**лАБОРАТОРНАЯ РАБОТА №2**

**Основы языка программирования Transact−SQL СУБД MS SQL Server**

**Цель работы:** изучение алгоритмических конструкций языка Transact-SQL

**Задачи:**

1) Устранить дублирование данных в таблицах БД, созданных в ходе выполнения лабораторной работы №1 и установить связи между таблицами;

2) Обеспечить возможность изменения окладов сотрудников с течением времени и определения актуальных значений на любой период;

3) Обеспечить условия сохранения целостности данных.

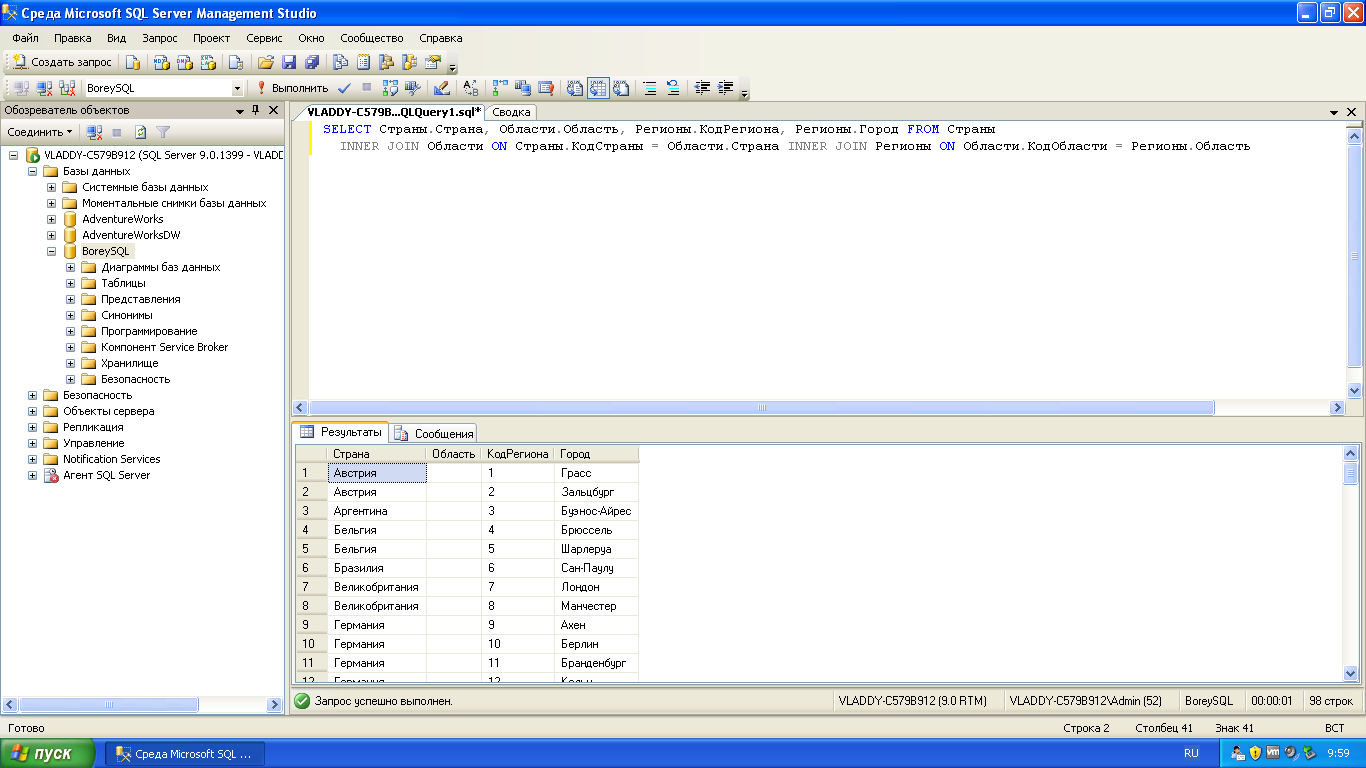
**Ход работы*:***

**1)** Заполним поле *Регион* таблиц *Клиенты, Сотрудники, Поставщики, РегионПолучателя* таблицы *Заказы* значениями поля *КодРегиона* таблица *Регионы*.

*Код запроса:*

SELECT Страны.Страна, Области.Область, Регионы.КодРегиона, Регионы.Город FROM Страны

INNER JOIN Области ON Страны.КодСтраны = Области.Страна INNER JOIN Регионы ON Области.КодОбласти = Регионы.Область

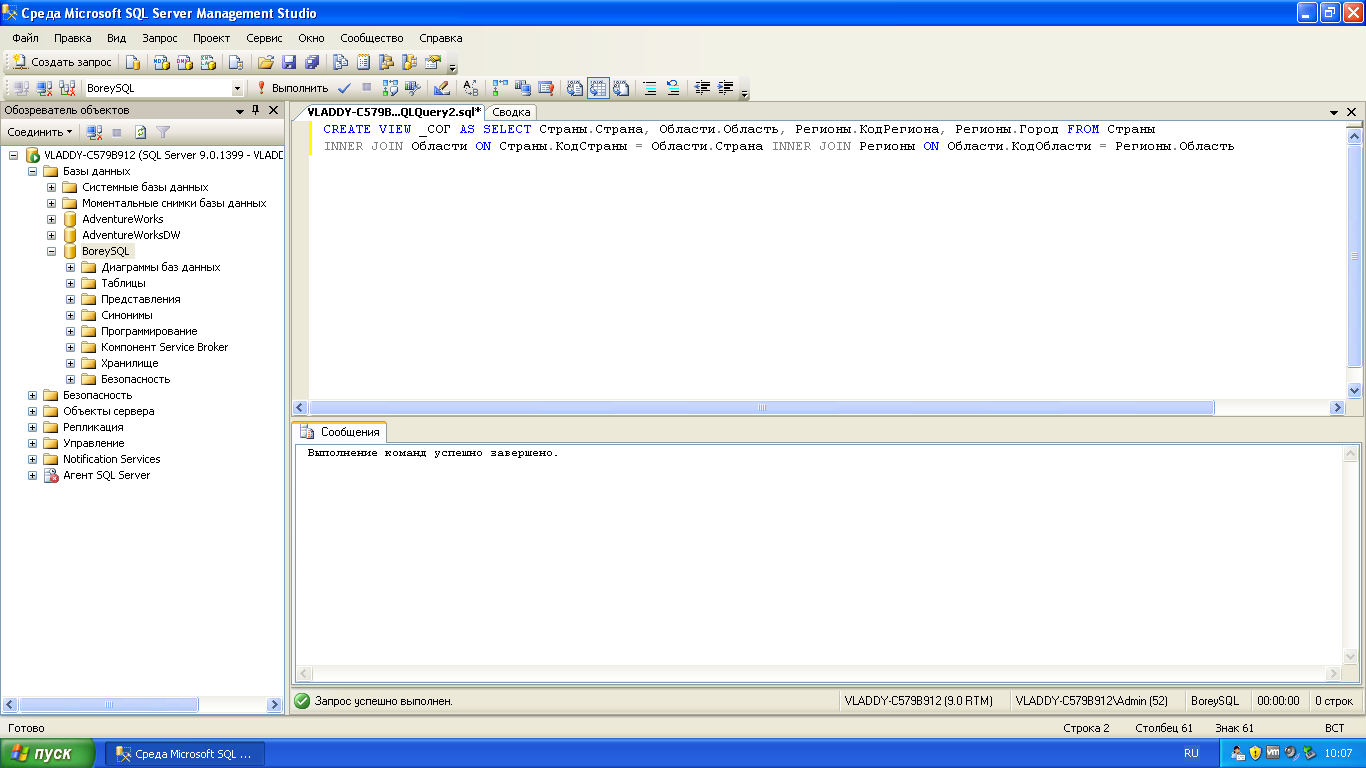


На основе данного запроса составим временное представление

*Код представления:*

CREATE VIEW \_СОГ AS SELECT Страны.Страна, Области.Область, Регионы.КодРегиона, Регионы.Город FROM Страны

INNER JOIN Области ON Страны.КодСтраны = Области.Страна INNER JOIN Регионы ON Области.КодОбласти = Регионы.Область



Создадим хранимую процедуру *Вставка-Регионов.*

CREATE PROCEDURE ВставкаРегионов AS

DECLARE @i Int, @РегВсего INT

DECLARE @КодРег INT, @Стр NCHAR(20), @Обл NCHAR(15), @Гор NCHAR(15)

