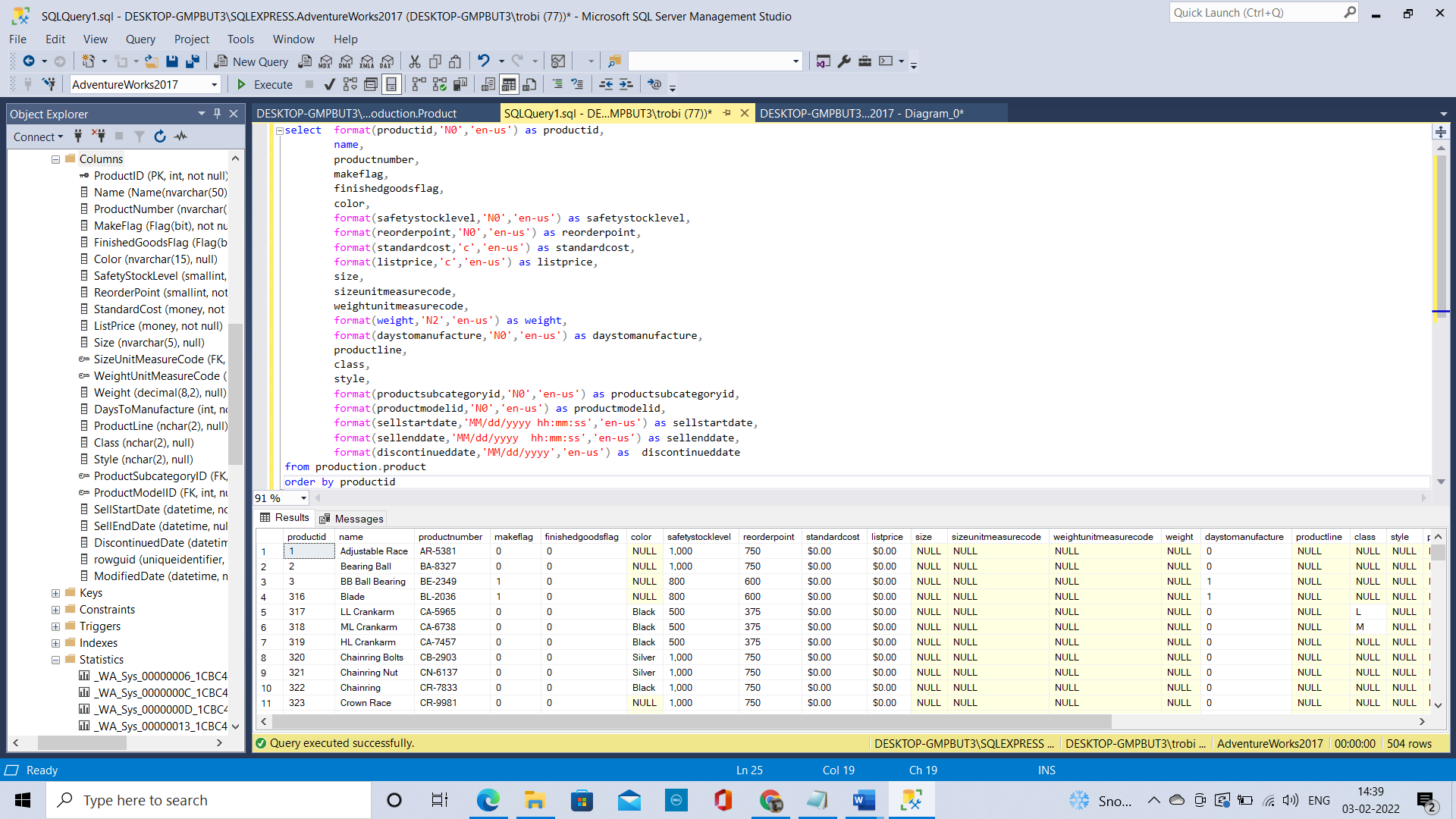
**Write Product Query**select format(productid,'N0','en-us') as productid,

name,

productnumber,

makeflag,

finishedgoodsflag,

color,

format(safetystocklevel,'N0','en-us') as safetystocklevel,

format(reorderpoint,'N0','en-us') as reorderpoint,

format(standardcost,'c','en-us') as standardcost,

format(listprice,'c','en-us') as listprice,

size,

sizeunitmeasurecode,

weightunitmeasurecode,

format(weight,'N2','en-us') as weight,

format(daystomanufacture,'N0','en-us') as daystomanufacture,

productline,

class,

style,

format(productsubcategoryid,'N0','en-us') as productsubcategoryid,

format(productmodelid,'N0','en-us') as productmodelid,

format(sellstartdate,'MM/dd/yyyy hh:mm:ss','en-us') as sellstartdate,

format(sellenddate,'MM/dd/yyyy hh:mm:ss','en-us') as sellenddate,

format(discontinueddate,'MM/dd/yyyy','en-us') as discontinueddate

from production.product

order by productid

**Write SubCategory query**First write queries to answer the following preliminary questions and answer the following questions in the Write SubCategory Query section of **A03.docx**

* + 1. How many products are there?

