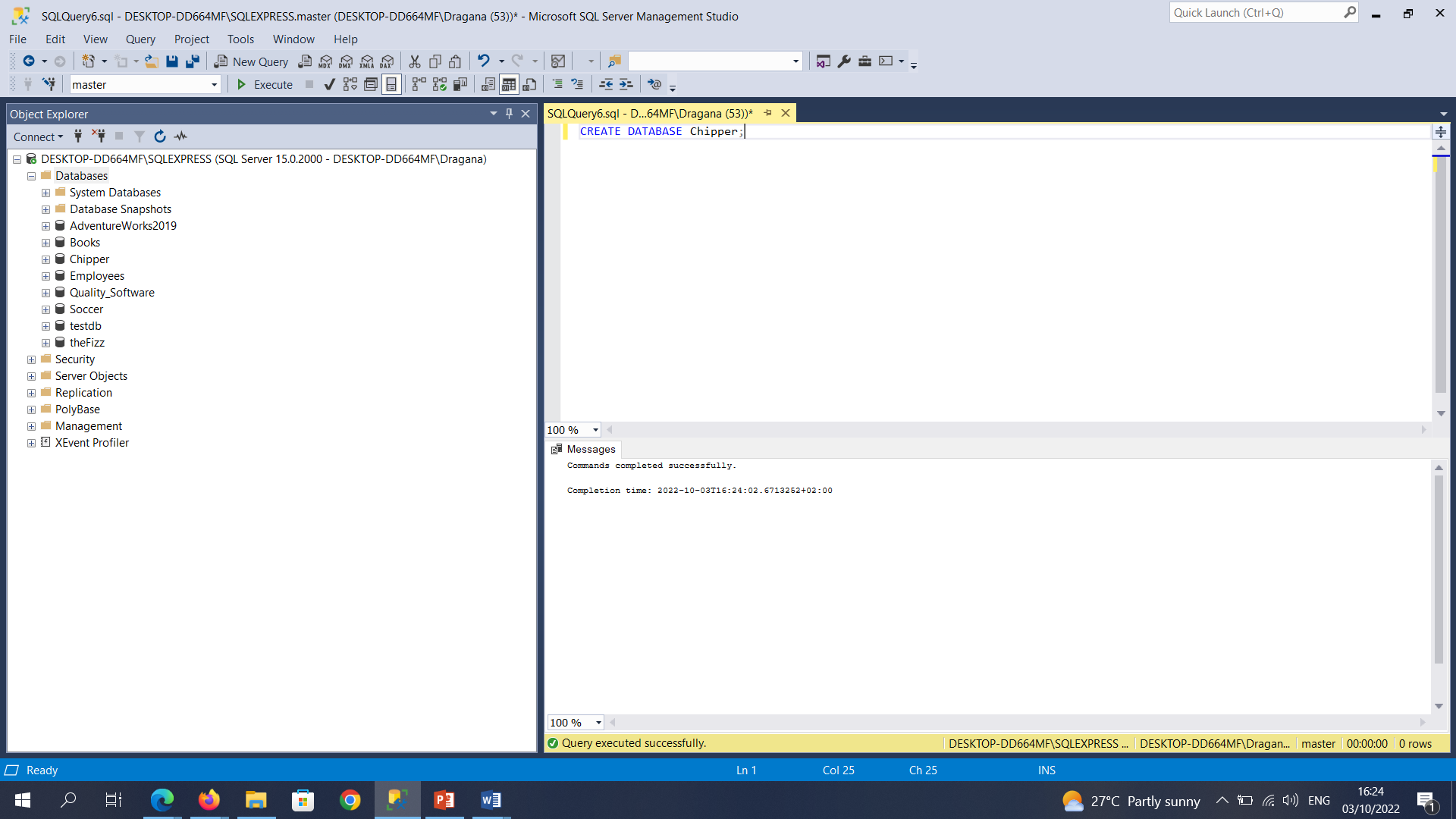
Exercise 4:

**Part A:**  
• Create a database named Chipper.

CREATE DATABASE Chipper;



**Part B:**  
• Create a table named Meals. The table should have the following fields.

Field Data Type  
id int  
name varchar(50)  
price money  
size varchar(50)

