

**Name: LAVU MEGHA SRAVANI**

**CSU ID: 2762646**

**Description: Triggers and stored procedure in Microsoft SQL server**

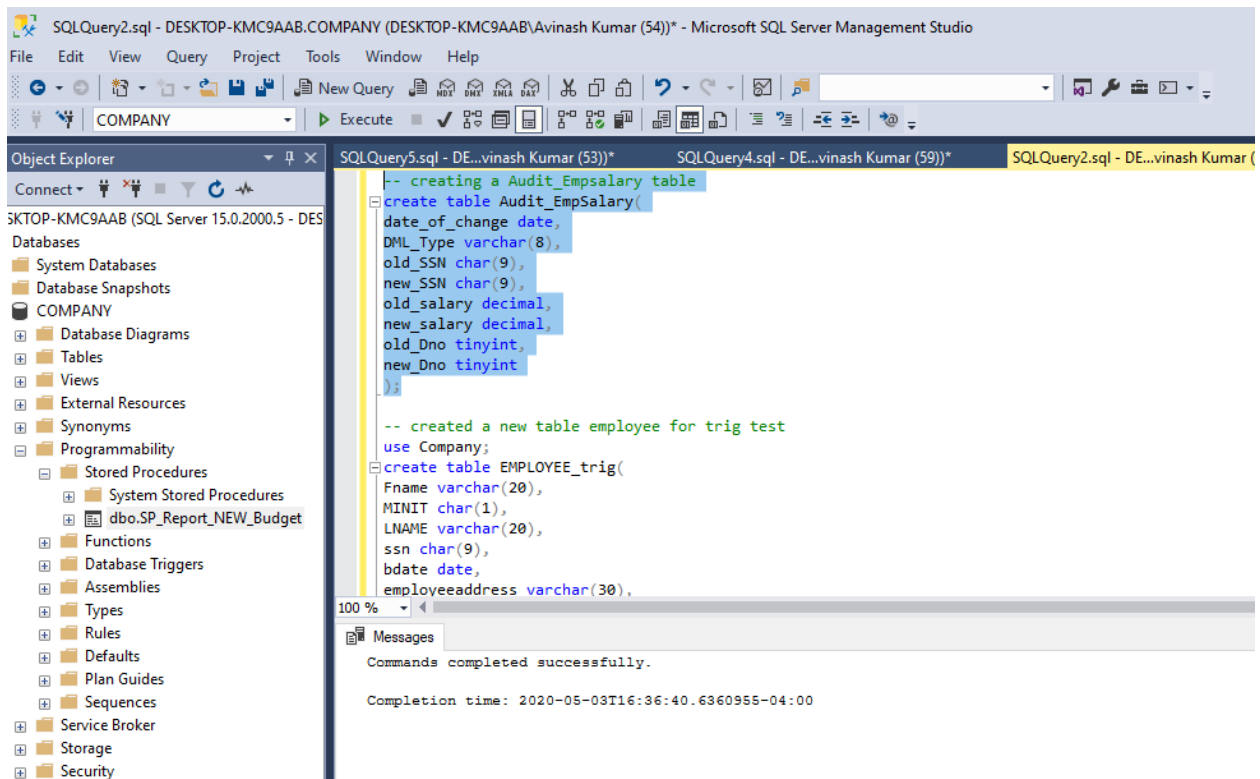
1)

-- created a new table employee for trigger test

```
use Company;
create table EMPLOYEE_trig(
Fname varchar(20),
MINIT char(1),
LNAME varchar(20),
ssn char(9),
bdate date,
employeeaddress varchar(30),
sex char(1),
employeesalary decimal,
superssn char(9),
DNO tinyint,
);
```

```
Insert into EMPLOYEE_trig values('John','B','Smith','123456789','09-Jan-55','731
Fondren,Houston,TX','M','30000','987654321','5');
Insert into EMPLOYEE_trig values('Franklin','T','Wong','333445555','08-Dec-45','638
Voss,Houston,TX','M','40000','888665555','5');
Insert into EMPLOYEE_trig values('Joyce','A','English','453453453','31-Jul-62','5631 Rice,
Houston,TX','F','25000','333445555','5');
Insert into EMPLOYEE_trig values('Ramesh','K','Narayan','666884444','15-Sep-52','975 Fire
Oak,Humble,TX','M','38000','333445555','5');
Insert into EMPLOYEE_trig values('James','E','Borg','888665555','10-Nov-27','450
Stone,Houston, TX','M','55000',NULL,'1');
Insert into EMPLOYEE_trig values('Jennifer','S','Wallace','987654321','20-Jun-31','291
Berry,Bellarie,TX','F','43000','888665555','4');
Insert into EMPLOYEE_trig values('Ahmad','V','Jabbar','987987987','29-Mar-59','980 Dallas,
Houston,TX','M','25000','987654321','4');
Insert into EMPLOYEE_trig values('Alicia','J','Zelaya','999887777','19-Jul-58','3321 Castle,
SPring,TX','F','25000','987654321','4');
```

```
-- creating a Audit_Empsalary table
create table Audit_EmpSalary(
date_of_change date,
DML_Type varchar(8),
old_SSN char(9),
new_SSN char(9),
old_salary decimal,
new_salary decimal,
old_Dno tinyint,
new_Dno tinyint
);
```



