**лАБОРАТОРНАЯ РАБОТА №1**

**Работа с объектами БД СУБД MS SQL Server**

**с помощью системы администрирования**

**SQL Management Studio 2020**

**Цель работы:** изучение основных приемов работы с объектами базы данных СУБД SQL Server 2020 с помощью инструмента SQL Management Studio 2020, изучение операторов языка манипулирования данными SQL-DML.

**Задачи:**

**1)** Заполнить данными созданную таблицу *Должности* данными колонки *Должность* таблицы *Сотрудники*;

**2)** Заполнить данными созданную таблицу *Страны* данными соответствующих колонок таблиц *Сотрудники*, *Клиенты*, *Поставщики*, *Заказы*;

**3)** Заполнить данными таблицу *Области* данными соответствующих колонок таблиц *Сотрудники*, *Клиенты*, *Поставщики*, *Заказы*;

**4)** Заполнить данными таблицу *Регионы* данными соответствующих колонок таблиц *Сотрудники*, *Клиенты*, *Поставщики*, *Заказы*.

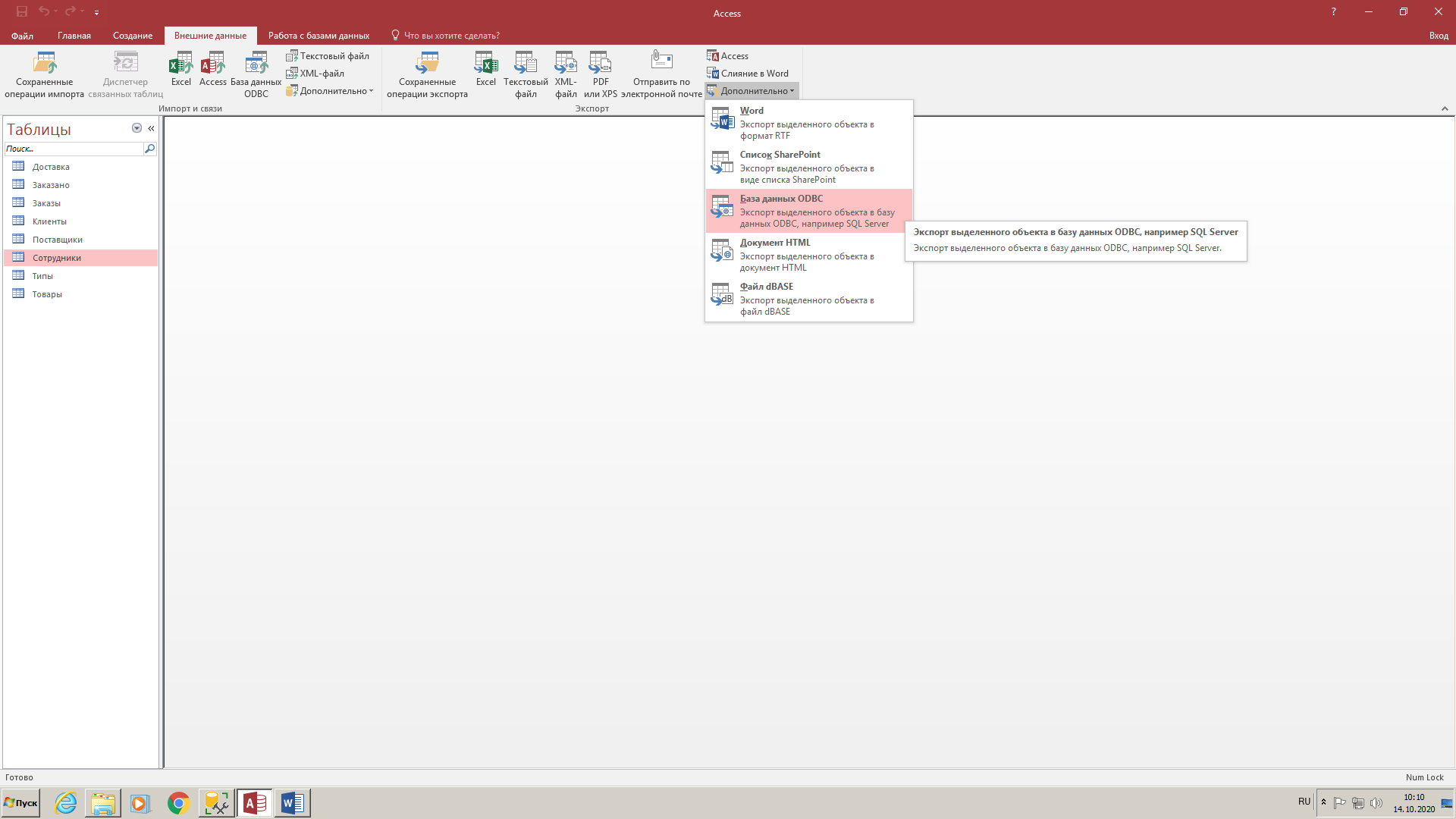
**Ход работы**

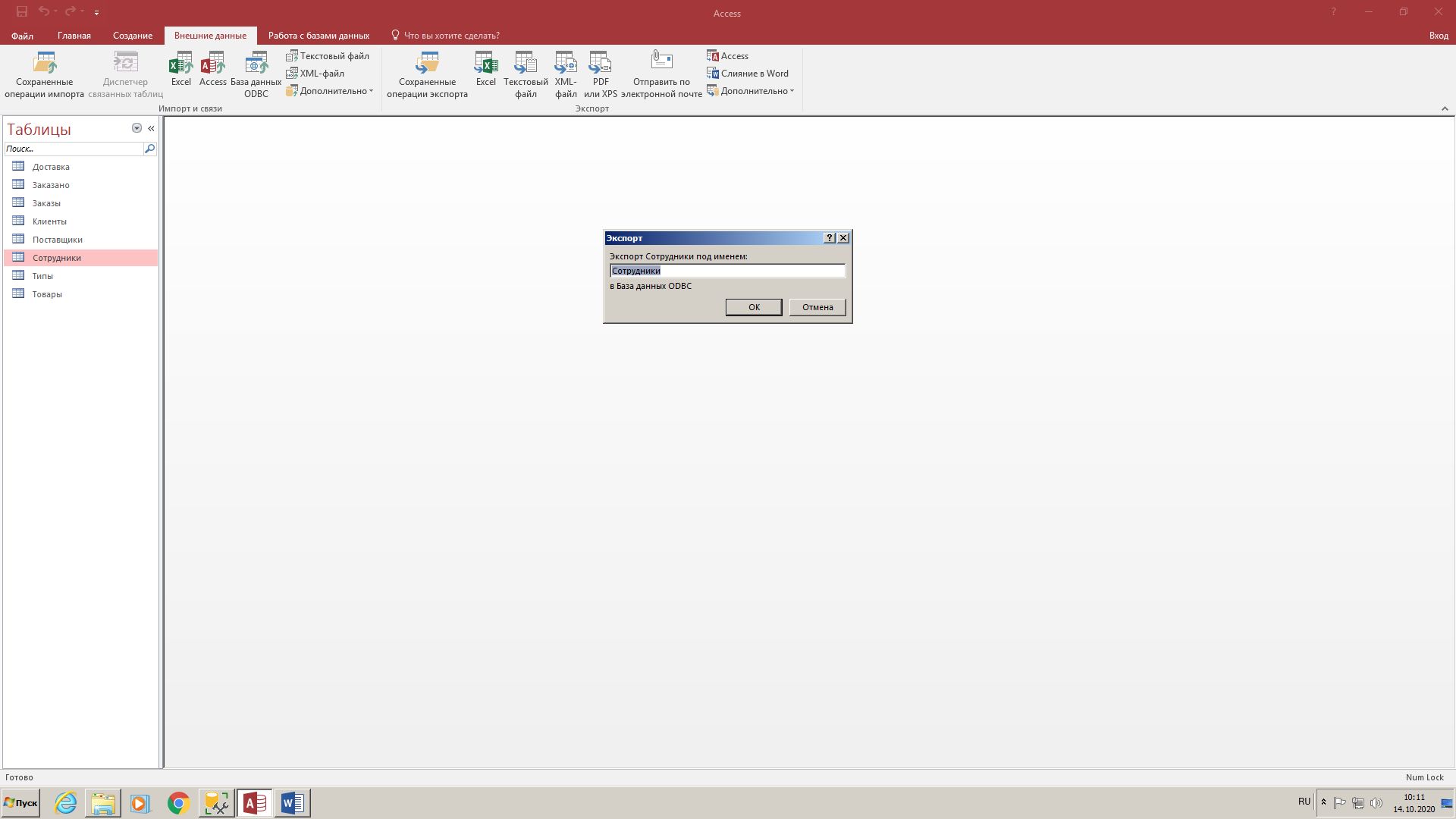
1. Создадим БД на платформе SQL Management Studio с именем «№группы\_фамилия».

Изучим схему данных назначение и функциональные возможности БД «Борей» СУБД Access.

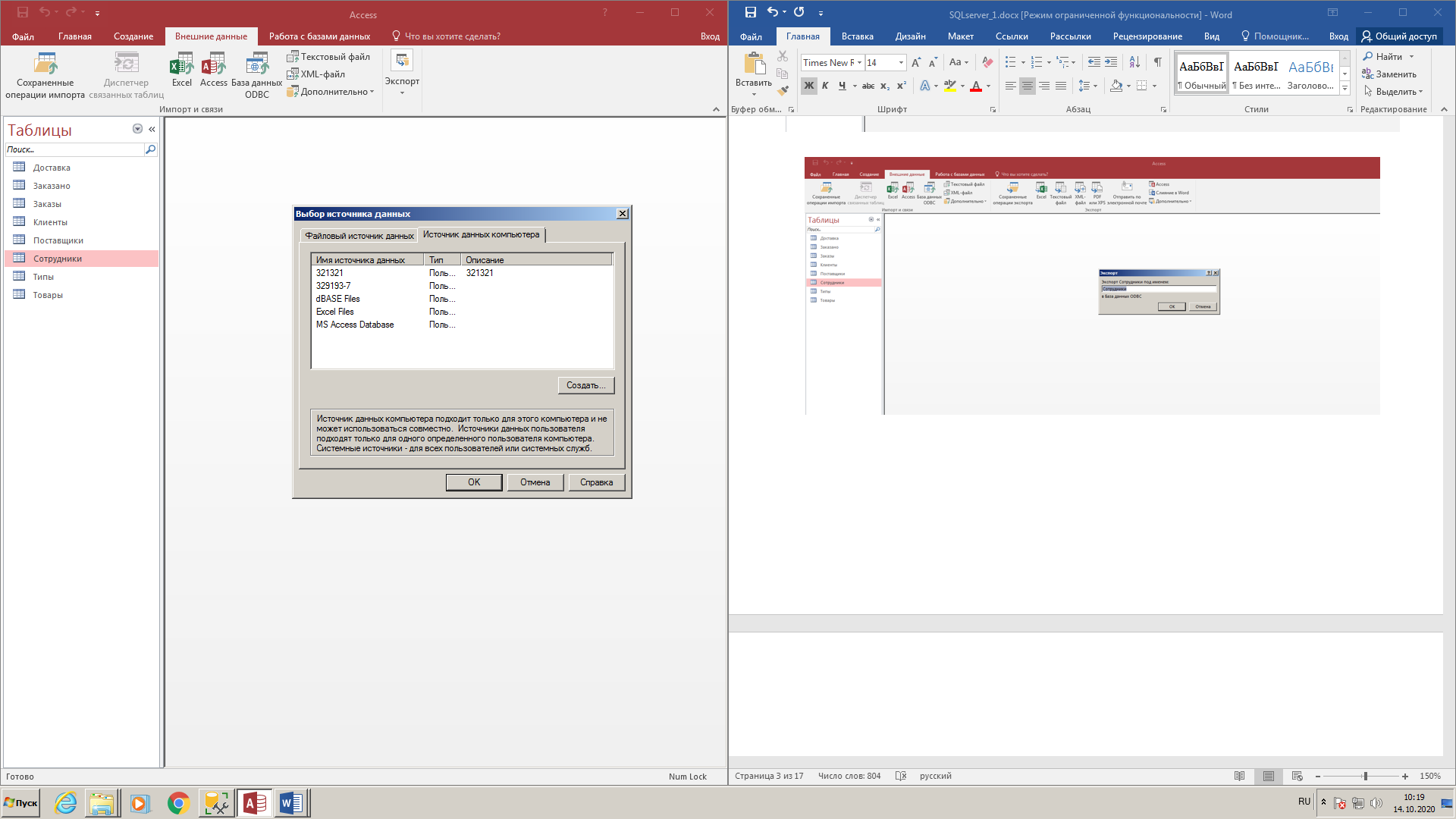


**2)** Перенести БД «Борей» СУБД Aсcess в БД СУБД SQL Server 2020 с помощью вкладки Внешние данные-> Дополнительно-> «База данных ODBC».

