**САМОСТОЯТЕЛЬНАЯ РАБОТА ПО МОДУЛЮ №4**

**Курс: «Теория Баз Данных»**

**Тема: «Многотабличные базы данных»**

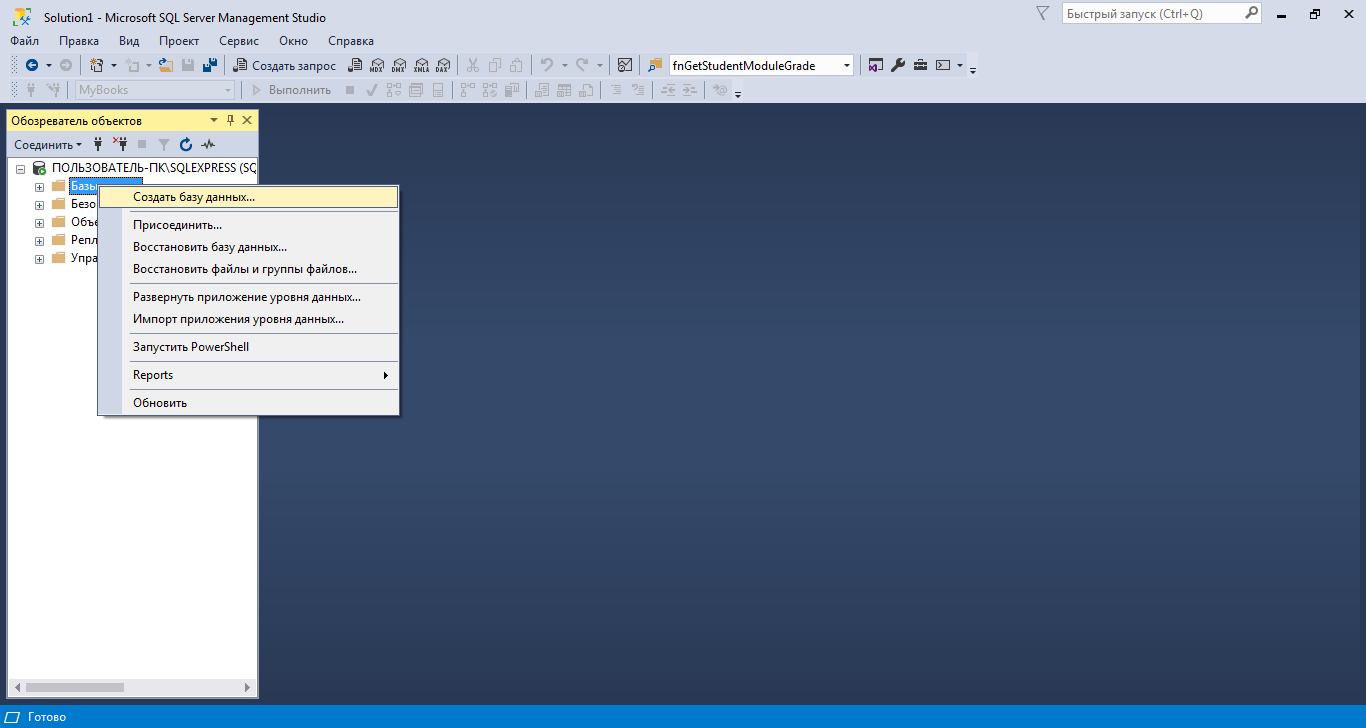
**Цель работы:** Ознакомить студентов с многотабличными базами данных. Научить их работать с запросами к многотабличным базам данных. Показать им главные отличия многотабличных баз данных от однотабличных баз данных. Научить их приемам, которые помогут им эффективно работать с многотабличными базами данных, а также с данными, которые могут быть импортированы из различных внешних источников.

