

ASSIGNMENT 1 – STORED PROCEDURES

Name- Amrit Pal Singh

Student ID- 110355211

1. Preparation

- **Create Employee Table**

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. The title bar reads "SQLQuery4.sql - DESKTOP-9VT8FQQ.Test2 (DESKTOP-9VT8FQQ\QSS (61)) - Microsoft SQL Server Management Studio". The left pane is the Object Explorer, displaying the database structure for "DESKTOP-9VT8FQQ (SQL Server 15.0)". The central pane contains three tabs representing open queries: "SQLQuery4.sql - D...9VT8FQQ\QSS (61)*", "SQLQuery2.sql - D...9VT8FQQ\QSS (58)*", and "SQLQuery1.sql - D...9VT8FQQ\QSS (75)*". The "SQLQuery4.sql" tab is active, showing a T-SQL script to create a table named "[dbo].[employee]" with various columns and constraints. The "Messages" tab at the bottom indicates that the command was completed successfully. The status bar at the bottom right shows the date and time as "10/22/2022 10:44 AM".

- **Insert data into Employee table**

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "SQLQuery4.sql - DESKTOP-9VT8FQQ\QSS (61)* - Microsoft SQL Server Management Studio". The Object Explorer sidebar on the left lists the database structure, including the Test2 database which contains the employee table. The central pane displays three queries being run simultaneously:

```
SQLQuery4.sql - D...9VT8FQQ\QSS (61)*  SQLQuery2.sql - D...9VT8FQQ\QSS (58)*  SQLQuery1.sql - D...9VT8FQQ\QSS (75)*  
INSERT [dbo].[employee] ([EMPNO], [FIRSTNAME], [MIDINIT], [LASTNAME], [WORKDEPT], [PHONENO], [HIREDATE], [JOB], [EDLEVEL], [SEX]  
INSERT [dbo].[employee] ([EMPNO], [FIRSTNAME], [MIDINIT], [LASTNAME], [WORKDEPT], [PHONENO], [HIREDATE], [JOB], [EDLEVEL], [SEX]  
INSERT [dbo].[employee] ([EMPNO], [FIRSTNAME], [MIDINIT], [LASTNAME], [WORKDEPT], [PHONENO], [HIREDATE], [JOB], [EDLEVEL], [SEX]  
INSERT [dbo].[employee] ([EMPNO], [FIRSTNAME], [MIDINIT], [LASTNAME], [WORKDEPT], [PHONENO], [HIREDATE], [JOB], [EDLEVEL], [SEX]  
INSERT [dbo].[employee] ([EMPNO], [FIRSTNAME], [MIDINIT], [LASTNAME], [WORKDEPT], [PHONENO], [HIREDATE], [JOB], [EDLEVEL], [SEX]  
INSERT [dbo].[employee] ([EMPNO], [FIRSTNAME], [MIDINIT], [LASTNAME], [WORKDEPT], [PHONENO], [HIREDATE], [JOB], [EDLEVEL], [SEX]  
INSERT [dbo].[employee] ([EMPNO], [FIRSTNAME], [MIDINIT], [LASTNAME], [WORKDEPT], [PHONENO], [HIREDATE], [JOB], [EDLEVEL], [SEX]  
INSERT [dbo].[employee] ([EMPNO], [FIRSTNAME], [MIDINIT], [LASTNAME], [WORKDEPT], [PHONENO], [HIREDATE], [JOB], [EDLEVEL], [SEX]  
INSERT [dbo].[employee] ([EMPNO], [FIRSTNAME], [MIDINIT], [LASTNAME], [WORKDEPT], [PHONENO], [HIREDATE], [JOB], [EDLEVEL], [SEX]  
INSERT [dbo].[employee] ([EMPNO], [FIRSTNAME], [MIDINIT], [LASTNAME], [WORKDEPT], [PHONENO], [HIREDATE], [JOB], [EDLEVEL], [SEX]  
INSERT [dbo].[employee] ([EMPNO], [FIRSTNAME], [MIDINIT], [LASTNAME], [WORKDEPT], [PHONENO], [HIREDATE], [JOB], [EDLEVEL], [SEX]  
INSERT [dbo].[employee] ([EMPNO], [FIRSTNAME], [MIDINIT], [LASTNAME], [WORKDEPT], [PHONENO], [HIREDATE], [JOB], [EDLEVEL], [SEX]  
INSERT [dbo].[employee] ([EMPNO], [FIRSTNAME], [MIDINIT], [LASTNAME], [WORKDEPT], [PHONENO], [HIREDATE], [JOB], [EDLEVEL], [SEX]  
INSERT [dbo].[employee] ([EMPNO], [FIRSTNAME], [MIDINIT], [LASTNAME], [WORKDEPT], [PHONENO], [HIREDATE], [JOB], [EDLEVEL], [SEX]  
INSERT [dbo].[employee] ([EMPNO], [FIRSTNAME], [MIDINIT], [LASTNAME], [WORKDEPT], [PHONENO], [HIREDATE], [JOB], [EDLEVEL], [SEX]  
INSERT [dbo].[employee] ([EMPNO], [FIRSTNAME], [MIDINIT], [LASTNAME], [WORKDEPT], [PHONENO], [HIREDATE], [JOB], [EDLEVEL], [SEX]  
INSERT [dbo].[employee] ([EMPNO], [FIRSTNAME], [MIDINIT], [LASTNAME], [WORKDEPT], [PHONENO], [HIREDATE], [JOB], [EDLEVEL], [SEX]  
INSERT [dbo].[employee] ([EMPNO], [FIRSTNAME], [MIDINIT], [LASTNAME], [WORKDEPT], [PHONENO], [HIREDATE], [JOB], [EDLEVEL], [SEX]  
INSERT [dbo].[employee] ([EMPNO], [FIRSTNAME], [MIDINIT], [LASTNAME], [WORKDEPT], [PHONENO], [HIREDATE], [JOB], [EDLEVEL], [SEX]  
100% 4  
# Messages  
  
(1 row affected)  
  
(1 row affected)
```

The status bar at the bottom indicates "Query executed successfully." and "DESKTOP-9VT8FQQ (15.0 RTM) DESKTOP-9VT8FQQ\QSS (61)* Test2 00:00:00 0 rows".

SQLQuery4.sql - DESKTOP-9VT8FQQ.Test2 (DESKTOP-9VT8FQQ\QSS (61))* - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Test2

Object Explorer

Connect ▾ Test2

Databases

- System Databases
- Database Snapshots
- NZWalksDb
- Test2
- Tables
- System Tables
- FileTables
- External Tables
- Graph Tables
- dbo.employee
- Views
- External Resources
- Synonyms
- Programmability
- Service Broker
- Storage
- Security
- Security
- Server Objects
- Replication
- PolyBase
- Management
- XEvent Profiler

SQLQuery4.sql - D...9VT8FQQ\QSS (61)* SQLQuery2.sql - D...9VT8FQQ\QSS (58)* SQLQuery1.sql - D...9VT8FQQ\QSS (75)*

Select count(*) from employee

Results Messages

(No column name)

1 42

Query executed successfully.

DESKTOP-9VT8FQQ (15.0 RTM) | DESKTOP-9VT8FQQ\QSS (61) | Test2 | 00:00:00 | 1 rows

Activate Windows

Ready Type here to search

Ln 1 Col 30 Ch 30 INS

27°C 🔍 1048 AM 10/22/2022

SQLQuery4.sql - DESKTOP-9VT8FQQ.Test2 (DESKTOP-9VT8FQQ\QSS (61))* - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Test2

Object Explorer

Connect ▾ Test2

Databases

- System Databases
- Database Snapshots
- NZWalksDb
- Test2
- Tables
- System Tables
- FileTables
- External Tables
- Graph Tables
- dbo.employee
- Views
- External Resources
- Synonyms
- Programmability
- Service Broker
- Storage
- Security
- Security
- Server Objects
- Replication
- PolyBase
- Management
- XEvent Profiler

SQLQuery4.sql - D...9VT8FQQ\QSS (61)* SQLQuery2.sql - D...9VT8FQQ\QSS (58)* SQLQuery1.sql - D...9VT8FQQ\QSS (75)*

Select * from employee

Results Messages

	EMPNO	FIRSTNAME	MIDINIT	LASTNAME	WORKDEPT	PHONE NO	HIREDATE	JOB	EDLEVEL	SEX	BIRTHDATE	SALARY	BONUS	COMM	RATING
1	10	CHRISTINE	I	HAAS	A00	3978	1995-01-01	PRES	18	F	1963-08-24	152750	1000	4220	1
2	20	MICHAEL	L	THOMPSON	B01	3476	2003-10-10	MANAGER	18	M	1978-02-02	94250	800	3300	2
3	30	SALLY	A	KWAN	C01	4738	2005-04-05	MANAGER	20	F	1971-05-11	98250	800	3060	3
4	50	JOHN	B	GEYER	E01	6789	1979-08-17	MANAGER	16	M	1955-09-15	80175	800	3214	1
5	60	IRVING	F	STERN	D11	6423	2003-09-14	MANAGER	16	M	1975-07-07	72250	500	2580	2
6	70	EVA	D	PULASKI	D21	7831	2005-09-30	MANAGER	16	F	2003-05-26	96170	700	2893	3
7	90	EILEEN	W	HENDERSON	E11	5498	2000-08-15	MANAGER	16	F	1971-05-15	89750	600	2380	3
8	100	THEODORE	Q	SPENSER	E21	972	2000-06-19	MANAGER	14	M	1980-12-18	86150	500	2092	2
9	110	VINCENZO	G	LUCCHESI	A00	3490	1988-05-16	SALESREP	19	M	1959-11-05	66500	900	3720	1
10	120	SEAN	NULL	O'CONNELL	A00	2167	1993-12-05	CLERK	14	M	1972-10-18	49250	600	2340	1
11	130	DELORES	M	QUINTANA	C01	4578	2001-07-28	ANALYST	16	F	1955-09-15	73800	500	1904	2
12	140	HEATH	X	MONROE	C01	3200	2008-12-31	ANALYST	16	F	1978-03-10	100000	10000	20000	5

Query executed successfully.

