

## СОЗДАНИЕ ТАБЛИЦ БАЗЫ ДАННЫХ С ПОМОЩЬЮ КОНСТРУКТОРА ТАБЛИЦ

**Задание:** Изучить материал, создать таблицы с помощью конструктора по своей итоговой схеме данных из лабораторной работы 2, объединить таблицы в схему данных, используя мастер диаграмм и заполнить таблицы. Все поместить в отчет.

Как выполнять:

1. Сначала создать проекты таблиц, задать ключи
2. Затем объединить в схему данных
3. Сначала заполнить родительские таблицы
4. Затем заполнить дочерние таблицы

В СУБД MS SQL Server у каждого столбца, локальной переменной, выражения и параметра есть определенный тип данных. Тип данных — атрибут, определяющий, какого рода данные могут храниться в объекте: целые числа, символы, данные денежного типа, метки времени и даты, двоичные строки и так далее.

При назначении типа данных объекту определяются следующие атрибута объекта:

- а) вид данных, содержащихся в объекте;
- б) размер или длина хранимого объектом значения;
- в) точность числа (только в случае численных типов);
- г) масштаб числа (только в случае численных типов).

SQL Server предоставляет набор системных типов данных, определяющих все типы данных, которые могут использоваться в нем. Можно также определять собственные типы данных в Transact-SQL или Microsoft.NET Framework. Псевдонимы типов данных основываются на системных типах. Дополнительные сведения о псевдонимах типов данных см. в разделе Работа с псевдонимами типов данных. Пользовательские типы данных обладают свойствами, зависящими от методов и операторов класса, который создается для них на одном из языков программирования, которые поддерживаются .NET Framework.

Типы данных в SQLServer объединены в следующие категории:

- а) точные числа;
- б) символьные строки;
- в) символьные строки в Юникоде;
- г) приблизительные числа;
- д) двоичные данные;
- е) дата и время;
- ж) прочие типы данных.

Таблица – Сводная таблица типов данных

Наименование	Описание типа данных
Двоичные данные	
binary [ ( n ) ]	максимальная длина 8 000 байт (n). Содержит последовательность нулей и единиц
varbinary [ ( n ) max]	данные переменной длины, максимальная длина 8 000 байт (n) Содержит последовательность нулей и единиц
Image	максимальная длина 2 147 483 647 байт
Bit	тип данных, который принимает значения 1 или 0
Символьные данные	
char [(n)]	максимальная длина 8 000 символов (n)
varchar [(n)]	тип переменной длины, максимально 8 000 символов (n)
Text	максимальная длина 1 073 741 823 символов
Символьные данные в кодировке Unicode	
nchar (n)	максимальная длина 4 000 символов (n)
nvarchar (n)	переменной длины в кодировке Unicode максимальная длина 4 000 символов (n)
Ntext	максимальная длина 1 073 741 823 символов
Числовые целые данные (Точные числа)	
Bigint	диапазон от -922 337 203 685 4775808 до 922 337 203 685 4775807
Int	диапазон от -2 147 483 648 до 2 147 483 647
Smallint	диапазон от - 32 768 до 32 767
Tinyint	диапазон от 0 до 255
Числовые данные с дробной частью числа (Приблизительные числа)	
decimal[(p[, s])]	диапазон от -1038-1 до 1038-1 с задание фиксированного количества знаков (p - всего и s - дробной части), максимальное общее количество знаков 38
Numeric	то же, что и decimal
float [ ( n ) ]	диапазон от +2.29*10-308 до +1.79*10308
Real	числа с 7-значной точностью в диапазоне от +1.18*10-38 до +3.40*1038.
Тип дата и время	
Datetime	диапазон от 1.01.1753 до 31.12.9999 с точностью 3.33 мс
smalldatetime	диапазон от 1.01.1900 до 6.06.2079 с точностью 1 мин.
Денежный тип (Точные числа)	
money	диапазон от 7 203 685 477.5808 до +922 337 203 685 477.5807
smallmoney	диапазон от 214 748.3648 до +214 748.3647
Данные специальных типов	
timestamp	счетчик, автоматически увеличивающийся, имеющий уникальное значение для базы данных (тип binary(8) или varbinary(8))
uniqueidentifier	тип, который содержит уникальный идентификационный номер (GUID), сохраняемый как 16-битная двоичная строка
sql_variant	тип, который сохраняет значения различных типов, кроме text, ntext, timestamp и sql_variant.
Sysname	тип - синоним nvarchar , используется для ссылок на имена объектов базы данных

В системе SQL-сервера имеется поддержка пользовательских типов данных. Они могут использоваться при определении какого-либо специфического или часто употребляемого формата.

