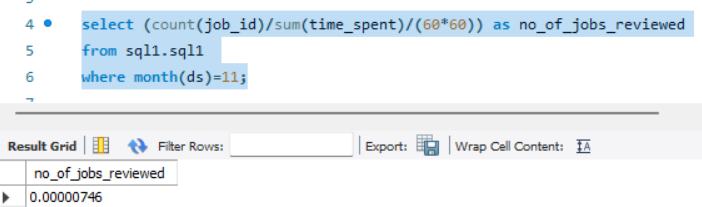
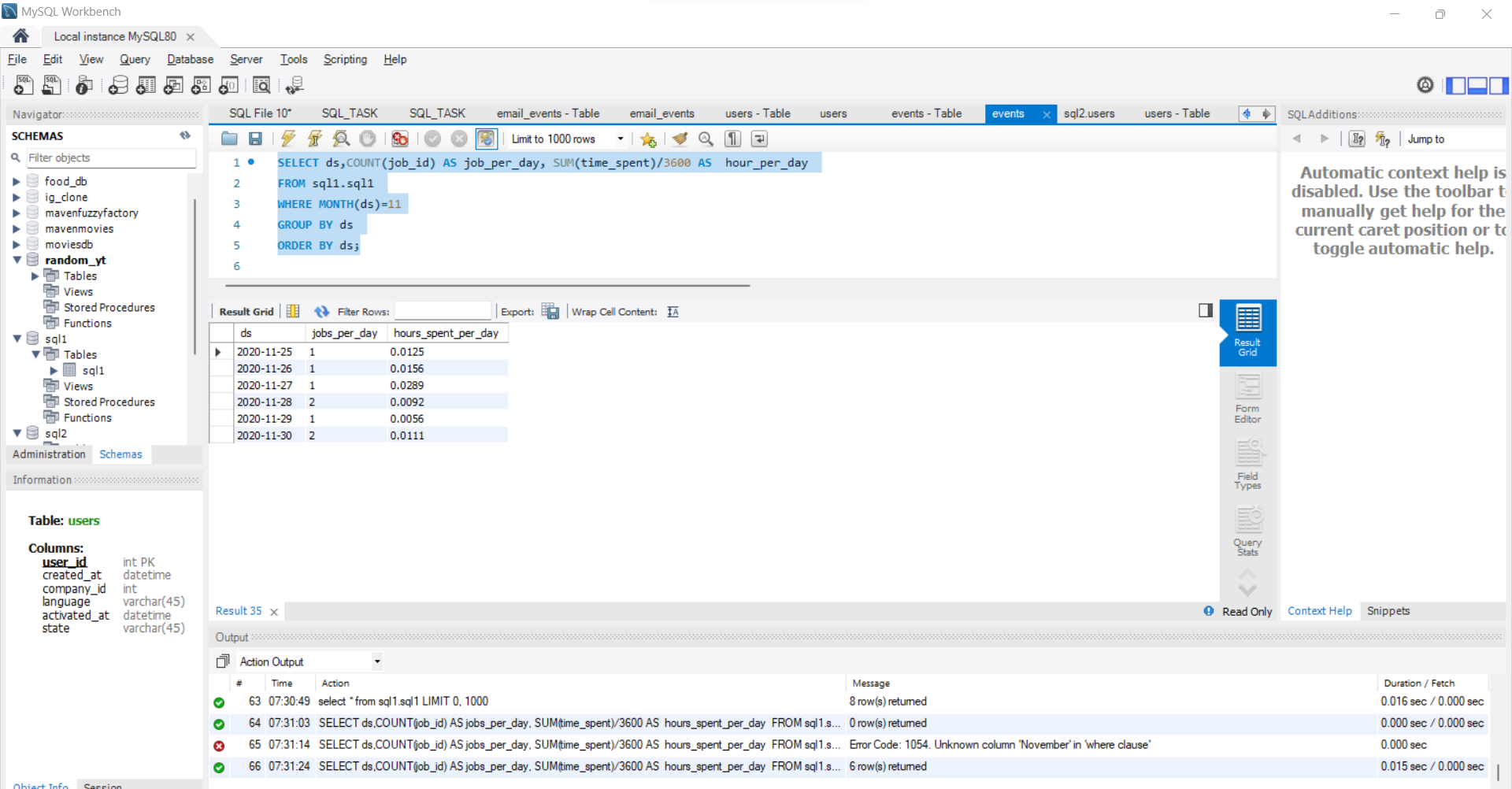
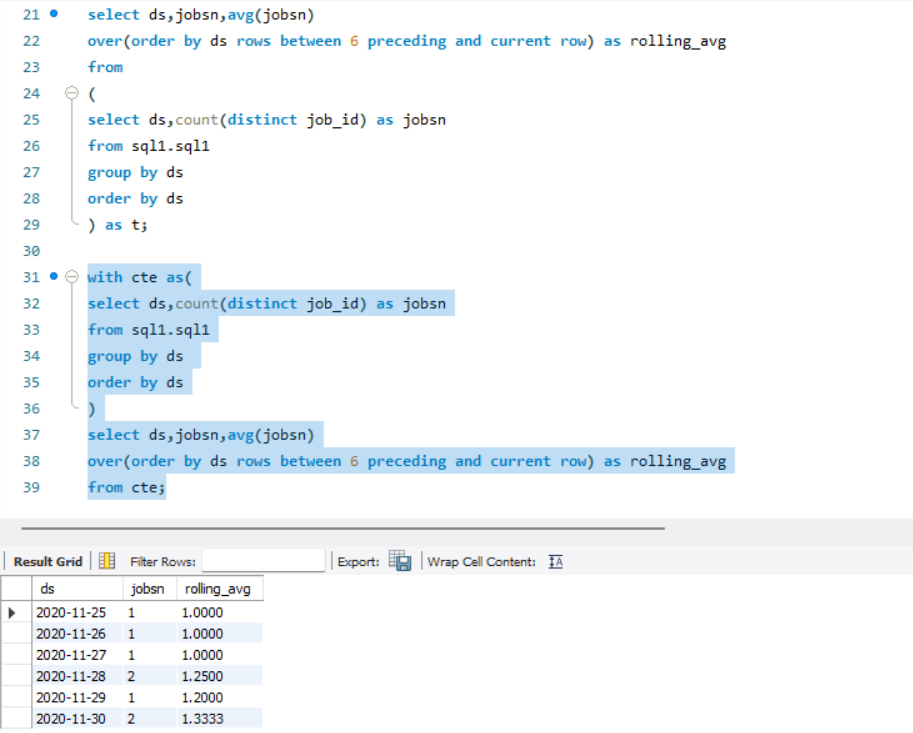
PROJECT DESCRIPTION: Operation Analytics is the analysis done for the complete end to end operations of a company. With the help of this, the company then finds the areas on which it must improve upon. Being one of the most important parts of a company, this kind of analysis is further used to predict the overall growth or decline of a company’s fortune. It means better automation, better understanding between cross-functional teams, and more effective workflows.

