INFO1085 - SQL Server

School of Workforce Development, Continuing Education, and Online Learning

Network Technical Support (# 1470)

Course Information

Name	**SQL Server**
Code	**INFO1085 - Section 1**

Faculty Information

Professor/Instructor	Firas Chahine, Ph.D.
Email	fchahine@conestogac.on.ca
Office Number	113
Office Hours	By appointment only

Lab 9.1

Working with database monitoring and troubleshooting tools:

1- Preparations for this lab on DB1 server:

   a. Create database ASSIGNMENT9:

```
CREATE DATABASE ASSIGNMENT9;
```

   b. Create tables EmployeePay, OrderDetail and OrderHeader from copying contents from AdventureWorks2016:

```
USE ASSIGNMENT9;
SELECT * INTO EmployeePay
FROM AdventureWorks2016.HumanResources.EmployeePayHistory
SELECT * INTO OrderDetail
FROM AdventureWorks2016.Sales.SalesOrderDetail
SELECT * INTO OrderHeader
FROM AdventureWorks2016.Sales.SalesOrderHeader
```

   c. Create database user RogueAdmin with Sysadmin access to your instance DB1\SQLSVR01:

```
CREATE LOGIN RogueAdmin WITH PASSWORD = 'Secret55!';
CREATE USER RogueAdmin FOR LOGIN RogueAdmin;
ALTER SERVER ROLE SysAdmin ADD MEMBER RogueAdmin;
```

2- Working with Performance Monitor (perfmon.exe):

a. Launch Performance Monitor and observe the following performance counters live to determine the health of your DB1 server:

i. *Processor Information: % Processor Time* - measures how busy the processor is

1. Sustained values over 70% during normal activity should be investigated

ii. *System: Processor Queue Length* - displays a count of the number of processes that are waiting for the processor to finish processing the active request

iii. *Memory: Available MBytes* - measures the amount of physical memory that is available for allocation

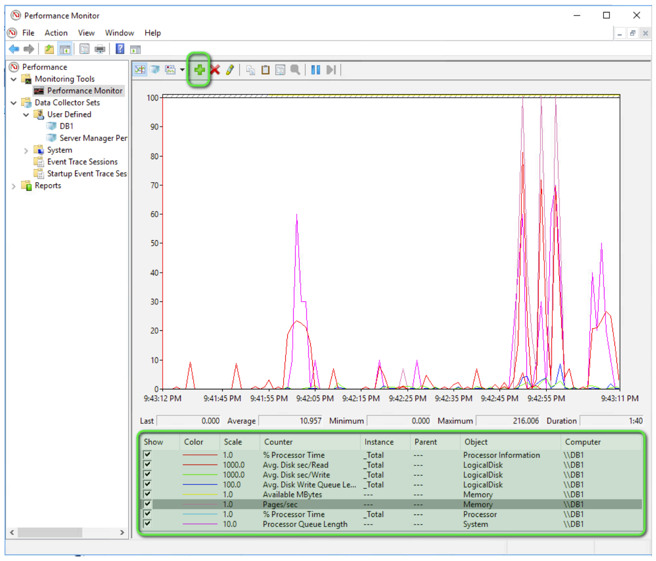
iv. *Memory: Pages/sec* - measures the activity transferring data between the physical disk and memory

v. *Logical Disk: Avg. Disk sec/Read* - measures disk throughput

1. A high number indicates the disk if operating efficiently

vi. *Logical Disk: Avg. Disk sec/Write* - measures disk throughput for processes writing data to the disk

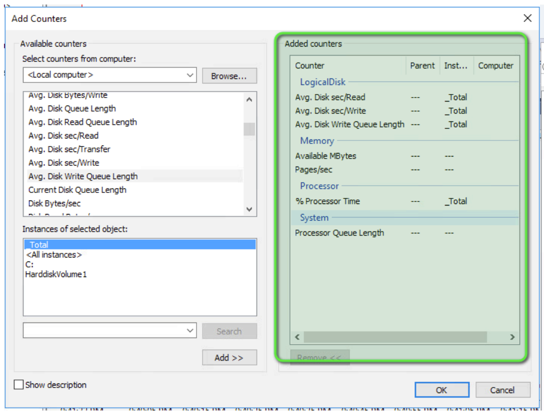
vii. *Logical Disk: Avg. Disk Queue Length* - provides a count of the number of requests waiting to read or write data from the physical disk



Last 0.000 Average 10.957 Minimum 0.000 Maximum 216.006 Duration 1:40

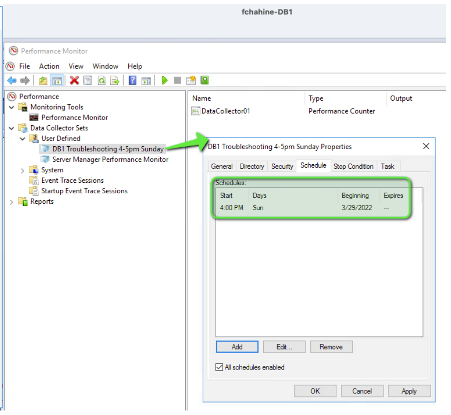
Show	Color	Scale	Counter	Instance	Parent	Object	Computer
✓	🟥	1.0	% Processor Time	_Total	---	Processor Information	\\DB1
✓	🟥	1000.0	Avg. Disk sec/Read	_Total	---	LogicalDisk	\\DB1
✓	🟩	1000.0	Avg. Disk sec/Write	_Total	---	LogicalDisk	\\DB1
✓	🟦	100.0	Avg. Disk Write Queue Le...	_Total	---	LogicalDisk	\\DB1
✓	🟨	1.0	Available MBytes	---	---	Memory	\\DB1
✓	🟪	1.0	Pages/sec	---	---	Memory	\\DB1
✓	🟦	1.0	% Processor Time	_Total	---	Processor	\\DB1
✓	🟪	10.0	Processor Queue Length	---	---	System	\\DB1

b. Users are complaining that performance of their applications degrade between 4-5pm on Sundays. From your DB1 server, create a Data Collector Set and schedule it and collect logs during that time:
Data Collector Sets > User Defined > [Right click] New > Data Collector Set
   i. Name: DB1 Troubleshooting 4-5pm Sunday
   ii. Create manually (Advanced)
   iii. Create data logs:
      1. [Check] Performance Counter
   iv. Add 7 performance counters below:
      1. *Processor > % Processor Time*
      2. *System > Processor Queue Length*
      3. *Memory > Available MBytes*
      4. *Memory > Pages/sec*
      5. *Logical Disk > Avg. Disk sec/Read*
      6. *Logical Disk > Avg. Disk sec/Write*
      7. *Logical Disk > Avg. Disk Queue Length*
   v. Change Sample internal to 2 seconds
   vi. Choose location of log file (C:\Users\SQLAdmin\Desktop\DB1 Troubleshooting 4-5pm Sunday)
   vii. Set schedule to run at 4pm for 1 hour on a Sunday



Added counters

Counter	Parent	Inst...	Computer
LogicalDisk			
Avg. Disk sec/Read	---	_Total	
Avg. Disk sec/Write	---	_Total	
Avg. Disk Write Queue Length	---	_Total	
Memory			
Available MBytes	---	---	
Pages/sec	---	---	
Processor			
% Processor Time	---	_Total	
System			
Processor Queue Length	---	---	

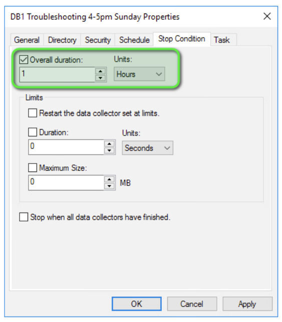


Name	Type	Output
DataCollector01	Performance Counter	

Schedules:

Start	Days	Beginning	Expires
4:00 PM	Sun	3/29/2022	---

☑ All schedules enabled



☑ Overall duration: 1 Units: Hours

Limits

☐ Restart the data collector set at limits.

☐ Duration: 0 Units: Seconds

☐ Maximum Size: 0 MB

☐ Stop when all data collectors have finished.

