

Бурков А.В.

**Проектирование информационных систем по технологии
клиент – сервер в «Microsoft SQL Server 2008» и «Microsoft
Visual Studio 2008»**

Часть 2. Практический курс

Лабораторная работа 1. Установка «Microsoft SQL Server 2008»

Цель: научиться устанавливать «Microsoft SQL Server 2008»

Начнём создание БД с установки «Microsoft SQL Server 2008». Вставьте диск с сервером в привод компакт дисков вашего компьютера. Установка запустится автоматически. Компьютер будет проверен на наличие пакета «Microsoft .NET Framework». Если данный пакет не установлен, то появится окно начала установки пакета (1.1).

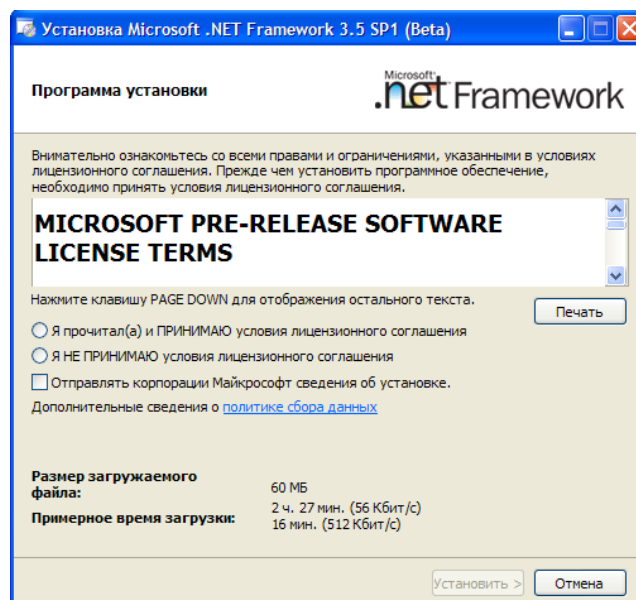


Рис.1.1

В появившемся окне представлено лицензионное соглашение об использовании пакета «Microsoft .NET Framework». Для того чтобы установить пакет необходимо согласиться с соглашением. Включите переключатель «Я прочитал(а) и ПРИНИМАЮ условия лицензионного соглашения» и нажмите кнопку «Установить».

После завершения установки появится окно с сообщением о завершении установки (Рис.1.2).

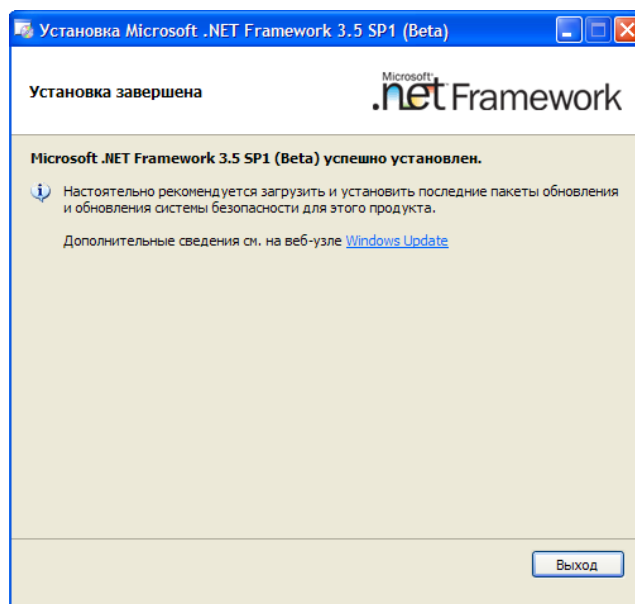


Рис.1.2

В данном окне нажмите кнопку «Выход».

Если на вашем компьютере не установлен пакет обновлений для «Windows XP», называемый «KB942288-v2», то появится следующее окно (1.3):

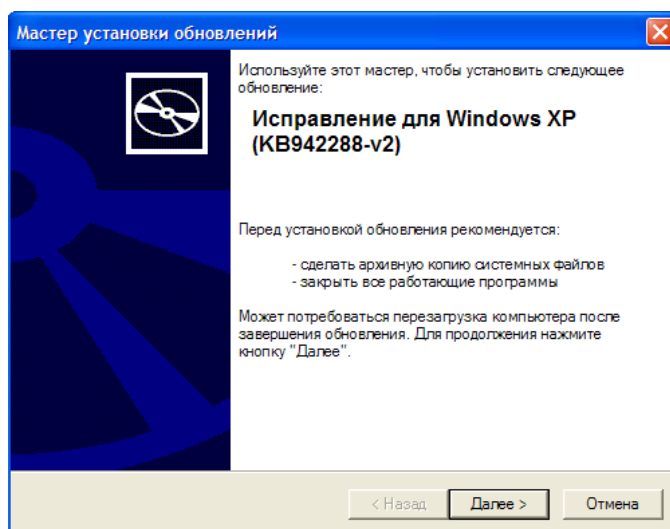


Рис.1.3

Для установки обновления нажмите кнопку «Далее». После установки обновления появится вопрос о перезагрузке компьютера (Рис.1.4).

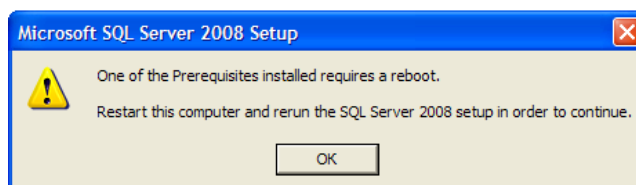


Рис.1.4

После перезагрузки компьютера появится окно выбора типа установки сервера (Рис.1.5).

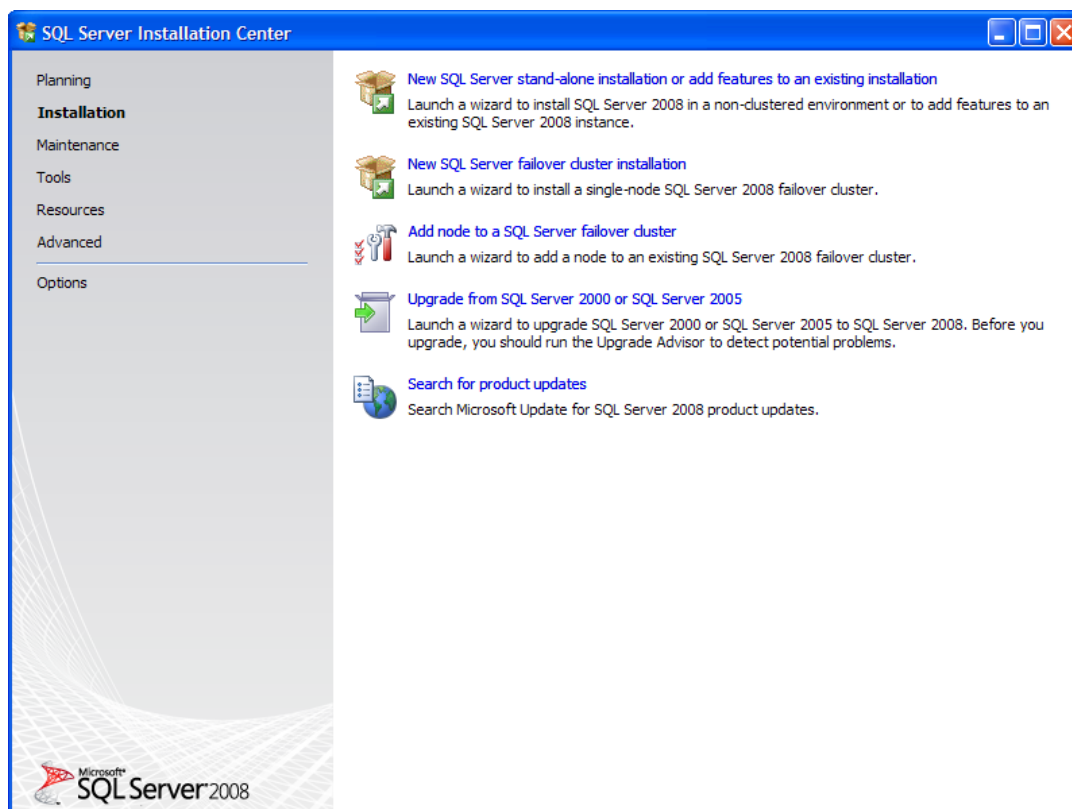


Рис.1.5

Выберите тип установки «New SQL Server stand-alone installation or add to an existing installation». Начнётся установка файлов обеспечивающих установку сервера (Рис.1.6).

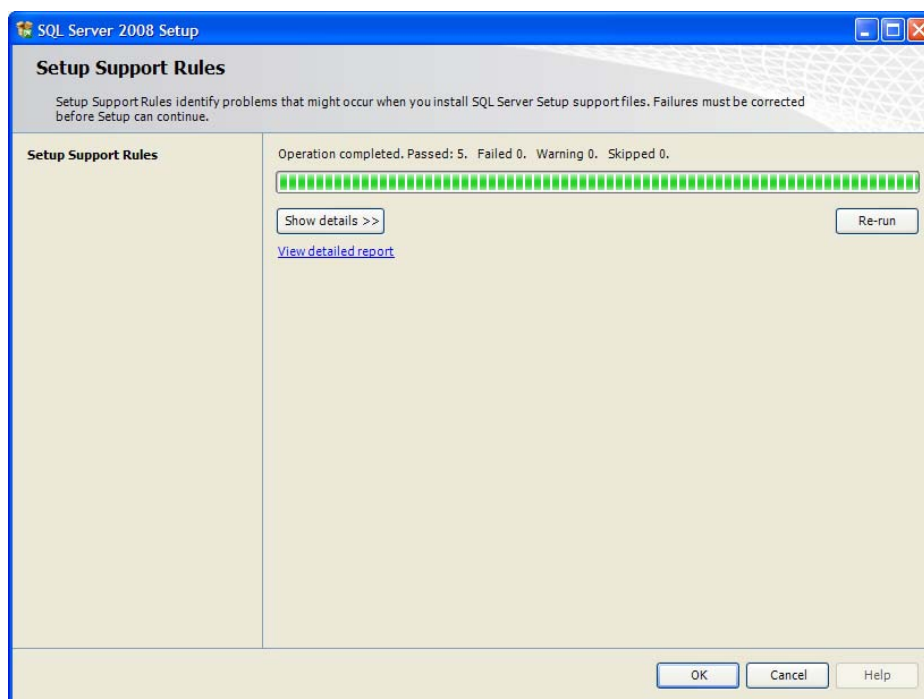


Рис.1.6

После установки вышеперечисленных файлов в окне установки нажмите кнопку «Ok». Появится окно выбора типа лицензии использования, где можно выбрать один из двух видов лицензии:

- Specify a free edition – бесплатная версия сервера, работает 180 дней, после чего требует приобретения коммерческой версии;
- Enter the product key – коммерческая версия, в поле ввода, расположенного ниже необходимо ввести ключ коммерческой лицензии продукта.

После выбора вида лицензии нажмите кнопку «Next» (Далее). Появится окно, предназначенное для ознакомления с лицензией. Прочитайте лицензионное соглашение, включите опцию «I accept the license terms» (Я согласен с условиями лицензии) и нажмите кнопку «Next» (Рис.1.7).

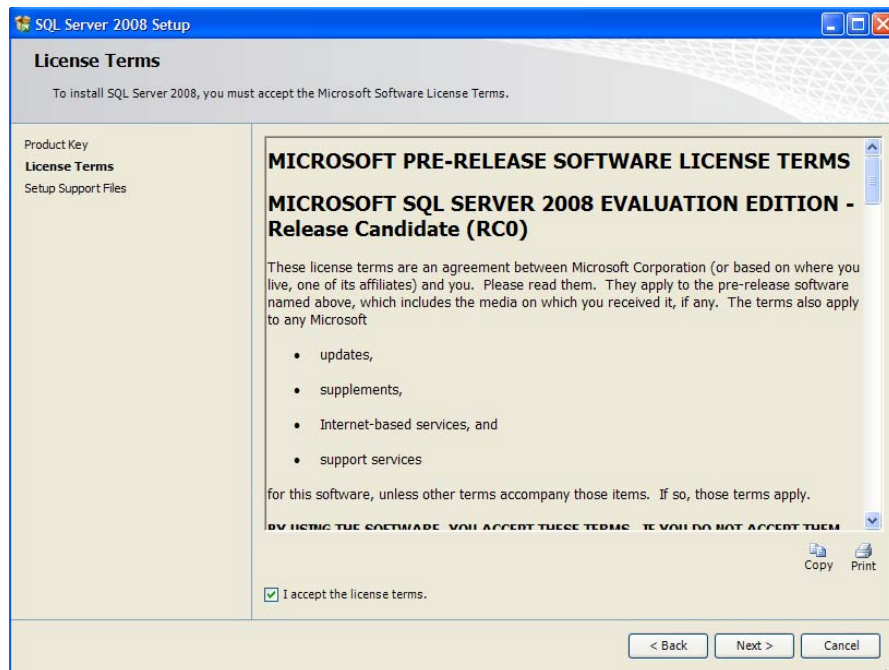


Рис.1.7

Появится окно начала установки дополнительных файлов (Рис.1.8).

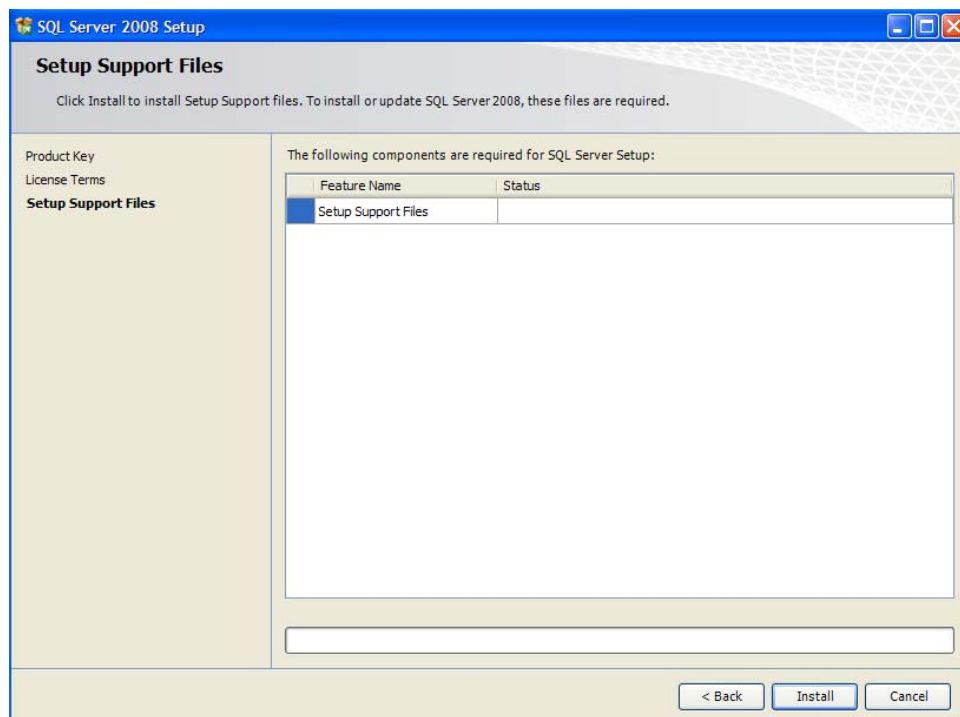


Рис.1.8

В окне начала установки дополнительных файлов нажмите кнопку «Install» (Установить). Появится окно определения правил установки дополнительных файлов (Рис.1.9).

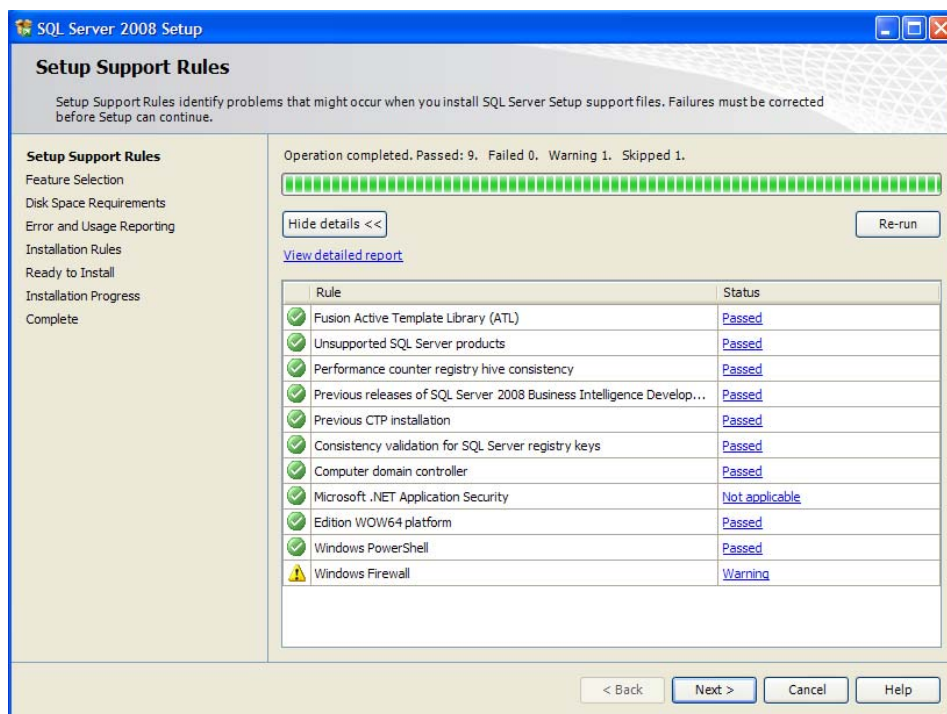


Рис.1.9

В выше представленном окне нажмите кнопку «Next». Появится окно выбора устанавливаемых компонентов сервера (Рис.1.10).

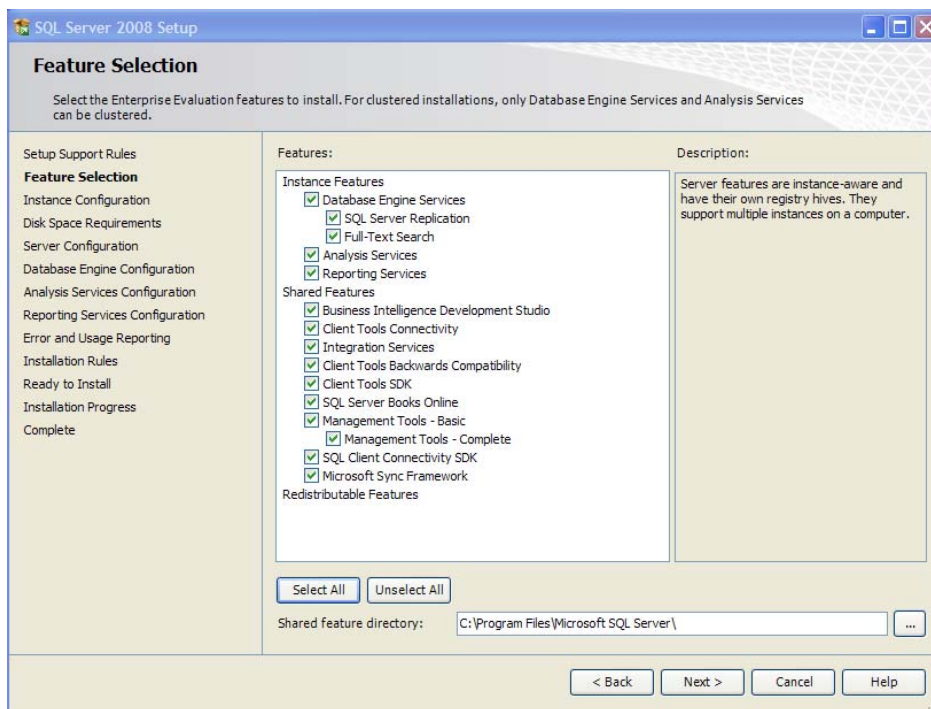


Рис.1.10

Выберите все компоненты сервера и нажмите кнопку «Next». Появится окно настройки устанавливаемого экземпляра сервера (Рис.1.11).

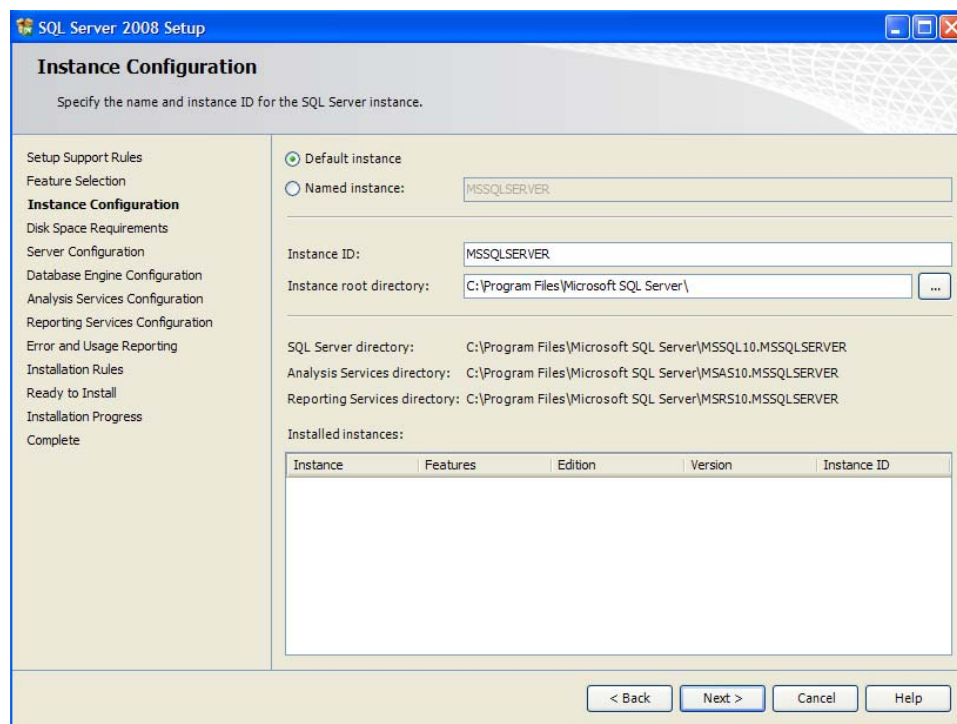


Рис.1.11

В данном окне определите следующие настройки:

- «Default instance» или «Named instance» - установка экземпляра по умолчанию или поименованного экземпляра;
- «Instance ID» – имя устанавливаемого экземпляра сервера;
- «Instance root directory» - папка на диске, куда будет установлен экземпляр сервера.

Задайте вышеописанные параметры и нажмите кнопку «Next». Появится окно с отчетом о количестве свободного места на диске, куда устанавливается экземпляр сервера (Рис.1.12)

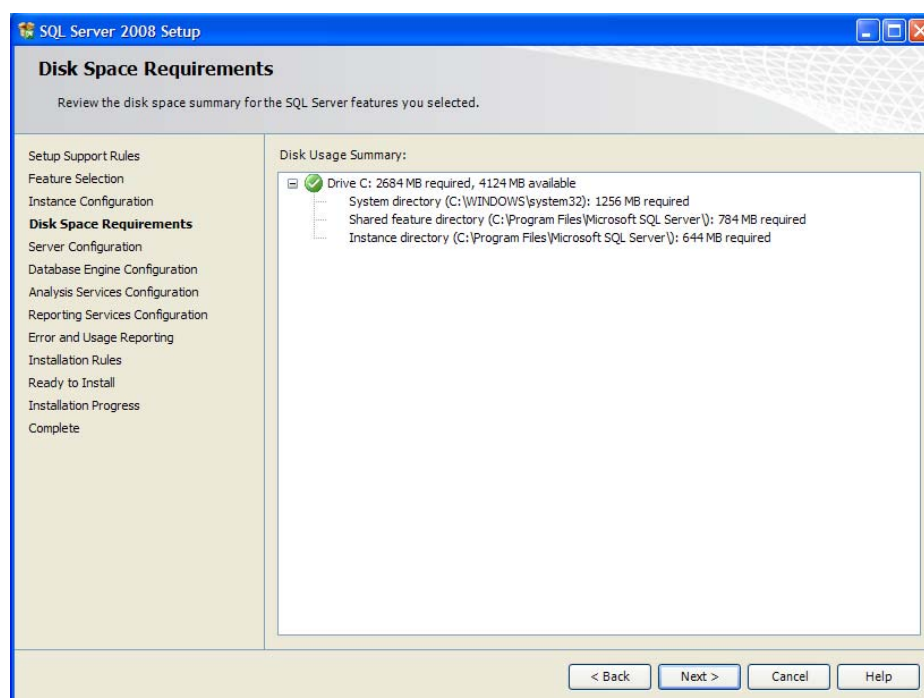


Рис.1.12

Нажмите кнопку «Next». Появится окно настройки экземпляра сервера (Рис.1.13).

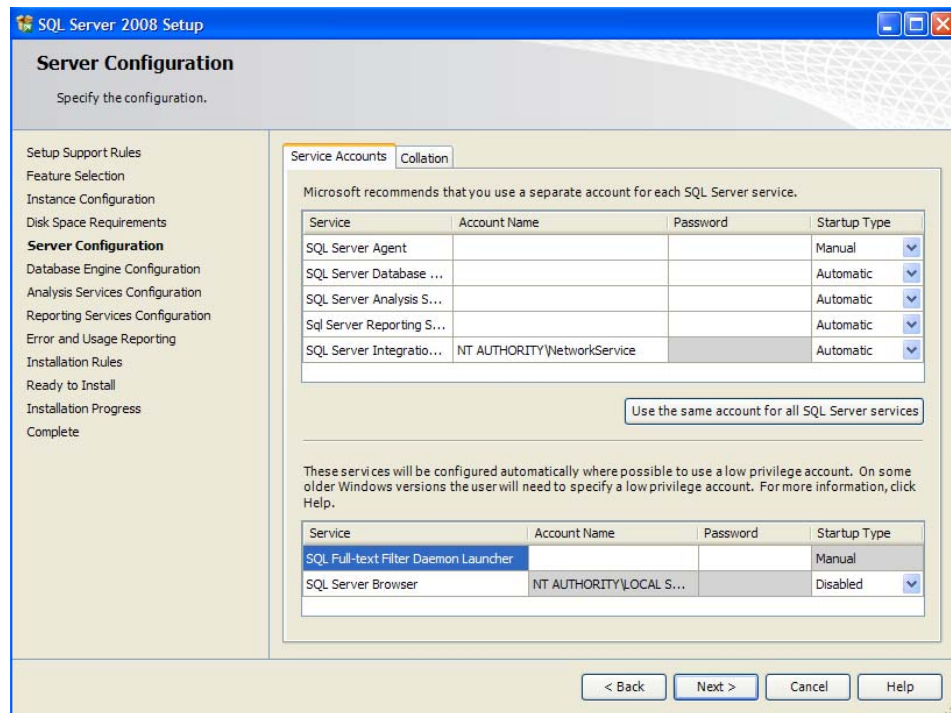


Рис.1.13

Нажмите кнопку «Next». Появится окно создания учётной записи пользователя сервера (Рис.1.14).

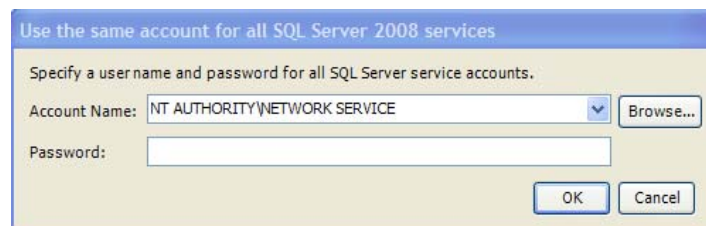


Рис.1.14

В данном окне оставьте настройки по умолчанию и нажмите кнопку «Ok». Появится окно настройки ядра сервера (Рис.1.15).

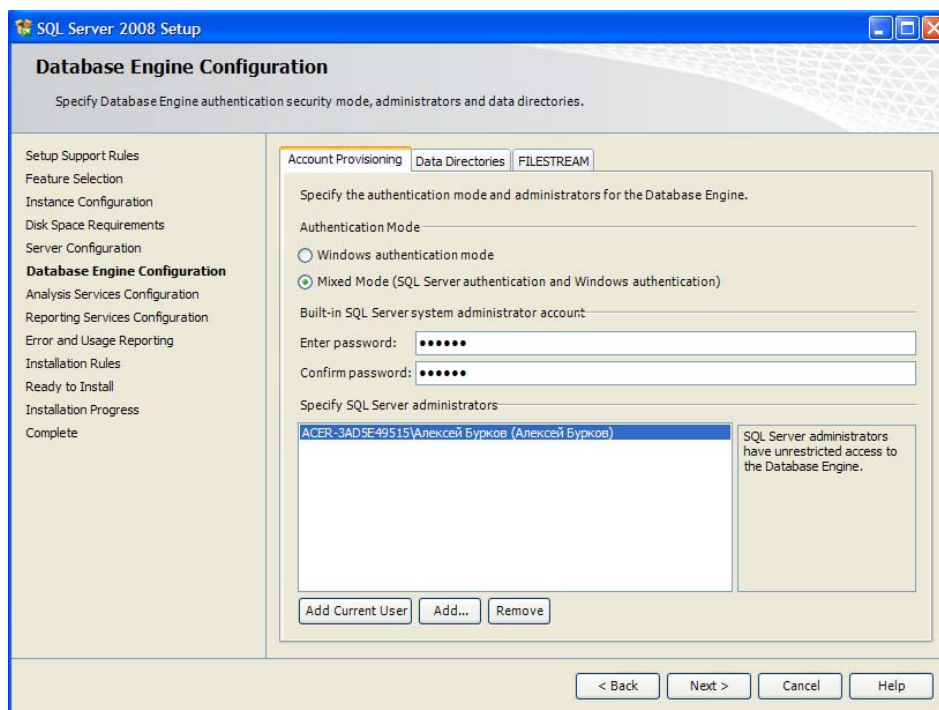


Рис.1.15

В выше представленном окне выберите смешанный режим подключения (Mixed Mode). Задайте пароль (Enter password) и его подтверждение (Confirm password). Добавьте нового администратора сервера, нажав кнопку «Add Current User» (Добавить текущего пользователя) и нажмите кнопку «Next». Появится окно для определения пользователя управляющего всеми службами сервера (Рис.1.16).

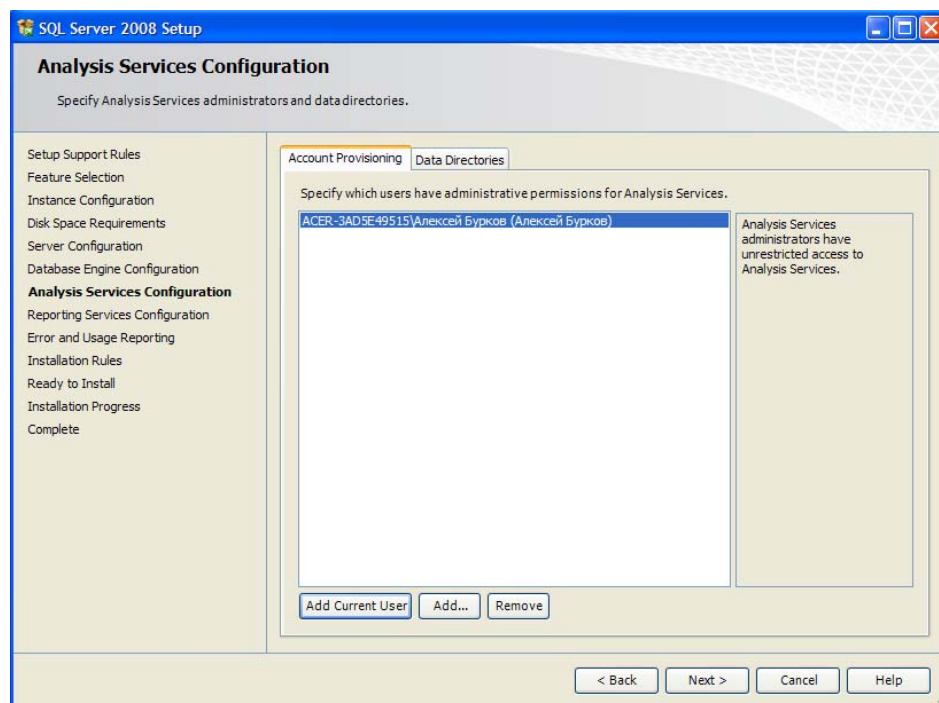


Рис.1.16

Для добавления пользователя нажмите кнопку «Add Current User», а затем нажмите кнопку «Next». Появится окно настройки сервера отчётов (Рис.1.17).

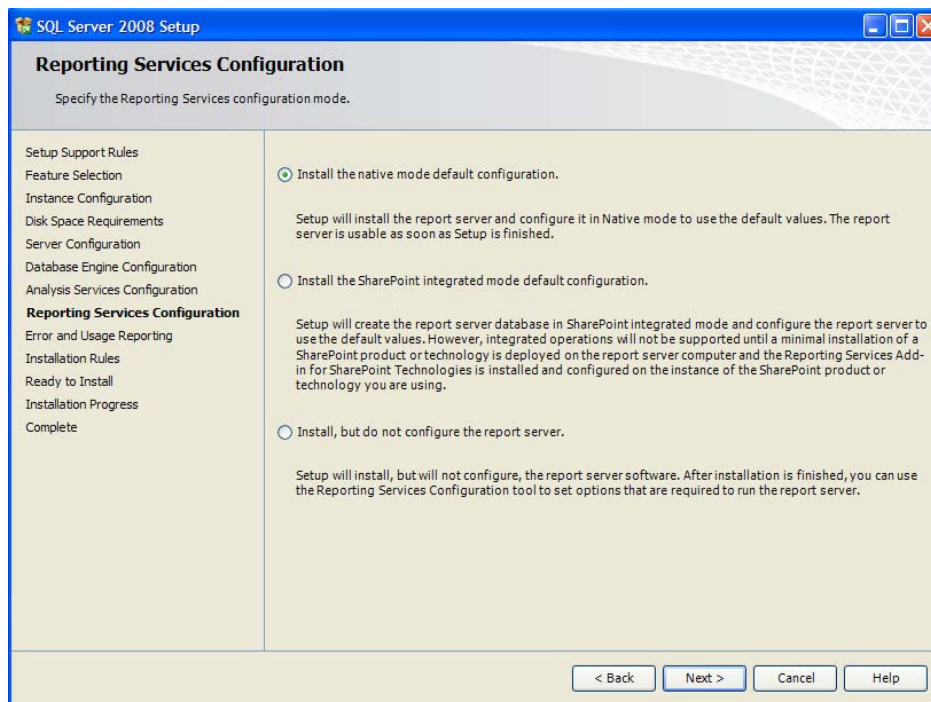


Рис.1.17

В данном окне можно выбрать один из трёх вариантов настройки отчёта:

- Install the native mode default configuration – установить настройки по умолчанию;
- Install the SharePoint integrated mode configuration – настройка отчётов для интеграции их в систему SharePoint.
- Install, but do not configure the report server – установить но не настраивать сервер отчётов.

Выберите первый вариант настройки и нажмите кнопку «Next». Появится окно настройки отчётов об ошибках в работе сервера (Рис.1.18).

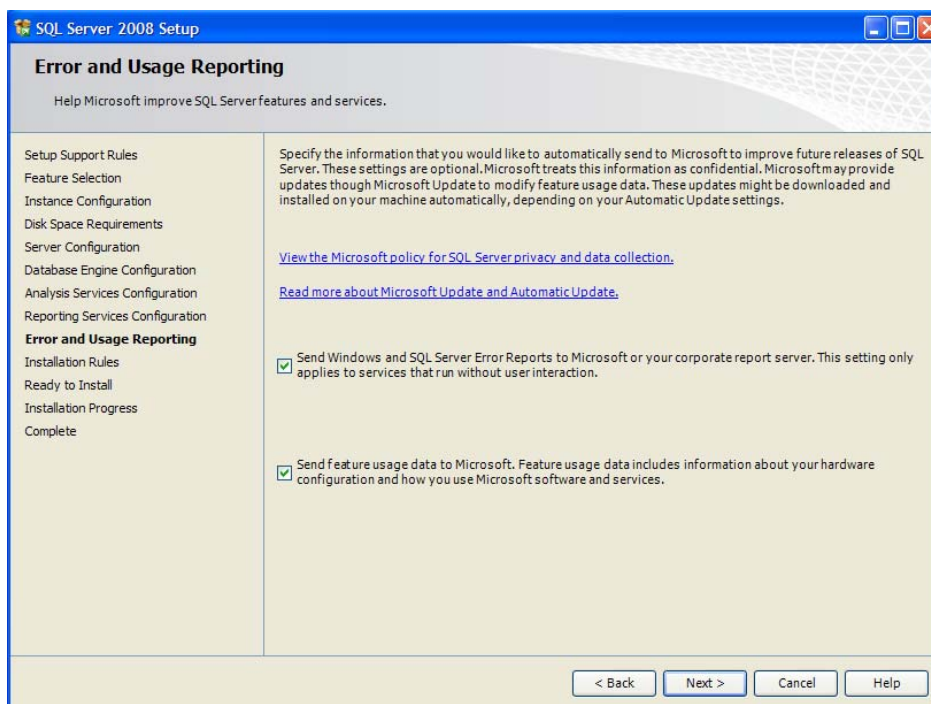


Рис.1.18

Если в окне настройки отчёта об ошибках включить все опции, то в случае ошибки вся информация об ошибке будет отправлена на сервер разработчика (Microsoft) для анализа. Если вам это необходимо то можете включить все опции, если нет, то не включайте. Нажмите кнопку «Next». Появится окно, отображающее правила установки (Рис.1.19).

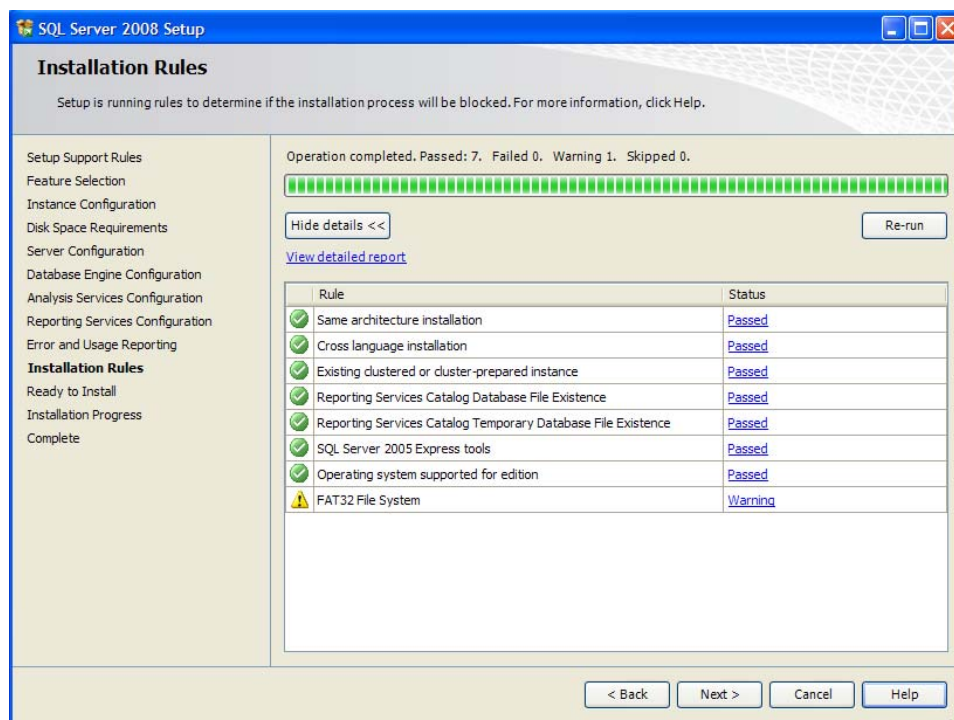


Рис.1.19

В выше представленном окне нажмите кнопку «Next»/ Появится оно с полным отчётом по установке сервера (Рис.1.20).

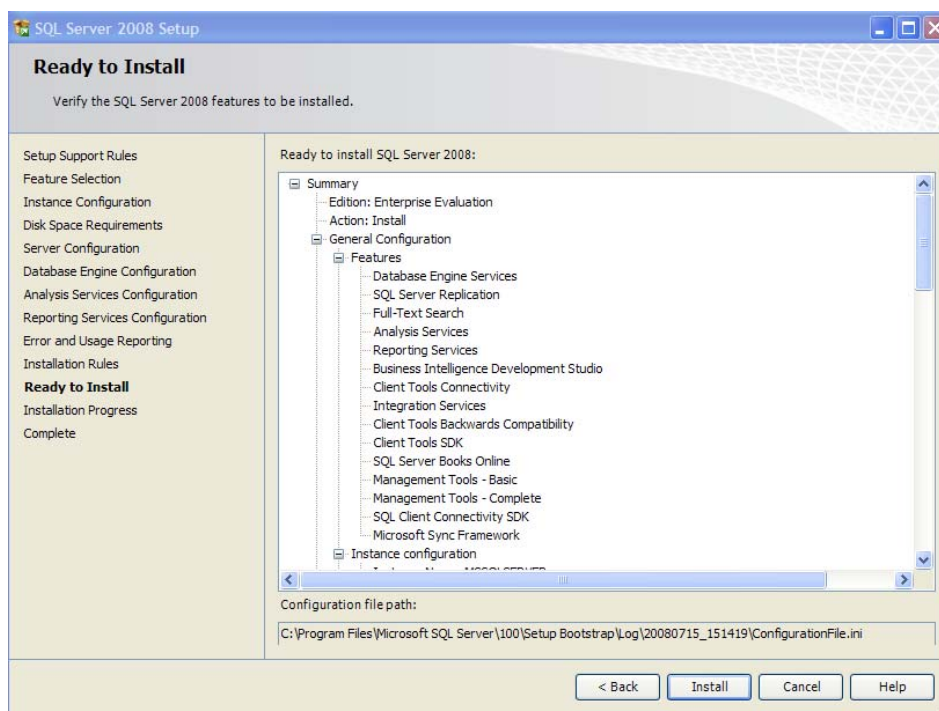


Рис.1.20

Для начала установки нажмите кнопку «Install» (Установить). Начнётся процесс установки сервера. Появится окно, отображающее ход установки сервера (Рис.1.21).

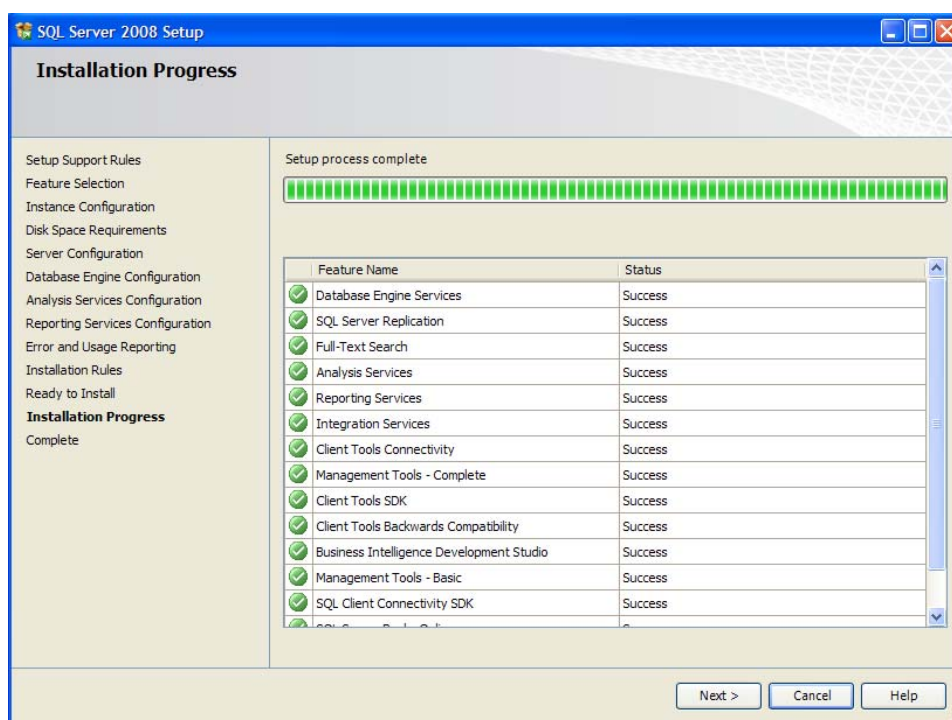


Рис.1.21

После окончания процесса установки нажмите кнопку «Next». Появится окно отчёта по результатам установки (Рис.1.22).

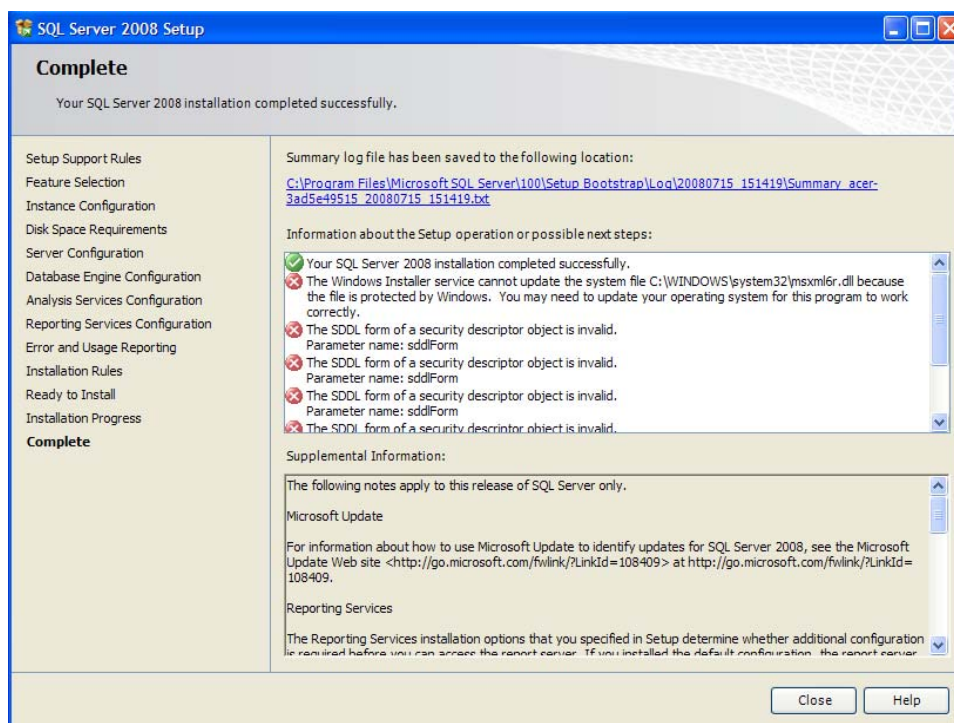


Рис.1.22

Для завершения установки сервера в данном окне нажмите кнопку «Close» (Закреть).

На этом мы заканчиваем описание установки «Microsoft SQL Server 2008» и переходим к созданию баз данных.

Лабораторная работа 2. Создание файла данных и журнала транзакций

Цель: научиться создавать файла данных и журнал транзакций

Создание любой БД начинается с создания файла данных. Рассмотрим этот процесс в «Microsoft SQL Server 2008» на примере создания простой БД по учёту успеваемости студентов.

Для начала необходимо запустить среду разработки «SQL Server Management Studio». Для этого в меню «Пуск» выбираем пункт «Программы\Microsoft SQL Server 2008\SQL Server Management Studio» (Рис.2.1).

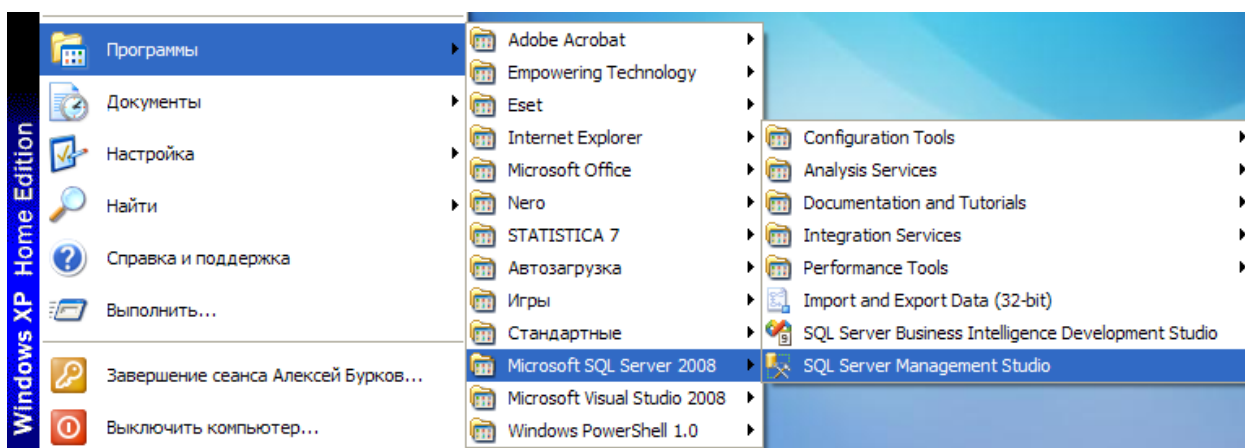


Рис.2.1

После запуска среды разработки появится окно подключения к серверу «Connect to Server» (Рис.2.2).

