**Дополнительный материал**

**Проектирование базы данных. Разработка ER-диаграммы.**

**Создание базы данных**

Основные темы цикла занятий

[1. Среда разработки 1](#_Toc92574530)

[2. Проектирование базы данных 2](#_Toc92574531)

[3. Работа с таблицами базы данных 5](#_Toc92574532)

[4. Использование скрипта базы данных 12](#_Toc92574533)

# 1. Среда разработки

Среда SQL Server Management Studio — это интегрированная среда для управления инфраструктурой SQL Server и базы данных SQL Azure. Среда Management Studio предоставляет средства для настройки, наблюдения и администрирования экземпляров SQL Server. Она также предоставляет средства для развертывания, отслеживания и обновления данных-уровня компонентов, таких как базы данных и хранилищ данных, используется приложениями, а также для создания запросов и скриптов.

Среда SQL Server Management Studio (SSMS)реализована только под Windows, поэтому если Вам нужен инструмент для работы с Microsoft SQL Server, который будет работать на других платформах, например, на Linux или macOS, то Вам следует использовать инструмент Azure Data Studio, который также является официальным инструментом, разработанным компанией Microsoft.

Для запуска среды SSMS зайдите в Пуск и найдите приложение.

|  |  |  |
| --- | --- | --- |
| Рис.1. | В режиме проверки подлинности сервера программа предложит ввести имя SQL сервера, к которому следует выполнить подключение, а также данные для авторизации на этом сервере (ввести имя пользователя и пароль существующего пользователя SQL Server).  В случае проверки подлинности Windows (Windows Authentication), используются данные текущей учетной записи Windows. Если необходимо использовать учетные данные другого пользователя, то необходимо и программу запустить от имени этого пользователя. | |
| Рис.2. | | Введя имя экземпляра сервера и данные для авторизация, необходимо нажать  «*Соединить*» (Connect) для подключения к выбранному SQL серверу.  (В качестве имени сервера укажите *LAB315-<номер компьютера>*. Например, LAB315-02) | |

|  |  |
| --- | --- |
| Рис.3. | Если окно подключения не появилось, то нажмите Файл/Подключить к обозревателю объектов. Или соответствующую кнопку на панели. |

# 2. Проектирование базы данных

**2.1 Создание базы данных в конструкторе**

|  |  |
| --- | --- |
| Рис.4 | Запустите Microsoft SQL Server Management Studio и подключитесь к серверу. В списке «Обозреватель объектов» (если не отображается, нажмите клавишу F8) кликните правой кнопкой мышки по папке «Базы данных». В появившемся меню выберите пункт «Создать базу данных». |

|  |  |
| --- | --- |
| Рис.5. | В появившемся окне введите имя базы данных, например «test». Поле «владелец» можно оставить по умолчанию. В принципе, этого достаточно, чтобы создать новую базу данных. Нажмите ОК. |

Новая база данных появится в списке. Вы в любое время сможете изменить параметры созданных ранее баз данных через контекстное меню «Свойства».

Если при попытке создать базу данных вы получить сообщение:

Действие Создать завершилось неудачно для объекта "База данных" "test". (Microsoft.SqlServer.Smo)

Разрешение CREATE DATABASE запрещено в базе данных "master". (Microsoft SQL Server, ошибка: 262)

значит, что у вас недостаточно прав и вы не сможете самостоятельно создать новую базу данных на сервере. Для решения этого вопроса обратитесь к администратору вашего сервера баз данных.

**2.2 Удаление базы данных с помощью графического интерфейса**

|  |  |
| --- | --- |
| Рис. 6 | Чтобы удалить базу данных с сервера с помощью ПКМ вызовите контекстное меню и выберите опцию Удалить |

|  |  |
| --- | --- |
| Скриншот 9Рис.7 | В окне «Удаление объекта» нажимаем «ОК». Для принудительного закрытия существующих подключений к БД можете поставить галочку «Закрыть существующие соединения». |

**Примечание!** Удалить базу данных возможно, только если к ней нет никаких подключений, т.е. в ней никто не работает, даже Ваш собственный контекст подключения в SSMS должен быть настроен на другую БД (например, с помощью команды USE). Поэтому предварительно перед удалением необходимо попросить всех завершить сеансы работы с БД, или в случае с тестовыми базами данных принудительно закрыть все соединения.

**2.3. Создание базы данных с помощью запросов**

Процесс создания базы данных на языке SQL, наверное, еще проще, так как для того чтобы создать БД с настройками по умолчанию (как мы это сделали чуть выше), необходимо написать всего три слова в редакторе SQL запросов – инструкцию CREATE DATABASE и название БД.