3- Working with SQL Profiler:

a. Your manager suspects that there is a rogue DBA and you are tasked with monitoring his activities on the database and collecting evidence of any malicious activities:

Launch SQL Server Profiler > File > New Trace…

i. > General:

1- Connect to your SQL instance

2- Trace name: RogueDBA Trace

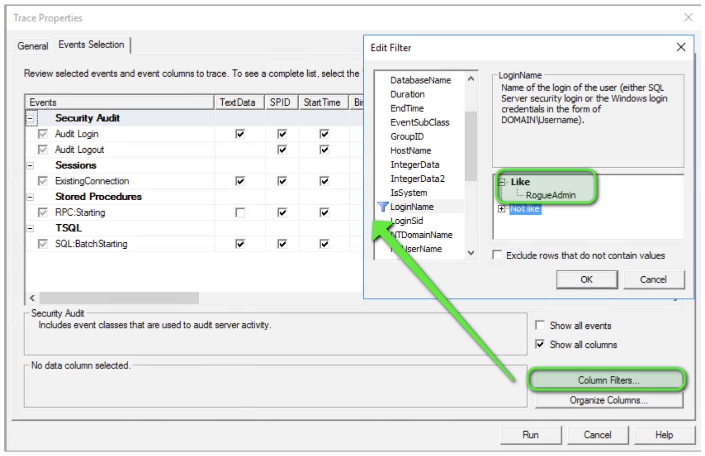
3- [Check] Save to file and select location (C:\Users\SQLAdmin\Desktop\SQL Profiler\ RogueDBA Trace.trc)

ii. > Event Selection > [Check] Show all columns

iii. > Event Selection > Columns Filter >

1- LoginName = RogueAdmin

iv. Run trace and leave it collecting events



Trace Properties

General | Events Selection

Review selected events and event columns to trace. To see a complete list, select the

Events	TextData	SPID	StartTime	Bi
Security Audit				
Audit Login	✓	✓	✓	
Audit Logout		✓	✓	
Sessions				
ExistingConnection	✓	✓	✓	
Stored Procedures				
RPC:Starting	☐	✓	✓	
TSQL				
SQL:BatchStarting	✓	✓	✓	

Security Audit
Includes event classes that are used to audit server activity.

No data column selected.

☐ Show all events
✓ Show all columns

Edit Filter

DatabaseName
Duration
EndTime
EventSubClass
GroupID
HostName
IntegerData
IntegerData2
IsSystem
LoginName
LoginSid
NTDomainName
NTUserName

LoginName
Name of the login of the user (either SQL Server security login or the Windows login credentials in the form of DOMAIN\Username).

Like
RogueAdmin
Not like

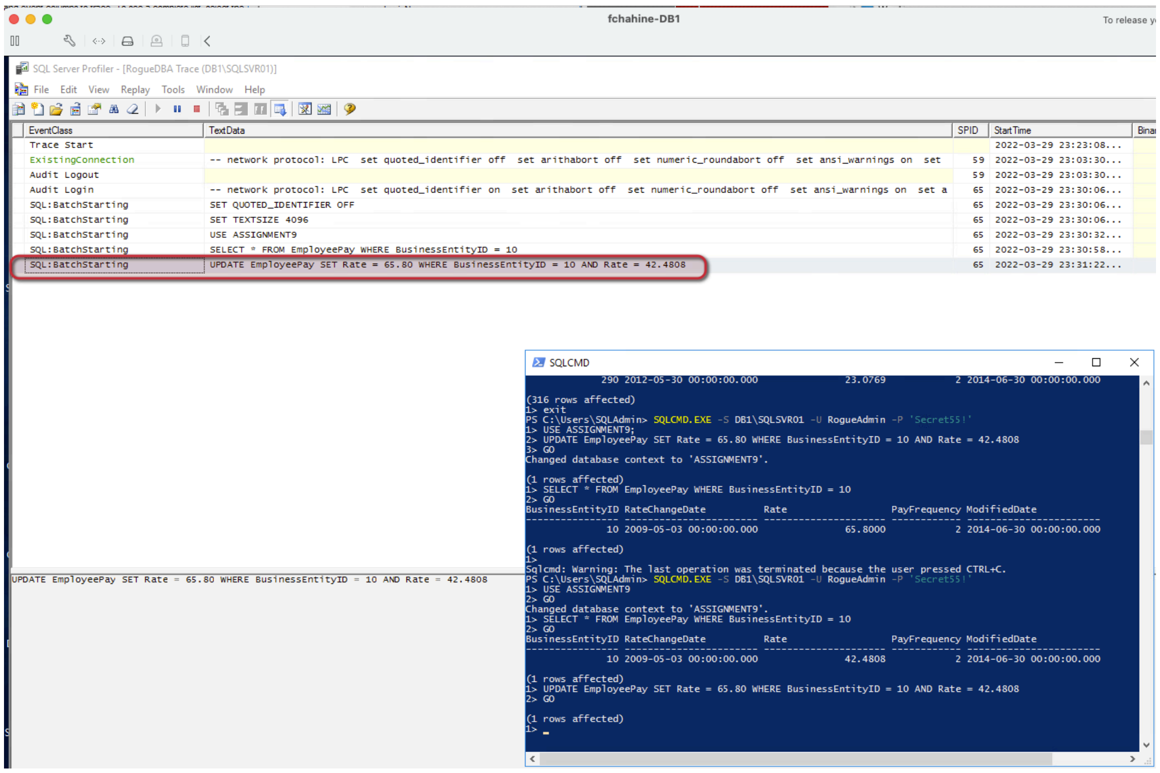
☐ Exclude rows that do not contain values

b. Become a rogue DBA, login to DB1\SQLSVR01 using RogueAdmin and attempt to modify sensitive records:

Using PowerShell:

```
SQLCMD.EXE -S DB1\SQLSVR01 -U RogueAdmin -P 'Secret55!'
USE ASSIGNMENT9
GO
SELECT * FROM EmployeePay WHERE BusinessEntityID = 10
GO
UPDATE EmployeePay SET Rate = 65.80 WHERE BusinessEntityID = 10 AND Rate
= 42.4808
GO
```

c. Go back to SQL Profiler and observe the rogue DBA and see if you can find any evidence of malicious acts:
   i. When done, stop the trace.


fchahine-DB1
To release y
SQL Server Profiler - [RogueDBA Trace (DB1\SQLSVR01)]
File Edit View Replay Tools Window Help
SQLCMD

EventClass	TextData	SPID	StartTime	Binar
Trace Start			2022-03-29 23:23:08...	
ExistingConnection	-- network protocol: LPC set quoted_identifier off set arithabort off set numeric_roundabort off set ansi_warnings on set	59	2022-03-29 23:03:30...	
Audit Logout		59	2022-03-29 23:03:30...	
Audit Login	-- network protocol: LPC set quoted_identifier on set arithabort off set numeric_roundabort off set ansi_warnings on set a	65	2022-03-29 23:30:06...	
SQL:BatchStarting	SET QUOTED_IDENTIFIER OFF	65	2022-03-29 23:30:06...	
SQL:BatchStarting	SET TEXTSIZE 4096	65	2022-03-29 23:30:06...	
SQL:BatchStarting	USE ASSIGNMENT9	65	2022-03-29 23:30:32...	
SQL:BatchStarting	SELECT * FROM EmployeePay WHERE BusinessEntityID = 10	65	2022-03-29 23:30:58...	
SQL:BatchStarting	UPDATE EmployeePay SET Rate = 65.80 WHERE BusinessEntityID = 10 AND Rate = 42.4808	65	2022-03-29 23:31:22...	

```
UPDATE EmployeePay SET Rate = 65.80 WHERE BusinessEntityID = 10 AND Rate = 42.4808
```

```
             290 2012-05-30 00:00:00.000                23.0769            2 2014-06-30 00:00:00.000

(316 rows affected)
1> exit
PS C:\Users\SQLAdmin> SQLCMD.EXE -S DB1\SQLSVR01 -U RogueAdmin -P 'Secret55!'
1> USE ASSIGNMENT9;
2> UPDATE EmployeePay SET Rate = 65.80 WHERE BusinessEntityID = 10 AND Rate = 42.4808
3> GO
Changed database context to 'ASSIGNMENT9'.

(1 rows affected)
1> SELECT * FROM EmployeePay WHERE BusinessEntityID = 10
2> GO
BusinessEntityID RateChangeDate          Rate                  PayFrequency ModifiedDate
---------------- ----------------------- --------------------- ------------ -----------------------
              10 2009-05-03 00:00:00.000               65.8000            2 2014-06-30 00:00:00.000

(1 rows affected)
1>
Sqlcmd: Warning: The last operation was terminated because the user pressed CTRL+C.
PS C:\Users\SQLAdmin> SQLCMD.EXE -S DB1\SQLSVR01 -U RogueAdmin -P 'Secret55!'
1> USE ASSIGNMENT9
2> GO
Changed database context to 'ASSIGNMENT9'.
1> SELECT * FROM EmployeePay WHERE BusinessEntityID = 10
2> GO
BusinessEntityID RateChangeDate          Rate                  PayFrequency ModifiedDate
---------------- ----------------------- --------------------- ------------ -----------------------
              10 2009-05-03 00:00:00.000               42.4808            2 2014-06-30 00:00:00.000

(1 rows affected)
1> UPDATE EmployeePay SET Rate = 65.80 WHERE BusinessEntityID = 10 AND Rate = 42.4808
2> GO

(1 rows affected)
1> _
```

