**Лабораторная работа № 1. Знакомство с инструментами СУБД Microsoft SQL Server**

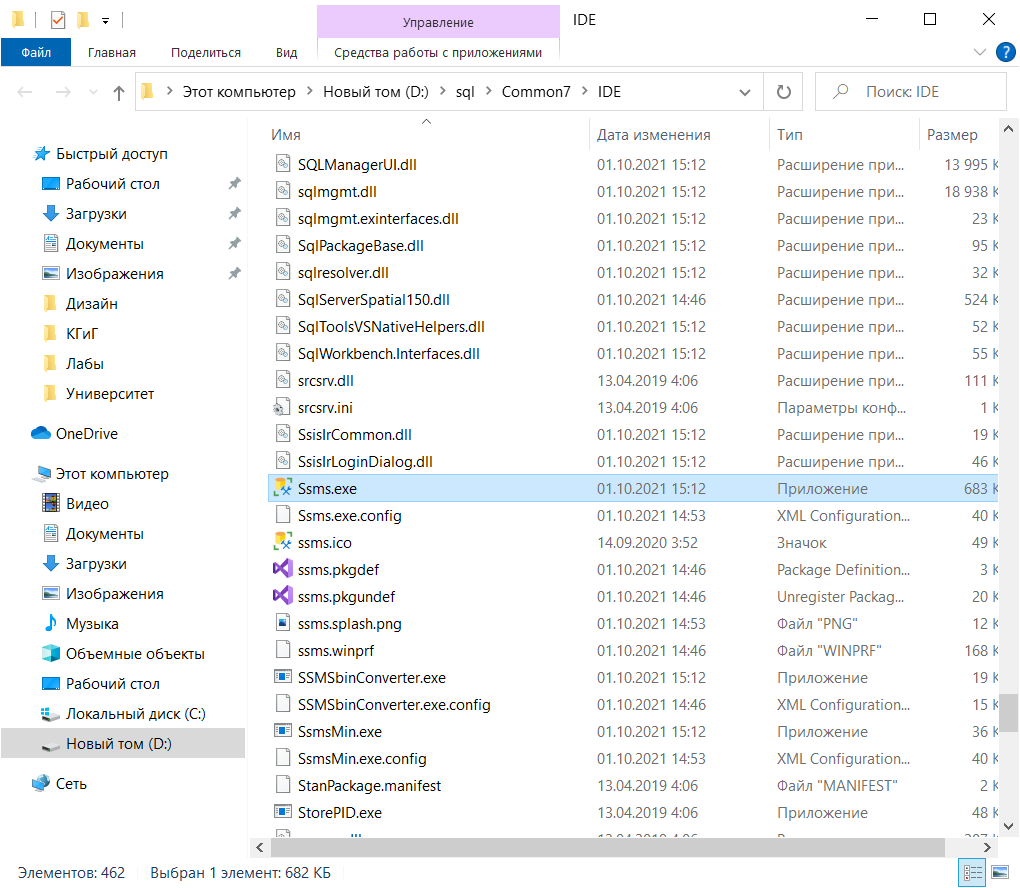
**КУРНОСЕНКО СОФЬЯ**

Создавать базы данных можно с использованием утилиты **SQL Server Management Studio** (**SSMS**), входящей в состав программного обеспечения **СУБД Microsoft SQL Server**,или с помощью языка **Transact**-**SQL**. В данной лабораторной работе рассматривается первый вариант.

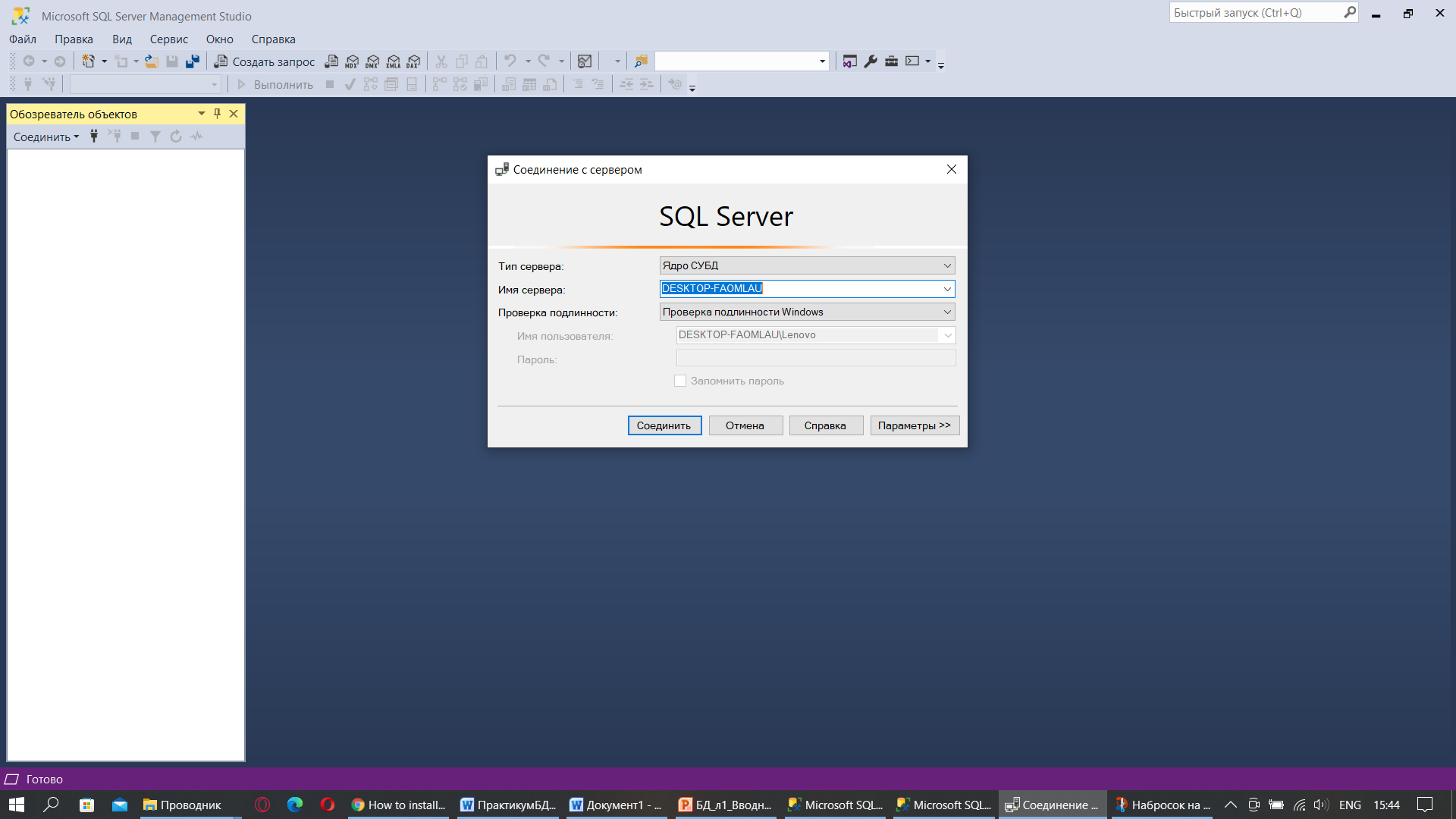
Сначала установила Sql Server, затем SSMS по видео:

<https://www.youtube.com/watch?v=Y_J6zMvV_jM&list=LL&index=5>

Далее открываю:



Начало работы, жму «Соединить», ничего не меняя:



**ЗАДАНИЯ**

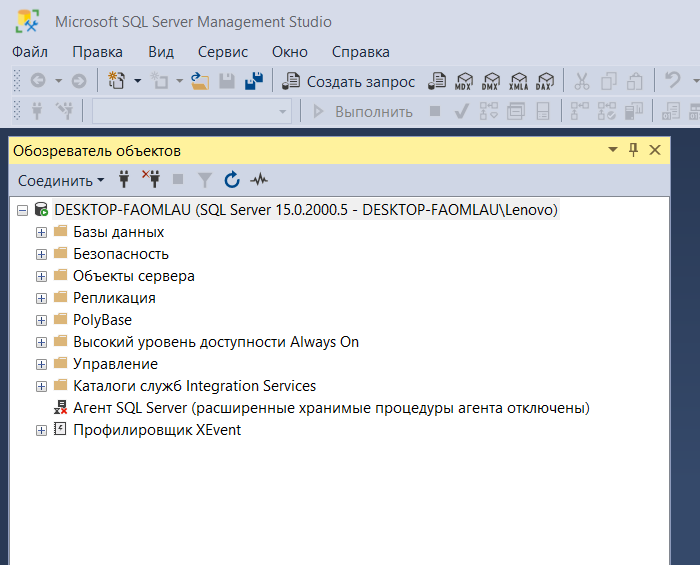
*1. Создать базу данных с помощью утилиты* ***SQL Server Management Studio.*** *Имя базы − фамилия студента и слово* ***ПРОДАЖИ****.*

*Изучить файлы, которые при этом создаются.*

→ ***Теория (методичка)***

Для запуска утилиты **SQL Server Management Studio** (**SSMS**) надо выполнить: **Пуск** / **Программы** / **Microsoft SQL Server** / **SQL Server Management Studio**. Утилитаиспользуется для создания, администрирования и управления базами данных.

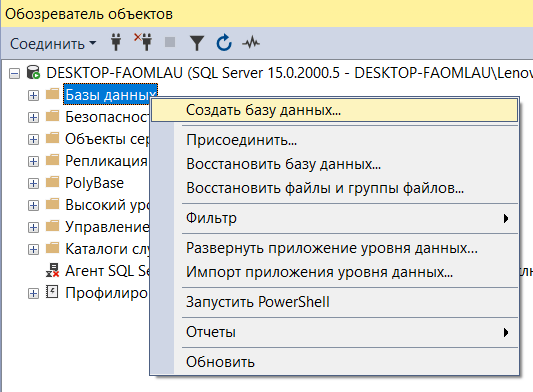
Если панель **Обозреватель объектов** (**Object Explorer**)не видна, то надо выполнить **Вид** / **Обозреватель объектов** (**View / Object Explorer**).

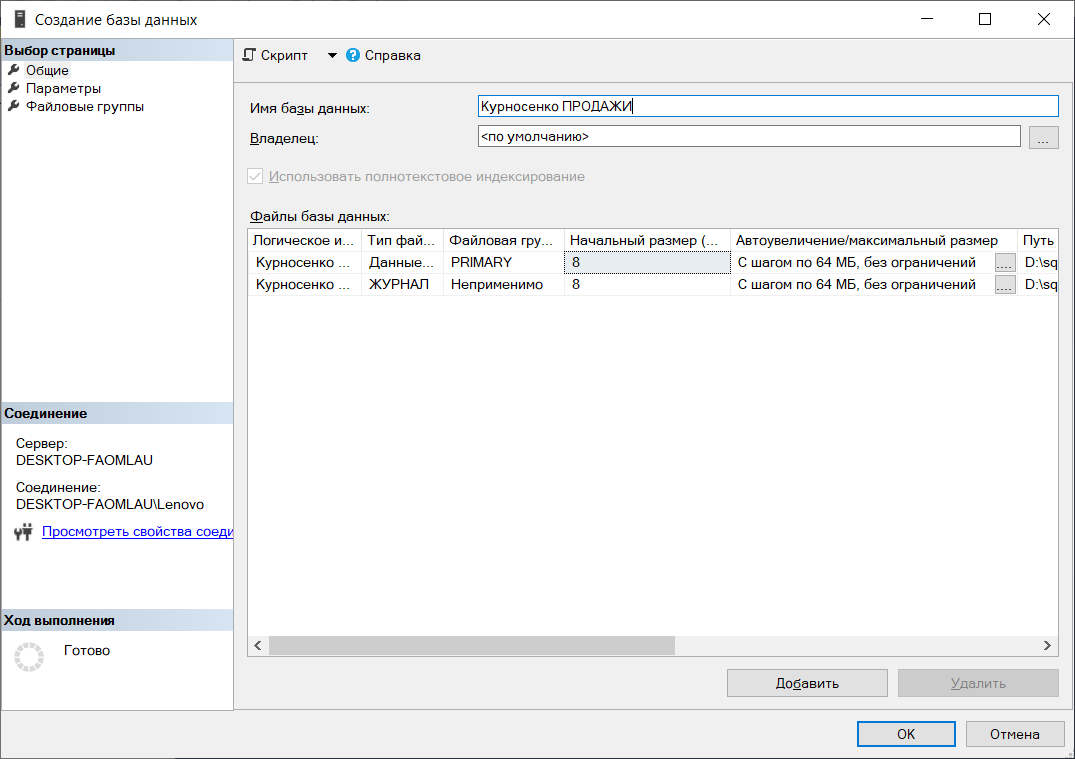


Для подключения к серверу следует выбрать **Соединить** / **Компонент** **Database Engine** / **Соединить** (**Connect** / **Database Engine** / **Connect**).