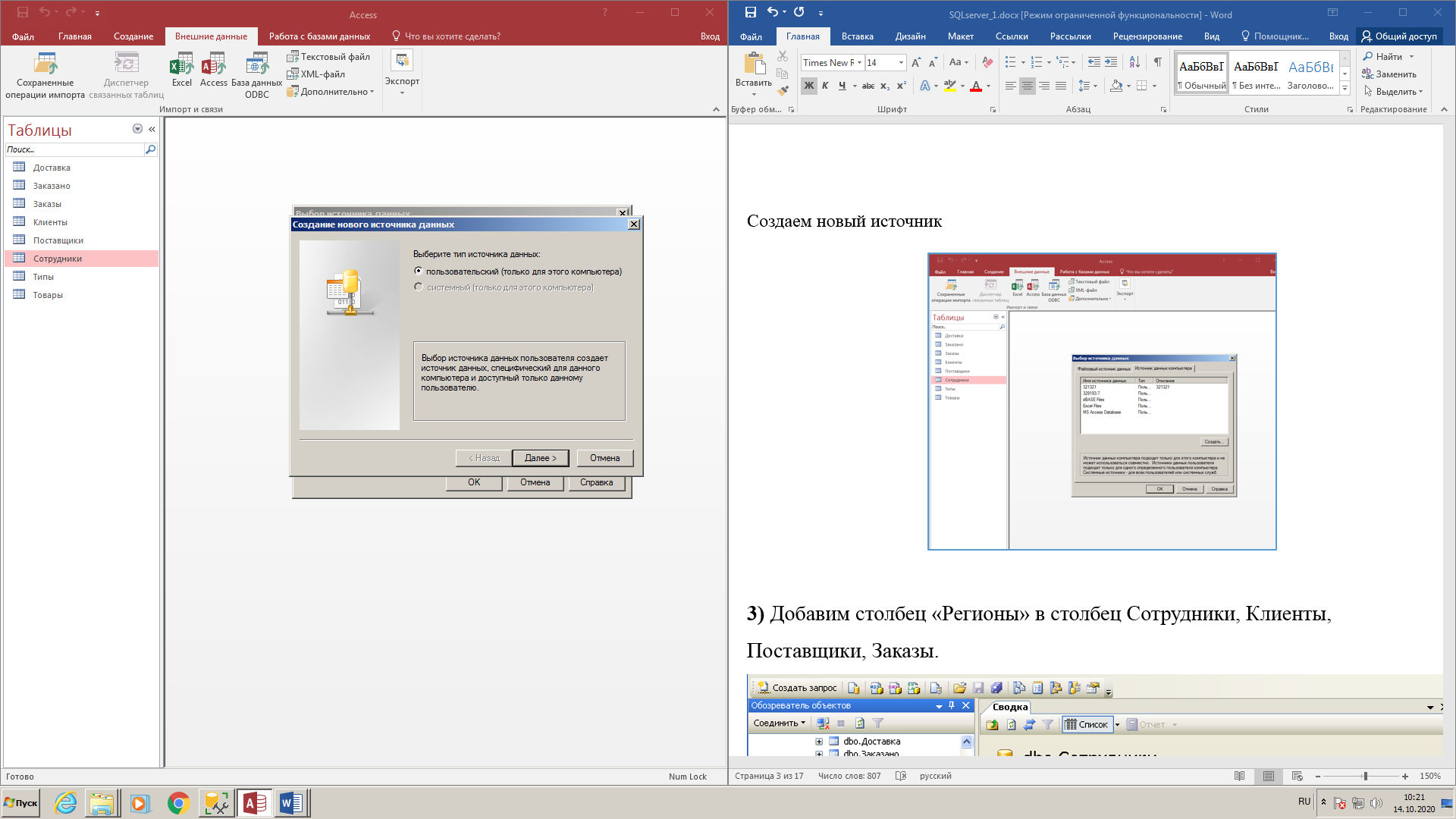




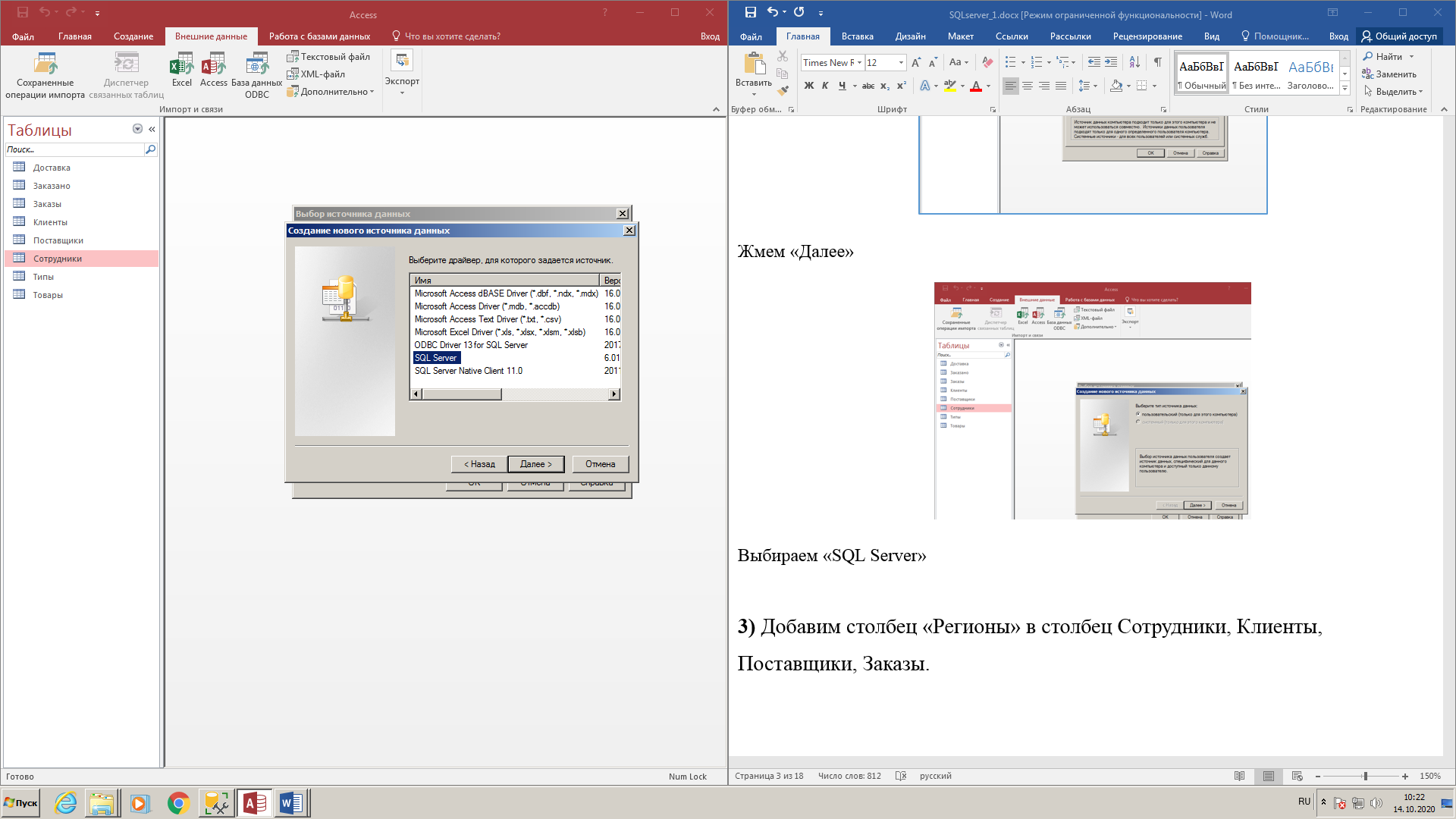
* + Создаем новый источник.



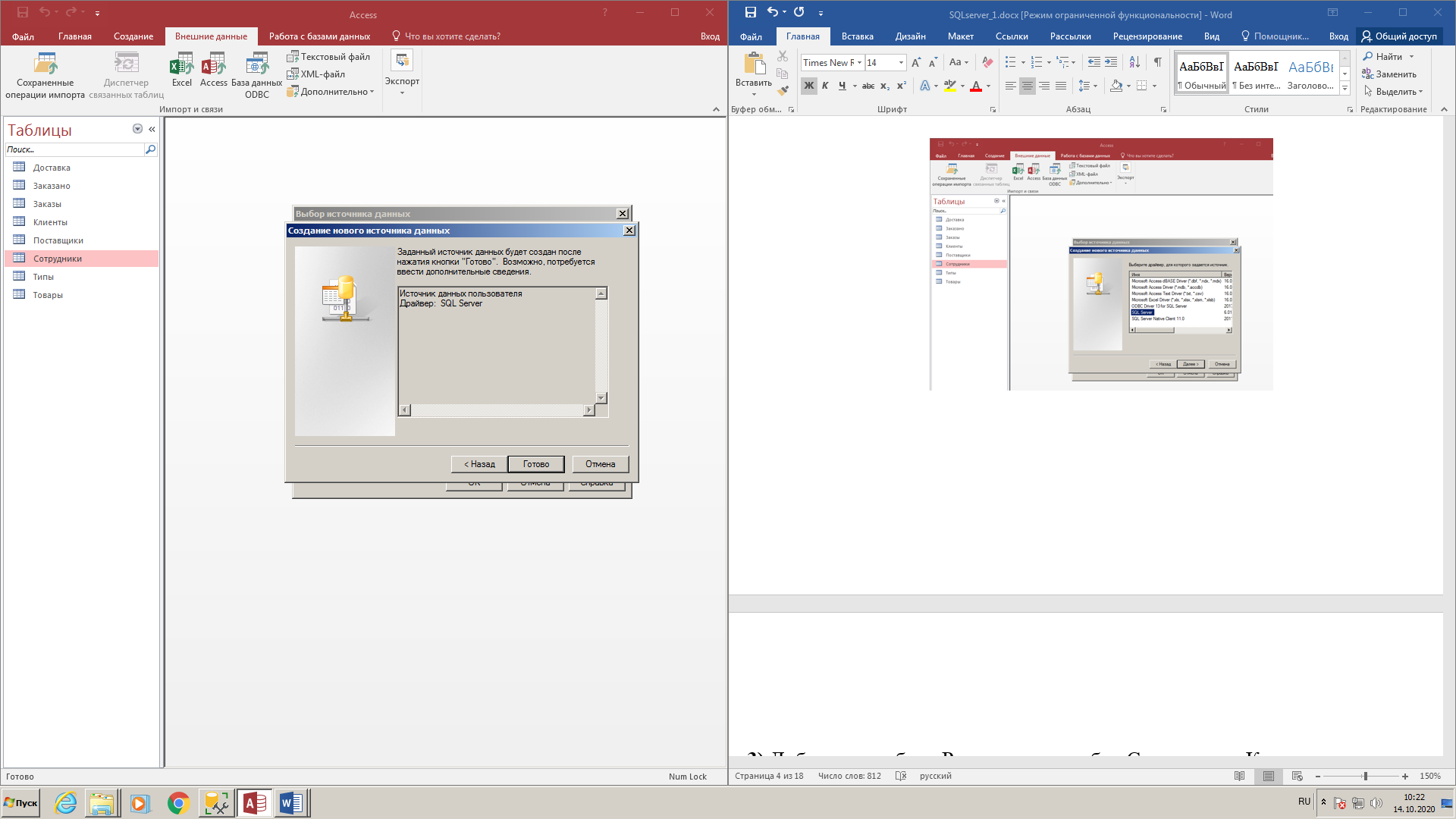
* + Жмем «Далее».



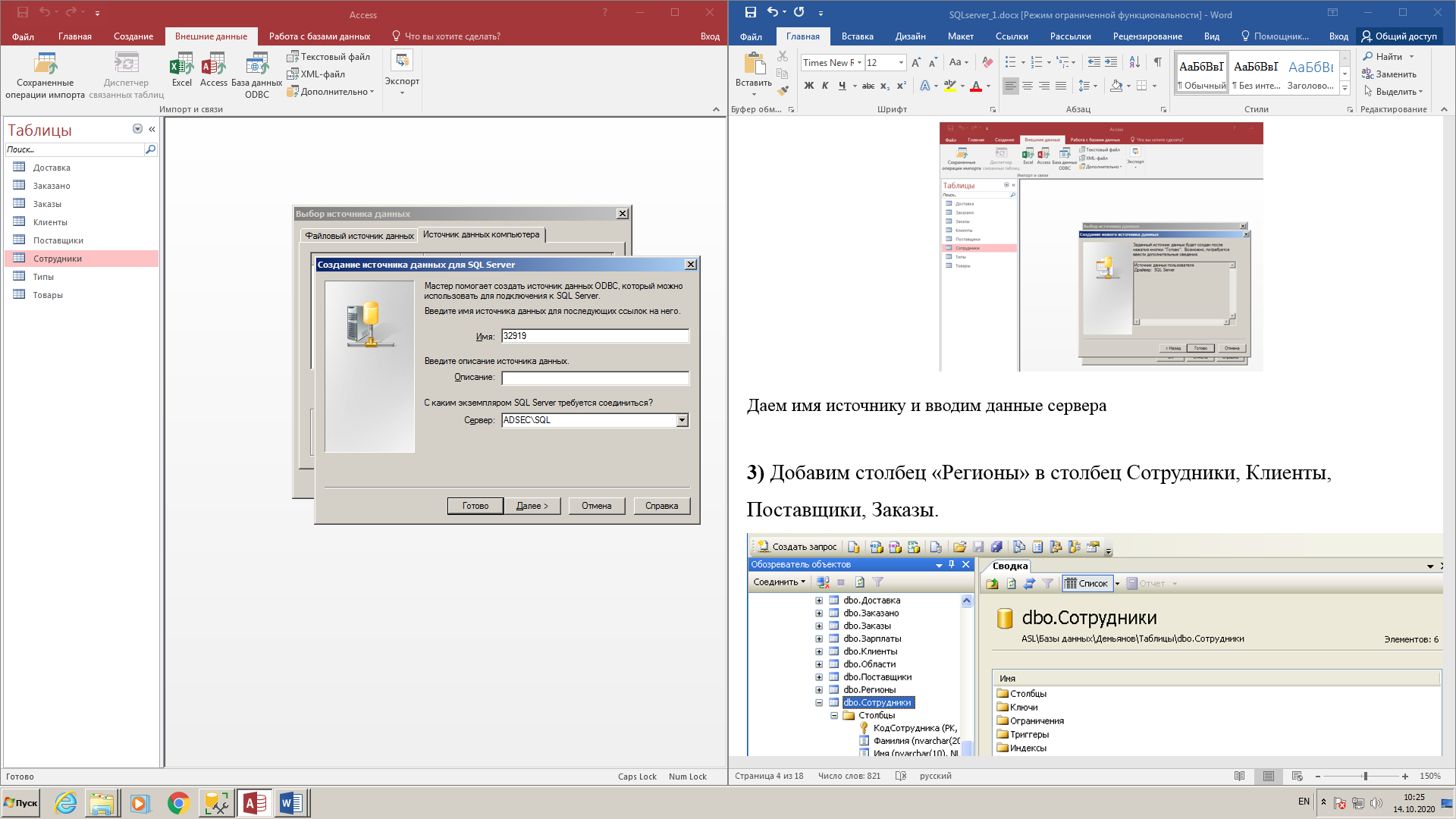
* + Выбираем «SQL Server».