DESKTOP-9VT8FQQ (15.0 RTM) | DESKTOP-9VT8FQQ\QSS (61) | Test2 | 00:00:00 | 42 rows

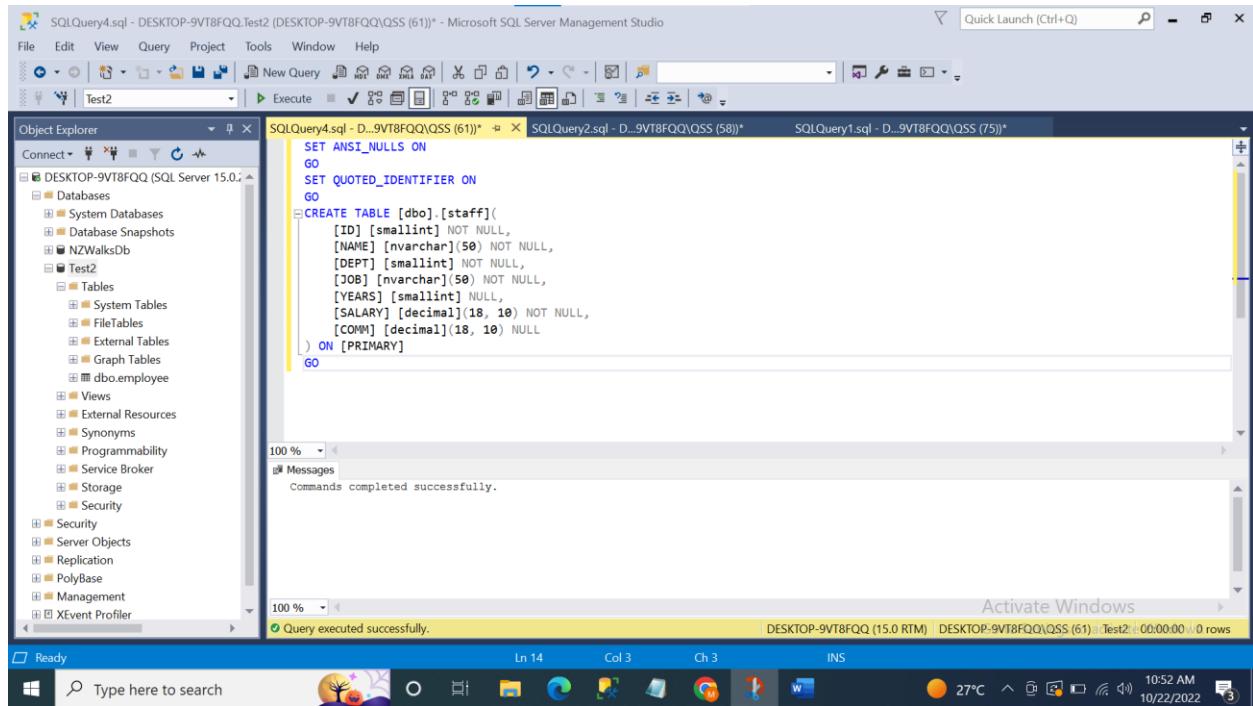
Ready Type here to search

Ln 1 Col 9 Ch 9 INS

27°C 🔍 1049 AM 10/22/2022

Overview of the employee table

• Create Staff Table



The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, a database named 'Test2' is selected. In the center pane, a query window displays the creation script for the 'staff' table:

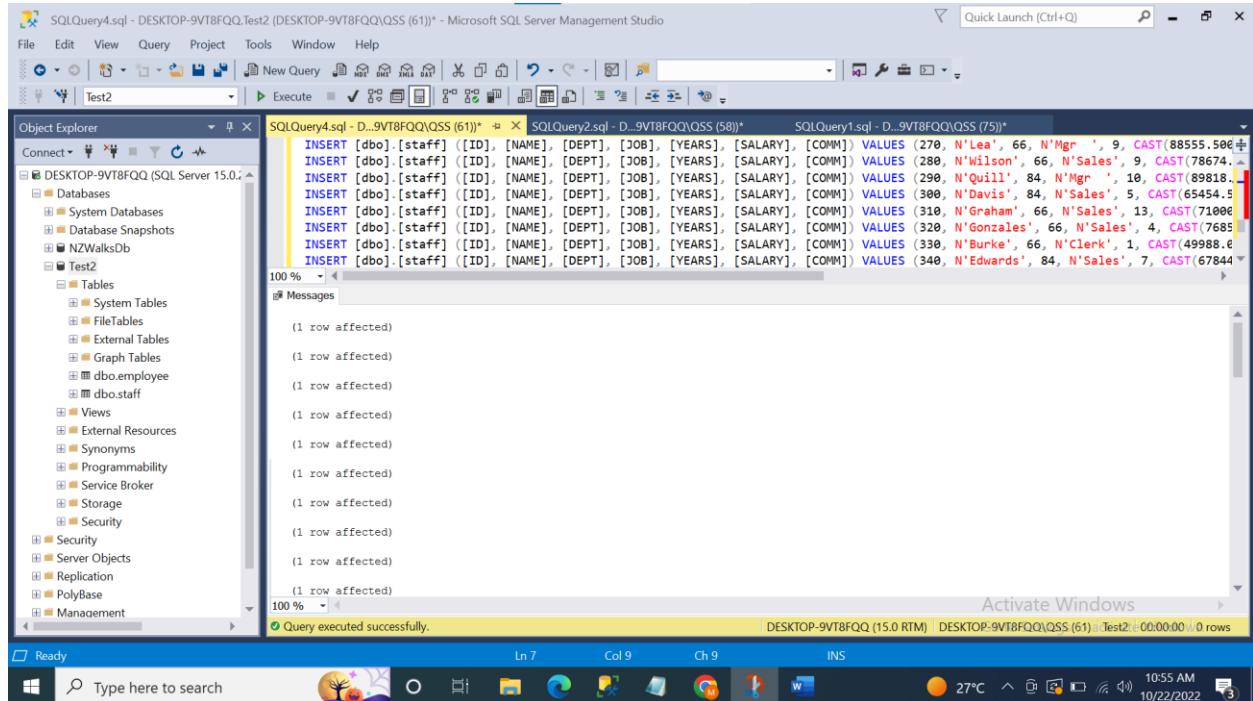
```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[staff](
    [ID] [smallint] NOT NULL,
    [NAME] [nvarchar](50) NOT NULL,
    [DEPT] [smallint] NOT NULL,
    [JOB] [nvarchar](50) NOT NULL,
    [YEARS] [smallint] NULL,
    [SALARY] [decimal](18, 10) NOT NULL,
    [COMM] [decimal](18, 10) NULL
) ON [PRIMARY]
GO

```

The status bar at the bottom indicates "Query executed successfully." and "10 rows".

• Insert data into Staff table



The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, a database named 'Test2' is selected. In the center pane, a query window displays the insertion of data into the 'staff' table:

```

INSERT [dbo].[staff] ([ID], [NAME], [DEPT], [JOB], [YEARS], [SALARY], [COMM]) VALUES (270, N'Lea', 66, N'Mgr', 9, CAST(88555.500 AS decimal(18, 2)))
INSERT [dbo].[staff] ([ID], [NAME], [DEPT], [JOB], [YEARS], [SALARY], [COMM]) VALUES (280, N'Wilson', 66, N'Sales', 9, CAST(78674.500 AS decimal(18, 2)))
INSERT [dbo].[staff] ([ID], [NAME], [DEPT], [JOB], [YEARS], [SALARY], [COMM]) VALUES (290, N'Quill', 84, N'Mgr', 18, CAST(89818.500 AS decimal(18, 2)))
INSERT [dbo].[staff] ([ID], [NAME], [DEPT], [JOB], [YEARS], [SALARY], [COMM]) VALUES (300, N'Davis', 84, N'Sales', 5, CAST(65454.500 AS decimal(18, 2)))
INSERT [dbo].[staff] ([ID], [NAME], [DEPT], [JOB], [YEARS], [SALARY], [COMM]) VALUES (310, N'Graham', 66, N'Sales', 13, CAST(71000.000 AS decimal(18, 2)))
INSERT [dbo].[staff] ([ID], [NAME], [DEPT], [JOB], [YEARS], [SALARY], [COMM]) VALUES (320, N'Gonzales', 66, N'Sales', 4, CAST(76850.000 AS decimal(18, 2)))
INSERT [dbo].[staff] ([ID], [NAME], [DEPT], [JOB], [YEARS], [SALARY], [COMM]) VALUES (330, N'Burke', 66, N'Clerk', 1, CAST(49988.000 AS decimal(18, 2)))
INSERT [dbo].[staff] ([ID], [NAME], [DEPT], [JOB], [YEARS], [SALARY], [COMM]) VALUES (340, N'Edwards', 84, N'Sales', 7, CAST(67844.000 AS decimal(18, 2)))

```

The status bar at the bottom indicates "Query executed successfully." and "10 rows".

SQLQuery4.sql - DESKTOP-9VT8FQQ.Test2 (DESKTOP-9VT8FQQ\QSS (61)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Test2

Object Explorer

Connect ▾

DESKTOP-9VT8FQQ (SQL Server 15.0.)

- Databases
 - System Databases
 - Database Snapshots
 - NZWalksDb
- Test2
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.employee
 - dbo.staff
 - Views
 - External Resources
 - Synonyms
 - Programmability
 - Service Broker
 - Storage
 - Security
- Security
- Server Objects
- Replication
- PolyBase
- Management

100 %

Results Messages

(No column name)

1	35
---	----

Query executed successfully.

