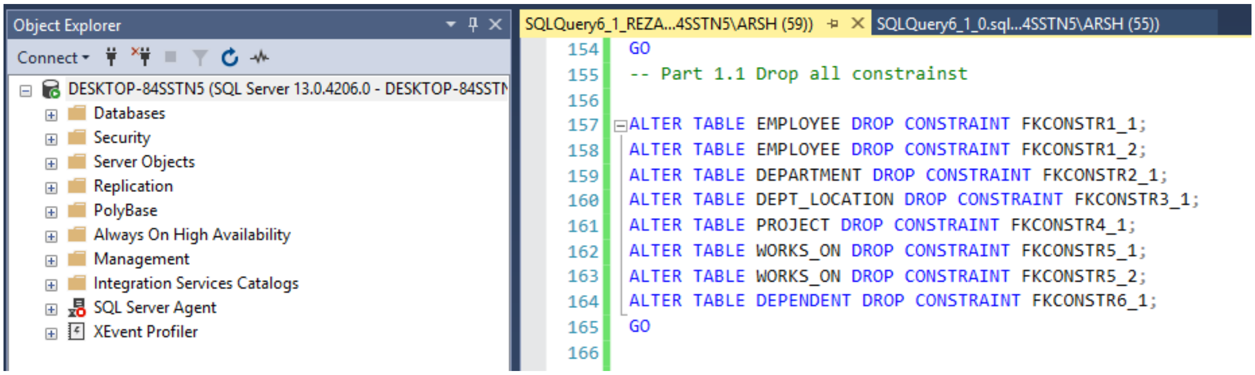
CIS530– Lab Assignment 6: Trigger and Stored Procedure
Name: Reza Shisheie
ID: 2708062

1. Write Triggers that implement Referential Integrity for FK Dno of Employee that is named as Constraint EMPDEPTFK in Table Employee as specified in Figure 4.2 in the textbook in the following DDL statements below for Company database that you created for Lab2. (You may keep the all the new tuples inserted in Lab3 and Lab4)

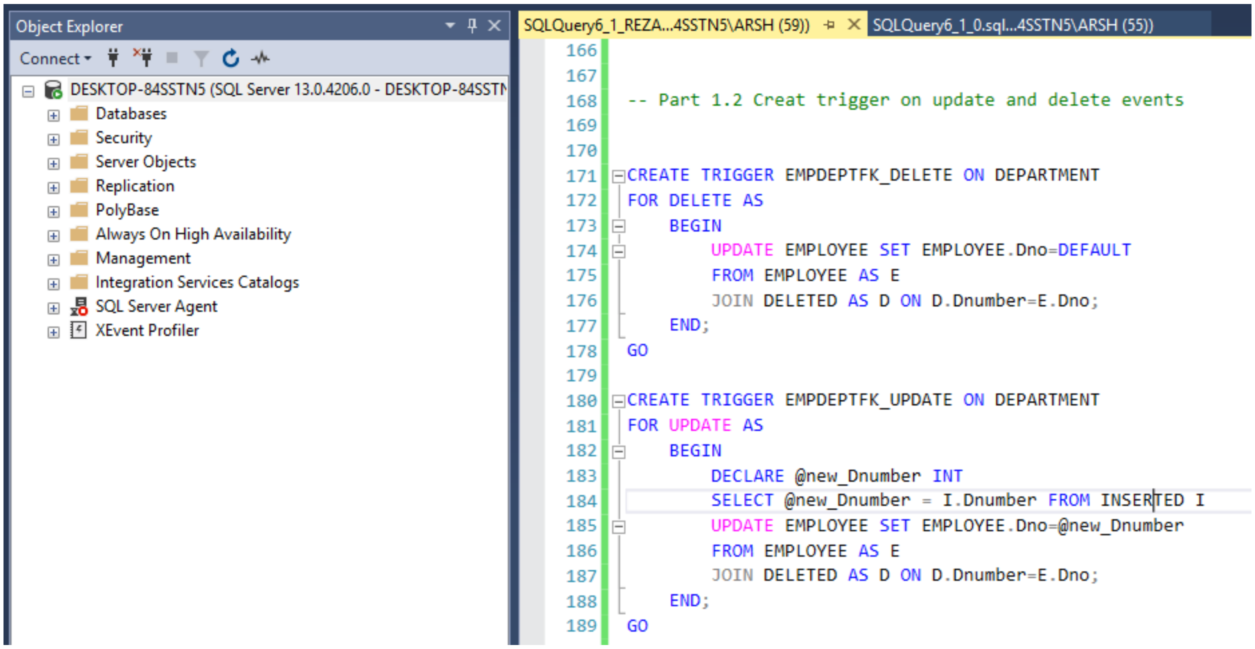
1) Before creating any trigger for this lab, Alter Table to Drop all the PK, FK, Unique Constraints, Cascade, Check options from the Tables Employee and Department for this lab to avoid any possible conflict with a system trigger or any table mutating problem.


Object Explorer
Connect ▾
DESKTOP-84SSTN5 (SQL Server 13.0.4206.0 - DESKTOP-84SSTN
Databases
Security
Server Objects
Replication
PolyBase
Always On High Availability
Management
Integration Services Catalogs
SQL Server Agent
XEvent Profiler
SQLQuery6_1_REZA...4SSTN5\ARSH (59))
SQLQuery6_1_0.sql...4SSTN5\ARSH (55))

```
GO
-- Part 1.1 Drop all constrainst

ALTER TABLE EMPLOYEE DROP CONSTRAINT FKCONSTR1_1;
ALTER TABLE EMPLOYEE DROP CONSTRAINT FKCONSTR1_2;
ALTER TABLE DEPARTMENT DROP CONSTRAINT FKCONSTR2_1;
ALTER TABLE DEPT_LOCATION DROP CONSTRAINT FKCONSTR3_1;
ALTER TABLE PROJECT DROP CONSTRAINT FKCONSTR4_1;
ALTER TABLE WORKS_ON DROP CONSTRAINT FKCONSTR5_1;
ALTER TABLE WORKS_ON DROP CONSTRAINT FKCONSTR5_2;
ALTER TABLE DEPENDENT DROP CONSTRAINT FKCONSTR6_1;
GO
```

```
-- Part 1.1 Drop all constrainst

ALTER TABLE EMPLOYEE DROP CONSTRAINT FKCONSTR1_1;
ALTER TABLE EMPLOYEE DROP CONSTRAINT FKCONSTR1_2;
ALTER TABLE DEPARTMENT DROP CONSTRAINT FKCONSTR2_1;
ALTER TABLE DEPT_LOCATION DROP CONSTRAINT FKCONSTR3_1;
ALTER TABLE PROJECT DROP CONSTRAINT FKCONSTR4_1;
ALTER TABLE WORKS_ON DROP CONSTRAINT FKCONSTR5_1;
ALTER TABLE WORKS_ON DROP CONSTRAINT FKCONSTR5_2;
ALTER TABLE DEPENDENT DROP CONSTRAINT FKCONSTR6_1;
GO
```

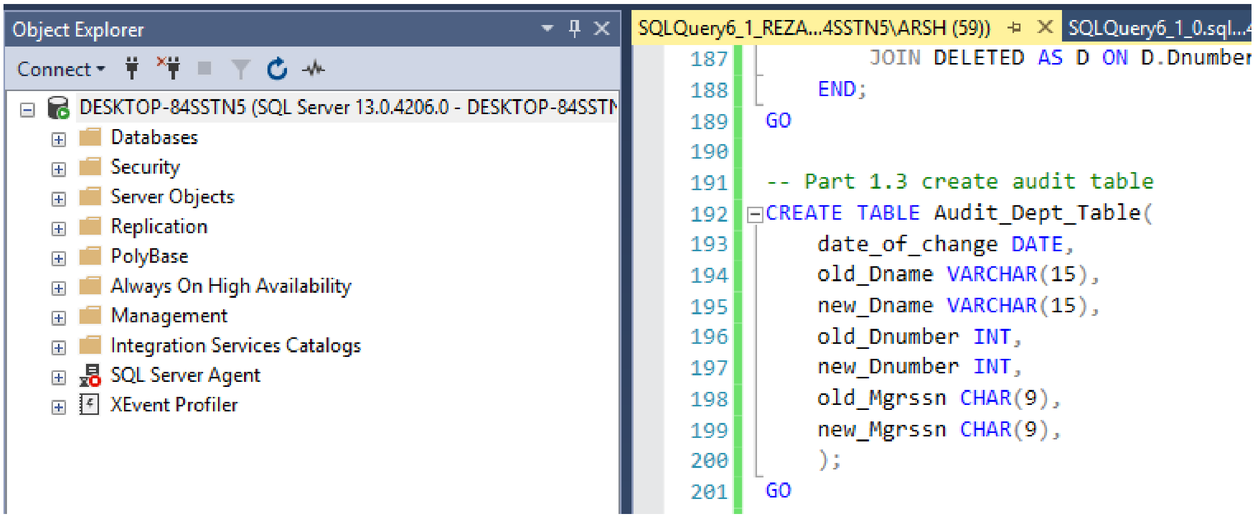
2) Write(Create) triggers to implement Constraint EMPDEPTFK in Table Employee based on the following rules as defined in DDL for Employee as in Figure 4.2:
FK Dno of Employee On Delete SET DEFAULT (= 1) and
FK Dno of Employee On Update CASCADE of Dnumber PK of Department