**504**select count(\*) from Production.Product

* + 1. How many subcategories are there?  
       **37**select count(\*) from Production.ProductSubcategory
    2. Do all products have subcategories?   
       **No all products do not have subcategories**
       - 1. If not, how did you know they did not?   
            **we can check whether ProductSubcategoryID column is not null in product table**select count(\*) from Production.Product

where ProductSubcategoryID is null

* + - * 1. If not, how many products have subcategories?   
           **295**select count(\*) from Production.Product

where ProductSubcategoryID is not null

* + - * 1. If not, how many products did not have subcategories?  
           **209**select count(\*) from Production.Product

where ProductSubcategoryID is null

iv)Given the goal of the query (see above), how many rows should there be when you are done?  
 **295** select count(\*)

from Production.ProductSubcategory

left join Production.Product

on Production.ProductSubcategory.ProductSubcategoryID = Production.Product.ProductSubcategoryID

* 1. Write a query from the product table that includes all rows and the following fields in the order listed below:
     1. ProductID
     2. ProductNumber
     3. Name

select p.ProductID,p.ProductNumber,p.Name from Production.Product p

order by p.ProductID,p.ProductNumber,p.Name

* 1. Add a join to the query
     1. Add a left join to the ProductSubCategory table   
          
        select \* from Production.Product p

left join Production.ProductSubcategory

on p.ProductSubcategoryID = Production.ProductSubcategory.ProductSubcategoryID

order by p.ProductID,p.ProductNumber,p.Name

* + 1. Add ProductSubCategory.Name as the first field in the list. What field in the ProductSubCategory table must match what field in the Product table in order for the join to work?  
         
       **ProductSubcategoryID field should be used to join the tables.**  
         
       select Production.ProductSubcategory.name as Name,p.\*,Production.ProductSubcategory.\* from Production.Product p

left join Production.ProductSubcategory

on p.ProductSubcategoryID = Production.ProductSubcategory.ProductSubcategoryID

order by p.ProductID,p.ProductNumber,p.Name

* + 1. Order by ProductSubCategory.Name and then by Product.Name, both ascending.   
         
       select Production.ProductSubcategory.name as Name,p.\*,Production.ProductSubcategory.\* from Production.Product p

left join Production.ProductSubcategory

on p.ProductSubcategoryID = Production.ProductSubcategory.ProductSubcategoryID

order by Production.ProductSubcategory.name,p.name

* 1. Did you get the correct number of rows (base this on the question “Given the goal of the query (see above), how many rows should there be when you are done?” answer above) If not, then:

No, the number of rows should be 295 but it is showing 504 rows. There are three ways it can be corrected.

1.By bringing the Production.ProductSubcategory first

select psc.name,p.ProductID,p.ProductNumber,p.Name

from Production.ProductSubcategory psc

left join Production.Product p

on psc.ProductSubcategoryID = p.ProductSubcategoryID

order by psc.name,p.name

2.Filtering the records in the product table where ProductSubcategoryID is not null

select psc.name,p.ProductID,p.ProductNumber,p.Name

from Production.Product p

left join Production.ProductSubcategory psc

on p.ProductSubcategoryID = psc.ProductSubcategoryID

where p.ProductSubcategoryID is not null

order by psc.name,p.name

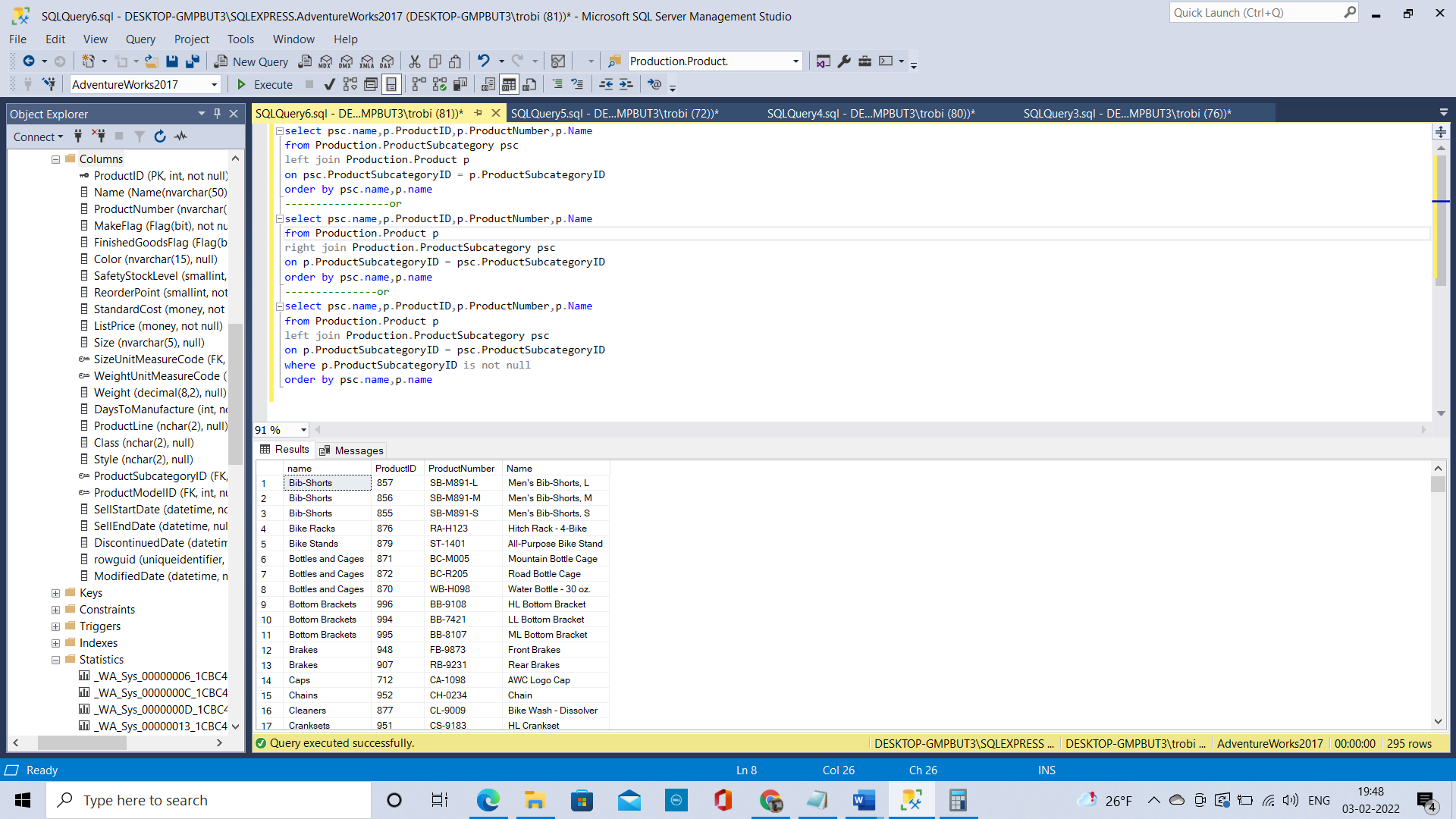
3.Using right join and keeping the product table first  
  