**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?

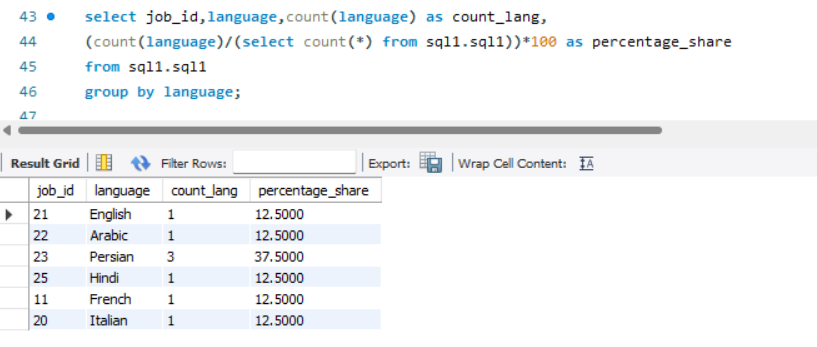




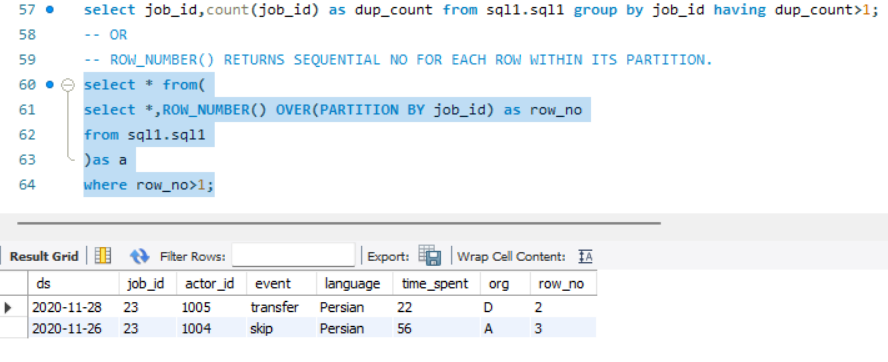
**Throughput:**It is the no. of events happening per second.  
**Your task:** Let’s say the above metric is called throughput. Calculate 7 day rolling average of throughput? For throughput, do you prefer daily metric or 7-day rolling and why?