Object Explorer
Connect
DESKTOP-84SSTN5 (SQL Server 13.0.4206.0 - DESKTOP-84SSTN
Databases
Security
Server Objects
Replication
PolyBase
Always On High Availability
Management
Integration Services Catalogs
SQL Server Agent
XEvent Profiler
SQLQuery6_1_REZA...4SSTN5\ARSH (59))
SQLQuery6_1_0.sql...4SSTN5\ARSH (55))

```sql


-- Part 1.2 Creat trigger on update and delete events


CREATE TRIGGER EMPDEPTFK_DELETE ON DEPARTMENT
FOR DELETE AS
	BEGIN
		UPDATE EMPLOYEE SET EMPLOYEE.Dno=DEFAULT
		FROM EMPLOYEE AS E
		JOIN DELETED AS D ON D.Dnumber=E.Dno;
	END;
GO

CREATE TRIGGER EMPDEPTFK_UPDATE ON DEPARTMENT
FOR UPDATE AS
	BEGIN
		DECLARE @new_Dnumber INT
		SELECT @new_Dnumber = I.Dnumber FROM INSERTED I
		UPDATE EMPLOYEE SET EMPLOYEE.Dno=@new_Dnumber
		FROM EMPLOYEE AS E
		JOIN DELETED AS D ON D.Dnumber=E.Dno;
	END;
GO
```

```sql
-- Part 1.2 Creat trigger on update and delete events


CREATE TRIGGER EMPDEPTFK_DELETE ON DEPARTMENT
FOR DELETE AS
	BEGIN
		UPDATE EMPLOYEE SET EMPLOYEE.Dno=DEFAULT
		FROM EMPLOYEE AS E
		JOIN DELETED AS D ON D.Dnumber=E.Dno;
	END;
GO

CREATE TRIGGER EMPDEPTFK_UPDATE ON DEPARTMENT
FOR UPDATE AS
	BEGIN
		DECLARE @new_Dnumber INT
		SELECT @new_Dnumber = I.Dnumber FROM INSERTED I
		UPDATE EMPLOYEE SET EMPLOYEE.Dno=@new_Dnumber
		FROM EMPLOYEE AS E
		JOIN DELETED AS D ON D.Dnumber=E.Dno;
	END;
GO
```

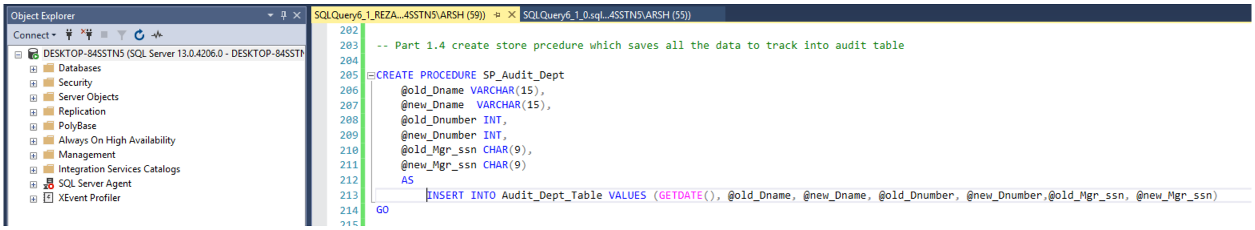
3) Write (Create) Stored Procedure *SP_Audit_Dept* that inserts all the history of the data of changes by the trigger you created in 1) above into a table *Audit_Dept_Table.* See for the more specific instructions that are given in 2 below.


Object Explorer
Connect
DESKTOP-84SSTN5 (SQL Server 13.0.4206.0 - DESKTOP-84SSTN
Databases
Security
Server Objects
Replication
PolyBase
Always On High Availability
Management
Integration Services Catalogs
SQL Server Agent
XEvent Profiler
SQLQuery6_1_REZA...4SSTN5\ARSH (59))
SQLQuery6_1_0.sql...

```
            JOIN DELETED AS D ON D.Dnumber
        END;
GO

-- Part 1.3 create audit table
CREATE TABLE Audit_Dept_Table(
        date_of_change DATE,
        old_Dname VARCHAR(15),
        new_Dname VARCHAR(15),
        old_Dnumber INT,
        new_Dnumber INT,
        old_Mgrssn CHAR(9),
        new_Mgrssn CHAR(9),
        );
GO
```

```
-- Part 1.3 create audit table
CREATE TABLE Audit_Dept_Table(
       date_of_change DATE,
       old_Dname VARCHAR(15),
       new_Dname VARCHAR(15),
       old_Dnumber INT,
       new_Dnumber   INT,
       old_Mgrssn CHAR(9),
       new_Mgrssn CHAR(9),
       );
GO
```

4) Call the Stored procedure *SP_Audit_Dept* at the end of your Trigger to record all the history of the changes by the trigger.

