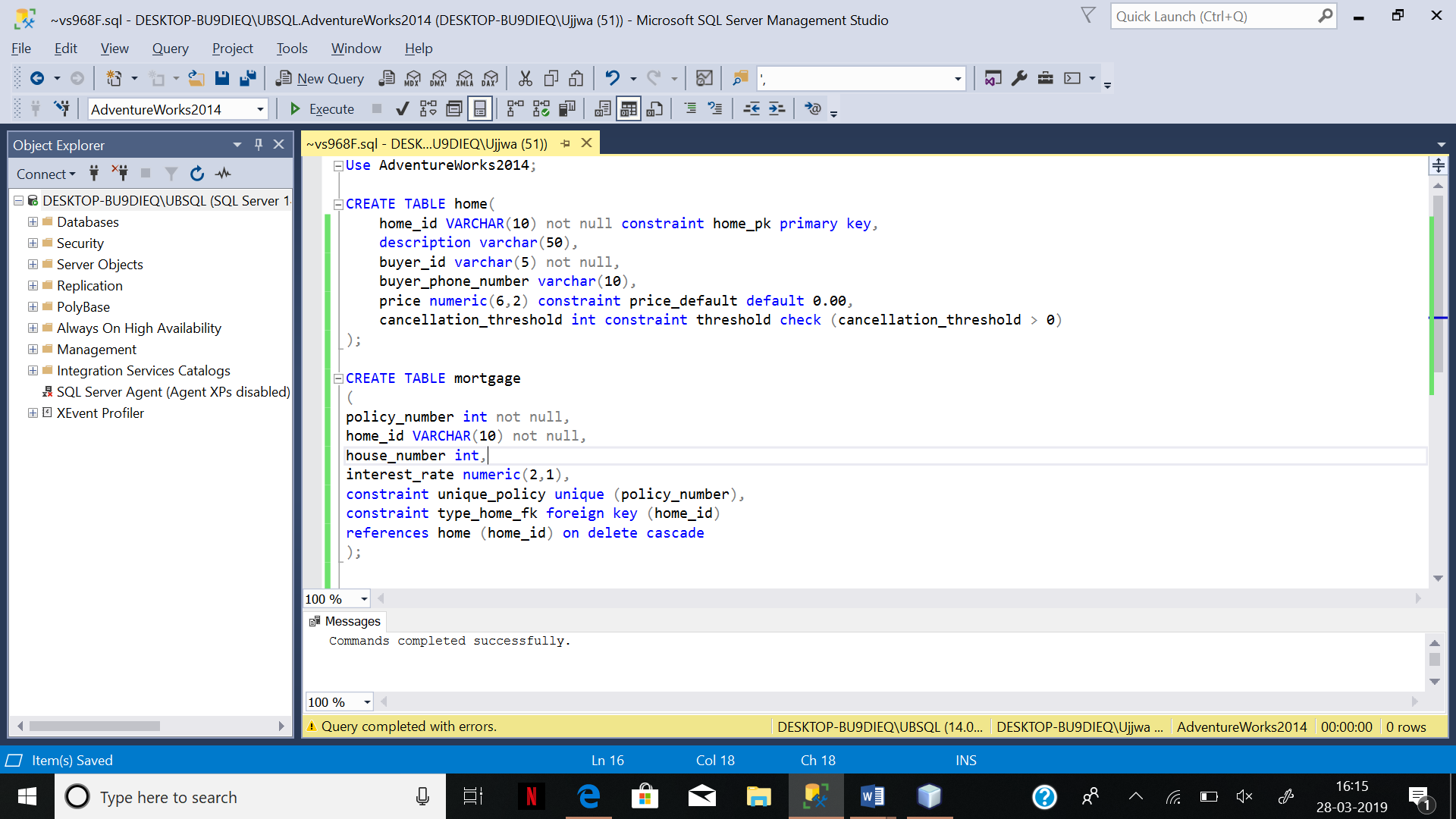
Assignment Part 1

Section A

|  |
| --- |
| Demonstrate with examples the following constraints:   * + DEFAULT   + PRIMARY KEY   + UNIQUE   + FOREIGN KEY   + CHECK   You may use an existing db or create your own, for testing purposes  Do not use the GUI for this part of the lab, everything you do had to be done with T-SQL code |
|  |



Section B

In this practice, we use the ALTER TABLE command to add and drop constraints to a table, including primary key, unique, and foreign key constraints.