Создание пользовательского типа данных с помощью обозревателя объектов. Раскройте в обозревателе объектов по очереди узел **Базы данных**, узел конкретной базы данных, узел **Programmability** и **Types**, щелкните правой кнопкой мыши узел **User-DefinedDataTypes** и выберите пункт **NewUser-DefinedDataTypes**.

В диалоговом окне **Создание NewUser-DefinedDataTypes** введите в поле **Schema** схему, которая будет владеть новым типом данных, или выберите схему, нажав кнопку обзора.

В поле **Name** введите имя нового типа данных.

В поле Тип данных выберите тип, на основе которого будет создан новый тип данных.

Заполните поля **Storage(Длина)**, **Precision (Точность)**, если это требуется.

Если новый тип данных должен поддерживать значения NULL, установите флажок **AllowsNULLs**.

Если нужно связать с новым типом данных умолчание или правило, заполните в области **Biding** поле **Default** или **Rule**. Создать умолчания и правила в среде Среда SQLServerManagementStudio нельзя. Используйте для этого инструкции языка Transact-SQL. Примеры кода, создающего умолчания и правила, доступны в окне обозревателя шаблонов.

Создание пользовательского типа данных с помощью SQL запроса. Необходимо вызвать процедуру по имени `sp_addtype`. Для вызова хранимой процедуры используется ключевое слово `Exec`, либо `Execute`.

```
sp_addtype [@typename=]type,[@phystype=]  
system_data_type  
[,@nulltype=] 'null_type'
```

Пример.

```
Execute sp_addtypeMyStringvarchar(25) not null;
```

Тем самым будет создан собственный тип данных с именем `MyString`, представляющий из себя строку `varchar` длиной 25 символов и столбец этого типа данных не может быть пустым (пропуск записи значения в это поле приведет к ошибке), если только при создании таблицы не переопределить этот параметр.

Хранимая процедура `sp_droptype` удаляет пользовательский тип данных. Пример:

```
Executesp_droptypeMyString;
```

Удалить пользовательский тип данных можно, только если ни в одной таблице текущей базы данных он не был использован.

## Создание таблиц с помощью конструктора

Для создания таблицы в EnterpriseManager нужно в требуемой базе данных вызвать контекстное меню к папке Tables, в котором необходимо выбрать пункт меню NewTable. Создание таблицы с помощью конструктора приведено на рисунок 1.

В появившемся окне необходимо заполнить данные о таблице.

**Columnname** - имя поля (столбца) создаваемой таблицы.

**Datatype** - тип данных для текущего поля.

**Length** - количество символов (если создаваемое поле текстовое), либо кол-во требуемых байт (если поле числовое либо битовое).

**AllowNULLs** - Если галочка установлена, то при заполнении данных в таблицу, это поле можно пропустить. В противном случае, оно будет обязательным для заполнения.

Для каждого поля можно установить дополнительные параметры.