DESKTOP-9VT8FQQ (15.0 RTM) | DESKTOP-9VT8FQQ\QSS (61) : Test2 : 00:00:00 | 1 rows

Ready Type here to search

Ln 1 Col 23 Ch 23 INS

Activate Windows

27°C 10:55 AM 10/22/2022

SQLQuery4.sql - DESKTOP-9VT8FQQ.Test2 (DESKTOP-9VT8FQQ\QSS (61)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Test2

Object Explorer

Connect ▾

DESKTOP-9VT8FQQ (SQL Server 15.0.)

- Databases
 - System Databases
 - Database Snapshots
 - NZWalksDb
- Test2
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.employee
 - dbo.staff
 - Views
 - External Resources
 - Synonyms
 - Programmability
 - Service Broker
 - Storage
 - Security
- Security
- Server Objects
- Replication
- PolyBase
- Management

100 %

Results Messages

ID NAME DEPT JOB YEARS SALARY COMM

ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
1	Sanders	20	Mgr	7	983575000000000	NULL
2	Pernal	20	Sales	8	781712500000000	612450000000
3	Marenghi	38	Mgr	5	775067500000000	NULL
4	O'Brien	38	Sales	6	780060000000000	846550000000
5	Hanes	15	Mgr	10	806598000000000	NULL
6	Quigley	38	Sales	NULL	668083000000000	650250000000
7	Rothman	15	Sales	7	765028300000000	115200000000
8	James	20	Clerk	NULL	435046000000000	128200000000
9	Koontz	42	Sales	6	380017500000000	138670000000
10	Plotz	42	Mgr	7	783528000000000	NULL
11	Ngan	15	Clerk	5	425082000000000	206600000000
12	Naughton	38	Clerk	NULL	429547500000000	180000000000
13	Yamaguchi	42	Clerk	6	405059000000000	75600000000
14	Fraye	51	Mgr	6	911500000000000	NULL
15	Williams	51	Sales	6	794565000000000	637650000000
16	Molnare	10	Mgr	7	829592000000000	NULL
17	Kermisch	15	Clerk	4	422585000000000	110100000000
18	Abrahams	38	Clerk	3	370097500000000	236500000000

Query executed successfully.

DESKTOP-9VT8FQQ (15.0 RTM) | DESKTOP-9VT8FQQ\QSS (61) : Test2 : 00:00:00 | 35 rows

Ready Type here to search

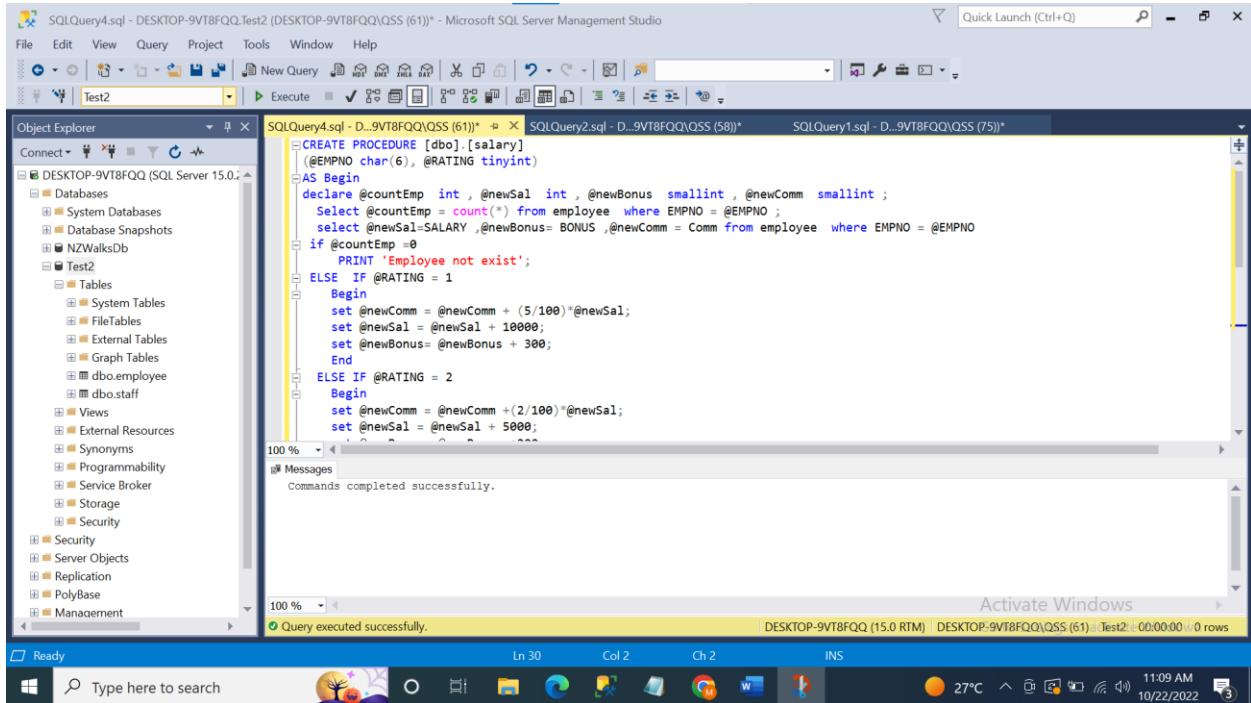
Ln 1 Col 9 Ch 9 INS

Activate Windows

27°C 10:56 AM 10/22/2022

Overview of the staff table

2. Stored procedure called SALARY for the EMPLOYEE table

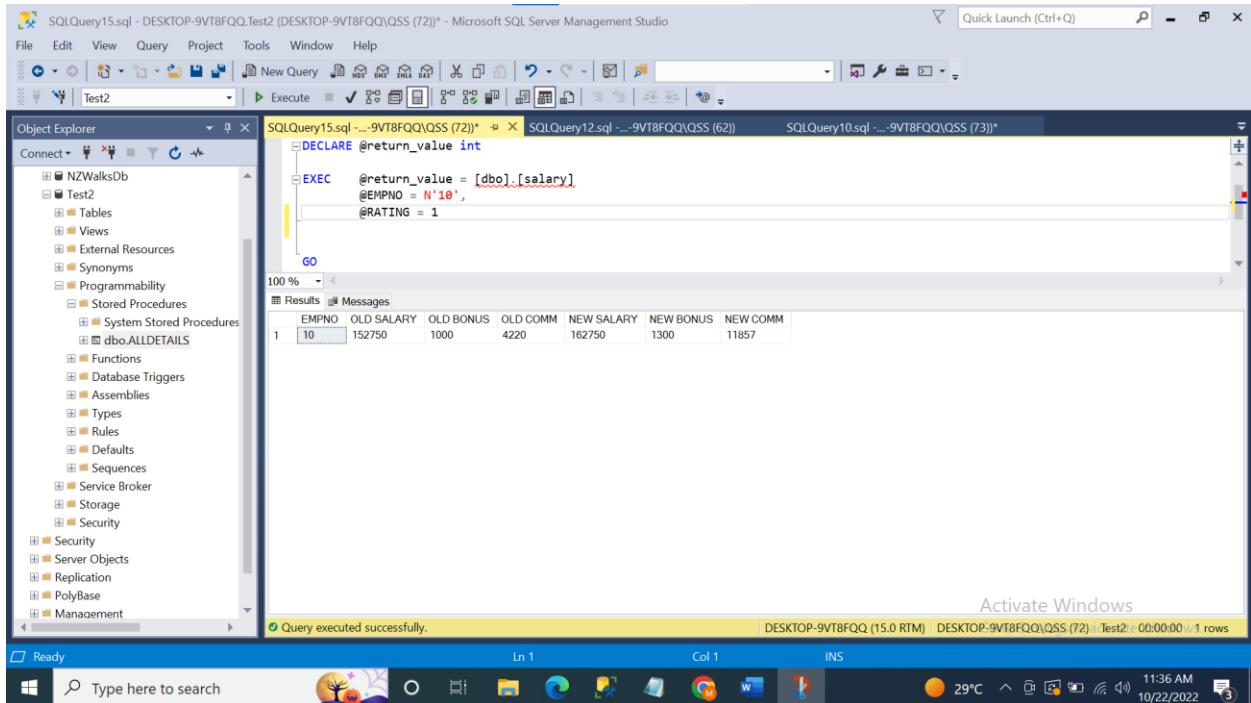


The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure, including the Test2 database which contains the dbo.employee and dbo.staff tables. The central pane displays the following T-SQL code for creating a stored procedure:

```
CREATE PROCEDURE [dbo].[salary]
(@EMPNO char(6), @RATING tinyint)
AS
Begin
    declare @countEmp int , @newSal int , @newBonus smallint , @newComm smallint ;
    Select @countEmp = count(*) from employee where EMPNO = @EMPNO ;
    select @newSal=SALARY ,@newBonus= BONUS ,@newComm = Comm from employee where EMPNO = @EMPNO
    if @countEmp < 0
        PRINT 'Employee not exist';
    ELSE IF @RATING = 1
        Begin
            set @newComm = @newComm + (5/100)*@newSal;
            set @newSal = @newSal + 10000;
            set @newBonus= @newBonus + 300;
        End
    ELSE IF @RATING = 2
        Begin
            set @newComm = @newComm +(2/100)*@newSal;
            set @newSal = @newSal + 5000;
        End
End
```

The Messages pane at the bottom shows the command completed successfully.

