

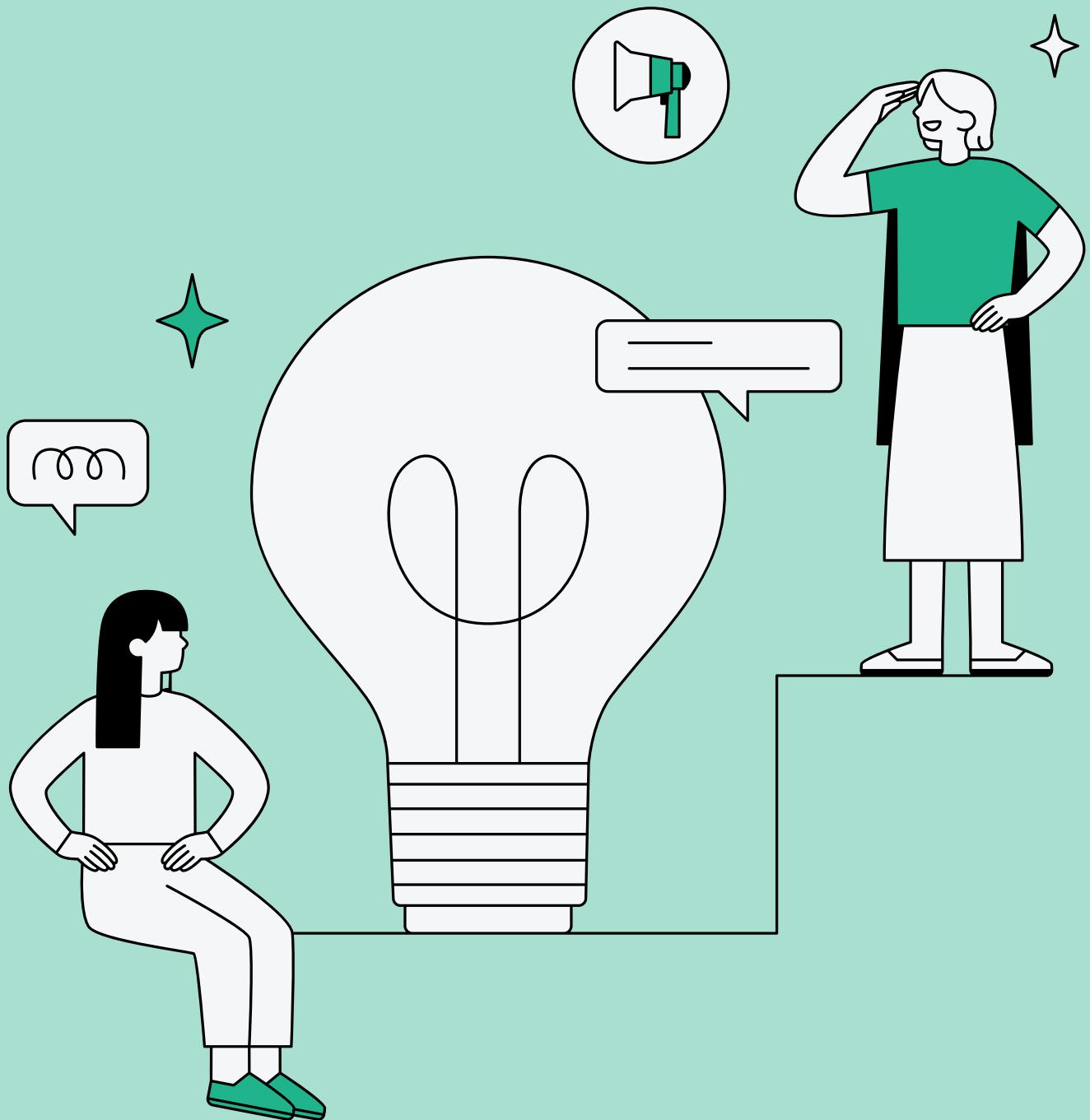
Operation Analytics and Investigating Metric Spike



Project Description

Operation Analytics is the analysis done for the complete end to end operations of a company. With the help of this, the company then finds the areas on which it must improve upon. I've worked with the ops team, support team, marketing team and help them derive insights out of the data they collect.