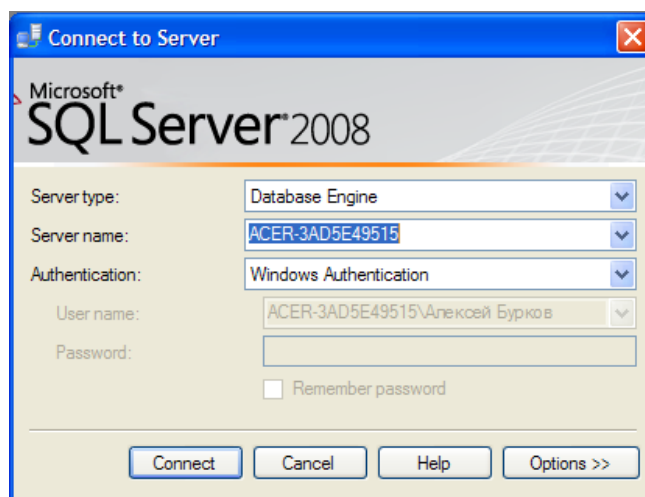


Рис.2.2

В этом окне необходимо нажать кнопку «Connect»

Замечание: Если при установке «Microsoft SQL Server 2008» был задан логин и пароль подключения к серверу, то перед нажатием кнопки «Connect», в выпадающем списке «Authentication» нужно выбрать «SQL Server Authentication», а затем необходимо ввести заданные при установке логин и пароль.

После нажатия кнопки «Connect» появится окно среды разработки «SQL Server Management Studio» (Рис.2.3).

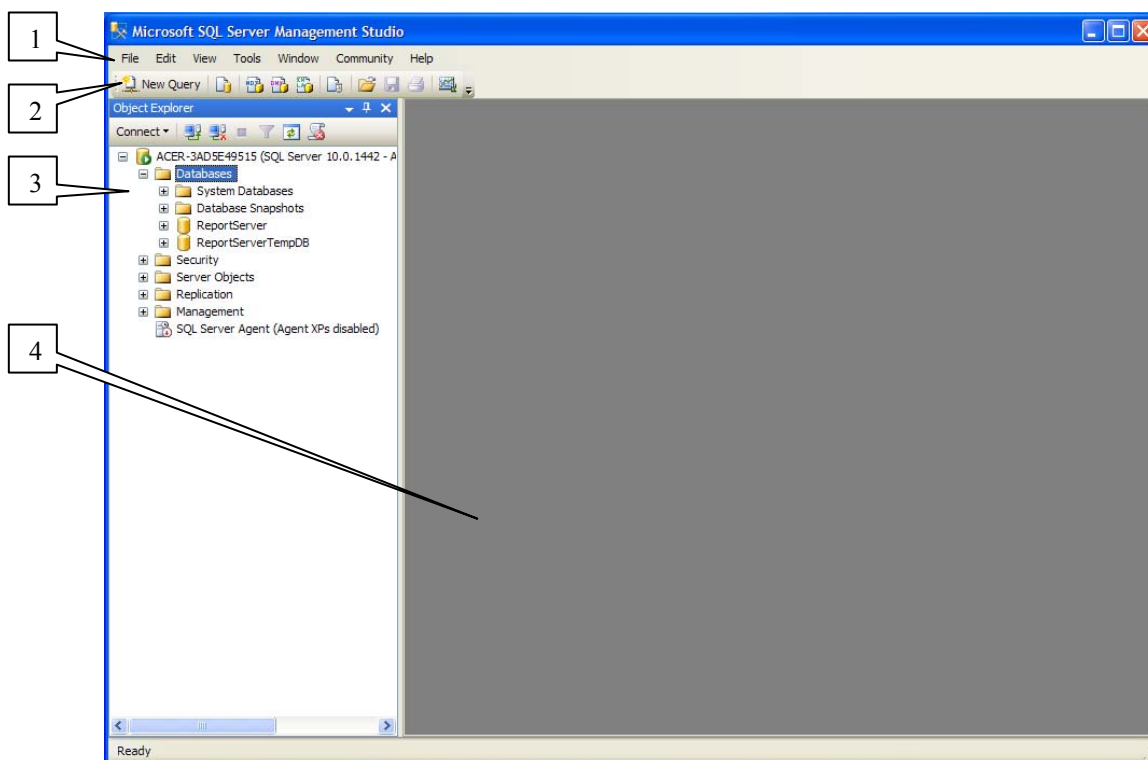


Рис.2.3

Данное окно имеет следующую структуру (Рис.2.3):

1. Оконное меню – содержит полный набор команд для управления сервером и выполнения различных операций.
2. Панель инструментов – содержит кнопки для выполнения наиболее часто производимых операций. Внешний вид данной панели зависит от выполняемой операции.
3. Панель «Object Explorer» - обозреватель объектов. Обозреватель объектов – это панель с древовидной структурой, отображающая все объекты сервера, а также позволяющая производить различные операции, как с самим сервером, так и с БД. Обозреватель объектов является основным инструментом для разработки БД.
4. Рабочая область. В рабочей области производятся все действия с БД, а также отображается её содержимое.

Замечание: В обозревателе объектов сами объекты находятся в папках. Чтобы открыть папку необходимо щёлкнуть по знаку «+» слева от изображения папки.

Теперь перейдём непосредственно к созданию файла данных. Для этого в обозревателе объектов щёлкните ПКМ на папке «Databases» (Базы данных) (Рис.2.3) и в появившемся меню выберите пункт «New Database» (Новая БД). Появится окно настроек параметров файла данных новой БД «New Database» (Рис.2.4). В левой части окна настроек имеется список «Select a page». Этот список позволяет переключаться между группами настроек.

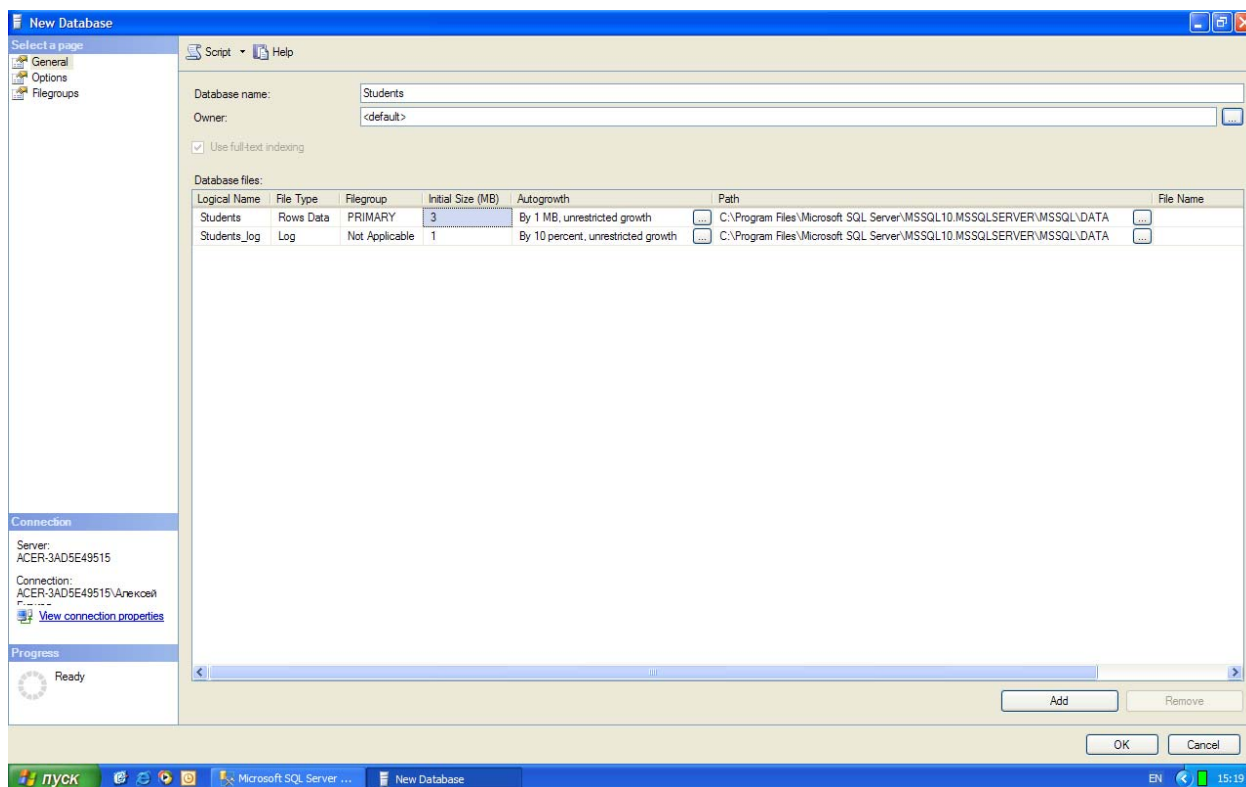


Рис.2.4

Для начала настроим основные настройки «General». Для выбора основных настроек нужно просто щёлкнуть мышью по пункту «General» в списке «Select a page». В правой части окна «New Database» появятся основные настройки (Рис.2.4)

Рассмотрим их более подробно. Верхней части окна расположено два параметра: «Database name» (Имя БД) и «Owner» (Владелец). Задайте параметр «Database name» равным «Students». Параметр «Owner» оставьте без изменений.

Под вышеприведёнными параметрами в виде таблицы располагаются настройки файла данных и журнала транзакций. Таблица имеет следующие столбцы:

- Logical Name – логическое имя файла данных и журнала транзакций. По этим именам будет происходить обращение к вышеприведённым файлам в БД. Можно заметить, что файл данных имеет то же имя что и БД, а имя файла журнала транзакций составлено из имени БД и суффикса «_log».
- File Type – тип файла. Этот параметр показывает, является ли файл файлом данных или журналом транзакций.
- Filegroup – группа файлов, показывает к какой группе файлов относится файл. Группы файлов настраиваются в группе настроек «Filegroups».
- Initial Size (MB) – начальный размер файла данных и журнала транзакций в мегабайтах.
- Autogrowth – автоувеличение размера файла. Как только файл заполняется информацией его размер автоматически увеличивается на величину, указанную в параметре «Autogrowth». Увеличение можно задавать как в мегабайтах так и в процентах. Здесь же можно задать максимальный размер файлов. Для изменения этого параметра надо нажать кнопку «...». В нашем случае (Рис.2.4) размер файлов не ограничен. Файл данных увеличивается на 1 мегабайт, а файл журнала транзакций на 10%.
- Path- путь к папке, где хранятся файлы. Для изменения этого параметра также надо нажать кнопку «...».

- File Name – имена файлов. По умолчанию имена файлов аналогичны логическим именам. Однако файл данных имеет расширение «mdf», а файл журнала транзакций – расширение «ldf».

Замечание: Для добавления новых файлов данных или журналов транзакций используется кнопка «Add», а для удаления кнопка «Remove».

В нашем случае мы оставим все основные настройки без изменений.

Теперь перейдём к другим второстепенным настройкам файла данных. Для доступа к этим настройкам необходимо щёлкнуть мышью по пункту «Options» в списке «Select a page». Появится следующее окно (Рис.2.5).

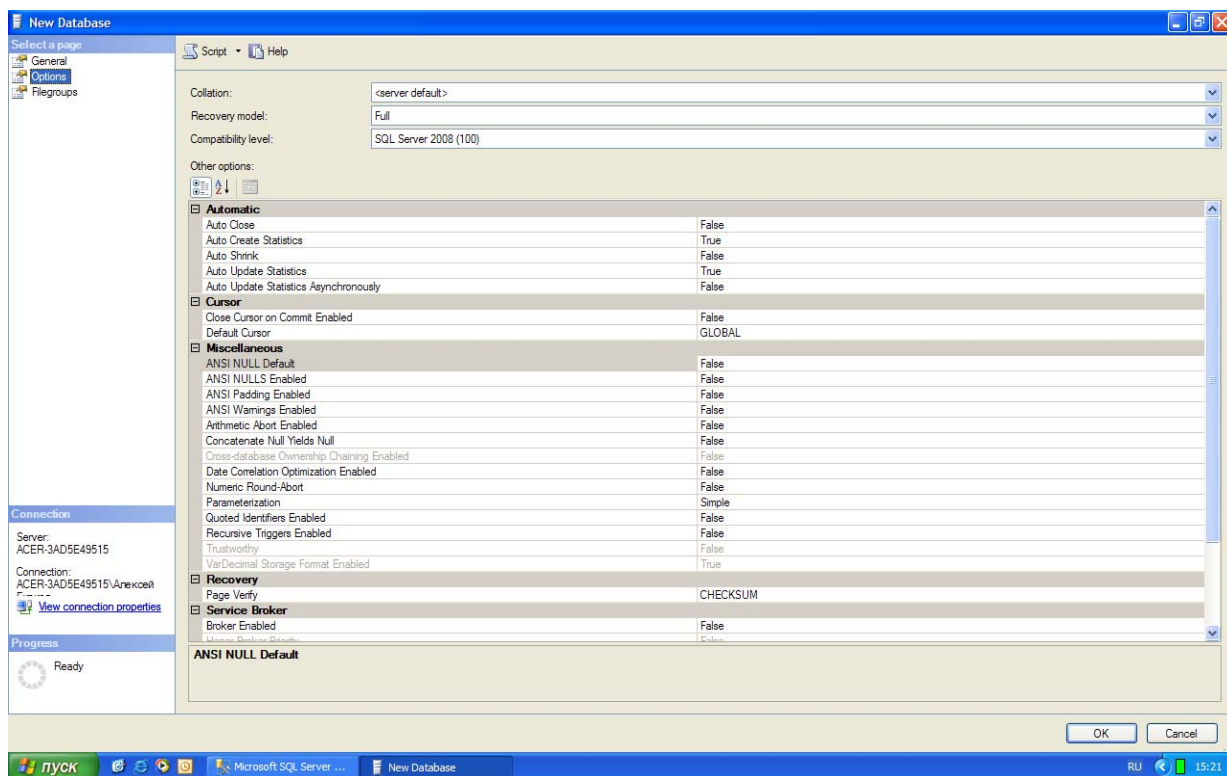


Рис.2.5

В правой части окна мы видим следующие настройки:

- Collation – этот параметр отвечает за обработку текстовых строк, их сравнение, текстовый поиск и т.д. Рекомендуется оставить его как «<server default>». При этом данный параметр будет равен значению, заданному на вкладке «Collation», при установке сервера.
- Recovery Model – модель восстановления. Данный параметр отвечает за информацию, предназначенную для восстановления БД, хранящуюся в файле транзакций. Чем полнее модель восстановления, тем больше вероятность восстановления данных при сбое системы или ошибках пользователей, но и больше размер файла журнала транзакций. При наличии места на диске, рекомендуется оставить этот параметр в значении «Full».
- Compatibility level – уровень совместимости, определяет совместимость файла данных с более ранними версиями сервера. Если планируется перенос данных на другую, более раннюю версию сервера, то её необходимо указать в этом параметре.
- Other options – второстепенные параметры. Данные параметры являются необязательными для изменения.

В нашем случае все параметры в разделе «Options», рекомендуется оставить как на рисунке 2.5.

Наконец рассмотрим последнюю группу настроек «Filegroups». Данная группа настроек отвечает за группы файлов. Для её отображения в списке «Select a page» необходимо щёлкнуть мышью по пункту «Options». Отобразятся настройки групп файлов (Рис.2.6).

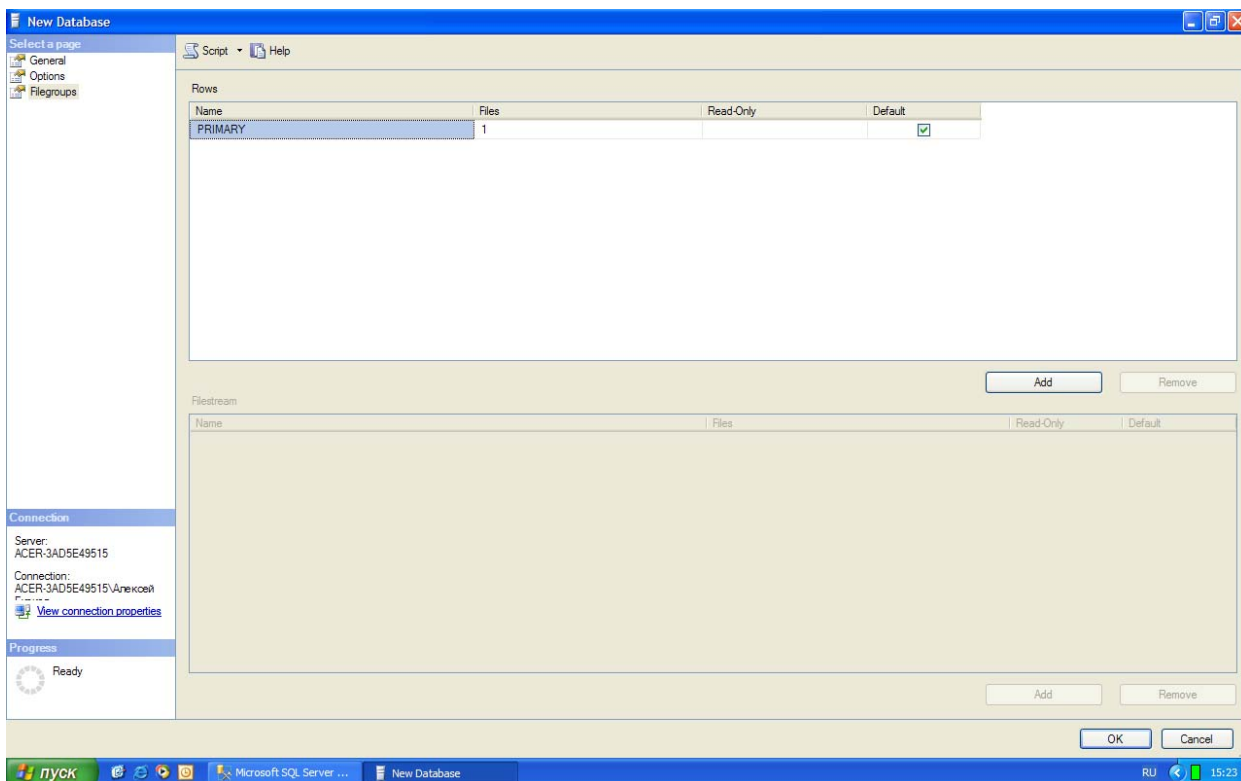


Рис.2.6

Группы файлов представлены в таблице «Rows» в правой части окна (Рис.2.6). Данная таблица имеет следующие столбцы:

- Name – имя группы файлов.
- Files – количество файлов входящих в группу.
- Read only – файлы в группе будут только для чтения. То есть, их можно только просматривать, но нельзя изменять.
- Default – группа по умолчанию. Все новые файлы данных будут входить в эту группу.

Замечание: Как и в случае с файлами данных, для добавления новых групп используется кнопка «Add», а для удаления кнопка «Remove».

В рассматриваемой БД нет необходимости добавлять новые группы файлов. Поэтому оставим группу настроек «Filegroups» без изменений.

На этом мы заканчиваем настройку свойств наших файлов. Для принятия всех настроек и создание фала данных и журнала транзакций нашей БД в окне «New Database» нажмём кнопку «Ok».

Произойдёт возврат в окно среду разработки «SQL Server Management Studio». На панели обозревателя объектов в папке «Databases» появиться новая БД «Students» (Рис.2.7).

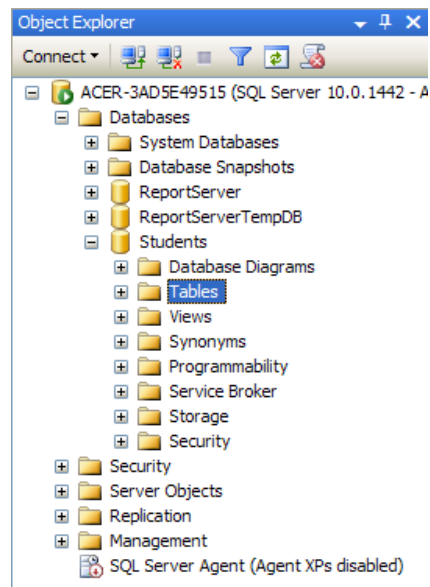


Рис.2.7

Замечание: Для переименования БД необходимо в обозревателе объектов щёлкнуть по ней ПКМ и в появившемся меню выбрать пункт «Rename». Для удаления в это же меню выбираем пункт «Delete», для обновления – пункт «Refresh», а для изменения свойств описанных выше – пункт «Properties».

Лабораторная работа 3. Создание и заполнение таблиц

Цель: научиться создавать и заполнять таблицы

Перейдём теперь к созданию таблиц. Все таблицы нашей БД находятся в подпапке «Tables» папки «Students» в окне обозревателя объектов (Рис.3.1).

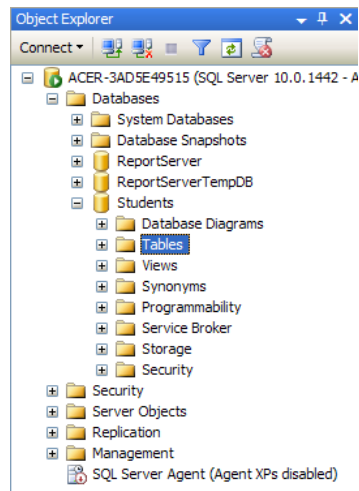


Рис.3.1

Создадим таблицу «Специальности». Для этого щёлкните ПКМ по папке «Tables» и в появившемся меню выберите пункт «New Table». Появится окно создания новой таблицы (Рис.3.2).

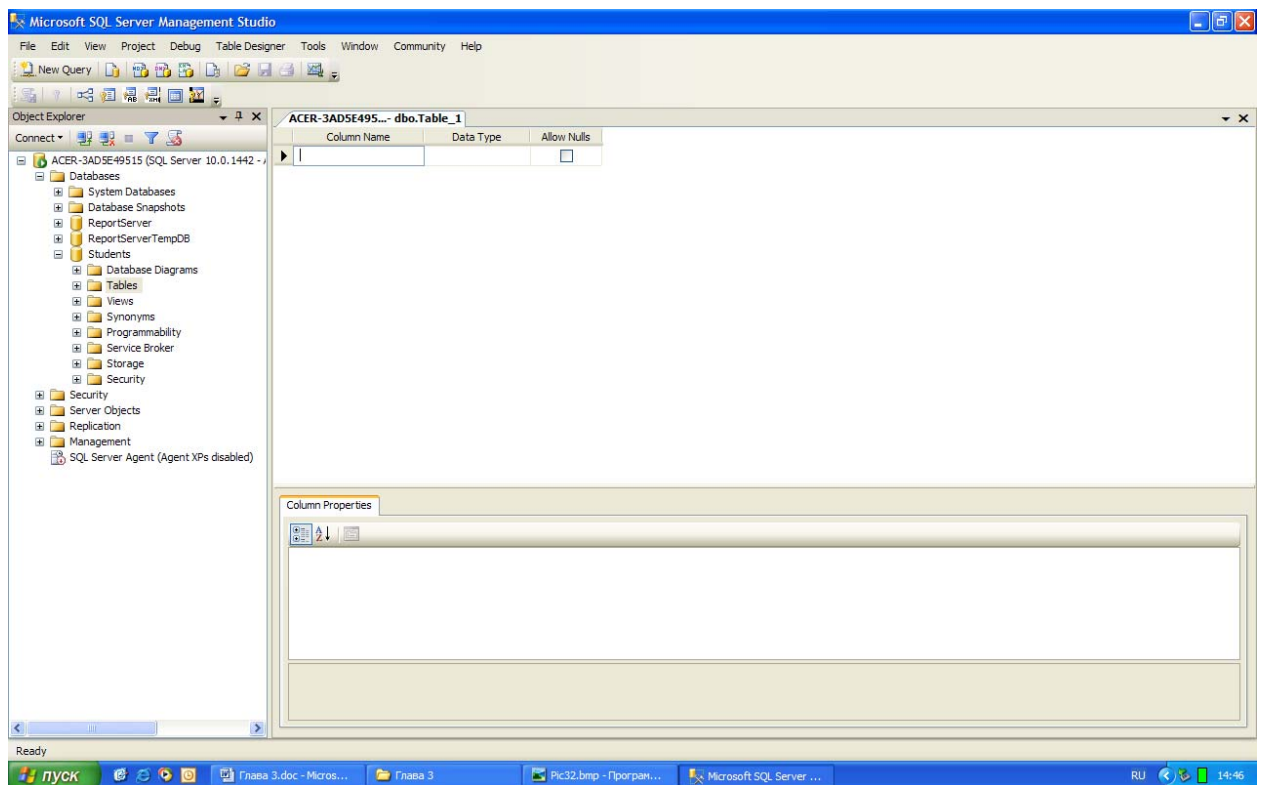


Рис.3.2

В правой части окна расположена таблица определения полей новой таблицы. Данная таблица имеет следующие столбцы:

- Column Name – имя поля. Имя поля должно всегда начинаться с буквы и не должно содержать различных специальных символов и знаков препинания. Если имя поля содержит пробелы, то оно автоматически заключается в квадратные скобки.
- Data Type – тип данных поля.
- Allow Nulls – допуск значения Null. Если эта опция поля включена, то в случае незаполнения поля в него будет автоматически подставлено значение Null. То есть, поле необязательно для заполнения.

Замечание: Под таблицей определения полей располагается таблица свойств выделенного поля «Column Properties». В данной таблице настраиваются свойства выделенного поля. Некоторые из них будут рассмотрены ниже.

Перейдём к созданию полей и настройке их свойств. В таблице определения полей задайте значения столбцов «Column Name», «Data Type» и «Allow Nulls», как показано на рисунке ниже (Рис.3.3).

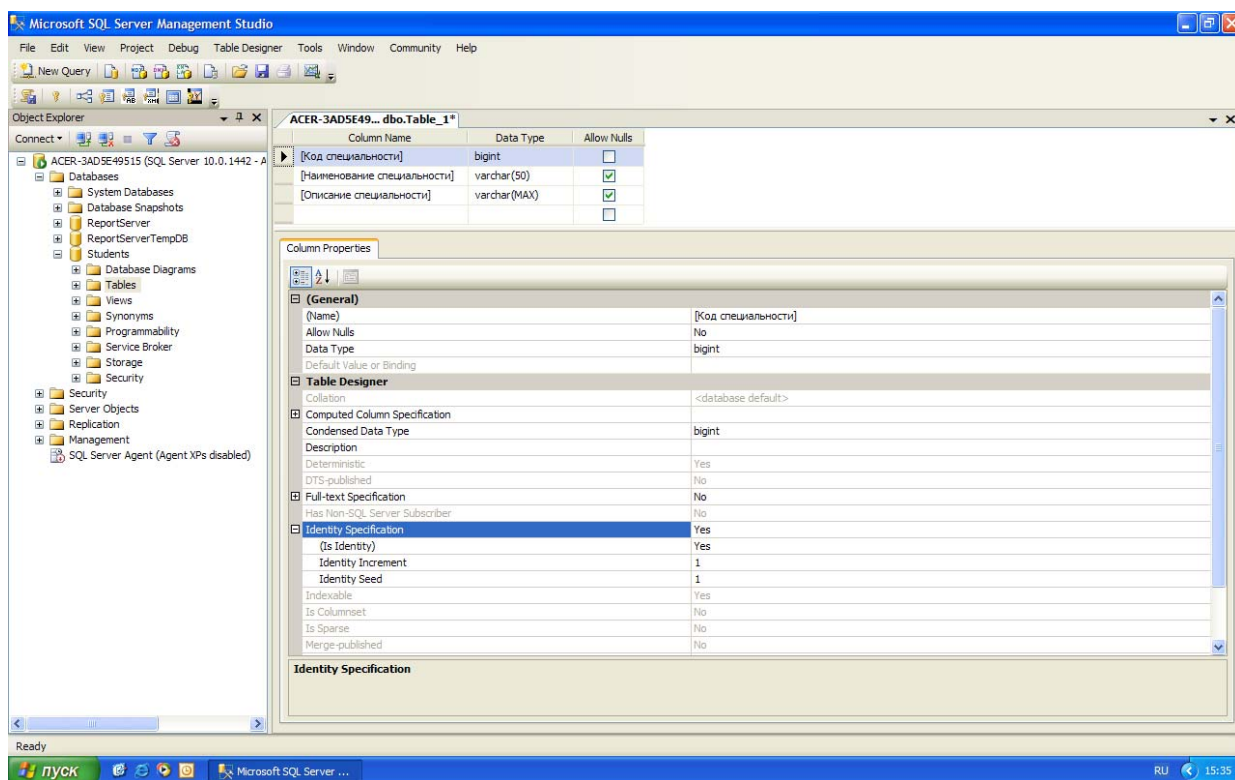


Рис.3.3


Из рисунка 3.3 следует, что наша таблица «Специальности» имеет три поля:


- Код специальности – числовое поле для связи с таблицей студенты,
- Наименование специальности – текстовое поле, предназначенное для хранения строк, имеющих длину не более 50 символов.
- Описание специальности - текстовое поле, предназначенное для хранения строк, имеющих неограниченную длину.

Замечание: Так как, поле «Код специальности» будет являться первичным полем связи в запросе, связывающем таблицы «Студенты» и «Специальности». То мы должны сделать

его числовым счётчиком. То есть данное поле должно автоматически заполняться числовыми значениями. Более того, оно должно быть ключевым.

Сделаем поле «Код специальности» счётчиком. Для этого выделите поле, просто щёлкнув по нему мышкой в таблице определения полей. В таблице свойств поля отобразятся свойства поля «Код специальности». Разверните группу свойств «Identity Specification» (Настройка особенности). Свойство «(Is Identity)» (Особенное) установите в значение «Yes» (Да). Задайте свойства «Identity Increment» (Увеличение особенности, шаг счётчика) и «Identity Seed» (Начало особенности, начальное значение счётчика) равными 1 (Рис.3.3). Эти настройки показывают, что значение поля «Код специальности» у первой записи в таблице будет равным 1, у второй – 2, у третьей 3 и т.д.

Теперь сделаем поле «Код специальности» ключевым полем. Выделите поле, а затем на панели инструментов нажмите кнопку с изображением ключа . В таблице определения полей, рядом с полем «Код специальности» появится изображение ключа, говорящее о том, что поле ключевое.

На этом настройку таблицы «Специальности» можно считать завершённой. Закройте окно создания новой таблицы, нажав кнопку закрытия  в верхнем правом углу окна, над таблицей определения полей. Появится окно с запросом о сохранении таблицы (Рис.3.4).

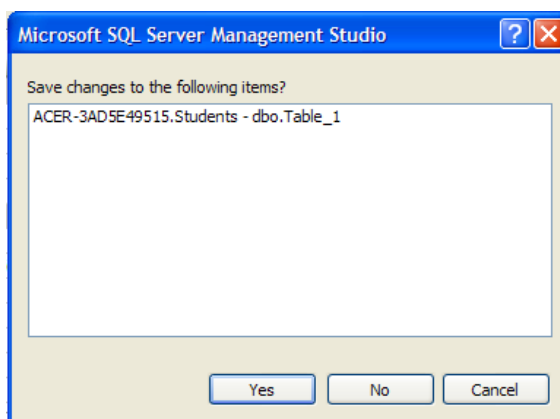


Рис.3.4

В этом окне необходимо нажать «Yes» (Да). Появится окно «Choose Name» (Задайте имя), предназначенное для определения имени новой таблицы (Рис.3.5).

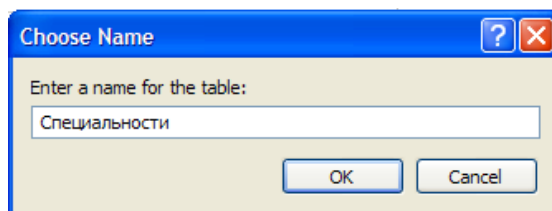


Рис.3.5

В этом окне задайте имя новой таблицы как «Специальности» и нажмите кнопку «Ok». Таблица «Специальности» отобразится в обозревателе объектов в папке «Tables» БД «Students» (Рис.3.6).

Замечание: В обозревателе объектов таблица «Специальности» отображается как «dbo.Специальности». Префикс «dbo» обозначает, что таблица является объектом БД (Data Base Object). В дальнейшем при работе с объектами БД префикс «dbo» можно опускать.

Теперь перейдём к созданию таблицы «Предметы». Как и в случае с таблицей «Специальности» щёлкните ПКМ по папке «Tables» и в появившемся меню выберите пункт «New Table». Создайте поля представленные на рисунке ниже (Рис.3.6).

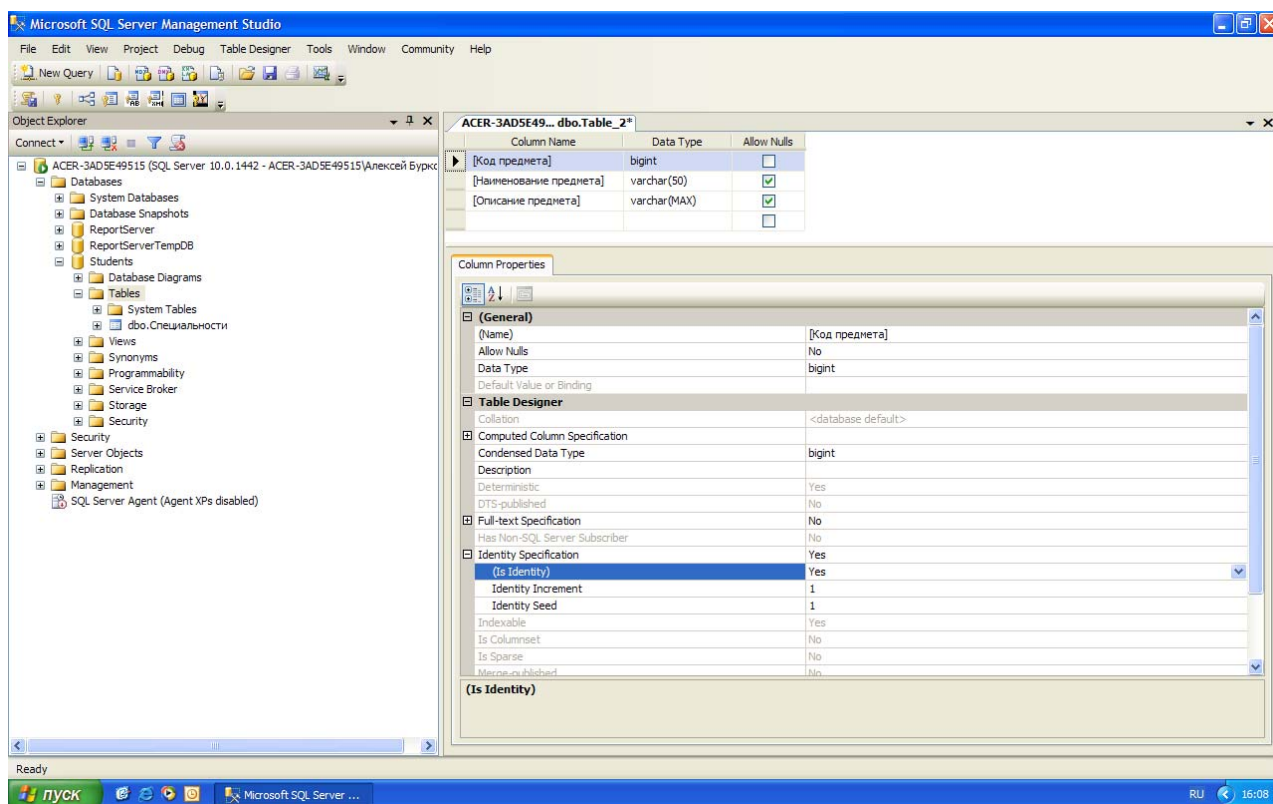


Рис.3.6

Сделайте поле «Код предмета» числовым счётчиком и ключевым полем, как это было сделано в таблице «Специальности». Закройте окно создания новой таблицы. В появившемся окне «Choose Name» задайте имя «Предметы» (Рис.3.7).

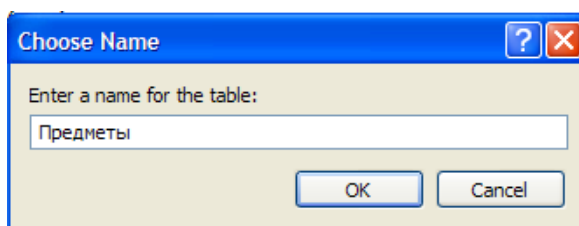


Рис.3.7

Таблица «Предметы» появится в папке «Tables» в обозревателе объектов (Рис.3.8).

После создания таблицы «Предметы» создайте таблицу «Студенты». Создайте новую таблицу аналогичную таблице представленной на рисунке 3.8.

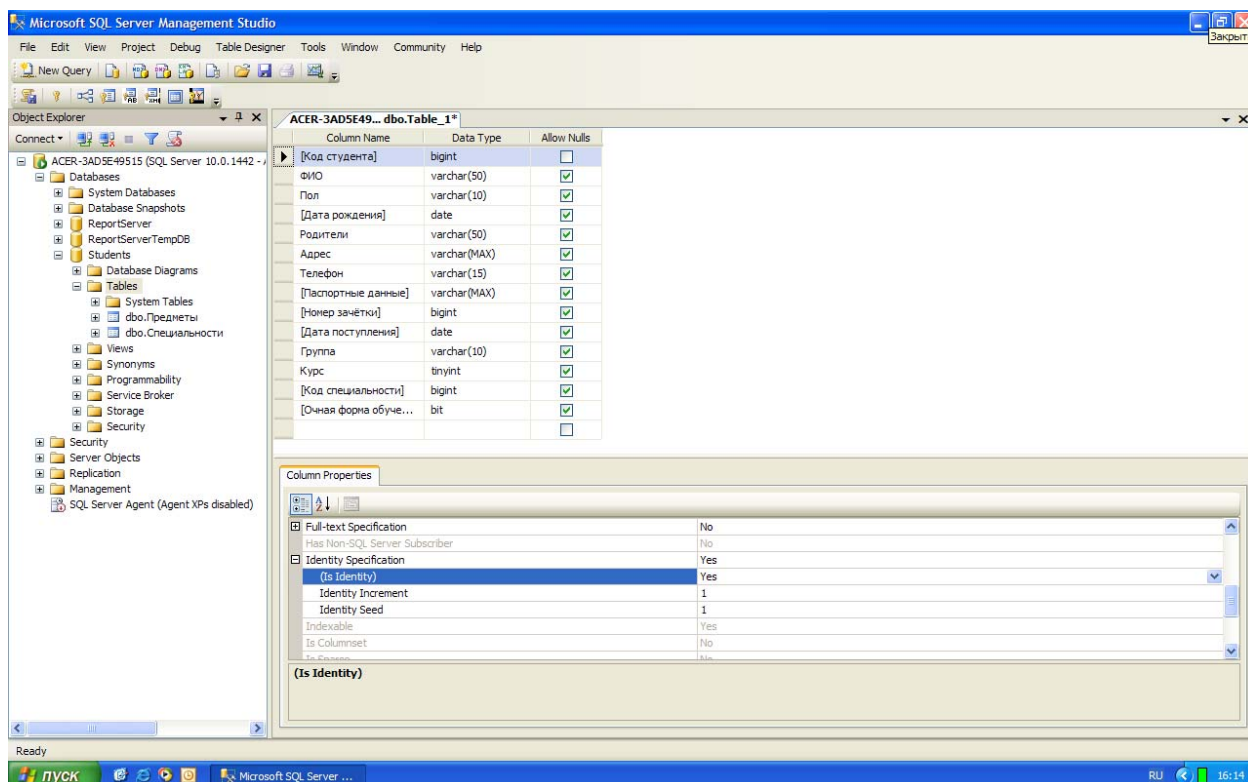


Рис.3.8

Рассматривая поля новой таблицы можно прийти к следующим выводам:

- Поле «Код студента» - это первичное поле для связи с таблицей оценки. Следовательно, данное поле необходимо сделать числовым счётчиком и ключевым (см. создание таблицы «Специальности» выше);
- Поля «ФИО», «Пол», «Родители», «Адрес», «Телефон», «Паспортные данные» и «Группа» являются текстовыми полями различной длины (для задания длины выделенного текстового поля необходимо в таблице свойств выделенного поля установить свойство Length равное максимальному количеству знаков текста вводимого в поле);
- Поля «Дата рождения» и «Дата поступления» предназначены для хранения дат. Поэтому они имеют тип данных «date»;
- Поле «Очная форма обучения» является логическим полем. В «Microsoft SQL Server 2008» такие поля должны иметь тип данных «bit»;
- Поля «Номер зачётки» и «Курс» являются целочисленными. Единственным отличием является размер полей. Поле «Номер зачётки» предназначено для хранения целых чисел в диапазоне $-2^{63} \dots +2^{63}$ (тип данных «bigint»). Поле «Курс » предназначено для хранения целых чисел в диапазоне $0 \dots 255$ (тип данных «tinyint»);
- Поле «Код специальности» - это поле связи с таблицей «Специальности». Однако, данное поле связи является вторичным, поэтому его можно сделать просто целочисленным, то есть, «bigint».

После определения полей таблицы «Студенты», закройте окно создания новой таблицы. В появившемся окне «Chose Name» задайте имя новой таблицы как «Студенты» (Рис.3.9).

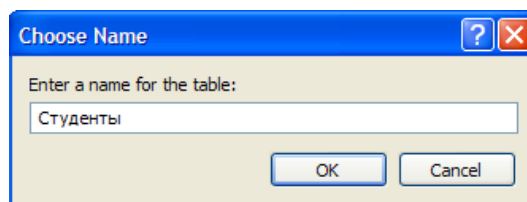


Рис.3.9

Таблица «Студенты» появится в папке «Tables» в обозревателе объектов (Рис.3.10).
Наконец, создадим таблицу «Оценки». Создайте поля, представленные на рисунке 3.10.

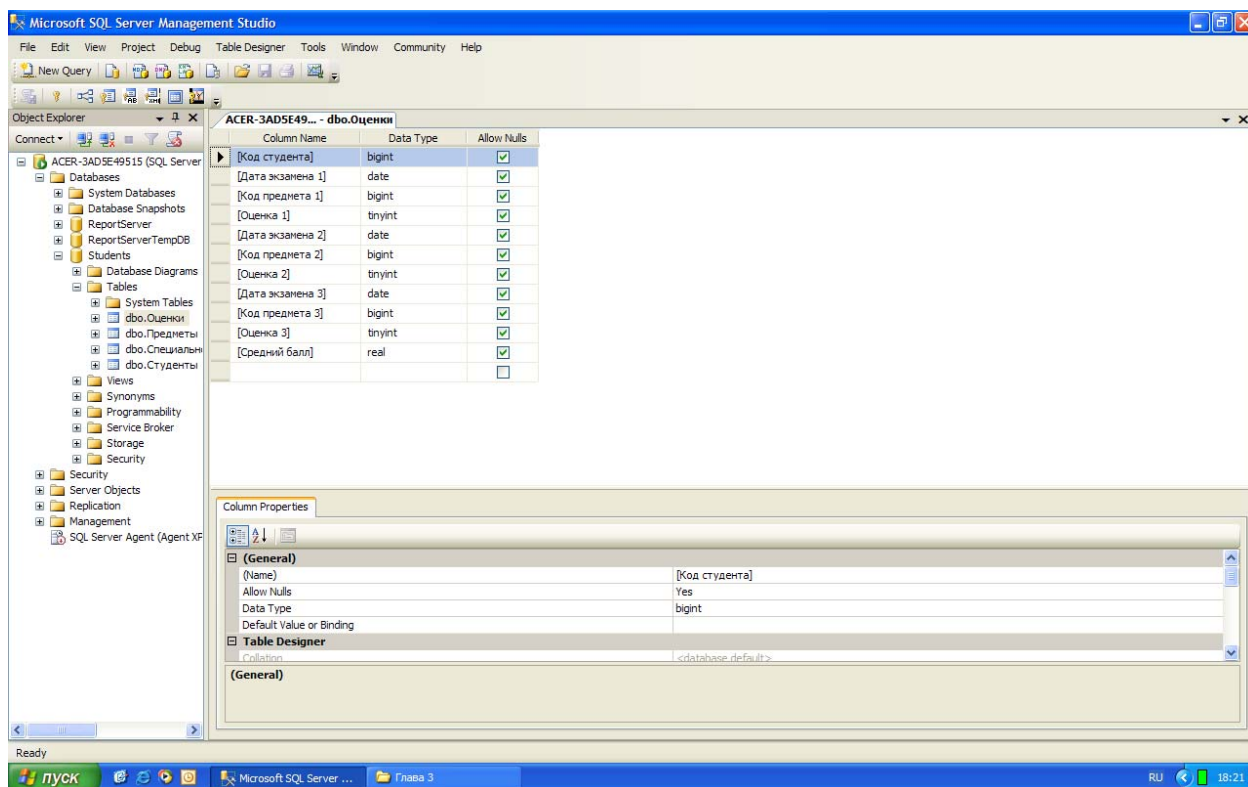


Рис.3.10

Таблица «Оценки» не имеет первичных полей связи. Следовательно, эта таблица не имеет ключевых полей. Поля «Код предмета 1», «Код предмета 2» и «Код предмета 3» являются вторичными полями связи, предназначенными для связи с таблицей «Предметы», поэтому они являются целочисленными (тип данных «bigint»). Поля «Дата экзамена 1», «Дата экзамена 2» и «Дата экзамена 3» предназначены для хранения дат (тип данных «date»). Поля «Оценка 1», «Оценка 2», и «Оценка 3» предназначены для хранения оценок. Задайте тип данных для этого поля «tinyint». Наконец, поле «Средний балл» хранит дробные числа и имеет тип «real».

Закройте окно создания новой таблицы, задав имя таблицы как «Оценки» (Рис.3.11).

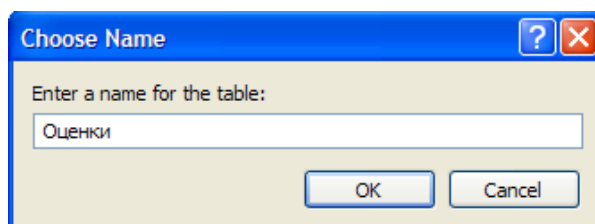


Рис.3.11

На этом мы заканчиваем создание таблиц БД «Students». После создания всех таблиц окно обозревателя объектов будет выглядеть так (Рис.3.12):

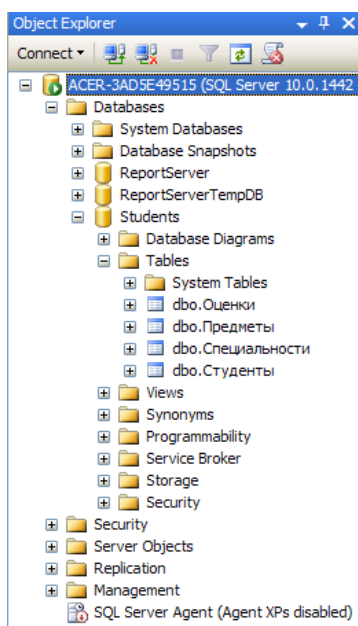


Рис.3.12

Теперь рассмотрим операцию заполнения таблиц начальными данными.


Для начала заполним таблицу «Специальности». Для заполнения этой таблицы в обозревателе объектов щёлкните правой кнопкой мыши по таблице «Специальности» (Рис.3.12) и в появившемся меню выберите пункт «Edit Top 200 Rows» (Редактировать первые 200 записей.). В рабочей области «Microsoft SQL Server Management Studio» проявится окно заполнения таблиц. Заполните таблицу «Специальности», как показано на рисунке 3.13.

	Код специальности	Наименование специальности	Описание специальности
	1	ММ	Математические методы
	2	ПИ	Прикладная информатика
	3	СТ	Статистика
	4	МО	Менеджмент организаций
►	5	БУ	Бухгалтерский учёт
*	NULL	NULL	NULL

Рис.3.13

Замечание: Заполнение таблиц происходит полностью аналогично табличному процессору «Microsoft Excel 2000».

Замечание: Так как поле «Код специальности» является первичным полем связи и ключевым числовым счётчиком, то оно заполняется автоматически (заполнять его не нужно).

Закройте окно заполнения таблицы «Специальность» щелкнув по кнопке закрытия окна  в верхнем правом углу, над таблицей.

После заполнения таблицы «Специальности» заполним таблицу «Предметы». Откройте её для заполнения как описано выше, и заполните, как показано на рисунке 3.14.