**Данные:** Текстовый файл в формате CSV или файл в формате Microsoft Excel.

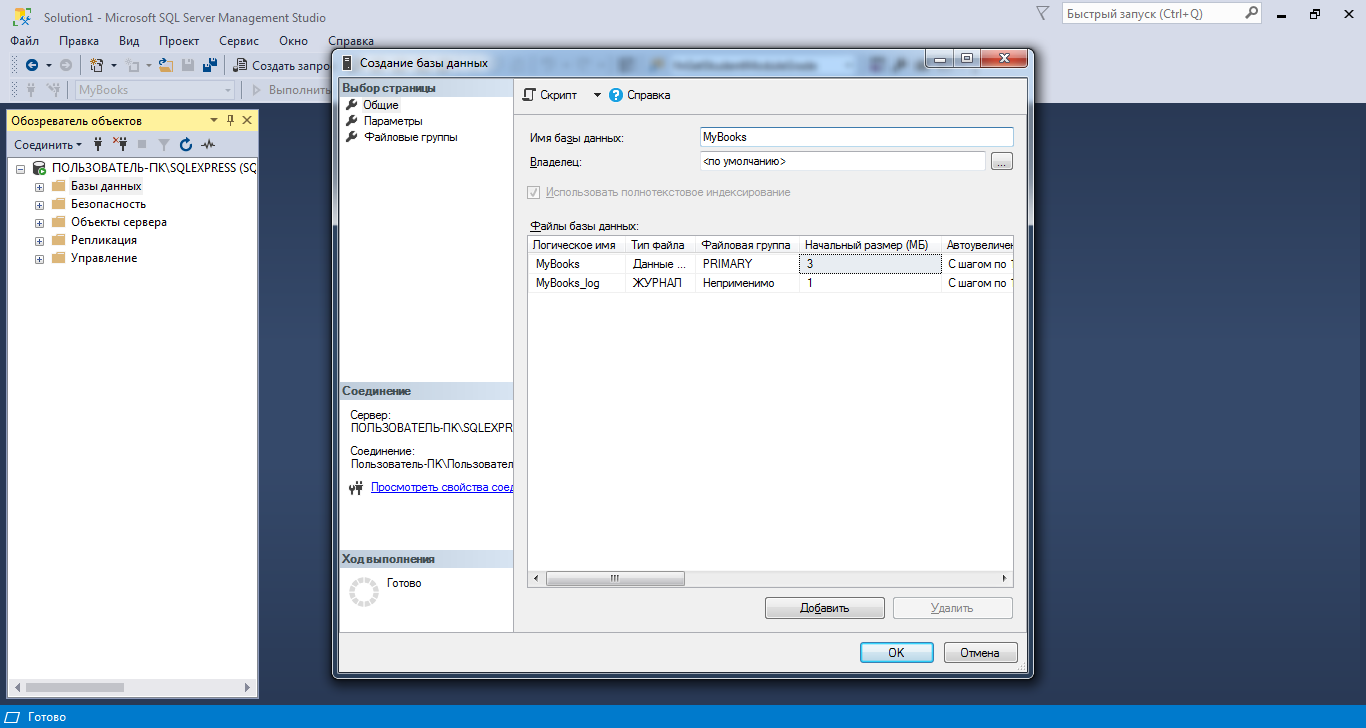
**Результат:** Многотабличная база данных “MyBooks”, полученная в результате операций над исходной таблицей.

**Шаг №1**

Необходимо создать базу данных “MyBooks” средствами Microsoft SQL Server Management Studio (MSSMS). Для этого необходимо сначала щелкнуть правой кнопкой мыши в обозревателе объектов, а затем выбрать в контекстном меню команду «Создать базу данных…».