DECLARE КурсРег CURSOR LOCAL SCROLL FOR SELECT КодРегиона, Страна, Область, Город FROM \_СОГ

OPEN КурсРег

SELECT @РегВсего = Count(\*) FROM \_СОГ

SET @i = 1

WHILE @i <= @РегВсего

BEGIN

FETCH ABSOLUTE @i FROM КурсРег INTO @КодРег, @Стр, @Обл, @Гор

UPDATE Клиенты SET Регионы = @КодРег WHERE Страна = @Стр AND Область = @Обл AND Город = @Гор

SET @i = @i + 1

END

CLOSE КурсРег

DEALLOCATE КурсРег

GO

EXEC ВставкаРегионов

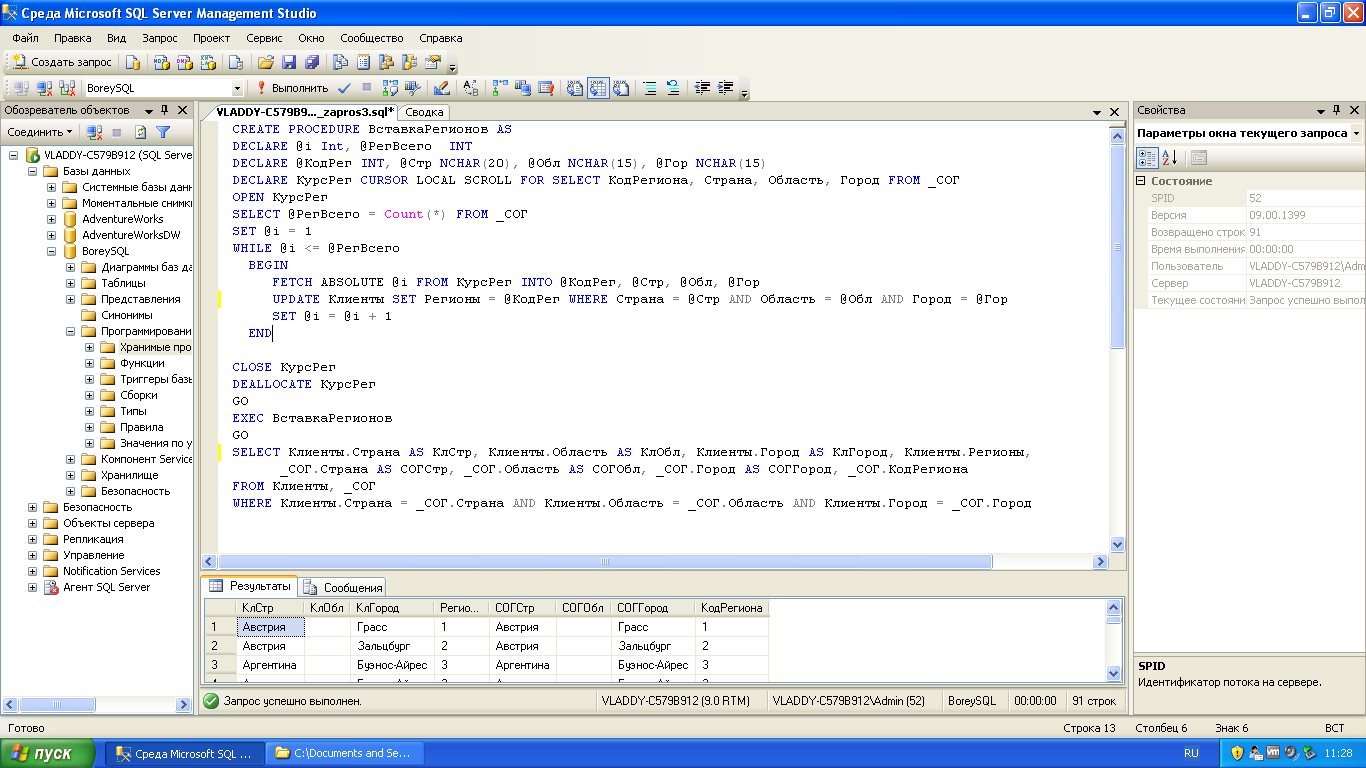
GO

SELECT Клиенты.Страна AS КлСтр, Клиенты.Область AS КлОбл, Клиенты.Город AS КлГород, Клиенты.Регионы,

\_СОГ.Страна AS СОГСтр, \_СОГ.Область AS СОГОбл, \_СОГ.Город AS СОГГород, \_СОГ.КодРегиона

FROM Клиенты, \_СОГ

WHERE Клиенты.Страна = \_СОГ.Страна AND Клиенты.Область = \_СОГ.Область AND Клиенты.Город = \_СОГ.Город



Заполним поле *Регион* таблицы *Поставщики*

*Код запроса:*

CREATE PROCEDURE ВставкаРегионовПост AS

DECLARE @i Int, @РегВсего INT

DECLARE @КодРег INT, @Стр NCHAR(20), @Обл NCHAR(15), @Гор NCHAR(15)

DECLARE КурсРег CURSOR LOCAL SCROLL FOR SELECT КодРегиона, Страна, Область, Город FROM \_СОГ

OPEN КурсРег

SELECT @РегВсего = Count(\*) FROM \_СОГ

SET @i = 1

WHILE @i <= @РегВсего

BEGIN

FETCH ABSOLUTE @i FROM КурсРег INTO @КодРег, @Стр, @Обл, @Гор

UPDATE Поставщики SET Регионы = @КодРег WHERE Страна = @Стр AND Область = @Обл AND Город = @Гор