CREATE TABLE Meals (

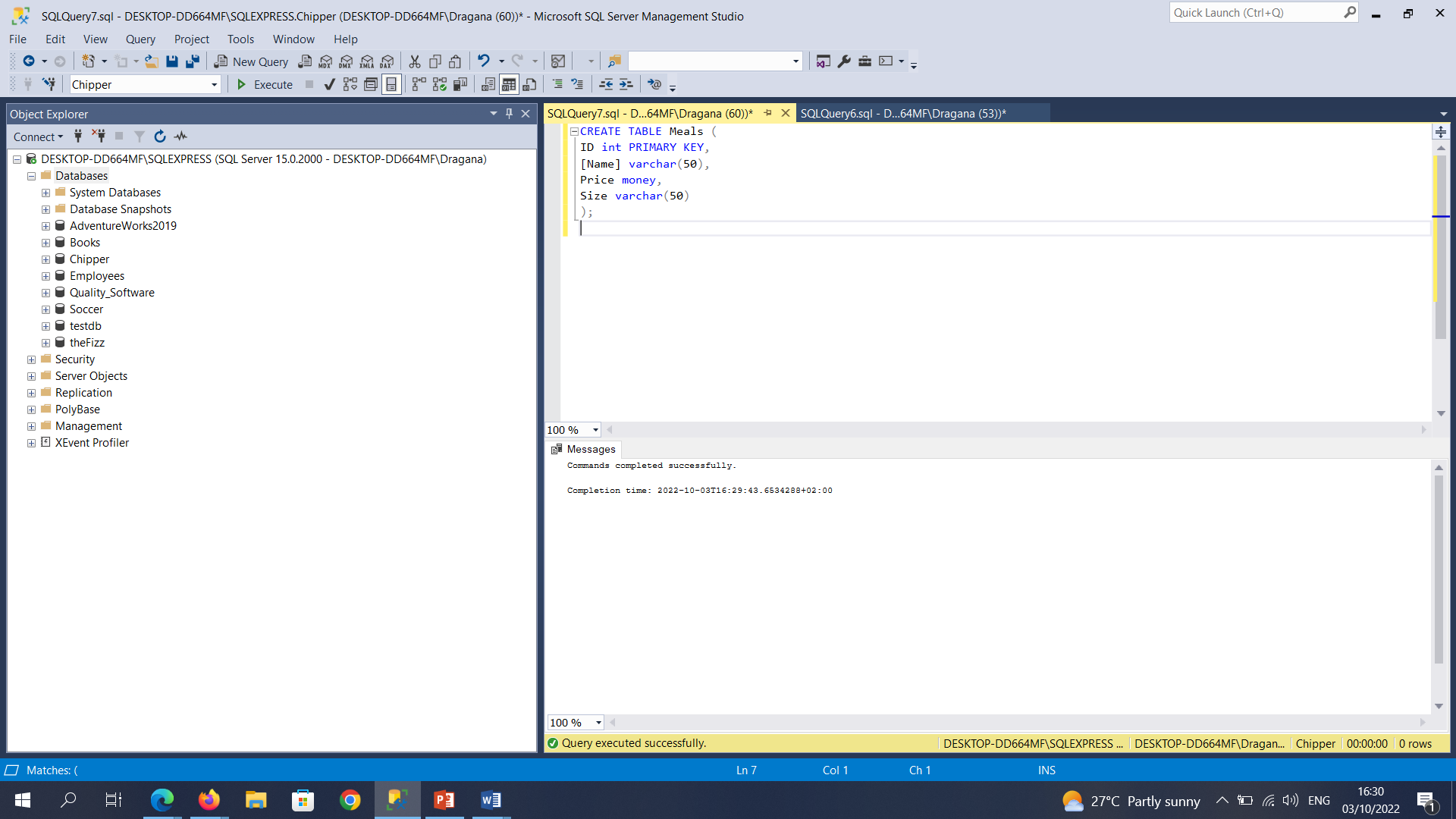
ID int PRIMARY KEY,

[Name] varchar(50),

Price money,

Size varchar(50)

);



**Part C:**  
• Insert three records into the table.

ID Name Price Size  
1 Garlic Bread 2.50 Large  
2 Cheese Burger 1.80 Large  
3 Small Fries 1.30 Small