Salary Procedure



The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure, including the Test2 database which contains the dbo.salary stored procedure. The central pane displays the following T-SQL code for executing the stored procedure:

```
DECLARE @return_value int
EXEC  @return_value = [dbo].[salary]
@EMPNO = N'10',
@RATING = 1
GO
```

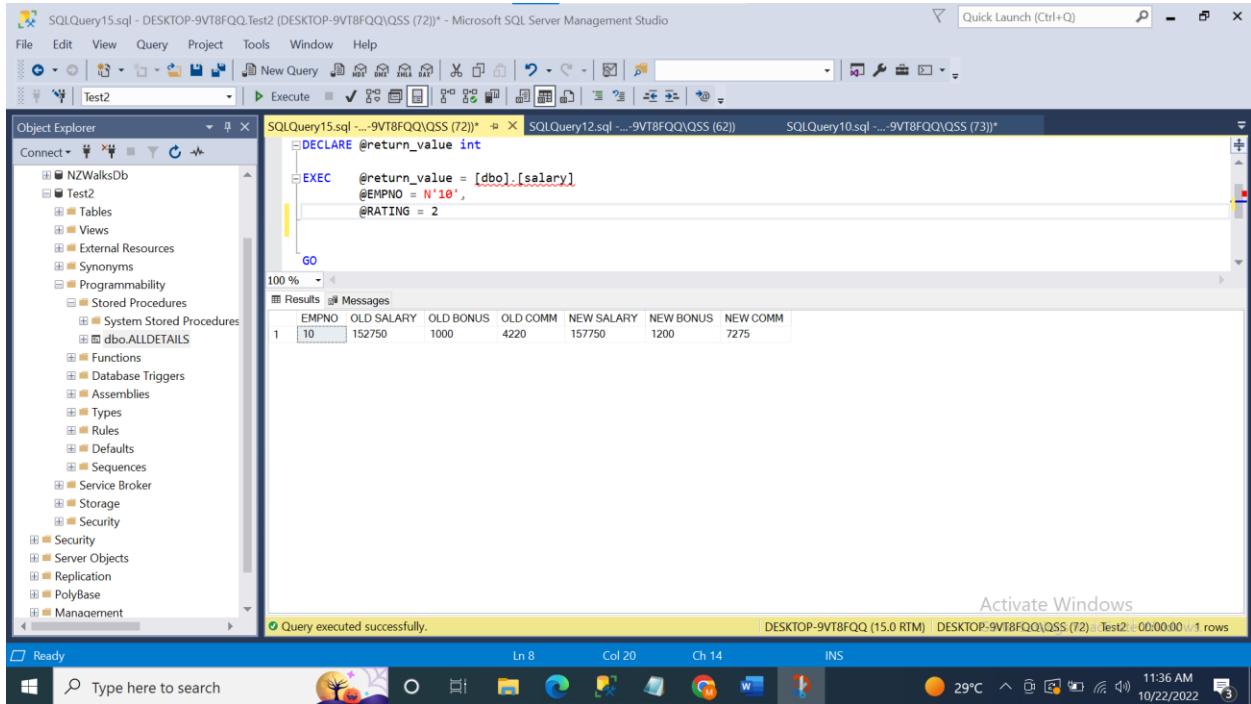
The Results pane shows the output of the query, which is a single row of data:

	EMPNO	OLD SALARY	OLD BONUS	OLD COMM	NEW SALARY	NEW BONUS	NEW COMM
1	10	152750	1000	4220	162750	1300	11857

The Messages pane at the bottom shows the command executed successfully.

Details of employees for rating 1

If the employee was rated a 1 – they receive a \$10,000 salary increase, additional \$300 in bonus and an additional 5% of salary as commission



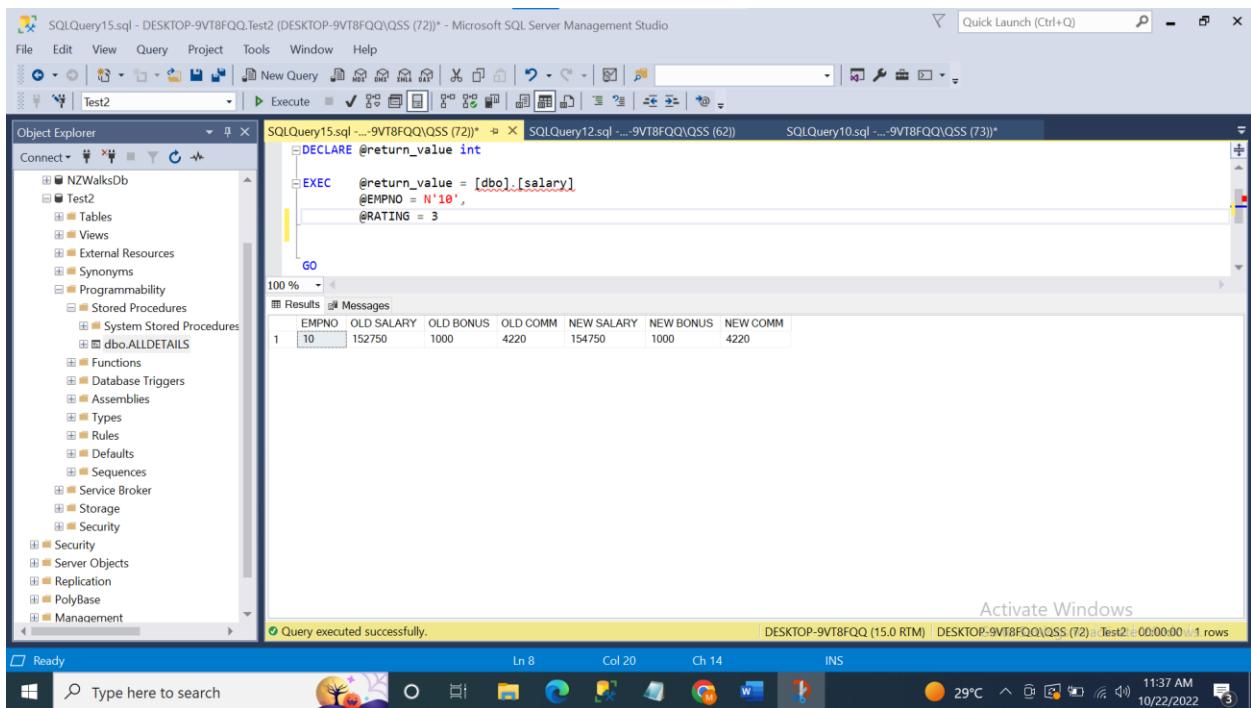
```

SQLQuery15.sql - DESKTOP-9VT8FQQ.Test2 (DESKTOP-9VT8FQQ\QSS (72)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
Execute New Query Object Explorer Test2 Results Messages
SQLQuery15.sql -...-9VT8FQQ\QSS (72)* SQLQuery12.sql -...-9VT8FQQ\QSS (62) SQLQuery10.sql -...-9VT8FQQ\QSS (73)*
DECLARE @return_value int
EXEC   @return_value = [dbo].[salary]
        @EMPNO = N'10',
        @RATING = 2
GO
100 %
Results Messages
EMPNO OLD SALARY OLD BONUS OLD COMM NEW SALARY NEW BONUS NEW COMM
1 10 152750 1000 4220 157750 1200 7275
1 row(s) affected.
Query executed successfully.
DESKTOP-9VT8FQQ (15.0 RTM) | DESKTOP-9VT8FQQ\QSS (72) : Test2 | 00:00:00 | 1 rows
Activate Windows
Ready Type here to search Ln 8 Col 20 Ch 14 INS
29°C 11:36 AM 10/22/2022

```

Details of employees for rating 2

If the employee was rated a 2 – they receive a \$5,000 salary increase, additional \$200 in bonus and an additional 2% of salary as commission.



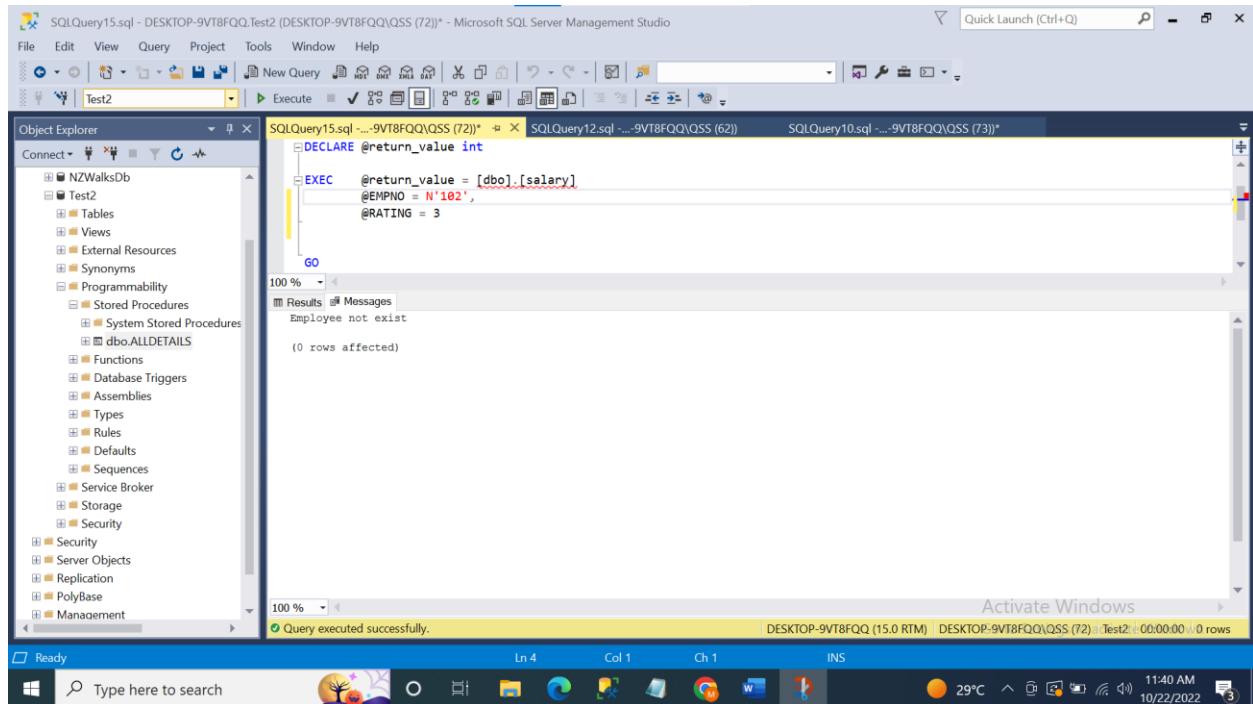
```

SQLQuery15.sql - DESKTOP-9VT8FQQ.Test2 (DESKTOP-9VT8FQQ\QSS (72)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
Execute New Query Object Explorer Test2 Results Messages
SQLQuery15.sql -...-9VT8FQQ\QSS (72)* SQLQuery12.sql -...-9VT8FQQ\QSS (62) SQLQuery10.sql -...-9VT8FQQ\QSS (73)*
DECLARE @return_value int
EXEC   @return_value = [dbo].[salary]
        @EMPNO = N'10',
        @RATING = 3
GO
100 %
Results Messages
EMPNO OLD SALARY OLD BONUS OLD COMM NEW SALARY NEW BONUS NEW COMM
1 10 152750 1000 4220 154750 1000 4220
1 row(s) affected.
Query executed successfully.
DESKTOP-9VT8FQQ (15.0 RTM) | DESKTOP-9VT8FQQ\QSS (72) : Test2 | 00:00:00 | 1 rows
Activate Windows
Ready Type here to search Ln 8 Col 20 Ch 14 INS
29°C 11:37 AM 10/22/2022

