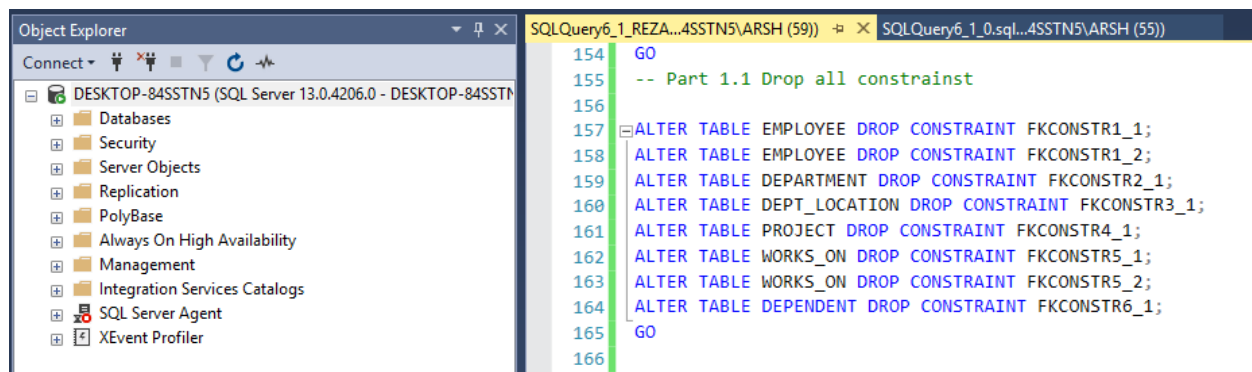


**1. Write Triggers that implement Referential Integrity for FK Dno of Employee that is named as Constraint EMPDEPTFK in Table Employee as specified in Figure 4.2 in the textbook in the following DDL statements below for Company database that you created for Lab2. (You may keep the all the new tuples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



-- Part 1.1 Drop all constraint

```

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**

```

166
167
168 -- Part 1.2 Creat trigger on update and delete events
169
170
171 CREATE TRIGGER EMPDEPTFK_DELETE ON DEPARTMENT
172 FOR DELETE AS
173 BEGIN
174     UPDATE EMPLOYEE SET EMPLOYEE.Dno=DEFAULT
175     FROM EMPLOYEE AS E
176     JOIN DELETED AS D ON D.Dnumber=E.Dno;
177 END;
178 GO
179
180 CREATE TRIGGER EMPDEPTFK_UPDATE ON DEPARTMENT
181 FOR UPDATE AS
182 BEGIN
183     DECLARE @new_Dnumber INT
184     SELECT @new_Dnumber = I.Dnumber FROM INSERTED I
185     UPDATE EMPLOYEE SET EMPLOYEE.Dno=@new_Dnumber
186     FROM EMPLOYEE AS E
187     JOIN DELETED AS D ON D.Dnumber=E.Dno;
188 END;
189 GO
  
```

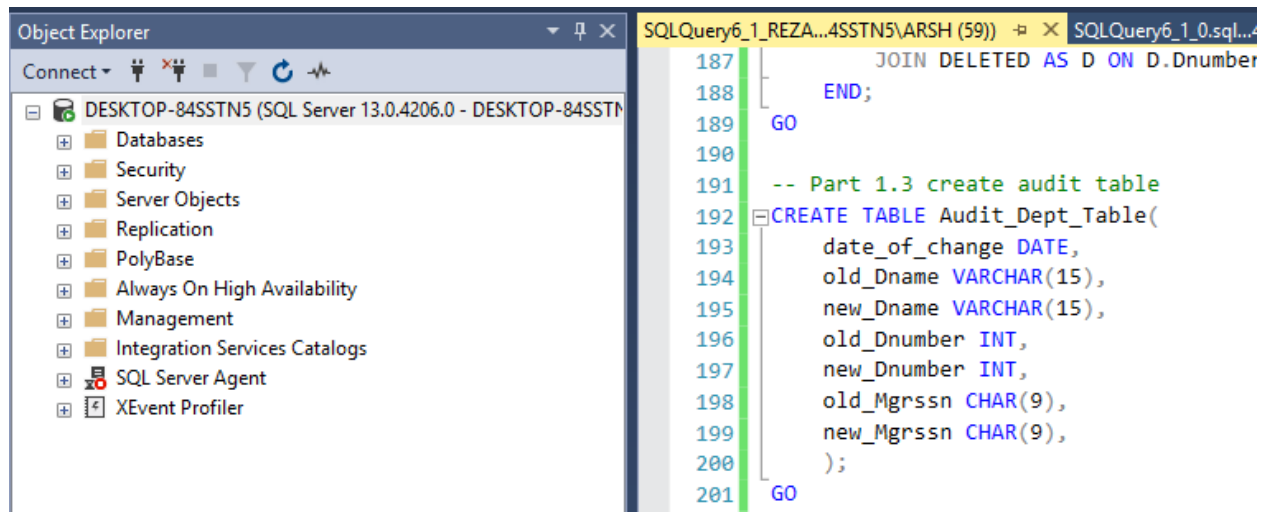
-- Part 1.2 Creat trigger on update and delete events