	Код предмета	Наименование предмета	Описание предмета
	1	Операционные системы	Microsoft Windows Vista
	2	Офисные пакеты	Microsoft Office 2007
	3	Базы данных	Microsoft Access 2007
	4	Языки программирования	Microsoft Visual Studio 2008
►	5	Проектирование информационных систем	Microsoft SQL server 2008
*	NULL	NULL	NULL

Рис.3.14

Закройте окно заполнения таблицы «Предметы» и перейдите к заполнению таблицы «Студенты». Откройте таблицу «Студенты» для заполнения и заполните её как показано ниже (Рис.3.15).

ACER-3AD5E49... - dbo.Студенты														
	Код сту...	ФИО	Пол	Дата рож...	Родители	Адрес	Телефон	Паспортные д...	Номер за...	Дата пост...	Группа	Курс	Код спец...	Очная форма...
	1	Иванов А.И.	Мужской	1983-12-12	Отец, Мать	Москва	+74957895674	8567-567543	13245	2007-09-01	ММ11	1	1	True
	2	Петрова И.И.	Женский	1982-11-01	Мать	Москва	+74957889876	4567-765432	34563	2006-08-01	ПИ21	2	2	False
	3	Мухин М.А.	Мужской	1982-05-14	Отец	Самара	+78462875690	5438-098787	56732	2006-07-05	СТ22	2	3	False
	4	Сидорова В.К.	Женский	1981-09-27	Нет	Саратов	+79027868909	1287-987509	27543	2005-06-23	МО31	3	4	True
	5	Кожевников А.А.	Мужской	1981-04-12	Мать	Казань	+79168563476	2312-675468	34217	2005-07-21	БУ33	3	5	True
	6	Пальчикова Н.Е.	Женский	1983-09-02	Отец, Мать	Челябинск	+74569098723	8743-856780	43278	2007-08-01	ММ12	1	1	False
	7	Царегородцев Е.В.	Мужской	1980-02-17	Отец	Самара	+78462234769	6543-834521	43765	2004-07-04	ПИ41	4	2	True
	8	Баранова Г.В.	Женский	1980-07-09	Отец, Мать	Чебоксары	+79027874638	2133-896567	10387	2004-08-09	СТ42	4	3	False
	9	Леухин П.Г.	Мужской	1979-02-26	Нет	Казань	+79067453678	2769-634904	67348	2003-07-23	МО51	5	4	True
▶	10	Николаева А.П.	Женский	1979-03-17	Мать	Саратов	+78546456432	3256-490932	45287	2003-06-21	БУ53	5	5	False
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рис.3.15

Замечание: Для заполнения дат в качестве разделителя можно использовать знак «.». Даты можно заполнять в формате «день.месяц.год».

Замечание: Поле «Код специальности» является вторичным полем связи (для связи с таблицей «Специальности»). Следовательно, значения этого поля необходимо заполнять значениями поля «Код специальности» таблицы «Специальности». В нашем случае это значения от 1 до 5 (Рис.3.13). Если у Вас коды специальностей в таблице «Специальности» имеют другие значения, то внесите их в таблицу «Студенты».

По окончании заполнения, закройте окно заполнения таблицы «Студенты». Наконец заполним таблицу «Оценки», как это показано на рисунке 3.16.

ACER-3AD5E49... - dbo.Оценки											
	Код студента	Дата экзамена 1	Код предмета 1	Оценка 1	Дата экзамена 2	Код предмета 2	Оценка 2	Дата экзамена 3	Код предмета 3	Оценка 3	Средний балл
	1	2008-02-01	1	5	2008-02-09	4	3	2008-02-14	2	4	0
	2	2008-01-30	5	4	2008-02-23	3	5	2008-02-27	1	5	0
	3	2008-01-26	3	5	2008-02-05	1	3	2008-02-15	5	3	0
	4	2007-12-26	2	3	2008-01-05	4	4	2008-01-21	3	4	0
	5	2008-01-12	4	4	2008-01-18	5	4	2008-01-25	1	4	0
	6	2007-12-17	2	4	2007-12-26	4	5	2008-01-05	1	3	0
	7	2008-02-21	5	2	2008-02-25	1	2	2008-02-27	2	4	0
	8	2008-02-03	3	3	2008-02-12	5	3	2008-02-20	4	5	0
	9	2008-01-25	1	5	2008-02-02	3	5	2008-02-14	5	5	0
▶	10	2007-12-28	4	4	2008-01-11	1	4	2008-01-23	2	3	0
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рис.3.16

Замечание: Поля с датами заполняются, как и в таблице «Студенты» (см. выше).

Замечание: Поля «Код предмета 1», «Код предмета 2» и «Код предмета 3» являются вторичными полями связи с таблицей «Предметы». По этому они должны быть заполнены значениями поля «Код предмета из этой таблицы», то есть значениями от 1 до 5 (см. рис.3.14).

Закройте окно заполнения таблицы «Оценки». На этом мы заканчиваем создание и заполнение таблиц нашей БД «Students».

Лабораторная работа 4. Создание запросов и фильтров

Цель: научиться создавать запросы и фильтры

Перейдём к созданию статических запросов. В обозревателе объектов «Microsoft SQL Server 2008» все запросы БД находятся в папке «Views» (Рис 4.1).

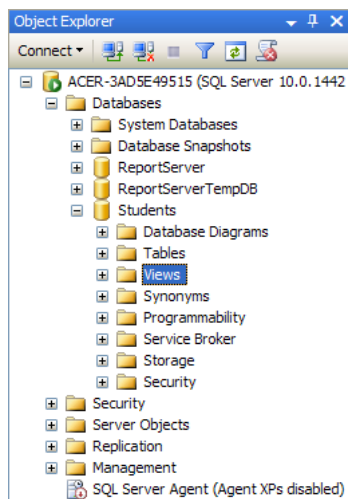


Рис.4.1

Создадим запрос «Запрос Студенты+Специальности», связывающий таблицы «Студенты» и «Специальности» по полю связи «Код специальности». Для создания нового запроса необходимо в обозревателе объектов в БД «Students» щёлкнуть ПКМ по папке «Views», затем в появившемся меню выбрать пункт «New View». Появится окно «Add Table» (Добавить таблицу), предназначенное для выбора таблиц и запросов, участвующих в новом запросе (Рис.4.2).

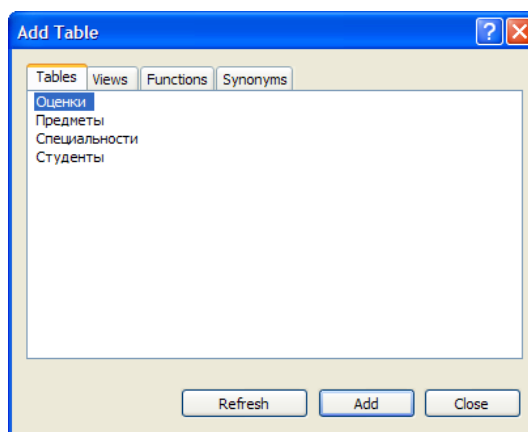


Рис.4.2

Добавим в новый запрос таблицы «Студенты» и «Специальности». Для этого в окне «Add Table» выделите таблицу «Студенты» и нажмите кнопку «Add» (Добавить). Аналогично добавьте таблицу «Специальности». После добавления таблиц участвующих в запросе закройте окно «Add Table» нажав кнопку «Close» (Закрыть). Появится окно конструктора запросов (Рис.4.3).

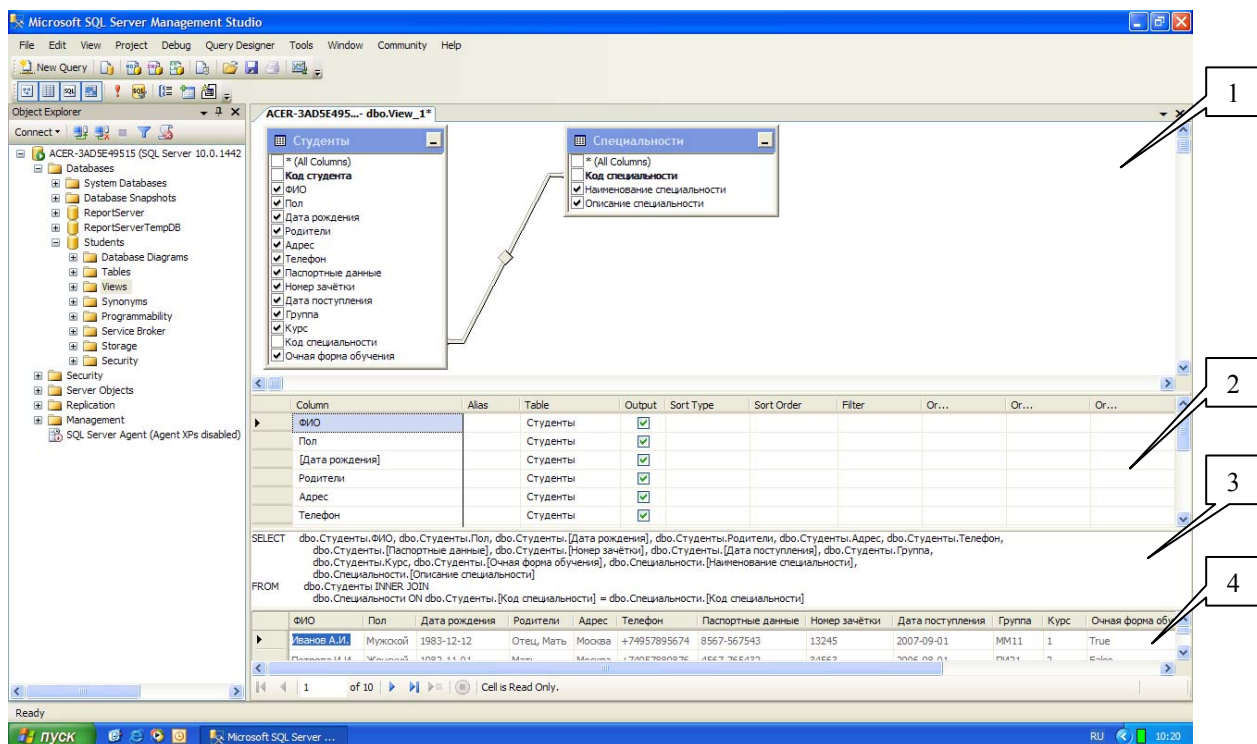







Рис.4.3

Замечание: Окно конструктора запросов состоит из следующих панелей:

1. Схема данных – отображает поля таблиц и запросов, участвующих в запросе, позволяет выбирать отображаемые поля, позволяет устанавливать связи между участниками запроса по специальным полям связи. Эта панель включается и выключается следующей кнопкой на панели инструментов ;
2. Таблица отображаемых полей – показывает отображаемые поля (столбец «Column»), позволяет задавать им псевдонимы (столбец «Alias»), позволяет устанавливать тип сортировки записей по одному или нескольким полям (столбец «Sort Type»), позволяет задавать порядок сортировки (столбец «Sort Order»), позволяет задавать условия отбора записей в фильтрах (столбцы «Filter» и «Or...»). Также эта таблица позволяет менять порядок отображения полей в запросе. Эта панель включается и выключается следующей кнопкой на панели инструментов ;
3. Код SQL – код создаваемого запроса на языке T-SQL. Эта панель включается и выключается следующей кнопкой на панели инструментов ;
4. Результат – показывает результат запроса после его выполнения. Эта панель включается и выключается следующей кнопкой на панели инструментов .

Замечание: Если необходимо снова отобразить окно «Add Table» для добавления новых таблиц или запросов, то для этого на панели инструментов «Microsoft SQL Server 2008» нужно нажать кнопку .

Замечание: Если необходимо удалить таблицу или запрос из схемы данных, то для этого нужно щёлкнуть ПКМ и в появившемся меню выбрать пункт «Remove» (Удалить).

Теперь перейдём к связыванию таблиц «Студенты» и «Специальности» по полям связи «Код специальности». Чтобы создать связь необходимо в схеме данных перетащить мышью поле «Код специальности» таблицы «Специальности» на такое же поле таблицы

«Студенты». Связь отобразится в виде ломаной линии соединяющей эти два поля связи (Рис.4.3).

Замечание: Если необходимо удалить связь, то для этого необходимо щёлкнуть по ней ПКМ и в появившемся меню выбрать пункт «Remove».


Замечание: После связывания таблиц (а также при любых изменениях в запросе) в области кода T-SQL будет отображаться T-SQL код редактируемого запроса.

Теперь определим поля, отображаемые при выполнении запроса. Отображаемые поля обозначаются галочкой (слева от имени поля) на схеме данных, а также отображаются в таблице отображаемых полей. Чтобы сделать поле отображаемым при выполнении запроса необходимо щёлкнуть мышью по пустому квадрату (слева от имени поля) на схеме данных, в квадрате появится галочка.


Замечание: Если необходимо сделать поле невидимым при выполнении запроса, то нужно убрать галочку, расположенную слева от имени поля на схеме данных. Для этого просто щёлкните мышью по галочке.

Замечание: Если необходимо отобразить все поля таблицы, то необходимо установить галочку слева от пункта «* (All Columns)» (Все поля), принадлежащего соответствующей таблице на схеме данных.

Определите отображаемые поля нашего запроса, как это показано на рисунке 4.3 (Отображаются все поля кроме полей с кодами, то есть полей связи).

На этом настройку нового запроса можно считать законченной. Перед сохранением запроса проверим его работоспособность, выполнив его. Для запуска запроса на панели инструментов нажмите кнопку . Либо щёлкните ПКМ в любом месте окна конструктора запросов и в появившемся меню выберите пункт «Execute SQL» (Выполнить SQL). Результат выполнения запроса появится в виде таблицы в области результата (Рис.4.3).

Замечание: Если после выполнения запроса результат не появился, а появилось сообщение об ошибке, то в этом случае проверьте, правильно ли создана связь. Ломаная линия связи должна соединять поля «Код специальности» в обеих таблицах. Если линия связи соединяет другие поля, то её необходимо удалить и создать заново, как это описано выше.

Если запрос выполняется правильно, то необходимо сохранить. Для сохранения запроса закройте окно конструктора запросов, щёлкнув мышью по кнопке закрытия , расположенной в верхнем правом углу окна конструктора (над схемой данных). Появится окно с вопросом о сохранении запроса (Рис.4.4).

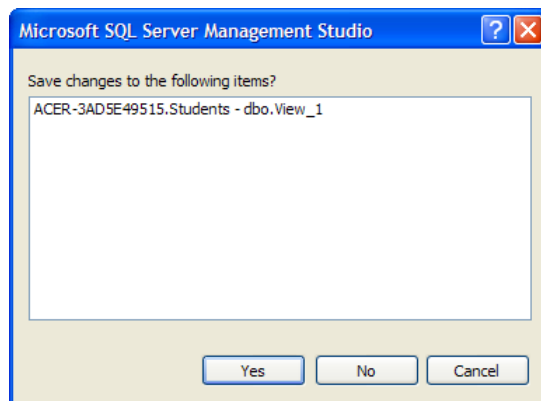


Рис.4.4

В данном окне необходимо нажать кнопку «Yes» (Да). Появится окно «Choose Name» (Выберите имя) (Рис.4.5).

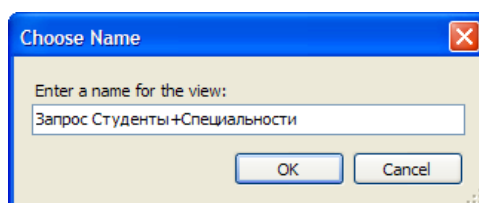


Рис.4.5

В данном окне зададим имя нового запроса «Запрос Студенты+Специальности» и нажмём кнопку «Ok». Запрос появится в папке «Views» БД «Students» в обозревателе объектов (Рис.4.6).

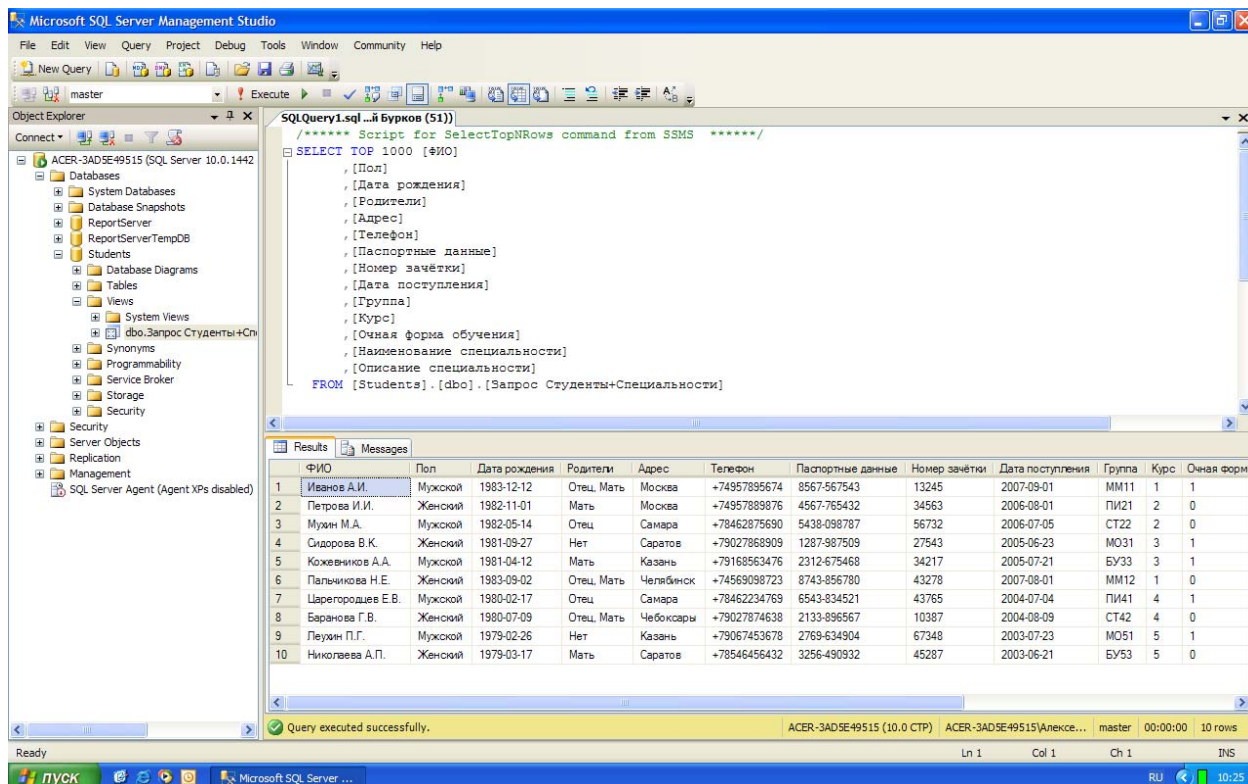
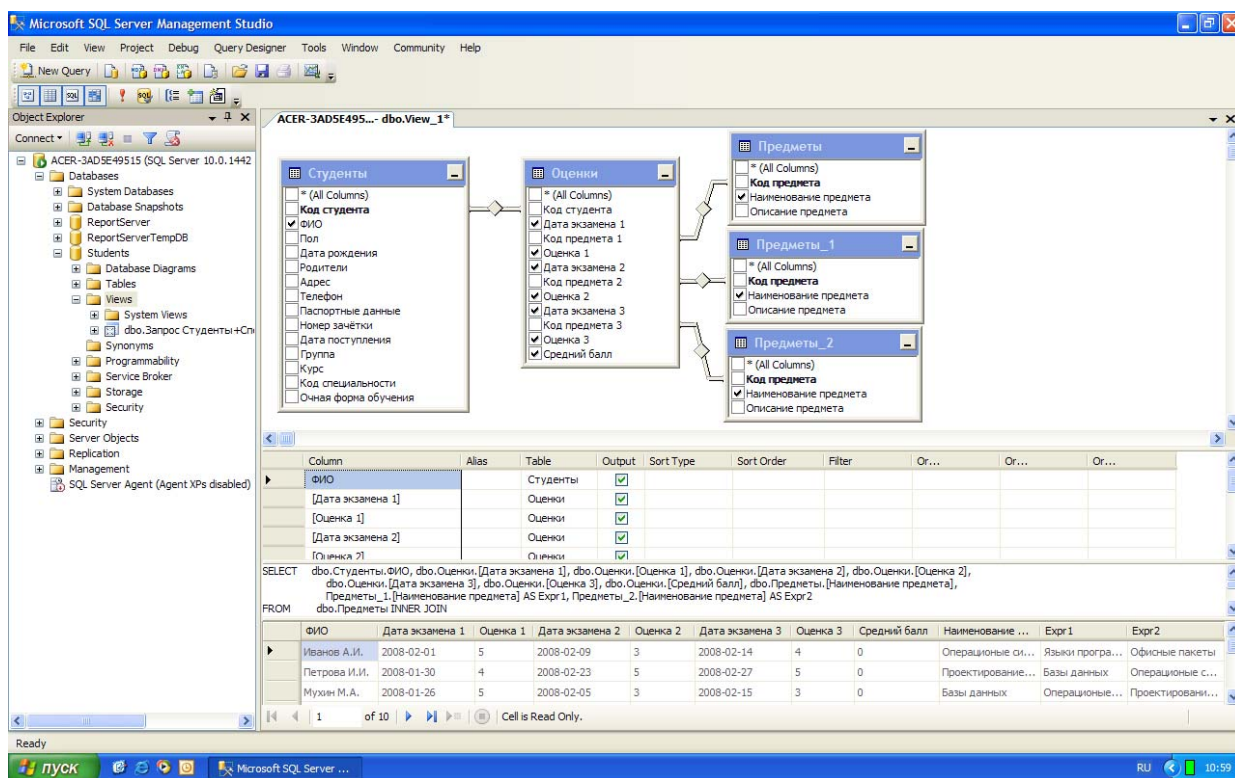


Рис.4.6

Проверим работоспособность созданного запроса вне конструктора запросов. Запустим вновь созданный запрос «Запрос Студенты+Специальности» без использования

В окне конструктора запросов установите связи между таблицами и определите отображаемые поля, как показано на рисунке 4.7.



Теперь поменяем порядок отображаемых полей в запросе, для этого в таблице отображаемых полей необходимо перетащить поля мышью вверх или вниз за заголовок строки таблицы (столбец перед столбцом «Column»). Расположите отображаемые поля в в таблице отображаемых полей как показано на рисунке 4.8.

Column	Alias	Table	Output	Sort Type	Sort Order	Filter	Or...	Or...	Or...
ФИО	[ФИО студента]	Студенты	<input checked="" type="checkbox"/>						
[Дата экзамена 1]	[Дата первого экзамена]	Оценки	<input checked="" type="checkbox"/>						
[Наименование предмета]	[Наименование предмета первого экзамена]	Предметы	<input checked="" type="checkbox"/>						
[Оценка 1]	[Оценка первого экзамена]	Оценки	<input checked="" type="checkbox"/>						
[Дата экзамена 2]	[Дата второго экзамена]	Оценки	<input checked="" type="checkbox"/>						
[Наименование предмета]	[Наименование предмета второго экзамена]	Предметы_1	<input checked="" type="checkbox"/>						
[Оценка 2]	[Оценка второго экзамена]	Оценки	<input checked="" type="checkbox"/>						
[Дата экзамена 3]	[Дата третьего экзамена]	Оценки	<input checked="" type="checkbox"/>						
[Наименование предмета]	[Наименование предмета третьего экзамена]	Предметы_2	<input checked="" type="checkbox"/>						
[Оценка 3]	[Оценка третьего экзамена]	Оценки	<input checked="" type="checkbox"/>						
[Средний балл]	[Средний балл студента за сессию]	Оценки	<input checked="" type="checkbox"/>						


```

SELECT dbo.Студенты.ФИО AS [ФИО студента], dbo.Оценки.[Дата экзамена 1] AS [Дата первого экзамена],
dbo.Предметы.[Наименование предмета] AS [Наименование предмета первого экзамена], dbo.Оценки.[Оценка 1] AS [Оценка первого экзамена],
dbo.Оценки.[Дата экзамена 2] AS [Дата второго экзамена],
dbo.Предметы_1.[Наименование предмета] AS [Наименование предмета второго экзамена], dbo.Оценки.[Оценка 2] AS [Оценка второго экзамена],
dbo.Оценки.[Дата экзамена 3] AS [Дата третьего экзамена],
dbo.Предметы_2.[Наименование предмета] AS [Наименование предмета третьего экзамена], dbo.Оценки.[Оценка 3] AS [Оценка третьего экзамена],
Предметы_2.[Наименование предмета] AS [Наименование предмета третьего экзамена],

```

ФИО студента	Дата первого экзамена	Наименование предмета пер...	Оценка первого экза...	Дата второго экзамена	Наименование предмета вто...	Оценка втор...	Дат
Иванов А.И.	2008-02-01	Операционные системы	5	2008-02-09	Языки программирования	3	2008
Петрова И.И.	2008-01-30	Проектирование информацио...	4	2008-02-23	Базы данных	5	2008
Мухин М.А.	2008-01-26	Базы данных	5	2008-02-05	Операционные системы	3	2008
Сидорова В.К.	2007-12-26	Офисные пакеты	3	2008-01-05	Языки программирования	4	2008

Рис.4.8

Задайте псевдонимы для каждого из полей, просто записав псевдонимы в столбце «Alias» таблицы отображаемых полей, как на рисунке 4.8.

Проверьте работоспособность нового запроса, выполнив его. Обратите внимание на то, что реальные названия полей были заменены их псевдонимами. Закройте окно конструктора запросов. В появившемся окне «Choose Name» задайте имя нового запроса «Запрос Студенты+Оценки» (Рис.4.9).

Choose Name

Enter a name for the view:

Запрос Студенты+Оценки

OK

Cancel

Рис.4.9

Проверьте работоспособность нового запроса вне конструктора. Для этого запустите запрос. Результат выполнения запроса «Запрос Студенты+Оценки» должен выглядеть как на рисунке 4.10.

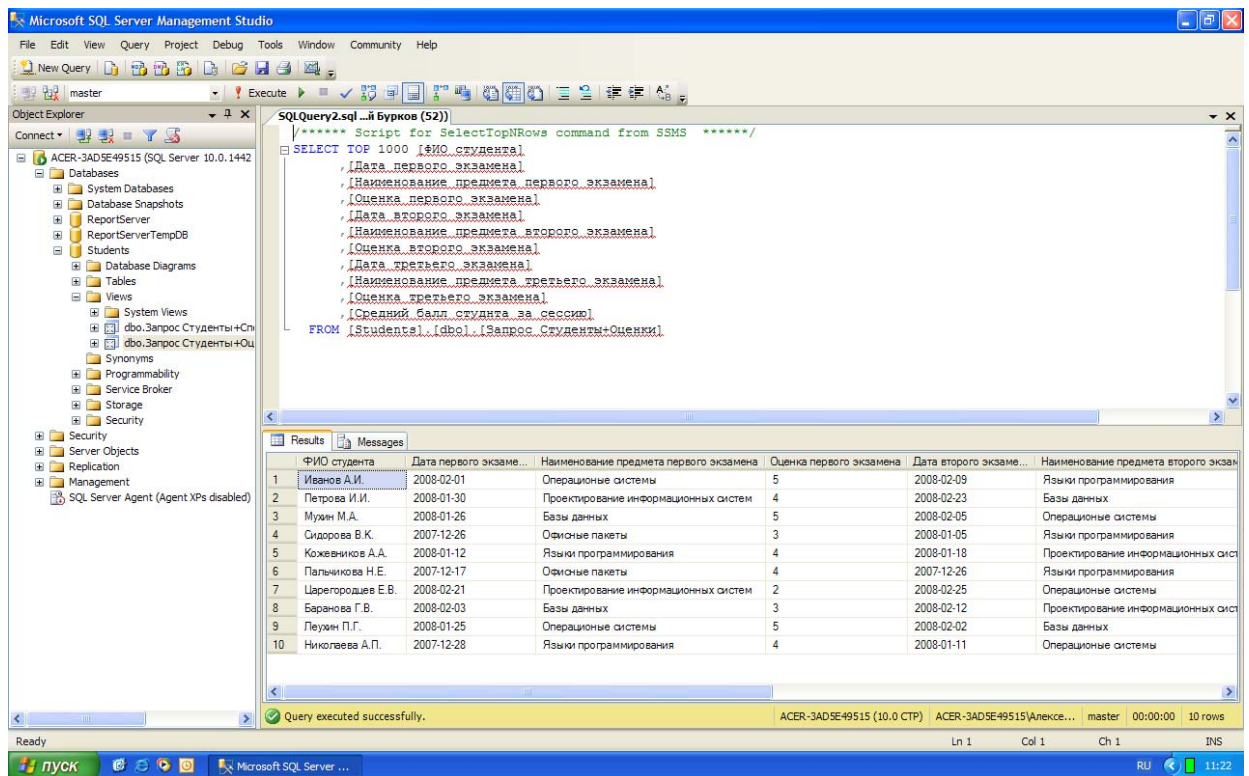


Рис.4.10

На этом мы заканчиваем рассмотрение обычных запросов и переходим к созданию фильтров.

На основе запроса «Запрос Студенты+Специальности» создадим фильтры, отображающие студентов отдельных специальностей. Создайте новый запрос. Так как он будет основан на запросе «Запрос Студенты+Специальности», то в окне «Add Table» перейдите на вкладку «Views» и добавьте в новый запрос «Запрос Студенты+Специальности» (Рис.4.11). Затем закройте окно «Add Table».

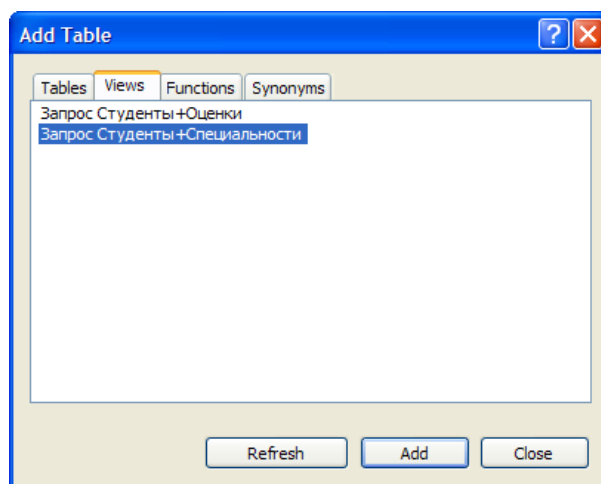


Рис.4.11

В появившемся окне конструктора запросов определите в качестве отображаемых полей все поля запроса «Запрос Студенты+Специальности» (Рис.4.12).

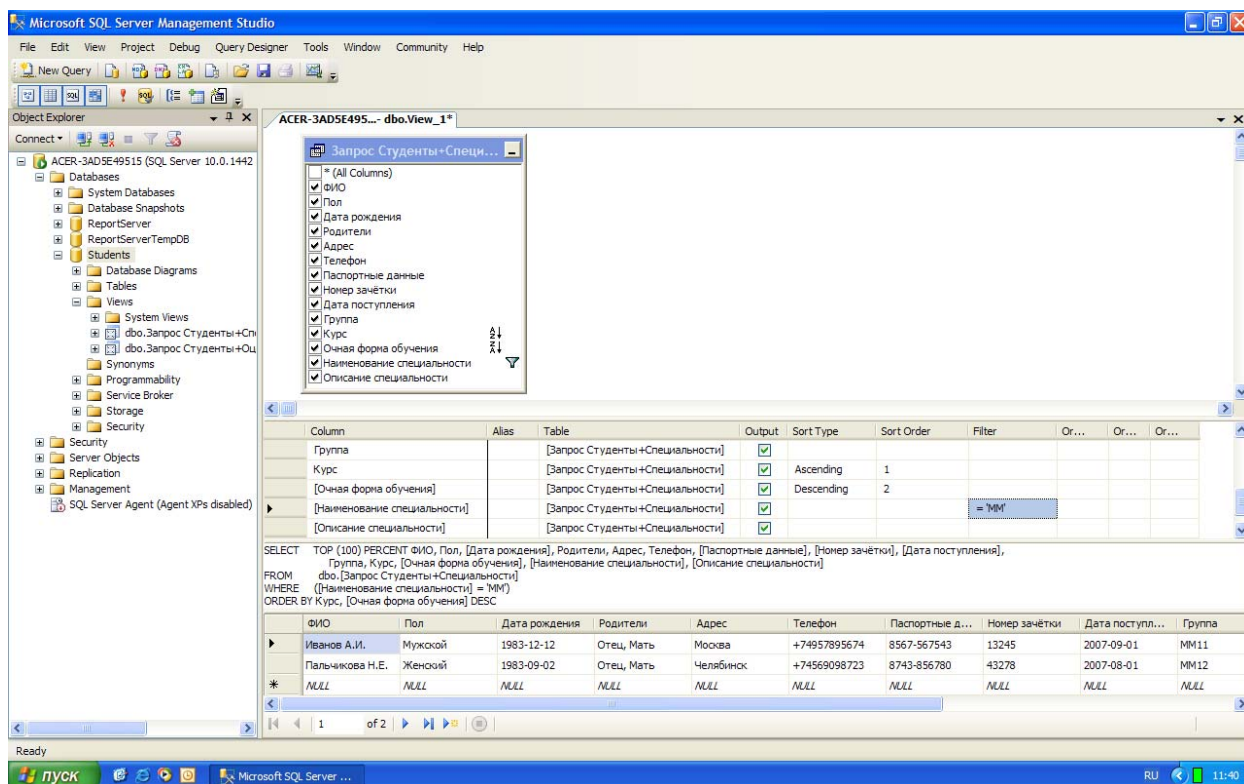


Рис.4.12

Замечание: Для отображения всех полей запроса, в данном случае, мы не можем использовать пункт «* (All Columns)» (Все поля). Так как в этом случае мы не можем устанавливать критерий отбора записей в фильтре, а также невозможно установить сортировку записей.

Теперь установим критерий отбора записей в фильтре. Пусть наш фильтр отображает только студентов имеющих специальность «ММ». Для определения условия отбора записей в таблице отображаемых полей в строке, соответствующей полю, на которое накладывается условие, в столбце «Filter», необходимо задать условие. В нашем случае условие накладывается на поле «Наименование специальности». Следовательно, в строке «Наименование специальности», в столбце «Filter» нужно задать следующее условие отбора «='ММ'» (Рис.4.12).

В заключение настроим сортировку записей в фильтре. Пусть при выполнении фильтра сначала происходит сортировка записей по возрастанию по полю «Очная форма обучения», а затем по убыванию по полю «Курс». Для установки сортировки записей по возрастанию, в таблице определяемых полей, в строке для поля «Очная форма обучения», в столбце «Sort Type» (Тип сортировки), задайте «Ascending» (По возрастанию), а в строке для поля «Курс» - задайте «Descending» (По убыванию). Для определения порядка сортировки для поля «Очная форма обучения» в столбце «Sort Order» (Порядок сортировки) поставьте 1, а для поля «Курс» поставьте 2 (Рис.4.12). То есть, при выполнении запроса записи сначала сортируются по полю «Очная форма обучения», а затем по полю «Курс».

Замечание: После установки условий отбора и сортировки записей на схеме данных напротив соответствующих полей появятся специальные значки. Значки \uparrow и \downarrow обозначают сортировку по возрастанию и убыванию, а значок ∇ показывает наличие условия отбора.

После установки сортировки записей в фильтре проверим его работоспособность, выполнив его. Результат выполнения фильтра должен выглядеть как на рисунке 4.12. Закройте окно конструктора запросов. В качестве имени нового фильтра в окне «Choose Name» задайте «Фильтр ММ» (Рис.4.13) и нажмите кнопку «Ok».

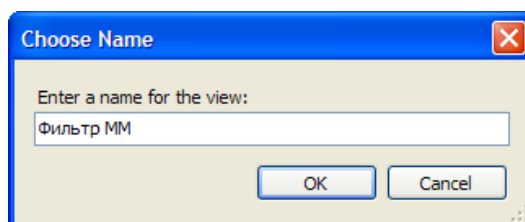


Рис.4.13

Фильтр «Фильтр ММ» появится в обозревателе объектов. Выполните созданный фильтр вне окна конструктора запросов. Результат должен быть таким же как на рисунке 4.14.

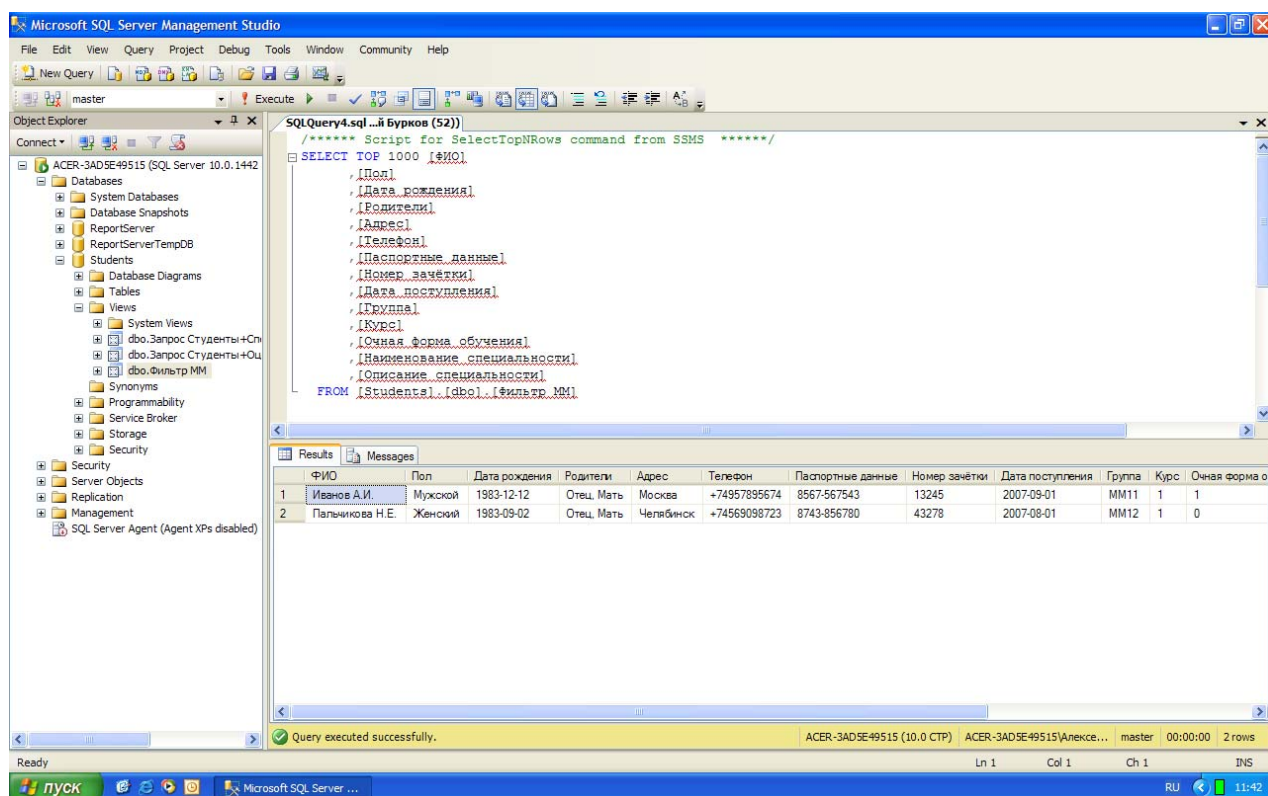


Рис.4.14

Самостоятельно создайте фильтры для отображения других специальностей. Данные фильтры создаются аналогично фильтру «Фильтр ММ» (смотри выше). Единственным отличием является условие отбора, накладываемое на поле «Наименование специальности», оно должно быть не «='ММ'», а «='ПИ'», «='СТ'», «='МО'» или «='БУ'». При сохранении фильтров задаём их имена соответственно их условиям отбора, то есть «Фильтр ПИ», «Фильтр СТ», «Фильтр МО» или «Фильтр БУ». Проверьте созданные фильтры на работоспособность.

Теперь на основе запроса «Запрос Студенты+Специальности» создадим фильтры, отображающие студентов имеющих отдельных родителей. Для начала создадим фильтр для студентов, из родителей только «Отец». Создайте новый запрос и добавьте в него запрос «Запрос Студенты+Специальности» (Рис.4.11). После закрытия окна «Add Table» сделайте отображаемыми все поля запроса (Рис.4.15).

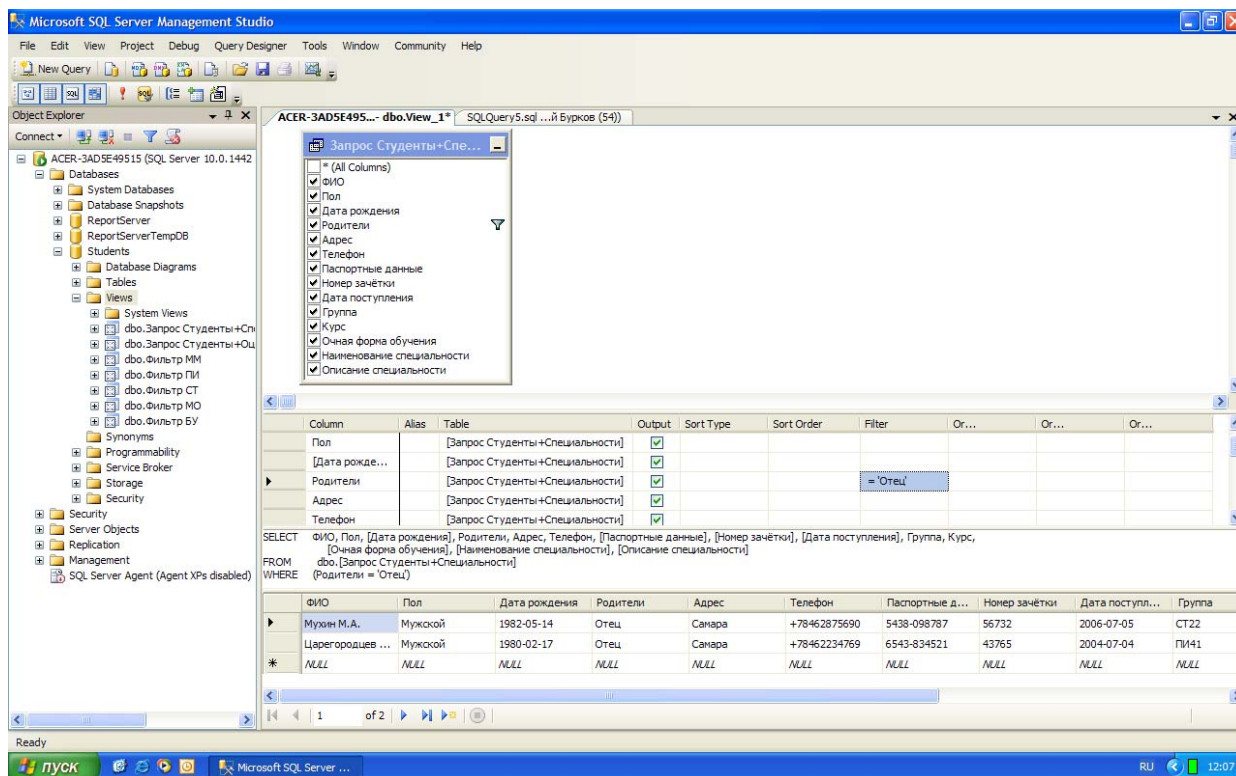


Рис.4.15

В таблице отображаемых полей в строке для поля «Родители», в столбце «Filter», задайте условие отбора равное «= 'Отец'». Проверьте работу фильтра, выполнив его. В результате выполнения фильтра окно конструктора запросов должно выглядеть как на рисунке 4.15.

Закройте окно конструктора запросов. В окне «Choose Name» задайте имя нового фильтра как «Фильтр Отец» (Рис.4.16).

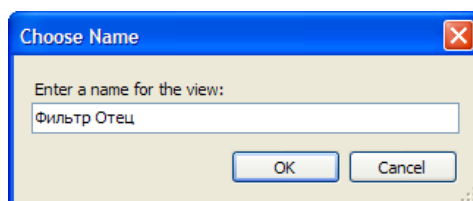


Рис.4.16

Выполните фильтр «Фильтр Отец» вне конструктора запросов. Результат должен быть аналогичен рисунку 4.17.

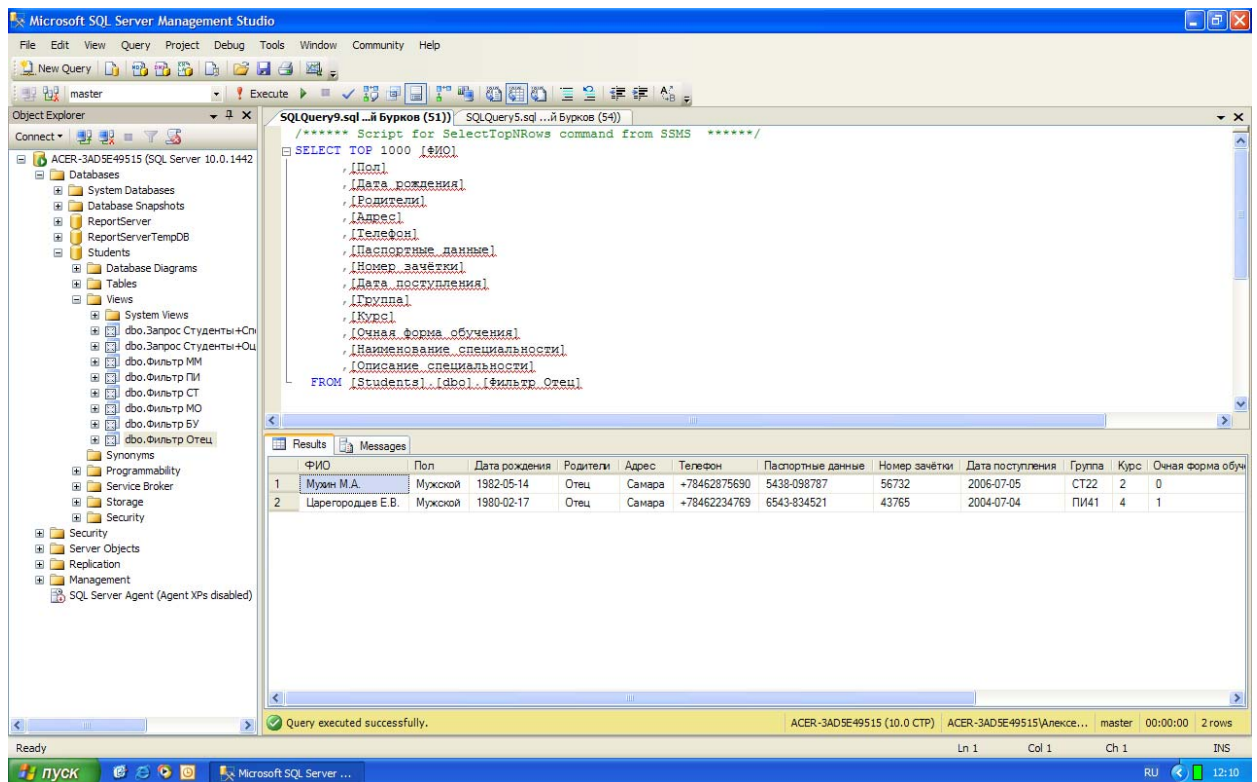


Рис.4.17

Создайте фильтры для отображения студентов с другими вариантами родителей. Данные фильтры создаются аналогично фильтру «Фильтр Отец» (смотри выше). Единственным отличием является условие отбора, накладываемое на поле «Родители», оно должно быть не «='Отец'», а «='Мать'», «='Отец, Мать'» или «='Нет'». При сохранении фильтров задаём их имена соответственно их условиям отбора, то есть «Фильтр Мать», «Фильтр Отец и Мать» или «Фильтр Нет родителей». Проверьте созданные фильтры на работоспособность.

Наконец создадим фильтры для отображения студентов очной и заочной формы обучения. Начнём с очной формы обучения. Создайте новый запрос и добавьте в него запрос «Запрос Студенты+Специальности». Как и ранее сделайте все поля запроса отображаемыми (Рис.4.18).

