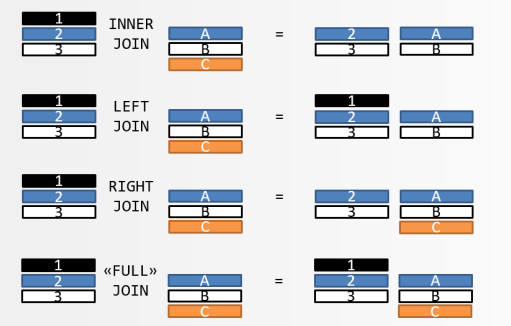
**Домашняя работа 3**

Продвинутый уровень SQL

1. Повторите изученный материал [здесь](https://www.w3schools.com/sql/sql_join.asp) : разделы JOINS и его виды (6 статей), CASE, UNION.

Выполните после каждого раздела задания.

Посмотрите, чем отличаются между собой различные типы JOINS:



Упражнение 1:

Вставьте недостающие части в предложение, чтобы соединить две таблицы и , использование поля в обеих таблицах в качестве связи между двумя таблицами.JOINOrdersCustomersCustomerID

SELECT \*

FROM Orders

LEFT JOIN Customers ON Orders.CustomerID=Customers.CustomerID;

Упражнение 2:

Выберите правильное предложение, чтобы выбрать все записи из двух таблиц где есть совпадение в обеих таблицах.JOIN

SELECT \*

FROM Orders

INNER JOIN Customers ON Orders.CustomerID=Customers.CustomerID;

Упражнение 3:

Выберите правильное предложение, чтобы выбрать все записи из таблицы плюс все матчи в таблице.JOINCustomersOrders

SELECT \*

FROM Orders

RIGHT JOIN Customers ON Orders.CustomerID=Customers.CustomerID;

Упражнение 4:

Выберите правильное предложение, чтобы выбрать все записи из двух таблиц где есть совпадение в обеих таблицах.JOIN

SELECT \*

FROM Orders

INNER JOIN Customers ON Orders.CustomerID=Customers.CustomerID;

1. Решите на базе данных AdventureWorks2017 следующие задачи (для поиска ключей можно использовать [схему БД](https://i.stack.imgur.com/CAGy7.gif) ).
2. Вывести список цен в виде текстового комментария, основанного на диапазоне цен для продукта:
   1. StandardCost равен 0 или не определен – ‘Not for sale’
   2. StandardCost больше 0, но меньше 100 – ‘<$100’
   3. StandardCost больше или равно 100, но меньше 500 - ‘ <$500'
   4. Иначе - ‘ >= $500'

Вывести имя продукта и новое поле PriceRange.

SELECT p.name,

CASE

WHEN p.standardcost = 0 OR p.standardcost is null THEN 'Not for sale'

