

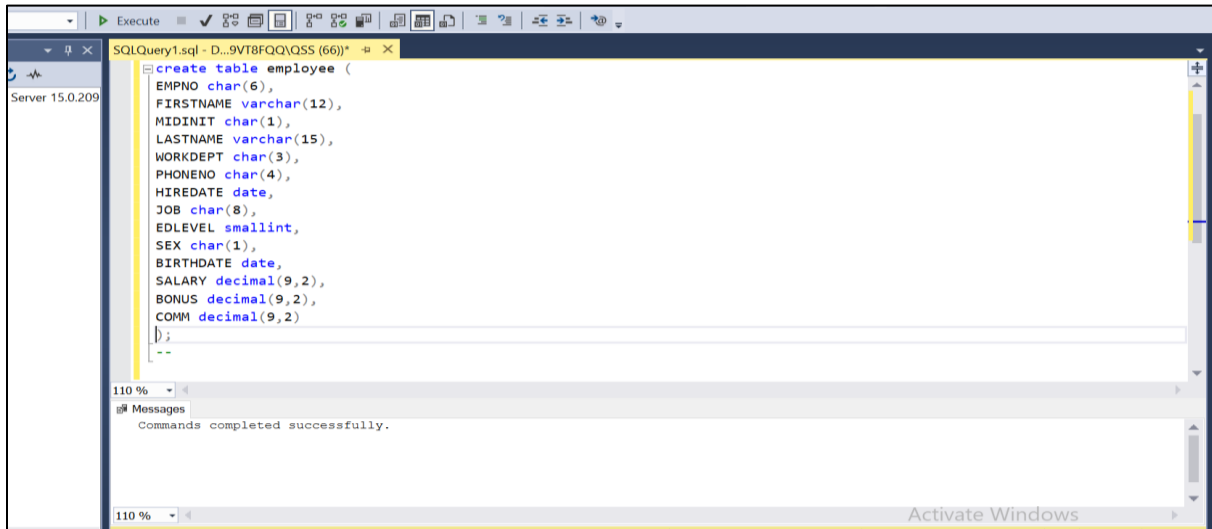
FINAL PROJECT

Name- Amrit Pal Singh

Student ID- 110355211

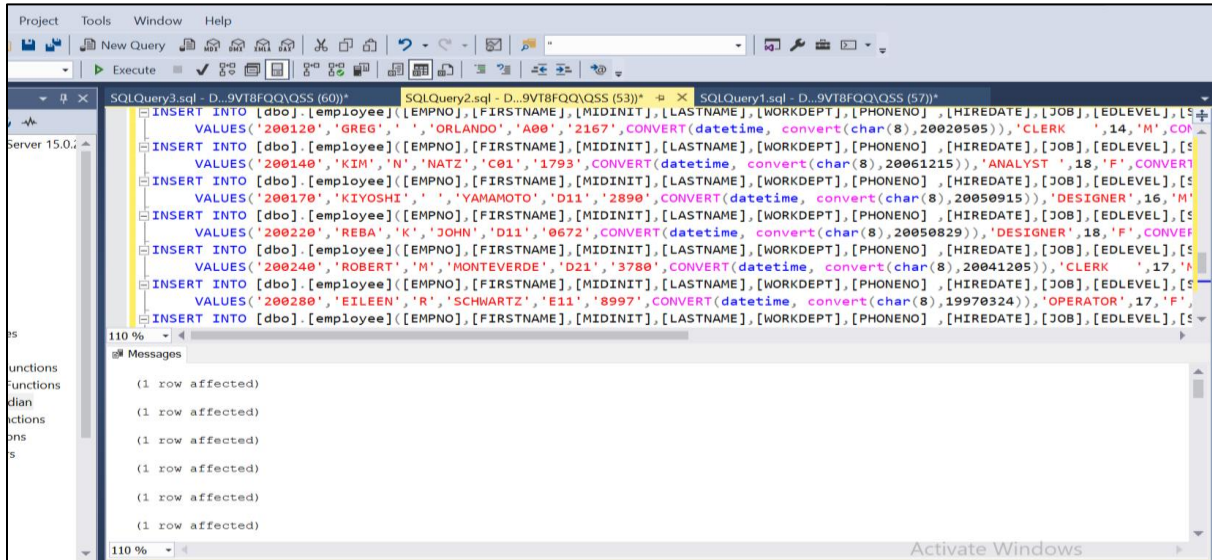
1. Preparation

Create Employee Table



The screenshot shows the SQL Server Enterprise Manager interface. The main window displays the SQL query: `create table employee (EMPNO char(6), FIRSTNAME varchar(12), MIDINIT char(1), LASTNAME varchar(15), WORKDEPT char(3), PHONENO char(4), HIREDATE date, JOB char(8), EDLEVEL smallint, SEX char(1), BIRTHDATE date, SALARY decimal(9,2), BONUS decimal(9,2), COMM decimal(9,2));`. The Messages pane at the bottom indicates "Commands completed successfully."

Insert data into Employee table



The screenshot shows the SQL Server Enterprise Manager interface with multiple SQL queries executed. The Messages pane displays the results of the INSERT statements, showing "(1 row affected)" for each. The queries include:

- `INSERT INTO [dbo].[employee] ([EMPNO], [FIRSTNAME], [MIDINIT], [LASTNAME], [WORKDEPT], [PHONENO], [HIREDATE], [JOB], [EDLEVEL], [SEX], [BIRTHDATE], [SALARY], [BONUS], [COMM]) VALUES ('200120', 'GREG', ' ', 'ORLANDO', 'A00', '2167', CONVERT(datetime, convert(char(8), 20020505)), 'CLERK', '14', 'M', 'C01', '19970324'), ('200140', 'KIM', 'N', 'NATZ', 'C01', '1793', CONVERT(datetime, convert(char(8), 20061215)), 'ANALYST', '18', 'F', 'C01', '19970324'), ('200170', 'KIYOSHI', ' ', 'YAMAMOTO', 'D11', '2890', CONVERT(datetime, convert(char(8), 20050915)), 'DESIGNER', '16', 'M', 'C01', '19970324'), ('200220', 'REBA', 'K', 'JOHN', 'D11', '0672', CONVERT(datetime, convert(char(8), 20050829)), 'DESIGNER', '18', 'F', 'C01', '19970324'), ('200240', 'ROBERT', 'M', 'MONTEVERDE', 'D21', '3780', CONVERT(datetime, convert(char(8), 20041205)), 'CLERK', '17', 'M', 'C01', '19970324'), ('200280', 'EILEEN', 'R', 'SCHWARTZ', 'E11', '8997', CONVERT(datetime, convert(char(8), 19970324)), 'OPERATOR', '17', 'F', 'C01', '19970324');`

SQLQuery3.sql - D...9VT8FQQ\QSS (60))* SQLQuery2.sql - D...9VT8FQQ\QSS (53))* SQLQuery1.sql - D...9VT8FQQ\QSS (57))*

```

Select count(*) from employee
Select * from employee

```

Server 15.0.2

Results

(No column name)

1 42

Messages

(No column name)

EMPNO	FIRSTNAME	MIDINIT	LASTNAME	WORKDEPT	PHONENO	HIREDATE	JOB	EDLEVEL	SEX	BIRTHDATE	SALARY	BONUS	COMM
1	CHRISTINE	I	HAAS	A00	3978	1995-01-01	PRES	18	F	1963-08-24	152750.00	1000.00	4220.00
2	MICHAEL	L	THOMPSON	B01	3476	2003-10-10	MANAGER	18	M	1978-02-02	94250.00	800.00	3300.00
3	SALLY	A	KWAN	C01	4738	2005-04-05	MANAGER	20	F	1971-05-11	98250.00	800.00	3060.00
4	JOHN	B	GEYER	E01	6789	1979-08-17	MANAGER	16	M	1955-09-15	80175.00	800.00	3214.00
5	IRVING	F	STERN	D11	6423	2003-09-14	MANAGER	16	M	1975-07-07	72250.00	500.00	2580.00
6	EVA	D	PULASKI	D21	7831	2005-09-30	MANAGER	16	F	2003-05-26	96170.00	700.00	2893.00
7	EILEEN	W	HENDER...	E11	5498	2000-08-15	MANAGER	16	F	1971-05-15	89750.00	600.00	2380.00
8	THEODORE	Q	SPENSER	E21	0972	2000-06-19	MANAGER	14	M	1980-12-18	86150.00	500.00	2092.00
9	VINCENZO	G	LUCCHES	A00	3490	1988-05-16	SALESMAN	19	M	1959-11-05	65000.00	800.00	3720.00

Overview of the employee table

Create Staff Table

Project Tools Window Help

Convert(datetime,%)

SQLQuery1.sql - D...9VT8FQQ\QSS (66))*

```

-- Create the STAFF table
--
create table staff (
  ID smallint,
  NAME varchar(9),
  DEPT smallint,
  JOB char(5),
  YEARS smallint,
  SALARY decimal(7,2),
  COMM decimal(7,2)
);

