**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

# Кафедра Систем автоматизированного проектирования

**ОТЧЕТ**

# по лабораторной работе №2 по дисциплине «Базы данных»

**Тема: Выполнение выборки из таблицы**

|  |  |  |
| --- | --- | --- |
| Студенты гр. 2308 |  | Попов Н.А. |
|  |  | Бебия Р.А. |
|  |  | Чиков А.А. |
| Преподаватель |  | Горяинов С.В. |

Санкт-Петербург 2024

# Цель работы

Знакомство с опциями GROUP BY и HAVING, а также агрегированием данных

**Используемая база данных**

AdventureWorks

# Выполнение работы

**Упражнение 1 – использование ключевого слова TOP в команде SELECT**

Запрос 1: Вывести значения полей и отсортировать запрос по Bonus по убыванию

SELECT SalesPersonID, Bonus

FROM Sales.SalesPerson ORDER BY Bonus DESC;

Результат выполнения запроса показан на рисунке 1.

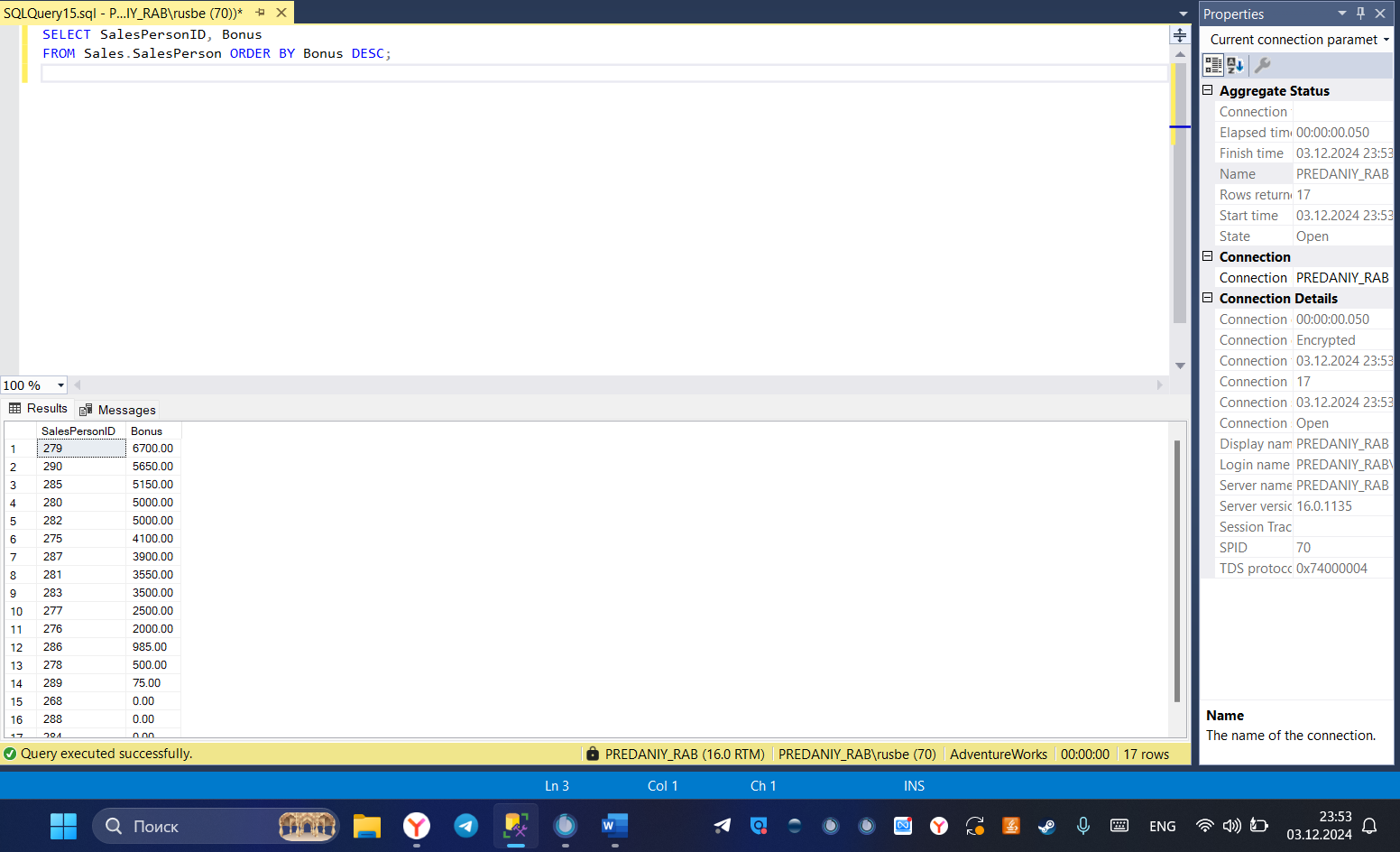


Рисунок 1 – строк результата выполнения запроса

Запрос 2: Модификация запроса, чтобы возвращались 4 самых больших бонуса

SELECT TOP(4) SalesPersonID, Bonus

From Sales.SalesPerson ORDER BY Bonus DESC;

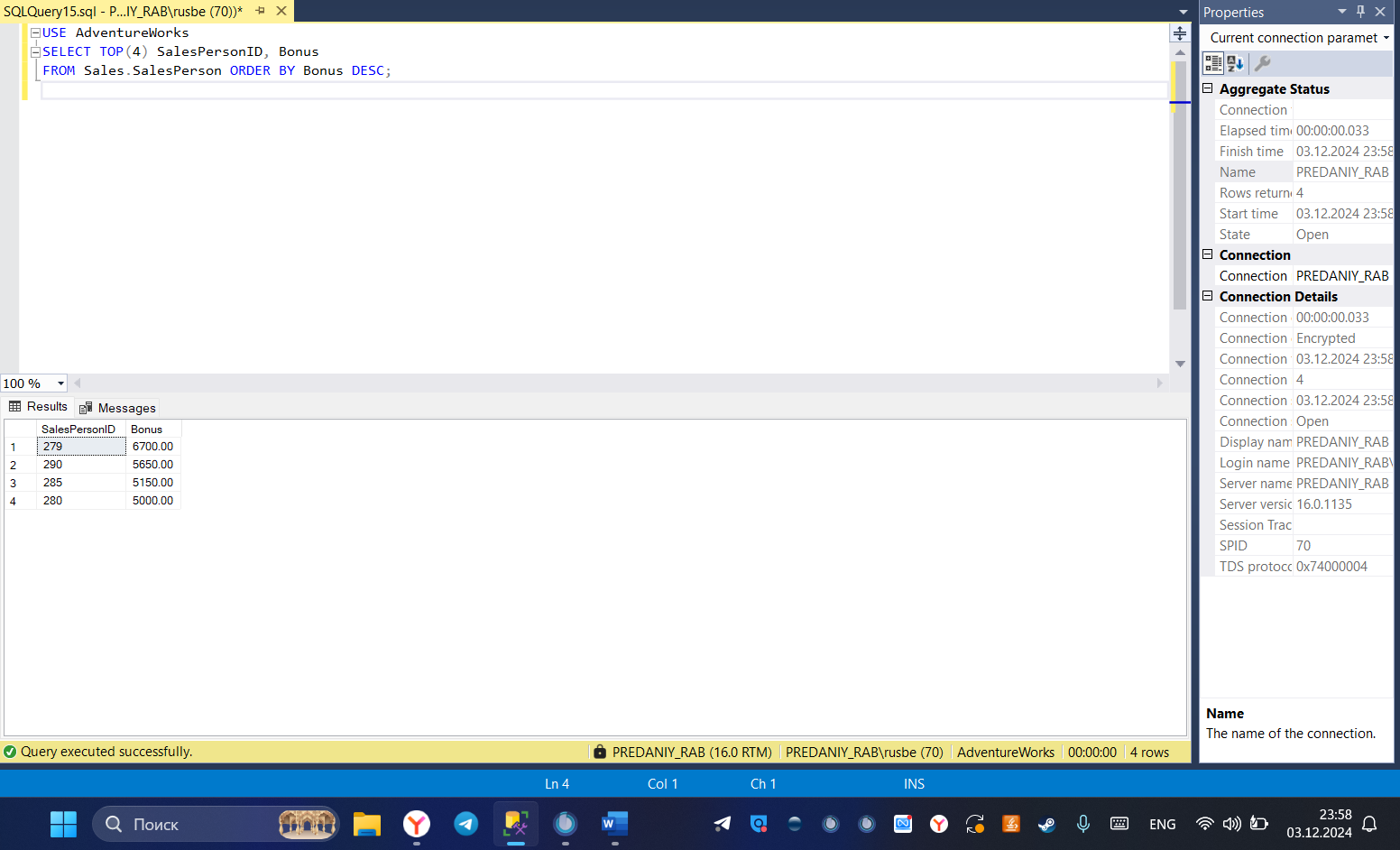
Результат выполнения показан на рисунке 2.