select psc.name,p.ProductID,p.ProductNumber,p.Name

from Production.Product p

right join Production.ProductSubcategory psc

on p.ProductSubcategoryID = psc.ProductSubcategoryID

order by psc.name,p.name



Final Query  
  
select psc.name,p.ProductID,p.ProductNumber,p.Name

from Production.ProductSubcategory psc

left join Production.Product p

on psc.ProductSubcategoryID = p.ProductSubcategoryID

order by psc.name,p.name

-----------------or

select psc.name,p.ProductID,p.ProductNumber,p.Name

from Production.Product p

right join Production.ProductSubcategory psc

on p.ProductSubcategoryID = psc.ProductSubcategoryID

order by psc.name,p.name

---------------or

select psc.name,p.ProductID,p.ProductNumber,p.Name

from Production.Product p

left join Production.ProductSubcategory psc

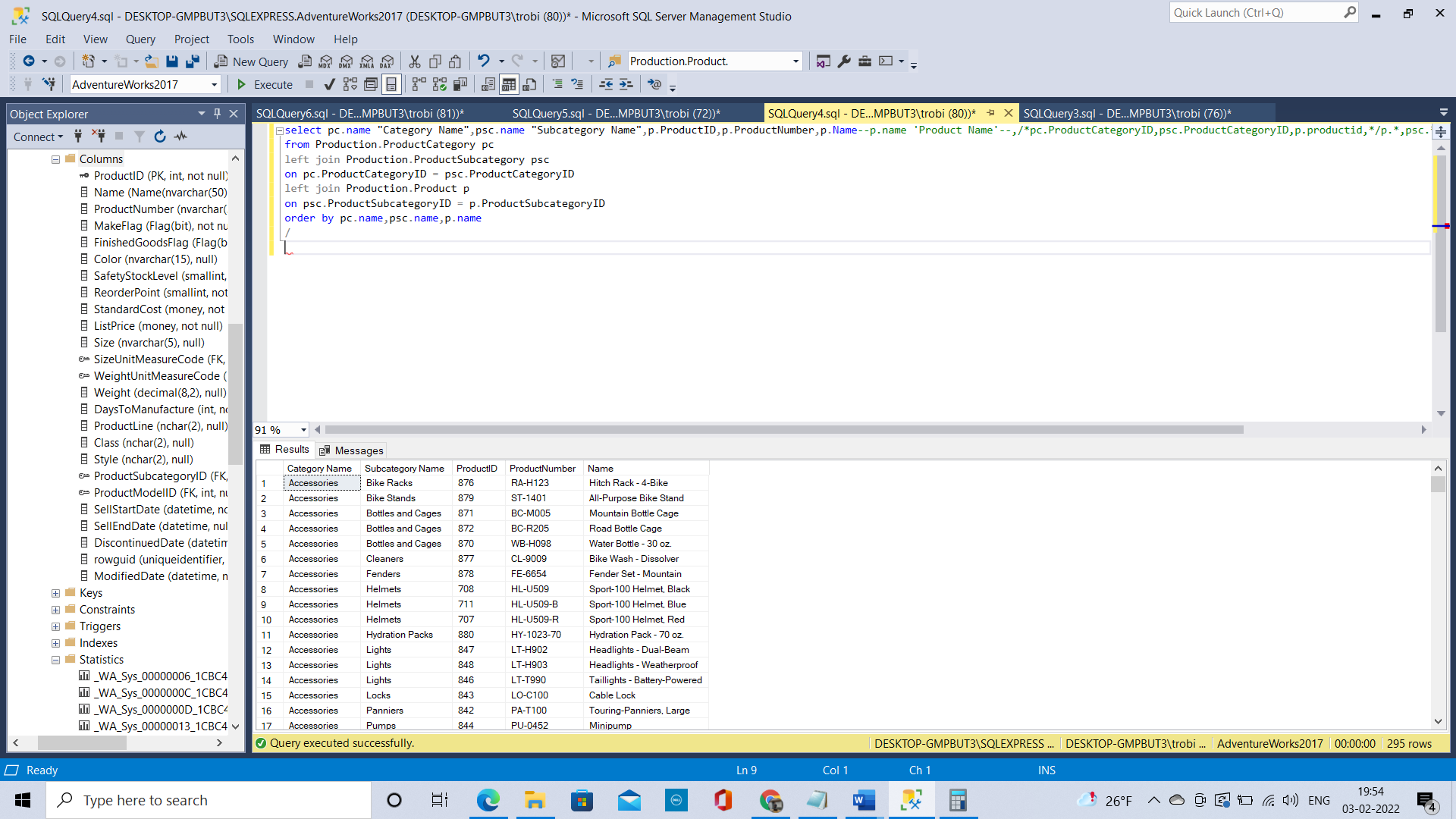
on p.ProductSubcategoryID = psc.ProductSubcategoryID