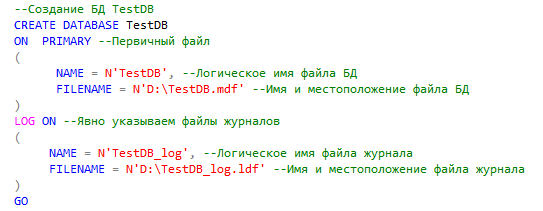
|  |  |
| --- | --- |
| Скриншот 6Рис. 7 | Сначала открываем редактор SQL запросов, для этого щелкаем на кнопку «Создать запрос» на панели инструментов. |

Затем вводим следующую инструкцию, и запускаем ее на выполнение, кнопка «Выполнить».

CREATE DATABASE TestDB;

Где CREATE – это команда языка SQL для создания объектов на SQL сервере, командой DATABASE мы указываем, что хотим создать базу данных, а TestDB — это имя новой базы данных.

С помощью инструкции CREATE DATABASE можно задать абсолютно все параметры, которые отображались у нас в графическом интерфейсе SSMS. Заменим вышеуказанную инструкцию следующей. База данных c таким же именем будет создана на диске D (выделенные зеленым цветом комментарии можно не писать).



**2.4 Удаление базы данных в Microsoft SQL Server**

В случае необходимости можно удалить базу данных. В реальности, конечно же, такое редко будет требоваться, но в процессе обучения, может быть, и часто. Это можно сделать также и с помощью языка SQL.

Для удаления базы данных достаточно написать следующую инструкцию (в БД также никто не должен работать).

DROP DATABASE TestDB;

Где DROP DATABASE — это инструкция для удаления базы данных, TestDB – имя базы данных. Иными словами, командой DROP объекты на SQL сервере удаляются.

|  |  |
| --- | --- |
| **Рис.8** | **Примечание!** Если при создании или удалении базы данных в обозревателе не появляются изменения (база данных не появилась или не удаляется), необходимо Обновить базы данных или Отключить базу данных от сервера и подключиться снова. |

# 3. Работа с таблицами базы данных

**3.1 Создание таблицы в конструкторе**

Рассмотрим примеры создания таблиц как с помощью графического интерфейса, специально для начинающих, так и с помощью инструкции CREATE TABLE языка SQL.

Как было уже отмечено, создать таблицу в Microsoft SQL Server можно двумя способами: первый — с помощью графического конструктора SSMS, и второй — с помощью инструкции на языке SQL.

**Исходные данные для примера**

Давайте представим, что нам нужно реализовать базу данных ***TestDB*** со следующей структурой (пример структуры тестовый). В ней у нас будет две таблицы, и они будут содержать следующие столбцы:

**Goods** – таблица будет содержать информацию о товарах:

ProductId – идентификатор товара, столбец не может содержать значения NULL, первичный ключ;

Category – ссылка на категорию товара, столбец не может содержать значения NULL, но имеет значение по умолчанию, например, для случаев, когда товар еще не распределили в необходимую категорию, в этом случае товару будет присвоена категория по умолчанию («Не определена» или «Не указана»);

ProductName – наименование товара, столбец не может содержать значения NULL;

Price – цена товара, столбец может содержать значения NULL, например, с ценой еще не определились.

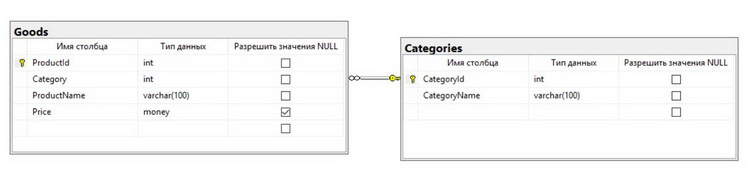
**Categories** — таблица будет содержать описание категорий товаров:

CategoryId – идентификатор категории, столбец не может содержать значения NULL, первичный ключ;

CategoryName – наименование категории, столбец не может содержать значения NULL.

При этом внести товар с несуществующей категорией нельзя, поэтому мы добавим еще и ограничение внешнего ключа.

Так будет выглядеть схема нашей базы данных.

Рис.9.

Запускаем среду SQL Server Management Studio.