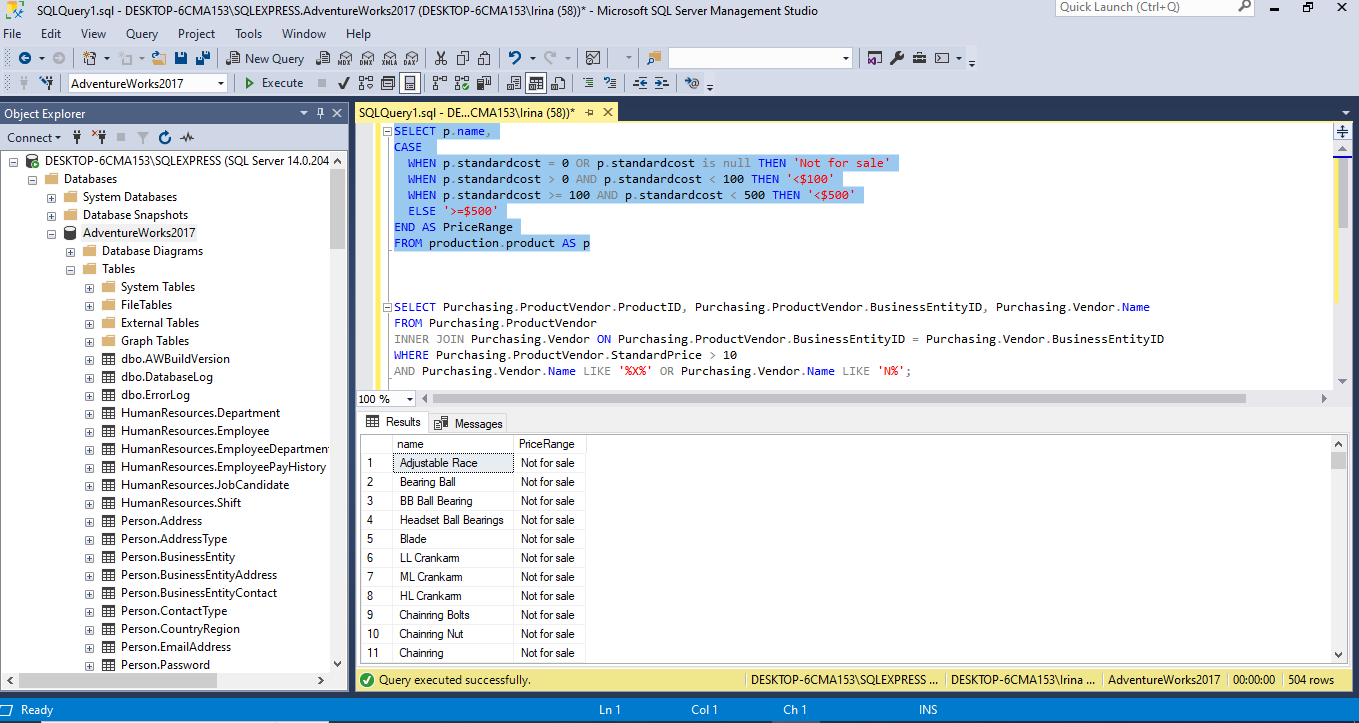
WHEN p.standardcost > 0 AND p.standardcost < 100 THEN '<$100'

WHEN p.standardcost >= 100 AND p.standardcost < 500 THEN '<$500'

ELSE '>=$500'

END AS PriceRange

FROM production.product AS p



1. Найти ProductID, BusinessEntityID и имя поставщика продукции из Purchasing.ProductVendor и Purchasing.Vendor, где StandardPrice больше $10. Также в имени вендора должна присутствовать (вне зависимости от положения) буква X или имя должно начинаться на букву N.