SET @i = @i + 1

END

CLOSE КурсРег

DEALLOCATE КурсРег

GO

EXEC ВставкаРегионовПост

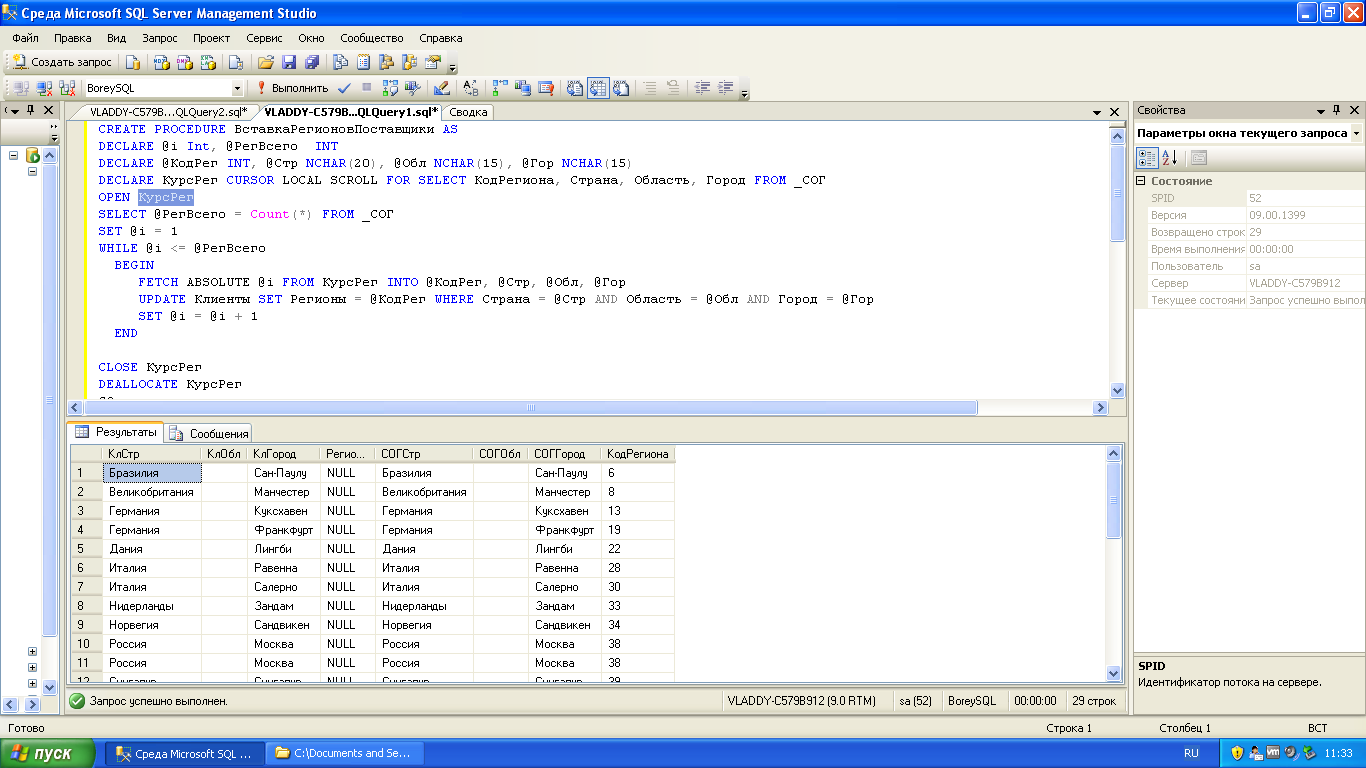
GO

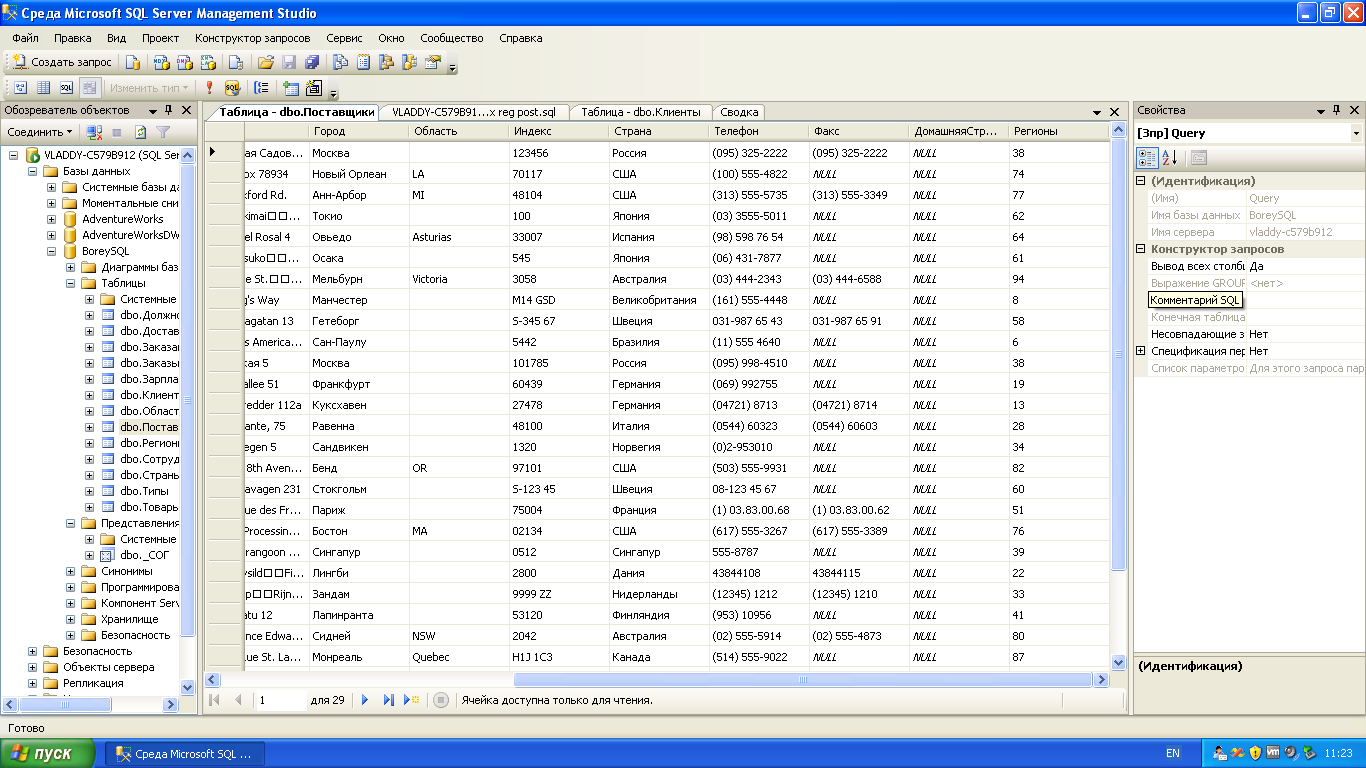
SELECT Поставщики.Страна AS КлСтр, Поставщики.Область AS КлОбл, Поставщики.Город AS КлГород, Поставщики.Регионы,

\_СОГ.Страна AS СОГСтр, \_СОГ.Область AS СОГОбл, \_СОГ.Город AS СОГГород, \_СОГ.КодРегиона

FROM Поставщики, \_СОГ

WHERE Поставщики.Страна = \_СОГ.Страна AND Поставщики.Область = \_СОГ.Область AND Поставщики.Город = \_СОГ.Город





Заполним данными поле *Регион* таблицы *Сотрудники.*

*Код запроса:*

CREATE PROCEDURE ВставкаРегионовСотр AS

DECLARE @i Int, @РегВсего INT

DECLARE @КодРег INT, @Стр NCHAR(20), @Обл NCHAR(15), @Гор NCHAR(15)

DECLARE КурсРег CURSOR LOCAL SCROLL FOR SELECT КодРегиона, Страна, Область, Город FROM \_СОГ

OPEN КурсРег

SELECT @РегВсего = Count(\*) FROM \_СОГ

SET @i = 1

WHILE @i <= @РегВсего

BEGIN

FETCH ABSOLUTE @i FROM КурсРег INTO @КодРег, @Стр, @Обл, @Гор

UPDATE Сотрудники SET Регионы = @КодРег WHERE Страна = @Стр AND Область = @Обл AND Город = @Гор

