Документ подписан простой электронной подписью Информация о владельце: ФИО: Шебзу уритатика перемпремана УКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ Должность и ракура Льное госуда РСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ федерального университета Дата подписания: 12.09.2023 10:21:20 Уникальный программный ключ: «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ» d74ce93cd40e39275c3ba2f58486412a1c8ef96f

Пятигорский институт (филиал) СКФУ

Методические указания

по выполнению лабораторных работ по дисциплине «ПРОЕКТНЫЙ ПРАКТИКУМ» для студентов направления подготовки /специальности

09.03.02 Информационные системы и технологии

(ЭЛЕКТРОННЫЙ ДОКУМЕНТ)

1 Цель и задачи освоения дисциплины	3
2 Место дисциплины в структуре образовательной программы	3
3 Связь с предшествующими дисциплинами	3
4 Связь с последующими дисциплинами	3
5 СОДЕРЖАНИЕ ЛАБОРАТОРНЫХ РАБОТ	4
6 КРИТЕРИИ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ	111
7 МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРЫ	
ОЦЕНИВАНИЯ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ И (ИЛИ) ОПЫТА	
ДЕЯТЕЛЬНОСТИ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ	
КОМПЕТЕНЦИЙ	112
8 УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ	
ДИСЦИПЛИНЫ	112

1 Цель и задачи освоения дисциплины

Целью изучения дисциплины является формирование набора общепрофессиональных и профессиональных компетенций бакалавра по направлению 09.03.02«Информационные системы и технологии».

Задачи освоения дисциплины:

- приобретение практических навыков комплексного использования методов, инструментальных средств проектирования и сопровождения
 - информационных систем; навыков управления ИТ- проектами;
- освоение методик проектирования обеспечивающих подсистем ИС и расчета экономической эффективности ИТ-проекта.

2 Место дисциплины в структуре образовательной программы

Дисциплина «Проектный практикум» относится к дисциплинам по выбору вариативной части блока Б1 учебного плана подготовки бакалавров направления 09.03.02 Информационные системы и технологии. Ее освоение происходит в 8 семестре.

3 Связь с предшествующими дисциплинами

Содержание данной учебной дисциплины опирается на знание дисциплин: Интеллектуальные системы и технологии, Методы и средства проектирования информационных систем и технологий, Безопасность информационных систем, Основы Web-технологий, Инструментальные средства информационных систем, Базы данных в распределенных системах обработки информации, Основы численного моделирования, Численные методы в научных расчетах.

4 Связь с последующими дисциплинами

Знания, полученные при изучении данной дисциплины, необходимы дисциплин: Преддипломная практика, Защита выпускной квалификационной работы, включая подготовку к процедуре защиты и процедуру защиты.

5 СОДЕРЖАНИЕ ЛАБОРАТОРНЫХ РАБОТ

Лабораторная работа № 1. «Выполнение индивидуальных заданий по проектированию информационных систем в Microsoft SQL Server 2012.»

Форма проведения: лабораторная работа (3 часа).

Цель работы:

- Изучение основных конструкций структурированного языка запросов SQL.
- Изучения среды MS SQL Server Management Studio.
- Приобретение навыков проектирования структур данных.

Базы данных составляют основу для построения информационных систем любого масштаба и предназначения. В теории баз данных одними из основных являются вопросы, связанные с анализом предметной области и моделированием структуры данных, управлением данными и их анализом.

Основой любой базы данных является реализованная в ней модель данных, представляющая собой множество структур данных, ограничений целостности и операций манипулирования данными. С помощью модели данных могут быть представлены объекты предметной области и существующие между ними связи.

Результатом лабораторной работы будет создание реляционной базы данных на основе MS SQL Server 2005.

В реляционной базе данных данные представлены в виде собрания таблиц. Таблица состоит из определенного числа столбцов (полей) и произвольного числа строк (записей). Планируемая база данных будет представлять собой информационное хранилище данных об успеваемости студентов и состоять из следующих таблиц:

- - **Speciality** (специальность)
 - Course (курс)
 - Group (группа)
 - **Discipline** (дисциплина)
 - Account (тип отчетности)
 - Mark (отметка)
 - Status (академический статус студента)
 - Position (должность)
 - **People** (люди)
 - Student (студент)
 - Teacher (преподаватель)
 - SemesterResults (результаты сессии, семестра)
- Структура данных таблиц приведена в Приложении.

Начало работы в Microsoft SQL Server Management Studio

Для создания баз данных используем среду Microsoft SQL Server Management Studio. На запрос соединения с сервером выбираем (рис. 1):

- Тип сервера: Компонент Database Engine
- Имя сервера: **SQL-MS.**
- Под таким именем в домене fizmat.vspu.ru. доступна машина, на которой установлены серверные компоненты MS SQL Server 2005. Можно попробовать

выбрать сервер из выпадающего списка серверов. Можно также обратиться к этой машине по IP-адресу 192.168.10.152 из локальной сети.

- Проверка подлинности: Проверка подлинности SQL Server.
- Такая настройка позволяет создавать пользователей данного экземпляра SQL Server независимо от компьютера, с которого производится вход.
- Имя входа: studentMBS21.
- *Пароль:* student.



- Рисунок 1. Окно входа в Microsoft SQL Server Management Studio 2005
- Примечание. Пользователь studentMBS21 обладает большими полномочиями на этом сервере, поэтому пользоваться им надо очень аккуратно. Под этим пользователем мы создадим базу данных, а заполнять её и производить поиск по ней мы будем под другими пользователями. Предпочтительнее всего использовать свою учетную запись в домене fizmat.vspu.ru. В этом случае надо выбирать проверку подлинности Windows.
- Теперь нажимаем кнопку «Параметры» и выбираем (рис. 2):
- Соединение с базой данных \rightarrow Обзор сервера... \rightarrow Пользовательские базы данных \rightarrow trial_base.
- Сетевой протокол → TCP/IP
- Нажимаем кнопку «Соединить».

ход Свойства соединения			
Введите или выберите имя базы да	нных для с	оединения.	
Соединение с базой данных:	<по умол	ианию>	~
Сеть			
Сетевой протокол:	TCP/IP		*
Размер сетевого пакета:	4096	😂 байт	
Соединение			
Время ожидания соединения:	15	😂 сек.	
Время ожидания выполнения:	0	🗘 сек.	
📃 Шифровать соединение			
		Сбросит	ьвсе

- Рисунок 2. Окно входа в Microsoft SQL Server Management Studio 2005 (вкладка Параметры)
- Примечание. База данных trial_base является базой данной по умолчанию для пользователя studentMBS21, она была создана при регистрации этого пользователя. В случае, когда права доступа пользователя не ограничены (как в рассматриваемом случае), вкладку Параметры можно не открывать. Если же пользователь имеет доступ только к определенным базам данных, при подключении к серверу нужно одну из этих баз указывать.
- После успешного соединения с базой данных на экране видим следующую картинку (рис. 3):

🍢 Среда Microsoft SQL Server Management S	tudio
Файл Правка Вид Сервис Окно Сообщест	во Справка
😫 Создать запрос 📑 📸 📸 🔓 🎽	a 🥔 📴 🌺 🦉 🛫 🖕
Обозреватель объектов 🛛 👻 🕂 🗙	Сводка
Соединить • 🖳 🔳 🝸	🔄 🖻 🤿 🍸 🎹 Список 👻 🚰 Отчет 👻
 SQL-MS (SQL Server 9.0.5000 - studentMBS21) Вазы данных Безопасность Объекты сервера Репликация Репликация Управление Notification Services Агент SQL Server 	 SQL-MS (SQL Server 9.0. sqL-M5 Имя Базы данных Базы данных Безопасность Объекты сервера Репликация Управление Управление Notification Services Агент SQL Server

- Рисунок 3. Подключение к SQL серверу установлено
- Среда MS SQL Management Studio предоставляет удобный инструментарий для создания, редактирования, заполнения баз данных. Но настоящие профессионалы в своей работе редко пользуются этой средой, а для выполнения своих задач используют SQL-запросы. Мы будем пользоваться, когда это удобно и наглядно, графическим режимом, но основной упор будем делать на освоении базы языка SQL.