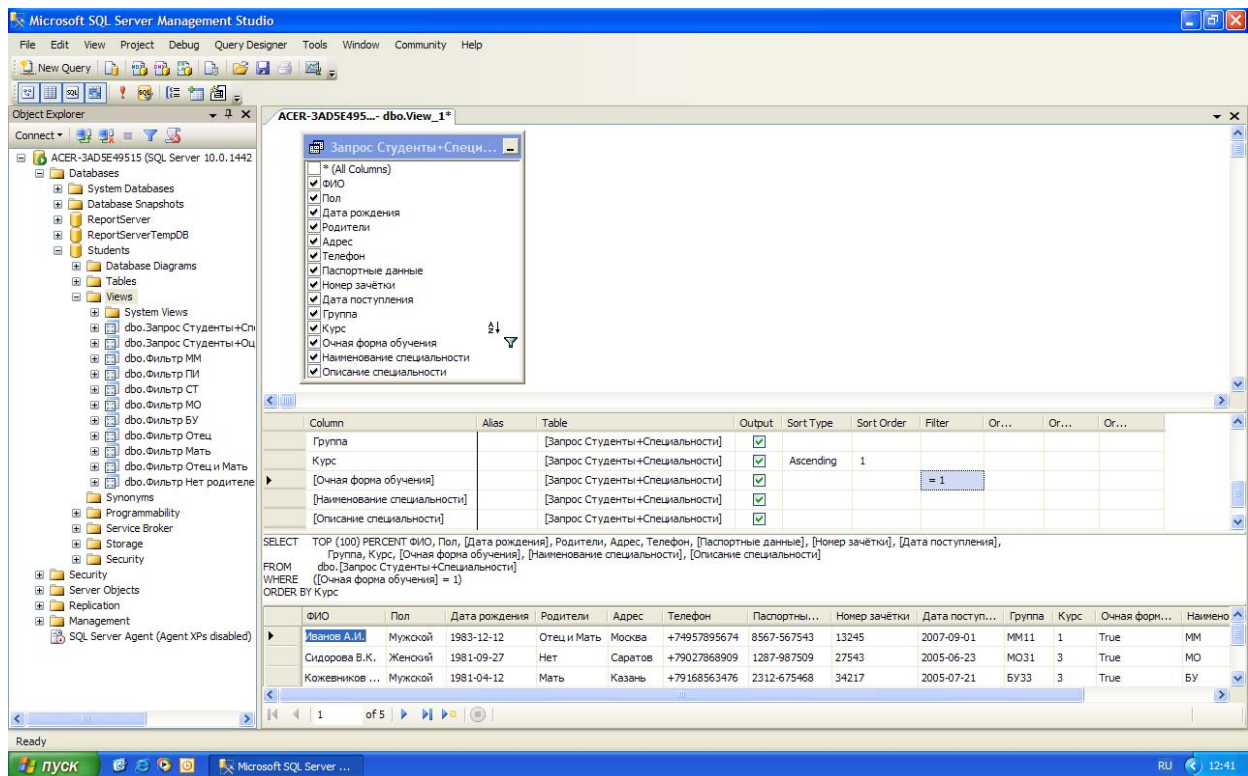


Рис.4.18

В таблице отображаемых полей в столбце «Filter», в строке для поля «Очная форма обучения» установите условие отбора равное «=1»

Замечание: Поле «Очная форма обучения» является логическим полем, оно может принимать значения либо «True» (Истина), либо «False» (Ложь). В качестве синонимов этих значений в «Microsoft SQL Server 2008» можно использовать 1 и 0 соответственно.

Установите сортировку по возрастанию, по полю курс, задав в строке для этого поля, в столбце «Sort Type», значение «Ascending».

Проверьте работу фильтра, выполнив его. После выполнения фильтра окно конструктора запросов должно выглядеть точно также как на рисунке 4.18.

Закройте окно конструктора запросов. Сохраните фильтр под именем «Фильтр очная форма обучения» (Рис.4.19).

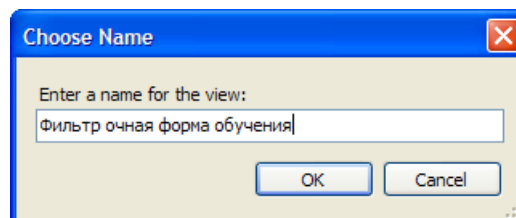


Рис.4.19

После появления фильтра «Фильтр очная форма обучения» в обозревателе объектов выполните фильтр вне окна конструктора запросов. Результат выполнения фильтра «Фильтр очная форма обучения» представлен на рисунке 4.20.

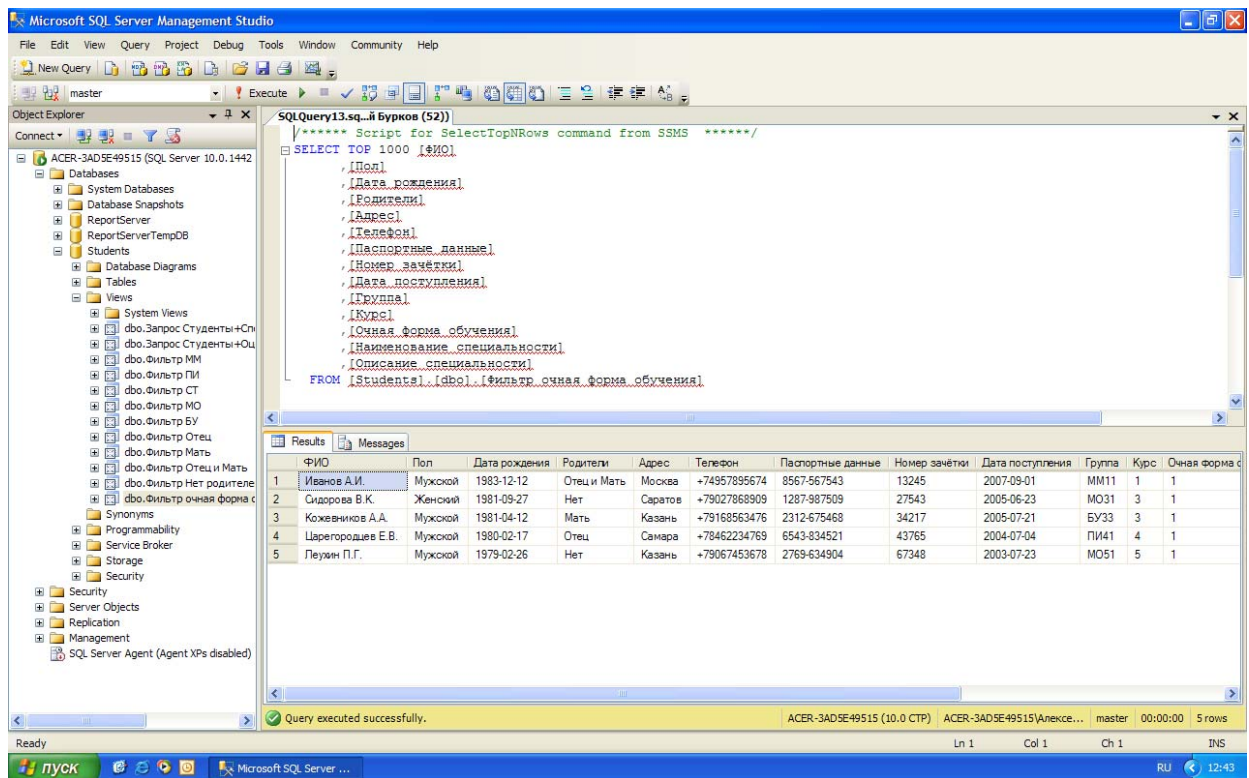


Рис.4.20

Самостоятельно создайте фильтр для отображения студентов заочной формы обучения. Данный фильтр создаётся точно также как и фильтр «Фильтр очная форма обучения». Единственным отличием является условие отбора, накладываемое на поле «Очная форма обучения», оно должно быть не «=1», а «=0». При сохранении фильтра задайте его имя как «Фильтр заочная форма обучения». Проверьте созданный фильтр на работоспособность.

В итоге, после создания всех запросов и фильтров окно обозревателя объектов должно выглядеть следующим образом (Рис.4.21):

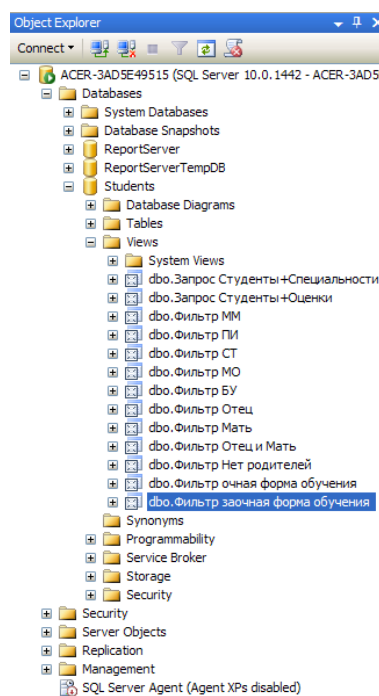


Рис.4.21

Лабораторная работа 5. Хранимые процедуры

Цель: научиться работать с хранимыми процедурами

Перейдём к созданию хранимых процедур. Для работы с хранимыми процедурами в обозревателе объектов необходимо выделить папку «Programmability/Stored Procedures» базы данных «Students» (Рис.5.1).

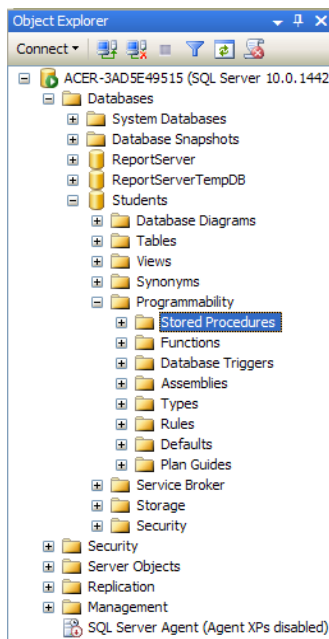


Рис.5.1

Создадим процедуру, вычисляющую среднее трёх чисел. Для создания новой хранимой процедуры щёлкните ПКМ по папке «Stored Procedures» (Рис.5.1) и в появившемся меню выберите пункт «New Stored Procedure». Появится окно кода новой хранимой процедуры (Рис.5.2).

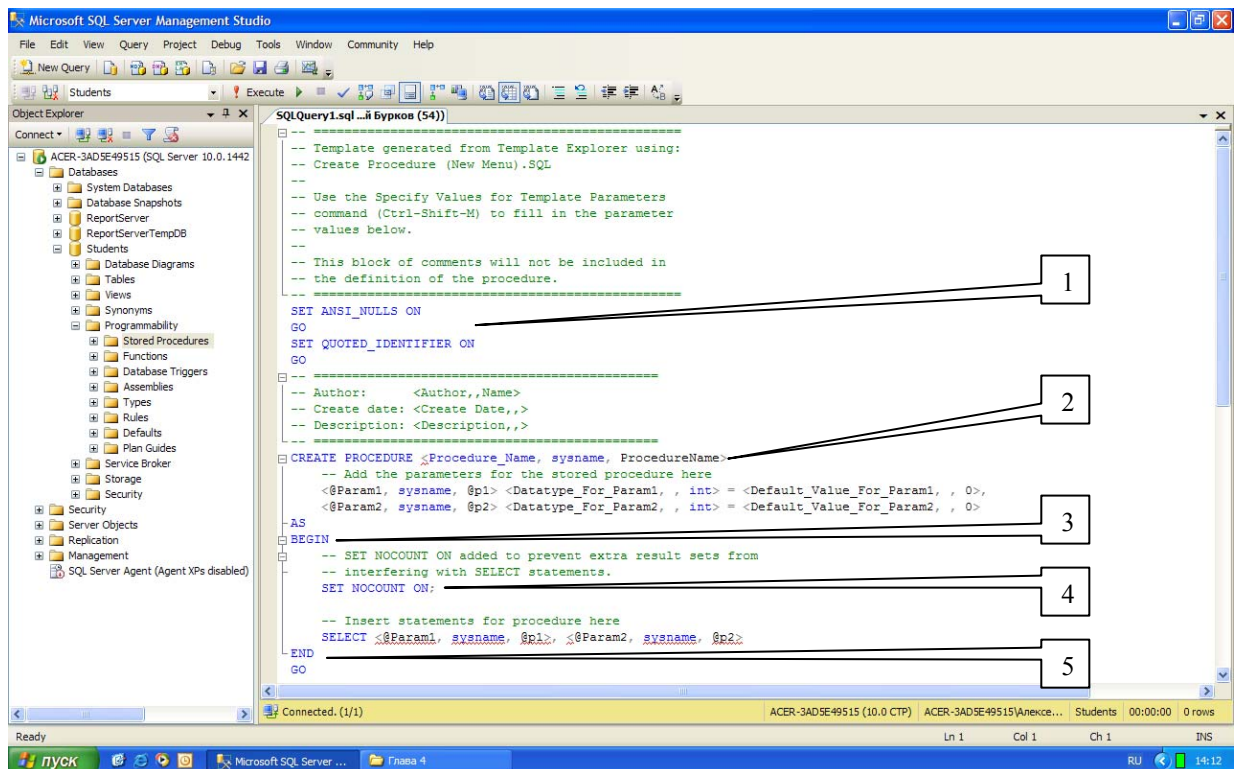


Рис.5.2

Хранимая процедура имеет следующую структуру (Рис.5.2):

1. Область настройки параметров синтаксиса процедуры. Позволяет настраивать некоторые синтаксические правила, используемые при наборе кода процедуры. В нашем случае это:
 - SET ANSI_NULLS ON – включает использование значений NULL (Пусто) в кодировке ANSI,
 - SET QUOTED_IDENTIFIER ON - включает возможность использования двойных кавычек для определения идентификаторов;
2. Область определения имени процедуры (Procedure_Name) и параметров передаваемых в процедуру (@Param1, @Param2). Определение параметров имеет следующий синтаксис:
 @<Имя параметра> <Тип данных> = <Значение по умолчанию>
 Параметры разделяются между собой запятыми;
3. Начало тела процедуры, обозначается служебным словом «BEGIN»;
4. Тело процедуры, содержит команды языка программирования запросов T-SQL;
5. Конец тела процедуры, обозначается служебным словом «END».

Замечание: В коде зелёным цветом выделяются комментарии. Они не обрабатываются сервером и выполняют функцию пояснений к коду. Строки комментариев начинаются с подстроки «--». Далее в коде, мы не будем отображать комментарии, они будут свёрнуты. Слева от раздела с комментариями будет стоять знак «+», щёлкнув по которому можно развернуть комментарий.

Наберём код процедуры вычисляющей среднее трёх чисел, как это показано на рисунке 5.3.

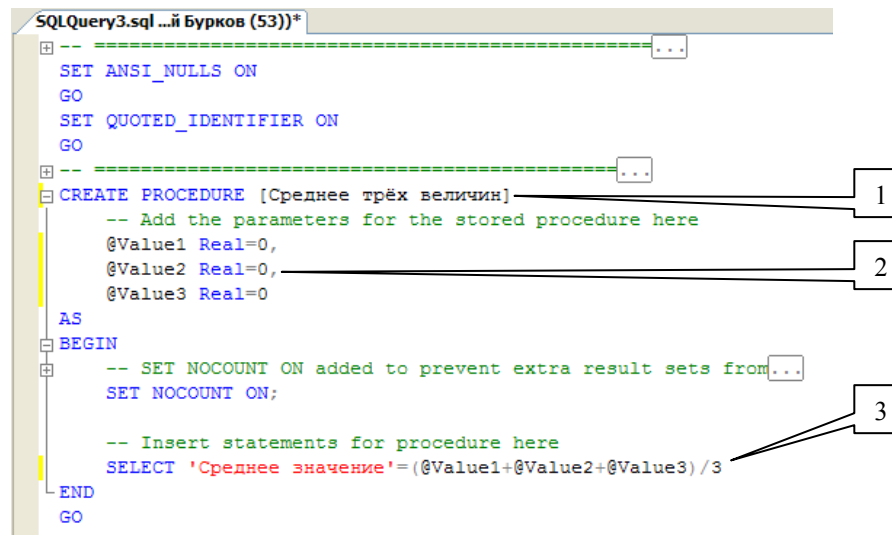
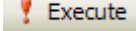



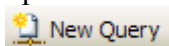
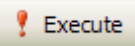
Рис.5.3

Рассмотрим код данной процедуры более подробно (Рис.5.3):

1. CREATE PROCEDURE [Среднее трёх величин] – определяет имя создаваемой процедуры как «Среднее трёх величин»;
2. @Value1 Real = 0, @Value2 Real = 0, @Value3 Real = 0 - определяют три параметра процедуры Value1, Value2 и Value3. Данным параметрам можно присвоить дробные числа (Тип данных Real), значения по умолчанию равны 0;
3. SELECT 'Среднее значение'=(@Value1+@Value2+@Value3)/3 – вычисляет среднее и выводит результат с подписью «Среднее значение».

Остальные фрагменты кода рассмотрены выше (Рис.5.2).

Для создания процедуры, выполним вышеописанный код, нажав кнопку  Execute (Выполнить) на панели инструментов. В нижней части окна с кодом появится сообщение «Command(s) completed successfully.». Закройте окно с кодом, щёлкнув мышью по кнопке закрытия , расположенной в верхнем правом углу окна с кодом процедуры.

Проверим работоспособность созданной хранимой процедуры. Для запуска хранимой процедуры необходимо создать новый пустой запрос, нажав на кнопку  New Query (Новый запрос) на панели инструментов. В появившемся окне с пустым запросом наберите команду EXEC [Среднее трёх величин] 1, 7, 9 и нажмите кнопку  Execute на панели инструментов (Рис.5.4).

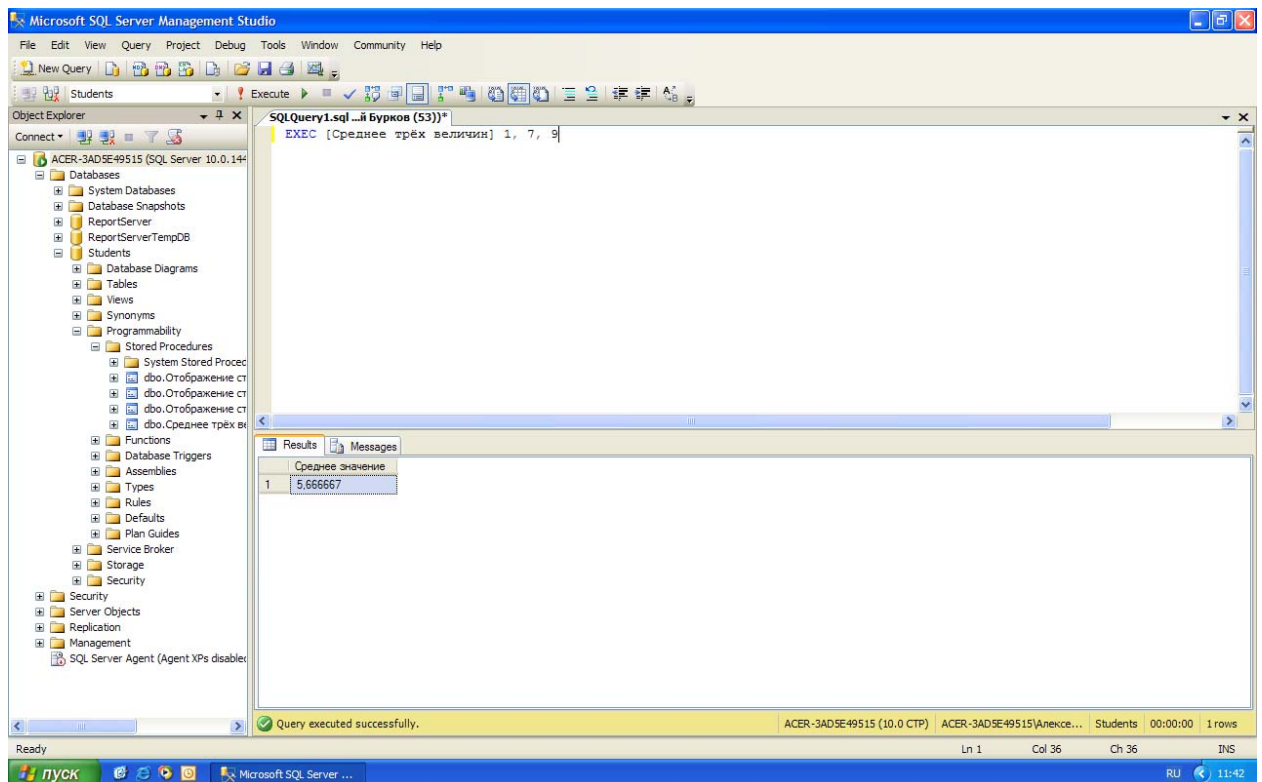


Рис.5.4

В нижней части окна с кодом появиться результат выполнения новой хранимой процедуры: Среднее значение 5,66667 (Рис.5.4).

Теперь создадим хранимую процедуру для отбора студентов из таблицы студенты по их «ФИО». Для этого создайте новую хранимую процедуру, как это описано выше, и наберите код новой процедуры как на рисунке 5.5.

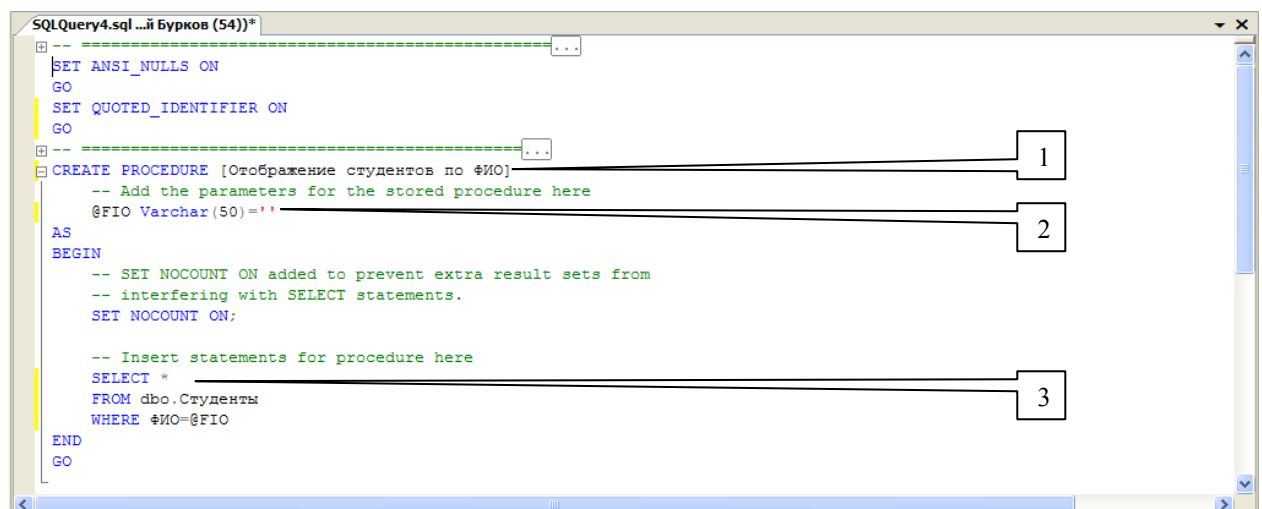


Рис.5.5

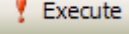
Рассмотрим код процедуры «Отображение студентов по ФИО» более подробно (Рис.5.5):

1. `CREATE PROCEDURE [Отображение студентов по ФИО]` – определяет имя создаваемой процедуры как «Отображение студентов по ФИО»;
2. `@FIO Varchar(50)=' '` - определяют единственный параметр процедуры FIO. Параметру можно присвоить текстовые сроки переменной длины,

длинной до 50 символов (Тип данных Varchar(50)), значения по умолчанию равны пустой строке;

3. SELECT * FROM dbo.Студенты WHERE ФИО=@ФИО – отобразить все поля (*) из таблицы студенты (dbo.Студенты), где значение поля ФИО равно значению параметра ФИО (ФИО=@ФИО).

Выполним вышеописанный код и закроем окно с кодом, как описано выше.

Проверим работоспособность созданной хранимой процедуры. Создайте новый пустой запрос. В появившемся окне с пустым запросом наберите команду EXEC [Отображение студентов по ФИО] 'Иванов А.И.' и нажмите кнопку  на панели инструментов (Рис.5.6).

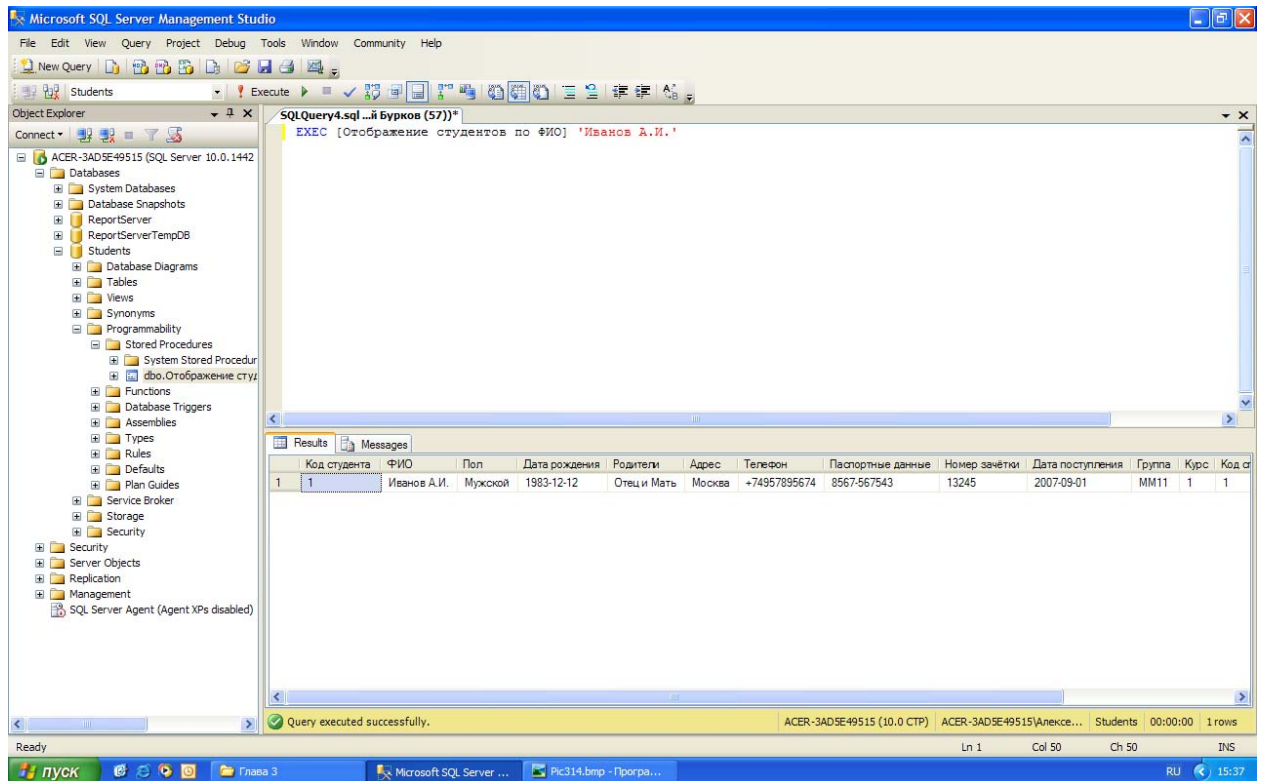


Рис.5.6

В нижней части окна с кодом появится результат выполнения хранимой процедуры «Отображение студентов по ФИО» (Рис.5.6).

Теперь перейдём к более сложной задаче – отобразить студентов, у которых средний балл выше заданного. Создайте новую хранимую процедуру и наберите код новой процедуры как на рисунке 5.7.

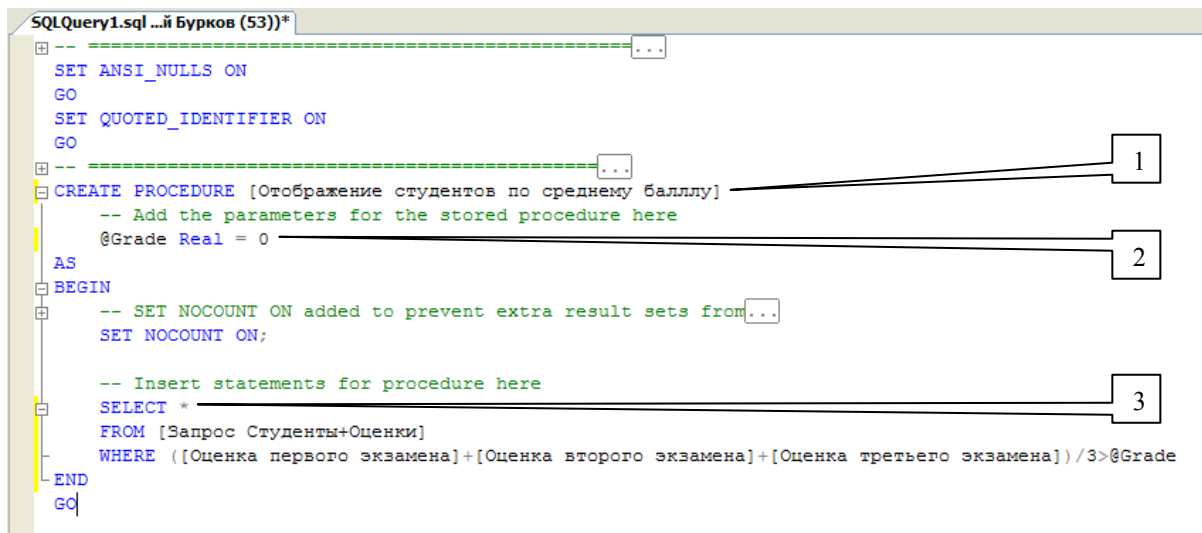


Рис.5.7

Рассмотрим код процедуры «Отображение студентов по ФИО» более подробно (Рис.5.7):

1. CREATE PROCEDURE [Отображение студентов по среднему баллу] – определяет имя создаваемой процедуры как «Отображение студентов по среднему баллу»;
2. @Grade Real=0 - определяют параметр процедуры Grade. Параметру можно присвоить дробные числа (Тип данных Real), значения по умолчанию равны 0;
3. SELECT * FROM [Запрос Студенты+Оценки] WHERE ([Оценка первого экзамена]+[Оценка второго экзамена]+[Оценка третьего экзамена])/3>@Grade– отобразить все поля (*) из запроса «Запрос Студенты+Оценки» (Запрос Студенты+Оценки), где средний балл больше чем значение параметра Grade (([Оценка первого экзамена]+[Оценка второго экзамена]+[Оценка третьего экзамена])/3>@Grade).

Выполним вышеописанный код и закроем окно с кодом, как описано выше. Проверим, как работает запрос, описанный выше. Для этого, создайте новый запрос и в нём наберите команду EXEC [Отображение студентов по среднему баллу] 3.5 и выполните её (Смотри выше) (Рис.5.8).

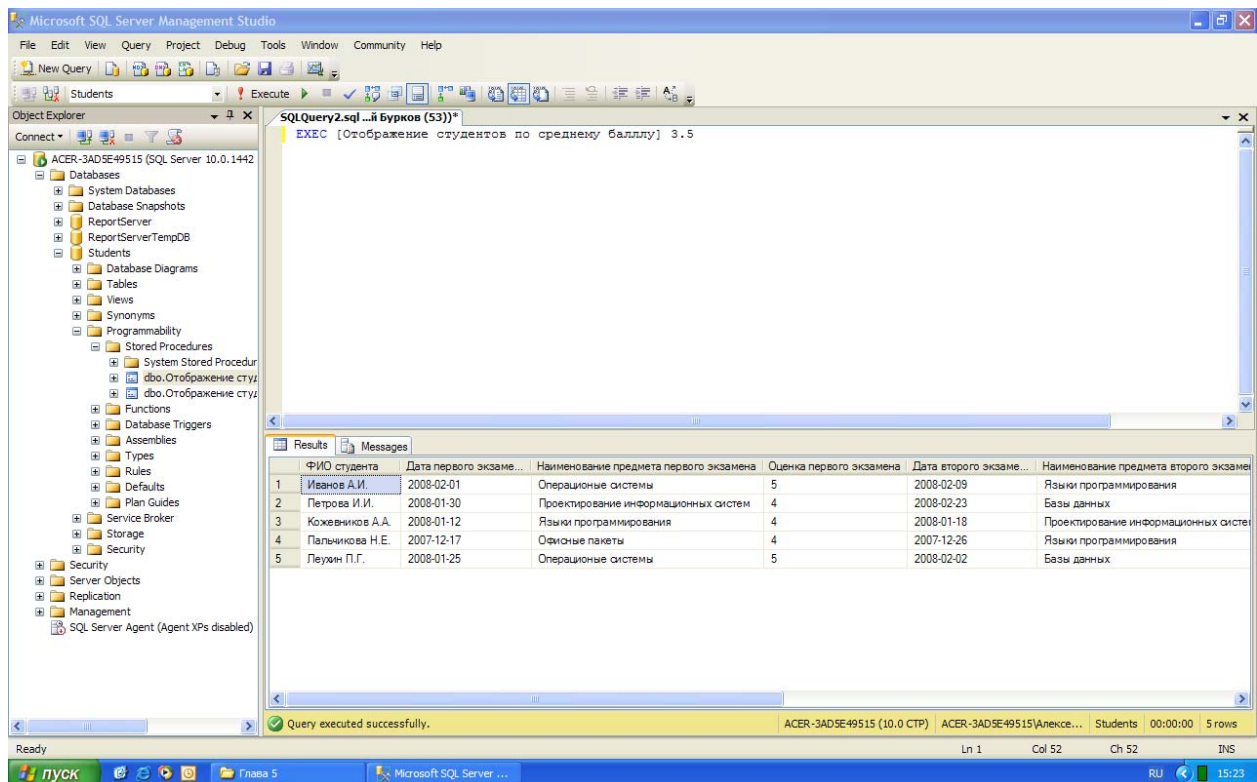


Рис.5.8

В нижней части окна с кодом появиться результат выполнения хранимой процедуры «Отображение студентов по среднему баллу» (Рис.5.8).

В заключение решим более сложную задачу- отображение студентов старше заданного возраста. При чём возраст будет автоматически вычисляться в зависимости от даты рождения.

Создадим новую хранимую процедуру и наберём код новой процедуры как представлено на рисунке 5.9.

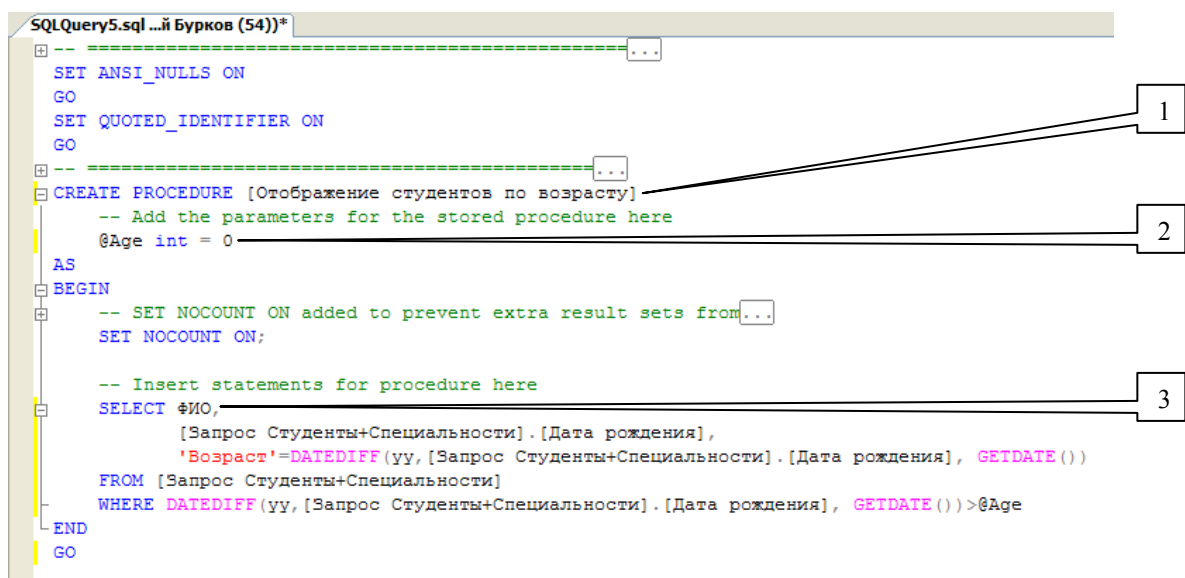


Рис.5.9

Рассмотрим код создаваемой процедуры «Отображение студентов по возрасту» более подробно (Рис.5.9):

1. CREATE PROCEDURE [Отображение студентов по возрасту] – определяет имя создаваемой процедуры как «Отображение студентов по возрасту»;
2. @Age int=0 - определяют параметр процедуры Grade. Параметру можно присвоить целые числа (Тип данных int), значения по умолчанию равны 0;
3. ФИО, [Запрос Студенты+Специальности].[Дата рождения], 'Возраст'=DATEDIFF (yy, [Запрос Студенты+Специальности].[Дата рождения], GETDATE()) – отображает из запроса «Запроса Студенты+Специальности» (FROM [Запрос Студенты+Специальности]) поля «ФИО» (ФИО) и «Дата рождения» ([Запрос Студенты+Специальности].[Дата рождения]), а также отображает возраст студента ('Возраст') в годах (yy), вычисленный исходя из его даты рождения и текущей даты (DATEDIFF(yy,[Запрос Студенты+Специальности].[Дата рождения], GETDATE())). Более того, выводятся студенты возраст которых больше определённого в параметре «Age» (DATEDIFF(yy,[Запрос Студенты+Специальности].[Дата рождения], GETDATE())>@Age).

Замечание: Встроенная функция DATEDIFF вычисляющая количество периодов между двумя датами, имеет следующий синтаксис: DATEDIFF(<период>, <начальная дата>, <конечная дата>)

Выполним код запроса «Отображение студентов по возрасту», а затем закроем окно с кодом, как описано выше. Проверим, как работает запрос. Для этого, создадим новый запрос и в нём наберём команду EXEC [Отображение студентов по возрасту] 26 и выполните её. Должен появиться результат аналогичный результату, представленному на рисунке 5.10.

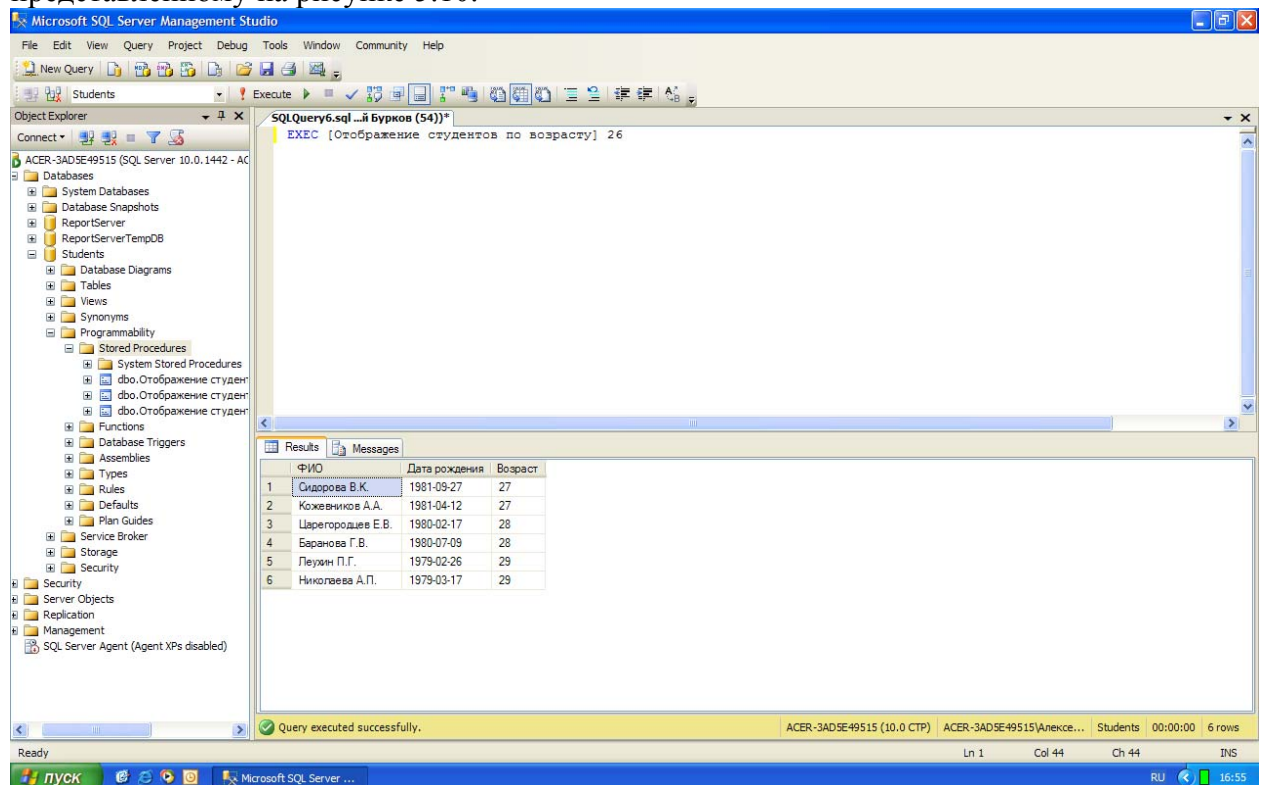


Рис.5.10

На этом мы заканчиваем описание хранимых процедур и переходим к рассмотрению пользовательских функций. В итоге, обозреватель объектов должен иметь вид как на рисунке 5.11.

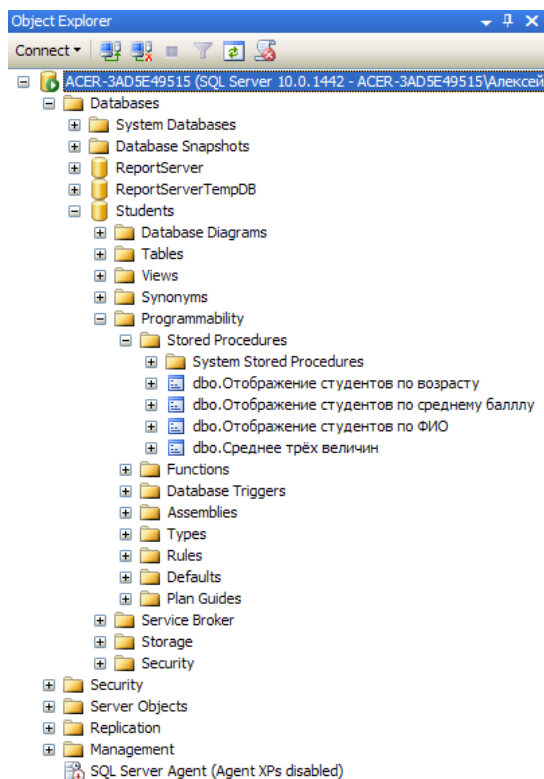


Рис.5.11

Лабораторная работа 6. Пользовательские функции

Цель: научиться работать с пользовательскими функциями

Теперь рассмотрим создание и применение пользовательских функций. В БД «Microsoft SQL Server 2008» все пользовательские функции находятся в папке «Functions» расположенной в папке «Programmability» в обозревателе объектов (Рис.6.1).

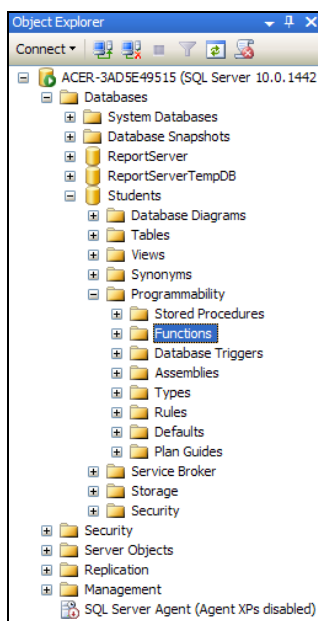


Рис.6.1

Начнём с создания скалярных пользовательских функций. Для создания новой скалярной пользовательской функции в обозревателе объектов, в БД «Students», в папке «Programmability», щёлкните ПКМ по папке «Functions» и в появившемся меню выберите пункт «New/Scalar-valued Function». Появится окно новой скалярной пользовательской функции (Рис.6.2)

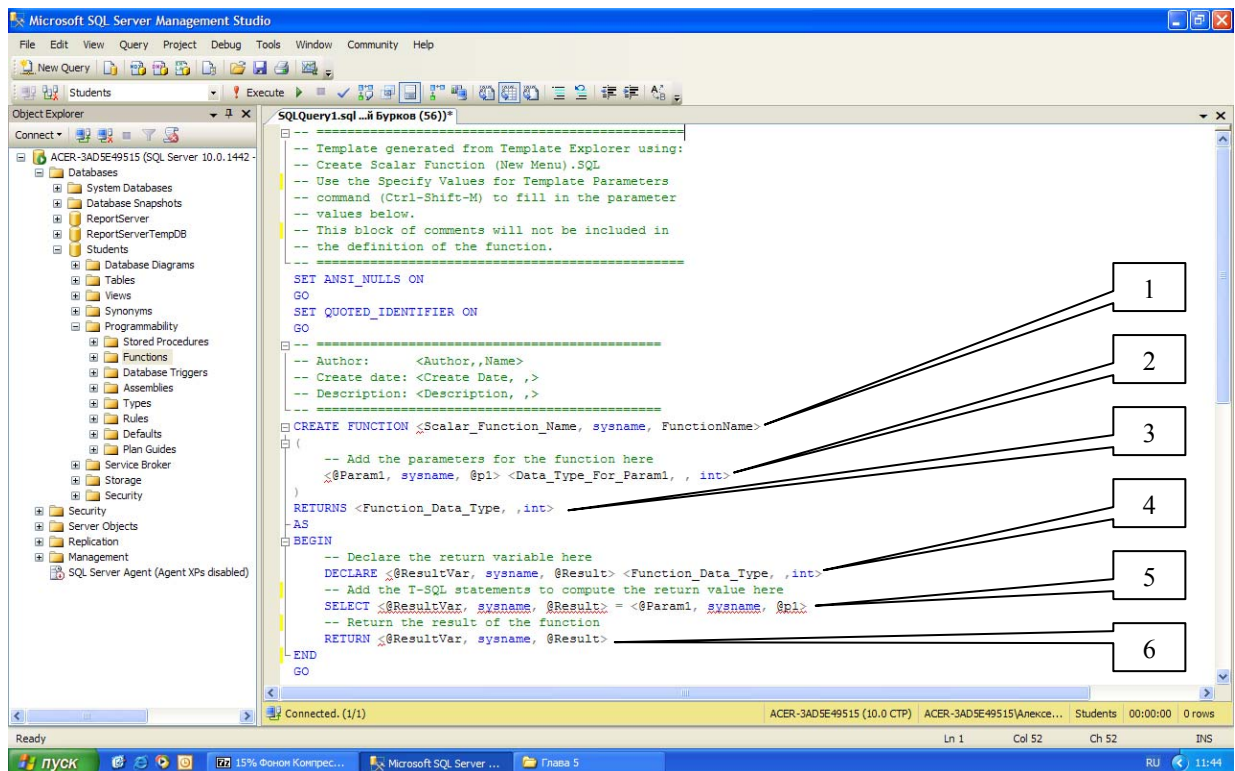


Рис.6.2

Синтаксис скалярной пользовательской функции похож на синтаксис хранимой процедуры (см. занятие 5). Однако имеется ряд существенных отличий (Рис.6.2):

6. Область определения имени функции (Scalar_Function_Name);
7. Параметры, передаваемые в процедуру (@Param1). Определение параметров аналогично определению параметров в хранимой процедуре (см. занятие 5);
8. Тип данных значения возвращаемого процедурой;
9. Область объявления переменных, используемых внутри функции. Объявление переменных имеет следующий синтаксис:
`DECLARE @<Имя переменной> <Тип данных>`
10. Тело самой пользовательской функции, содержит команды языка программирования запросов T-SQL;
11. Команда RETURN возвращающая результат выполнения функции. Имеет следующий синтаксис:

`RETURN @<Имя переменной с результатом>`

Переменная должна быть того же типа данных, который был указан в пункте 3.

Создадим скалярную пользовательскую функцию, вычисляющую среднее трёх величин. В окне новой пользовательской функции наберите код представленный на рисунке 6.3.

