

PROJECT 3

Operation Analytics and Investigating Metric Spike

Advanced SQL

Analyzed by:

Parag Jyoti Nath

PROJECT OVERVIEW

- To-do :
 - Analyzing a company's end-to-end operations to identify areas for improvement within the company.
- Goal :
 - Derive valuable insights from the data collected by various teams, such as operations, support, and marketing.
 - Investigate metric spikes - Understanding and explaining sudden changes in key metrics, such as a dip in daily user engagement or a drop in sales etc.

TECH-STACK USED

- In this project, SQL and MySQL Workbench is being used as the tool to analyze this data to answer questions posed by different departments within the company and provide valuable insights that can help improve the company's operations and understand sudden changes in key metrics.

PROJECT APPROACH

Case Study 1: Job Data Analysis

- Jobs Reviewed Over Time
- Throughput Analysis
- Language Share Analysis
- Duplicate Rows Detection

Case Study 2: Investigating Metric Spike

- Weekly User Engagement
- User Growth Analysis
- Weekly Retention Analysis
- Weekly Engagement Per Device
- Email Engagement Analysis

JOB DATA ANALYSIS

CASE STUDY 1

CREATING DATABASE

```
1 • create database jobs;
2 • use jobs;
3
4 • create table job_data (
5     date_ date,
6     job_id int,
7     actor_id int,
8     event varchar(20),
9     language varchar(20),
10    time_spent int,
11    org char(1)
12 );
13
```

```
14 • insert into job_data (date_, job_id, actor_id, event, language, time_spent, org ) values
15     ('2020-11-30', 21, 1001, 'skip', 'English', 15, 'A'),
16     ('2020-11-30', 22, 1006, 'transfer', 'Arabic', 25, 'B'),
17     ('2020-11-29', 23, 1003, 'decision', 'Persian', 20, 'C'),
18     ('2020-11-28', 23, 1005, 'transfer', 'Persian', 22, 'D'),
19     ('2020-11-28', 25, 1002, 'decision', 'Hindi', 11, 'B'),
20     ('2020-11-27', 11, 1007, 'decision', 'French', 104, 'D'),
21     ('2020-11-26', 23, 1004, 'skip', 'Persian', 56, 'A'),
22     ('2020-11-25', 20, 1003, 'transfer', 'Italian', 45, 'C');
23
```



```
select * from job_data;
```



Result Grid		Filter Rows:		Export:		Wrap Cell Content:	
	date_	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

1. JOBS REVIEWED OVER TIME

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

```
26 • select date_ as day,  
27    count(job_id) as jobs_reviewed,  
28    sum(time_spent) as minutes,  
29    count(job_id)/sum(time_spent)*60 as job_reviewed_per_hour  
30 from job_data  
31 group by day  
32 order by day;
```



	day	jobs_reviewed	minutes	job_reviewed_per_hour
▶	2020-11-25	1	45	1.3333
	2020-11-26	1	56	1.0714
	2020-11-27	1	104	0.5769
	2020-11-28	2	33	3.6364
	2020-11-29	1	20	3.0000
	2020-11-30	2	40	3.0000

2. THROUGHPUT ANALYSIS

- Write an SQL query to calculate the 7-day rolling average of throughput (number of events per second). Additionally, explain whether you prefer using the daily metric or the 7-day rolling average for throughput, and why.

```
34  -- step 1 - Calculate the daily total events and total time spent
35  •  select date_ as day,
36         count(job_id) as total_events,
37         sum(time_spent) as total_time
38         from job_data
39         group by date_;
```



	day	total_events	total_time
▶	2020-11-30	2	40
	2020-11-29	1	20
	2020-11-28	2	33
	2020-11-27	1	104
	2020-11-26	1	56
	2020-11-25	1	45

```
41  -- step 2 - Calculate the 7-day rolling average of events per second
42 •  select temp.day,
43      avg(temp.total_events / temp.total_time) as rolling_avg
44  from
45      (select date_ as day,
46         count(job_id) as total_events,
47         sum(time_spent) as total_time
48       from job_data
49       group by date_
50      ) temp
51  group by temp.day
52  order by temp.day;
```



	day	rolling_avg
▶	2020-11-25	0.02220000
	2020-11-26	0.01790000
	2020-11-27	0.00960000
	2020-11-28	0.06060000
	2020-11-29	0.05000000
	2020-11-30	0.05000000

Daily Metric vs. 7-Day Rolling Average

- Daily Metric:
- Advantages:
 - Suitable for short term trends - days.
 - Better for day-to-day analysis and immediate detection of spikes or drops in a trend.
- Disadvantages:
 - Harder to identify long-term trends.
- 7-Day Rolling Average:
- Advantages:
 - Suitable for long term trends - weeks.
 - Reduces the impact of outliers on any single day.
- Disadvantages:
 - Hides short-term issues that need immediate attention.