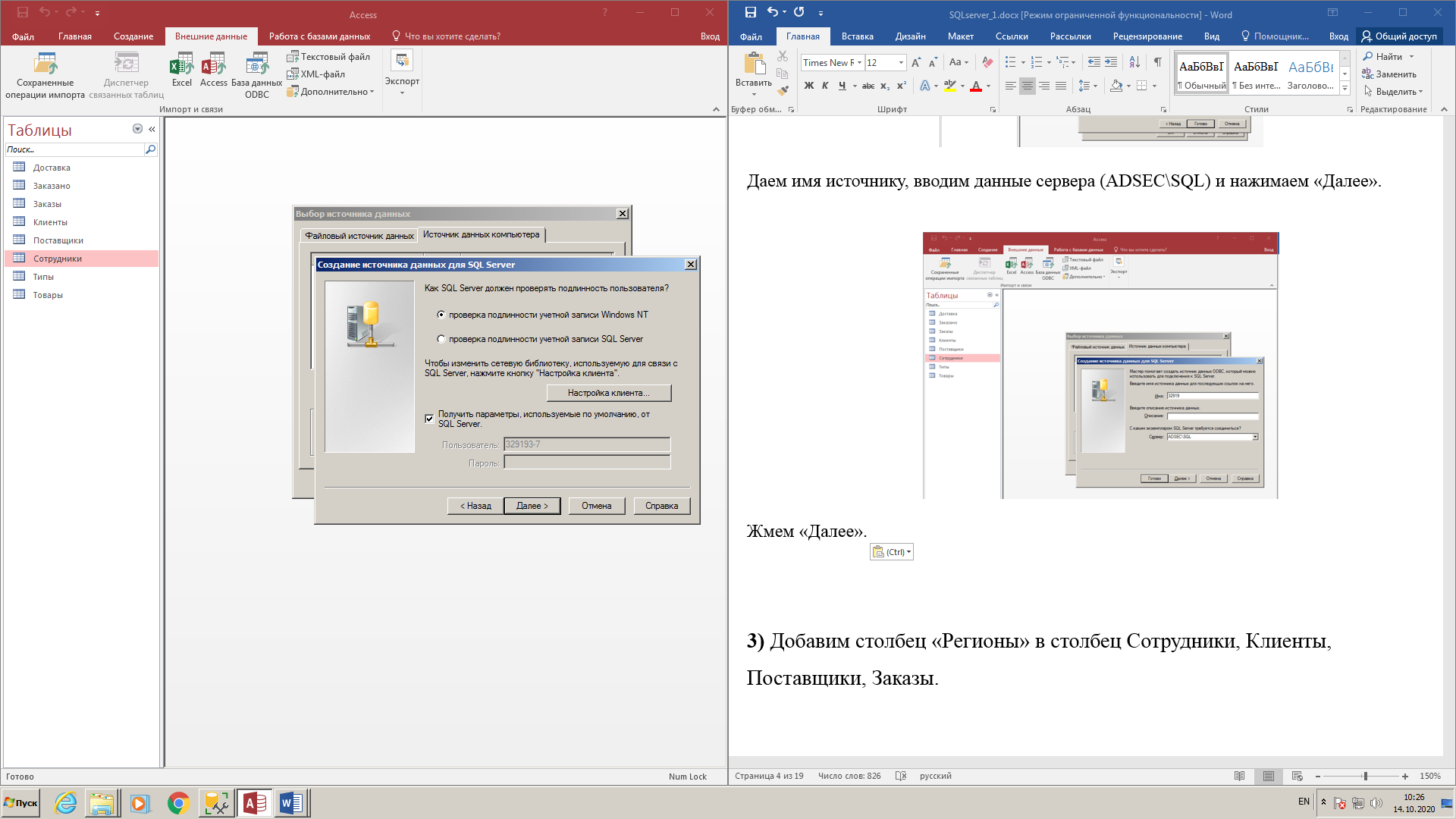
* + Жмем «Готово».



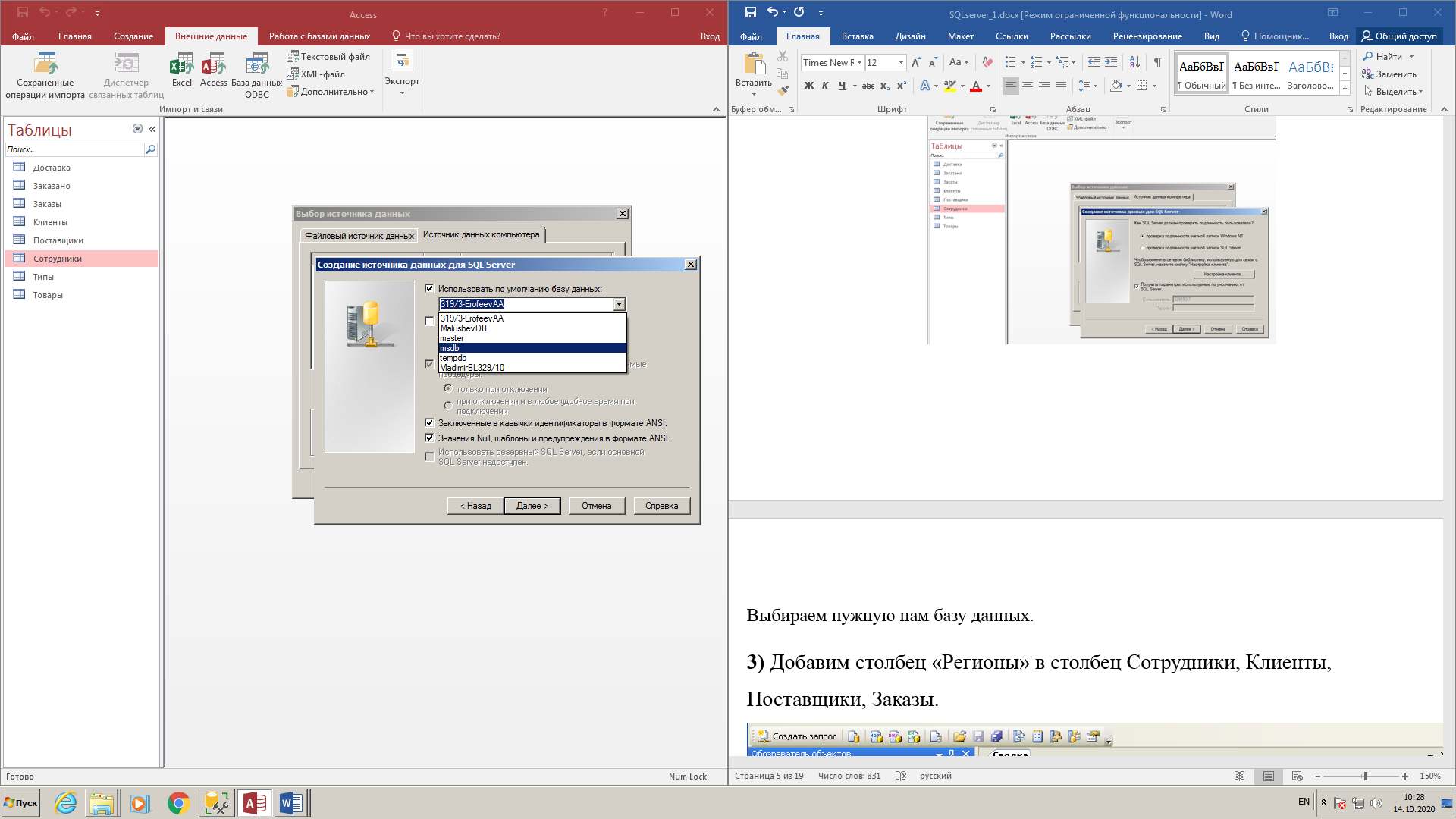
* + Даем имя источнику, вводим данные сервера (ADSEC\SQLSRV) и нажимаем «Далее».



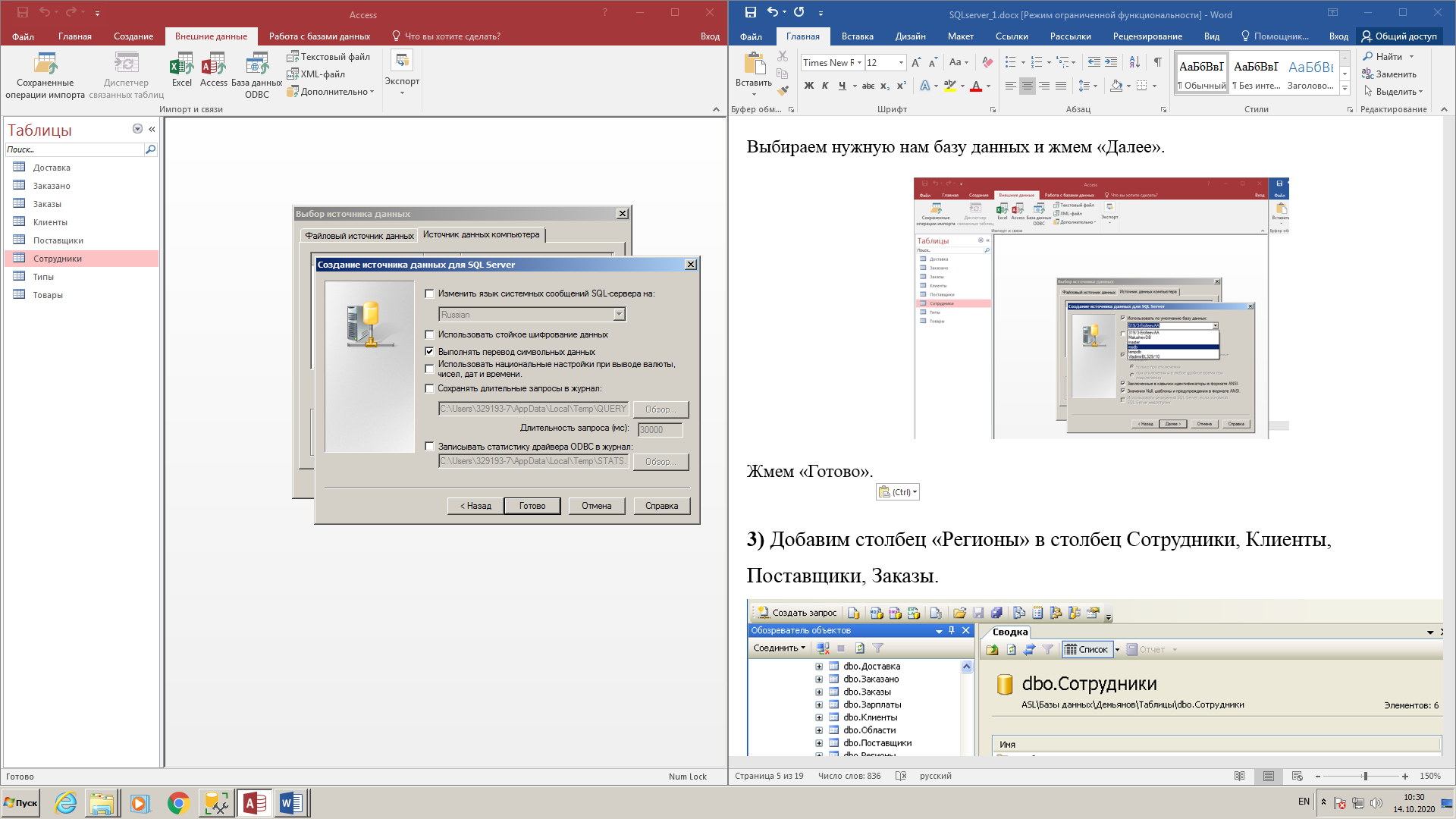
* + Жмем «Далее».



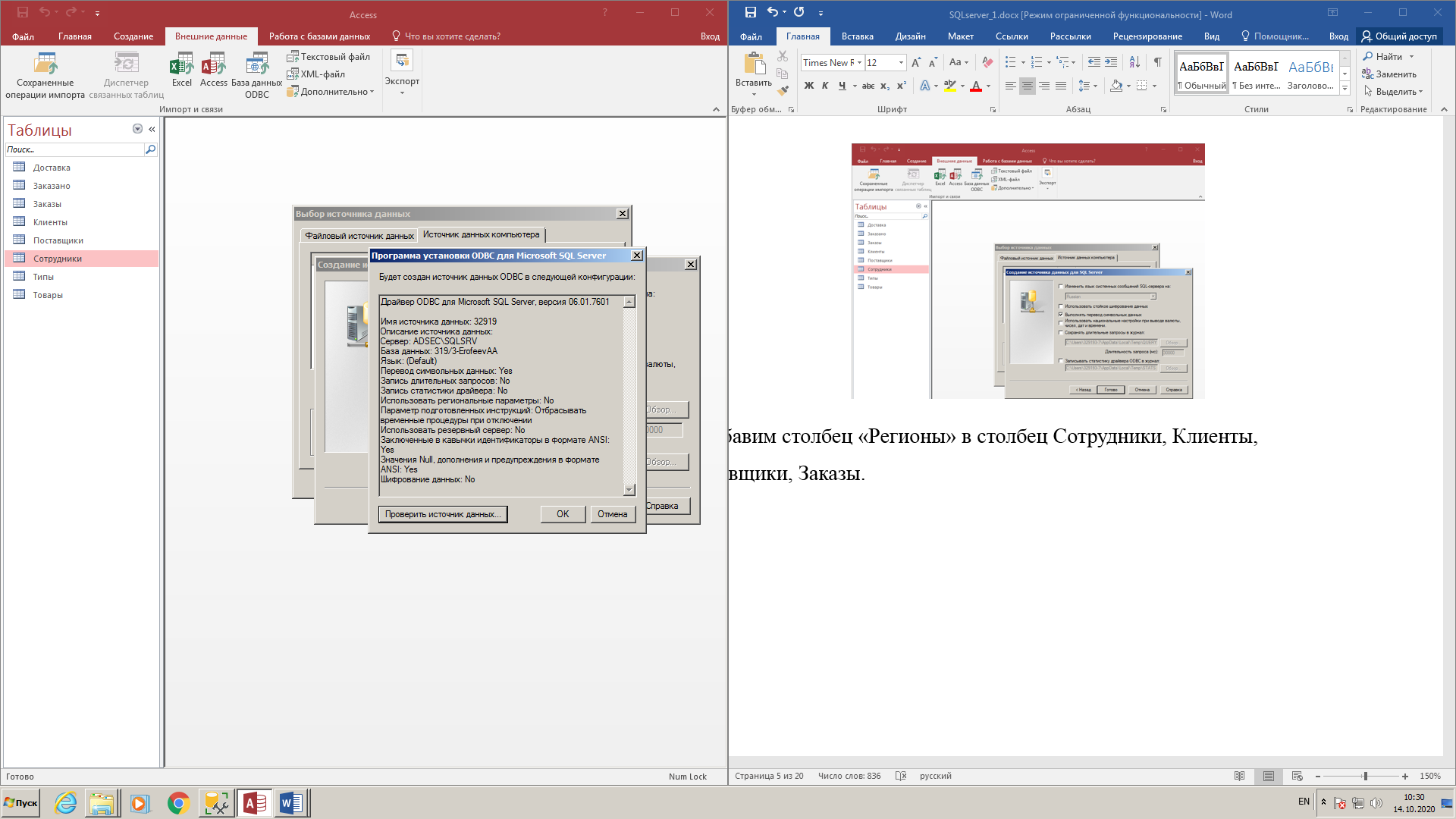
* + Выбираем нужную нам базу данных и жмем «Далее».



* + Жмем «Готово».

