(T26)尋找BlockingProcess(被鎖的流程)  
CourseGUID: e48417fc-9db5-4e99-822c-706c5ccef6cc  
=======================================================================  
(T26)尋找BlockingProcess(被鎖的流程)  
=======================================================================  
0. Summary

1. Create Sample Data

2. Display the oldest active transaction information.

4. Clean up  
=======================================================================

0. Summary

--Display all the active transaction information.

SELECT

    [s\_tst].[session\_id],

    [s\_es].[login\_name] AS [Login Name],

    DB\_NAME (s\_tdt.database\_id) AS [Database],

    [s\_tdt].[database\_transaction\_begin\_time] AS [Begin Time],

    [s\_tdt].[database\_transaction\_log\_bytes\_used] AS [Log Bytes],

    [s\_tdt].[database\_transaction\_log\_bytes\_reserved] AS [Log Rsvd],

    [s\_est].text AS [Last T-SQL Text],

    [s\_eqp].[query\_plan] AS [Last Plan]

FROM

    sys.dm\_tran\_database\_transactions [s\_tdt]

JOIN

    sys.dm\_tran\_session\_transactions [s\_tst]

ON

    [s\_tst].[transaction\_id] = [s\_tdt].[transaction\_id]

JOIN

    sys.[dm\_exec\_sessions] [s\_es]

ON

    [s\_es].[session\_id] = [s\_tst].[session\_id]

JOIN

    sys.dm\_exec\_connections [s\_ec]

ON

    [s\_ec].[session\_id] = [s\_tst].[session\_id]

LEFT OUTER JOIN

    sys.dm\_exec\_requests [s\_er]

ON

    [s\_er].[session\_id] = [s\_tst].[session\_id]

CROSS APPLY

    sys.dm\_exec\_sql\_text ([s\_ec].[most\_recent\_sql\_handle]) AS [s\_est]

OUTER APPLY

    sys.dm\_exec\_query\_plan ([s\_er].[plan\_handle]) AS [s\_eqp]

ORDER BY

    [Begin Time] ASC;

GO -- Run the previous command and begins new batch

==================================================

1. Create Sample Data

--==============================================================

--T026\_01\_01

--Create Sample Data

IF ( EXISTS ( SELECT    \*

              FROM      INFORMATION\_SCHEMA.TABLES

              WHERE     TABLE\_NAME = 'TableA' ) )

    BEGIN

        TRUNCATE TABLE dbo.TableA;

        DROP TABLE TableA;

    END;

GO -- Run the previous command and begins new batch

IF ( EXISTS ( SELECT    \*

              FROM      INFORMATION\_SCHEMA.TABLES

              WHERE     TABLE\_NAME = 'TableB' ) )

    BEGIN

        TRUNCATE TABLE dbo.TableB;

        DROP TABLE TableB;

    END;

GO -- Run the previous command and begins new batch

CREATE TABLE TableA

(

  ID INT IDENTITY

         PRIMARY KEY ,

  Name NVARCHAR(50)

);

       GO -- Run the previous command and begins new batch

INSERT  INTO TableA

VALUES  ( 'TableAName1' );

INSERT  INTO TableA

VALUES  ( 'TableAName2' );

INSERT  INTO TableA

VALUES  ( 'TableAName3' );

INSERT  INTO TableA

VALUES  ( 'TableAName4' );

INSERT  INTO TableA

VALUES  ( 'TableAName5' );

GO -- Run the previous command and begins new batch

CREATE TABLE TableB

(

  ID INT IDENTITY

         PRIMARY KEY ,

  Name NVARCHAR(50)

);

INSERT  INTO TableB

VALUES  ( 'TableBName1' );

INSERT  INTO TableB

VALUES  ( 'TableBName2' );

INSERT  INTO TableB

VALUES  ( 'TableBName3' );

INSERT  INTO TableB

VALUES  ( 'TableBName4' );

INSERT  INTO TableB

VALUES  ( 'TableBName5' );

GO -- Run the previous command and begins new batch

SELECT  \*

FROM    TableA;

SELECT  \*

FROM    TableB;

GO -- Run the previous command and begins new batch

Graphical user interface, application, table

Description automatically generated with medium confidence

2. Display the oldest active transaction information.

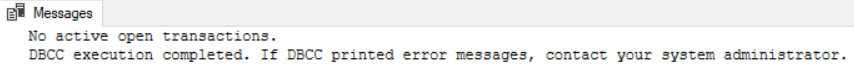
--==============================================================

--T026\_01\_02

--display the oldest active transaction information.