**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?



**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?



**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.

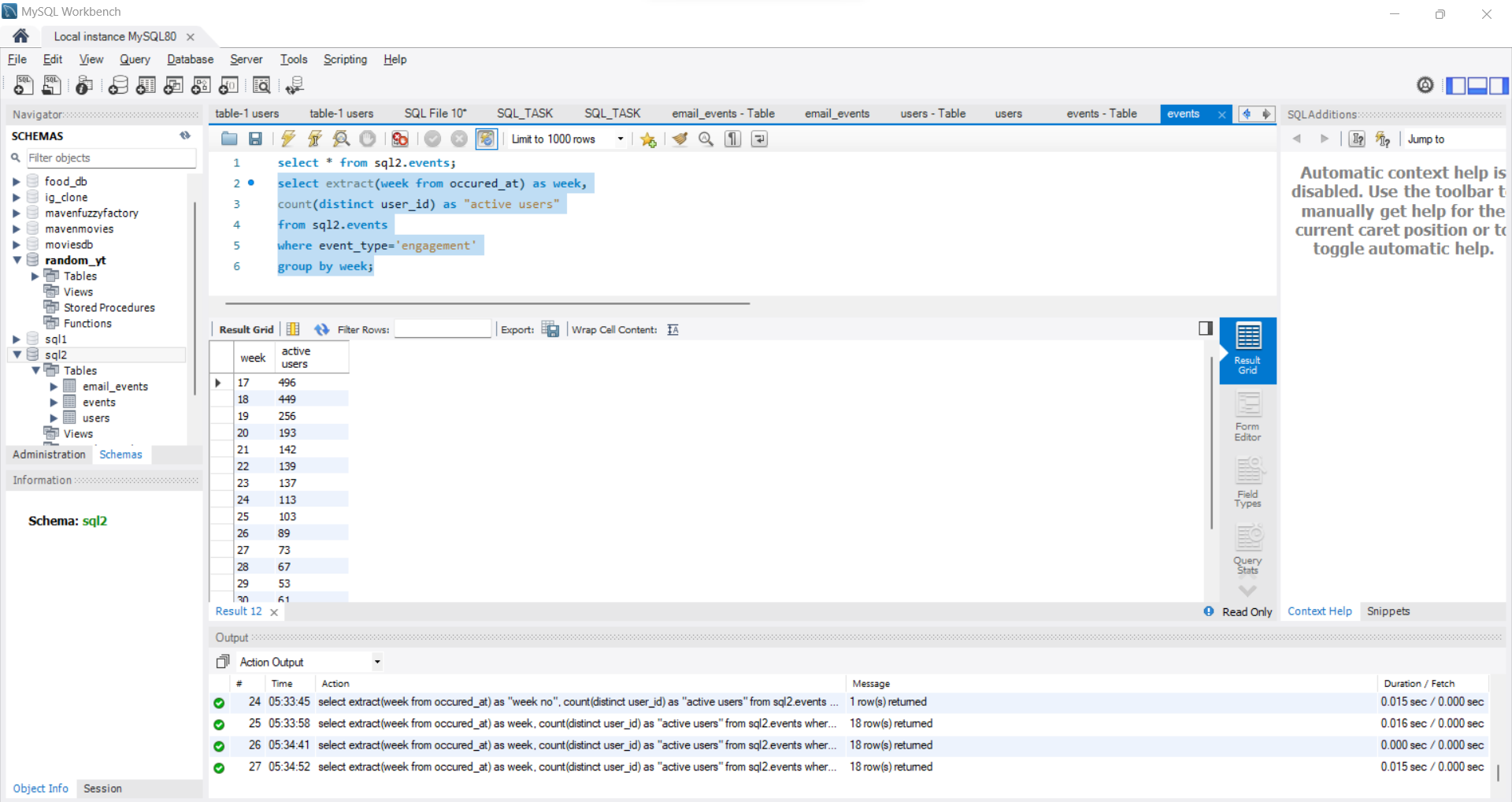
select extract(week from occured\_at) as week,

count(distinct user\_id) as "active users"

from sql2.events

where event\_type='engagement'

group by week;



