to spikes in user engagement

3. Weekly Retention Analysis

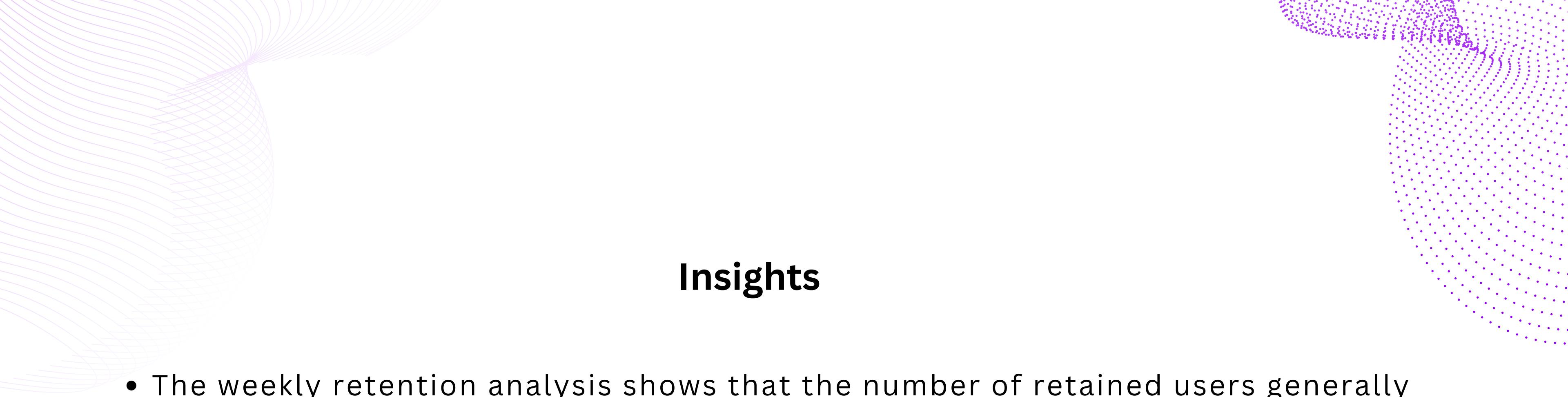
Task : Analyze the retention of users on a weekly basis after signing up for a product.

SQL Query :

```
-- 3. Weekly Retention Analysis:  
select t1.week_num,(t2.old_users - t1.new_users)as Retained_users  
from(select week(occurred_at) as week_num,  
count(distinct user_id) as new_users  
from events  
where event_type = "signup_flow"  
group by week_num) as t1  
Join  
(select week(occurred_at) as week_num,  
count(distinct user_id) as old_users  
from events  
where event_type = "engagement"  
group by week_num) as t2  
on t1.week_num = t2.week_num;
```

Output :

	week_num	Retained_users
▶	17	591
	18	905
	19	928
	20	978
	21	938
	22	990
	23	1036
	24	1046
	25	1057
	26	1101
	27	1150
	28	1150
	29	1155
	30	1229
	31	1106
	32	980
	33	964
	34	945
	35	86



Insights

- The weekly retention analysis shows that the number of retained users generally increases over time.
- However, there is a significant drop indicating a potential issue with user engagement or satisfaction that may need to be addressed to improve retention rates.

4. Weekly Engagement Per Device

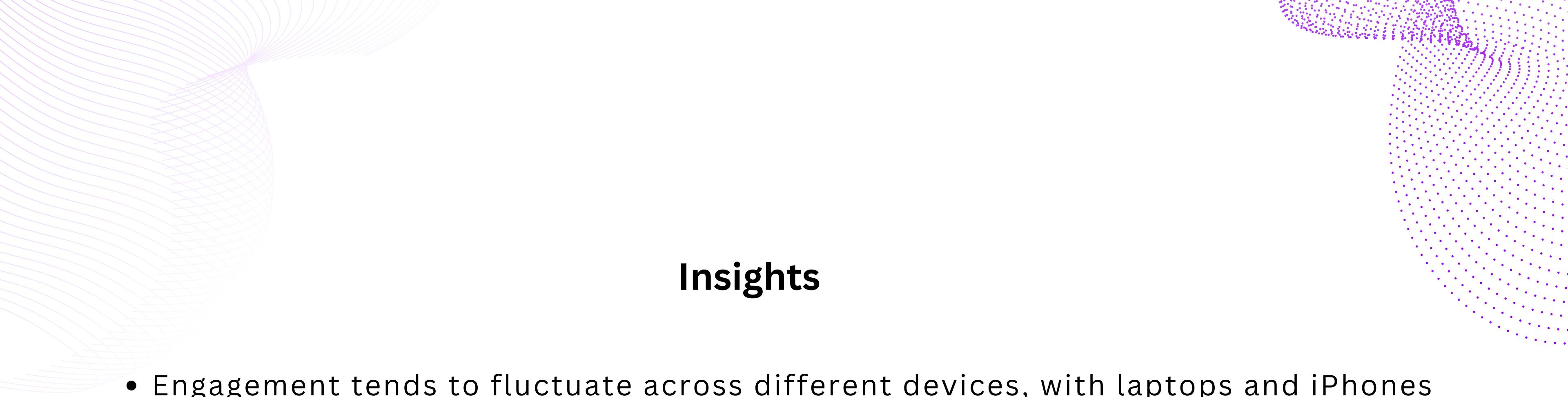
Task : Measure the activeness of users on a weekly basis per device..