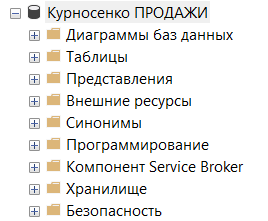
Чтобы создать базу данных надо в контекстном меню пункта **Базы** **данных** (**Databases**) выполнить команду **Создать базу данных** (**New Database**). В диалоговом окне указать имя новой базы данных.

→ ***Выполнение заданий***





При этом создаются файлы:



*2. Создать таблицу ТОВАРЫ, содержащую поля:*

***Наименование*** *(nvarchar(20))****,***

***Цена*** *(real)****,***

***Количество*** *(int).*

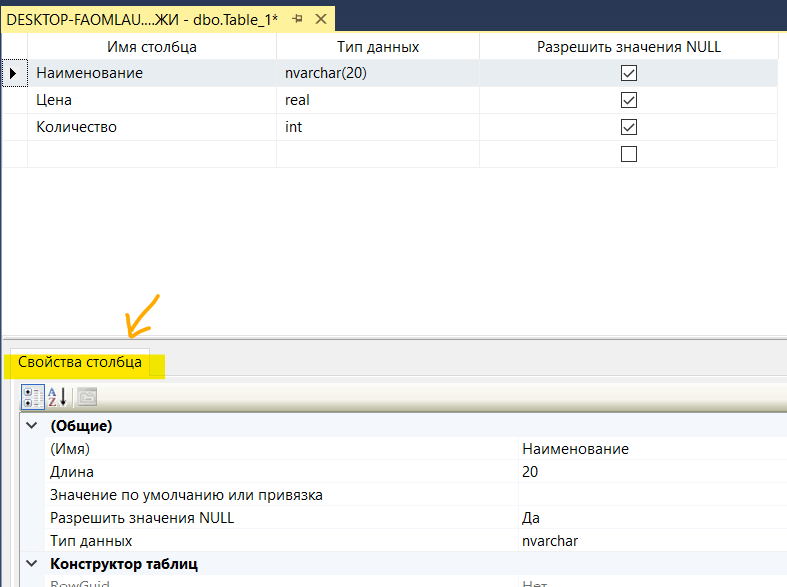
*Сделать первый столбец первичным ключом.*

→ ***Теория (методичка) для заданий 2-4***

Для создания таблицы следует раскрыть папку **Базы данных** (**Databases**) двойным щелчком, раскрыть нужную базу данных, в контекстном меню пункта **Таблицы** (**Tables**) выполнить команду **Создать таблицу** (**New Table**).

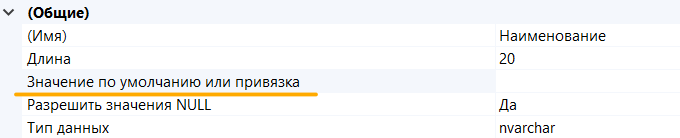
Ввести имена столбцов (атрибутов) в таблицу и их свойства. Все типы данных отображаются, если щелкнуть мышью по знаку стрелки в столбце **Тип данных** (**DataТуре**). Флажок в поле **Разрешить** (**Allow Nulls**)должен быть установлен, если можно вводить в этот столбец значение null.

Для выбранного типа данных можно определять свойства в таблице **Свойства столбца** (**Column Properties**).



Например, типу **nvarchar** (строковый тип *переменной* длины) требуется определить значение для строки **Длина** (**Length**),типу **decimal** требуется указать значения в строках **Точность** (**Precision**)и **Масштаб** (**Scale**).Типу **int** можно не задавать никаких значений в этих строках.

Если для столбца существует значение по умолчанию, то его нужно ввести в строку **Значение по умолчанию или привязка** (**Default Value or Binding**).



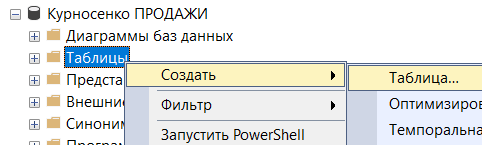
Чтобы сделать столбец первичным ключом надо щелкнуть по столбцу правой кнопкой мыши и выбрать пункт **Задать первичный ключ** (**Set Primary Key**).

После создания таблицы следует закрыть окно с таблицей и в появившемся окне ввести **имя** таблицы для ее сохранения. Нажать кнопку **Обновить** (**Refresh**).

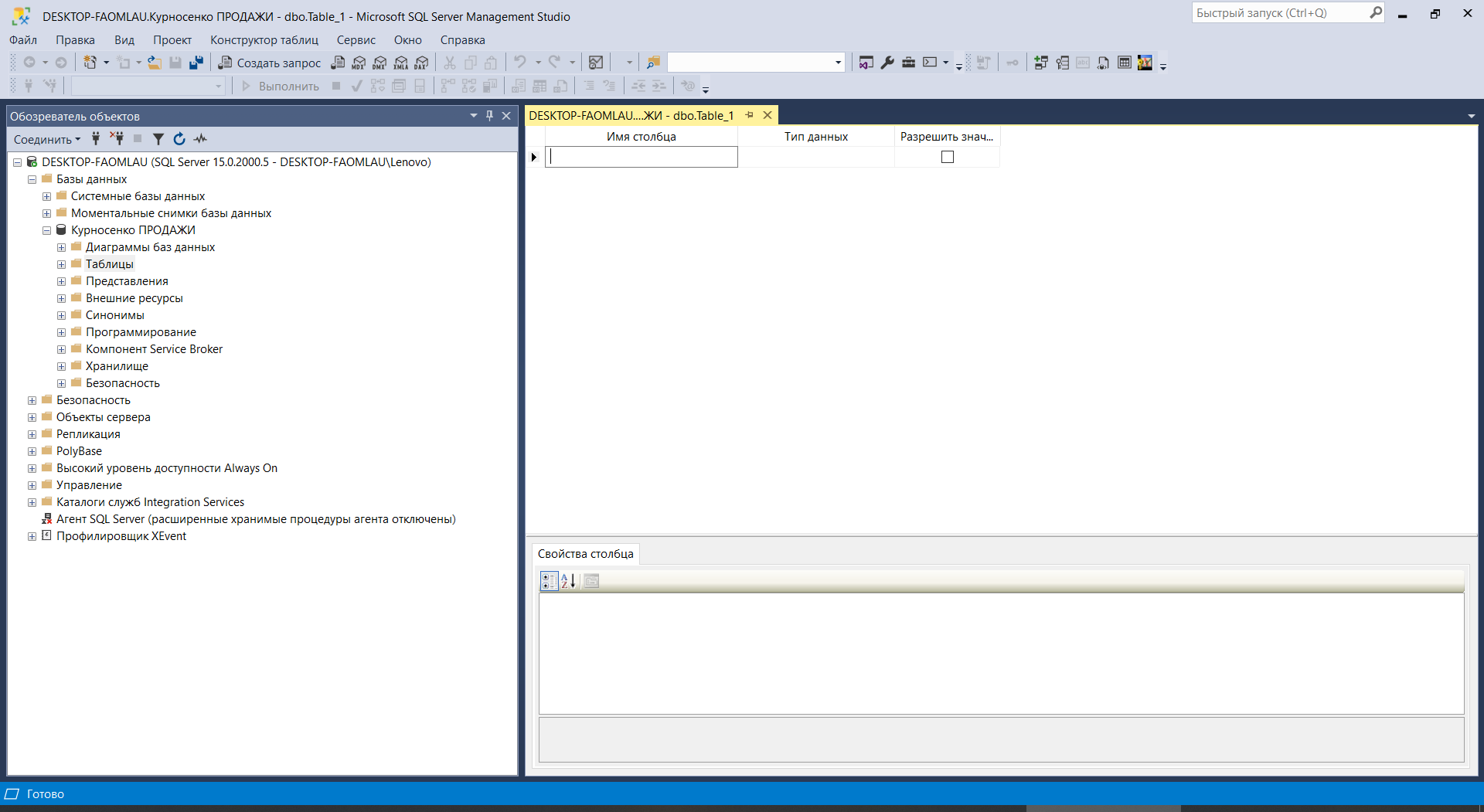
С помощью контекстного меню можно просматривать свойства таблицы, переименовывать ее, удалять и т. п.

→ ***Выполнение заданий***

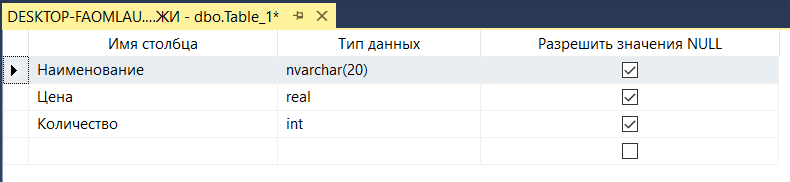
Создание таблицы:



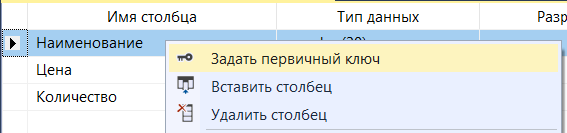
Вот, что мы видим:

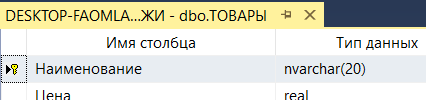


Вводим названия столбцов и типы данных, которые можно будет записывать в поля данного столбца:

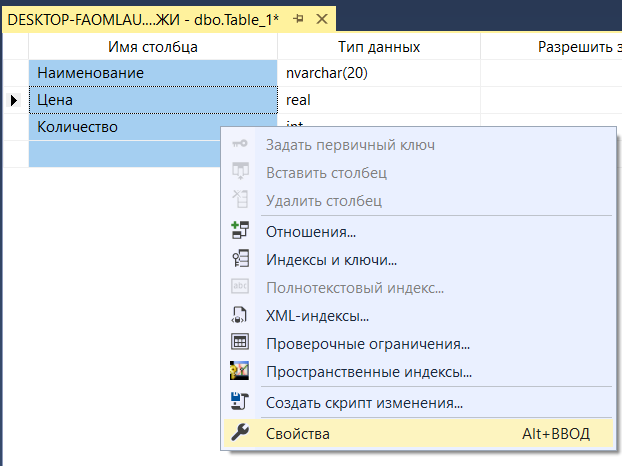


Делаем первый столбец первичным ключом:

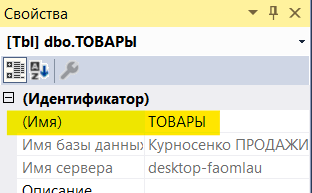
****



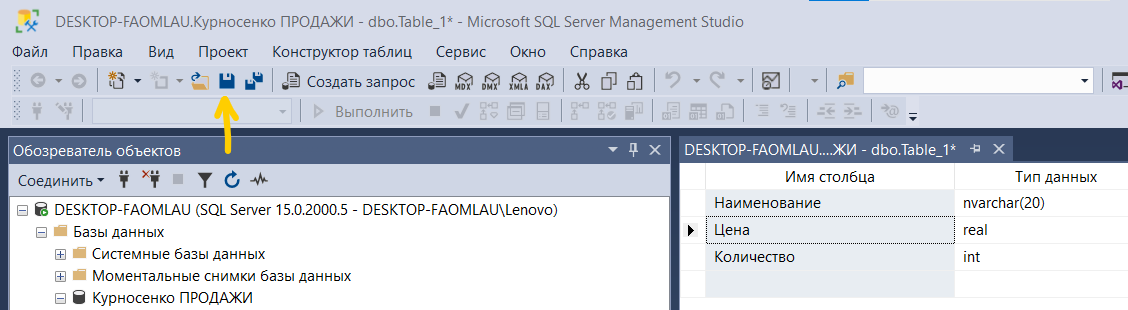
Теперь назовем таблицу. Нажимаю правой кнопкой мыши в любом месте таблицы и выбираю **Свойства**:



В открывшемся справа окне ввожу имя таблицы:

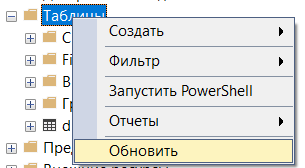


Сохраним таблицу:

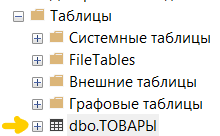


После сохранения значок станет серым: .