В обозревателе объектов открываем контейнер «Базы данных», затем создаем нужную базу данных ***TestDB*** любым способом.

|  |  |
| --- | --- |
| Скриншот 3Рис.10 | Щелкаем правой кнопкой мыши по пункту «Таблицы», и выбираем «Таблица».  У Вас откроется конструктор таблиц. В нем будет всего три колонки:  Имя столбца – сюда пишем название столбца;  Тип данных – выбираем тип данных для этого столбца.  Разрешить значения NULL – если поставить галочку, то столбец сможет принимать значение NULL. |

|  |  |
| --- | --- |
| Скриншот 4Рис.11 | Заполняем эти колонки, сначала в соответствии с нашей тестовой структурой таблицы Categories. |

|  |  |
| --- | --- |
| Скриншот 5Рис.12 | После этого нам нужно определить первичный ключ, для этого щелкаем правой кнопкой мыши по нужному столбцу (в нашем случае это CategoryId) и выбираем пункт «Задать первичный ключ». |

Также для этого столбца определим спецификацию идентификатора, т.е. зададим свойство IDENTITY, для того чтобы данный столбец автоматически генерировал уникальный идентификатор записи.

|  |  |
| --- | --- |
| Чтобы это сделать, в свойствах столбца в нижней части конструктора ищем раздел «Спецификация идентификатора» и включаем его, т.е. ставим «Да». В случае необходимости Вы можете задать начальное значение идентификатора, например, для того чтобы начать идентификацию с определённого значения, а также можете изменить шаг приращения, т.е. на какое значение будет увеличиваться Ваш идентификатор. | Скриншот 6Рис.13 |

|  |  |
| --- | --- |
| Скриншот 7Рис.14. | Определение нашей таблицы готово, теперь нам ее необходимо сохранить. Для этого щелкаем по вкладке правой кнопкой мыши и нажимаем «Сохранить» или просто нажимаем сочетание клавиш «Ctrl+S», также кнопка «Сохранить» доступна и в меню «Файл».  Далее вводим название таблицы, в нашем случае это Categories, и нажимаем «OK». |

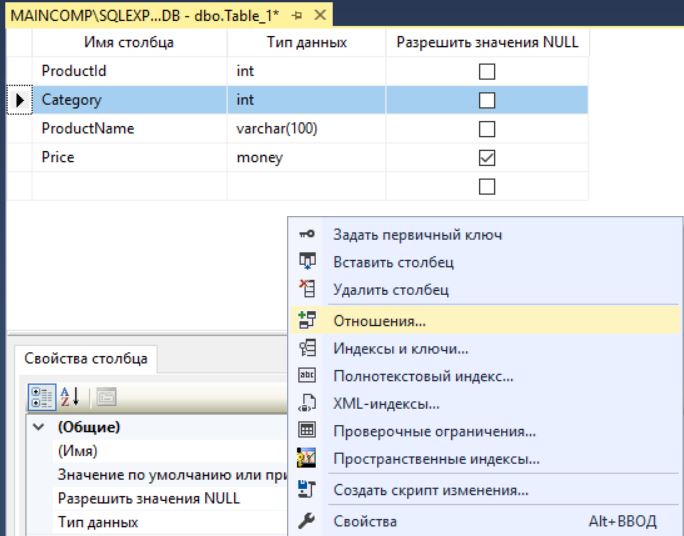
Все, конструктор можно закрыть, можете обновить обозреватель объектов, чтобы таблица у Вас отобразилась.

Теперь переходим к таблице Goods. В этом случае делаем все то же самое, т.е. определяем столбцы, задаем первичный ключ и задаем спецификацию идентификатора. Только в данном случае нам нужно дополнительно задать значение по умолчанию для столбца Category и создать ограничение внешнего ключа (FOREIGN KEY).

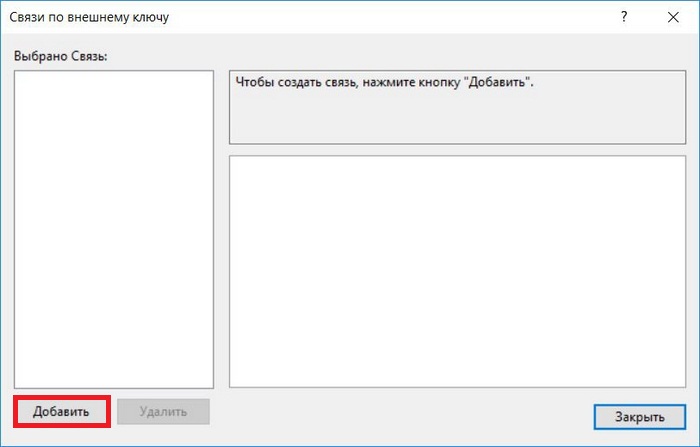
FOREIGN KEY - это поле (или набор полей) в одной таблице, которое ссылается на первичный ключ в другой таблице. Таблица, содержащая внешний ключ, называется дочерней таблицей, а таблица, содержащая ключ - кандидат, называется ссылочной или родительской таблицей.

