ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»  
(РУТ (МИИТ))

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

ОТЧЁТ  
ПО ЛАБОРАТОРНОЙ РАБОТЕ №3

«Соединение данных из множества таблиц»

По дисциплине «Информационное обеспечение систем управления»

Выполнил: ст. гр. ТУУ – 411

Прокопьева Е.Е.

Проверил: к.т.н., доц.

Васильева М.А.

Москва 2021

Оглавление

[Цель работы 2](#_Toc89785847)

[Формулировка задания 2](#_Toc89785848)

[Выполнение лабораторной работы 3](#_Toc89785849)

[Вывод 10](#_Toc89785850)

Цель работы

Изучить операторы MS SQL Server, необходимые для работы с данными, находящимися в разных таблицах. Научиться создавать запросы на соединение данных из множества таблиц. Работа производится с учебной базой данной AdwentureWorks2019.

# Формулировка задания

1. Показать количество товаров в корзине покупателя (Quantity) и поля Name, ProductNumber, ShoppingCartID из таблиц Production.Product, Sales.ShoppingCartItem.
2. Показать список ID работников (BusinessEntityID), в котором указано имеет работник резюме или нет, из таблиц HumanResources.Employee, HumanResources.JobCandidate, используя LEFT OUTER JOIN.
3. Показать список товаров (поле Name), в котором указан, есть ли у товара отслеживаемая цена (поле ListPrice из Production.ProductListPriceHistory) или нет, из таблицы Production.ProductListPriceHistory, Production.Product, используя RIGHT OUTER JOIN.
4. Показать список имен (FirstName), которые имеют несколько фамилий (LastName) из таблицы Person.Person, используя SELF JOIN.
5. Показать список фамилий (LastName), которые имеют несколько имен (FirstName) из таблицы Person.Person, используя SELF JOIN.
6. Показать список продуктов из таблицы Production.ProductInventory, содержащихся на нескольких складах (LocationID), ProductID которых начинается с 3, используя SELF JOIN.
7. Показать комбинированный список таблиц Production.Product по полям Name, WeightUnitMeasureCode, Production.UnitMeasure по полям Name, UnitMeasureCode, используя UNION.
8. Показать список BusinessEntityID, которые содержатся в таблице Person.Person, но не содержатся в таблице HumanResources.Employee.

# Выполнение лабораторной работы

--Показать количество товаров в корзине покупателя (Quantity) и поля Name,

--ProductNumber, ShoppingCartID из таблиц Production.Product, Sales.ShoppingCartItem.

SELECT p.[Name]

, p.ProductNumber

, s.ShoppingCartID

FROM Production.Product AS p

INNER JOIN Sales.ShoppingCartItem AS s

ON p.ProductID = s.ProductID;