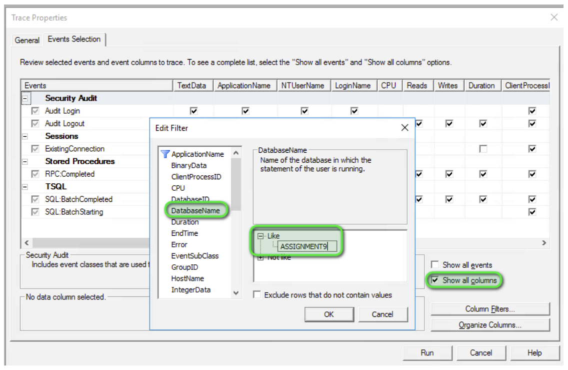
d. Users are complaining that performance of their applications degrade between 4-5pm on Sundays. From Performance Monitor logs, you can see that during that time DB1 server experiences a jump in a number of performance counters. You suspect that this might be related to activities on the SQL database instance. You decide to monitor SQL Server events using SQL Profiler during that period.
   i. Assume it is Sunday 4pm. Run the Data Collector set you create manually:
      Performance Monitor > Data Collector Sets > User Defined > [Right-click] DB1 Troubleshooting 4-5pm Sunday > Start
   ii. Run a trace on SQL Profiler as follows:
      Launch SQL Server Profiler > File > New Trace…
      1- > General:
         - Connect to your SQL instance
         - Trace name: DB1 Troubleshooting 4-5pm Sunday

2- [Check] Save to file and select location (C:\Users\SQLAdmin\Desktop\SQL Profiler\ DB1 Troubleshooting 4-5pm Sunday.trc)

3- > Event Selection > [Check] Show all columns

4- > Event Selection > Columns Filter >

  - DatabaseName = ASSIGNMENT9



Trace Properties

Review selected events and event columns to trace. To see a complete list, select the "Show all events" and "Show all columns" options.

Edit Filter

DatabaseName: Name of the database in which the statement of the user is running.

Like: ASSIGNMENT9

[✓ Show all columns]

5- Run trace and leave it collecting events

iii. Let's simulate some SQL activities that for some reason (could be scheduled weekly reports for management) takes place during that period (select and run all commands together as a batch):

```
USE ASSIGNMENT9;
SELECT a.SalesOrderID, a.Status, a.OnlineOrderFlag, a.AccountNumber,
a.CustomerID, a.ShipToAddressID, a.CreditCardID, b.LineTotal FROM
OrderHeader a
JOIN OrderDetail b ON b.SalesOrderID = a.SalesOrderID

SELECT a.SalesOrderID, a.Status, a.OnlineOrderFlag, a.AccountNumber,
a.CustomerID, a.ShipToAddressID, a.CreditCardID, b.LineTotal FROM
OrderHeader a
JOIN OrderDetail b ON b.SalesOrderID = a.SalesOrderID
WHERE a.SalesOrderID = '44403'

SELECT a.SalesOrderID, a.Status, a.OnlineOrderFlag, a.AccountNumber,
a.CustomerID, a.ShipToAddressID, a.CreditCardID, b.LineTotal FROM
OrderHeader a
JOIN OrderDetail b ON b.SalesOrderID = a.SalesOrderID
WHERE a.SalesOrderID = '44404'
```

CONESTOGA

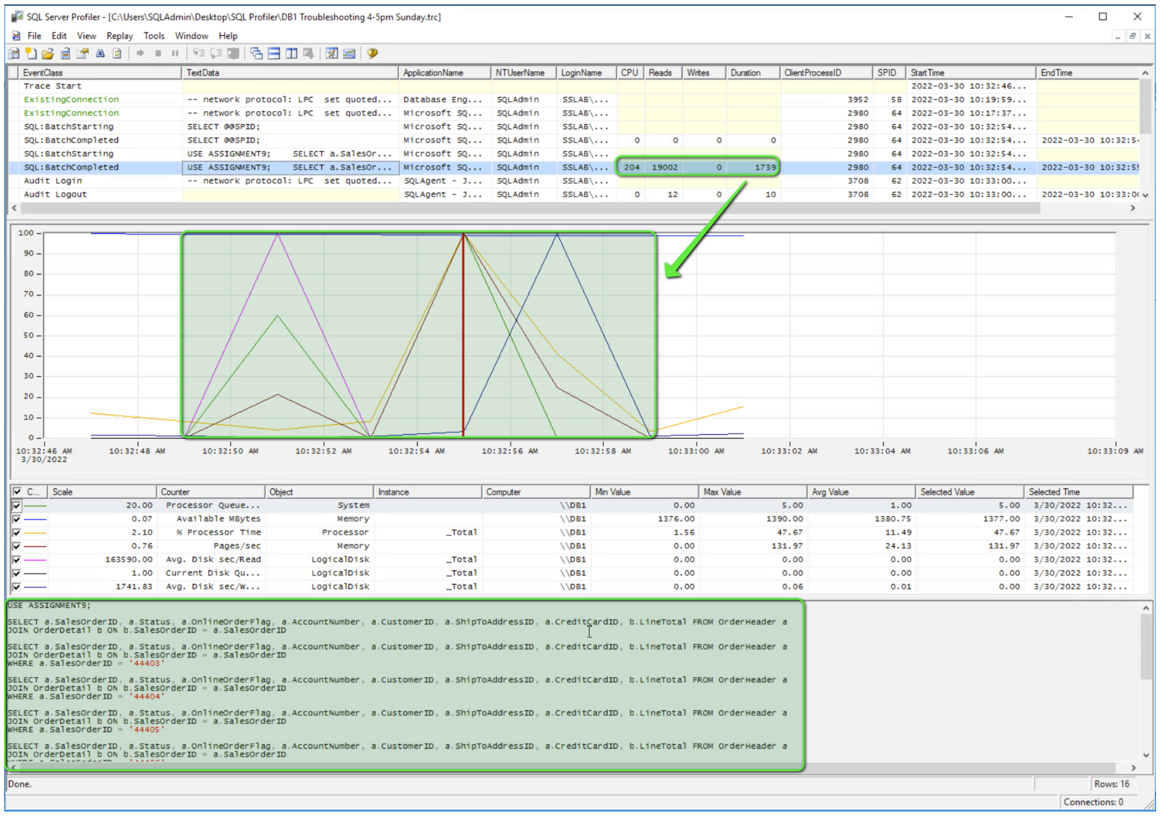
```
SELECT a.SalesOrderID, a.Status, a.OnlineOrderFlag, a.AccountNumber,
a.CustomerID, a.ShipToAddressID, a.CreditCardID, b.LineTotal FROM
OrderHeader a
JOIN OrderDetail b ON b.SalesOrderID = a.SalesOrderID
WHERE a.SalesOrderID = '44405'

SELECT a.SalesOrderID, a.Status, a.OnlineOrderFlag, a.AccountNumber,
a.CustomerID, a.ShipToAddressID, a.CreditCardID, b.LineTotal FROM
OrderHeader a
JOIN OrderDetail b ON b.SalesOrderID = a.SalesOrderID
WHERE a.SalesOrderID = '44406'

SELECT SalesOrderID, SalesOrderDetailID, ProductID, OrderQty
FROM OrderDetail

SELECT SalesOrderID, SalesOrderDetailID, ProductID, OrderQty
FROM OrderDetail
WHERE SalesOrderID='44403'

SELECT SalesOrderID, SalesOrderDetailID, ProductID, OrderQty
FROM OrderDetail
WHERE SalesOrderID='44404'

SELECT SalesOrderID, SalesOrderDetailID, ProductID, OrderQty
FROM OrderDetail
WHERE SalesOrderID='44405'

SELECT SalesOrderID, SalesOrderDetailID, ProductID, OrderQty
FROM OrderDetail
WHERE SalesOrderID='44406'
```

iv. Stop both the SQL Profiler trace and Performance Monitor Data Collector.

v. Now it is time to corelate your SQL Profiler trace against the Performance Monitor log to see if your suspicion that certain SQL activities are the culprit causing performance degradation:

  1- Launch SQL Server Profiler > File > Open > Trace File > [Select] DB1 Troubleshooting 4-5pm Sunday.trc

2- Using same SQL Server Profile > File > Import Performance Data > [Select] logs for Data Collector (DB1 Troubleshooting 4-5pm Sunday) > [Select] All counters
3- Notice that around the time the SQL batch was executing our CPU and other counters were reaching 100%. This is our culprit.