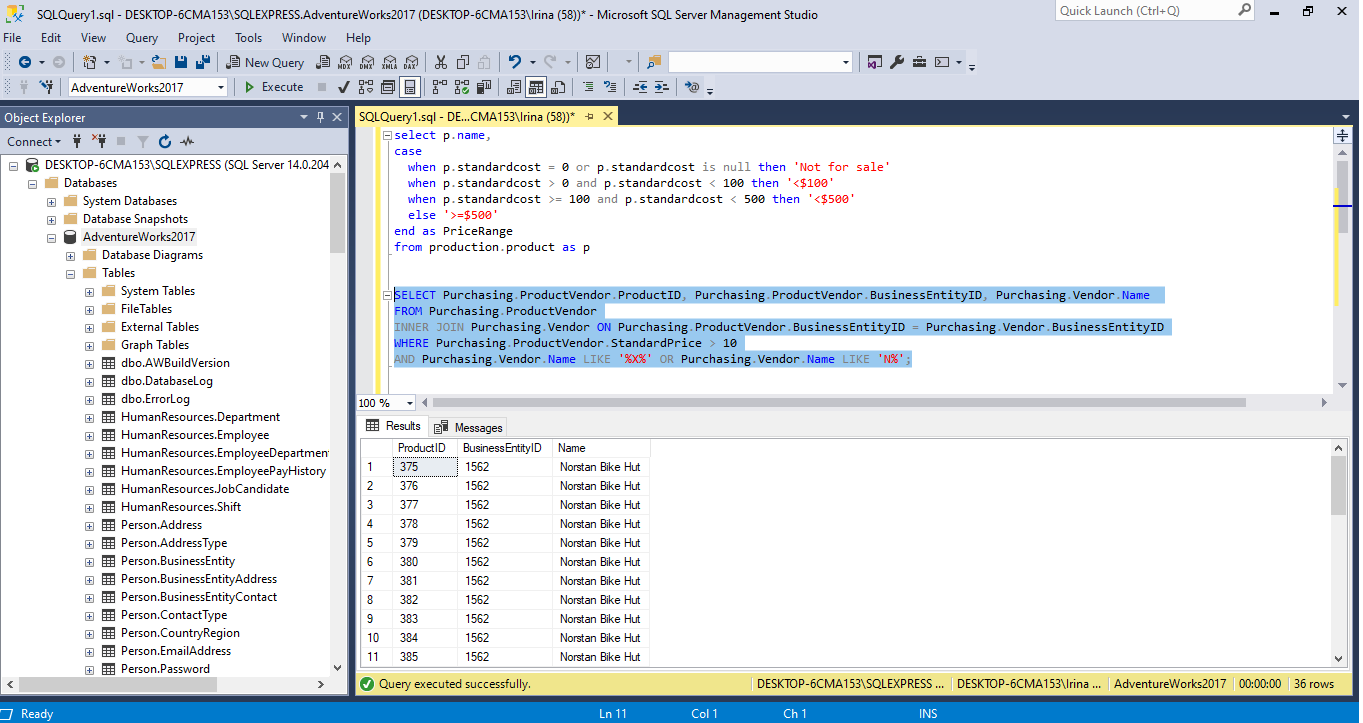
SELECT Purchasing.ProductVendor.ProductID, Purchasing.ProductVendor.BusinessEntityID, Purchasing.Vendor.Name

FROM Purchasing.ProductVendor

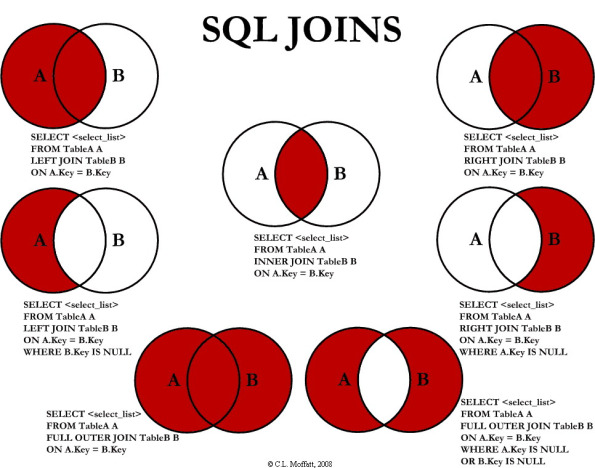
INNER JOIN Purchasing.Vendor ON Purchasing.ProductVendor.BusinessEntityID = Purchasing.Vendor.BusinessEntityID

WHERE Purchasing.ProductVendor.StandardPrice > 10

AND Purchasing.Vendor.Name LIKE '%X%' OR Purchasing.Vendor.Name LIKE 'N%';



1. Найти имена всех вендоров, продукция которых не продавалась за всё время. Необходимо использовать следующую схему соединения таблиц Purchasing.ProductVendor и Purchasing.Vendor:

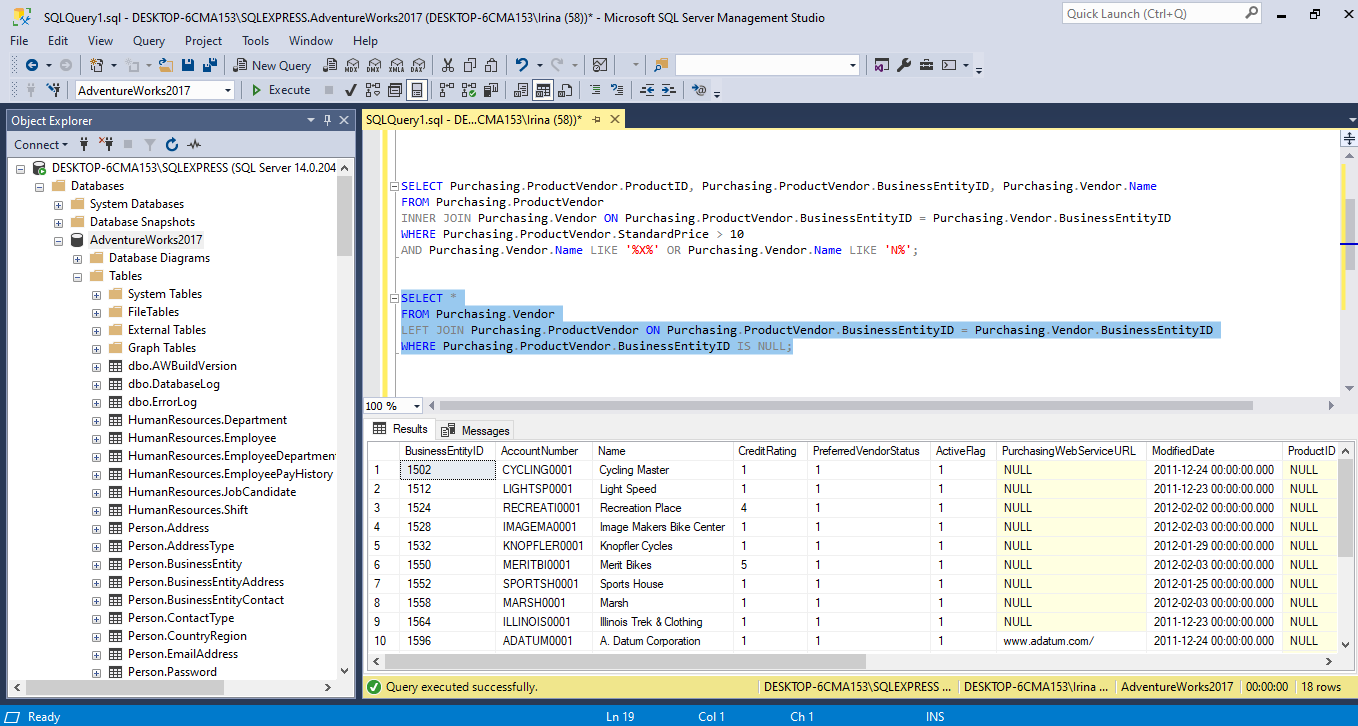


SELECT \*

FROM Purchasing.Vendor

LEFT JOIN Purchasing.ProductVendor ON Purchasing.ProductVendor.BusinessEntityID = Purchasing.Vendor.BusinessEntityID

WHERE Purchasing.ProductVendor.BusinessEntityID IS NULL;



1. Решите на базе данных AdventureWorks2017 следующие задачи.
2. Выведите названия продуктов и их цену, модель которых начинается на ‘LL’. Таблицы [Production].[ProductModel] и [Production].[Product].