SET @i = @i + 1

END

CLOSE КурсРег

DEALLOCATE КурсРег

GO

EXEC ВставкаРегионовСотрв

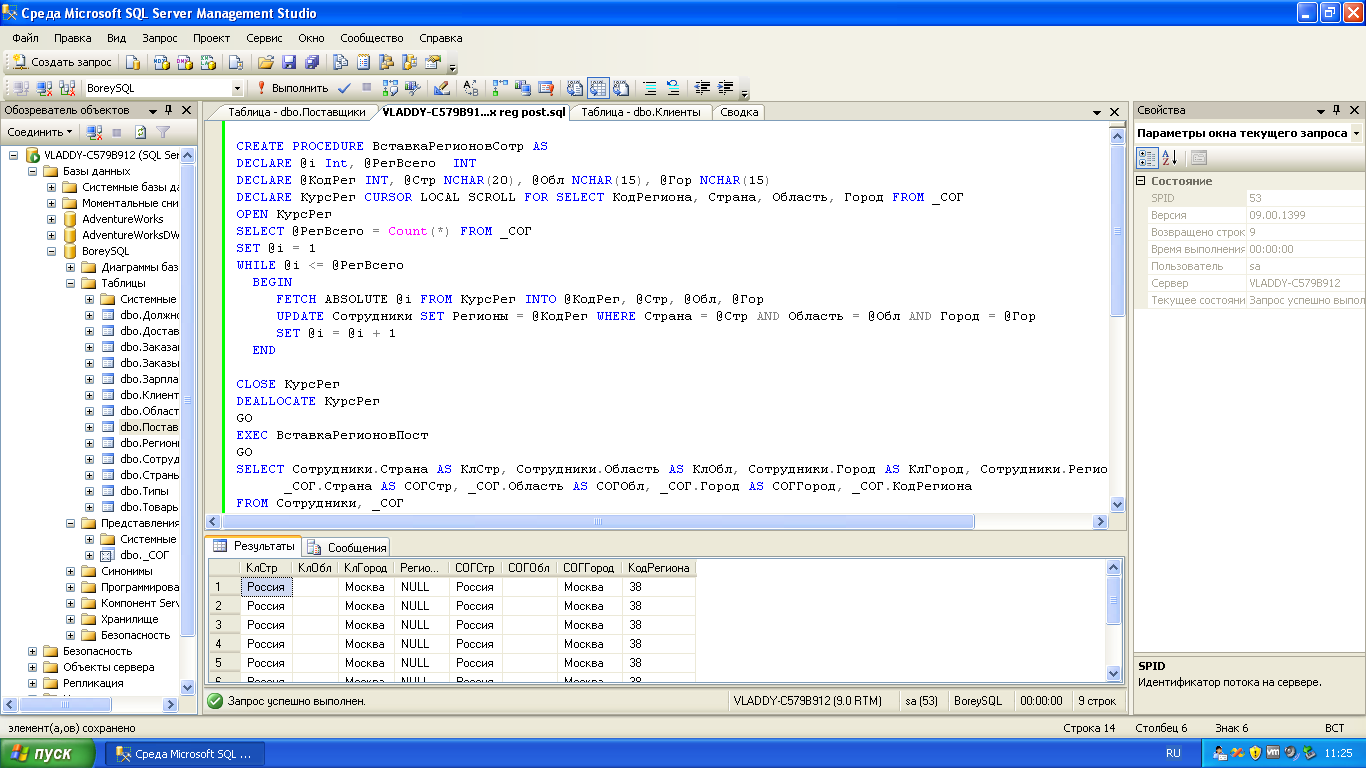
GO

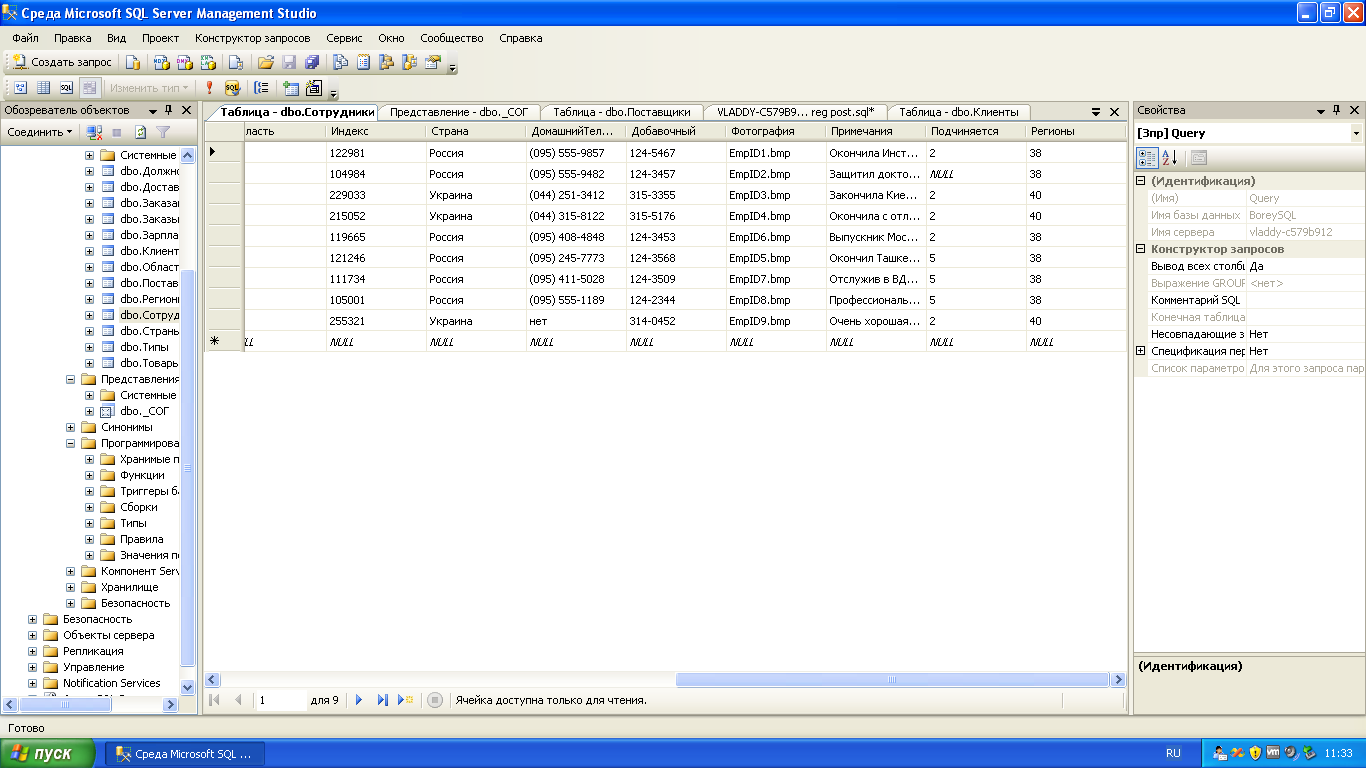
SELECT Сотрудники.Страна AS КлСтр, Сотрудники.Область AS КлОбл, Сотрудники.Город AS КлГород, Сотрудники.Регионы,

\_СОГ.Страна AS СОГСтр, \_СОГ.Область AS СОГОбл, \_СОГ.Город AS СОГГород, \_СОГ.КодРегиона

FROM Сотрудники, \_СОГ

WHERE Сотрудники.Страна = \_СОГ.Страна AND Сотрудники.Область = \_СОГ.Область AND Сотрудники.Город = \_СОГ.Город





Заполним поле *Регионы* таблицы *Заказы.*

*Код запроса:*

CREATE PROCEDURE ВставкаРегионовЗаказы1 AS

DECLARE @i Int, @РегВсего INT

DECLARE @КодРег INT, @Стр NCHAR(20), @Обл NCHAR(15), @Гор NCHAR(15)

DECLARE КурсРег CURSOR LOCAL SCROLL FOR SELECT КодРегиона, Страна, Область, Город FROM \_СОГ

OPEN КурсРег

SELECT @РегВсего = Count(\*) FROM \_СОГ

SET @i = 1

WHILE @i <= @РегВсего

BEGIN

FETCH ABSOLUTE @i FROM КурсРег INTO @КодРег, @Стр, @Обл, @Гор

UPDATE Заказы SET Регионы = @КодРег WHERE СтранаПолучателя = @Стр AND ОбластьПолучателя = @Обл AND ГородПолучателя = @Гор

SET @i = @i + 1

END

CLOSE КурсРег

DEALLOCATE КурсРег

GO

EXEC ВставкаРегионовЗаказы1

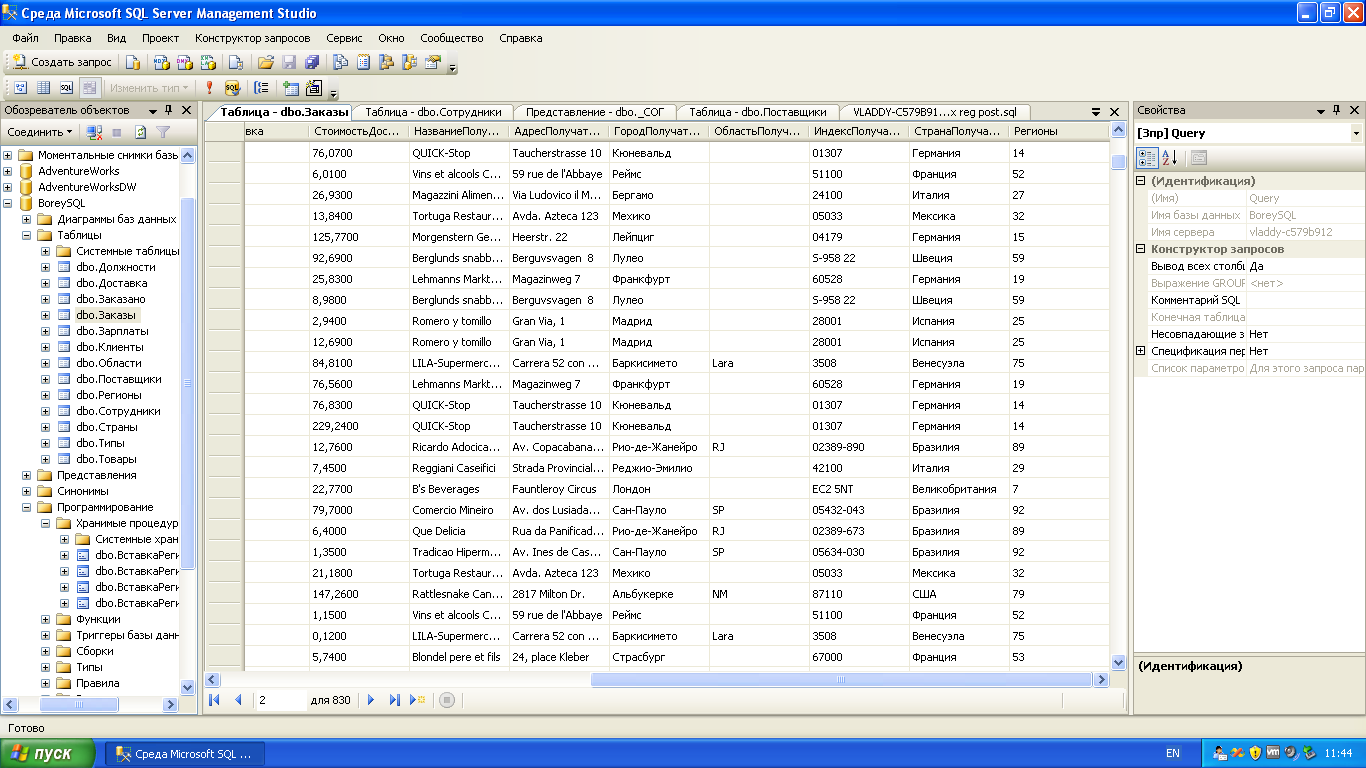
GO

SELECT Заказы.СтранаПолучателя AS КлСтр, Заказы.ОбластьПолучателя AS КлОбл, Заказы.ГородПолучателя AS КлГород, Заказы.Регионы,