DBCC OPENTRAN;

GO -- Run the previous command and begins new batch



/\*

Output

--No active open transactions.

--DBCC execution completed. If DBCC printed error messages, contact your system administrator.

\*/

3. Create Blocking Query

--==============================================================

--T026\_01\_03

--Create Blocking Query

-------------------------------------------------------------

--T026\_01\_03\_01

--Transaction1

BEGIN TRAN;

UPDATE  TableA

SET     Name += ' Tran1'

WHERE   ID = 1;

--Thus, TableA will be block.

/\*

Open another query windows to execute the query.

Begin a transaction and do not commit.

Thus, TableA will be locked by this uncommited transaction.

All other transaction will not be able to use TableA.

\*/

-------------------------------------------------------------

--T026\_01\_03\_02

--Transaction2

--T026\_01\_03\_03\_01

SELECT  COUNT(\*)

FROM    TableA;

--T026\_01\_03\_03\_02

DELETE  FROM TableA

WHERE   ID = 1;

--T026\_01\_03\_03\_03

TRUNCATE TABLE TableA;

--T026\_01\_03\_03\_04

DROP TABLE TableA;

/\*

Open another query windows to execute the query.

Execute 1~4 query in Transaction2 separately.

All these query will be blocked

by the previous uncommited transaction, Transaction1.

TableA has beem locked by Transaction1.

\*/

-------------------------------------------------------------

--T026\_01\_03\_03

--Transaction3

--Create Blocking Query

BEGIN TRAN;

UPDATE  dbo.TableB

SET     Name += ' Tran3'

WHERE   ID = 1;

--Thus, TableB will be block.

/\*

Open another query windows to execute the query.

Begin another transaction and do not commit.

Thus, TableB will be locked by this uncommited transaction.

All other transaction will not be able to use TableB.

\*/

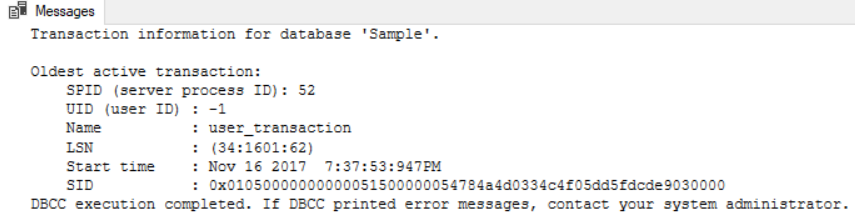
-------------------------------------------------------------

--T026\_01\_03\_04

--Transaction4

--display the oldest active transaction information.

DBCC OPENTRAN;



--Display all the active transaction information.

SELECT

    [s\_tst].[session\_id],

    [s\_es].[login\_name] AS [Login Name],

    DB\_NAME (s\_tdt.database\_id) AS [Database],

    [s\_tdt].[database\_transaction\_begin\_time] AS [Begin Time],

    [s\_tdt].[database\_transaction\_log\_bytes\_used] AS [Log Bytes],

    [s\_tdt].[database\_transaction\_log\_bytes\_reserved] AS [Log Rsvd],

    [s\_est].text AS [Last T-SQL Text],

    [s\_eqp].[query\_plan] AS [Last Plan]

FROM

    sys.dm\_tran\_database\_transactions [s\_tdt]

JOIN

    sys.dm\_tran\_session\_transactions [s\_tst]

ON

    [s\_tst].[transaction\_id] = [s\_tdt].[transaction\_id]

JOIN

    sys.[dm\_exec\_sessions] [s\_es]

ON

    [s\_es].[session\_id] = [s\_tst].[session\_id]

JOIN

    sys.dm\_exec\_connections [s\_ec]

ON

    [s\_ec].[session\_id] = [s\_tst].[session\_id]

LEFT OUTER JOIN

    sys.dm\_exec\_requests [s\_er]