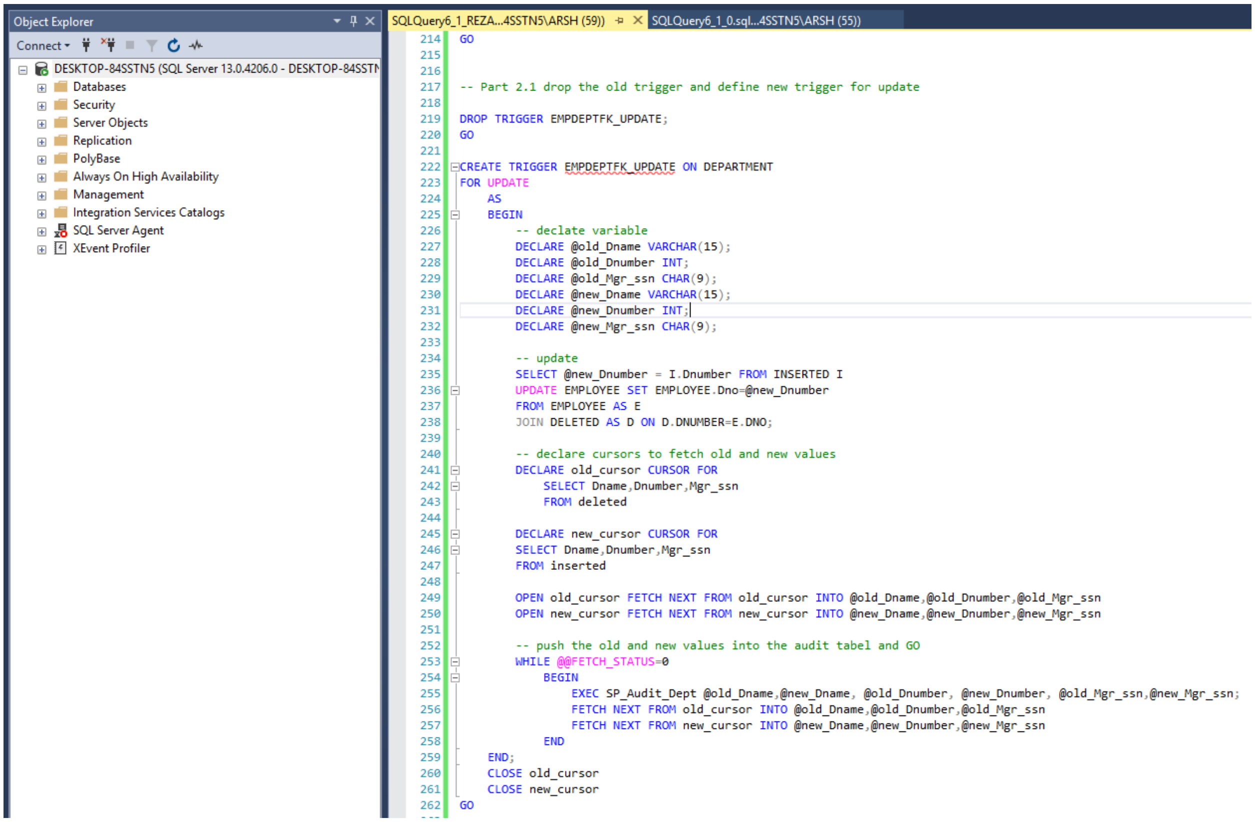

Object Explorer
Connect
DESKTOP-84SSTN5 (SQL Server 13.0.4206.0 - DESKTOP-84SSTN
Databases
Security
Server Objects
Replication
PolyBase
Always On High Availability
Management
Integration Services Catalogs
SQL Server Agent
XEvent Profiler
SQLQuery6_1_REZA...4SSTN5\ARSH (59))
SQLQuery6_1_0.sql...4SSTN5\ARSH (55))

```sql

-- Part 1.4 create store prcedure which saves all the data to track into audit table

CREATE PROCEDURE SP_Audit_Dept
    @old_Dname VARCHAR(15),
    @new_Dname  VARCHAR(15),
    @old_Dnumber INT,
    @new_Dnumber INT,
    @old_Mgr_ssn CHAR(9),
    @new_Mgr_ssn CHAR(9)
    AS
        INSERT INTO Audit_Dept_Table VALUES (GETDATE(), @old_Dname, @new_Dname, @old_Dnumber, @new_Dnumber,@old_Mgr_ssn, @new_Mgr_ssn)
GO
```

```sql
CREATE PROCEDURE SP_Audit_Dept
       @old_Dname VARCHAR(15),
       @new_Dname     VARCHAR(15),
       @old_Dnumber INT,
       @new_Dnumber INT,
       @old_Mgr_ssn CHAR(9),
       @new_Mgr_ssn CHAR(9)
       AS
              INSERT INTO Audit_Dept_Table VALUES (GETDATE(), @old_Dname, @new_Dname,
@old_Dnumber, @new_Dnumber,@old_Mgr_ssn, @new_Mgr_ssn)
GO
```

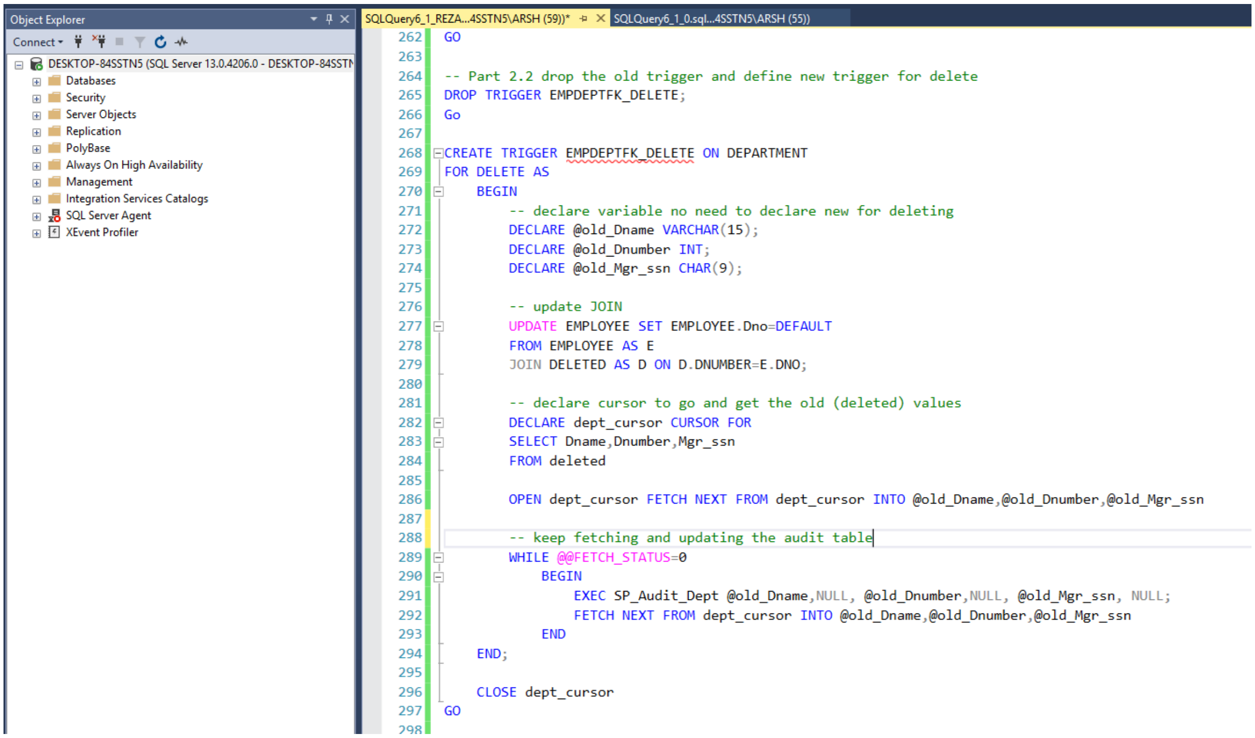
1) On Update of the trigger, Insert the new record into a table named *Audit_ Dept_Table* as follow:

(date_of_change, old_Dname, new_Dname, old_Dnumber, new_Dnumber, old_Mgrssn, new_Mgrssn)


Object Explorer
Connect
DESKTOP-84SSTN5 (SQL Server 13.0.4206.0 - DESKTOP-84SSTN
Databases
Security
Server Objects
Replication
PolyBase
Always On High Availability
Management
Integration Services Catalogs
SQL Server Agent
XEvent Profiler
SQLQuery6_1_REZA...4SSTN5\ARSH (59))
SQLQuery6_1_0.sql...4SSTN5\ARSH (55))

```sql
GO


-- Part 2.1 drop the old trigger and define new trigger for update

DROP TRIGGER EMPDEPTFK_UPDATE;
GO

CREATE TRIGGER EMPDEPTFK_UPDATE ON DEPARTMENT
FOR UPDATE
    AS
    BEGIN
        -- declate variable
        DECLARE @old_Dname VARCHAR(15);
        DECLARE @old_Dnumber INT;
        DECLARE @old_Mgr_ssn CHAR(9);
        DECLARE @new_Dname VARCHAR(15);
        DECLARE @new_Dnumber INT;
        DECLARE @new_Mgr_ssn CHAR(9);

        -- update
        SELECT @new_Dnumber = I.Dnumber FROM INSERTED I
        UPDATE EMPLOYEE SET EMPLOYEE.Dno=@new_Dnumber
        FROM EMPLOYEE AS E
        JOIN DELETED AS D ON D.DNUMBER=E.DNO;

        -- declare cursors to fetch old and new values
        DECLARE old_cursor CURSOR FOR
            SELECT Dname,Dnumber,Mgr_ssn
            FROM deleted

        DECLARE new_cursor CURSOR FOR
        SELECT Dname,Dnumber,Mgr_ssn
        FROM inserted

        OPEN old_cursor FETCH NEXT FROM old_cursor INTO @old_Dname,@old_Dnumber,@old_Mgr_ssn
        OPEN new_cursor FETCH NEXT FROM new_cursor INTO @new_Dname,@new_Dnumber,@new_Mgr_ssn

        -- push the old and new values into the audit tabel and GO
        WHILE @@FETCH_STATUS=0
            BEGIN
                EXEC SP_Audit_Dept @old_Dname,@new_Dname, @old_Dnumber, @new_Dnumber, @old_Mgr_ssn,@new_Mgr_ssn;
                FETCH NEXT FROM old_cursor INTO @old_Dname,@old_Dnumber,@old_Mgr_ssn
                FETCH NEXT FROM new_cursor INTO @new_Dname,@new_Dnumber,@new_Mgr_ssn
            END
    END;
    CLOSE old_cursor
    CLOSE new_cursor
GO
```