Рисунок 2 – Первые 4 строк результата выполнения запроса

Запрос 3: SELECT с условием WITH TIES  
SELECT TOP (4) WITH TIES SalesPersonID, Bonus

FROM Sales.SalesPersonID ORDER BY Bonus DESC;

Результат выполнения запроса представлен на рисунке 3

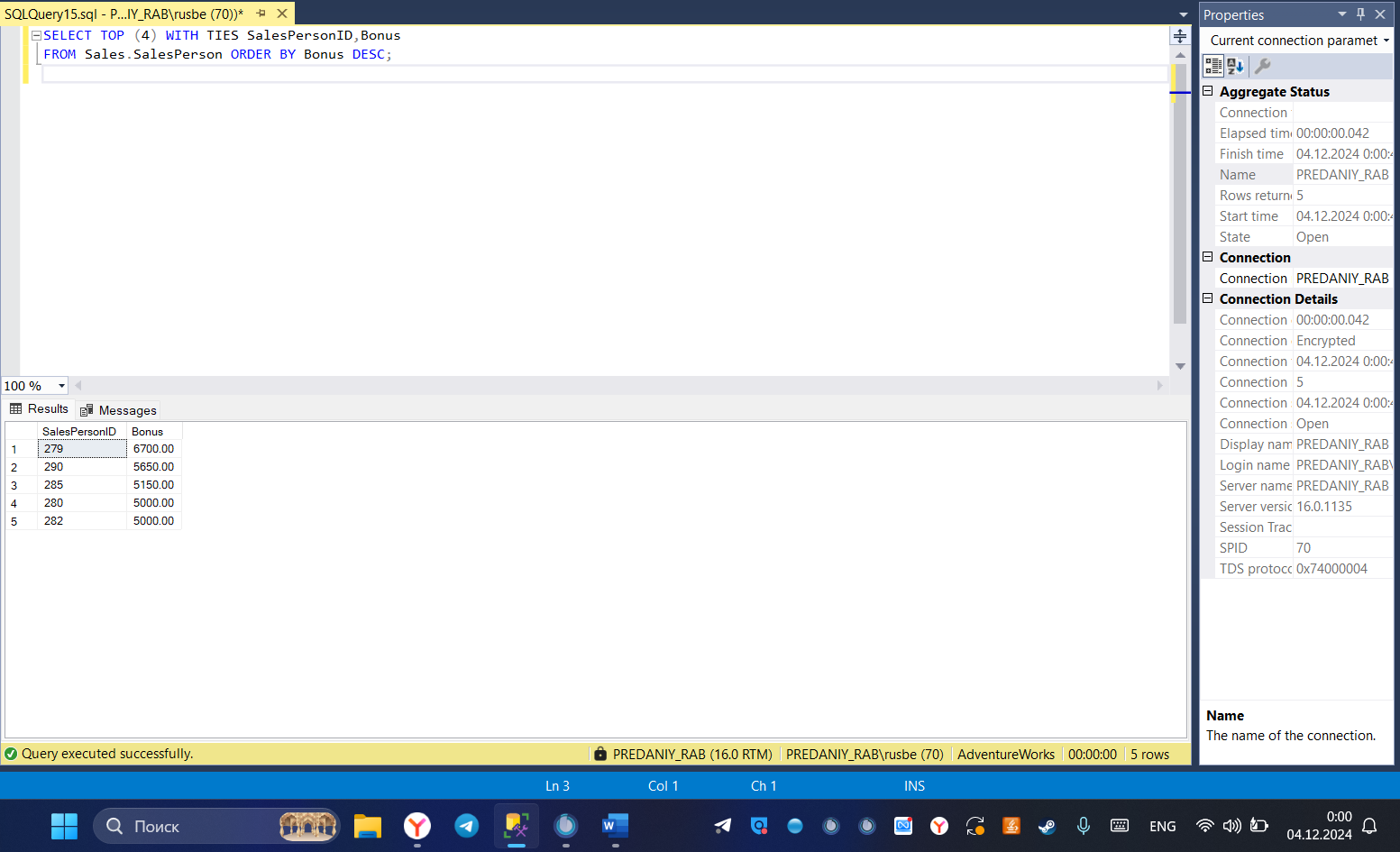


Рисунок 3 – Первые 5 строк результата выполнения запроса

**Упражнение 2 – использование агрегатных функций и конструкций GROUP BY и HAVING**

Запрос 1: Подсчитать общее количество строк в таблице Employee

SELECT COUNT(\*) AS TotalEmployees

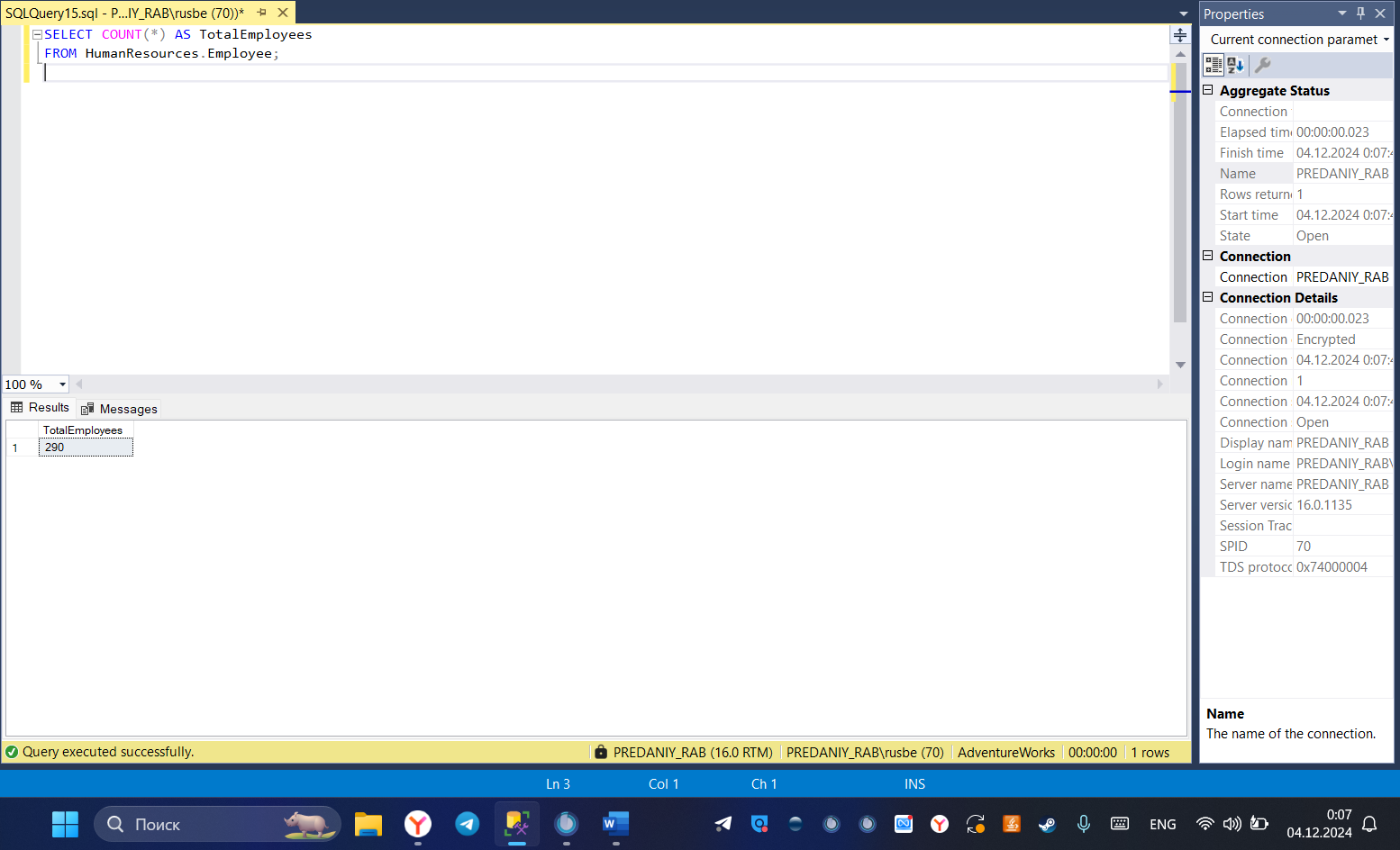
FROM HumanResourсes.Employee; 