where p.ProductSubcategoryID is not null

order by psc.name,p.name

e) How many rows did your query produce? Did that row count match the goal?  
 **295 rows and it matched the goal**

* + 1. Did all the subcategories have products? If there had been a subcategory without any products what would have happened to your row count? What would the product name be in that case?  
       **All subcategories are available in product table.  
       The row count would remain same if there is an extra row in subcategory table. However, if in the product table subcategoryid is not present in subcategory table the row count would have reduced.  
       The product name would be null in that case and there would be no relationship for that particular row.**
    2. Why did we use a left join?
       1. What would have happened if we had used a right join? How many rows would a right join have produced?   
          If we use a right join product table needs to come first.  
          If we use a left join productsubcategory table needs to come first.  
          If we do other way around **504 records** will get displayed.
       2. What would have happened if we had used an inner join? How many rows would an inner join have produced?  
          Inner join would have produced **295 rows** which is the matching records.
       3. What would have happened if we had used an outer join? How many rows would an outer join have produced?  
          It will display all the matching values first then the missing value in the right and then the missing values in the left. The full(Outer) join would have produced **504 rows.**

**Write Category Query:**Given the goal of the query (see above), how many rows should there be when you are done?   
**295 rows**  
  


Final Code:  
  
select pc.name "Category Name",psc.name "Subcategory Name",p.ProductID,p.ProductNumber,p.Name--p.name 'Product Name'--,/\*pc.ProductCategoryID,psc.ProductCategoryID,p.productid,\*/p.\*,psc.\*,pc.\*

from Production.ProductCategory pc

left join Production.ProductSubcategory psc

on pc.ProductCategoryID = psc.ProductCategoryID

left join Production.Product p

on psc.ProductSubcategoryID = p.ProductSubcategoryID

order by pc.name,psc.name,p.name  
  
Did all the subcategories have categories? If not, what did, or would, the query produce?  
 Yes, all subcategories have categories, if not there will be no relationship for that particular row and the query will produce less number of records.

* + 1. Based on your subcategory query, how many rows should your query have produced and did it?   
       **295 rows**
    2. Why did we use a left join?
       1. What would have happened if we had used a right join? How many rows would a right join have produced?   
          All records in product table would have been displayed with null values in some of the name fields. The right join would have produced **504 rows**
       2. What would have happened if we had used an inner join? How many rows would an inner join have produced?  
          The inner join would have produced **295 rows** same as left join because all subcategory id is present in product table and all categoryid are present in category table.
       3. What would have happened if we had used an outer join? How many rows would an outer join have produced?  
          It will display all the matching values first then the missing value in the right and then the missing values in the left. The full(Outer) join would have produced **504 rows.**

