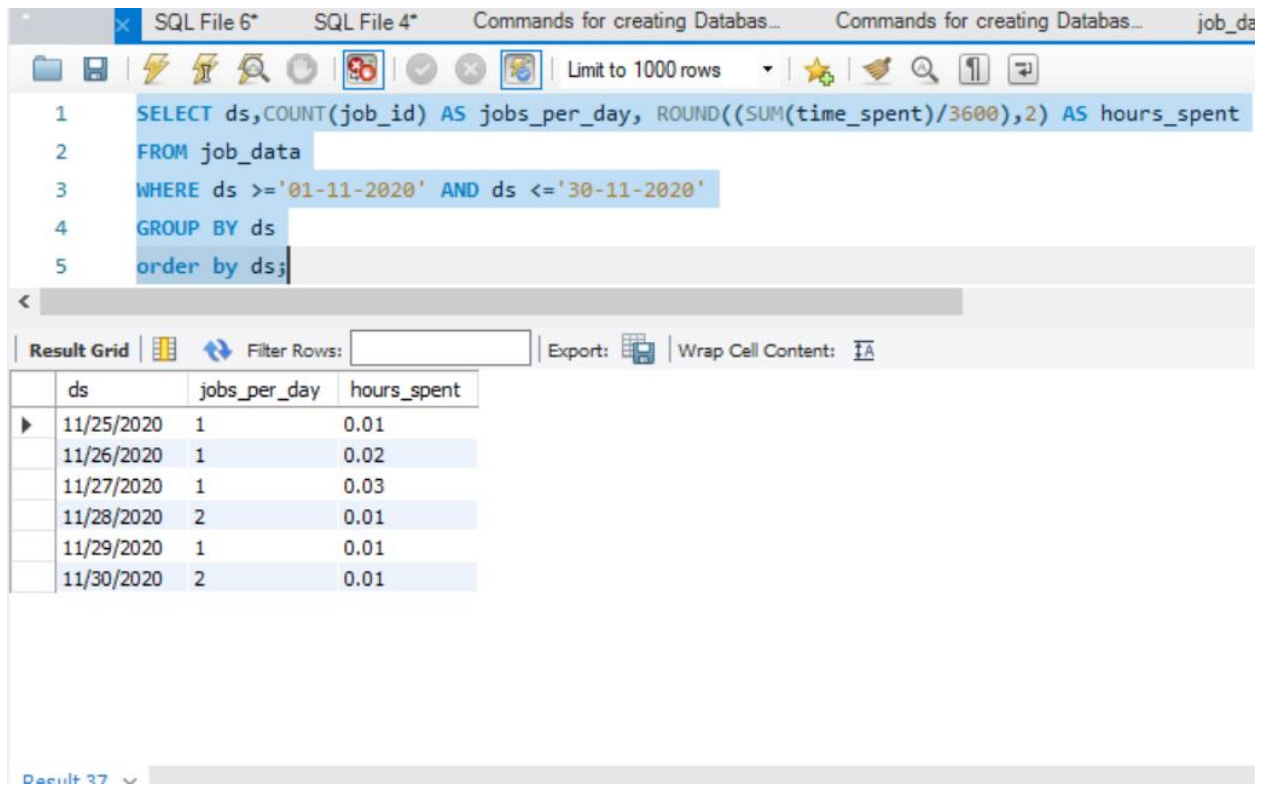


Write an SQL query to calculate the number of jobs reviewed per hour for each day in November 2020



The screenshot shows a SQL IDE interface with a query editor and a result grid. The query editor contains the following SQL query:

```
1 SELECT ds,COUNT(job_id) AS jobs_per_day, ROUND((SUM(time_spent)/3600),2) AS hours_spent
2 FROM job_data
3 WHERE ds >='01-11-2020' AND ds <='30-11-2020'
4 GROUP BY ds
5 order by ds;
```

The result grid displays the following data:

ds	jobs_per_day	hours_spent
11/25/2020	1	0.01
11/26/2020	1	0.02
11/27/2020	1	0.03
11/28/2020	2	0.01
11/29/2020	1	0.01
11/30/2020	2	0.01

The IDE interface includes a toolbar with various icons, a status bar at the bottom, and a filter row section above the result grid.

Write an SQL query to calculate the 7-day rolling average of throughput. Additionally, explain whether you prefer using the daily metric or the 7-day rolling average for throughput, and why.