```

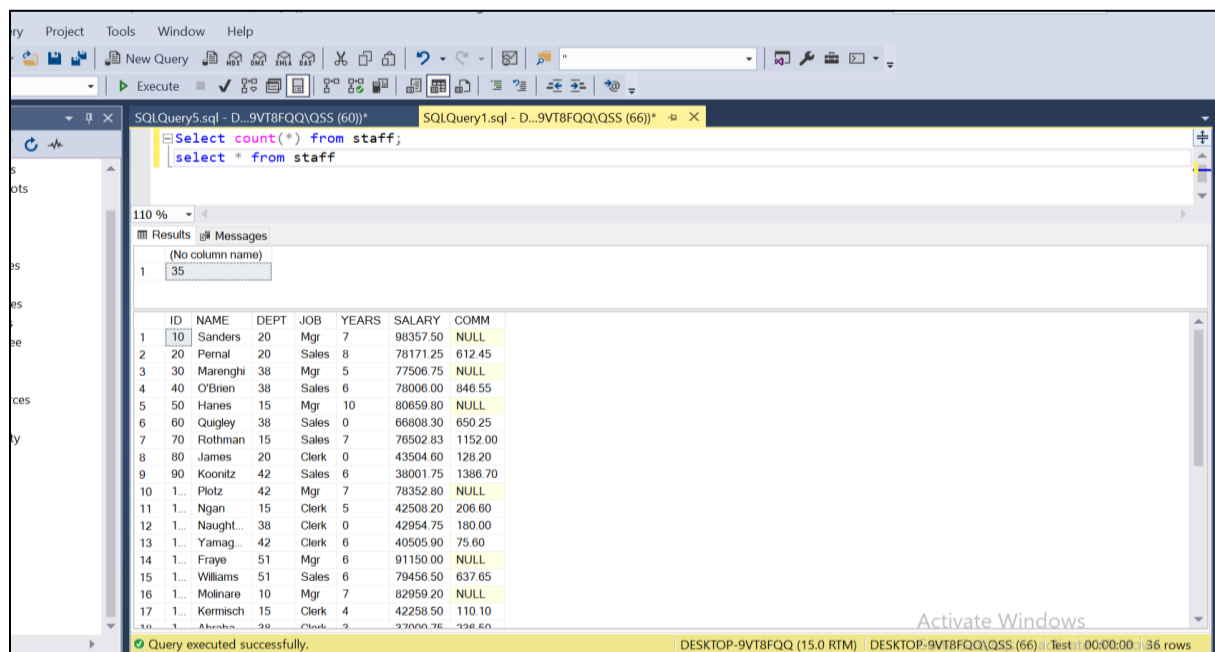
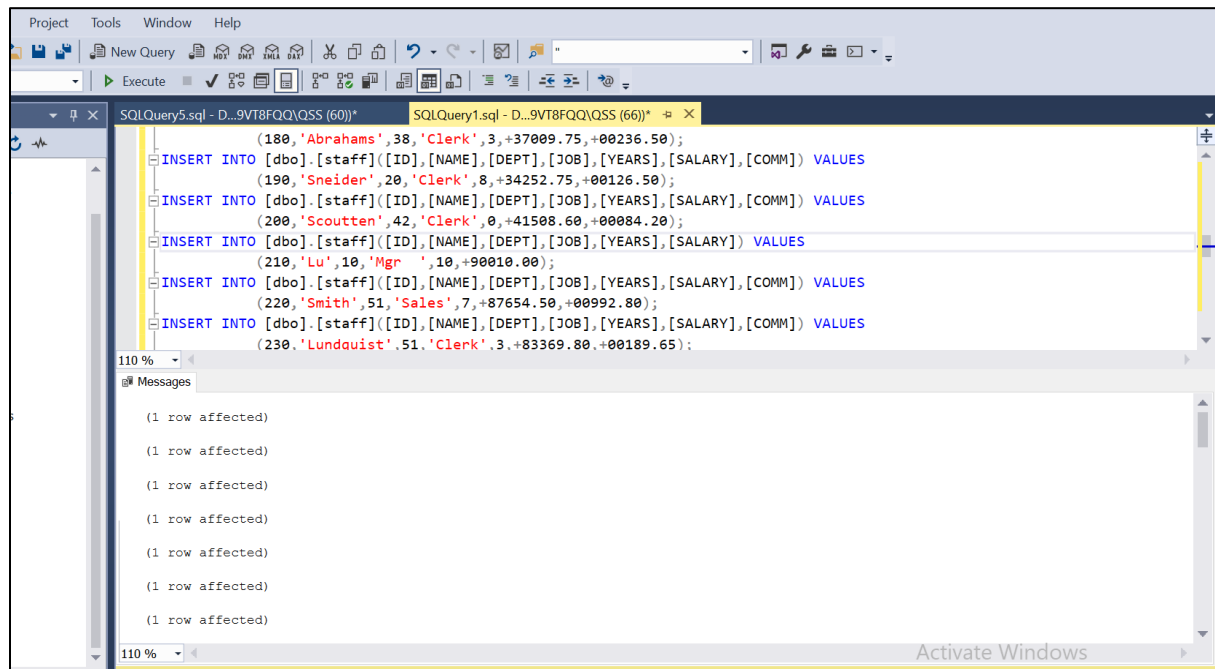
Server 15.0.209

Messages

Commands completed successfully.

Activate Windows

Insert data into Staff table



Overview of the staff table

2. stored procedure – *staff_add*

```
SQLQuery3.sql - lo...EU3H3Q\aaashi (58)) * SQLQuery2.sql - lo...EU3H3Q\aaashi (52)) * SQLQuery1.sql - lo...EU3H3Q\aaashi (60)) *  
Create procedure staff_add(@name as varchar(10),@job as varchar(10),@salary as decimal(18,2),@comm  
As Begin  
if @job in ('Sales','Clerk','Mgr')  
Begin  
Declare @Id as int  
Set @Id=0;  
Select @Id=max(ID) from staff;  
Set @Id=@Id+10;  
INSERT INTO [dbo].[staff]([ID],[NAME],[DEPT],[JOB],[YEARS],[SALARY],[COMM])  
VALUES(@Id, @name, 90, @job, 1, @salary,@comm);  
end  
else  
print 'Invalid Job'  
End
```

133 %
Messages
Commands completed successfully.
Completion time: 2022-12-01T22:54:15.6420189+05:30

Testing

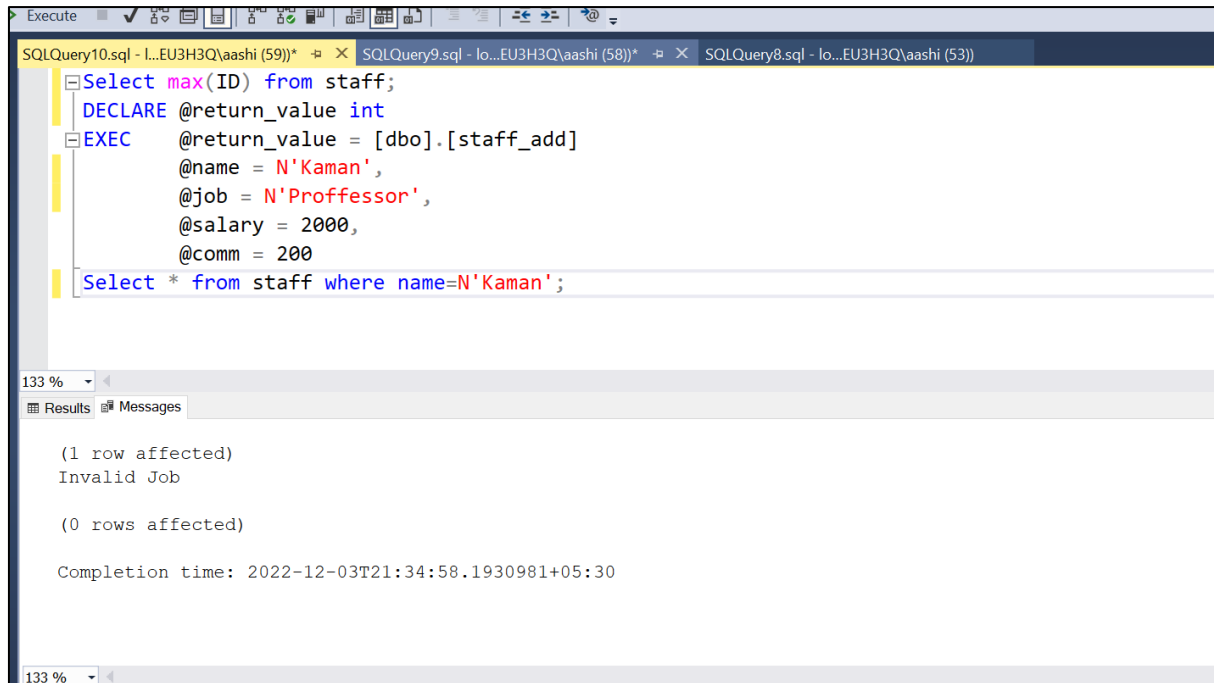
- Valid Job

```
SQLQuery10.sql - lo...EU3H3Q\aaashi (59)) * SQLQuery9.sql - lo...EU3H3Q\aaashi (58)) * SQLQuery8.sql - lo...EU3H3Q\aaashi (53))  
Select max(ID) from staff;  
DECLARE @return_value int  
EXEC @return_value = [dbo].[staff_add]  
@name = N'Anika',  
@job = N'Sales',  
@salary = 2000,  
@comm = 200  
Select * from staff;
```