-- creation of stored procedure

create procedure SP\_Audit\_Empsalary

```
-- @date_of_change date,  
@DML_Type varchar(8),  
@old_SSN char(9),  
@new_SSN char(9),  
@old_salary decimal,  
@new_salary decimal,  
@old_Dno tinyint,  
@new_Dno tinyint
```

AS

Begin

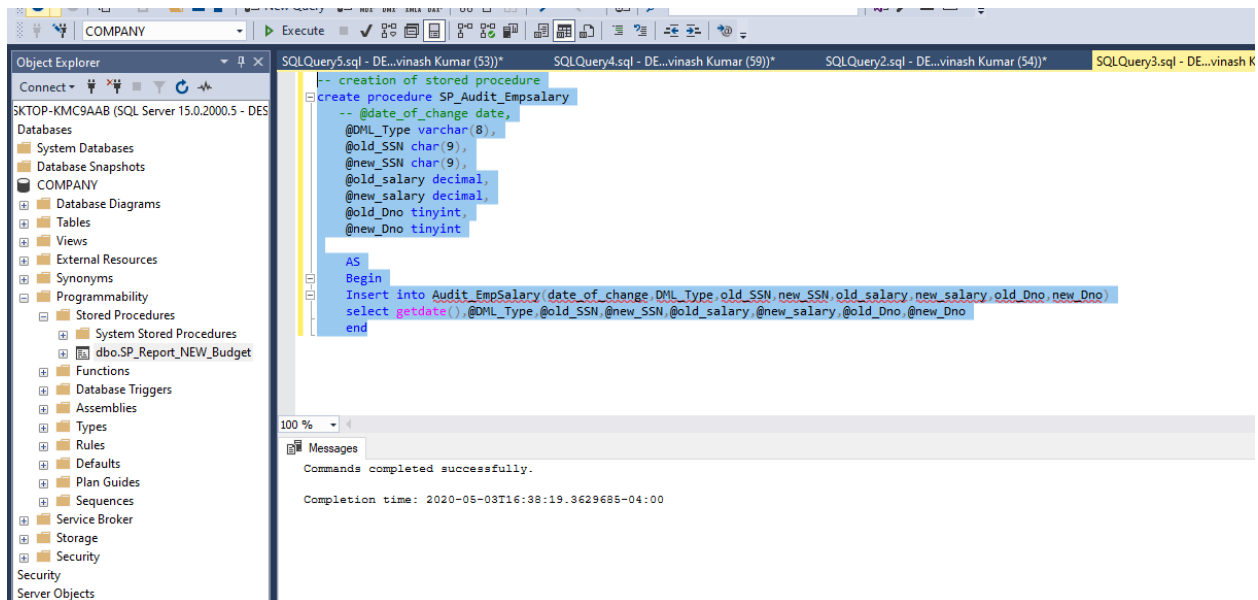
Insert into

Audit\_EmpSalary(date\_of\_change,DML\_Type,old\_SSN,new\_SSN,old\_salary,new\_salary,old\_Dno,new\_Dno)

select

getdate(),@DML\_Type,@old\_SSN,@new\_SSN,@old\_salary,@new\_salary,@old\_Dno,@new\_Dno

end



```
Create Trigger Trig_Update_Audit_empsalary
on EMPLOYEE_trig For Update
AS
begin
Declare @DML_Type varchar(8);
  Declare @old_SSN char(9);
  Declare @new_SSN char(9);
  Declare @old_salary decimal;
  Declare @new_salary decimal;
  Declare @old_Dno tinyint;
  Declare @new_Dno tinyint;

if update(employeeesalary)
begin

  --select @old_salary= d.employeeesalary from deleted d;
  --select @new_salary= i.employeeesalary from inserted i;
  Declare cursorid cursor for select d.ssn,d.employeeesalary,d.DNO from deleted d
  Declare cursorid1 cursor for select i.ssn,i.employeeesalary,i.DNO from inserted i
  open cursorid
  fetch next from cursorid into @old_SSN,@old_salary,@old_Dno
  open cursorid1
  fetch next from cursorid1 into @new_SSN,@new_salary,@new_Dno

  while @@fetch_status = 0
  begin
    set @DML_type ='Update';

    EXEC SP_Audit_Empsalary
    @DML_type,@old_SSN,@new_SSN,@old_salary,@new_salary,@old_Dno,@new_Dno
    fetch next from cursorid into @old_SSN,@old_salary,@old_Dno
    fetch next from cursorid1 into @new_SSN,@new_salary,@new_Dno

  end
  close cursorid1
  close cursorid
end
end
```

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the Object Explorer shows the database structure for 'COMPANY', including tables, views, and triggers. The main window shows a SQL query editor with the following script:

```
Create Trigger Trig_Update_Audit_empSalary
on EMPLOYEE_trig For Update
AS
begin
    Declare @DML_Type varchar(8);
    Declare @old_SSN char(9);
    Declare @new_SSN char(9);
    Declare @old_salary decimal;
    Declare @new_salary decimal;
    Declare @old_Dno tinyint;
    Declare @new_Dno tinyint;

    if update(employeeSalary)
    begin
        --select @old_salary= d.employeeSalary from deleted d;
        --select @new_salary= i.employeeSalary from inserted i;
        Declare cursorid cursor for select d.ssn,d.employeeSalary,d.DNO from deleted d
        Declare cursorid1 cursor for select i.ssn,i.employeeSalary,i.DNO from inserted i
        open cursorid
        fetch next from cursorid into @old_SSN,@old_salary,@old_Dno
```