ON

    [s\_er].[session\_id] = [s\_tst].[session\_id]

CROSS APPLY

    sys.dm\_exec\_sql\_text ([s\_ec].[most\_recent\_sql\_handle]) AS [s\_est]

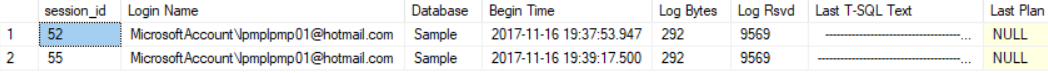
OUTER APPLY

    sys.dm\_exec\_query\_plan ([s\_er].[plan\_handle]) AS [s\_eqp]

ORDER BY

    [Begin Time] ASC;

GO -- Run the previous command and begins new batch



/\*

--DBCC OpenTran

Reference:

<https://docs.microsoft.com/en-us/sql/t-sql/database-console-commands/dbcc-opentran-transact-sql>

<https://www.sqlskills.com/blogs/paul/script-open-transactions-with-text-and-plans/>

DBCC OPENTRAN can only display

the oldest active transaction information.

If you want to see all the active transaction information,

You have to use the following script.

--SELECT

--    [s\_tst].[session\_id],

--    [s\_es].[login\_name] AS [Login Name],

--    DB\_NAME (s\_tdt.database\_id) AS [Database],

--    [s\_tdt].[database\_transaction\_begin\_time] AS [Begin Time],

--    [s\_tdt].[database\_transaction\_log\_bytes\_used] AS [Log Bytes],

--    [s\_tdt].[database\_transaction\_log\_bytes\_reserved] AS [Log Rsvd],

--    [s\_est].text AS [Last T-SQL Text],

--    [s\_eqp].[query\_plan] AS [Last Plan]

--FROM

--    sys.dm\_tran\_database\_transactions [s\_tdt]

--JOIN

--    sys.dm\_tran\_session\_transactions [s\_tst]

--ON

--    [s\_tst].[transaction\_id] = [s\_tdt].[transaction\_id]

--JOIN

--    sys.[dm\_exec\_sessions] [s\_es]

--ON

--    [s\_es].[session\_id] = [s\_tst].[session\_id]

--JOIN

--    sys.dm\_exec\_connections [s\_ec]

--ON

--    [s\_ec].[session\_id] = [s\_tst].[session\_id]

--LEFT OUTER JOIN

--    sys.dm\_exec\_requests [s\_er]

--ON

--    [s\_er].[session\_id] = [s\_tst].[session\_id]

--CROSS APPLY

--    sys.dm\_exec\_sql\_text ([s\_ec].[most\_recent\_sql\_handle]) AS [s\_est]

--OUTER APPLY

--    sys.dm\_exec\_query\_plan ([s\_er].[plan\_handle]) AS [s\_eqp]

--ORDER BY

--    [Begin Time] ASC;

--GO

2.1.

Output

--session\_id ...

--81...

--60...

\*/

-------------------------------------------------------------

--T026\_01\_03\_05

--Go back to the query window of

--the Transaction1:

--and perform COMMIT;

COMMIT;

-------------------------------------------------------------

--T026\_01\_03\_06

--Go back to the query window of

--the Transaction3:

--and perform COMMIT;

COMMIT;

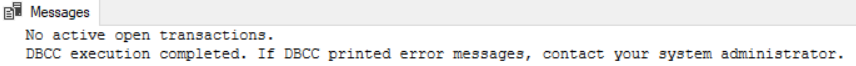
-------------------------------------------------------------

--T026\_01\_03\_07

--Transaction4

--display the oldest active transaction information.

DBCC OPENTRAN;



/\*

Output

--No active open transactions.

--DBCC execution completed.

--If DBCC printed error messages,

--contact your system administrator.

\*/

--Display all the active transaction information.