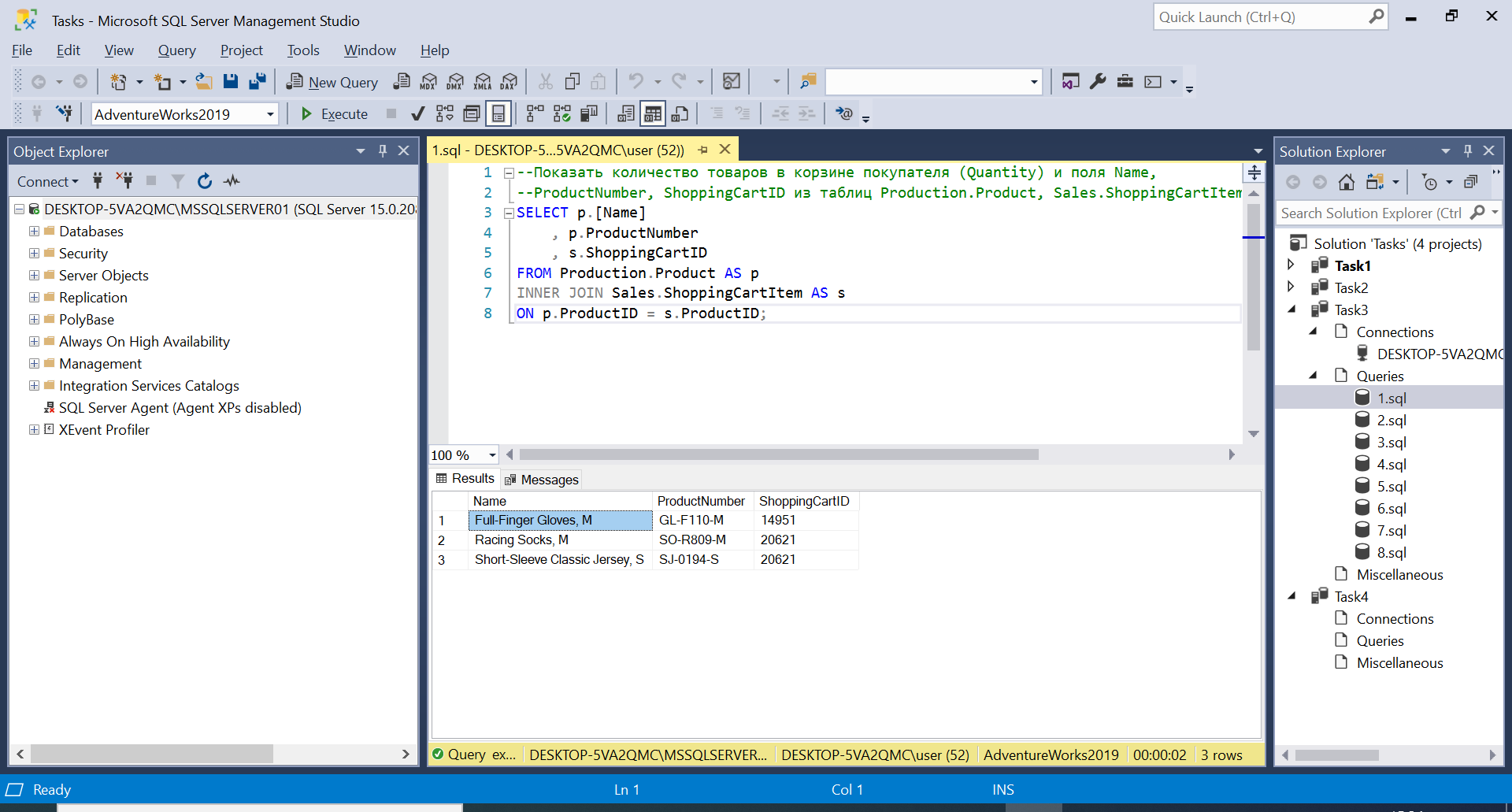


Рисунок 1 – Результат выполнения запроса №1

--Показать список ID работников (BusinessEntityID), в котором указано имеет

--работник резюме или нет, из таблиц HumanResources.Employee,

--HumanResources.JobCandidate, используя LEFT OUTER JOIN

SELECT e.BusinessEntityID

, j.[Resume]

FROM HumanResources.Employee AS e

LEFT OUTER JOIN HumanResources.JobCandidate AS j

ON e.BusinessEntityID = j.BusinessEntityID

WHERE j.[Resume] IS NOT NULL;