Note: do not create the table, it already exists, just run the code below

/\*

CREATE TABLE Production.Products (

productid INT NOT NULL IDENTITY,

productname NVARCHAR(40) NOT NULL,

supplierid INT NOT NULL,

categoryid INT NOT NULL,

unitprice MONEY NOT NULL

CONSTRAINT DFT\_Products\_unitprice DEFAULT(0),

discontinued BIT NOT NULL

CONSTRAINT DFT\_Products\_discontinued DEFAULT(0),

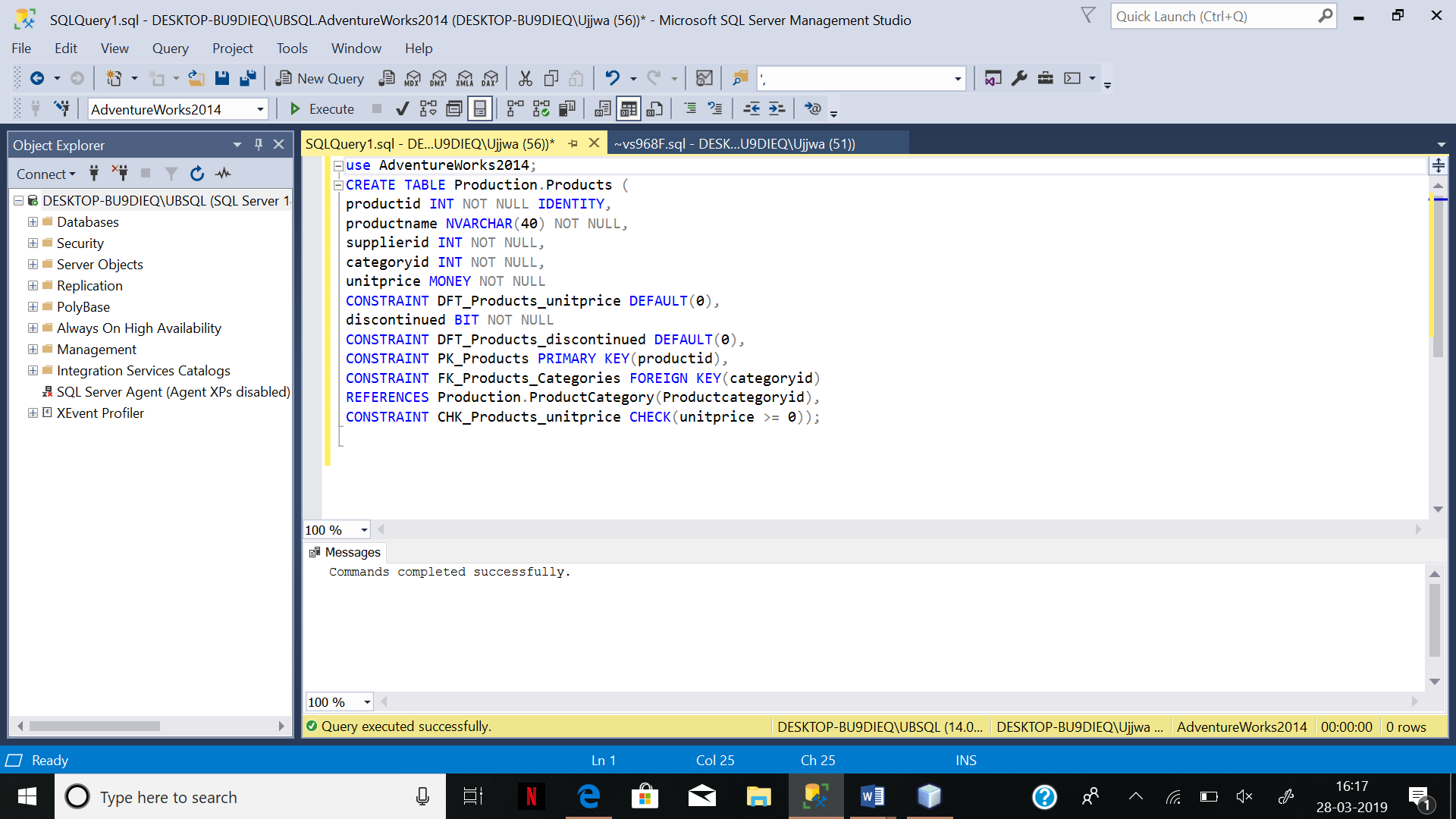
CONSTRAINT PK\_Products PRIMARY KEY(productid),

CONSTRAINT FK\_Products\_Categories FOREIGN KEY(categoryid)

REFERENCES Production.ProductCategory(Productcategoryid),

CONSTRAINT CHK\_Products\_unitprice CHECK(unitprice >= 0));

\*/



1. Test the primary key using the following.

SELECT productname FROM Production.Products

WHERE productid = 1;

SET IDENTITY\_INSERT Production.Products ON;

