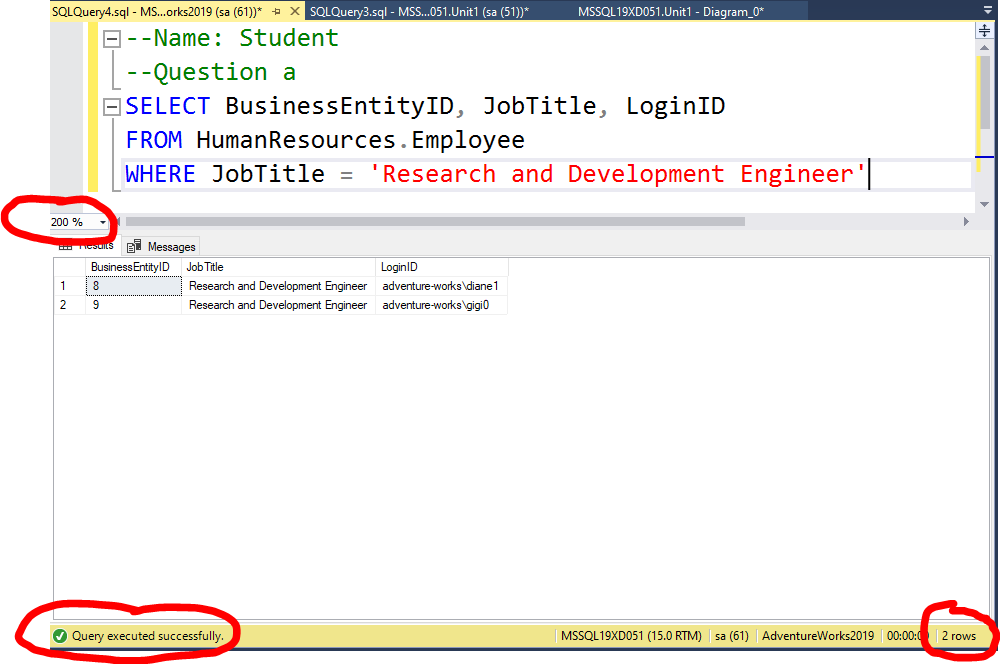
**Performance Assessment: Scripts, Procedures, Functions, and Triggers**

All these questions use the **AdventureWorks** database. These tables have the schema name in front of it, so use that in your queries. Submit screenshots like the one below, make sure your query shows and is readable!!. Also, ensure that the yellow bar that has the number of rows returned and the fact that the query “Executed Successfully” can be seen.

*Example: Write a query using a WHERE clause that displays all the employees listed in the*