Создание базы данных в среде Microsoft SQL Server Management Studio

 В разделе «Базы данных» правой кнопкой выбираем «Создать базу данных...» (рис. 4). Назовем базу данных по индексу группы – mbs21. Владельцем базы данных назначим пользователя, под именем которого был произведен вход – studentMBS21. В разделе «Параметры» выбираем тип сортировки Cyrillic_General_BIN (для примера), нажимаем ОК.

Выбор страницы Сбщие Параметры Файловые группы	🖾 Сценарий 🔻 🚺 Справка				
	Имя базы данных: mbs21 Владелец: studentMBS21		nbs21		
	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1				
	Использовать г Файлы базы данны	полнотекстов ых:	ое индексирование		
	 Использовать г Файлы базы данна Логическое имя 	полнотекстов ых: Тип файла	ое индексирование Файловая группа	Начальный размер (МБ)	Авт
	Использовать г Файлы базы дання Логическое имя mbs21	полнотекстов ых: Тип файла Данные	ое индексирование Файловая группа PRIMARY	Начальный размер (МБ) 3	Авто

- Рисунок 4. Создание базы данных
- В разделе «Базы данных» Обозревателя объектов появилась вновь созданная mbs21 (проверьте!):



- Рисунок 5. Обозреватель объектов

Создание таблиц базы данных в среде Microsoft SQL Server Management Studio

– Начнем с создания таблицы Speciality. Структура таблицы приведена ниже:

Имя поля		Тип	Возможность содержать
(столбца)	Содержание	данных	NULL
Num	Первичный ключ	int	нет
Name	Название специальности	varchar(60)	нет

- В реляционных базах данных первичный ключ используется как уникальный идентификатор записи. Это поле является обязательным, оно используется для связи таблиц по внешним ключам (примеры такого связывания будут рассмотрены далее). Первичный ключ должен иметь целочисленный тип (в данном случае - *int*). Во втором поле будет храниться название специальности некоторая строка, поэтому мы выбираем для этого поля тип *varchar*(60). Число в скобках означает максимальное число символов в строке. Детальную информацию об этих типах можно посмотреть в справке.
- Простейшим образом можно создавать таблицы средствами MS SQL Server Management Studio (правая кнопка мыши на заголовке «Таблицы» > Создать таблицу.). Получаем следующее:

Обозреватель объектов 🛛 👻 🗣	×	Таблица - dbo.Table_1	Сводка	
Соединить 🕶 🕎 🔳 📝 🍸		Имя столбца	Тип данных	Разрешить з
 SQL-MS (SQL Server 9.0.5000 - studentMBS) Базы данных 	:1)	•		

- Рисунок 6. Создание таблицы
- Вводим имя первого столбца Num (первичный ключ в том столбце хранится номер записи), выбираем из выпадающего списка тип данных int. Первичный ключ не может быть пустым, поэтому и оставляем неотмеченным поле «Разрешить

значения null». Затем аналогичным образом вводим имя второго столбца, задаем тип, запрещаем полю иметь значение null. Таблица принимает следующий вид:

1	Габлица - dbo.Table_	1* Сводка	
	Имя столбца	Тип данных	Разрешить значения null
	Num	int	
•	Name	varchar(60)	

- Рисунок 7.
- Теперь необходимо указать, что поле Num будет являться первичным ключом.
 Правой кнопкой мыши щелкаем по этому полю и выбираем «Задать первичный ключ»:

аблица - dbo.Table_1*	Сводка		
Имя столбца	Тип данных	Разрешить знач	чения null
Num	int		
Name	varchar(60)		у Задать первичный ключ
		[FT]	Вставить столбец
		10	🆞 Удалить столбец
			- OTVOLUCIUM

- Рисунок 8.
- Сохраняем таблицу под именем Speciality (после этого таблица должна появиться в обозревателе объектов). Теперь можно перейти к заполнению этой таблицы (для этого нужно в обозревателе объектов выбрать эту таблицу и в контекстном меню нажать «Открыть таблицу»):

14	олица - uno.:	speciality Taolinga - upo. Speciality	Сводка
	Num	Name	
	1	Математика	
	2	Информатика	
	3	Физика	
▶*	NULL	NULL	

- Рисунок 9.
- При заполнении вы обнаружите, что каждый раз приходится вводить не только полезную информацию (название специальности), но и номер записи. Чтобы вводить номер записи автоматически, нужно задать спецификацию идентифицирующего столбца. Для этого необходимо в свойствах столбца указать, что данный столбец является идентифицирующим (рис. 10):

	Table_1*	Сводк	
Иня столбца	Тип данных Р	Разрешит	
Num	int		
Свойства столбцов]		
Не для репликаца	91	Нет	
Описание			
Paswep		4	
С публикацией DTS		Нет	
С публикацией объединениен		Her	
С репликацией		Het	
Сжатый тип данных		int	
П Спецификация вы	мисляеного столбца		
П Спецификация ид	ентифицирующего столбца	Да	
(Является иде	нтифицирующим столбцом)	Да	
Начальное зна	чение	1	
Прирашение		1	
П Спецификация полнотекстового столбца		Her	
П Спецификация по	the reaction of the second s		

Рисунок 10. Определение свойств идентифицирующего столбца

Создание таблиц базы данных с помощью SQL-запроса

- Создание таблиц в графическом режиме, безусловно, удобно, однако не универсально. При использовании других средств разработки баз данных (например, IBM DB2) придется привыкать к новым приемам работы. Использование конструкций языка SQL позволяет работать с базами данных, исходя из единого подхода, в любой среде управления базами данных.
- Выберите на панели инструментов «Создать запрос»:

🍢 Сре,	ga Micro	soft S	QL Serv
<u>Ф</u> айл	Правка	Вид	Проект
🔁 Co3	дать запр	ooc 🛛 🗋	1
8	21	Изме	нить тип

Рисунок 11.