Below the script, the Messages pane shows the execution results:

```
Commands completed successfully.
Completion time: 2020-05-03T16:44:32.1194197-04:00
```

```
-- before update
select * from EMPLOYEE_trig;

select * from Audit_EmpSalary;
```

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the Object Explorer shows the database structure for 'COMPANY'. The main window shows a SQL script in the 'SQLQuery5.sql' tab. The script includes a trigger definition, a 'before update' trigger body, and a series of SQL statements to update the trigger and the audit table. Below the script, the 'Results' pane shows the output of the 'select \* from EMPLOYEE\_trig;' statement, displaying a table with 10 columns: Fname, MINIT, LNAME, ssn, bdate, employeeaddress, sex, employeesalary, superssn, and DNO. The results show 8 rows of employee data. Below the results, there is a table with 8 columns: date\_of\_change, DML\_Type, old\_SSN, new\_SSN, old\_salary, new\_salary, old\_Dno, and new\_Dno.

```
close cursorid1
close cursorid
end
end

-- before update
select * from EMPLOYEE_trig;
select * from Audit_EmpSalary;

Update EMPLOYEE_trig set employeesalary= '99000' where DNO=5;
Update EMPLOYEE_trig set DNO=1;

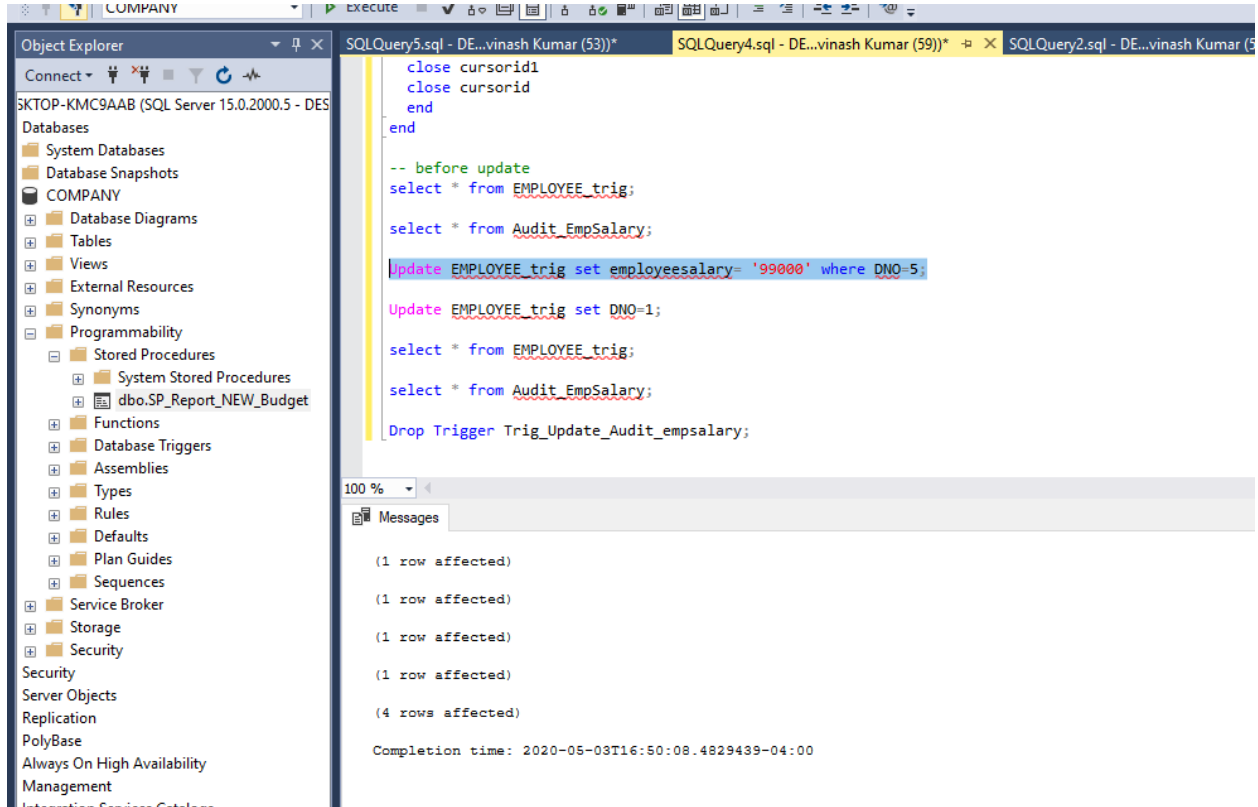
select * from EMPLOYEE_trig;
select * from Audit_EmpSalary;

Drop Trigger Trig_Update_Audit_empSalary;
```

	Fname	MINIT	LNAME	ssn	bdate	employeeaddress	sex	employeesalary	superssn	DNO
1	John	B	Smith	123456789	1955-01-09	731 Fondren, Houston, TX	M	30000	987654321	5
2	Franklin	T	Wong	333445555	2045-12-08	638 Voss, Houston, TX	M	40000	888665555	5
3	Joyce	A	English	453453453	1962-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
4	Ramesh	K	Narayan	666884444	1952-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
5	James	E	Borg	888665555	2027-11-10	450 Stone, Houston, TX	M	55000	NULL	1
6	Jennifer	S	Wallace	987654321	2031-06-20	291 Berry, Bellarie, TX	F	43000	888665555	4
7	Ahmad	V	Jabbar	987987987	1959-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
8	Alicia	J	Zelaya	999887777	1958-07-19	3321 Castle, SPring, TX	F	25000	987654321	4

date_of_change	DML_Type	old_SSN	new_SSN	old_salary	new_salary	old_Dno	new_Dno
----------------	----------	---------	---------	------------	------------	---------	---------