Рисунок 4 – Результат выполнения запроса

Запрос 2: Подсчитать общее количество сотрудников, имеющих менеджеров

SELECT COUNT (\*) AS TotalEmployees

FROM HumanResources.Employee

WHERE ManagerID IS NOT NULL

Результат выполнения запроса представлен на рисунке

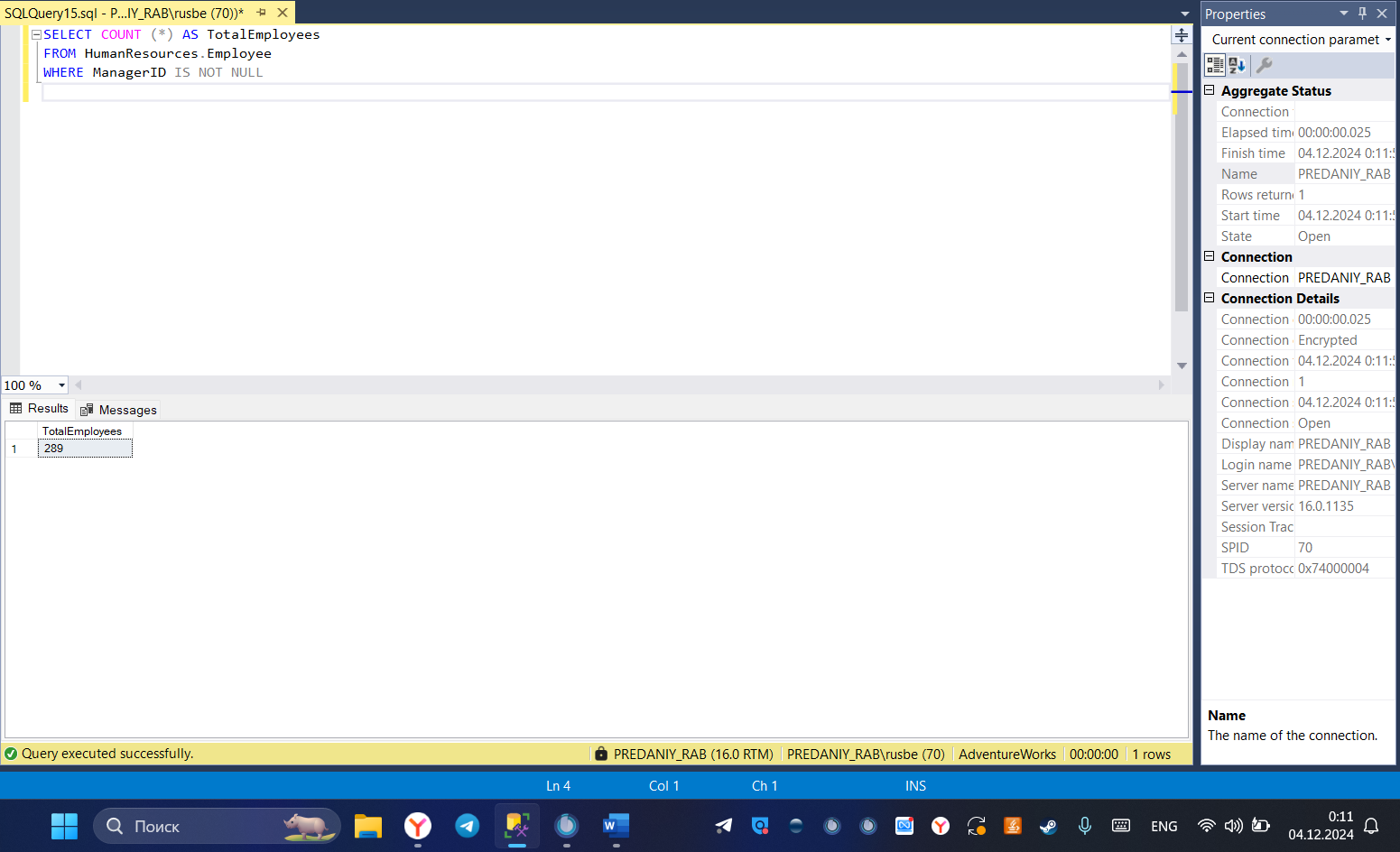


Рисунок 5 – Результат выполнения запроса с WHERE

Запрос 3: Подсчитать поле OrderQty для каждого продукта (ProductID)