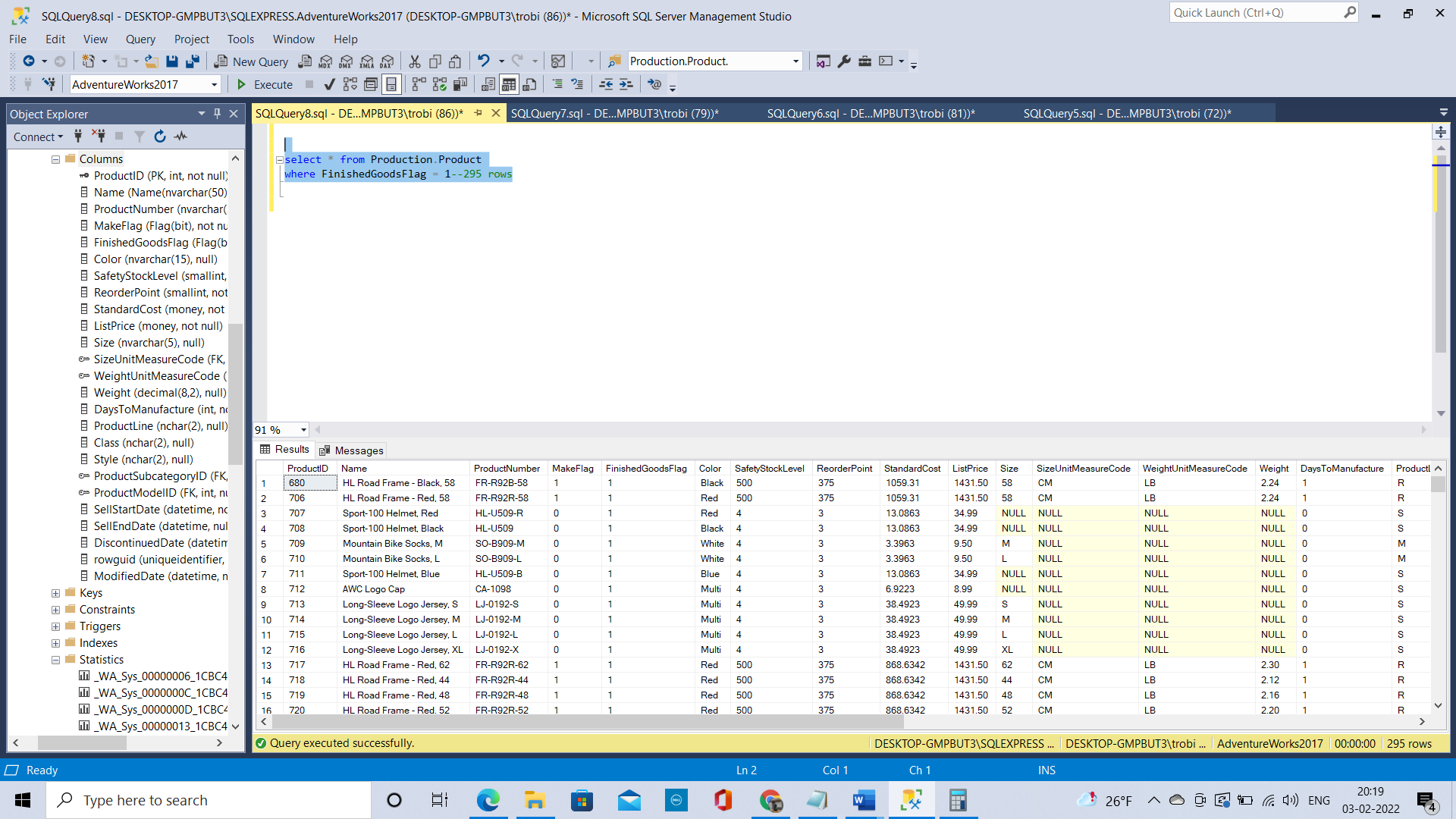
**Corelating fields**a) What fields appear to have a specific value when ProductSubcategoryID has a value and what fields appear to have a specific value when ProductSubcategoryID is NULL?  
  
when ProductSubcategoryID has a value  
FinishedGoodsFlag is 1 (Product is salable) This is a finished good which needs a lot of sub products which will be mentioned in the sub category table