```

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
  
```

**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.**

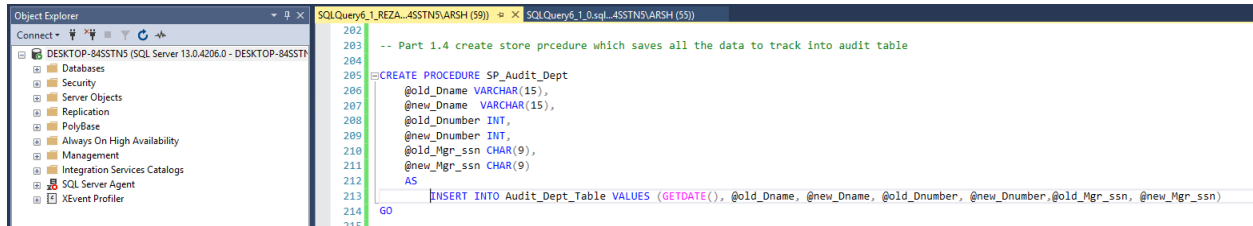


```

-- 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

```

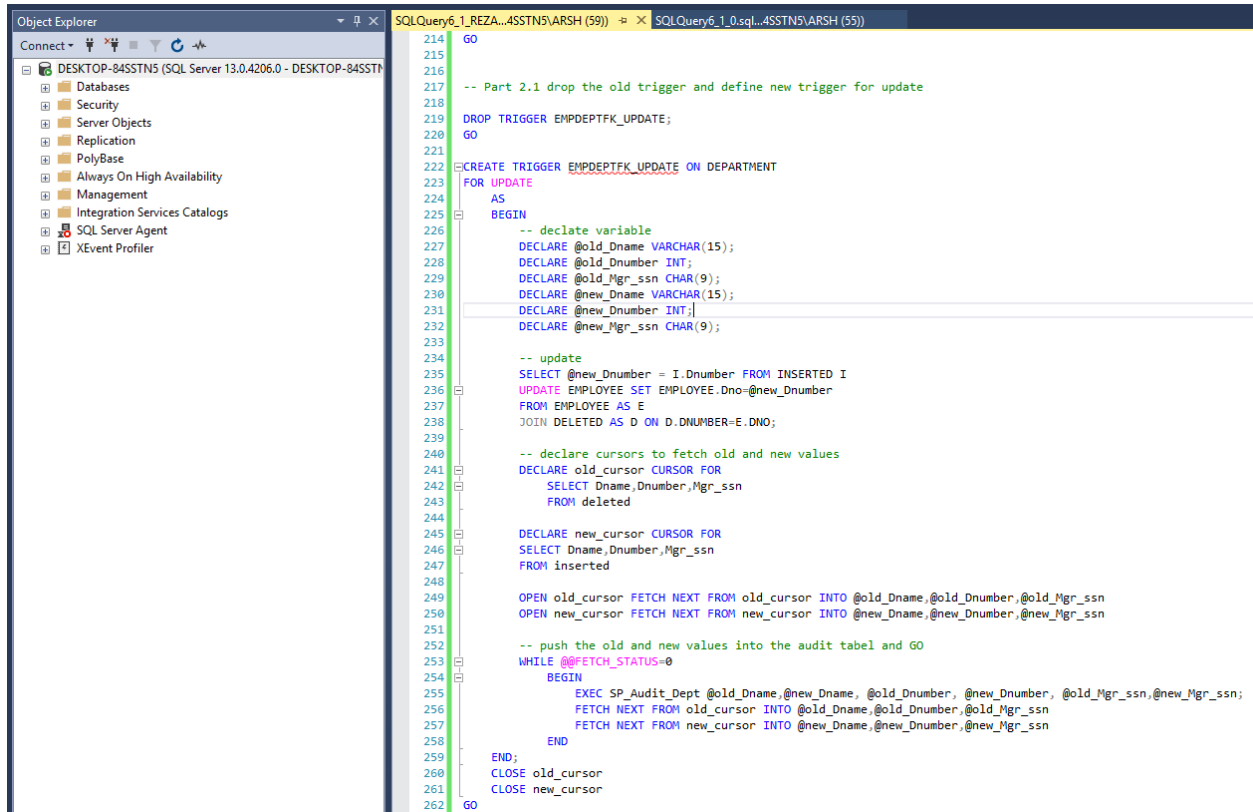
#### 4) Call the Stored procedure *SP\_Audit\_Dept* at the end of your Trigger to record all the history of the changes by the trigger.



