https://www.dotnettricks.com/learn/sqlserver/sql-server-exception-handling-by-try-catch#:~:text=To%20handle%20exception%20in%20Sql,to%20the%20corresponding%20CATCH%20blocks.

Exception Handling in SQL Server by TRY…CATCH

[[](https://www.dotnettricks.com/mentors/shailendra-chauhan)   Shailendra Chauhan](https://www.dotnettricks.com/mentors/shailendra-chauhan)

[**Print**](javascript:void(0);) 3 min read

 31 Aug 2022

 Intermediate

 231K Views

Like C#, SQL Server also has an exception model to handle exceptions and errors that occurs in T-SQL statements. To handle exception in Sql Server we have TRY..CATCH blocks. We put T-SQL statements in TRY block and to handle exception we write code in CATCH block. If there is an error in code within TRY block then the control will automatically jump to the corresponding CATCH blocks. In Sql Server, against a Try block, we can have only one CATCH block.

TRY..CATCH Syntax

BEGIN TRY

--T-SQL statements

--or T-SQL statement blocks

END TRY

BEGIN CATCH

--T-SQL statements

--or T-SQL statement blocks

END CATCH

Error Functions used within CATCH block

1. ERROR\_NUMBER()

This returns the error number and its value is the same as for @@ERROR function.

1. ERROR\_LINE()

This returns the line number of T-SQL statement that caused an error.

1. ERROR\_SEVERITY()

This returns the severity level of the error.

1. ERROR\_STATE()

This returns the state number of the error.

[](https://www.dotnettricks.com/training/masters-program/dotnet-online-training?utm_source=scholarhat&utm_medium=banner)

1. ERROR\_PROCEDURE()

This returns the name of the stored procedure or trigger where the error occurred.

1. ERROR\_MESSAGE()

This returns the full text of error message. The text includes the values supplied for any substitutable parameters, such as lengths, object names, or times.

Exception handling example

BEGIN TRY

DECLARE @num INT, @msg varchar(200)

---- Divide by zero to generate Error

SET @num = 5/0

PRINT 'This will not execute'

END TRY

BEGIN CATCH

PRINT 'Error occured that is'

set @msg=(SELECT ERROR\_MESSAGE())

print @msg;

END CATCH

GO



BEGIN TRY

DECLARE @num INT

---- Divide by zero to generate Error

SET @num = 5/0

PRINT 'This will not execute'

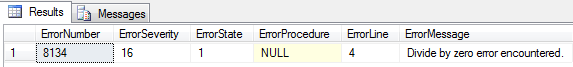
END TRY

BEGIN CATCH

SELECT ERROR\_NUMBER() AS ErrorNumber, ERROR\_SEVERITY() AS ErrorSeverity, ERROR\_STATE() AS ErrorState, ERROR\_PROCEDURE() AS ErrorProcedure, ERROR\_LINE() AS ErrorLine, ERROR\_MESSAGE() AS ErrorMessage;

END CATCH;

GO



Note

1. A TRY..CATCH block combination catches all the errors that have a severity between 11 and 19.
2. The CATCH block is executed only if there is an error occurs in T-SQL statements within TRY block otherwise the CATCH block is ignored.
3. Each TRY block is associated with only one CATCH block and vice versa
4. TRY and CATCH blocks can’t be separated with the GO statement. We need to put both TRY and CATCH blocks within the same batch.
5. TRY..CATCH blocks can be used with transactions. We check the number of open transactions by using @@TRANCOUNT function in Sql Server.
6. XACT\_STATE function within the TRY..CATCH block can be used to check whether an open transaction is committed or not. It will return -1 if the transaction is not committed else returns 1.

[Read More Articles Related to SQL Server](https://www.dotnettricks.com/learn/sqlserver)

Summary

In this article, I try to explain Exception handling in Sql Server with example. I hope after reading this article you will be know how to handle exception in Sql Server. I would like to have feedback from my blog readers. Please post your feedback, question, or comments about this article.