GO

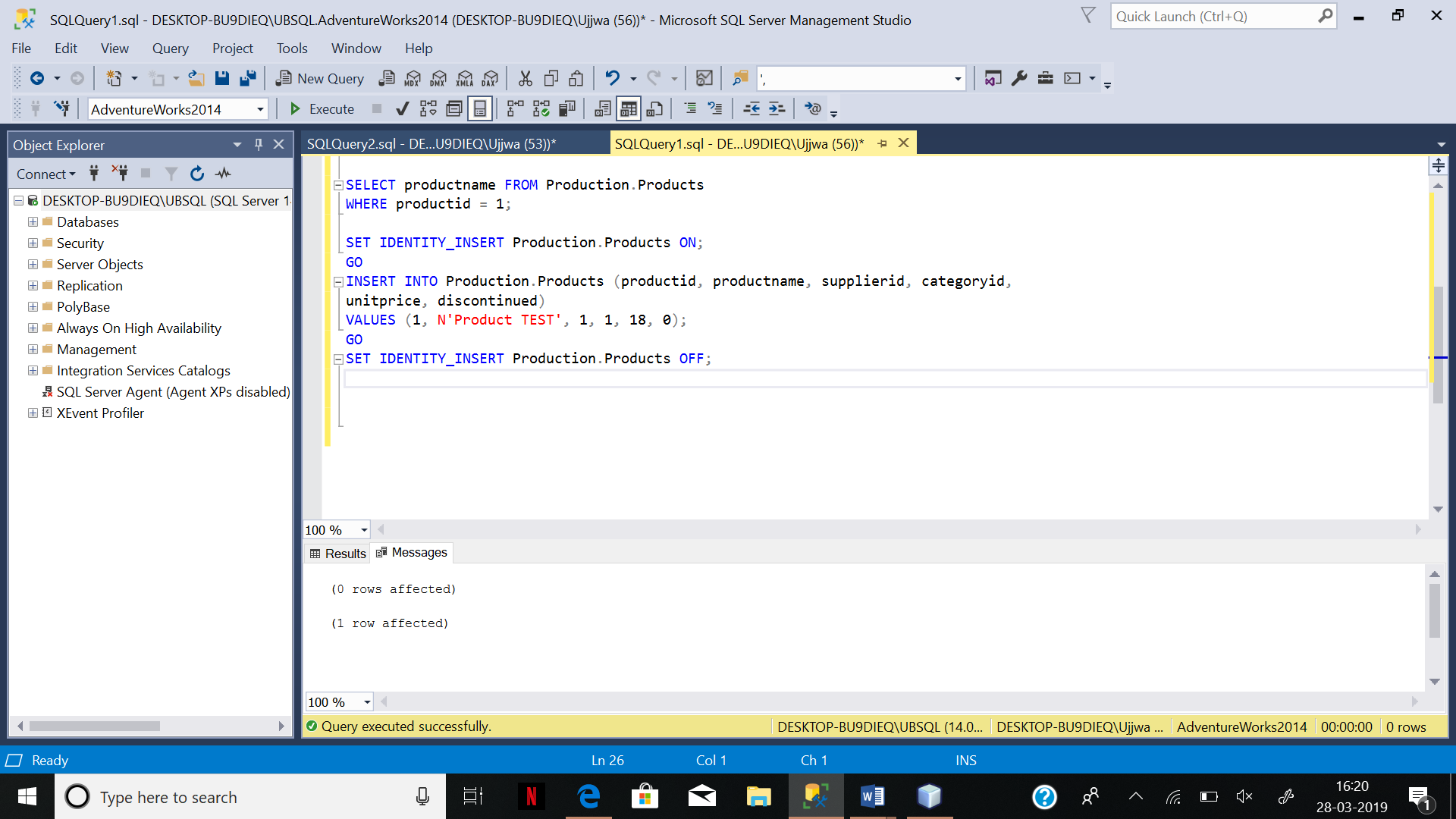
INSERT INTO Production.Products (productid, productname, supplierid, categoryid,

unitprice, discontinued)

VALUES (1, N'Product TEST', 1, 1, 18, 0);

GO

SET IDENTITY\_INSERT Production.Products OFF;  
  
Include a screenshot of any message you get.