- Создадим новую базу данных запросом. Напишем
- CREATE DATABASE mbs21_query
- и нажмем F5. В обозревателе объектов должна появиться новая база (если сразу не появилась, то надо выделить мышью раздел «Базы данных» и в контекстном меню выбрать «Обновить»).
- Теперь создадим таблицу Speciality. Упрощенный синтаксис создания таблиц следующий:

- CREATE TABLE <имя таблицы> (
- <имя столбца 1> <тип данных> [NOT NULL] [DEFAULT <значение по умолчанию>],
- <имя столбца 2> <тип данных> [NOT NULL] [DEFAULT <значение по умолчанию>],
- -
-)
- Введем новый запрос:
- /* создание таблицы Специальность*/
- USE mbs21_query
 -- определяем базу данных, в которую входит таблица
- CREATE TABLE Speciality(
- Num INT IDENTITY(1,1) PRIMARY KEY NOT NULL, -- первичный ключ
- NameSpec VARCHAR(60) -- название специальности
-)
- В обозревателе объектов видим, что таблица действительно создана. Файл с SQLзапросом сохраняем в своей папке (в конце работы необходимо показать запросы, которые были выполнены, преподавателю). Слово IDENTITY(1,1) добавлено, чтобы поле первичного ключа Num автоматически нумеровалось начиная с единицы (фактически, эта конструкция определяет спецификацию идентифицирующего столбца).
- Таким же образом необходимо создать остальные таблицы. Рассмотрим таблицу Course.
- Таблица **Course** (курс)

Имя поля (столбца)	Содержание	Тип данных	Возможность содержать NULL
Num	Первичный ключ	int	нет
Name	Название специальности	varchar(60)	нет
YearEntry	Год поступления Год	int	нет
YearFinal	выпуска	int	да
Speciality	Специальность (внешний ключ ссылается на первичный ключ таблицы Speciality)	int	нет

- Эта таблица содержит поле *Speciality*, которое ссылается на первичный ключ таблицы Speciality. Чтобы создать такую таблицу, необходимо выполнить запрос:
- /* создание таблицы Курс */
- USE mbs21_query
 -- определяем базу данных, в которую входит таблица
- CREATE TABLE Course(
- Num INT IDENTITY(1,1) PRIMARY KEY NOT NULL, -- первичный ключ
- YearEntry INT NOT NULL, -- год поступления
- YearFinal INT, -- год окончания
- Speciality INT FOREIGN KEY REFERENCES Speciality(Num) -- специальность,
- -- ссылка по внешнему ключу на поле Num таблицы Speciality
-)
- Примечание. Ссылку можно создать только на существующую таблицу. Задать ссылку по внешнему ключу можно и после создания таблицы (подробно будет рассмотрено в следующей лабораторной работе).
- Задание. Создайте все остальные таблицы, указанные в Приложении, используя SQL – запросы.

Перейдем к созданию статических запросов. В обозревателе объектов "Microsoft SQL Server 2008" все запросы БД находятся в папке "Views" (рис. 8.1).



Рис. 8.1.

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" щелкнуть 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. 8.2).

Add Tab	le				? 🗙
Tables	Views	Functions	Synonyms		
Оценки Предме	ты				
Специа Студен	льности іты				
			Refresh	Add	Close
		_			

Рис. 8.2.

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. 8.3).

A Marager SC, Server Management Sta The Edit Year Project Debug QueryD Query Duers (2) (2) (2) (2) (2) (2) (2) (2) (2) (2) (2) (2) (2) (2) (2) (2)	dia nagrat Init - 1	Tues tinds	in Canna	nity indu													1
Report Encoderer Connect + Connect +	2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	 ЗАВОСТНО- ФОДСАНИИ: Карстранина тото Пол Дата роказина тото Пол Дата роказина Паслартные Телефон Телефон Телефон Телефон Телефон Телефон Паслартные Дата поступе Группе Корк Корк Корк Сода стецияние Сода	- 663 Xes 		/		- * (0) - * (0) - * (0) - * (0) - * (0) - * (0) - * (0)	od (Lindo) Olama) Majakato Ta robove (Pajaka	9 9079 9079						,	1	
# Call Security	1.4	Course defines a	NOT MENT	_												. *	2
ini ini Server Cityerts		Calum			Alex.	Table		Output Se	tTipe	Sort Circles	Filer	Or	÷0		(Dim	1	_
in Ca Hanagement		910				Стурента		8									
To advise when index in any end of		flait	11263			Crabevia	÷	8								/	s0
		QSatra perka	mere)			Chadwith		105							10	/	-
		PERMIT		-		Chiperty		100									3
		Addec		-		Cristiants		10									-
	ROM	dio.Cruan- dio.Cruan- dio.Crua dio.Cruan- dio.Cruan- dio.Cruan- dio.Cruan-	пы. 640, da рипы. (Тасп рипы. Курс, ратиности.) иты (1868.) ратиности.)	а. Студенты артныя для- dos. Студен Отысления о ОТы ОТы ОТы ОТы ОТы ОТы ОТы ОТы ОТы ОТы	non, d non, d nu jos nu jos nu jos nu jos	ID. CTy ammunities ID. CT	(Anta per Posiepi sa Interne), r	annel, de C annel, de C le Creanne	студанты осто ра	- Радиталь, фо. С (Дата постронно опосебност (пеца а очидимисти)	Praema Agos es], dia Cryaen nineechij,	dbo Студенты Тылы ы Группа.	pon.			/	4
		640	The	Data pos	-	Piarretu	Apec	Tanaĝor	flag	THE RATES	Harep send har	Дата поступления	Terra	Rype	O-man d	cou de f	-
		Distance and the	Hystocali	1985-12-0	1	Oning Matte	Note	+745170916	14 8187	1067040	12245	3007-09-01	149114	. 6	Thue	/	
	1	P	wanted			1910	****	- have there	×	e merer :	1100.0	100 A 10 10	deres .	4			
STATES AND		100			100	Read Pro-						and the second second					
•	111	1/4 1		1.	Car	warme Centr.	_		_							1	
Ready	-	The second s	_	_		_		_		_			_	_	-	-	
🐮 прок 🖉 🗖 С. 🧐 📰 – Кана	100010-000	Serrer													- W. 4	O-mail 1	

Рис. 8.3.

SQL

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 щелкнуть **ПКМ** и в появившемся меню выбрать пункт "**Remove**" (Удалить).

14

Теперь перейдем 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. 8.3).

Замечание: Если необходимо удалить связь, то для этого необходимо щелкнуть 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 щелкнуть 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 щелкните 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 рис. 8.3 (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

2

Либо щелкните **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. 8.3).

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 ее необходимо удалить и создать заново, как это описано выше.

Если запрос выполняется правильно, то необходимо сохранить. Для сохранения запроса закройте окно конструктора запросов, щелкнув мышью по кнопке закрытия

×

расположенной в верхнем правом углу окна конструктора (над схемой данных). Появится окно с вопросом о сохранении запроса (рис. 8.4).

Microsoft SQL Server Management Studio	? 🔀
Save changes to the following items?	
ACER-3AD5E49515.Students - dbo.View_1	
Yes No	Cancel

Рис. 8.4.

В данном окне необходимо нажать кнопку "Yes" (Да). Появится окно "Choose Name" (Выберите имя) (<u>рис. 8.5</u>).

ти	
ОК Са	ncel
-	ок Са

Рис. 8.5.

В данном окне зададим имя нового запроса "Запрос Студенты+Специальности" и нажмем кнопку "Ok". Запрос появится в папке "Views" БД "Students" в обозревателе объектов (рис. 8.6).