В контекстном меню раздела **Таблицы** нажимаем **Обновить**:



Теперь таблицу можно найти в списке:



*3. Создать таблицу ЗАКАЗЧИКИ, содержащую поля:*

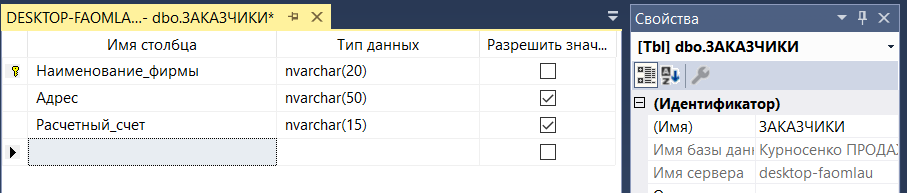
***Наименование\_фирмы*** *(nvarchar(20)),*

***Адрес*** *(nvarchar(50)),*

***Расчетный\_счет*** *(nvarchar(15)).*

*Сделать первый столбец первичным ключом.*

→ ***Выполнение заданий***



*4. Создать таблицу ЗАКАЗЫ, содержащую поля:*

***Номер\_заказа*** *(nvarchar(10)),*

***Наименование\_товара*** *(nvarchar(20)),*

***Цена\_продажи*** *(real),*

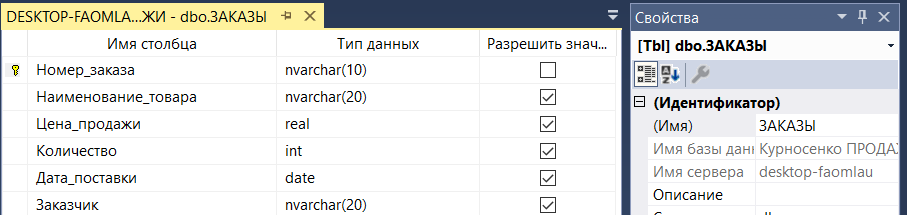
***Количество*** *(int),*

***Дата\_поставки*** *(date),*

***Заказчик*** *(nvarchar(20)).*

*Сделать первый столбец первичным ключом.*

→ ***Выполнение заданий***



*5. Установить связи между таблицами ЗАКАЗЧИКИ и ЗАКАЗЫ по полям* ***Наименование\_фирмы*** *и* ***Заказчик.***

*Установить связи между таблицами ТОВАРЫ и ЗАКАЗЫ по полям* ***Наименование*** *и* ***Наименование\_ товара****.*

*Заполнить таблицы информацией (5-10 строк в каждой).*

→ ***Теория (методичка)***

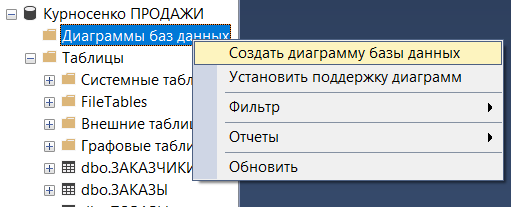
Для создания связей надо в контекстном меню **Диаграммы баз данных** (**Database Diagrams**) выполнить команду **Создать диаграмму базы данных** (**New Database Diagrams**), в появившемся окне выделить таблицу ТОВАРЫ и нажать **Добавить** (**Add**), аналогично выбрать остальные таблицы. Сначала надо определить, какая таблица главная, и какая – подчиненная. Затем тянуть мышкой поле (**Наименование)** от главной таблицы (ТОВАРЫ) к нужному полю (**Наименование\_товара**) в подчиненной таблице (ЗАКАЗЫ).

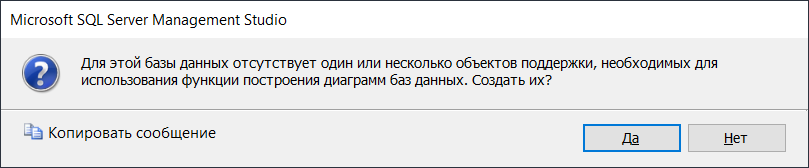
Первыми заполняются главные таблицы (ТОВАРЫ и ЗАКАЗЧИКИ), затем подчиненные. Чтобы заполнить таблицу надо в ее контекстном меню выполнить команду **Изменить первые 200 строк** (**Edit Top 200 Rows**), в появившемся окне ввести данные.

Чтобы увидеть созданные таблицы в списке, надо нажать **Обновить**.

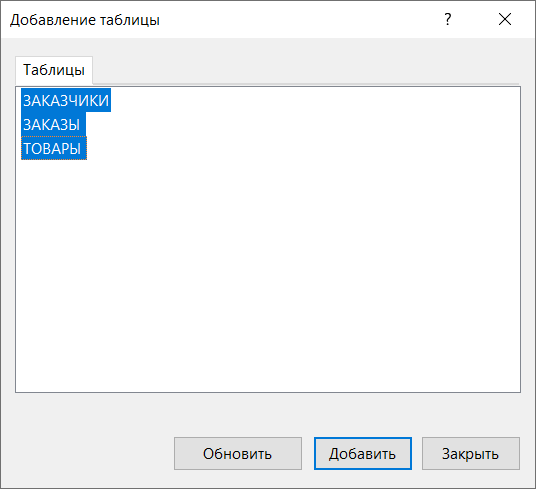
→ ***Выполнение заданий***

Создание отношений:

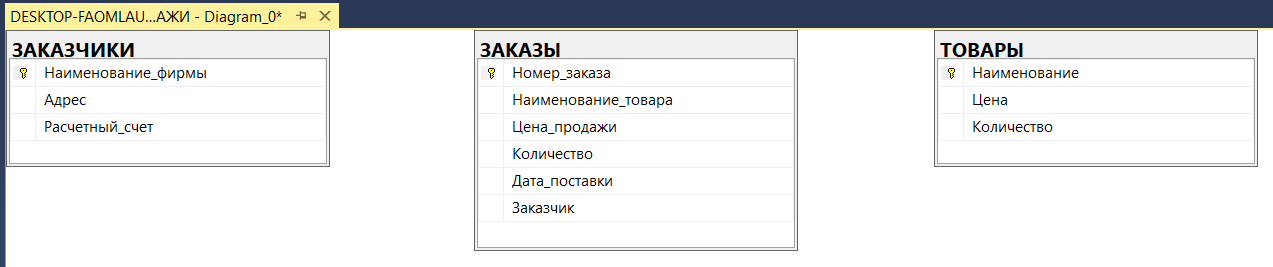
******



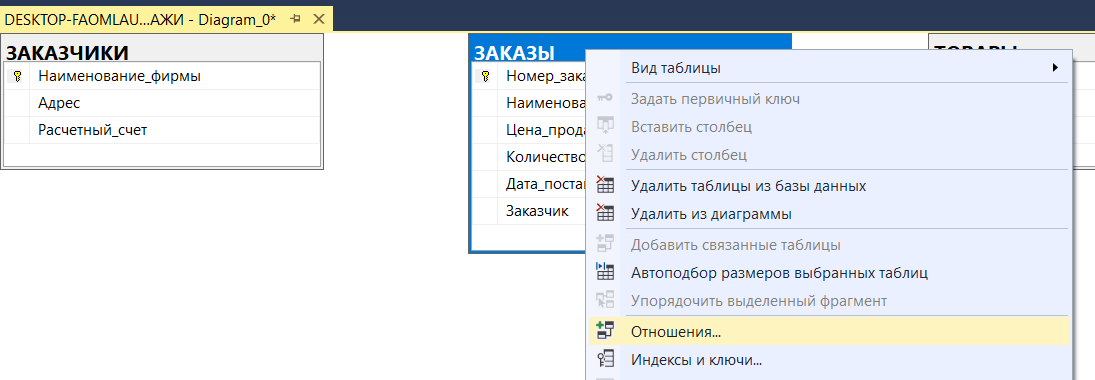
Нажимаем по первой таблице, зажимаем Shift, выбираем последнюю таблицу, таким образом, выделяются таблицы с первой по последнюю, и нажимаем Добавить:



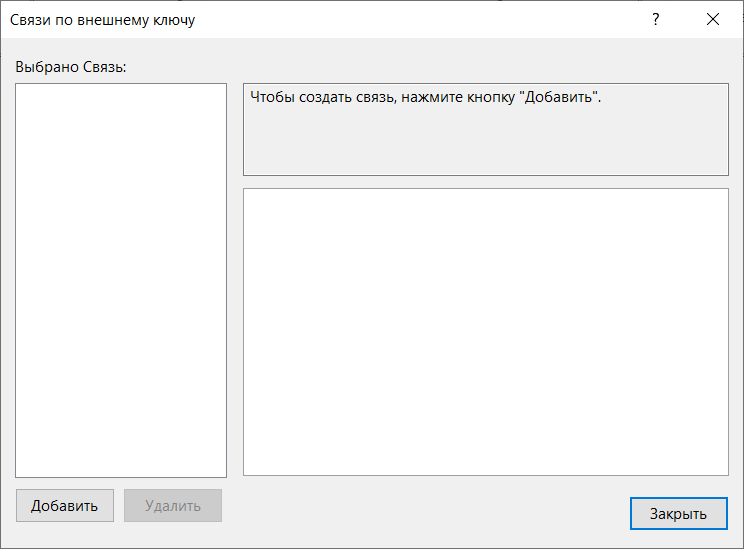
Вот что будет создано:



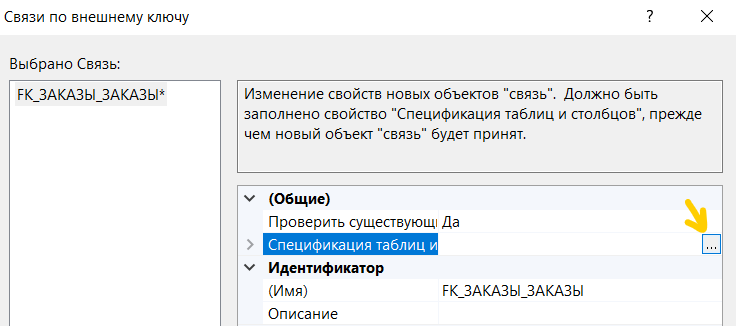
Устанавливаем связи:



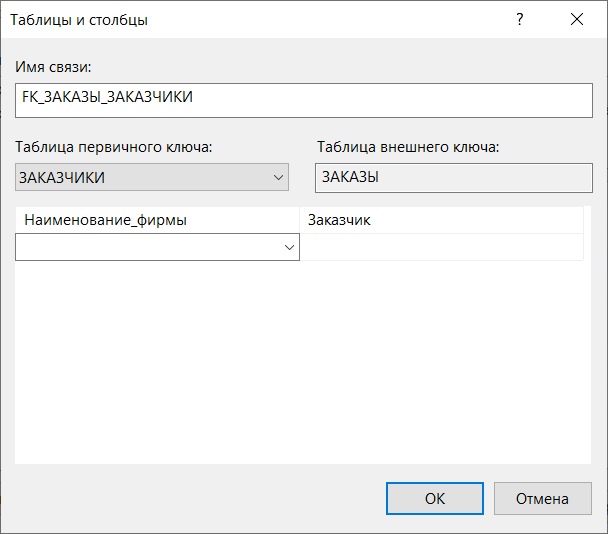
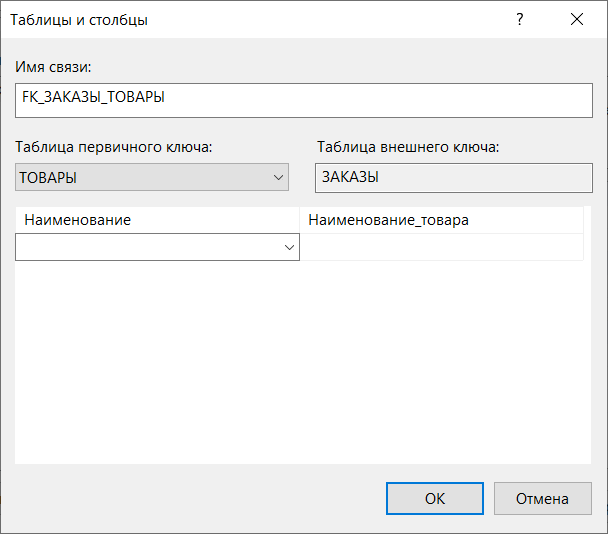
Добавить:



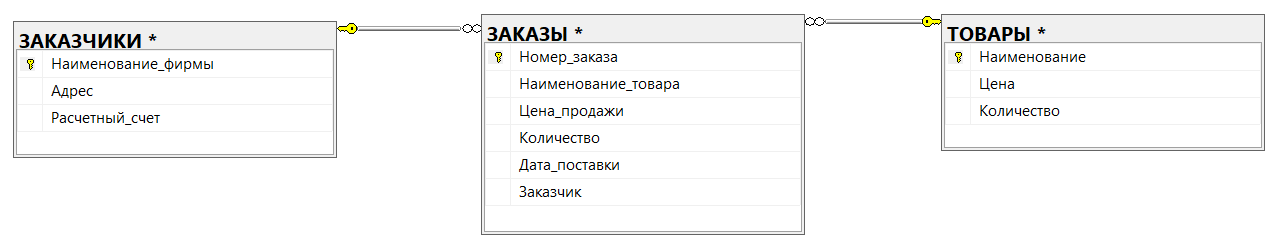
Нажимаем троеточие в пункте **Спецификация таблиц и столбцов**:



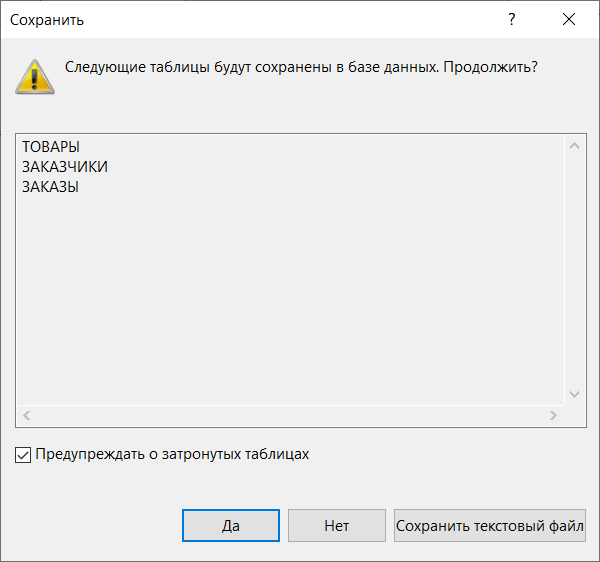
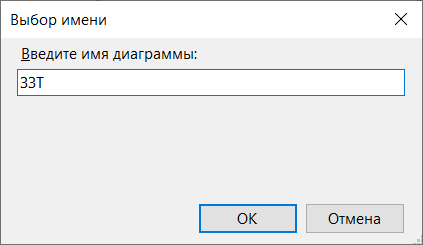
Устанавливаем поля согласно заданию:



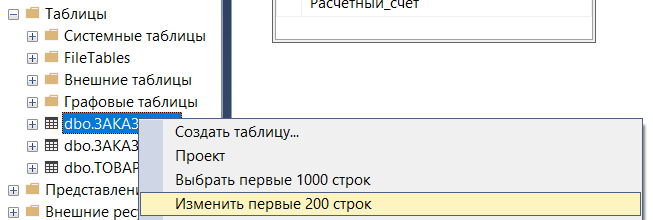
В итоге получаем такой вид:



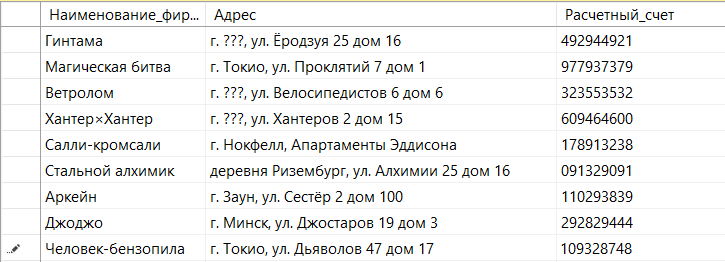
Сохраним диаграмму :



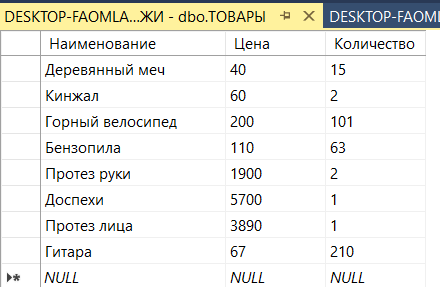
Теперь заполним таблицы:



Заказчики:



Товары:



Заказы:



*6. Сформировать следующие запросы и проанализировать результаты:*

*– определить товары, поставки которых должны осуществиться после некоторой даты;*

*– найти товары, цена которых находится в некоторых пределах;*

*– определить названия фирм, заказавших конкретный товар;*

*– найти заказы определенной фирмы по ее названию, отсортировать их по датам поставки.*

*Сохранить запросы в sql-скрипте.*

→ ***Теория (методичка)***

Для формирования запроса надо щелкнуть по кнопке **Создать запрос** на панели инструментов. В контекстном меню появившегося окна следует выполнить команду **Создать запрос в редакторе** (**Design Query in Editor**), в окне **Добавление таблицы** (**Add Table**) выбрать нужные таблицы. В таблице или таблицах поставить галочки у полей, которые должны быть включены в запрос. Эти поля будут отображены в нижней части бланка запроса.

В столбце **Фильтр** (**Filter**) бланка запроса следует указать условия отбора для нужных полей. При этом могут использоваться операторы сравнения (=, >, <, >=, <=, ><) и логические операторы **AND**, **OR**, **NOT**, а также оператор **IN**, который проверяет значение поля на равенство любому значению из списка (элементы списка отделяются друг от друга запятыми, список заключается в круглые скобки).

Если условие налагается на несколько полей, и они связаны логическим оператором **И**, то условия вводятся в одном столбце **Фильтр** у нужных полей, если логическим оператором **ИЛИ** – то в разных столбцах **Or**.

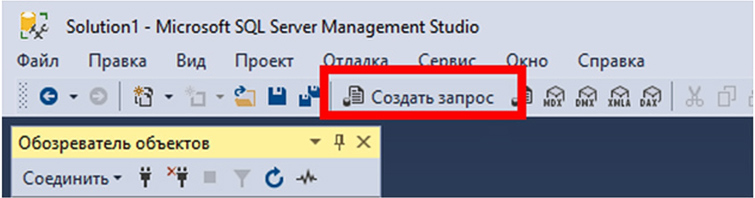
Чтобы отсортировать результат надо в столбце **Порядок сортировки** (**Sort Type**) для соответствующего поля выбрать тип сортировки (по возрастанию или по убыванию). После формирования запроса нажать **Ок**.

Чтобы выполнить запрос надо щелкнуть по кнопке **Выполнить** (**Execute**) на панели инструментов. Результаты появятся на экране.

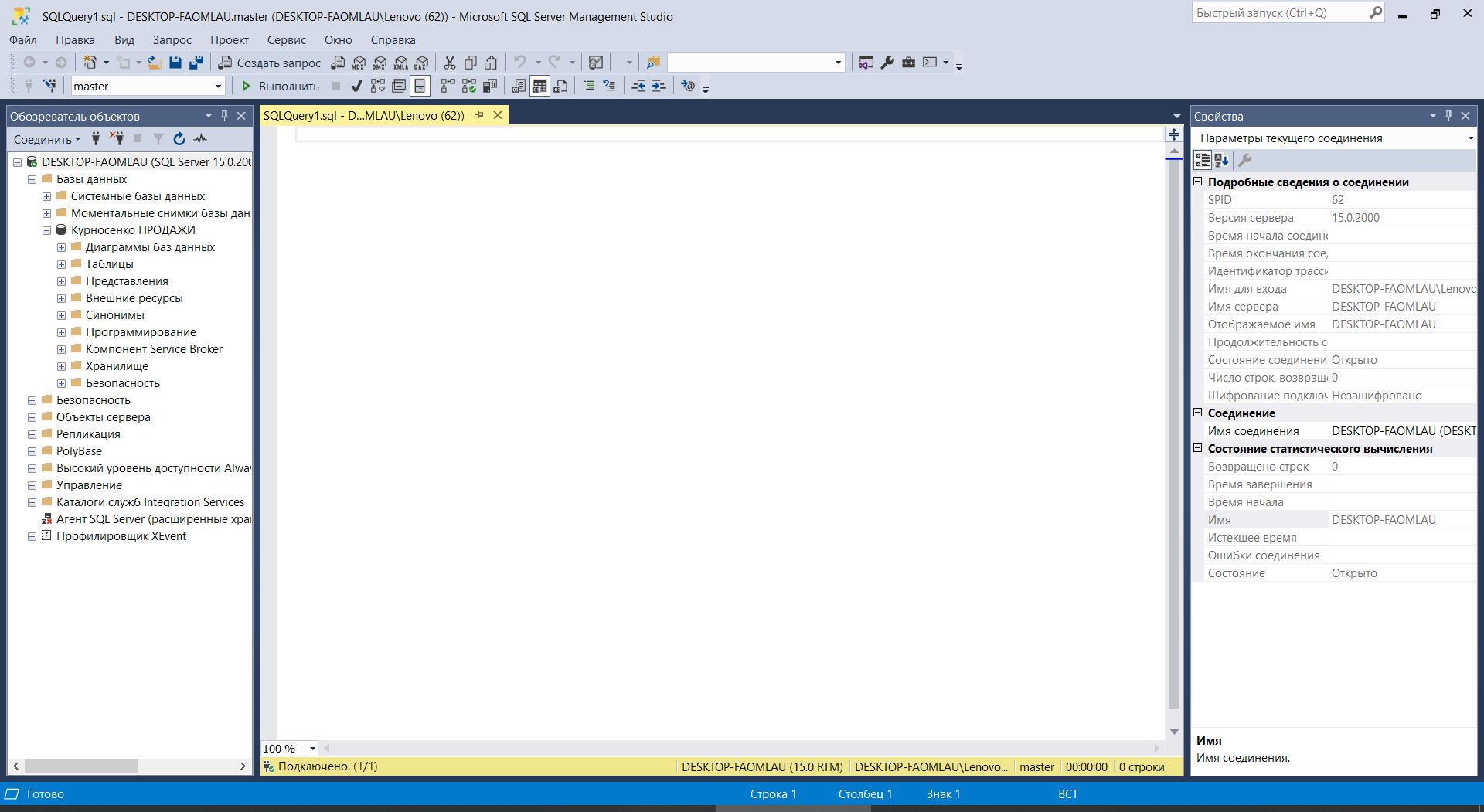
Для *сохранения* запроса можно использовать кнопку **Save** на панели инструментов.