SELECT week\_number, COUNT(DISTINCT user\_id) as number\_of\_users

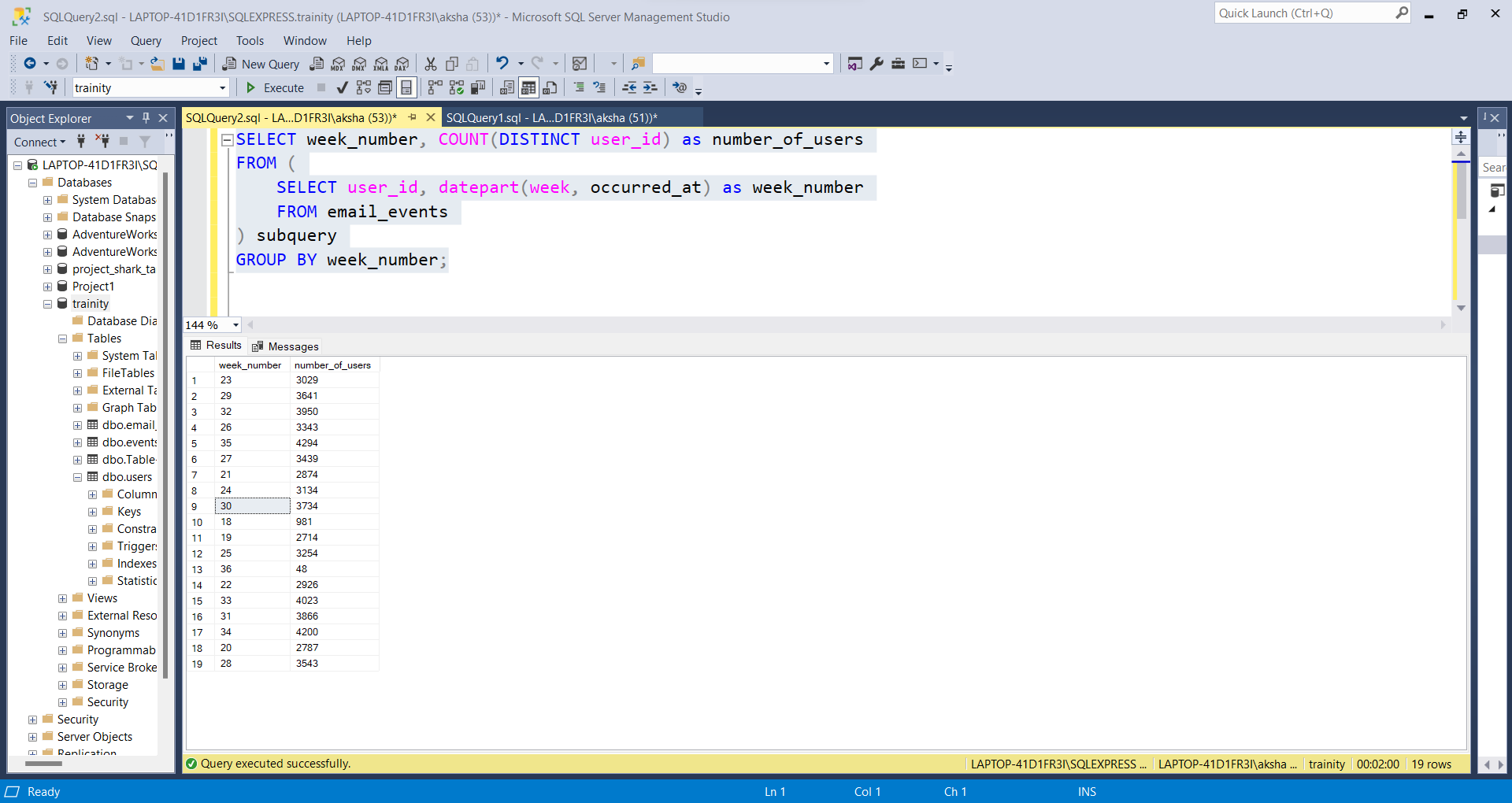
FROM (

SELECT user\_id, datepart(week, occurred\_at) as week\_number

FROM email\_events

) subquery

GROUP BY week\_number;



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

select \* from users;

select

year,

weeknum,

new\_active\_user,

sum(new\_active\_user) over (order by year,weeknum rows between unbounded preceding and current row) as cum\_active\_user