EventClass	TextData	ApplicationName	NTUserName	LoginName	CPU	Reads	Writes	Duration	ClientProcessID	SPID	StartTime	EndTime
Trace Start											2022-03-30 10:32:46...	
ExistingConnection	-- network protocol: LPC set quoted...	Database Eng...	SQLAdmin	SSLAB\...					3952	58	2022-03-30 10:19:59...	
ExistingConnection	-- network protocol: LPC set quoted...	Microsoft SQ...	SQLAdmin	SSLAB\...					2980	64	2022-03-30 10:17:37...	
SQL:BatchStarting	SELECT @@SPID;	Microsoft SQ...	SQLAdmin	SSLAB\...					2980	64	2022-03-30 10:32:54...	
SQL:BatchCompleted	SELECT @@SPID;	Microsoft SQ...	SQLAdmin	SSLAB\...	0	0	0	0	2980	64	2022-03-30 10:32:54...	2022-03-30 10:32:5
SQL:BatchStarting	USE ASSIGNMENT9; SELECT a.SalesOr...	Microsoft SQ...	SQLAdmin	SSLAB\...					2980	64	2022-03-30 10:32:54...	
SQL:BatchCompleted	USE ASSIGNMENT9; SELECT a.SalesOr...	Microsoft SQ...	SQLAdmin	SSLAB\...	204	19002	0	1739	2980	64	2022-03-30 10:32:54...	2022-03-30 10:32:5
Audit Login	-- network protocol: LPC set quoted...	SQLAgent - J...	SQLAdmin	SSLAB\...					3708	62	2022-03-30 10:33:00...	
Audit Logout		SQLAgent - J...	SQLAdmin	SSLAB\...	0	12	0	10	3708	62	2022-03-30 10:33:00...	2022-03-30 10:33:0

C...	Scale	Counter	Object	Instance	Computer	Min Value	Max Value	Avg Value	Selected Value	Selected Time
✓	20.00	Processor Queue...	System		\\DB1	0.00	5.00	1.00	5.00	3/30/2022 10:32...
✓	0.07	Available MBytes	Memory		\\DB1	1376.00	1390.00	1380.75	1377.00	3/30/2022 10:32...
✓	2.10	% Processor Time	Processor	_Total	\\DB1	1.56	47.67	11.49	47.67	3/30/2022 10:32...
✓	0.76	Pages/sec	Memory		\\DB1	0.00	131.97	24.13	131.97	3/30/2022 10:32...
✓	163590.00	Avg. Disk sec/Read	LogicalDisk	_Total	\\DB1	0.00	0.00	0.00	0.00	3/30/2022 10:32...
✓	1.00	Current Disk Qu...	LogicalDisk	_Total	\\DB1	0.00	0.00	0.00	0.00	3/30/2022 10:32...
✓	1741.83	Avg. Disk sec/W...	LogicalDisk	_Total	\\DB1	0.00	0.06	0.01	0.00	3/30/2022 10:32...

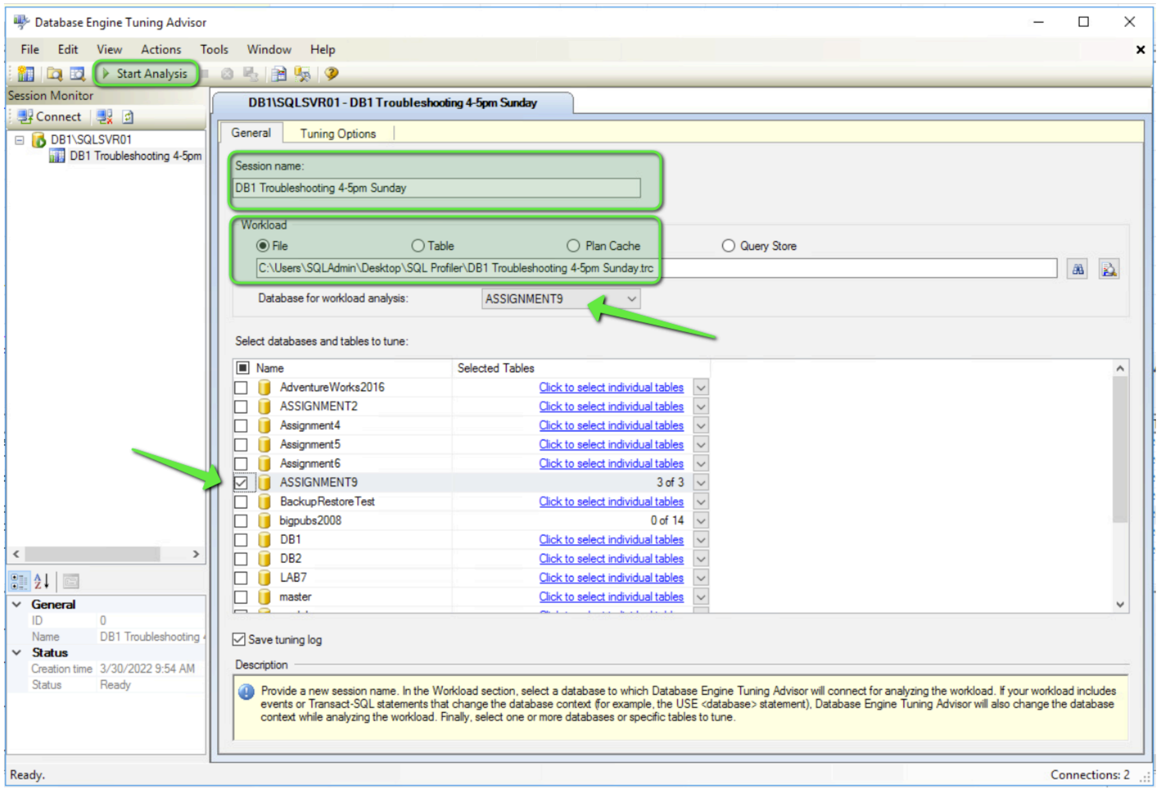
```
USE ASSIGNMENT9;

SELECT a.SalesOrderID, a.Status, a.OnlineOrderFlag, a.AccountNumber, a.CustomerID, a.ShipToAddressID, a.CreditCardID, b.LineTotal FROM OrderHeader a
JOIN OrderDetail b ON b.SalesOrderID = a.SalesOrderID

SELECT a.SalesOrderID, a.Status, a.OnlineOrderFlag, a.AccountNumber, a.CustomerID, a.ShipToAddressID, a.CreditCardID, b.LineTotal FROM OrderHeader a
JOIN OrderDetail b ON b.SalesOrderID = a.SalesOrderID
WHERE a.SalesOrderID = '44403'

SELECT a.SalesOrderID, a.Status, a.OnlineOrderFlag, a.AccountNumber, a.CustomerID, a.ShipToAddressID, a.CreditCardID, b.LineTotal FROM OrderHeader a
JOIN OrderDetail b ON b.SalesOrderID = a.SalesOrderID
WHERE a.SalesOrderID = '44404'

SELECT a.SalesOrderID, a.Status, a.OnlineOrderFlag, a.AccountNumber, a.CustomerID, a.ShipToAddressID, a.CreditCardID, b.LineTotal FROM OrderHeader a
JOIN OrderDetail b ON b.SalesOrderID = a.SalesOrderID
WHERE a.SalesOrderID = '44405'

SELECT a.SalesOrderID, a.Status, a.OnlineOrderFlag, a.AccountNumber, a.CustomerID, a.ShipToAddressID, a.CreditCardID, b.LineTotal FROM OrderHeader a
JOIN OrderDetail b ON b.SalesOrderID = a.SalesOrderID
```

Done.