→ ***Выполнение заданий***

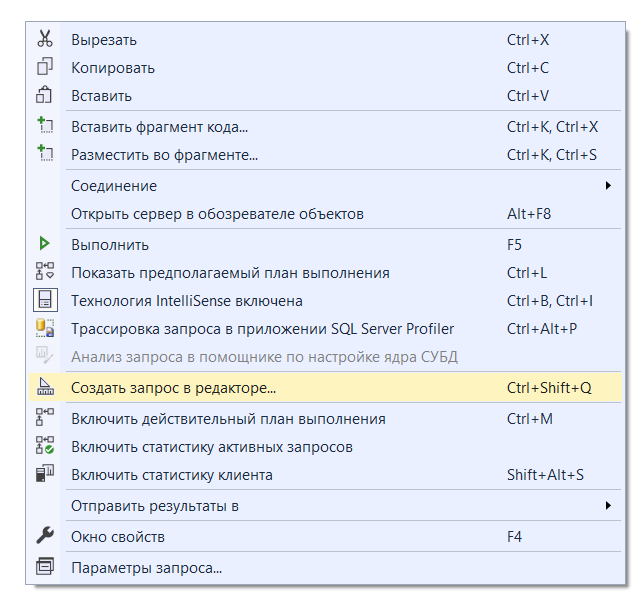
Воспользовавшись Management Studio, для начала создадим новый редактор скриптов. Чтобы это сделать, на стандартной панели инструментов выберем «Создать запрос».



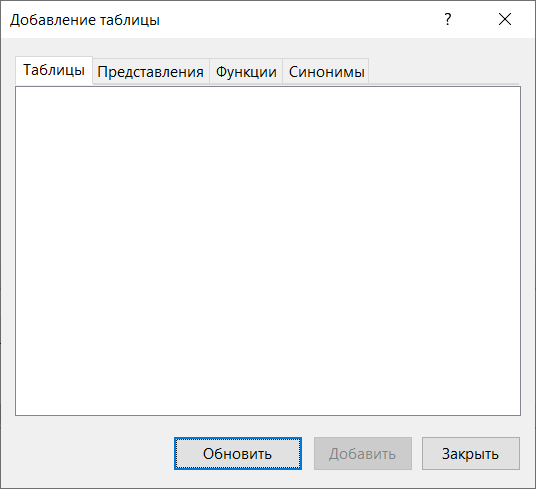
Нажимая кнопку «Создать запрос» в Management Studio, мы открываем тестовый редактор, используя который можно производить написание SQL запросов, сохранять их и запускать.



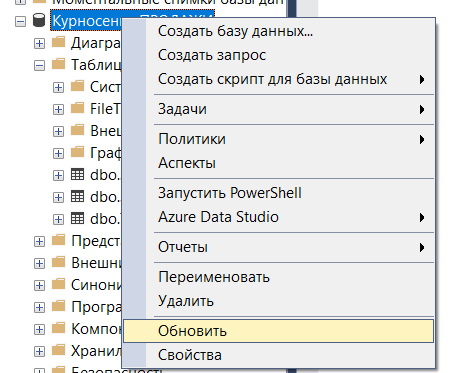
Правой кнопкой мыши **Создать з апрос в редакторе**:



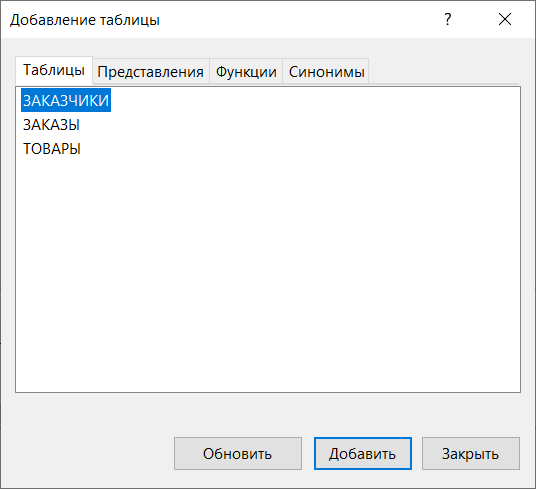
Появилось окно Добавление таблицы, но даже после нажатия *Обновить* таблицы не отобразились.



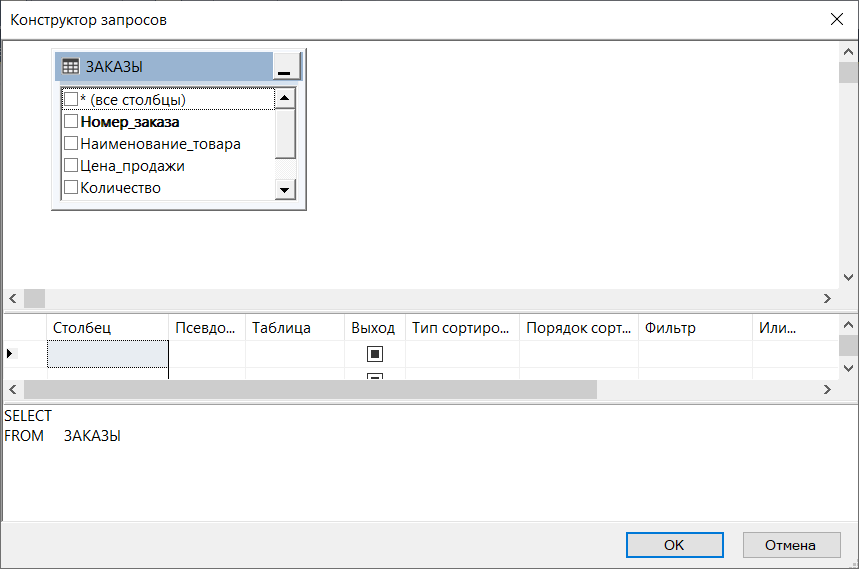
Я обновила БД:

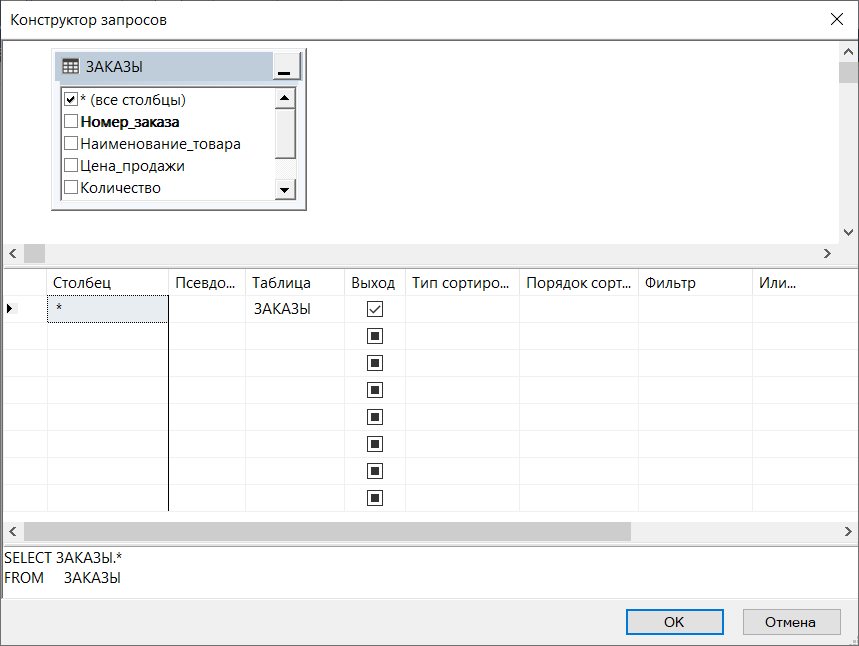


Создала запрос снова. В этот раз всё заработало, как надо:

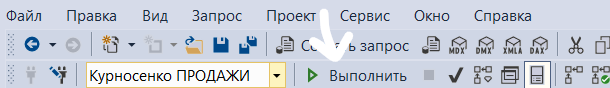


Добавила таблицу «ЗАКАЗЫ»:

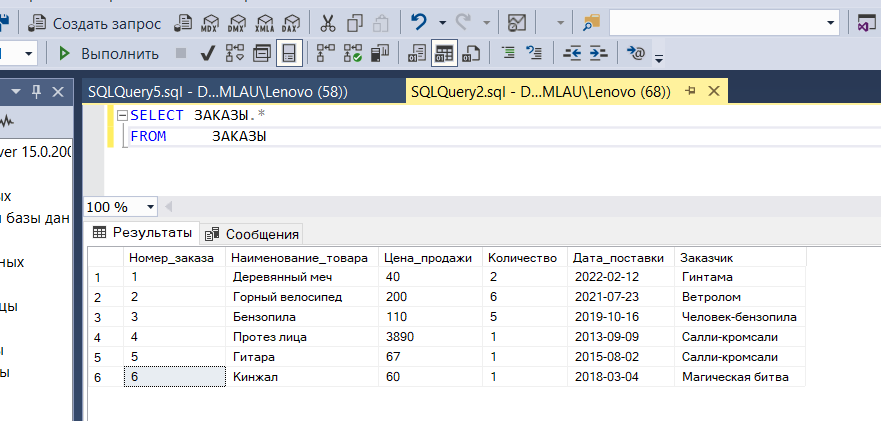




Для выполнения запроса жмем **Выполнить** в панели инструментов:



Результат:



Это было для общего ознакомления, а теперь составим запросы согласно заданиям.

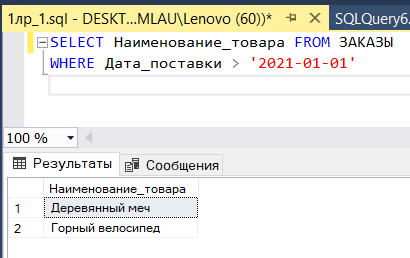
Инструкции Transact-SQL (т.е. запросы) для исполнения вводятся в верхнюю панель, а результаты обработки системой этих запросов отображаются в нижней панели.

– определить товары, поставки которых должны осуществиться после некоторой даты;

Обязательно используем одинарные кавычки:

SELECT Наименование\_товара FROM ЗАКАЗЫ

WHERE Дата\_поставки > '2021-01-01'

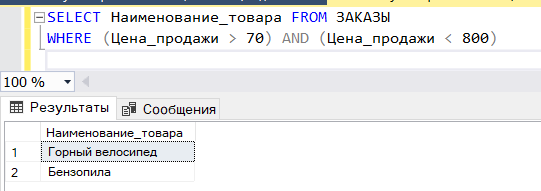


Сохраним запрос .