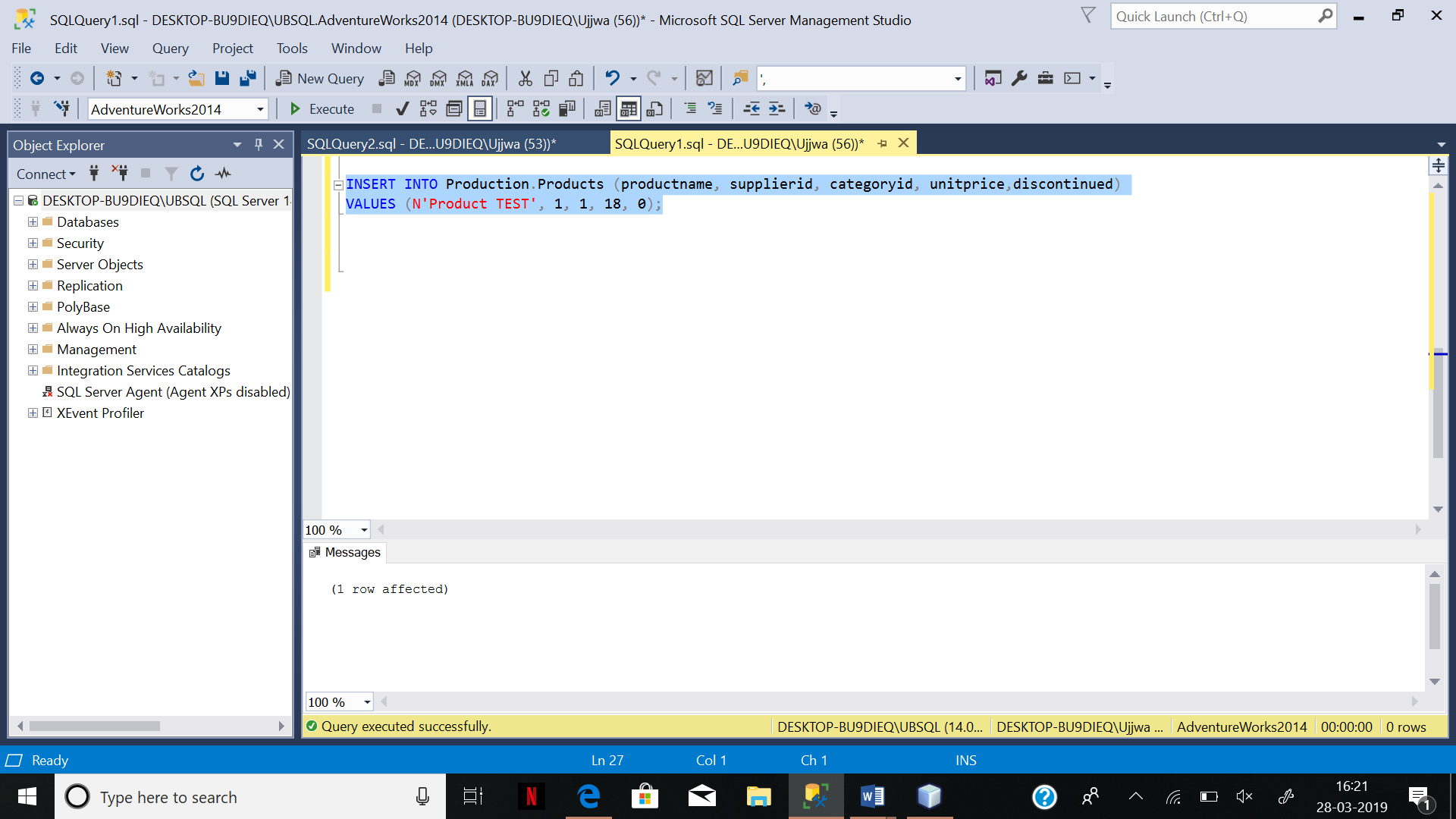


1. Insert a new row that lets the Identity property assign a new productid.

INSERT INTO Production.Products (productname, supplierid, categoryid, unitprice,discontinued)

VALUES (N'Product TEST', 1, 1, 18, 0);

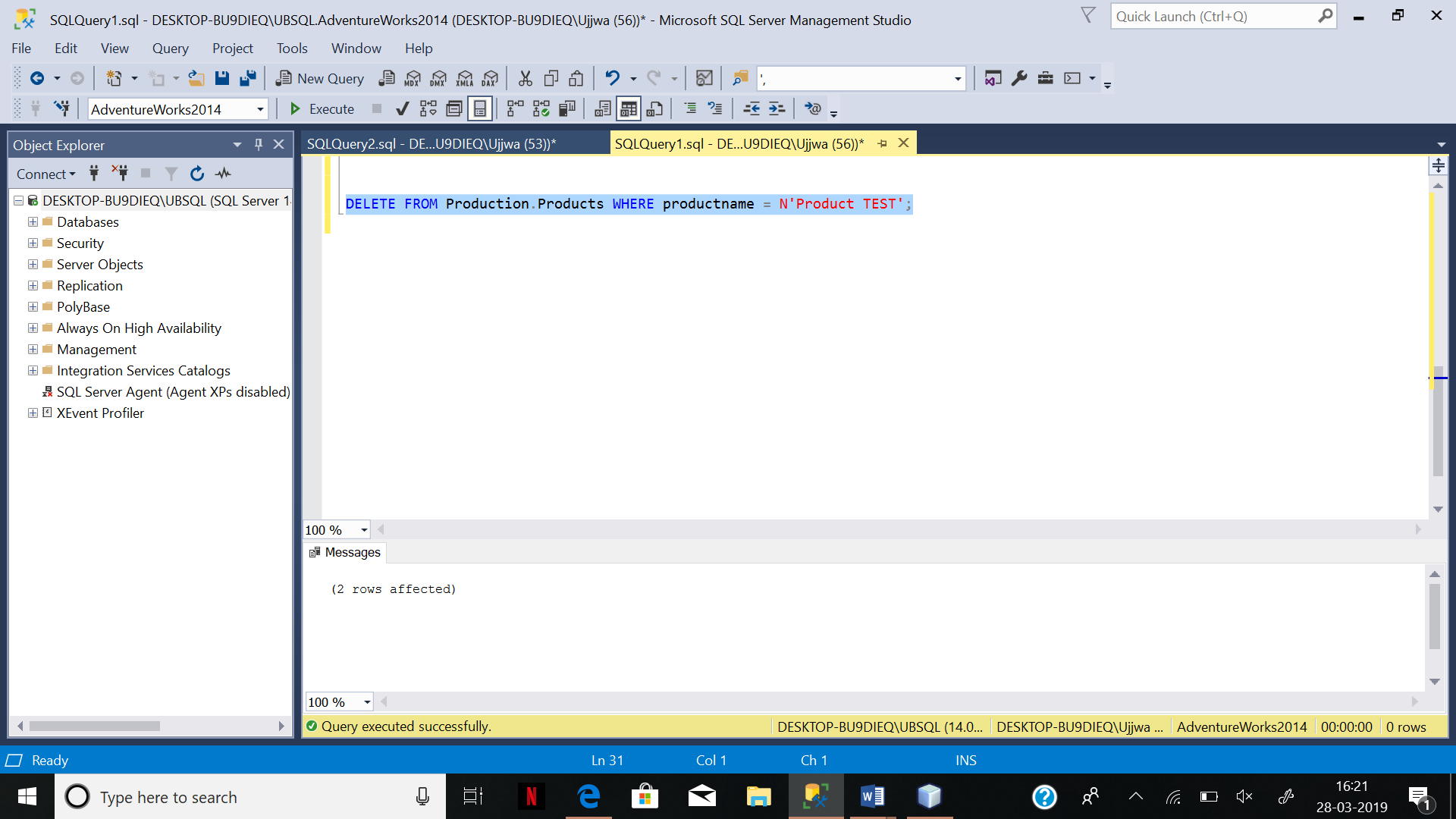
Include a screenshot of any message you get.



1. Delete the test row.

DELETE FROM Production.Products WHERE productname = N'Product TEST';

Include a screenshot of any message you get.



