

Programming SQL Server Database Triggers and Functions

PROTECTING THE DATABASE WITH DDL AND LOGON TRIGGERS



Ryan Booz

AUTHOR AND SPEAKER

@ryanbooz <https://www.softwareandbooz.com>



DDL & Login Triggers



Understand DDL Triggers and how they differ from DML Triggers

Using Login Triggers to manage security events

Common Use Cases



What is a DDL Trigger?



DDL

Data Definition Language is a vocabulary of standard T-SQL commands that modify or manipulate schema.

CREATE, ADD, DROP, ALTER, etc.



A Tool for the Database Administrator



DDL Triggers watch for schema modification events at the Database or Server level

React to those modification events

Not attached to a specific table or schema

Forces extra work when the modification is truly necessary



DDL Triggers provide an easy and accessible way to prevent mistakes and log schema events



Anatomy of a DDL Trigger



Anatomy of a DDL Trigger

```
CREATE OR ALTER TRIGGER [Trigger Name]
ON {ALL SERVER | DATABASE }
FOR { event_type | event_group }
AS

    /*

        Insert business logic here...

    */

GO;
```



DDL Triggers execute after
the event has completed.



DDL Triggers can be defined for events at the database or server level



```
CREATE OR ALTER TRIGGER NoDrop  
ON DATABASE  
FOR DROP_TRIGGER...
```

◀ React to the “DROP TRIGGER” event for the specified database only

```
CREATE OR ALTER TRIGGER NoDrop  
ON ALL SERVER  
FOR DROP_TRIGGER...
```

◀ React to the “DROP TRIGGER” event for all databases on the server



DDL Triggers provide information about the event through a special function called 'EVENTDATA()'



DDL EVENTDATA() Function

```
<EVENT_INSTANCE>
  <EventType>DROP_TRIGGER</EventType>
  <PostTime>2019-04-29T22:10:54.530</PostTime>
  <SPID>76</SPID>
  <ServerName>RYANB-DEV</ServerName>
  <LoginName>RYANB-DEV\Ryan</LoginName>
  <UserName>dbo</UserName>
<DatabaseName>WideWorldImporters</DatabaseName>
  <SchemaName>Sales</SchemaName>
  <ObjectName>TU_Orders_AFTER_One</ObjectName>
  <ObjectType>TRIGGER</ObjectType>
  <TargetObjectName>Orders</TargetObjectName>
  <TargetObjectType>TABLE</TargetObjectType>
  <TSQLCommand>
    <SetOptions ANSI_NULLS="ON" />
    <CommandText>
      drop trigger Sales.TU_Orders_AFTER_One
    </CommandText>
  </TSQLCommand>
</EVENT_INSTANCE>
```

Provides detailed data about the DDL event

Useful for logging DDL audit events

Displays the actual CommandText that triggered the event



DDL Trigger Events and Event Groups



275+ DDL Trigger Events



Database DDL Events

Table	CREATE_TABLE	ALTER_TABLE	DROP_TABLE
Index	CREATE_INDEX	ALTER_INDEX	DROP_INDEX
Function	CREATE_FUNCTION	ALTER_FUNCTION	DROP_FUNCTION
Procedure	CREATE_PROCEDURE	ALTER_PROCEDURE	DROP_PROCEDURE
Queue	CREATE_QUEUE	ALTER_QUEUE	DROP_QUEUE
Trigger	CREATE_TRIGGER	ALTER_TRIGGER	DROP_TRIGGER
Statistics	CREATE_STATISTICS	UPDATE_STATISTICS	DROP_STATISTICS
...			



DDL Event Groups

DDL_TABLE_EVENTS	DDL_INDEX_EVENTS	DDL_SSB_EVENTS
CREATE_TABLE	CREATE_INDEX	DDL_BROKER_PRIORITY_EVENTS
ALTER_TABLE	ALTER_INDEX	DDL_CONTRACT_EVENTS
DROP_TABLE	DROP_INDEX	DDL_MESSAGE_TYPE_EVENTS
	CREATE_FULLTEXT_INDEX	DDL_QUEUE_EVENTS
	DROP_FULLTEXT_INDEX	DDL_ROUTE_EVENTS

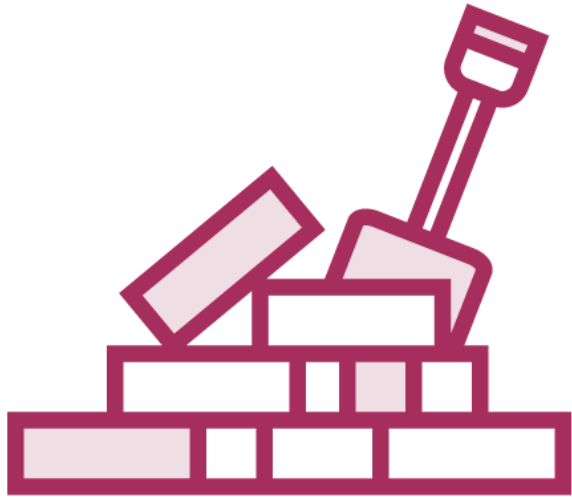
DDL_DATABASE_EVENTS	DDL_TRIGGER_EVENTS	DDL_LOGIN_EVENTS
CREATE_DATABASE	CREATE_TRIGGER	CREATE_LOGIN
ALTER_DATABASE	ALTER_TRIGGER	ALTER_LOGIN
DROP_DATABASE	DROP_TRIGGER	DROP_LOGIN