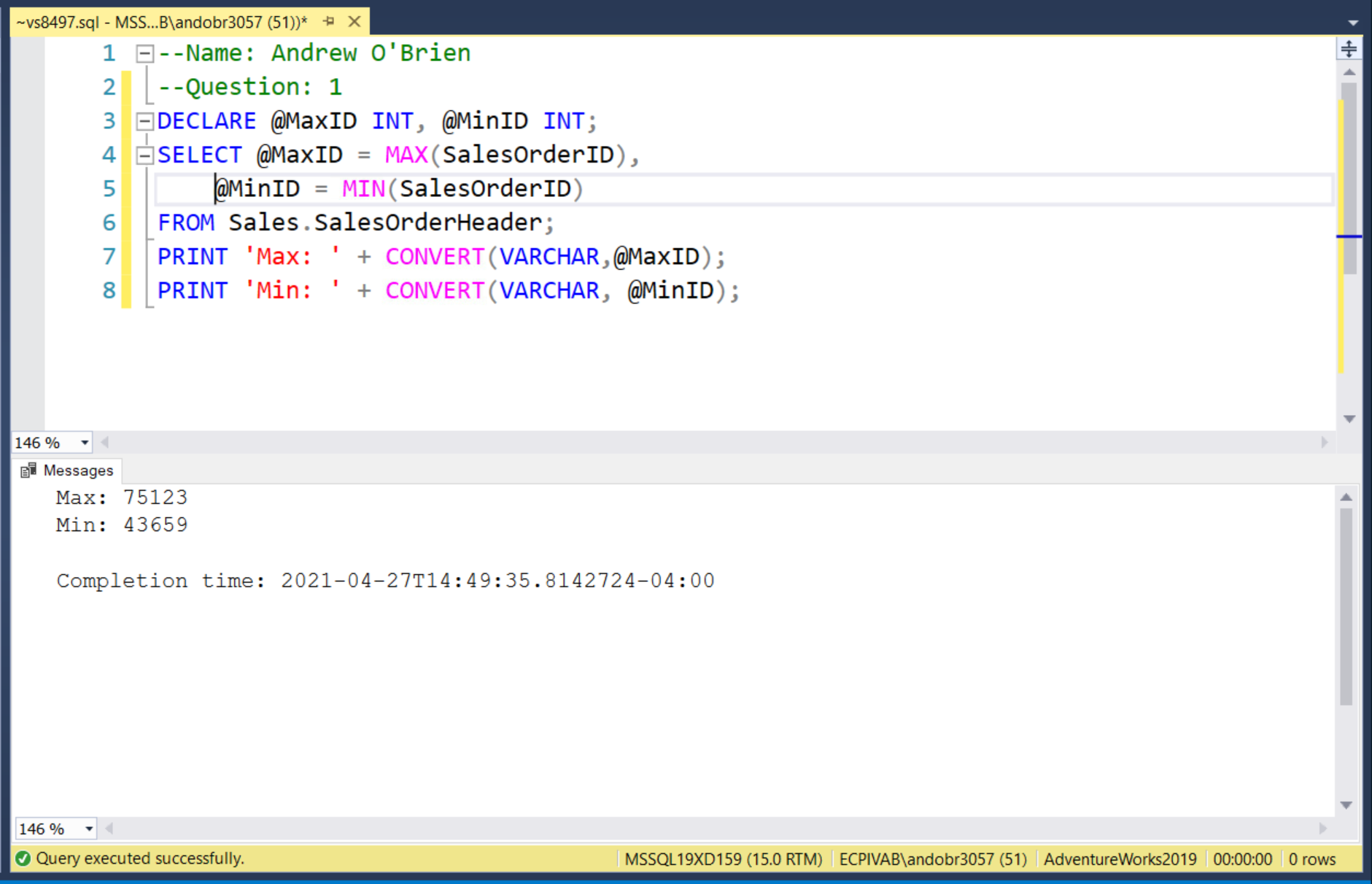
*HumanResources.Employee table who have the job title Research and Development Engineer.*

*Display the business entity ID number, the login ID, and the title for each one.*