```
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)



The screenshot shows the SQL Server Enterprise Manager interface. On the left, the Object Explorer displays the server structure for 'DESKTOP-84SSTN5'. The main window shows a SQL script for creating and testing a trigger named 'EMPDEPTFK\_UPDATE' on the 'DEPARTMENT' table. The script includes variable declarations, cursor declarations, and a loop to insert audit data into the 'Audit\_Dept\_Table'.

```

214 GO
215
216
217 -- Part 2.1 drop the old trigger and define new trigger for update
218
219 DROP TRIGGER EMPDEPTFK_UPDATE;
220 GO
221
222 CREATE TRIGGER EMPDEPTFK_UPDATE ON DEPARTMENT
223 FOR UPDATE
224 AS
225 BEGIN
226     -- declare variable
227     DECLARE @old_Dname VARCHAR(15);
228     DECLARE @old_Dnumber INT;
229     DECLARE @old_Mgr_ssn CHAR(9);
230     DECLARE @new_Dname VARCHAR(15);
231     DECLARE @new_Dnumber INT;
232     DECLARE @new_Mgr_ssn CHAR(9);
233
234     -- update
235     SELECT @new_Dnumber = I.Dnumber FROM INSERTED I
236     UPDATE EMPLOYEE SET EMPLOYEE.Dno=@new_Dnumber
237     FROM EMPLOYEE AS E
238     JOIN DELETED AS D ON D.DNUMBER=E.DNO;
239
240     -- declare cursors to fetch old and new values
241     DECLARE old_cursor CURSOR FOR
242     SELECT Dname,Dnumber,Mgr_ssn
243     FROM deleted
244
245     DECLARE new_cursor CURSOR FOR
246     SELECT Dname,Dnumber,Mgr_ssn
247     FROM inserted
248
249     OPEN old_cursor FETCH NEXT FROM old_cursor INTO @old_Dname,@old_Dnumber,@old_Mgr_ssn
250     OPEN new_cursor FETCH NEXT FROM new_cursor INTO @new_Dname,@new_Dnumber,@new_Mgr_ssn
251
252     -- push the old and new values into the audit tabel and GO
253     WHILE @@FETCH_STATUS=0
254     BEGIN
255         EXEC SP_Audit_Dept @old_Dname,@new_Dname, @old_Dnumber, @new_Dnumber, @old_Mgr_ssn,@new_Mgr_ssn;
256         FETCH NEXT FROM old_cursor INTO @old_Dname,@old_Dnumber,@old_Mgr_ssn
257         FETCH NEXT FROM new_cursor INTO @new_Dname,@new_Dnumber,@new_Mgr_ssn
258     END
259     END;
260     CLOSE old_cursor
261     CLOSE new_cursor
262 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*.

```

262 GO
263
264 -- Part 2.2 drop the old trigger and define new trigger for delete
265 DROP TRIGGER EMPDEPTFK_DELETE;
266 GO
267
268 CREATE TRIGGER EMPDEPTFK_DELETE ON DEPARTMENT
269 FOR DELETE AS
270 BEGIN
271     -- declare variable no need to declare new for deleting
272     DECLARE @old_Dname VARCHAR(15);
273     DECLARE @old_Dnumber INT;
274     DECLARE @old_Mgr_ssn CHAR(9);
275
276     -- update JOIN
277     UPDATE EMPLOYEE SET EMPLOYEE.Dno=DEFAULT
278     FROM EMPLOYEE AS E
279     JOIN DELETED AS D ON D.DNUMBER=E.DNO;
280
281     -- declare cursor to go and get the old (deleted) values
282     DECLARE dept_cursor CURSOR FOR
283     SELECT Dname,Dnumber,Mgr_ssn
284     FROM deleted
285
286     OPEN dept_cursor FETCH NEXT FROM dept_cursor INTO @old_Dname,@old_Dnumber,@old_Mgr_ssn
287
288     -- keep fetching and updating the audit table
289     WHILE @@FETCH_STATUS=0
290     BEGIN
291         EXEC SP_Audit_Dept @old_Dname,NULL, @old_Dnumber,NULL, @old_Mgr_ssn, NULL;
292         FETCH NEXT FROM dept_cursor INTO @old_Dname,@old_Dnumber,@old_Mgr_ssn
293     END
294 END;
295
296 CLOSE dept_cursor
297 GO
298
299

```

**1. Date before the following change:**

UPDATE Department Dnumber 4 to 99

The screenshot shows the SQL Server Enterprise Manager interface. On the left is the Object Explorer showing the server structure. The main window displays a SQL query window with the following code:

```

370 -- Part 3
371
372 use [COMPANY1];
373
374 -- Part 3.1 data before update trigger
375 SELECT * FROM DEPARTMENT;
376 SELECT * FROM EMPLOYEE;
377 SELECT * FROM Audit_Dept_Table;
378 GO
  
```

Below the query window, the Results pane shows a grid of data. The first table has 4 columns: Dname, Dnumber, Mgr\_ssn, and Mgr\_start\_date. The second table has 10 columns: Fname, Minit, Lname, Ssn, Bdate, Address, Sex, Salary, Super\_ssn, and Dno. The third table has 7 columns: date\_of\_change, old\_Dname, new\_Dname, old\_Dnumber, new\_Dnumber, old\_Mgrssn, and new\_Mgrssn.

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

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

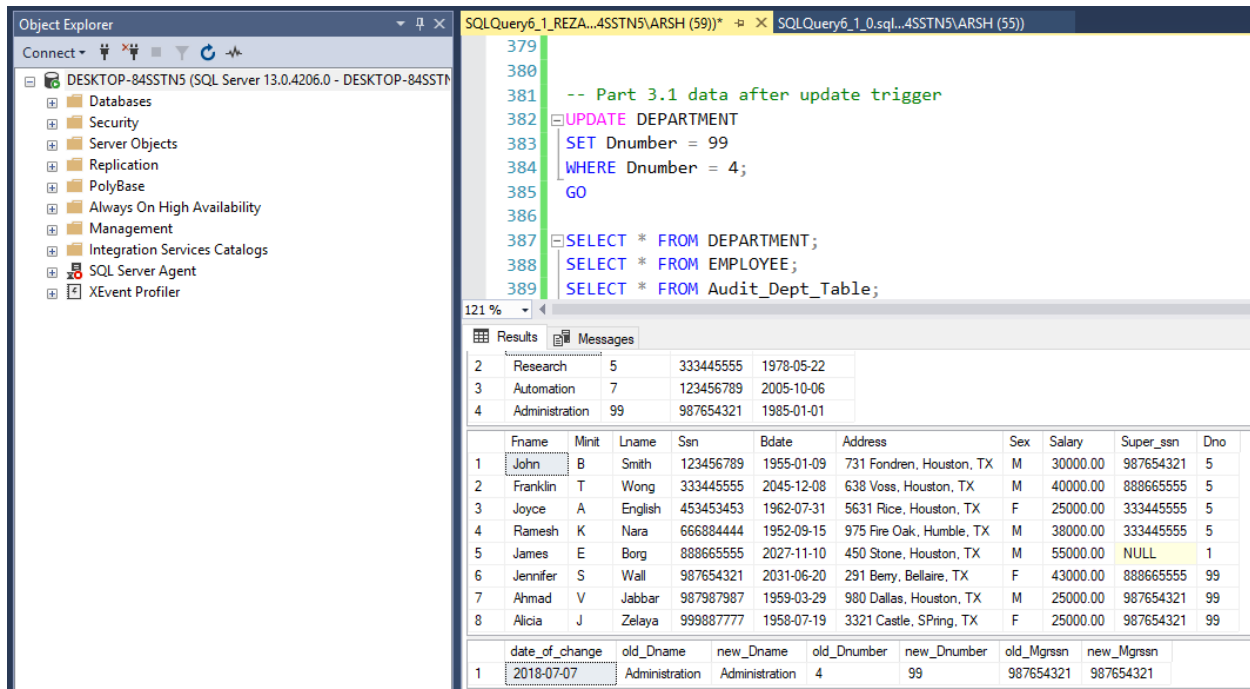
  

date_of_change	old_Dname	new_Dname	old_Dnumber	new_Dnumber	old_Mgrssn	new_Mgrssn
----------------	-----------	-----------	-------------	-------------	------------	------------

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



The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the Object Explorer with the 'DESKTOP-84SSTN5' server selected. The right pane shows a query window with the following SQL code:

```

379
380
381 -- Part 3.1 data after update trigger
382 UPDATE DEPARTMENT
383 SET Dnumber = 99
384 WHERE Dnumber = 4;
385 GO
386
387 SELECT * FROM DEPARTMENT;
388 SELECT * FROM EMPLOYEE;
389 SELECT * FROM Audit_Dept_Table;
  
```

Below the query window, the 'Results' tab shows the output of the query. The first table is the 'DEPARTMENT' table, which now has the following data:

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

The second table is the 'EMPLOYEE' table, which shows the updated department numbers for all employees:

Empid	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

The third table is the 'Audit\_Dept\_Table', which shows the change recorded in the audit table:

date_of_change	old_Dname	new_Dname	old_Dnumber	new_Dnumber	old_Mgrsname	new_Mgrsname
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

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the Object Explorer with the server 'DESKTOP-84SSTN5' selected. The right pane shows a query window with the following SQL code:

```

391 -- */
392
393 -- Part 3.2 data before delete trigger
394
395 SELECT * FROM DEPARTMENT;
396 SELECT * FROM EMPLOYEE;
397 SELECT * FROM Audit_Dept_Table;
398 GO
  
```

Below the query window, the 'Results' tab displays the output of the query. It contains two tables. The first table shows the state of the 'DEPARTMENT' table before the delete operation. The second table shows the state of the 'EMPLOYEE' table before the delete operation. The third table shows the state of the 'Audit\_Dept\_Table' after the delete operation.

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

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

The screenshot displays the SQL Server Enterprise Manager interface. The left pane shows the Object Explorer with the server 'DESKTOP-84SSTN5' selected. The right pane shows a SQL query window with the following code:

```

397 SELECT * FROM Audit_Dept_Table;
398 GO
399
400 -- Part 3.2 data after delete trigger
401 DELETE DEPARTMENT
402 WHERE Dnumber = 5;
403 GO
404
405 SELECT * FROM DEPARTMENT;
406 SELECT * FROM EMPLOYEE;
407 SELECT * FROM Audit_Dept_Table;
408
409 --*/
410 -- drop data base
411 USE master;
412 Drop DATABASE [COMPANY1];
413
414

```

Below the query window, the 'Results' pane shows two tables. The first table, 'DEPARTMENT', has columns Dname, Dnumber, Mgr\_ssn, and Mgr\_start\_date. The second table, 'EMPLOYEE', has columns Fname, Minit, Lname, Ssn, Bdate, Address, Sex, Salary, Super\_ssn, and Dno. The third table shows the 'date\_of\_change' table with columns date\_of\_change, old\_Dname, new\_Dname, old\_Dnumber, new\_Dnumber, old\_Mgrsn, and new\_Mgrsn.

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

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1955-01-09	731 Fondren, Houston, TX	M	30000.00	987654321	1
Franklin	T	Wong	333445555	2045-12-08	638 Voss, Houston, TX	M	40000.00	888665555	1
Joyce	A	Engli	453453453	1962-07-31	5631 Rice, Houston, TX	F	25000.00	333445555	1
Ramesh	K	Nara	666884444	1952-09-15	975 Fire Oak, Humble, TX	M	38000.00	333445555	1
James	E	Borg	888665555	2027-11-10	450 Stone, Houston, TX	M	55000.00	NULL	1
Jennifer	S	Wall	987654321	2031-06-20	291 Berry, Bellaire, TX	F	43000.00	888665555	99
Ahmad	V	Jabbar	987987987	1959-03-29	980 Dallas, Houston, TX	M	25000.00	987654321	99
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_Mgrsn	new_Mgrsn
2018-07-07	Administration	Administration	4	99	987654321	987654321
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.

## 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

-- dropping all dependencies and constraints
/*
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;

-- dropping 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
```

## CIS530– Lab Assignment 6: Trigger and Stored Procedure

Name: Reza Shisheie

ID: 2708062

```
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);
```

## CIS530– Lab Assignment 6: Trigger and Stored Procedure

Name: Reza Shisheie

ID: 2708062

```
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 constraints

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
```

## CIS530– Lab Assignment 6: Trigger and Stored Procedure

Name: Reza Shisheie

ID: 2708062

-- 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
```

## CIS530– Lab Assignment 6: Trigger and Stored Procedure

Name: Reza Shisheie

ID: 2708062

```
CREATE TRIGGER EMPDEPTFK_UPDATE ON DEPARTMENT
FOR UPDATE
AS
BEGIN
    -- declare 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);
```

# CIS530– Lab Assignment 6: Trigger and Stored Procedure

Name: Reza Shisheie

ID: 2708062

```

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');

```



## CIS530– Lab Assignment 6: Trigger and Stored Procedure

Name: Reza Shisheie

ID: 2708062

```
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];
```

## CIS530– Lab Assignment 6: Trigger and Stored Procedure

Name: Reza Shisheie

ID: 2708062

-- 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];
```