INSERT INTO Meals (ID, [Name], Price, Size)

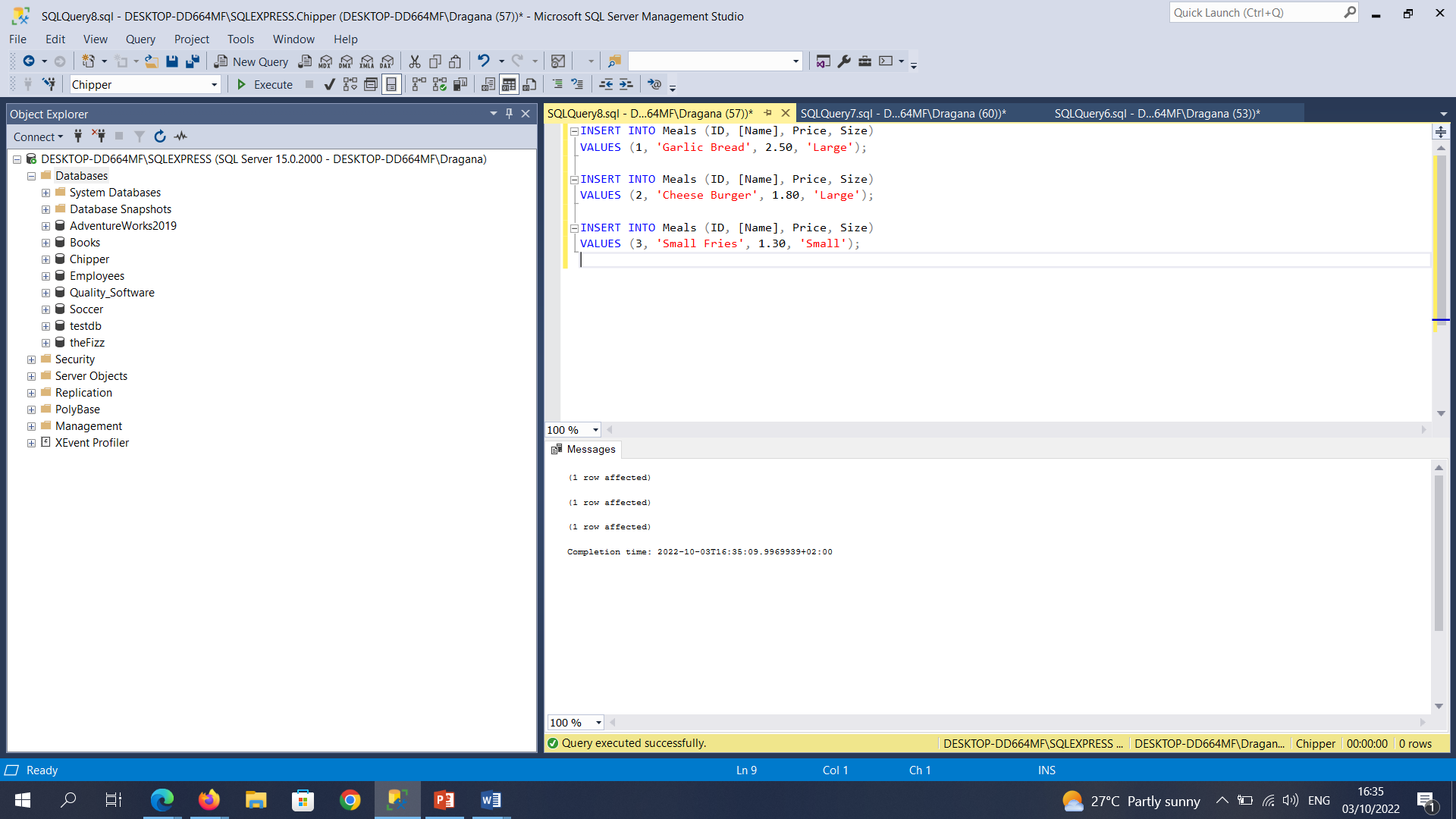
VALUES (1, 'Garlic Bread', 2.50, 'Large');

INSERT INTO Meals (ID, [Name], Price, Size)

VALUES (2, 'Cheese Burger', 1.80, 'Large');

INSERT INTO Meals (ID, [Name], Price, Size)

VALUES (3, 'Small Fries', 1.30, 'Small');



**Part D:**

* Write a SQL statement, which creates a stored procedure that shows details of records stored in the Meals table. Execute the stored procedure.