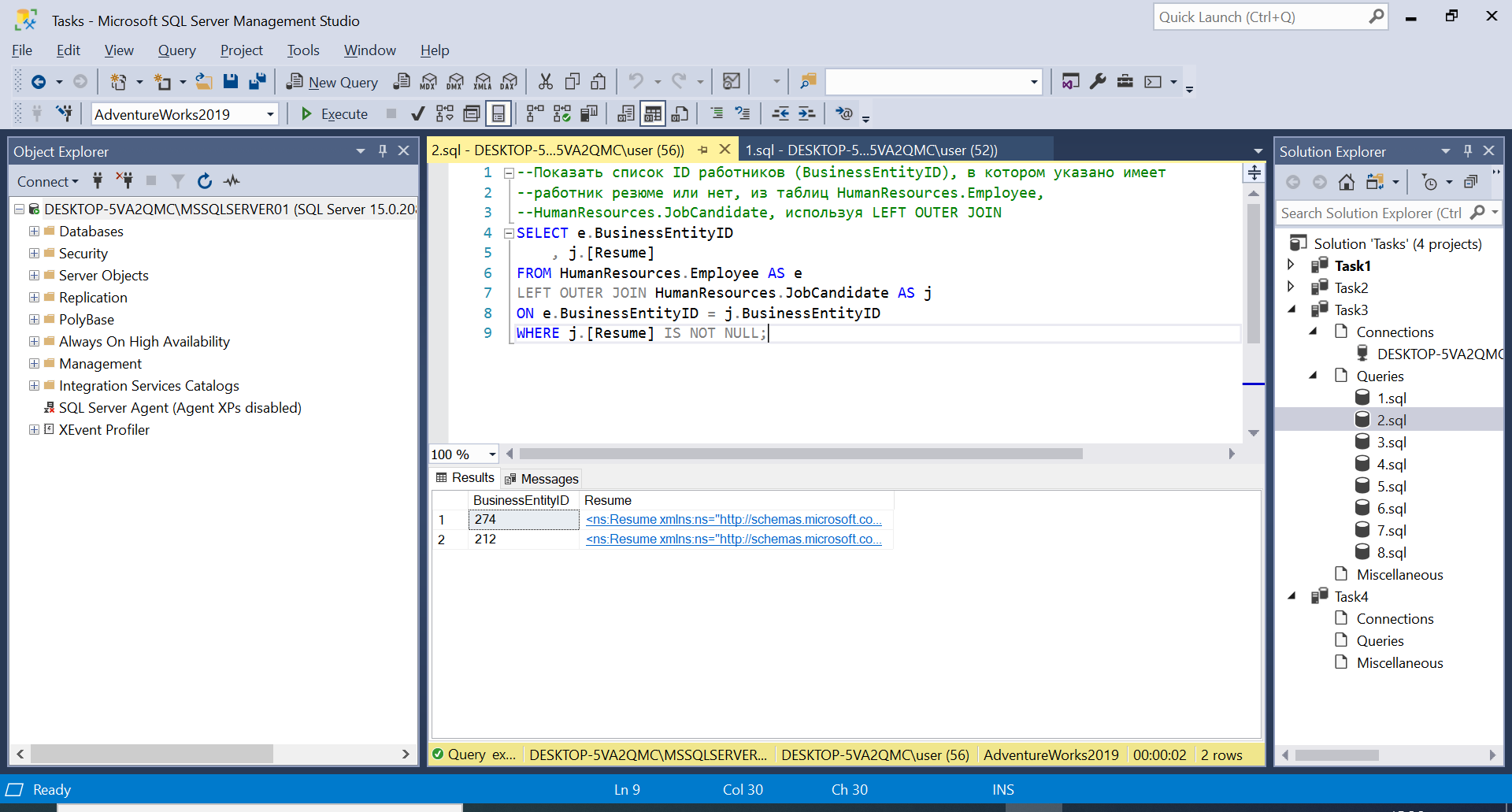


Рисунок 2 – Результат выполнения запроса №2

--Показать список товаров (поле Name), в котором указан, есть ли у товара

--отслеживаемая цена (поле ListPrice из Production.ProductListPriceHistory) или нет,

--из таблицы Production.ProductListPriceHistory, Production.Product,

--используя RIGHT OUTER JOIN.

SELECT p.[Name]

, h.ListPrice

FROM Production.ProductListPriceHistory AS h

RIGHT OUTER JOIN Production.Product AS p

ON h.ProductID = p.ProductID

WHERE h.ListPrice IS NOT NULL;

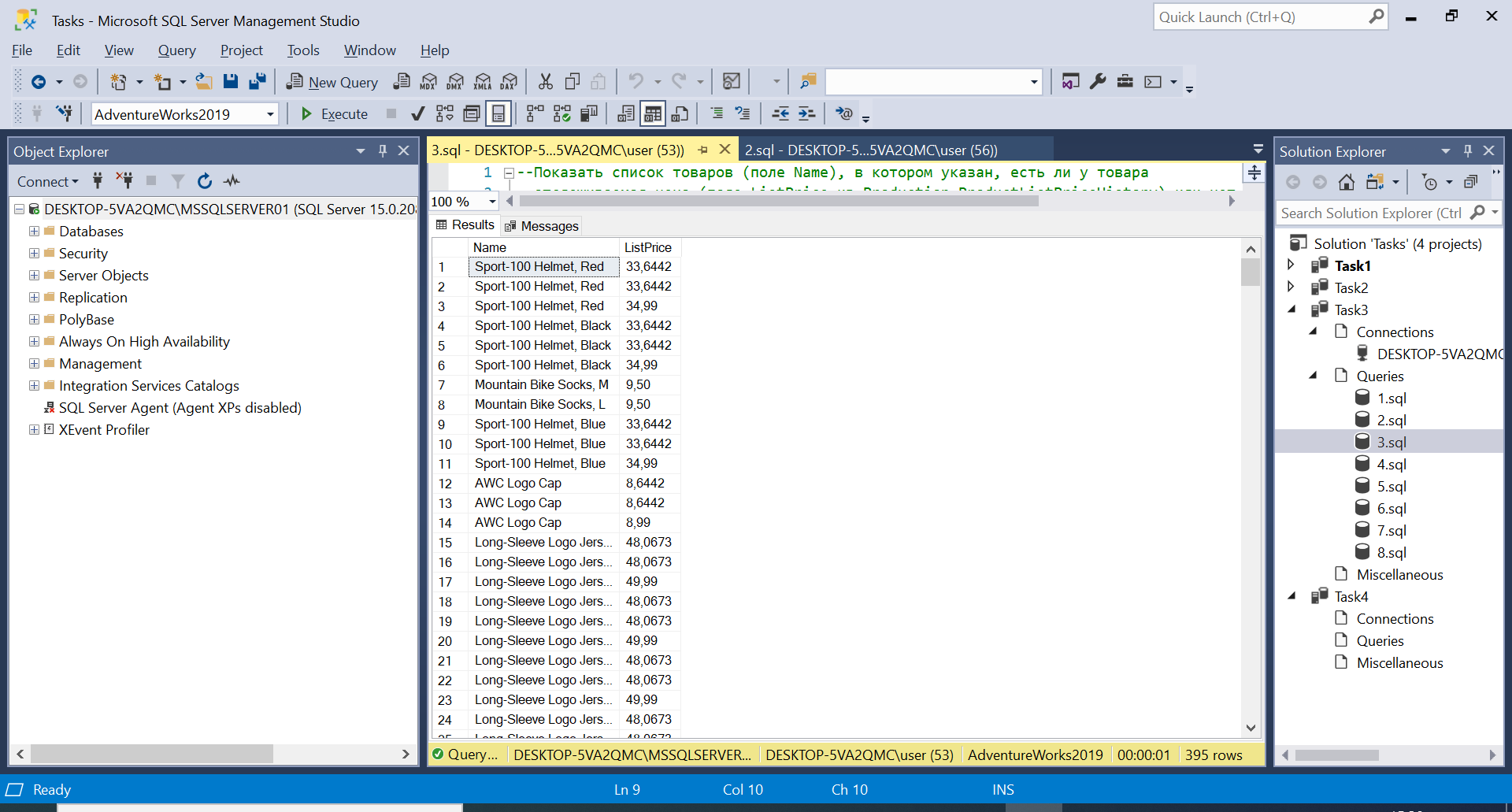


Рисунок 3 – Результат выполнения запроса №3

--Показать список имен (FirstName), которые имеют несколько фамилий (LastName)

--из таблицы Person.Person, используя SELF JOIN.