4- Working with Database Engine Tuning Advisor:
   a. We know the SQL batch that runs every Sunday between 4-5pm that produces management reports is the culprit for the performance degradation. You need to present your findings to your manager along with recommendations on how to resolve the issue.
      i. Since the issue is SQL query related, we can use Database Engine Tuning Advisor to validate the tables and databases being used. Launch Database Engine Tuning Advisor >
         1- Connect to your SQL instance
         2- Session name: DB1 Troubleshooting 4-5pm Sunday
         3- Workload: File > [Select] DB1 Troubleshooting 4-5pm Sunday.trc
         4- Database for workload analysis: ASSIGNMENT9

5- Select databases and tables to tune: ASSIGNMENT9

6- Start Analysis


Database Engine Tuning Advisor
File Edit View Actions Tools Window Help
Start Analysis
Session Monitor
Connect
DB1\SQLSVR01
DB1 Troubleshooting 4-5pm
General
Tuning Options
Ready.
Connections: 2

DB1\SQLSVR01 - DB1 Troubleshooting 4-5pm Sunday

Session name: DB1 Troubleshooting 4-5pm Sunday

Workload: [✓ File] Table Plan Cache Query Store

C:\Users\SQLAdmin\Desktop\SQL Profiler\DB1 Troubleshooting 4-5pm Sunday.trc

Database for workload analysis: ASSIGNMENT9

Select databases and tables to tune:

Name	Selected Tables
☐ AdventureWorks2016	Click to select individual tables
☐ ASSIGNMENT2	Click to select individual tables
☐ Assignment4	Click to select individual tables
☐ Assignment5	Click to select individual tables
☐ Assignment6	Click to select individual tables
☑ ASSIGNMENT9	3 of 3
☐ BackupRestoreTest	Click to select individual tables
☐ bigpubs2008	0 of 14
☐ DB1	Click to select individual tables
☐ DB2	Click to select individual tables
☐ LAB7	Click to select individual tables
☐ master	Click to select individual tables

☑ Save tuning log

Description

Provide a new session name. In the Workload section, select a database to which Database Engine Tuning Advisor will connect for analyzing the workload. If your workload includes events or Transact-SQL statements that change the database context (for example, the USE <database> statement), Database Engine Tuning Advisor will also change the database context while analyzing the workload. Finally, select one or more databases or specific tables to tune.

General: ID 0; Name DB1 Troubleshooting

Status: Creation time 3/30/2022 9:54 AM; Status Ready

ii. [ASSESSMENT]: What are the recommendations that Database Engine Tuning Advisor suggested?

iii. [ASSESSMENT]: How much of an improvement would these suggestions provide?

iv. Apply the recommendations from Database Engine Tuning Advisor:
Database Engine Tuning Advisor > Actions > Apply Recommendations

CONESTOGA

5- Working with and managing deadlocks and blocking locks:

a. Execute an SQL transaction that selects OrderDetail and updates OrderHeader tables in ASSIGNMENT9 database:

```
USE ASSIGNMENT9
DECLARE @SalesOrderId BIGINT;
DECLARE @SumValue BIGINT;
SET @SalesOrderId = 43659;

BEGIN TRANSACTION

SELECT @SumValue = SUM(LineTotal) -- we will round up to next int
FROM dbo.OrderDetail
WHERE SalesOrderId = @SalesOrderId

UPDATE dbo.OrderHeader
SET SubTotal = @SumValue , TaxAmt   = @SumValue * .21
WHERE SalesOrderId = @SalesOrderId;
```

b. Execute an SQL transaction that updates OrderDetail table (LineTotal) in ASSIGNMENT9 database:

```
USE ASSIGNMENT9
DECLARE @SalesOrderId BIGINT;
DECLARE @SumValue BIGINT;
SET @SalesOrderId = 43659;

BEGIN TRANSACTION

UPDATE dbo.OrderDetail SET LineTotal = 0 WHERE SalesOrderId =
@SalesOrderId;
```

c. [Again] Execute an SQL transaction that updates OrderDetail table (LineTotal) in ASSIGNMENT9 database:

```
USE ASSIGNMENT9
DECLARE @SalesOrderId BIGINT;
DECLARE @SumValue BIGINT;
SET @SalesOrderId = 43659;

BEGIN TRANSACTION
```

```
UPDATE dbo.OrderDetail SET LineTotal = 0 WHERE SalesOrderId =
@SalesOrderId ;
```

d. [Again] Execute an SQL transaction that selects OrderDetail and updates OrderHeader tables in ASSIGNMENT9 database:

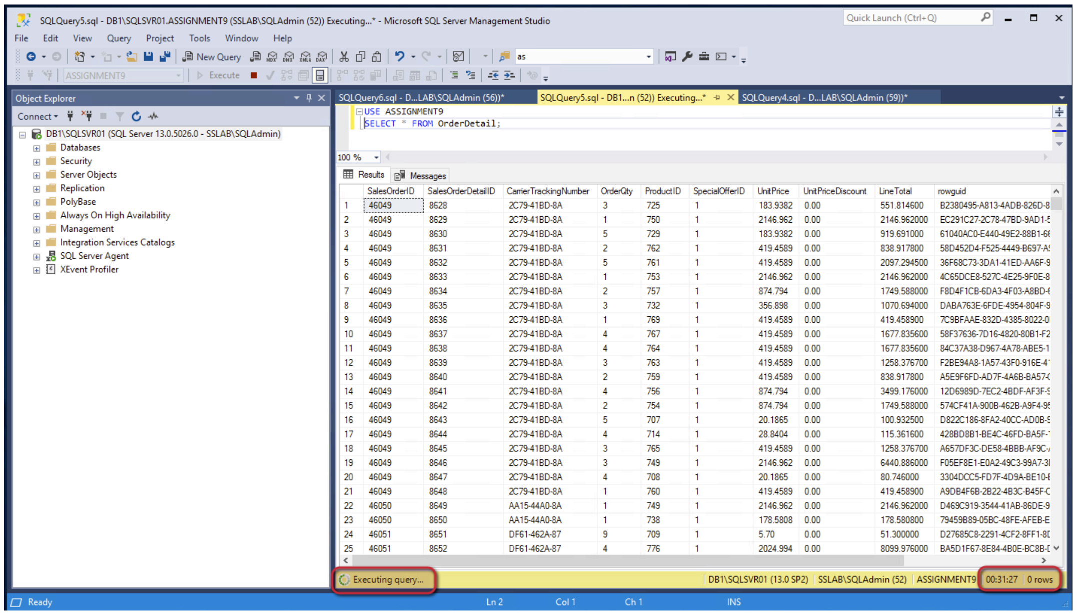
```
USE ASSIGNMENT9
DECLARE @SalesOrderId BIGINT;
DECLARE @SumValue BIGINT;
SET @SalesOrderId = 43659;

BEGIN TRANSACTION

SELECT @SumValue = SUM(LineTotal) -- we will round up to next int
FROM dbo.OrderDetail
WHERE SalesOrderId = @SalesOrderId

UPDATE dbo.OrderHeader
SET SubTotal = @SumValue , TaxAmt   = @SumValue * .21
WHERE SalesOrderId = @SalesOrderId
```

e. Now commit the above transactions and observe the deadlock:

```
COMMIT TRANSACTION
```

i. Open a New Query window and run:

```
SELECT * FROM OrderDetail;
```



SQLQuery5.sql - DB1\SQLSVR01.ASSIGNMENT9 (SSLAB\SQLAdmin (52)) Executing...* - Microsoft SQL Server Management Studio

```sql
USE ASSIGNMENT9
SELECT * FROM OrderDetail;
```