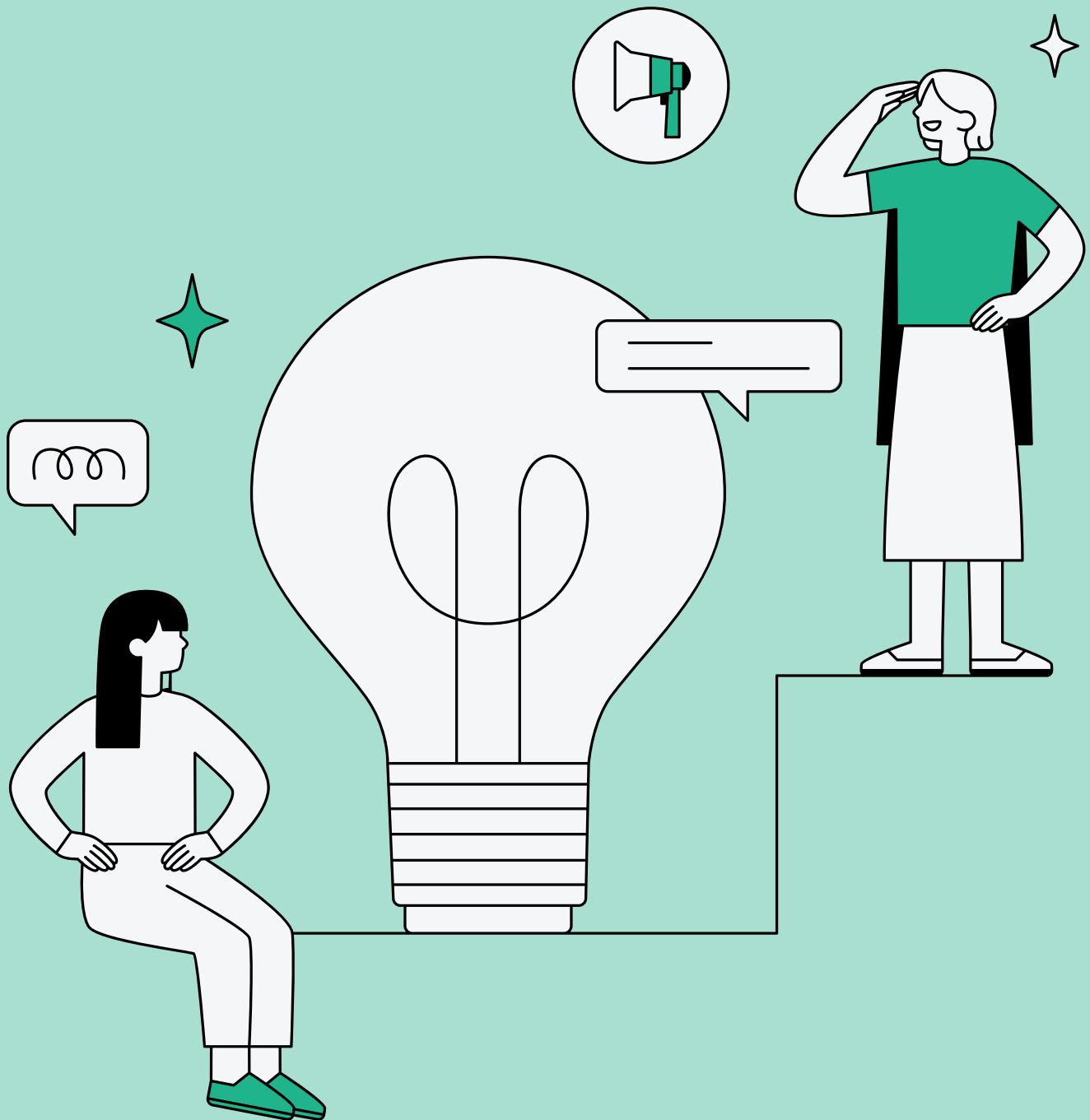
Being one of the most important parts of a company, this kind of analysis is further used to predict the overall growth or decline of a company's future.



