In this video we will discuss **DDL Triggers in sql server**.   
  
   
  
**In SQL Server there are 4 types of triggers**   
**1.** DML Triggers - Data Manipulation Language. Discussed in Parts 43 to 47 of [SQL Server](https://www.youtube.com/playlist?list=PL08903FB7ACA1C2FB) Tutorial.  
**2.** DDL Triggers - Data Definition Language  
**3.** CLR triggers - Common Language Runtime  
**4.** Logon triggers  
  
**What are DDL triggers**  
**DDL triggers fire in response to DDL events** - CREATE, ALTER, and DROP (Table, Function, Index, Stored Procedure etc...). For the list of all DDL events please visit https://msdn.microsoft.com/en-us/library/bb522542.aspx   
  
**Certain system stored procedures** that perform DDL-like operations can also fire DDL triggers. Example - sp\_rename system stored procedure   
  
**What is the use of DDL triggers**

* If you want to execute some code in response to a specific DDL event
* To prevent certain changes to your database schema
* Audit the changes that the users are making to the database structure

**Syntax for creating DDL trigger**

CREATE TRIGGER [Trigger\_Name]

ON [Scope (Server|Database)]

FOR [EventType1, EventType2, EventType3, ...],

AS

BEGIN

   -- Trigger Body

END

**DDL triggers scope :** DDL triggers can be created in a specific database or at the server level.    
  
**The following trigger will fire in response to CREATE\_TABLE DDL event.**

CREATE TRIGGER trMyFirstTrigger

ON Database

FOR CREATE\_TABLE

AS

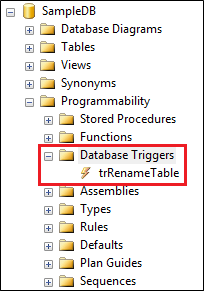
BEGIN

   Print 'New table created'

END

**To check if the trigger has been created**

1. In the Object Explorer window, expand the **SampleDB**database by clicking on the plus symbol.
2. Expand **Programmability**folder
3. Expand **Database Triggers** folder

   
  
**Please note :** If you can't find the trigger that you just created, make sure to refresh the Database Triggers folder.  
  
When you execute the following code to create the table, the trigger will automatically fire and will print the message - New table created  
Create Table Test (Id int)  
  
The above trigger will be fired only for one DDL event CREATE\_TABLE. If you want this trigger to be fired for multiple events, for example when you alter or drop a table, then separate the events using a comma as shown below.

ALTER TRIGGER trMyFirstTrigger

ON Database

FOR CREATE\_TABLE, ALTER\_TABLE, DROP\_TABLE

AS

BEGIN

   Print 'A table has just been created, modified or deleted'

END

Now if you create, alter or drop a table, the trigger will fire automatically and you will get the message - A table has just been created, modified or deleted.    
  
The 2 DDL triggers above execute some code in response to DDL events

Now let us look at an example of how to prevent users from creating, altering or dropping tables. To do this modify the trigger as shown below. 

ALTER TRIGGER trMyFirstTrigger

ON Database

FOR CREATE\_TABLE, ALTER\_TABLE, DROP\_TABLE

AS

BEGIN

   Rollback

   Print 'You cannot create, alter or drop a table'

END

To be able to create, alter or drop a table, you either have to disable or delete the trigger.   
  
**To disable trigger**  
**1.** Right click on the trigger in object explorer and select **"Disable"**from the context menu   
**2.** You can also disable the trigger using the following T-SQL command

DISABLE TRIGGER trMyFirstTrigger ON DATABASE

**To enable trigger**  
**1.** Right click on the trigger in object explorer and select "Enable" from the context menu   
**2.** You can also enable the trigger using the following T-SQL command  
ENABLE TRIGGER trMyFirstTrigger ON DATABASE  
  
**To drop trigger**  
**1.** Right click on the trigger in object explorer and select "Delete" from the context menu   
**2.** You can also drop the trigger using the following T-SQL command  
DROP TRIGGER trMyFirstTrigger ON DATABASE  
  
Certain system stored procedures that perform DDL-like operations can also fire DDL triggers. The following trigger will be fired when ever you rename a database object using sp\_rename system stored procedure.

CREATE TRIGGER trRenameTable

ON DATABASE

FOR RENAME

AS

BEGIN

    Print 'You just renamed something'

END

The following code changes the name of the TestTable to NewTestTable. When this code is executed, it will fire the trigger trRenameTable

sp\_rename 'TestTable', 'NewTestTable'

The following code changes the name of the Id column in NewTestTable to NewId. When this code is executed, it will fire the trigger trRenameTable

sp\_rename 'NewTestTable.Id' , 'NewId', 'column'