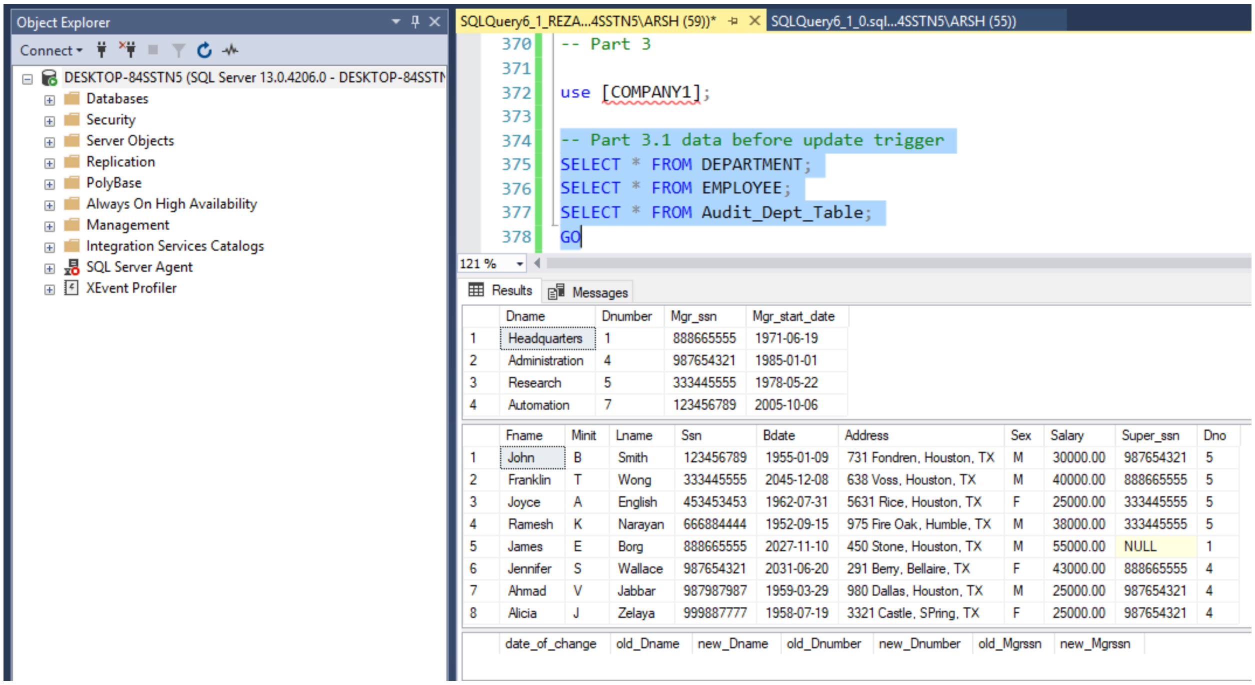
2) On Delete of the trigger, Insert the changes into *Audit_ Dept_Table* table as well. Since there is no new record for delete, so insert NULL for the new record columns new_Dname, new_Dnumber, new_Mgrssn of *Audit_ Dept_Table*.


Object Explorer
Connect
DESKTOP-84SSTN5 (SQL Server 13.0.4206.0 - DESKTOP-84SSTN
Databases
Security
Server Objects
Replication
PolyBase
Always On High Availability
Management
Integration Services Catalogs
SQL Server Agent
XEvent Profiler
SQLQuery6_1_REZA...4SSTN5\ARSH (59))*
SQLQuery6_1_0.sql...4SSTN5\ARSH (55))

```sql
GO

-- Part 2.2 drop the old trigger and define new trigger for delete
DROP TRIGGER EMPDEPTFK_DELETE;
Go

CREATE TRIGGER EMPDEPTFK_DELETE ON DEPARTMENT
FOR DELETE AS
    BEGIN
        -- declare variable no need to declare new for deleting
        DECLARE @old_Dname VARCHAR(15);
        DECLARE @old_Dnumber INT;
        DECLARE @old_Mgr_ssn CHAR(9);

        -- update JOIN
        UPDATE EMPLOYEE SET EMPLOYEE.Dno=DEFAULT
        FROM EMPLOYEE AS E
        JOIN DELETED AS D ON D.DNUMBER=E.DNO;

        -- declare cursor to go and get the old (deleted) values
        DECLARE dept_cursor CURSOR FOR
        SELECT Dname,Dnumber,Mgr_ssn
        FROM deleted

        OPEN dept_cursor FETCH NEXT FROM dept_cursor INTO @old_Dname,@old_Dnumber,@old_Mgr_ssn

        -- keep fetching and updating the audit table
        WHILE @@FETCH_STATUS=0
            BEGIN
                EXEC SP_Audit_Dept @old_Dname,NULL, @old_Dnumber,NULL, @old_Mgr_ssn, NULL;
                FETCH NEXT FROM dept_cursor INTO @old_Dname,@old_Dnumber,@old_Mgr_ssn
            END
    END;

    CLOSE dept_cursor
GO
```

1. Date before the following change:

UPDATE Department Dnumber 4 to 99



Object Explorer
Connect
DESKTOP-84SSTN5 (SQL Server 13.0.4206.0 - DESKTOP-84SSTN
Databases
Security
Server Objects
Replication
PolyBase
Always On High Availability
Management
Integration Services Catalogs
SQL Server Agent
XEvent Profiler
SQLQuery6_1_REZA...4SSTN5\ARSH (59))*
SQLQuery6_1_0.sql...4SSTN5\ARSH (55))
121 %
Results
Messages

```sql
-- Part 3

use [COMPANY1];

-- Part 3.1 data before update trigger
SELECT * FROM DEPARTMENT;
SELECT * FROM EMPLOYEE;
SELECT * FROM Audit_Dept_Table;
GO
```

	Dname	Dnumber	Mgr_ssn	Mgr_start_date
1	Headquarters	1	888665555	1971-06-19
2	Administration	4	987654321	1985-01-01
3	Research	5	333445555	1978-05-22
4	Automation	7	123456789	2005-10-06

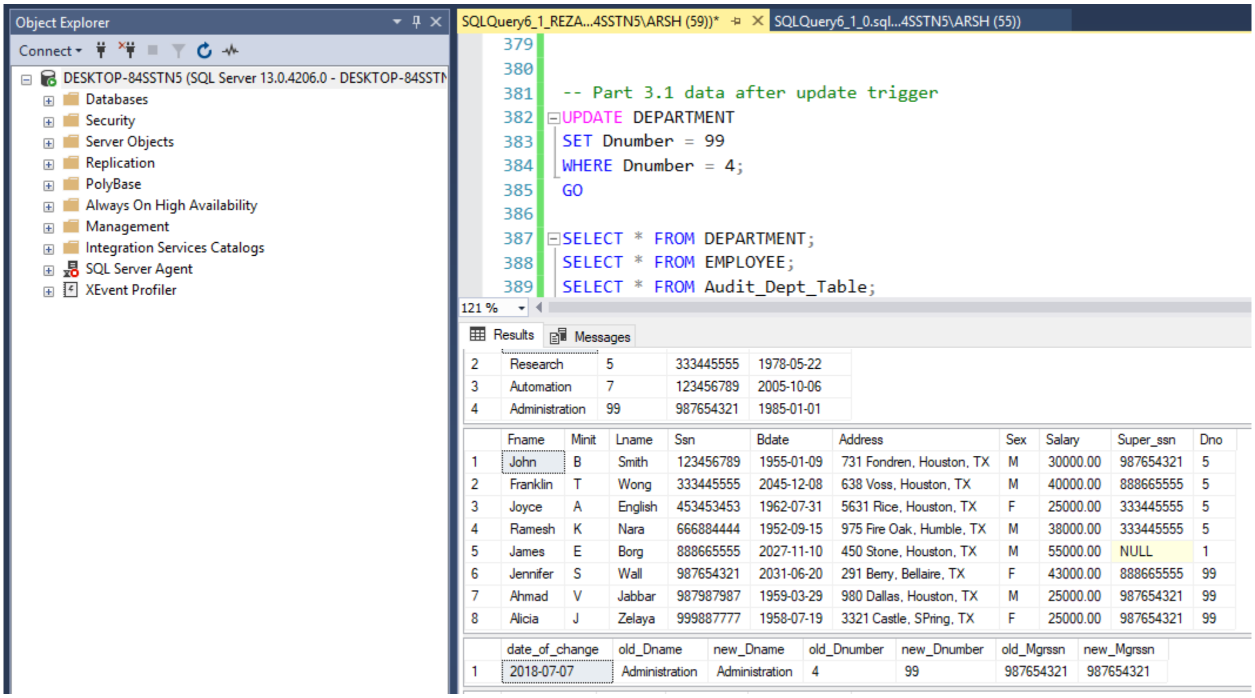
	Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
1	John	B	Smith	123456789	1955-01-09	731 Fondren, Houston, TX	M	30000.00	987654321	5
2	Franklin	T	Wong	333445555	2045-12-08	638 Voss, Houston, TX	M	40000.00	888665555	5
3	Joyce	A	English	453453453	1962-07-31	5631 Rice, Houston, TX	F	25000.00	333445555	5
4	Ramesh	K	Narayan	666884444	1952-09-15	975 Fire Oak, Humble, TX	M	38000.00	333445555	5
5	James	E	Borg	888665555	2027-11-10	450 Stone, Houston, TX	M	55000.00	NULL	1
6	Jennifer	S	Wallace	987654321	2031-06-20	291 Berry, Bellaire, TX	F	43000.00	888665555	4
7	Ahmad	V	Jabbar	987987987	1959-03-29	980 Dallas, Houston, TX	M	25000.00	987654321	4
8	Alicia	J	Zelaya	999887777	1958-07-19	3321 Castle, SPring, TX	F	25000.00	987654321	4

	date_of_change	old_Dname	new_Dname	old_Dnumber	new_Dnumber	old_Mgrssn	new_Mgrssn

As shown above all values are what they were inserted, and the audit table is empty