I would use 7-day rolling average as while plotting the graph it will be much more accurate and will not have sharp dip or rise which can hinder the management while taking decisions.


```
1 select ds, avg(time_spent) over (order by ds rows between 6 preceding and current row) as 7_day_average
2 from job_data;
```


Result Grid	Filter Rows:	Export:	Wrap Cell Content:
ds	7_day_average		
11/25/2020	45.0000		
11/26/2020	50.5000		
11/27/2020	68.3333		
11/28/2020	56.7500		
11/28/2020	47.6000		
11/29/2020	43.0000		
11/30/2020	39.0000		
11/30/2020	36.1429		

Write an SQL query to calculate the percentage share of each language over the last 30 days.

```
1 select language, count(time_spent)/sum(time_spent) * 100 as percent_share from job_data
2 group by language
```


Result Grid






Filter Rows:

Export:






Wrap Cell Content:



language	percent_share
English	6.6667
Arabic	4.0000
Persian	3.0612
Hindi	9.0909
French	0.9615
Italian	2.2222

Write an SQL query to display duplicate rows from the **job_data** table.

```
• SELECT job_id,actor_id,count(*) FROM job_data
  group by job_id,actor_id
  having count(*)>1
```

Alt Grid |   Filter Rows: | Export: 

job_id	actor_id	count(*)
--------	----------	----------

CASE STUDY 2

Q1 Write an SQL query to calculate the weekly user engagement

```
1 Select Extract(week from occurred_at) as number_of_week,  
2 count(DISTINCT user_id)  
3 From tutorial.yammer_events  
4 Group by number_of_week  
5
```

	Data	Fields	Source
		number_of_week	count
1		18	791
2		19	1244
3		20	1270
4		21	1341
5		22	1293
6		23	1366
7		24	1434
8		25	1133
9		26	1100

Here we are trying to measure the activeness of a user on a weekly basis which is nothing but to understand how engaged the user is with the product. We see that the number grow with the rising number of weeks

Write an SQL query to calculate the user growth for the product.

▶ Run

✓ Limit 100

Format

↺ View history

```
1 select year_num,active_users, sum(active_users) over (order by year_num) as running_total
2 from(
3 select extract(year from activated_at) as year_num,
4 count(distinct user_id) as active_users from tutorial.yammer_users
5 where state='active'
6 group by year_num
7 order by year_num
8 ) as a
```

	Data	Fields	Source
		year_num	active_users
1	2013	3283	running_total
2	2014	6098	9381

The above result shows the number of users increased with time in 2013 and 2014. As per the running total the numbers almost increased by thrice of what they were initially.

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

```
SELECT
  signup_week,
  activity_week,
  active_users,
  active_users / FIRST_VALUE(active_users) OVER (PARTITION BY signup_week ORDER BY activity_week) * 100 as retention_rate
FROM (
  SELECT
    Extract(week from created_at) as signup_week,
    Extract(week from activated_at) as activity_week,
    COUNT(DISTINCT user_id) as active_users
  FROM tutorial.yammer_users
  GROUP BY signup_week, activity_week
) as a
ORDER BY signup_week, activity_week;
```

Fields		Source			
signup_week	activity_week	active_users	retention_rate		
1	1	158	100		
1		131	0		
2	2	151	100		
2		148	0		
3	3	159	100		
3		149	0		

In this query, we can see if the user who created his profile on a particular week has used the product on the same or not and for how much time. It also signifies how involved the user has been with the product.

Write an SQL query to calculate the weekly engagement per device

```
select extract(week from occurred_at) as week, extract(year from occurred_at) as year, device,
count (distinct user_id) as active_users from tutorial.yammer_events
where event_type = 'engagement'
group by week, year, device
order by week, year, device
```

Fields		Source	
week	year	device	active_users
1	18	acer aspire desktop	10
2	18	acer aspire notebook	21
3	18	amazon fire phone	4
4	18	asus chromebook	23
5	18	dell inspiron desktop	21
6	18	dell inspiron notebook	49
7	18	hp pavilion desktop	15
8	18	htc one	16
9	18	ipad air	30
0	18	ipad mini	21
1	18	iphone 4s	21
2	18	iphone 5	70

By above result we can understand that users prefer using mobile phones while using the product.

Write an SQL query to calculate the email engagement metrics.

```
select count(*) as number, action from tutorial.yammer_emails
group by action
```

Fields		Source	
number	action		
20459	email_open		
9010	email_clickthrough		
57267	sent_weekly_digest		
3653	sent_reengagement_email		

This result show how much people are engaged with their mailboxes which can help the managers to decide to start any email ad campaign.