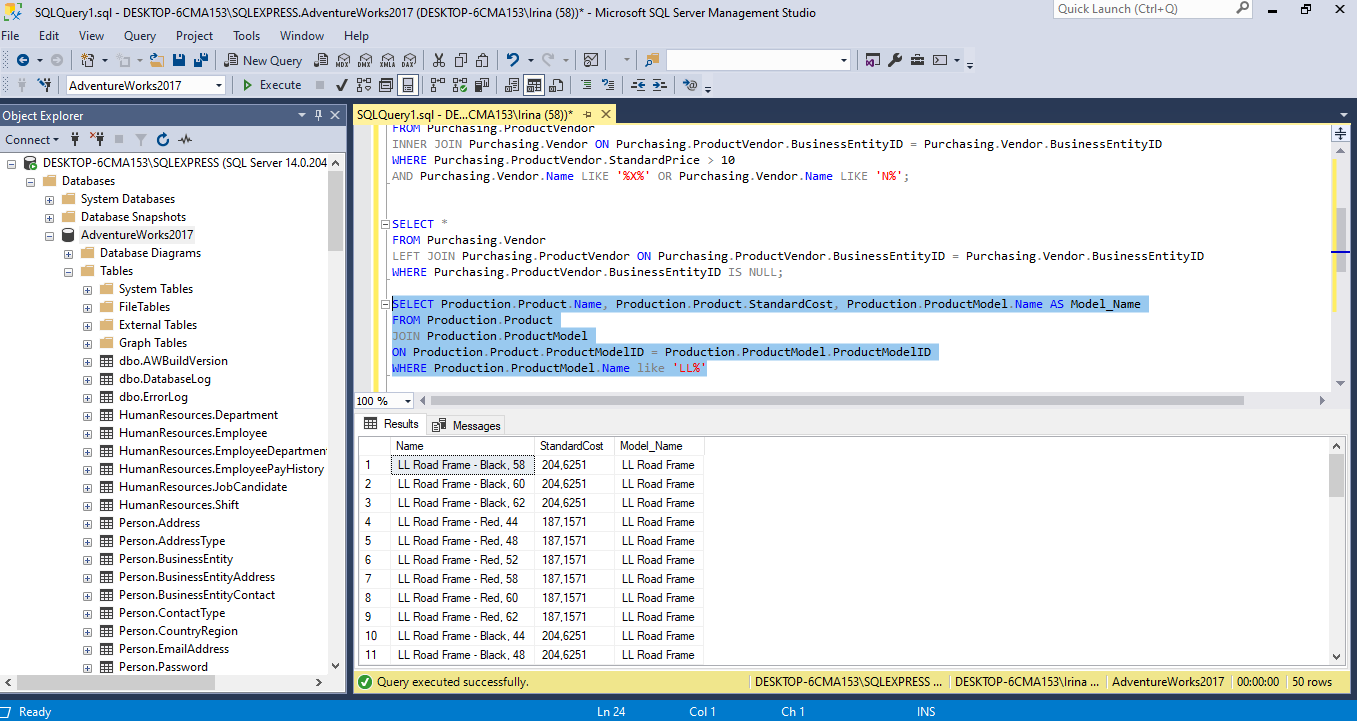
SELECT Production.Product.Name, Production.Product.StandardCost, Production.ProductModel.Name AS Model\_Name

FROM Production.Product

JOIN Production.ProductModel

ON Production.Product.ProductModelID = Production.ProductModel.ProductModelID

WHERE Production.ProductModel.Name like 'LL%'



1. Выведите имена всех вендоров [Purchasing].[Vendor] и имена магазинов [Sales].[Store] одним списком в отсортированном порядке по алфавиту и без дубликатов.