from

(

select

year(activated\_at) as year,

datepart(week,activated\_at) as weeknum,

count(distinct user\_id) as new\_active\_user

from

users

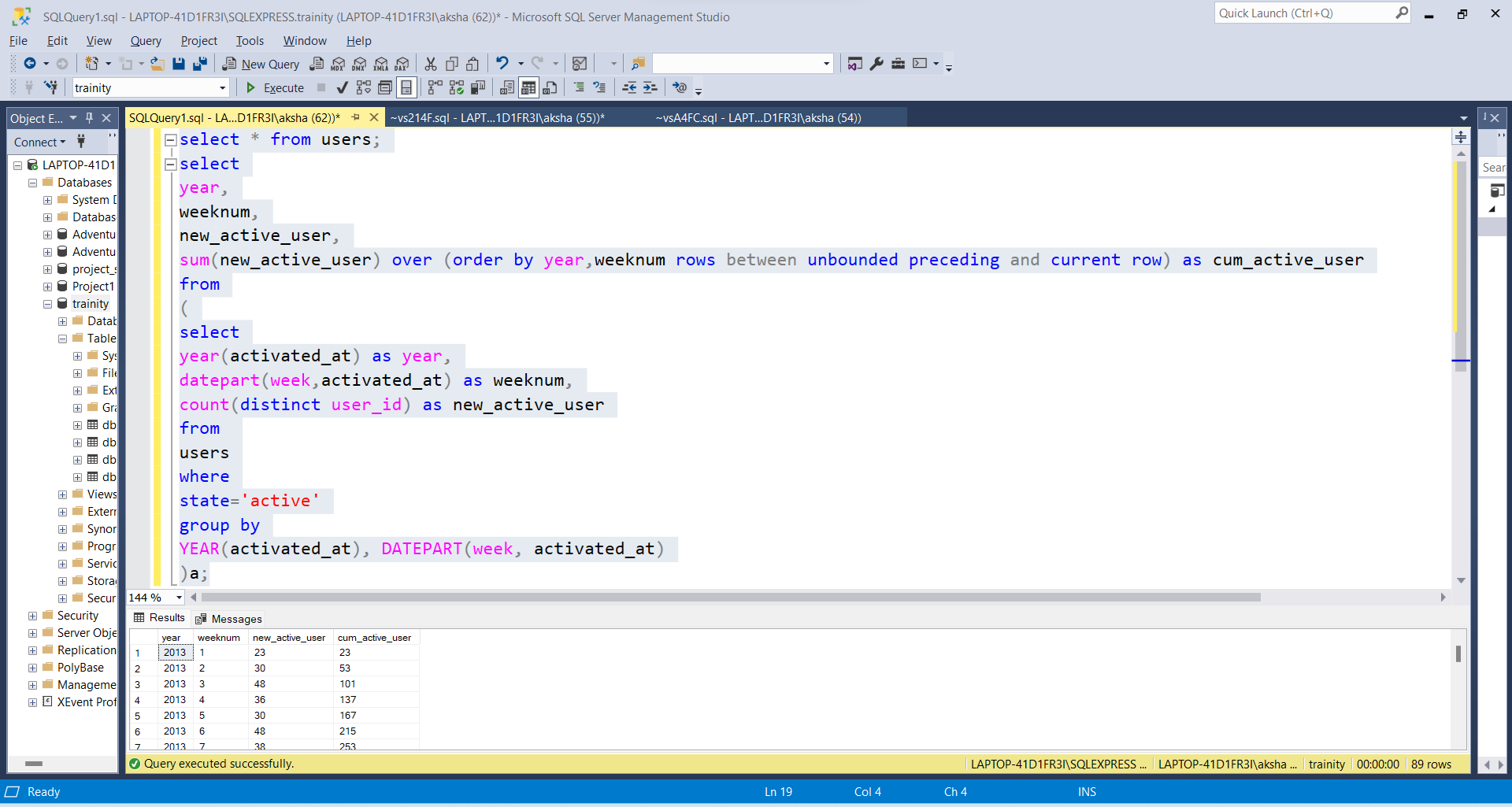
where

state='active'

group by

YEAR(activated\_at), DATEPART(week, activated\_at)

)a;



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

SELECT

first AS [Week Numbers],

SUM(CASE WHEN week\_number = 0 THEN 1 else 0 end) as [week 0],

SUM(CASE WHEN week\_number = 1 THEN 1 else 0 end) as [week 1],

SUM(CASE WHEN week\_number = 2 THEN 1 else 0 end) as [week 2],

SUM(CASE WHEN week\_number = 3 THEN 1 else 0 end) as [week 3],

SUM(CASE WHEN week\_number = 4 THEN 1 else 0 end) as [week 4],

SUM(CASE WHEN week\_number = 5 THEN 1 else 0 end) as [week 5],

SUM(CASE WHEN week\_number = 6 THEN 1 else 0 end) as [week 6],

SUM(CASE WHEN week\_number = 7 THEN 1 else 0 end) as [week 7],

SUM(CASE WHEN week\_number = 8 THEN 1 else 0 end) as [week 8],

SUM(CASE WHEN week\_number = 9 THEN 1 else 0 end) as [week 9],

SUM(CASE WHEN week\_number = 10 THEN 1 else 0 end) as [week 10],

SUM(CASE WHEN week\_number = 11 THEN 1 else 0 end) as [week 11],

SUM(CASE WHEN week\_number = 12 THEN 1 else 0 end) as [week 12],

SUM(CASE WHEN week\_number = 13 THEN 1 else 0 end) as [week 13],

SUM(CASE WHEN week\_number = 14 THEN 1 else 0 end) as [week 14],

SUM(CASE WHEN week\_number = 15 THEN 1 else 0 end) as [week 15],

SUM(CASE WHEN week\_number = 16 THEN 1 else 0 end) as [week 16],

SUM(CASE WHEN week\_number = 17 THEN 1 else 0 end) as [week 17],

SUM(CASE WHEN week\_number = 18 THEN 1 else 0 end) as [week 18]

FROM (

SELECT m.user\_id, m.login\_week, n.first, m.login\_week - first AS week\_number

FROM

(

SELECT user\_id, DATEPART(WEEK, occurred\_at) AS login\_week

FROM events

GROUP BY user\_id, DATEPART(WEEK, occurred\_at)