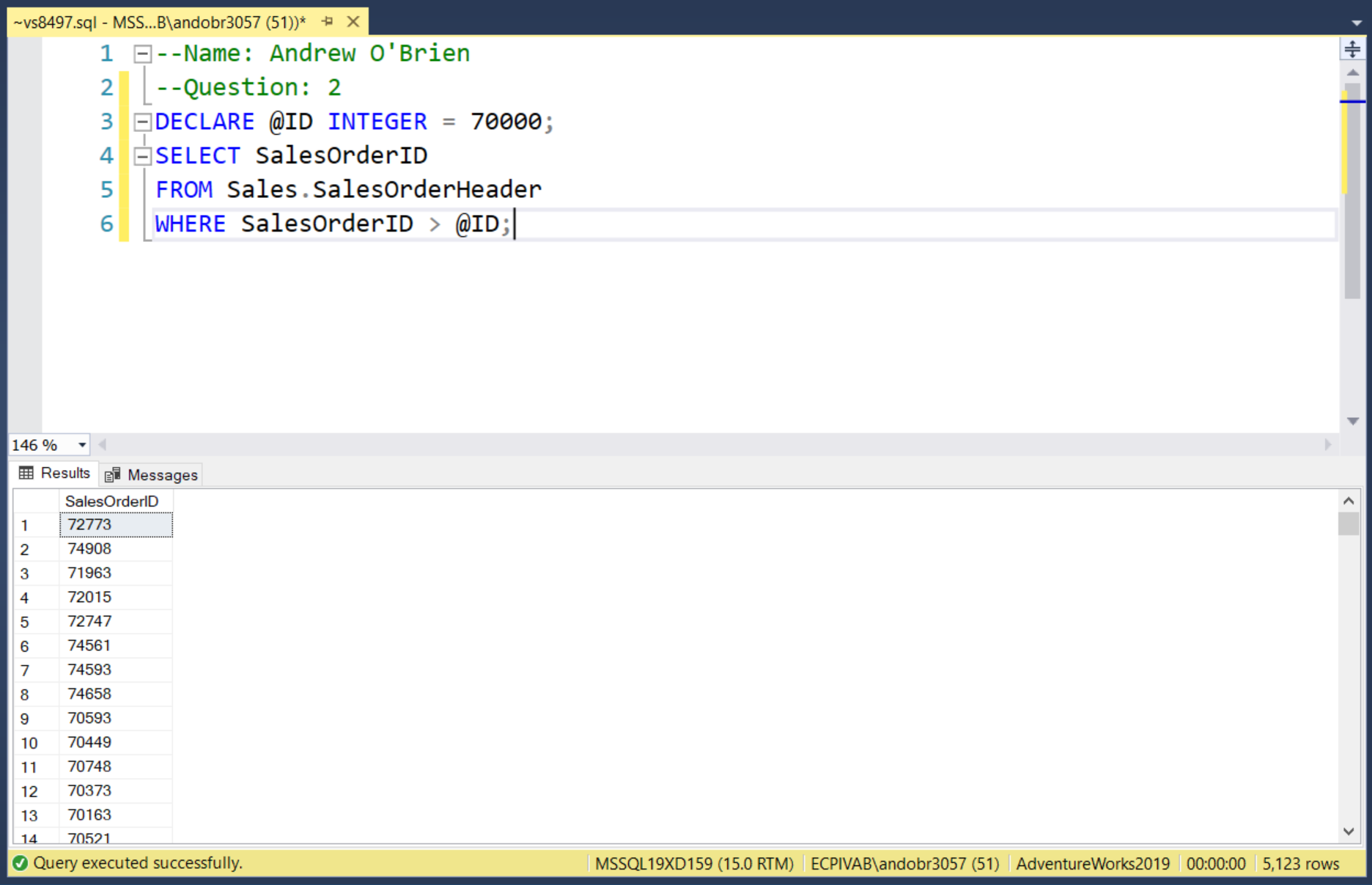


**Chapter 14. Scripts**

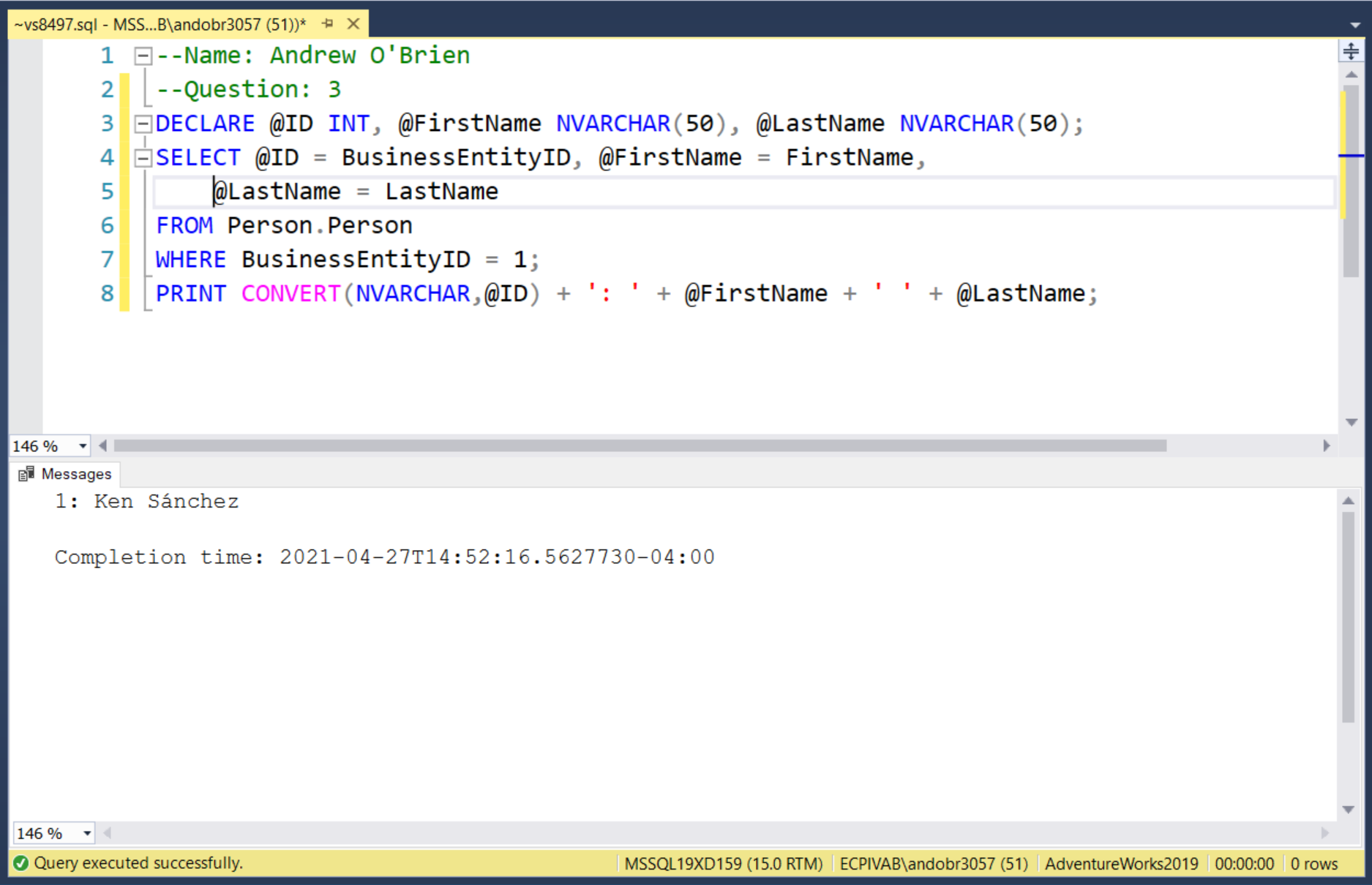
1. Write a script that declares two integer variables called @MaxID and @MinID. Use the variables to print the highest and lowest SalesOrderID values from the Sales.SalesOrderHeader table.