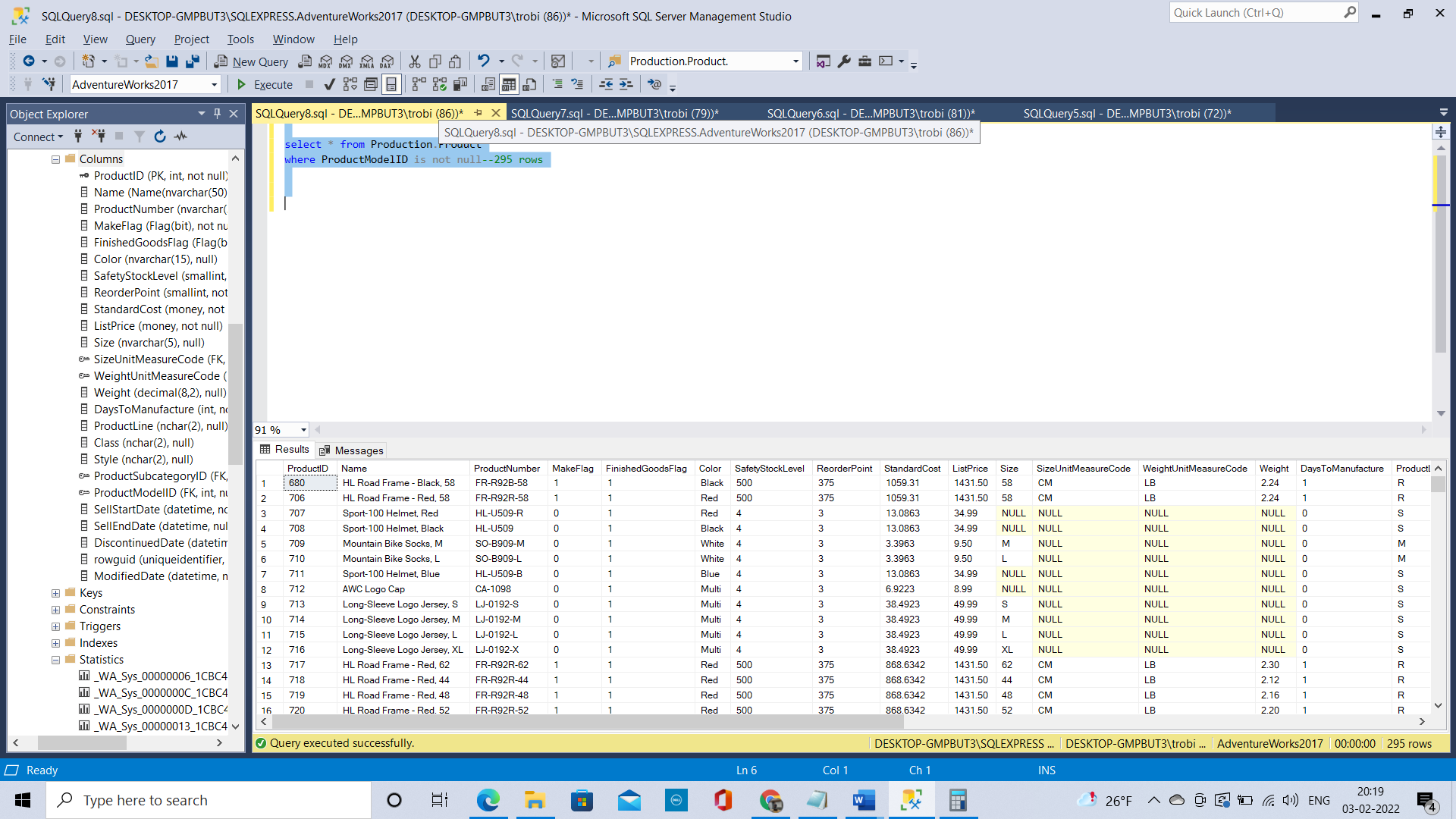
ProductModelID is not null  
  
when ProductSubcategoryID is NULL  
FinishedGoodsFlag is 0 (Product is not salable) It is a part which is required to make another finished product which we can’t sell.

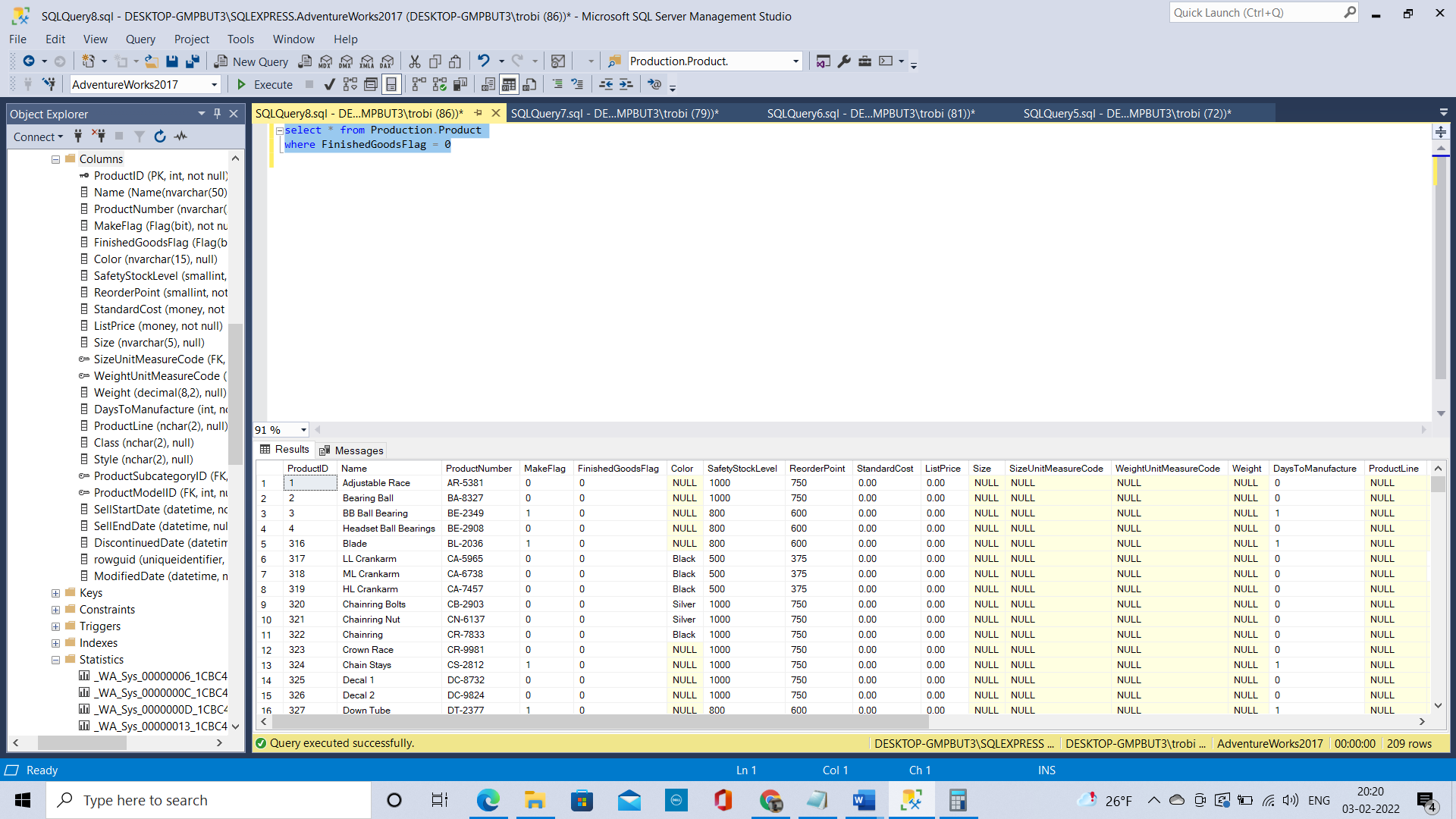
ProductModelID is null  
  
b) Think about and answer the following questions in the Correlations section of **A03.docx**