SELECT ProductID, SUM(OrderQty) AS TotalOrderQty

FROM Sales.SalesOrderDetail

GROUP BY ProductID

Результат выполнения запроса представлен на рисунке

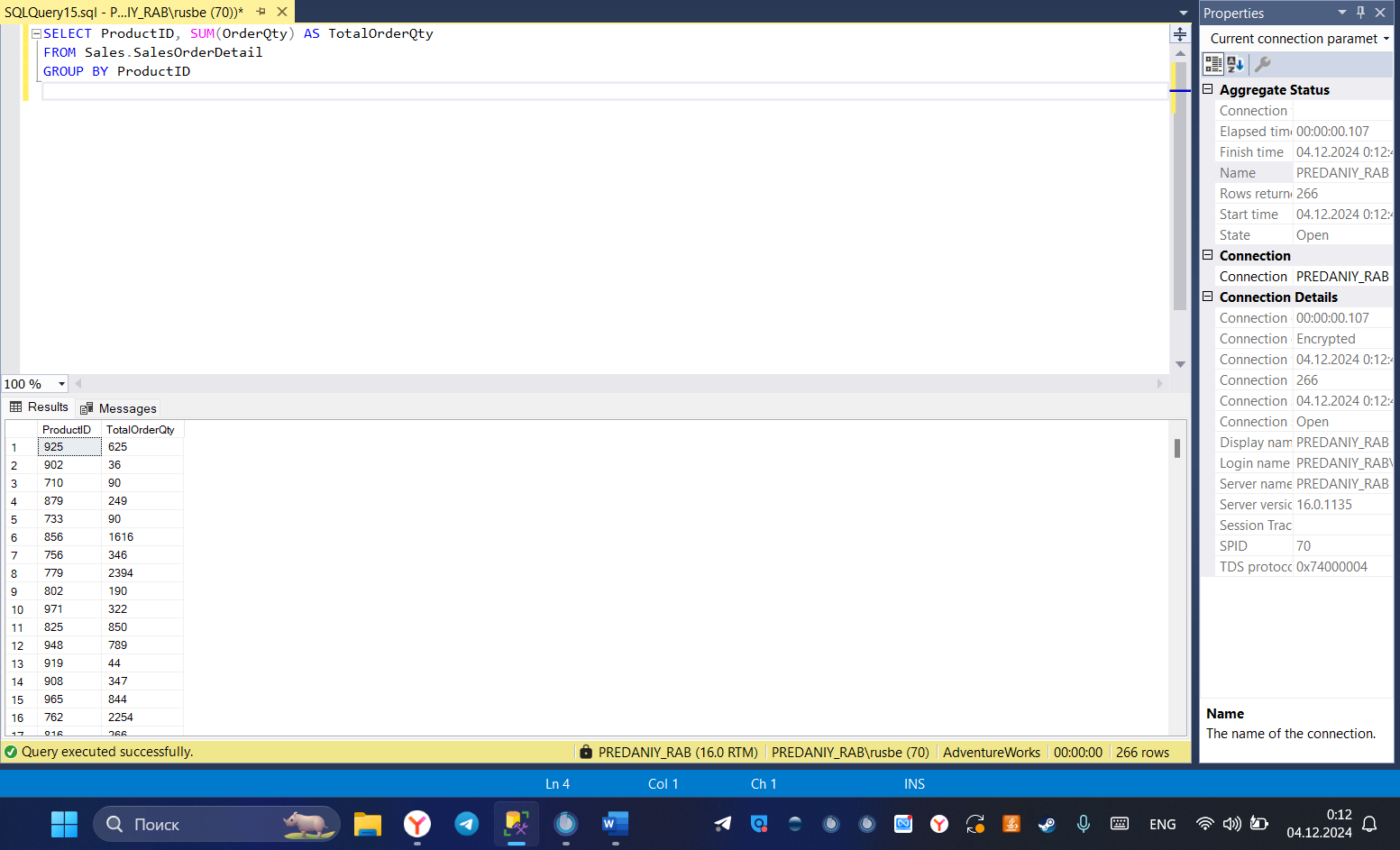


Рисунок 6 – Результат выполнения запроса с SUM

Запрос 4: Отсортировать результат запроса по суммарному количеству

SELECT ProductID, SUM(OrderQty) AS TotalOrderQty

FROM Sales.SalesOrderDetail

GROUP BY ProductID

ORDER BY TotalOrderQty DESC

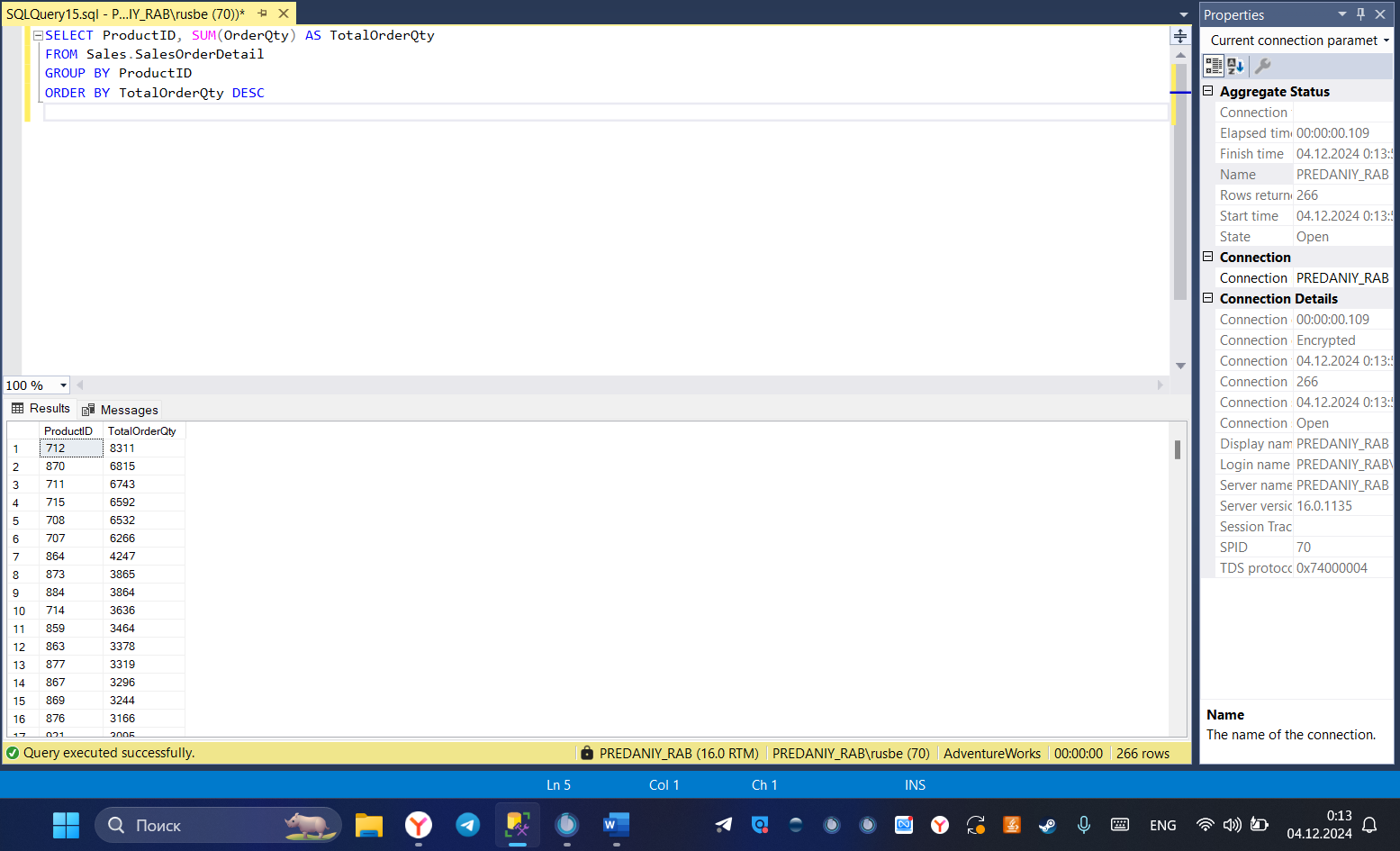


Рисунок 7 – Результат выполнения запроса по суммарному количеству

Анализ: Запрос корректно выводит суммированные значения, в порядке убывания

Запрос 5: Отсортировать результат запроса по суммарному количеству с модификацией (>=2000)