|  |  |
| --- | --- |
| Скриншот 8Рис.15 | Для того чтобы задать значение по умолчанию, необходимо выбрать столбец, и в свойствах этого столбца в параметре «Значение по умолчанию или привязка» указать желаемое значение по умолчанию, в нашем случае давайте напишем 1. |

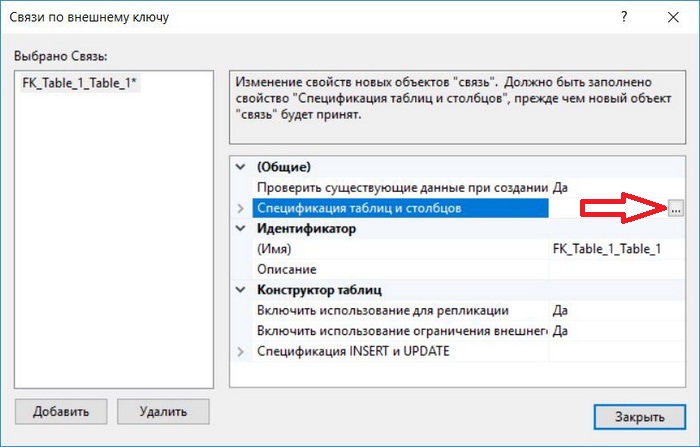
Чтобы создать внешний ключ, щелкаем в любом месте конструктора правой кнопкой мыши и выбираем пункт «Отношения…».

Рис.16

Затем нажимаем добавить.

 Рис.17

Далее задаем спецификацию таблиц и столбцов, для этого щелкаем на три точки напротив соответствующего свойства.

 Рис.18

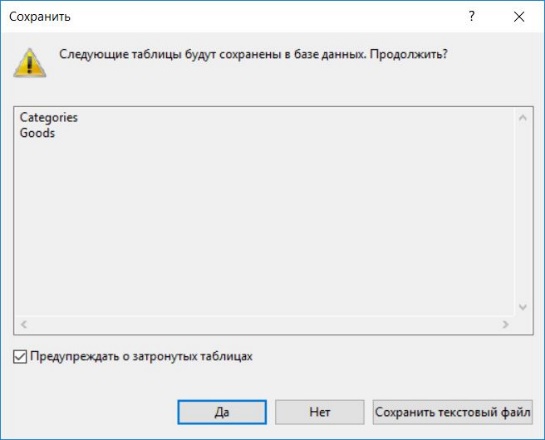
|  |  |
| --- | --- |
| Скриншот 12Рис.19 | Потом откроется окно, в котором мы указываем следующее:   * Таблица первичного ключа – выбираем из списка таблицу Categories, а также ее первичный ключ, по которому будет осуществляться связь; * Таблица внешнего ключа – это как раз наша текущая таблица, пока она еще не создана, поэтому она отображается как Table\_1, в этом случае выбираем столбец Category этой таблицы, который будет выполнять роль внешнего ключа, т.е. это и будет ссылка на внешнюю таблицу (т.е. сопоставление таблиц будет осуществляться как CategoryId = Category); * Имя связи — название ограничения, допустим, у нас это будет FK\_Category.   Нажимаем «ОК». |

Нам осталось задать правила обновления и удаления, т.е. что будет происходить с записями таблицы Goods (они же ссылаются на таблицу Categories) если категория (запись таблицы Categories) будет изменена или удалена.

