

**Operation Analytics and Investigating Metric Spike**

**Case Study 1 (Job Data)**

**Table-1:**job\_data

CREATE DATABASE investigating;

USE investigating;

CREATE TABLE job\_data

(

ds DATE,

job\_id INT NOT NULL,

actor\_id INT NOT NULL,

event VARCHAR(15) NOT NULL,

language VARCHAR(15) NOT NULL,

time\_spent INT NOT NULL,

org CHAR(2)

);

INSERT INTO job\_data (ds, job\_id, actor\_id, event, language, time\_spent, org)

VALUES ('2020-11-30',21, 1001, 'skip', 'English', 15, 'A'),

('2020-11-30', 22, 1006, 'transfer', 'Arabic', 25, 'B'),

('2020-11-29', 23, 1003, 'decision', 'Persian', 20, 'C'),

('2020-11-28', 23, 1005,'transfer', 'Persian', 22, 'D'),

('2020-11-28', 25, 1002, 'decision', 'Hindi', 11, 'B'),

('2020-11-27', 11, 1007, 'decision', 'French', 104, 'D'),

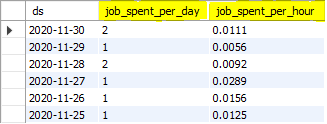
('2020-11-26', 23, 1004, 'skip', 'Persian', 56, 'A'),

('2020-11-25', 20, 1003, 'transfer', 'Italian', 45, 'C');

table job\_data;

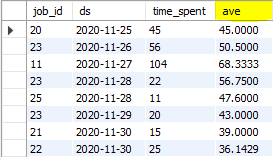
**A. Number of jobs reviewed:**

select ds,count(job\_id) as job\_spent\_per\_day, sum(time\_spent)/3600 as job\_spent\_per\_hour from job\_data where ds >='2020-11-01' and ds <='2020-11-30' group by ds;



**B. Throughput:**

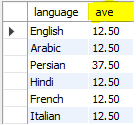
select job\_id,ds,time\_spent,avg(time\_spent) over(order by ds rows between 6 preceding and current row) as ave from job\_data order by ds;



**I prefer both daily metric and 7-days rolling because daily metric gives daily calculation which is easy to calculate and 7-days rolling values gives extreme values. Both datas are needed and useful.**

**C. Percentage share of each language:**

select language,round(count(language)\*100/8,2) as ave from job\_data where ds >='2020-11-01' and ds <='2020-11-30' group by language;



**D. Duplicate rows:**

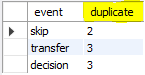
SELECT job\_id,count(\*) as duplicate FROM job\_data GROUP BY job\_id HAVING duplicate > 1;



SELECT ds,count(\*) as duplicate FROM job\_data GROUP BY ds HAVING duplicate > 1;



SELECT event,count(\*) as duplicate FROM job\_data GROUP BY event HAVING duplicate > 1;



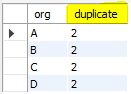
SELECT language,count(\*) as duplicate FROM job\_data GROUP BY language HAVING duplicate > 1;



SELECT actor\_id,count(\*) as duplicate FROM job\_data GROUP BY actor\_id HAVING duplicate > 1;