) m

INNER JOIN

(

SELECT user\_id, MIN(DATEPART(WEEK, occurred\_at)) AS first

FROM events

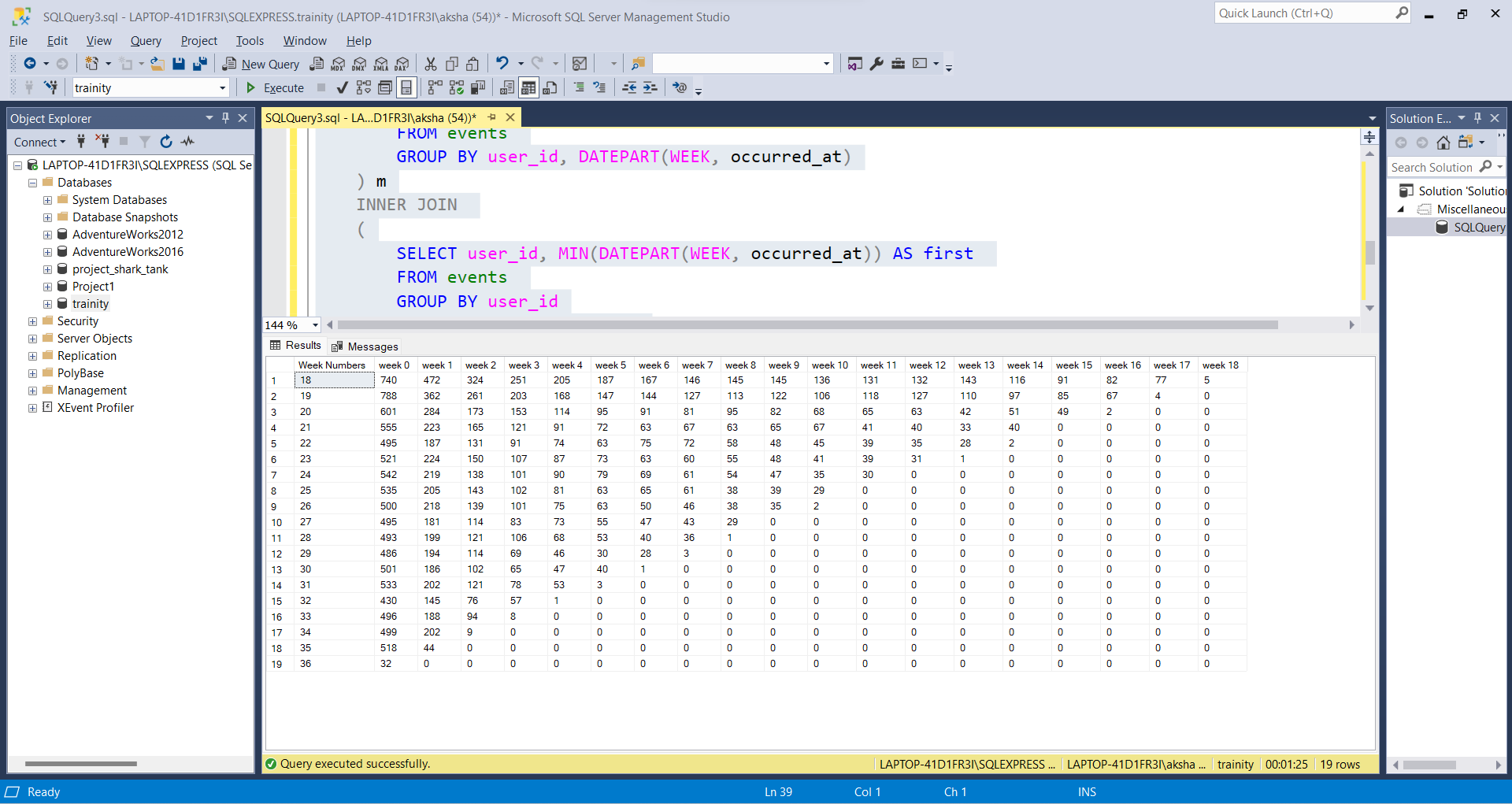
GROUP BY user\_id

) n ON m.user\_id = n.user\_id

) sub

GROUP BY first

ORDER BY first;



MYSQL QUERY->

SELECT first AS "Week Numbers",

SUM(CASE WHEN week\_number = 0 THEN 1 else 0 end) as 'week 0',

SUM(CASE WHEN week\_number = 1 THEN 1 else 0 end) as 'week 1',

SUM(CASE WHEN week\_number = 2 THEN 1 else 0 end) as 'week 2',

SUM(CASE WHEN week\_number = 3 THEN 1 else 0 end) as 'week 3',

SUM(CASE WHEN week\_number = 4 THEN 1 else 0 end) as 'week 4',

SUM(CASE WHEN week\_number = 5 THEN 1 else 0 end) as 'week 5',

SUM(CASE WHEN week\_number = 6 THEN 1 else 0 end) as 'week 6',

SUM(CASE WHEN week\_number = 7 THEN 1 else 0 end) as 'week 7',

SUM(CASE WHEN week\_number = 8 THEN 1 else 0 end) as 'week 8',

SUM(CASE WHEN week\_number = 9 THEN 1 else 0 end) as 'week 9',

SUM(CASE WHEN week\_number = 10 THEN 1 else 0 end) as 'week 10',

SUM(CASE WHEN week\_number = 11 THEN 1 else 0 end) as 'week 11',

SUM(CASE WHEN week\_number = 12 THEN 1 else 0 end) as 'week 12',

SUM(CASE WHEN week\_number = 13 THEN 1 else 0 end) as 'week 13',

SUM(CASE WHEN week\_number = 14 THEN 1 else 0 end) as 'week 14',

SUM(CASE WHEN week\_number = 15 THEN 1 else 0 end) as 'week 15',

SUM(CASE WHEN week\_number = 16 THEN 1 else 0 end) as 'week 16',

SUM(CASE WHEN week\_number = 17 THEN 1 else 0 end) as 'week 17',

SUM(CASE WHEN week\_number = 18 THEN 1 else 0 end) as 'week 18'

FROM (

SELECT m.user\_id, m.login\_week, n.first, m.login\_week - first AS week\_number

FROM

(SELECT user\_id, EXTRACT(WEEK FROM occured\_at) AS login\_week from

sql2.events GROUP BY 1, 2) m,

(SELECT user\_id, MIN(EXTRACT(WEEK FROM occured\_at)) AS first from

sql2.events GROUP BY 1) n

WHERE m.user\_id = n.user\_id) sub

GROUP BY first

ORDER BY first;

**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

YEAR(occurred\_at) AS year\_num,

DATEPART(WEEK, occurred\_at) AS week\_num,

device,

COUNT(DISTINCT user\_id) AS no\_of\_users

FROM

events

WHERE

event\_type = 'engagement'

