

From the above given tables perform the following queries:

**PART-A**

1. Create a trigger that fires on INSERT, UPDATE and DELETE operation on the Person table to display a message "Record is Affected."

```
CREATE TRIGGER trg_RecordAffected
ON Person
AFTER INSERT, UPDATE, DELETE
AS
BEGIN
    PRINT 'Record is Affected.';
END;
```

2. Create a trigger that fires on INSERT operation on the Person table to convert person name into uppercase whenever the record is inserted.

```
CREATE OR ALTER TRIGGER trg_UppercasePersonName
ON Person
FOR INSERT
AS
BEGIN
    DECLARE @pID INT
    SELECT @pID = PersonID FROM INSERTED
    UPDATE Person
    SET PersonName = UPPER(PersonName)
    WHERE PersonId = @pID;
END;
```

3. Create an INSERT trigger on person table, which calculates the age and update that age in Person table.

```
CREATE TRIGGER trg_CalculateAge
ON Person
AFTER INSERT
AS
BEGIN
    DECLARE @pID INT
    SELECT @pID = PersonID FROM INSERTED
    UPDATE Person
    SET Age = DATEDIFF(YEAR, BirthDate, GETDATE())
    WHERE PersonId = @pID;
END;
```

**PART-B**

4. Create a FOR/AFTER trigger those fires on INSERT, UPDATE and DELETE operation on the Person table. For that, create a new table PersonLog to log (enter) all operations performed on the person table.

– For Insert Operation on Person table

```
CREATE OR ALTER TRIGGER TR_Person_INSERT
ON Person
FOR INSERT
AS
BEGIN
DECLARE @PersonID INT, @PersonName VARCHAR(50)
SELECT @PersonId = INSERTED.PersonId,
       @PersonName = INSERTED.PersonName
FROM INSERTED
INSERT INTO PersonLog (PersonID, PersonName, Operation, UpdateDate)
VALUES (@PersonID, @PersonName, 'Insert', getdate())
END;
```

– For Update Operation on Person table

```
CREATE OR ALTER TRIGGER TR_Person_UPDATE
ON Person
FOR UPDATE
AS
BEGIN
DECLARE @PersonID INT, @PersonName VARCHAR(50)
SELECT @PersonId = INSERTED.PersonId,
       @PersonName = INSERTED.PersonName
FROM INSERTED
INSERT INTO PersonLog (PersonID, PersonName, Operation, UpdateDate)
VALUES (@PersonID, @PersonName, 'Update', getdate())
END;
```

– For Delete Operation on Person table

```
CREATE OR ALTER TRIGGER TR_Person_DELETE
ON Person
FOR DELETE
AS
BEGIN
DECLARE @PersonID INT, @PersonName VARCHAR(50)
SELECT @PersonId = INSERTED.PersonId,
       @PersonName = INSERTED.PersonName
FROM INSERTED
INSERT INTO PersonLog (PersonID, PersonName, Operation, UpdateDate)
VALUES (@PersonID, @PersonName, 'Delete', getdate())
```

END;

5. Create an INSTEAD OF trigger that fires on INSERT, UPDATE and DELETE operation on the Person table to display a message "Record is Affected...!!!"

```
CREATE TRIGGER tr_Person_RecordAffected
ON Person
INSTEAD OF INSERT, UPDATE, DELETE
AS
BEGIN
    PRINT 'Record is Affected...!!!';
END;
```

### PART-C

6. Create DELETE trigger on PersonLog table, when we delete any record of PersonLog table it prints 'Record deleted successfully from PersonLog'.

```
CREATE TRIGGER trg_DeletePersonLog
ON PersonLog
AFTER DELETE
AS
BEGIN
    PRINT 'Record deleted successfully from PersonLog table';
END;
```

**Darshan**  
UNIVERSITY  
योग: कर्मसु कौशलम्