PROJECT DESCRIPTION:

To analyse and give suitable solutions for the given queries through the structured query language.

APPROACH:

Understanding the given question and thinking logically and analytically to solve it.

TECH-STACK USED:

The software used in these projects is MYSQL creates a table database and creates a query in SQL language for achieving the solution.

INSIGHTS:

By using SQL queries for solving the table data it is easy to separate specific data needed from the table and obtain the specific value for a problem which will be helpful for analytical purposes.

RESULT:

By solving the given problems, I have learned to think differently while shorting and aggregating particular data and removing unwanted data to get the required data from the table.

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

SELECT

ds

count(job_id) as number of jobs

time spent/60*60 as per hour

from project-1 table

group by ds

B.Throughput: It is the no. of events happening per second.
Your 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?

SELECT*

AVG(time_spent) over (partition by event ORDERBY ds rows between 6 preceding and current row as _7 days rolling average

From project-1 table

->For the moving values 7-day rolling metric is used where there is a need for predicting the next preceding value.

->In throughput for calculating the next value, 7-day rolling average is the best way to predict the 7th value

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

```
SELECT

Top 30

Count *100/sum(count) As language percentage

From(

Select *

Count (language ) over ( partition by language order by ds) as count

From project-1 table

)
```

D.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?

```
SELECT *

ROLL_NUMBER() over(partition by actor_id order by actor_id) as repeated

From project-1 table

Where repeated>1
```

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

```
SELECT

DISTINCT User_id

ROLL_NUMBER() Week(occred_at) as week

Count (event_type ) as product/services engaged

Where event_type=enagagement

From
```

Event.csv

Group by user_id,week

Order by user_id

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

SELECT

COUNT(DISTINCT user_id) as growth of product by month

ROLL_NUMBER() Month(occured_at) as month

growth of product by month – LAG(growth of product by month) order by (month) as user growth for product

From

Event.csv

GROUP BY month

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

Select

Count(DISTINCt user_id) as number of users per week

ROLL_NUMBER () Week(ordered_at) as week

number of users per week – LAG(number of users per week) order by(week) ad weekly retention

From

Event.csv

Group by week

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

Select

device

ROLL_NUMBER()Week(ordered_at) as week

Count (event_type) as engagement per device

WHERE event_type = engagement

From

Event.csv

Group by device

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

Select

Count(DISTINCT user_id) as id

DIFFERENCE(id , active) as email engagement

From(

Select

Count (event_type) as active

From

Event.csv

Where event_type =sighup_flow OR null

)