GROUP BY

YEAR(occurred\_at),

DATEPART(WEEK, occurred\_at),

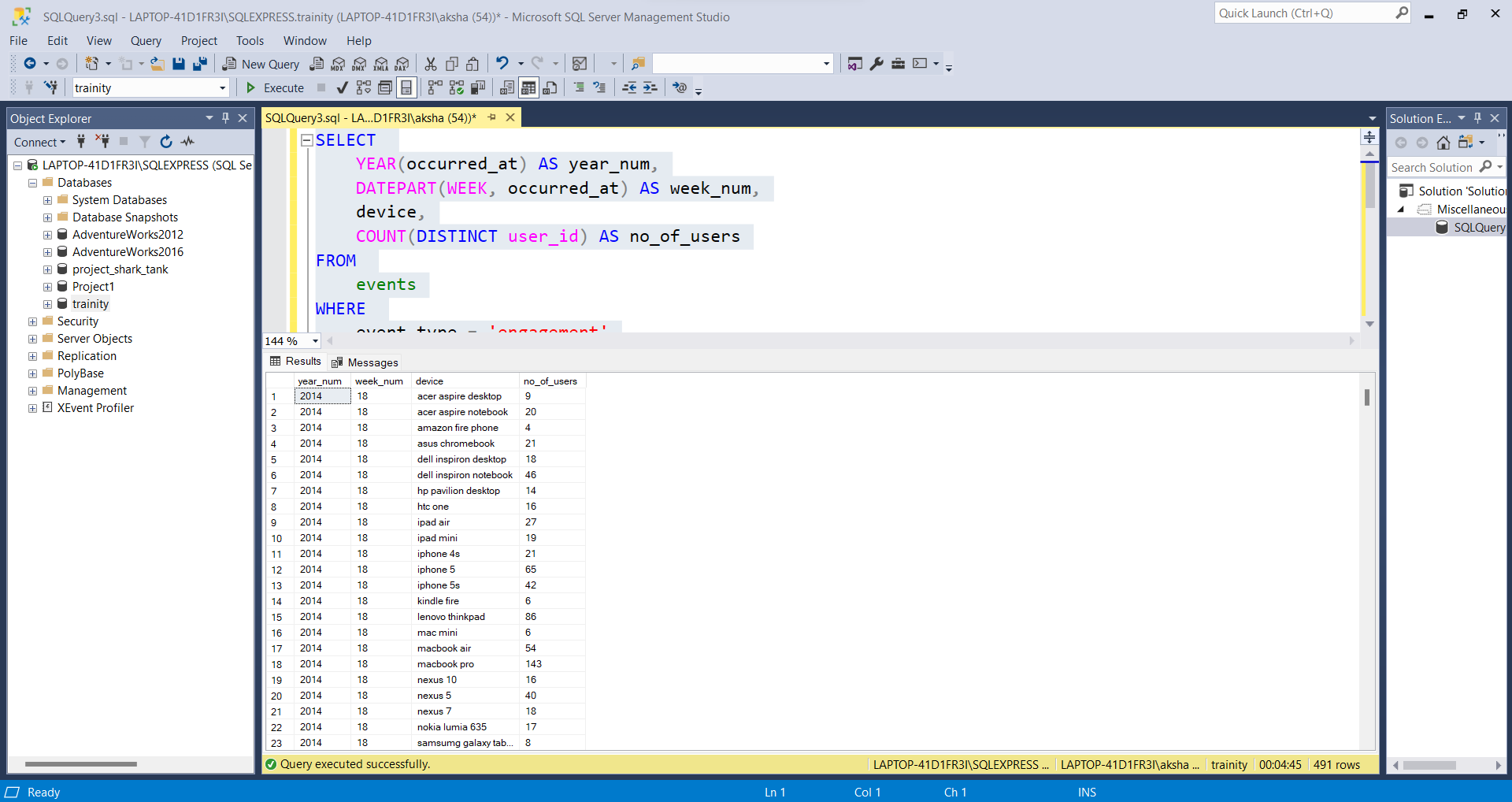
device

ORDER BY

YEAR(occurred\_at),

DATEPART(WEEK, occurred\_at),

device;



MYSQL QUERY->

SELECT

extract(year from occured\_at) as year\_num,

extract(week from occured\_at) as week\_num,

device,

COUNT(distinct user\_id) as no\_of\_users

FROM

sql2.events

where event\_type = 'engagement'

GROUP by 1,2,3

order by 1,2,3;

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

SELECT

DATEPART(WEEK, occurred\_at) AS week,

ROUND((SUM(CASE WHEN action = 'sent\_weekly\_digest' THEN 1 ELSE 0 END) / CAST(COUNT(user\_id) AS FLOAT) \* 100), 2) AS 'Weekly Digest Rate',

ROUND((SUM(CASE WHEN action = 'email\_open' THEN 1 ELSE 0 END) / CAST(COUNT(user\_id) AS FLOAT) \* 100), 2) AS 'Email Open Rate',

ROUND((SUM(CASE WHEN action = 'email\_clickthrough' THEN 1 ELSE 0 END) / CAST(COUNT(user\_id) AS FLOAT) \* 100), 2) AS 'Email Clickthrough Rate',

ROUND((SUM(CASE WHEN action = 'sent\_reengagement\_email' THEN 1 ELSE 0 END) / CAST(COUNT(user\_id) AS FLOAT) \* 100), 2) AS 'Reengagement Email'