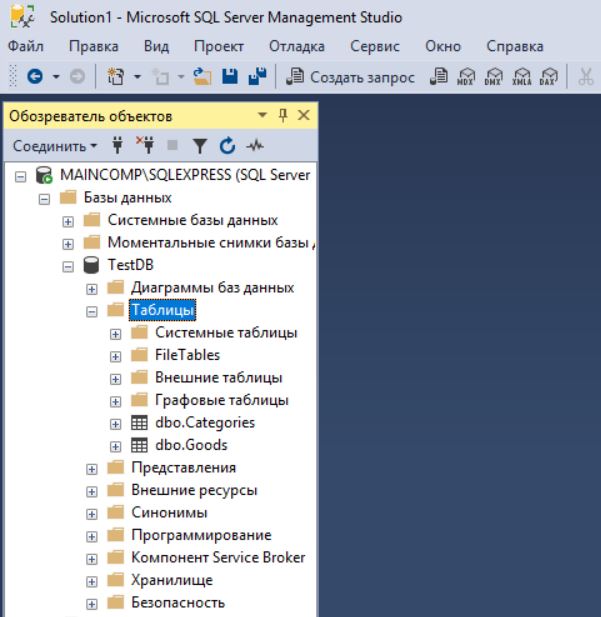
|  |  |
| --- | --- |
| Скриншот 13Рис.20 | Изменять идентификатор категории вряд ли придётся, а если и придётся, то пусть в этих случаях появится ошибка, иными словами, правило обновление просто не задаем. А вот в случае с удалением категории, пусть всем товарам присвоится значение по умолчанию, т.е. неопределенная категория. Для этого определяем правило удаления как «Присвоить значение по умолчанию». |

Нажимаем закрыть.

Затем можем сохранить таблицу тем же способом, что и раньше. Называем ее Goods. В случае если появится предупреждающее сообщение о том, что будут затронуты следующие таблицы, отвечаем «Да», т.е. продолжаем.

Рис.21

После обновления объектов в обозревателе, созданная таблица отобразится.

 Рис.22

Теперь Вы можете добавлять данные в эти таблицы.

**3.2 Создание таблицы с помощью инструкции CREATE TABLE языка SQL**

Рассмотрим процесс создания тех же самых таблиц, но только на языке SQL с использованием инструкции **CREATE TABLE**.

Для этого создадим еще одну базу данных и назовем ее ***trade.***

Упрощённый синтаксис создания таблиц следующий:

CREATE TABLE Название таблицы (

[Название столбца] [Тип данных] [Возможность принятия значения NULL] [Определение ограничения],

…

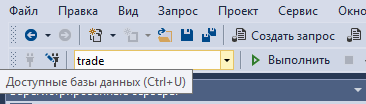
)

В реальности синтаксис инструкции CREATE TABLE очень большой и с первого взгляда сложный, поэтому начинающим лучше сначала понять принцип создания таблицы, а потом углубляться в детали.

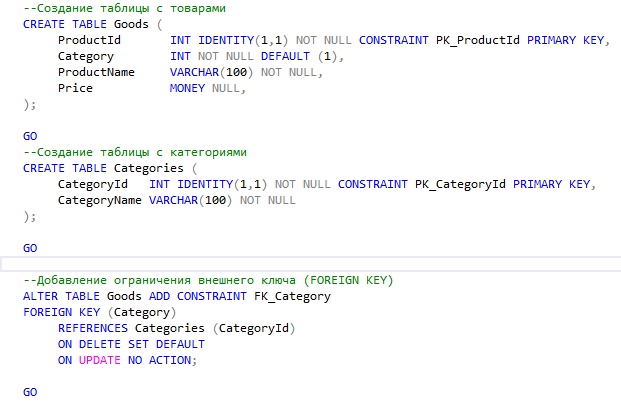
Чтобы написать и выполнить инструкцию SQL, открываем редактор SQL запросов, для этого нажимаем кнопку «Создать запрос» и пишем необходимую инструкцию, она представлена чуть ниже. Эта инструкция эквивалентна всем действиям, которые мы делали в графическом интерфейсе.

**Примечание!** Если таблицы уже существуют в заданной базе данных, а вы хотите протестировать следующую инструкцию SQL по созданию таблиц, то Вам предварительно нужно удалить эти таблицы, так как сервер выдаст ошибку. Для этого специально включена в инструкцию команда DROP TABLE IF EXISTS, которая удаляет таблицы, в случае если они существуют.

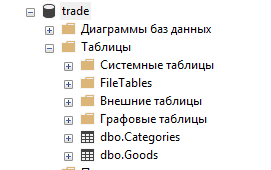
**Примечание!** Чтобы создаваемые таблицы попали в нужную базу данных, ее имя должно быть выделено в окне «Доступные базы данных»

 Рис.23

Выполните следующий скрипт.

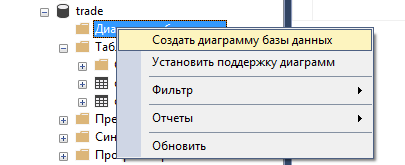


Убедитесь, что таблицы созданы в заданной базе данных.