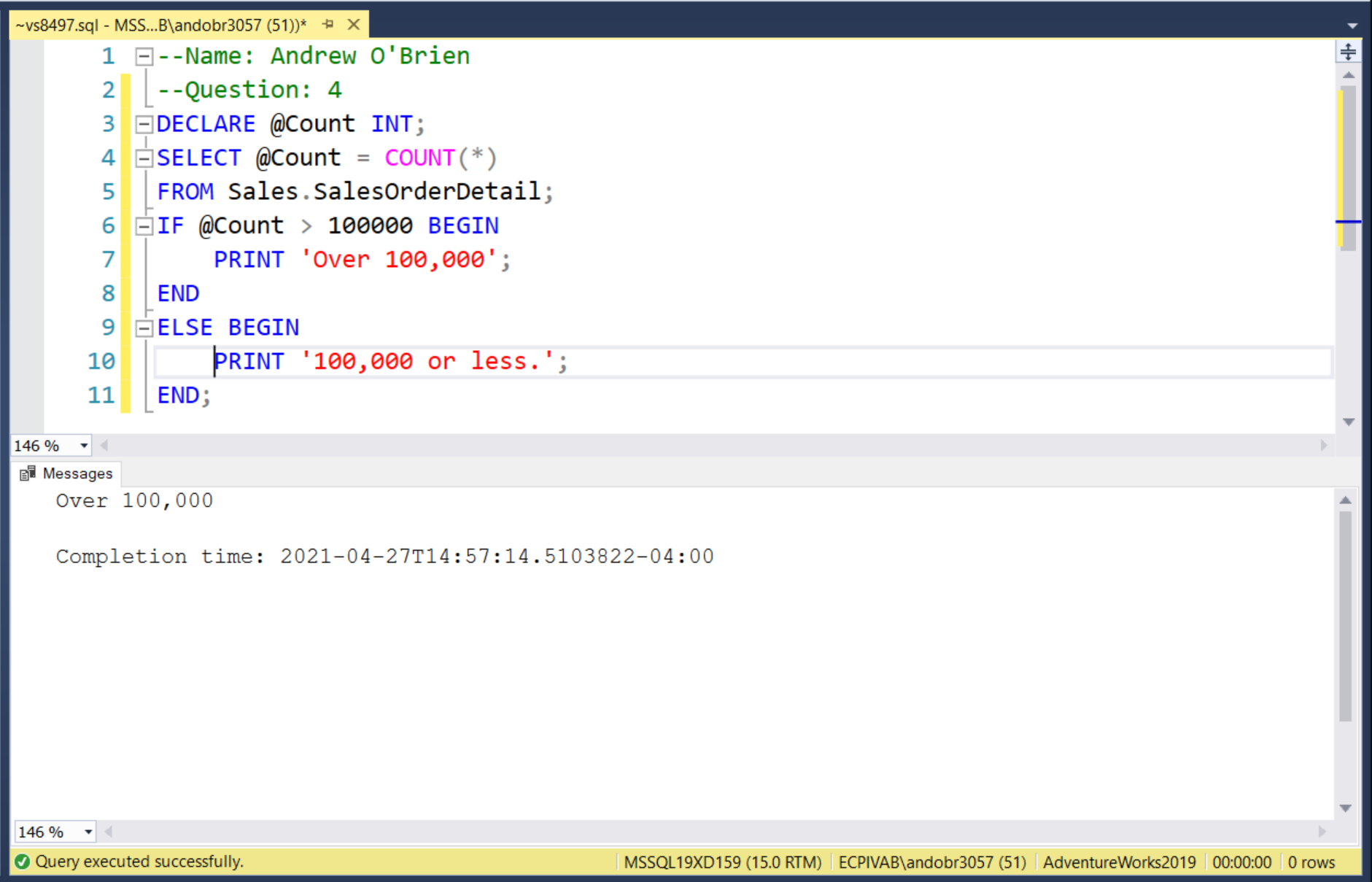
1. Write a script that declares an integer variable called @ID. Assign the value 70000 to the variable. Use the variable in a SELECT statement that returns all the SalesOrderID values from the Sales.SalesOrderHeader table that have a SalesOrderID greater than the value of the variable.