SELECT DISTINCT p1.FirstName

, p1.LastName

FROM Person.Person AS p1

INNER JOIN Person.Person AS p2

ON p1.FirstName = p2.FirstName

AND p1.LastName <> p2.LastName;

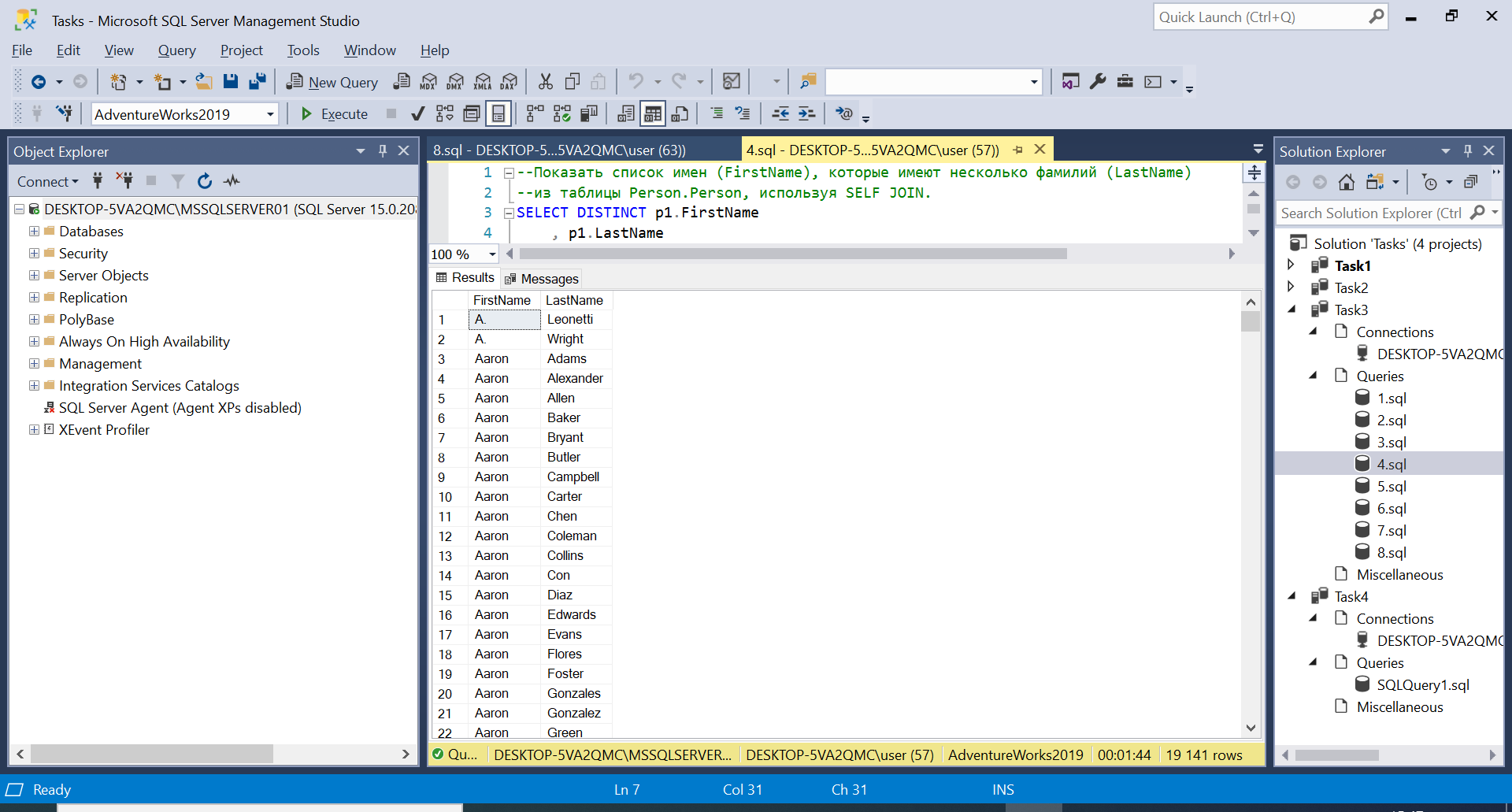


Рисунок 4 – Результат выполнения запроса №4

--Показать список фамилий (LastName), которые имеют несколько имен (FirstName)

--из таблицы Person.Person, используя SELF JOIN

SELECT DISTINCT p1.LastName

, p1.FirstName

FROM Person.Person AS p1

INNER JOIN Person.Person AS p2

ON p1.LastName = p2.LastName

AND p1.FirstName <> p2.FirstName;