133 %
Results Messages
(No column name)
1 350

	ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
1	10	Sanders	20	Mgr	7	98357.50	NULL
2	360	Anika	90	Sales	1	2000.00	200.00
3	20	Pernal	20	Sales	8	78171.25	612.45
4	30	Marengi	38	Mgr	5	77506.75	NULL
5	40	O'Brien	38	Sales	6	78006.00	846.55
6	50	Hanes	15	Mgr	10	80659.80	NULL
7	60	Quigley	38	Sales	0	66808.30	650.25
8	70	Rothman	15	Sales	7	76502.83	1152...
9	80	James	20	Clerk	0	43504.60	128.20

- **Invalid Job**



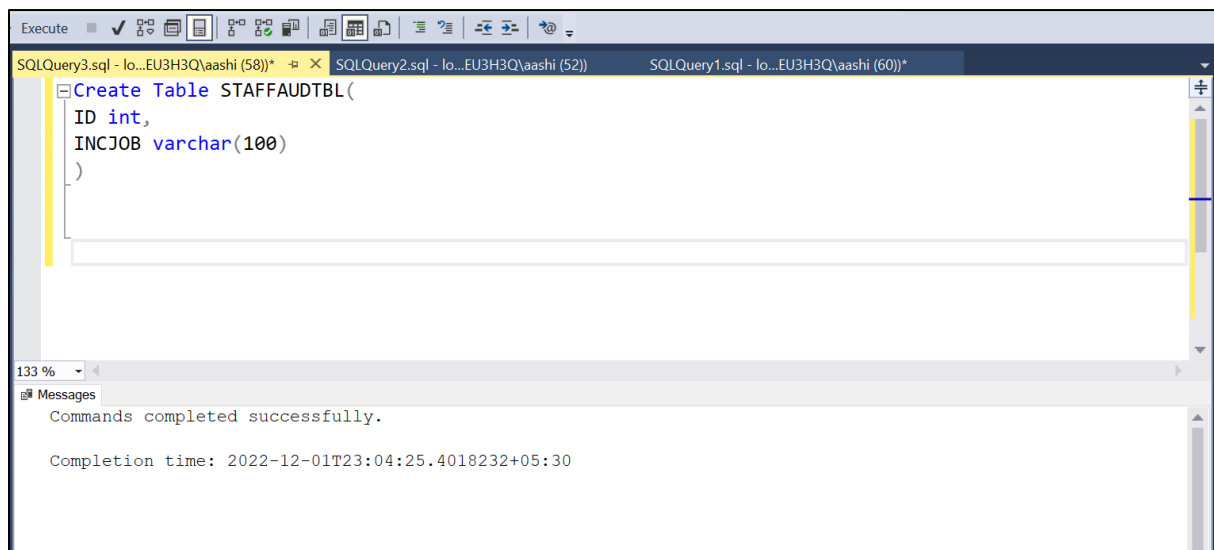
The screenshot shows a SQL Server Enterprise Manager window with the following SQL query:

```
Select max(ID) from staff;  
DECLARE @return_value int  
EXEC @return_value = [dbo].[staff_add]  
    @name = N'Kaman',  
    @job = N'Proffessor',  
    @salary = 2000,  
    @comm = 200  
Select * from staff where name=N'Kaman';
```

The execution results pane shows the following messages:

```
(1 row affected)  
Invalid Job  
  
(0 rows affected)  
  
Completion time: 2022-12-03T21:34:58.1930981+05:30
```

3. trigger – *ins_job*

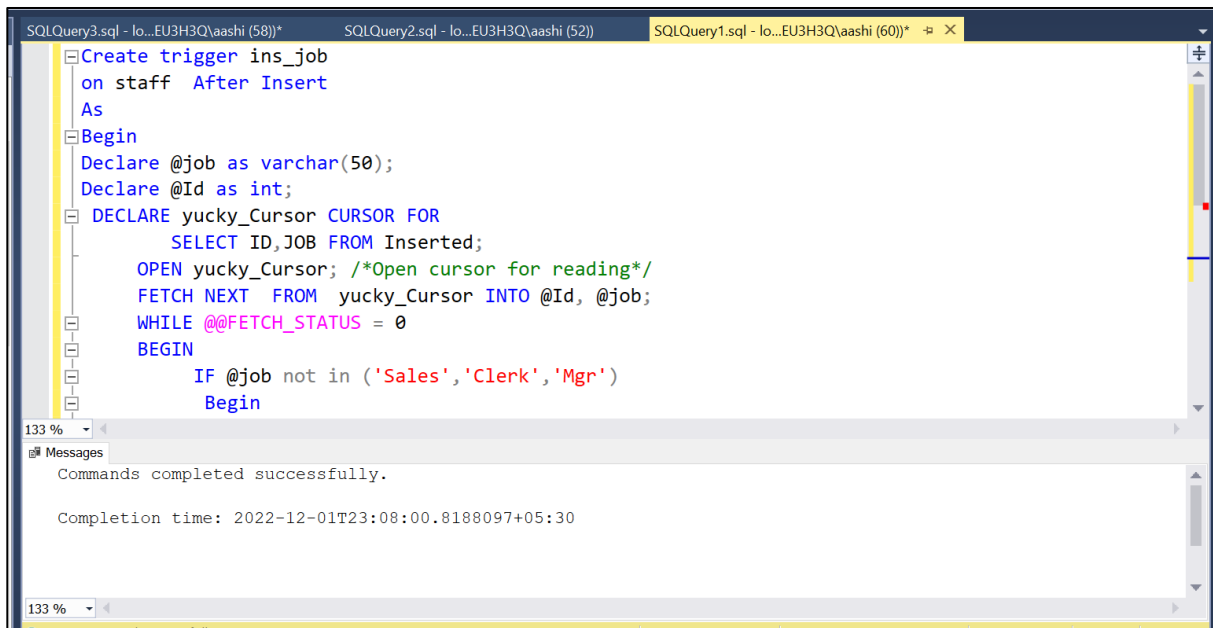


The screenshot shows a SQL Server Enterprise Manager window with the following SQL query:

```
Create Table STAFFAUDTBL(  
    ID int,  
    INCJOB varchar(100)  
)
```

The execution results pane shows the following messages:

```
Commands completed successfully.  
  
Completion time: 2022-12-01T23:04:25.4018232+05:30
```



```
SQLQuery3.sql - lo...EU3H3Q\aaashi (58)*  SQLQuery2.sql - lo...EU3H3Q\aaashi (52)  SQLQuery1.sql - lo...EU3H3Q\aaashi (60)*  X
-- Create trigger ins_job
-- on staff After Insert
-- As
-- Begin
-- Declare @job as varchar(50);
-- Declare @Id as int;
-- DECLARE yucky_Cursor CURSOR FOR
--     SELECT ID,JOB FROM Inserted;
-- OPEN yucky_Cursor; /*Open cursor for reading*/
-- FETCH NEXT FROM yucky_Cursor INTO @Id, @job;
-- WHILE @@FETCH_STATUS = 0
-- BEGIN
--     IF @job not in ('Sales', 'Clerk', 'Mgr')
--     Begin
--     End
-- End
```

133 %

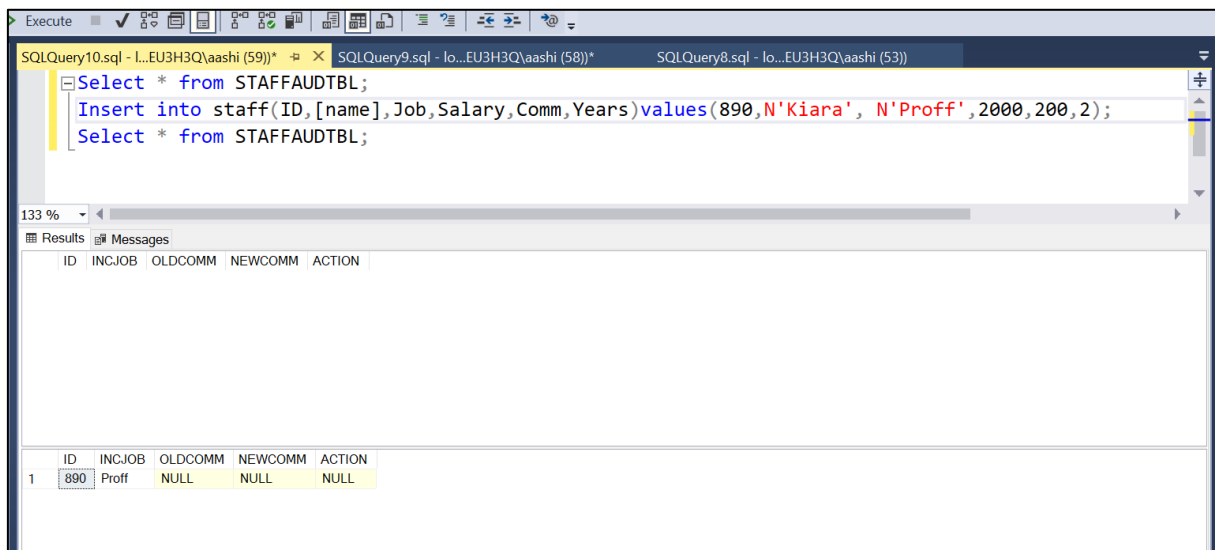
Messages

Commands completed successfully.