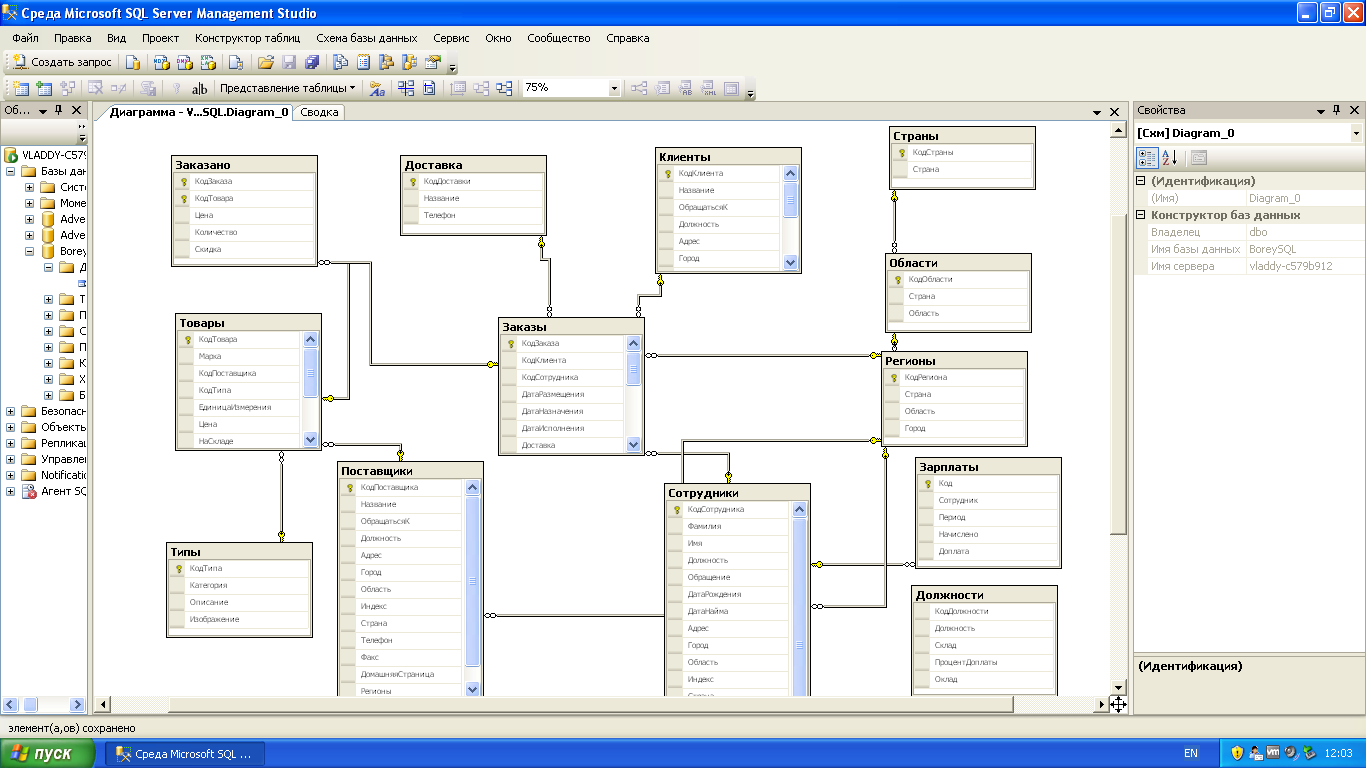
\_СОГ.Страна AS СОГСтр, \_СОГ.Область AS СОГОбл, \_СОГ.Город AS СОГГород, \_СОГ.КодРегиона

FROM Заказы, \_СОГ

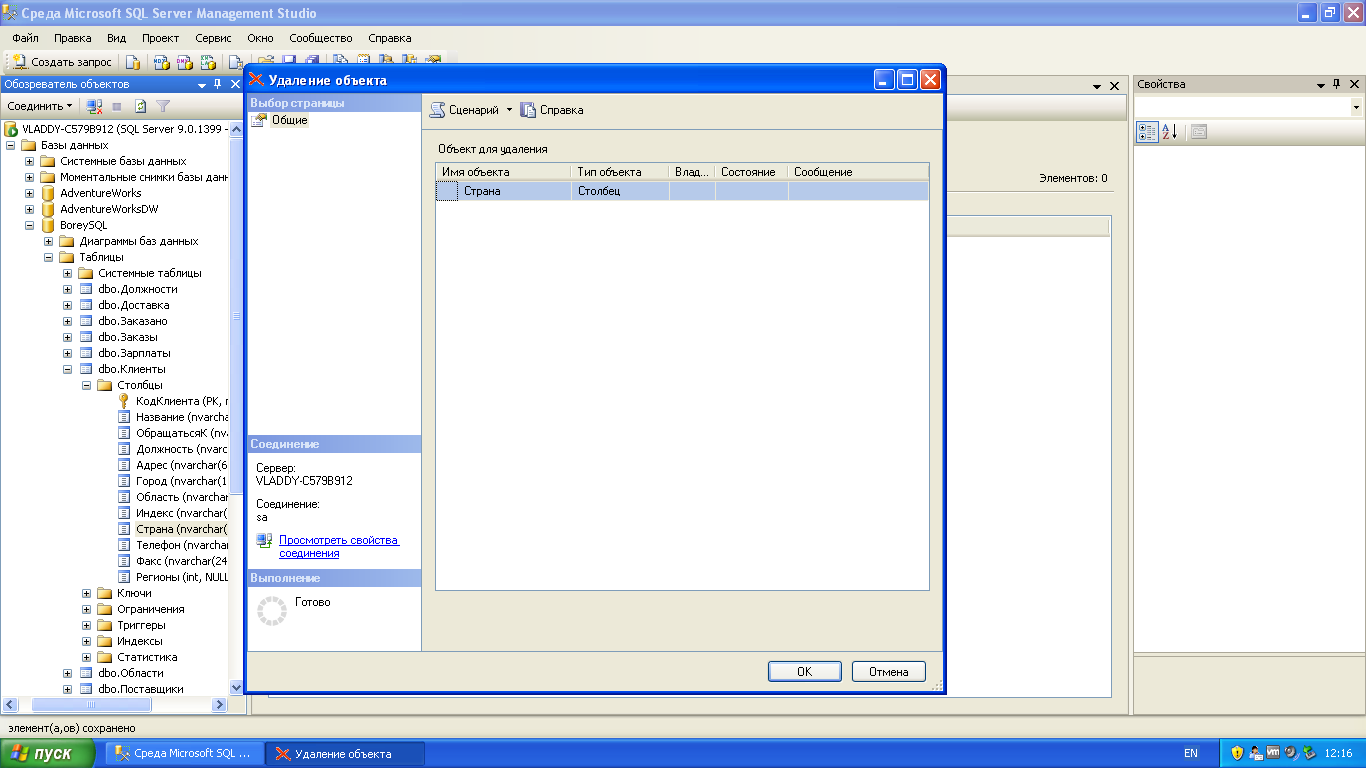
WHERE Заказы.СтранаПолучателя = \_СОГ.Страна AND Заказы.ОбластьПолучателя = \_СОГ.Область AND Заказы.ГородПолучателя = \_СОГ.Город



