

## Лабораторная работа 3:

# Проектирование и создание Таблиц

## Создание таблицы в SSMS

В SSMS с помощью Конструктора таблиц вы создадите таблицу **Customers**.

1. Запустите SSMS:  
**Start → Programs → Microsoft SQL Server 20XX → SQL Server Management Studio**
2. В **Object Explorer** раскройте узел **Databases**, чтобы увидеть созданную в лабораторной работе 1 БД **ApressFinancial**.
3. Раскройте БД **ApressFinancial** до узла **Tables** (Таблицы).
4. Щелкните ЛПКМ по узлу **Tables** (Таблицы) и в контекстном меню выберите команду **New Table...** (Создать таблицу). Откроется **Table Designer** (Конструктор таблиц) и соответствующая ПИ.
5. В верхней части Конструктора необходимо ввести следующие данные для каждого столбца:
  - имя столбца – поле **Column Name**;
  - тип столбца – поле с раскрывающимся списком **Data Type**;
  - флажок **Allow Nulls** (Разрешить значения Null).

	Column Name	Data Type	Allow Nulls
	CustomerId	bigint	<input type="checkbox"/>
	CustomerTitleId	int	<input type="checkbox"/>
	CustomerFirstName	nvarchar(50)	<input type="checkbox"/>
	CustomerOtherInitials	nvarchar(10)	<input checked="" type="checkbox"/>
	CustomerLastName	nvarchar(50)	<input type="checkbox"/>
	AddressId	bigint	<input type="checkbox"/>
	AccountNumber	nchar(15)	<input type="checkbox"/>
	AccountTypeId	int	<input type="checkbox"/>
	CleareBalance	money	<input type="checkbox"/>
	UndeareBalance	money	<input type="checkbox"/>
►	DateAdded	date	<input type="checkbox"/>

Рис. 3.1 Вид создаваемой таблицы в верхней части Table Designer (Конструктора таблиц).

**Примечание.** При именовании столбцов старайтесь избегать пробелов: либо части имени начинайте с прописной буквы, либо используйте символ подчеркивания (\_).

Использовать пробелы в именах столбцов допустимо, но в коде T-SQL (например, при написании запросов к таблице) имена столбцов придется писать в квадратных скобках ([ ]), что обременительно. Обратите внимание, что раздел **Column Properties** (Свойства столбца) в нижней части Конструктора заполняется только после начала ввода типа данных (который следует после имени столбца). Заполнение этого раздела также важно, как и заполнение верхней части Конструктора таблиц. Постарайтесь внимательно просмотреть этот раздел. Здесь показаны: имя столбца, признак – разрешено ли значение Null, а также тип хранимых данных. В зависимости от выбранного типа данных в разделе возможны изменения.

6. Определите первичный ключ таблицы. Щелкните ЛПКМ по столбцу **CustomerId** и выберите из контекстного меню пункт **Set Primary Key** (Задать первичный ключ): слева от имени столбца отобразится значок ключа. Еще раз щелкните по столбцу **CustomerId** и выберите из контекстного меню пункт **Indexes/Keys...** (Индексы/Ключи). В открывшемся ОД **Indexes/Keys** (Индексы/Ключи) в разделе

**Table Designer** проверьте значение свойства **Create As Clustered**: значение надо установить в **No**.

7. Настройте в разделе **Column Properties** (Свойства столбца) следующие свойства:
  - для столбца **CustomerId** свойство **(Is Identity)** установите **Yes** (Да);
  - для столбца **DateAdded** в свойстве **Default Value or Binding** (Значение по умолчанию) введите `getdate()` – при добавлении строки будут вставляться текущие дата и время;
8. После определения столбцов таблицы, но перед ее сохранением необходимо определить некоторые свойства таблицы. Справа от Конструктора таблиц должно находиться ОД **Table Properties** (Свойства таблицы). Если оно не отображается, то либо нажмите клавишу <F4>, либо меню **View → Properties Window**. В ОД **Table Properties** укажите:
  - имя таблицы: **Customers**;
  - схема (поле с раскрывающимся списком **Schema**) – **CustomerDetails** (была создана в лабораторной работе 2);
  - поле **Descriptions** (Описание) – дайте некое описание таблицы:

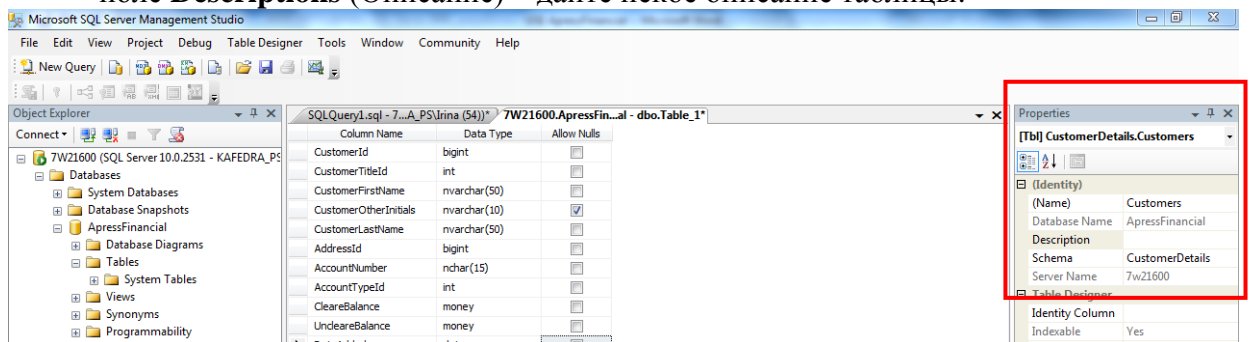


Рис. 3.2 ОД Properties (Свойства таблицы) в Конструкторе таблиц.

9. Теперь можно сохранять таблицу: нажмите на стандартной ПИ кнопку **Save** (Сохранить) либо кнопку <X> (Заккрыть) в Конструкторе таблиц. Любое из этих действий приведет к открытию ОД с запросом о сохранении изменений. Нажмите **Yes** (Да).

**Примечание.** После нажатия кнопки **Yes** (Да), может появиться ОД, запрашивающее имя таблицы, если только это имя не было введено в ОД **Table Properties** (Свойства таблицы).

10. Таблица сохраняется. Происходит возврат в SSMS.
11. Щелкните ЛПКМ по созданной таблице **Customers** и в контекстном меню выберите команду **Properties** (Свойства). Можно просмотреть важные подробные данные о таблице: дата ее создания, ее имя и имя схемы, имя владельца таблицы. В одном из разделов указываются текущие соединения с таблицей.

## Создание таблицы в Query Editor

Для выполнения задания среда SSMS должна быть активной. Если необходимо, выполните:

**Start → Programs → Microsoft SQL Server 20XX → SQL Server Management Studio**

1. В **Object Explorer** выберите БД **ApressFinancial**: щелкните ЛПКМ по узлу **ApressFinancial**.

**Примечание.** Будьте внимательны, работая в Редакторе запросов, вы можете создать объект в другой базе данных! Желательно использовать в коде инструкцию **USE**.

2. На ПИ Стандартная нажмите кнопку **New Query** (Создать запрос). Откроется **Query Editor** (Редактор запросов).

3. В окне запроса введите следующий код T-SQL:

```
USE ApressFinancial
GO
CREATE TABLE TransactionDetails.Transactions (
    TransactionId bigint IDENTITY (1,1) PRIMARY KEY NOT NULL,
    CustomerId bigint NOT NULL,
    TransactionType int NOT NULL,
    DateEntered datetime NOT NULL,
    Amount numeric(18,5) NOT NULL,
    ReferenceDetails nvarchar(50) NULL,
    Notes nvarchar(max) NULL,
    RelatedShareId bigint NULL,
    RelatedProductId bigint NOT NULL )
```

4. Выполните код. На панели **Results** (Результаты) должно появиться следующее сообщение:

Command(s) completed successfully.

(Выполнение команд успешно завершено).