Project Description

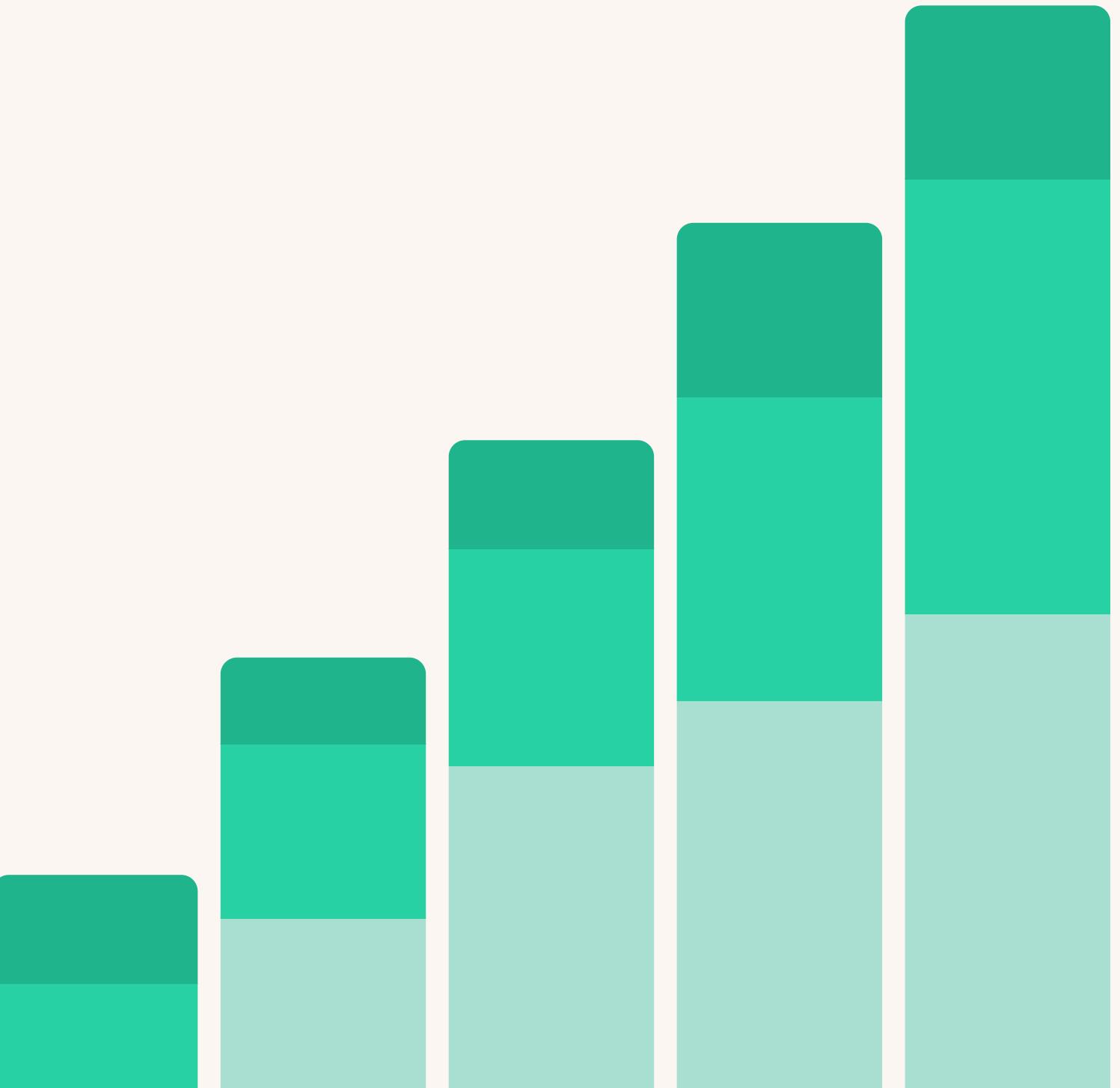
Investigating Metric Spike is also an important part of operation analytics as being a Data Analyst I've approach with the other teams and understand questions like - Why is there a dip in daily engagement Etc.

Questions like these must be answered daily and for that its very important to Investigate the Spike.



Approach

- Created the Database and Tables : Created a database and then the alter table using the structure and links provided.
- Perform Analysis: Used MySQL to perform entire analysis answering the questions asked. I went through the given dataset and then created the tables for calculating various queries.
- Additionally, I joined the data bits and structured the tables to derive business insights, fetched the required results and hence, created useful insights for the company to take calculated and planned decisions.



Tech-Stack Used

MySQL

Used MySQL Community Server
8.0.38
For creating database and
analysis of Data.

Excel

For Import and Export of data

PowerPoint

For the Presentation

Insights

Case Study 1: Job Data Analysis

Jobs Reviewed Over Time:

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

Your Task: Write an SQL query to calculate the number of jobs reviewed per hour for each day in November 2020.

```
SELECT DATE(ds) AS day, HOUR(ds) AS hour,  
COUNT(*) AS jobs_reviewed  
FROM job_data  
GROUP BY DATE(ds), HOUR(ds)  
ORDER BY day, hour;
```

| | day | hour | jobs_reviewed |
|---|------------|------|---------------|
| ▶ | 2020-11-25 | 0 | 1 |
| | 2020-11-26 | 0 | 1 |
| | 2020-11-27 | 0 | 1 |
| | 2020-11-28 | 0 | 2 |
| | 2020-11-29 | 0 | 1 |
| | 2020-11-30 | 0 | 2 |

Insights

Case Study 1: Job Data Analysis

Throughput Analysis:

Objective: Calculate the 7-day rolling average of throughput (number of events per second).