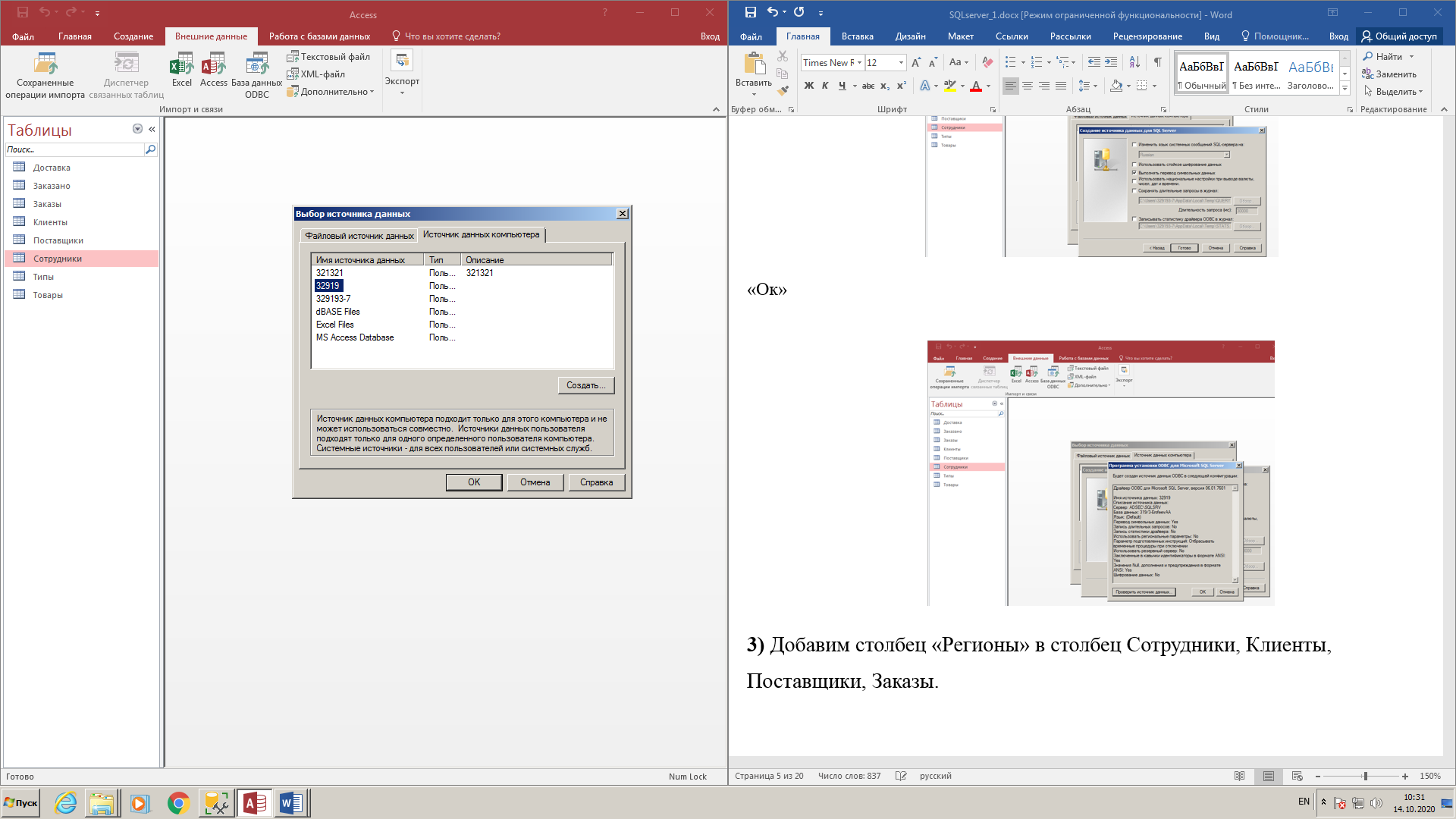


* + «Ок».



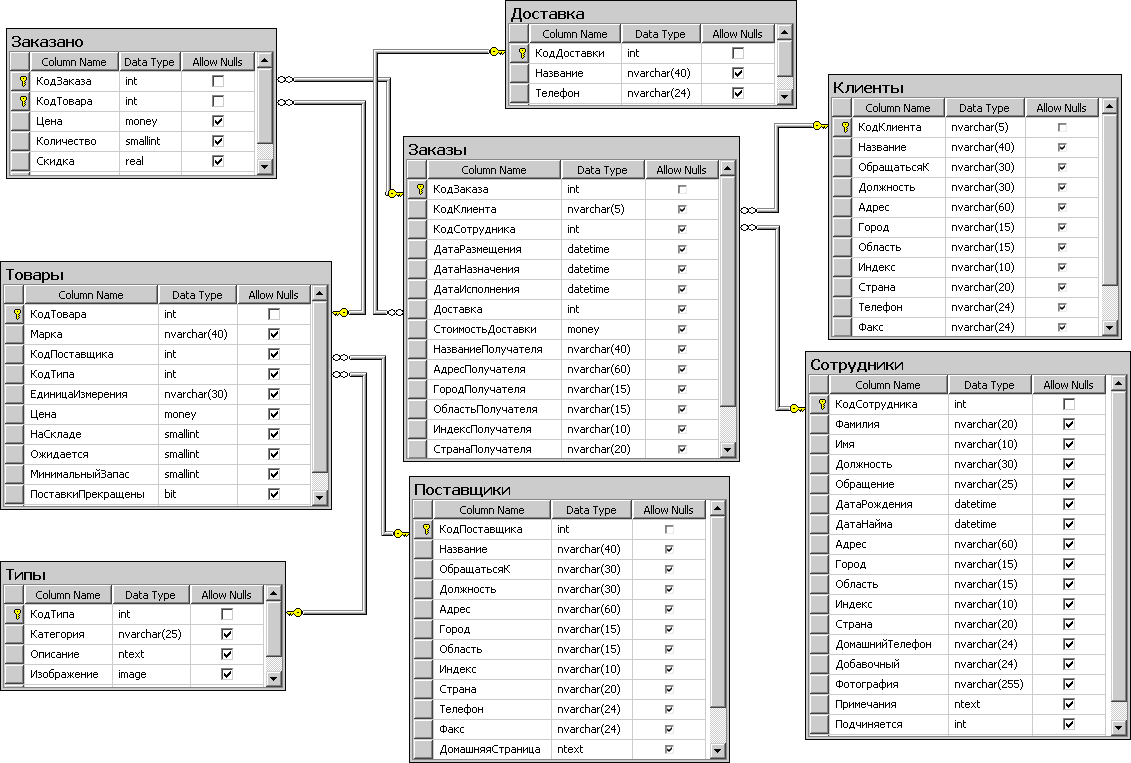
* + Наш источник данных готов.

Выбираем нужный и нажимаем «Ок»

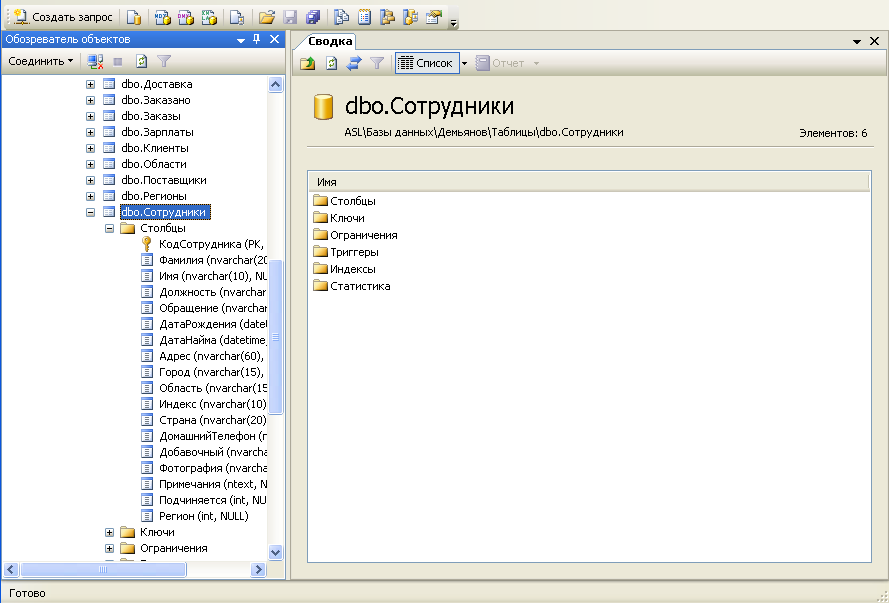


Уже без создания источника повторить с каждой таблицей.

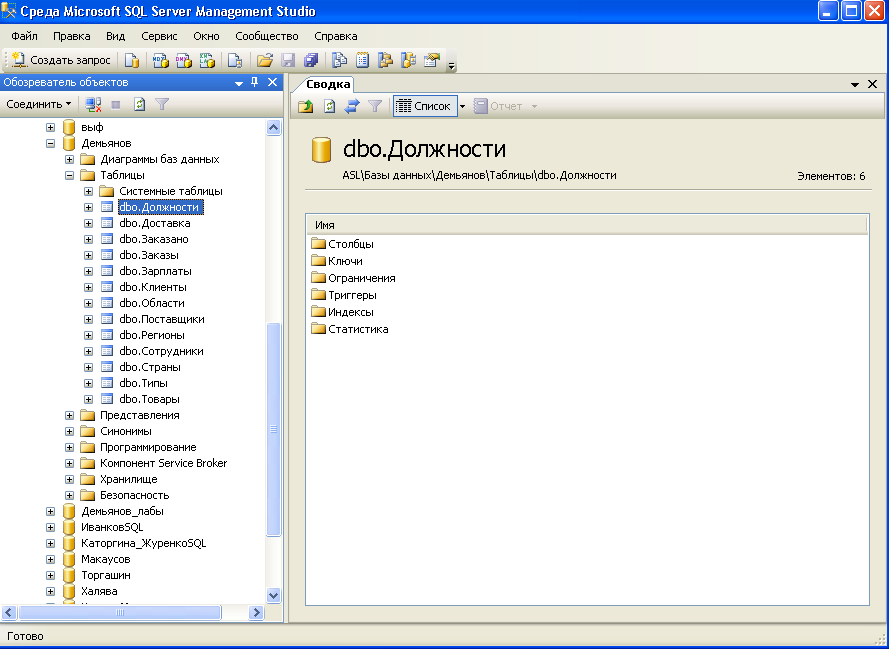
В результате получим базу данных в SQL Server.