Microsoft SQL Server Management S	tudio												0
File Edit View Query Project Debug	Teols Wind	on Community	Heb										
2 mm Query Di 12 13 13 Di 14	G (4 14												
10 922 waster + 1	Execute #	× 23 m	10 2 4	1000	12.0	BR AL							
Object Explorer . 3 :	K 501.0wm	ry1.aul_à lives	om (513)		A A	ALL CALLS							* X
Correct - 20 20 2 2 3 3 Correct - 20 20 2 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5	g = 5212	Загарь (П тор 100 (Пал) (Дата р (Родите, Адрес) (Тальфо (Пальфо (Пальфо) (Дата п (Группа (Куро) (Очкая) (Сикаа) (Сикаа) (Сикаа)	for Sel for Sel (scales) (scal	ест?оряйска жие; жи; учежия; бешиль вости ил кости; ; (Seppo Ct	Commania I G Yanestw-Cr	208.3388	PW]						
e 🖬 Security											-		
ili 🥁 Server Objects	The first	I Do Marrowski											
ix 🔤 Replication	40	Cod conserves	Dee	Data sciences	Pranear	Liner	Telefon	Damonical second	Hissen sealings	flama recommission	Inera	Keel 6	Owners dermos
Management R SOL Server Acent (Acent XPs deable)	1 1 1000	NAson	Myxciell	1983-12-12	Отец Мать	Москва	+74957295674	0562-567543	13245	2007-29-01	MMIT	1	1
The second second second second	2 Det	DOBB M.M.	Xeroal	1962-11-01	Мать	Mockee	+74957009076	4567-765432	34563	2006-08-01	10421	2	0
	3 Ma	A.M en	Myxceek	1982-05-14	Oneu	Санара	+76462875690	\$438-098787	86732	2006-07-05	CT22	2	0
	4 Gra	X 8 escon	Xerced	1961-09-27	Het	Capaton.	+79027068909	1287-967505	27543	2005-06-23	M021	1	1
	5 Kcx	AA sources	Мужской	1987-04-12	Marte	Kasara	+79168563476	2312-675468	34217	2005-07-21	EY33	3	t
	6 fler	3.H secreted	Ханский	1983-09-02	One, Mare	Verederox	+74569098723	8743-056700	43278	2007-08-01	IMM12	1	0
	7 Liai	eropogues E.B.	Moxpet	1905-02-17	One	Санара	+70462234769	6543-034521	43795	2008-07-04	Closel	4	E.
	8 5ac	avosa 7 8.	Xeroud	1980-07-09	OneL Mate	Hedexcape	+79027874638	2133-896567	10387	2004-08-09	CT42	4	0
	3 Ter		Muncherk	1975-02-26	Her	Katara	+79067453678	2753-634904	67348	2003-07-22	MO51	5	t -
	10 Hare	ланая А.П.	Хеновё	1979-03-17	Мать	Caparce	+78546456432	3256-490532	45297	2003-06-21	67/53	5	0
	2												
	(2 Danse	secure before	hille:		-		1	ACTE-3404E49515 (10.1	ACTEN ACTENTIA	DEFECTION OF	master	00-00-00	10 room
Rents	Contra la							Contraction of the last	in t	Col 1	01		The C
	and the second	SHOT							014			COLOR OF	0

Рис. 8.6.

Проверим работоспособность созданного запроса вне конструктора запросов. Запустим вновь созданный запрос **"Запрос Студенты+Специальности"** без использования конструктора запросов. Для выполнения уже сохраненного запроса необходимо щелкнуть **ПКМ** по запросу и в появившемся меню выбрать пункт **"Select top 1000 rows"** (Отобразить первые 1000 записей). Выполните эту операцию для запроса **"Запрос Студенты+Специальности"**. Результат представлен на <u>рис. 8.6.</u>

Перейдем к созданию запроса "Запрос Студенты+Оценки". В обозревателе объектов в БД "Students" щелкните ПКМ по папке "Views", затем в появившемся меню выберите пункт "New View". Появиться окно "Add Table" (<u>рис. 8.2</u>).

В запросе "Запрос Студенты+Оценки" мы связываем таблицы "Студенты" и "Оценки" по полям связи "Код студента". Следовательно, в окне "Add Table" в новый запрос добавляем таблицы "Студенты" и "Оценки". Более того, в данном запросе таблица "Оценки" связывается с таблицей "Предметы" не по одному полю, а по трем полям. То есть поля "Код предмета 1", "Код предмета 2" и "Код предмета 3" таблицы "Оценки" связаны с полем "Код предмета" таблицы "Предметы". Поэтому добавим в запрос три экземпляра таблицы "Предметы" (по одному экземпляру для каждого поля связи таблицы оценки). В итоге в запросе должны участвовать таблицы "Студенты", "Оценки" и три экземпляра таблицы "Предметы" (в запросе они будут называться "Предметы", "Предметы_1" и "Предметы_2"). После добавления таблиц закройте окно "Add Table", появится окно конструктора запросов.

В окне конструктора запросов установите связи между таблицами и определите отображаемые поля, как показано на <u>рис. 8.7.</u>

17

e Microsoft SQL Server Management S	tudio												2 0 X
File Edit Vew Project Debug Query	Designer	Tools Winds	e Community H	Wip .									and the second
2 mm Query (3) (3) (3) (3) (3) (3)		14 .											
alaisis) : + a ta ta /s .													
Object Explorer + 3 3	AL AC	IR-JADSE495	dbo.View 1*										
Correct - 1 1 = 7 3								in the second se	a management				
Correct 4 4 5 = 7 3 Correct 5 5 5 (2), Server 10.0.1442 Distabases Distabas	8 A	Сурантия « (и Солити) Как стурного оно Она Она Она Она Она Она Она Она	a ana ana ana drama		Currents All Columns All Column Kay creater Courses 1 Courses 1 Courses 1 Courses 2 Courses 2 Courses 3 Courses 3 Courses 3 Courses 3 Courses 3 Courses 3 Courses 3	4) te me 1 ta 1 ta 2 ta 2 ta 3 ta 3			Column) reputeria existence operation existence operation reputeria Column) repateria existence operation and topological column) repateria repateria repateria	pera			
· Ca Replication		Column		Ales	Table	Durbut	SetTro	Sort On	der Fil	er or		- 0'-	
 Management SOL Server Asent (Asent VPs deabled) 		0110			Стуренты								
100		(дата жове	e+a 12		Outwor	1							
		[Dutres 1]			Outrear	23							
		(Date How	ena 23		0.200	2							
		Dama 2	S - OSconsure	terret	Durrent.	171				111000000000000000000000000000000000000			
	ROM	бо.Студен бо.Оце Преднет фо.Предн	ты: 6ИО, dbo. Оцен ни: [Дата жаанон и_1.]Эзэлтонобани гты: 29462 2021	ux. [Дата экз 3], dtó. Оце г преднета] /	anena 1), dbo.0 nun (Ouenia 3), Ki Expr1, Ripean	dba.Ou etw_2.	Ouerria I), erria (Cpe) (Harrieros)	dbo. Ouzeroi (S) sriaž Sant], dbo. sriat npeziretaj /	па жанена 3 Греднети (на Кобърга Кобърга), do.Ouevol.(Ou interobarine rocum	क्षण्ड हो. बाबो.		0.0
		0.10	Дета жанно а	Ouriera 1	Дата экзано-	02 0	Dames 2	Дата жанена	1 Oureal	Средний балл	Наменование	. bp1	Expr2
		Howen AM.	2008-02-01	5	2008-02-08	1		2005-02-14		4	Orepayore or	. Reentroppin.	Ob-On-R TakeTa
	10	Dempolea M.M.	2008-01-30	1 A	3008-02-23	3		2008-02-27	5	0	Проектирования		Orepaperer 1
		Mpage N.A.	2008-01-26	3	2008-02-05	3		2008-03-18	2		Same garmen	Orepapore.	Проектировани
	115	11		(R) Call	Read Only.								
	STR.	10.72 3	12 12 12 12	Citer Create	100000000								
Ready	_	_		_			_		_				1000
📲 луск: 🖉 с 🗢 🔟 🗛	kcroseft 56	s Server											AU (\$ 10.00

Рис. 8.7.