SELECT ProductID, SUM (OrderQty) AS TotalOrderQty

FROM Sales.SalesOrderDetail

GROUP BY ProductID

HAVING SUM (OrderQty) >= 2000

ORDER BY TotalOrderQty DESC

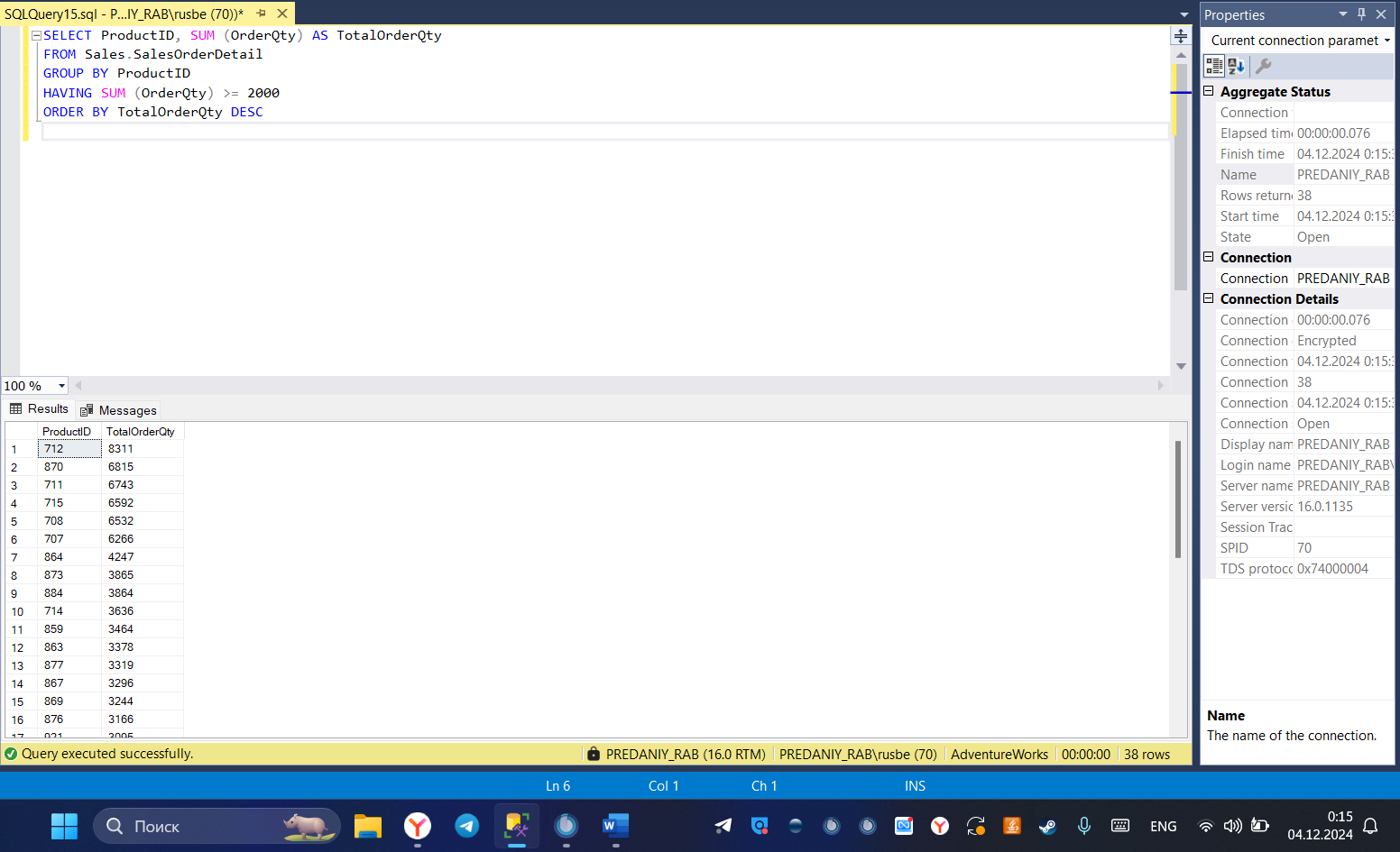


Рисунок 8 –Выполнение запроса по суммарному количеству с условием

Анализ: Запрос корректно выводит значения, удовлетворяя условие

Запрос 6: Написать запрос к таблице Sales.SalesOrderDetail, выполнить группировку

SELECT ProductID, SpecialOfferID, AVG(UnitPrice) AS AvgUnitPrice, SUM (LineTotal) AS TotallineAmount

FROM Sales.SalesOrderDetail

GROUP BY ProductID, SpecialOfferID

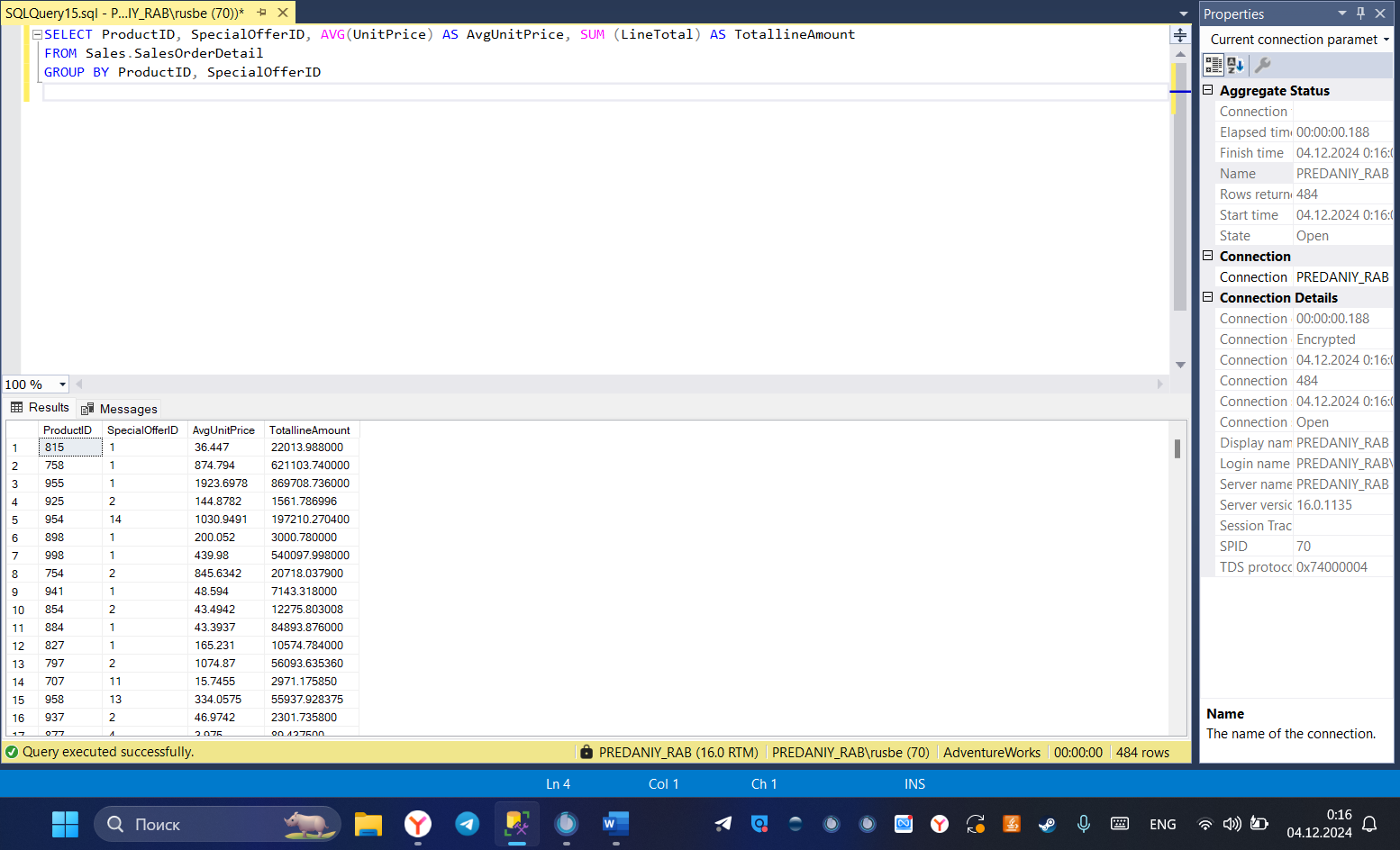


Рисунок 9 –Выполнение запроса с использованием GROUP BY для нескольких групп

Запрос 7: Написать запрос к таблице Sales.SalesOrderDetail с сортировкой по ProductID по возрастанию, дать псевдонимы, выполнить группировку