2. Date after the following change:

UPDATE Department Dnumber 4 to 99


Object Explorer
Connect
DESKTOP-84SSTN5 (SQL Server 13.0.4206.0 - DESKTOP-84SSTN
Databases
Security
Server Objects
Replication
PolyBase
Always On High Availability
Management
Integration Services Catalogs
SQL Server Agent
XEvent Profiler
SQLQuery6_1_REZA...4SSTN5\ARSH (59))*
SQLQuery6_1_0.sql...4SSTN5\ARSH (55))
121 %
Results
Messages

```
-- Part 3.1 data after update trigger
UPDATE DEPARTMENT
SET Dnumber = 99
WHERE Dnumber = 4;
GO

SELECT * FROM DEPARTMENT;
SELECT * FROM EMPLOYEE;
SELECT * FROM Audit_Dept_Table;
```

2	Research	5	333445555	1978-05-22
3	Automation	7	123456789	2005-10-06
4	Administration	99	987654321	1985-01-01

	Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
1	John	B	Smith	123456789	1955-01-09	731 Fondren, Houston, TX	M	30000.00	987654321	5
2	Franklin	T	Wong	333445555	2045-12-08	638 Voss, Houston, TX	M	40000.00	888665555	5
3	Joyce	A	English	453453453	1962-07-31	5631 Rice, Houston, TX	F	25000.00	333445555	5
4	Ramesh	K	Nara	666884444	1952-09-15	975 Fire Oak, Humble, TX	M	38000.00	333445555	5
5	James	E	Borg	888665555	2027-11-10	450 Stone, Houston, TX	M	55000.00	NULL	1
6	Jennifer	S	Wall	987654321	2031-06-20	291 Berry, Bellaire, TX	F	43000.00	888665555	99
7	Ahmad	V	Jabbar	987987987	1959-03-29	980 Dallas, Houston, TX	M	25000.00	987654321	99
8	Alicia	J	Zelaya	999887777	1958-07-19	3321 Castle, SPring, TX	F	25000.00	987654321	99

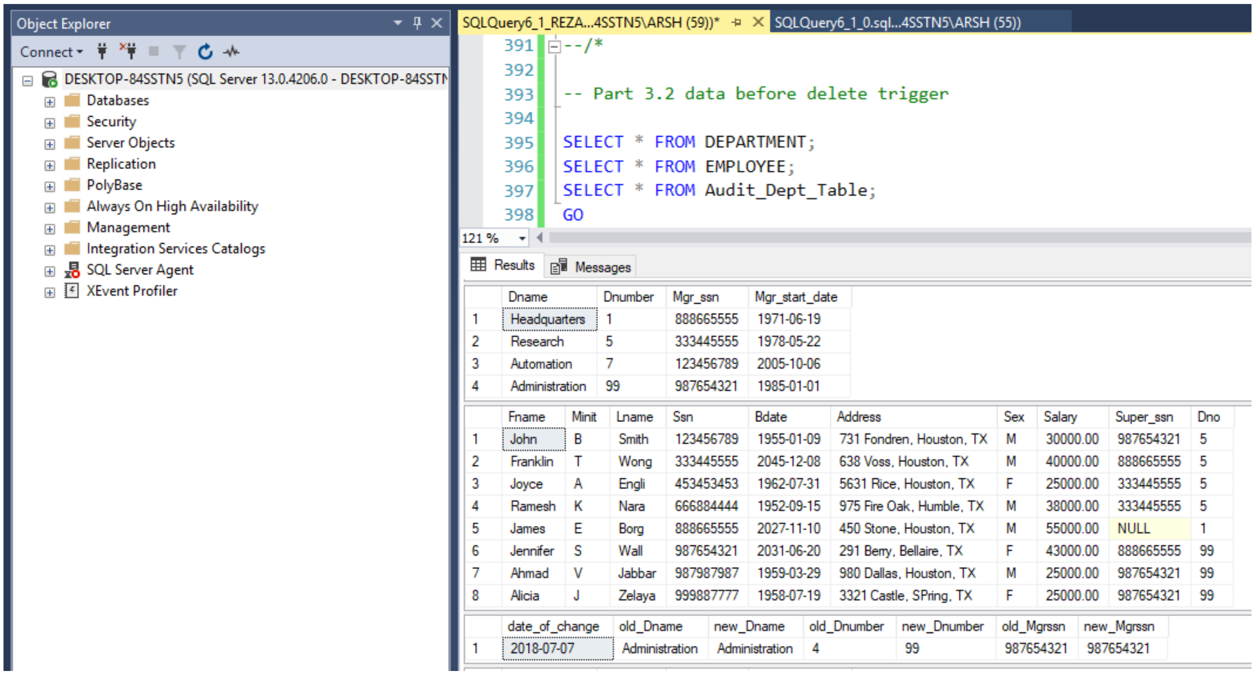
	date_of_change	old_Dname	new_Dname	old_Dnumber	new_Dnumber	old_Mgrssn	new_Mgrssn
1	2018-07-07	Administration	Administration	4	99	987654321	987654321

As shown, the department number in department table is changed from 4 to 99 and consequently the department of all employees has to change from 4 to 99. This change is shown above.

In addition, the change is recorded into audit table with the new data changed added to the table

3. Date after the following change:

DELETE Department Dnumber 5


Object Explorer
Connect
DESKTOP-84SSTN5 (SQL Server 13.0.4206.0 - DESKTOP-84SSTN
Databases
Security
Server Objects
Replication
PolyBase
Always On High Availability
Management
Integration Services Catalogs
SQL Server Agent
XEvent Profiler
SQLQuery6_1_REZA...4SSTN5\ARSH (59))*
SQLQuery6_1_0.sql...4SSTN5\ARSH (55))
121 %
Results
Messages

```
--/*

-- Part 3.2 data before delete trigger

SELECT * FROM DEPARTMENT;
SELECT * FROM EMPLOYEE;
SELECT * FROM Audit_Dept_Table;
GO
```

	Dname	Dnumber	Mgr_ssn	Mgr_start_date
1	Headquarters	1	888665555	1971-06-19
2	Research	5	333445555	1978-05-22
3	Automation	7	123456789	2005-10-06
4	Administration	99	987654321	1985-01-01