1. Try again with an invalid categoryid = 99.

INSERT INTO Production.Products (productname, supplierid, categoryid, unitprice,

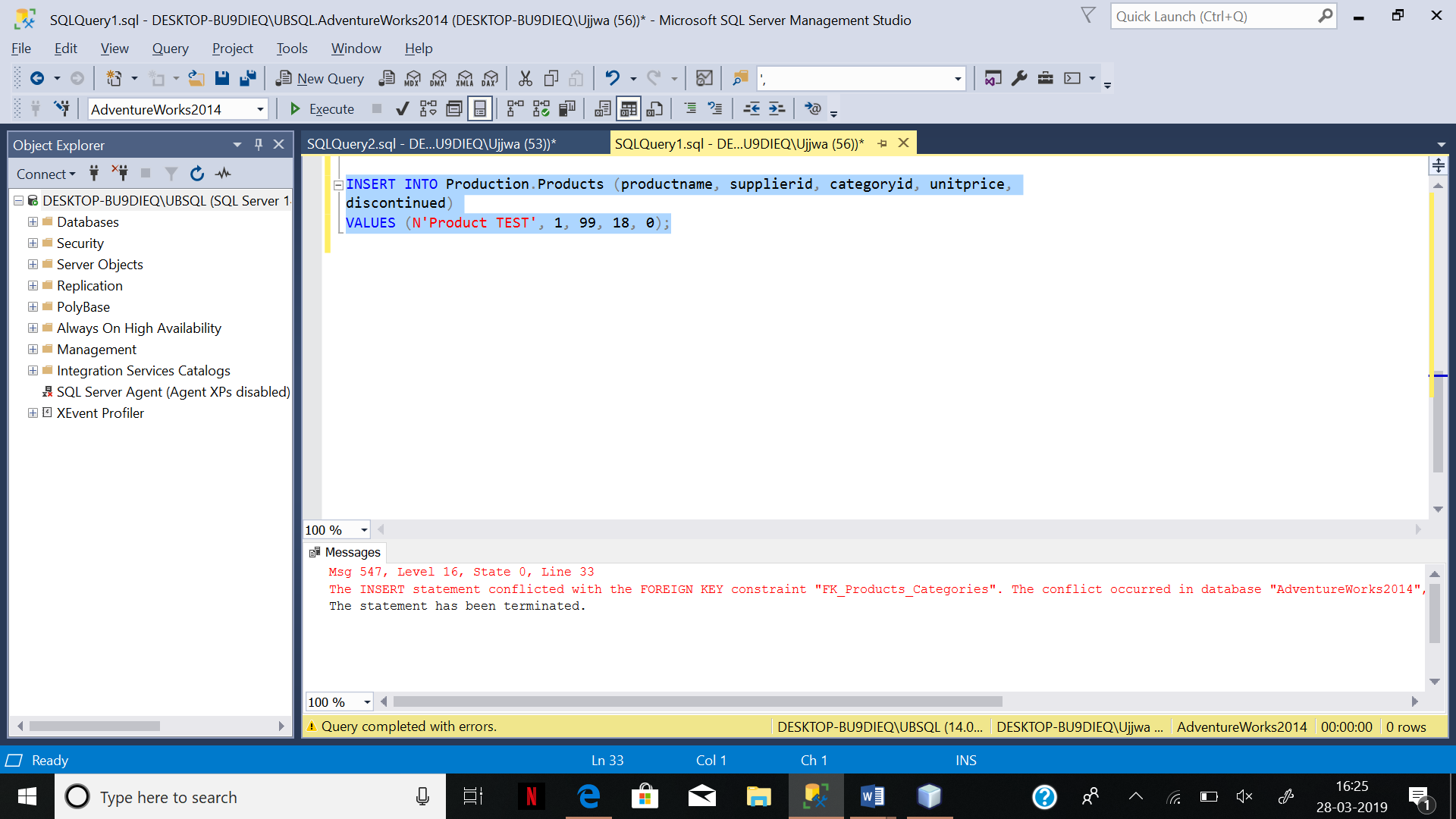
discontinued)

VALUES (N'Product TEST', 1, 99, 18, 0);

GO

Explain the reason the INSERT failed and provide screenshot of the error message

Answer: the INSERT failed as table ProductCategory doesn’t has any ProductCategoryID = 18, and categoryID in Production.Products is a foreign key with the reference of ProductCategoryID