SELECT ProductID, SpecialOfferID, AVG(UnitPrice) AS AvgUnitPrice,

SUM (LineTotal) AS TotallineAmount

FROM Sales.SalesOrderDetail

GROUP BY ProductID, SpecialOfferID

ORDER BY ProductID

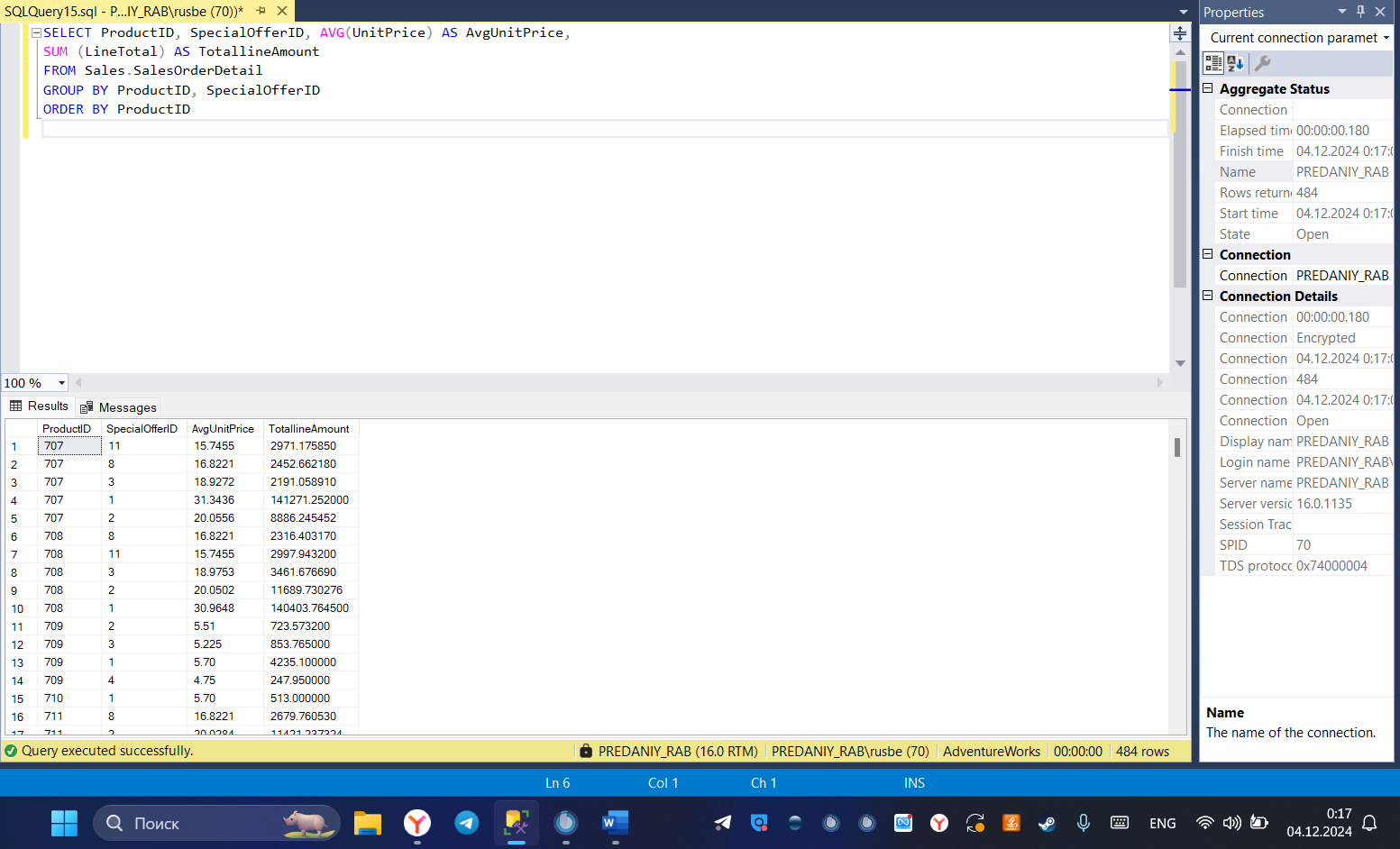


Рисунок 10 –Выполнение запроса с использованием GROUP BY для нескольких групп с сортировкой и группировкой

**Упражнение 3 – использование операторов ROLLUP и CUBE**

Запрос 1:Написать запрос к таблице Sales.SalesOrderDetail,выполнить группировку ,дать псевдоним

SELECT SalesQuota, SUM (SalesYTD) AS TotalSalesYTD

FROM Sales.SalesPerson

GROUP BY SalesQuota

Результат выполнения запроса представлен на рисунке

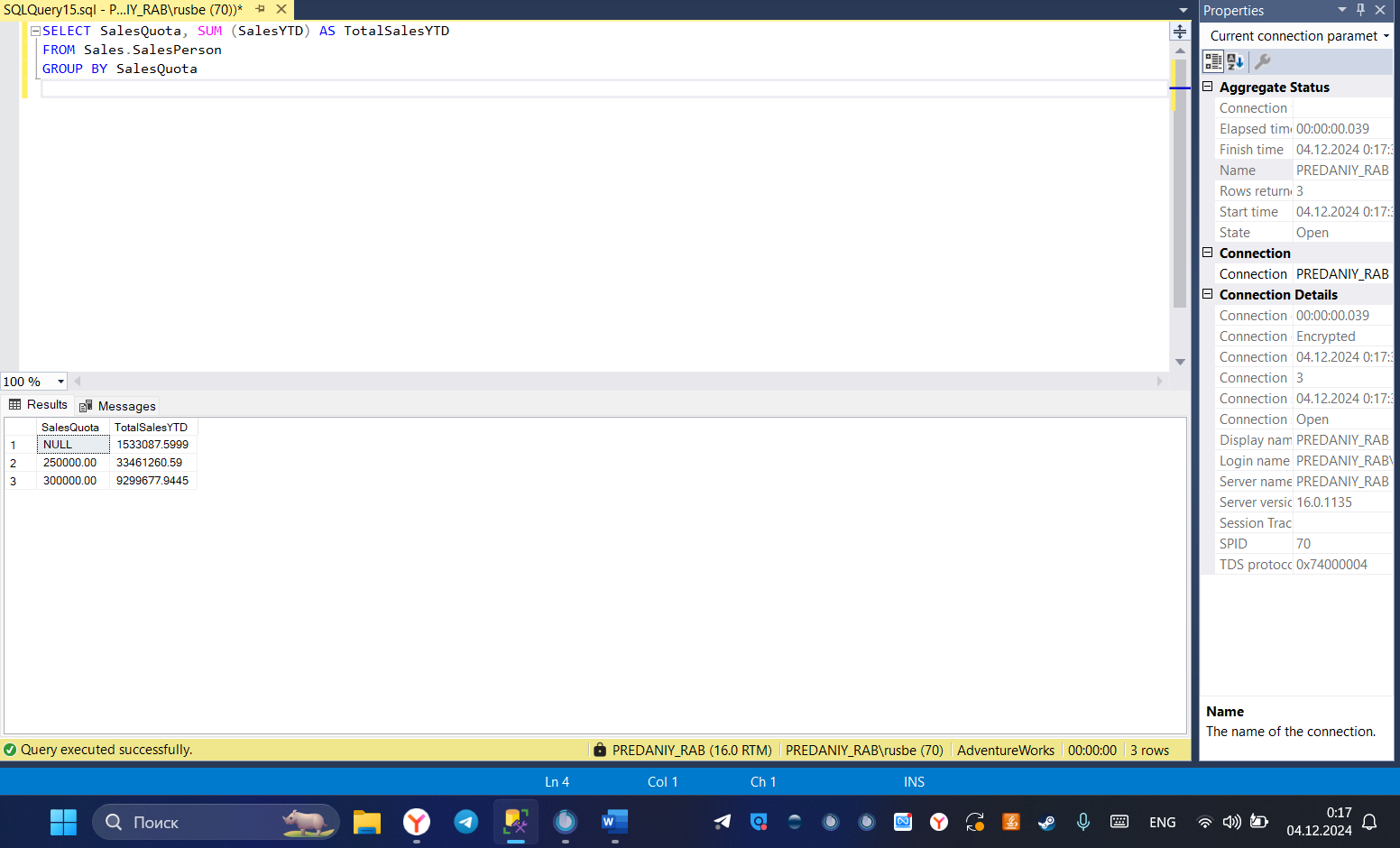


Рисунок 11 –Выполнение запроса с использованием GROUP BY для нескольких групп с суммарным значением и группировкой

Запрос 2: Написать запрос к таблице Sales.SalesOrderDetail, выполнить группировку, дать псевдоним и применить GROUPING

SELECT SalesQuota, SUM (SalesYTD) ‘TotalSalesYTD’ , GROUPING(SalesQuota) AS‘Grouping’