Update EMPLOYEE\_trig set employeesalary= '99000' where DNO=5;



-- After Update

select \* from EMPLOYEE\_trig;

select \* from Audit\_EmpSalary;

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the Object Explorer displays the database structure for 'COMPANY'. The main window shows a SQL script in the 'SQLQuery4.sql' file. The script includes comments for 'Before Update' and 'After Update' events, and a 'Drop Trigger' statement. Below the script, the 'Results' tab shows the output of the 'select \* from EMPLOYEE\_trig;' query, displaying a list of employees with their details. The 'Messages' tab shows the output of the 'select \* from Audit\_EmpSalary;' query, displaying a list of audit records.

```

close cursorid1
close cursorid
end
end

-- before update
select * from EMPLOYEE_trig;

select * from Audit_EmpSalary;

Update EMPLOYEE_trig set employeesalary= '99000' where DNO=5;

Update EMPLOYEE_trig set DNO=1;

-- After Update
select * from EMPLOYEE_trig;

select * from Audit_EmpSalary;

Drop Trigger Trig_Update_Audit_empsalary;

```

	Fname	MINIT	LNAME	ssn	bdate	employeeaddress	sex	employeesalary	superssn	DNO
1	John	B	Smith	123456789	1955-01-09	731 Fondren,Houston,TX	M	99000	987654321	5
2	Franklin	T	Wong	333445555	2045-12-08	638 Voss,Houston,TX	M	99000	888665555	5
3	Joyce	A	English	453453453	1962-07-31	5631 Rice, Houston,TX	F	99000	333445555	5
4	Ramesh	K	Narayan	666884444	1952-09-15	975 Fire Oak,Humble,TX	M	99000	333445555	5
5	James	E	Borg	888665555	2027-11-10	450 Stone,Houston, TX	M	55000	NULL	1
6	Jennifer	S	Wallace	987654321	2031-06-20	291 Berry,Bellarie,TX	F	43000	888665555	4
7	Ahmad	V	Jabbar	987987987	1959-03-29	980 Dallas, Houston,TX	M	25000	987654321	4
8	Alicia	J	Zelaya	999887777	1958-07-19	3321 Castle, SPring,TX	F	25000	987654321	4

	date_of_change	DML_Type	old_SSN	new_SSN	old_salary	new_salary	old_Dno	new_Dno
1	2020-05-03	Update	666884444	666884444	38000	99000	5	5
2	2020-05-03	Update	453453453	453453453	25000	99000	5	5
3	2020-05-03	Update	333445555	333445555	40000	99000	5	5
4	2020-05-03	Update	123456789	123456789	30000	99000	5	5

Create Trigger Trig\_delete\_Audit\_empsalary  
on EMPLOYEE\_trig For delete



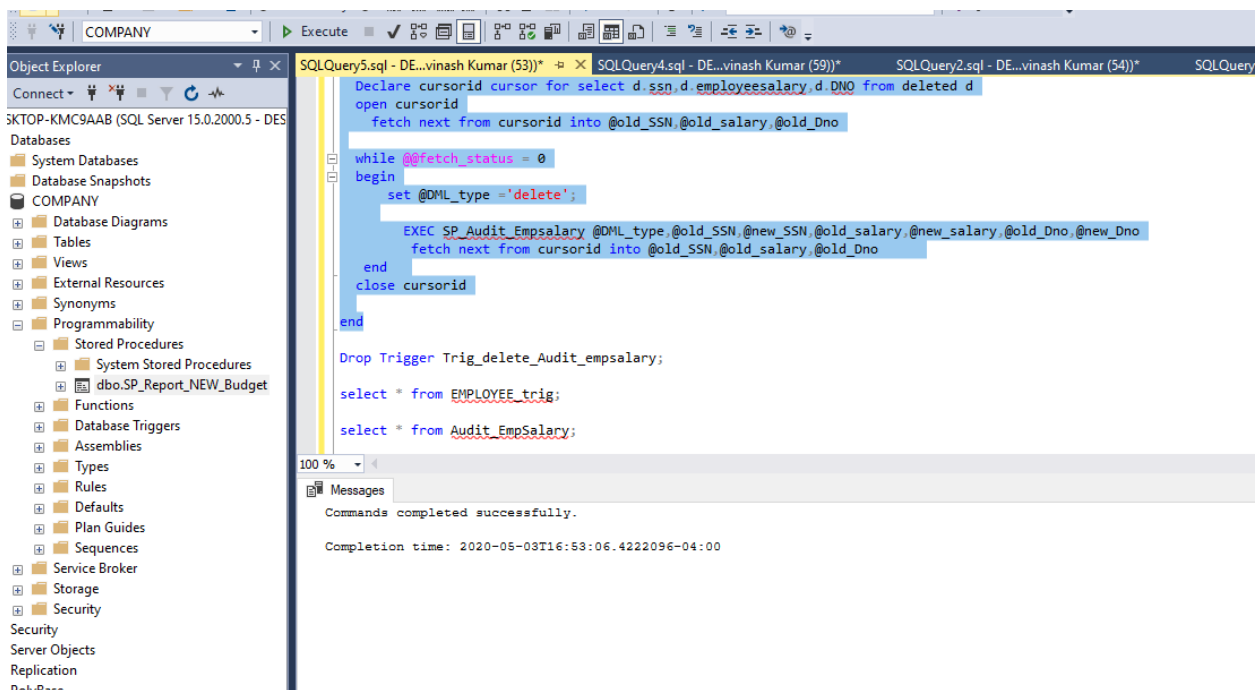
```

AS
begin
    Declare @DML_Type varchar(8);
    Declare @old_SSN char(9);
    Declare @new_SSN char(9);
    Declare @old_salary decimal;
    Declare @new_salary decimal;
    Declare @old_Dno tinyint;
    Declare @new_Dno tinyint;
    Declare cursorid cursor for select d.ssn,d.employeesalary,d.DNO from deleted d
    open cursorid
    fetch next from cursorid into @old_SSN,@old_salary,@old_Dno
    while @@fetch_status = 0
    begin
        set @DML_type ='delete';

        EXEC SP_Audit_Empsalary
        @DML_type,@old_SSN,@new_SSN,@old_salary,@new_salary,@old_Dno,@new_Dno
        fetch next from cursorid into @old_SSN,@old_salary,@old_Dno

    end
    close cursorid
end