1. Drop the foreign key constraint on Production.Products. Show your SQL code here.

ALTER TABLE Production.Products

DROP

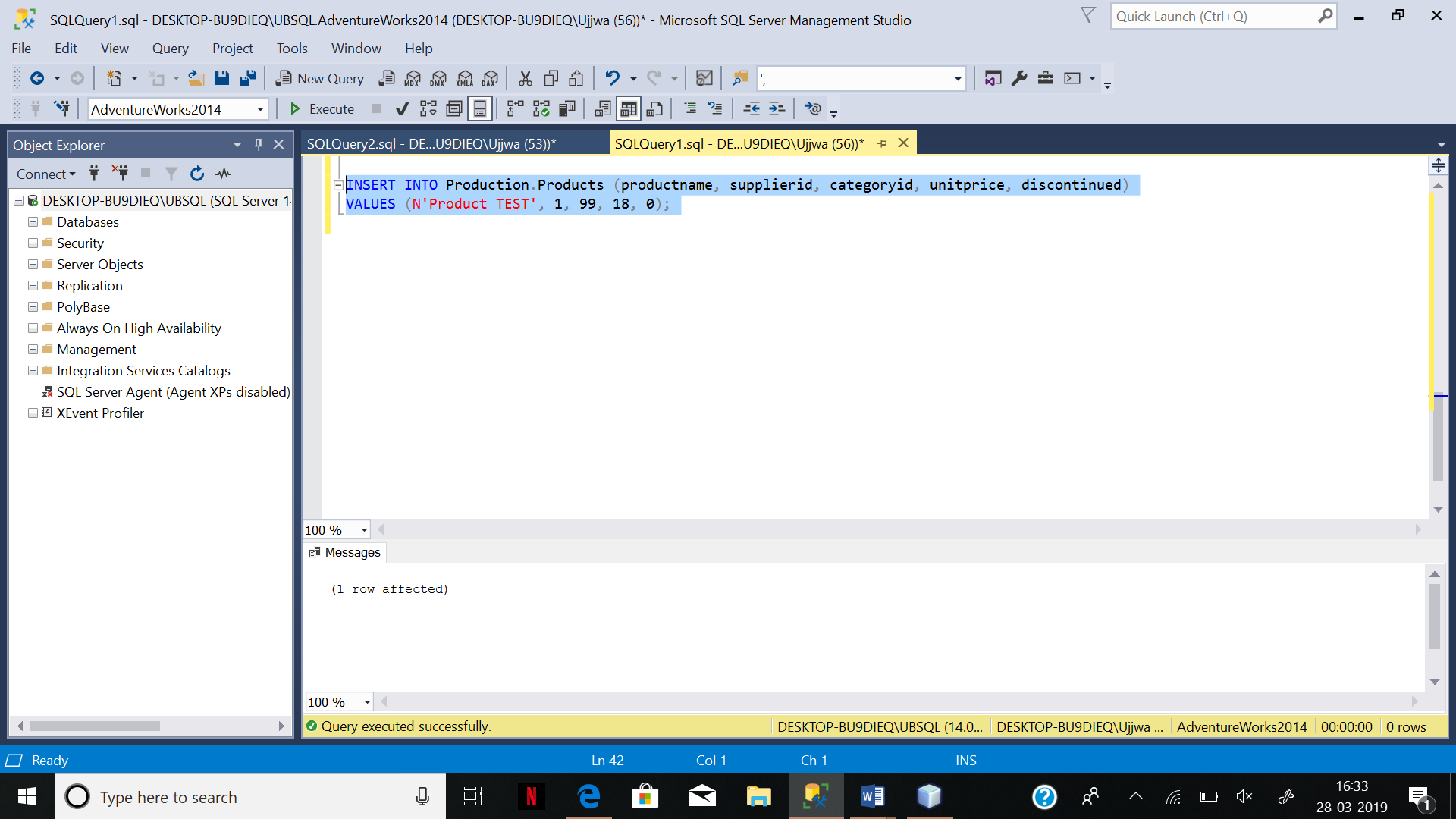
CONSTRAINT FK\_Products\_Categories;

1. After doing #5 above, try the insert now with the invalid categoryid = 99.

INSERT INTO Production.Products (productname, supplierid, categoryid, unitprice, discontinued)

VALUES (N'Product TEST', 1, 99, 18, 0);

Did it succeed? Provide a screenshot of your success message in the messages window.



1. Try to add the foreign key constraint back in using WITH CHECK.

ALTER TABLE Production.Products WITH CHECK

ADD CONSTRAINT FK\_Products\_Categories FOREIGN KEY(categoryid)

REFERENCES Production.Categories (categoryid);

GO

What happened and why?