Common DDL Trigger Use Cases



Common Use Cases – DDL Triggers



**Prevent modifications of
schema objects**



**Log information about
DDL Events**

Demo



Preventing ALTER_TABLE and
DROP_TABLE events

Logging DDL events for auditing



What is a Logon Trigger?



Logon Trigger

A Trigger that responds to the LOGON event and executes after a user has successfully authenticated with the server.





LOGON Triggers can render
your server inaccessible
and unusable.

Proceed with CAUTION!




```
EXEC sp_configure 'remote  
admin connections', 1;  
GO
```

```
RECONFIGURE  
GO
```

Enable the DAC!!

- ◀ 'Dedicated Administrator Connection'
- ◀ This is your ONE 'get out of jail free' card
- ◀ All servers should have this enabled and you should test using it!
- ◀ Find more at <http://docs.microsoft.com>





There Be Dragons!



Anatomy of a LOGON Trigger



Anatomy of a LOGON Trigger

```
CREATE OR ALTER TRIGGER [Trigger Name]
ON ALL SERVER {EXECUTE AS} [UserName]
FOR LOGON
AS
```

```
/*
```

```
    Insert business logic here...
```

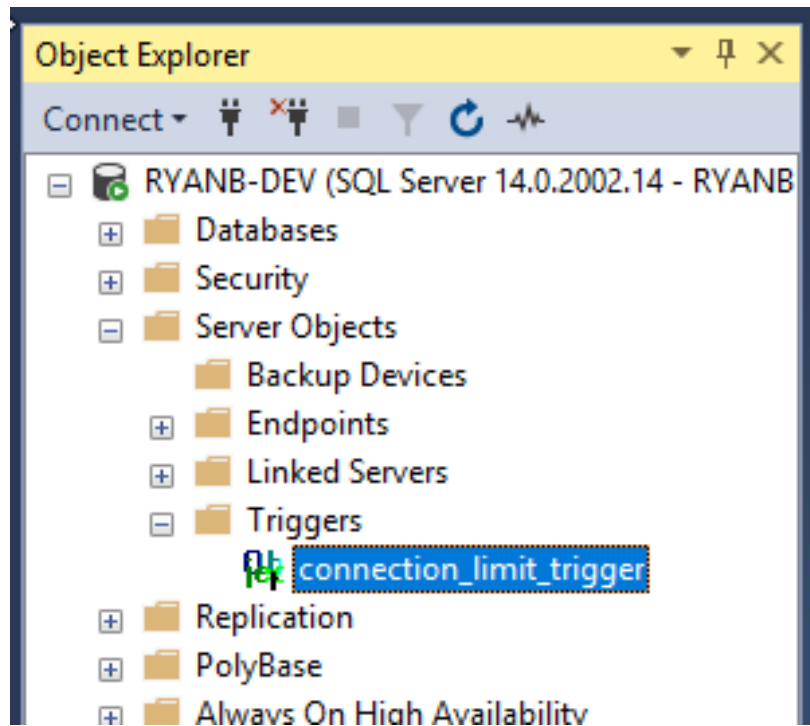
```
*/
```

```
GO;
```



LOGON Triggers are always
created in the
'master' database





```
select * from sys.server_triggers
```