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 таблице отображаемых полей как показано на рис. 8.8.

	Column	Alies		Table	Output	Sort Type	Sort Order	Filter	Or Or.	Or	**		
•	@#10	[ФИЮ СТУДИ	нта]	Студенты	1								
	Дата жанени	а 1] [Дата пере	ого экзанена]	Outreal	1								
	Наченование	преднета] (Наменова	ние преднета переого экзанена]	Преднеты	1								
	[Ouevea 1]	(Оценка пе	реого экзанена]	Outreat	1								
	[Дата экзанени	в 2] [Дата втор	ого экзанена]	Outrest									
	Наченование	преднета] (Налненова	ние преднета второго экзанена]	Преднеты_1	1								
	[Оценка 2]	(Оценка вт	орого экзанена]	Ouewor	2								
	Дата жалени	а 3] [Дата трет	ьего экзанена]	Outreat	2								
	Паленование	преднета] [Наименова	ние преднета третьего экзанена]	Преднеты_2	2								
	[Oue-Ka 3]	[Оценка тр	етьего жзанена]	Outrest	1								
	(Средний балл	[Средний б	алл студнта за сесоно]	Outreport	2								
		11 11			1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1								
SELECT	dbo.Студенты.	0/10 A5 (0/10 студента).	dbo. Оценкон. [Дата экзанена 1] Аб	(Дата nepeoro s	(in the second s	1. 1. 193	130523		- V2	- 11 S			~
SELECT	dbo. Студенты dbo. Предне dbo. Оценки Преднеты dbo. Оценки Преднеты	 (втинеское студента) изделение соболение (1997) (Дата изделение соболение) (Дата изделение соболение (Дата изделение соболение соболение соболение соболение (Дата изделение соболение соболение соболение соболение соболение соболение соболение соболение соболение соболение соболение соболение соболение соболение соболение	dbo.Oueven.[Дата жаленна 1] AS cra] AS [Накен-кванке преднета с та второго жаленна], a] AS [Накен-кованке преднета вт та третьего жаленна], a] AS [Накен-кованке преднета тр	(Дата переого з ереого экзанена) арого экзанена), етьего экзанена	ecamena) a], dos Ou , dos Oue], dos Oue	arean (Outre ean (Outrea trean (Outrea	a 1) AS (Oue 2) AS (Oue++ a 3) AS (Oue++	жа перво в второго ка треты	го эсзанена) і экзанена), гго экзанена	2			~ ~
SELECT	фо.Студенты, do.Предне do.Ouewor Преднеты, do.Ouewor Преднеты, 0ИО студента	 сио 45 (640 студента), ты, Нажненование предн (Дата экзаненование преднет Дата экзаненование преднет Дата экзаненование преднет Дата первого экзанена	dbo.Ouerrow.[Дата экзанена 1] AS cra] AS [Накненование преднета г та второго экзанена], a] AS [Накненование преднета вт та третьего жизнена], a] AS [Накненование преднета пр Накненование преднета пр	(Дата переого з ереого экзанена) орого экзанена) пъего экзанена) Оценка переог	(300-04 a), doo.04 doo.04 doo.04 a), doo.04 a), doo.04 a), doo.04	инол. (Оцена нол. (Оцена нол. (Оцена пнол. (Оцена Дата втор	oa 1) AS (Ouer 2) AS (Ouereo a 3) AS (Ouereo aro socianeea	ка перво а второго ка треты Начне	го жанена) жанена), го жанена нование пре	.)]. днета вт	ro	Оценка втор	Der o
SELECT	фо.Студенты фо.Предне фо.Оценко Преднеты фо.Оценко Преднеты ФИО студента Изанов А.И.	ско А5 (640 студента), ти, Накненование предн (дата экзаненование предн (дата экзаненование преднет (дата экзаненование преднет Дата экзаненование преднет Дата первого экзанена 2008-02-01	dbo.Ouewor.[Дата жаанена 1] AS cra] AS [Наиненование преднета г та второго жаанена], a] AS [Наиненование преднета вт та третьего жаанена], a] AS [Наиненование преднета тр Наиненование преднета пер Операционые окстены	[Дата переого з ереого экзанена орого экзанена], пъего экзанена Оценка переог 5	() () () () () () () () () () () () () (аннон (Оцанн нон (Оцанна нион (Оцанна Дата втор 2008-02-09	oa 1) AS (Ouen 2) AS (Ouenno 2) AS (Ouenno 3) AS (Ouenno 3) AS (Ouenno 30 antianno 30 antianno 30 antianno 30 antianno 31 antiannno 31 antianno 31 antiannno 31 antianno 31 an	жа перво в второго ка треты Начее Языки	го экзанена) і экзанена), го экзанена нование пре	,], дмета вт мания	ro	Оценка втор 2	Дет 0
SELECT	фо. Студенты фо. Предне фо. Оценко Преднеты фо. Оценко Преднеты ФИО студента Изанов А.И. Петрова И.И.	оно А5 (оно студента), ти, Разненование предн (дата экзаниенование предне (дата экзаниенование преднет Дата экзаниенова) А5 (да 2014). 2014 оказание преднет Дата первого экзаниена 2008-02-01 2008-01-30	dbo.Ouewor.[Дата жаанена 1] AS cra] AS [Накненование преднета г та второго жаанена], a) AS [Накненование преднета вто та третьего жаанена], a) AS [Накненование преднета тре Накненование преднета пер Операционые окстены Проектирование информацио	[Дата переого з ереого жзанена], прого жзанена], пъего жзанена] Оценка переог 5 4	() () () () () () () () () ()	иннол. [Оцинн нол. [Оцинна сол. [Оцинна перал. [Оцинна Дата вторя 2008-02-09 2008-02-23	oa 1] AS (Ouen 2] AS (Ouenno a 3] AS (Ouenno a 3] AS (Ouenno aro aksamena	ка перво а второго ка треты Наите Языки Бази д	го экзанена) го экзанена), по экзанена нование пре програннор анных	,], днета вт изания	ro	Оценка втор 2 5	Aer 0
SELECT	Фо. Студенты do. Предне do. Оценко Преднеты do. Оценко Преднеты GHO студента ИЗанов А.И. Петрова И.И. Музон М.А.	Сию А5 (бИО студента), ти, Разненование предне (Дата экзаненование преднет (Дата экзанена 2) А5 (д. 2) Разненование преднет Дата экзанена 3) А5 (д. 2) Разненование преднет Дата первого экзанена 2008-02-01 2008-01-05 2008-01-25	do. Оценки. [Дата жланена 1] AS tra] AS [Накненование преднета г та второто зкланена], а) AS [Накненование преднета вти та третьего жланена], AS [Накненование преднета пер Операционые октень Гроектирование лифорнацио Базы Данных	(Дата переого з ереого экзанена), арого экзанена), стьего экзанена) Оценка переог 5 4 5	(300-04 (30	стол. (Оценка нол. (Оценка Сата втор 2008-02-09 2008-02-23 2008-02-05	oa 1] AS (Ouen 2] AS (Ouenn a 3] AS (Ouenn a 3] AS (Ouenn ro ansanena	ка перво в второго ка треты Нанне Языки Базы д Опера	го жаанена) і жаанена), го жаанена), посание пре програнног деньк дногые окт	,], 2010-10 BT 2010-1018	ro	Оценка втор 2. 5. 3.	Дет 2001 2001
SELECT	фо. Студенты фо. Предне фо. Оценки преднеты фо. Оценки преднеты ФИО студента Разнов А.И. Петрова И.И. Музон М.А. Спророва В.К.	сию А5 (бИО студента), ти, (Наненование предн (Дата экзаненование предн (Дата экзаненование преднет (Дата экзаненование преднет Дата экзаненование преднет Дата акзаненование преднет Дата акзаненование преднет 2008-02-01 2008-01-05 2008-01-25 2007-12-26	dbo.Outereor.[Дата экзанена 1] AS cra] AS [Наиненование преднета т та второго экзанена], a) AS [Наиненование преднета эти та третьего экзанена], a) AS [Наиненование преднета тр Наиненование преднета пер Операционые окстены Проектирование информацио Базы данных. Офновые пакаты	(Дата первого з ереско жизнена) орого жизнена), стыего жизнена) Ощенка первол 5 4 5 3	(), dbo.Ou (), dbo.Ou (), dbo.Ou (), dbo.Ou	атнол. (Оценка нол. (Оценка Дата втор 2008-02-09 2008-02-09 2008-02-05 2008-02-05 2008-02-05	os I) AS (Ouen 2] AS (Ouens a 3] AS (Ouens ro skisanena	ка перео а второго ка треты Наиче Язьки Базы д Опера Язько	го экзанена), го экзанена), го экзанена нование пре програнног анных ционые окти), Ameta at mai states	ro	Оценка втор 2 5 3 4	Дет 2001 2005 2006