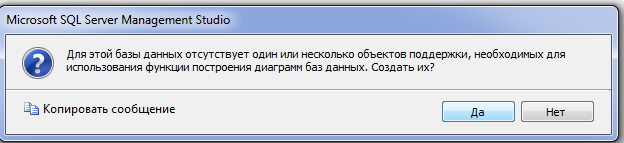
Рис.24

**3.3 Создание схемы базы данных**

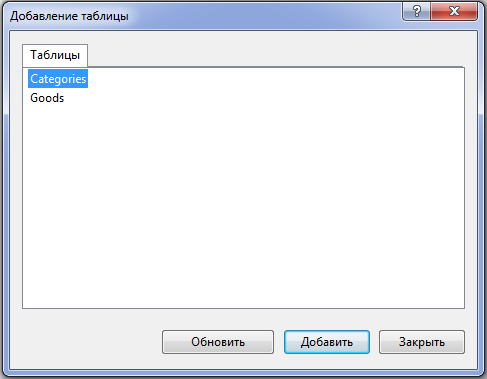
В случае, если таблицы базы данных и связи между ними уже определены, схема базы данных может быть создана в конструкторе. Для этого кликнем ПКМ по разделу Диаграммы базы данных.

Рис.24

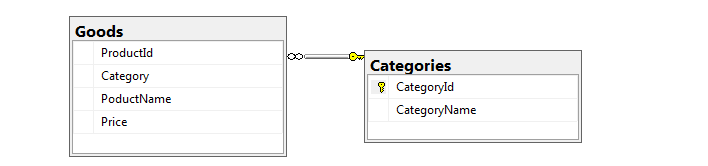
В появившемся сообщении отметим Да.

Рис.25

В окне Добавление таблицы добавим имеющиеся таблицы в диаграмму.

Рис.26

В нашем примере был задан внешний ключ, поэтому в диаграмме сразу отразилась связь между таблицами.

Рис.27

# 4. Использование скрипта базы данных

В SQL Server Management Studio есть очень полезный функционал, который позволяет сгенерировать скрипт создания объектов базы данных, чтобы, например, использовать этот скрипт на другом экземпляре SQL Server для создания точно таких же объектов.

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

С помощью данного мастера мы можем очень легко сгенерировать SQL скрипт создания практически любого объекта на SQL Server.

Такой скрипт может включать:

- Инструкции создания таблиц (CREATE);

- Инструкции добавления данных (INSERT);

- Определение представлений, функций, хранимых процедур, триггеров;

- Определение ограничений и индексов;

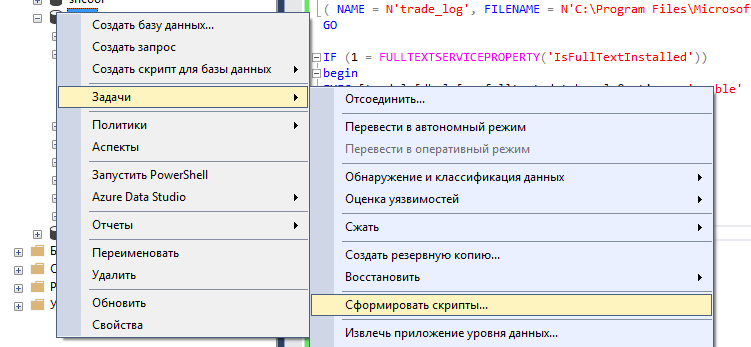
- другие SQL инструкции.

Такие SQL скрипты могут быть очень полезны администраторам или разработчикам, например, для того, чтобы в случае необходимости иметь возможность быстро восстановить объекты базы данных с помощью этих скриптов, или для того, чтобы передать эти SQL скрипты другому администратору, разработчику или заказчику, чтобы он создал подобные объекты на своем экземпляре SQL Server.

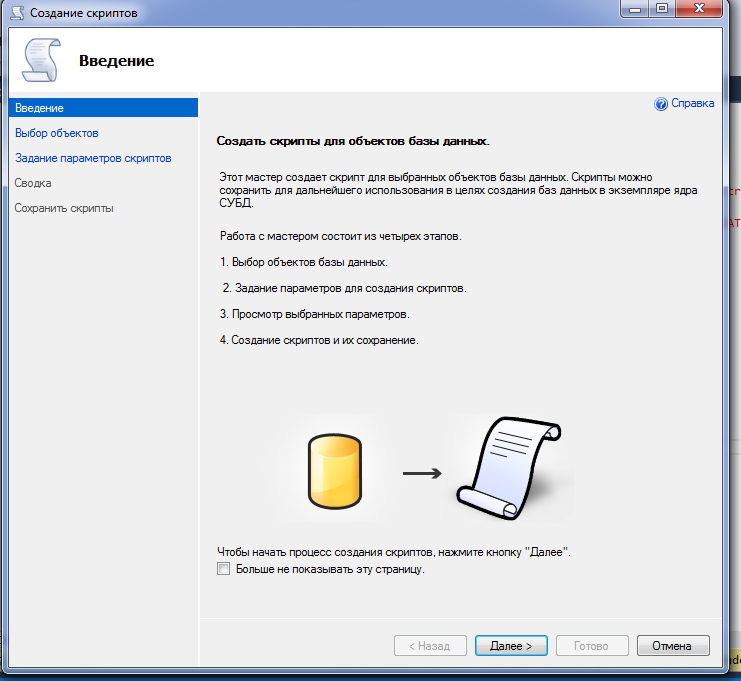
Таким образом, с помощью данного мастера мы можем создать некий дамп базы данных, содержащий SQL скрипты создания объектов этой базы данных и наполнения их данными.

