

AFTER triggers (DML)

BUILDING AND OPTIMIZING TRIGGERS IN SQL SERVER



Florin Angelescu
Instructor

Definition and properties

- Performs a set of actions when fired
- The actions are performed only after the DML event is finished
- Used with `INSERT` , `UPDATE` , and `DELETE` statements for tables or views

AFTER trigger prerequisites

- Table or view needed for DML statements
- The trigger will be attached to the same table

Output from table `Products` used for this example:

```
| Product | Price | Currency | Quantity | Measure |
|-----|-----|-----|-----|-----|
| Apple   | 2.00  | USD      | 25000    | kg       |
| Apricot | 1.55  | USD      | 2500     | kg       |
| Avocado | 3.00  | USD      | 1000     | kg       |
| Banana  | 1.80  | USD      | 35000    | kg       |
```

AFTER trigger prerequisites

- Target table
 - Description of the trigger
 - Trigger firing event (DML)
 - Trigger name
- Products
 - Save info of deleted products to a history table
 - DELETE
 - TrackRetiredProducts

AFTER trigger prerequisites summary

When rows are removed from the `Products` table...

Save the required information from those rows to the table `RetiredProducts`.

Product	Price	Currency	Quantity	Measure	
Apple	2.00	USD	25000	kg	
Apricot	1.55	USD	2500	kg	X
Avocado	3.00	USD	1000	kg	
Banana	1.80	USD	35000	kg	

Product	Measure	RemovalDate
Apricot	kg	19.04.2019

AFTER trigger definition

```
CREATE TRIGGER TrackRetiredProducts  
  
ON Products  
  
AFTER DELETE  
  
AS  
  
    INSERT INTO RetiredProducts (Product, Measure)  
    SELECT Product, Measure  
    FROM deleted;
```

"inserted" and "deleted" tables

- Special tables used by DML triggers
- Created automatically by SQL Server

"inserted" and "deleted" tables

- Special tables used by DML triggers
- Created automatically by SQL Server

Special table	INSERT	UPDATE	DELETE
inserted	new rows	new rows	N/A
deleted	N/A	updated rows	removed rows

The complete AFTER trigger

```
CREATE TRIGGER TrackRetiredProducts
ON Products
AFTER DELETE
AS
    INSERT INTO RetiredProducts (Product, Measure)
    SELECT Product, Measure
    FROM deleted;
```

Let's practice!

BUILDING AND OPTIMIZING TRIGGERS IN SQL SERVER

INSTEAD OF triggers (DML)

BUILDING AND OPTIMIZING TRIGGERS IN SQL SERVER



Florin Angelescu
Instructor

Definition and properties

- Performs a set of actions when fired
- The actions are performed instead of the DML event
- The DML event does not run anymore
- Used with `INSERT` , `UPDATE` , and `DELETE` statements for tables or views

INSTEAD OF trigger prerequisites

ID	Customer	Product	Price	Quantity	OrderDate	TotalAmount
284	VitaFruit	Orange	0.76	75	2018-09-17	57.00
330	Healthy Choices	Raspberry	1.96	200	2018-10-29	392.00
302	World of Fruits	Pineapple	4.16	25	2018-10-09	104.00
248	Fruit Mania	Plum	1.11	300	2018-08-11	333.00
358	Fruit Mag	Pomelo	1.11	750	2018-11-29	832.50
100	VitaFruit	Avocado	2.91	500	2018-03-29	1455.00

INSTEAD OF trigger prerequisites

- Target table
 - Description of the trigger
 - Trigger firing event (DML)
 - Trigger name
- Orders
 - Prevent updates of existing entries
 - UPDATE
 - PreventOrdersUpdate

INSTEAD OF trigger definition

```
CREATE TRIGGER PreventOrdersUpdate  
  
ON Orders  
  
INSTEAD OF UPDATE  
  
AS  
  
    RAISERROR ('Updates on "Orders" table are not permitted.  
              Place a new order to add new products', 16, 1);
```

Let's practice!

BUILDING AND OPTIMIZING TRIGGERS IN SQL SERVER

DDL triggers

BUILDING AND OPTIMIZING TRIGGERS IN SQL SERVER



Florin Angelescu
Instructor

Definition and properties

DML triggers

- events associated with DML statements
`INSERT`, `UPDATE`, `DELETE`
- used with `AFTER` or `INSTEAD OF`
- attached to tables or views
- `inserted` and `deleted` special tables

DDL triggers

- events associated with DDL statements
`CREATE`, `ALTER`, `DROP`
- only used with `AFTER`
- attached to databases or servers
- no special tables

AFTER and FOR

```
CREATE TRIGGER DatabaseChangeLog  
FOR CREATE_TABLE  
[...]
```

AFTER and FOR

```
CREATE TRIGGER TrackDatabaseTables  
FOR CREATE_TABLE  
[...]
```

AFTER = FOR

DDL trigger prerequisites

- Target object (server or database)
 - Description of the trigger
 - Trigger firing events (DDL)
 - Trigger name
- DATABASE
 - Log table with definition changes
 - CREATE_TABLE, ALTER_TABLE, DROP_TABLE
 - TrackTableChanges

DDL trigger definition

```
CREATE TRIGGER TrackTableChanges  
  
ON DATABASE  
  
FOR CREATE_TABLE,  
    ALTER_TABLE,  
    DROP_TABLE  
  
AS  
  
    INSERT INTO TablesChangeLog (EventData, ChangedBy)  
    VALUES (EVENTDATA(), USER);
```

Preventing the triggering events for DML triggers

```
CREATE TRIGGER PreventTableDeletion
ON DATABASE
FOR DROP_TABLE
AS
    RAISERROR ('You are not allowed to remove tables from this database.', 16, 1);
ROLLBACK;
```

Let's practice!

BUILDING AND OPTIMIZING TRIGGERS IN SQL SERVER

Logon triggers

BUILDING AND OPTIMIZING TRIGGERS IN SQL SERVER



Florin Angelescu
Instructor

Definition and properties

- Performs a set of actions when fired
- The actions are performed for `LOGON` events
- After authentication phase, but before the session establishment

Logon trigger prerequisites

- Trigger firing event
 - Description of the trigger
 - Trigger name
- LOGON
 - Audit successful / failed logons to the server
 - LogonAudit

Logon trigger definition

```
CREATE TRIGGER LogonAudit
ON ALL SERVER WITH EXECUTE AS 'sa'
FOR LOGON
AS
    INSERT INTO ServerLogonLog
        (LoginName, LoginDate, SessionID, SourceIPAddress)
    SELECT ORIGINAL_LOGIN(), GETDATE(), @@SPID, client_net_address
    FROM SYS.DM_EXEC_CONNECTIONS WHERE session_id = @@SPID;
```

Logon trigger definition summary

```
CREATE TRIGGER LogonAudit
ON ALL SERVER WITH EXECUTE AS 'sa'
FOR LOGON
AS
    INSERT INTO ServerLogonLog
        (LoginName, LoginDate, SessionID, SourceIPAddress)
    SELECT ORIGINAL_LOGIN(), GETDATE(), @@SPID, client_net_address
    FROM SYS.DM_EXEC_CONNECTIONS WHERE session_id = @@SPID;
```

Let's practice!

BUILDING AND OPTIMIZING TRIGGERS IN SQL SERVER