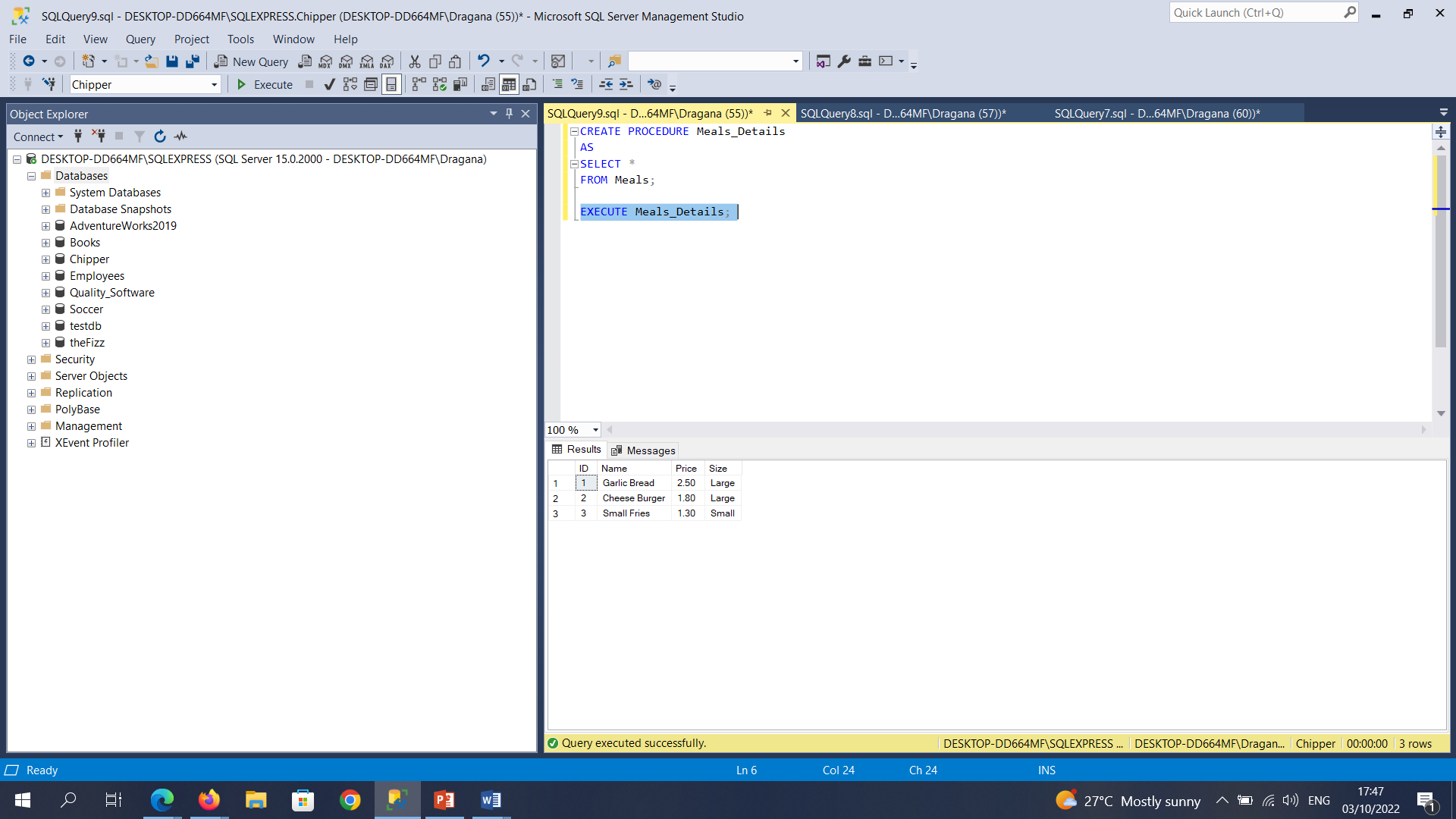
CREATE PROCEDURE Meals\_Details

AS

SELECT \*

FROM Meals;

EXEC Meals\_Details;



**Part E:**

* Create another stored procedure that only displays details of meals of a certain size.  
  The size of the meal should be passed into the stored procedure.