**Description** - описание создаваемого поля. (Чтоб легче вспомнить, если, вдруг, забудете.)

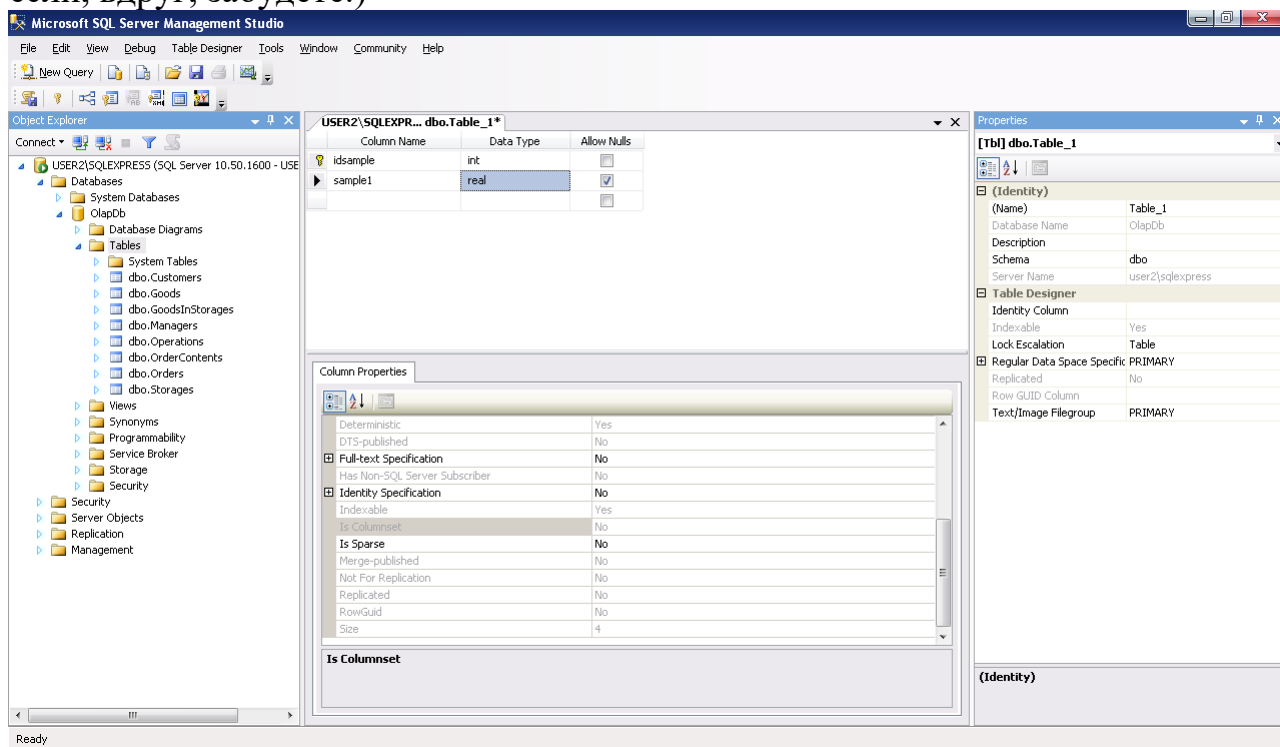


Рисунок 1 – Создание таблицы с помощью конструктора

**Defaultvalue** - значение по умолчанию. Если при заполнении таблицы пропустить поле, у которого установлен параметр по умолчанию, то записано будет значение, заданное первоначально.

**Precision** - здесь указывается общее кол-во разрядов для типа данных с плавающей точкой.

**Scale** - используется для указания требуемого кол-ва разрядов после запятой.

**Identity** - служит для задания счетчика у поля (обычно используется для первичного ключа). Счетчик представляет из себя поле, каждое последующее значение в котором увеличивается на определенную величину. Может принимать три значения:

**No** - счетчиком не является.

**Yes** - установить счетчик.

**Yes (NotForReplication)** - установить счетчик, но в репликация не использовать.

**IdentitySeed** - с какого числа начинать счетчик.

**IdentityIncrement** - на какую величину увеличивать каждое последующее значение.

**Collation** - этот параметр определяет, как должна производиться сортировка поля при вычитке данных.

После настройки всех полей можно смело нажать на закрытие окна создания таблицы и вам будет предложено сохранить вашу таблицу. Нажмите Да и введите имя для вашей таблицы.

Иногда, после создания таблицы, необходимо изменить параметры того или иного поля. Для этого необходимо выбрать пункт DesignTable в контекстном меню таблицы. Откроется окно создания таблицы.

Для этого достаточно вызвать в контекстном меню требуемой таблицы пункт меню "Удалить".

В следующем окне нужно нажать DropAll, тем самым, удалив все таблицы из списка (в том случае, если контекстное меню вызывалось к набору выделенных таблиц).

Чтобы просмотреть все содержимое таблицы выберете в контекстном меню таблицы Opentable->Returnallrows.

При работе БД должна обеспечиваться целостность данных. Под целостностью данных понимают обеспечения целостности связей между записями в таблицах при удалении записей из первичных таблиц. То есть, при удалении записей из первичных таблиц автоматически должны удаляться связанные с ними записи из вторичных таблиц.

В случае несоблюдения целостности данных со временем в БД накопится большое количество записей во вторичных таблицах связанных с несуществующими записями в первичных таблицах, что приведёт к сбоям в работе БД и её засорению неиспользуемыми данными.