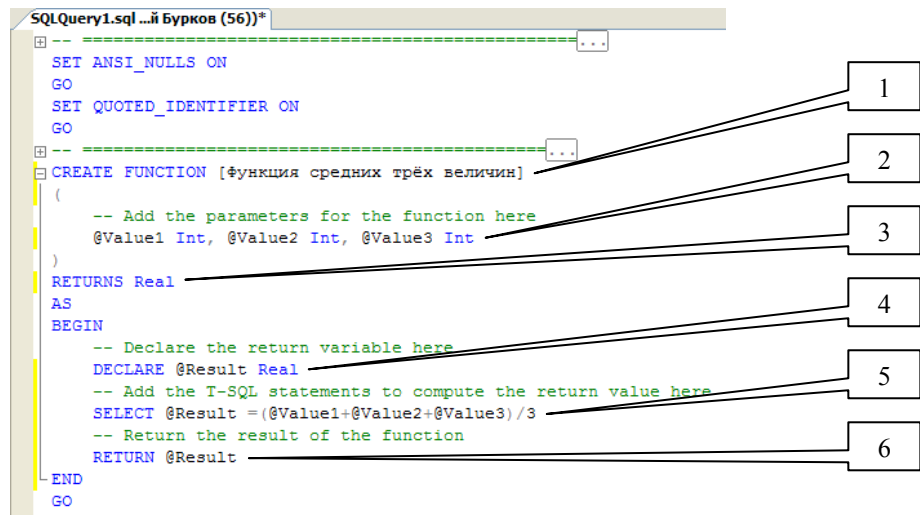
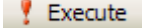



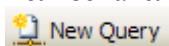
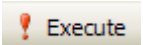
Рис.6.3

Рассмотрим более подробно код данной скалярной пользовательской функции (Рис.6.3):

4. CREATE FUNCTION [Функция средних трёх величин] – определяет имя создаваемой функции как «Функция средних трёх величин»;
5. @Value1 Real, @Value2, @Value3 - определяют три параметра процедуры Value1, Value2 и Value3. Данным параметрам можно присвоить целые числа (Тип данных Int);
6. RETURNS Real – показывает, что функция возвращает дробные числа (Тип данных Real);
7. DECLARE @Result Real – объявляется переменная @Result для хранения результата работы функции, то есть дробного числа (Тип данных Real);
8. SELECT @Result=(@Value1+@Value2+@Value3)/3 – вычисляет среднее и помещает результат в переменную @Result;
9. RETURN @Result – возвращает значение переменной @Result.

Остальные фрагменты кода рассмотрены выше (Рис.6.2).

Для создания функции, выполним вышеописанный код, нажав кнопку  Execute (Выполнить) на панели инструментов. В нижней части окна с кодом появится сообщение «Command(s) completed successfully». Закройте окно с кодом, щёлкнув мышью по кнопке закрытия , расположенной в верхнем правом углу окна с кодом функции.

Проверим работу созданной скалярной пользовательской функции. Для запуска пользовательской функции необходимо создать новый пустой запрос, нажав на кнопку  New Query (Новый запрос) на панели инструментов. В появившемся окне с пустым запросом наберите команду SELECT dbo.[Функция средних трёх величин] (3, 5, 4) и нажмите кнопку  Execute на панели инструментов (Рис.6.4).

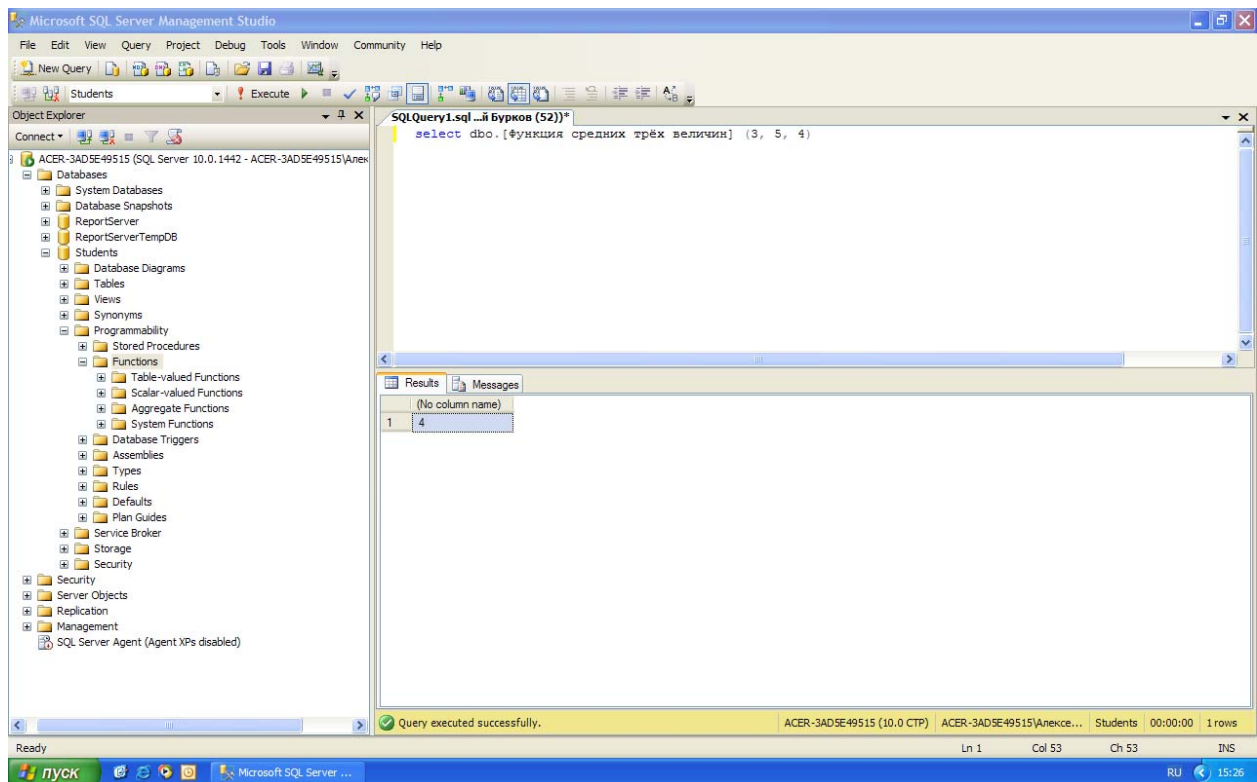


Рис.6.4

В нижней части окна с кодом появится результат выполнения новой скалярной пользовательской функции: 4 (Рис.6.4).

Теперь создадим более сложную скалярную пользовательскую функцию, предназначенную для определения последнего дня месяца введенной даты.

Создайте новую скалярную пользовательскую функцию, так как об этом сказано выше. В окне новой пользовательской функции наберите следующий код (Рис.6.5):

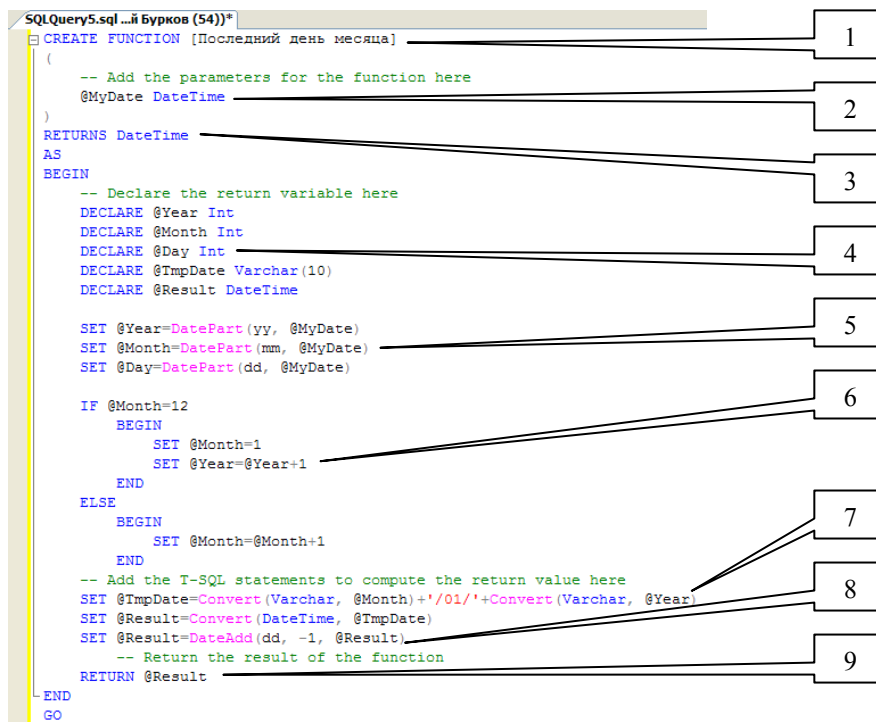


Рис.6.5

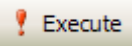
Перейдём к рассмотрению вышеприведенного кода (Рис.6.5). Код состоит из следующих групп команд:

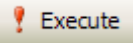
1. CREATE FUNCTION [Последний день месяца] – определяет имя создаваемой функции как «Последний день месяца»;
2. @MyDate - определяют параметр процедуры MyDate. Параметру можно присвоить значения дат или времени (Тип данных DateTime);
3. RETURNS DateTime – показывает, что функция возвращает дату или время (Тип данных DateTime);
4. DECLARE @Year Int, DECLARE @Month Int, DECLARE @Day Int – объявляются переменные @Year, @Month и @Day для хранения целочисленных значений года, месяца и дня введенной даты (Тип данных Int).
DECLARE @TmpDate VarChar(10) объявляет переменную «TmpDate» для хранения промежуточного значения даты в строке длиной до 10 символов (Тип данных VarChar(10)).
DECLARE @Result DateTime объявляет переменную «Result» для хранения результата – даты последнего дня месяца (Тип данных DateTime).
5. SET @Year=DatePart(yy, @MyDate), SET @Month=DatePart(mm, @MyDate), SET @Day=DatePart(dd, @MyDate) – определяются части введенной даты и помещаются в переменные @Year, @Month и @Day. Для определения частей даты используется функция DatePart, имеющая следующий синтаксис: DatePart(<часть даты>, <дата>). Здесь «часть даты» - это закодированная специальными символами определяемая часть даты (yy – год, mm – месяц, dd - день), «дата» - это дата, части которой определяем.
6. IF @Month=12
 BEGIN
 SET @Month=1
 SET @Year=@Year+1
 END
ELSE
 BEGIN
 SET @Month=@Month+1
 END

Вышереведённый фрагмент кода выполняет следующие действия: Если номер месяца равен 12 то установить номер месяца (@Month) равным 1 и увеличить год (@Year) на 1, иначе увеличить месяц на 1.

7. SET @TmpDate=Convert(VarChar, @Month)+'/01/'+Convert(VarChar, @Year),
SET @Result=Convert(DateTime, @TmpDate) – переводит числовые значения даты в дату в строковом формате и записывает её в переменную @TmpDate, затем переводит дату в строковом формате в тип данных даты и времени и помещает её в переменную @Result. Для конвертации используется функция Convert, имеющая следующий синтаксис:
Convert(<тип данных>, <значение>), здесь «тип данных» это тип данных в который переводится «значение».
8. SET @Result=DateAdd(dd, -1, @Result) – из даты, хранимой в переменной @Result вычитается 1 день, для этого используется функция DateAdd, имеющая следующий синтаксис:
DateAdd(<часть даты>, <количество периодов>, <дата>) - здесь «часть даты» - это закодированная специальными символами определяемая часть даты (см. функцию DatePart), «количество периодов» - это количество частей даты прибавляемой к введенной дате (параметр «дата»).

9. RETURN @Result – возвращает значение, хранимое в переменной @Result.

Для создания функции, выполним вышеописанный код, как и в случае с предыдущей функцией, нажав кнопку . После появления сообщения «Command(s) completed successfully.» закройте окно с кодом.

Проверим работу функции «Последний день месяца» выполнив её. Создайте новый пустой запрос, затем в окне с пустым запросом наберите команду SELECT dbo.[Последний день месяца] ('12/07/08') и нажмите кнопку  на панели инструментов (Рис.6.6).

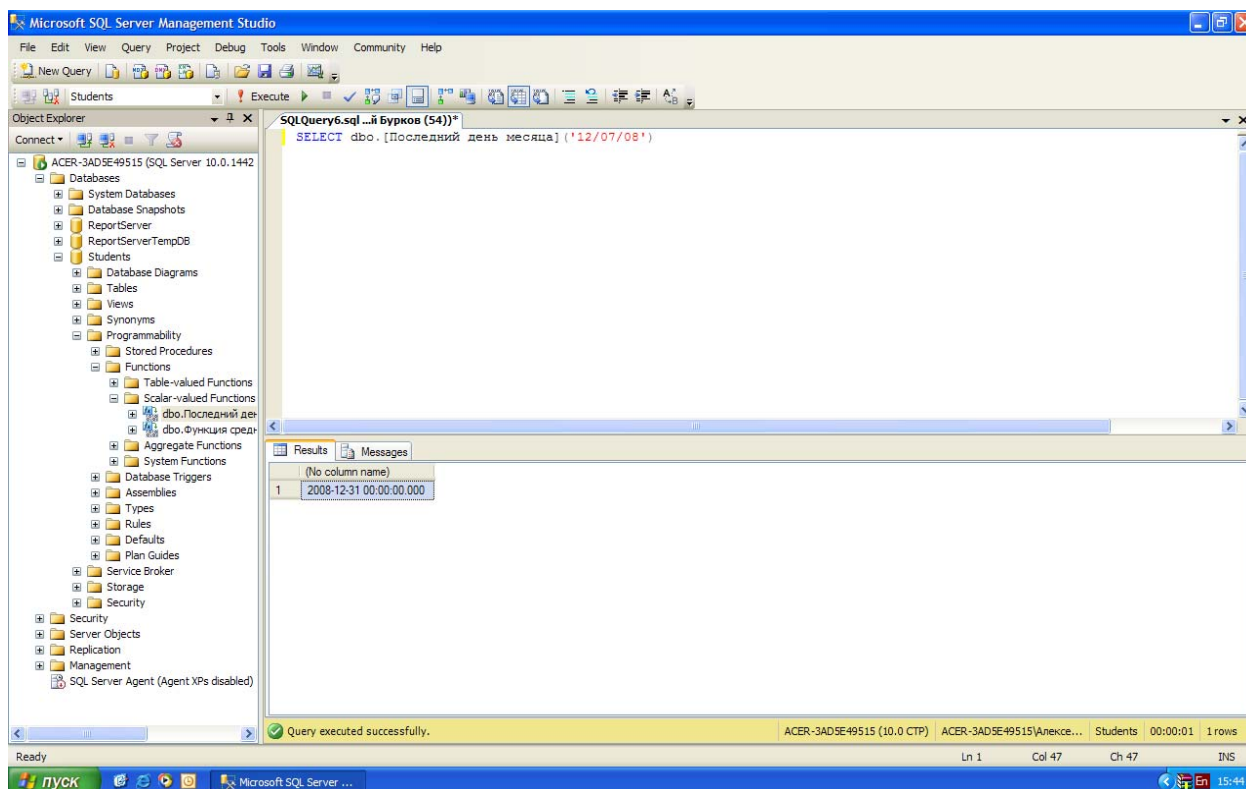


Рис.6.6

Появится результат выполнения новой скалярной пользовательской функции: 2008-12-31 (Рис.6.6).

Теперь перейдём к созданию табличных пользовательских функций. Для создания табличной пользовательской функции в обозревателе объектов, в БД «Students», в папке «Programmability», щёлкните ПКМ по папке «Functions» и в появившемся меню выберите пункт «New/Table-valued Function». Появится окно новой табличной пользовательской функции (Рис.6.7)

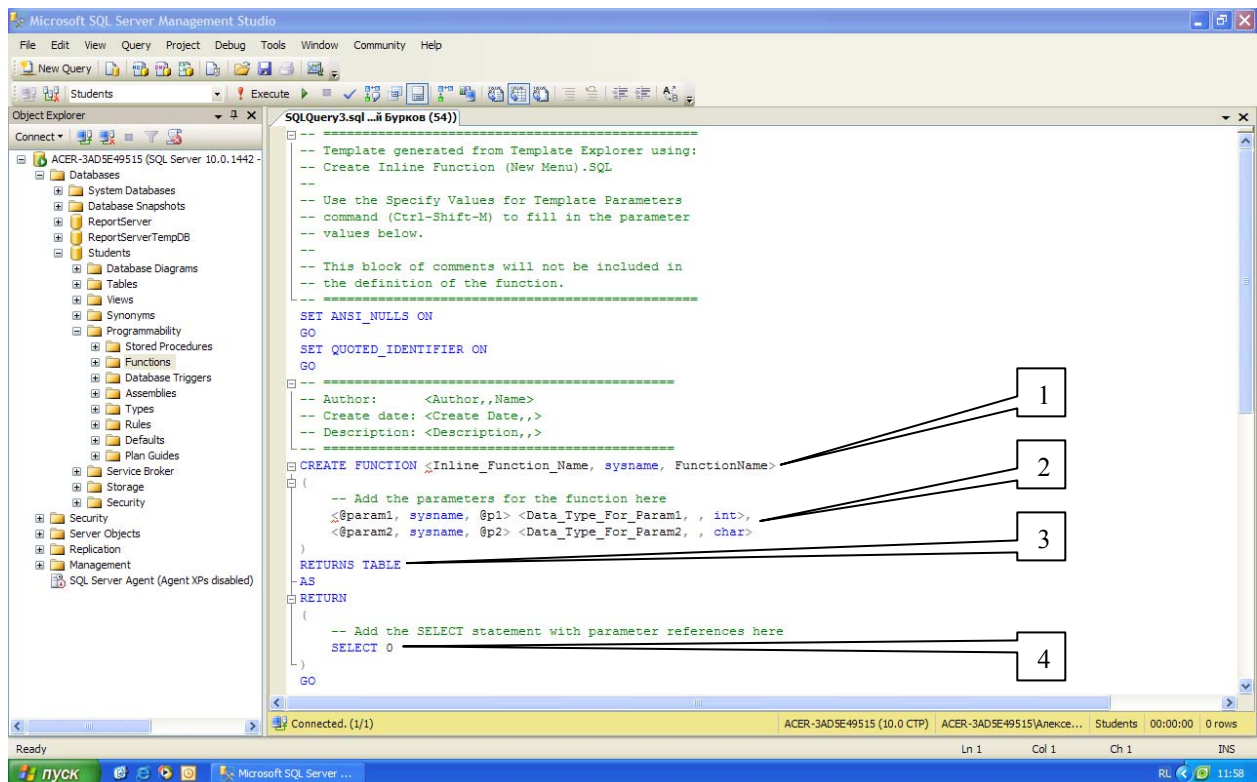


Рис.6.7

Рассмотрим структуру кода табличной пользовательской функции. Табличная пользовательская функция состоит из следующих разделов:

1. Область определения имени функции (Scalar_Function_Name);
2. Параметры, передаваемые в процедуру (@Param1, @Param2);
3. RETURNS TABLE показывает что функция является табличной, то есть возвращает таблицу;
4. Тело самой пользовательской функции, состоит из команды SELECT языка программирования запросов T-SQL.

Остальные разделы табличной пользовательской функции аналогичны таким же разделам хранимых процедур и скалярных пользовательских функций.

В заключение рассмотрим создание табличной пользовательской функции «Функция отбора по возрасту», вычисляющих текущий возраст студентов в зависимости от их даты рождения. В окне новой пользовательской функции (Рис.6.7) наберите следующий код (Рис.6.8):

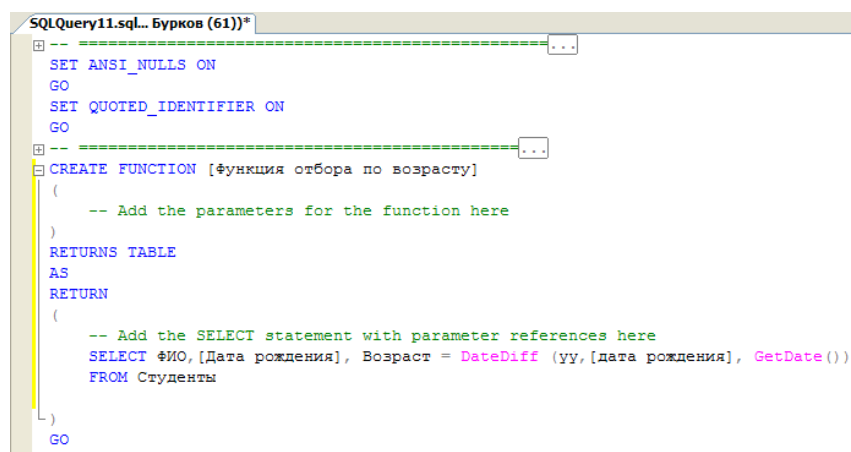


Рис.6.8

Из кода представленного на рисунке 6.8 видно, что данная табличная функция не имеет параметров и реализуется командой

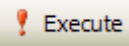
```
SELECT ФИО, [Дата рождения], Возраст = DateDiff(yy, [Дата рождения], GetDate())
FROM Студенты.
```

Из вышепредставленной команды видно, что из таблицы «Студенты» отображаются поля «ФИО» и «Дата рождения», а также вычисляемое поле «Возраст». Поле «Возраст» вычисляется при помощи встроенной функции DateDiff вычисляющей разницу между датами в определённых единицах измерения (частях даты) и имеющей следующий синтаксис:

DateDiff(<часть даты>, <начальная дата>, <конечная дата>).

Здесь «часть даты» - это закодированные специальными символами единицы измерения (часть даты) (yy – год, mm – месяц, dd - день), «начальная дата» - дата начала периода и «конечная дата» - дата конца периода. В нашем случае в качестве начальной даты берём дату рождения студента, а в качестве конечной даты берём текущую дату (функция GetDate()).

Для создания функции, выполним вышеописанный код, как и в случае с предыдущей функцией. После появления сообщения «Command(s) completed successfully.» закройте окно с кодом.

Проверим работоспособность новой табличной пользовательской функции. Создайте новый пустой запрос, затем в окне с пустым запросом наберите команду `SELECT * FROM dbo.[Функция отбора по возрасту] ()` и нажмите кнопку  на панели инструментов (Рис.6.9).

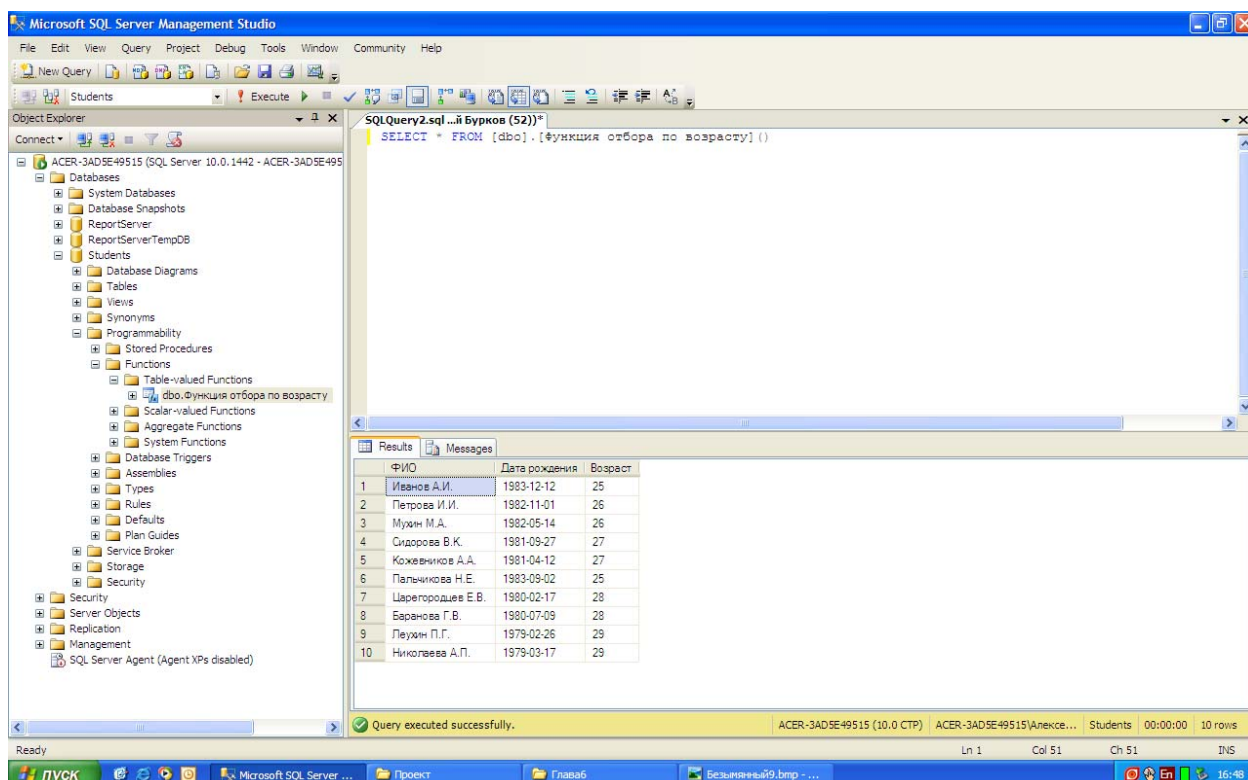


Рис.6.9

В нижней части окна появится таблица с фамилиями, датами рождения и возрастом студентов на данный момент времени (Рис.6.9).

Замечание: Обратите внимание на тот факт, что мы работаем с табличной функцией как с обыкновенной таблицей.

На этом мы заканчиваем рассмотрение пользовательских функций и переходим к рассмотрению целостности данных, диаграмм и триггеров. По окончании выполнения главы 6 обозреватель объектов будет иметь следующий вид (Рис.6.10):

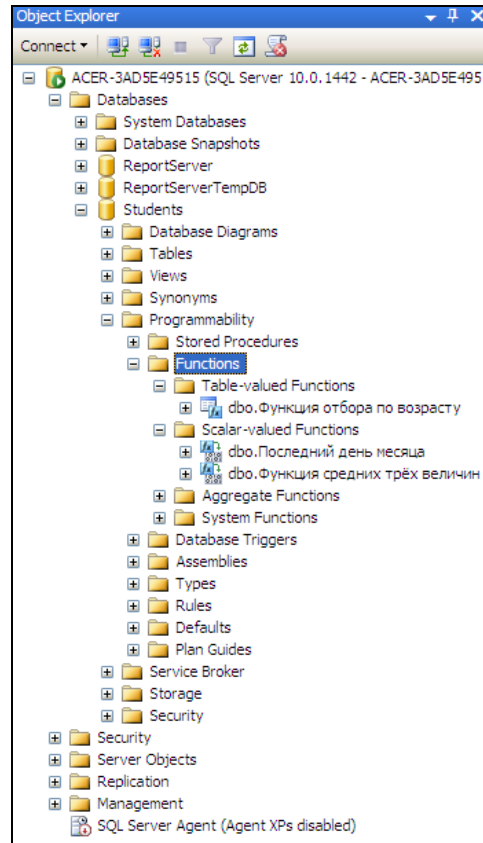


Рис.6.10

Лабораторная работа 7. Диаграммы и триггеры

Цель: научиться создавать диаграммы и триггеры

Перейдём теперь к созданию диаграмм. В БД «Microsoft SQL Server 2008» все диаграммы находятся в папке «Database Diagrams» обозревателя объектов (Рис.7.1).

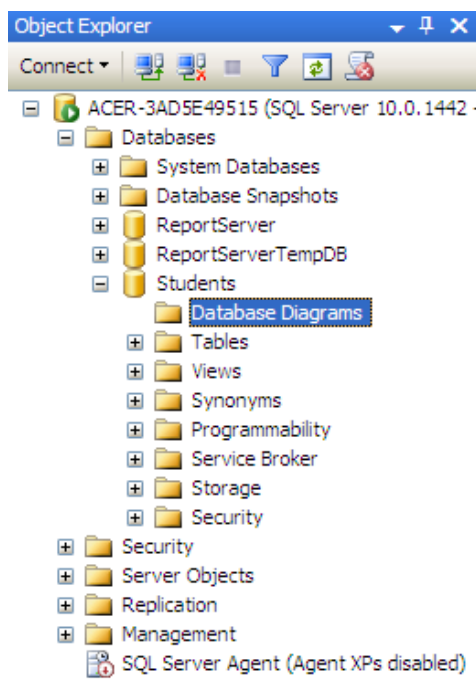


Рис.7.1

Создадим диаграмму, обеспечивающую целостность данных нашей БД «Students». Для создания новой диаграммы в БД «Students» щёлкните 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.7.2).

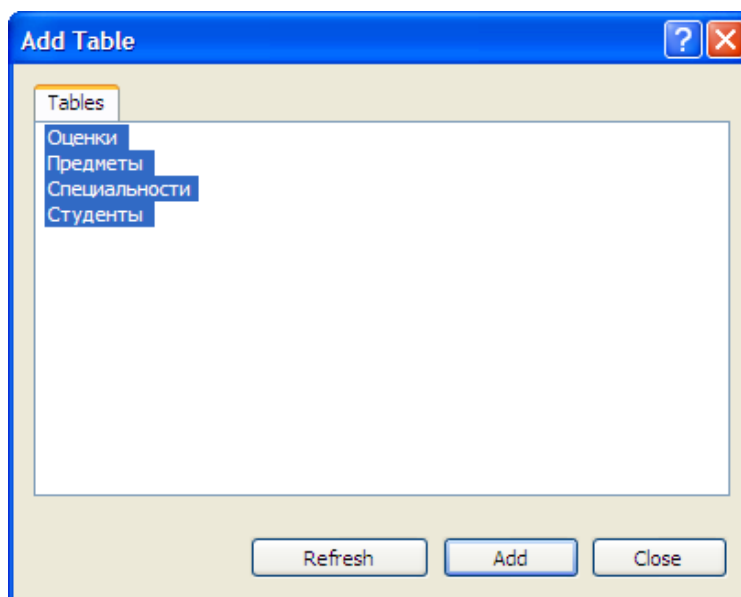


Рис.7.2

В окне добавления таблиц выделите все таблицы нашей БД и нажмите кнопку «Add» (Рис.7.2). 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.7.3).

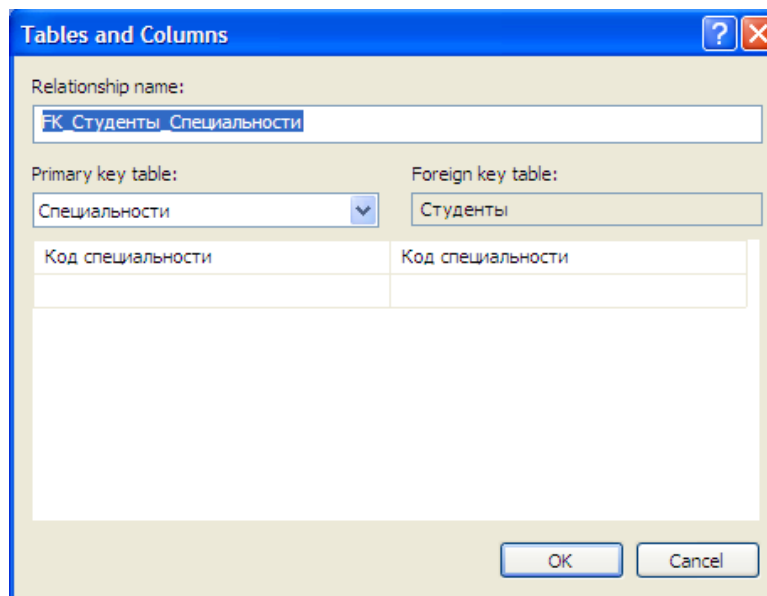


Рис.7.3

В окне создания связи нажмите кнопку «Ok». Появится окно настройки свойств связи «Foreign Key Relationship» (Рис.7.4).

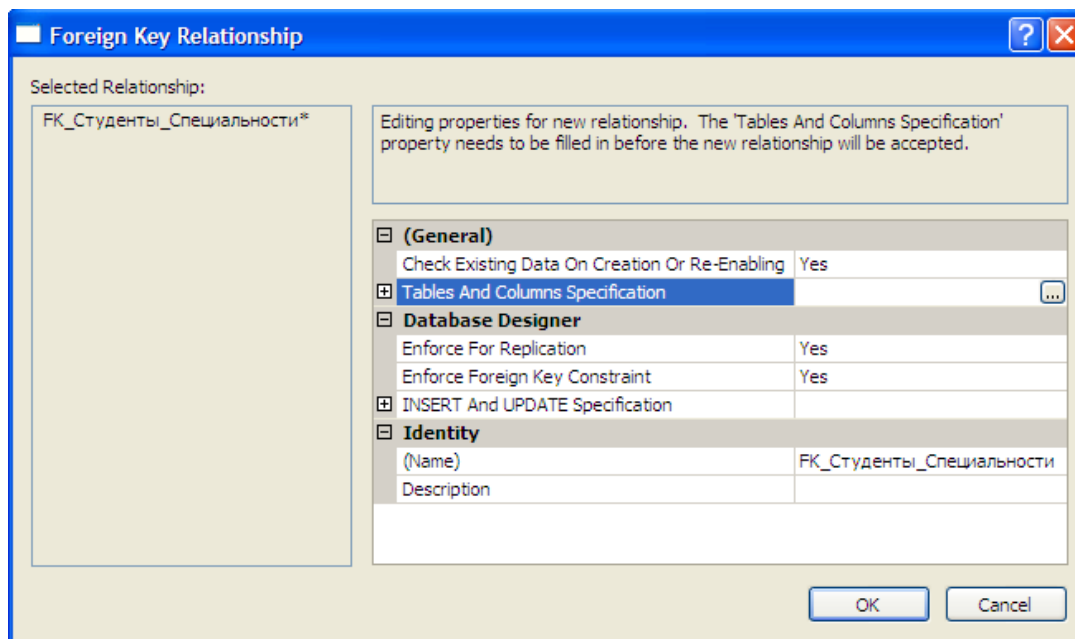


Рис.7.4

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.7.5).

Аналогичным образом создайте связь таблицы «Студенты» с таблицей «Оценки», перетащив поле «Код студента» из таблицы «Студенты» на одноимённое 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.7.5).

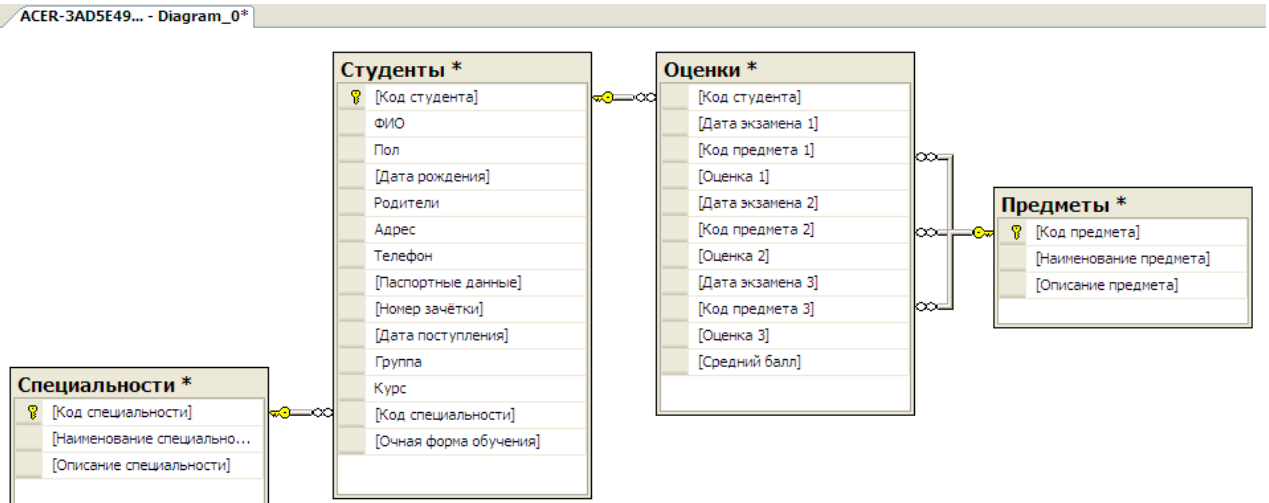



Рис.7.5

Закройте окно с диаграммой, щёлкнув 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.7.6).

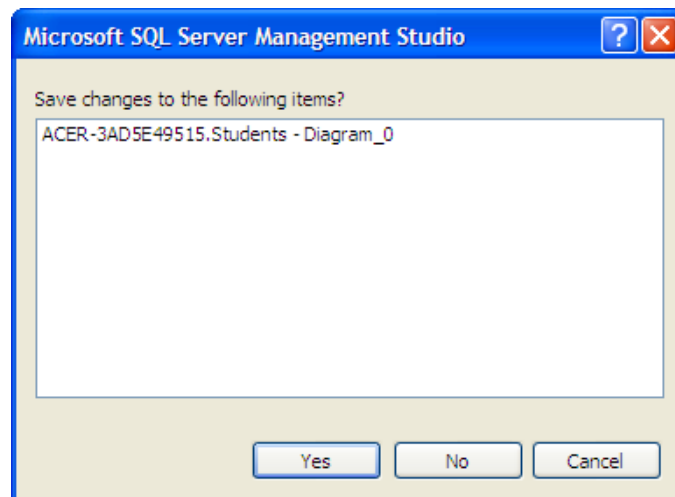


Рис.7.6

Появится окно определения имени новой диаграммы «Choose Name». В окне определения имени, задайте имя диаграммы как «Диаграмма БД Студенты» и нажмите кнопку «Ok» (Рис.7.7).

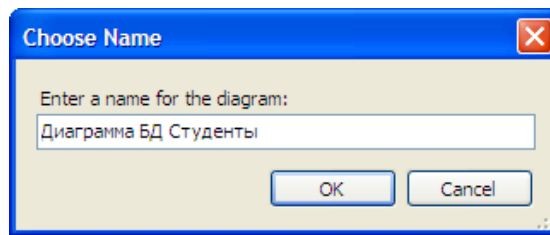


Рис.7.7

Появится окно «Save» с запросом сохранения таблиц, входящих в диаграмму. В данном окне необходимо нажать кнопку «Yes» (Рис.7.8).

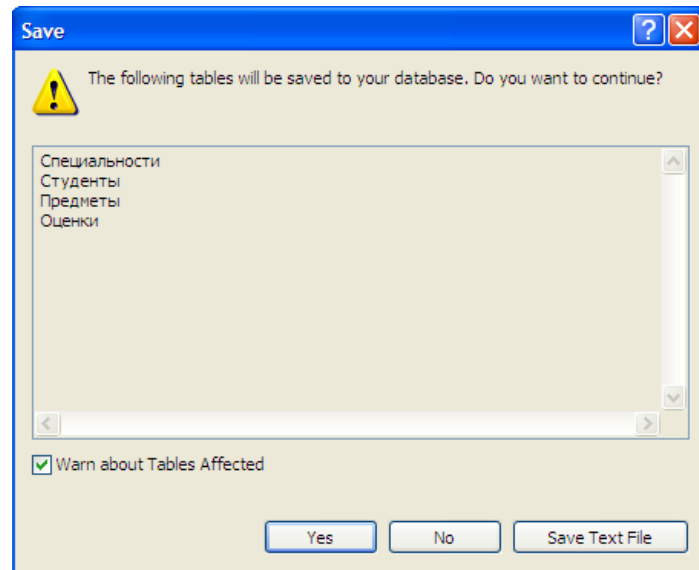


Рис.7.8

Перейдём к созданию триггеров. Создадим триггеры для таблицы «Студенты». Триггеры создаются отдельно для каждой таблицы и располагаются в обозревателе объектов в папке «Triggers». В нашем случае, папка «Triggers» входит в состав таблицы «Студенты» (Рис.7.9).

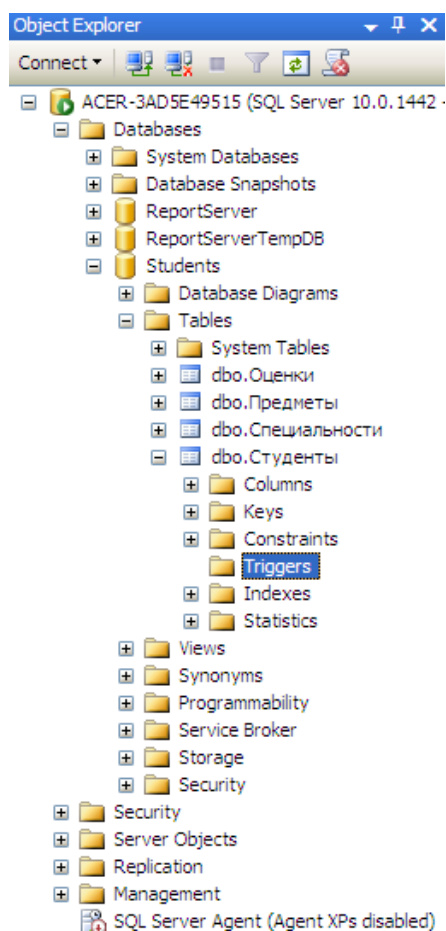


Рис.7.9

Для начала создадим триггер, выводящий сообщение «Запись добавлена» при добавлении записи в таблицу «Студенты». Создадим новый триггер, щёлкнув ПКМ по папке «Triggers» в таблице «Студенты» и в появившемся меню выбрав пункт «New Trigger». Появится следующее окно с новым триггером (Рис.7.10):

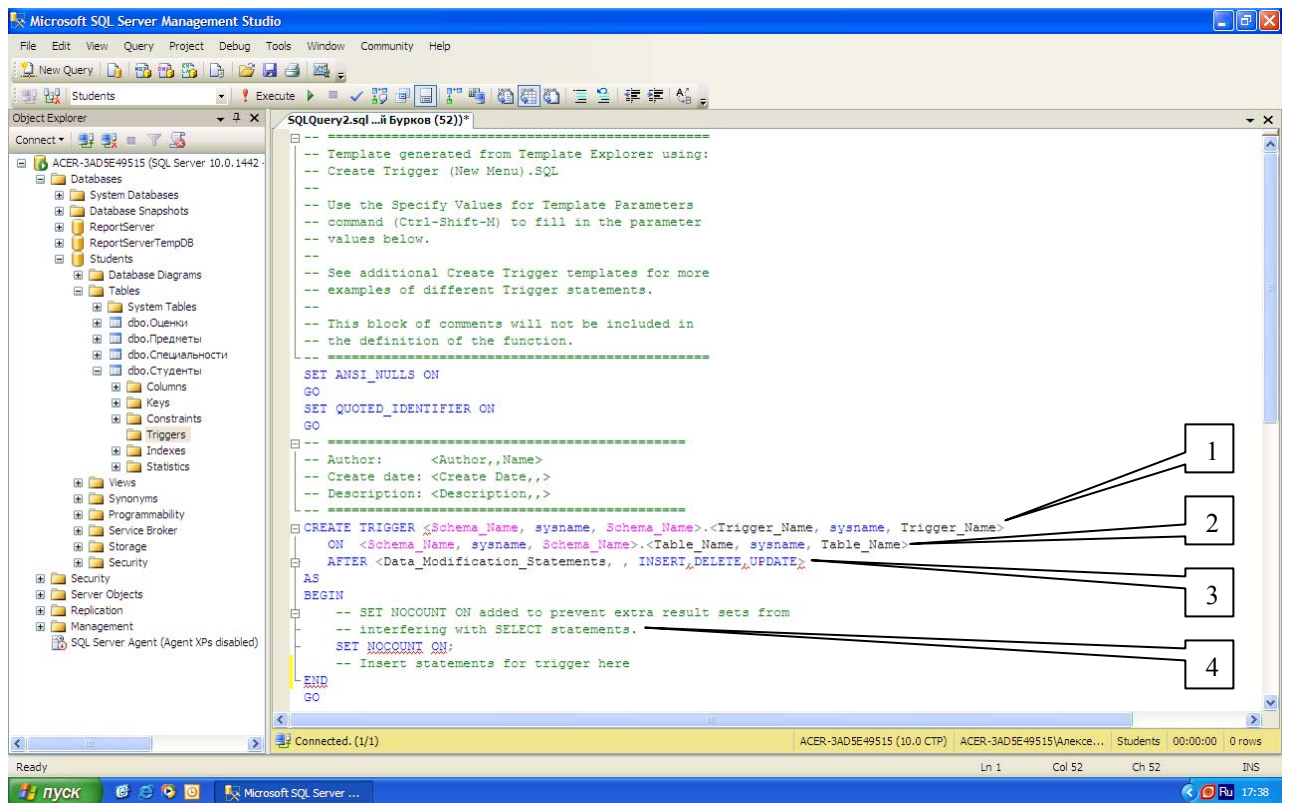


Рис.7.10

Рассмотрим структуру триггеров:

- 1) Область определения имени функции (Trigger_Name);
- 2) Область, показывающая для какой таблицы создаётся триггер (Table_Name);
- 3) Область, показывающая когда выполнять триггер (INSERT – при создании записи в таблице, DELETE – при удалении и UPDATE – при изменении) и как его выполнять (ALTER – после выполнения операции, INSTEAD OF – вместо выполнения операции);
- 4) Тело триггера, содержит команды языка программирования запросов T-SQL.

В окне нового триггера наберите код как показано на рисунке 7.11.

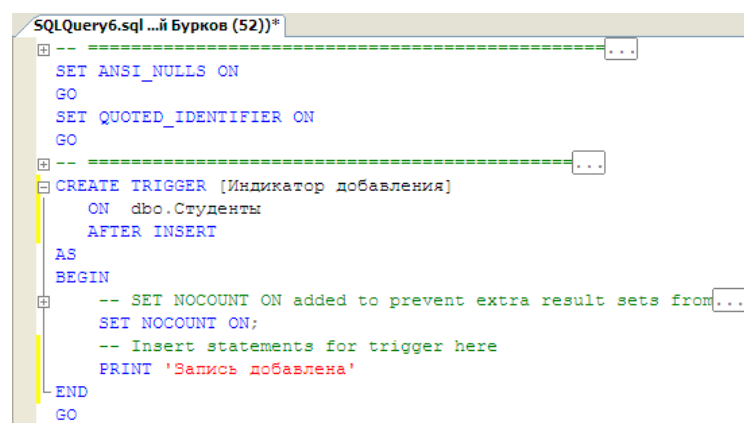
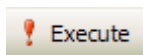


Рис.7.11

Из рисунка 7.11 видно, что создаваемый триггер «Индикатор добавления» выполняется после добавления записи (AFTER INSERT) в таблицу «Студенты» (ON dbo.Студенты). После добавления записи триггер выведет на экран сообщение «Запись добавлена» (PRINT 'Запись добавлена'). Выполните набранный код, нажав кнопку



на панели инструментов. В нижней части окна с кодом появится сообщение «Command(s) completed successfully.».

Проверим, как работает новый триггер. Создайте новый пустой запрос и в нём наберите следующую команду для добавления новой записи в таблицу «Студенты» (Рис.7.12):

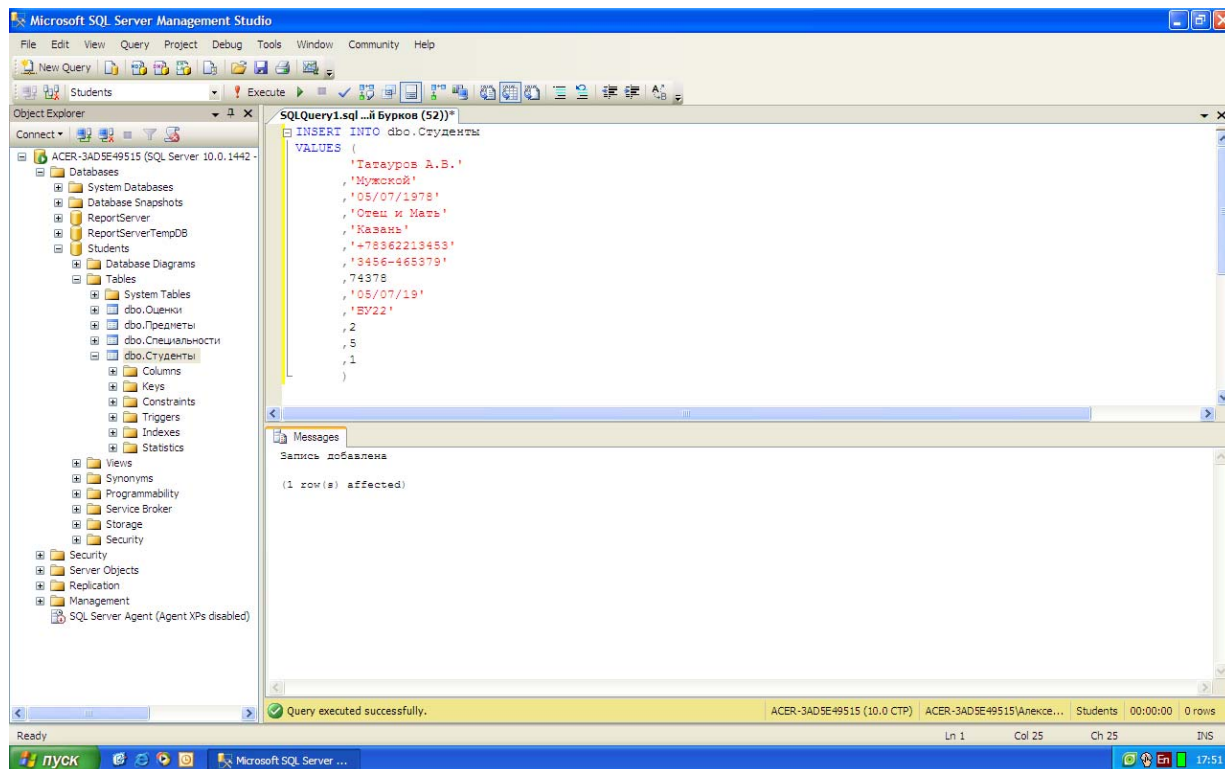
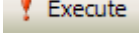


Рис.7.12

Выполните набранную команду, нажав кнопку  на панели инструментов. В таблицу будет добавлена новая запись, и триггер выведет сообщение «Запись добавлена» (Рис.7.12).