FROM Sales.SalesPerson

GROUP BY SalesQuota WITH ROLLUP

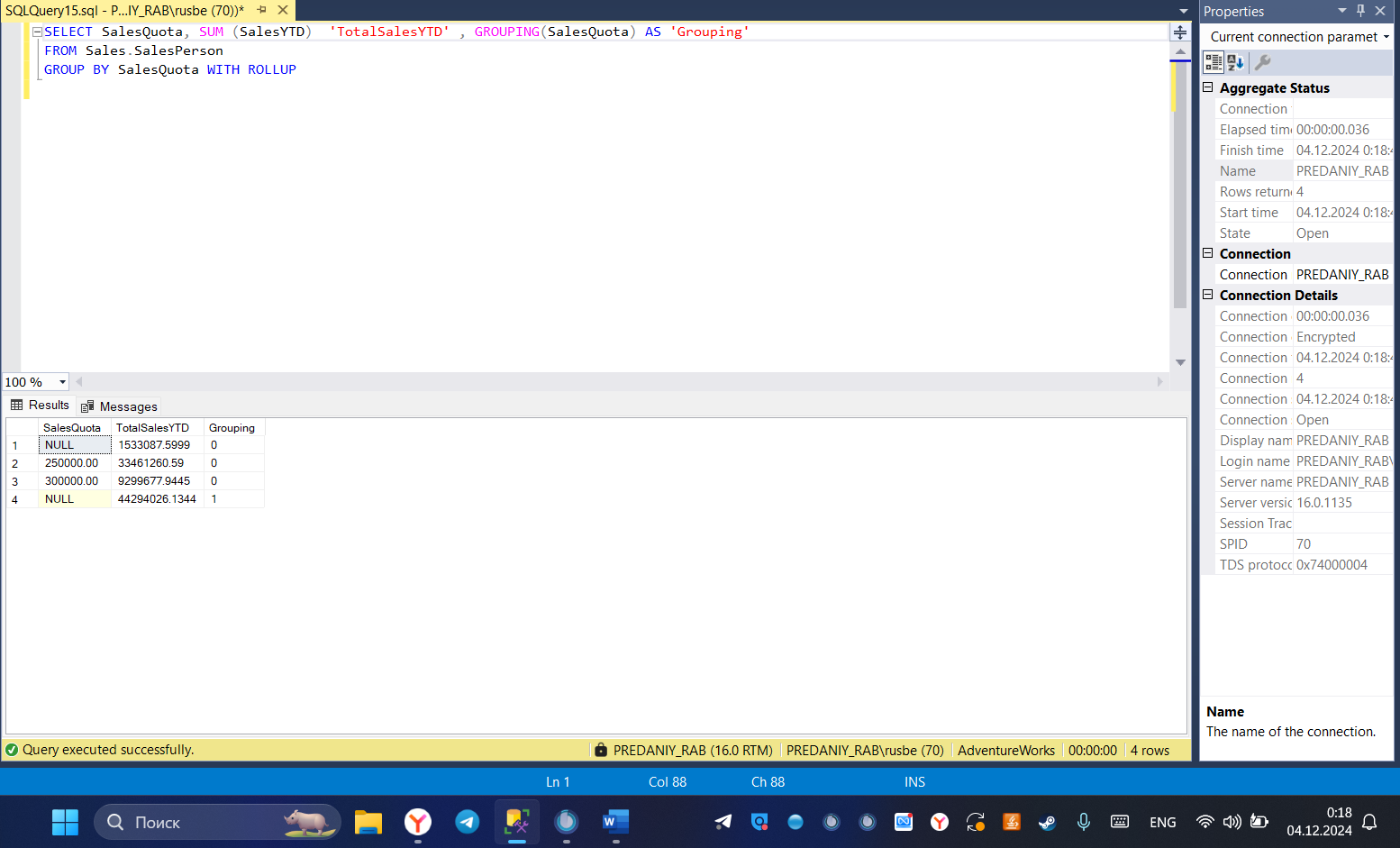


Рисунок 12 –Выполнение запроса с использованием ROLLUP

NULL в поле SalesQuota указывает на то, что это сводная строка, обобщающая данные по всем SalesQuota. NULL значения позволяют легче интерпретировать итоги, отделяя их от обычных записей, что полезно для анализа данных.

Запрос 3: Написать запрос к таблице Sales.SalesOrderDetail, выполнить группировку, сортировку и ограничение < 5.00

SELECT ProductID, OrderQty SUM (LineTotal) AS TotalLineAmount

FROM Sales.SalesOrderDetail

WHERE UnitPrice < 5.00

GROUP BY CUBE (ProductID,OrderQty)

ORDER BY ProductID

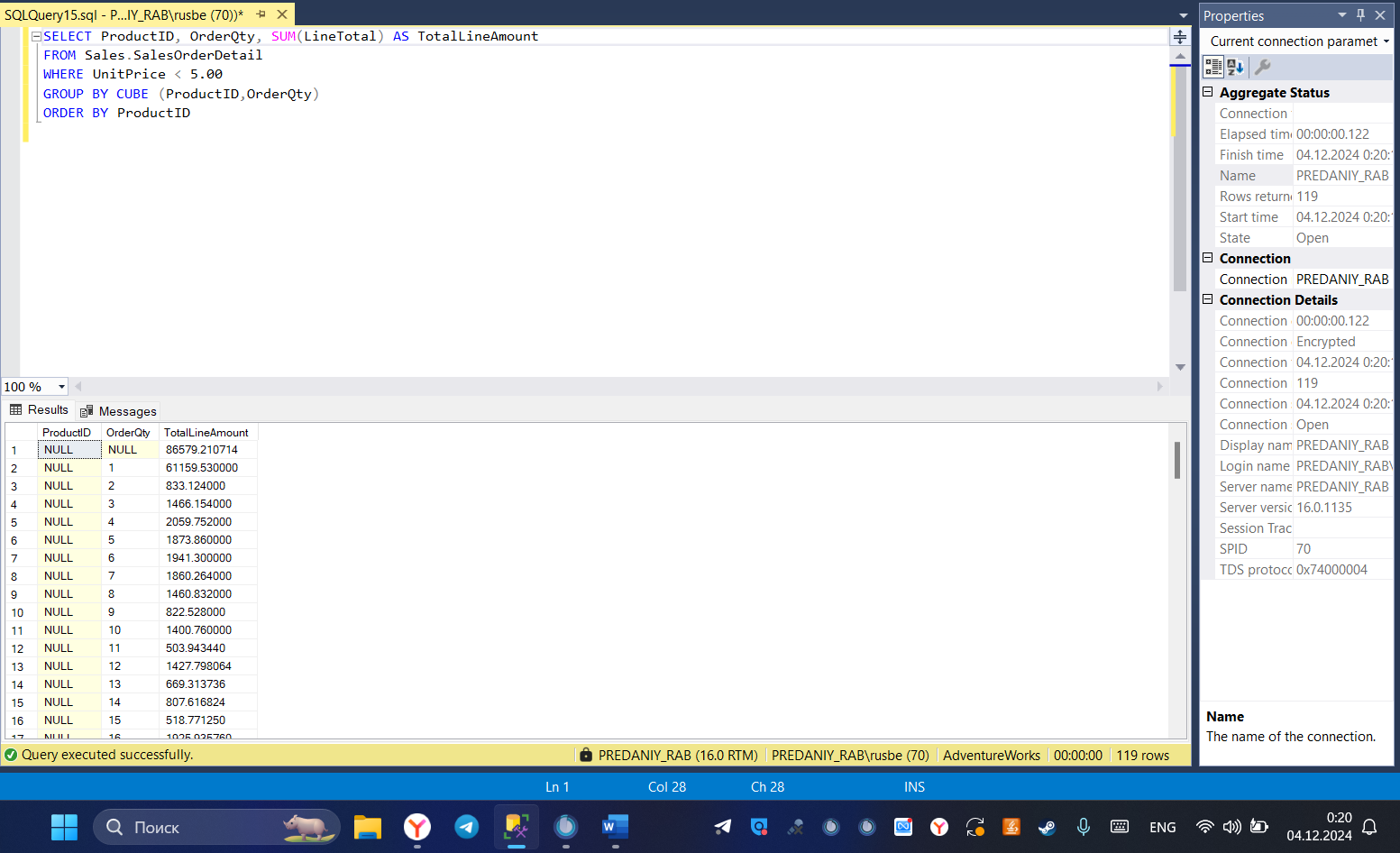


Рисунок 13 –Выполнение запроса с использованием WHERE, GROUP BY CUBE, ORDER BY

Запрос 4: Модифицировать прошлый запрос, добавив оператор CUBE и поле OrderQty

SELECT ProductID, OrderQty SUM(LineTotal) AS TotalLineAmount

FROM Sales.SalesOrderDetail

WHERE UnitPrice < 5.00

GROUP BY OrderQty, CUBE (ProductID)

ORDER BY ProductID DESC;