5. Может появиться сообщение об ошибке. Это может произойти по целому ряду причин: начиная с ошибки при печати кода и заканчивая отсутствием разрешения на создание таблиц. Генерируемое сообщение об ошибке, как правило, понятно без пояснения. Например, нижеследующее сообщение информирует, что при наборе сделана ошибка в строке 5:

Msg 102, Level 15, State 1, Line 5  
Incorrect syntax near 'NUL'.

6. Перейдите в **Object Explorer** (Обозреватель объектов), обновите узел **Tables** (Таблицы) и проверьте: должна появиться таблица **TransactionDetails.Transactions**.

## Создание таблицы с помощью шаблона

В SQL Server существует еще один (третий) способ создания таблиц: с помощью шаблона. Может быть, это не самый удобный способ, но для повседневных задач в SSMS встроено большое количество шаблонов. Также можно создавать собственные шаблоны для повторяющихся задач.

1. Откройте окно шаблонов в SSMS:  
**View** (Вид) → **Template Explorer** (Обозреватель шаблонов) или <Ctrl>+<Alt>+<T>  
Это окно отображается в правой области SSMS.
2. В **Template Explorer** (Обозреватель шаблонов) раскройте узел **Table** и найдите шаблон с именем **Create Table**

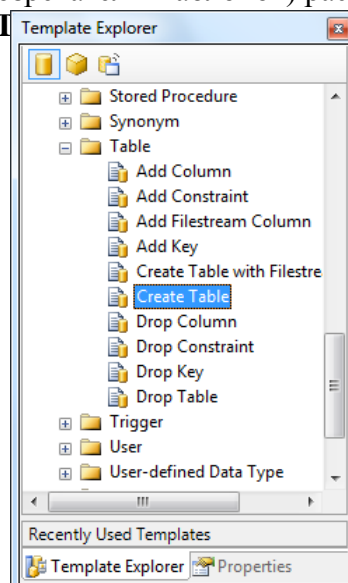


Рис. 3.3 ОД Template Explorer (Обозреватель шаблонов) с выделенным шаблоном Create Table.

- Щелкните 2ЛКМ по шаблону **Create Table**. Откроется новая панель **Query Editor** (Редактор запросов) с шаблоном для создания таблицы.
- Изучите приведенный ниже листинг шаблона. Он включает несколько параметров, которые заключены в угловые скобки (<>):

```
-- =====
-- Create table template
-- =====
USE <database,>sysname, AdventureWorks>
GO

IF OBJECT_ID('<schema_name, sysname, dbo>.<table_name, sysname, sample_table>', 'U') IS NOT NULL
    DROP TABLE <schema_name,>sysname, dbo>.<table_name,>sysname, sample_table>
GO

CREATE TABLE <schema_name,>sysname, dbo>.<table_name,>sysname, sample_table>
(
    <columns_in_primary_key,, c1><column1_datatype,,int><column1_nullability,, NOT NULL>,
    <column2_name,>sysname, c2><column2_datatype,,char(10)><column2_nullability,, NULL>,
    <column3_name,>sysname, c3><column3_datatype,,datetime><column3_nullability,, NULL>,
    CONSTRAINT <constraint_name,>sysname, PK_sample_table> PRIMARY KEY
    (<columns_in_primary_key,, c1>)
)
GO
```

- Для изменения параметров и превращения шаблона в осмысленный код необходимо нажать <Ctrl>+<Shift>+<M> и в открывшемся окне **Specify Values for Template Parameters** заполнить столбец **Value** (см. рис.3.4)