Для обеспечения целостности данных в SQL Server используют диаграммы и триггеры.

**Диаграммы** - это компоненты БД, которые блокируют удаление записей из первичных таблиц если существуют связанные с ними записи во вторичных таблицах. Следовательно, диаграммы предотвращают нарушение целостности данных. В SQL Server диаграммы создаются при помощи мастера диаграмм.

## Создание диаграмм

В БД "Microsoft SQL Server" все диаграммы находятся в папке "**Database Diagrams**" обозревателя объектов (рис. 2).

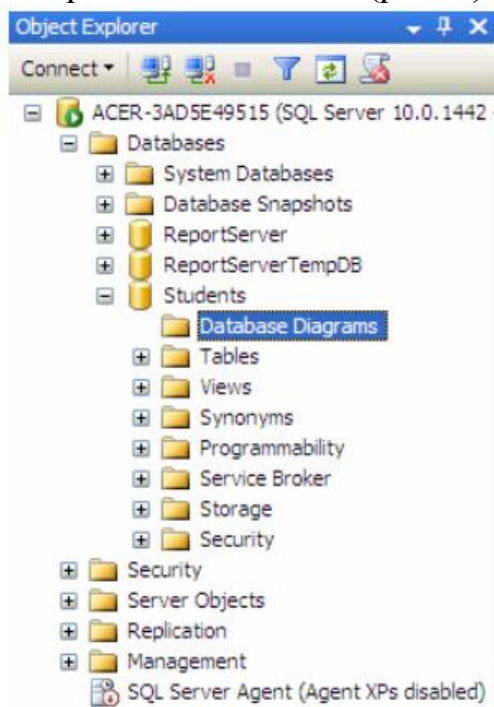


Рисунок 2

Создадим диаграмму, обеспечивающую целостность данных БД "**Students**". Для создания новой диаграммы в БД "**Students**" щелкните ПКМ по папке "**Database Diagrams**" и в появившемся меню выберем пункт "**New Database Diagram**". Сначала появится окно с вопросом о добавлении нового объекта "Диаграмма". В этом окне нужно нажать кнопку "**Yes**". Затем появится окно "**Add Table**" предназначенное для добавления таблиц в новую диаграмму (рис. 3).

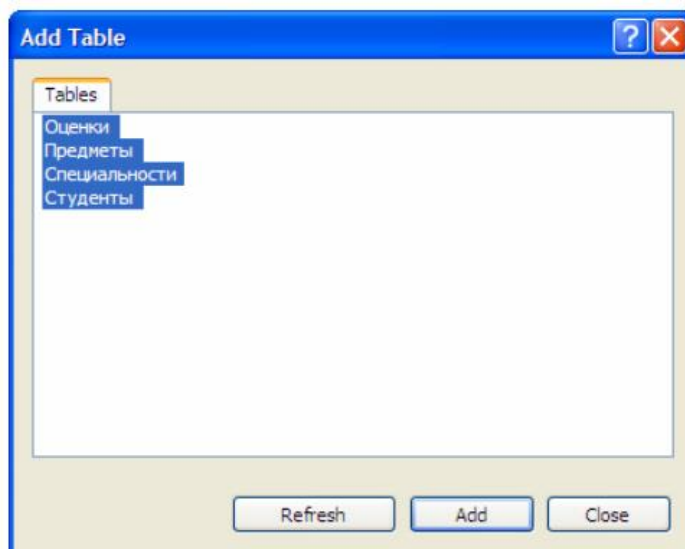


Рисунок 3

В окне добавления таблиц выделите все таблицы нашей БД и нажмите кнопку "Add". Закройте окно "Add Table" нажатием на кнопку "Close".

Появится окно диаграммы, где будут отображены отобранные таблицы. Теперь необходимо определить связи между таблицами. Перетащите поле "Код специальности" из таблицы "Специальности" на такое же поле в таблице "Студенты". Появится окно создания связи между таблицами "Tables and Columns" (рис. 4).