```

Details of employees for rating 3

If the employee was rated a 3 – they receive a \$2,000 salary increase with no change to their variable pay

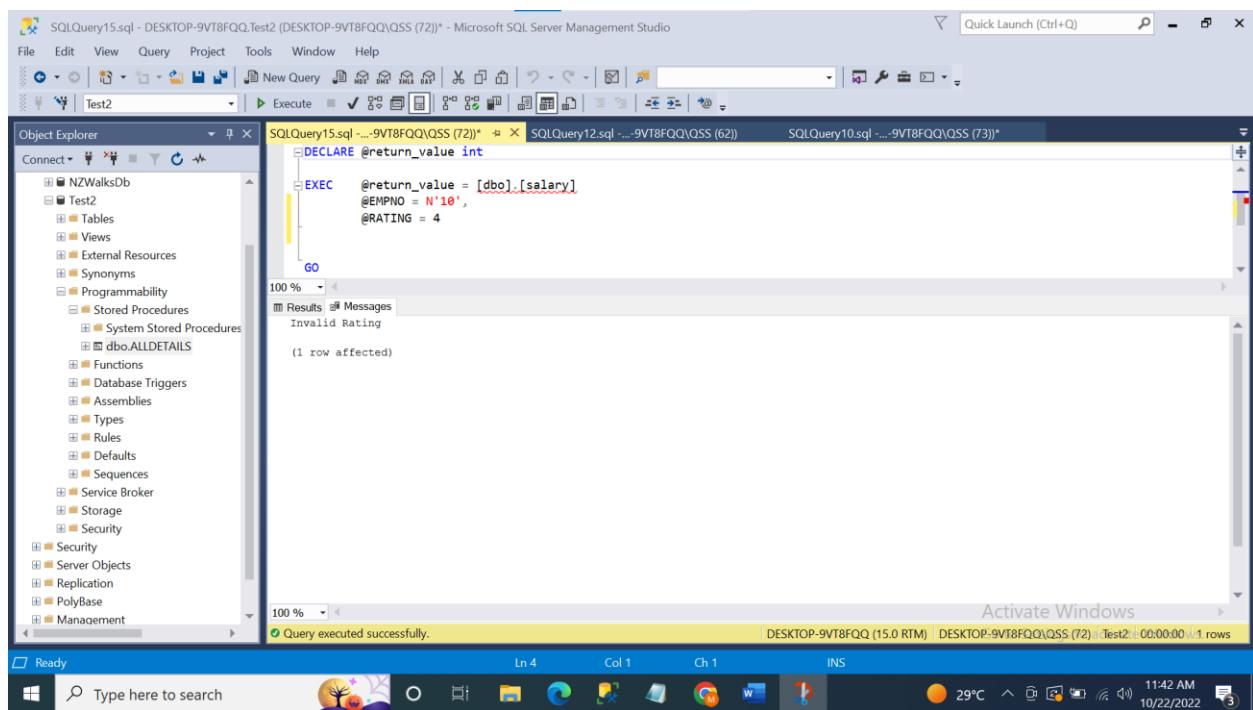


The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, a database named 'Test2' is selected. In the center pane, a query window displays the following T-SQL code:

```
SQLQuery15.sql - DESKTOP-9VT8FQQ.Test2 (DESKTOP-9VT8FQQ\QSS (72)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
Object Explorer Test2
SQLQuery15.sql -...-9VT8FQQ\QSS (72)* SQLQuery12.sql -...-9VT8FQQ\QSS (62) SQLQuery10.sql -...-9VT8FQQ\QSS (73)*
DECLARE @return_value int
EXEC   @return_value = [dbo].[salary]
        @EMPLNO = N'102',
        @RATING = 3
GO
```

The 'Messages' tab shows an error message: "Employee not exist" and "(0 rows affected)". Below the messages, a status bar indicates "Query executed successfully." and "DESKTOP-9VT8FQQ (15.0 RTM) | DESKTOP-9VT8FQQ\QSS (72) : Test2 | 00:00:00 | 0 rows".

Invalid Employee No



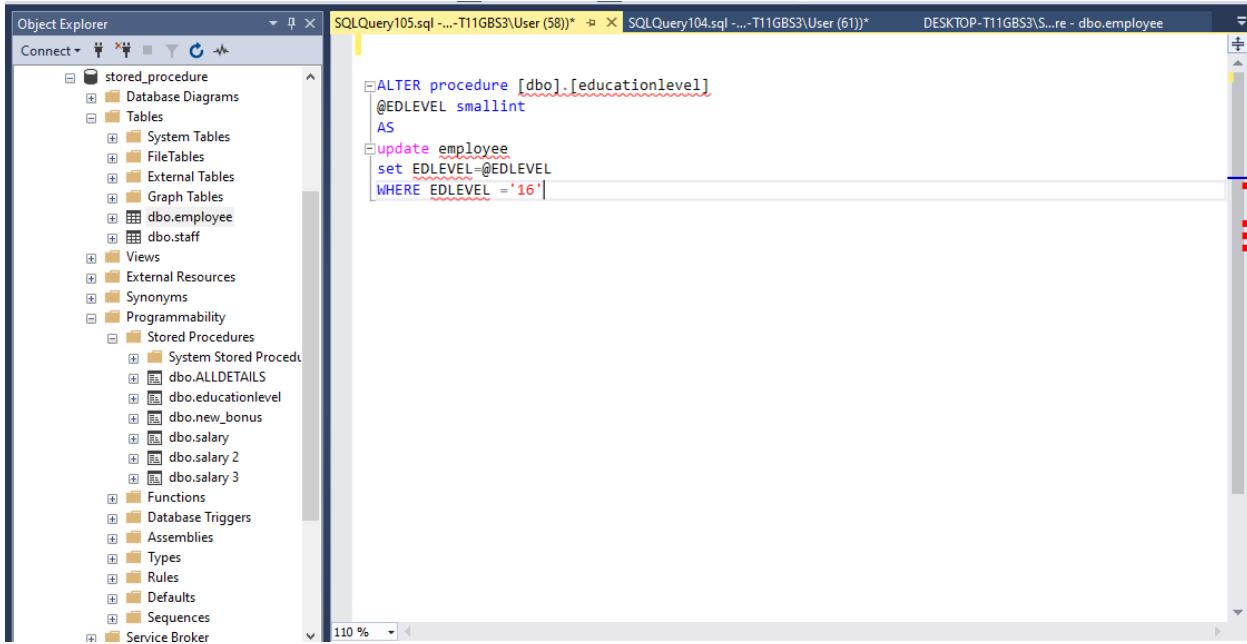
The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, a database named 'Test2' is selected. In the center pane, a query window displays the following T-SQL code:

```
SQLQuery15.sql - DESKTOP-9VT8FQQ.Test2 (DESKTOP-9VT8FQQ\QSS (72)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
Object Explorer Test2
SQLQuery15.sql -...-9VT8FQQ\QSS (72)* SQLQuery12.sql -...-9VT8FQQ\QSS (62) SQLQuery10.sql -...-9VT8FQQ\QSS (73)*
DECLARE @return_value int
EXEC   @return_value = [dbo].[salary]
        @EMPLNO = N'10',
        @RATING = 4
GO
```

The 'Messages' tab shows an error message: "Invalid Rating" and "(1 row affected)". Below the messages, a status bar indicates "Query executed successfully." and "DESKTOP-9VT8FQQ (15.0 RTM) | DESKTOP-9VT8FQQ\QSS (72) : Test2 | 00:00:00 | 1 rows".

Invalid Rating

3. Stored procedure for employee number and education level



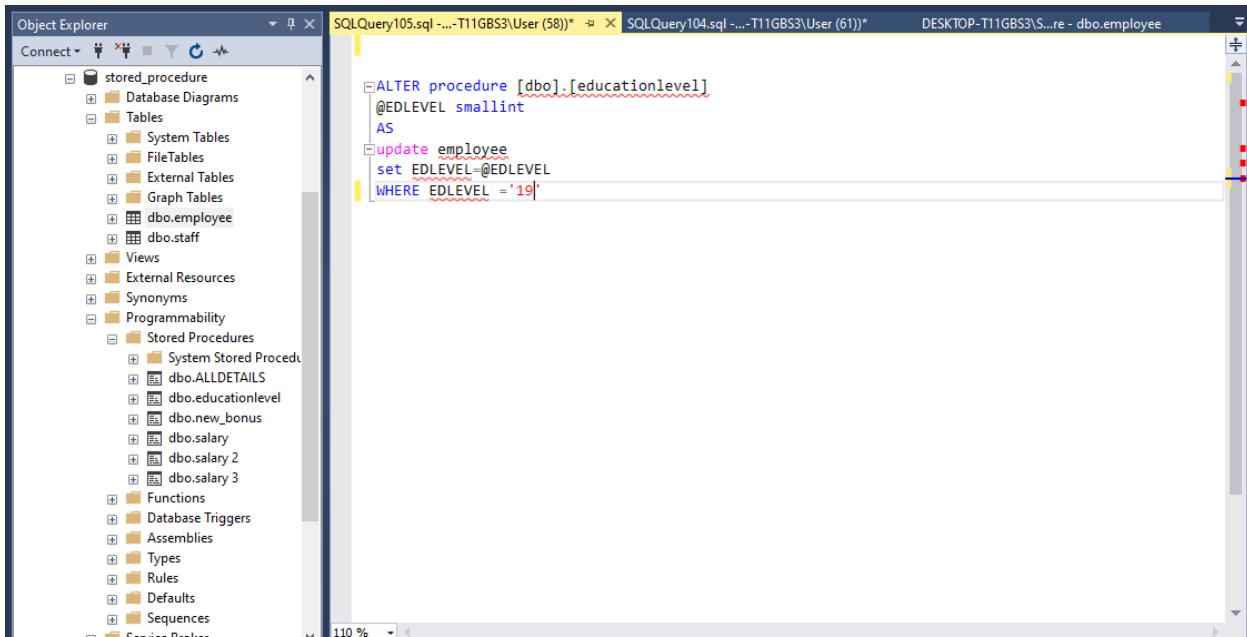
The screenshot shows the SQL Server Management Studio interface. The Object Explorer on the left lists various database objects under the 'stored_procedure' category. The central pane displays a query window with the following code:

```
ALTER procedure [dbo].[educationlevel]
@EDLEVEL smallint
AS
update employee
set EDLEVEL=@EDLEVEL
WHERE EDLEVEL ='16'
```

The word 'EDLEVEL' is highlighted in red, indicating a syntax error.