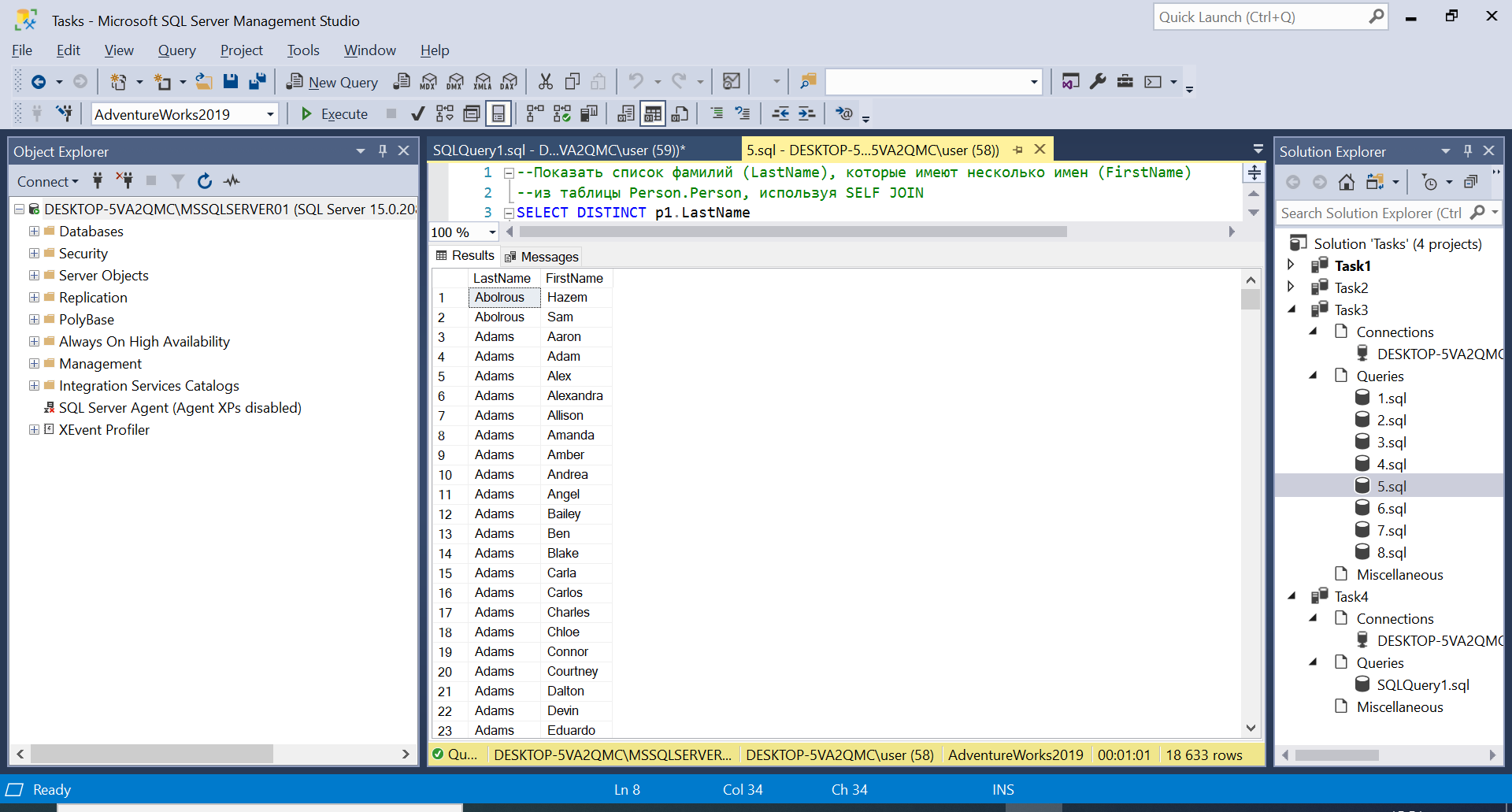


Рисунок 5 – Результат выполнения запроса №5

--Показать список продуктов из таблицы Production.ProductInventory, содержащихся

--на нескольких складах (LocationID), ProductID которых начинается с 3,

--используя SELF JOIN.

SELECT DISTINCT i1.ProductID

, i2.LocationID

FROM Production.ProductInventory AS i1

INNER JOIN Production.ProductInventory AS i2

ON i1.ProductID = i2.ProductID

WHERE i1.ProductID >= 3

AND i2.ProductID >= 3;