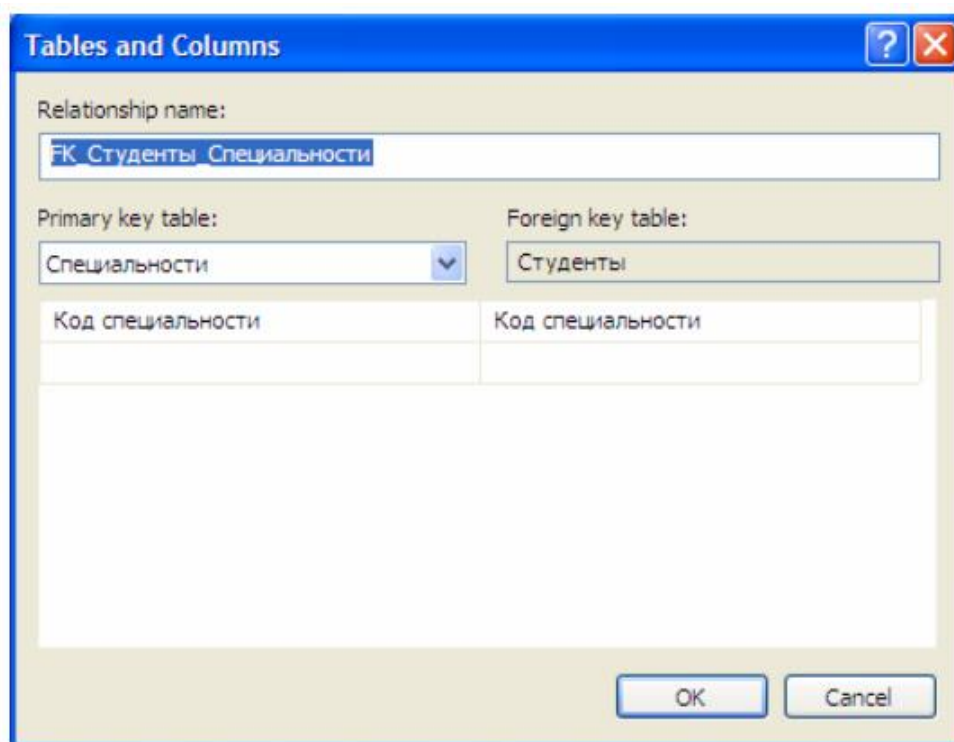


Рисунок 4

В окне создания связи нажмите кнопку "Ok". Появится окно настройки свойств связи "Foreign Key Relationship" (рис. 5).