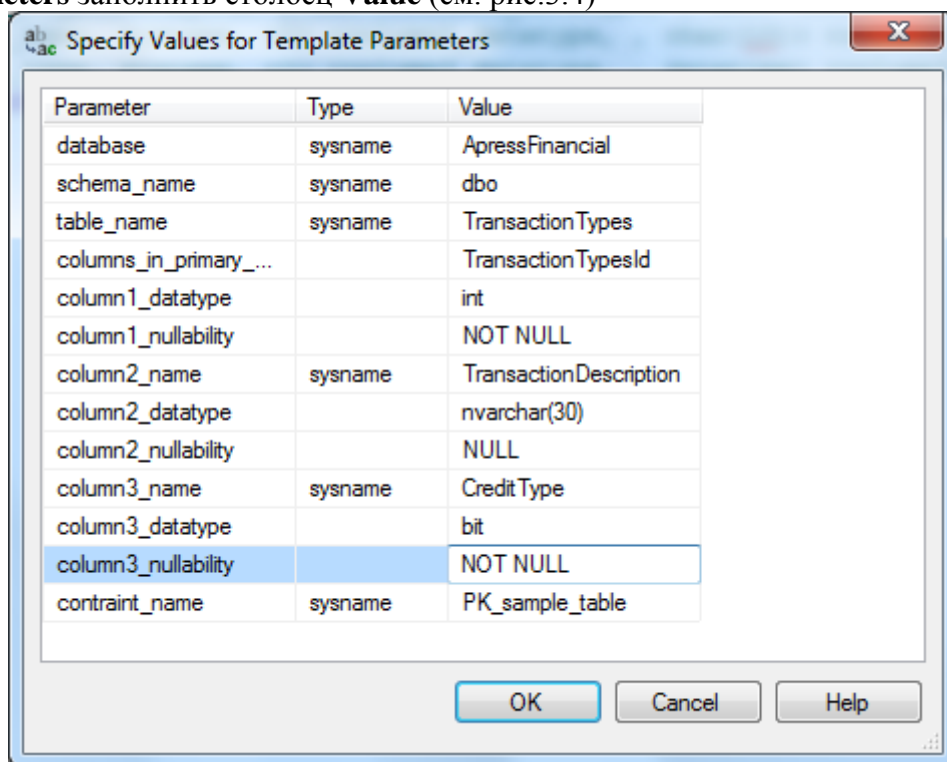


Рис. 3.4. Параметры шаблона для таблицы **TransactionTypes**

- После нажатия кнопки **OK**, в окне **Query Editor** (Редактора запросов) вы получите следующий код:

```
-- =====
-- Create table template
-- =====
USE ApressFinancial
GO

IF OBJECT_ID('dbo.TransactionTypes','U') IS NOT NULL
```

```

DROP TABLE dbo.TransactionTypes
GO

CREATE TABLE dbo.TransactionTypes
(
    TransactionTypesId int NOT NULL,
    TransactionDescription nvarchar(30) NULL,
    CreditType bit NOT NULL,
    CONSTRAINT PK_sample_table PRIMARY KEY (TransactionTypesId)
)
GO

```

7. Перед выполнением код необходимо откорректировать и дополнить следующим образом:

```

-- =====
-- Create table template
-- =====
USE ApressFinancial
GO

IF OBJECT_ID('TransactionDetails.TransactionTypes','U') IS NOT NULL
    DROP TABLE TransactionDetails.TransactionTypes
GO

CREATE TABLE TransactionDetails.TransactionTypes
(
    TransactionTypesId int IDENTITY (1,1) NOT NULL,
    TransactionDescription nvarchar(30) NOT NULL,
    CreditType bit NOT NULL
)
GO

```

8. Теперь код можно выполнить. Проверьте, что в БД **ApressFinancial** появилась третья таблица **TransactionDetails.TransactionTypes**.

## Изменение таблицы

---

В предыдущем задании с помощью шаблона была создана таблица с тремя столбцами. Необходимо добавить столбец. Если таблица была создана с ошибками, то можно таблицу удалить с помощью инструкции **DROP TABLE** и создать заново. Но если в таблицу уже поместили какие-либо данные, то такое решение не подходит. Есть альтернативный способ: инструкция **ALTER TABLE**, которая допускает ограниченные(!) преобразования формата и при этом сохраняет ее содержимое.

1. Добавьте столбец в таблицу. Нажмите кнопку **New Query** (Создать запрос). Откроется **Query Editor** (Редактор запросов).
2. В окне запроса введите следующий код T-SQL:

```

USE ApressFinancial
GO

ALTER TABLE TransactionDetails.TransactionTypes
ADD AffectCashBalance bit NULL
GO

```

**Примечание.** *Так как мы не хотим, чтобы существующие данные принимали значения по умолчанию, то при добавлении столбца следует допустить значения NULL.*

3. Выполните код. На панели **Results** (Результаты) должно появиться следующее сообщение:

Command(s) completed successfully.