Unable to add foreignkey as there is no table named Production.Category

1. Update the row so that it has a valid categoryid.

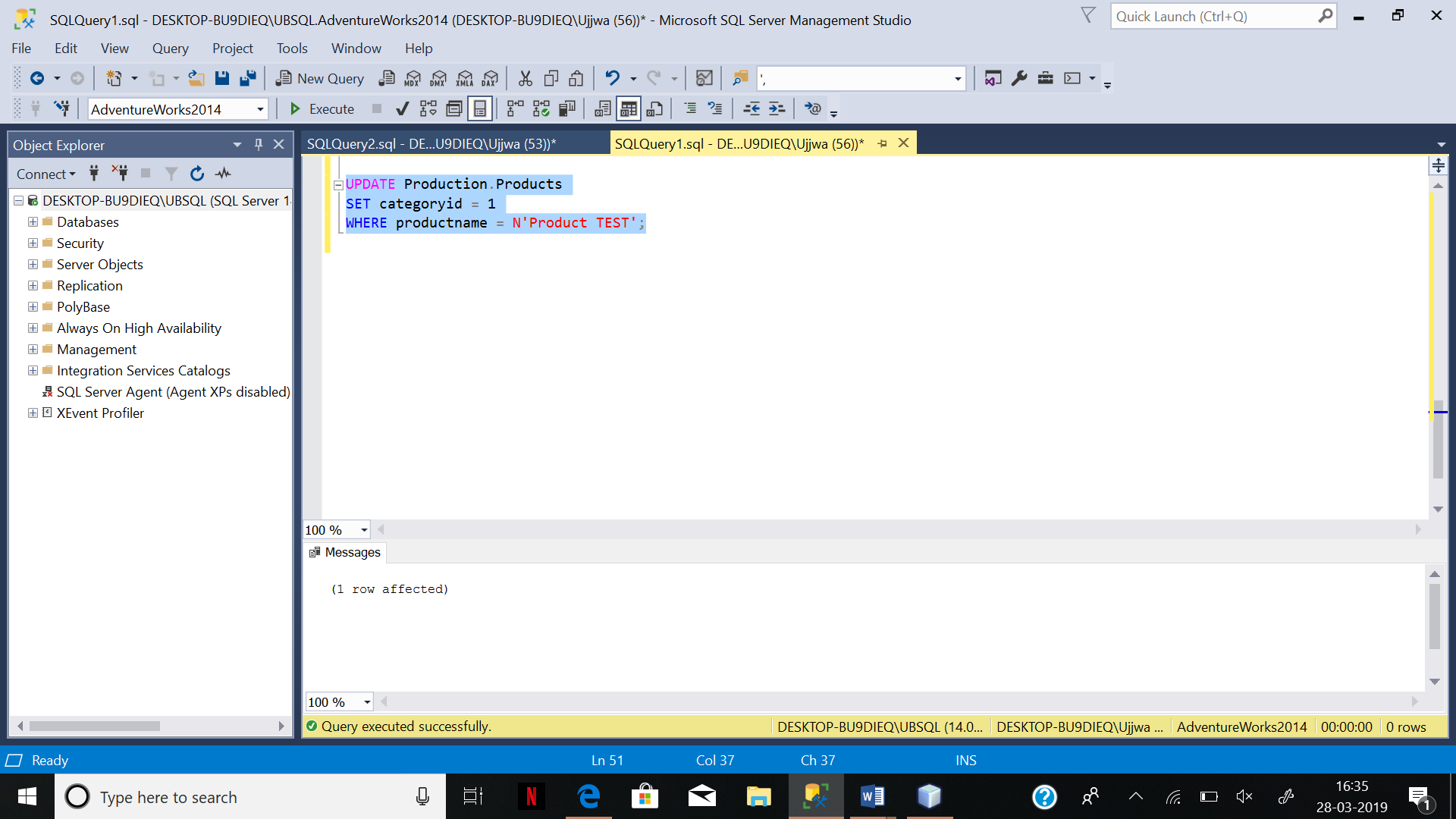
UPDATE Production.Products

SET categoryid = 1

WHERE productname = N'Product TEST';

GO

Include a screenshot of any message you get.



9. Now try to add the foreign key constraint back to the table.

ALTER TABLE Production.Products WITH CHECK

ADD CONSTRAINT FK\_Products\_Categories FOREIGN KEY(categoryid)

REFERENCES Production.Categories (categoryid);

GO

This time what happened? Can you explain why?

Unable to add foreignkey as there is no table by the name Production.Category.

10. Drop the test row from the table.

DELETE FROM Production.Products

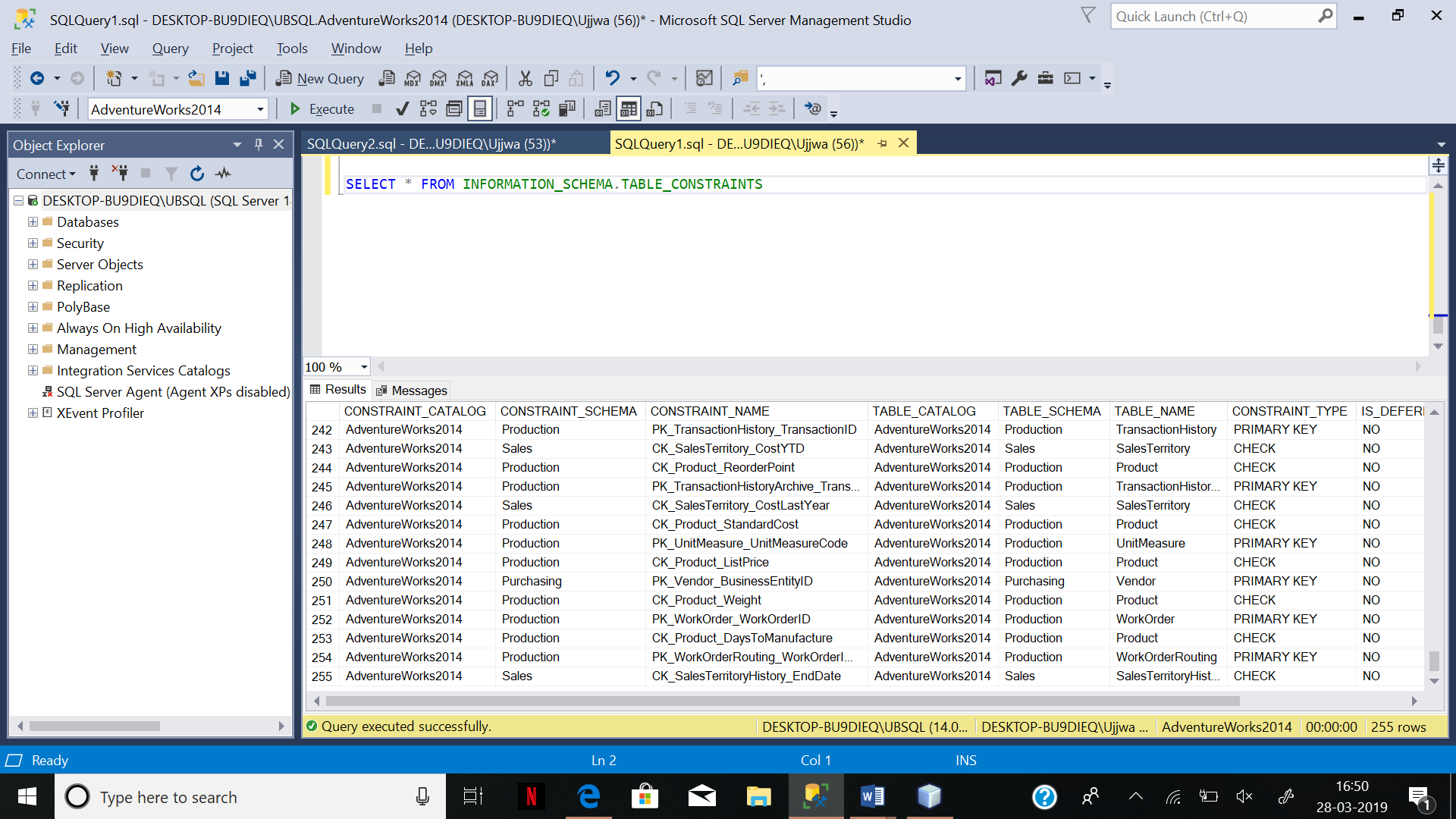
WHERE productname = N'Product TEST';