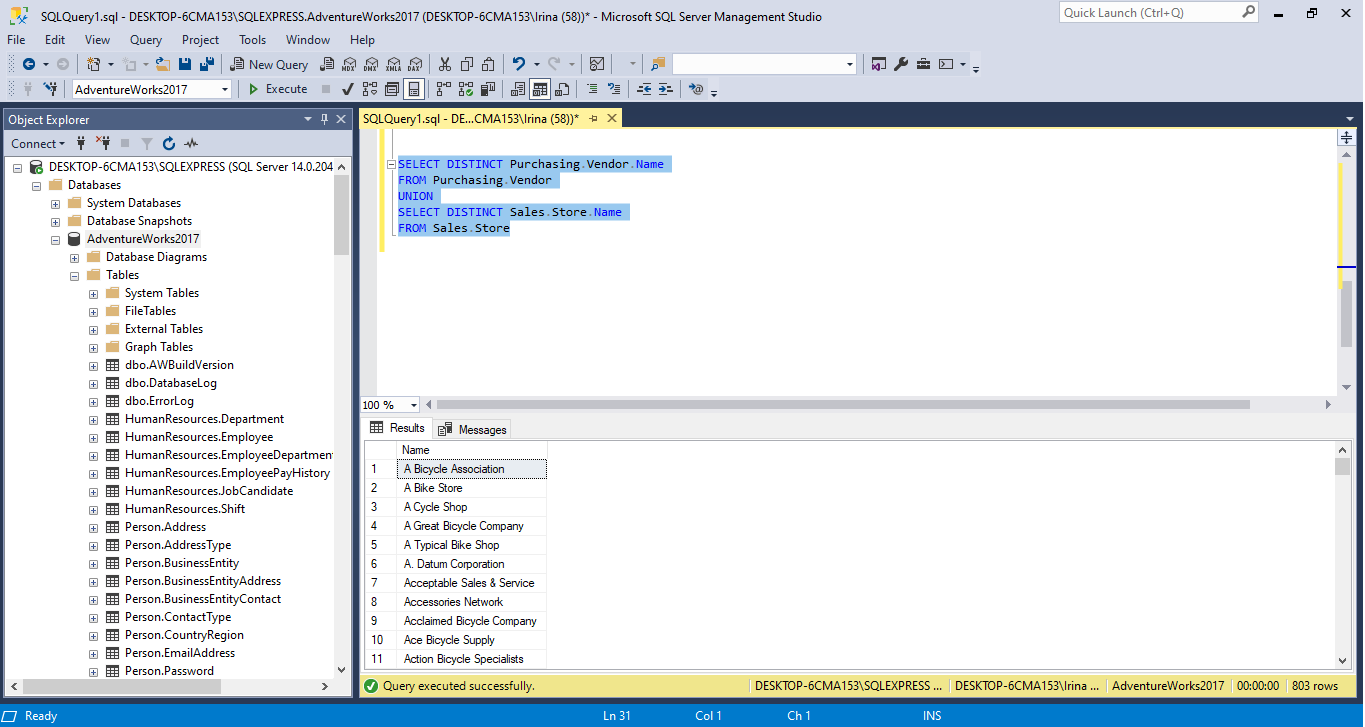
SELECT DISTINCT Purchasing.Vendor.Name

FROM Purchasing.Vendor

UNION

SELECT DISTINCT Sales.Store.Name

FROM Sales.Store



1. Найдите имена продуктов, на которых действовало больше, чем 1 специальное предложение. Таблицы [Sales].[SpecialOffer],

[Sales].[SpecialOfferProduct], [Production].[Product].

SELECT Production.Product.Name, COUNT(Sales.SpecialOfferProduct.SpecialOfferID) AS Cnt\_Special\_Offer