Теперь создадим триггер отображающий сообщение «Запись изменена». Создайте новый триггер, как в предыдущем случае. В окне нового триггера наберите следующий код (Рис.7.13):

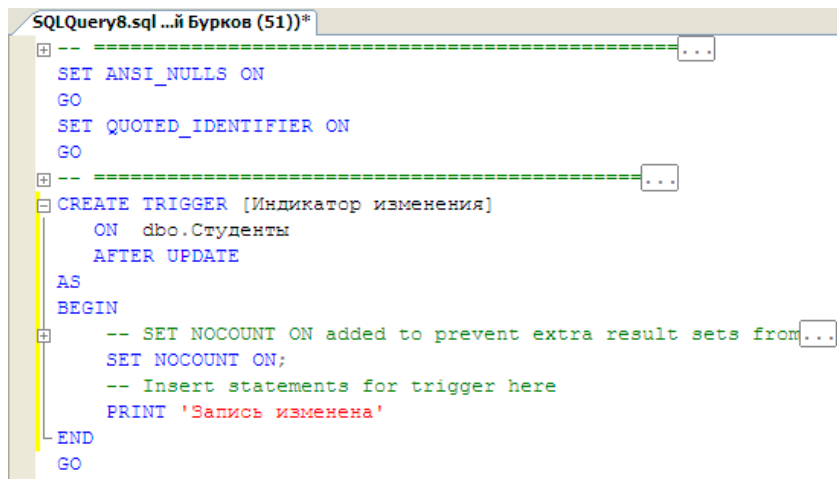


Рис.7.13

Из рисунка 7.13 видно, что новый триггер «Индикатор добавления» выполняется после изменения записи (AFTER UPDATE) в таблице «Студенты» (ON dbo.Студенты). После изменения записи триггер выведет на экран сообщение «Запись изменена» (PRINT 'Запись изменена'). Выполните набранный код. В нижней части окна с кодом появится сообщение «Command(s) completed successfully.».

Проверим работоспособность созданного триггера. Создайте новый запрос и в нём наберите команду, представленную на рисунке 7.14.

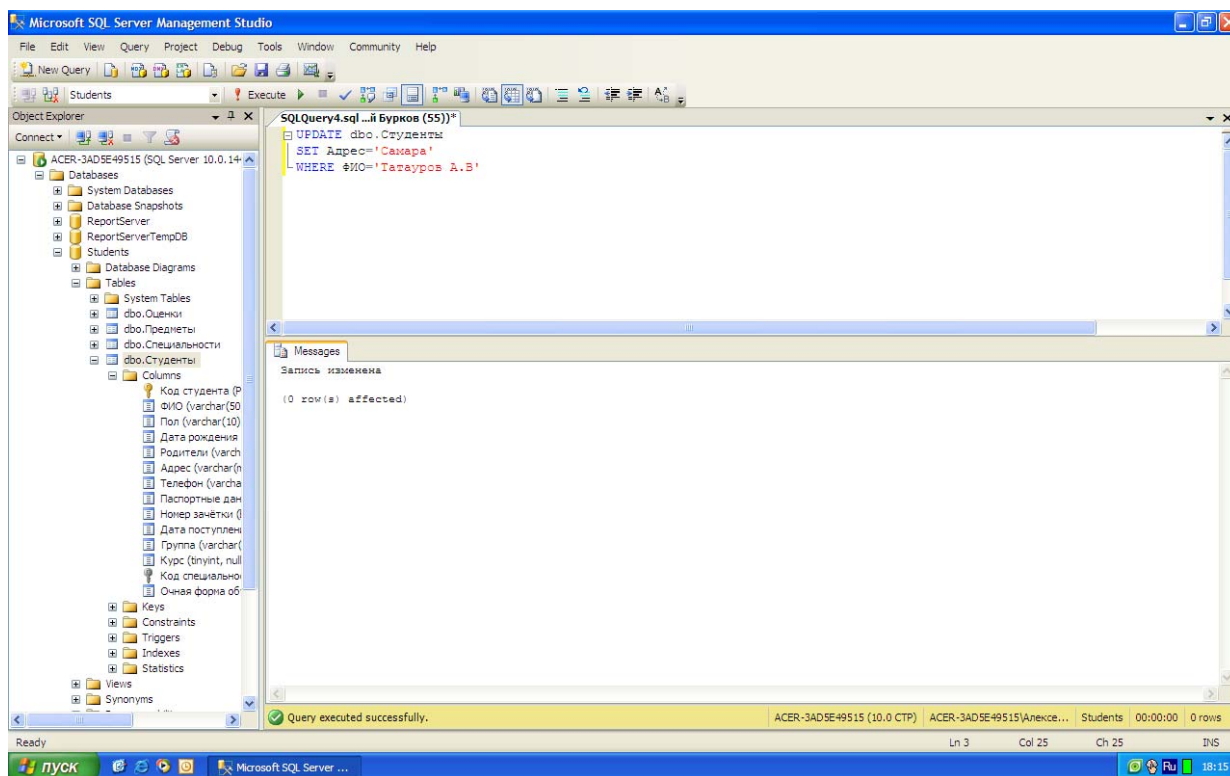
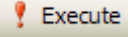


Рис.7.14

Выполните набранную команду, нажав кнопку  на панели инструментов. В таблицу будет добавлена новая запись, и триггер выведет сообщение «Запись добавлена» (Рис.7.14).

Для полноты картины создадим триггер, выводящий сообщение при удалении записи из таблицы «Студенты». Создайте новый триггер и в нём наберите код, показанный на рисунке 7.15.

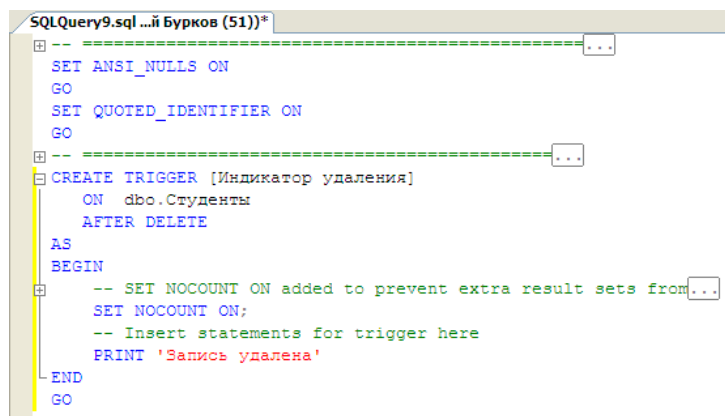


Рис.7.15

Создаваемый триггер «Индикатор удаления» выполняется после удаления записи (ALTER DELETE) из таблицы студенты (ON dbo.Студенты). После удаления записи триггер выводит сообщение «Запись удалена» (PRINT 'Запись удалена').

Выполните код, представленный на рисунке 7.15. В нижней части окна с кодом появиться сообщение «Command(s) completed successfully.».

Проверим работу триггера «Индикатор удаления» удалив созданную ранее запись из таблицы «Студенты». Для этого создайте новый запрос и в нём наберите следующую команду (Рис.7.16):

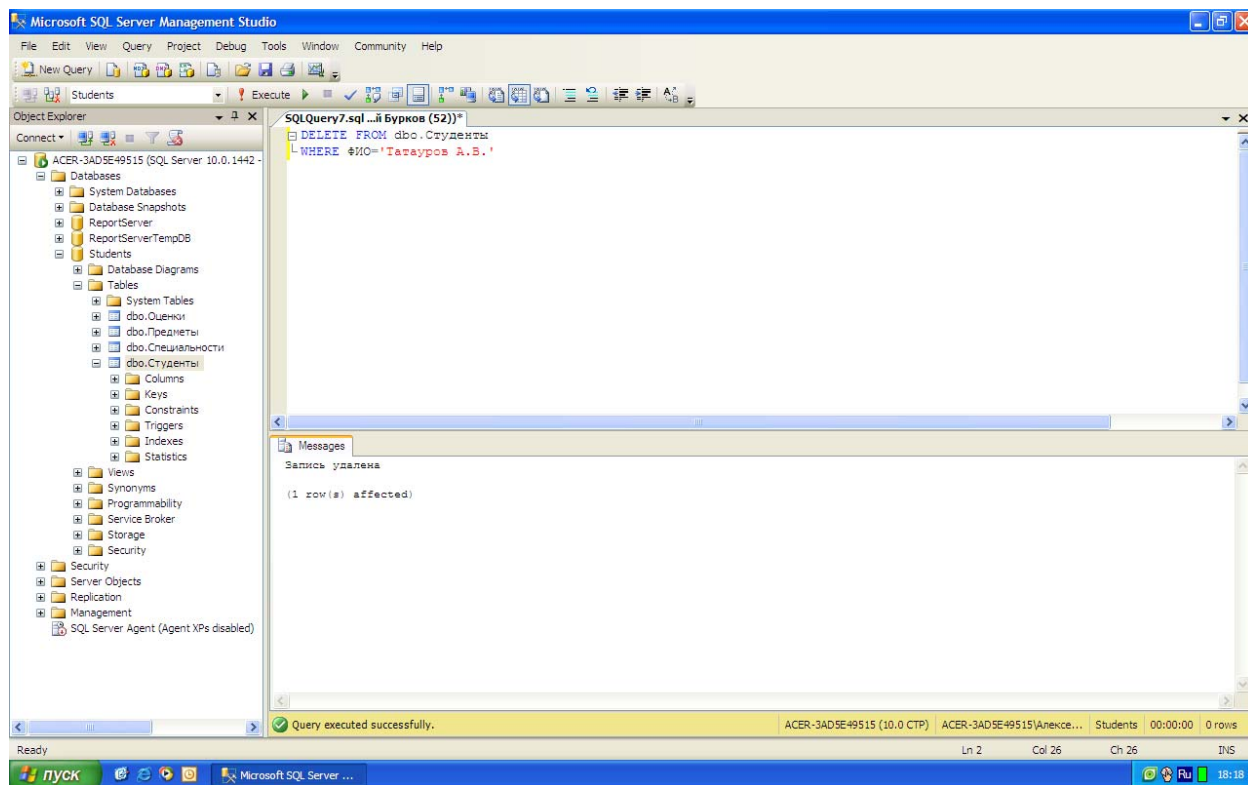
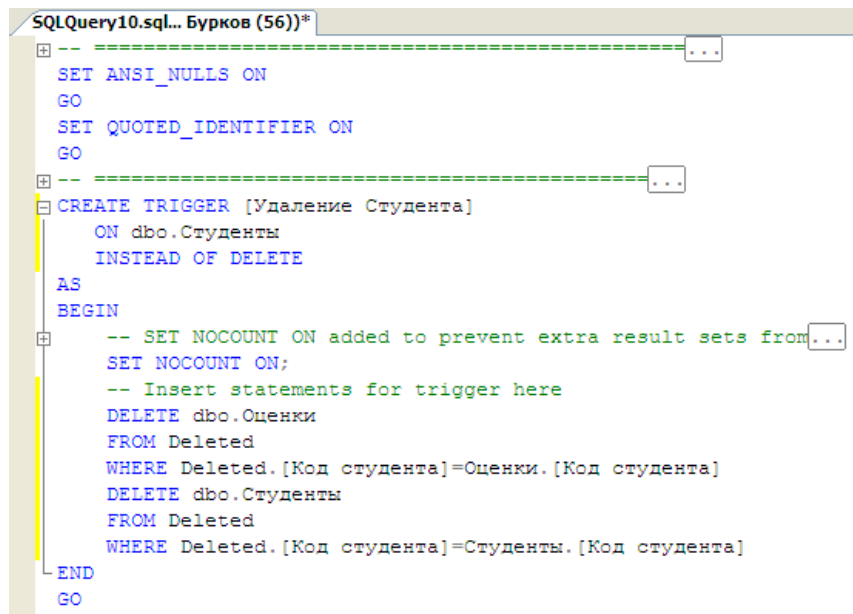


Рис.7.16

Выполните вышеприведённую команду. После удаления записи триггер «Индикатор удаления» отобразит сообщение «Запись удалена» (Рис.7.16).

В заключение рассмотрим пример применения триггеров для обеспечения целостности данных. Создадим триггер «Удаление студента», который при удалении записи из таблицы студенты сначала удаляет все связанные с ней записи из таблицы «Оценки», а затем удаляет саму запись из таблицы «Студенты», тем самым обеспечивается целостность данных.

Создайте новый триггер и в нём наберите следующий код (Рис.7.17):

The image shows a screenshot of a SQL Server Enterprise Manager window. The title bar reads "SQLQuery10.sql... Бурков (56)*". The window contains a SQL script for creating a trigger. The script starts with "SET ANSI_NULLS ON" and "SET QUOTED_IDENTIFIER ON", followed by "GO". Then, it defines a trigger named "[Удаление Студента]" on the "dbo.Студенты" table, using "INSTEAD OF DELETE". The trigger body begins with "BEGIN", followed by a comment "-- SET NOCOUNT ON added to prevent extra result sets from". Then, it sets "NOCOUNT ON;" and includes a comment "-- Insert statements for trigger here". The main logic consists of two "DELETE" statements: the first deletes from "dbo.Оценки" where "Deleted.[Код студента] = Оценки.[Код студента]", and the second deletes from "dbo.Студенты" where "Deleted.[Код студента] = Студенты.[Код студента]". The script ends with "END" and "GO".

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
CREATE TRIGGER [Удаление Студента]
ON dbo.Студенты
INSTEAD OF DELETE
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    SET NOCOUNT ON;
    -- Insert statements for trigger here
    DELETE dbo.Оценки
    FROM Deleted
    WHERE Deleted.[Код студента]=Оценки.[Код студента]
    DELETE dbo.Студенты
    FROM Deleted
    WHERE Deleted.[Код студента]=Студенты.[Код студента]
END
GO
```

Рис.7.17

Создаваемый триггер «Удаление студента» выполняется вместо удаления записи (INSTEAD OF DELETE) из таблицы «Студенты» (ON dbo.Студенты).

Замечание: При срабатывании триггера вместо удаления записи создаётся временная константа Deleted, содержащая имя таблицы из которой должно было быть произведено удаление.

После срабатывания триггера из таблицы «Оценки» удаляется запись, у которой значение поля «Код студента» равно значению такого же поля у удаляемой записи из таблицы «Студенты». Эту операцию выполняют следующие команды:

```
DELETE dbo.Оценки
FROM Deleted
WHERE Deleted.[Код студента] = Оценки.[Код студента]
```

Затем удаляется запись из таблицы «Студенты», которую удаляли до срабатывания триггера. Удаление выполняется следующими командами:

```
DELETE dbo.Студенты
FROM Deleted
WHERE Deleted.[Код студента] = Студенты.[Код студента]
```

Выполните код, представленный на рисунке 7.17. В нижней части окна с кодом появиться сообщение «Command(s) completed successfully.».

Проверим, как работает триггер «Удаление студента». Для этого создайте новый запрос и в нём наберите следующий код (Рис.7.18):

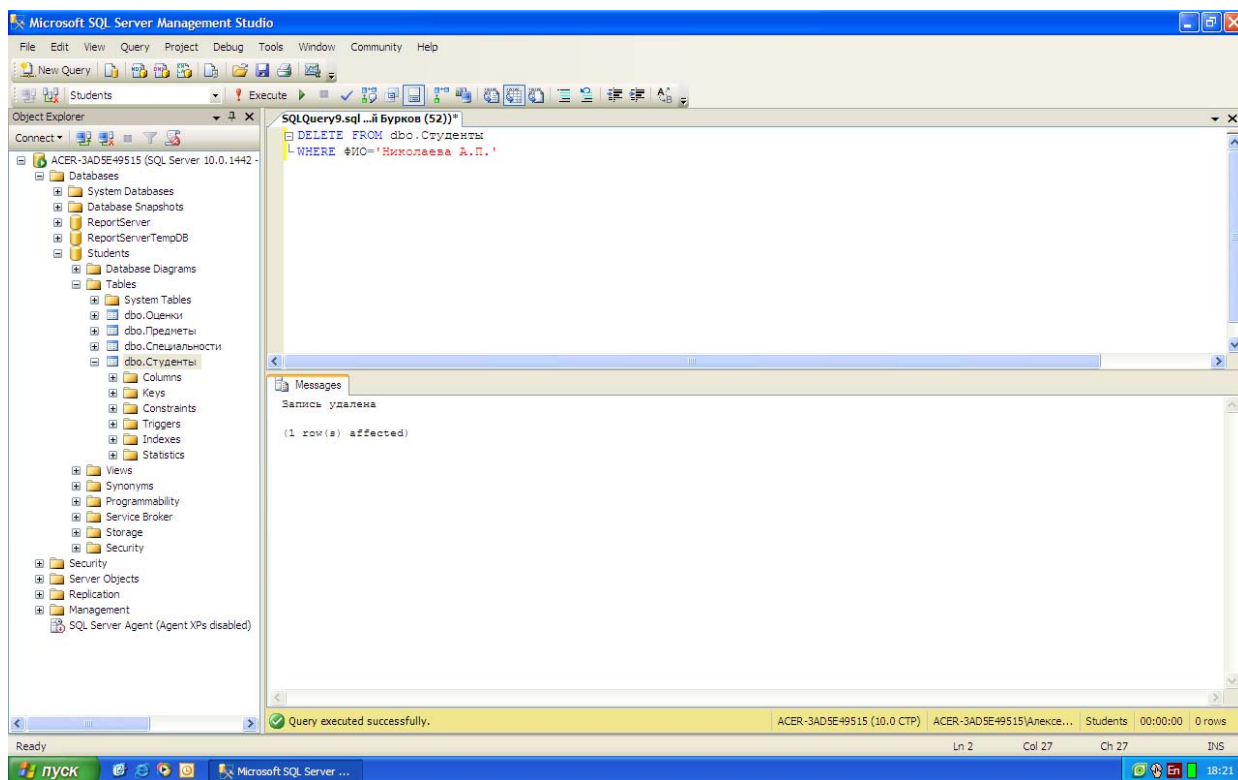


Рис.7.18

При срабатывании триггера сначала из таблицы «Оценки» удалятся все связанные с удаляемой записью записи, а затем удаляется сама удаляемая запись из таблицы «Студенты», при этом сохраняется целостность данных.

Замечание: Хотелось бы заметить, что без использования триггера «Удаление студента» нам бы не удалось удалить запись из таблицы «Студенты». Команда удаления была бы заблокирована диаграммой «Диаграмма БД Студенты» во избежание нарушения целостности данных.

На этом мы завершаем работу с диаграммами и триггерами. После выполнения всех вышеописанных действий обозреватель объектов будет иметь следующий вид (Рис.7.19):

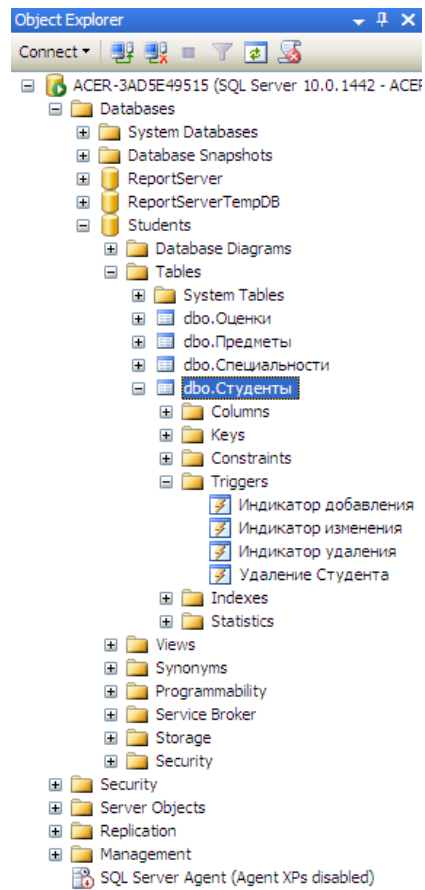


Рис.7.19

Лабораторная работа 8. Создание проекта. Подключение файла данных к проекту

Цели:

- 1. Научиться создавать проекты.***
- 2. Научиться подключать файлы данных к проекту***

Теперь рассмотрим создание пользовательского интерфейса нашей БД в «Microsoft Visual Studio 2008».

Прежде чем создавать новый проект в «Microsoft Visual Studio 2008», необходимо запустить «SQL Server Browser» (Обозреватель SQL серверов) сервис, обеспечивающий доступ сторонним приложениям к «Microsoft SQL Server 2008». Для запуска данного сервиса необходимо запустить утилиту «SQL Server Configuration Manager» (Менеджер настроек SQL сервера), входящую в состав «Microsoft SQL Server 2008». Для запуска данной утилиты в меню «Пуск» выберите пункт «Программы/Microsoft SQL Server 2008/Configuration Tools/ SQL Server Configuration Manager» как это представлено на рисунке 8.1.

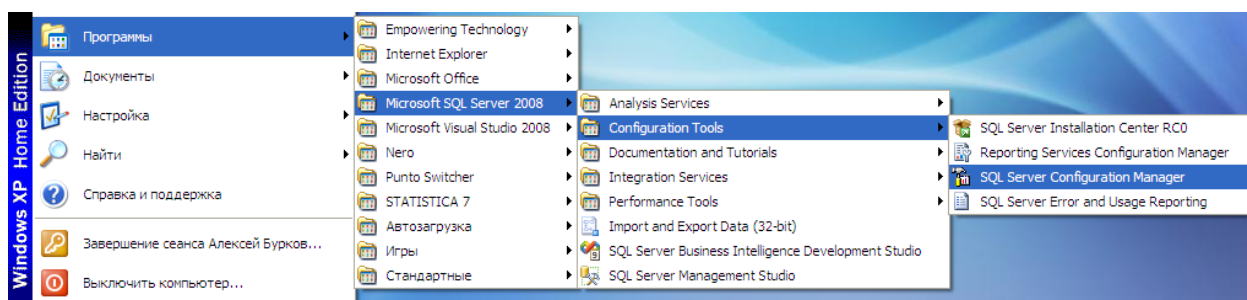


Рис.8.1

Появится окно менеджера настроек SQL сервера имеющее следующий вид (Рис.8.2):

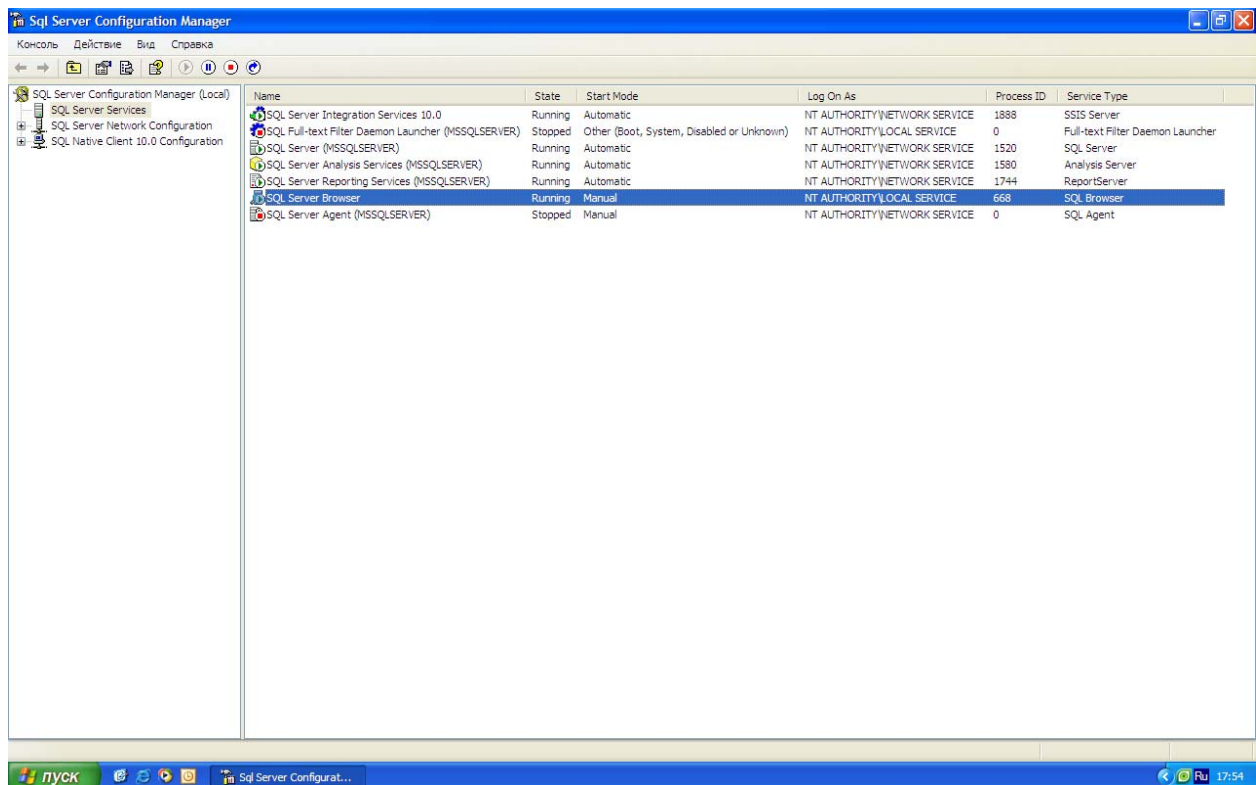



Рис.8.2

В правой части окна менеджера настроек SQL сервера, в списке сервисов «Microsoft SQL Server 2008» выделите сервис «SQL Server Browser», затем на панели инструментов нажмите кнопку «Start Service» (Запустить сервис)  (Рис.8.2). После запуска вышеуказанного сервиса окно менеджера настроек SQL сервера можно закрыть.

Теперь перейдём непосредственно к созданию пользовательского интерфейса нашей БД «Students» в «Microsoft Visual Studio 2008». Для начала необходимо создать новый проект. Для этого запустите «Microsoft Visual Studio 2008», выбрав в меню «Пуск» пункт «Программы/ Microsoft Visual Studio 2008/ Microsoft Visual Studio 2008» (Рис.8.3).

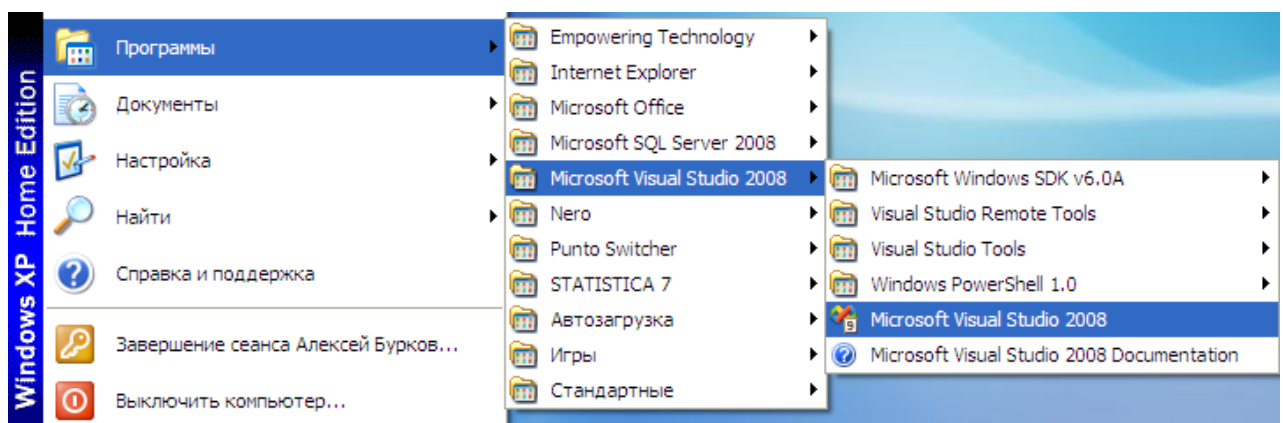


Рис.8.3

Появится окно со стартовой страницей «Microsoft Visual Studio 2008 (Start Page)» (Рис.8.4).

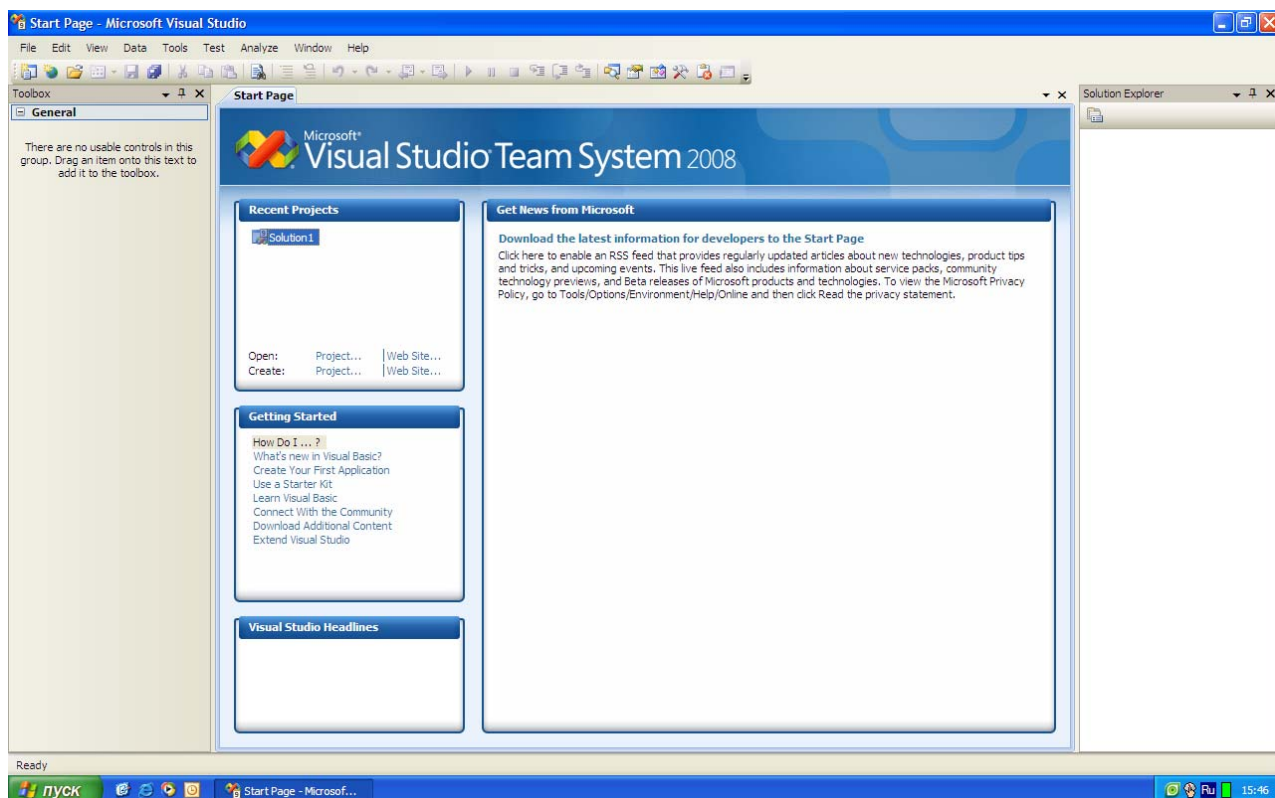


Рис.8.4

Для создания нового проекта на стартовой странице в области «Recent Projects» (Недавние проекты) необходимо щёлкнуть ЛКМ по ссылке «Create: Project...» (Создать: проект...) (Рис.8.4). Появится окно выбора типа создаваемого проекта, и используемого языка программирования «New Project» (Новый проект) (Рис.8.5).

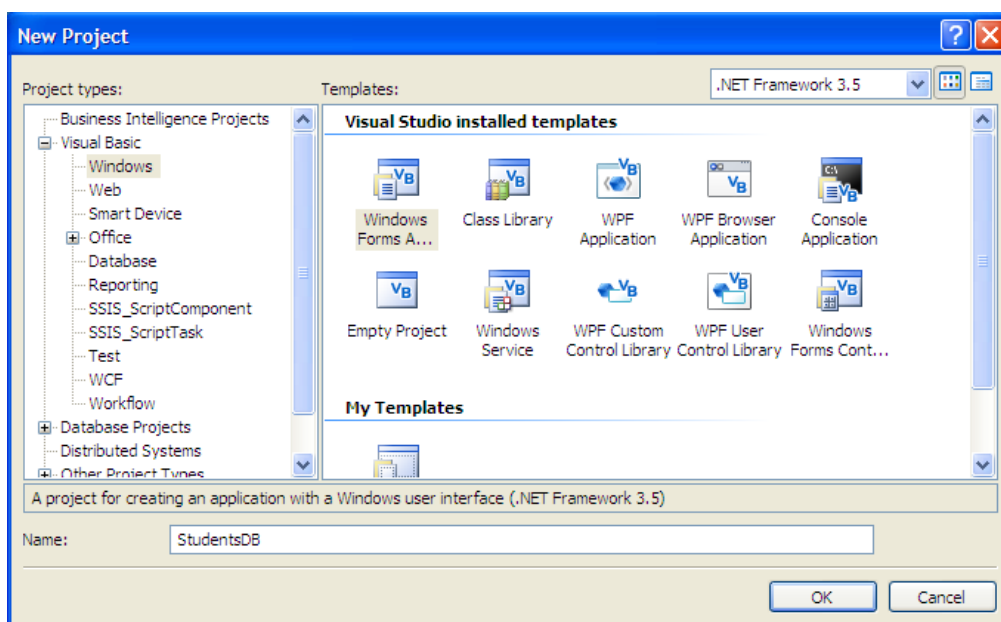


Рис.8.5

В нашем случае на дереве типов проекта «Project types:» (Типы проектов) выберите «Visual Basic\Windows», а в качестве шаблона проекта (Область Templates:) выберите «Windows Forms Application» (Приложение Windows). В качестве имени проекта (Поле ввода Name:) задайте «StudentsDB» и нажмите кнопку «Ok» (Рис.8.5).

Появится стандартное окно разработки языка «Microsoft Visual Basic 2008» (Рис.8.6).

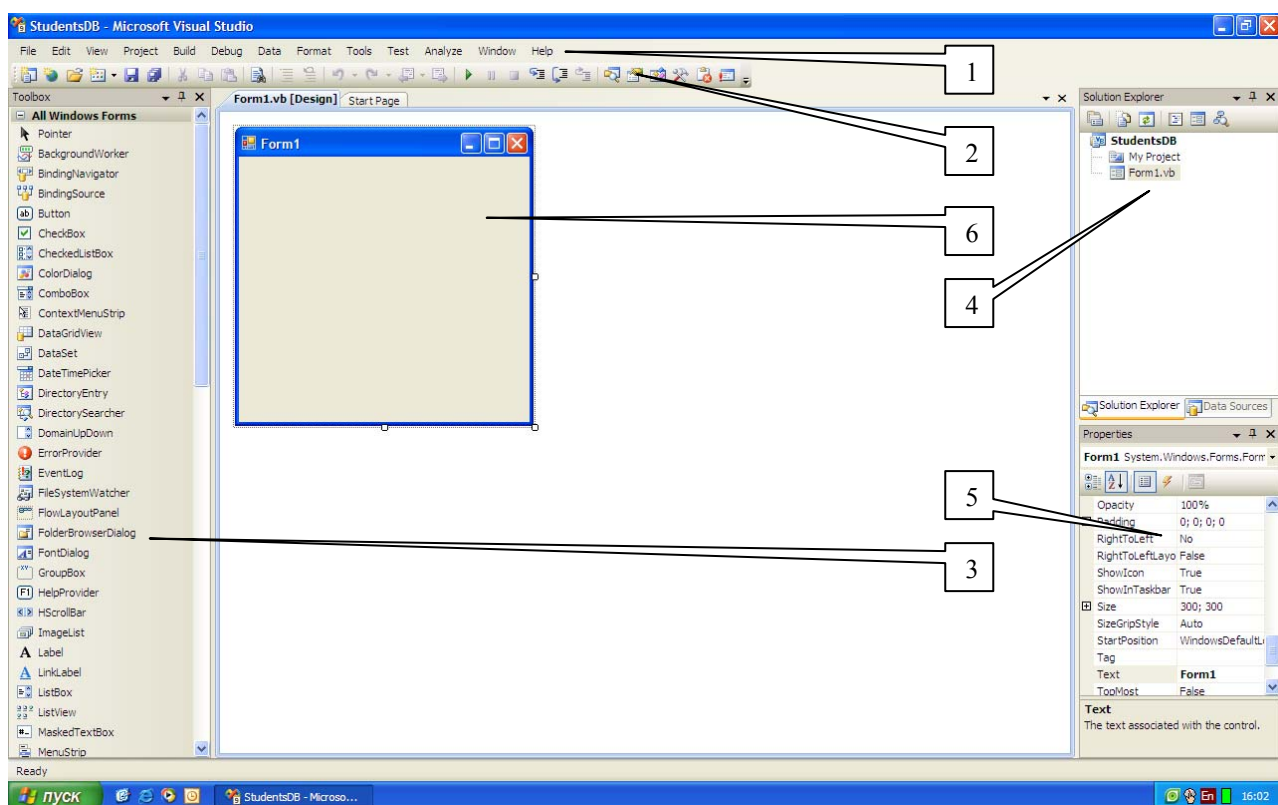


Рис.8.6

Окно разработки языка «Microsoft Visual Basic 2008» имеет следующую структуру (рис.8.6):

1. Оконное меню – содержит полный набор команд для управления средой разработки;
2. Панель инструментов – содержит кнопки с наиболее часто используемыми командами среды разработки;
3. Панель объектов (Toolbox) – содержит кнопки классов для создания различных объектов (Элементов управления);
4. Обзорщик проекта/Источники данных (Solution Explorer/Data Sources)- в зависимости от активизированной в нижней части данной панели вкладки, отображает обзорщик проекта или источники данных, подключённые к проекту. Обзорщик проекта (Solution Explorer) отображает все файлы, входящие в проект и позволяет переключаться между ними. Источники данных это базы данных, службы или объекты из которых поступают данные в проект;
5. Панель свойств (Properties) – отображает и позволяет изменять свойства выбранного объекта;
6. Рабочая область – в зависимости от выбранной вкладки, расположенной в верхней части области, отображает область дизайна формы, код формы или стартовую страницу.

После создания нового проекта необходимо подключить к проекту созданную ранее в «Microsoft SQL Server 2008» БД «Students». Для подключения БД к проекту в оконном меню среды разработки выберите пункт «Data\Add New Data Source...». Появится окно мастера подключения к новому источнику данных «Data Source Configuration Wizard» (Рис.8.7).

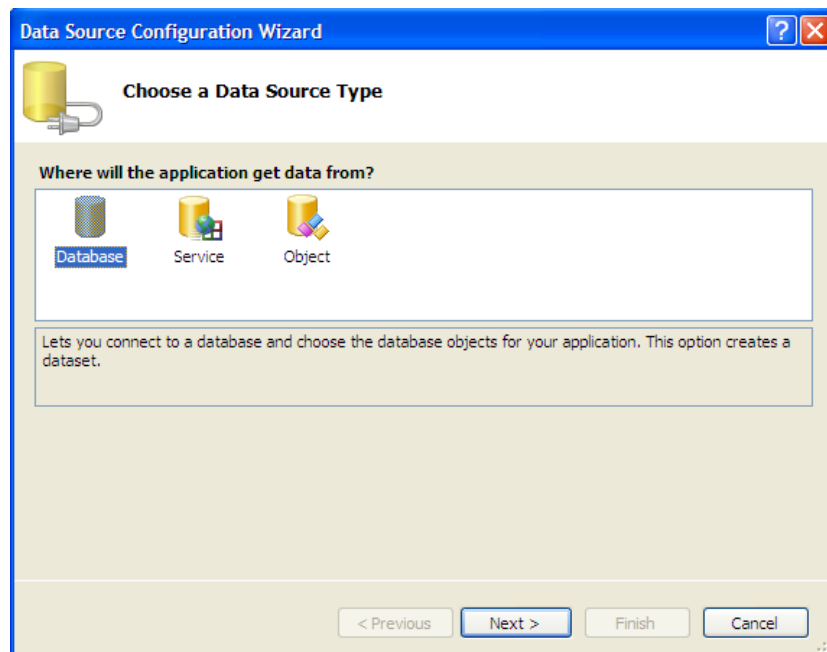


Рис.8.7

В данном окне можно выбрать один из трёх видов источников данных (Choose a Data Source Type):

- БД (Database);
- Служба (Service);
- Объект (Object).

Так как мы подключаем наш проект к БД «Students» то выбираем вариант БД (Database) и нажимаем кнопку «Next» (Далее). Появится окно выбора подключения к БД (Choose Your Data Connection) (Рис.8.8).

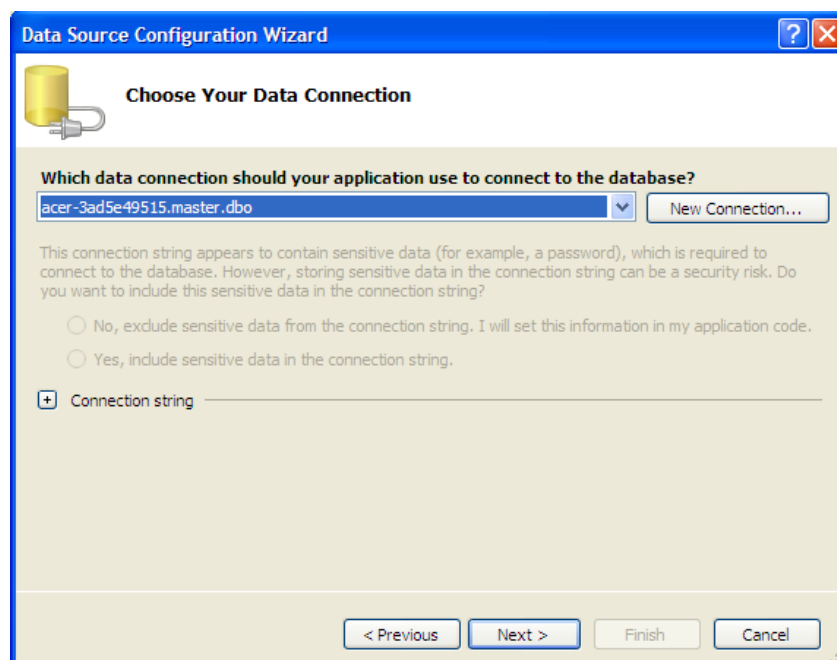


Рис.8.8

В окне выбора подключения к БД, для создания нового подключения нажмите кнопку «New Connection...». Появится окно добавления нового соединения «Add Connection» (Рис.8.9).

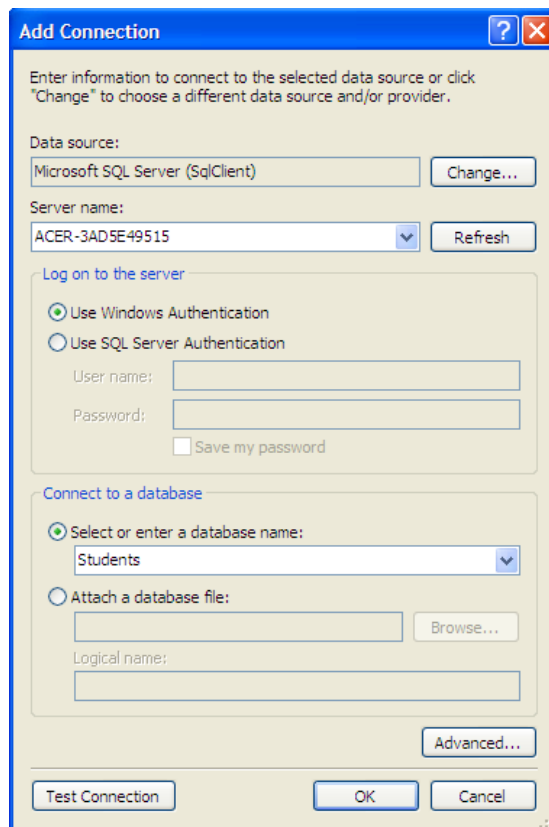


Рис.8.9

В окне «Add Connection» в выпадающем списке «Server Name» (Имя сервера) выберите имя сервера заданное при установке SQL сервера. В качестве логина и пароля для входа на сервер (Log on to the server) выберите «Use Windows Authentication» (Использовать логин и пароль учётной записи Windows). В качестве БД для подключения (Connect to a database) из выпадающего списка «Select or enter a database name:» (Выберите или введите имя БД) выберите БД «Students» (Рис.8.9).

Для проверки работоспособности создаваемого соединения нажмите кнопку «Test Connection». Появится сообщение «Test connection succeeded» (Рис.8.10), говорящее о том, что соединение работоспособно.

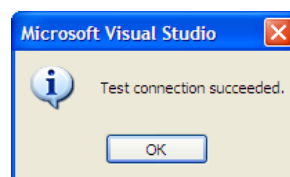


Рис.8.10

Закройте окно, представленное на рисунке 8.10, а затем в окне добавления нового соединения «Add Connection» (Рис.8.9) нажмите кнопку «Ok». Произойдёт возвращение к окну выбора подключения к БД (Choose Your Data Connection). Просмотрите созданную строку подключения «Connection string», щёлкнув по знаку «+» в нижней части окна (Рис.8.11).

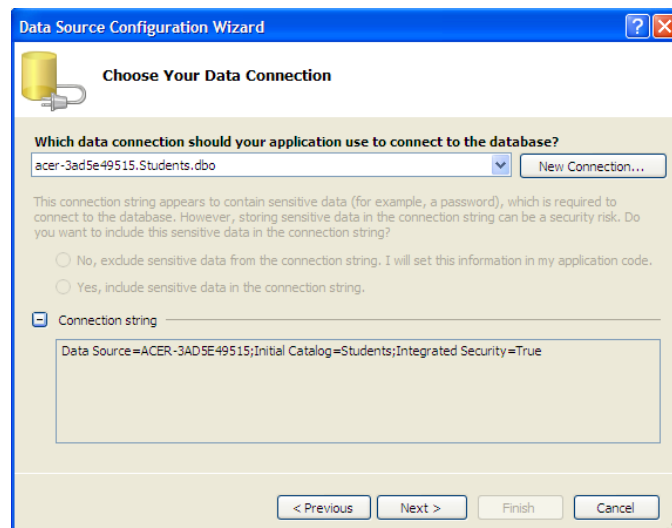


Рис.8.11

В окне выбора подключения к БД (Choose Your Data Connection) нажмите кнопку «Next» (Далее). Появится окно с запросом о сохранении строки подключения «Save the Connection String to the Application Configuration File» (Сохранить строку подключения в файл настроек приложения) (Рис.8.12).

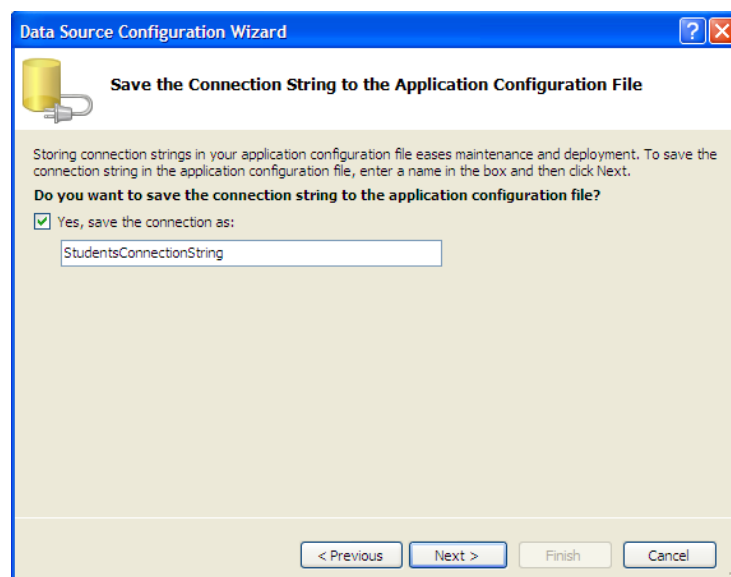


Рис.8.12

Для сохранения строки подключения включите опцию «Yes, save the connection as:» (Да, сохранить подключение как:) и нажмите кнопку «Next».

Появится окно выбора объектов подключаемой БД (Choose Your Database Objects) (Рис.8.13).

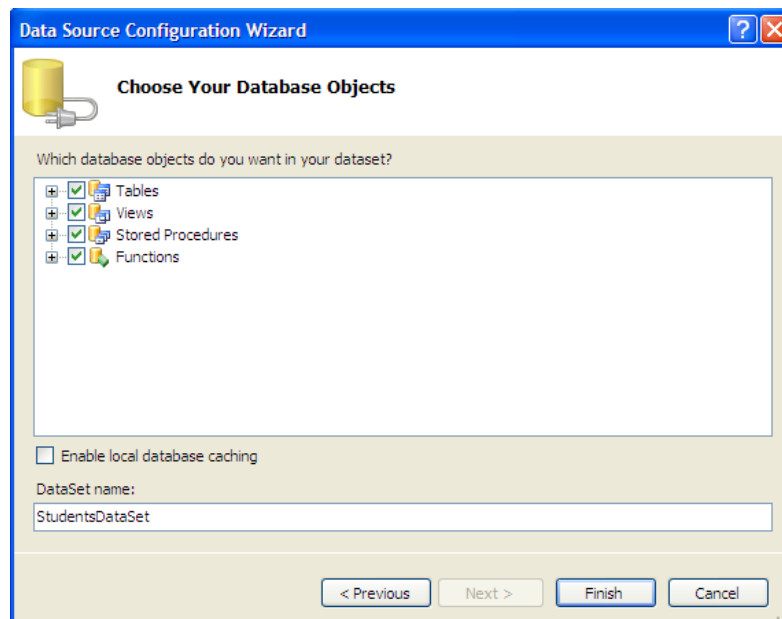


Рис.8.13

Выберите все объекты как это показано на рисунке 8.13 и нажмите кнопку «Finish» (Готово). Подключение завершено.

Для просмотра источника данных щёлкните по вкладке «Data Sources» в окне разработки языка «Microsoft Visual Basic 2008» (Рис.8.6 пункт 4). Вкладка «Data Sources» имеет вид (Рис.8.14):

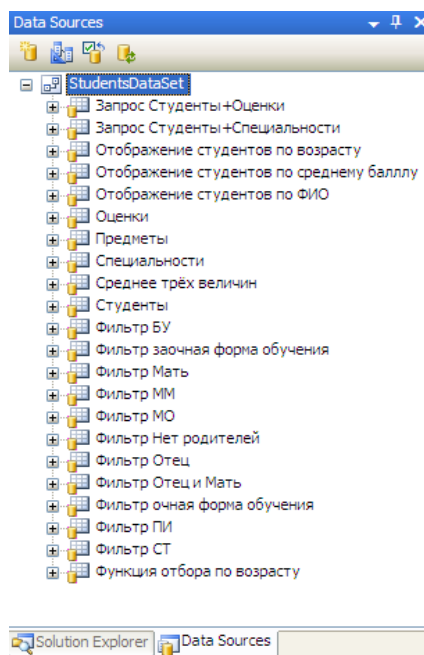


Рис.8.14

На этом мы заканчиваем рассмотрение подключения БД к новому проекту. Закройте окно среды разработки. Появится окно сохранения нового проекта «Save Project» (Рис.8.15).

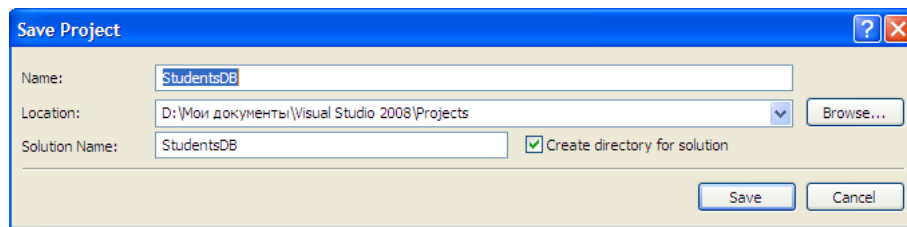


Рис.8.15

В данном окне в поле ввода «Name:» (Имя) задайте имя проекта «StudentsDB». В поле «Location:» (Расположение:) задайте папку, где будет сохранён новый проект, нажав кнопку «Browse...» (Выбрать). После определения имени проекта и его местоположения нажмите кнопку «Save» (Сохранить).

Лабораторная работа 9. Главная кнопочная форма. Создание простых ленточных форм для работы с данными

Цель: научиться создавать пользовательский интерфейс (главная кнопочная форма, простые ленточные формы для работы с данными)

Перейдём теперь к созданию пользовательского интерфейса. Его создание начнём с создания главной кнопочной формы. Запустите «Microsoft Visual Studio 2008» и откройте созданный ранее проект «StudentsDB», щёлкнув по его значку в области «Recent Projects» стартовой страницы «Start Page» (Рис.9.1).

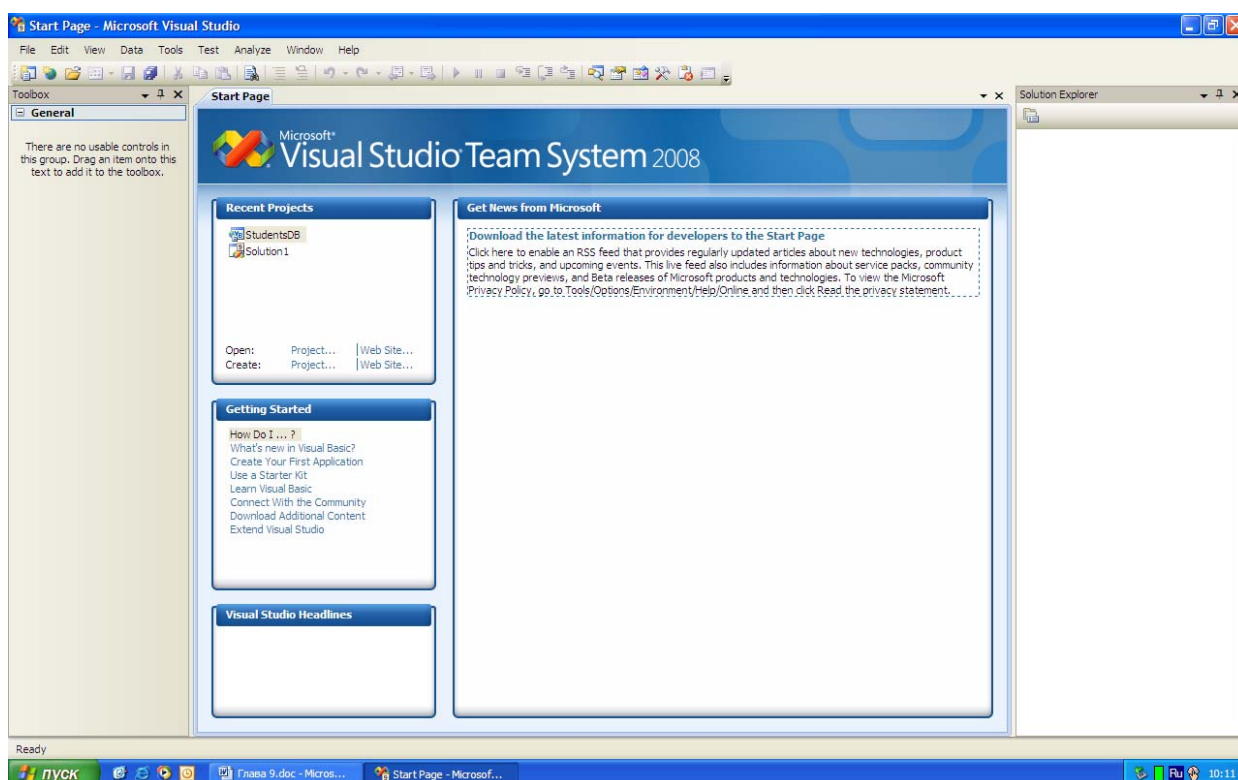


Рис.9.1

После появления стандартного окна среды разработки в рабочей области на форму поместите надпись (Label) и четыре кнопки (Button) как показано на рисунке 9.2.

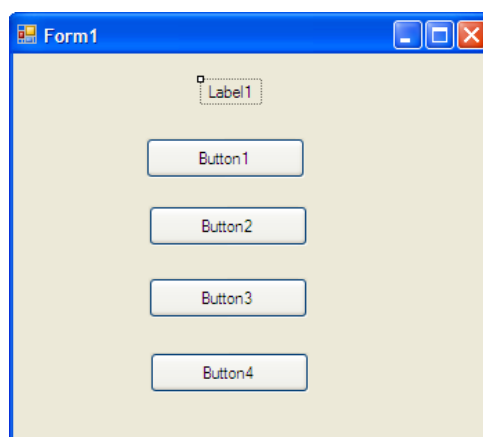




Рис.9.2

Замечание: Для создания надписи на панели объектов необходимо нажать кнопку,

 Label

а затем нарисовать прямоугольник мышью на форме, удерживая ЛКМ. Кнопки создаются таким же образом, только на панели объектов нажмите кнопку  Button.

После создания объектов перейдём к настройке их свойств. Начнём с настройки свойств формы. Выделите форму, щёлкнув ЛКМ в пустом месте формы. На панели свойств задайте свойства формы как представлено ниже:

- FormBorderStyle (Стиль границы формы): Fixed3D;
- MaximizeBox (Кнопка развёртывания формы во весь экран): False;
- MinimizeBox (Кнопка свёртывания формы на панель задач): False;
- Text (Текст надписи в заголовке формы): База данных «Студент».

На форме выделите надпись, щёлкнув по ней ЛКМ и на панели свойств, задайте свойства надписи следующим образом:

- AutoSize (Автора размер): False;
- Font (Шрифт): Microsoft Sans Serif, размер 14;
- ForeColor (Цвет текста): Тёмно синий;
- Text (Текст надписи): База данных «Студент»;
- TextAlign (Выравнивание текста): MiddleCenter.

У кнопок задайте надписи (свойство «Text») как показано на рисунке 9.3.

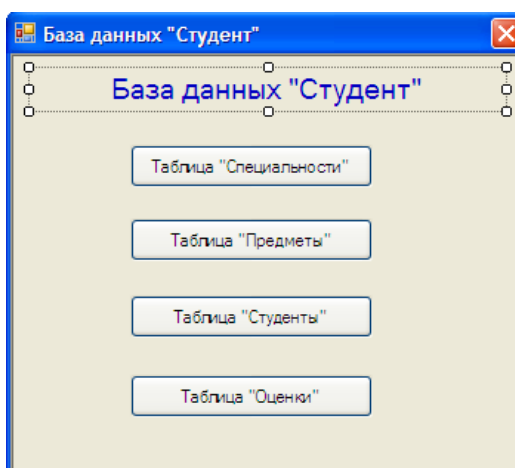


Рис.9.3

После настройки свойств вышеперечисленных объектов форма примет вид представленный на рисунке 9.3.

Теперь перейдём к созданию простых ленточных форм для работы с данными. Для начала создадим ленточную форму, отображающую таблицу «Специальности». Добавим в проект новую пустую форму. Для этого в оконном меню выберите пункт «Project/Add Windows Form». Появится окно «Add New Item - StudentsDB» (Добавить новый компонент) (Рис.9.4).

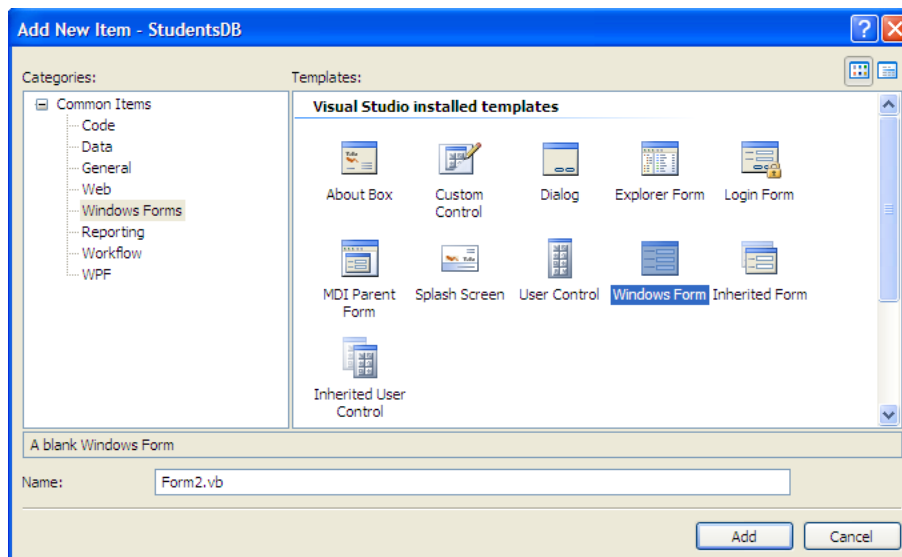


Рис.9.4

В данном окне в разделе «Categories:» (Категории) выберите «Windows Forms» (Формы Windows), затем в разделе «Templates:» (Шаблоны) выберите «Windows Form» (Форма Windows) и нажмите кнопку «Add» (Добавить). Новая пустая форма появится в рабочей области среды разработки.

В верхней части новой формы создайте надпись (Label), как это показано на рисунке 9.5.



Рис.9.5

Перейдём к настройке свойств формы и надписи. Выделите форму, щёлкнув ЛКМ в пустом месте формы. На панели свойств задайте свойства формы следующим образом:

- FormBorderStyle (Стиль границы формы): Fixed3D;
- MaximizeBox (Кнопка развёртывания формы во весь экран): False;
- MinimizeBox (Кнопка свёртывания формы на панель задач): False;
- Text (Текст надписи в заголовке формы): Таблица «Специальности».

На форме выделите надпись, щёлкнув по ней ЛКМ и на панели свойств, задайте свойства надписи как показано ниже:

- AutoSize (Авторазмер): False;
- Font (Шрифт): Microsoft Sans Serif, размер 14;
- ForeColor (Цвет текста): Тёмно синий;
- Text (Текст надписи): Таблица «Специальности»;
- TextAlign (Выравнивание текста): MiddleCenter.

После настройки всех вышеперечисленных свойств форма будет выглядеть следующим образом (Рис.9.6):

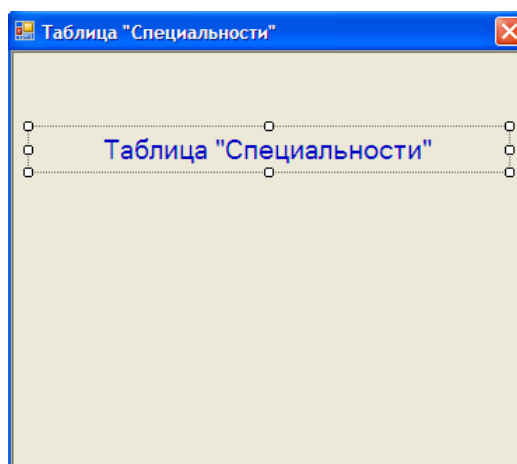


Рис.9.6

Теперь поместим на форму поля таблицы «Специальности». Сначала откройте панель «Источники данных» (Data Sources), щёлкнув по её вкладке в правой части окна среды разработки (смотри Рис.8.6). На панели «Источники данных» отобразите поля таблицы «Специальности», щёлкнув по значку «+», расположенному слева от имени таблицы (Рис.9.7).

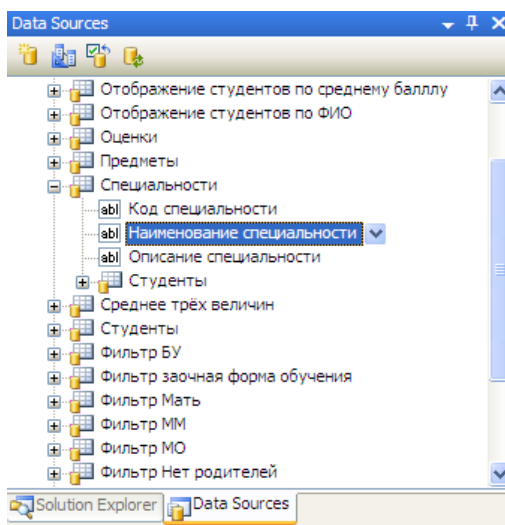


Рис.9.7

Панель «Источники данных» примет вид, представленный на рисунке 9.7.

Замечание: Под полями таблицы специальности в виде подтаблицы располагается таблица «Студенты» (Рис.9.7). Подтаблица показывает, что таблица «Студенты» является вторичной по отношению к таблице специальности.

Замечание: При выделении, какого либо поля таблицы, оно будет отображаться в виде выпадающего списка (Рис.9.7), позволяющего выбрать объект, отображающий содержимое выделенного поля (Рис.9.8).

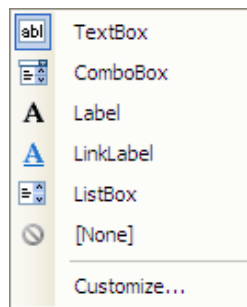


Рис.9.8

Для того чтобы поместить на новую форму поля таблицы их необходимо перетащить из панели «Источники данных» на форму. Из таблицы «Специальности» перетащите мышью на форму поля «Наименование специальности» и «Описание специальности». Форма примет вид, представленный на рисунке 9.9

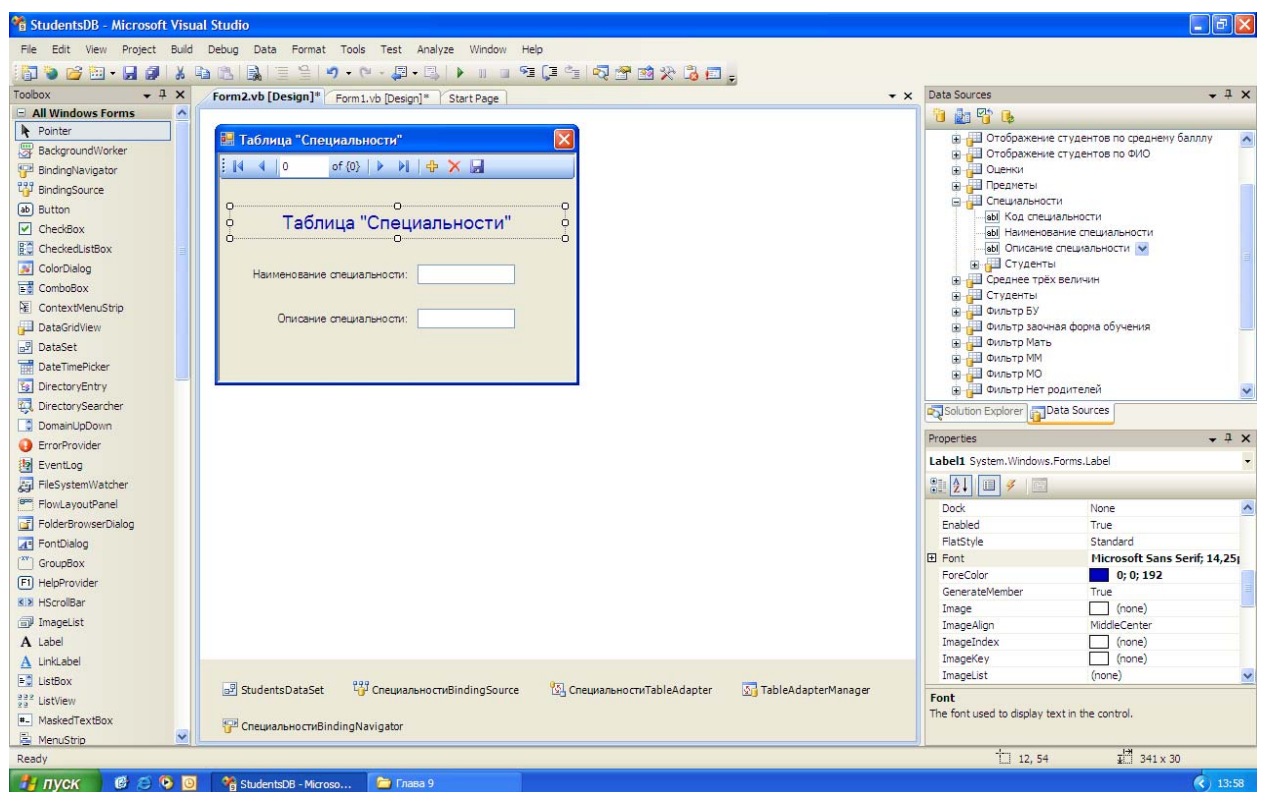


Рис.9.9

Замечание: Мы не помещаем поле «Код специальности» на нашу форму, так как данное поле является первичным полем связи и заполняется автоматически. Конечный пользователь не должен видеть такие поля.

Замечание: Обратите внимание, что после перетаскивания полей с панели «Источники данных» на форму в верхней части формы появилась навигационная панель, а в нижней части рабочей области среды разработки появились пять невидимых объектов. Эти объекты предназначены для связи нашей формы с таблицей «Специальности», расположенной на сервере. Рассмотрим функции этих объектов:

- StudentDataSet (Набор данных Student) – обеспечивает подключение формы к конкретной БД на сервере (в нашем случае это БД Students);
- СпециальностиBindingSource (Источник связи для таблицы «Специальности») – обеспечивает подключение к конкретной таблице (в

нашем случае к таблице специальности), а также позволяет управлять таблицей;


- СпециальностиTableAdapter (Адаптер таблиц для таблицы «Специальности») – обеспечивает передачу данных с формы в таблицу и наоборот.
- TableAdapterManager (Менеджер адаптера таблиц) – управляет работой объекта СпециальностиTableAdapter;
- СпециальностиBindingNavigator (Панель управления таблицей «Специальности») – голубая панель с кнопками управления таблицей, расположенная в верхней части формы.

Теперь нам необходимо проверить работоспособность новой формы. Для отображения формы «Специальности» её необходимо подключить к главной кнопочной форме, а затем запустить проект и открыть форму «Специальности» при помощи кнопки на главной кнопочной форме.

Отобразите главную кнопочную форму в рабочей области среды разработки, щёлкнув по вкладке «Form1.vb [Design]» в верхней части рабочей области. Для подключения новой формы «Специальности» к главной кнопочной форме дважды щёлкните ЛКМ по кнопке «Таблица «Специальности»», расположенной на главной кнопочной форме (рис.9.3). В появившемся окне кода формы в процедуре «Button1_Click» наберите команду «Form2.Show()», предназначенную для открытия формы «Таблица «Специальности»» (Form2), как это показано на рисунке 9.10.