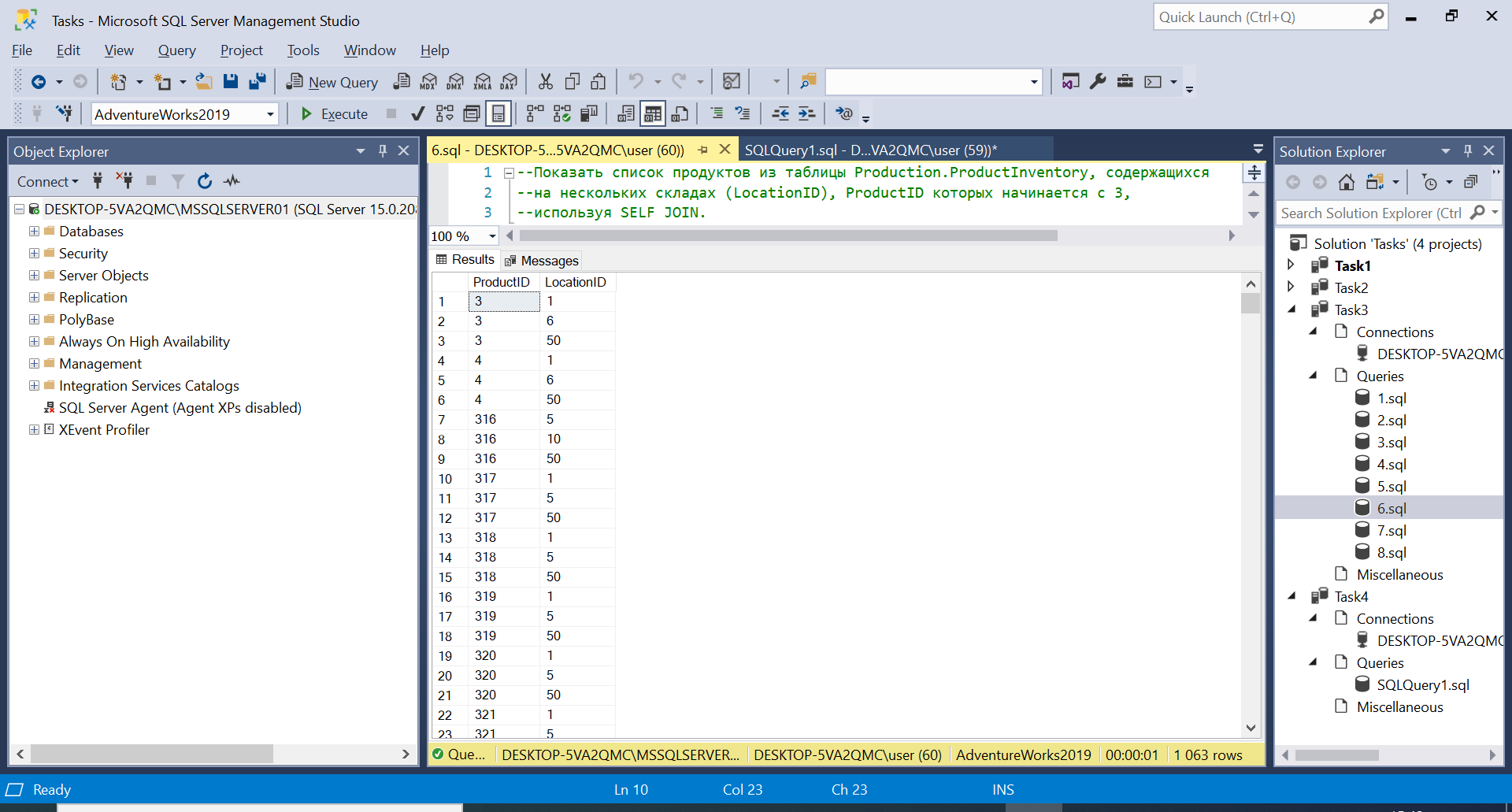


Рисунок 6 – Результат выполнения запроса №6

--Показать комбинированный список таблиц Production.Product по полям Name,

--WeightUnitMeasureCode, Production.UnitMeasure по полям Name, UnitMeasureCode,

--используя UNION

SELECT [Name]

, WeightUnitMeasureCode

FROM Production.Product

UNION

SELECT [Name]

, UnitMeasureCode

FROM Production.UnitMeasure;

