

Version

SQL Server 2019 ▾

🔍 Filter by title

SQL Server

Docs navigation tips

Previous versions 2005-2014

> Overview

> Big Data Clusters

> Business continuity

▾ Database design

Hierarchical Data

> Collation

> Databases

> Filestream, Filetable & BLOB

> Indexes

> SQL Graph

> Sequence numbers

> Service broker

> Spatial data

> Stored procedures

> Tables

Capture Logon Trigger Event Data

Manage Trigger Security

> DDL Triggers

▾ DML Triggers

DML Triggers

Create DML Triggers

Create CLR Triggers

Create DML Triggers to Handle

Multiple Rows of Data

Create Nested Triggers

Specify First & Last Triggers

Use the inserted & deleted Tables

Modify or Rename DML Triggers

Get Information About DML

Triggers

Delete or Disable DML Triggers

> User-defined functions

> Views

> XML data

> Development

> Internals & Architecture

> Installation

> Migrate & load data

> Manage, monitor, & tune

> Query data

> Reporting & Analytics

> Security

> Tools

> Tutorials

> SQL Server on Linux





> SQL on Azure

> Resources

> Reference

DML Triggers

03/14/2017 • 5 minutes to read • 👤 👤 👤

APPLIES TO:  SQL Server  Azure SQL Database  Azure Synapse Analytics (SQL DW)  Parallel Data Warehouse

DML triggers is a special type of stored procedure that automatically takes effect when a data manipulation language (DML) event takes place that affects the table or view defined in the trigger. DML events include INSERT, UPDATE, or DELETE statements. DML triggers can be used to enforce business rules and data integrity, query other tables, and include complex Transact-SQL statements. The trigger and the statement that fires it are treated as a single transaction, which can be rolled back from within the trigger. If a severe error is detected (for example, insufficient disk space), the entire transaction automatically rolls back.

DML Trigger Benefits

DML triggers are similar to constraints in that they can enforce entity integrity or domain integrity. In general, entity integrity should always be enforced at the lowest level by indexes that are part of PRIMARY KEY and UNIQUE constraints or are created independently of constraints. Domain integrity should be enforced through CHECK constraints, and referential integrity (RI) should be enforced through FOREIGN KEY constraints. DML triggers are most useful when the features supported by constraints cannot meet the functional needs of the application.

The following list compares DML triggers with constraints and identifies when DML triggers have benefits over .

- DML triggers can cascade changes through related tables in the database; however, these changes can be executed more efficiently using cascading referential integrity constraints. FOREIGN KEY constraints can validate a column value only with an exact match to a value in another column, unless the REFERENCES clause defines a cascading referential action.

- They can guard against malicious or incorrect INSERT, UPDATE, and DELETE operations and enforce other restrictions that are more complex than those defined with CHECK constraints.

Unlike CHECK constraints, DML triggers can reference columns in other tables. For example, a trigger can use a SELECT from another table to compare to the inserted or updated data and to perform additional actions, such as modify the data or display a user-defined error message.

- They can evaluate the state of a table before and after a data modification and take actions based on that difference.
- Multiple DML triggers of the same type (INSERT, UPDATE, or DELETE) on a table allow multiple, different actions to take place in response to the same modification statement.
- Constraints can communicate about errors only through standardized system error messages. If your application requires, or can benefit from, customized messages and more complex error handling, you must use a trigger.
- DML triggers can disallow or roll back changes that violate referential integrity, thereby canceling the attempted data modification. Such a trigger might go into effect when you change a foreign key and the new value does not match its primary key. However, FOREIGN KEY constraints are usually used for this purpose.
- If constraints exist on the trigger table, they are checked after the INSTEAD OF trigger execution but prior to the AFTER trigger execution. If the constraints are violated, the INSTEAD OF trigger actions are rolled back and the AFTER trigger is not executed.

Types of DML Triggers

AFTER trigger

AFTER triggers are executed after the action of the INSERT, UPDATE, MERGE, or DELETE statement is performed. AFTER triggers are never executed if a constraint violation occurs; therefore, these triggers cannot be used for any processing that might prevent constraint violations. For every INSERT, UPDATE, or DELETE action specified in a MERGE statement, the corresponding trigger is fired for each DML operation.

INSTEAD OF trigger

INSTEAD OF triggers override the standard actions of the triggering statement. Therefore, they can be used to perform error or value checking on one or more columns and then perform additional actions before insert, updating or deleting the row or rows. For example, when the value being updated in an hourly wage column in a payroll table exceeds a specified value, a trigger can be defined to either produce an error message and roll back the transaction, or insert a new record into an audit trail before inserting the record into the payroll table. The primary advantage of INSTEAD OF triggers is that they enable views that would not be updatable to support updates. For example, a view based on multiple base tables must use an INSTEAD OF trigger to support inserts, updates, and deletes that reference data in more than one table. Another advantage of INSTEAD OF triggers is that they enable you to code logic that can reject parts of a

Is this page helpful?

 Yes  No

In this article

[DML Trigger Benefits](#)[Types of DML Triggers](#)[Related Tasks](#)[See Also](#)

batch while letting other parts of a batch to succeed.

This table compares the functionality of the AFTER and INSTEAD OF triggers.

Function	AFTER trigger	INSTEAD OF trigger
Applicability	Tables	Tables and views
Quantity per table or view	Multiple per triggering action (UPDATE, DELETE, and INSERT)	One per triggering action (UPDATE, DELETE, and INSERT)
Cascading references	No restrictions apply	INSTEAD OF UPDATE and DELETE triggers are not allowed on tables that are targets of cascaded referential integrity constraints.
Execution	After:	Before: Constraint processing
	Constraint processing	In place of: The triggering action
	Declarative referential actions	After: inserted and deleted tables creation
	inserted and deleted tables creation	
	The triggering action	
Order of execution	First and last execution may be specified	Not applicable
varchar(max) , nvarchar(max) , and varbinary(max) column references in inserted and deleted tables	Allowed	Allowed
text , ntext , and image column references in inserted and deleted tables	Not allowed	Allowed

CLR Triggers

A CLR Trigger can be either an AFTER or INSTEAD OF trigger. A CLR trigger can also be a DDL trigger. Instead of executing a Transact-SQL stored procedure, a CLR trigger executes one or more methods written in managed code that are members of an assembly created in the .NET Framework and uploaded in SQL Server.

Related Tasks

Task	Topic
Describes how to create a DML trigger.	Create DML Triggers
Describes how to create a CLR trigger.	Create CLR Triggers
Describes how to create a DML trigger to handle both single-row and multi-row data modifications.	Create DML Triggers to Handle Multiple Rows of Data
Describes how to nest triggers.	Create Nested Triggers
Describes how to specify the order in which AFTER triggers are fired.	Specify First and Last Triggers
Describes how to use the special inserted and delete tables in trigger code.	Use the inserted and deleted Tables
Describes how to modify or rename a DML trigger.	Modify or Rename DML Triggers
Describes how to view information about DML triggers.	Get Information About DML Triggers
Describes how to delete or disable DML triggers.	Delete or Disable DML Triggers
Describes how to manage trigger security.	Manage Trigger Security

See Also

- [CREATE TRIGGER \(Transact-SQL\)](#)
- [ALTER TRIGGER \(Transact-SQL\)](#)
- [DROP TRIGGER \(Transact-SQL\)](#)
- [DISABLE TRIGGER \(Transact-SQL\)](#)
- [Trigger Functions \(Transact-SQL\)](#)

Feedback

Send feedback about

This product 

 This page 

There is currently no feedback for this document. Page feedback will appear here.

↓ Download PDF