увеличить изображение

Рис. 8.8.

Задайте псевдонимы для каждого из полей, просто записав псевдонимы в столбце "Alias" таблицы отображаемых полей, как на <u>puc. 8.8.</u>

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" (<u>рис. 8.9</u>).



Рис. 8.9.

Проверьте работоспособность нового запроса вне конструктора. Для этого запустите запрос. Результат выполнения запроса "Запрос Студенты+Оценки" должен выглядеть как на рис. 8.10.

THE COURT NO. Server Baragement Stad						85	100
the dat lies dury front Doing	Yesh Mindes Dermants	198	and the second			1	-
Lesson in the in it was	Hittin A						
to tal loaner	and A. R. v. C. of	01119 D/01	0.22.9.9.4.				
Gerifater + F K	NA German and	00 (Nal)					
fament 1 (5) (5) (5) (7) (5) (5) (5) (5) (5) (5) (5) (5) (5) (5	Dennis Class. In Class. In Class. In Class.	но мараттери 1.00. полнена кала далинно кала далинно полне далинно полне далинно полне далинно полне далинно полнен д	neg command from Table torontal seguring: Holonomal si entiplice: Holonomal si entiplice: Holonomal si comment sciences; sc				1
 Eal Programshifty Eal Terroral Potential Eal Terroral Potential Eal Terroral Eal Terroral 							
a Californi Marco	The sector of messages	Photo and a state	and the second se		The second second		
a Calification	1 Basels A.R.	2006-02-09	Conception of the large of the	A REAL PROPERTY AND ADDRESS.	200505.04	Read and the second second	-
7, 93, Server spect object of a manual	2 fammen H.H.	2008-011-000	Parameters and an and an and an and		10000000	Strin strengt	
	1 House M.S.	2008-01-26	fatte percent	4	2008-02-05	Oraceptour actives	
	A Greater S.K.	2011 12 28	Gentral revenue	1	2006/01-08	Playie too tan separate	
	5 Scenesce A.A.	2006-03-02	Place porparing and	4	2008-05-16	Tenerophanee antenna	ind
	4 falaanserik	2961/01/7	General	*	2007.52.26	Phase receiptions	100
	7 UperStrate LB	2006-0224	Figure and a state of the state	1.	3999-00.25	Orectanovae sectoria	
	3 50000178	2008 (2.53	Epis aprila	1	2000.0210	formation and an and a second	(mag
	R Payment !!	2008-01-25	Creek, much increase		3969-02.02	Date Amount	- 1
	10 months 1/1	200-1248	Place destance destance		200401-01	Contraction decision	- 1
	4						18
4 A	Conty matured automat	ww		ACER INDIREMENTS (30.0 C	NUMBER ADA	University - Hantler (00.00100) 48 m	-0
August .					915 - 2	ali dek 1	-
Trans. C. 9 B King	addition of the					10 B I	. 11

увеличить изображение

Рис. 8.10.

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. 8.11). Затем закройте окно "Add Table".



Рис. 8.11.

В появившемся окне конструктора запросов определите в качестве отображаемых полей все поля запроса "Запрос Студенты+Специальности" (рис. 8.12).





Рис. 8.12.

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. 8.12).

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 Type**". О**гder**" (Порядок сортировки) поставьте 1, а для поля "**Курс**" поставьте 2 (<u>рис. 8.12</u>). 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

₹↓

обозначают сортировку по возрастанию и убыванию, а значок

Y

показывает наличие условия отбора.

После установки сортировки записей в фильтре проверим его работоспособность, выполнив его. Результат выполнения фильтра должен выглядеть как на <u>рис. 8.12.</u> Закройте окно конструктора запросов. В качестве имени нового фильтра в окне "Choose Name" задайте "Фильтр MM" (<u>рис. 8.13</u>) и нажмите кнопку "Ok".

Choose Name		X
Enter a name for th Фильтр MM	e view:	
	ОК	Cancel

Рис. 8.13.

Фильтр **"Фильтр ММ"** появится в обозревателе объектов. Выполните созданный фильтр вне окна конструктора запросов. Результат должен быть таким же как на <u>рис. 8.14.</u>





Рис. 8.14.

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 задаем 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. 8.11). После закрытия окна "Add Table" сделайте отображаемыми все поля запроса (рис. 8.15).