Your Task: Write an SQL query to calculate the 7-day rolling average of throughput. Additionally, explain whether you prefer using the daily metric or the 7-day rolling average for throughput, and why.

```
SELECT DATE(ds) AS day, COUNT(*) AS jobs_reviewed,  
       AVG(COUNT(*)) OVER (ORDER BY DATE(ds) ROWS BETWEEN 6 PRECEDING AND CURRENT ROW) AS rolling_avg_7_day  
FROM job_data  
GROUP BY day  
ORDER BY day;
```

| day | jobs_reviewed | rolling_avg_7_day |
|------------|---------------|-------------------|
| 2020-11-25 | 1 | 1.0000 |
| 2020-11-26 | 1 | 1.0000 |
| 2020-11-27 | 1 | 1.0000 |
| 2020-11-28 | 2 | 1.2500 |
| 2020-11-29 | 1 | 1.2000 |
| 2020-11-30 | 2 | 1.3333 |

I would prefer using the 7-day rolling average for throughput analysis in most cases because it gives a clearer picture of long-term trends. Daily metrics can be too volatile and influenced by random variations, such as holidays, weekends, or specific one-off events. The rolling average provides a more reliable indicator of underlying performance trends, making it easier to draw actionable insights over time.

Insights

Case Study 1: Job Data Analysis

Language Share Analysis:

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

Your Task: Write an SQL query to calculate the percentage share of each language over the last 30 days.

```
SELECT language, COUNT(*) AS language_count,
ROUND((COUNT(*) / (SELECT COUNT(*) FROM job_data)) * 100, 2) AS percentage_share
FROM job_data
GROUP BY language
ORDER BY percentage_share DESC;
```

| language | language_count | percentage_share |
|----------|----------------|------------------|
| Persian | 3 | 37.50 |
| English | 1 | 12.50 |
| Arabic | 1 | 12.50 |
| Hindi | 1 | 12.50 |
| French | 1 | 12.50 |
| Italian | 1 | 12.50 |

Insights

Case Study 1: Job Data Analysis

Duplicate Rows Detection:

Objective: Identify duplicate rows in the data.

Your Task: Write an SQL query to display duplicate rows from the job_data table.

```
SELECT job_id, COUNT(*) AS occurrence_count  
FROM job_data  
GROUP BY job_id  
HAVING COUNT(*) > 1  
ORDER BY occurrence_count DESC;
```

| job_id | occurrence_count |
|--------|------------------|
| 23 | 3 |

```
SELECT actor_id, COUNT(*) AS occurrence_count  
FROM job_data  
GROUP BY actor_id  
HAVING COUNT(*) > 1  
ORDER BY occurrence_count DESC;
```

| actor_id | occurrence_count |
|----------|------------------|
| 1003 | 2 |

Insights

Case Study 2: Investigating Metric Spike

Weekly User Engagement:

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

Your Task: Write an SQL query to calculate the weekly user engagement.

```
SELECT week(occurred_at) as week_num,  
       COUNT(DISTINCT user_id)  
  FROM events  
 WHERE event_type= 'engagement'  
 GROUP BY week_num;
```

| | week_num | COUNT(DISTINCT user_id) |
|---|----------|-------------------------|
| ▶ | 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 |

- Highest user week : 30
- Lowest user week : 35

Insights

Case Study 2: Investigating Metric Spike

Weekly User Engagement:

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

Your Task: Write an SQL query to calculate the weekly user engagement.

```
• WITH weekly AS (
    SELECT year(created_at) as year,
    week(created_at) AS week_num,COUNT(user_id) AS num_users
    FROM users
    GROUP BY year, week_num
    ORDER BY year, week_num
)
SELECT year, week_num, num_users,
    SUM(num_users) OVER (ORDER BY year, week_num) AS cumulative_users
FROM weekly;
```

| year | week_num | num_users | cumulative_users |
|------|----------|-----------|------------------|
| 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 |

The 33rd week of 2014 saw highest number of users.

The lowest was on 35th week of 2014.