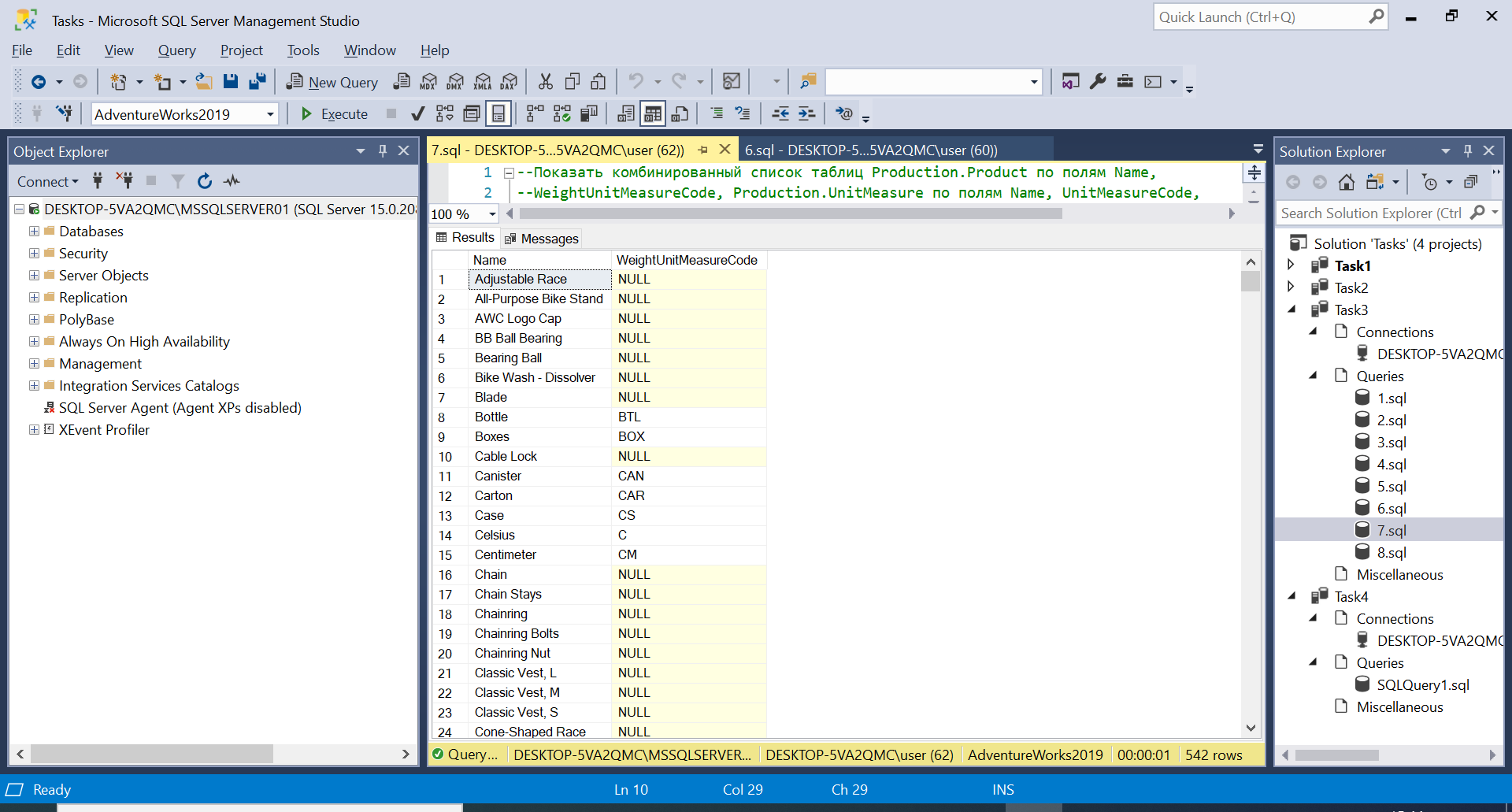


Рисунок 7 – Результат выполнения запроса №7

--Показать список BusinessEntityID, которые содержатся в таблице Person.Person,

--но не содержатся в таблице HumanResources.Employee

SELECT BusinessEntityID

FROM Person.Person

INTERSECT

SELECT BusinessEntityID

FROM HumanResources.Employee;