```



```

-- before delete
select * from EMPLOYEE_trig;

```

`select * from Audit_EmpSalary;`

The screenshot shows the SQL Server Enterprise Manager interface. On the left is the Object Explorer showing the database structure. The main window displays a SQL query script for dropping a trigger and performing a delete operation with before and after triggers. Below the script, the 'Results' tab shows a grid of employee data.

**SQL Query Script:**

```
Drop Trigger Trig_delete_Audit_empSalary;

-- before delete
select * from EMPLOYEE_trig;

select * from Audit_EmpSalary;

Delete from EMPLOYEE_trig where FName='John' AND LNAME='Smith';

-- after delete
select * from EMPLOYEE_trig;

select * from Audit_EmpSalary;
```

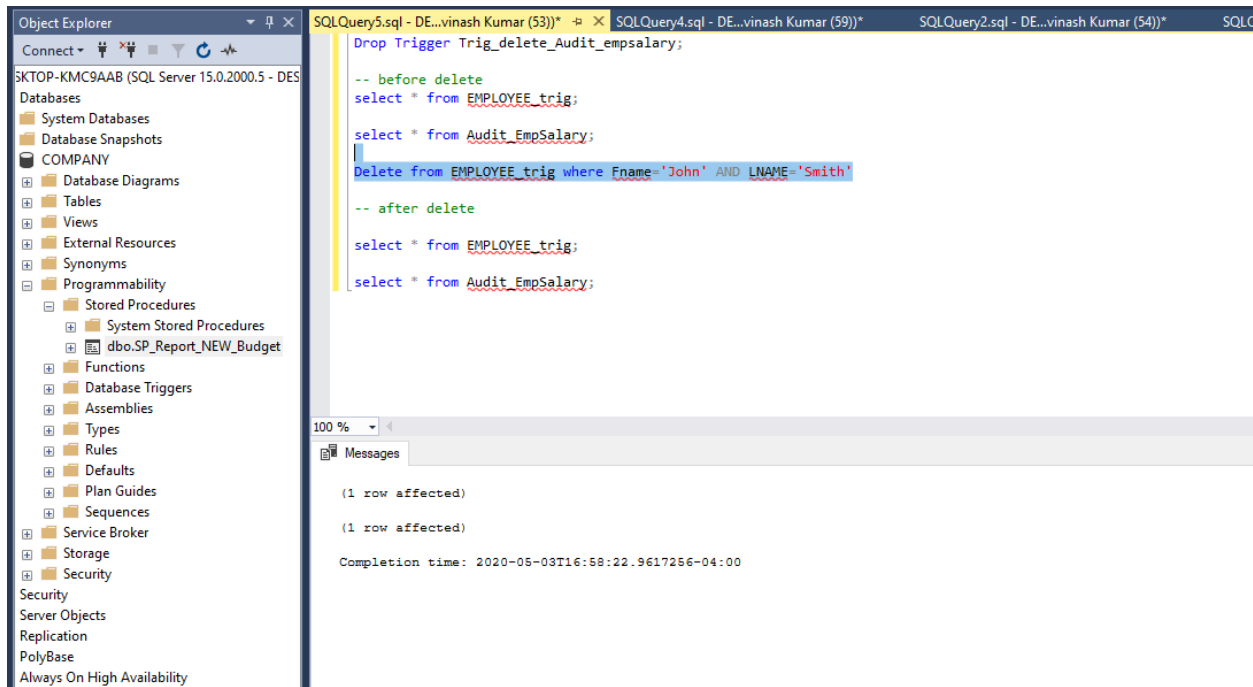
**Results Grid:**

	Fname	MINIT	LNAME	ssn	bdate	employeeaddress	sex	employeesalary	superssn	DNO
1	John	B	Smith	123456789	1955-01-09	731 Fondren,Houston,TX	M	99000	987654321	5
2	Franklin	T	Wong	333445555	2045-12-08	638 Voss,Houston,TX	M	99000	888665555	5
3	Joyce	A	English	453453453	1962-07-31	5631 Rice, Houston,TX	F	99000	333445555	5
4	Ramesh	K	Narayan	666884444	1952-09-15	975 Fire Oak,Humble,TX	M	99000	333445555	5
5	James	E	Borg	888665555	2027-11-10	450 Stone,Houston, TX	M	55000	NULL	1
6	Jennifer	S	Wallace	987654321	2031-06-20	291 Berry,Bellarie,TX	F	43000	888665555	4
7	Ahmad	V	Jabbar	987987987	1959-03-29	980 Dallas, Houston,TX	M	25000	987654321	4
8	Alicia	J	Zelaya	999887777	1958-07-19	3321 Castle, SPring,TX	F	25000	987654321	4