Completion time: 2022-12-01T23:08:00.8188097+05:30

133 %

Testing



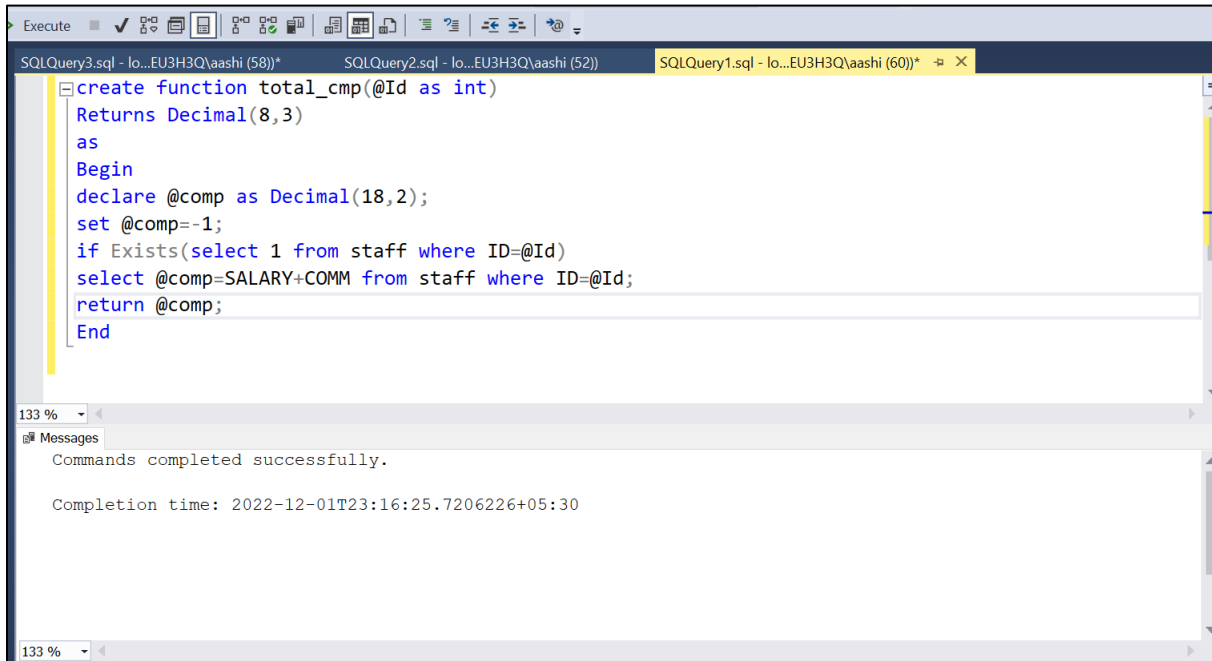
```
SQLQuery10.sql - I...EU3H3Q\aaashi (59)*  X  SQLQuery9.sql - lo...EU3H3Q\aaashi (58)*  SQLQuery8.sql - lo...EU3H3Q\aaashi (53)
-- Select * from STAFFAUDTBL;
-- Insert into staff(ID,[name],Job,Salary,Comm,Years) values(890,N'Kiara', N'Proff',2000,200,2);
-- Select * from STAFFAUDTBL;
```

133 %

Results Messages

	ID	INCJOB	OLDCOMM	NEWCOMM	ACTION
1	890	Proff	NULL	NULL	NULL

4. function – *total_cmp*



The screenshot shows the SQL Server Enterprise Manager interface with a query window titled 'SQLQuery1.sql - lo...EU3H3Q\vaashi (60))'. The query contains the following T-SQL code:

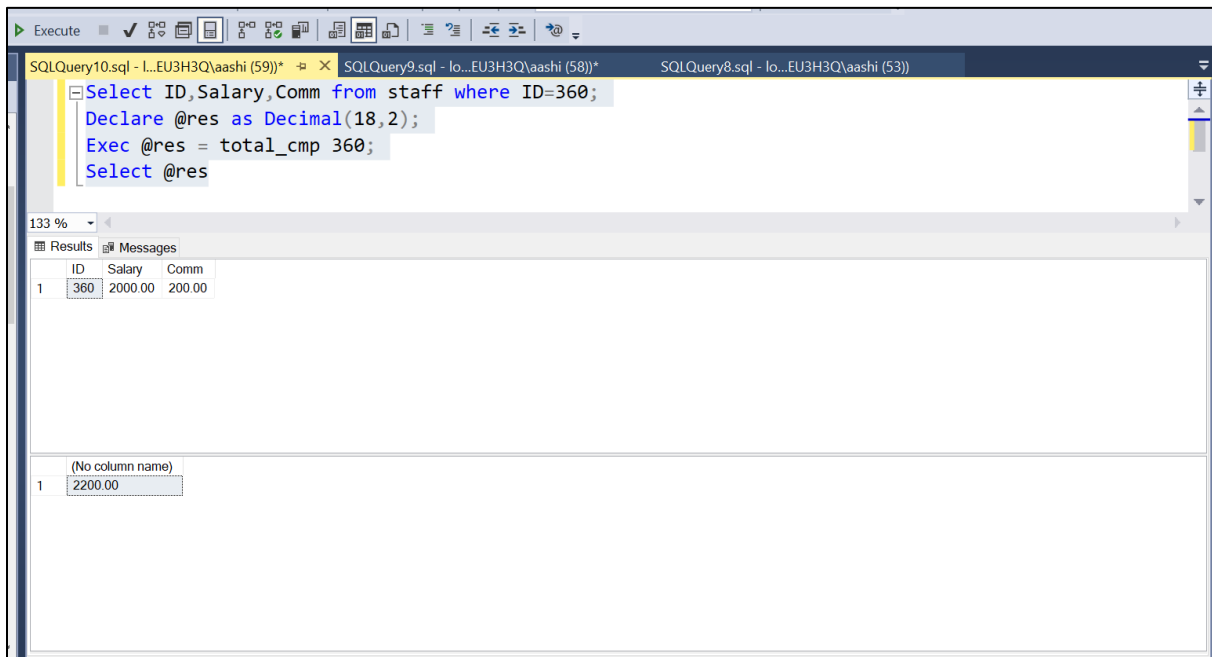
```
create function total_cmp(@Id as int)
Returns Decimal(8,3)
as
Begin
declare @comp as Decimal(18,2);
set @comp=-1;
if Exists(select 1 from staff where ID=@Id)
select @comp=SALARY+COMM from staff where ID=@Id;
return @comp;
End
```

Below the query window, the 'Messages' pane shows the following output:

```
Commands completed successfully.

Completion time: 2022-12-01T23:16:25.7206226+05:30
```

Testing



The screenshot shows the SQL Server Enterprise Manager interface with a query window titled 'SQLQuery10.sql - I...EU3H3Q\vaashi (59))'. The query contains the following T-SQL code:

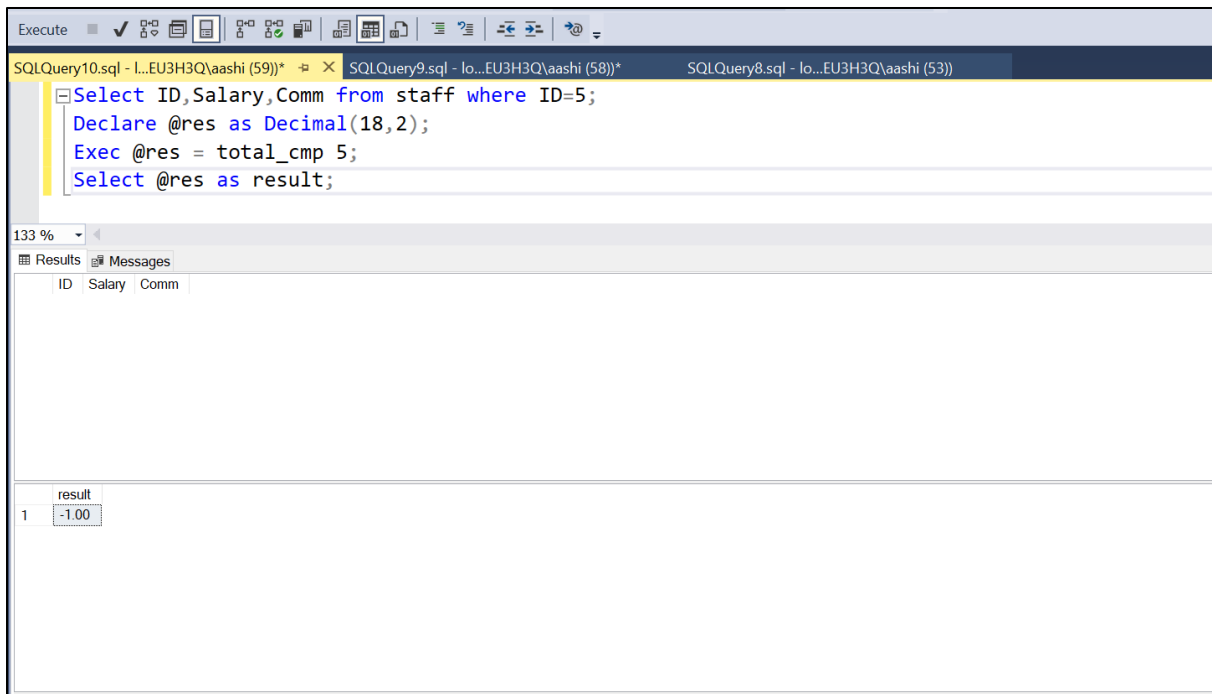
```
Select ID,Salary,Comm from staff where ID=360;
Declare @res as Decimal(18,2);
Exec @res = total_cmp 360;
Select @res
```