* + 1. Identify the fields (**there is more than one**) in Product that identify which products have a ProductSubCategory and which products do not have a ProductSubCategory. Include definitions from the data dictionary to support your findings.
       1. **These fields will not include** the ProductSubCategoryID field  
          **1. FinishedGoodsFlag (**1 is a finished good which needs a lot of sub products which will be mentioned in the sub category table on the contrary 0 is unfinished good for which ProductSubCategory value is null)

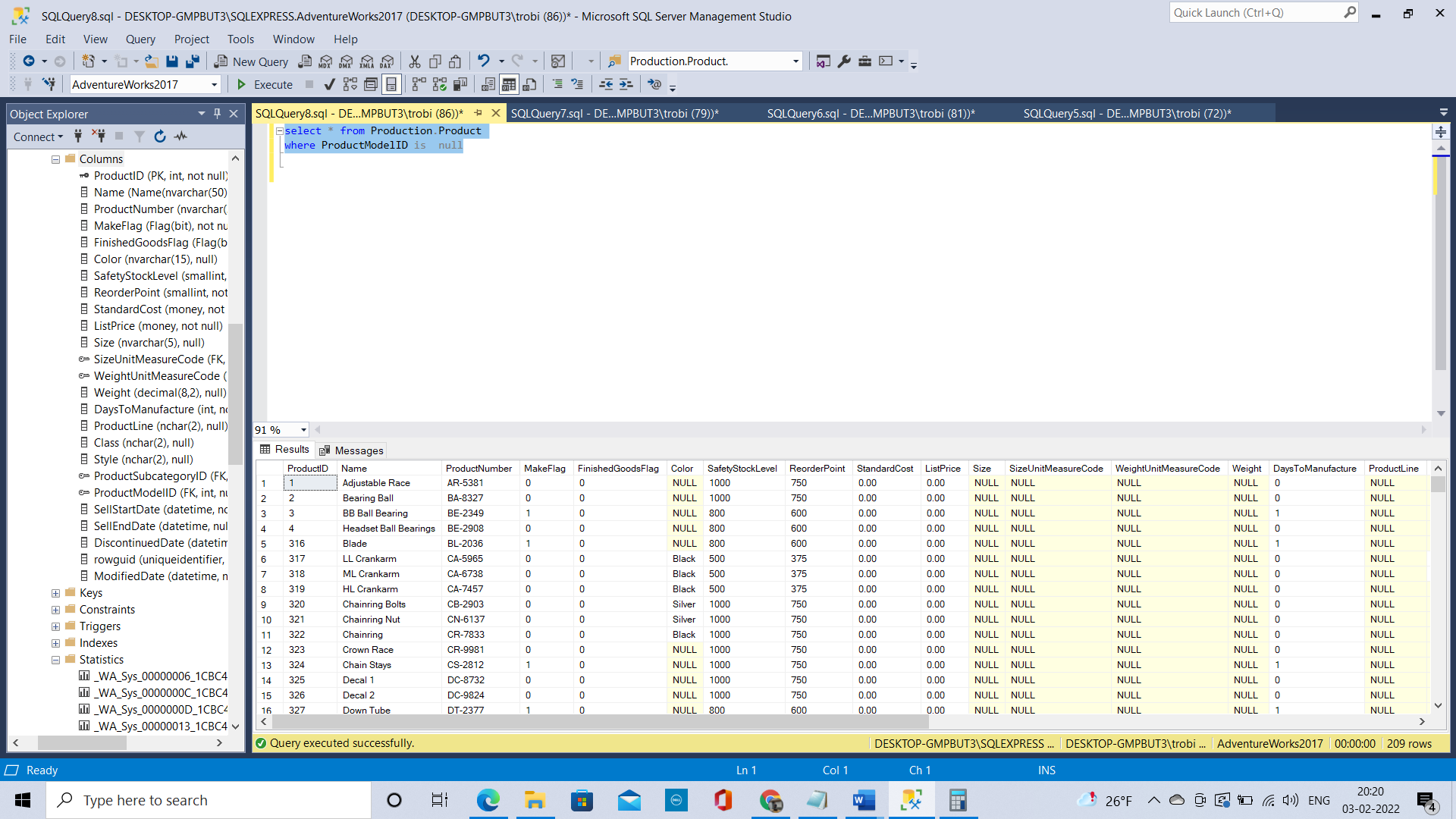
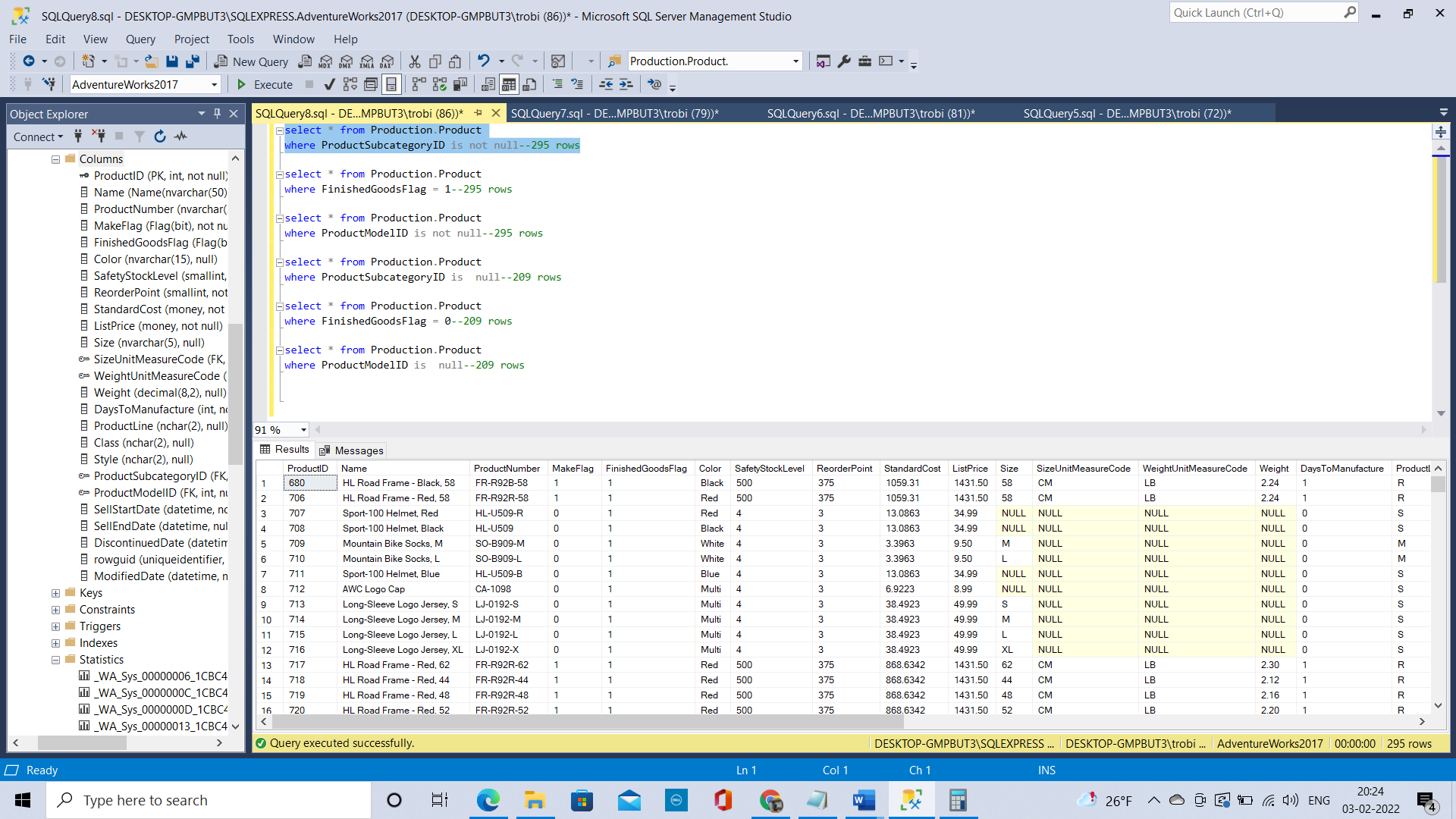
**2. ProductModelID (**Product is a member of this product model which is present in ProductModel table**)**