Multiple LOGON Triggers
are allowed on the server



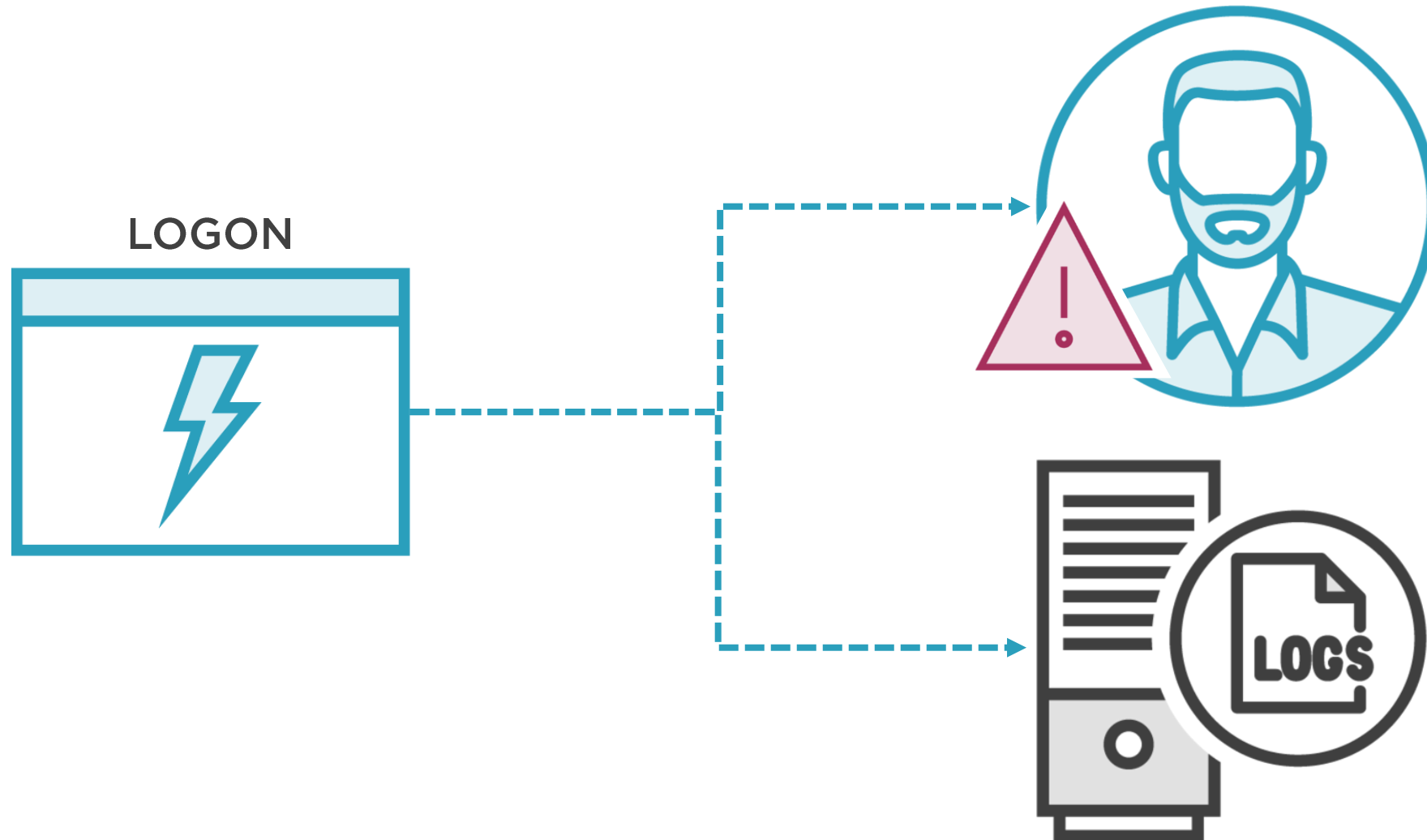
LOGON Trigger Execution Order



All output from the LOGON
Trigger is diverted to the
SQL Server error log



LOGON Trigger Output



LOGON Triggers provide
information about the event
with EVENTDATA()



LOGON EVENTDATA() Function

```
<EVENT_INSTANCE>  
  <EventType>LOGON</EventType>  
    <PostTime>2019-04-29T22:10:54.530</PostTime>  
    <SPID>76</SPID>  
    <ServerName>RYANB-DEV</ServerName>  
    <LoginName>RYANB-DEV\Ryan</LoginName>  
    <LoginType>Windows (NT) Login</LoginType>  
    <SID>0x0123456789ABCDEF0123456789ABC...</SID>  
    <ClientHost></ClientHost>  
    <IsPooled>0</IsPooled>  
</EVENT_INSTANCE>
```

Provides detailed data about the LOGON event

Useful for logging LOGON audit events



Make your LOGON
Triggers FAST...

...every connection
uses them



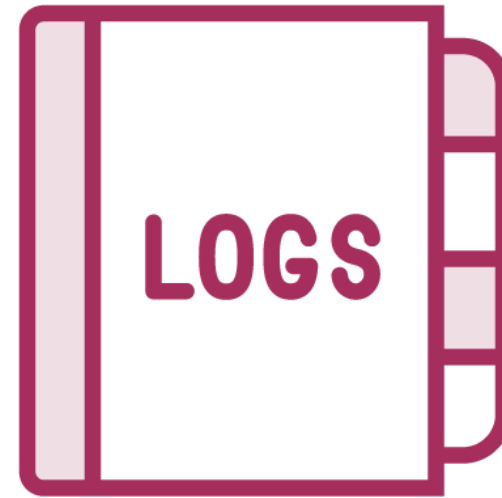
Common Use Cases for LOGON Triggers



Common Use Cases – LOGON Triggers



**Prevent user connection
after successful
authentication**



**Log information about
LOGON Events**



Demo



Prevent database users from connecting multiple times



Demo



Create a LOGON Trigger to provide customized logging





Summary



DDL and LOGON Triggers

EVENTDATA() Function

Common Use Cases for each

Use caution with LOGON Triggers