	SalesOrderID	SalesOrderDetailID	CarrierTrackingNumber	OrderQty	ProductID	SpecialOfferID	UnitPrice	UnitPriceDiscount	LineTotal	rowguid
1	46049	8628	2C79-41BD-8A	3	725	1	183.9382	0.00	551.814600	B2380495-A813-4ADB-826D-8
2	46049	8629	2C79-41BD-8A	1	750	1	2146.962	0.00	2146.962000	EC291C27-2C78-47BD-9AD1-5
3	46049	8630	2C79-41BD-8A	5	729	1	183.9382	0.00	919.691000	61040AC0-E440-49E2-88B1-66
4	46049	8631	2C79-41BD-8A	2	762	1	419.4589	0.00	838.917800	58D452D4-F525-4449-B697-A
5	46049	8632	2C79-41BD-8A	5	761	1	419.4589	0.00	2097.294500	36F68C73-3DA1-41ED-AA6F-9
6	46049	8633	2C79-41BD-8A	1	753	1	2146.962	0.00	2146.962000	4C65DCE8-527C-4E25-9F0E-8
7	46049	8634	2C79-41BD-8A	2	757	1	874.794	0.00	1749.588000	F8D4F1CB-6DA3-4F03-A8BD-6
8	46049	8635	2C79-41BD-8A	3	732	1	356.898	0.00	1070.694000	DABA763E-6FDE-4954-804F-9
9	46049	8636	2C79-41BD-8A	1	769	1	419.4589	0.00	419.458900	7C9BFAAE-832D-4385-8022-0
10	46049	8637	2C79-41BD-8A	4	767	1	419.4589	0.00	1677.835600	58F37636-7D16-4820-80B1-F2
11	46049	8638	2C79-41BD-8A	4	764	1	419.4589	0.00	1677.835600	84C37A38-D967-4A78-ABE5-1
12	46049	8639	2C79-41BD-8A	3	763	1	419.4589	0.00	1258.376700	F2BE94A8-1A57-43F0-916E-4
13	46049	8640	2C79-41BD-8A	2	759	1	419.4589	0.00	838.917800	A5E9F6FD-AD7F-4A6B-BA57-0
14	46049	8641	2C79-41BD-8A	4	756	1	874.794	0.00	3499.176000	12D6989D-7EC2-4BDF-AF3F-9
15	46049	8642	2C79-41BD-8A	2	754	1	874.794	0.00	1749.588000	574CF41A-900B-462B-A9F4-95
16	46049	8643	2C79-41BD-8A	5	707	1	20.1865	0.00	100.932500	D822C186-8FA2-40CC-AD0B-9
17	46049	8644	2C79-41BD-8A	4	714	1	28.8404	0.00	115.361600	428BD8B1-BE4C-46FD-BA5F-1
18	46049	8645	2C79-41BD-8A	3	765	1	419.4589	0.00	1258.376700	A657DF3C-DE58-4BBB-AF9C-A
19	46049	8646	2C79-41BD-8A	3	749	1	2146.962	0.00	6440.886000	F05EF8E1-E0A2-49C3-99A7-3
20	46049	8647	2C79-41BD-8A	4	708	1	20.1865	0.00	80.746000	3304DCC5-FD7F-4D9A-BE10-E
21	46049	8648	2C79-41BD-8A	1	760	1	419.4589	0.00	419.458900	A9DB4F6B-2B22-4B3C-B45F-C
22	46050	8649	AA15-44A0-8A	1	749	1	2146.962	0.00	2146.962000	D469C919-3544-41AB-86DE-9
23	46050	8650	AA15-44A0-8A	1	738	1	178.5808	0.00	178.580800	79459B89-05BC-48FE-AFEB-E
24	46051	8651	DF61-462A-87	9	709	1	5.70	0.00	51.300000	D27685C8-2291-4CF2-8FF1-8D
25	46051	8652	DF61-462A-87	4	776	1	2024.994	0.00	8099.976000	BA5D1F67-8E84-4B0E-BC8B-D

Executing query... | DB1\SQLSVR01 (13.0 SP2) | SSLAB\SQLAdmin (52) | ASSIGNMENT9 | 00:31:27 | 0 rows

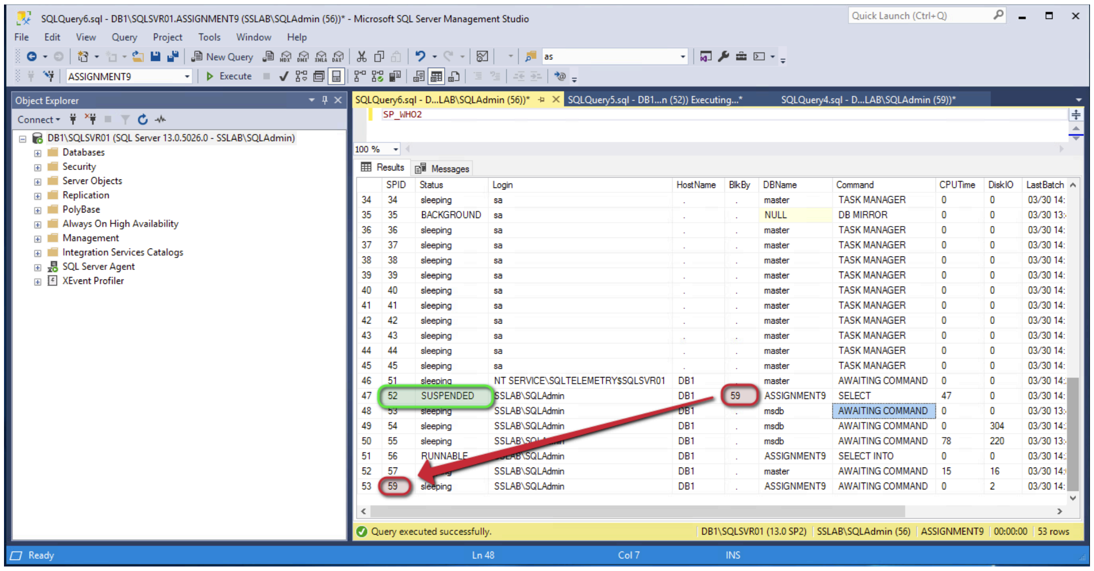
ii. Congratulations, you have caused a deadlock!

iii. Let's identify the process IDs of the deadlocks:

1. Open a New Query window and run:

```sql
SP_WHO2;
```

2. Locate the blocking process ID (BlkBy):



SQLQuery6.sql - DB1\SQLSVR01.ASSIGNMENT9 (SSLAB\SQLAdmin (56))* - Microsoft SQL Server Management Studio

```sql
SP_WHO2
```

	SPID	Status	Login	HostName	BlkBy	DBName	Command	CPUTime	DiskIO	LastBatch
34	34	sleeping	sa	.	.	master	TASK MANAGER	0	0	03/30 14:
35	35	BACKGROUND	sa	.	.	NULL	DB MIRROR	0	0	03/30 13:
36	36	sleeping	sa	.	.	master	TASK MANAGER	0	0	03/30 14:
37	37	sleeping	sa	.	.	master	TASK MANAGER	0	0	03/30 14:
38	38	sleeping	sa	.	.	master	TASK MANAGER	0	0	03/30 14:
39	39	sleeping	sa	.	.	master	TASK MANAGER	0	0	03/30 14:
40	40	sleeping	sa	.	.	master	TASK MANAGER	0	0	03/30 14:
41	41	sleeping	sa	.	.	master	TASK MANAGER	0	0	03/30 14:
42	42	sleeping	sa	.	.	master	TASK MANAGER	0	0	03/30 14:
43	43	sleeping	sa	.	.	master	TASK MANAGER	0	0	03/30 14:
44	44	sleeping	sa	.	.	master	TASK MANAGER	0	0	03/30 14:
45	45	sleeping	sa	.	.	master	TASK MANAGER	0	0	03/30 14:
46	51	sleeping	NT SERVICE\SQLTELEMETRY$SQLSVR01	DB1	.	master	AWAITING COMMAND	0	0	03/30 14:
47	52	SUSPENDED	SSLAB\SQLAdmin	DB1	59	ASSIGNMENT9	SELECT	47	0	03/30 14:
48	53	sleeping	SSLAB\SQLAdmin	DB1	.	msdb	AWAITING COMMAND	0	0	03/30 13:
49	54	sleeping	SSLAB\SQLAdmin	DB1	.	msdb	AWAITING COMMAND	0	304	03/30 14:
50	55	sleeping	SSLAB\SQLAdmin	DB1	.	msdb	AWAITING COMMAND	78	220	03/30 13:
51	56	RUNNABLE	SSLAB\SQLAdmin	DB1	.	ASSIGNMENT9	SELECT INTO	0	0	03/30 14:
52	57	sleeping	SSLAB\SQLAdmin	DB1	.	master	AWAITING COMMAND	15	16	03/30 14:
53	59	sleeping	SSLAB\SQLAdmin	DB1	.	ASSIGNMENT9	AWAITING COMMAND	0	2	03/30 14:

Query executed successfully. | DB1\SQLSVR01 (13.0 SP2) | SSLAB\SQLAdmin (56) | ASSIGNMENT9 | 00:00:00 | 53 rows

iv. Identify which database objects are causing the deadlock:

```sql
SP_LOCK;
```