**4.1 Генерация скрипта базы данных**

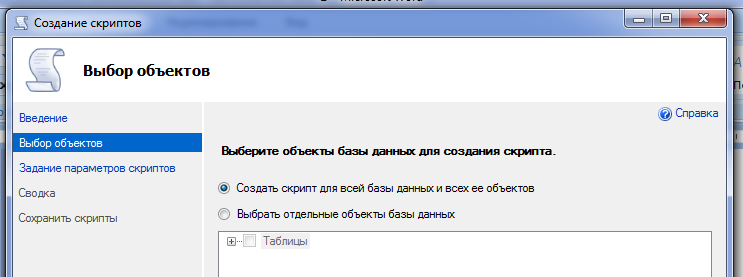
Щелкните правой кнопкой мыши базу данных **trade** и последовательно выберите пункты **Задачи/Сформировать скрипты**.

Рис.28

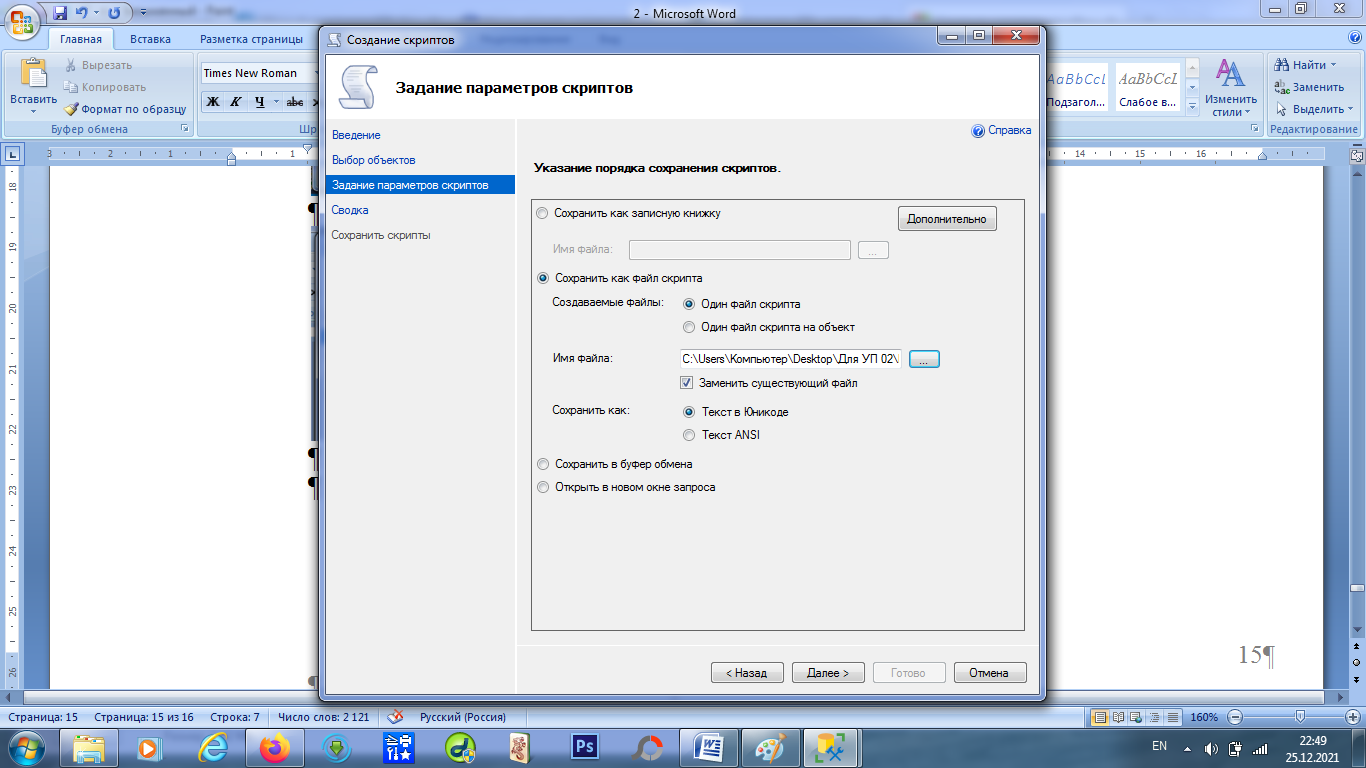
Появится окно Создание скриптов. В нем отмечаем Далее.