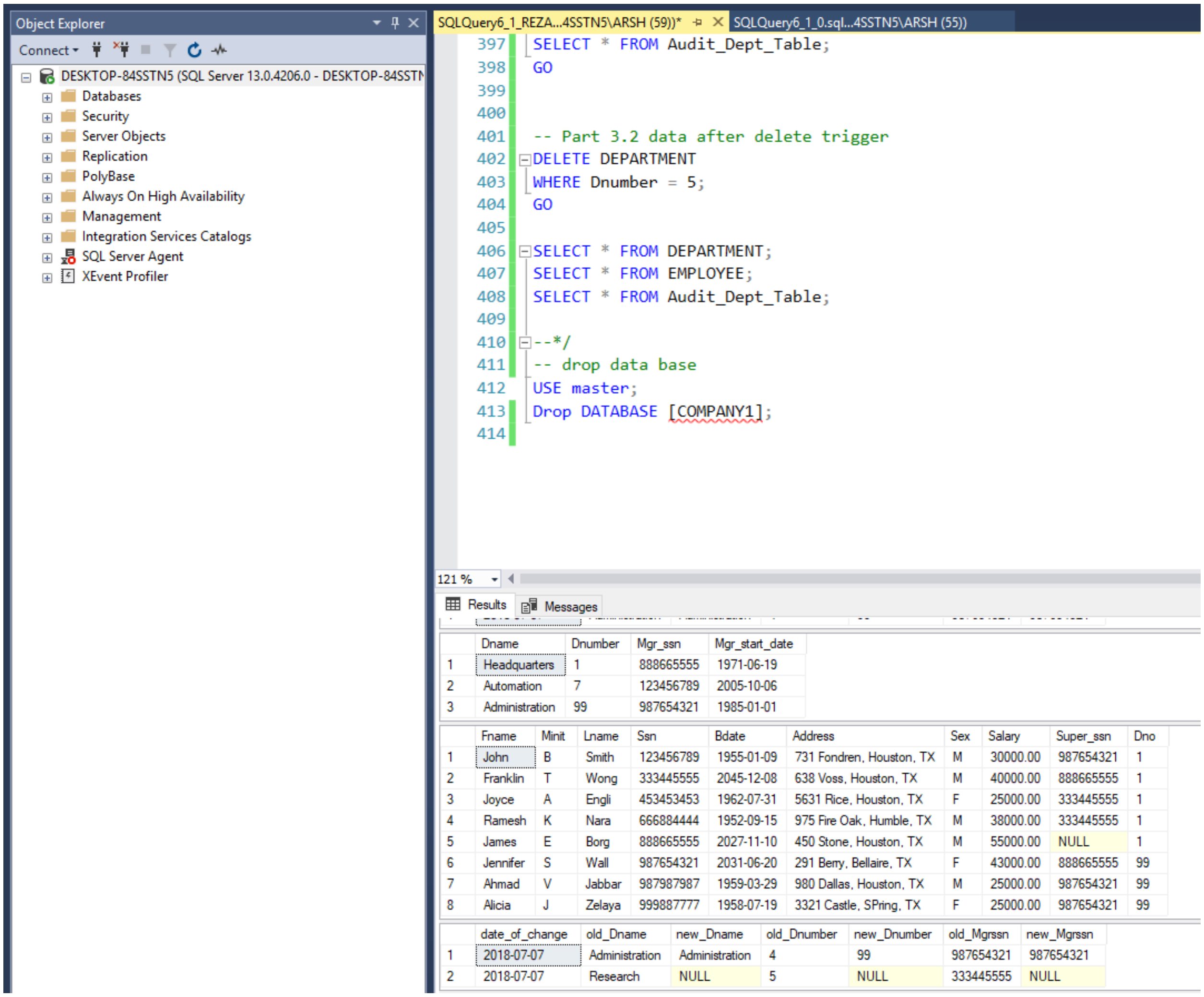
	Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
1	John	B	Smith	123456789	1955-01-09	731 Fondren, Houston, TX	M	30000.00	987654321	5
2	Franklin	T	Wong	333445555	2045-12-08	638 Voss, Houston, TX	M	40000.00	888665555	5
3	Joyce	A	Engli	453453453	1962-07-31	5631 Rice, Houston, TX	F	25000.00	333445555	5
4	Ramesh	K	Nara	666884444	1952-09-15	975 Fire Oak, Humble, TX	M	38000.00	333445555	5
5	James	E	Borg	888665555	2027-11-10	450 Stone, Houston, TX	M	55000.00	NULL	1
6	Jennifer	S	Wall	987654321	2031-06-20	291 Berry, Bellaire, TX	F	43000.00	888665555	99
7	Ahmad	V	Jabbar	987987987	1959-03-29	980 Dallas, Houston, TX	M	25000.00	987654321	99
8	Alicia	J	Zelaya	999887777	1958-07-19	3321 Castle, SPring, TX	F	25000.00	987654321	99

	date_of_change	old_Dname	new_Dname	old_Dnumber	new_Dnumber	old_Mgrssn	new_Mgrssn
1	2018-07-07	Administration	Administration	4	99	987654321	987654321

this is essentially the same as the result of previous section.

4. Date after the following change:

DELETE Department Dnumber 5


Object Explorer
Connect
DESKTOP-84SSTN5 (SQL Server 13.0.4206.0 - DESKTOP-84SSTN
Databases
Security
Server Objects
Replication
PolyBase
Always On High Availability
Management
Integration Services Catalogs
SQL Server Agent
XEvent Profiler
SQLQuery6_1_REZA...4SSTN5\ARSH (59))*
SQLQuery6_1_0.sql...4SSTN5\ARSH (55))
121 %
Results
Messages

```
SELECT * FROM Audit_Dept_Table;
GO

-- Part 3.2 data after delete trigger
DELETE DEPARTMENT
WHERE Dnumber = 5;
GO

SELECT * FROM DEPARTMENT;
SELECT * FROM EMPLOYEE;
SELECT * FROM Audit_Dept_Table;

--*/
-- drop data base
USE master;
Drop DATABASE [COMPANY1];

```

	Dname	Dnumber	Mgr_ssn	Mgr_start_date
1	Headquarters	1	888665555	1971-06-19
2	Automation	7	123456789	2005-10-06
3	Administration	99	987654321	1985-01-01

	Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
1	John	B	Smith	123456789	1955-01-09	731 Fondren, Houston, TX	M	30000.00	987654321	1
2	Franklin	T	Wong	333445555	2045-12-08	638 Voss, Houston, TX	M	40000.00	888665555	1
3	Joyce	A	Engli	453453453	1962-07-31	5631 Rice, Houston, TX	F	25000.00	333445555	1
4	Ramesh	K	Nara	666884444	1952-09-15	975 Fire Oak, Humble, TX	M	38000.00	333445555	1
5	James	E	Borg	888665555	2027-11-10	450 Stone, Houston, TX	M	55000.00	NULL	1
6	Jennifer	S	Wall	987654321	2031-06-20	291 Berry, Bellaire, TX	F	43000.00	888665555	99
7	Ahmad	V	Jabbar	987987987	1959-03-29	980 Dallas, Houston, TX	M	25000.00	987654321	99
8	Alicia	J	Zelaya	999887777	1958-07-19	3321 Castle, SPring, TX	F	25000.00	987654321	99

	date_of_change	old_Dname	new_Dname	old_Dnumber	new_Dnumber	old_Mgrssn	new_Mgrssn
1	2018-07-07	Administration	Administration	4	99	987654321	987654321
2	2018-07-07	Research	NULL	5	NULL	333445555	NULL

As shown above which ever department is selected (department 5) must be removed. The first table shows that there is department with department number 5 available which means it is removed.

In addition, all changes have to be recorded into the audit table. The first line of audit table shows the changes made in previous section (update) and the second row shows changes in the second section (delete). In the second row there is no new value added since the value is deleted and thus does not exist anymore.

CIS530– Lab Assignment 6: Trigger and Stored Procedure
Name: Reza Shisheie
ID: 2708062