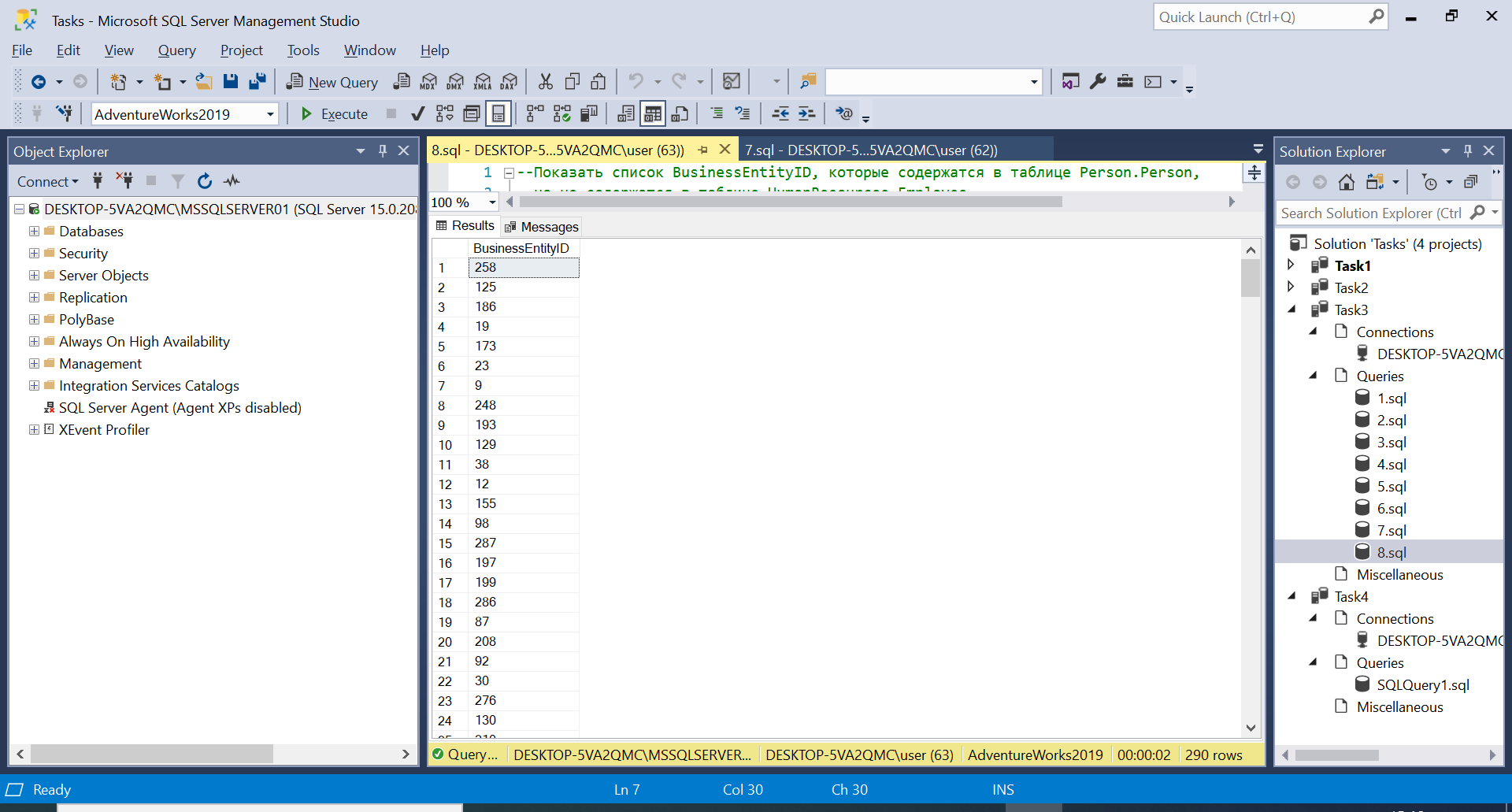


Рисунок 8 – Результат выполнения запроса №8

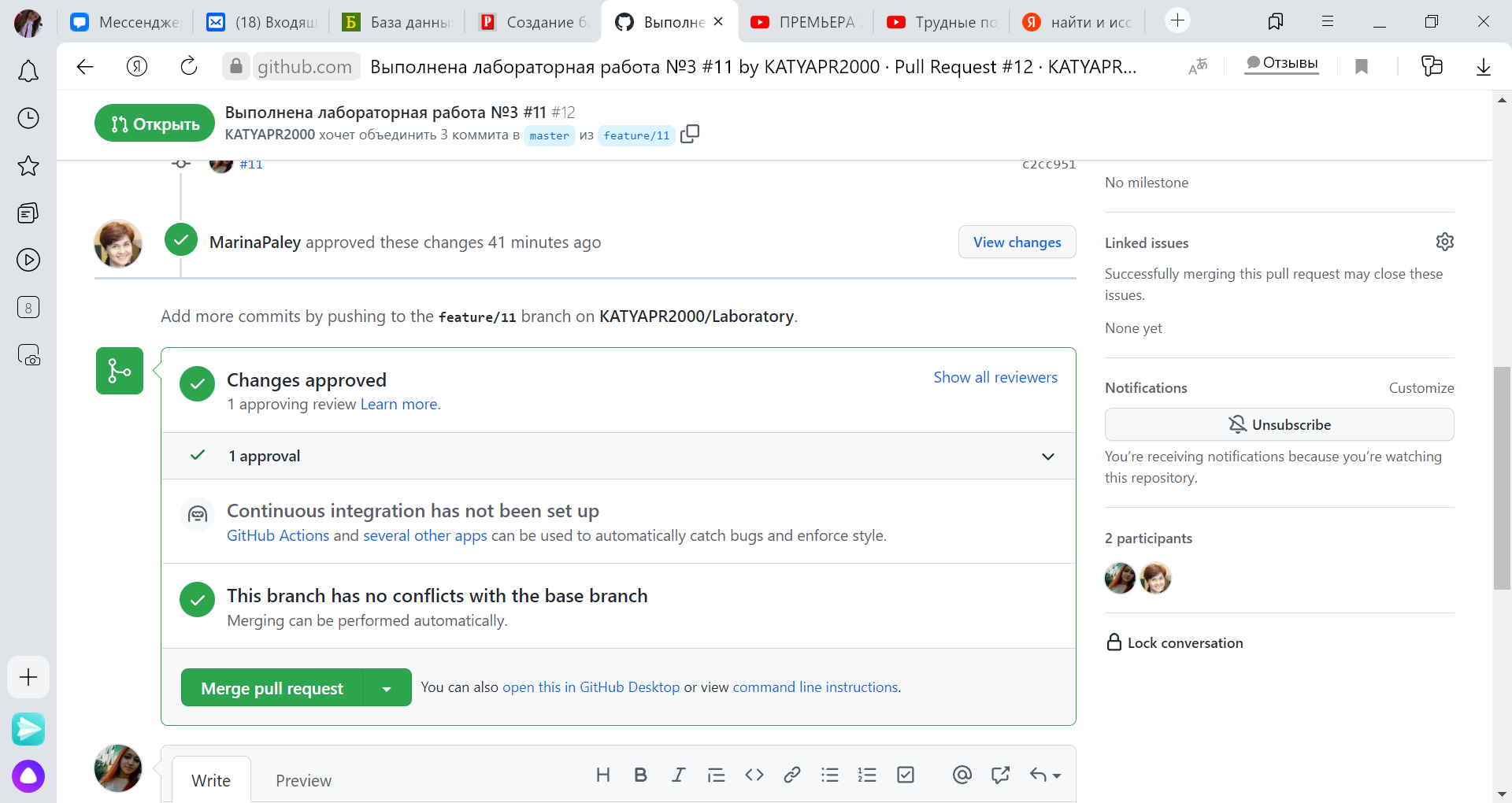


Рисунок 9 – Одобренное решение

Вывод

Изучены операторы MS SQL Server, необходимые для работы с данными, находящимися в разных таблицах. Научилась создавать запросы на соединение данных из множества таблиц. Работа производилась с учебной базой данной AdwentureWorks2019.