Nothing to show here

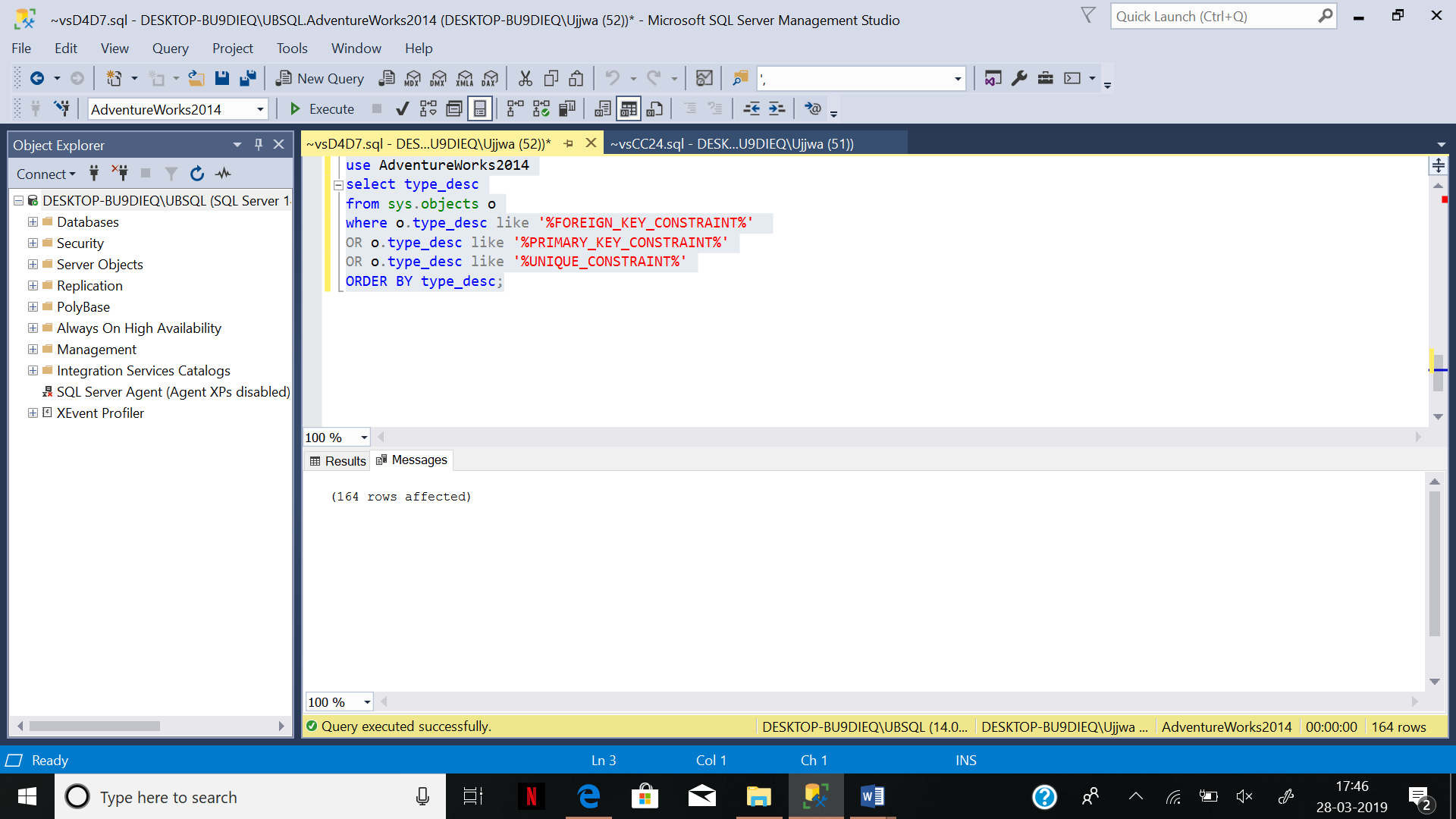
Part 3

Do some research to find out how to list all the constraints using a T-SQL command.

Run the command and list:



All the Primary Key, Foreign Key and Unique constraints in the dventureWorks table.



You can use this webpage:

https://bhaveshgpatel.wordpress.com/2009/11/04/sql-server-list-all-constraints-of-database-or-table/