

Session 4

TRIGGER

1. Concept

- Triggers are routines which are fired automatically whenever an event occurs:
 - (Trigger DML) After INSERT, UPDATE, DELETE against a table.
 - (Trigger DDL) Creating, dropping a table,...
 - (Trigger Logon) Logon

2. Create trigger (DML)

```
CREATE TRIGGER [schema_name.]trigger_name  
ON table_name  
AFTER {[INSERT],[UPDATE],[DELETE]}  
[NOT FOR REPLICATION]  
AS  
{sql_statements}
```

```
CREATE TRIGGER [schema_name.] trigger_name  
ON {table_name | view_name }  
INSTEAD OF {[INSERT] [,] [UPDATE] [,] [DELETE] }  
AS  
{sql_statements}
```

3. Virtual tables: INSERTED, DELETED

DML event	INSERTED table holds	DELETED table holds
INSERT	rows to be inserted	empty
UPDATE	new rows modified by the update	existing rows modified by the update
DELETE	empty	rows to be deleted

3. Example

```
CREATE TRIGGER
production.trg_product_audit
ON production.products
AFTER INSERT, DELETE
AS
BEGIN
    SET NOCOUNT ON;
    INSERT INTO production.product_audits(
        product_id, product_name,
        brand_id, category_id,
        model_year, list_price,
        updated_at, operation
    )
```

```
        SELECT
            i.product_id, product_name,
            brand_id, category_id,
            model_year, i.list_price,
            GETDATE(), 'INS'
        FROM
            inserted i
    UNION ALL
        SELECT
            d.product_id, product_name,
            brand_id, category_id,
            model_year, d.list_price,
            GETDATE(), 'DEL'
        FROM
            deleted d;
END
```

4. Exercises

1. Writing logs every time somebody changes (updates/inserts/deletes) the table employee.
2. Do not allow update more than 10% of salary