	date_of_change	DML_Type	old_SSN	new_SSN	old_salary	new_salary	old_Dno	new_Dno
1	2020-05-03	Update	666884444	666884444	38000	99000	5	5
2	2020-05-03	Update	453453453	453453453	25000	99000	5	5
3	2020-05-03	Update	333445555	333445555	40000	99000	5	5
4	2020-05-03	Update	123456789	123456789	30000	99000	5	5

Delete from EMPLOYEE\_trig where Fname='John' AND LNAME='Smith'



The screenshot displays the SQL Server Enterprise Manager interface. On the left, the Object Explorer shows the database structure for 'SKTOP-KMC9AAB (SQL Server 15.0.2000.5 - DES)'. The main window shows a query script in 'SQLQuery5.sql'. The script includes a 'Drop Trigger' statement, followed by 'select \* from EMPLOYEE\_trig;' and 'select \* from Audit\_EmpSalary;' before the delete operation. The delete statement is 'Delete from EMPLOYEE\_trig where Fname='John' AND LNAME='Smith''. After the delete, there are more 'select \* from EMPLOYEE\_trig;' and 'select \* from Audit\_EmpSalary;' statements. The execution results pane at the bottom shows two messages: '(1 row affected)' and '(1 row affected)', followed by the completion time '2020-05-03T16:58:22.9617256-04:00'.

```
Drop Trigger Trig_delete_Audit_empSalary;

-- before delete
select * from EMPLOYEE_trig;

select * from Audit_EmpSalary;

Delete from EMPLOYEE_trig where Fname='John' AND LNAME='Smith'

-- after delete

select * from EMPLOYEE_trig;

select * from Audit_EmpSalary;
```

100 %

Messages

(1 row affected)

(1 row affected)

Completion time: 2020-05-03T16:58:22.9617256-04:00

-- after delete

select \* from EMPLOYEE\_trig;

select \* from Audit\_EmpSalary;

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the Object Explorer displays the database structure for 'SKTOP-KMC9AAB (SQL Server 15.0.2000.5 - DES)'. The main window shows a SQL script for 'SQLQuery5.sql - DE...vinash Kumar (53)\*'. The script includes a trigger definition and execution steps.

```

Drop Trigger Trig_delete_Audit_empSalary;

-- before delete
select * from EMPLOYEE_trig;

select * from Audit_EmpSalary;

Delete from EMPLOYEE_trig where Fname='John' AND LNAME='Smith'

-- after delete
select * from EMPLOYEE_trig;

select * from Audit_EmpSalary;

```

Below the script, the 'Results' tab shows the output of the queries. The first table lists employee details, and the second table shows the audit log.

	Fname	MINIT	LNAME	ssn	bdate	employeeaddress	sex	employeesalary	superssn	DNO
1	Franklin	T	Wong	333445555	2045-12-08	638 Voss,Houston,TX	M	99000	888665555	5
2	Joyce	A	English	453453453	1962-07-31	5631 Rice, Houston,TX	F	99000	333445555	5
3	Ramesh	K	Narayan	666884444	1952-09-15	975 Fire Oak,Humble,TX	M	99000	333445555	5
4	James	E	Borg	888665555	2027-11-10	450 Stone,Houston, TX	M	55000	NULL	1
5	Jennifer	S	Wallace	987654321	2031-06-20	291 Berry,Bellare,TX	F	43000	888665555	4
6	Ahmad	V	Jabbar	987987987	1959-03-29	980 Dallas, Houston,TX	M	25000	987654321	4
7	Alicia	J	Zelaya	999887777	1958-07-19	3321 Castle, SPring,TX	F	25000	987654321	4