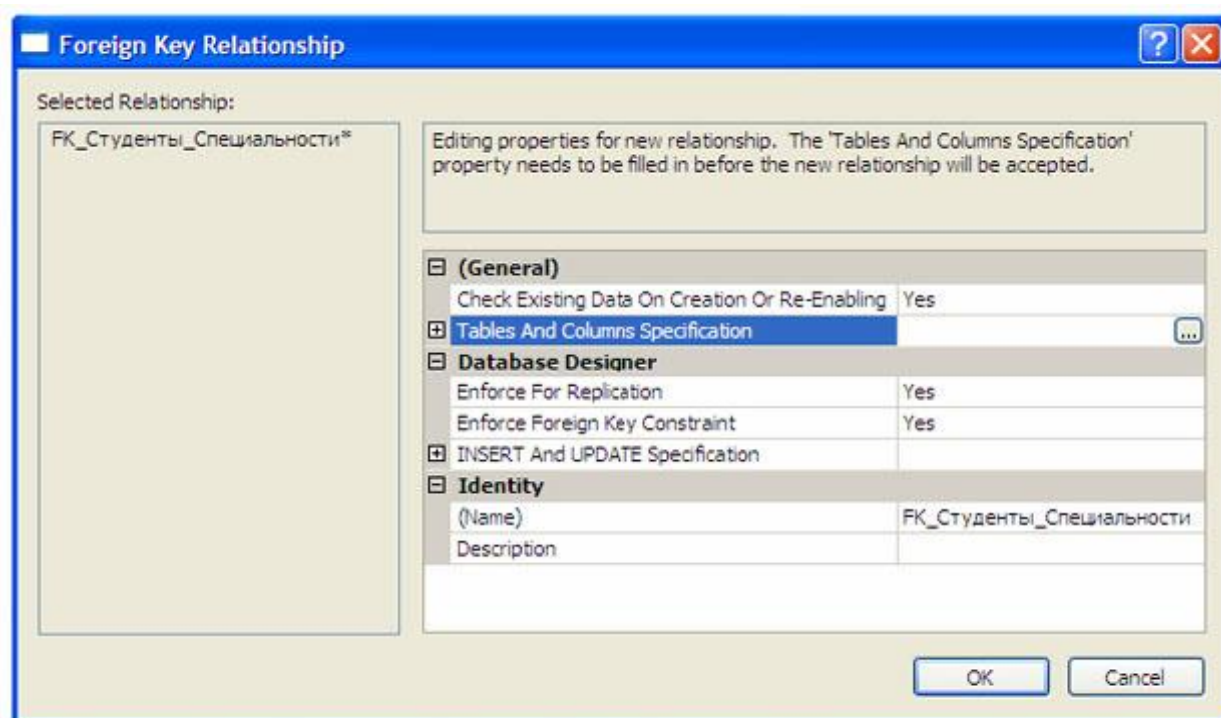


Рисунок 5

Оставьте свойства связи без изменений и в окне свойств связи нажмите кнопку "Ok". В диаграмме между таблицами "Студенты" и "Специальности" появится связь в виде ломанной линии (рис. 6).

Аналогичным образом создайте связь таблицы "Студенты" с таблицей "Оценки", перетащив поле "Код студента" из таблицы "Студенты" на одноименное поле в таблице "Оценки".

Затем, свяжите таблицы "Предметы" и "Оценки", перетащив поле "Код предмета" из таблицы "Предметы" на поля "Код предмета 1", "Код предмета 2" и "Код предмета 3" таблицы "Оценки".

После выполнения вышеперечисленных действий диаграмма примет следующий вид (рис. 6).

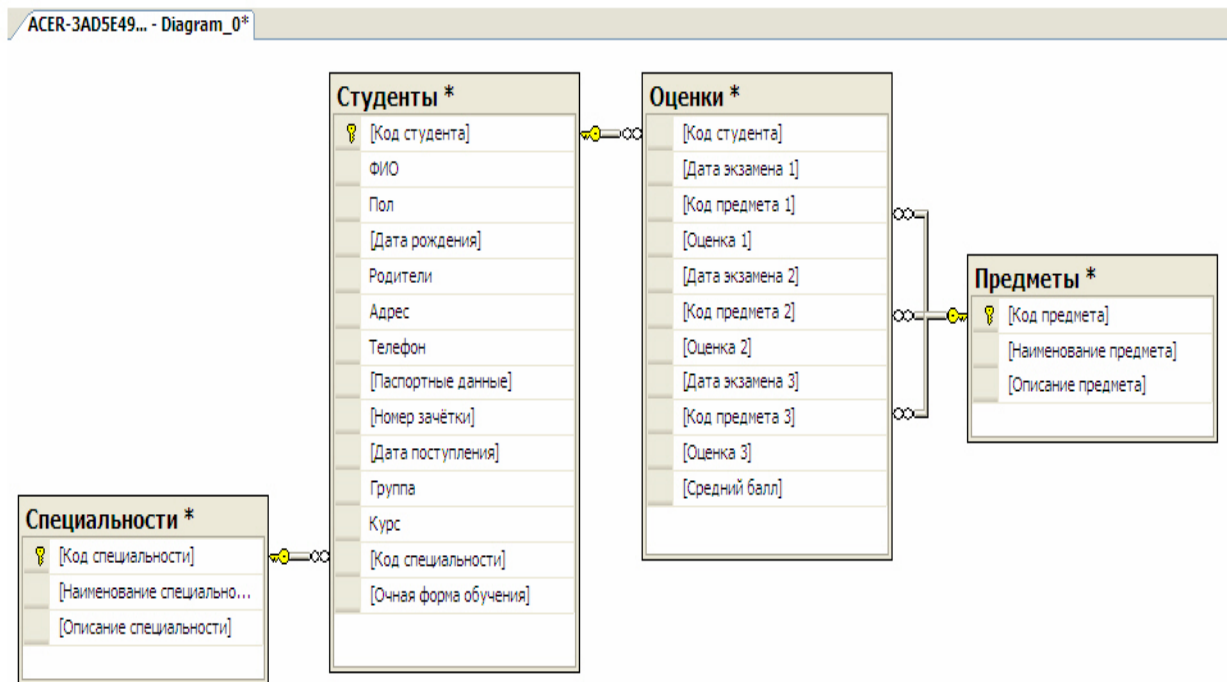



Рисунок 6

Закройте окно с диаграммой, щелкнув мышью по кнопке закрытия  расположенной в верхнем правом углу окна с диаграммой. Появится окно с вопросом о сохранении новой диаграммы, где необходимо нажать кнопку "Yes" (рис 7).