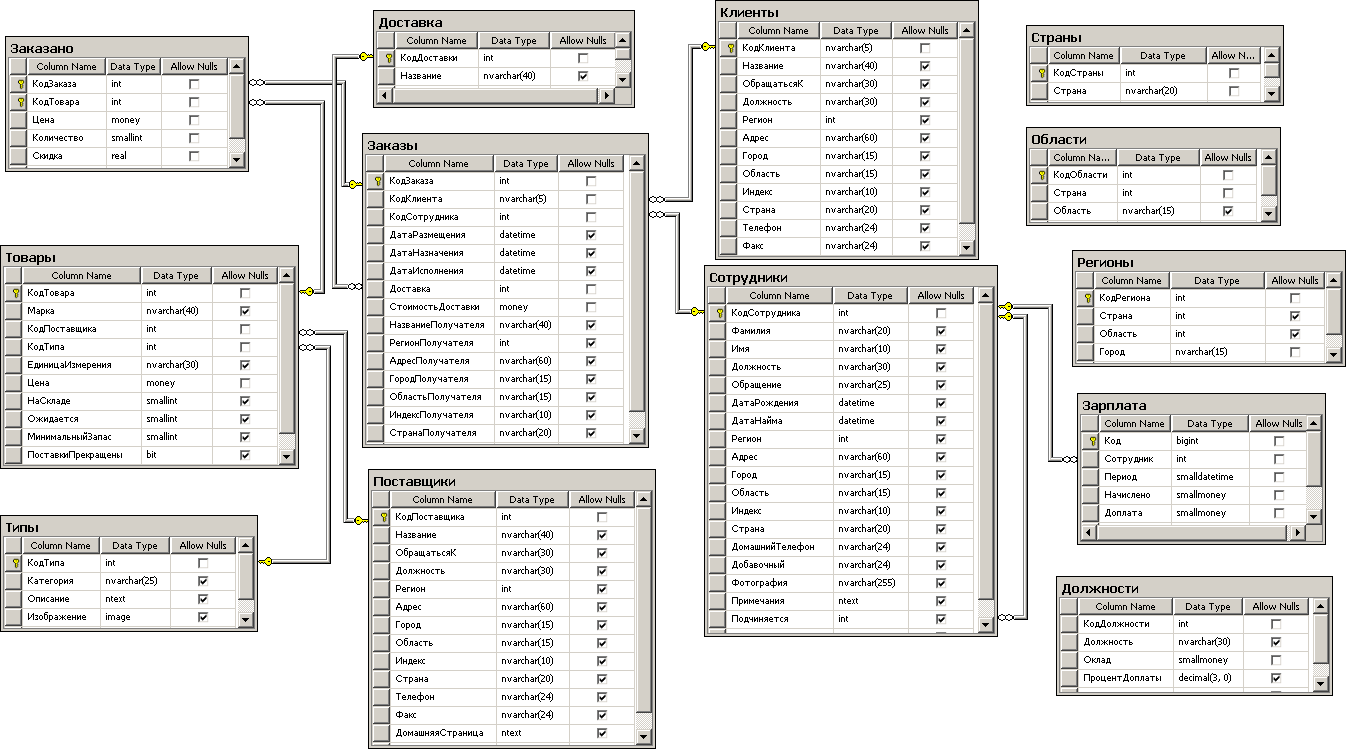


**3)** Добавим столбец «Регионы» в таблицы Сотрудники, Клиенты, Поставщики, Заказы.

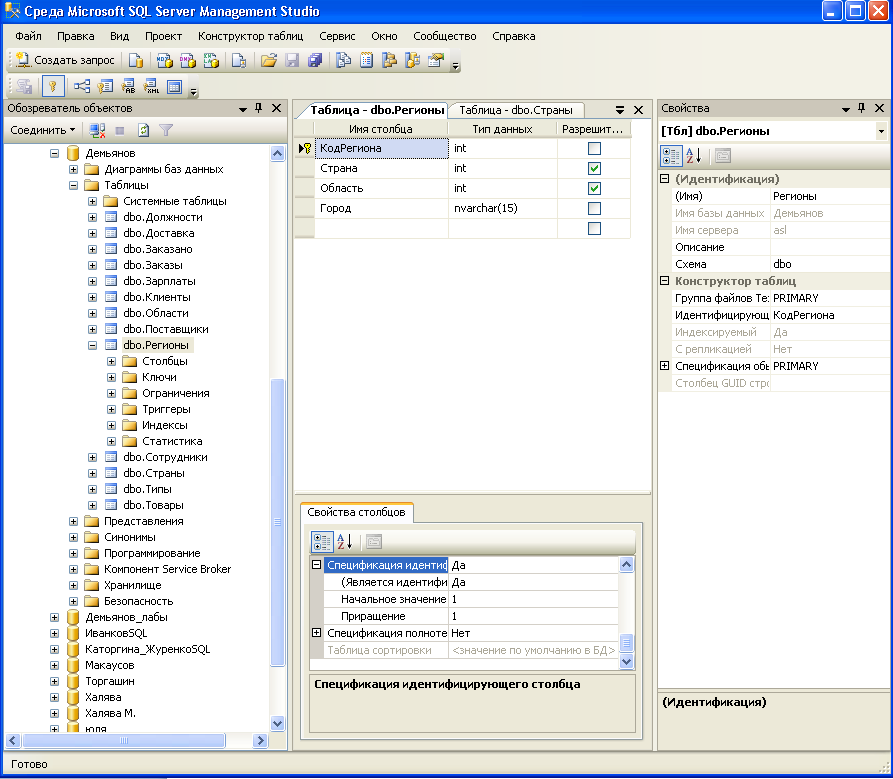


**4)** Создадим новые таблицы Страны, Области, Регионы, Зарплата, Должности.



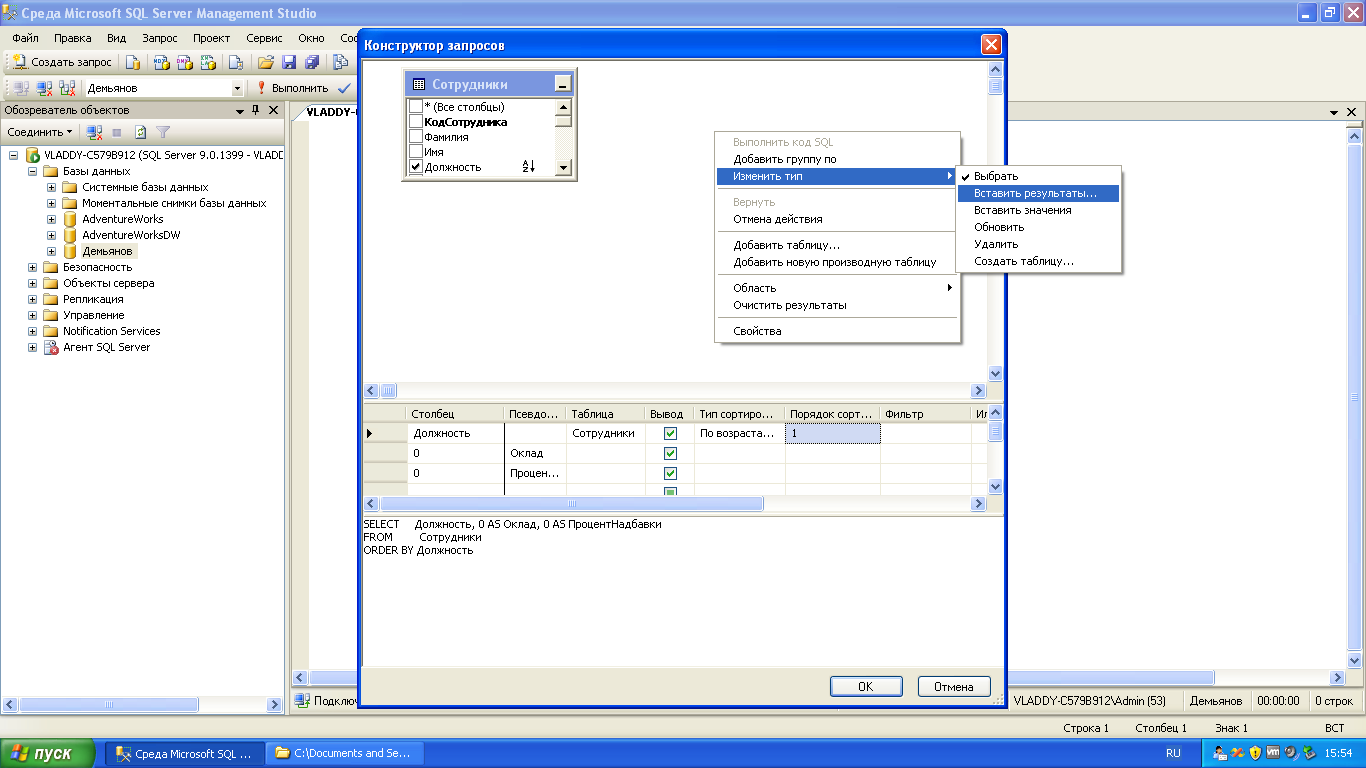


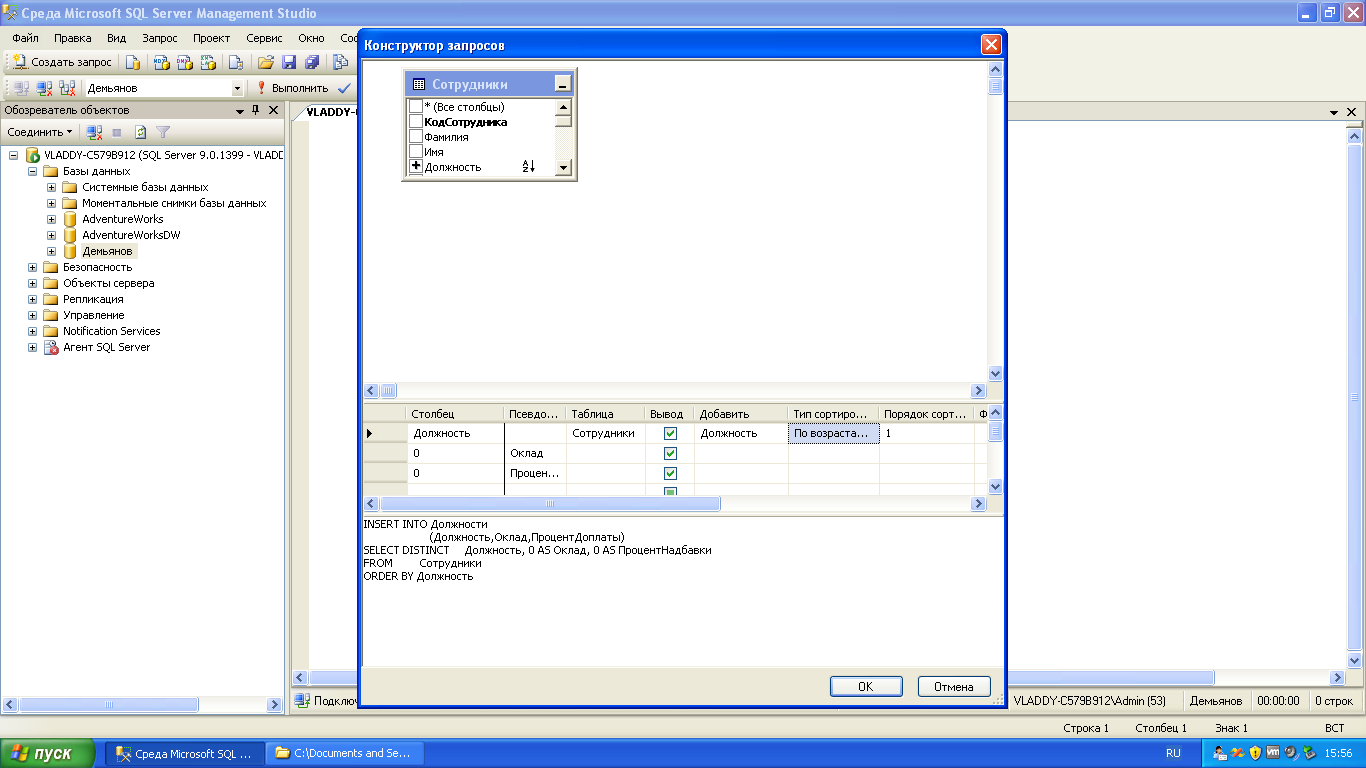
**5)** Первичные целочисленные ключи новых таблиц сделаем счетчиками (включить свойство Identify (1, 1)).

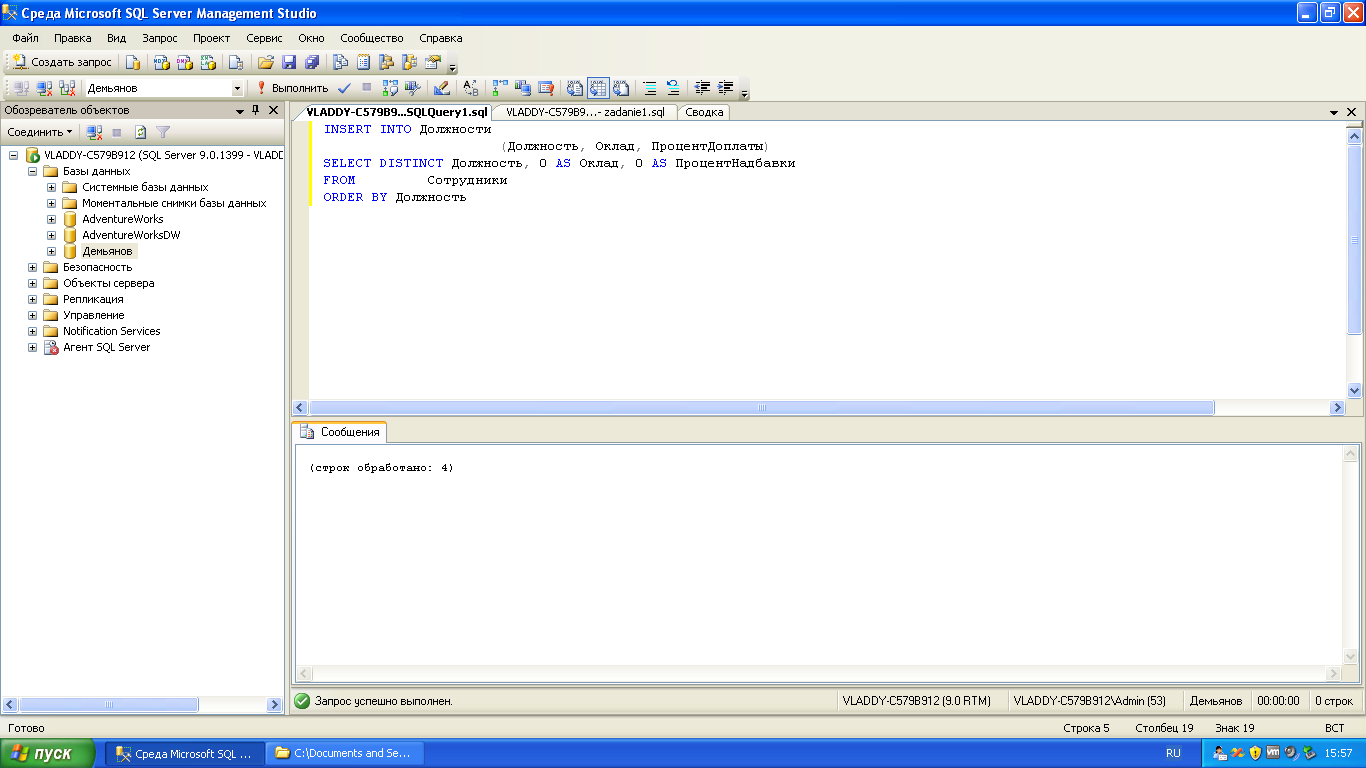


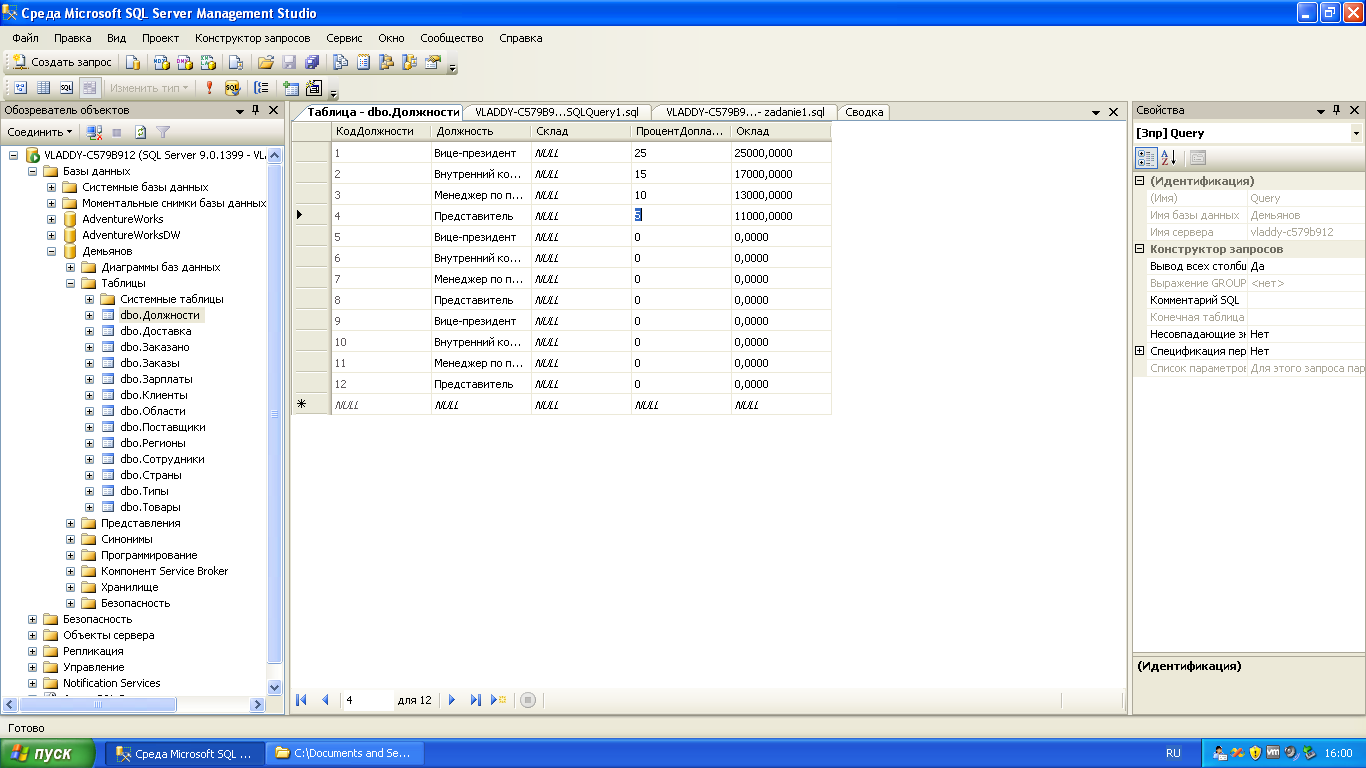
**6)** Заполнить данными созданную таблицу *Должности* данными колонки *Должность* таблицы *Сотрудники*. Сделать это удобнее с помощью редактора запросов, который можно открыть на панели инструментов (после команды- Выполнить запрос, появится вкладка Запрос).