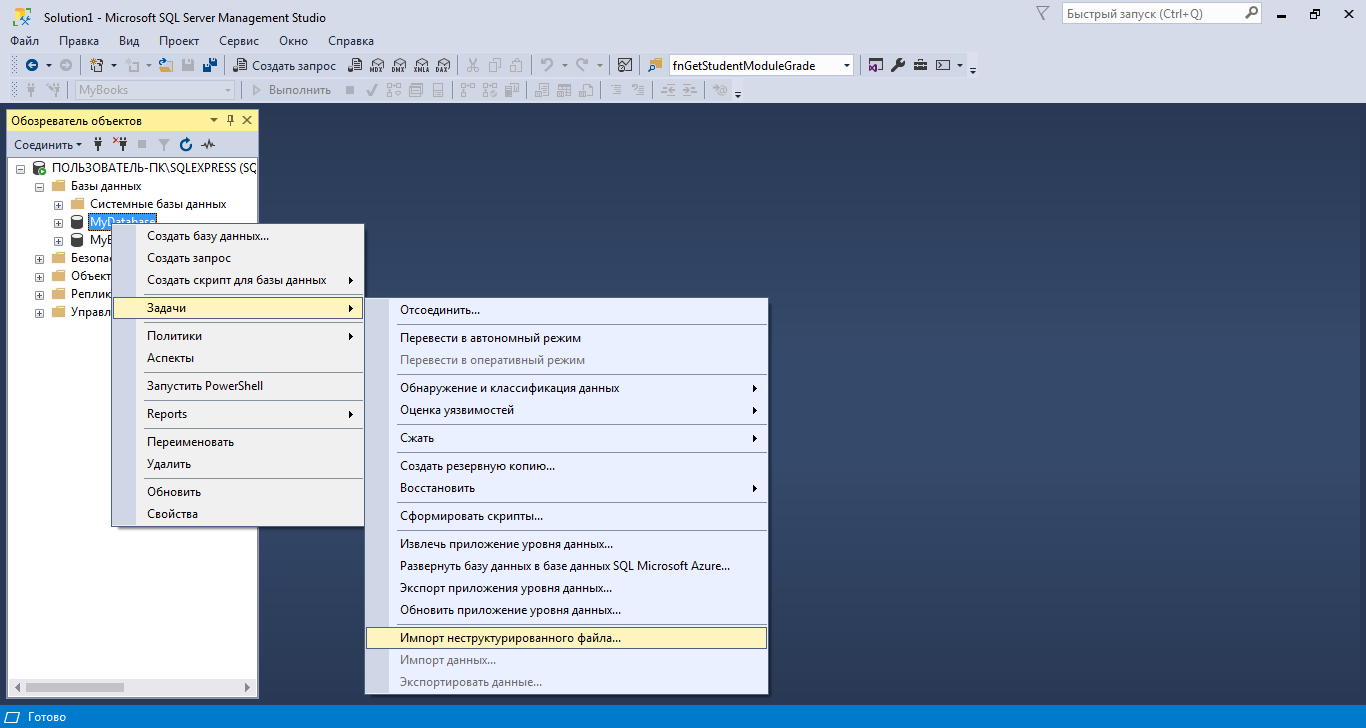


В открывшемся окне указать имя базы данных: “MyBooks”



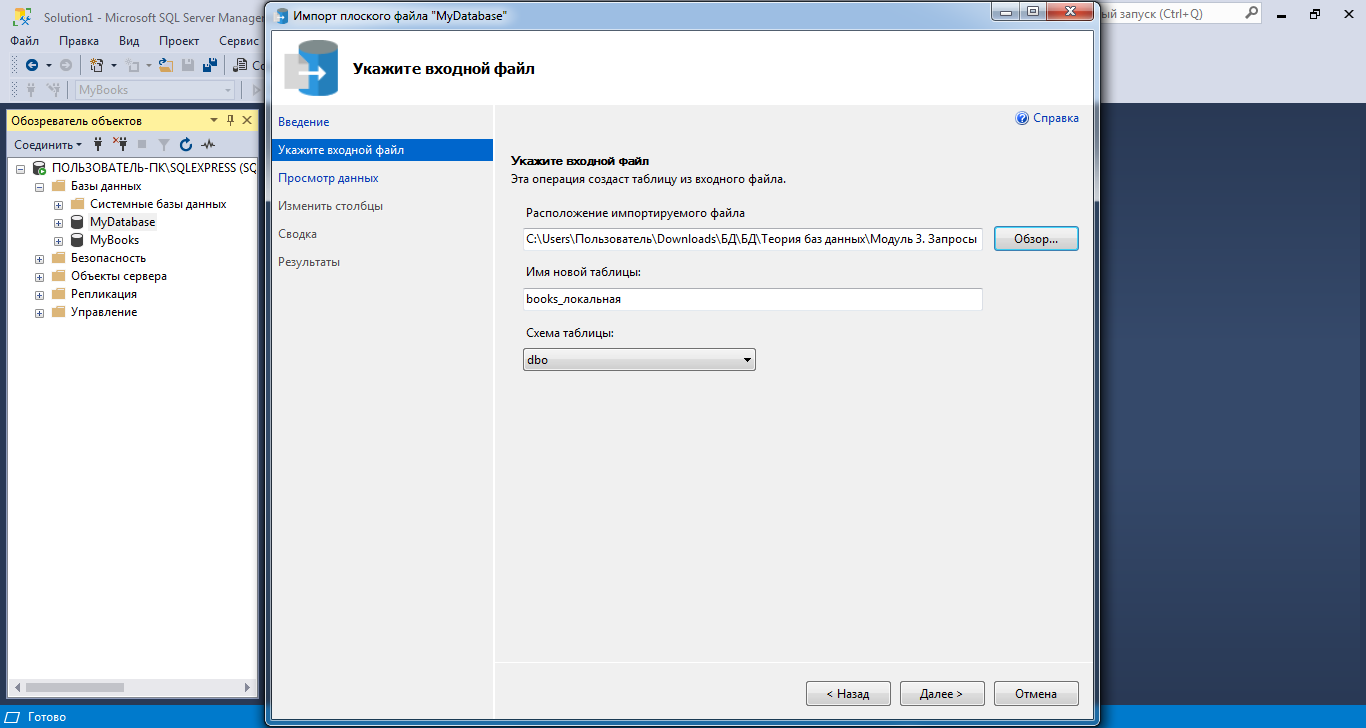
Введите имя базы данных и нажмите “OK”.