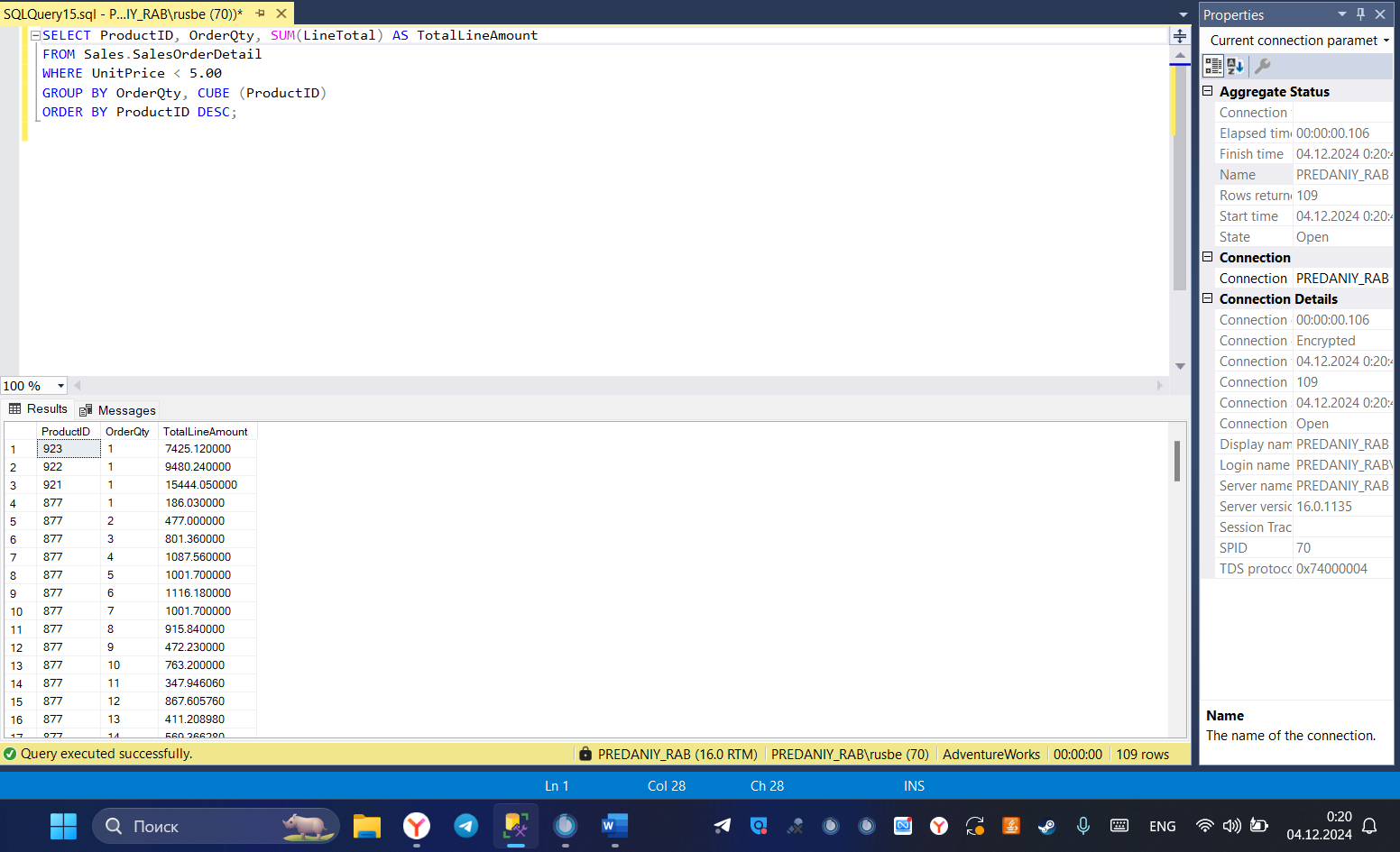


Рисунок 14 –Выполнение запроса с использованием CUBE

**Упражнение 4 – использование предложений COMPUTE и COMPUTE BY в команде SELECT для создания отчетов**

Запрос 1: Написать запрос к таблице Sales.SalesOrderDetail,указать 5 полей, отсортировать по двум заданным полям

Результат выполнения запроса представлен на рисунке

SELECT SalesPersonID, CustomerID, OrderDate, SubTotal, TotalDue

FROM Sales.SalesOrderHeader

ORDER BY SalesPersonID, OrderDate;

SELECT

SUM(SubTotal) AS TotalSubTotal

SUM(TotalDue) AS TotalTotalDue

FROM Sales.SalesOrderHeader;

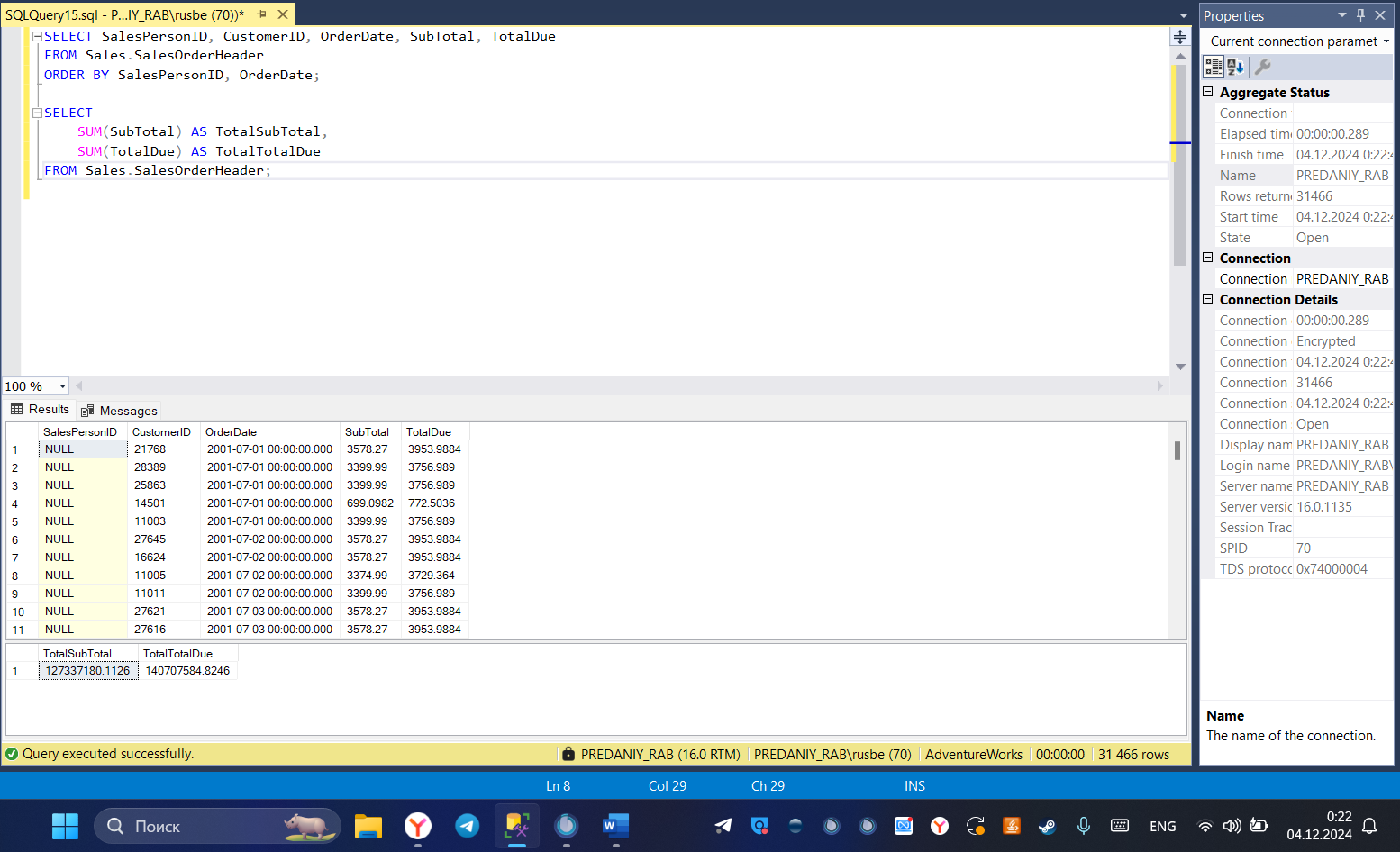


Рисунок 15 –Выполнение запроса с использованием Sum вместо Compute

# Вывод

В ходе лабораторной работы были изучены TOP, WITH TIES, GROUP BY, HAVING, CUBE, ROLLUP используемые для анализа и агрегирования данных из базы данных. В процессе выполнения заданий были проведены выборки из таблиц базы данных Adventure Works с применением различных ключевых слов, таких как WHERE, AS, NULL, FROM, ORDER BY. Это позволило закрепить методы фильтрации и упорядочивания данных. Эти навыки и знания обеспечат более эффективное извлечение и анализ данных из базы данных в будущих проектах.