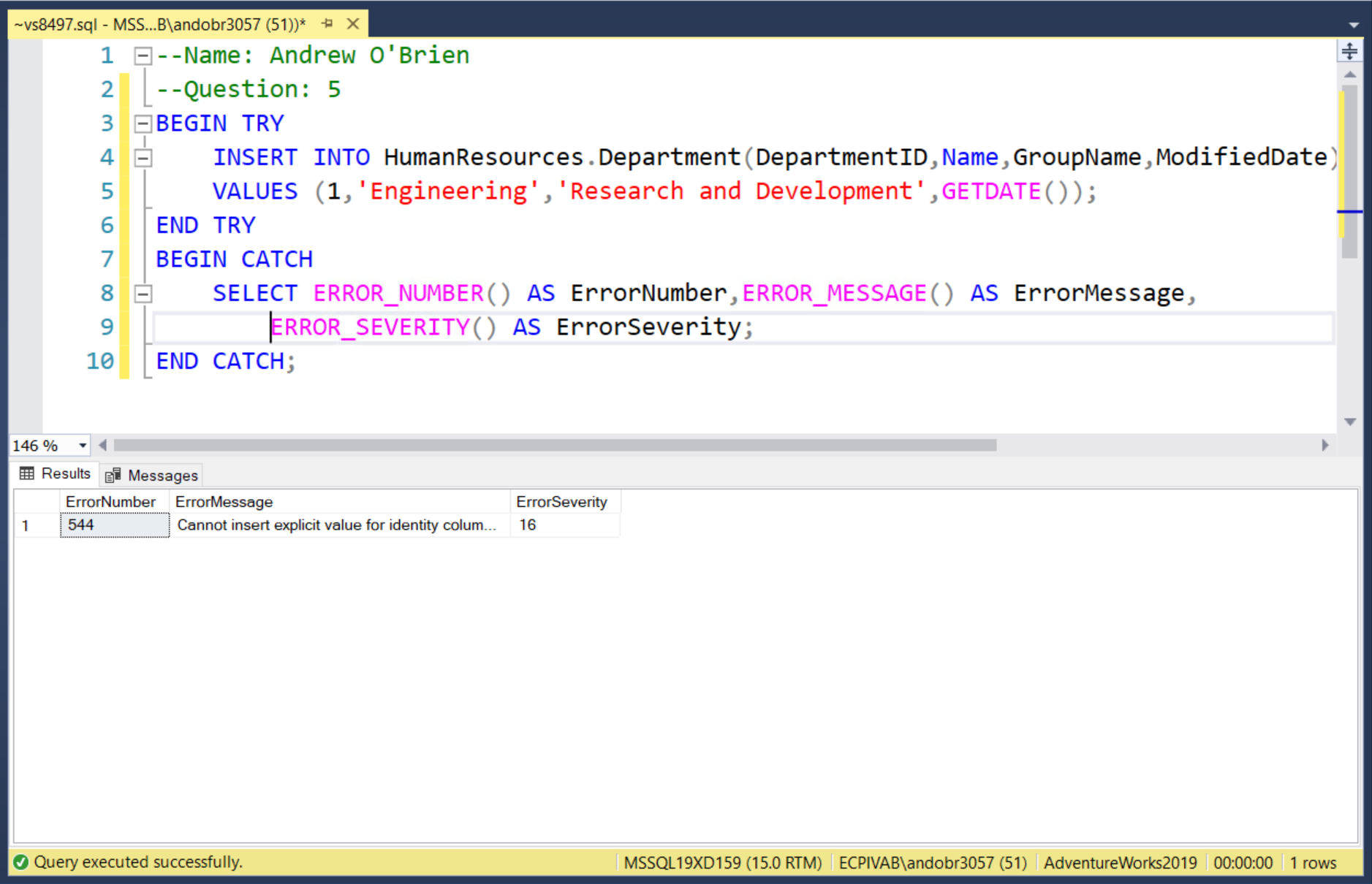
1. Write a script that declares three variables, one integer variable called @ID, an NVARCHAR(50) variable called @FirstName, and an NVARCHAR(50) variable called @LastName. Use a SELECT statement to set the value of the variables with the row from the Person.Person table with BusinessEntityID = 1. Print a statement in the “BusinessEntityID: FirstName LastName” format.



1. Write a batch that declares an integer variable called @Count to save the count of all the Sales.SalesOrderDetail records. Add an IF block that that prints “Over 100,000” if the value exceeds 100,000. Otherwise, print “100,000 or less.”

