SQL Queries used for Operation Analytics and Investigating Metric Spike Project

```
create database trainity_project3;
use trainity_project3;
```

########################## Job Data Case Study ##############################

```
create table job_data3( job_id int, actor_id int, eve varchar(30), time_spent int, org
varchar(100), ds date, lang varchar(30)
);
select * from job_data3;
insert into job_data3(job_id,actor_id,eve,time_spent,org,ds,lang)
values
(21,1001,'skip',15,'A','2020-11-30','English'),
(22,1006,'transfer',25,'B','2020-11-30','Arabic'),
(23,1003,'decision',20,'C','2020-11-29','Persian'),
(23,1005,'transfer',22,'D','2020-11-28','Persian'),
(25,1002,'decision',11,'B','2020-11-28','Hindi'),
(11,1007,'decision',104,'D','2020-11-27','French'),
(23,1004,'skip',56,'A','2020-11-26','Persian'),
(20,1003,'transfer',45,'C','2020-11-25','Italian');
commit;
```

#Task 1: Calculate the number of jobs reviewed per hour per day for November 2020?

```
select ds, count(job_id)/(30*24) as Num_of_jobs, round(sum(time_spent)/3600, 2) as
spent_per_hour from job_data3 where ds between '2020-11-01' and '2020-11-30'
group by ds;
```

#Task 2: It is the no. of events happening per second. Calculate 7 day rolling average of throughput?

```
#For throughput, do you prefer daily metric or 7-day rolling and why?
select a.*,
avg(events_sum) over(partition by job_id order by ds, ds rows between 6 preceding and current
row) as rolling_avg from
```

```
(select job_id,ds,sum(time_spent) as events_sum from job_data3
group by job_id,ds) a;
```

#Task 3: Calculate the percentage share of each language in the last 30 days?

```
select lang,time_spent, time_spent * 100 /(select sum(time_spent) from
job_data3) as perc_per_lang from job_data3 group by lang;
```

#Task 4: Let's say you see some duplicate rows in the data. How will you display duplicates from the table?

```
select * from job_data3
group by eve,lang
having count(job_id) > 1;
```

###################### Investigating Metric Spike ############################

```
create table users1(
user_id int, created_at
varchar(255), company_id
int, lang varchar(255),
activated_at
varchar(255), state
varchar(255)
);
```

```
create table eventss(
user_id int, occured_at
varchar(255), event_type
varchar(255),
event_name
varchar(255), location
varchar(255), device
```

varchar(255), user_type

int

);

create table email_event(

user_id int, occured_at

varchar(255), `action`

varchar(255), user_type

int

);

############################ TASKS ##################################

 # Task 1: Write an SQL query to calculate the weekly user engagement.

```
select count(distinct(user_id)) as Users ,
week(date_format(str_to_date(occured_at,'%d-%m-%Y'),'%Y-%m-%d')) as
Week_number from eventss group by 2;
```

 # Task 2: Write an SQL query to calculate the user growth for the product.

```
select
Year_num, Week_num, Users ,  sum(Users) over(rows between unbounded preceding
and current row) as User_growth from( select
year(date_format(str_to_date(created_at,'%Y-%m-%d'),'%Y-%m-%d')) as Year_num,
week(date_format(str_to_date(created_at,'%Y-%m-%d'),'%Y-%m-%d')) as Week_num,
count(distinct(user_id)) as Users from users1 where state = 'active' group by 1,2 order by
1,2) a;
```

 # Task 3: Write an SQL query to calculate the weekly retention of users based on their sign-up cohort.

```
select count(e.user_id),week(date_format(str_to_date(occured_at,'%d-%m-%Y'),'%Y-%m-%d')) as
Week_number
```

```
from eventss e join users1 u on e.user_id = u.user_id where
e.event_name = 'complete_signup' and u.state = 'active' group
by 2;
```

 # Task 4: Write an SQL query to calculate the weekly engagement per device.

```
select device, count(user_id) as User_count,
week(date_format(str_to_date(occured_at,'%d-%m-%Y'),'%Y-%m-%d')) as
Week_number from eventss group by 1,3 order by 3;
```

 # Task 5: Write an SQL query to calculate the email engagement
metrics.

```
select email_action,count(*)
from ( select * , case when action = 'email_clickthrough' then
'email_clicked'
when action = 'email_open' then 'email_opened'
                    else 'email_sent' end as email_action
from email_event) a
group by email_action;
```