**7)** Заполнить данными созданную таблицу *Страны* данными соответствующих колонок таблиц *Сотрудники*, *Клиенты*, *Поставщики*, *Заказы*.

Для добавления записей в таблицу *Страны* без повторений из таблиц *Сотрудники, Клиенты, Поставщики, Заказы* выполним следующий запрос

*Код запроса:*

SELECT DISTINCT Клиенты.Страна AS Страна FROM Клиенты

UNION SELECT Сотрудники.Страна AS Страна FROM Сотрудники

UNION SELECT Заказы.СтранаПолучателя AS Страна FROM Заказы

UNION SELECT Поставщики.Страна AS Страна FROM Поставщики

ORDER BY Страна



Проверяя результаты запроса, мы увидим что в исходных данных имеется несогласованность – страна «Венесуэла» представлена двумя названиями. Исправим неверное название запросом

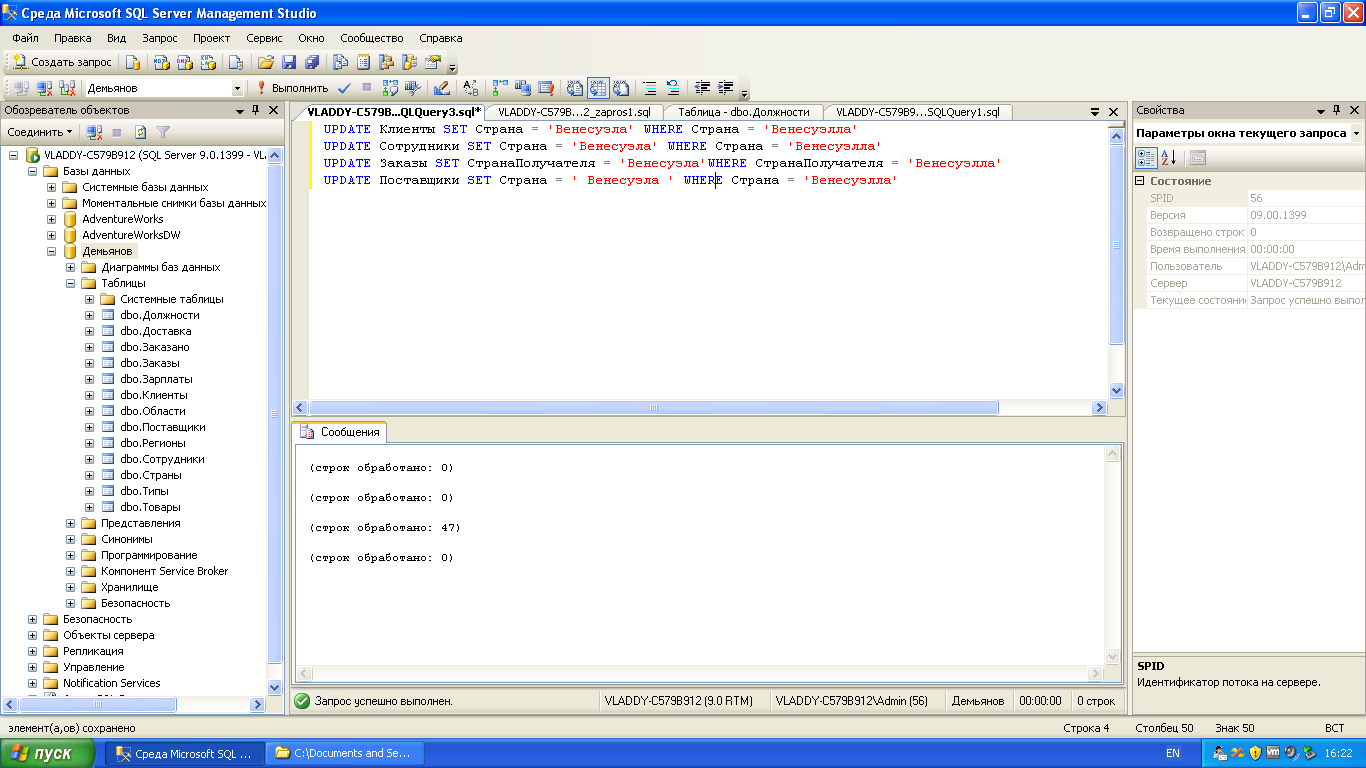
*Код запроса:*

UPDATE Клиенты SET Страна = 'Венесуэла' WHERE Страна = 'Венесуэлла'

UPDATE Сотрудники SET Страна = 'Венесуэла' WHERE Страна = 'Венесуэлла'

UPDATE Заказы SET СтранаПолучателя = 'Венесуэла'WHERE СтранаПолучателя = 'Венесуэлла'

UPDATE Поставщики SET Страна = ' Венесуэла ' WHERE Страна = 'Венесуэлла'



Убедившись, что «Венесуэлла» стала «Венесуэлой», добавим список стран в таблицу Страны

*Код запроса:*

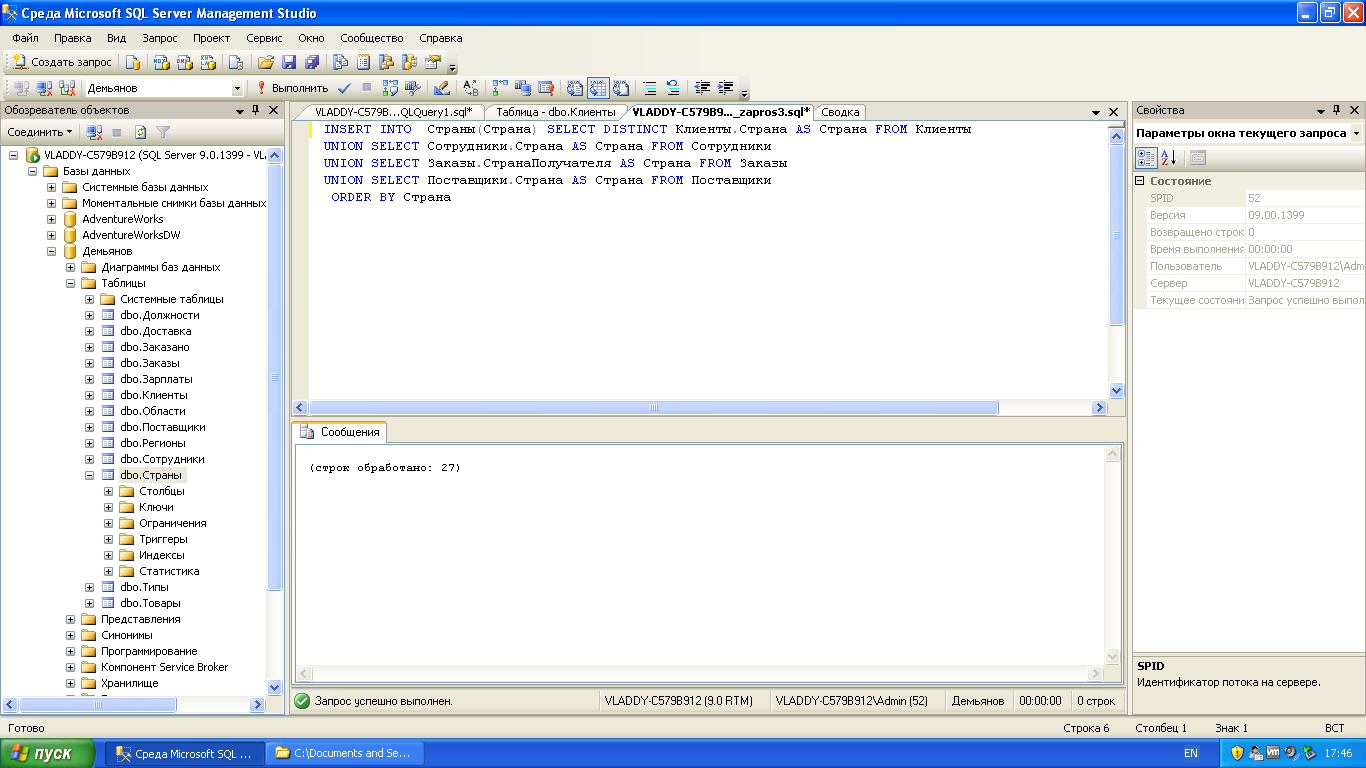
INSERT INTO Страны(Страна) SELECT DISTINCT Клиенты.Страна AS Страна FROM Клиенты

UNION SELECT Сотрудники.Страна AS Страна FROM Сотрудники

UNION SELECT Заказы.СтранаПолучателя AS Страна FROM Заказы

UNION SELECT Поставщики.Страна AS Страна FROM Поставщики

ORDER BY Страна



**8)** Заполнить данными таблицу *Области* данными соответствующих колонок таблиц *Сотрудники*, *Клиенты*, *Поставщики*, *Заказы*;

Для создания соответствия *Страна−Область* создадим временное представление *Области−Страны* с имеющимися данными о странах и их областях

*Код запроса:*

CREATE VIEW \_ОбластиСтраны AS SELECT DISTINCT Клиенты.Область AS Область, Клиенты.Страна AS Страна FROM Клиенты

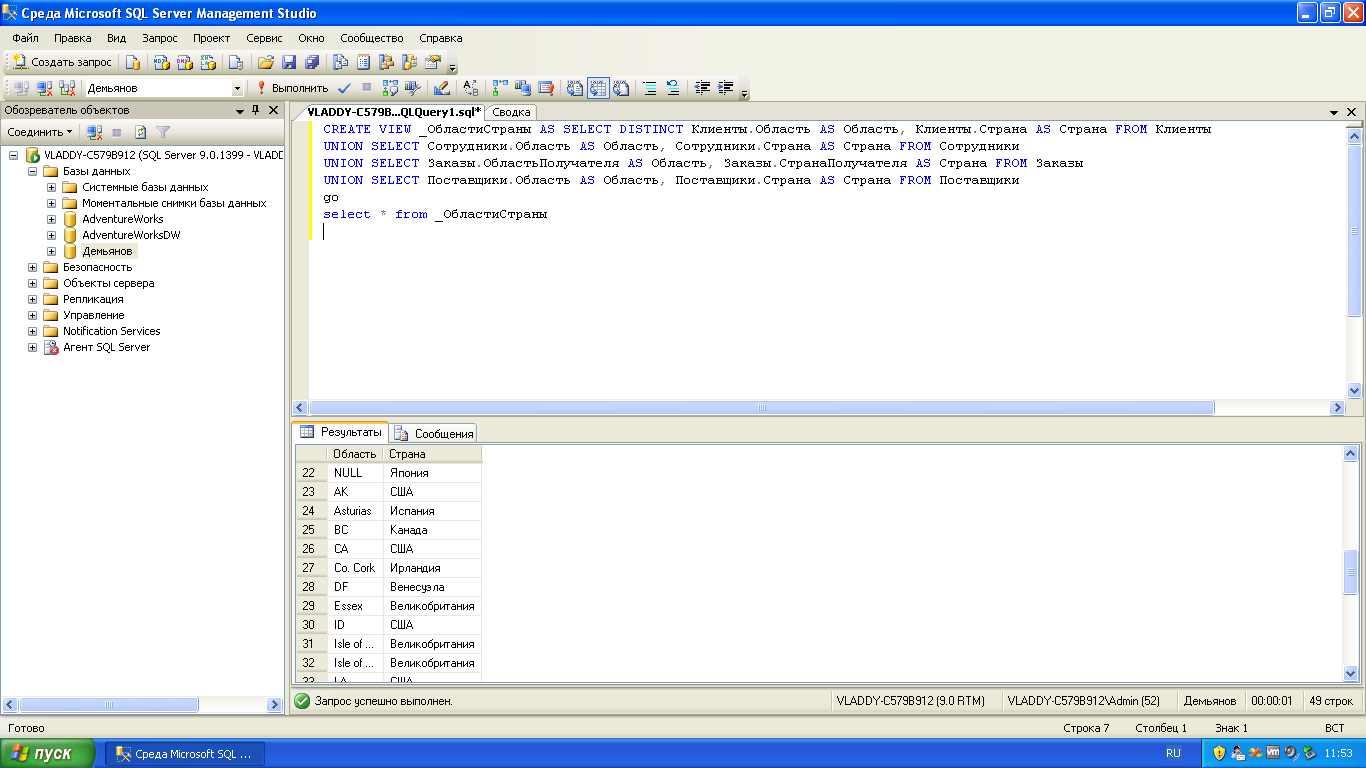
UNION SELECT Сотрудники.Область AS Область, Сотрудники.Страна AS Страна FROM Сотрудники

UNION SELECT Заказы.ОбластьПолучателя AS Область, Заказы.СтранаПолучателя AS Страна FROM Заказы

UNION SELECT Поставщики.Область AS Область, Поставщики.Страна AS Страна FROM Поставщики

go

select \* from \_ОбластиСтраны



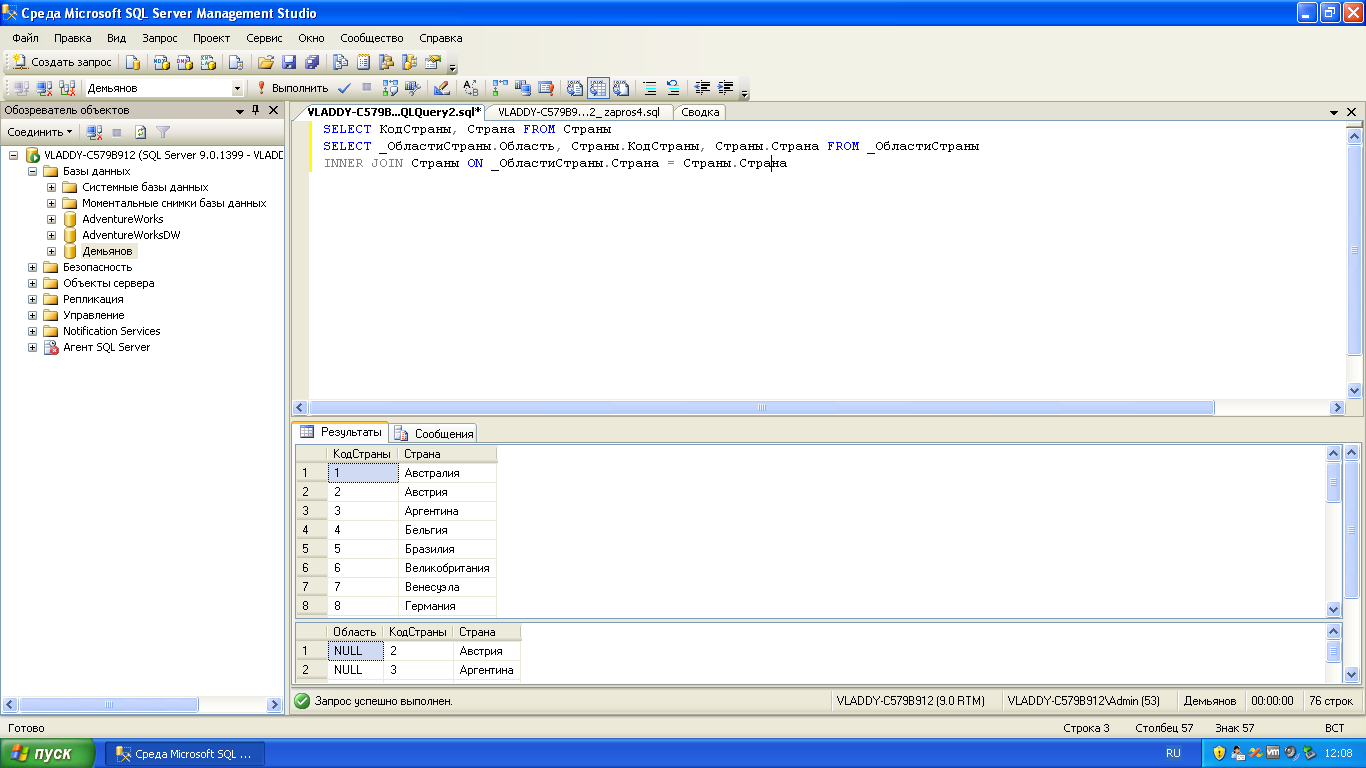
Выбираем поля *КодСтраны* и *Страна* из таблицы *Страны*, и *Область* и *Страна* из представления, при условии объединения записей таблицы и представления по равенству значений поля *Страна*:

*Код запроса:*

SELECT КодСтраны, Страна FROM Страны

SELECT \_ОбластиСтраны.Область, Страны.КодСтраны, Страны.Страна FROM \_ОбластиСтраны

INNER JOIN Страны ON \_ОбластиСтраны.Страна = Страны.Страна



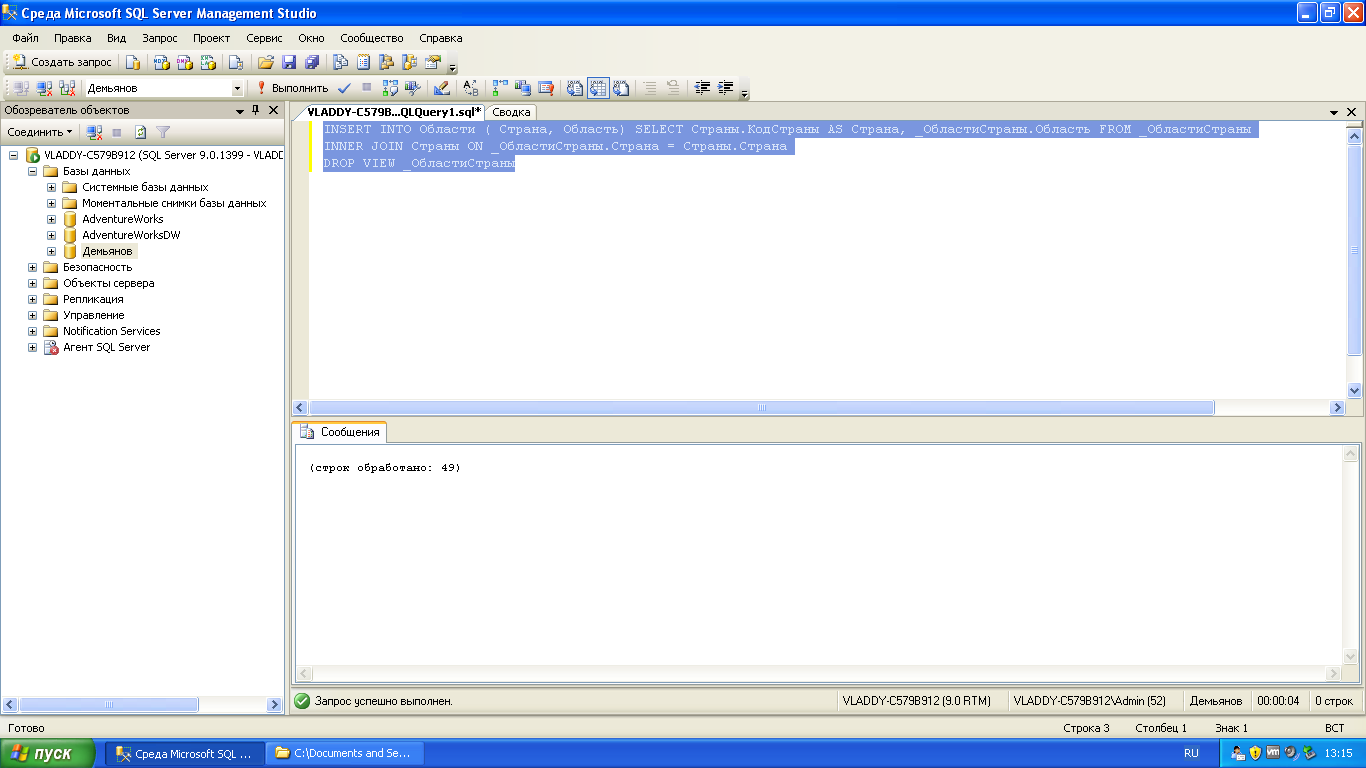
Вставим результаты запроса в таблицу *Области* и удалим временное представление *\_ОбластиСтраны.*

*Код запроса:*

INSERT INTO Области ( Страна, Область) SELECT Страны.КодСтраны AS Страна, \_ОбластиСтраны.Область FROM \_ОбластиСтраны

INNER JOIN Страны ON \_ОбластиСтраны.Страна = Страны.Страна

DROP VIEW \_ОбластиСтраны



Заменим все значения NULL в столбцах *Область* всех таблиц пустой строкой.

*Код запроса:*

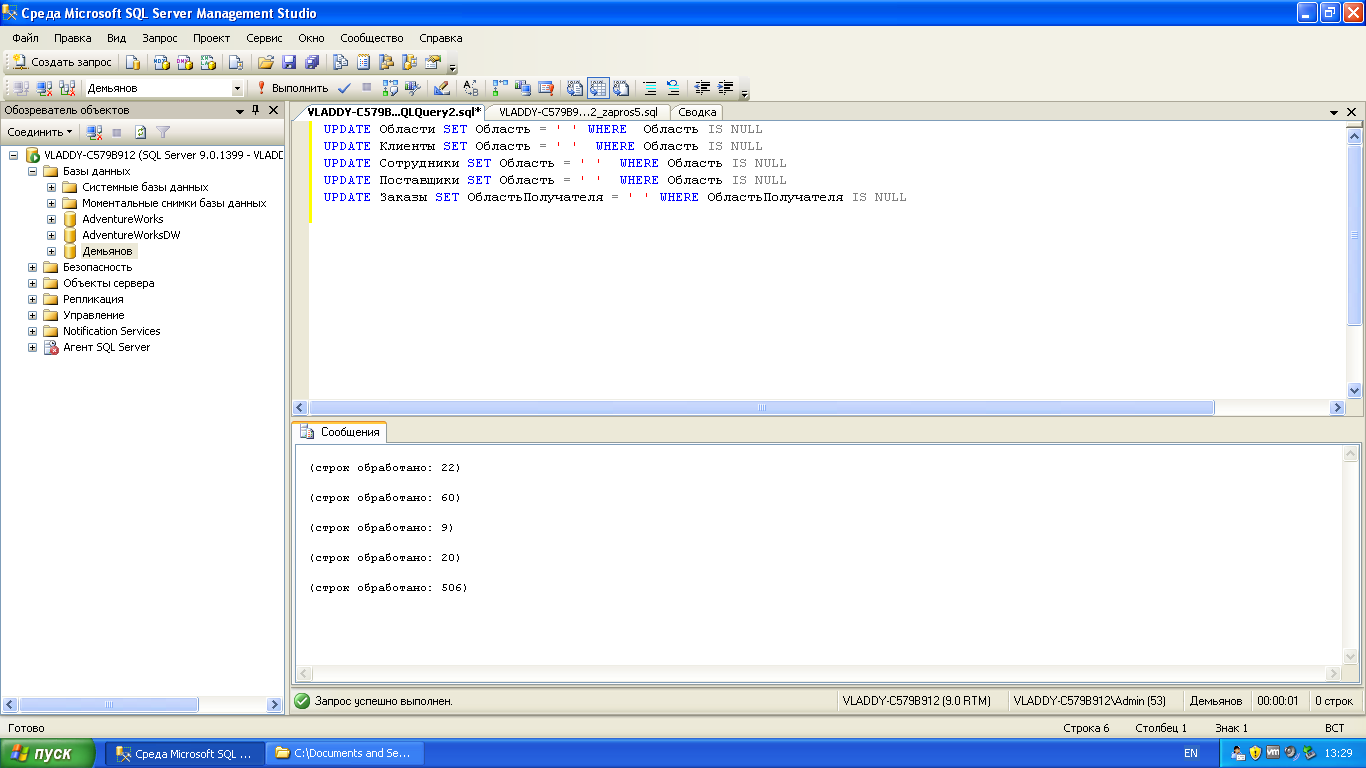
UPDATE Области SET Область = ' ' WHERE Область IS NULL

UPDATE Клиенты SET Область = ' ' WHERE Область IS NULL

UPDATE Сотрудники SET Область = ' 'WHERE Область IS NULL

UPDATE Поставщики SET Область = ' ' WHERE Область IS NULL

UPDATE Заказы SET ОбластьПолучателя = ' 'WHERE ОбластьПолучателя IS NULL



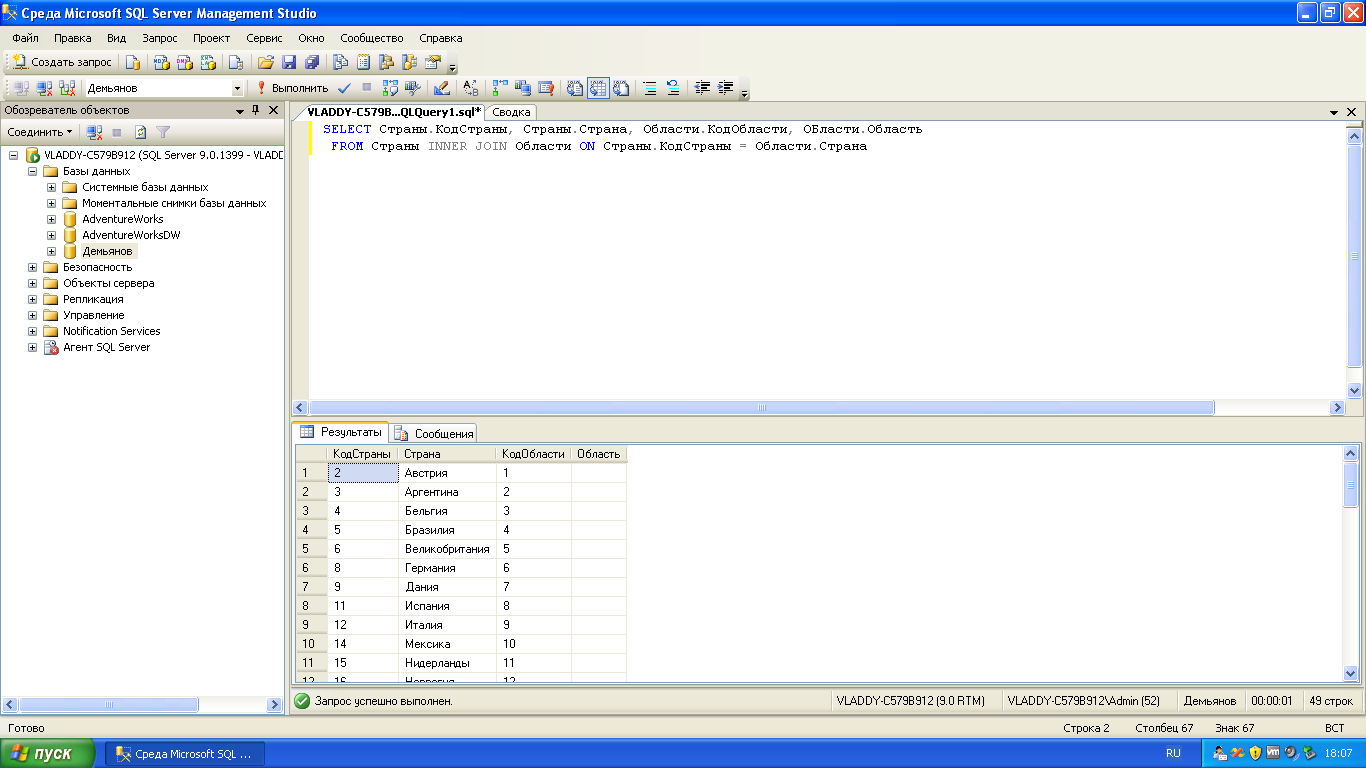
**9)** Заполнить данными таблицу *Регионы* данными соответствующих колонок таблиц *Сотрудники*, *Клиенты*, *Поставщики*, *Заказы*;

Определим страны, области и их коды в таблицах *Страна* и *Область.*

*Код запроса:*

SELECT Страны.КодСтраны, Страны.Страна, Области.КодОбласти, ОБласти.Область

FROM Страны INNER JOIN Области ON Страны.КодСтраны = Области.Страна



Выберем города с соответствующими областями и странами из таблиц *Клиенты, Сотрудники, Поставщики* и *Заказы.*

