OPERATION ANALYTICS AND INVESTIGATION MATRIC SPIKE

By- ANJALI RAJPUT

PROJECT DESCRIPTION

THE MAIN PURPOSE OF THE PROJECT IS TO UNDERSTAND SUDDEN CHANGES IN THE KEY INDICATORS AND DISCOVER MEANINGFUL INSIGHTS THAT WILL ENRICH THE OPERATION OF THE BUSINESS. WE LOOK AT THIS INFORMATION FROM THE FOLLOWING PERSPECTIVE:

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

APPROACH:

- 1. READ THE PROVIDES DATA DESCRIPTION AND UNDERSTAND THE PROBLEM.
- 2. CAREFULLY GO THROUGH THE DATA AND MAKE SURE YOU UNDERSTAND THE PROVIDED VARIABLES AND ATTRIBUTE.
- 3. I IMPORTED FILES BY USING MYSQL WORKBENCH INTO THE OPERATIONAL ANALYTICS DATABASES AND STARTED MAKING QUERIES WITH THE QUESTIONS PRESENTED TO GET THE DESIRED OUTCOMES.
- 4. I WRITE QUERIES AND EXECUTE THEM IN ORDER TO GET RESULTANT OUTCOME.
- 5. FINALLY, TAKE SCREENSHOTS AND DOCUMENT THEM TOGETHER.

TECH USE IN THIS PROJECT:





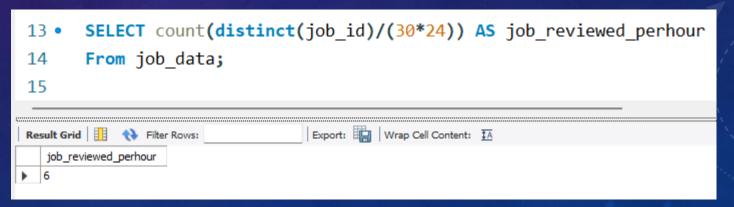


CASE STUDY 1: JOB DATA ANALYSIS

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.



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 ds , job nums, total time,
sum(job nums) over (order by ds rows between 6 preceding and current row)/sum(total time) as 7day rolling average
select ds ,count(job_id) as job_nums,sum(time_spent) as total_time
from job data where ds >='2020-11-01' and ds <='2020-11-30'
group by ds
                                 Export: Wrap Cell Content: IA
                        7day_rolling_average
                        0.0288
                        0.1515
                        0.3000
```

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, lang count,
sum(lang_count)/(sum(lang_count)over(order by language rows between unbounded preceding and unbounded following )) * 100.0 as perc_langu
select language, count(language) as lang count
from job_data
group by language
                                  Export: Wrap Cell Content: IX
   lang_count perc_language
             12,50000
             12,50000
              12,50000
              12,50000
             37,50000
```

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.

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.

```
34 • select

35  extract(year FROM occurred_at) as year,

36  extract(week from occurred_at) as weeknum,

37  count(distinct user_id) AS user_engagement

38  FROM

39  events

40  group by year, weeknum

41  order by year, weeknum;

Result Grid  Filter Rows: Export: W
```

USER GROWTH ANALYSIS:

OBJECTIVE: ANALYZE THE GROWTH OF USERS OVER TIME FOR A PRODUCT.

YOUR TASK: WRITE AN SQL QUERY TO CALCULATE THE USER GROWTH FOR THE PRODUCT.

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.

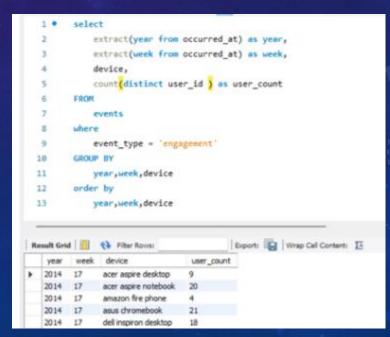
```
SELECT user id, count(user id),
sum(case when retention week = 1 then 1 else 0 end) as week 1.
sum(case when retention week = 2 then 1 else 0 end) as week 2
sum(case when retention_week = 3 then 1 else 0 end) as week_3,
sum(case when retention week = 4 then 1 else 0 end) as week 4,
sum(case when retention_week = 5 then 1 else 0 end) as week 5,
sum(case when retention_week = 6 then 1 else 0 end) as week_6,
sum(case when retention week = 7 then 1 else 0 end) as week 7,
sum(case when retention_week = 8 then 1 else 0 end) as week_8,
sum(case when retention_week = 9 then 1 else 0 end) as week_9
SELECT
   a.user_id,
    a.signup_week,
    b.engagement week - a.signup week as retention week
```

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.

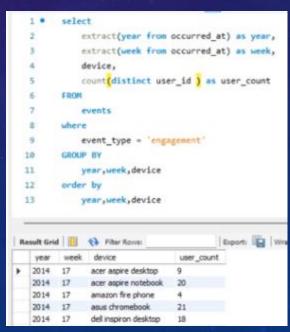


EMAIL ENGAGEMENT ANALYSIS:

OBJECTIVE: ANALYZE HOW USERS ARE ENGAGING WITH THE EMAIL SERVICE.

YOUR TASK: WRITE AN SQL QUERY TO CALCULATE THE

EMAIL ENGAGEMENT METRICS.



THANK YOU