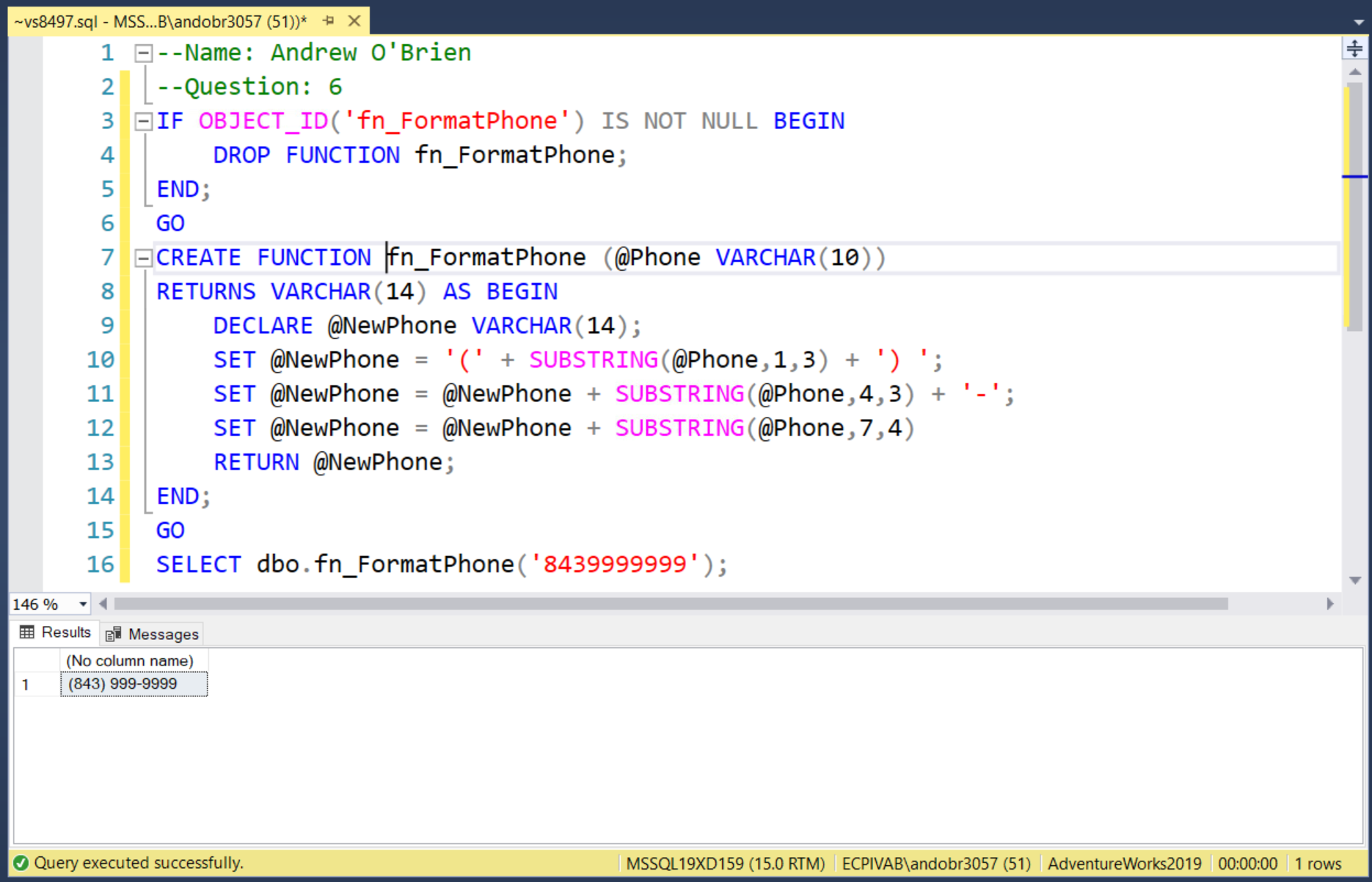


1. Write a statement that attempts to insert a duplicate row into the HumanResources.Department table. Use a TRY CATCH block to trap the error. Display the error number, message, and severity.



**Chapter 15. Code Procedures and Functions**

1. Write a function called fn\_FormatPhone that takes a string of ten numbers. The function will format the string into this phone number format: “(###) ###-####.” Test the function.



1. Create a stored procedure called usp\_ProductSales that accepts a ProductID for a parameter and has an OUTPUT parameter that returns the number sold for the product. Test the stored procedure.

