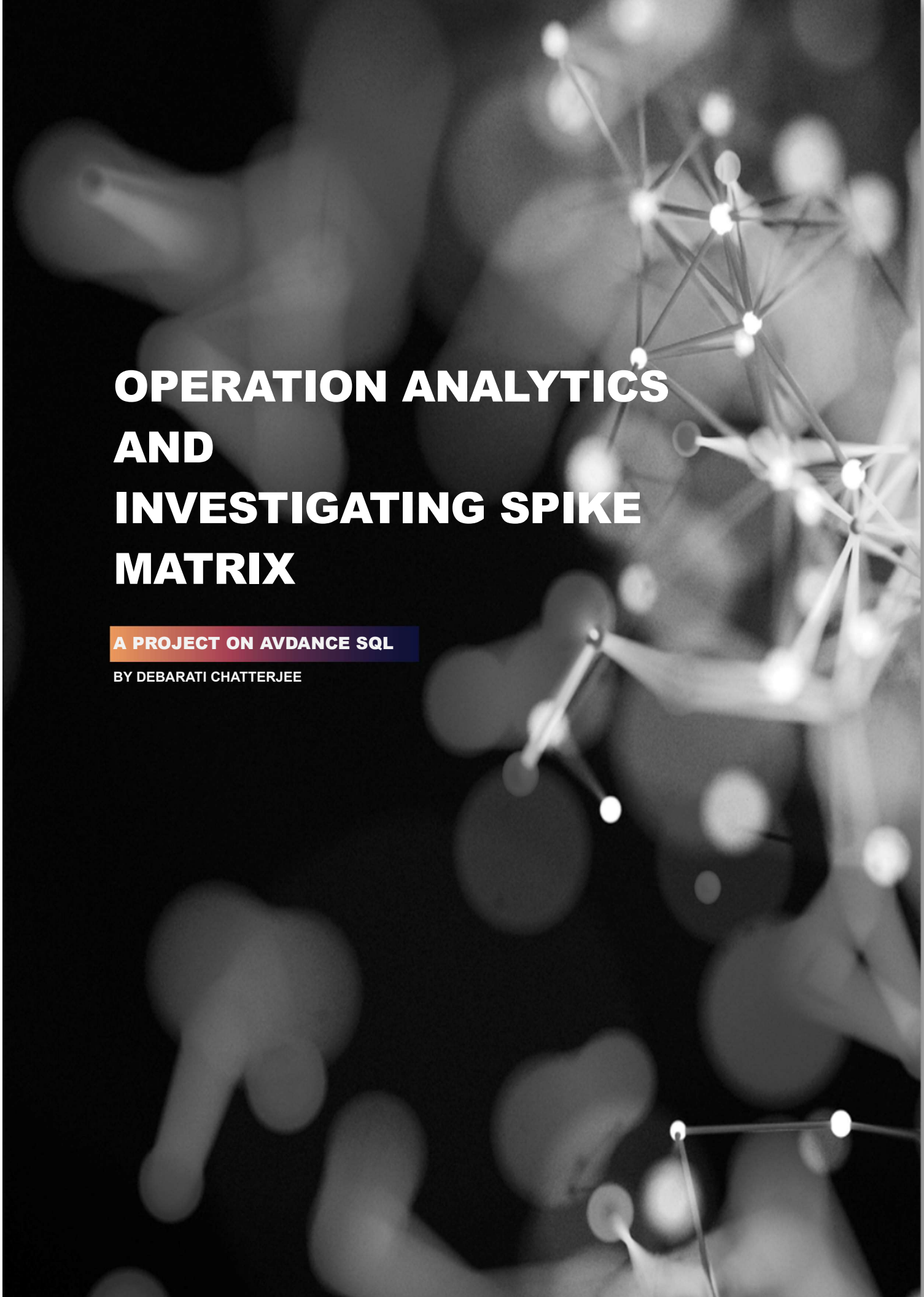

OPERATION ANALYTICS AND INVESTIGATING SPIKE MATRIX

A PROJECT ON AVDANCE SQL

BY DEBARATI CHATTERJEE

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. You work closely with the ops team, support team, marketing team, etc 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 fortune. It means better automation, better understanding between cross-functional teams, and more effective workflows.
- Investigating metric spike is also an important part of operation analytics as being a Data Analyst you must be able to understand or make other teams understand questions like- Why is there a dip in daily engagement? Why have sales taken a dip? Etc. Questions like these must be answered daily and for that it's very important to investigate metric spike.
- I am working for a company like Microsoft designated as Data Analyst Lead and is provided with different data sets, tables from which I have to derive certain insights out of it and answer the questions asked by different departments provide a detailed report for the below two operations mentioning the answers for the related questions:

APPROACH

- I have reviewed the data structure and following schema while understanding the given questions.
- I have created a database and then the tables using the structure and links provided.
- I used SQL for entire analysis answering the questions asked where I use Advance SQL concept like Window Functions, CTE, Multiple Table Join and find valuable insights for the company.

