Database Design & Applications

The Database Language - Triggers

Introduction to Trigger

- A trigger is a block of code that is stored in the database and fired (executed) in response to a specified event.
- The SQLServer database automatically executes the trigger when specified conditions occur.
- A trigger could be defined on Table, View or database.

Business Application Scenarios for Implementing Triggers

- **Security:** Triggers allow table access according to data values.
- **Auditing:** Triggers track values for data operations on tables.
- **Data Integrity:** Triggers helps in implementing complex integrity rules.
- **Derived data:** Triggers helps to compute derived data automatically
- **Event Logging:** Trigger log events transparently.

Triggers

In SQLServer there are three types of triggers:

- **DML Trigger**
  - DML Triggers are fired automatically in response to DML events (Insert, Update, Delete)
- **DDL Trigger**
  - DDL triggers respond to DDL events like CREATE, ALTER, DROP, GRANT, DENY, REVOKE, or UPDATE STATISTICS.
- **Logon Trigger**
  - In SQL Server, the Logon trigger is fired automatically on a LOGON event.
  - A LOGON trigger can be used in controlling server sessions by tracking login activity, restricting logins to the SQL Server, or limiting the number of sessions for a particular login.

DML Triggers

DML triggers can be classified into 2 types:

- After Triggers (sometime called FOR triggers):
  - After trigger fires after the triggering action.
  - The INSERT, UPDATE and DELETE statements, causes an after trigger to fire after the respective statement complete execution
- Instead of Triggers
  - Instead of trigger fires instead of triggering action.
  - The INSERT, UPDATE and DELETE statements, causes an instead of trigger to fire INSTEAD OF the respective statement execution.

FOR Triggers

```
CREATE TRIGGER tr_employee_for_insert
ON Employee
FOR INSERT
AS
BEGIN
    DECLARE @id int
    SELECT @id=employee_id FROM inserted
    INSERT INTOaudit_employee
    VALUES('Employee added with employee_id =‘+
    Cast(@id as nvarchar(5))+ 'is added at’+
    Cast(Getdate() as nvarchar(20)))
END
```

ALTER and DROP Trigger

- **ALTER Trigger**
  - Is used to modify the definition of an existing trigger without altering the permissions or dependencies.
- **DROP Trigger**
  - Drops one or more triggers from the database.
  - You can delete multiple triggers using the DROP TRIGGER statement by specifying the trigger names separated by a comma.
  - DROP TRIGGER [IF EXISTS] [schema_name.]trigger1, trigger2, ...];
  - DROP TRIGGER tr_employee_for_insert;

THANK YOU!