(Выполнение команд успешно завершено).

Ваша таблица пока не содержит никаких данных. Но если бы данные были, то после добавления столбца можно было бы внести в них изменения, а затем отменить разрешение на значение NULL.

4. Изменим добавленный столбец: он обязательно должен содержать значения 0 или 1.  
Для этого введите и выполните следующий код T-SQL:

```
USE ApressFinancial
GO

ALTER TABLE TransactionDetails.TransactionTypes
ALTER COLUMN AffectCashBalance bit NOT NULL
GO
```

5. Добавим первичный ключ.

```
USE ApressFinancial
GO

ALTER TABLE TransactionDetails.TransactionTypes
ADD CONSTRAINT
    PK_TransactionTypes PRIMARY KEY NONCLUSTERED (TransactionTypesId)
    WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
        ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
GO
```

6. Теперь таблица **TransactionDetails.TransactionTypes** стала правильной.

## Создание остальных таблиц

Необходимо к трем созданным таблицам добавить еще четыре.

Для этого в окне запроса введите и выполните следующий код T-SQL:

```
USE ApressFinancial
GO

CREATE TABLE CustomerDetails.CustomersProducts
(
    CustomerFinancialProductID bigint IDENTITY (1,1) NOT NULL,
    CustomerId bigint NOT NULL,
    FinancialProductID bigint NOT NULL,
    AmountToCollect money NOT NULL,
    Frequency smallint NOT NULL,
    LastCollected datetime NOT NULL,
    LastCollection datetime NOT NULL
    Renewable bit NOT NULL,
) ON [PRIMARY]
GO
```

```
CREATE TABLE CustomerDetails.FinancialProducts
(
    ProductID bigint NOT NULL,
    ProductName nvarchar(50) NOT NULL
) ON [PRIMARY]
GO
```

-- Создадим схему ShareDetails и в ней таблицу SharePrices одной инструкцией!

[illegible]

```

PriceDate datetime NOT NULL
)

GO

CREATE TABLE ShareDetails.Shares
(
    ShareID bigint IDENTITY (1,1) NOT NULL,
    ShareDesc nvarchar(50) NOT NULL,
    ShareTickerID nvarchar(50) NULL,
    CurrentPrice numeric(18,5) NOT NULL
) ON [PRIMARY]
GO

```

**Примечание.** Обратите внимание, что в версии SQL Server одной (!) инструкцией **CREATE SCHEMA** создается не только сама схема, но и объекты, содержащиеся внутри создаваемой схемы - таблицы и представления, а также разрешения **GRANT**, **REVOKE** или **DENY** на любой защищаемый объект. Эта инструкция должна выполняться как отдельный пакет. Для выполнения требуется разрешение **CREATE SCHEMA** в базе данных, а чтобы создать объект, указанный в инструкции **CREATE SCHEMA**, у пользователя должно быть соответствующее разрешение **CREATE**. Транзакции **CREATE SCHEMA** являются атомарными. Если в процессе выполнения инструкции **CREATE SCHEMA** возникают ошибки, ни один из указанных защищаемых объектов не создается, и ни одно разрешение не предоставляется.

**Примечание.** В создаваемых таблицах не определены первичные ключи: они будут определены далее с использованием дополнительных возможностей среды SSMS.

## Создание отношения в Table Designer

---

1. Если необходимо, запустите SSMS:  
**Start → Programs → Microsoft SQL Server 20XX → SQL Server Management Studio**
2. В **Object Explorer** раскройте узел **Databases**, чтобы увидеть БД **ApressFinancial**.
3. Раскройте БД **ApressFinancial** до узла **Tables** (Таблицы).  
Раскройте узел **Tables** (Таблицы).
4. Найдите и выделите таблицу **TransactionDetails.Transactions**.  
Щелкните по этой таблице ПКМ и в контекстном меню выберите команду **Design**, чтобы открыть таблицу в **Table Designer** (Конструктор таблиц).
5. В окне **Table Designer** (Конструктор таблиц) щелкните ПКМ по верхней части и из контекстного меню выберите строку **Relationships...** (Отношения), или нажмите кнопку **Relationships** (Отношения) на ПИ **Table Designer**.  
Откроется конструктор отношений **Foreign Key Relationships** (Связи по внешнему ключу).
6. Окно пустое, необходимо нажать кнопку **Add** (Добавить). Экран заполнится.
7. В группе (**General**) раскройте узел **Tables And Columns Specified** (Спецификация таблиц и столбцов). Справа от узла найдите и нажмите кнопку с многоточием (см. рис.3.5).