FROM

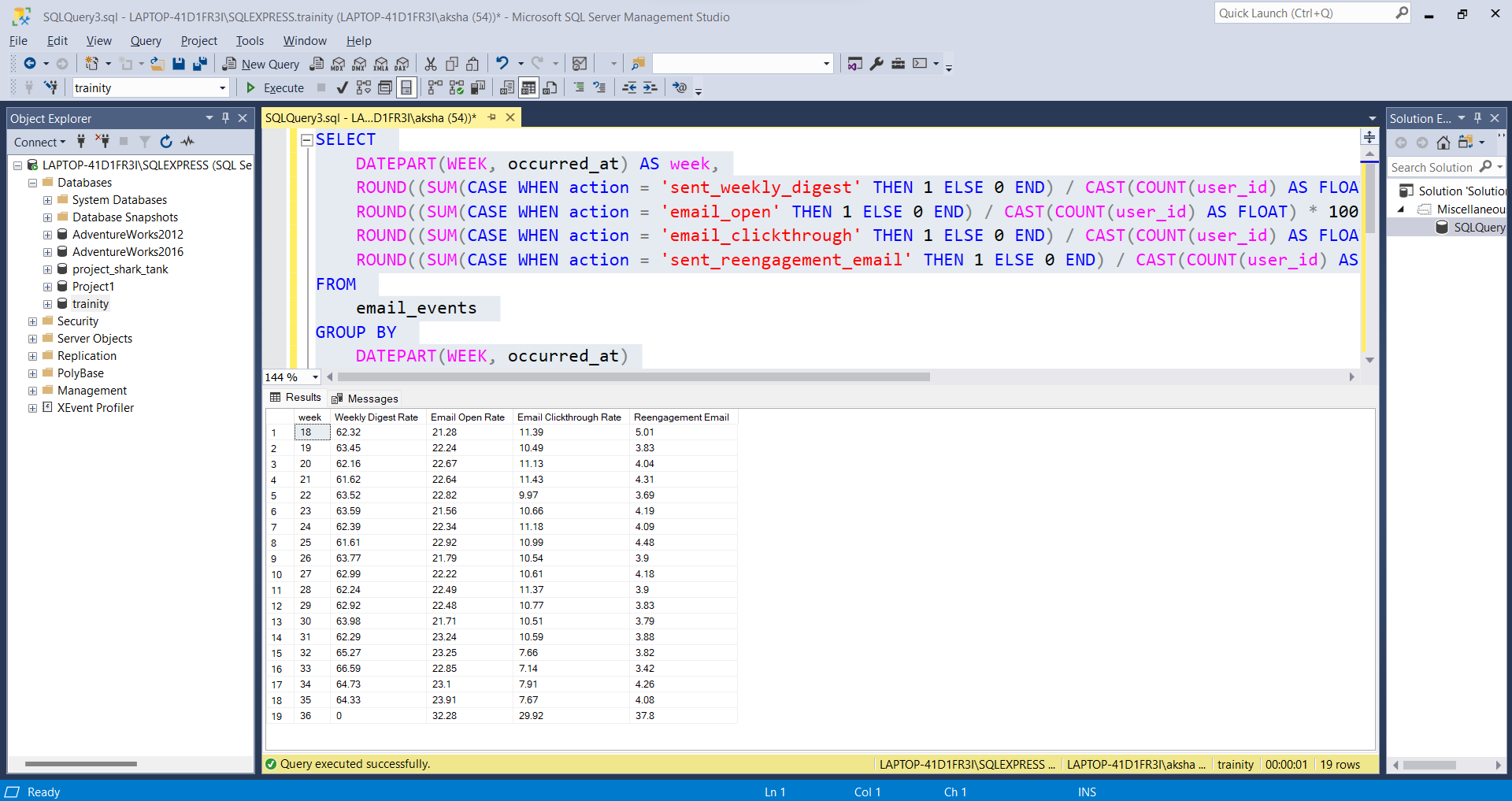
email\_events

GROUP BY

DATEPART(WEEK, occurred\_at)

ORDER BY

DATEPART(WEEK, occurred\_at);



MYSQL QUERY->

SELECT Week,

ROUND((weekly\_digest/total\*100),2) as 'Weekly Digest Rate',

ROUND((email\_opens/total\*100),2) as 'Email Open Rate',

ROUND((email\_clickthroughs/total\*100),2) as 'Email Clickthrough Rate',

ROUND((reengagement\_emails/total\*100),2) as 'Reengagement Email'

FROM

(

SELECT EXTRACT(WEEK FROM occured\_at) as week,

COUNT(CASE WHEN action = 'sent\_weekly\_digest' then user\_id else null end)

as weekly\_digest,

COUNT(CASE WHEN action = 'email\_open' then user\_id else null end) as

email\_opens,

COUNT(CASE WHEN action = 'email\_clickthrough' then user\_id else null end)

as email\_clickthroughs,

COUNT(CASE WHEN action = 'sent\_reengagement\_email' then user\_id else null

end) as reengagement\_emails,

COUNT(user\_id) AS total

FROM sql2.email\_events

GROUP BY 1 ) sub

GROUP BY 1

ORDER BY 1;