SELECT

    [s\_tst].[session\_id],

    [s\_es].[login\_name] AS [Login Name],

    DB\_NAME (s\_tdt.database\_id) AS [Database],

    [s\_tdt].[database\_transaction\_begin\_time] AS [Begin Time],

    [s\_tdt].[database\_transaction\_log\_bytes\_used] AS [Log Bytes],

    [s\_tdt].[database\_transaction\_log\_bytes\_reserved] AS [Log Rsvd],

    [s\_est].text AS [Last T-SQL Text],

    [s\_eqp].[query\_plan] AS [Last Plan]

FROM

    sys.dm\_tran\_database\_transactions [s\_tdt]

JOIN

    sys.dm\_tran\_session\_transactions [s\_tst]

ON

    [s\_tst].[transaction\_id] = [s\_tdt].[transaction\_id]

JOIN

    sys.[dm\_exec\_sessions] [s\_es]

ON

    [s\_es].[session\_id] = [s\_tst].[session\_id]

JOIN

    sys.dm\_exec\_connections [s\_ec]

ON

    [s\_ec].[session\_id] = [s\_tst].[session\_id]

LEFT OUTER JOIN

    sys.dm\_exec\_requests [s\_er]

ON

    [s\_er].[session\_id] = [s\_tst].[session\_id]

CROSS APPLY

    sys.dm\_exec\_sql\_text ([s\_ec].[most\_recent\_sql\_handle]) AS [s\_est]

OUTER APPLY

    sys.dm\_exec\_query\_plan ([s\_er].[plan\_handle]) AS [s\_eqp]

ORDER BY

    [Begin Time] ASC;

GO



/\*

--DBCC OpenTran

Reference:

<https://docs.microsoft.com/en-us/sql/t-sql/database-console-commands/dbcc-opentran-transact-sql>

<https://www.sqlskills.com/blogs/paul/script-open-transactions-with-text-and-plans/>

DBCC OPENTRAN can only display

the oldest active transaction information.

If you want to see all the active transaction information,

You have to use the following script.

--SELECT

--    [s\_tst].[session\_id],

--    [s\_es].[login\_name] AS [Login Name],

--    DB\_NAME (s\_tdt.database\_id) AS [Database],

--    [s\_tdt].[database\_transaction\_begin\_time] AS [Begin Time],

--    [s\_tdt].[database\_transaction\_log\_bytes\_used] AS [Log Bytes],

--    [s\_tdt].[database\_transaction\_log\_bytes\_reserved] AS [Log Rsvd],

--    [s\_est].text AS [Last T-SQL Text],

--    [s\_eqp].[query\_plan] AS [Last Plan]

--FROM

--    sys.dm\_tran\_database\_transactions [s\_tdt]

--JOIN

--    sys.dm\_tran\_session\_transactions [s\_tst]

--ON

--    [s\_tst].[transaction\_id] = [s\_tdt].[transaction\_id]

--JOIN

--    sys.[dm\_exec\_sessions] [s\_es]

--ON

--    [s\_es].[session\_id] = [s\_tst].[session\_id]

--JOIN

--    sys.dm\_exec\_connections [s\_ec]

--ON

--    [s\_ec].[session\_id] = [s\_tst].[session\_id]

--LEFT OUTER JOIN

--    sys.dm\_exec\_requests [s\_er]

--ON

--    [s\_er].[session\_id] = [s\_tst].[session\_id]

--CROSS APPLY

--    sys.dm\_exec\_sql\_text ([s\_ec].[most\_recent\_sql\_handle]) AS [s\_est]

--OUTER APPLY

--    sys.dm\_exec\_query\_plan ([s\_er].[plan\_handle]) AS [s\_eqp]

--ORDER BY

--    [Begin Time] ASC;

--GO

\*/

4. Clean up

--==============================================================

--T026\_01\_04

--Clean up

--If Table exists then DROP it

IF ( EXISTS ( SELECT    \*

              FROM      INFORMATION\_SCHEMA.TABLES

              WHERE     TABLE\_NAME = 'TableA' ) )

    BEGIN

        TRUNCATE TABLE dbo.TableA;

        DROP TABLE TableA;

    END;

GO -- Run the previous command and begins new batch

--clean up

--If Table exists then DROP it

IF ( EXISTS ( SELECT    \*

              FROM      INFORMATION\_SCHEMA.TABLES

              WHERE     TABLE\_NAME = 'TableB' ) )

    BEGIN

        TRUNCATE TABLE dbo.TableB;

        DROP TABLE TableB;

    END;

GO -- Run the previous command and begins new batch