In this video we will discuss **how to write the deadlock information to the SQL Server error log**  
  
   
  
**When deadlocks occur**, SQL Server chooses one of the transactions as the deadlock victim and rolls it back. There are several ways in SQL Server to track down the queries that are causing deadlocks. One of the options is to use SQL Server trace flag 1222 to write the deadlock information to the SQL Server error log.   
  
**Enable Trace flag :** To enable trace flags use DBCC command. -1 parameter indicates that the trace flag must be set at the global level. If you omit -1 parameter the trace flag will be set only at the session level. 

DBCC Traceon(1222, -1)   
  
To check the status of the trace flag  
DBCC TraceStatus(1222, -1)   
  
To turn off the trace flag  
DBCC Traceoff(1222, -1)  
  
The following SQL code generates a dead lock. This is the same code we discussed in [Part 78](http://csharp-video-tutorials.blogspot.com/2015/08/sql-server-deadlock-example.html) of [SQL Server Tutorial](https://www.youtube.com/playlist?list=PL08903FB7ACA1C2FB). 

**--SQL script to create the tables and populate them with test data**

Create table TableA

(

    Id int identity primary key,

    Name nvarchar(50)

)

Go

Insert into TableA values ('Mark')

Go

Create table TableB

(

    Id int identity primary key,

    Name nvarchar(50)

)

Go

Insert into TableB values ('Mary')

Go

**--SQL Script to create stored procedures**

Create procedure spTransaction1

as

Begin

    Begin Tran

    Update TableA Set Name = 'Mark Transaction 1' where Id = 1

    Waitfor delay '00:00:05'

    Update TableB Set Name = 'Mary Transaction 1' where Id = 1

    Commit Transaction

End

Create procedure spTransaction2

as

Begin

    Begin Tran

    Update TableB Set Name = 'Mark Transaction 2' where Id = 1

    Waitfor delay '00:00:05'

    Update TableA Set Name = 'Mary Transaction 2' where Id = 1

    Commit Transaction

End

Open 2 instances of SQL Server Management studio. From the first window execute **spTransaction1** and from the second window execute **spTransaction2**.    
  
After a few seconds notice that one of the transactions complete successfully while the other transaction is made the deadlock victim and rollback.   
  
The information about this deadlock should now have been logged in sql server error log.   
  
**To read the error log**  
execute sp\_readerrorlog   
  
**Next video :**How to read and understand the deadlock information that is logged in the sql server error log