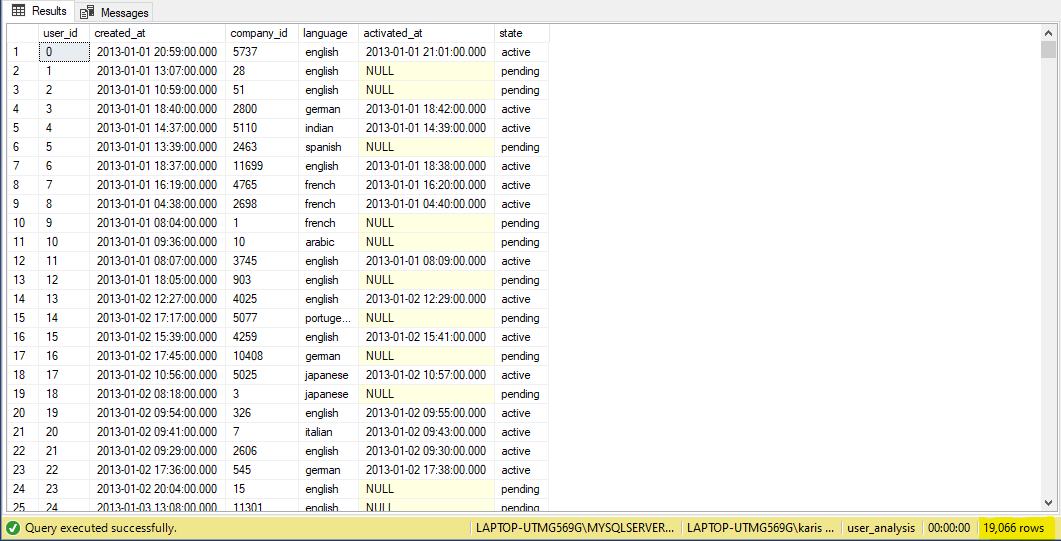


SELECT org,count(\*) as duplicate FROM job\_data GROUP BY org HAVING duplicate > 1;