	date_of_change	DML_Type	old_SSN	new_SSN	old_salary	new_salary	old_Dno	new_Dno
1	2020-05-03	Update	666884444	666884444	38000	99000	5	5
2	2020-05-03	Update	453453453	453453453	25000	99000	5	5
3	2020-05-03	Update	333445555	333445555	40000	99000	5	5
4	2020-05-03	Update	123456789	123456789	30000	99000	5	5
5	2020-05-03	delete	123456789	NULL	99000	NULL	5	NULL

```

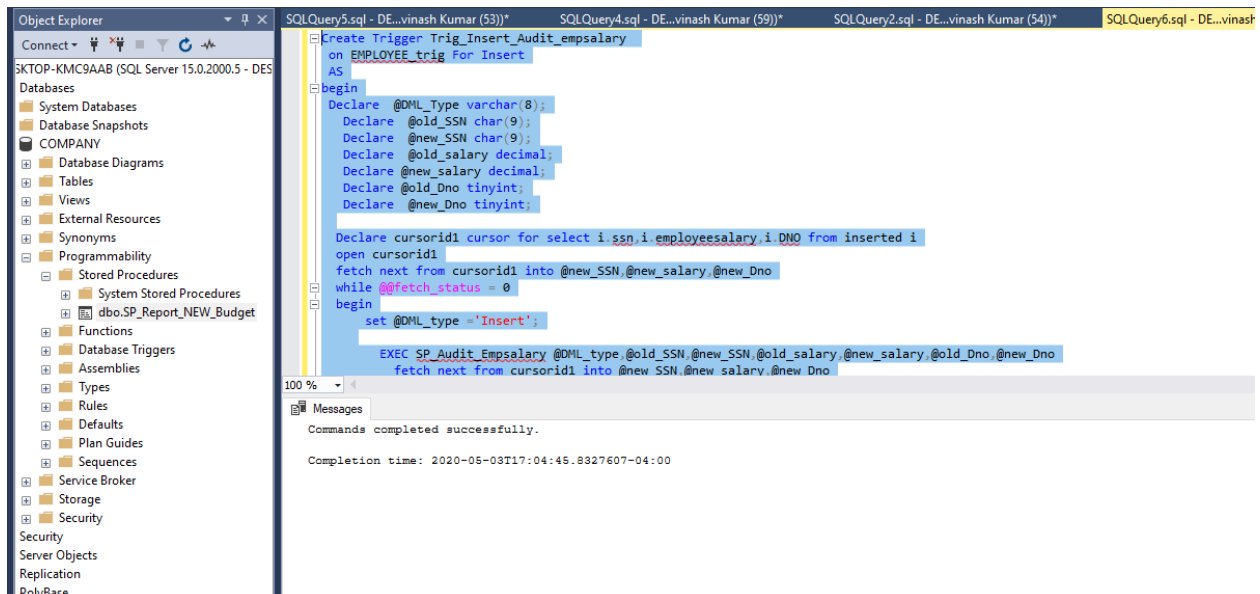
Create Trigger Trig_Insert_Audit_empsalary
on EMPLOYEE_trig For Insert
AS
begin
Declare @DML_Type varchar(8);
  Declare @old_SSN char(9);
  Declare @new_SSN char(9);
  Declare @old_salary decimal;
  Declare @new_salary decimal;
  Declare @old_Dno tinyint;
  Declare @new_Dno tinyint;

  Declare cursorid1 cursor for select i.ssn,i.employeesalary,i.DNO from inserted i
  open cursorid1
  fetch next from cursorid1 into @new_SSN,@new_salary,@new_Dno
  while @@fetch_status = 0
  begin
    set @DML_type ='Insert';

    EXEC SP_Audit_Empsalary
    @DML_type,@old_SSN,@new_SSN,@old_salary,@new_salary,@old_Dno,@new_Dno
    fetch next from cursorid1 into @new_SSN,@new_salary,@new_Dno

  end
  close cursorid1
end

