**DDL TRIGGERS IN SQL SERVER:**

• Data Definition Language (DDL) triggers execute stored procedures when DDL events such as CREATE, ALTER and DROP statements occur in the database or the server.

**What is the use of DDL Triggers:**

• DDL triggers can be used to prevent modifications in the database schema. A schema is a collection of objects such as tables, views, and so forth in a database.

• If we want to execute some code or message in response to a specific DDL events.

• To prevent certain changes to your database schema.

• Audit or check the changes that the users are making to the database structure.

**SYNTAX:**

**CREATE TRIGGER <trigger\_name>**

**ON {ALL SERVER | DATABASE}**

**[WITH ENCRYPTION]**

**{FOR | AFTER} {<EVENT\_TYPE>}**

**AS <SQL\_STATEMENT>**

Where,

**ALL SERVER:** specifies that the DDL trigger executes when DDL events occur in the current server.

**DATABASE:** specifies that the DDL trigger executes when DDL events occur in the current database.

**EVENT\_TYPE:** specifies the name of the DDL event that invokes the DDL trigger.

• DDL triggers scope: DDL triggers can be created in a specific database or at the server level.

• We can use ROLLBACK command to prevent for creating, altering, and dropping something in the database or server.

• We can use **WITH ENCRYPTION** with DDL triggers to hide the code definition.

• We can use DROP and ALTER commands with DDL triggers.

• We can **Enable** or **Disable** triggers for a time as being per requirement.

• **RENAME** event is used in the DDL trigger when we invoke the system stored procedure **sp\_rename.**

**Scope of DDL Triggers:**

• DDL triggers are invoked by SQL statements executed either in the current database or on the current server.

• Scope of DDL trigger depends on whether the trigger executes for database events or server events.

**Database-Scoped DDL Triggers:**

• Are invoked by the events that modify the database schema.

• Stores the triggers in the database and execute on DDL events, except those related to temporary tables.

**Server-Scoped DDL Triggers:**

• Are invoked by DDL events at the server level.

• Are stored in the master database.

• We can also ENABLE and DISABLE Server Scoped DDL Triggers.

• We can also use ALTER and DROP command with Server Scoped DDL Triggers.

• We can also use WITH ENCRYPTION command with Server Scoped DDL triggers.

**Execution Order of DML Triggers:**

• SQL Server 2012 allows users to specify which AFTER trigger is to be executed first and which is to be executed last.

• It is used when we have more than one triggers on a table

• We have one system stored procedure to maintain execution order of DML triggers:

**SP\_SETTRIGGERORDER**

**•** SP\_SETTRIGGERORDER stored procedure have following arguments:

1. **@triggername:** is the name of the DML or DDL trigger and the schema to which it belongs and whose order needs to be specified.
2. **@Order:** specifies the execution order of the trigger as FIRST, LAST and NONE. If FIRST is specified, then the trigger is fired first.
3. **@Stmttype (Statement Type):** specifies the type of SQL statement (INSERT, UPDATE, or DELETE) that invokes the DML trigger.