Re Microsoft SQL Server Management Sto	dia											E 18 🛛	3
File Solt Vew Project Debug Query De	NO.	Toola Window	Community Help										1
A Mer Quers Di ES ES ES Di Di ES	a	14 .											
et alester + es as *3/8													
Object Explorer - 3 X	- 40	8-1405F495	Maria 1º / 1/1	here's set is there	4.040							* *	a
Covert . 1 3/ = 7 3		Canal Section 1											à.
ACRIV.34058-09115 (IQ), Berver 20.6, 1443 Databases Acris.34058-09115 (IQ), Berver 20.6, 1443 Databases Sequences Acris.34058-09115 (IQ), Berver 20.6, 1443 Acris.34058 Sequences Acris.34058 Sequenc		" (Al Column) " (Al Column) " (Al Column) " (an e poradree " Alare " Alare " Alare " Alare " Alare " Tonego sector " Tonego " T	а аныя ная очна обучаная отрадочносто	*									
A C dec. Remarks DA a C dec. Remarks C f	-	Orscasse crea	and the octor	_									
a (do.boxrp10)	97.0	Colore	Alas Talas			Come.	Sert Tube	East Oxfor	10w 0		0	1.1	
Cal Synanume		Paul	Darpoc	Crype-rui+Cresaian	ward	64	dest citize		1.000				
a Programmability		(data powar	Derpor	Cryaminu + Cression	worrd	123							
* Call Storage		Pedurteno	Darpic	Crysleite (Creupon	(ursona	8			- '0+m4'				
in Cal Security		Apper	(lienpoc	Студенты Юпециил	emocted	12							
a Server Objects		Телефон	[larpoc	стаденты нОтецики	wette	[7]							1
III Cal Replication III Cal Hendgement SIG. Server Agent (Agent XPs deabled)	ARION VINERA	BND, Nov. Joan (Oweat Book dbs. Danper Cr (Podurtecki + Dr	а рокрения), Родит а обучения), (Нание уденты «Специалы» ец()	ель, Адрес, Тялефо новаже специельно юстн)	en () jon	Carve a	uemeanneur) seres): [aoimh m	evention), [Darta roct)	noeeee), Coyrne, Ky	pc.			
		0140	Ten	Датарондачна	Popure	Pet.	Appel	Terredoin	Пистортные д.	. Honey severings	Дета поступи	(Dyrma	
		Hysen H.A.	Myreceal .	1982-05-24	Oreu		Санара	+76463875690	\$438-098787	16752	2006-07-05	C722	
		Lisberopodume	Monapal.	1980-02-37	Orei		Cavagia	+78462234268	6543-034921	43765	2004-07-04	(1944)	
		MAL	ALL	ALLI	MAL		MAI	1661	A641	NUG.	1663	NAI.	
	-								-				
	1.	11 42	P. N. P. 100										2
Ready			-			-	_				_	_	÷,
Илуск С. С. С. В. П. На	orașin fere	Liseyer									10	(C) 🛛 12.07	



Рис. 8.15.

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 <u>рис. 8.15</u>.

Закройте окно конструктора запросов. В окне "Choose Name" задайте имя нового фильтра как "Фильтр Отец" (рис. 8.16).

Choose Name		
Enter a name for the	view:	
Фильтр Отец		
	ОК	Carcel

Рис. 8.16.

Выполните фильтр **"Фильтр Отец"** вне конструктора запросов. Результат должен быть аналогичен рис. 8.17.





Рис. 8.17.

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 задаем 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. Начнем с очной формы обучения. Создайте новый запрос и добавьте в него запрос **"Запрос Студенты+Специальности"**. Как и ранее сделайте все поля запроса отображаемыми (<u>рис. 8.18</u>).

Microsoft SQL Server Masagement Stu	die		124												C 10	1
Pile Billt Very Project Debug Query De	ng n	Tools Window	Communit	r. Help												
Then Query Dr. 15 13 23 23 24	1.1	ile a														
Object Explorer + 3 ×	MI	R-34056495.	dio.Verw_1	1												×
Connect - 3/ 3: - 7 5		Concernation in the	-	Contraction of the												1
Dividurer Exclusion Exclusion		 (M Column (M Column (M Column (M Column (M Column 	он (1993) в данный тар Пария	11 V												
al all disc. oversing 191		[Allowersed a	retard/bring/													-
ai 👔 dbil dwarang HD	<u>.</u>														3	
a C dbo. Overanto EV		Column		Alas	Table	111-011-01		Output	SersType	Sert Onter	Pilter d	A	Qr	Otre		10
a D dbb. dvine to Marie		Partie -			Damping City	perty co	er transmission	100	Animation							
a doo Broherto Orinia o Matte		Deses doors	(and a second s		Darper Cry	dente «Ch	[hrmming	100	Pace ser		1.1					
Ca Synanytee		Planetosa	e cheuptanter	acrod (Danpos City	caternu +Ch	evanancial	R								
in Car Programmability		(Drucawa ch	-	1	(Berpot Chi	AB-TH-CT	[interesting	121								-
in Storege in Storege in Security in Security in Security	SELECT PROM UMERE ORDER	TOP (1993) PB (294796, K) (Be (394995) (Dream dope BT Kips)	ковніт фино, і урс, (Очник ф Студенть ні ф обучення)	ton, (Darta ponçan Norva oti yvatneni), (Vraturanuvotitiri) + 10	une), Pagurteno Plane - cedana	п, Адрес, Те стециальн	eredon, jTarros actul, (Drucan	1796-8 28 19 (719-84)	read), (Hor Marcoline)	ep sevenor), (j),	Hancer provene	ł.				
a Ca Management		440	fien	fare possesses	Paperene	Appen:	Terebos	flain	CEPTION	Nonep several	Дата поступ.	. Perria	Apr	Overe dam.	Harm	10
5QL Server Agent (Agent XPs deabled)		Distant AM	Hynoisk	1983-12-12	Disa y Hata	Moose	+7485.789567	4 8567	567543	13248	2007-09-01	MHUS		tice	MM	
		Caucopone B.H.	Hancosh	1881-05-27	1941	Capetile	+7902796890	9 1297	987508	27840	2005-06-23	99033	3	That	MD	
	*	Countration of the	Mysopoli	1981-0+12	Nata	1004	+7915858347	6 2212	475468	94217	2005-07-21	623	1	The .	87 3	1
e and a second sec	14 4	1 11	1 9 95	P. (82)												
Ready	-															
TE MYCK C C C C C C	un neo	Server 1												17,000	6.00	-1

увеличить изображение Рис. 8.18.

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 рис. 8.18.

Закройте окно конструктора запросов. Сохраните фильтр под именем "Фильтр очная форма обучения" (<u>рис. 8.19</u>).

Choose Name		
Enter a name for th	ne view:	
Фильтр очная фор	ма обучения	
	ОК	Cancel

Рис. 8.19.

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 рис. 8.20.

e Microsoft SQL Server Management Stud	lio													
Ne 58t View Query Project Debug 1	Tools	Window Community	Help											100.00
There Query Di B B B Di 10 10	13	14 .												
117 22 master . 1 Ex	eo.te	> = < 13 =	1 7 4	0000	12 R	R.A.								
Object Explorer - 3 ×	50	LQuery13.sq_# Sype	oe (52))											* X
Correct- 1 1 = 7 3	12	Present Bersen	for Jel	ectTopSEva	command fr	con 32N	******/							~
ACIR-SACK-MELLS (SQ, Server 10.0.1442 E Distbases E System Cetabases E Distbases E D		. Inoni .IAara se .IRaaras .IAagesi .IAagesi .IAaras .IAA .IAaras .IAA .IAA .IAA .IAA .IAA .IAA .IAA .IA	I LENOL IMAGENORE EL I I TOME CAN NAME CAN I COMMAND I COMMAND I COMMAND I COMMAND I COMMAND I COMMAND I COMMAND I LENOL	num] Hal Distantion Adaptor 1 (Switz o	1 mar Borna	stywaw	18]							
a Coloraro Ci														. *
# D dot. Swarp Creu	1	Reside 1 the Mainsoner												
a 🚺 do. Evitero Marte		410 G 141	Ten	Дата служаения	Poprary	Apec	Талерон	Decrement device	House Several	Dana reconstruction	forma	Kec 1	Owies de	ene d
a (1) do. ovinaro trer populare	1	Haavee A.H.	Мужений	1983-12-12	Oneu a Mara	Москве	+74957295674	8567-567543	13245	2007-09-01	MMT1	1	1	
· 👔 фо. фильтр очная форма (2	Grappose B.K.	Xerowit	1981-09-27	Het	Саранзе	+79027868909	1267-987509	27543	2005-06-23	M031	1	\$	
Synonyma	3	Kowenerge A.A.	Myxceal	1981-04-12	Маты	Kasana	+79168563476	2312-675468	34217	2005-07-21	EX33	3	1	
# Programmability	4	Laperopoauee E.B.	Мужрей	1360-02-17	Oneu	Санара	+78462234789	8543-834521	43765	2004-07-04	13/41	4	1	
Bonge Bonge Bonge Bonge Bonge Bonge Bonge Bonge Bongenent Bongenent Bongenent Bongenent	3	Depare D.C.	Мукони	1979-02-26	Her	Kasama	+79007453678	2765-634964	67341	2003-07-23	MOSI	5	1	
	s .													
s	30	very executed successf	ully,					ACER-3405049515 (10.0 CTP) ACER-	3405049515\Anexce.	maste	00:00	100 S rs	-
Ready									Ln 1	Ce 1	Ohi			105
📲 луск. 💼 С 🤏 🔤 🔩 нач	with Si	Q Server							500	ur unser-				2-0

увеличить изображение

Рис. 8.20.