TECH-STACK USED

MySQL Workbench 8.0: Creating databases, tables and finding insights using SQL queries on the tables.

Microsoft Power Point: Presenting the detailed report for company.

JOB DATA (CASE STUDY 1)

Below is the structure of the table with the definition of each column:

Table-1: job_data

- **job_id:** unique identifier of jobs
- **actor_id:** unique identifier of actor
- **event:** decision/skip/transfer
- **language:** language of the content
- **time_spent:** time spent to review the job in seconds
- **org:** organization of the actor
- **ds:** date in the yyyy/mm/dd format. It is stored in the form of text and we use presto to run. no need for date function

Using the dataset above I have answer the following questions .

QUESTION-1:

Number of jobs reviewed: Amount of jobs reviewed over time.
Task: Calculate the number of jobs reviewed per hour per day for November 2020?

SQL QUERY:

```
select ds,
round((count(distinct job_id)/sum(time_spent))*3600,2) as jobs_reviewed_per_hour_per_day
from jobdata
group by ds
order by ds;
```

OUTPUT:

ds	jobs_reviewed_per_hour_per_day
2020-11-25	80.00
2020-11-26	64.29
2020-11-27	34.62
2020-11-28	218.18
2020-11-29	180.00
2020-11-30	180.00

QUESTION-2:

Throughput: It is the no. of events happening per second.
Task: Let's say the above metric is called throughput. Calculate 7 day rolling average of throughput? For throughput, do you prefer daily metric or 7-day rolling and why?

SQL QUERY:

```
with event_table as
(select ds,
round((count(distinct event)/sum(time_spent)),2) as event_per_second_daily
from jobdata
group by ds
order by ds
)
select ds, event_per_second_daily,
avg(event_per_second_daily) over(order by ds rows between 6 preceding and current row)
as 7_days_rolling_avg
from event_table
group by ds
order by ds ;
```

OUTPUT:

ds	event_per_second_daily	7_days_rolling_avg
2020-11-25	0.02	0.020000
2020-11-26	0.02	0.020000
2020-11-27	0.01	0.016667
2020-11-28	0.06	0.027500
2020-11-29	0.05	0.032000
2020-11-30	0.05	0.035000

QUESTION-3:

Percentage share of each language: Share of each language for different contents.
Task: Calculate the percentage share of each language in the last 30 days?

SQL QUERY:

```
select language,
round(count(*)*100/total,2)   as percentage
from jobdata cross join
(select count(*) as total
 from jobdata) as totaldata
group by language
order by count(*) desc;
```

OUTPUT:

language	percentage
Persian	37.50
English	12.50
Arabic	12.50
Hindi	12.50
French	12.50
Italian	12.50

QUESTION-4:

Duplicate rows: Rows that have the same value present in them.
Your task: Let's say you see some duplicate rows in the data. How will you display duplicates from the table?

SQL QUERY:

```
select * from
(
select *,
row_number()over(partition by job_id) as rownum
from jobdata
)a
where rownum>1;
```

OUTPUT:

ds	job_id	actor_id	event	language	time_spent	org	rownum
2020-11-28	23	1005	transfer	Persian	22	D	2
2020-11-26	23	1004	skip	Persian	56	A	3

CASE STUDY 1 INSIGHTS:

- November 28th 2020 has highest number of job review per hour.
- 7 day rolling average is best for throughput as it is easier to check the event per second.
- The Persian Language had the highest share (37.5%) while other languages have equal share of 12.5%.
- There is 2 duplicate rows in the data

INVESTING METRIC SPIKE (CASE STUDY 2)

The structure of the tables with the definition of each column :

- **Table-1:** users
  This table includes one row per user, with descriptive information about that user's account.
- **Table-2:** events
  This table includes one row per event, where an event is an action that a user has taken. These events include login events, messaging events, search events, events logged as users progress through a signup funnel, events around received emails.
- **Table-3:** email_events
  This table contains events specific to the sending of emails. It is similar in structure to the events table above.

Using the dataset I have to answer the following questions.

QUESTION 1:

User Engagement: To measure the activeness of a user. Measuring if the user finds quality in a product/service.
Task: Calculate the weekly user engagement?