Затем необходимо щелкнуть правой кнопкой в обозревателе объектов на созданной базе данных и последовательно выбрать команды «Задачи» и «Импорт неструктурирован-ного файла».



В открывшемся окне сначала укажите путь к файлу “books\_локальная”, а затем нажатием кнопки «Далее» завершите операцию импорта данных в созданную ранее базу данных. В результате в ней должна появиться таблица с именем “books\_локальная”.

Примечание: В том случае если в обозревателе объектов не отображается база данных, которая была создана вами, то просто нажмите кнопку «Обновить» на верхней панели.



Мы получили однотабличную базу данных.

**Шаг №2**

Выполните первый запрос для того, чтобы просмотреть содержащуюся в этой базе данных информацию.

Запрос №1

-- выбираем базу данных MyBooks по умолчанию

USE [MyBooks]

GO

-- извлекаем все данные из исходной таблицы для просмотра

SELECT \* FROM [books\_локальная]

GO

Как вы можете заметить, в этой однотабличной базе данных много избыточной информации, которую для начала необходимо нормализовать. Для этого нам необходимо вынести избыточные данные в другие таблицы.

Нам нужно создать три дополнительные таблицы:

* “Publishers” – для хранения данных об издательствах;
* “Categories” – для хранения данных о категориях книг;
* “Themes” – для хранения данных о тематике книг.

Создадим эти таблицы при помощи запросов, указанных ниже:

Запрос №2

-- выбираем базу данных MyBooks по умолчанию

USE [MyBooks]

GO

-- удаляем таблицу для хранения всех издателей средствами T-SQL

IF (OBJECT\_ID('Publishers') IS NOT NULL)

DROP TABLE [Publishers]

GO

-- создаем таблицу для хранения всех издателей средствами T-SQL

CREATE TABLE [Publishers] (

[ID] INT IDENTITY(1,1) PRIMARY KEY, -- это первичный ключ

[Name] NVARCHAR(50)

)

GO

-- выбираем все данные по издателям из исходной таблицы и импортируем их в созданную таблицу издателей средствами T-SQL

INSERT INTO [Publishers] SELECT DISTINCT [Izd] FROM [books\_локальная] WHERE [Izd] IS NOT NULL

GO

Запрос №3

-- выбираем базу данных MyBooks по умолчанию

USE [MyBooks]

GO

-- удаляем таблицу для хранения всех категорий средствами T-SQL

IF (OBJECT\_ID('Categories') IS NOT NULL)

DROP TABLE [Categories]

GO

-- создаем таблицу для хранения всех категорий средствами T-SQL

CREATE TABLE [Categories] (

[ID] INT IDENTITY(1,1) PRIMARY KEY, -- это первичный ключ

[Name] NVARCHAR(50)

)

GO

-- выбираем все данные по издателям из исходной таблицы и импортируем их в созданную таблицу категорий средствами T-SQL

INSERT INTO [Categories] SELECT DISTINCT [Category] FROM [books\_локальная] WHERE [Category] IS NOT NULL

GO

Запрос №4

-- выбираем базу данных MyBooks по умолчанию

USE [MyBooks]

GO

-- удаляем таблицу для хранения всех тематик средствами T-SQL

IF (OBJECT\_ID('Themes') IS NOT NULL)

DROP TABLE [Themes]

GO

-- создаем таблицу для хранения всех тематик средствами T-SQL

CREATE TABLE [Themes] (

[ID] INT IDENTITY(1,1) PRIMARY KEY, -- это первичный ключ

[Name] NVARCHAR(50)

)

GO

-- выбираем все данные по издателям из исходной таблицы и импортируем их в созданную таблицу тематик средствами T-SQL

INSERT INTO [Themes] SELECT DISTINCT [Themes] FROM [books\_локальная] WHERE [Themes] IS NOT NULL

GO

Поскольку у нас уже есть три созданные таблицы, то мы можем сравнить данные при помощи проецирования (то есть, отображения информации из разных столбцов и из разных таблиц в одной таблице). Таким образом, мы можем увидеть, что эта информация может быть взаимосвязана между собой, даже если она находится в разных таблицах.