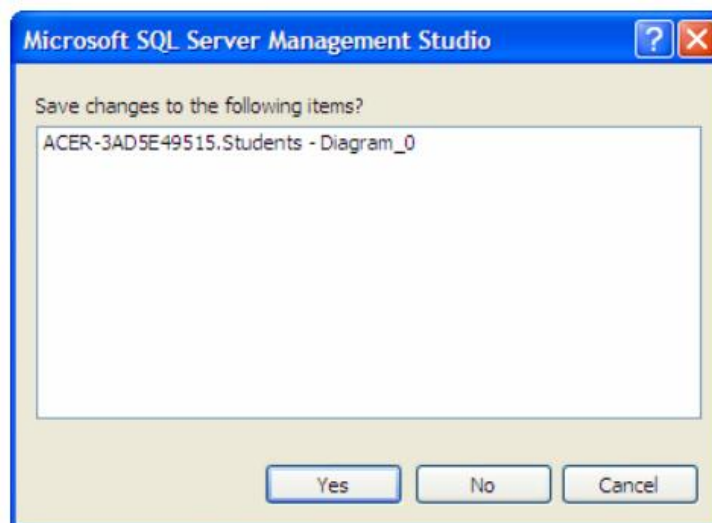


Рисунок 7

Появится окно определения имени новой диаграммы **"Choose Name"**. В окне определения имени, задайте имя диаграммы как "Диаграмма БД Студенты" и нажмите кнопку **"Ok"** (рис. 8).

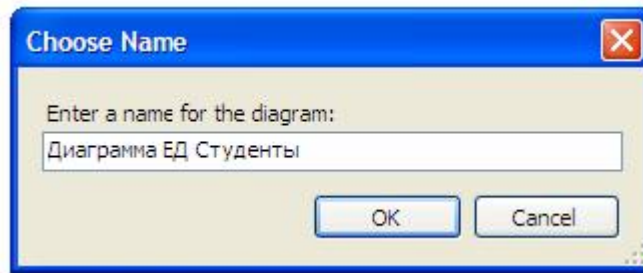


Рисунок 8

Появится окно **"Save"** с запросом сохранения таблиц, входящих в диаграмму. В данном окне необходимо нажать кнопку **"Yes"** (рис. 9).

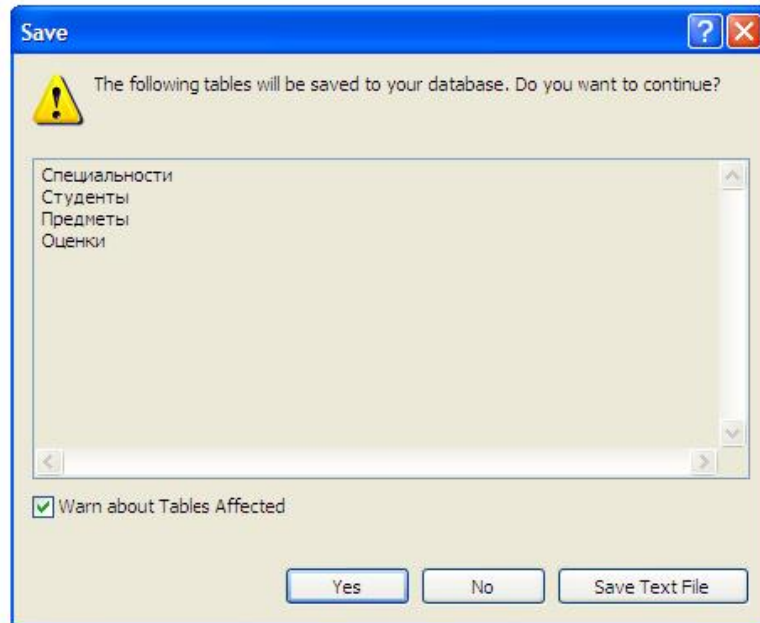


Рисунок 9

После построения диаграммы, заполните таблицы.

Вопросы на защиту:

1. Microsoft SQL Server – это?
2. MS SQL Server включает?
3. MS SQL Server – версии и их назначение?
4. Серверная часть системы MS SQL Server – службы и их назначение?
5. Клиентская часть системы MS SQL Server – утилиты и их назначение?
6. Системные базы данных и их назначение?