SQL QUERY:

```sql
select extract(week from occurred_at) as week_of_year,
count(distinct user_id) as total_user
from events
where event_type ="engagement"
group by 1
```

OUTPUT:

week_of_year	total_user
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

QUESTION-2:

User Growth: Amount of users growing over time for a product.
Task: Calculate the user growth for product?

SQL QUERY:

```sql
with active_table as
(select extract(week from created_at) as created_week, count(user_id) as active_users
from users
where state ="active"
group by 1
order by 1)

select created_week, active_users,
round(((active_users/lag(active_users,1) over (order by created_week)-1)*100),2) as growth_rate
from active_table;
```

OUTPUT:

created_week	active_users	growth_rate
0	106	NULL
1	156	47.17
2	157	0.64
3	149	-5.10
4	160	7.38
5	181	13.13
6	173	-4.42
7	167	-3.47
8	163	-2.40
9	176	7.98
10	186	5.68
11	161	-13.44
12	181	12.42
13	206	13.81
14	197	-4.37
15	207	5.08
16	225	8.70

QUESTION-3:

Weekly Retention: Users getting retained weekly after signing-up for a product.
Task: Calculate the weekly retention of users-sign up cohort?

SQL QUERY:

```sql
select extract(week from occurred_at) as week, count(user_id) as engaged_user,
 sum(case when retention_week >0 then 1 else 0 end) as
retained_user
from
(
select distinct a.user_id,
 a.sign_up_week,
 b.engagement_week,
 b.engagement_week - a.sign_up_week as retention_week
from
(
select distinct user_id, extract(week from occurred_at) as sign_up_week
from events
where event_type = "signup_flow" and event_name ="complete_signup"
and extract(week from occurred_at)=18
)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)c
left join
events using(user_id)
group by 1
order by 1;
```

QUTPUT:

week	engaged_user	retained_user
19	10618	8605
20	7431	6234
21	6014	5197
22	3445	2940
23	2688	2371
24	2048	1773
25	2430	2109
26	1741	1570
27	2952	2630
28	1592	1454
29	1402	1263
30	1925	1732
31	1577	1429
32	1341	1204
33	884	769
34	483	408
35	96	84

QUESTION-4:

Weekly Engagement: To measure the activeness of a user. Measuring if the user finds quality in a product/service weekly.
Task: Calculate the weekly engagement per device?

SQL QUERY:

```
select extract(week from occurred_at) as week_of_year,device,
count(distinct user_id) as user_count
from events
where event_type ="engagement"
group by 1,2
order by 1;
```

OUTPUT:

week_of_year	device	user_count
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

QUESTION-5:

Email Engagement: Users engaging with the email service.
Your task: Calculate the email engagement metrics?

SQL QUERY:

```
select
100.0 * sum(case when email_category = 'email_opened' then 1 else 0 end)
 /sum(case when email_category = 'email_sent' then 1 else 0 end)
as opening_rate,
100.0 * sum(case when email_category = 'email_clicked' then 1 else 0 end)
 /sum(case when email_category = 'email_sent' then 1 else 0 end)
as clicking_rate
from
(
select *,
case when action in ('sent_weekly_digest', 'sent_reengagement_email')
 then 'email_sent'
 when action in ('email_open')
 then 'email_opened'
 when action in ('email_clickthrough')
 then 'email_clicked'
end as email_category
from email_event
) as email engagement;
```

OUTPUT:

opening_rate	clicking_rate
33.58339	14.78989

CASE STUDY 2 INSIGHTS:

• The weekly user engagement is highest in 30th week of 2014.

• 1st week has highest user growth rate(47%) where 52st wee has lowest growth rate (-54%) compared to previous week.

• 19th week of 2014 has Maximum retained users and most users were only retained for a week, the retention rates dropped weekly.

• Users who had 'MacBook Pro‘ has the highest engagement per week.

• Out of the total emails sent, around 34% of them were opened and only 15% of those emails were clicked.

CONCLUSION

The primary outcomes of the study included the identification of evaluated occupations and their distribution across languages, the determination of retention rates, and the identification of retained users via an in-depth survey. Predefined assumptions were used in the analysis. SQL is a critical ability for anybody working in a data-driven environment. Furthermore, this project assisted me in gaining an understanding of numerous variables that are critical for the firm to function for a long time and expand. Also, this project taught me how to use sophisticated SQL concepts such as Windows Functions, Sub-queries, CTE etc. I comprehended how the real-world industry operates. It aided me in understanding SQL principles. I learnt how to ask the appropriate questions given the circumstances.

THANK YOU