3. LANGUAGE SHARE ANALYSIS

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

```
54  -- calculate the percentage share of each language over the last 30 days.
55  •  select language,
56     (count(language)/total_lang.total_count*100) as percent_share
57  from job_data
58  join
59     (select count(language) as total_count
60      from job_data) as total_lang
61  group by language, total_lang.total_count
62  order by percent_share desc;
```



	language	percent_share
▶	Persian	37.5000
	English	12.5000
	Arabic	12.5000
	Hindi	12.5000
	French	12.5000
	Italian	12.5000

Persian has the highest share and all others have equal share.

4. DUPLICATE ROWS DETECTION

- Write an SQL query to display duplicate rows from the job_data table.

```
64 -- display all rows from the job_data table.
65 • select date_, job_id, actor_id, event, language, time_spent, org, count(*)
66 from job_data
67 group by date_, job_id, actor_id, event, language, time_spent, org;
```



	date_	job_id	actor_id	event	language	time_spent	org	count(*)
▶	2020-11-30	21	1001	skip	English	15	A	1
	2020-11-30	22	1006	transfer	Arabic	25	B	1
	2020-11-29	23	1003	decision	Persian	20	C	1
	2020-11-28	23	1005	transfer	Persian	22	D	1
	2020-11-28	25	1002	decision	Hindi	11	B	1
	2020-11-27	11	1007	decision	French	104	D	1
	2020-11-26	23	1004	skip	Persian	56	A	1
	2020-11-25	20	1003	transfer	Italian	45	C	1

```
64 -- filter the groups to include only those with more than one occurrence, i.e., duplicate rows.
65 • select date_, job_id, actor_id, event, language, time_spent, org, count(*)
66 from job_data
67 group by date_, job_id, actor_id, event, language, time_spent, org
68 having count(*) > 1;
```



	date_	job_id	actor_id	event	language	time_spent	org	count(*)

No Duplicate rows have been detected.

INVESTIGATING METRIC SPIKE

CASE STUDY 2

1. WEEKLY USER ENGAGEMENT

- Write an SQL query to calculate the weekly user engagement.

```
7 • select
8     users.user_id,
9     year(events.occurred_at) as year,
10    week(events.occurred_at) as week,
11    count(distinct events.event_name) as total_events,
12    count(distinct email_events.action) as total_emails
13 from users
14 left join events on users.user_id = events.user_id
15 left join email_events on users.user_id = email_events.user_id
16 group by
17     users.user_id,
18     year(events.occurred_at),
19     week(events.occurred_at)
20 order by
21     users.user_id, year, week;
```



	user_id	year	week	total_events	total_emails
▶	0	NULL	NULL	0	2
	3	NULL	NULL	0	0
	4	2014	19	3	3
	4	2014	20	4	3
	4	2014	21	13	3
	4	2014	22	4	3
	4	2014	23	5	3
	4	2014	24	4	3
	4	2014	25	4	3
	4	2014	26	4	3
	4	2014	27	4	3
	6	NULL	NULL	0	0
	7	NULL	NULL	0	0
	8	2014	17	2	3
	8	2014	18	6	3
	8	2014	19	3	3
	8	2014	20	5	3
	8	2014	30	4	3
	11	2014	24	6	3
	11	2014	25	6	3

Null values mean these users IDs from users table have no records in events and email_events tables.

2. USER GROWTH ANALYSIS

- Write an SQL query to calculate the user growth for the product.

```
25 • select
26     year(created_at) as year,
27     month(created_at) as month,
28     count(user_id) as new_users
29 from users
30 group by
31     year(created_at),
32     month(created_at)
33 order by year, month;
```



	year	month	new_users
▶	2013	1	160
	2013	2	160
	2013	3	150
	2013	4	181
	2013	5	214
	2013	6	213
	2013	7	284
	2013	8	316
	2013	9	330
	2013	10	390
	2013	11	399
	2013	12	486
	2014	1	552
	2014	2	525
	2014	3	615
	2014	4	726
	2014	5	779
	2014	6	873
	2014	7	997
	2014	8	1031

Highest growth, i.e. number of new users is observed in August, 2014 and lowest in March, 2013.