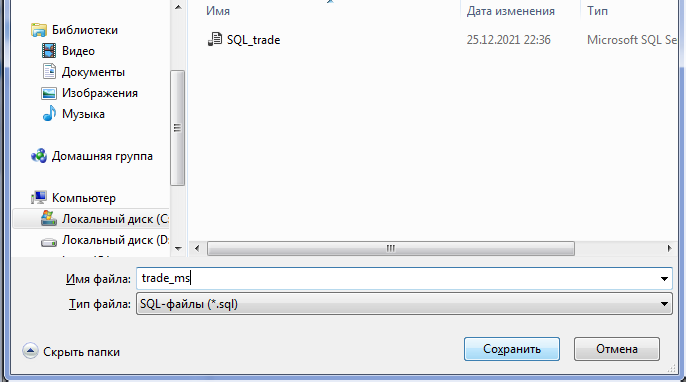
Рис.29

В разделе Выбор объектов отмечаем Создать скрипт для всей базы данных и всех ее объектов.

Рис.30

Далее в разделе Задание параметров скриптов вносим необходимые сведения, задаем имя файла и расположение скрипта.

Рис.31

Рис.32

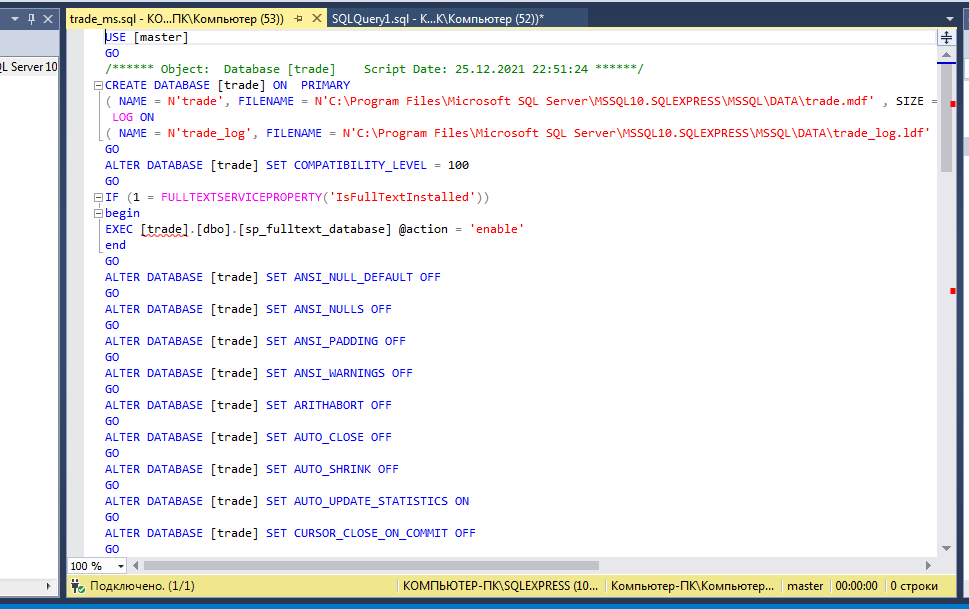
Убедитесь, что в вашей папке появился скрипт с заданным именем.

**4.2 Восстановление базы данных из скрипта**

Для проверки работы скрипта удалите базу данных **trade** с сервера с помощью опции Удалить или скрипта

*DROP database trade;*

Войдите в свою папку, в которой находится скрипт базы данных. Откройте его двумя кликами. При этом скрипт откроется в окне нового запроса среды SSMS.

Рис.33

Выполните запрос. В обозревателе снова появится база данных **trade**. Если база данных не появилась в обозревателе, отключите соединение с сервером и снова подключитесь, или используйте кнопку Обновить.