Below the query window, the 'Results' pane shows the output of the first query:

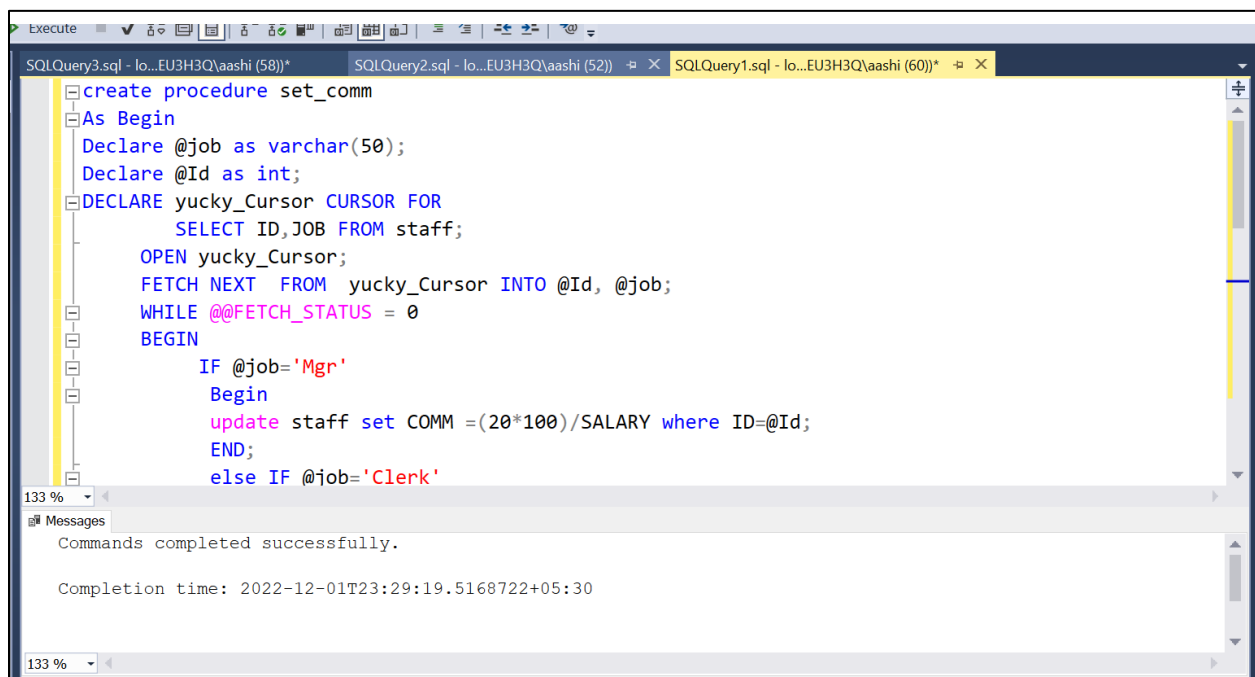
ID	Salary	Comm
360	2000.00	200.00

The 'Messages' pane shows the output of the second query:

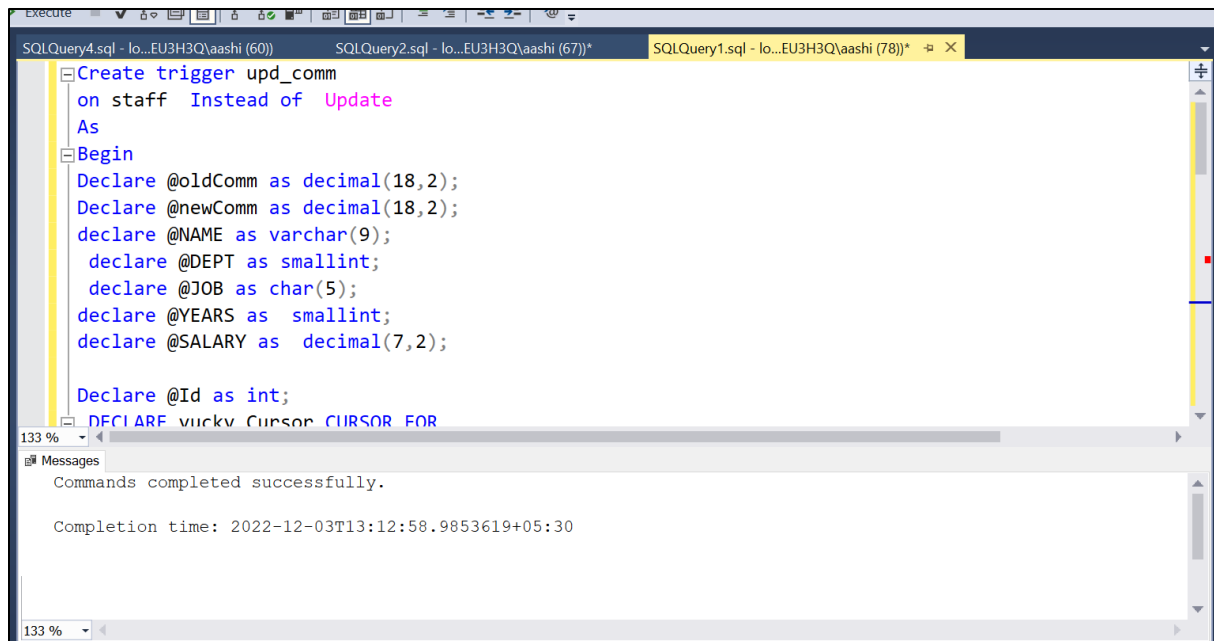
(No column name)
2200.00



5. stored procedure – *set_comm*



UPDATE trigger – upd_comm



The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays the SQL query for creating a trigger named 'upd_comm' on the 'staff' table. The trigger is set to fire 'Instead of Update'. The query includes variable declarations for @oldComm, @newComm, @NAME, @DEPT, @JOB, @YEARS, @SALARY, and @Id. The bottom pane shows the execution messages, indicating that the commands completed successfully and providing the completion time: 2022-12-03T13:12:58.9853619+05:30.

```
Create trigger upd_comm
on staff Instead of Update
As
Begin
Declare @oldComm as decimal(18,2);
Declare @newComm as decimal(18,2);
declare @NAME as varchar(9);
declare @DEPT as smallint;
declare @JOB as char(5);
declare @YEARS as smallint;
declare @SALARY as decimal(7,2);

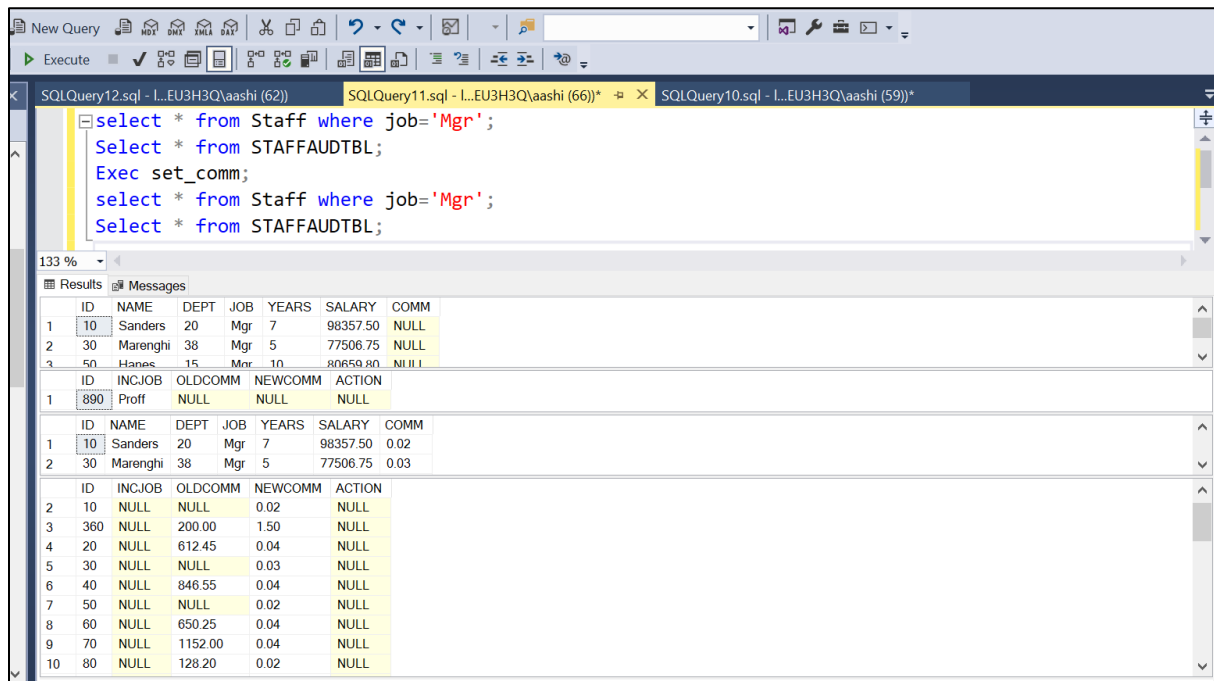
Declare @Id as int;
DECLARE vucky Cursor CURSOR FOR
```

Messages

Commands completed successfully.