3. WEEKLY RETENTION ANALYSIS

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

```
36 • select s.signup_year, s.signup_week,  
37        count(distinct s.user_id) as retained_users_count  
38 from (  
39     select user_id,  
40            year(created_at) as signup_year,  
41            week(created_at) as signup_week  
42     from users  
43 ) s  
44 left join (  
45     select user_id,  
46            year(occurred_at) as activity_year,  
47            week(occurred_at) as activity_week  
48     from events  
49     group by user_id, year(occurred_at), week(occurred_at)  
50 ) a  
51 on s.user_id = a.user_id  
52    and a.activity_year = s.signup_year  
53    and a.activity_week >= s.signup_week  
54    and a.activity_week < s.signup_week + 1  
55    -- only counting activities in the same week as sign-up  
56 group by s.signup_year, s.signup_week  
57 order by s.signup_year, s.signup_week;
```



	signup_year	signup_week	retained_users_count
▶	2013	0	23
	2013	1	30
	2013	2	48
	2013	3	36
	2013	4	30
	2013	5	48
	2013	6	38
	2013	7	42
	2013	8	34
	2013	9	43
	2013	10	32
	2013	11	31
	2013	12	33
	2013	13	39
	2013	14	35
	2013	15	43
	2013	16	46
	2013	17	49
	2013	18	44
	2013	19	57
	2013	20	39
	2013	21	49
	2013	22	54
	2013	23	50

4. WEEKLY ENGAGEMENT PER DEVICE

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

```
74 • select
75     year(occurred_at) as year,
76     week(occurred_at) as week,
77     device,
78     count(user_id) as total_events
79 from events
80 group by
81     year(occurred_at), week(occurred_at), device
82 order by
83     year, week, device;
```



	year	week	device	total_events
▶	2014	17	acer aspire desktop	69
	2014	17	acer aspire notebook	207
	2014	17	amazon fire phone	84
	2014	17	asus chromebook	254
	2014	17	dell inspiron desktop	188
	2014	17	dell inspiron notebook	506
	2014	17	hp pavilion desktop	134
	2014	17	htc one	192
	2014	17	ipad air	331
	2014	17	ipad mini	208
	2014	17	iphone 4s	219
	2014	17	iphone 5	715
	2014	17	iphone 5s	476
	2014	17	kindle fire	57
	2014	17	lenovo thinkpad	801
	2014	17	mac mini	60
	2014	17	macbook air	493
	2014	17	macbook pro	1527
	2014	17	nexus 10	145
	2014	17	nexus 5	385
	2014	17	nexus 7	181
	2014	17	nokia lumia 635	130
	2014	17	samsung galaxy tablet	71

Highest engagement (3649) was observed from MacBook pro on week 31, 2014 and lowest (5) from Dell Inspiron desktop in week 35, 2014.

5. EMAIL ENGAGEMENT ANALYSIS

Write an SQL query to calculate the email engagement metrics.

```
89 • select
90     year(occurred_at) as year,
91     week(occurred_at) as week,
92     count(user_id) as total_email_actions
93 from email_events
94 group by
95     year(occurred_at), week(occurred_at)
96 order by
97     year, week;
```



	year	week	total_email_actions
▶	2014	17	1457
	2014	18	4101
	2014	19	4287
	2014	20	4435
	2014	21	4443
	2014	22	4578
	2014	23	4813
	2014	24	5040
	2014	25	5029
	2014	26	5242
	2014	27	5461
	2014	28	5561
	2014	29	5614
	2014	30	5950
	2014	31	5811
	2014	32	5852
	2014	33	6198
	2014	34	6390
	2014	35	127

Highest engagement (6390) was observed in week 34, 2014 and lowest (127) in week 35, 2014.

RESULTS

- **Data-Driven Insights:** By analyzing user engagement and email interactions, the project provided data-driven insights into user behavior, which helps in understanding patterns, preferences, and areas of engagement.
- **Improved User Retention:** The weekly retention analysis can be used to understand retention trends to brainstorm strategies to increase user retention.
- **Optimized Engagement Strategies:** The engagement metrics by device and email interactions helped in identifying which devices and email strategies are most effective. This knowledge can be used to improve product experiences and targeting specific users.
- **Informed Decision-Making:** The insights gained from these analyses support informed decision-making by providing a clearer picture of user engagement and retention. This helps the company in strategy planning, resource allocation, and targeted marketing.

THANK YOU