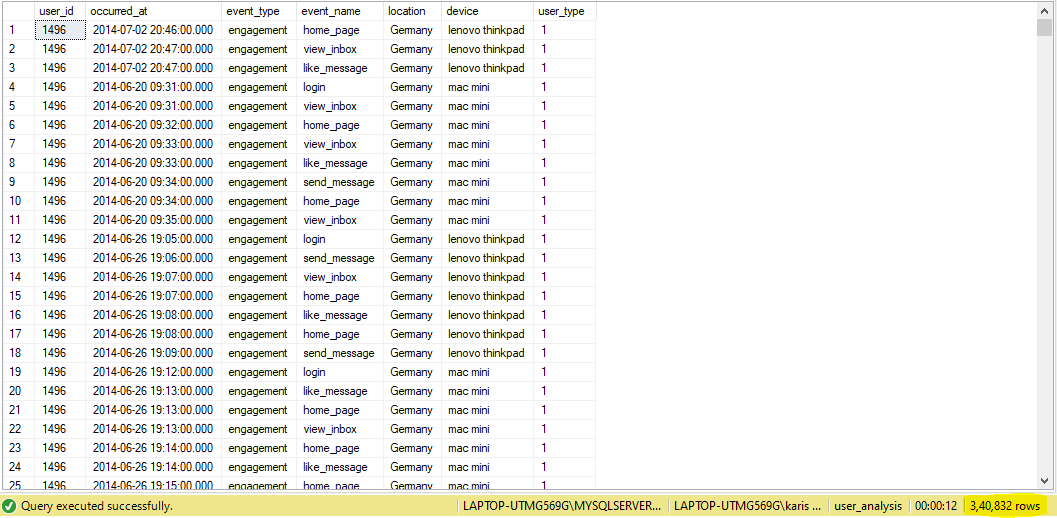


**Case Study 2 (Investigating metric spike)**

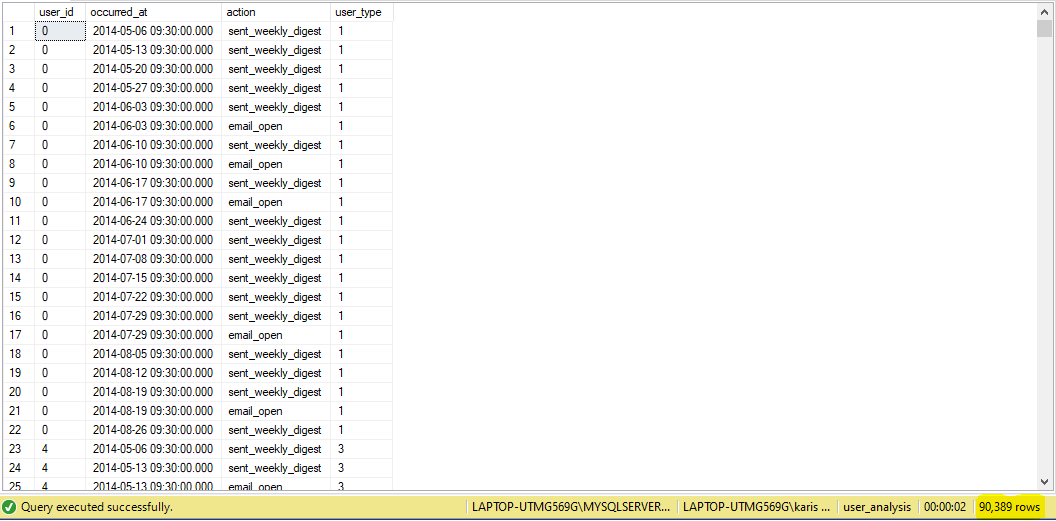
**Table-1:**users



**Table-2:**events



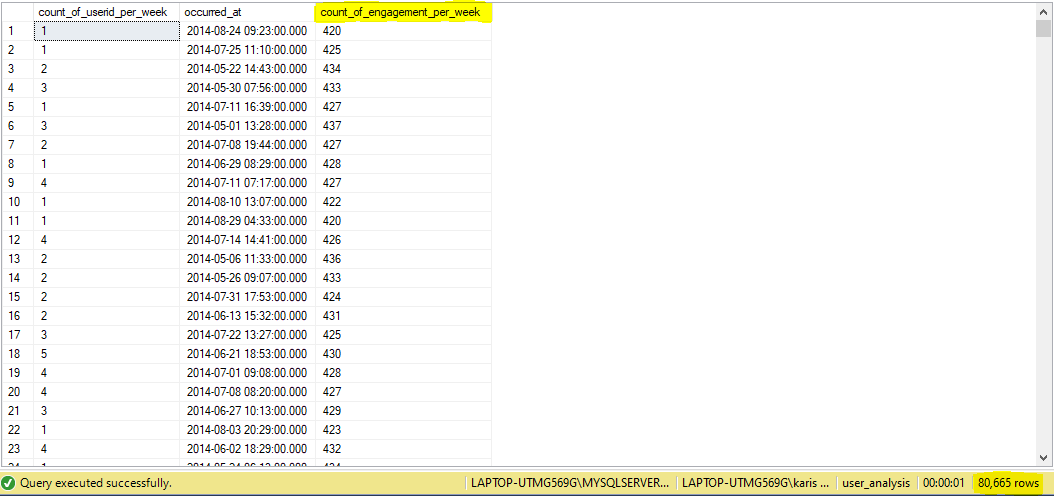
**Table-3:**email\_events



**A. User Engagement:**

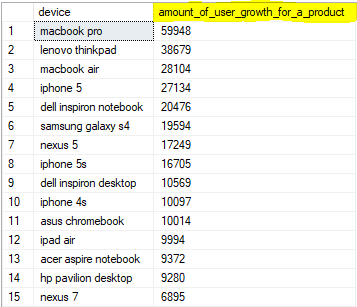
select count(distinct user\_id) as count\_of\_userid\_per\_week,

occurred\_at,(DATEDIFF(week,occurred\_at,getdate())) as count\_of\_engagement\_per\_week from events group by occurred\_at order by count\_of\_engagement\_per\_week desc



**B. User Growth:**

select device,count(device) as amount\_of\_user\_growth\_for\_a\_product from events group by device order by amount\_of\_user\_growth\_for\_a\_product desc





**C: Weekly Retention:**

select action,count(action) as weekly\_retention\_after\_signup from email\_events where action='sent\_weekly\_digest' group by action

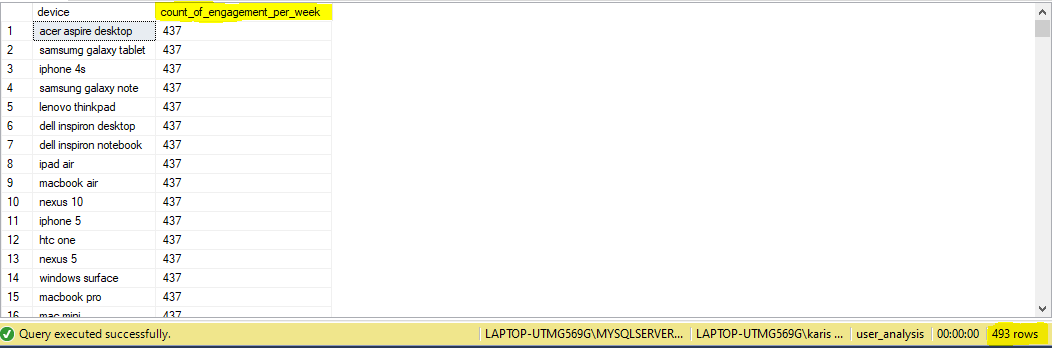


**D. Weekly Engagement:**

select distinct device,

DATEDIFF(week,occurred\_at,getdate()) as count\_of\_engagement\_per\_week from events

group by occurred\_at,device order by count\_of\_engagement\_per\_week desc



**E. Email Engagement:**

select count(distinct events.user\_id) as email\_engagement\_metrics,events.event\_type,email\_events.action from events,email\_events where event\_type='engagement' and action='email\_open' group by event\_type,action