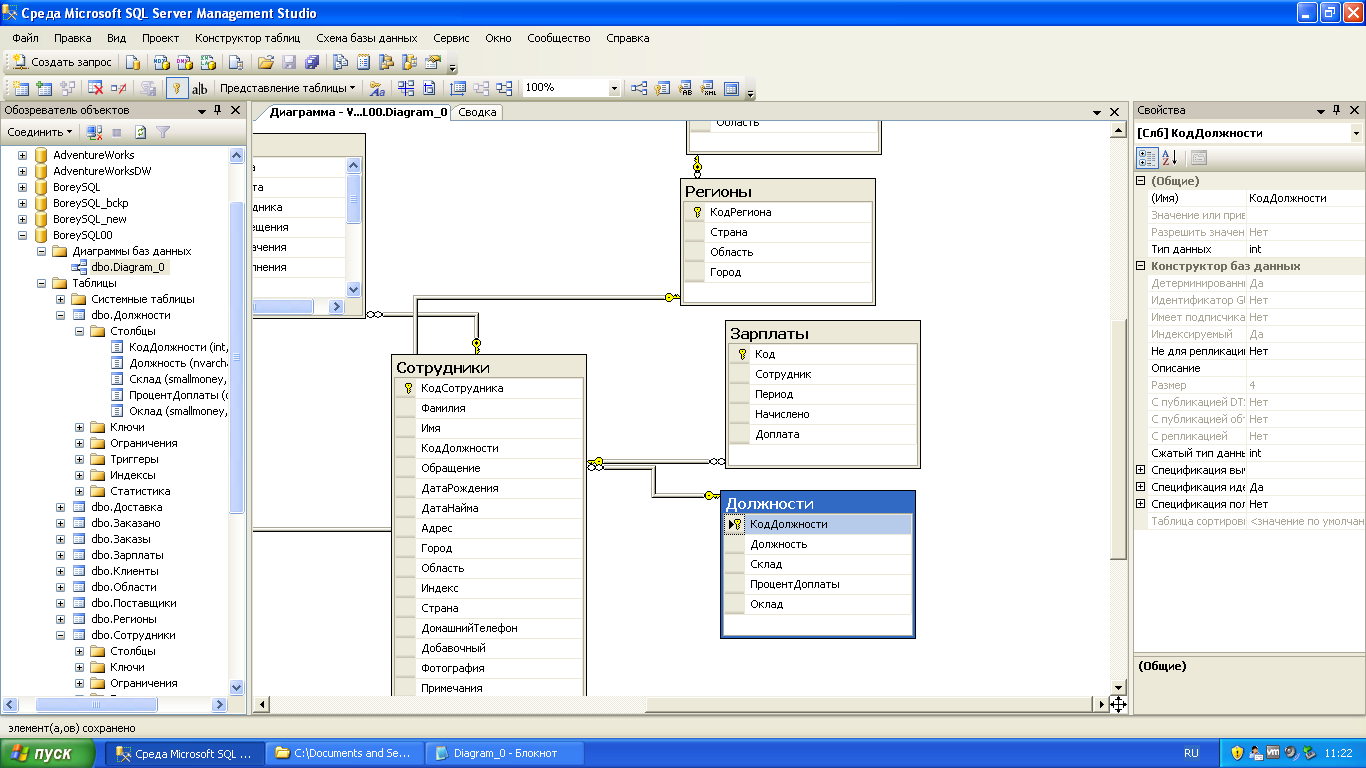
Установим связи между таблицами:



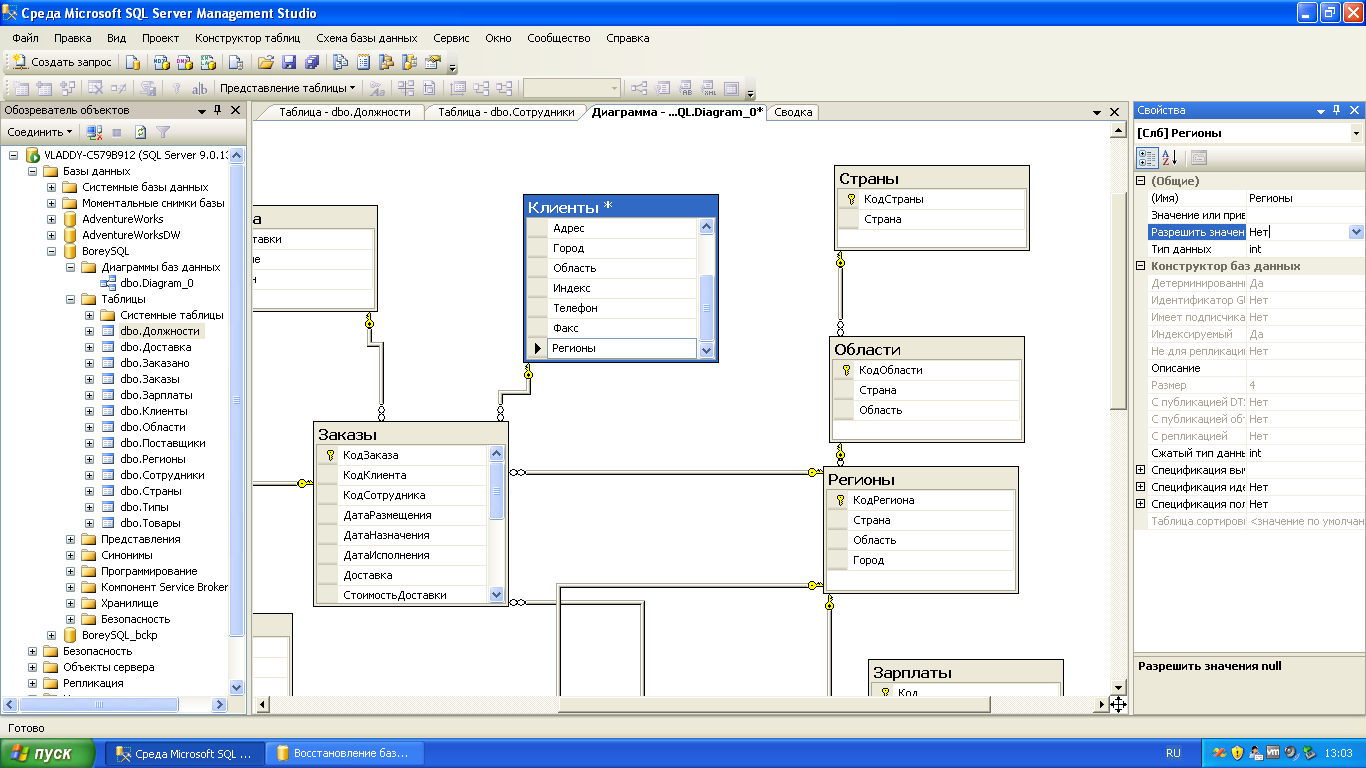
Удалим столбцы с данными о странах, областях и городах из таблиц *Клиенты*, *Сотрудники*, *Поставщики* и *Заказы.*



Установим связи между таблицами *Должности* и *Сотрудники*:

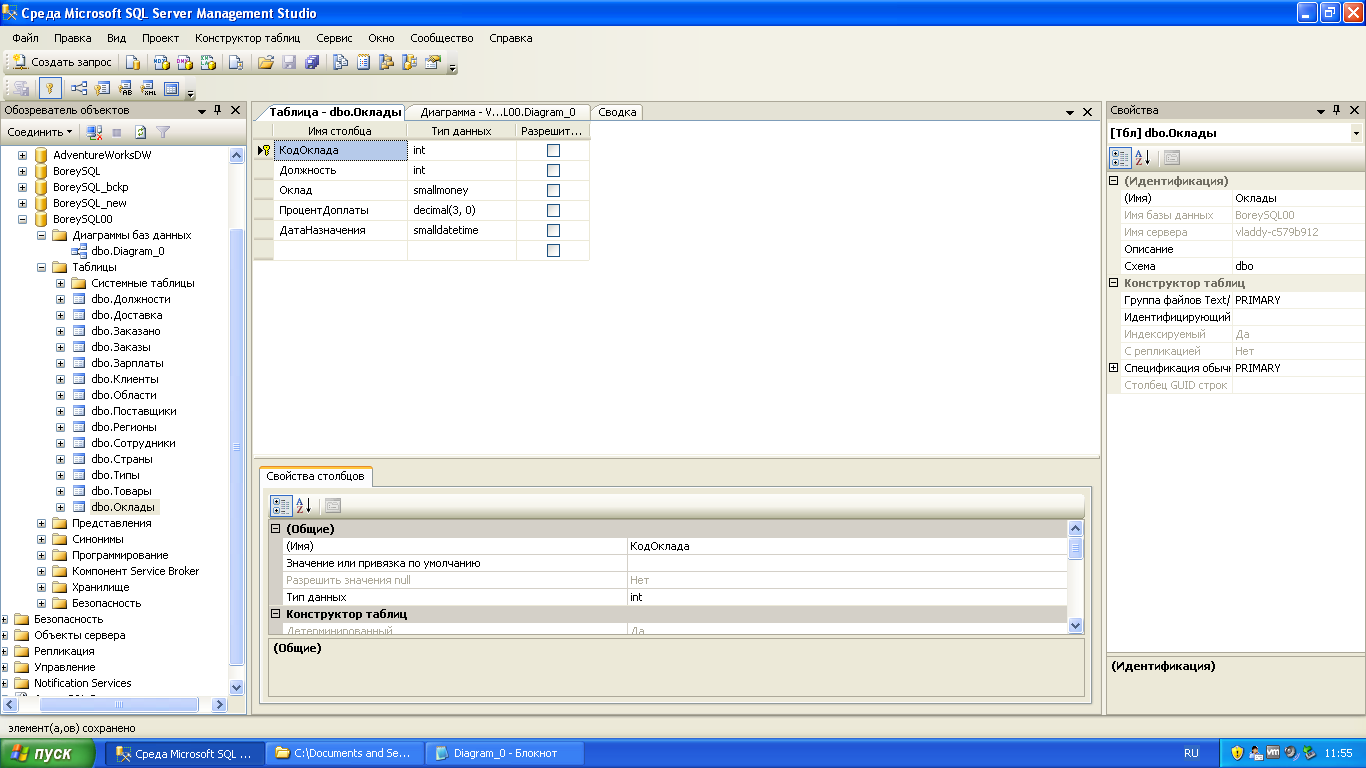


Отключаем возможность иметь значение NUL в столбцах *Регионы* таблиц *Сотрудники*, *Клиенты*, *Поставщики*, *Заказы.*

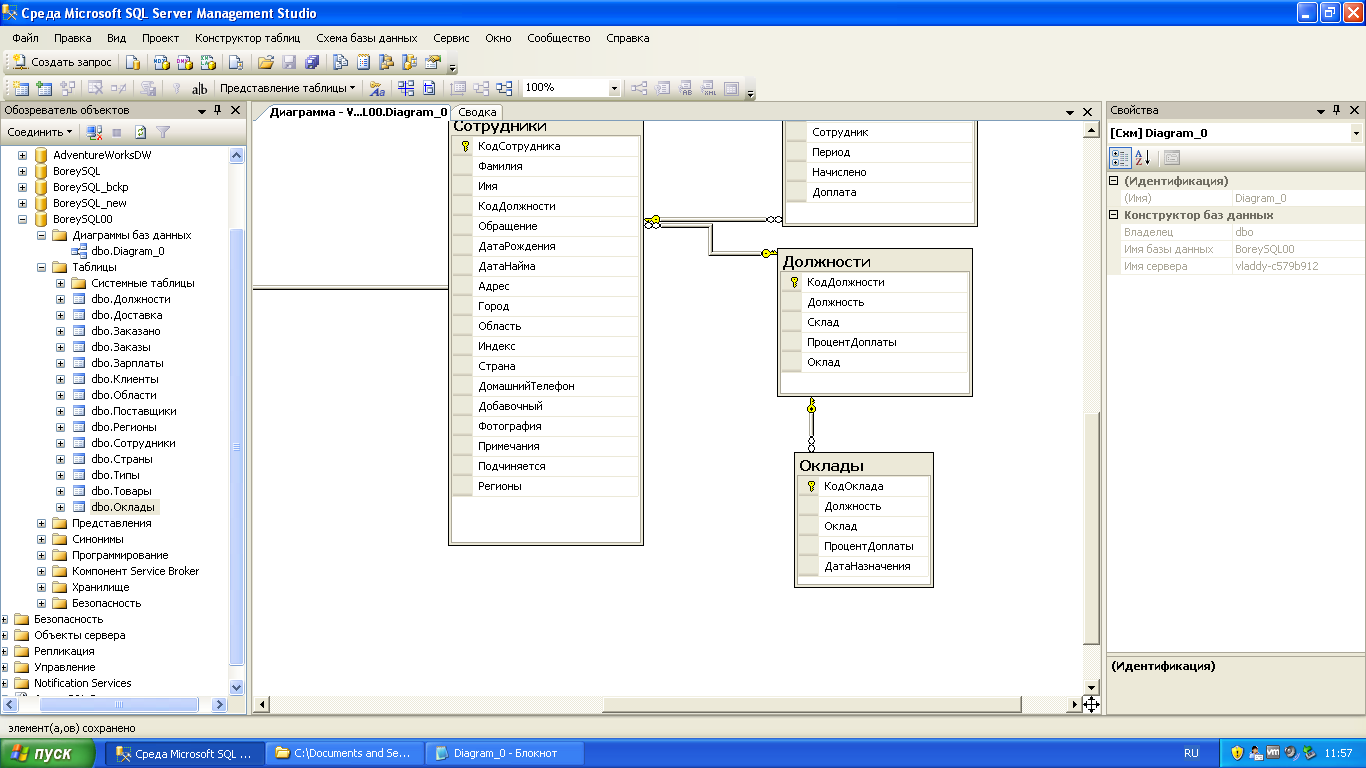


Из таблицы *Регионы* удаляется столбец *Страна* и переименовывается столбец *Область* в *СтранаОбласть.*

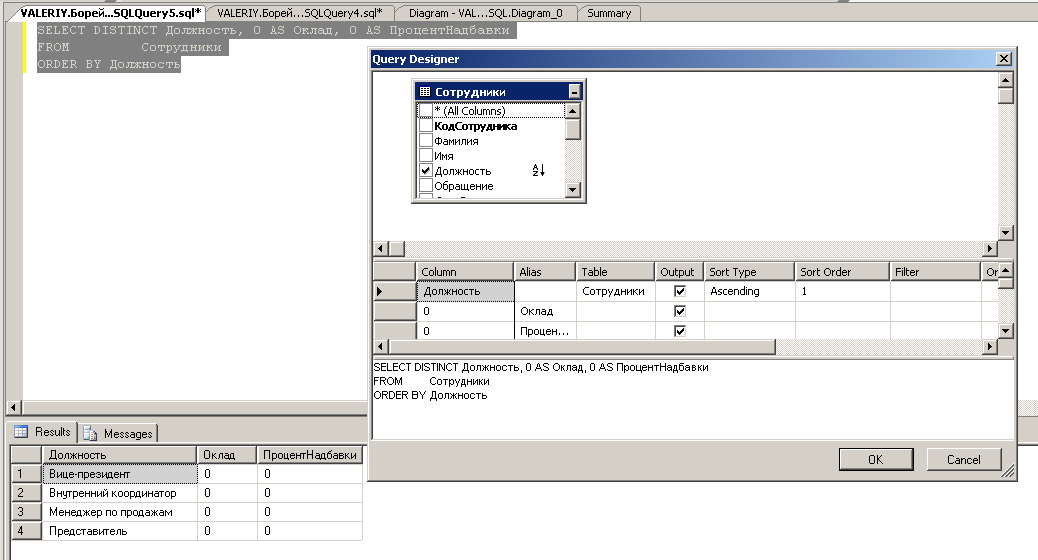
**2)** Создадим таблицу *Оклады* с полями *КодОклада*, *Должность*, *Склад*, *ПроцентДоплаты*, *ДатаНазначения*.



Свяжем таблицы *Оклады* и *Должности*.



Для заполнения данными таблицы Должности следует выбрать без повторений данные колонки Должность таблицы Сотрудники и вставить их в таблицу Должности. Запрос на выборку проще всего сформировать с помощью конструктора Запросов (выполните команду Выполнить-запрос и на панели инструментов появиться вкладка Запрос). В запросе дополнительно созданы колонки Оклад и ПроцентНадбавки с постоянными значениями равными 0.