Insights

Case Study 2: Investigating Metric Spike

Weekly Retention Analysis:

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

Your Task: Write an SQL query to calculate the weekly retention of users based on their sign-up cohort

```
WITH Signup_Cohorts AS (
  SELECT user_id, EXTRACT(WEEK FROM occurred_at) AS signup_week
  FROM events
  WHERE event_name = 'complete_signup' AND event_type = 'signup_flow'
),
Engagements AS (
  SELECT user_id, EXTRACT(WEEK FROM occurred_at) AS engagement_week
  FROM events
  WHERE event_type = 'engagement'
)
SELECT S.signup_week,
  COUNT(DISTINCT S.user_id) AS users_signed_up, COUNT(DISTINCT E.user_id) AS users_retained
FROM
  Signup_Cohorts S
  LEFT JOIN
  Engagements E ON S.user_id = E.user_id AND E.engagement_week > S.signup_week
GROUP BY
  S.signup_week
ORDER BY
  S.signup_week;
```

| signup_week | users_signed_up | users_retained |
|-------------|-----------------|----------------|
| 17 | 72 | 64 |
| 18 | 163 | 130 |
| 19 | 185 | 157 |
| 20 | 176 | 146 |
| 21 | 183 | 150 |
| 22 | 196 | 166 |
| 23 | 196 | 161 |
| 24 | 229 | 177 |
| 25 | 207 | 177 |
| 26 | 201 | 155 |
| 27 | 222 | 178 |
| 28 | 215 | 174 |
| 29 | 221 | 174 |
| 30 | 238 | 185 |
| 31 | 193 | 144 |
| 32 | 245 | 180 |

85.5% of users retained in week 25th.

Insights

Case Study 2: Investigating Metric Spike

Weekly Engagement Per Device:

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

Your Task: Write an SQL query to calculate the weekly engagement per device.

```
select extract(week from occurred_at) as week, extract(year from occurred_at) as year, device,  
count(distinct user_id) as total from events  
where event_type = 'engagement'  
group by 1,2,3  
order by 1,2,3;
```

| week | year | device | total |
|------|------|------------------------|-------|
| 33 | 2014 | nokia lumia 635 | 27 |
| 33 | 2014 | nexus 7 | 30 |
| 33 | 2014 | 2014 5 | 70 |
| 33 | 2014 | nexus 10 | 23 |
| 33 | 2014 | macbook pro | 312 |
| 33 | 2014 | macbook air | 133 |
| 33 | 2014 | mac mini | 32 |
| 33 | 2014 | lenovo thinkpad | 191 |
| 33 | 2014 | kindle fire | 14 |
| 33 | 2014 | iphone 5s | 65 |
| 33 | 2014 | iphone 5 | 110 |
| 33 | 2014 | iphone 4s | 35 |
| 33 | 2014 | ipad mini | 28 |
| 33 | 2014 | ipad air | 40 |
| 33 | 2014 | htc one | 19 |
| 33 | 2014 | hp pavilion desktop | 38 |
| 33 | 2014 | dell inspiron notebook | 110 |
| 33 | 2014 | dell inspiron desktop | 37 |

MacBook Pro has the highest sales in every week of the year

Insights

Case Study 2: Investigating Metric Spike

Email Engagement Analysis:

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

Your Task: Write an SQL query to calculate the email engagement metrics.

```
1 • select
2   100*sum(case when email_category = 'email_open' then 1 else 0 end)/sum(case when email_category='email_sent' then 1 else 0 end) as open_rate,
3   100*sum(case when email_category='email_click' then 1 else 0 end)/sum(case when email_category='email_sent' then 1 else 0 end) as click_rate
4   from
5   (select *, case
6     when action in ('sent_weekly_digest','sent_reengagement_email') then 'email_sent'
7     when action in ('email_open') then 'email_open'
8     when action in ('email_clickthrough') then 'email_click'
9   end as email_category
10  from email_events) a;
```

Out of total emails sent, around 33.5% of them were opened and only 15% of those emails were clicked.

| open_rate | click_rate |
|-----------|------------|
| 33.5834 | 14.7899 |

Results

- Less than 0.01 jobs were reviewed each hour of the day throughout the month of November.
- 7Day rolling average is best for analysis.
- The Persian Language has the highest share among other languages.
- The weekly user engagement is higher in 30th week.
- 30th and 35th week having the highest and lowest of user activity engagement respectively.
- Maximum retained users were only retained for a week, after it is dropping. In 25th signup week retention is highest.
- Users using 'MacBook Pro' is highest among all.
- Out of the total emails sent, around 33.5% of them were opened and only 15% of emails were clicked.



Thank
you very
much!

