Case study 1

Name : krisha borana

1. Calculate the number of jobs reviewed per hour per day for November 2020?

```
SELECT ds AS Jobs_Reviewed_Date, COUNT(job_id) AS Jobs_Count_per_day,
(time_spent)/3600 AS Hours_Spent_per_day
FROM job_data
where ds >='2020-11-01'  and ds <='2020-11-30'
GROUP BY Jobs_Reviewed_Date;
```

2. Calculate 7 day rolling average of throughput?

We will be calculating the rolling average of the time spent by the User with respect to the number of Jobs he/she searched for.

```
WITH throughput AS
(SELECT ds AS Jobs_Reviewed_Date, COUNT(job_id) AS Jobs_Count_per_day,
SUM(time_spent) AS Time_Spent
FROM job_data
WHERE ds >='2020-11-01'  and ds <='2020-11-30'
GROUP BY ds)

SELECT *,
SUM(Jobs_Count_per_day) OVER (ORDER BY Jobs_Reviewed_Date ROWS BETWEEN 6 PRECEDING
AND CURRENT ROW) / SUM(Time_Spent) OVER (ORDER BY Jobs_Reviewed_Date ROWS BETWEEN
6 PRECEDING AND CURRENT ROW) as 7_day_rolling
FROM throughput;
```

3. Calculate the percentage share of each language in the last 30 days?

```
with cte as
(SELECT language, count(job_id) AS number_of_jobs
FROM job_data
GROUP BY language)

SELECT *, ROUND(number_of_jobs*100/(SELECT sum(number_of_jobs) FROM cte),2) as
Percent_Share
FROM cte;
```

4. Display duplicate rows present in the table.

```
SELECT ds, job_id, actor_id, org, count(*) AS occurences
```

```
FROM job_data
GROUP BY ds, job_id, actor_id, org
HAVING occurences>1;
```

Case study 2

1. Calculate the weekly user engagement

We will be extracting week from the event occur date and then will group it and will see the number of distinct users in a week asnd thus we will get the weekly engaged users count.

```
SELECT EXTRACT(WEEK FROM occurred_at) AS Week_Number, COUNT(DISTINCT(user_id)) AS
Weekly_Engagement_Number
FROM yammer_events
GROUP BY Week_Number;
```

2. Calculate the weekly user growth

We will be using a temporary table by using WITH Clause, by extracting the week and year from the profile creation date and will check only those users that are active.

```
With Weekly_data as (
SELECT EXTRACT(year from created_at) AS Year_Number,
       EXTRACT(week from created_at) AS Week_Number,
       COUNT(DISTINCT(user_id)) AS Weekly_Active_Users
FROM yammer_users
WHERE state = 'active'
GROUP BY Year_Number, Week_Number)

SELECT *,
SUM(Weekly_Active_Users) OVER (ORDER BY Year_Number, Week_Number asc)  as
Total_Users
FROM Weekly_data;
```

3. Calculate the weekly engagement per device

We will calculate this by again extracting the year and week from the event occuring date and GROUP it by the device with the condition that the event type is engagement.

```
SELECT EXTRACT(year FROM occurred_at) AS Year_Name,
       EXTRACT(week FROM occurred_at) AS Week_Num,
       device,
       COUNT(DISTINCT(user_id)) as Device_num
FROM yammer_events
WHERE event_type = 'engagement'
GROUP BY 1,2,3
ORDER BY 1,2,3 ASC;
```

4. Calculate the email engagement metrics

Since we are not being told about any specific metric so we will calculate the Email Clicks Rate and Email Opening Rate

```
WITH Email_Data AS
(
SELECT *,
       CASE WHEN (action = 'email_clickthrough') THEN 'Email Clicked'
            WHEN (action = 'email_open') THEN 'Email Opened'
            ELSE 'Email Sent'
        END AS action_details
FROM yammer_emails)

SELECT
ROUND((100.0 *SUM(CASE WHEN action_details IN ('Email Clicked') THEN 1 ELSE 0
END)/SUM(CASE WHEN action_details IN ('Email Sent') THEN 1 ELSE 0 END)),2) AS
Email_Click_Rate,
ROUND((100.0 *SUM(CASE WHEN action_details IN ('Email Opened') THEN 1 ELSE 0
END)/SUM(CASE WHEN action_details IN ('Email Sent') THEN 1 ELSE 0 END)),2) AS
Email_Open_Rate
FROM Email_Data;
```

5. Calculate the Customer Retention post sign up

```
SELECT COUNT(user_id) AS Total_Users,
       SUM(CASE WHEN Retention_Activity =1 THEN 1 ELSE 0 END) AS Total_Retained
FROM
(
SELECT  Account_Created_Users.user_id, Account_Created_Users.SignUp_Week,
Engagement_By_Users.engagement_week,
        (Engagement_By_Users.engagement_week - Account_Created_Users.SignUp_Week)
AS Retention_Activity
FROM
(
(SELECT DISTINCT(user_id), EXTRACT(week FROM occurred_at) AS SignUp_Week FROM
yammer_events
WHERE event_type = 'signup_flow'
AND event_name = 'complete_signup'
) AS Account_Created_Users
LEFT JOIN
(
SELECT user_id, DATE(occurred_at) AS Engagement_Date, EXTRACT(week FROM
occurred_at) AS Engagement_Week
FROM yammer_events
WHERE event_type = 'engagement') Engagement_By_Users
ON Account_Created_Users.user_id = Engagement_By_Users.user_id
)
ORDER BY Account_Created_Users.user_id
) AS All_Data
```