```
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Form2.Show()
    End Sub
End Class
```

Рис.9.10

Теперь запустим проект, нажав на панели инструментов кнопку . На экране появится главная кнопочная форма. Для открытия формы, отображающей таблицу «Специальности» на главной кнопочной форме нажмите кнопку «Таблица «Специальности»». Появится форма с соответствующей таблицей (Рис.9.11).

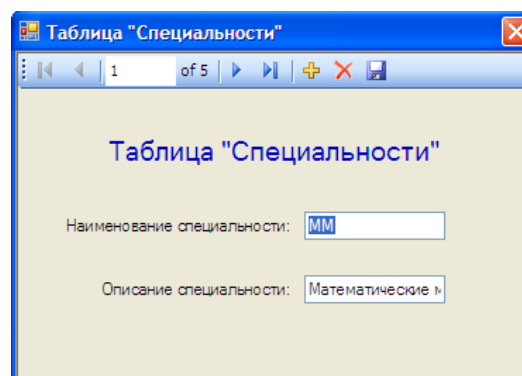


Рис.9.11

Проверьте работу панели навигации, расположенной в верхней части формы, понажимав на ней различные кнопки. Вернитесь в среду разработки, просто закрыв форму с таблицей «Специальности» и главную кнопочную форму.

Теперь создадим форму для просмотра таблицы предметы. Добавьте в проект новую форму. На форму добавьте надпись. Настройте свойства формы и надписи, как это было сделано для формы таблицы «Специальности». Затем из таблицы «Предметы» на

новую форму поместите поля «Наименование предмета» и «Описание предмета». После выполнения всех вышеописанных действий форма для таблицы предметы будет выглядеть следующим образом (Рис.9.12):

Рис.9.12

На главной кнопочной форме дважды щёлкните ЛКМ по кнопке «Таблица «Предметы»» и в появившемся окне кода в процедуре «Button2_Click» наберите «Form3.Show()» (Рис.9.13).

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    Form3.Show()
End Sub
End Class
```

Рис.9.13

Проверим работу новой формы, отображающей таблицу «Предметы». Запустите проект и на главной кнопочной форме нажмите кнопку «Таблица «Предметы»». Отобразится таблица предметы имеющая следующий вид (Рис.9.14):

Рис.9.14

Проверьте работу панели навигации, нажатием на кнопки на данной панели в верхней части формы. Для возвращения в среду разработки закройте форму таблицы «Предметы» и главную кнопочную форму.

Теперь создадим простую ленточную форму для отображения таблицы «Студенты». Для начала отобразите поля таблицы «Студенты» на панели «Источники данных», щёлкнув ЛКМ по знаку «+», расположенному слева от названия таблицы. Отобразятся все поля таблицы «Студенты» (Рис.9.15).

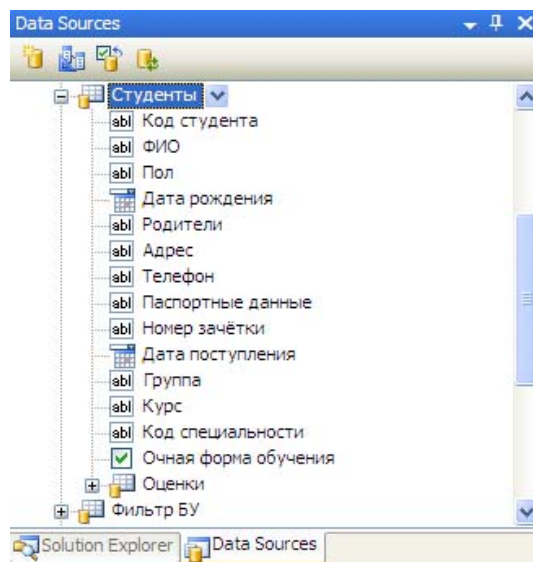


Рис.9.15

Замечание: Обратите внимание на тот факт, что поля «Дата рождения» и «Дата поступления» отображаются объектом «Выбор даты» (DatePicker), так как данные поля содержат значения дат. Поле «Очная форма обучения» является логическим, следовательно, для его отображения используется объект «Переключатель» (CheckBox). Остальные поля отображаются при помощи текстовых полей ввода (TextBox) (Рис.9.15).

Создайте новую форму и поместите в её верхнюю часть надпись. Задайте заголовок формы как «Таблица «Студенты»». В верхнюю часть формы поместите надпись. В качестве текста надписи задайте тот же самый текст, что был задан в качестве заголовка формы. Настройте свойства формы и надписи, аналогично формам созданным ранее. На форму с панели «Источники данных» переместите все поля кроме поля «Код студента». Так как данное поле является первичным полем связи. Новая форма примет вид (Рис.9.16):

Рис.9.16

Обратите внимание на объекты, отображающие поля «Дата рождения», «Дата поступления» и «Очная форма обучения».

Подключим форму, отображающую таблицу «Студенты» к главной кнопочной форме. Отобразите главную кнопочную форму и на ней дважды щелкните ЛКМ по кнопке «Таблица «Студенты»». В появившемся окне кода, в процедуре «Button3_Click» наберите следующую команду для открытия формы таблицы «Студенты» - «Form4.Show» (Рис.9.17).

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
    Form4.Show()
End Sub
```

Рис.9.17

Теперь запустим проект. На экране появится главная кнопочная форма. Для открытия формы, отображающей таблицу «Студенты» на главной кнопочной форме нажмите кнопку «Таблица «Студенты»». Появится форма с соответствующей таблицей (Рис.9.18).

Рис.9.18

Проверьте работу формы нажатием кнопок на панели навигации, расположенной в верхней части формы. Закройте форму, отображающую таблицу «Студенты» и главную кнопочную форму.

Аналогичным образом создайте форму для отображения таблицы «Оценки». Добавьте на новую форму надпись, добавьте на форму все поля из таблицы «Оценки» и настройте их свойства, как описано выше. В итоге, форма для отображения таблицы «Оценки» будет выглядеть следующим образом (Рис.9.19):

Таблица "Оценки"

Код студента:

Дата экзамена 1: 4 ноября 2008 г. ▼

Код предмета 1:

Оценка 1:

Дата экзамена 2: 4 ноября 2008 г. ▼

Код предмета 2:

Оценка 2:

Дата экзамена 3: 4 ноября 2008 г. ▼

Код предмета 3:

Оценка 3:

Средний балл:

Рис.9.19

Подключите вновь созданную форму таблицы «Оценки» к главной кнопочной форме. Для этого отобразите главную кнопочную форму и на ней дважды щёлкните ЛКМ по кнопке «Таблица «Оценки»». В появившемся окне с кодом, в процедуре «Button4_Click» наберите команду «Form5.Show» (Рис.9.20).

```
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click
    Form5.Show()
End Sub
```

Рис.9.20

Проверьте работу формы таблицы «Оценки», запустив проект, и на главной кнопочной форме нажмите кнопку «Таблица «Оценки»». Появится вновь созданная форма (Рис.9.21).

Таблица "Оценки"

Код студента: 1

Дата экзамена 1: 1 февраля 2008 г. ▼

Код предмета 1: 1

Оценка 1: 5

Дата экзамена 2: 9 февраля 2008 г. ▼

Код предмета 2: 4

Оценка 2: 3

Дата экзамена 3: 14 февраля 2008 г. ▼

Код предмета 3: 2

Оценка 3: 4

Средний балл: 0

Рис.9.21

В заключение, откройте обозреватель проекта (Solution Explorer) щёлкнув по его вкладке в правой части окна среды разработки. На данной панели должны отобразиться все выше созданные формы (Рис.9.22).

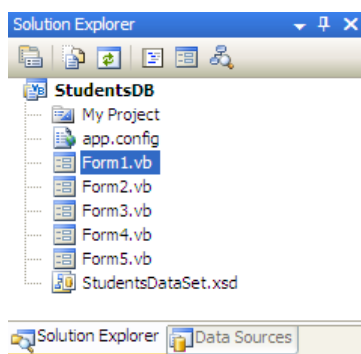


Рис.9.22

Лабораторная работа 10. Создание сложных ленточных форм для работы с данными

Цель: научиться создавать сложные ленточные формы для работы с данными

Модернизируем форму для таблицы «Студенты». Сначала программно продублируем кнопки панели навигации, расположенной в верхней части формы. Откройте проект «StudentsDB» и отобразите форму таблицы студенты (Form4). В нижней части формы расположите семь кнопок, как это показано на рисунке 10.1.

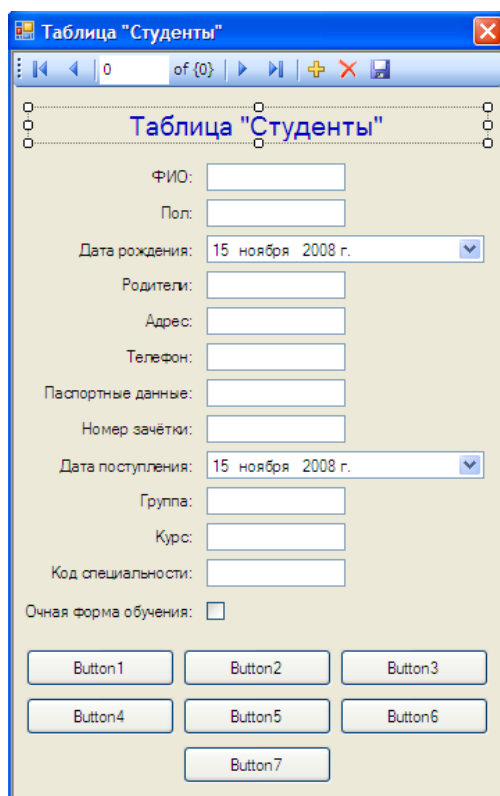
The image shows a screenshot of a Microsoft Access form titled "Таблица 'Студенты'". The form has a light beige background and a blue border. At the top, there is a navigation bar with a status bar showing "0 of {0}" and navigation icons. Below the navigation bar, the title "Таблица 'Студенты'" is displayed in a large, bold, blue font. The form contains several text boxes and dropdown menus for data entry: "ФИО:", "Пол:", "Дата рождения:" (with a dropdown menu showing "15 ноября 2008 г."), "Родители:", "Адрес:", "Телефон:", "Паспортные данные:", "Номер зачёта:", "Дата поступления:" (with a dropdown menu showing "15 ноября 2008 г."), "Группа:", "Курс:", "Код специальности:", and a checkbox labeled "Очная форма обучения:". At the bottom of the form, there are seven buttons arranged in three rows: "Button1", "Button2", "Button3" in the first row; "Button4", "Button5", "Button6" in the second row; and "Button7" centered in the third row.

Рис.10.1

В качестве надписей на созданных кнопках (Свойство «Caption») задайте как: «Первая», «Предыдущая», «Добавить», «Последняя», «Следующая», «Удалить» и «Сохранить» (Рис.10.2).

Рис.10.2

Дважды щёлкните ЛКМ по кнопке «Первая» и в появившемся окне кода формы «Form4» в процедуре «Button1_Click» наберите команду для перехода к первой записи «СтудентыBindingSource.MoveFirst()» (Рис.10.3).

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    СтудентыBindingSource.MoveFirst()
End Sub
```

Рис.10.3

Дважды щёлкните ЛКМ по кнопке «Предыдущая» и в появившемся окне кода формы «Form4» в процедуре «Button2_Click» наберите команду для перехода к предыдущей записи «СтудентыBindingSource.MovePrevious()» (Рис.10.4).

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    СтудентыBindingSource.MovePrevious()
End Sub
```

Рис.10.4

Дважды щёлкните ЛКМ по кнопке «Добавить» и в появившемся окне кода формы «Form4» в процедуре «Button3_Click» наберите команду для добавления новой записи «СтудентыBindingSource.AddNew()» (Рис.10.5).

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
    СтудентыBindingSource.AddNew()
End Sub
```

Рис.10.5

Дважды щёлкните ЛКМ по кнопке «Последняя» и в появившемся окне кода формы «Form4» в процедуре «Button4_Click» наберите команду для перехода к последней записи «СтудентыBindingSource.MoveLast()» (Рис.10.6).

```
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click
    СтудентыBindingSource.MoveLast()
End Sub
```

Рис.10.6

Дважды щёлкните ЛКМ по кнопке «Следующая» и в появившемся окне кода формы «Form4» в процедуре «Button5_Click» наберите команду для перехода к следующей записи «СтудентыBindingSource.MoveNext()» (Рис.10.7).

```
Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button5.Click
    СтудентыBindingSource.MoveNext()
End Sub
```

Рис.10.7

Дважды щёлкните ЛКМ по кнопке «Удалить» и в появившемся окне кода формы «Form4» в процедуре «Button6_Click» наберите команду для удаления текущей записи «СтудентыBindingSource.RemoveCurrent()» (Рис.10.8).

```
Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button6.Click
    СтудентыBindingSource.RemoveCurrent()
End Sub
```

Рис.10.8

Дважды щёлкните ЛКМ по кнопке «Сохранить» и в появившемся окне кода формы «Form4» в процедуре «Button7_Click» наберите команду для сохранения изменений, отображённую на рисунке 10.9.

```
Private Sub Button7_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button7.Click
    Me.Validate()
    Me.СтудентыBindingSource.EndEdit()
    Me.TableAdapterManager.UpdateAll(Me.StudentsDataSet)
End Sub
```

Рис.10.9

Рассмотрим последнюю процедуру более подробно. Она содержит следующие команды:

- Me.Validate() – проверяет введенные в поля данные на соответствие типам данных полей;
- Me.СтудентыBindingSource.EndEdit() – закрывает подключение с сервером;
- Me.TableAdapterManager.UpdateAll(Me.StudentsDataSet) – обновляет данные на сервере.

Для проверки работы созданных кнопок запустите проект откройте форму «Таблица «Студенты»» и нажмите каждую из кнопок.

Теперь изменим объекты, отображающие поля для более удобного ввода информации. Для начала удалите текстовые поля ввода (TextBox), отображающие следующие поля таблицы «Студенты»: «Пол», «Родители», «Телефон», «Паспортные данные», «Номер зачётки», «Курс» и «Код специальности». После удаления, перечисленных полей форма, отображающая таблицу «Студенты» примет следующий вид (Рис.10.10):

Таблица "Студенты"

0 of {0}

Таблица "Студенты"

ФИО:

Пол:

Дата рождения: 15 ноября 2008 г.

Родители:

Адрес:

Телефон:

Паспортные данные:

Номер зачётки:

Дата поступления: 15 ноября 2008 г.

Группа:

Курс:

Код специальности:

Очная форма обучения: ☐

Первая Предыдущая Добавить

Последняя Следующая Удалить

Сохранить

Рис.10.10


Для отображения полей «Телефон», «Паспортные данные» и «Номер зачётки» будем использовать текстовые поля ввода по маске (MaskedTextBox). Объект текстовое поле ввода по маске отсутствует в выпадающем списке объектов для отображения полей в окне «Источники данных», поэтому будем создавать данные объекты при помощи панели объектов (Toolbox), а затем подключать их к соответствующим полям вручную. Для создания текстовых полей ввода по маске на панели объектов используется кнопка  MaskedTextBox. Создайте текстовые поля ввода по маске справа от надписей «Телефон», «Паспортные данные» и «Номер зачётки», как это показано на рисунке 10.11.

Таблица "Студенты"

ФИО:

Пол:

Дата рождения: 15 ноября 2008 г.

Родители:

Адрес:

Телефон:

Паспортные данные:

Номер зачётки:

Дата поступления: 15 ноября 2008 г.

Группа:

Курс:

Код специальности:

Очная форма обучения: ☐

Первая Предыдущая Добавить

Последняя Следующая Удалить

Сохранить

Рис.10.11

Теперь у созданных объектов настроим маски ввода. Начнём с объекта, отображающего номер зачётки. На форме выделите соответствующее поле «Номер зачётки» текстовое поле ввода по маске. Для задания маски в меню действий с объектом выберите пункт «Set Mask...» (Установить маску...) (Рис.10.12).

Таблица "Студенты"

ФИО:

Пол:

Дата рождения: 15 ноября 2008 г.

Родители:

Адрес:

Телефон:

Паспортные данные:

Номер зачётки:

Дата поступления: 15 ноября 2008 г.

Группа:

Курс:

Код специальности:

Очная форма обучения: ☐

Первая Предыдущая Добавить


Последняя Следующая Удалить

Сохранить

MaskedTextBox Tasks

Set Mask...

Рис.10.12

Замечание: Для отображения меню действий в верхнем правом углу объекта необходимо нажать кнопку  (Рис.10.12).

После выбора пункта «Set Mask...» на экране появится окно задания маски «Input Mask» (Введите маску) (Рис.10.13).

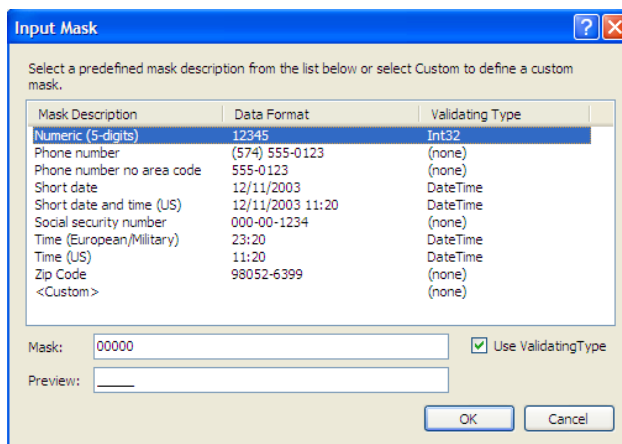


Рис.10.13

В окне «Input Mask» выберите маску «Numeric (5-digits)» (Числовое (5-цифр)) и нажмите кнопку «Ok» (Рис.10.13).

Для текстового поля ввода по маске для поля «Паспортные данные» задайте маску как показано на рисунке 10.14.

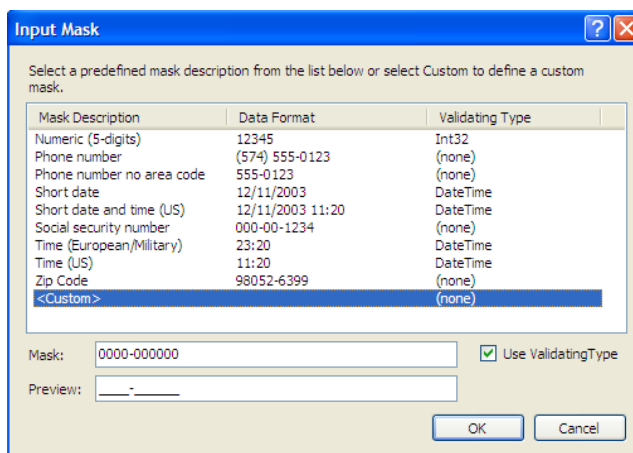


Рис.10.14

Замечание: Обратите внимание, что паспортные данные отображаются как четыре числа, тире, шесть чисел. То есть в поле «Mask» (Маска) нужно задать «0000-000000». Знак «0» обозначает цифру. В поле «Preview» (Предварительный просмотр) отображается вид текстового поля ввода по маске на форме.

После определения маски для поля «Паспортные данные» в окне «Input Mask» нажмите кнопку «Ok».

Теперь зададим маску для текстового поля ввода по маске отображающего поле «Телефон». Задайте маску как показано на рисунке 10.15.

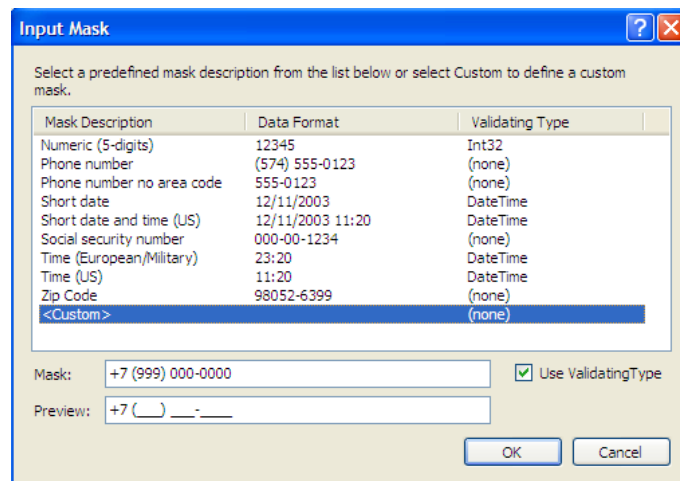


Рис.10.15

Теперь нам необходимо подключить созданные текстовые поля ввода по маске к соответствующим полям. Для этого с панели «Источники данных» (DataSources) перетащите поле «Номер зачётки» на текстовое поле ввода по маске, расположенное справа от надписи «Номер зачётки». Прodelайте такую же операцию с полями «Паспортные данные» и «Телефон», перетащив их на соответствующие им текстовые поля ввода по маске.

На этом мы заканчиваем работу с текстовыми полями ввода по маске и переходим к отображению поля «Курс» при помощи числового счётчика (объект NumericUpDown). Для этого, на панели «Источники данных» нажмите кнопку, расположенную справа от поля «Курс» и в выпадающем списке выберите объект для отображения данного поля как «NumericUpDown» (Рис.10.16).

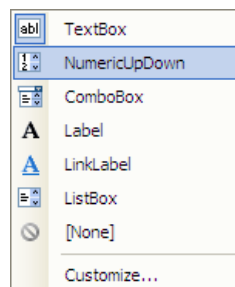


Рис.10.16

Затем перетащите поле на форму мышью, расположив, его справа от надписи «Курс».

Замечание: После перетаскивания поля «Курс» на форму слева от него появится ещё одна надпись «Курс». Удалите ее, щёлкнув по ней ЛКМ, а затем нажав кнопку «Delete» на клавиатуре.

Отобразим поля «Пол» и «Родители» в виде выпадающих списков (Объект ComboBox). Для этого, на панели «Источники данных» нажмите кнопку, расположенную справа от поля «Пол» и в выпадающем списке выберите объект для отображения данного поля как «ComboBox» (Рис.10.17).

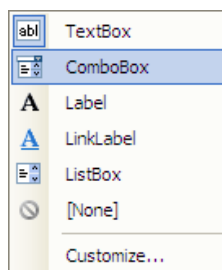


Рис.10.17

Такую же операцию проделайте с полем «Родители». Затем перетащите мышью поля на форму, расположив их напротив соответствующих надписей. Удалите лишние надписи.

Теперь заполним выпадающие списки. Выделите выпадающий список, отображающий поле «Пол». На панели свойств (Properties) и нажмите кнопку в свойстве «Items» (Элементы списка). Появится окно «String Collection Editor» (Редактор строковых коллекций) (Рис.10.18).

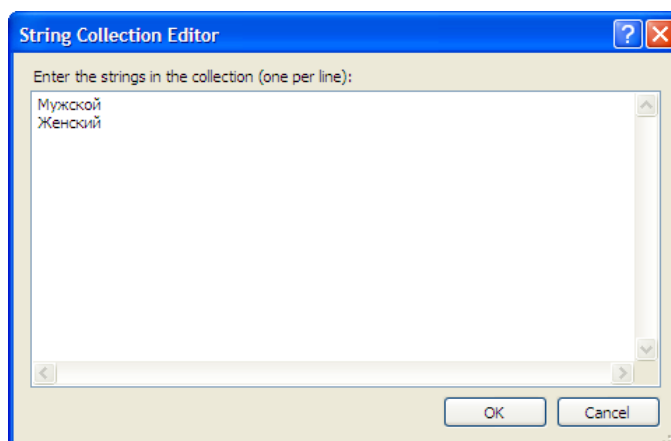


Рис.10.18

В появившемся окне в отдельных строках наберите элементы выпадающего списка: «Мужской» и «Женский» (Рис.10.18). Затем нажмите кнопку «Ok».

Для выпадающего списка, отображающего поле «Родители», проделайте аналогичную операцию, только в качестве пунктов списка задайте: «Отец и Мать», «Мать», «Отец» и «Нет» (Рис.10.19).

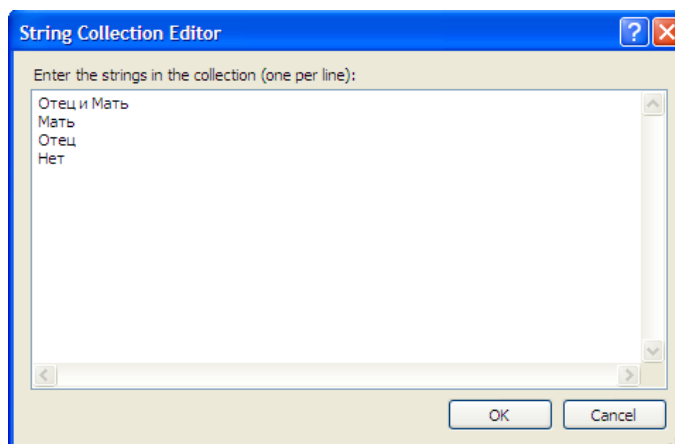



Рис.10.19

В заключение отобразим вместо поля «Код специальности» специальность соответствующую заданному коду, при помощи выпадающего списка. При этом сам

выпадающий список будет заполнен специальностями из таблицы «Специальности» и при выборе специальности её код будет автоматически подставляться в поле «Код специальности» таблицы «Студенты».

Поместите справа от надписи «Код специальности», неподключённый ни к каким полям выпадающий список. Для создания выпадающего списка на панели объектов воспользуйтесь кнопкой  **ComboBox**.

После создание выпадающего списка подключим его к полю «Код специальности» из таблицы «Студенты» и настроим заполнение списка значениями поля «Наименование специальности» из таблицы студенты. Для этого выделите вновь созданный выпадающий список, отобразите меню действий и в меню действий включите опцию «Use data bound items» (Использовать связанные с данными элементы списка) (Рис.10.20).

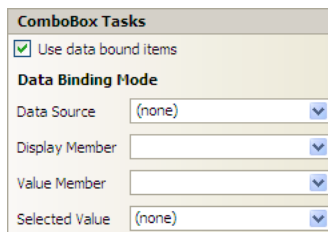



Рис.10.20

В панели действий под опцией «Use data bound items» расположены следующие параметры:

- Data Source (Источник данных) – определяет таблицу или запрос из которого заполняется список;
- Display Member (Член отображения) – определяет поле значениями которого заполняется список;
- Value Member (Член значений) – определяет значения какого поля подставляются в связанное с выпадающим списком поле;
- Selected Value (Выбранное значение) – определяет связанное с выпадающим списком поле.

Для изменения параметров необходимо нажать кнопку  в нутрии поля параметра. Появится древовидная структура выбора источника данных (Рис.10.21).

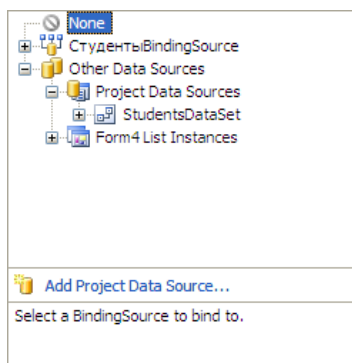


Рис.10.21

В нашем случае зададим вышеперечисленные параметры следующим образом:

- Параметр «DataSource» задайте как «Other Data Sources\Project Data Sources\StudentsDataSet\Специальности»;
- Параметр «DataMember» задайте как «Наименование специальности»;
- Параметр «Value Member» задайте как «Код специальности»;

- Параметр «Selected Value» задайте как «СтудентыBindingSource\Код специальности».