Completion time: 2022-12-03T13:12:58.9853619+05:30

Testing



The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays the SQL query for testing the trigger. The query includes a select statement for staff with job 'Mgr', followed by a call to the 'set_comm' stored procedure, and another select statement for staff with job 'Mgr'. The bottom pane shows the results of the query, including the 'Results' tab and the 'Messages' tab. The 'Results' tab displays two tables: one showing staff details and another showing the results of the 'set_comm' procedure.

```
select * from Staff where job='Mgr';
Select * from STAFFAUDTBL;
Exec set_comm;
select * from Staff where job='Mgr';
Select * from STAFFAUDTBL;
```

Results

ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
10	Sanders	20	Mgr	7	98357.50	NULL
30	Marengi	38	Mgr	5	77506.75	NULL
50	Hanes	15	Mgr	10	80650.80	NULL

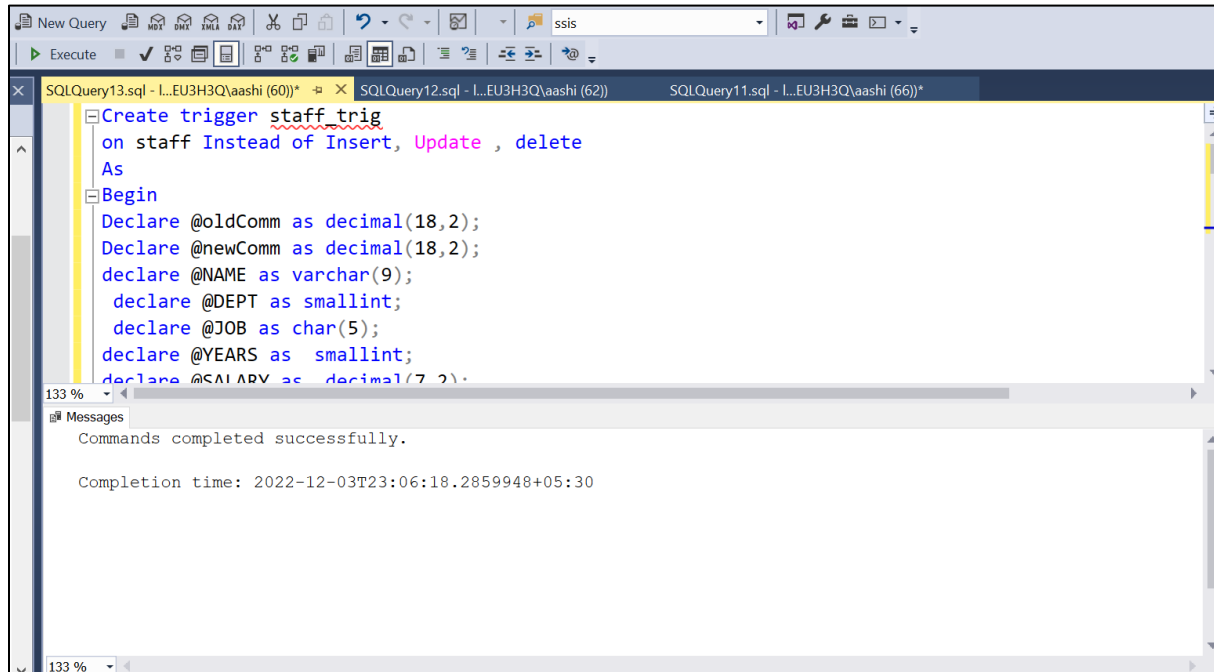
ID	INCJOB	OLDCOMM	NEWCOMM	ACTION
890	Proff	NULL	NULL	NULL

Messages

ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
10	Sanders	20	Mgr	7	98357.50	0.02
30	Marengi	38	Mgr	5	77506.75	0.03

ID	INCJOB	OLDCOMM	NEWCOMM	ACTION
10	NULL	NULL	0.02	NULL
360	NULL	200.00	1.50	NULL
20	NULL	612.45	0.04	NULL
30	NULL	NULL	0.03	NULL
40	NULL	846.55	0.04	NULL
50	NULL	NULL	0.02	NULL
60	NULL	650.25	0.04	NULL
70	NULL	1152.00	0.04	NULL
80	NULL	128.20	0.02	NULL

6. trigger – *staff_trig*



The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays the SQL code for creating a trigger named `staff_trig` on the `staff` table. The trigger is set to fire `Instead of Insert, Update, delete` operations. The code includes variable declarations for `@oldComm`, `@newComm`, `@NAME`, `@DEPT`, `@JOB`, `@YEARS`, and `@SALARY`. The bottom pane shows the execution results, indicating that the commands completed successfully and providing the completion time: 2022-12-03T23:06:18.2859948+05:30.

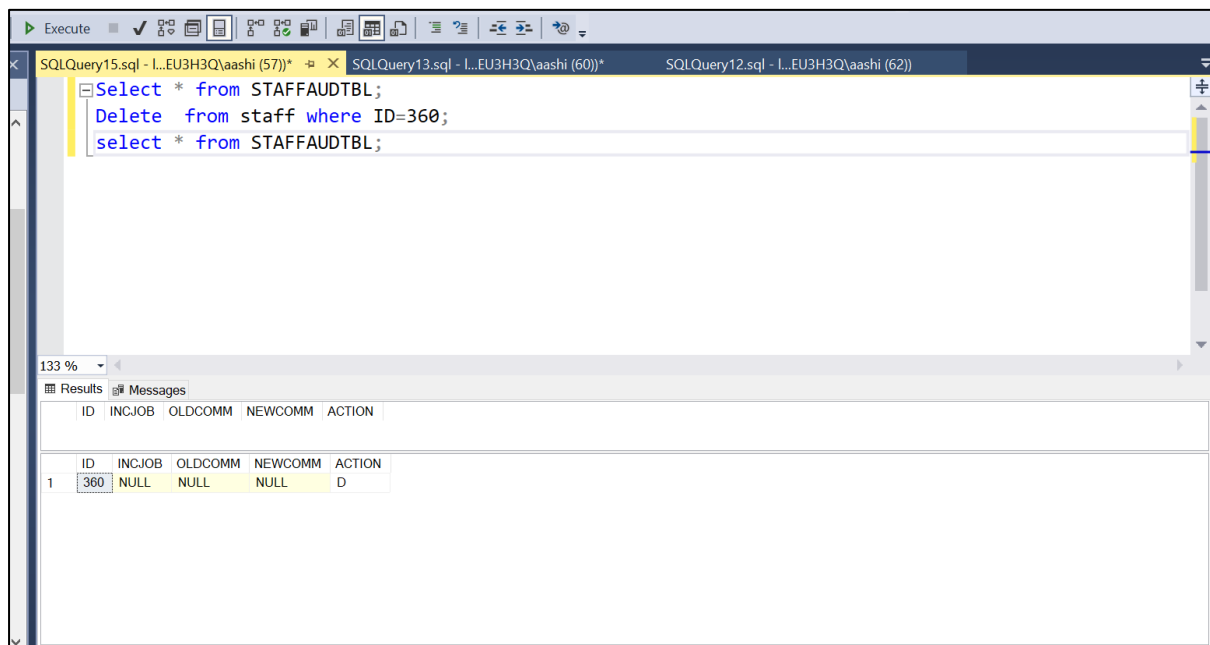
```
Create trigger staff_trig
on staff Instead of Insert, Update, delete
As
Begin
Declare @oldComm as decimal(18,2);
Declare @newComm as decimal(18,2);
declare @NAME as varchar(9);
declare @DEPT as smallint;
declare @JOB as char(5);
declare @YEARS as smallint;
declare @SALARY as decimal(7,2);
```

Messages

Commands completed successfully.