CREATE PROCEDURE Meals\_of\_Certain\_Sizes

@Size varchar(50)

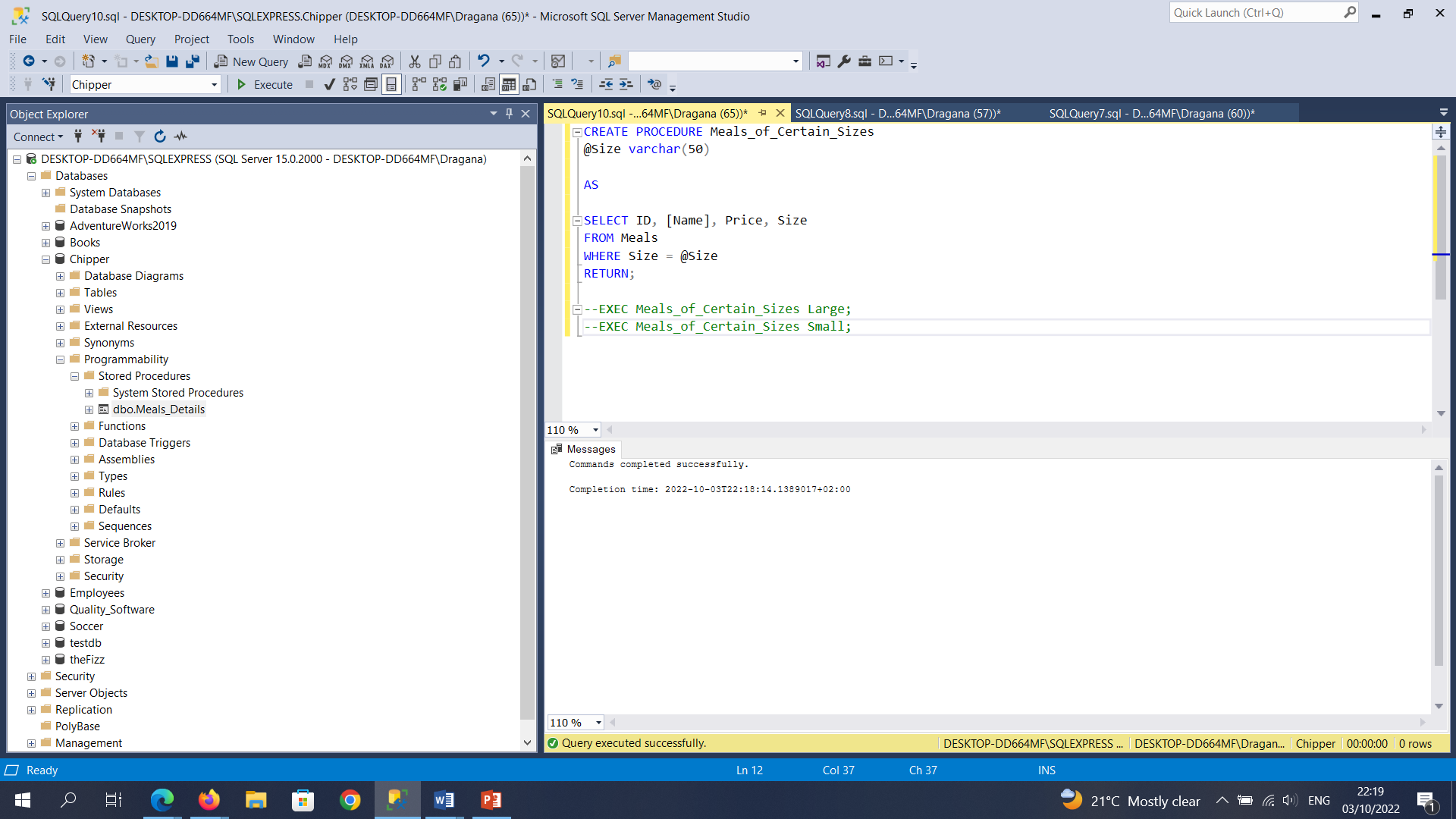
AS

SELECT ID, [Name], Price, Size

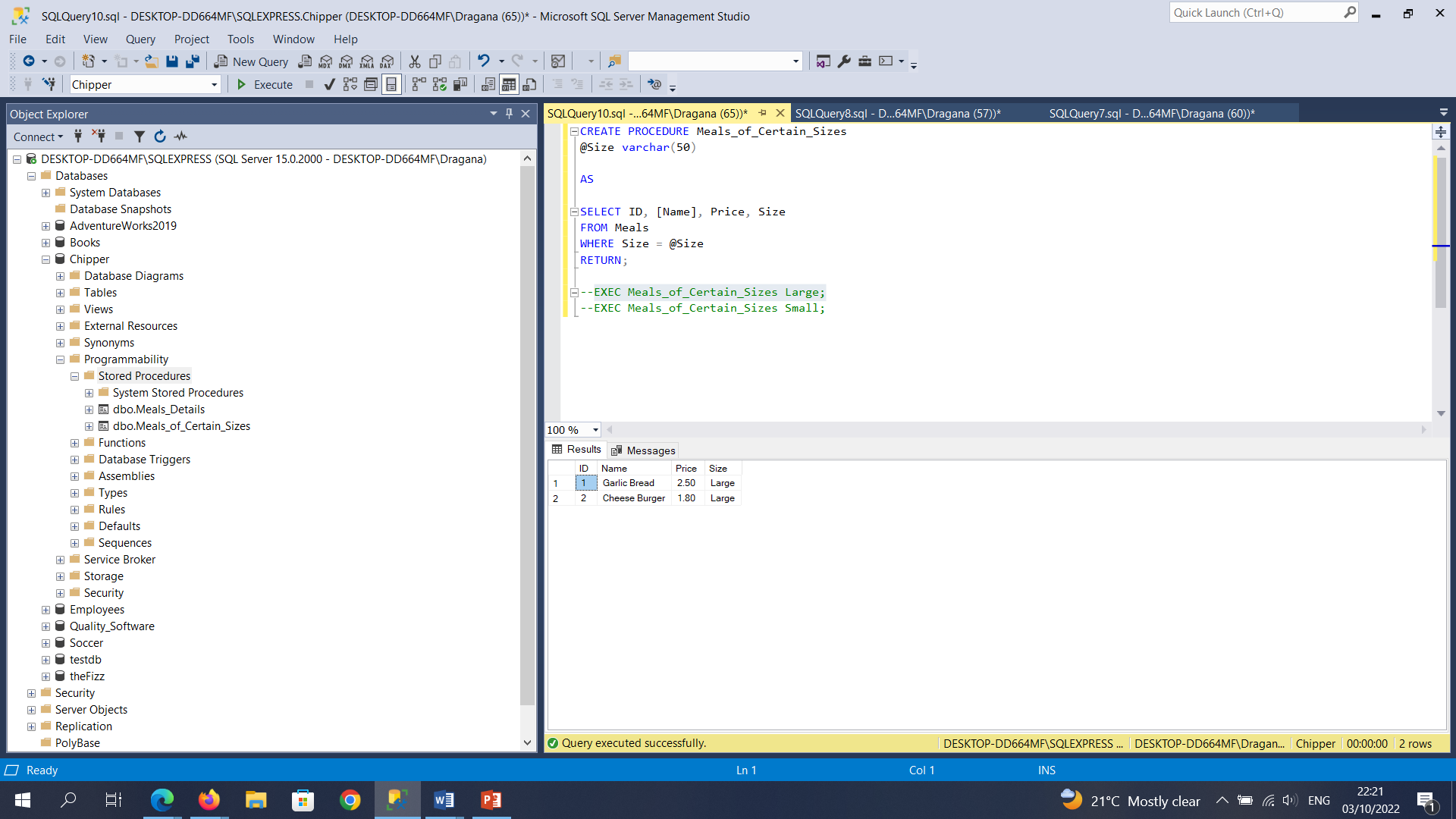
FROM Meals

WHERE Size = @Size

RETURN;



* For example, if the value ‘Large’ is passed to the stored procedure, only details of large-sized meals should be displayed.



* If however the value ‘Small’ is passed to the stored procedure, then only details of small-sized meals should be displayed.