После задания всех вышеперечисленных параметров панель действий выпадающего списка примет вид (Рис.10.22):

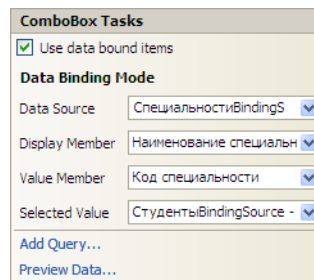


Рис.10.22

Обратите внимание на то, что на панели невидимых объектов, расположенной в нижней части рабочей области среды разработки, появилось два новых объекта: «СпециальностиBindingSource» и «СпециальностиTableAdapter» (Рис.10.23).

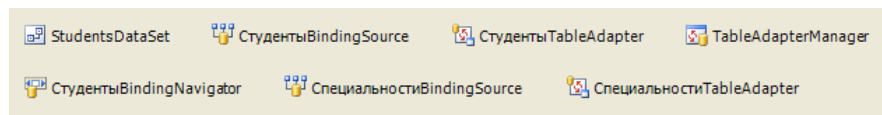


Рис.10.23

Данные объекты предназначены для заполнения выпадающего списка значениями поля «Наименование специальности» таблицы «Специальности».

После всех вышеперечисленных действий форма, отображающая таблицу «Студенты» примет вид, представленный на рисунке 10.24.

Рис.10.24

Проверим работу формы, отображающей таблицу «Студенты». Запустите проект и на главной кнопочной форме нажмите кнопку «Таблица «Студенты»». Появится форма, имеющая следующий вид (Рис.10.25):

Таблица "Студенты"

ФИО: Иванов А.И.

Пол: Мужской

Дата рождения: 12 декабря 1983 г.

Родители: Отец и Мать

Адрес: Москва

Телефон: +7 (495) 789-5674

Паспортные данные: 8567-567543

Номер зачётки: 13245

Дата поступления: 1 сентября 2007 г.

Группа: ММ11

Курс: 1

Код специальности: ММ

Очная форма обучения: ☒

Первая Предыдущая Добавить

Последняя Следующая Удалить

Сохранить

Рис.10.25

На этом мы заканчиваем работу с формой, отображающей таблицу «Студенты» и переходим к реализации вычисляемых полей. Для этого рассмотрим форму, отображающую таблицу «Оценки» (Form5). Рассмотрим вычисление поля «Средний балл» на основе среднего трёх полей:

Отобразите форму для таблицы «Оценки», щёлкнув ЛКМ по её вкладке в верхней части рабочей области среды разработки. На форму, справа от поля «Средний балл» поместите кнопку (Рис.10.26).

Таблица "Оценки"

Код студента:

Дата экзамена 1: 15 ноября 2008 г.

Код предмета 1:

Оценка 1:

Дата экзамена 2: 15 ноября 2008 г.

Код предмета 2:

Оценка 2:

Дата экзамена 3: 15 ноября 2008 г.

Код предмета 3:

Оценка 3:

Средний балл: Button1

Рис.10.26

Задайте свойство «Text» у вновь созданной кнопки как «Вычислить» (Рис.10.27).

Таблица "Оценки"

Код студента:

Дата экзамена 1: 15 ноября 2008 г.

Код предмета 1:

Оценка 1:

Дата экзамена 2: 15 ноября 2008 г.

Код предмета 2:

Оценка 2:

Дата экзамена 3: 15 ноября 2008 г.

Код предмета 3:

Оценка 3:

Средний балл: Вычислить


Рис.10.27

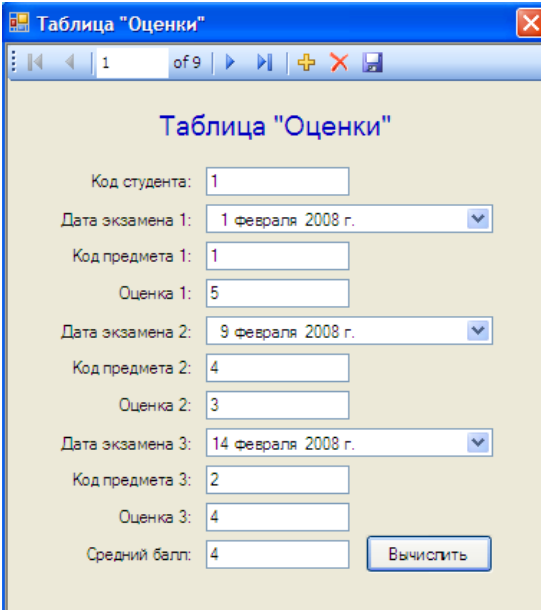
Теперь дважды щёлкните ЛКМ по кнопке «Вычислить» и в появившемся коде процедуры «Button1_Click» наберите код, представленный на рисунке 10.28, вычисляющий среднее оценок.

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Средний_баллTextBox.Text = (Val(Оценка_1TextBox.Text) + Val(Оценка_2TextBox.Text) + Val(Оценка_3TextBox.Text)) / 3
End Sub
```

Рис.10.28

Теперь проверим, как работает наша вновь созданная кнопка для вычисления поля «Средний балл». Запустите проект и на главной кнопочной форме нажмите кнопку «Таблица «Оценки»». Появится форма, отображающая таблицу «Оценки», на форме нажмите кнопку «Вычислить». Будет вычислен средний балл по оценкам. Если нажать

кнопку сохранения на панели инструментов формы , то средний балл будет сохранен в таблицу «Оценки» (Рис.10.29).



The screenshot shows a software window titled "Таблица "Оценки"". It features a toolbar at the top with navigation and action icons. The main area contains a form with the following fields and values:

Field	Value
Код студента:	1
Дата экзамена 1:	1 февраля 2008 г.
Код предмета 1:	1
Оценка 1:	5
Дата экзамена 2:	9 февраля 2008 г.
Код предмета 2:	4
Оценка 2:	3
Дата экзамена 3:	14 февраля 2008 г.
Код предмета 3:	2
Оценка 3:	4
Средний балл:	4

A button labeled "Вычислить" is located at the bottom right of the form.

Рис.10.29

На этом мы заканчиваем рассмотрение ленточных форм и переходим к рассмотрению табличных форм.

Лабораторная работа 11. Создание табличных форм

Цель: научиться создавать табличные формы

Перейдём теперь к созданию табличных форм для отображения данных. В данной главе также затрагиваются вопросы фильтрации и сортировки данных, а также реализуется поиск информации в таблице.

Рассмотрим создание табличной формы на примере формы, отображающей таблицу «Студенты». Добавьте в проект новую форму и на неё поместите следующие объекты:

- четыре надписи (Label),
- пять кнопок (Button),
- выпадающий список (ComboBox),
- текстовое поле ввода (TextBox),
- группирующую рамку (GroupBox),
- список (ListBox),
- два переключателя (RadioButton).

Расположите объекты как показано на рисунке 11.1.

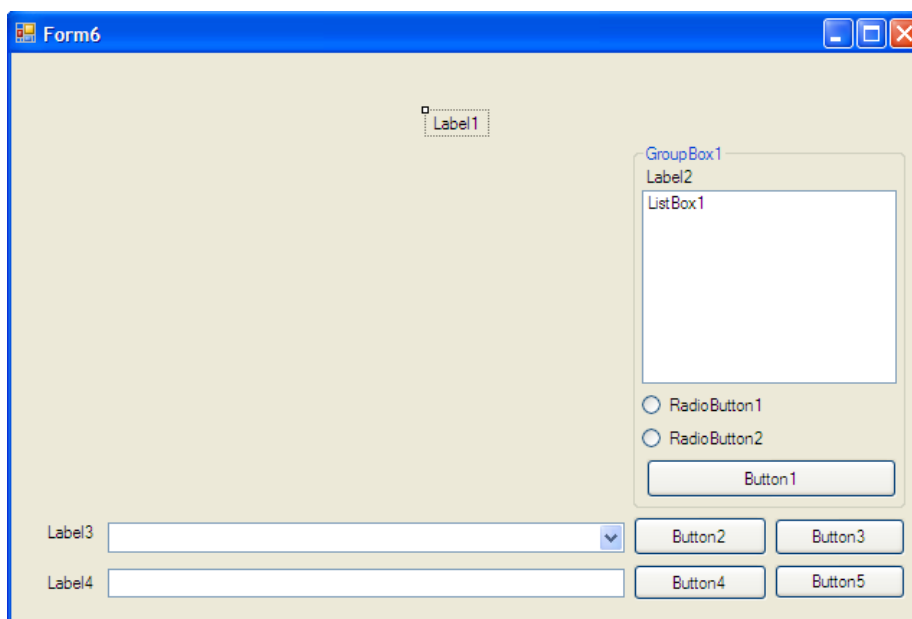



Рис.11.1

Замечание: Для создания объекта группирующая рамка используется кнопка ### на панели объектов (Toolbox), а для создания переключателя – кнопка ##.

Добавим на форму таблицу для отображения данных (DataGridView) из таблицы «Студенты». Для этого на панели «Источники данных» (Data Sources), нажмите кнопку , расположенную справа от таблицы «Студенты». В появившемся списке объектов для отображения всей таблицы выберите «DataGridView» (Рис.11.2).

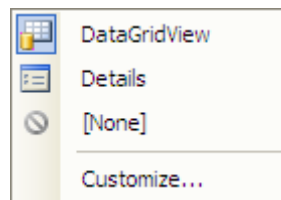


Рис.11.2

Перетащите таблицу «Студенты» из панели «Источники данных» на форму. Форма примет следующий вид (Рис.11.3):

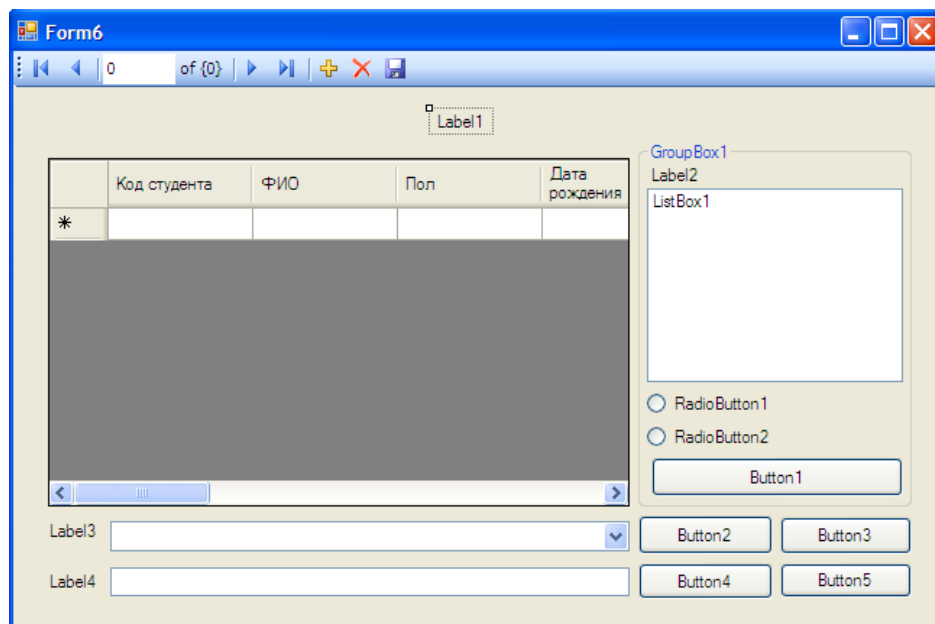


Рис.11.3

Обратите внимание на то, что на форме появилась таблица для отображения данных, подключённая к таблице «Студенты». Также появились объекты связи и панель навигации (Рис.11.4).

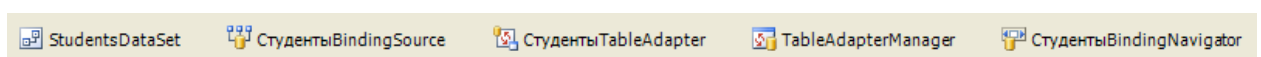


Рис.11.4

Теперь перейдём к настройке свойств объектов. Начнём с настройки свойств формы. Задайте свойства формы следующим образом:

- FormBorderStyle (Стиль границы формы): Fixed3D;
- MaximizeBox (Кнопка развёртывания формы во весь экран): False;
- MinimizeBox (Кнопка свёртывания формы на панель задач): False;
- Text (Текст надписи в заголовке формы): Таблица «Студенты» (Табличный вид).

Задайте свойства надписей (Label1, Label2, Label3 и Label4) как:

- AutoSize (Авторазмер): False;
- Text (Текст надписи): «Таблица «Студенты» (Табличный вид)», «Поле для сортировки», «ФИО:» и «Критерий» (Соответственно для Label1, Label2, Label3 и Label4).

Для надписи Label1 задайте:

- Font (Шрифт): Microsoft Sans Serif, размер 14;
- ForeColor (Цвет текста): Тёмно синий;

- TextAlign (Выравнивание текста): MiddleCenter.

Задайте надписи на кнопках как: «Сортировать», «Фильтровать», «Показать все», «Найти» и «Закрыть» (Соответственно для кнопок Button1, Button2, Button3, Button4 и Button5). Для того чтобы нельзя было произвести сортировку не выбрав поля изначально заблокируем кнопку «Сортировать» (Button1).

У группирующей рамки задайте заголовок (Свойство Text) равным «Сортировка». У переключателей (Объекты RadioButton1 и RadioButton2) задайте надписи как «Сортировка по возрастанию» и «Сортировка по убыванию», а у переключателя «Сортировка по возрастанию» (RadioButton1) задайте свойство Checked (Включён) равное True (Истина).

Заполните список (ListBox1) значениями, представленными на рисунке 11.5, а затем нажмите кнопку «Ok».

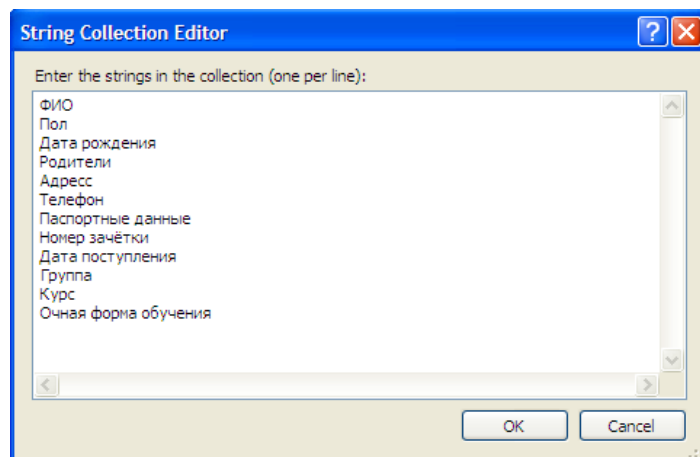



Рис.11.5

Настроим таблицу для отображения данных, удалив из неё поля с кодами. Выделите таблицу на форме и отобразите её меню действий, щёлкнув ЛКМ по кнопке , расположенной в верхнем правом углу таблицы. В меню действий выберите пункт «Edit columns...» (Рис.11.6).

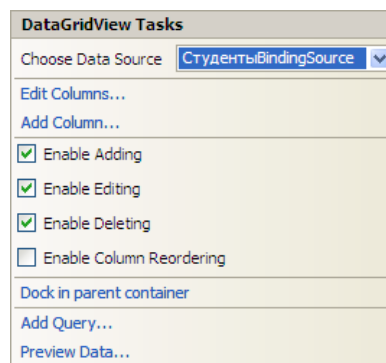


Рис.11.6

Появится окно настройки свойств полей таблицы «Edit Columns» (Рис.11.7).

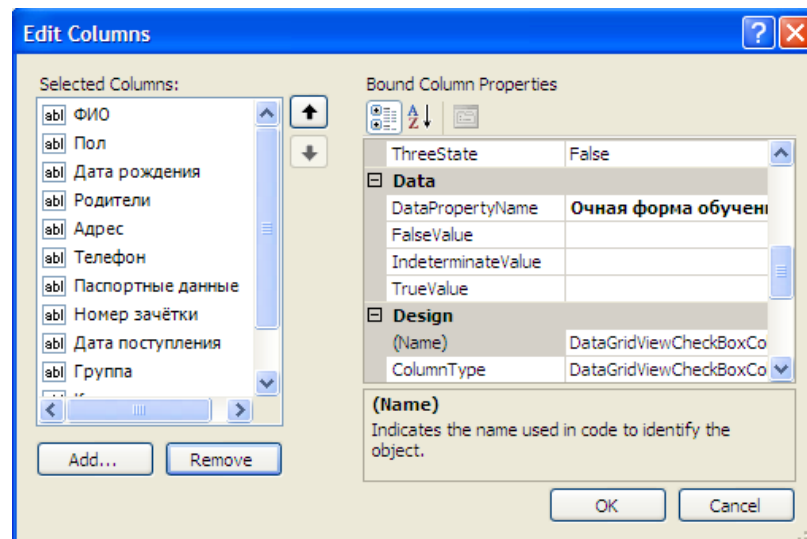


Рис.11.7

В окне «Edit Columns» из списка полей удалите поля «Код студента» и «Код специальности», выделив их и нажав кнопку «Remove» (Удалить). Список полей примет вид показанный на рисунке 11.7. Для закрытия окна редактирования полей, и сохранения изменений нажмите кнопку «Ok».

Настроим заполнение выпадающего списка именами студентов из таблицы студенты. Отобразите меню действий выпадающего списка. Включите опцию «Use Data Bound Items». Установите параметр «Data Source» равным «Other Data Sources\Project Data Sources\StudentsDataSet\Студенты», а параметр «Display Member» равным «ФИО». Остальные параметры оставьте без изменений (Рис.11.8).

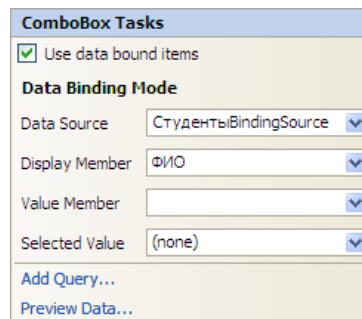


Рис.11.8

Закройте окно действий выпадающего списка. На панели невидимых объектов появится дополнительный объект связи «СтудентыBindingSource1», предназначенный для заполнения выпадающего списка (Рис.11.9).

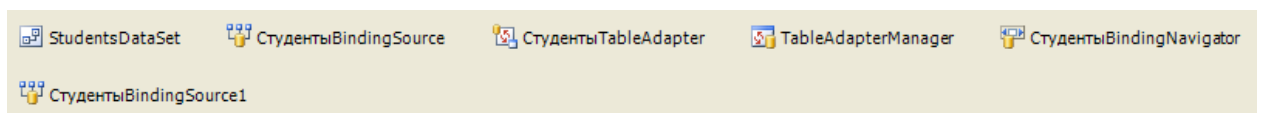


Рис.11.9

После настройки всех вышеперечисленных свойств объектов новая форма примет вид (Рис.11.10):

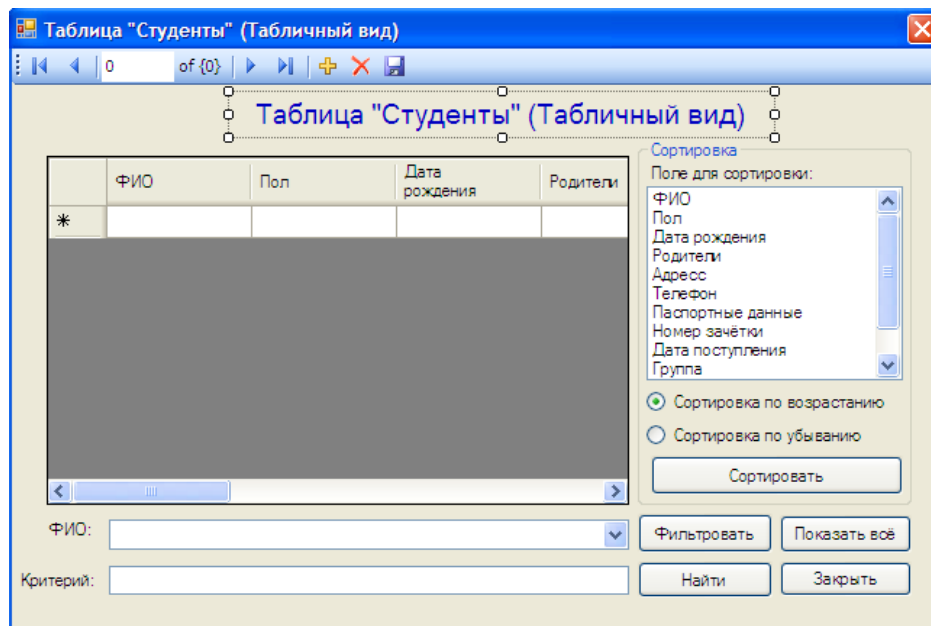


Рис.11.10

На этом мы заканчиваем настройку свойств объектов и переходим к написанию кода обработчиков событий объектов.

Работу с кодом начнём написания кода для разблокирования кнопки «Сортировать», при выборе пункта списка (ListBox1). Для создания процедуры события дважды щёлкните ЛКМ по списку. Появится процедура обработки события, происходящего при выборе пункта списка (ListBox1_SelectedIndexChanged). В процедуре наберите команду разблокировки кнопки «Сортировать» (Button1): Button1.Enabled = True (Рис.11.11).

```
Private Sub ListBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Button1.Enabled = True
End Sub
```

Рис.11.11

Теперь перейдём к созданию кода сортирующего нашу таблицу в зависимости от выбранного поля и порядка сортировки при нажатии кнопки «Сортировать». Дважды щёлкните ЛКМ по кнопке «Сортировать». Появится процедура «Button1_Click», выполняемая при щелчке ЛКМ по кнопке. В процедуре наберите код, представленный на рисунке 11.12.

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim Col As System.Windows.Forms.DataGridColumn
    Select Case ListBox1.SelectedIndex
        Case 0
            Col = DataGridViewTextBoxColumn2
        Case 1
            Col = DataGridViewTextBoxColumn3
        Case 2
            Col = DataGridViewTextBoxColumn4
        Case 3
            Col = DataGridViewTextBoxColumn5
        Case 4
            Col = DataGridViewTextBoxColumn6
        Case 5
            Col = DataGridViewTextBoxColumn7
        Case 6
            Col = DataGridViewTextBoxColumn8
        Case 7
            Col = DataGridViewTextBoxColumn9
        Case 8
            Col = DataGridViewTextBoxColumn10
        Case 9
            Col = DataGridViewTextBoxColumn11
        Case 10
            Col = DataGridViewTextBoxColumn12
    End Select
    If RadioButton1.Checked Then
        СтудентыDataGridView.Sort(Col, System.ComponentModel.ListSortDirection.Ascending)
    Else
        СтудентыDataGridView.Sort(Col, System.ComponentModel.ListSortDirection.Descending)
    End If
End Sub

```

Рис.11.12

Рассмотрим код более подробно:

- Команда `Dim Col As System.Windows.Forms.DataGridColumn` создаёт переменную `Col` для хранения имени выбранного столбца таблицы;
- Затем следует блок `Select Case...End Select`, присваивающий в переменную `Col` имя выбранного столбца таблицы в зависимости от номера выбранного пункта списка (`ListBox1.SelectedIndex`). Если выбран первый пункт списка, то в переменную `Col` записывается столбец `DataGridViewTextBoxColumn2`, если второй, то – `DataGridViewTextBoxColumn3` и так далее. Хотелось бы отметить тот факт, что нумерация пунктов списка начинается с нуля, а нумерация столбцов с единицы. Первый столбец «ФИО» носит имя `DataGridViewTextBoxColumn2`, так как имя `DataGridViewTextBoxColumn1` имеет столбец заголовков строк;
- Блок `If...End If` выполняет следующую операцию: если включён переключатель «Сортировка по возрастанию» (`RadioButton1`), то отсортировать таблицу по полю заданному в переменной `Col` по возрастанию (`СтудентыDataGridView.Sort (Col, System.ComponentModel.ListSortDirection.Ascending)`), иначе по убыванию (`СтудентыDataGridView.Sort (Col, System.ComponentModel.ListSortDirection.Descending)`).

Рассмотрим код обработчика события нажатия кнопки «Фильтровать» (`Button2`). Дважды щёлкните по кнопке «Фильтровать» и в процедуре обработки события «`Button2_Click`» наберите код: `СтудентыBindingSource.Filter = "ФИО='" & ComboBox1.Text & "'"` (Рис.11.13).

```

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    СтудентыBindingSource.Filter = "ФИО='" & ComboBox1.Text & "'"
End Sub

```

Рис.11.13

Замечание: У объекта `СтудентыBindingSource` имеется текстовое свойство `Filter` (Рис.11.13), которое определяет условие фильтрации. Условие фильтрации имеет синтаксис: "<Имя поля><Оператор>'<Значение>' ". В нашем случае значение поля «ФИО» приравнивается к значению, выбранному в выпадающем списке (`ComboBox1.Text`) (Рис.11.13).

Теперь перейдём к кнопке «Показать всё», отменяющей фильтрацию записей. Дважды щёлкните по вышеперечисленной кнопке. Появится процедура `Button2_Click`. В появившейся процедуре наберите команду `СтудентыBindingSource.Filter = ""` (Рис.11.14).

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
    СтудентыBindingSource.Filter = ""
End Sub
```

Рис.11.14

Заметим, что если присвоить свойству «`Filter`» значение пустой строки (""), то его действие будет отменено (Рис.11.14).

Далее рассмотрим реализацию поиска информации в таблице. Дважды щёлкните по кнопке «Найти». В появившейся процедуре обработки нажатия кнопки «`Button4_Click`» наберите следующий код (Рис.11.15).

```
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click
    For i = 0 To СтудентыDataGridView.ColumnCount - 1
        For j = 0 To СтудентыDataGridView.RowCount - 1
            СтудентыDataGridView.Item(i, j).Style.BackColor = Color.White
            СтудентыDataGridView.Item(i, j).Style.ForeColor = Color.Black
        Next j
    Next i
    For i = 0 To СтудентыDataGridView.ColumnCount - 1
        For j = 0 To СтудентыDataGridView.RowCount - 1
            If InStr(СтудентыDataGridView.Item(i, j).Value, TextBox1.Text) Then
                СтудентыDataGridView.Item(i, j).Style.BackColor = Color.AliceBlue
                СтудентыDataGridView.Item(i, j).Style.ForeColor = Color.Blue
            End If
        Next j
    Next i
End Sub
```

Рис.11.15

Рассмотрим более подробно код вышеприведённой процедуры. Данная процедура состоит из двух частей:

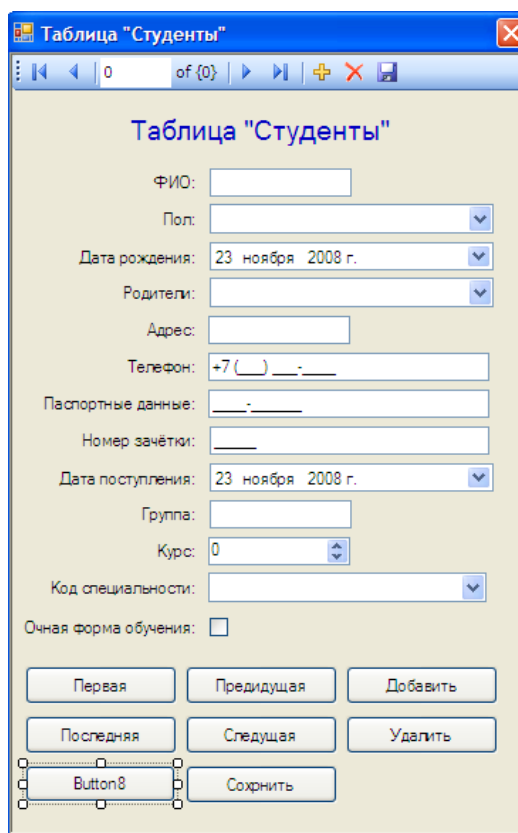
- Первый блок `For i=0.....Next i`. перебирает все ячейки таблицы и устанавливает в них белый цвет фона и чёрный цвет текста. То есть, отменяет результаты предыдущего поиска;
- Второй блок `For i=0.....Next i`. перебирает все ячейки таблицы и если они содержат текст, введённый в поле ввода (`TextBox1`), то устанавливает в них голубой цвет фона и синий цвет текста, чем выделяет искомые ячейки.

Наконец рассмотрим код для кнопки «Заккрыть». Дважды щёлкните ЛКМ по этой кнопке и в появившейся процедуре «`Button5_Click`» наберите команду «`Me.Close()`», закрывающую выше рассматриваемую форму (Рис.11.16).

```
Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button5.Click
    Me.Close()
End Sub
```

Рис.11.16

В заключение создадим кнопку на ленточной форме, отображающей таблицу «Студенты», для отображения соответствующей табличной формы. Откройте ленточную форму для таблицы «Студенты» (Form4) и поместите на неё новую кнопку, как это показано на рисунке 11.17.



The screenshot shows a Windows-style application window titled "Таблица "Студенты"". The window contains a form with various input fields and buttons. The fields are labeled: ФИО, Пол, Дата рождения, Родители, Адрес, Телефон, Паспортные данные, Номер зачётки, Дата поступления, Группа, Курс, Код специальности, and Очная форма обучения. The buttons are labeled: Первая, Предыдущая, Добавить, Последняя, Следующая, Удалить, Button8, and Сохранить. The Button8 button is highlighted with a dashed border and a small square handle, indicating it is being edited or moved.

Рис.11.17

Задайте надпись у новой кнопки (свойство Text), как «Таблица». Форма примет следующий вид (Рис.11.18):

Рис.11.18

Подключим к кнопке «Таблица» созданную ранее табличную форму (Form6). Для этого дважды щёлкните ЛКМ по кнопке «Таблица» и в появившейся процедуре «Button8_Click» наберите команду «Form6.Show» (Рис.11.19).

```
Private Sub Button8_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Form6.Show()
End Sub
```

Рис.11.19

Теперь проверим работоспособность созданной табличной формы. Запустите проект и на главной кнопочной форме нажмите кнопку «Таблица «Студенты»». На появившейся ленточной форме, отображающей таблицу «Студенты» нажмите кнопку «Таблица». Появится новая табличная форма (Рис.11.20).

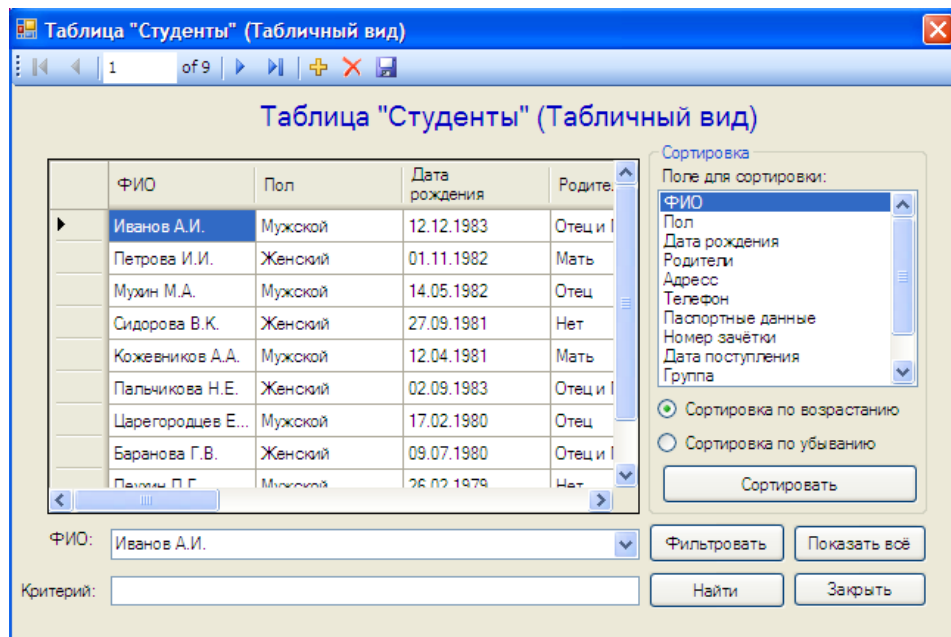


Рис.11.20

Проверьте, как работает поиск, фильтрация и сортировка записей в таблице, нажимая на соответствующие кнопки. После проверки работы формы для возвращения в среду разработки просто закройте все формы.

Хотелось бы отметить тот факт, что после проведения всех вышеописанных действий панель обозревателя проекта (Solution Explorer) примет вид (Рис.11.21):

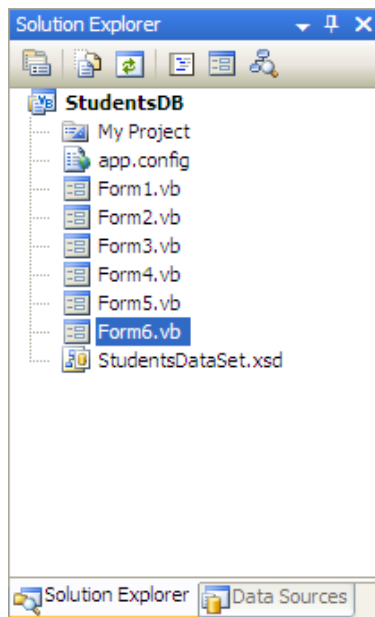


Рис.11.21

На этом мы заканчиваем работу с формами для работы с данными и переходим к отчётам.

Лабораторная работа 12. Отчёты

Цель: научить создавать отчёты

Начнём рассмотрение отчётов с создания ленточного отчёта, отображающего таблицу «Студенты». Для начала добавим в проект новый пустой отчёт. Для этого в оконном меню выберите пункт «Project\Add New Item...» (Рис.12.1).

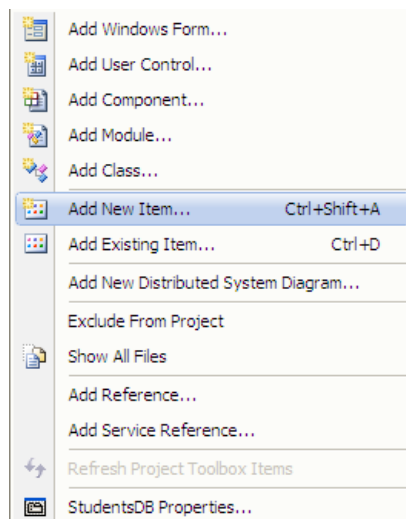


Рис.12.1

Появится окно «Add New Item-StudentsDB» (Добавить новый элемент - StudentsDB). В данном окне в списке «Categories» (Категории) выберите пункт «Reporting» (Отчёты), затем в области «Templates» (Шаблоны) выберите шаблон «Report» (Отчёт) и нажмите кнопку «Add» (Добавить) (Рис.12.2).

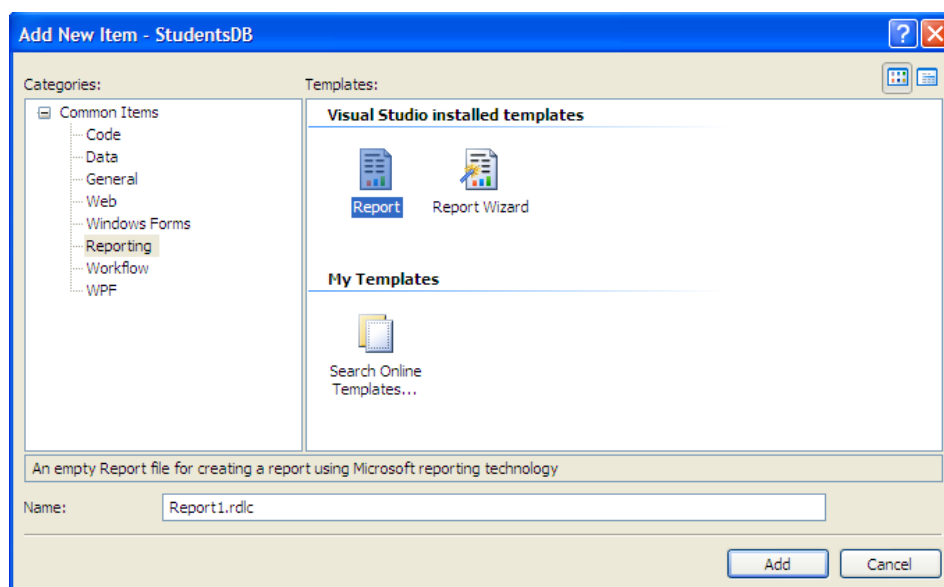


Рис.12.2

В рабочей области среды разработки появится пустой отчёт. Новый отчёт также отобразится и на панели обозревателя проекта (Solution Explorer) (Рис.12.3).

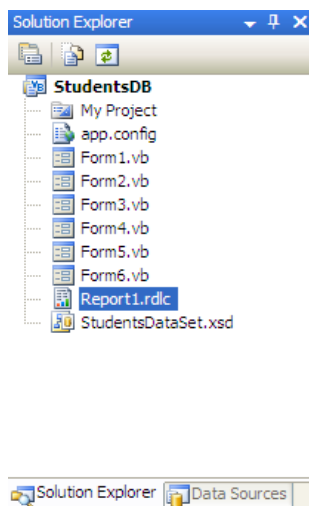



Рис.12.3


Для того чтобы в отчёт поместить поля таблицы «Студенты» в него необходимо добавить объект «Table» (Таблица). Для этого на панели объектов (Toolbox) нажмите кнопку  Table, а затем в отчёте нарисуйте прямоугольник. Отчёт примет вид, представленный на рисунке 12.4.

	Header	
	Detail	
	Footer	

Рис.12.4

Замечание: Объект таблица имеет три строки:

- Header (заголовок) – верхняя часть первой страницы отчёта, содержит заголовок отчёта;
- Detail (область данных) – средняя часть каждой страницы отчёта, содержит поля отображаемой таблицы;
- Footer (примечание) – нижняя часть последней страницы отчёта, содержит итоговую информацию по отчёту.

Добавим в таблицу в область данных дополнительные строки для отображения полей таблицы «Студенты». Выделите область данных, как это показано на рисунке 12.5, щёлкнув ЛКМ по заголовку строки области данных .

	Header	
	Detail	
	Footer	

Рис.12.5

Для вставки новой строки щёлкните ПКМ по заголовку выделенной строки  и в появившемся меню выберите пункт «Insert Row Below» (Вставить строку ниже) (Рис.12.6).

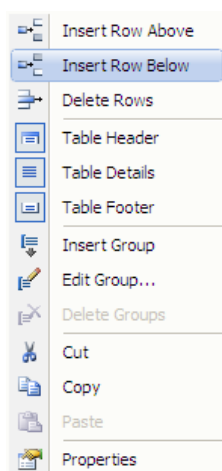


Рис.12.6

Проделайте эту операцию одиннадцать раз. Таблица примет вид (Рис.12.7)

	Header	
	Footer	

Рис.12.7

Наша таблица имеет три столбца. Один столбец является лишним, так как нам необходимо всего два столбца: один для отображения полей таблицы «Студенты», а другой для подписей к ним. Поэтому удалим первый столбец таблицы. Щёлкните ПКМ по заголовку первого столбца таблицы и в появившемся меню выберите пункт «Delete Columns» (Удалить строки) (Рис.12.8).

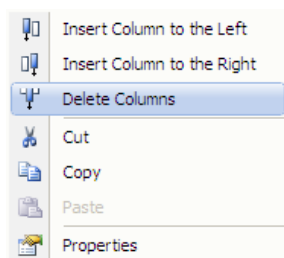


Рис.12.8

Теперь объединим ячейки в заголовке и примечании таблицы. Выделите ячейки заголовка таблицы, обведя их курсором мыши удерживая ЛКМ. Затем щёлкните по

выделению ПКМ и в появившемся меню выберите пункт «Merge Cells» (Объединить ячейки) (Рис.12.9).

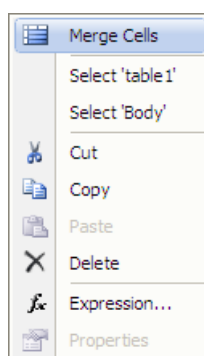


Рис.12.9

Проделайте аналогичную операцию с ячейками примечания. После выполнения вышеперечисленных операций таблица примет вид (Рис.12.10):

[illegible]

Рис.12.10

Поместим в таблицу поля таблицы «Студенты». Для этого перетащите поля таблицы «Студенты» с панели «Источники данных» (Data Sources) в ячейки правого столбца таблицы, расположенные под заголовком, как показано на рисунке 12.11. В одну ячейку перетаскивается одно поле. Поля связи, имеющие в своём имени слово «Код» перетаскивать не нужно.

	Header	
		=Fields!ФИО.Value
		=Fields!Пол.Value
		=Fields!Дата_рождения.Value
		=Fields!Родители.Value
		=Fields!Адрес.Value
		=Fields!Телефон.Value
		=Fields!Паспортные_данные.Value
		=Fields!Номер_зачётки.Value
		=Fields!Дата_поступления.Value
		=Fields!Группа.Value
		=Fields!Курс.Value
		=Fields!Очная_форма_обучения.Value
	Footer	

Рис.12.11

В левом столбце таблицы наберите имена полей и установите их выравнивание по правому краю (Свойство TextAlign). В заголовке наберите заголовок отчёта «Отчёт таблицы «Студенты»» и сделайте выравнивание текста в нём по центру (Рис.12.12).


Отчёт таблицы "Студенты"	
ФИО:	=Fields!ФИО.Value
Пол:	=Fields!Пол.Value
Дата рождения:	=Fields!Дата_рождения.Value
Родители:	=Fields!Родители.Value
Адрес:	=Fields!Адрес.Value
Телефон:	=Fields!Телефон.Value
Паспортные данные:	=Fields!Паспортные_данные.Value
Номер зачётки:	=Fields!Номер_зачётки.Value
Дата поступления:	=Fields!Дата_поступления.Value
Группа:	=Fields!Группа.Value
Курс:	=Fields!Курс.Value
Очная форма обучения:	=Fields!Очная_форма_обучения.Value
Footer	


Рис.12.12

Теперь выделим ячейки, отображающие поле «ФИО» серым цветом для логического отделения одного студента от другого. Выделите вторую строку таблицы и на панели свойств (Properties) в свойстве «BackColor» (Цвет фона) выберите серый цвет. Таблица примет следующий вид (Рис.12.13).

Отчёт таблицы "Студенты"	
ФИО:	=Fields!ФИО.Value
Пол:	=Fields!Пол.Value
Дата рождения:	=Fields!Дата_рождения.Value
Родители:	=Fields!Родители.Value
Адрес:	=Fields!Адрес.Value
Телефон:	=Fields!Телефон.Value
Паспортные данные:	=Fields!Паспортные_данные.Value
Номер зачётки:	=Fields!Номер_зачётки.Value
Дата поступления:	=Fields!Дата_поступления.Value
Группа:	=Fields!Группа.Value
Курс:	=Fields!Курс.Value
Очная форма обучения:	=Fields!Очная_форма_обучения.Value
Footer	

Рис.12.13


Заключительным шагом в настройке таблицы будет включение отображения границ ячеек. Выделите все ячейки с полями и подписями к ним. Затем на панели инструментов при помощи кнопки  включите границы выделенных ячеек таблицы (Рис.12.14).

Замечание: Если кнопка  отсутствует на панели инструментов, то необходимо включить панель редактирования границ отчётов (Report borders). Для этого щёлкните ПКМ по панели инструментов и в появившемся меню выберите пункт «Report borders».

Отчёт таблицы "Студенты"	
ФИО:	=Fields!ФИО.Value
Пол:	=Fields!Пол.Value
Дата рождения:	=Fields!Дата_рождения.Value
Родители:	=Fields!Родители.Value
Адрес:	=Fields!Адрес.Value
Телефон:	=Fields!Телефон.Value
Паспортные данные:	=Fields!Паспортные_данные.Value
Номер зачётки:	=Fields!Номер_зачётки.Value
Дата поступления:	=Fields!Дата_поступления.Value
Группа:	=Fields!Группа.Value
Курс:	=Fields!Курс.Value
Очная форма обучения:	=Fields!Очная_форма_обучения.Value
Footer	

Рис.12.14

Теперь создадим форму отображающий созданный отчёт. Добавьте в проект новую форму (Form7). Определите заголовок формы (Свойство Text) как «Отчёт таблицы «Студенты»».

Поместите на форму специальный объект, отображающий отчёты «MicrosoftReportViewer», используя кнопку  MicrosoftReportViewer, расположенную на панели объектов (Toolbox). К объекту, отображающему отчёты подключите, созданный ранее отчёт. Для этого в меню действий в выпадающем списке «Choose report» (Выберите отчёт) выберите отчёт «StudentsDB.Report1.rdlc».

Разверните объект, отображающий отчёты во всю форму. Для этого в меню действий объекта выберите пункт «Dock in Parent Container» (Развернуть в родительский контейнер). Меню действий примет вид (Рис.12.15):

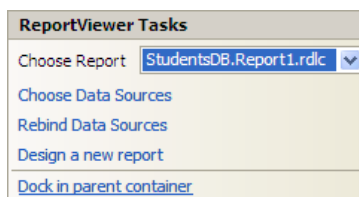


Рис.12.15

Замечание: Обратите внимание на тот факт, что после подключения отчёта к объекту, отображающему отчёты, на панели невидимых объектов появились объекты связи, подключающие отчёт к таблице «Студенты» (Рис.12.16).

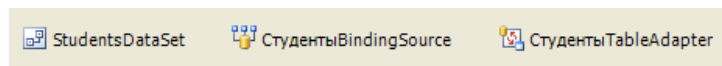


Рис.12.16

После выполнения всех вышеперечисленных действий форма, отображающая отчёт примет вид, представленный на рисунке 12.17.

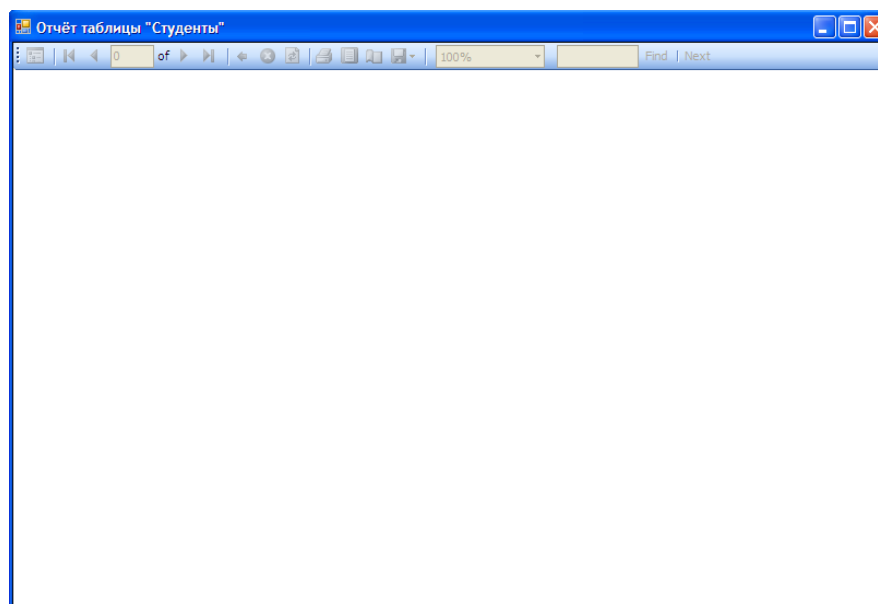


Рис.12.17

Проверим работоспособность нового отчёта, подключив форму для его отображения к кнопке на форме «Таблица «Студенты»». На форме, отображающей таблицу «Студенты» создайте кнопку (Button9) (Рис.12.18).

Рис.12.18

Задайте надпись на кнопке (Свойство Text) равную «Отчёт» (Рис.12.19).

Рис.12.19

Теперь определим код обработчика события нажатия кнопки. Дважды щёлкните ЛКМ по кнопке «Отчёт» и в появившейся процедуре «Button9_Click» наберите команду «Form7.Show()» (Рис.12.20).

```
Private Sub Button9_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button9.Click
    Form7.Show()
End Sub
```

Рис.12.20

Запустите проект и на главной кнопочной форме нажмите кнопку «Таблица «Студенты»». На появившейся ленточной форме, отображающей таблицу «Студенты» нажмите кнопку «Отчёт». Появится новая форма с отчётом, построенным по таблице «Студенты» (Рис.12.21).

Отчёт таблицы "Студенты"

ФИО:	Иванов А.И.
Пол:	Мужской
Дата рождения:	12/12/1983 12:00:00 AM
Родители:	Отец и Мать
Адрес:	Москва
Телефон:	+74957895674
Паспортные данные:	8567-567543
Номер зачётки:	13245
Дата поступления:	9/1/2007 12:00:00 AM
Группа:	ММ11
Курс:	1
Очная форма обучения:	True
ФИО:	Петрова И.И.
Пол:	Женский
Дата рождения:	11/1/1982 12:00:00 AM
Родители:	Мать
Адрес:	Москва
Телефон:	+74957889876
Паспортные данные:	4567-765432
Номер зачётки:	34563
Дата поступления:	8/1/2006 12:00:00 AM
Группа:	ПИ21
Курс:	2
Очная форма обучения:	False

Рис.12.21

Проверьте работу отчёта. Для завершения работы проекта просто закройте все открытые формы.

На этом мы завершаем разработку нашей БД «Студент».