* + 1. Which values of this/these identified field(s) predict when a Product will have a ProductSubCategory and which values of these fields predict when a Product will NOT have a ProductSubCategory.   
         
       Product will have a ProductSubCategory  
        **1. FinishedGoodsFlag - 1** 2. **ProductModelID – not null**Product will not have a ProductSubCategory  
        **1. FinishedGoodsFlag - 0** 2. **ProductModelID – null**
  1. Take Screen Shot for each correlating field  
     **1. FinishedGoodsFlag – 1 295 rows**  
       
     

2. **ProductModelID – not null 295 rows**   
  
  
**1. FinishedGoodsFlag – 0 209 rows**



2. **ProductModelID –null 209 rows**

  
  
  
  
 **Final Code**select \* from Production.Product

where ProductSubcategoryID is not null--295 rows

select \* from Production.Product

where FinishedGoodsFlag = 1--295 rows

select \* from Production.Product

where ProductModelID is not null--295 rows

select \* from Production.Product

where ProductSubcategoryID is null--209 rows

select \* from Production.Product

where FinishedGoodsFlag = 0--209 rows

select \* from Production.Product

where ProductModelID is null--209 rows  
  
**Business Meaning**  
  
**Graduate students:** Use the ERD, the data dictionary definitions, and your query results to explain in an MS Word document the business meaning behind what you found in 5 above.  
  
  
The Product table has the details of all the products  
The Product table has a field called FinishedGoodsFlag if the value is 0 then the product is not salable  
if the value is 1 then the product can be sold.  
**From the corelation what we have found is for all the records where the productsubcategoryid is null the FinishedGoodsFlag is 0 which means the product cannot be sold.  
From this we can understand that it a subproduct which is used to make another finished product** which can be sold (Where the FinishedGoodsFlag value will be 1 and the productsubcategory will not be null)

If the **productsubcategoryid** is null against a record that product is a sub-product  
  
Similarly  
Another corelation is ProductModelId   
Sub-product (productsubcategoryid is null) is not a part of ProductModel   
Only the Finished goods (productsubcategoryid is not null) have a ProductModel which will be available in ProductModel Table.