– найти товары, цена которых находится в некоторых пределах;

SELECT Наименование\_товара FROM ЗАКАЗЫ

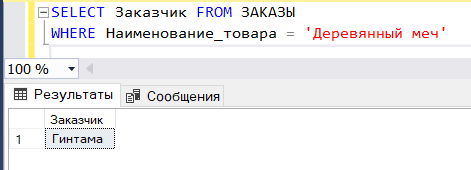
WHERE (Цена\_продажи > 70) AND (Цена\_продажи < 800)



– определить названия фирм, заказавших конкретный товар;

SELECT Заказчик FROM ЗАКАЗЫ

WHERE Наименование\_товара = 'Деревянный меч'

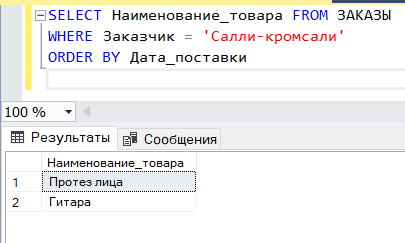


– найти заказы определенной фирмы по ее названию, отсортировать их по датам поставки.

SELECT Наименование\_товара FROM ЗАКАЗЫ

WHERE Заказчик = 'Салли-кромсали'

ORDER BY Дата\_поставки



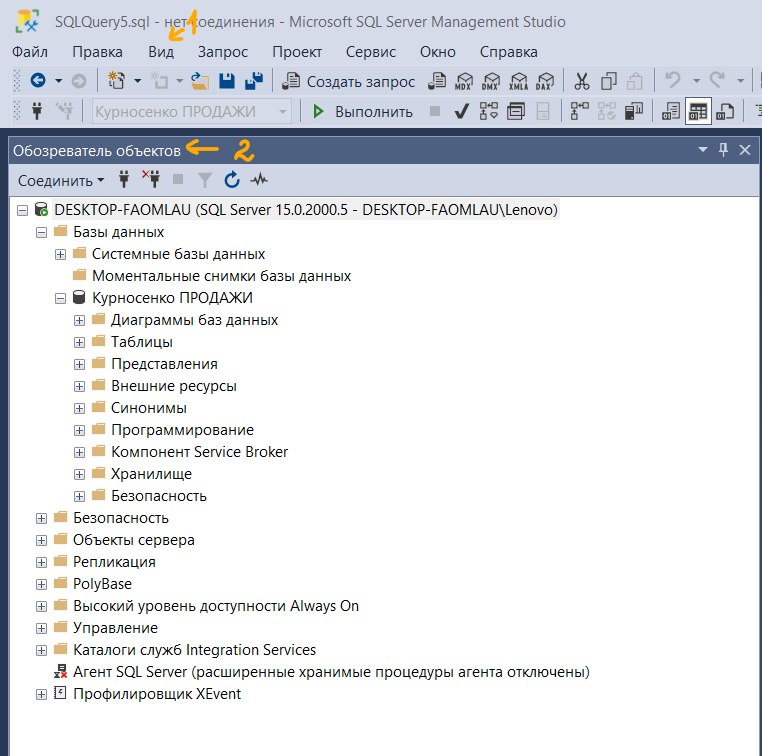
*7. Ознакомиться с sql-скриптами, сгенерировать скрипты для создания таблиц.*

→ ***Теория (методичка)***

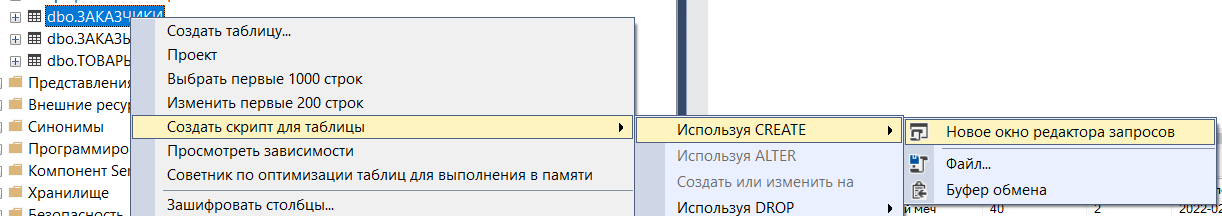
Просмотреть и отредактировать **sql**-скрипты можно с помощью **Обозревателя объектов** (**Object Explorer**).Если нужно, например, просмотреть оператор создания таблицы (**create table**), то следует в контекстном меню этой таблицы выполнить команду **Создать скрипт для таблицы** / **Используя** **CREATE** / **Новое окно редактора запросов** (**Script Table as** / **CREATE to / New Query Editor Window**).

→ ***Выполнение заданий***

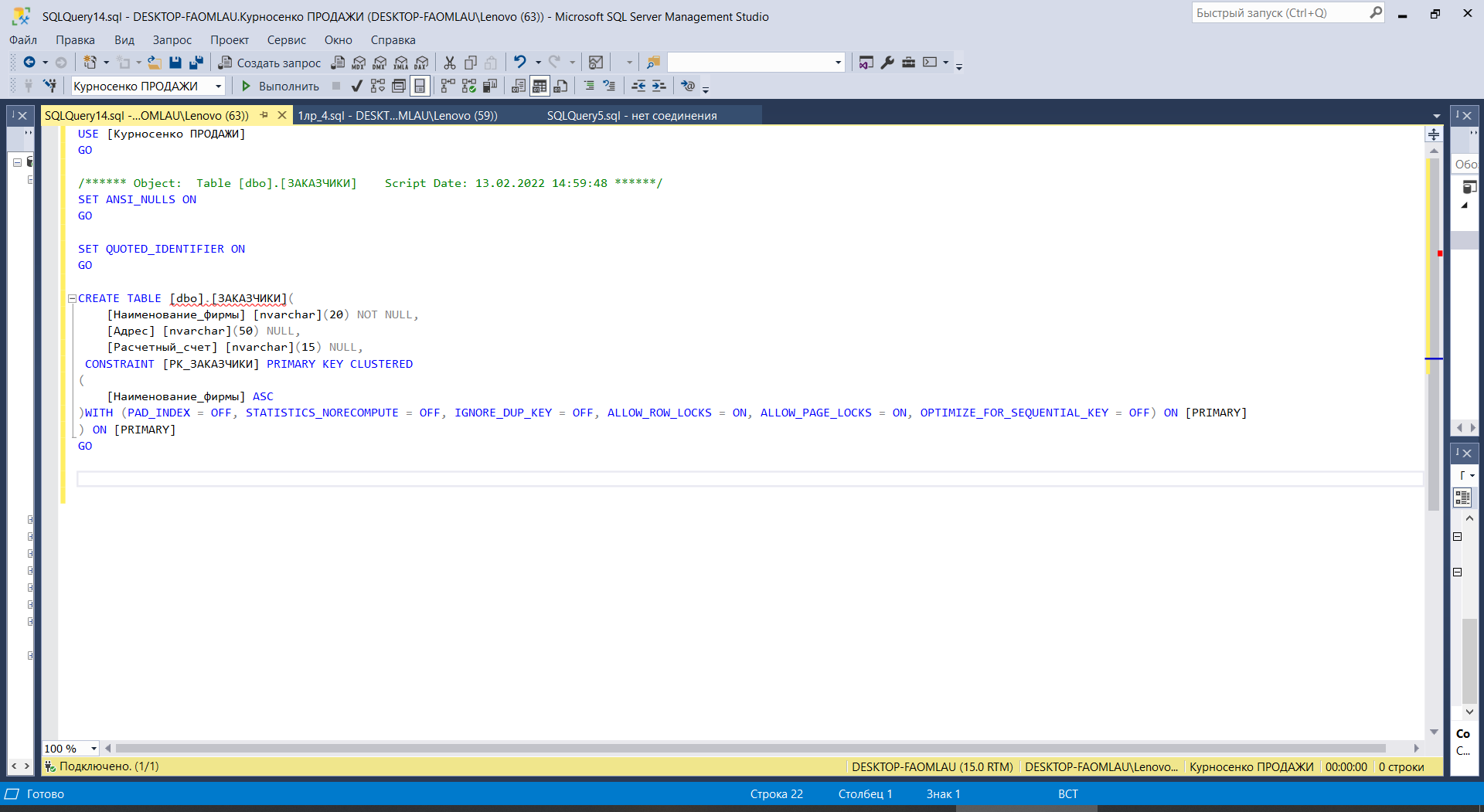
Открываем **Обозреватель объектов**, который можно найти на вкладке **Вид**:

******

Сгенерируем скрипт создания таблицы «ЗАКАЗЧИКИ»:

******

Откроется следующее:



Сохраним скрипт .

→ ***Теория (моя)***

**Что такое SQL скрипт объекта базы данных?**

**SQL скрипт объекта базы данных** – это SQL инструкция, сохраненная в текстовом файле, с помощью которой создается этот объект.

Иными словами, это простой SQL запрос, обычно сохраненный в текстовом файле с расширением .sql. В этом SQL запросе содержатся все необходимые инструкции создания объекта (или объектов), включая инструкции наполнения его данными.

Такой SQL скрипт можно открыть любым текстовым редактором, скопировать текст SQL запроса и выполнить, например, в среде SQL Server Management Studio, таким образом, создав объект базы данных, не разрабатывая соответствующие SQL инструкции самостоятельно.

Также SQL скрипты можно открыть специальными программами, которые умеют работать с SQL скриптами, и выполнять их на базе данных.

**Что могут содержать SQL скрипты?**

SQL скрипты объектов базы данных **могут содержать**:

▪ Инструкции создания таблиц (CREATE);

▪ Заполнение таблиц (инструкции INSERT);

▪ Определение представлений, функций, хранимых процедур, триггеров;

▪ Определение ограничений и индексов;

▪ Определение создания других объектов;

▪ И другие SQL инструкции.

**Для чего могут потребоваться SQL скрипты объектов базы данных?**

SQL скрипты объектов базы данных могут потребоваться администраторам или разработчикам, например, для того, чтобы в случае необходимости иметь возможность быстро восстановить эти объекты.

Или для того, чтобы передать эти SQL скрипты другому администратору, разработчику или заказчику, чтобы он создал подобные объекты на своем экземпляре SQL Server.

Таким образом, такие SQL скрипты необходимы для хранения копий SQL инструкций, с помощью которых создавались объекты базы данных.

**Как создать SQL скрипт объекта базы данных в Microsoft SQL Server?**

Так как SQL скрипт – это обычный текстовый файл, его можно создать вручную, например, когда Вы разрабатываете объект БД, Вы просто сохраняете SQL инструкцию в файл, и добавляете в него по мере необходимости другие SQL инструкции.

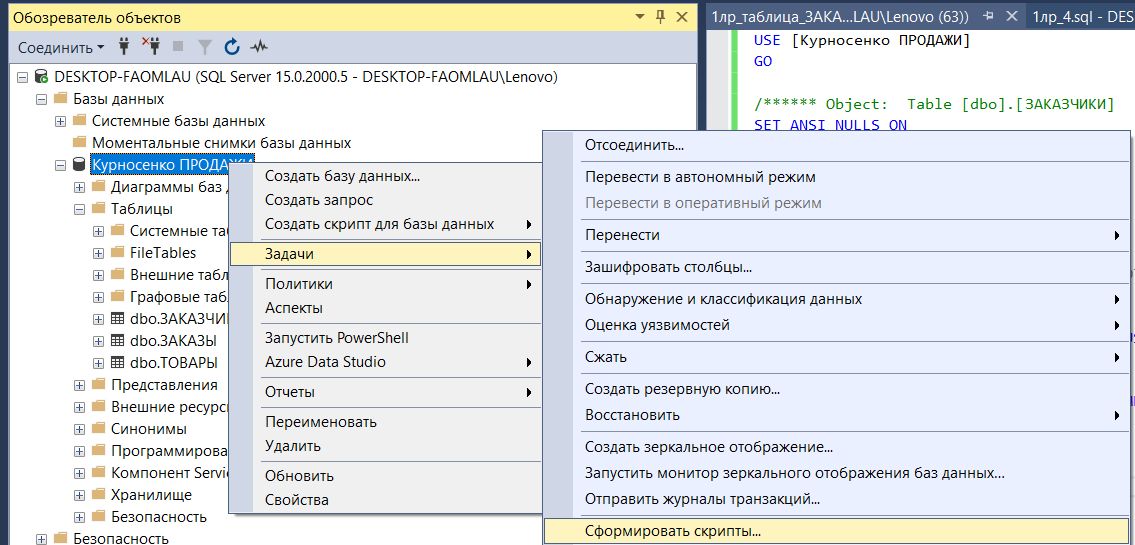
Однако также возможно автоматически сгенерировать SQL скрипты объектов базы данных специальными инструментами, например, в среде SQL Server Management Studio (SSMS).