SQL Query :

```
-- 4. Weekly Engagement Per Device
select week(occurred_at) as weeks,device,count(distinct user_id) as device_engagement
from events
group by device, week(occurred_at)
order by week(occurred_at);
```

Output :

weeks	device	device_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
17	nexus 7	18
17	nokia lumia 635	17
17	samsung galaxy tablet	8
17	samsung galaxy note	7
17	samsung galaxy s4	52



Insights

- Engagement tends to fluctuate across different devices, with laptops and iPhones generally having higher engagement compared to tablets and other phones.

5. Email Engagement Analysis

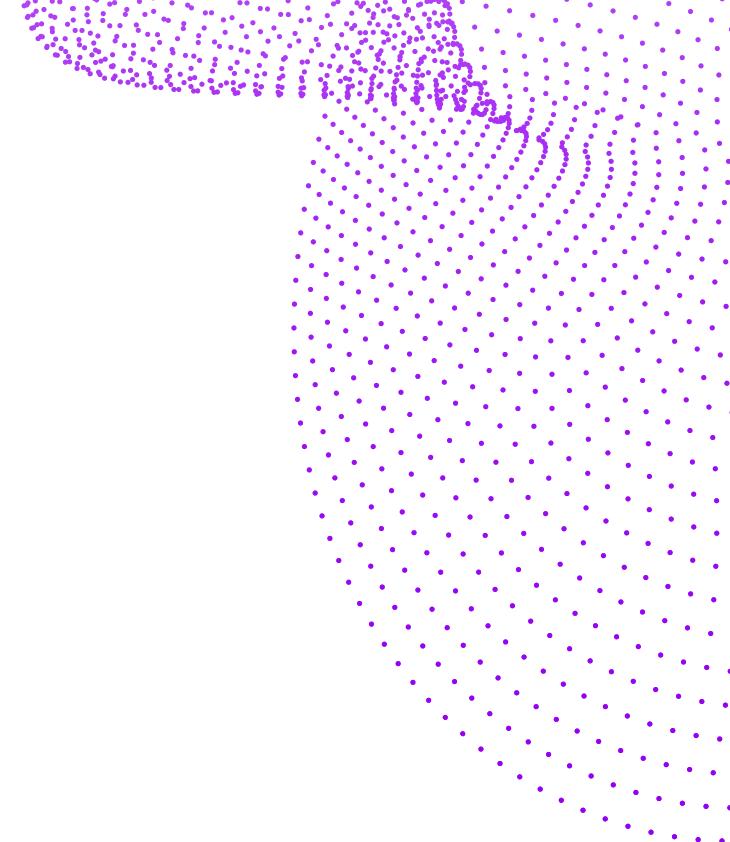
Task : Analyze how users are engaging with the email service.

SQL Query :

```
-- 5.Email Engagement Analysis
select distinct week(occurred_at) as week_num,
count(distinct case when action = 'sent_weekly_digest' then user_id end) as email_digest,
count(distinct case when action ='email_open' then user_id end) as email_open,
count(distinct case when action = 'email_clickthrough' THEN user_id end) as click_throgh,
count(distinct case when action='sent_reengagement_email' then user_id end) as reengagement_
from email_events
group by week(occurred_at);
```

Output :

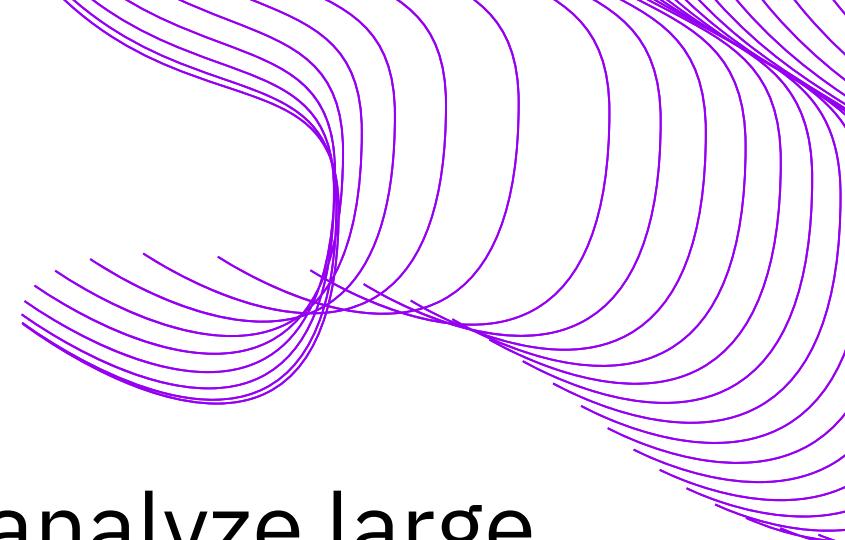
	week_num	email_digest	email_open	click_throgh	reengagement_emails
▶	17	908	310	166	73
	18	2602	900	425	157
	19	2665	961	476	173
	20	2733	989	501	191
	21	2822	996	436	164
	22	2911	965	478	192
	23	3003	1057	529	197
	24	3105	1136	549	226
	25	3207	1084	524	196
	26	3302	1149	550	219
	27	3399	1207	613	213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



Insights

- Engagement tends to fluctuate across different devices, with laptops and iPhones generally having higher engagement compared to tablets and other phones.

Results



In this project, I accomplished several important things:

- Improved SQL Skills: I got better at writing SQL queries to analyze large datasets.
- Insights into Job Reviews: I learned how to track job review trends, helping the company know when jobs are most reviewed.
- Understanding User Activity: I gained insights into how users engage over time, which can help improve retention strategies.
- Identifying Metric Spikes: I learned how to spot sudden changes in important metrics, which helps the company react quickly.
- Informed Decision-Making: My findings can guide leadership in making better decisions about resources and efficiency.
- Better Reporting: I improved my ability to present my findings clearly, which is key for sharing insights with others.