Completion time: 2022-12-03T23:06:18.2859948+05:30

Testing



The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays the SQL code for testing the trigger. The code consists of three statements: `Select * from STAFFAUDTBL;`, `Delete from staff where ID=360;`, and `select * from STAFFAUDTBL;`. The bottom pane shows the execution results, displaying a table with the following data:

ID	INCJOB	OLDCOMM	NEWCOMM	ACTION
1	360	NULL	NULL	D

SQLQuery15.sql - L...EU3H3Q\aaashi (57))* SQLQuery13.sql - L...EU3H3Q\aaashi (60))* SQLQuery12.sql - L...EU3H3Q\aaashi (62))

```

Select * from STAFFAUDTBL;
Update staff set Comm=120 where ID=10;
select * from STAFFAUDTBL;

```

133 %

Results Messages

	ID	INCJOB	OLDCOMM	NEWCOMM	ACTION
1	360	NULL	NULL	NULL	D

	ID	INCJOB	OLDCOMM	NEWCOMM	ACTION
1	360	NULL	NULL	NULL	D
2	10	NULL	NULL	120.00	U
3	10	NULL	NULL	NULL	D

SQLQuery15.sql - L...EU3H3Q\aaashi (57))* SQLQuery13.sql - L...EU3H3Q\aaashi (60))* SQLQuery12.sql - L...EU3H3Q\aaashi (62))

```

Select * from STAFFAUDTBL where ID=3008;
INSERT INTO [dbo].[staff]([ID],[NAME],[DEPT],[JOB],[YEARS],[SALARY],[COMM]) VALUES
(3008,'Davis',84,'Staff',5,+65454.50,+00806.10);
select * from STAFFAUDTBL where ID=3008;

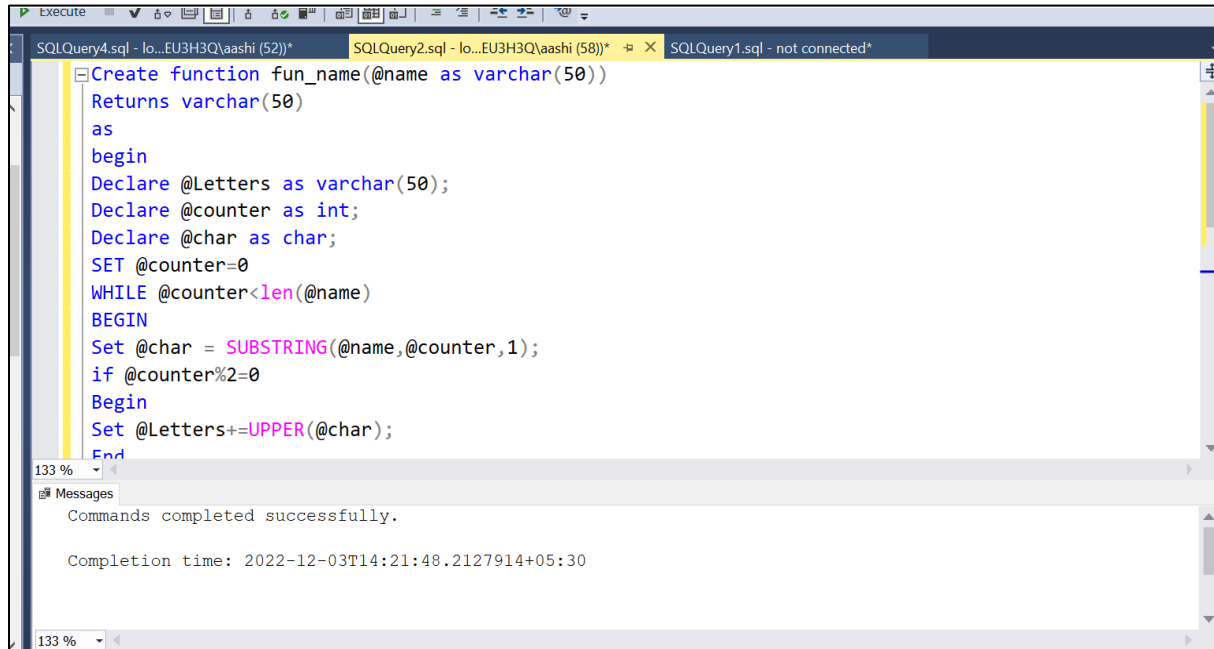
```

133 %

Results Messages

	ID	INCJOB	OLDCOMM	NEWCOMM	ACTION
1	3008	Staff	NULL	NULL	I

7. function – *fun_name*



The screenshot shows the SQL Server Enterprise Manager interface. The main window displays the following T-SQL code for creating a function:

```
Create function fun_name(@name as varchar(50))
Returns varchar(50)
as
begin
Declare @Letters as varchar(50);
Declare @counter as int;
Declare @char as char;
SET @counter=0
WHILE @counter<len(@name)
BEGIN
Set @char = SUBSTRING(@name,@counter,1);
if @counter%2=0
Begin
Set @Letters+=UPPER(@char);
End

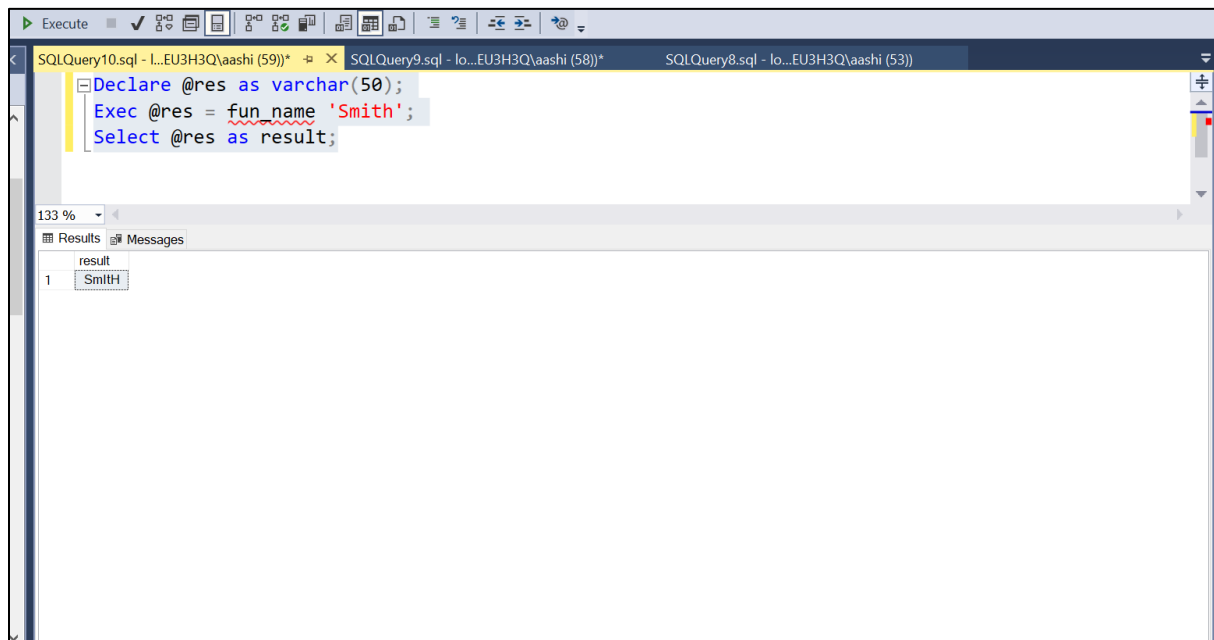
```

Below the code editor, the Messages pane shows the following output:

```
Commands completed successfully.

Completion time: 2022-12-03T14:21:48.2127914+05:30
```

Testing

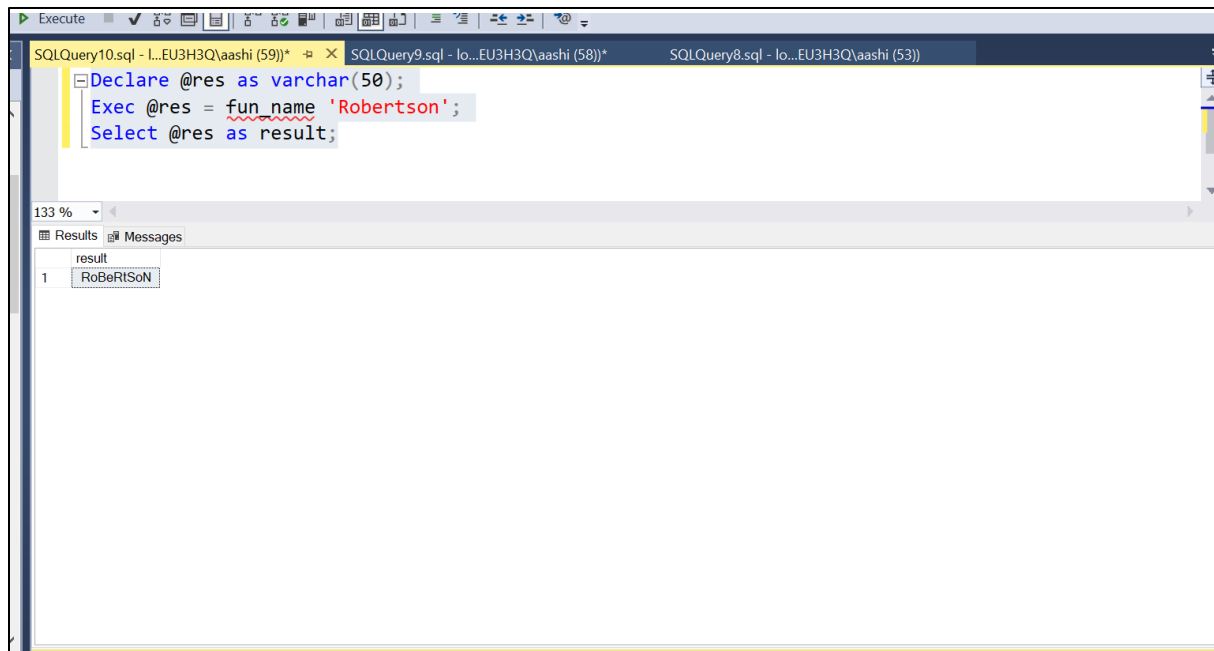


The screenshot shows the SQL Server Enterprise Manager interface. The main window displays the following T-SQL code for testing the function:

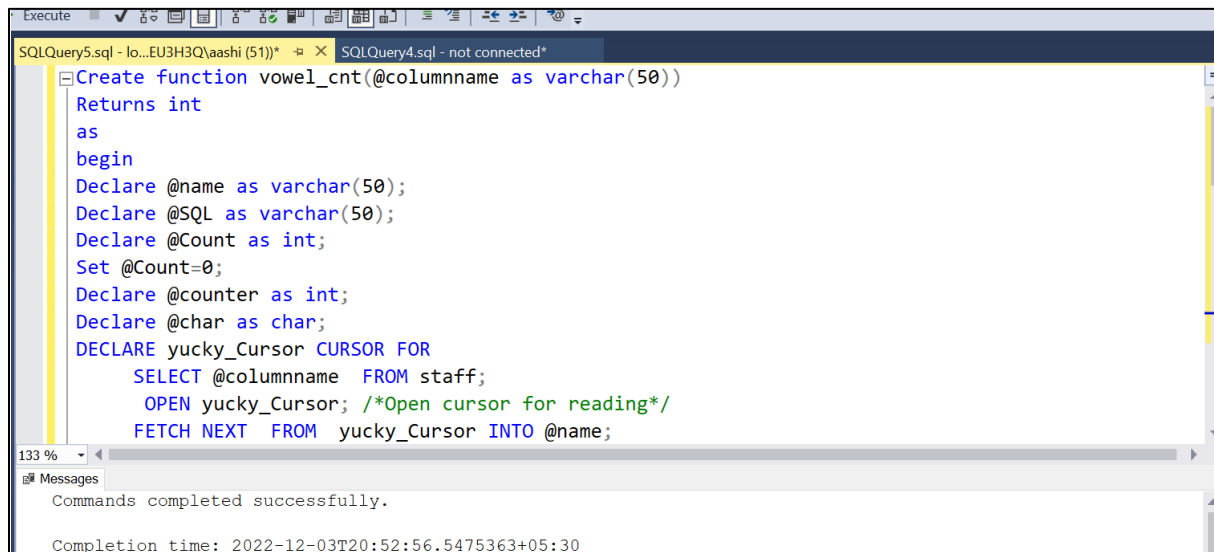
```
Declare @res as varchar(50);
Exec @res = fun_name 'Smith';
Select @res as result;
```

Below the code editor, the Results pane shows the following output:

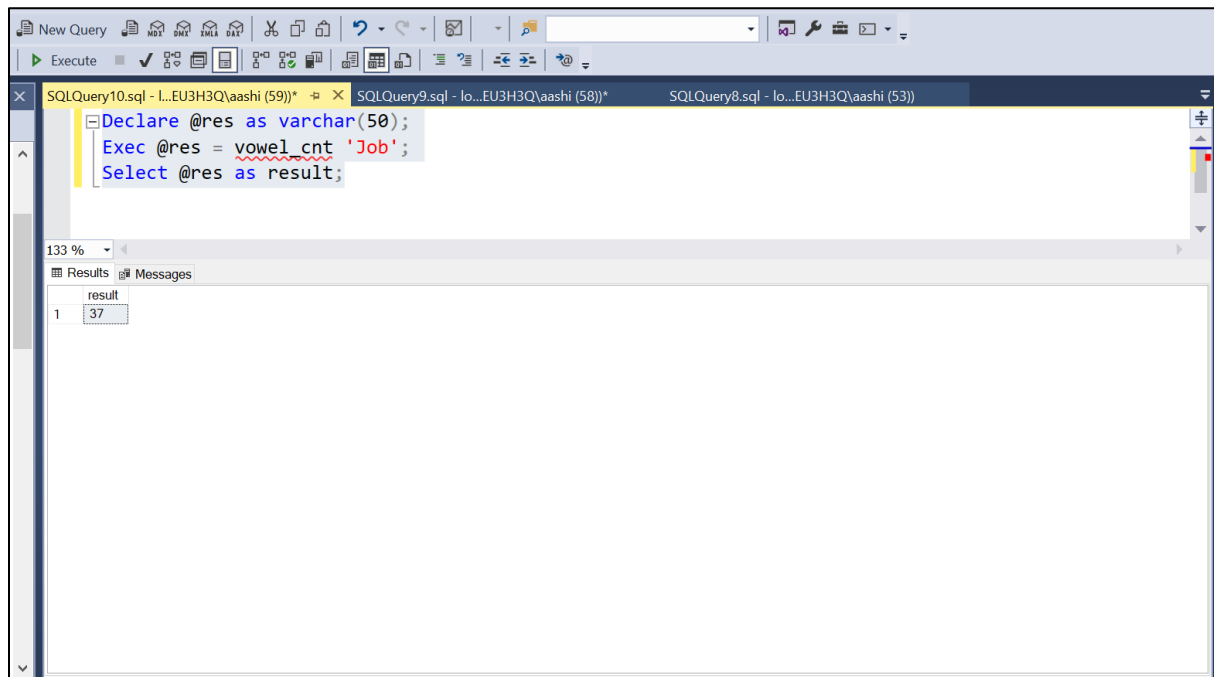
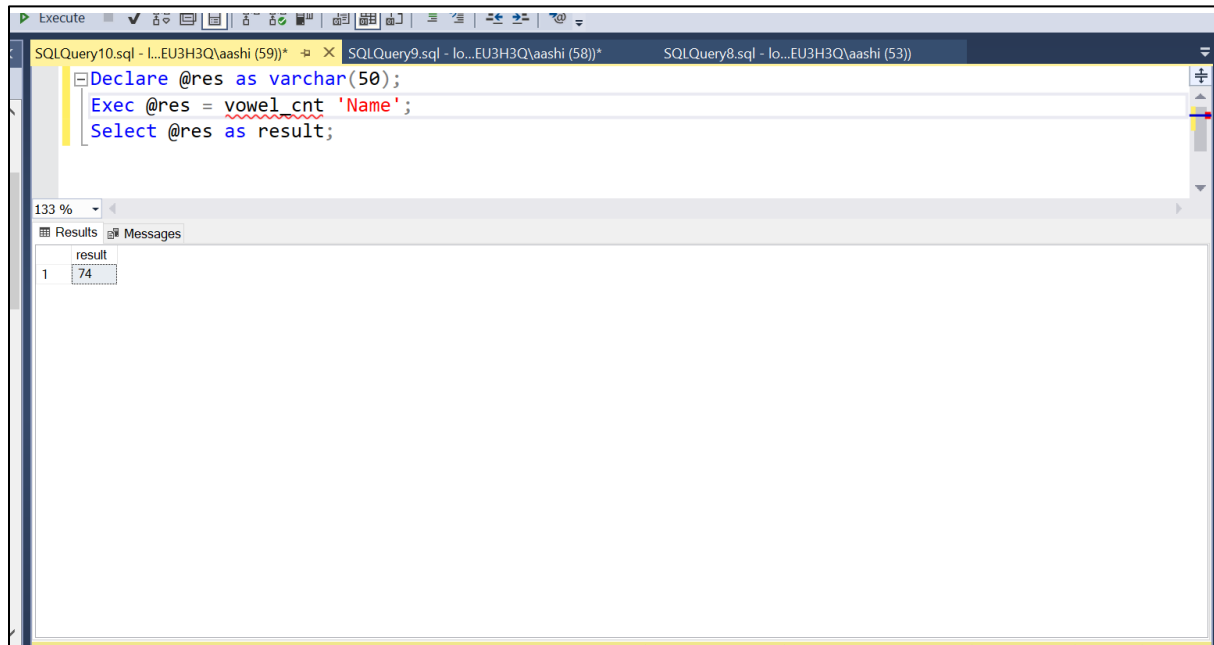
result
1 Smith



8. function – *vowel_cnt*



Testing



9. function – *staff_pck*

```
1 create or replace package staff_pck
2 as
3 procedure staff_add(name in varchar,job in varchar,salary in decimal,comm in decimal);
4 function total_cmp(Id in int) Return Decimal;
5 procedure set_comm;
6 function fun_name(name in varchar) Return varchar;
7 function vowel_cnt(columnname in varchar) Return int;
8 end;
```

Package created.