**Создание SQL скрипта объекта базы данных Microsoft SQL Server**

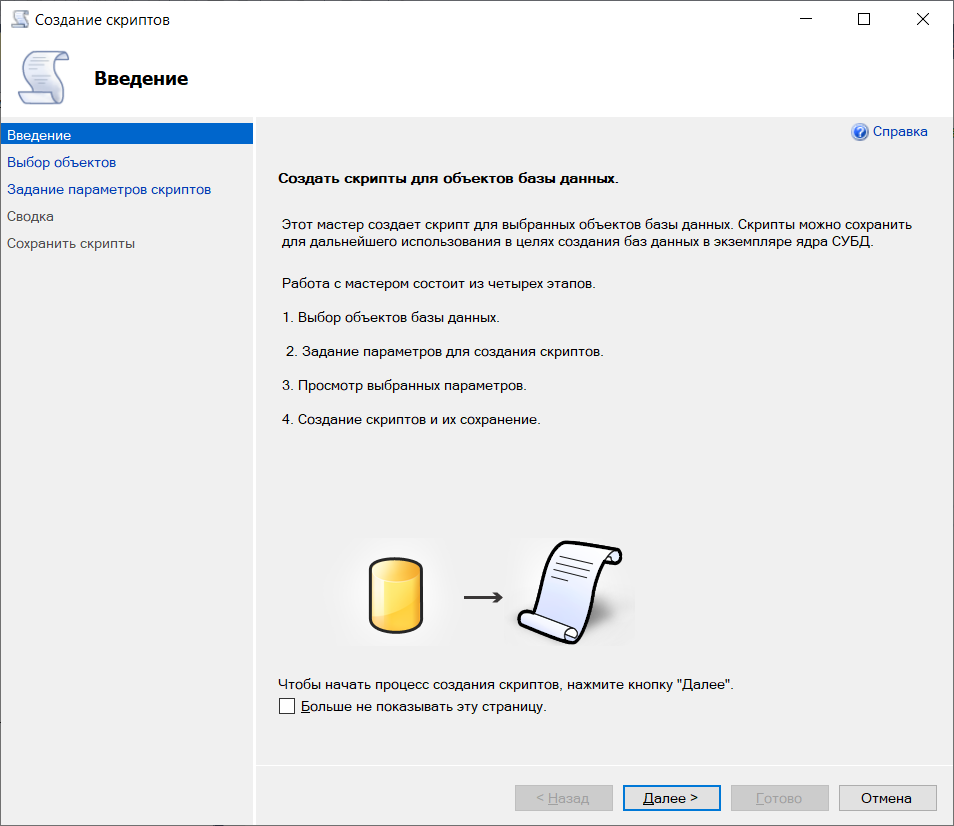
В качестве исходных данных у меня будет база данных «Курносенко ПРОДАЖИ» и таблицы «ЗАКАЗЫ», «ЗАКАЗЧИКИ» и «ТОВАРЫ», SQL скрипт которых мне и нужно создать. При этом мне необходимо, чтобы скрипт включал не только определение инструкции CREATE, но и данные, которые содержит эта таблица.

Итак, давайте начнем.

В обозревателе объектов находим нужную нам базу данных, затем открываем контекстное меню правой кнопкой мыши, ищем пункт «Задачи», и выбираем задачу «Сформировать скрипты».



В итоге запустится мастер создания скриптов. В окне «Введение» можем сразу нажать «Далее».



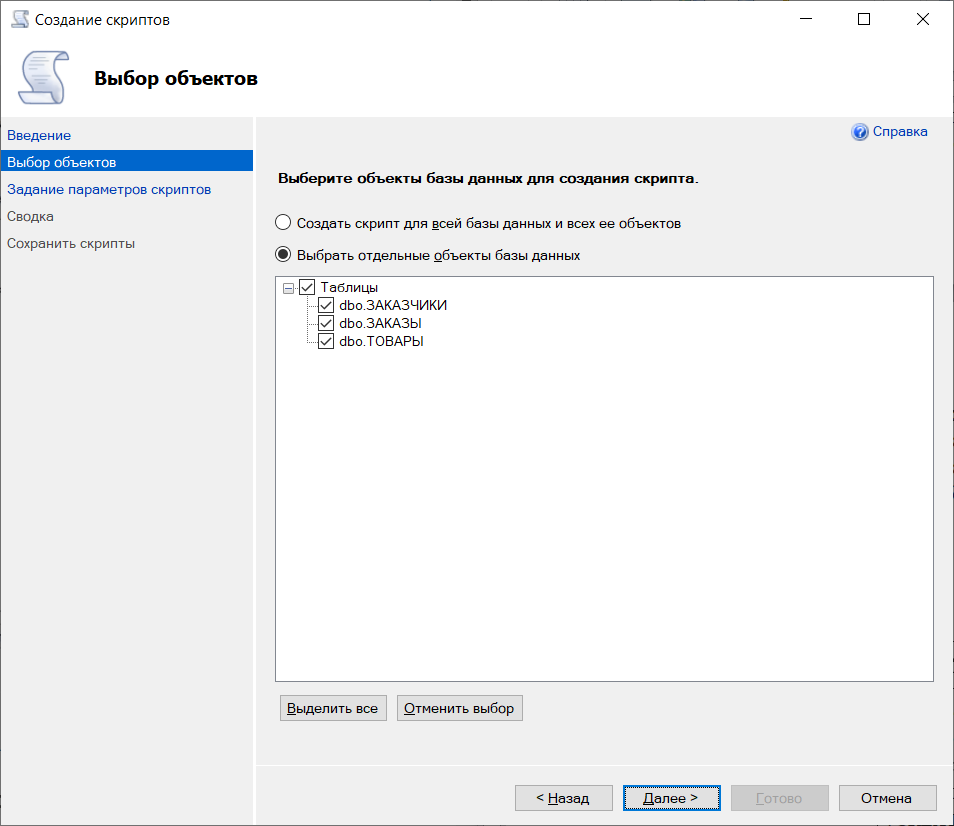
После этого нам нужно выбрать объекты базы данных, которые необходимо включить в SQL скрипт. При этом нам доступно два варианта:

▪ Создать скрипт для всей базы данных и всех ее объектов – этот вариант предполагает, что Вам нужен скрипт создания всех объектов в БД;

▪ Выбрать отдельные объекты базы данных – в данном случае в скрипт включатся SQL инструкции только тех объектов, которые Вы укажете.

Так как мне нужно сохранить только три таблицы, я выбираю второй вариант и отмечаю нужные.

Нажимаю «Далее».



Теперь мы можем указать параметры формирования скрипта, в частности, каким образом мы хотим получить скрипт.

Доступно 4 способа:

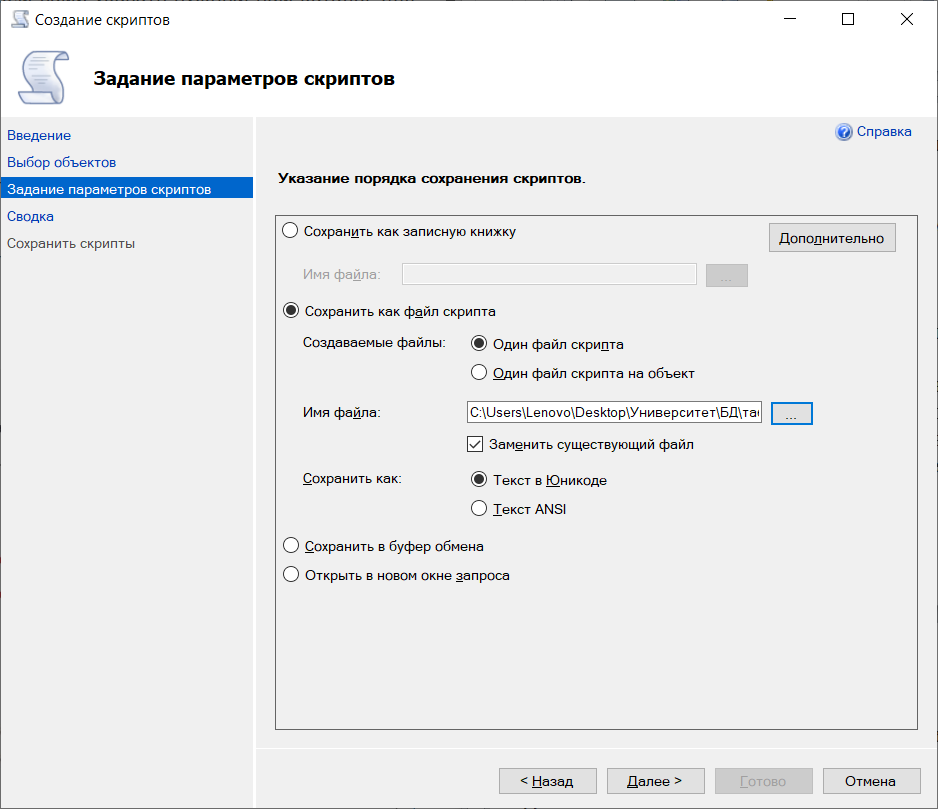
▪ Сохранить как записную книжку

▪ Сохранить в файл – при этом мы можем сами указать нужный нам каталог для сохранения;

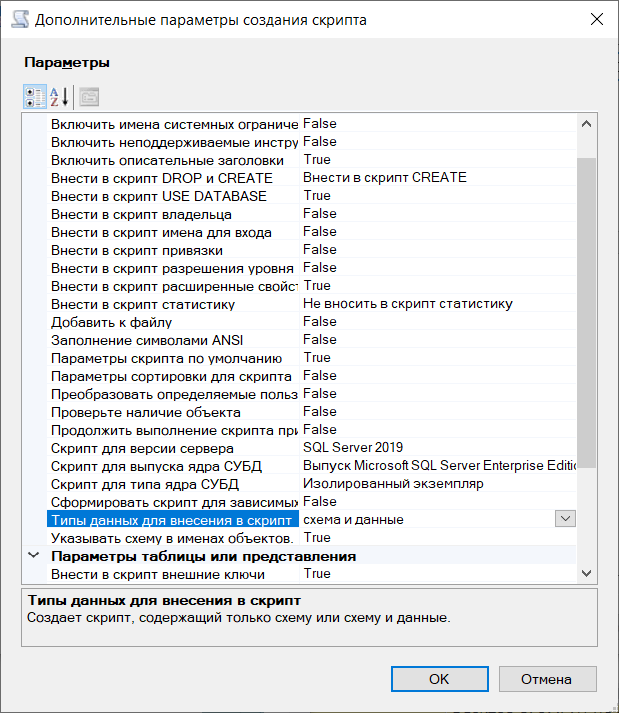
▪ Сохранить в буфер обмена;

▪ Сохранить в новое окно запросов.

Также есть и дополнительные параметры, чтобы их открыть, необходимо нажать на кнопку **Дополнительно**.