Education level 16

“H” (for high school diploma) – and – this will update the edlevel to 16



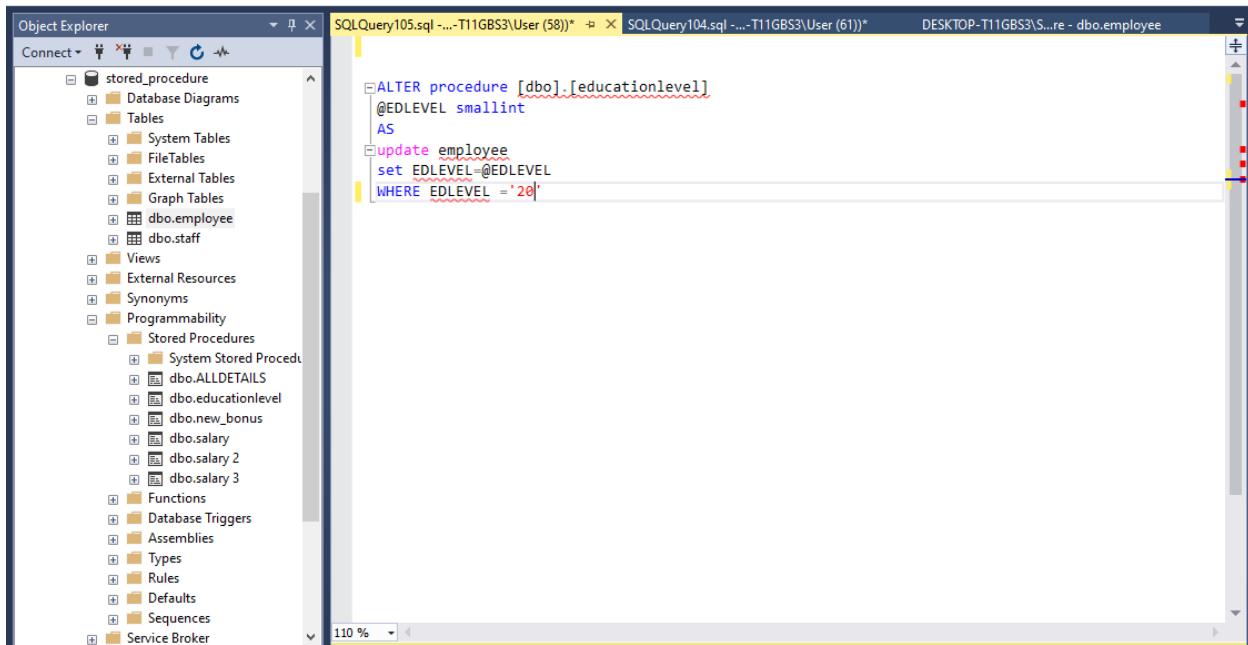
The screenshot shows the SQL Server Management Studio interface. The Object Explorer on the left lists various database objects under the 'stored_procedure' category. The central pane displays a query window with the following code:

```
ALTER procedure [dbo].[educationlevel]
@EDLEVEL smallint
AS
update employee
set EDLEVEL=@EDLEVEL
WHERE EDLEVEL = '19'
```

The word 'EDLEVEL' is now correctly written without the trailing apostrophe, and the code is valid.

Education level 19

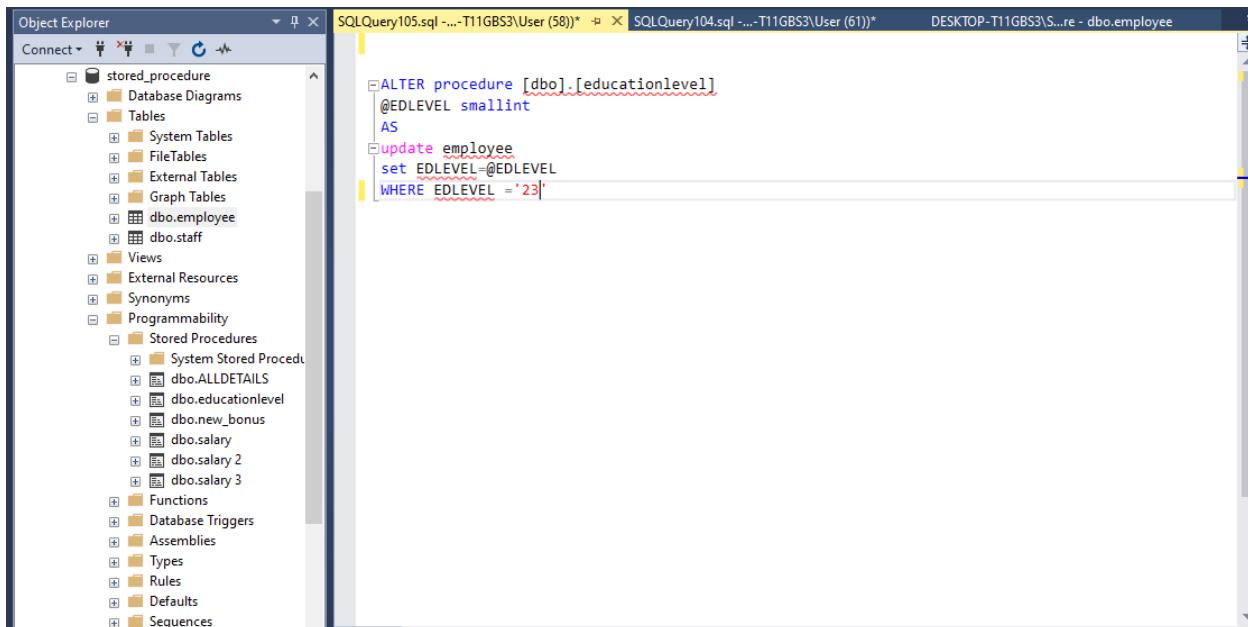
“C” (for college diploma) – and – this will update the edlevel to 19



```
Object Explorer
SQLQuery105.sql -...-T11GBS3\User (58)* SQLQuery104.sql -...-T11GBS3\User (61)* DESKTOP-T11GBS3\S...re - dbo.employee
ALTER procedure [dbo].[educationlevel]
@EDLEVEL smallint
AS
update employee
set EDLEVEL=@EDLEVEL
WHERE EDLEVEL = '20'
```

Education level 20

“U” (for university degree) – and – this will update the edlevel to 20.



```
Object Explorer
SQLQuery105.sql -...-T11GBS3\User (58)* SQLQuery104.sql -...-T11GBS3\User (61)* DESKTOP-T11GBS3\S...re - dbo.employee
ALTER procedure [dbo].[educationlevel]
@EDLEVEL smallint
AS
update employee
set EDLEVEL=@EDLEVEL
WHERE EDLEVEL = '23'
```

Education level 23

“M” (for masters) – and – this will update the edlevel to 23

The screenshot shows the SQL Server Management Studio interface. On the left, the Object Explorer pane displays a tree view of database objects under the 'stored_procedure' node, including tables like 'dbo.employee' and 'dbo.staff', and stored procedures like 'dbo.educationlevel'. In the center, the SQL Query Editor pane contains the following T-SQL code:

```
ALTER procedure [dbo].[educationlevel]
@EDLEVEL smallint
AS
update employee
set EDLEVEL=@EDLEVEL
WHERE EDLEVEL = '25'
```

Education level 25

“P” (for PhD) – and – this will update the edlevel to 25

The screenshot shows the SQL Server Management Studio interface. On the left, the Object Explorer pane displays a tree view of database objects under the 'stored_procedure' node, including tables like 'dbo.employee' and 'dbo.staff', and stored procedures like 'dbo.educationlevel'. In the center, the SQL Query Editor pane contains the following T-SQL code:

```
USE [stored_procedure]
GO

DECLARE @return_value int

EXEC    @return_value = [dbo].[new_bonus]

SELECT  'Return Value' = @return_value
GO
```

Below the code, the Results tab shows a table with the following data:

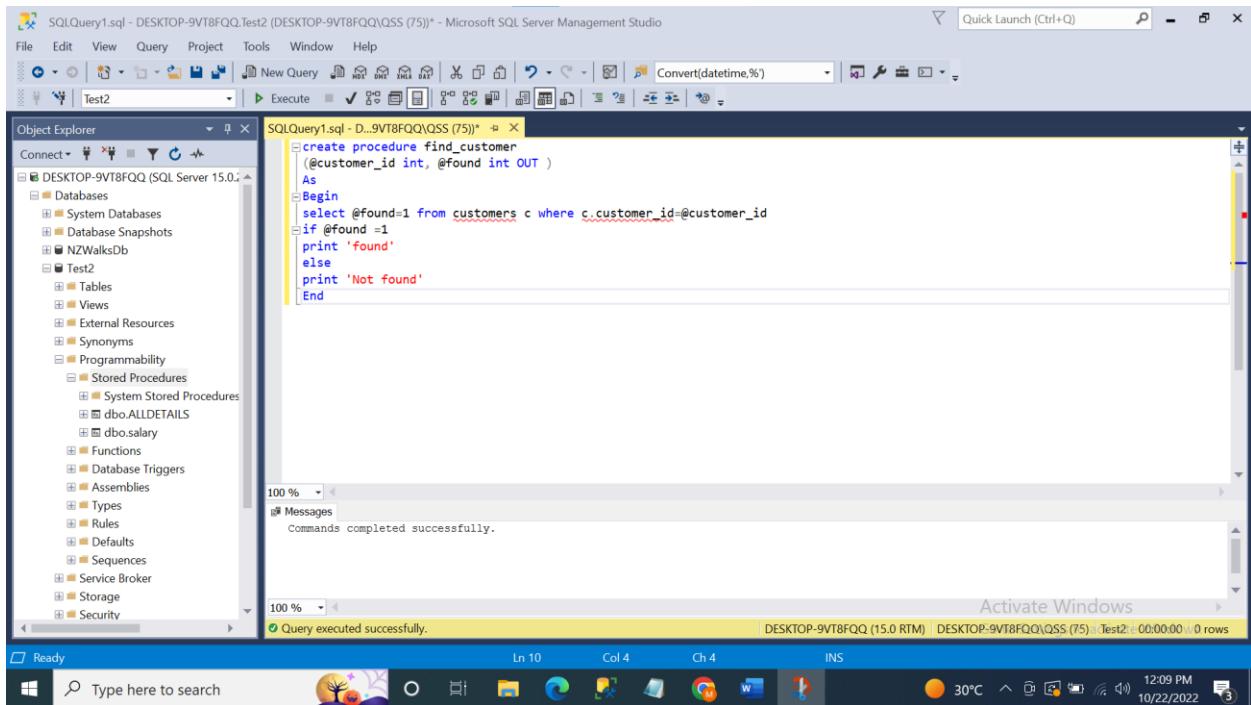
	BONUS	COMM	SALARY	RATING
1	1000	4220	152750	1
2	500	0	10000	2
3	800	3060	98250	3
4	800	3214	80175	1
5	500	0	10000	2
6	700	2893	96170	3
7	600	2380	89750	3
8	500	0	10000	2

A message window at the bottom shows a single row with 'Return Value' and '1'.

Display Order

4. Find Customer Procedure

This procedure has an input parameter to receive the customer ID and an output parameter named found. This procedure looks for the given customer ID in the database. If the customer exists, it sets the variable found to 1. Otherwise, the found variable is set to 0.



The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a connection to 'DESKTOP-9VT8FQQ (SQL Server 15.0)'. Under the 'Test2' database, the 'Programmability' node is expanded, showing 'Stored Procedures'. A new stored procedure named 'find_customer' is being created in the 'Script Results' pane. The script contains the following T-SQL code:

```
create procedure find_customer
(@customer_id int, @found int OUT )
As
Begin
select @found=1 from customers c where c.customer_id=@customer_id
End
if @found =1
print 'found'
else
print 'Not found'
```

The 'Messages' pane at the bottom shows the command completed successfully. The status bar at the bottom right indicates the system is at 30°C, the time is 12:09 PM, and the date is 10/22/2022.

5. Find Product Procedure By Product ID

This procedure has an input parameter to receive the product ID and an output parameter named price. This procedure looks for the given product ID in the database. If the product exists, it stores the product's list_price in the variable price. Otherwise, the price variable is set to 0.

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, a database named 'Test2' is selected. In the center pane, a stored procedure named 'find_product' is displayed:

```
create procedure find_product (@product_id int, @price int OUT )
As
Begin
declare @count smallint;
Select @count = count(*) from products p where p.product_id = @product_id ;
if @count =0
    PRINT 'Product Id not exist';
ELSE
select price = p.list_price from products p where p.product_id = @product_id
End
```

The status bar at the bottom indicates: DESKTOP-9VT8FQQ (15.0 RTM) | DESKTOP-9VT8FQQ\QSS (75) | Test2 | 00:00:00 | 0 rows.

6. Add Order Procedure

This procedure has an input parameter to receive the customer ID and an output parameter named new_order_id.

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "Test2 (DESKTOP-9VT8FQQ\QSS (61)) - Microsoft SQL Server Management Studio". The toolbar includes standard options like New Query, Execute, and various icons. The main window displays a T-SQL script for creating a stored procedure:

```
GO

create procedure [dbo].[add_order] (@customer_id int, @new_order_id int OUT )
As
declare @count smallint ;
Select @count = count(*) from customers where customer_id=@customer_id;
if @count = 0
PRINT 'Customer Id not exist';
ELSE
Begin
declare @max_order_id int
select @max_order_id=max(o.order_id) from orders o;
set @new order id = @max order id+1;
```

The script ends with a successful message in the Messages pane:

```
Commands completed successfully.
```

At the bottom, the status bar shows "Query executed successfully.", "DESKTOP-9VT8FQQ (15.0 RTM)", "DESKTOP-9VT8FQQ\QSS (61)", "Test2", "0:00:00", "0 rows", "Ln 11", "Col 6", "Ch 6", "INS", and "3:36 PM".

7. Add Order_Item Procedure

This procedure has five IN parameters. It stores the values of these parameters to the table `order_items`.

The screenshot shows the Microsoft SQL Server Management Studio interface. In the center, there is a query editor window titled 'SQLQuery24.sql -...9VT8FQQ\QSS (57)*'. The code inside the editor is a stored procedure:

```
@productId int,  
@quantity decimal(8,2),  
@price decimal(8,2)  
As  
Begin  
declare @count smallint ;  
Select @count = count(*) from orders where order_id=@orderId;  
if @count =0  
PRINT 'Order Id not exist';  
ELSE  
INSERT INTO [dbo].[order_items]([order_id],[item_id],[product_id],[quantity],[unit_price])  
VALUES(@orderId,@itemId,@productId,@quantity,@price)  
End
```

Below the code, the status bar shows 'Commands completed successfully.' and '110 %' zoom level.

In the bottom right corner of the status bar, there is an 'Activate Windows' button.

8. Display Order Procedure for specific Order ID

This procedure has an input parameter to receive the order ID and no output parameters. This procedure will display the order items associated with a particular order ID.

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists database objects like dbo.employee, dbo.orders, and dbo.order_items. The main window displays a T-SQL script for a stored procedure named 'display_order'. The script uses a cursor to count orders and then retrieves details for each order item. The status bar at the bottom indicates the command was executed successfully.

```

create procedure display_order(@orderId int)
As
Begin
    declare @count smallint ;
    Select @count = count(*) from orders where order_id=@orderId;
    if @count =0
        PRINT 'Order Id not exist';
    ELSE
        Begin
            select order_id as [Order ID],customer_id as [Customer ID] from orders where order_id=@orderId;
            select item_id as [Item ID],product_id as [Product ID],quantity as [Quantity],unit_price as [Price] from order_item
            select sum(unit_price) as [Total Price] from order_items where order_id=@orderId;
        End
    End

```

9. Master Procedure

This procedure is a master procedure for four of the five procedures above.

If task = 1, then, call find_customer(parm1)

If task = 2, then, call find_product(parm1)

If task = 3, then, call add_order(parm1)

If task = 4, then, call display-order(parm1)

The screenshot shows the Microsoft SQL Server Management Studio interface. In the center, there is a code editor window displaying a T-SQL script for a stored procedure named 'master_proc'. The script uses IF-ELSE logic to call other procedures based on the value of the parameter '@task'. The code is as follows:

```
create procedure master_proc (@task int, @parm1 int)
As
Begin
    declare @found int ;
    Set @found=0;
    If @task = 1
        EXEC find_customer @parm1 ,@found
    Else If @task = 2
        EXEC find_product @parm1, @found
    Else If @task = 3
        Exec add_order @parm1
    Else If @task = 4
        EXEC display_order @parm1
    Else
        print 'Invalid task number'
End
```

Below the code editor, the 'Messages' pane shows the output: 'Commands completed successfully.'.

At the bottom of the screen, the Windows taskbar is visible, showing various icons and system information. The taskbar includes the following details:

- Activate Windows
- Query executed successfully.
- DESKTOP-9VT8FQQ (15.0 RTM)
- DESKTOP-9VT8FQQ\QSS (75)
- Test2
- 00:00:00
- 0 rows
- 31°C
- 2:22 PM
- 10/22/2022

Call other procedures using Master Procedure

- **find_customer – with a valid customer ID**

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "Test2 (DESKTOP-9VT8FQQ\QSS (68)) - Microsoft SQL Server Management Studio". The toolbar includes standard options like New Query, Execute, and Save. The main window displays a query window titled "SQLQuery19.sql" (row 68). The code in the window is:

```
USE [Test2]
GO

DECLARE @return_value int
***** CALL CUSTOMER PROCEDURE for VALID ****/
EXEC    @return_value = [dbo].[master_proc]
        @task = 1,
        @parm1 = 177

--SELECT      'Return Value' = @return_value
GO
```

The status bar at the bottom shows "Query executed successfully." and "110 %". The taskbar at the bottom right includes icons for Start, Task View, File Explorer, Edge, and File Explorer.