Code

```
USE master;

-- define a data base called "COMPANY"
IF DB_ID('COMPANY1') IS NULL
       CREATE DATABASE COMPANY1;
       --DROP DATABASE COMPANY1;

GO

use COMPANY1;
go

-- droping all dependencies and constrainsts
/*
ALTER TABLE EMPLOYEE DROP CONSTRAINT FKCONSTR1_1;
ALTER TABLE EMPLOYEE DROP CONSTRAINT FKCONSTR1_2;
ALTER TABLE DEPARTMENT DROP CONSTRAINT FKCONSTR2_1;
ALTER TABLE DEPT_LOCATION DROP CONSTRAINT FKCONSTR3_1;
ALTER TABLE PROJECT DROP CONSTRAINT FKCONSTR4_1;
ALTER TABLE WORKS_ON DROP CONSTRAINT FKCONSTR5_1;
ALTER TABLE WORKS_ON DROP CONSTRAINT FKCONSTR5_2;
ALTER TABLE DEPENDENT DROP CONSTRAINT FKCONSTR6_1;

-- droping and removing all tables
drop table DEPARTMENT;
drop table EMPLOYEE;
drop table DEPT_LOCATION
drop table PROJECT
drop table WORKS_ON
drop table DEPENDENT
DROP VIEW VDept_Budget;
*/


IF OBJECT_ID('dbo.EMPLOYEE') IS NULL

       create table EMPLOYEE (
       Fname VARCHAR(15) NOT NULL,
       Minit CHAR,
       Lname VARCHAR(15) NOT NULL,
       Ssn CHAR(9) NOT NULL,
       Bdate DATE,
       Address VARCHAR(30),
       Sex CHAR,
       Salary DECIMAL(10,2),
       Super_ssn CHAR(9),
       Dno INT NOT NULL default 1,

       PRIMARY KEY (Ssn),
       );

-- Create a Table DEPARTMENT in COMPANY database

IF OBJECT_ID('dbo.DEPARTMENT') IS NULL
```

```
       CREATE TABLE dbo.DEPARTMENT(
       Dname VARCHAR(15) NOT NULL,
       Dnumber INT NOT NULL,
       Mgr_ssn CHAR(9) NOT NULL,
       Mgr_start_date DATE

       PRIMARY KEY (Dnumber),
       UNIQUE (Dname),
       );

       ALTER TABLE EMPLOYEE
              ADD CONSTRAINT FKCONSTR1_1
                     FOREIGN KEY (Super_ssn) REFERENCES EMPLOYEE(Ssn);

       ALTER TABLE EMPLOYEE
              ADD CONSTRAINT FKCONSTR1_2
                     FOREIGN KEY (Dno) REFERENCES DEPARTMENT(Dnumber);

       ALTER TABLE DEPARTMENT
              ADD CONSTRAINT FKCONSTR2_1
                     FOREIGN KEY (Mgr_ssn) REFERENCES EMPLOYEE(Ssn);



IF OBJECT_ID('dbo.DEPT_LOCATION') IS NULL
       CREATE TABLE dbo.DEPT_LOCATION(
       Dnumber INT NOT NULL,
       Dlocation VARCHAR(30),

       PRIMARY KEY (Dnumber, Dlocation),
       --FOREIGN KEY (Dnumber) REFERENCES DEPARTMENT(Dnumber),
       );
       ALTER TABLE DEPT_LOCATION
              ADD CONSTRAINT FKCONSTR3_1
                     FOREIGN KEY (Dnumber) REFERENCES DEPARTMENT(Dnumber);



IF OBJECT_ID('dbo.PROJECT') IS NULL
       CREATE TABLE dbo.PROJECT(
       Pname VARCHAR(15) NOT NULL,
       Pnumber INT NOT NULL,
       Plocation VARCHAR(30),
       Dnum INT NOT NULL,

       PRIMARY KEY (Pnumber),
       --FOREIGN KEY (Dnum) REFERENCES DEPARTMENT(Dnumber),
       );
       ALTER TABLE PROJECT
              ADD CONSTRAINT FKCONSTR4_1
                     FOREIGN KEY (Dnum) REFERENCES DEPARTMENT(Dnumber);
```

```
 IF OBJECT_ID('dbo.WORKS_ON') IS NULL
        CREATE TABLE dbo.WORKS_ON(
        Essn CHAR(9) NOT NULL,
        Pno INT NOT NULL,
        Hours VARCHAR(5),

        PRIMARY KEY (Essn, Pno),
        --FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn),
        --FOREIGN KEY (Pno) REFERENCES PROJECT(Pnumber),
        );
        ALTER TABLE WORKS_ON
               ADD CONSTRAINT FKCONSTR5_1
                      FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn);
        ALTER TABLE WORKS_ON
               ADD CONSTRAINT FKCONSTR5_2
                      FOREIGN KEY (Pno) REFERENCES PROJECT(Pnumber);


 IF OBJECT_ID('dbo.DEPENDENT') IS NULL
        CREATE TABLE dbo.DEPENDENT(
        Essn CHAR(9) NOT NULL,
        Dependent_name VARCHAR(15) NOT NULL,
        Sex VARCHAR(5) NOT NULL,
        Bdate DATE,
        Relationship VARCHAR(15) NOT NULL,

        PRIMARY KEY (Essn,Dependent_name),
        --FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn),
        );
        ALTER TABLE DEPENDENT
               ADD CONSTRAINT FKCONSTR6_1
                      FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn);


--  disabling  FK with NOCHECK Option as below
ALTER TABLE COMPANY1.dbo.EMPLOYEE NOCHECK CONSTRAINT FKCONSTR1_1;
ALTER TABLE COMPANY1.dbo.EMPLOYEE NOCHECK CONSTRAINT FKCONSTR1_2;
ALTER TABLE COMPANY1.dbo.DEPARTMENT NOCHECK CONSTRAINT FKCONSTR2_1;
ALTER TABLE COMPANY1.dbo.DEPT_LOCATION NOCHECK CONSTRAINT FKCONSTR3_1;
ALTER TABLE COMPANY1.dbo.PROJECT NOCHECK CONSTRAINT FKCONSTR4_1;
ALTER TABLE COMPANY1.dbo.WORKS_ON NOCHECK CONSTRAINT FKCONSTR5_1;
ALTER TABLE COMPANY1.dbo.WORKS_ON NOCHECK CONSTRAINT FKCONSTR5_2;
ALTER TABLE COMPANY1.dbo.DEPENDENT NOCHECK CONSTRAINT FKCONSTR6_1;
GO
-- Part 1.1 Drop all constrainst

ALTER TABLE EMPLOYEE DROP CONSTRAINT FKCONSTR1_1;
ALTER TABLE EMPLOYEE DROP CONSTRAINT FKCONSTR1_2;
ALTER TABLE DEPARTMENT DROP CONSTRAINT FKCONSTR2_1;
ALTER TABLE DEPT_LOCATION DROP CONSTRAINT FKCONSTR3_1;
ALTER TABLE PROJECT DROP CONSTRAINT FKCONSTR4_1;
ALTER TABLE WORKS_ON DROP CONSTRAINT FKCONSTR5_1;
ALTER TABLE WORKS_ON DROP CONSTRAINT FKCONSTR5_2;
ALTER TABLE DEPENDENT DROP CONSTRAINT FKCONSTR6_1;
GO
```

```
-- Part 1.2 Creat trigger on update and delete events


CREATE TRIGGER EMPDEPTFK_DELETE ON DEPARTMENT
FOR DELETE AS
       BEGIN
              UPDATE EMPLOYEE SET EMPLOYEE.Dno=DEFAULT
              FROM EMPLOYEE AS E
              JOIN DELETED AS D ON D.Dnumber=E.Dno;
       END;
GO

CREATE TRIGGER EMPDEPTFK_UPDATE ON DEPARTMENT
FOR UPDATE AS
       BEGIN
              DECLARE @new_Dnumber INT
              SELECT @new_Dnumber = I.Dnumber FROM INSERTED I
              UPDATE EMPLOYEE SET EMPLOYEE.Dno=@new_Dnumber
              FROM EMPLOYEE AS E
              JOIN DELETED AS D ON D.Dnumber=E.Dno;
       END;
GO

-- Part 1.3 create audit table
CREATE TABLE Audit_Dept_Table(
       date_of_change DATE,
       old_Dname VARCHAR(15),
       new_Dname VARCHAR(15),
       old_Dnumber INT,
       new_Dnumber   INT,
       old_Mgrssn CHAR(9),
       new_Mgrssn CHAR(9),
       );
GO

-- Part 1.4 create store prcedure which saves all the data to track into audit table

CREATE PROCEDURE SP_Audit_Dept
       @old_Dname VARCHAR(15),
       @new_Dname    VARCHAR(15),
       @old_Dnumber INT,
       @new_Dnumber INT,
       @old_Mgr_ssn CHAR(9),
       @new_Mgr_ssn CHAR(9)
       AS
              INSERT INTO Audit_Dept_Table VALUES (GETDATE(), @old_Dname, @new_Dname,
@old_Dnumber, @new_Dnumber,@old_Mgr_ssn, @new_Mgr_ssn)
GO


-- Part 2.1 drop the old trigger and define new trigger for update

DROP TRIGGER EMPDEPTFK_UPDATE;
GO
```

```
CREATE TRIGGER EMPDEPTFK_UPDATE ON DEPARTMENT
FOR UPDATE
        AS
        BEGIN
                -- declate variable
                DECLARE @old_Dname VARCHAR(15);
                DECLARE @old_Dnumber INT;
                DECLARE @old_Mgr_ssn CHAR(9);
                DECLARE @new_Dname VARCHAR(15);
                DECLARE @new_Dnumber INT;
                DECLARE @new_Mgr_ssn CHAR(9);

                -- update
                SELECT @new_Dnumber = I.Dnumber FROM INSERTED I