```



-- before insert

`select * from EMPLOYEE_trig;`

`select * from Audit_EmpSalary;`

100 %

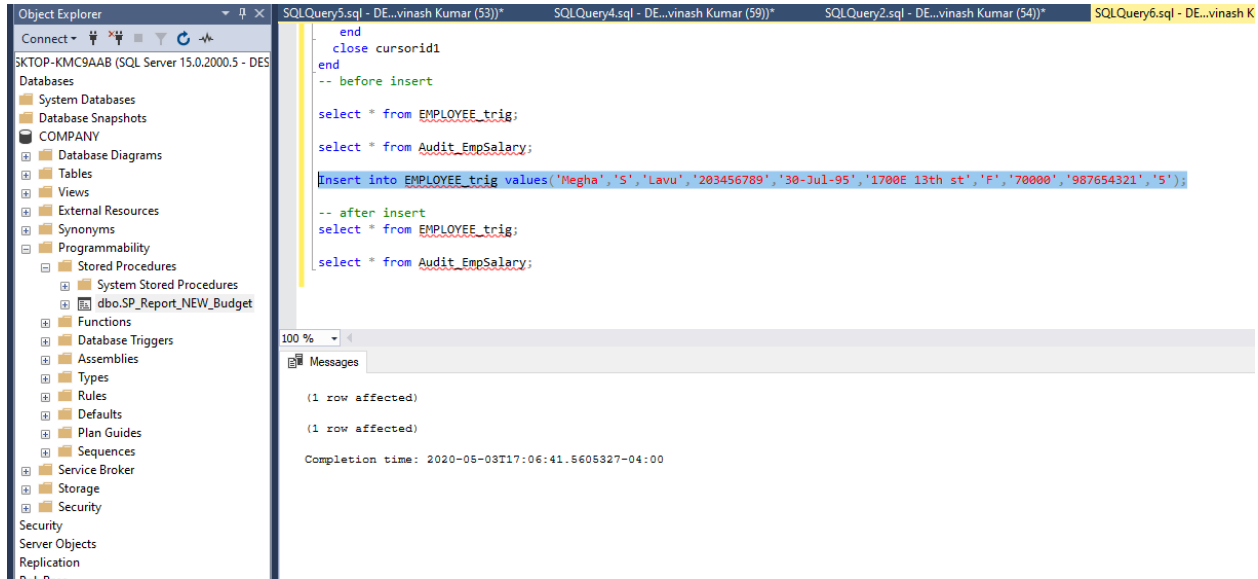
Results Messages

	Fname	MINIT	LNAME	ssn	bdate	employeeaddress	sex	employeesalary	superssn	DNO
1	Franklin	T	Wong	333445555	2045-12-08	638 Voss,Houston,TX	M	99000	888665555	5
2	Joyce	A	English	453453453	1962-07-31	5631 Rice, Houston,TX	F	99000	333445555	5
3	Ramesh	K	Narayan	666884444	1952-09-15	975 Fire Oak,Humble,TX	M	99000	333445555	5
4	James	E	Borg	888665555	2027-11-10	450 Stone,Houston, TX	M	55000	NULL	1
5	Jennifer	S	Wallace	987654321	2031-06-20	291 Berry,Bellarie,TX	F	43000	888665555	4
6	Ahmad	V	Jabbar	987987987	1959-03-29	980 Dallas, Houston,TX	M	25000	987654321	4
7	Alicia	J	Zelaya	999887777	1958-07-19	3321 Castle, SPring,TX	F	25000	987654321	4

	date_of_change	DML_Type	old_SSN	new_SSN	old_salary	new_salary	old_Dno	new_Dno
1	2020-05-03	Update	666884444	666884444	99000	99000	5	5
2	2020-05-03	Update	453453453	453453453	99000	99000	5	5
3	2020-05-03	Update	333445555	333445555	99000	99000	5	5
4	2020-05-03	Update	123456789	123456789	99000	99000	5	5
5	2020-05-03	delete	123456789	NULL	99000	NULL	5	NULL

Insert into EMPLOYEE\_trig values('Megha','S','Lavu','203456789','30-Jul-95','1700E 13th st','F','70000','987654321','5');



The screenshot displays the SQL Server Enterprise Manager interface. On the left, the Object Explorer shows the database structure for 'SKTOP-KMC9AAB (SQL Server 15.0.2000.5 - DES)'. The right pane shows a SQL query script in 'SQLQuery6.sql - DE...vinash K'. The script includes a cursor, a 'before insert' trigger that selects from 'EMPLOYEE\_trig' and 'Audit\_EmpSalary', an insert statement, and an 'after insert' trigger that also selects from the same tables. The execution results pane at the bottom shows two messages: '(1 row affected)' and '(1 row affected)', with a completion time of '2020-05-03T17:06:41.5605327-04:00'.

```
end
close cursorid1
end
-- before insert
select * from EMPLOYEE_trig;
select * from Audit_EmpSalary;
insert into EMPLOYEE_trig values('Megha','S','Lavu','203456789','30-Jul-95','1700E 13th st','F','70000','987654321','5');
-- after insert
select * from EMPLOYEE_trig;
select * from Audit_EmpSalary;
```

100 %  
Messages  
(1 row affected)  
(1 row affected)  
Completion time: 2020-05-03T17:06:41.5605327-04:00

```
-- after insert
select * from EMPLOYEE_trig;

select * from Audit_EmpSalary;
```

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'Object Explorer' with the 'COMPANY' database selected. The right pane shows a query window with the following SQL code:

```
select * from Audit_EmpSalary;

Insert into EMPLOYEE_trig values('Megha','S','Lavu','203456789','30-Jul-95','1700E 13th st','F','70000','987654321','5');

-- after insert
select * from EMPLOYEE_trig;

select * from Audit_EmpSalary;
```

Below the query window, the 'Results' tab is active, displaying a table with 10 columns: Fname, MINIT, LNAME, ssn, bdate, employeeaddress, sex, employeesalary, superssn, and DNO. The table contains 8 rows of employee data.

	Fname	MINIT	LNAME	ssn	bdate	employeeaddress	sex	employeesalary	superssn	DNO
1	Megha	S	Lavu	203456789	1995-07-30	1700E 13th st	F	70000	987654321	5
2	Franklin	T	Wong	333445555	2045-12-08	638 Voss,Houston,TX	M	99000	888665555	5
3	Joyce	A	English	453453453	1962-07-31	5631 Rice, Houston,TX	F	99000	333445555	5
4	Ramesh	K	Narayan	666884444	1952-09-15	975 Fire Oak,Humble,TX	M	99000	333445555	5
5	James	E	Borg	888665555	2027-11-10	450 Stone,Houston, TX	M	55000	NULL	1
6	Jennifer	S	Wallace	987654321	2031-06-20	291 Berry,Bellarie,TX	F	43000	888665555	4
7	Ahmad	V	Jabbar	987987987	1959-03-29	980 Dallas, Houston,TX	M	25000	987654321	4
8	Alicia	J	Zelaya	999887777	1958-07-19	3321 Castle, SPring,TX	F	25000	987654321	4

Below the employee data table, there is another table showing audit results with 8 columns: date\_of\_change, DML\_Type, old\_SSN, new\_SSN, old\_salary, new\_salary, old\_Dno, and new\_Dno. It contains 6 rows of audit data.

	date_of_change	DML_Type	old_SSN	new_SSN	old_salary	new_salary	old_Dno	new_Dno
1	2020-05-03	Update	666884444	666884444	99000	99000	5	5
2	2020-05-03	Update	453453453	453453453	99000	99000	5	5
3	2020-05-03	Update	333445555	333445555	99000	99000	5	5
4	2020-05-03	Update	123456789	123456789	99000	99000	5	5
5	2020-05-03	delete	123456789	NULL	99000	NULL	5	NULL
6	2020-05-03	Insert	NULL	203456789	NULL	70000	NULL	5