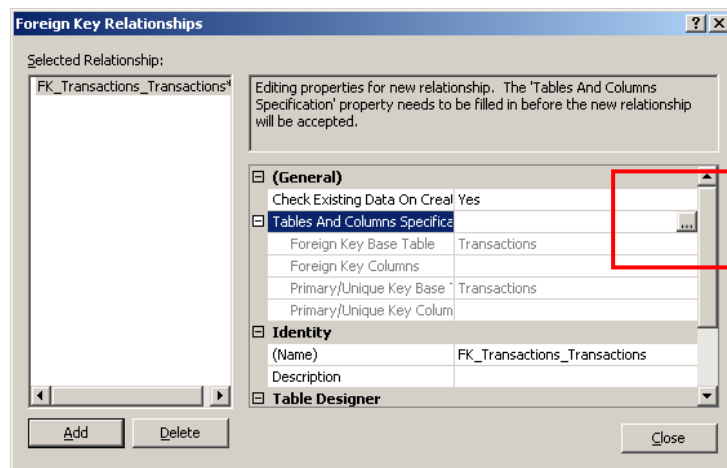


Рис. 3.5 ОД Foreign Key Relationships.

8. В открывшемся ОД **Tables And Columns** (Таблицы и Столбцы) сразу измените имя создаваемого отношения: в поле **Relationship Name** введите **FK\_Customers\_Transactions**.
9. В поле со списком **Primary key Table** выберите таблицу **Customers (CustomerDetails)**.
10. В поле **Foreign key Table** указана таблица **Transactions (TransactionDetails)**. Ниже поля **Foreign key Table** выберите из выпадающего списка поле **CustomersId**  
**Примечание.** В обеих таблицах поля имеют одинаковые имена, но это необязательно. Главное, чтобы хранимые данные в этих столбцах были одинаковыми!
11. Нажмите кнопку **ОК**. Снова появится окно **Foreign Key Relationships** (Связи по внешнему ключу). Теперь в этом окне можно видеть подробные данные только что определенного внешнего ключа.
12. Параметры в группе **Table Designer** оставьте по умолчанию.

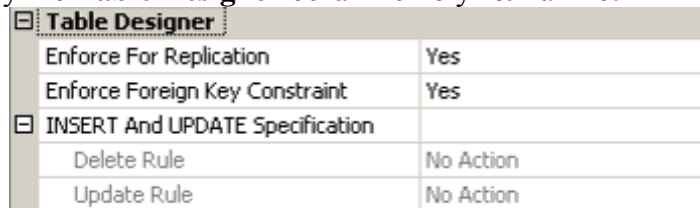


Рис. 3.6 Значения по умолчанию в группе Table Designer.

**Примечание.** При закрытии окна **Foreign Key Relationships** кнопкой **Close** изменения не сохраняются. Только после закрытия окна **Table Designer** (Конструктор таблиц) появится ОД, уведомляющее о том, что будут изменены две таблицы.

13. Закройте окно **Table Designer** (Конструктор таблиц) и сохраните изменения в обеих таблицах.

## Создание отношения с помощью кода T-SQL

В окне запроса введите и выполните следующий код T-SQL:

```
USE ApressFinancial
GO

ALTER TABLE TransactionDetails.Transactions
WITH NOCHECK
ADD CONSTRAINT FK_Transactions_Shares FOREIGN KEY (RelatedShareId)
REFERENCES ShareDetails.Shares (ShareID)
GO
```

Должно появиться сообщение об ошибке. Проанализируйте его, выполните необходимые действия, чтобы исправить ошибку, и выполните код еще раз, чтобы создать отношение.