

# Database Systems

## Session 2019 (spring)

### Semester 6<sup>th</sup>

---

**Instructor:** Samyan Qayyum Wahla, Sahar Waqar

**Graduate Assistants:** Muhammad Nasir Nawaz, Junaid Zafar

**Agenda:** Triggers in SQL

**Trigger:** A *trigger* is a special kind of stored procedure—one that cannot be executed explicitly. Instead, it's attached to an event. Whenever the event takes place, the trigger fires and the trigger's code runs.

**Types of Triggers:** There are two main types of SQL triggers 1) DML Triggers, 2) DDL Triggers.

#### 1) DML TRIGGERS:

These triggers are associated with DML (data manipulation language) events. These events include INSERT, UPDATE and DELETE. DML triggers fires when any of these events takes place.

There are two further sub-types of DML triggers as follows:

1. AFTER TRIGGERS (also called as FOR TRIGGERS)
2. INSTEAD OF TRIGGERS

**AFTER TRIGGERS:** These triggers fires and executes their code after a DML event (INSERT, UPDATE or DELETE) takes place. Syntax to create an after trigger (on INSERT event) is as follows:

```
CREATE TRIGGER trgTriggerName ON  
tableName AFTER INSERT  
AS  
BEGIN  
    SQL statement here  
END
```

**INSTEAD OF TRIGGERS:** These triggers fires and executes their code before a DML event (INSERT, UPDATE or DELETE) takes place. These triggers prevent the execution of actual SQL statement. Syntax to create an instead of trigger (on UPDATE event) is as follows:

```
CREATE TRIGGER trgTriggerName ON  
tableName INSTEAD OF UPDATE  
AS  
BEGIN  
    SQL statement here  
END
```

A DML trigger can be altered and dropped by using ALTER & DROP command.

## 2) DDL TRIGGERS:

These triggers are associated with DDL (data definition language) events. These events include CREATE, ALTER and DROP etc. DDL triggers fires when any of these events takes place. DDL TRIGGERS could be created at two scopes **1)** at database level scope for example CREATE TABLE, **2)** at server level scope for example CREATE DATABASE. *SQL server currently supports AFTER DDL Triggers, it does not support INSTEAD OF DDL Triggers.*

We'll discuss DDL triggers in two different parts.

1. DATABASE SCOPE LEVEL
2. SERVER SCOPE LEVEL

**DATABASE SCOPE LEVEL:** Scope of these triggers is only up to the database on which the trigger is created. It will work with CREATE, ALTER and DROP event when used with a table, stored procedure or view etc. Syntax is given as follows:

```
CREATE TRIGGER trgTriggerName ON  
DATABASE FOR EVENT_NAME  
AS  
BEGIN  
    SQL statement here  
END
```

**SERVER SCOPE LEVEL:** Scope of these triggers is up to whole server. All the databases on the servers comes under the scope of these triggers. Trigger will be executed if event CREATE, ALTER or DROP is invoked on any of the databases on the server. Syntax is given below:

```
CREATE TRIGGER trgTriggerName ON
ALL SERVER FOR EVENT_NAME
AS
BEGIN
    SQL statement here
END
```

A DDL trigger can be altered and dropped by using ALTER & DROP command.