1. Для добавления результатов запроса в таблицу *Должности* следует изменить в контекстном меню тип запроса на «Вставить результаты…» выбрать в качестве приемника данных таблицу *Должности* и указать соответствие колонок результатов запроса и колонок таблицы-приемника (установить соответствие между значениями столбцов Column и Append конструктора).

INSERT INTO Должности

(Должность, Оклад, ПроцентДоплаты)

SELECT DISTINCT Должность, 0 AS Оклад, 0 AS ПроцентНадбавки

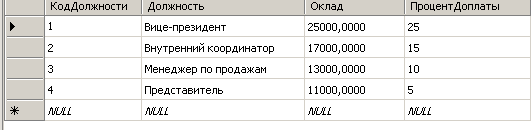
FROM Сотрудники

ORDER BY Должность

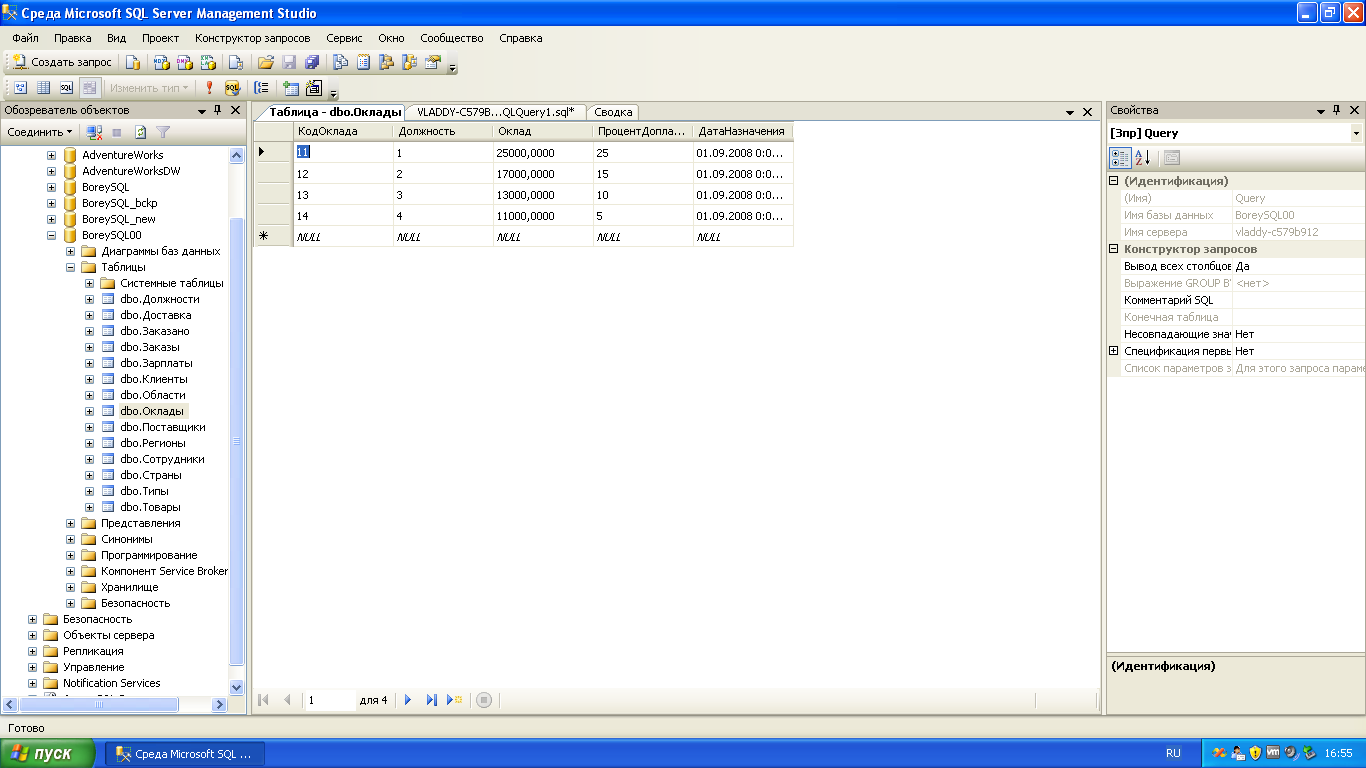
Запустить запрос на выполнение. В случае правильного оформления запроса и успешного выполнения должно быть выведено системное сообщение вида «4 row(s)) affected» (4 строки обработаны).

Проверить результат, открыв таблицу Должности.

Заполнить поля Оклад и ПроцентДоплаты.



Заполним таблицу *Оклады* соответствующими данными таблицы *Должности* при постоянном значении поля *ДатаНазначения* ’01.09.2018’.



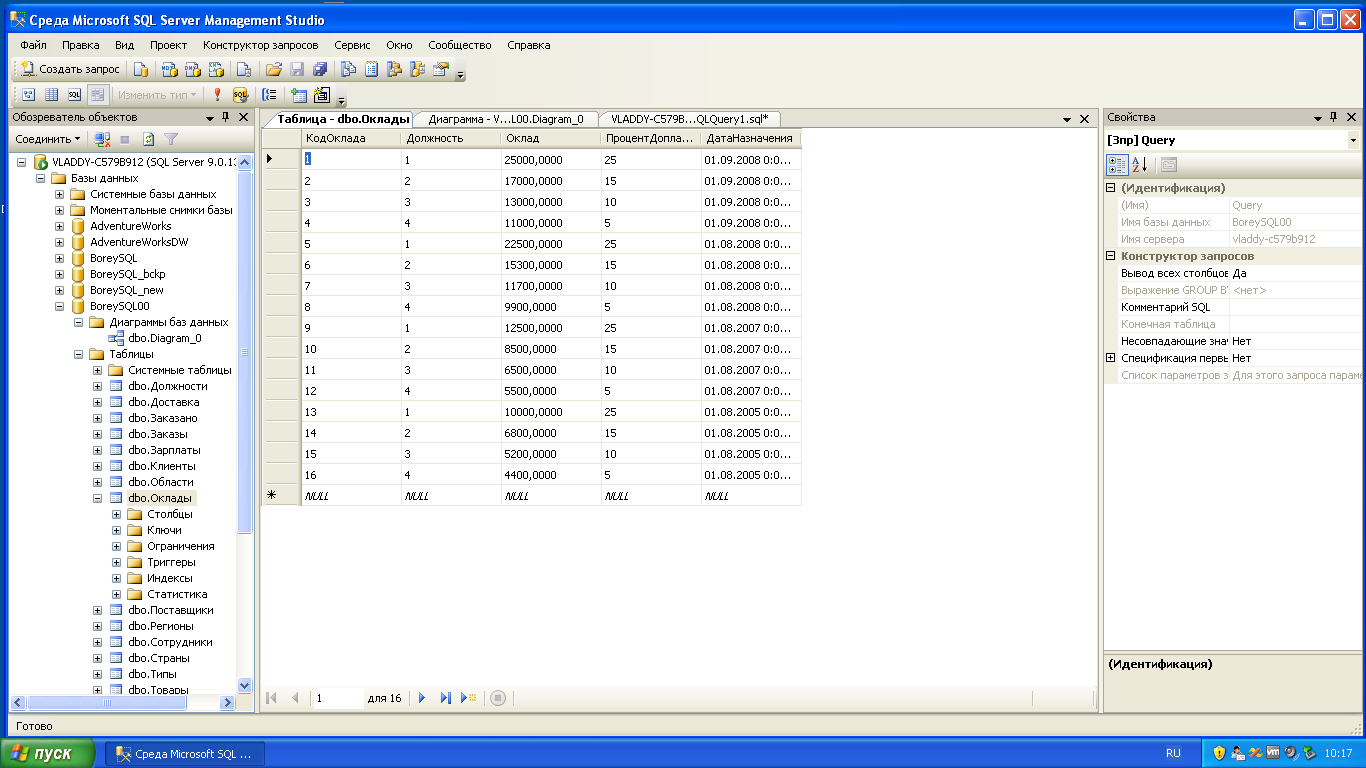
Добавим данные для оклада на 10% меньше для всех должностей, установленных на произвольную дату предыдущего года:

*Код Запроса:*

Insert Into Оклады (Должность, Оклад, ПроцентДоплаты, ДатаНазначения)

SELECT Должность, Оклад\*0.9, ПроцентДоплаты, '01.08.2007' FROM Оклады

Where ДатаНазначения = '01.01.2018'



В свойствах связей на диаграмме базы данных установим триггеры удаления и обновления в значение «Каскадное».