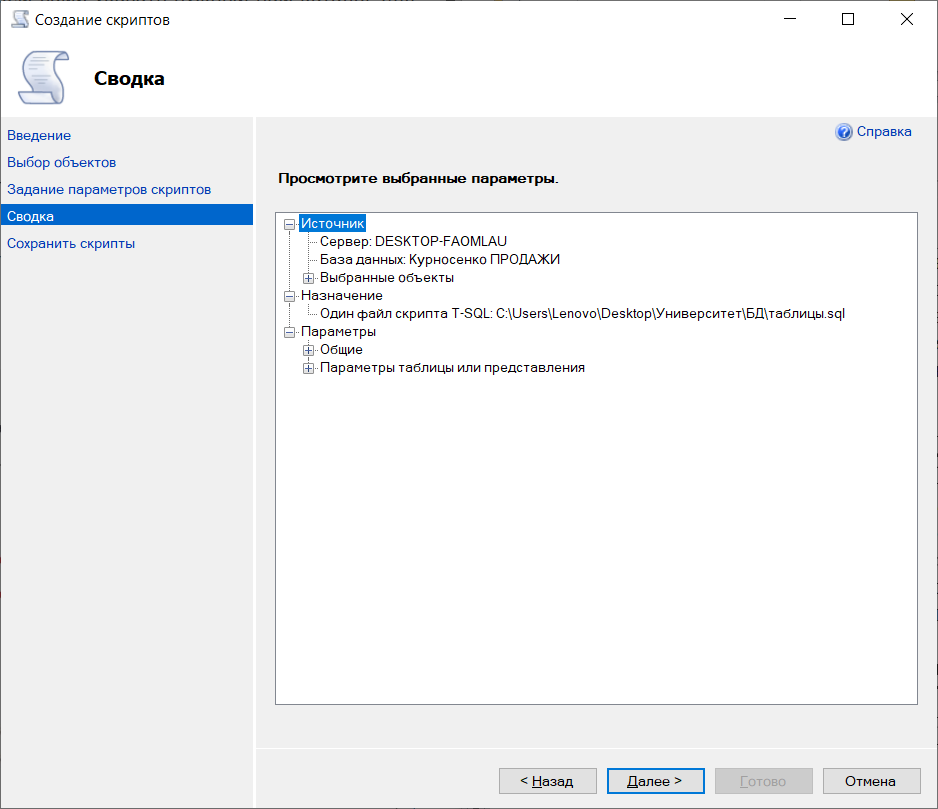


В дополнительных параметрах мы можем более тонко настроить принцип формирования SQL скрипта, например, для того чтобы скрипт включал еще и SQL инструкции создания данных, необходимо в параметрах отметить соответствующий пункт, т.е. указать «Схема и данные».

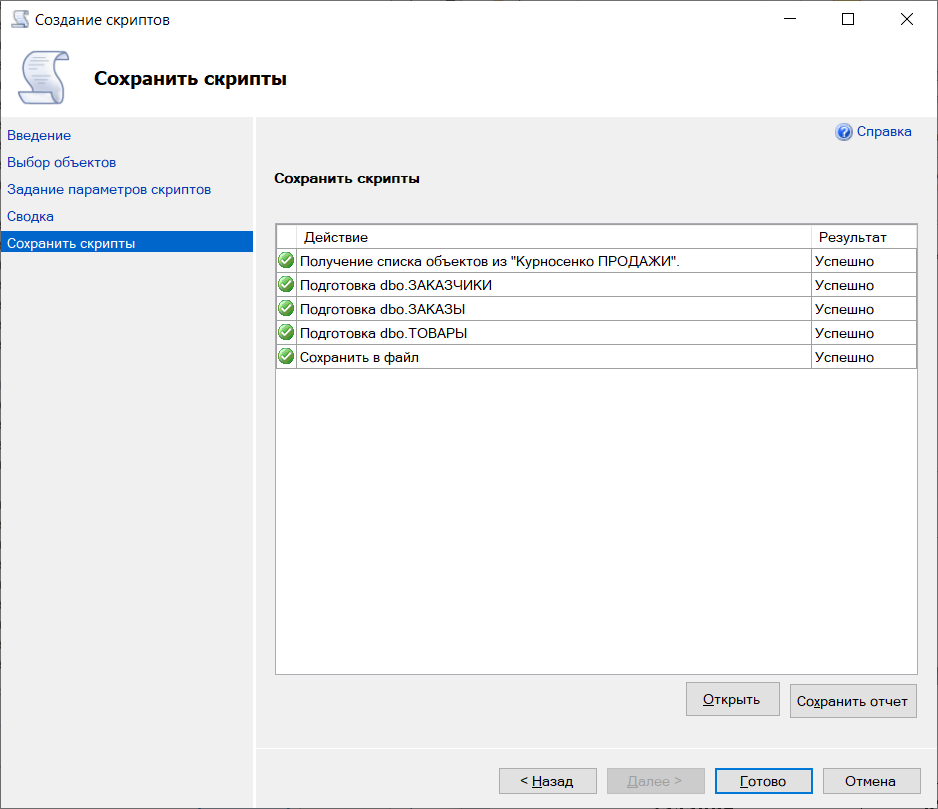


После того как все параметры заданы, нажимаем «ОК».

Убедившись в правильности выбранных параметров нажимаем кнопку «Далее».

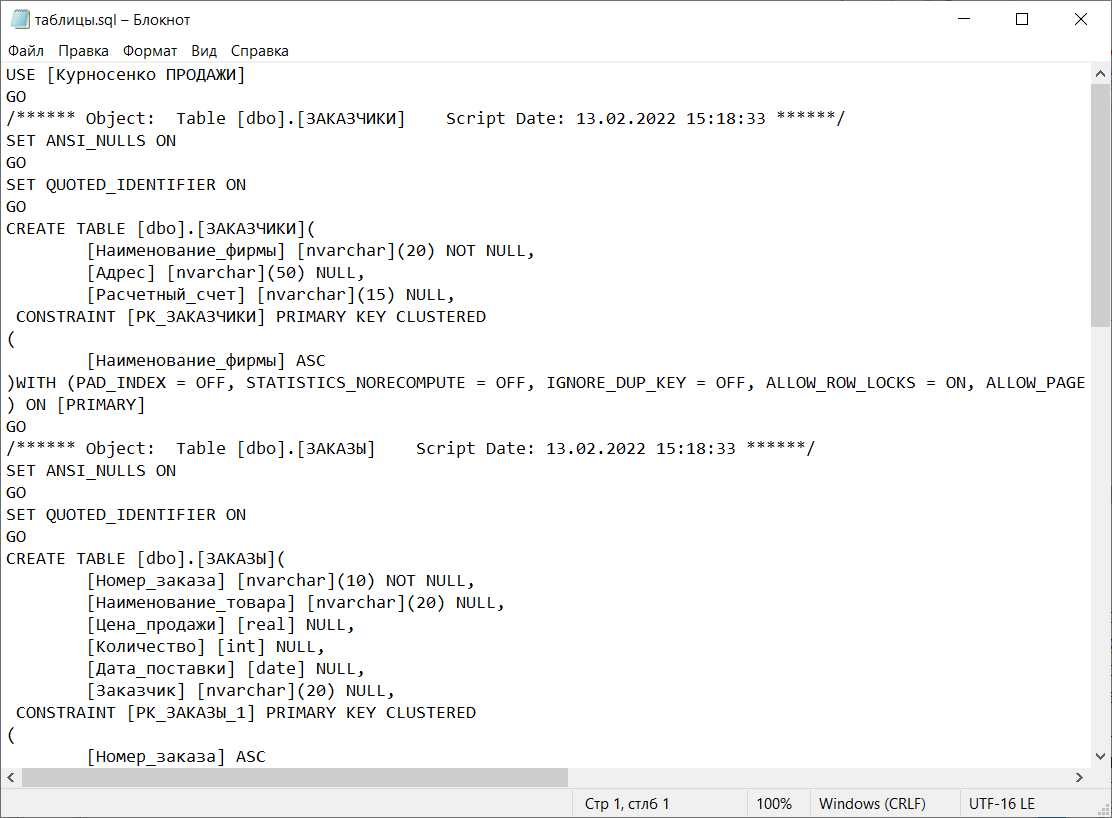


Когда процесс будет завершен, программа сообщит Вам об этом, нажимаем «Готово».



В результате в каталоге, который Вы указали в параметрах формирования скрипта, появится SQL скрипт, а по факту обычный текстовый файл с расширением .sql.

Если его открыть, например, текстовым редактором, то можно посмотреть на все SQL инструкции.

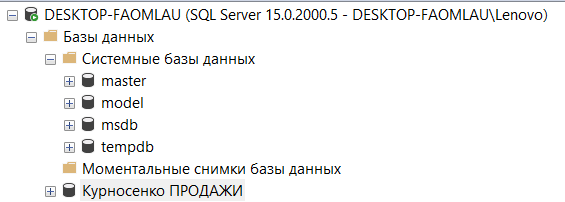


*8. Найти системные БД, просмотреть в них таблицы.*

→ ***Теория (методичка)***

Просмотреть системные базы данных можно, выполнив **Базы данных** / **Системные базы данных** (**Databases** / **System Databases**).

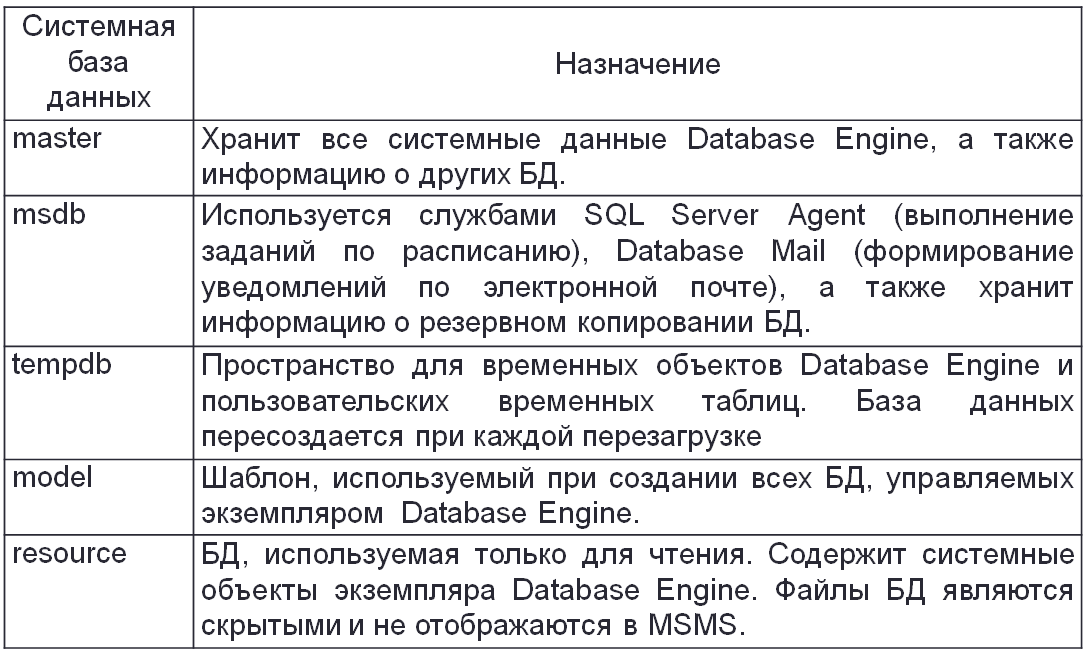
→ ***Выполнение заданий***

******

→ ***Теория (моя)***

Системные БД создаются при инсталляции сервера и используются сервером в процессе работы.

**Database Engine** является ядром системы управления реляционной БД

******