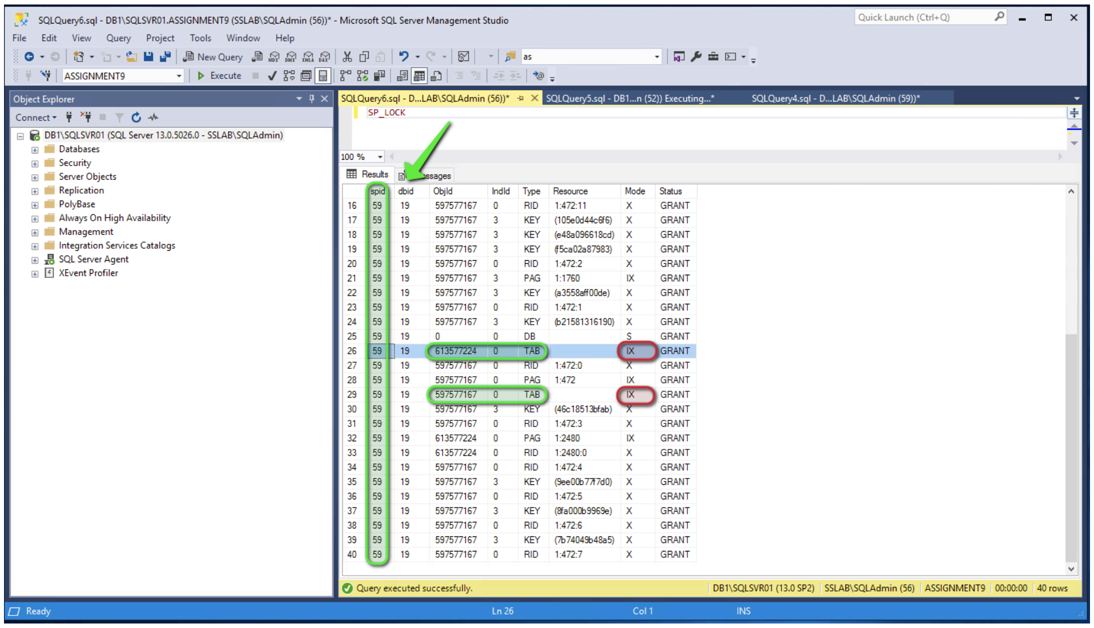
1. Locate the blocking process ID, database ID (dbid), Object ID (ObjId), object type (Type), and block type (Mode):
   - Blocking Type:

i. S/IS = Shared/Intent Shared

ii. X/IX = Exclusive/Intent Exclusive

- Object Type:

  i. TAB = Table



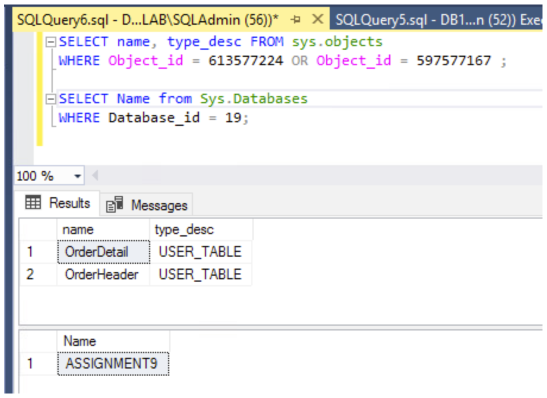
```
SP_LOCK
```

	spid	dbid	ObjId	IndId	Type	Resource	Mode	Status
16	59	19	597577167	0	RID	1:472:11	X	GRANT
17	59	19	597577167	3	KEY	(105e0d44c6f6)	X	GRANT
18	59	19	597577167	3	KEY	(e48a096618cd)	X	GRANT
19	59	19	597577167	3	KEY	(f5ca02a87983)	X	GRANT
20	59	19	597577167	0	RID	1:472:2	X	GRANT
21	59	19	597577167	3	PAG	1:1760	IX	GRANT
22	59	19	597577167	3	KEY	(a3558aff00de)	X	GRANT
23	59	19	597577167	0	RID	1:472:1	X	GRANT
24	59	19	597577167	3	KEY	(b21581316190)	X	GRANT
25	59	19	0	0	DB		S	GRANT
26	59	19	613577224	0	TAB		IX	GRANT
27	59	19	597577167	0	RID	1:472:0	X	GRANT
28	59	19	597577167	0	PAG	1:472	IX	GRANT
29	59	19	597577167	0	TAB		IX	GRANT
30	59	19	597577167	3	KEY	(46c18513bfab)	X	GRANT
31	59	19	597577167	0	RID	1:472:3	X	GRANT
32	59	19	613577224	0	PAG	1:2480	IX	GRANT
33	59	19	613577224	0	RID	1:2480:0	X	GRANT
34	59	19	597577167	0	RID	1:472:4	X	GRANT
35	59	19	597577167	3	KEY	(9ee00b77f7d0)	X	GRANT
36	59	19	597577167	0	RID	1:472:5	X	GRANT
37	59	19	597577167	3	KEY	(8fa000b9969e)	X	GRANT
38	59	19	597577167	0	RID	1:472:6	X	GRANT
39	59	19	597577167	3	KEY	(7b74049b48a5)	X	GRANT
40	59	19	597577167	0	RID	1:472:7	X	GRANT

Query executed successfully.

2. Identify the blocking database objects (use Object IDs and Database IDs from above):

```
SELECT name, type_desc FROM sys.objects
WHERE Object_id = 613577224 OR Object_id = 597577167 ;

SELECT Name from Sys.Databases
WHERE Database_id = 19;
```



```
SELECT name, type_desc FROM sys.objects
WHERE Object_id = 613577224 OR Object_id = 597577167 ;

SELECT Name from Sys.Databases
WHERE Database_id = 19;
```

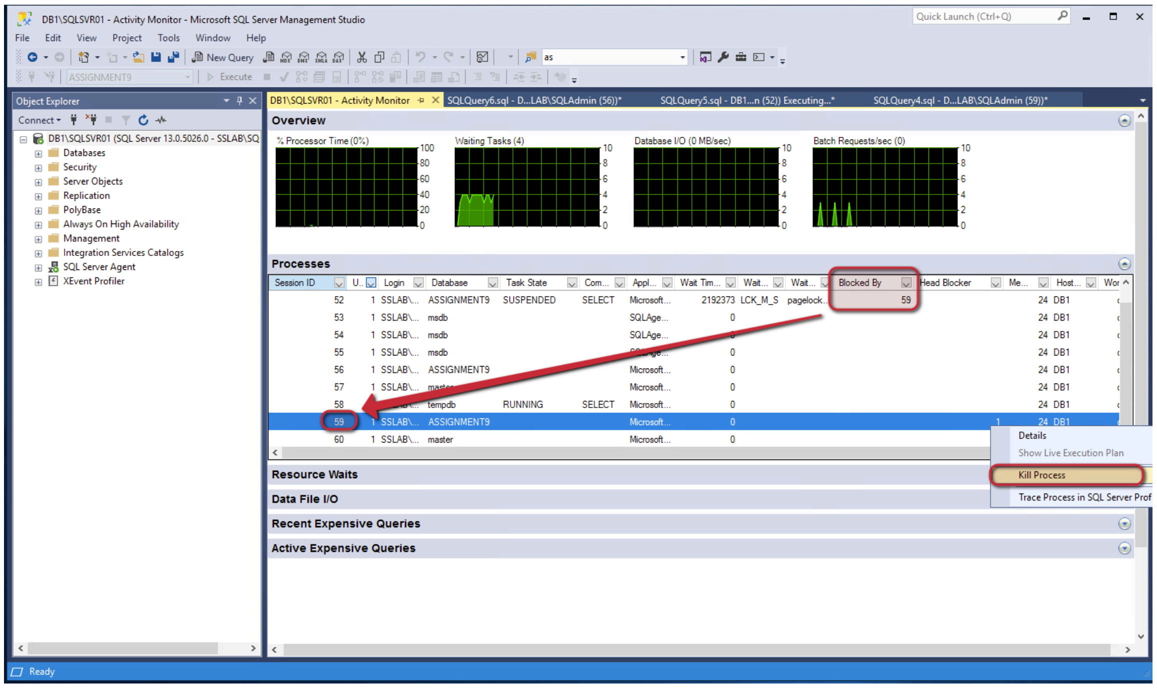
	name	type_desc
1	OrderDetail	USER_TABLE
2	OrderHeader	USER_TABLE

	Name
1	ASSIGNMENT9

f. Resolve the deadlock by killing the process that is causing it:

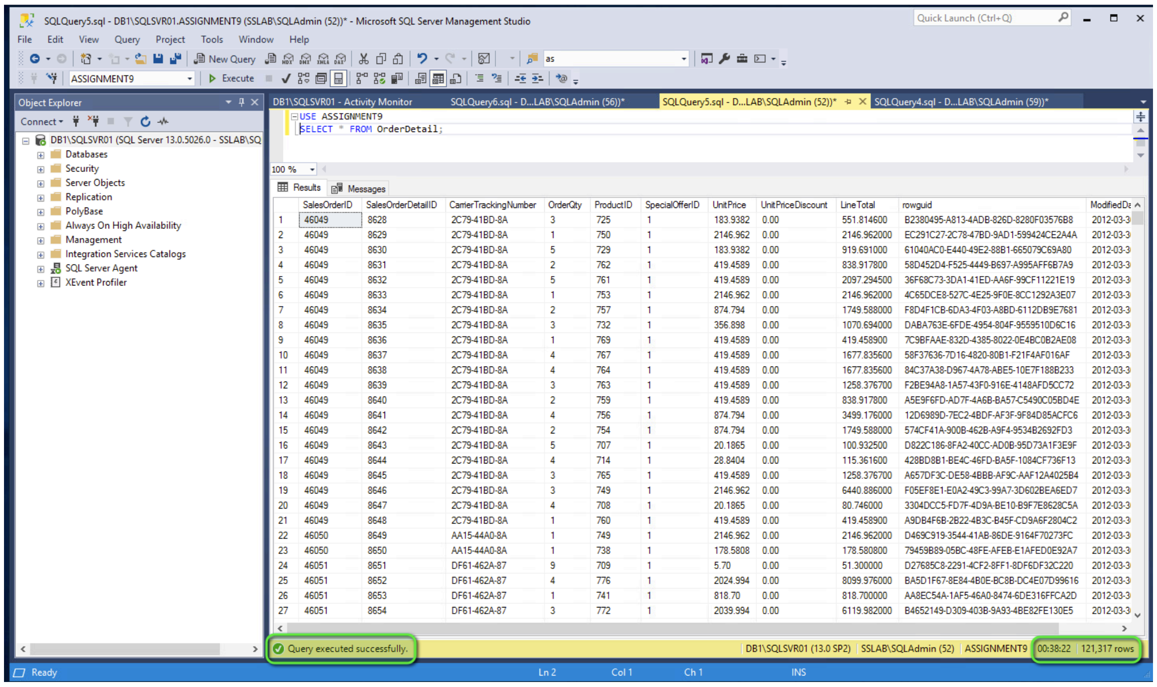
INFO1085 - SQL Server

SSMS > [Right-click] SQL Instance > Activity Monitor > Processes > [Right-click] Blocking Session ID > Kill Process:



Session ID	U..	Login	Database	Task State	Com...	Appl...	Wait Tim...	Wait...	Wait...	Blocked By	Head Blocker	Me...	Host...
52	1	SSLAB\...	ASSIGNMENT9	SUSPENDED	SELECT	Microsoft...	2192373	LCK_M_S	pagelock...	59		24	DB1
53	1	SSLAB\...	msdb			SQLAge...	0					24	DB1
54	1	SSLAB\...	msdb			SQLAge...	0					24	DB1
55	1	SSLAB\...	msdb			SQLAge...	0					24	DB1
56	1	SSLAB\...	ASSIGNMENT9			Microsoft...	0					24	DB1
57	1	SSLAB\...	master			Microsoft...	0					24	DB1
58	1	SSLAB\...	tempdb	RUNNING	SELECT	Microsoft...	0					24	DB1
59	1	SSLAB\...	ASSIGNMENT9			Microsoft...	0				1	24	DB1
60	1	SSLAB\...	master			Microsoft...	0						

g. Return to your suspended query, what do you see now?



SQLQuery5.sql - DB1\SQLSVR01.ASSIGNMENT9 (SSLAB\SQLAdmin (52))* - Microsoft SQL Server Management Studio
DB1\SQLSVR01 - Activity Monitor
SQLQuery6.sql - D...LAB\SQLAdmin (56))*
SQLQuery5.sql - D...LAB\SQLAdmin (52))*
SQLQuery4.sql - D...LAB\SQLAdmin (59))*
Results
Messages

```sql
USE ASSIGNMENT9
SELECT * FROM OrderDetail;
```

	SalesOrderID	SalesOrderDetailID	CarrierTrackingNumber	OrderQty	ProductID	SpecialOfferID	UnitPrice	UnitPriceDiscount	LineTotal	rowguid	ModifiedDa
1	46049	8628	2C79-41BD-8A	3	725	1	183.9382	0.00	551.814600	B2380495-A813-4ADB-826D-8280F03576B8	2012-03-3
2	46049	8629	2C79-41BD-8A	1	750	1	2146.962	0.00	2146.962000	EC291C27-2C78-47BD-9AD1-599424CE2A4A	2012-03-3
3	46049	8630	2C79-41BD-8A	5	729	1	183.9382	0.00	919.691000	61040AC0-E440-49E2-88B1-665079C69A80	2012-03-3
4	46049	8631	2C79-41BD-8A	2	762	1	419.4589	0.00	838.917800	58D452D4-F525-4449-B697-A995AFF6B7A9	2012-03-3
5	46049	8632	2C79-41BD-8A	5	761	1	419.4589	0.00	2097.294500	36F68C73-3DA1-41ED-AA6F-99CF11221E19	2012-03-3
6	46049	8633	2C79-41BD-8A	1	753	1	2146.962	0.00	2146.962000	4C65DCE8-527C-4E25-9F0E-8CC1292A3E07	2012-03-3
7	46049	8634	2C79-41BD-8A	2	757	1	874.794	0.00	1749.588000	F8D4F1CB-6DA3-4F03-A8BD-6112DB9E7681	2012-03-3
8	46049	8635	2C79-41BD-8A	3	732	1	356.898	0.00	1070.694000	DABA763E-6FDE-4954-804F-9559510D6C16	2012-03-3
9	46049	8636	2C79-41BD-8A	1	769	1	419.4589	0.00	419.458900	7C9BFAAE-832D-4385-8022-0E4BC0B2AE08	2012-03-3
10	46049	8637	2C79-41BD-8A	4	767	1	419.4589	0.00	1677.835600	58F37636-7D16-4820-80B1-F21F4AF016AF	2012-03-3
11	46049	8638	2C79-41BD-8A	4	764	1	419.4589	0.00	1677.835600	84C37A38-D967-4A78-ABE5-10E7F188B233	2012-03-3
12	46049	8639	2C79-41BD-8A	3	763	1	419.4589	0.00	1258.376700	F2BE94A8-1A57-43F0-916E-4148AFD5CC72	2012-03-3
13	46049	8640	2C79-41BD-8A	2	759	1	419.4589	0.00	838.917800	A5E9F6FD-AD7F-4A6B-BA57-C5490C05BD4E	2012-03-3
14	46049	8641	2C79-41BD-8A	4	756	1	874.794	0.00	3499.176000	12D6989D-7EC2-4BDF-AF3F-9F84D85ACFC6	2012-03-3
15	46049	8642	2C79-41BD-8A	2	754	1	874.794	0.00	1749.588000	574CF41A-900B-462B-A9F4-9534B2692FD3	2012-03-3
16	46049	8643	2C79-41BD-8A	5	707	1	20.1865	0.00	100.932500	D822C186-8FA2-40CC-AD0B-95D73A1F3E9F	2012-03-3
17	46049	8644	2C79-41BD-8A	4	714	1	28.8404	0.00	115.361600	428BD8B1-BE4C-46FD-BA5F-1084CF736F13	2012-03-3
18	46049	8645	2C79-41BD-8A	3	765	1	419.4589	0.00	1258.376700	A657DF3C-DE58-4BBB-AF9C-AAF12A4025B4	2012-03-3
19	46049	8646	2C79-41BD-8A	3	749	1	2146.962	0.00	6440.886000	F05EF8E1-E0A2-49C3-99A7-3D602BEA6ED7	2012-03-3
20	46049	8647	2C79-41BD-8A	4	708	1	20.1865	0.00	80.746000	3304DCC5-FD7F-4D9A-BE10-B9F7E8628C5A	2012-03-3
21	46049	8648	2C79-41BD-8A	1	760	1	419.4589	0.00	419.458900	A9DB4F6B-2B22-4B3C-B45F-CD9A6F2804C2	2012-03-3
22	46050	8649	AA15-44A0-8A	1	749	1	2146.962	0.00	2146.962000	D469C919-3544-41AB-86DE-9164F70273FC	2012-03-3
23	46050	8650	AA15-44A0-8A	1	738	1	178.5808	0.00	178.580800	79459B89-05BC-48FE-AFEB-E1AFED0E92A7	2012-03-3
24	46051	8651	DF61-462A-87	9	709	1	5.70	0.00	51.300000	D27685C8-2291-4CF2-8FF1-8DF6DF32C220	2012-03-3
25	46051	8652	DF61-462A-87	4	776	1	2024.994	0.00	8099.976000	BA5D1F67-8E84-4B0E-BC8B-DC4E07D99616	2012-03-3
26	46051	8653	DF61-462A-87	1	741	1	818.70	0.00	818.700000	AA8EC54A-1AF5-46A0-8474-6DE316FFCA2D	2012-03-3
27	46051	8654	DF61-462A-87	3	772	1	2039.994	0.00	6119.982000	B4652149-D309-403B-9A93-4BE82FE130E5	2012-03-3

```
Query executed successfully.    DB1\SQLSVR01 (13.0 SP2) | SSLAB\SQLAdmin (52) | ASSIGNMENT9 | 00:38:22 | 121,317 rows
Ready    Ln 2    Col 1    Ch 1    INS
```