Выполните пятый запрос, чтобы убедиться в этом:

Запрос №5

-- выбираем базу данных MyBooks по умолчанию

USE [MyBooks]

GO

-- проецируем данные из исходной таблицы таким образом, чтобы совместить идентификаторы всех издателей, категорий и тематик

SELECT N, New, Name, Price, Pages, Format, Date, Pressrun, ID AS [Izd\_ID], ID AS [Theme\_ID], ID AS [Category\_ID]

FROM [books\_локальная], [Categories] AS c, [Publishers] AS p, [Themes] AS t

WHERE Category = c.Name AND Izd = p.Name AND Themes = t.Name

GO

-- в запросе было использовано ключевое слово AS позволяющее заменить длинные имена таблиц или столбцов более коротким и понятным

-- видим, что в результатах запроса стало меньше записей, чем было в исходной таблице

-- это связано с тем, что выполнить объединение с теми записями, где издатель, категория или тематика не указаны, не удалось

Для того чтобы физически связать информацию, содержащуюся в этих таблицах, нам потребуются дополнительные поля в исходной таблице, которые в последствии могут стать внешними ключами и обеспечить создание связей между всеми таблицами базы данных.

Создадим эти столбцы при помощи запроса, указанного ниже:

Запрос №6

-- выбираем базу данных MyBooks по умолчанию

USE [MyBooks]

GO

-- изменим исходную таблицу средствами T-SQL

ALTER TABLE [books\_локальная]

ADD Izd\_ID INT NULL, Theme\_ID INT NULL, Category\_ID INT NULL

GO

На следующем этапе нам потребуется заполнить созданные столбцы данными, на данный момент все столбцы заполнены значениями NULL.

Выполните седьмой запрос, чтобы заполнить созданные столбцы значениями из созданных нами ранее таблиц “Publishers”, “Categories” и “Themes”. Эти значения будут являться ссылками на значения первичных ключей этих таблиц.

Запрос №7

-- выбираем базу данных MyBooks по умолчанию

USE [MyBooks]

GO

-- изменяем значение созданных в предыдущем запросе столбцов так чтобы они совпадали с идентификаторами изданий, категорий и тематик

UPDATE [books\_локальная]

SET Izd\_ID = (

SELECT p.ID FROM [Publishers] AS p WHERE p.Name = Izd)

GO

UPDATE [books\_локальная]

SET Theme\_ID = (

SELECT t.ID FROM [Themes] AS t WHERE t.Name = Themes)

GO

UPDATE [books\_локальная]

SET Category\_ID = (

SELECT c.ID FROM [Categories] AS c WHERE c.Name = Category)

GO

Теперь нам больше не нужны столбцы “Izd”, “Themes” и “Category”, поскольку данные из них были перенесены в другие таблицы, а в исходной таблице есть столбцы, которые ссылаются на эти таблицы посредством одинаковых значений в созданных столбцах “Izd\_ID”, “Theme\_ID” и “Category\_ID”. Можно удалить ненужные столбцы.

Удалим перечисленные выше столбцы при помощи запроса, указанного ниже, а затем тут же переименуем исходную таблицу:

Запрос №8

-- выбираем базу данных MyBooks по умолчанию

USE [MyBooks]

GO

-- удаляем ненужные столбцы средствами T-SQL

ALTER TABLE [books\_локальная]

DROP COLUMN [Izd],[Themes],[Category]

GO

-- изменяем название исходной таблицы средствами T-SQL

EXEC sp\_rename 'books\_локальная', 'Books'

GO

Наконец, все готово к созданию связей в нашей многотабличной базе данных. Осталось создать внешние ключи в исходной таблице и сохранить все изменения. Для создания внешних ключей нужно указать связи между всеми таблицами.

Выполните девятый запрос, чтобы создать связи между таблицами и сохранить изменения в многотабличной базе данных:

Запрос №9

-- выбираем базу данных MyBooks по умолчанию

USE [MyBooks]

GO

-- добавляем информацию по внешним ключам переименованной таблицы средствами T-SQL

ALTER TABLE [Books]

ADD

CONSTRAINT FK\_Books\_Publishers FOREIGN KEY (Izd\_ID)

REFERENCES [Publishers] (ID)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT FK\_Books\_Categories FOREIGN KEY (Category\_ID)

REFERENCES [Categories] (ID)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT FK\_Books\_Themes FOREIGN KEY (Theme\_ID)

REFERENCES [Themes] (ID)

ON DELETE CASCADE

ON UPDATE CASCADE

GO

Вы только что создали многотабличную базу данных на основе имеющегося в Вашем арсенале неструктурированного текстового файла. Вы выполнили все операции над имеющимися данными с целью преобразования однотабличной базы данных в многотабличную, а также привели эту базу данных к нормализованному состоянию. Кроме того, были созданы связи между всеми таблицами.

Смело задавайте вопросы по данной самостоятельной работе, а также делитесь выводами о ней с другими студентами.

Спасибо за внимание!