                UPDATE EMPLOYEE SET EMPLOYEE.Dno=@new_Dnumber
                FROM EMPLOYEE AS E
                JOIN DELETED AS D ON D.DNUMBER=E.DNO;

                -- declare cursors to fetch old and new values
                DECLARE old_cursor CURSOR FOR
                        SELECT Dname,Dnumber,Mgr_ssn
                        FROM deleted

                DECLARE new_cursor CURSOR FOR
                SELECT Dname,Dnumber,Mgr_ssn
                FROM inserted

                OPEN old_cursor FETCH NEXT FROM old_cursor INTO
@old_Dname,@old_Dnumber,@old_Mgr_ssn
                OPEN new_cursor FETCH NEXT FROM new_cursor INTO
@new_Dname,@new_Dnumber,@new_Mgr_ssn

                -- push the old and new values into the audit tabel and GO
                WHILE @@FETCH_STATUS=0
                        BEGIN
                                EXEC SP_Audit_Dept @old_Dname,@new_Dname, @old_Dnumber,
@new_Dnumber, @old_Mgr_ssn,@new_Mgr_ssn;
                                FETCH NEXT FROM old_cursor INTO
@old_Dname,@old_Dnumber,@old_Mgr_ssn
                                FETCH NEXT FROM new_cursor INTO
@new_Dname,@new_Dnumber,@new_Mgr_ssn
                        END
        END;
        CLOSE old_cursor
        CLOSE new_cursor
GO

-- Part 2.2 drop the old trigger and define new trigger for delete
DROP TRIGGER EMPDEPTFK_DELETE;
Go

CREATE TRIGGER EMPDEPTFK_DELETE ON DEPARTMENT
FOR DELETE AS
        BEGIN
                -- declare variable no need to declare new for deleting
                DECLARE @old_Dname VARCHAR(15);
```

```
              DECLARE @old_Dnumber INT;
              DECLARE @old_Mgr_ssn CHAR(9);

              -- update JOIN
              UPDATE EMPLOYEE SET EMPLOYEE.Dno=DEFAULT
              FROM EMPLOYEE AS E
              JOIN DELETED AS D ON D.DNUMBER=E.DNO;

              -- declare cursor to go and get the old (deleted) values
              DECLARE dept_cursor CURSOR FOR
              SELECT Dname,Dnumber,Mgr_ssn
              FROM deleted

              OPEN dept_cursor FETCH NEXT FROM dept_cursor INTO
@old_Dname,@old_Dnumber,@old_Mgr_ssn

              -- keep fetching and updating the audit table
              WHILE @@FETCH_STATUS=0
                     BEGIN
                            EXEC SP_Audit_Dept @old_Dname,NULL, @old_Dnumber,NULL,
@old_Mgr_ssn, NULL;
                            FETCH NEXT FROM dept_cursor INTO
@old_Dname,@old_Dnumber,@old_Mgr_ssn
                     END
       END;

       CLOSE dept_cursor
GO




--Let’s insert data into EMPLOYEE
insert into EMPLOYEE values ( 'John', 'B' ,'Smith','123456789','9-Jan-55', '731 Fondren,
Houston, TX', 'M', '30000', '987654321', '5');
insert into EMPLOYEE values ( 'Franklin', 'T' ,'Wong','333445555','08-Dec-45', '638 Voss,
Houston, TX', 'M', '40000', '888665555', '5');
insert into EMPLOYEE values ( 'Joyce', 'A' ,'English','453453453','31-Jul-62', '5631
Rice, Houston, TX ', 'F', '25000', '333445555', '5');
insert into EMPLOYEE values ( 'Ramesh', 'K' ,'Narayan','666884444','15-Sep-52', '975 Fire
Oak, Humble, TX', 'M', '38000', '333445555', '5');
insert into EMPLOYEE values ( 'James', 'E' ,'Borg','888665555','10-Nov-27', '450 Stone,
Houston, TX ', 'M', '55000', null, '1');
insert into EMPLOYEE values ( 'Jennifer', 'S' ,'Wallace','987654321','20-Jun-31', '291
Berry, Bellaire, TX', 'F', '43000', '888665555', '4');
insert into EMPLOYEE values ( 'Ahmad', 'V' ,'Jabbar','987987987','29-Mar-59', '980
Dallas, Houston, TX', 'M', '25000', '987654321', '4');
insert into EMPLOYEE values ( 'Alicia', 'J' ,'Zelaya','999887777','19-Jul-58', '3321
Castle, SPring, TX', 'F', '25000', '987654321', '4');


--Let’s insert data into DEPARTMENT
insert into DEPARTMENT values ( 'Headquarters', '1' ,'888665555','19-Jun-71');
insert into DEPARTMENT values ( 'Administration', '4' ,'987654321','01-Jan-85');
insert into DEPARTMENT values ( 'Research', '5' ,'333445555','22-May-78');
insert into DEPARTMENT values ( 'Automation', '7' ,'123456789','06-Oct-05');


--Let’s insert data into DEPENDENT
insert into DEPENDENT values ( '123456789', 'Alice' ,'F','31-Dec-78','Daughter');
insert into DEPENDENT values ( '123456789', 'Elizabeth' ,'F','05-May-57','Spouse');
insert into DEPENDENT values ( '123456789', 'Michael' ,'M','01-Jan-78','Son');
insert into DEPENDENT values ( '333445555', 'Alice' ,'F','05-Apr-76','Daughter');
insert into DEPENDENT values ( '333445555', 'Joy' ,'F','03-May-48','Spouce');
insert into DEPENDENT values ( '333445555', 'Theodore' ,'M','25-Oct-73','Son');
insert into DEPENDENT values ( '987654321', 'Abner' ,'M','29-Feb-32','Spouce');


--Let’s insert data into DEPT_LOCATION
insert into DEPT_LOCATION values ( '1', 'Houston');
insert into DEPT_LOCATION values ( '4', 'Stafford');
insert into DEPT_LOCATION values ( '5', 'Bellaire');
insert into DEPT_LOCATION values ( '5', 'Sugarland');
insert into DEPT_LOCATION values ( '5', 'Houston');


--Let’s insert data into PROJECT
insert into PROJECT values ( 'ProductX', '1', 'Bellaire', '5');
insert into PROJECT values ( 'ProductY', '2', 'Sugarland', '5');
insert into PROJECT values ( 'ProductZ', '3', 'Houston', '5');
insert into PROJECT values ( 'Computerization', '10', 'Stafford', '4');
insert into PROJECT values ( 'Reorganization', '20', 'Houston', '1');
insert into PROJECT values ( 'Newbenefits', '30', 'Stafford', '4');


--Let’s insert data into WORKS_ON
insert into WORKS_ON values ( '123456789', '1', '32.5');
insert into WORKS_ON values ( '123456789', '2', '7.5');
insert into WORKS_ON values ( '333445555', '2', '10');
insert into WORKS_ON values ( '333445555', '3', '10');
insert into WORKS_ON values ( '333445555', '10', '10');
insert into WORKS_ON values ( '333445555', '20', '10');
insert into WORKS_ON values ( '453453453', '1', '20');
insert into WORKS_ON values ( '453453453', '2', '20');
insert into WORKS_ON values ( '666884444', '3', '40');
insert into WORKS_ON values ( '888665555', '20', null);
insert into WORKS_ON values ( '987654321', '20', '15');
insert into WORKS_ON values ( '987654321', '30', '20');
insert into WORKS_ON values ( '987987987', '10', '35');
insert into WORKS_ON values ( '987987987', '30', '5');
insert into WORKS_ON values ( '999887777', '10', '10');
insert into WORKS_ON values ( '999887777', '30', '30');




-- Part 3

use [COMPANY1];
```

```
-- Part 3.1 data before update trigger
SELECT * FROM DEPARTMENT;
SELECT * FROM EMPLOYEE;
SELECT * FROM Audit_Dept_Table;
GO


-- Part 3.1 data after update trigger
UPDATE DEPARTMENT
SET Dnumber = 99
WHERE Dnumber = 4;
GO

SELECT * FROM DEPARTMENT;
SELECT * FROM EMPLOYEE;
SELECT * FROM Audit_Dept_Table;

--/*

-- Part 3.2 data before delete trigger

SELECT * FROM DEPARTMENT;
SELECT * FROM EMPLOYEE;
SELECT * FROM Audit_Dept_Table;
GO


-- Part 3.2 data after delete trigger
DELETE DEPARTMENT
WHERE Dnumber = 5;
GO

SELECT * FROM DEPARTMENT;
SELECT * FROM EMPLOYEE;
SELECT * FROM Audit_Dept_Table;

--*/
-- drop data base
USE master;
Drop DATABASE [COMPANY1];
```