- **find_customer – with an invalid customer ID**

s2 (DESKTOP-9VT8FQQ\QSS (68)) - Microsoft SQL Server Management Studio

File Window Help

New Query MDX DML XML DAX | Convert(datetime,%)

Execute ✓ | SQL | Script | Grid | Table | List | Advanced | Parameters | Parameters | @ |

SQLQuery19.sql -...-9VT8FQQ\QSS (68)* SQLQuery18.sql -...-9VT8FQQ\QSS (59)* SQLQuery1.sql - D...9VT8FQQ\QSS (75)*

```
USE [Test2]
GO

DECLARE @return_value int
***** CALL CUSTOMER PROCEDURE for INVALID ****/
EXEC    @return_value = [dbo].[master_proc]
        @task = 1,
        @parm1 = 1776

--SELECT      'Return Value' = @return_value
GO
```

110 %

Messages Not found

Activate Windows

Query executed successfully.

DESKTOP-9VT8FQQ (15.0 RTM) DESKTOP-9VT8FQQ\QSS (68) Test2 00:00:00 0 rows

In 5 Col 39 Ch 39 INS

- **find_product – with a valid Product ID**

The screenshot shows the Microsoft SQL Server Management Studio interface. The query window contains the following T-SQL code:

```

USE [Test2]
GO

DECLARE @return_value int
/****** CALL find_product PROCEDURE for VALID and print price of product *****/
EXEC  @return_value = [dbo].[master_proc]
@task = 2,
@parm1 = 1

--SELECT      'Return Value' = @return_value
GO

```

The results pane shows a single row with the value 640.99 under the column 'price'.

Status bar message: Query executed successfully.

- **find_product –with an Invalid ProductID**

The screenshot shows the Microsoft SQL Server Management Studio interface. The query window contains the same T-SQL code as the previous screenshot, but with a different parameter value:

```

USE [Test2]
GO

DECLARE @return_value int
/****** CALL find_product PROCEDURE for VALID and print price of product *****/
EXEC  @return_value = [dbo].[master_proc]
@task = 2,
@parm1 = 1455

--SELECT      'Return Value' = @return_value
GO

```

The results pane shows an error message: "Product Id not exist".

Status bar message: Query executed successfully.

- **add_order – with a valid customer ID**

The screenshot shows the Microsoft SQL Server Management Studio interface. The query window displays the following T-SQL code:

```

USE [Test2]
GO

DECLARE @return_value int
/****** CALL add_order PROCEDURE for VALID Customer ID *****/
EXEC @return_value = [dbo].[master_proc]
@task = 3,
@parm1 = 1

SELECT 'Return Value' = @return_value
GO

```

The output pane shows the result of the execution:

```

(1 row affected)

```

- **add_order – with an invalid customer ID**

The screenshot shows the Microsoft SQL Server Management Studio interface. The query window displays the same T-SQL code as the previous screenshot, but with a different customer ID:

```

USE [Test2]
GO

DECLARE @return_value int
/****** CALL add_order PROCEDURE for Invalid VALID Customer ID *****/
EXEC @return_value = [dbo].[master_proc]
@task = 3,
@parm1 = 789

SELECT 'Return Value' = @return_value
GO

```

The output pane shows the error message:

```

Customer Id not exist
(1 row affected)
(0 rows affected)

```

A status bar at the bottom indicates:

- Query executed successfully.
- DESKTOP-9VT8FQQ (15.0 RTM)
- DESKTOP-9VT8FQQ\QSS (68)
- Test2
- 00:00:00
- 1 rows

- **display_order – with a valid order ID which has at least 5 order items**

QQ.Test2 (DESKTOP-9VT8FQQ\QSS (65)) - Microsoft SQL Server Management Studio

Tools Window Help

Convert(datetime,%)

Execute

SQLQuery22.sql -...-9VT8FQQ\QSS (69)* SQLQuery21.sql -...-9VT8FQQ\QSS (65)* SQLQuery19.sql -...-9VT8FQQ\QSS (68)*

```

DECLARE @return_value int
EXEC   @return_value = [dbo].[display_order]
        @orderId = 1

SELECT  'Return Value' = @return_value
GO

```

10 %

Results Messages

Order ID	Customer ID
1	1
2	4
3	
4	
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12

Item ID	Product ID	Quantity	Price
5	5	138	131.00
6	6	126	95.00
7	7	30	41.00
8	8	170	129.00
9	9	196	139.00
10	10	64	147.00
11	11	169	105.00
12	12	226	103.00

Total Price
10809.73

Return Value
0

Activate Windows

Query executed successfully.

DESKTOP-9VT8FQQ (15.0 RTM) DESKTOP-9VT8FQQ\QSS (65) Test2 00:00:00 16 rows

Ln 1 Col 1 Ch 1 INS

31°C 2:50 PM 10/22/2022

- **display_order – with an invalid order ID**

Test2 (DESKTOP-9VT8FQQ\QSS (68)) - Microsoft SQL Server Management Studio

Tools Window Help

Convert(datetime,%)

Execute

SQLQuery22.sql -...-9VT8FQQ\QSS (69)* SQLQuery21.sql -...-9VT8FQQ\QSS (65)* SQLQuery19.sql -...-9VT8FQQ\QSS (68)*

```

DECLARE @return_value int
***** CALL display_order PROCEDURE for Invalid VALID order ID ****/
EXEC   @return_value = [dbo].[master_proc]
        @task = 4,
        @parm1 = 789

--SELECT  'Return Value' = @return_value
GO

```

110 %

Results Messages

Order Id not exist

110 %

Query executed successfully.

DESKTOP-9VT8FQQ (15.0 RTM) DESKTOP-9VT8FQQ\QSS (68) Test2 00:00:00 0 rows

Ln 8 Col 21 Ch 15 INS

31°C 2:53 PM 10/22/2022

- **add_order_item** – should execute successfully 5 times

The screenshot shows the Microsoft SQL Server Management Studio interface. In the center pane, there is a query window containing the following T-SQL code:

```

DECLARE @return_value int
EXEC   @return_value = [dbo].[add_order_item]
        @orderId = 2,
        @itemId = 1,
        @productId = 1,
        @quantity = 20,
        @price = 29
SELECT  'Return Value' = @return_value
GO

```

Below the code, the results pane displays the output:

```

Order Id not exist
Order inserted successfully
(1 row affected)

```

In the status bar at the bottom, it says "Query executed successfully." and shows system information like "DESKTOP-9VT8FQQ (15.0 RTM) | DESKTOP-9VT8FQQ\QSS (52) | Test2 | 00:00:00 | 1 rows".

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads 'DESKTOP-9VT8FQQ\QSS (52)* - Microsoft SQL Server Management Studio'. The toolbar includes various icons for file operations like New Query, Open, Save, and Print. A dropdown menu shows 'Convert(datetime,%')' as the current format. Below the toolbar is a ribbon with tabs for Execute, View, Tools, Options, and Help. The main window displays a T-SQL script in the 'Query' tab:

```
DECLARE @return_value int
EXEC    @return_value = [dbo].[add_order_item]
        @orderId = 2,
        @itemId = 2,
        @productId = 1,
        @quantity = 6.67,
        @price = 45.9
SELECT  'Return Value' = @return_value
GO
```

The 'Results' pane at the bottom shows the output of the query:

```
Order Id not exist
Order inserted successfully

(1 row affected)
```



Test2 (DESKTOP-9VT8FQQ\QSS (52))* - Microsoft SQL Server Management Studio

File Tools Window Help

New Query DTS DML XML DAX | Convert(datetime,%)

Execute

SQLQuery26.sql -...9VT8FQQ\QSS (52)* SQLQuery24.sql -...9VT8FQQ\QSS (57)* SQLQuery19.sql -...9VT8FQQ\QSS (68)*

```
DECLARE @return_value int

EXEC    @return_value = [dbo].[add_order_item]
        @orderId = 3,
        @itemId = 2,
        @productId = 1,
        @quantity = 56,
        @price = 89.9

SELECT  'Return Value' = @return_value

GO
```

110 %

Results Messages

Order Id not exist
Order inserted successfully
(1 row affected)

Activate Windows

Query executed successfully.

DESKTOP-9VT8FQQ (15.0 RTM) DESKTOP-9VT8FQQ\QSS (52) Test2 00:00:00 1 rows

Ln 5 Col 1 Ch 1 INS

Test2 (DESKTOP-9VT8FQQ\QSS (52))* - Microsoft SQL Server Management Studio

File Tools Window Help

New Query DTS DML XML DAX | Convert(datetime,%)

Execute

SQLQuery26.sql -...9VT8FQQ\QSS (52)* SQLQuery24.sql -...9VT8FQQ\QSS (57)* SQLQuery19.sql -...9VT8FQQ\QSS (68)*

```
DECLARE @return_value int

EXEC    @return_value = [dbo].[add_order_item]
        @orderId = 22,
        @itemId = 2,
        @productId = 1,
        @quantity = 689,
        @price = 34.9

SELECT  'Return Value' = @return_value

GO
```

110 %

Results Messages

Order Id not exist
Order inserted successfully
(1 row affected)

Activate Windows

Query executed successfully.

DESKTOP-9VT8FQQ (15.0 RTM) DESKTOP-9VT8FQQ\QSS (52) Test2 00:00:00 1 rows

Ln 5 Col 1 Ch 1 INS

- **add_order_item – should execute with an invalid order**

st2 (DESKTOP-9VT8FQQ\QSS (52))* - Microsoft SQL Server Management Studio

File Window Help

New Query MDW DML XML DAX | Convert(datetime,%) | Execute

SQLQuery28.sql -...9VT8FQQ\QSS (64)* SQLQuery26.sql -...9VT8FQQ\QSS (52)* SQLQuery24.sql -...9VT8FQQ\QSS (57)*

```
DECLARE @return_value int  
EXEC    @return_value = [dbo].[add_order_item]  
        @orderId = 3,  
        @itemId = 2,  
        @productId = 1,  
        @quantity = 689,  
        @price = 34.9  
SELECT  'Return Value' = @return_value  
GO
```

110 %

Results Messages Order Id not exist
(1 row affected)

Activate Windows

Query executed successfully.

DESKTOP-9VT8FQQ (15.0 RTM) DESKTOP-9VT8FQQ\QSS (52)\Test2 00:00:00 1 rows

Ln 4 Col 1 Ch 1 INS 31°C 3:28 PM