FROM Production.Product

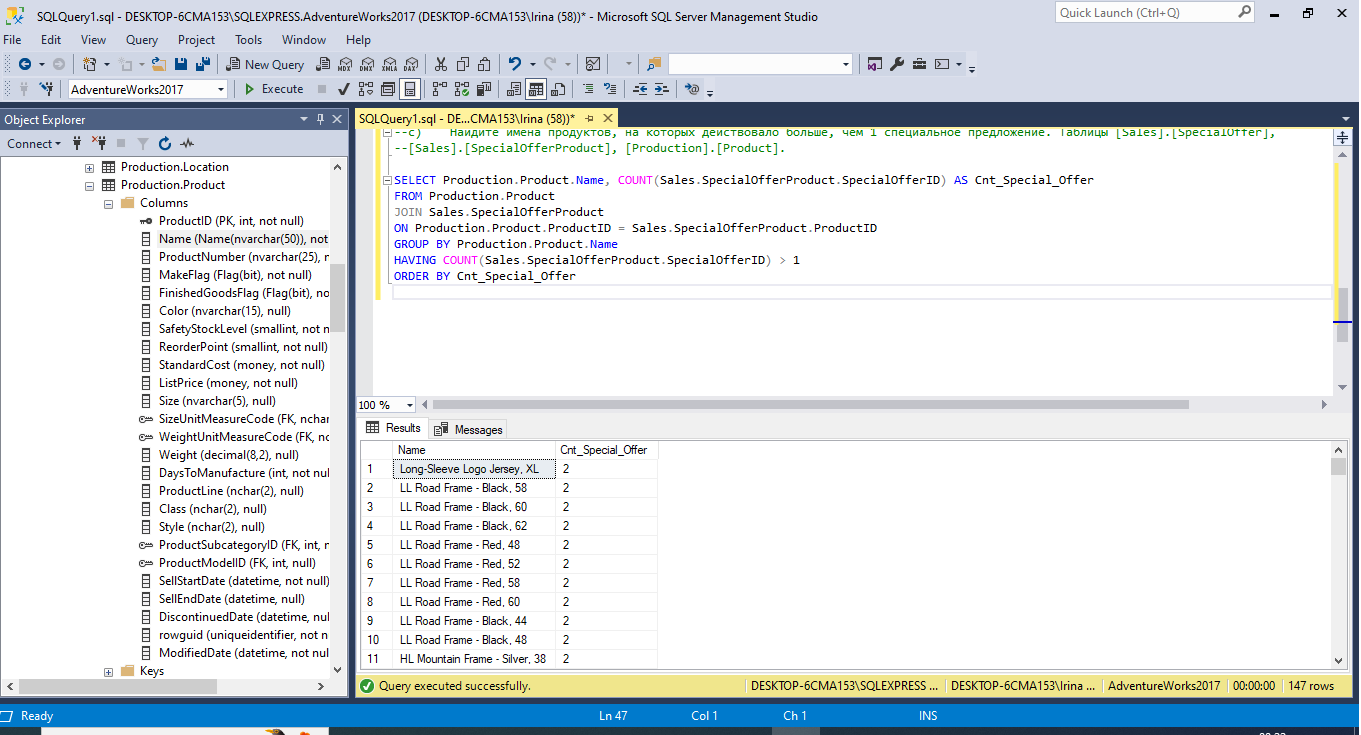
JOIN Sales.SpecialOfferProduct

ON Production.Product.ProductID = Sales.SpecialOfferProduct.ProductID

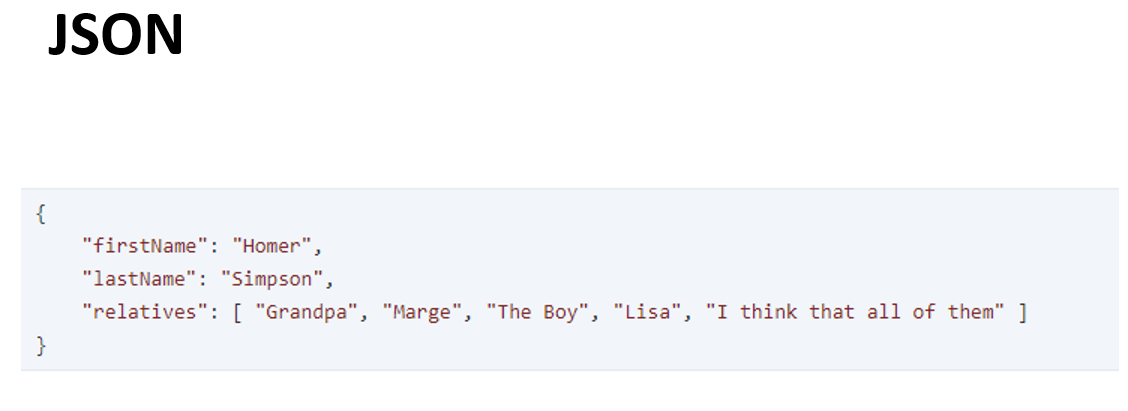
GROUP BY Production.Product.Name

HAVING COUNT(Sales.SpecialOfferProduct.SpecialOfferID) > 1

ORDER BY Cnt\_Special\_Offer



1. Изучите [Функции даты и времени](https://docs.microsoft.com/ru-ru/sql/t-sql/functions/date-and-time-data-types-and-functions-transact-sql?view=sql-server-ver15%20%20) . Необходимо знать синтаксис и определение следующих функций – GETDATE, DATENAME, DATEPART, DAY, MONTH, YEAR, DATEDIFF, DATEADD, EOMONTH.
2. Изучите информацию по типам данных [XML и JSON](https://appmaster.io/ru/blog/json-protiv-xml):



Описание: Изображение выглядит как текст

Автоматически созданное описание

Описание: Изображение выглядит как стол

Автоматически созданное описание

Поисследуйте, как XML данные хранятся в реляционной базе данных (AdventureWorks2019.Production.ProductModel таблица).