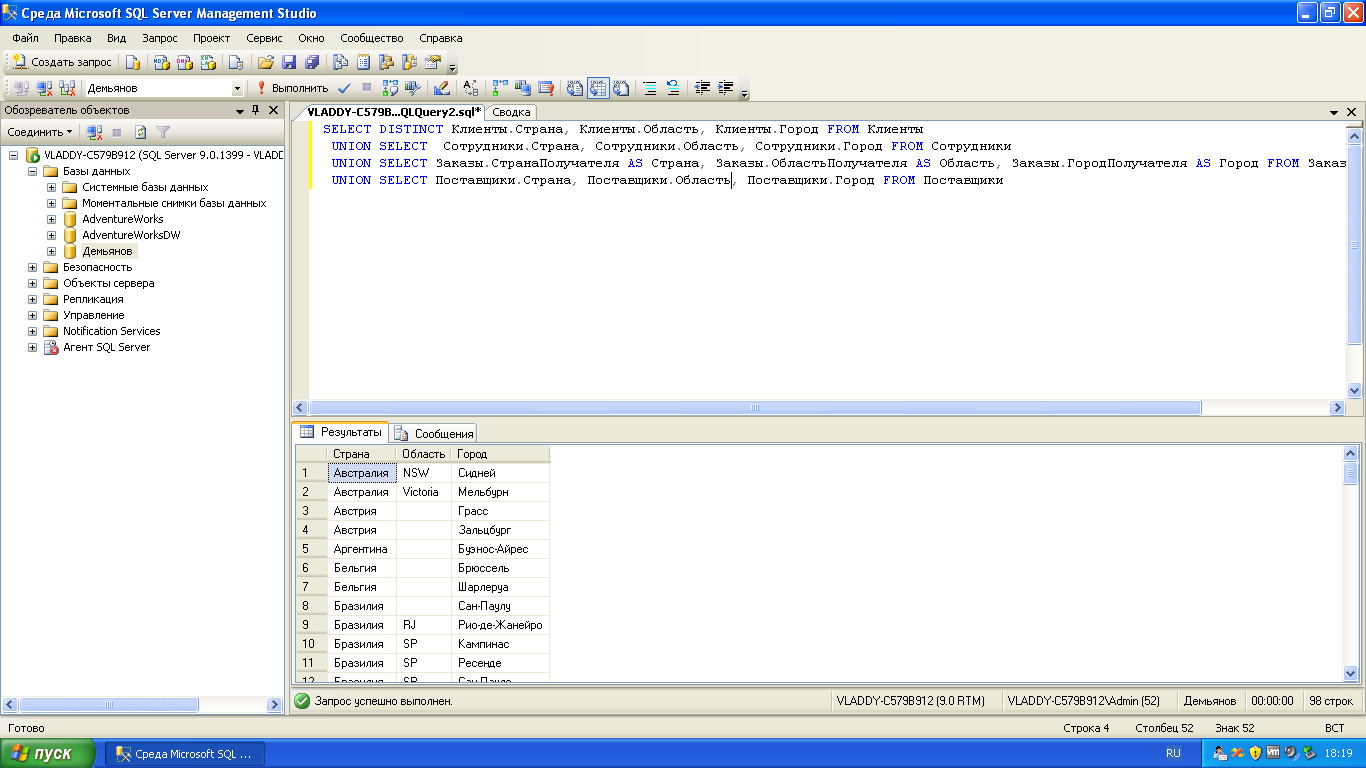
*Код запроса:*

SELECT DISTINCT Клиенты.Страна, Клиенты.Область, Клиенты.Город FROM Клиенты

UNION SELECT Сотрудники.Страна, Сотрудники.Область, Сотрудники.Город FROM Сотрудники

UNION SELECT Заказы.СтранаПолучателя AS Страна, Заказы.ОбластьПолучателя AS Область, Заказы.ГородПолучателя AS Город FROM Заказы

UNION SELECT Поставщики.Страна, Поставщики.Область, Поставщики.Город FROM Поставщики



На основе запросов выше создадим представление

*Код представления:*

CREATE VIEW \_СО AS SELECT Страны.КодСтраны, Страны.Страна, Области.КодОбласти, Области.Область

FROM Страны INNER JOIN Области ON Страны.КодСтраны = Области.Страна

GO

CREATE VIEW \_СОГ AS SELECT DISTINCT Клиенты.Страна, Клиенты.Область, Клиенты.Город FROM Клиенты

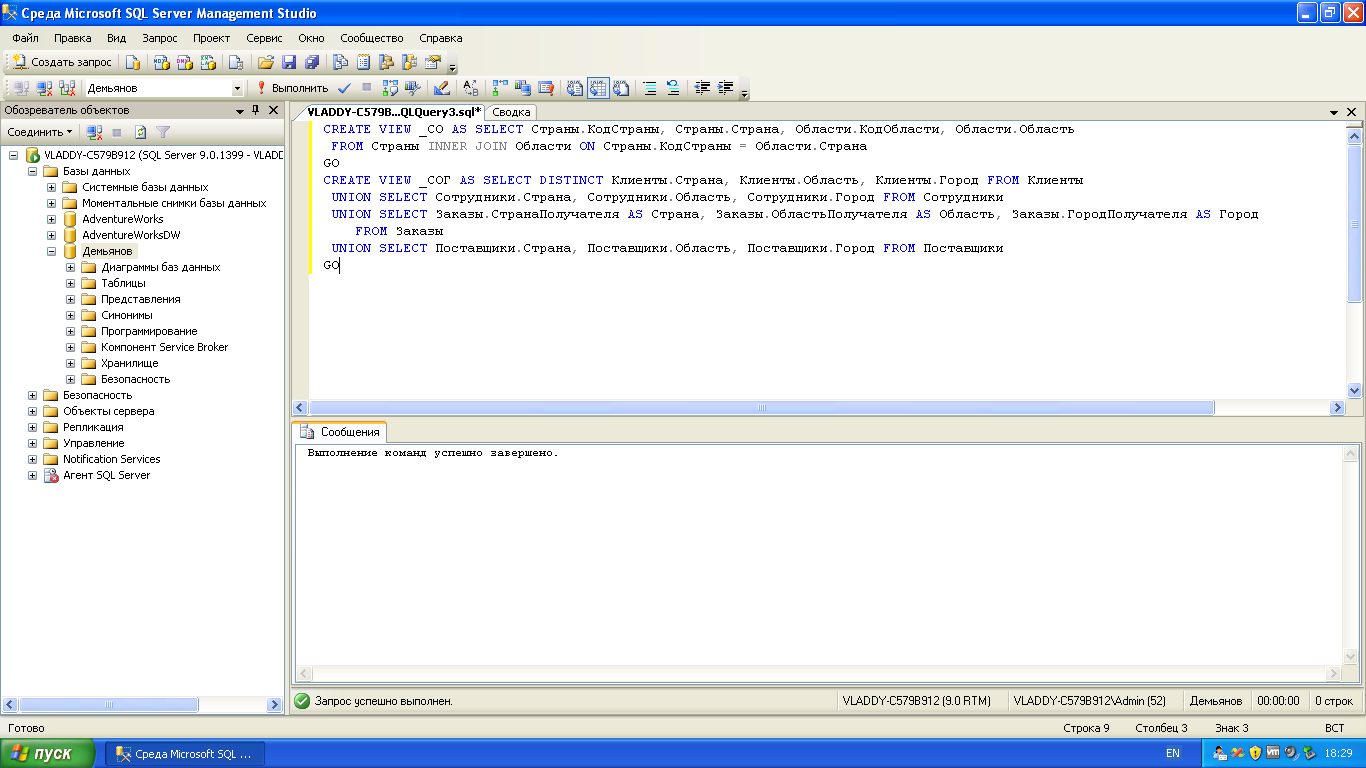
UNION SELECT Сотрудники.Страна, Сотрудники.Область, Сотрудники.Город FROM Сотрудники

UNION SELECT Заказы.СтранаПолучателя AS Страна, Заказы.ОбластьПолучателя AS Область, Заказы.ГородПолучателя AS Город

FROM Заказы

UNION SELECT Поставщики.Страна, Поставщики.Область, Поставщики.Город FROM Поставщики

GO

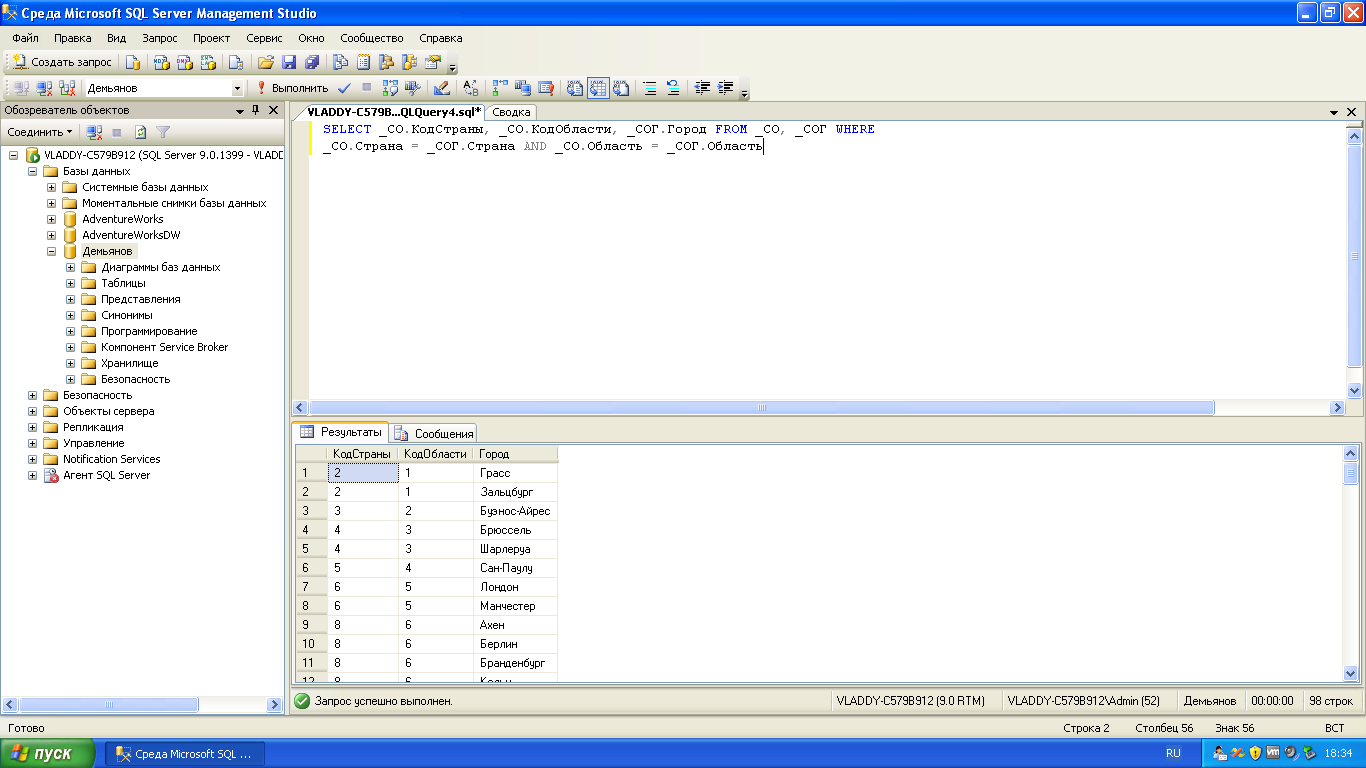


Объединим данные представлений и проверим результат.

*Код запроса:*

SELECT \_СО.КодСтраны, \_СО.КодОбласти, \_СОГ.Город FROM \_СО, \_СОГ WHERE

\_СО.Страна = \_СОГ.Страна AND \_СО.Область = \_СОГ.Область



Объединенные данные обоих представлений вставим в таблицу *Регионы*

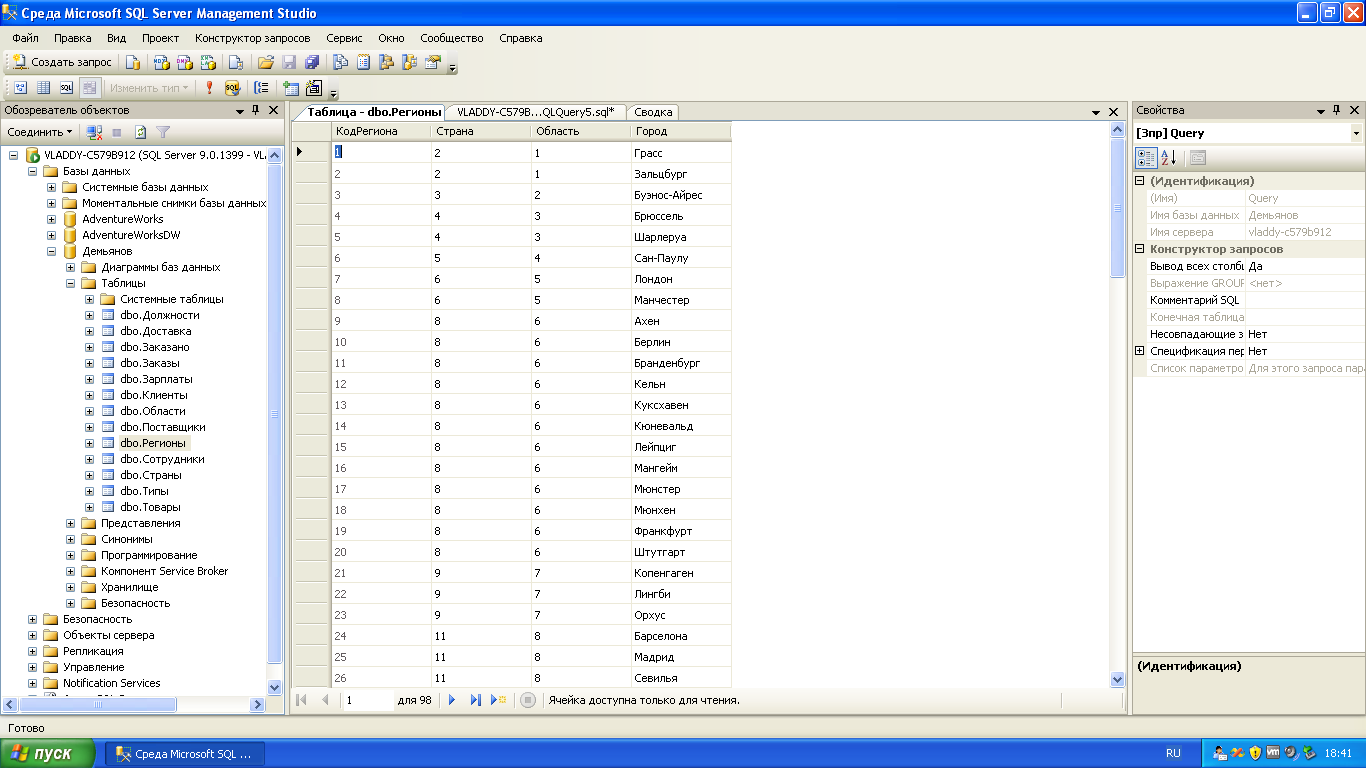
*Код запроса:*

INSERT INTO Регионы(Страна, Область, Город) SELECT \_СО.КодСтраны, \_СО.КодОбласти AS Область, \_СОГ.Город

FROM \_СО, \_СОГ WHERE

\_СО.Страна = \_СОГ.Страна AND \_СО.Область = \_СОГ.Область

Таблица *Регионы* показывает успешное выполнение операции.



Удалим временные представления.

*Код запроса:*

DROP VIEW \_СО

DROP VIEW \_СОГ