Самостоятельно создайте фильтр для отображения студентов заочной формы обучения. Данный фильтр создается 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. 8.21):



Рис. 8.21.

Перейдем к созданию хранимых процедур. Для работы с хранимыми процедурами в обозревателе объектов необходимо выделить папку "**Programmability/Stored Procedures**" базы данных "**Students**" (рис. 10.1).



Рис. 10.1.

Создадим процедуру, вычисляющую среднее трех чисел. Для создания новой хранимой процедуры щелкните **ПКМ** по папке "Stored Procedures" (рис. 10.1) и в появившемся меню выберите пункт "New Stored Procedure". Появится окно кода новой хранимой процедуры (рис. 10.2).



Рис. 10.2.

Хранимая процедура имеет следующую структуру (рис. 10.2):

- 1. Область настройки параметров синтаксиса процедуры. Позволяет настраивать некоторые синтаксические правила, используемые при наборе кода процедуры. В нашем случае это:
 - оSET 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 зеленым 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 свернуты. Слева от раздела с комментариями будет стоять знак "+", щелкнув по которому можно развернуть комментарий.

Наберем код процедуры вычисляющей среднее трех чисел, как это показано на рис. 10.3.

```
SQLQuery3.sql ...й Бурков (53))*
```

```
F -- =
 SET ANSI NULLS ON
 GO
 SET QUOTED IDENTIFIER ON
 GO
-- ======
E CREATE PROCEDURE [Среднее трёх величин]-
      -- Add the parameters for the stored procedure here
     @Value1 Real=0,
     @Value2 Real=0,.
      @Value3 Real=0
 ΛS
 BEGIN
      -- SET NOCOUNT ON added to prevent extra result sets from ...
r‡
     SET NOCOUNT ON;
                                                                          3
      -- Insert statements for procedure here
     SELECT 'Cpeghee shavehue'=(@Value1+@Value2+@Value3)/3
 END
 GO
```

Рис. 10.3.

Рассмотрим код данной процедуры более подробно (рис. 10.3):

- 1. CREATE PROCEDURE [Среднее трех величин] определяет имя создаваемой процедуры как "Среднее трех 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. 10.2).

Для создания процедуры, выполним вышеописанный код, нажав кнопку

? Execute

(Выполнить) на панели инструментов. В нижней части окна с кодом появиться сообщение "Command(s) completed successfully.". Закройте окно с кодом, щелкнув 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 трех величин] 1, 7, 9 и нажмите кнопку

? Execute

на панели инструментов (рис. 10.4).



увеличить изображение

Рис. 10.4.

В нижней части окна с кодом появится результат выполнения новой хранимой процедуры: Среднее значение **5,66667** (рис. 10.4).

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 рис. 10.5.



увеличить изображение

Рис. 10.5.

Рассмотрим код процедуры **"Отображение студентов по ФИО"** более подробно (рис. 10.5):

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

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

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

```
? Execute
```

на панели инструментов (рис. 10.6).



увеличить изображение

Рис. 10.6.

В нижней части окна с кодом появиться результат выполнения хранимой процедуры "Отображение студентов по ФИО" (рис. 10.6).

Теперь перейдем к более сложной задаче - отобразить студентов, у которых средний балл выше заданного. Создайте новую хранимую процедуру и наберите код новой процедуры как на рис. 10.7.



Рис. 10.7.

Рассмотрим код процедуры "Отображение студентов по среднему баллу" более подробно (рис. 10.7):

- 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;
- 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 нем наберите команду EXEC [Отображение студентов по среднему баллу] 3.5 и выполните ее (Смотри выше) (рис. 10.8).

 Microsoft SQL Server Management Stud 								
Pile Edit Vew Query Project Debug 1	Teols	Window Communi	ty Help			1		and the second sec
These Query Dr 13 13 13 13 13 14	A	14 .						
1 22 Students . 1 Ex	eo./e	* * × 13 *	00000	O REAL.				
Object Explorer + 3 ×	50	Query2.sql A by	PROB (53))*					* X
Connect - 1 1 = 7 3		EXEC (Orofipan	when crypentos of	o openseiny Sannty] 3.5				-
ACBR 3400549515 (CQ, Server 30.0.1440 Databases Databases Databases Snaphots Database Snaphots Database Snaphots Societs Societs Societs Societs Tables Tables Societs Societs								
 Cetabase Triggers Description 	-							8
iii 🛄 Types	1	Health []] Meanag	PR					
ie 🛄 Rules		4910 ch28418	Дата пересло знавне	Наиненсевние преднета переого знавнена	Оценна переого экзанена	Дата второго знавне	Нанивновачи предне	Па второго знавнее
a Defaults	1	Freehoe A.M.	2008-02-01	Спераноные системы		2008-02-08	Новни програмнеров	pure.
Service Broker	15	retore in in	2000-01-00	poertyposave indepilations and the	2	2000-02-23	Data garras	100000000000000000000000000000000000000
* 🦲 Storage	1	ACKEDNINGE A.A.	20001112	Harry (Collinger Collins)	1	2002-01-18	1 DOEKS DOBRINE IN DO	CHRYSOMAL OCTAL
🕷 🎑 Security	1	Developer n.c.	20071617	Company and a second	;	2005.02.02	Frank reviser were	Para
 a Security a Security Bapticaton Bapticaton Management SQL Server Agent (Agent XPs dashled) 	×							2
< ×	30	very executed succes	afuly.		ACER-3405E49515 (10.0	CTP) ACER-SADSE-4951	SyAmerice Students (0:00:00 S rows
Ready	-					int	Cel 52 Ch 52	26
🧌 пуск 🛛 😅 🖉 😂 ге	1	1	National SQL Server			- idino		U R/ 1520



Рис. 10.8.

В нижней части окна с кодом появится результат выполнения хранимой процедуры "Отображение студентов по среднему баллу" (рис. 10.8).

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

Создадим новую хранимую процедуру и наберем код новой процедуры как представлено на рис. 10.9.



Рис. 10.9.

Рассмотрим код создаваемой процедуры "Отображение студентов по возрасту" более подробно (рис. 10.9):

- 1. CREATE PROCEDURE [Отображение студентов по возрасту] определяет имя создаваемой процедуры как "Отображение студентов по возрасту";
- 2. @Age int=0 определяют параметр процедуры **Age**. Параметру можно присвоить целые числа (Тип данных int), значения по умолчанию равны 0;
- 3. ФИО, [Запрос Студенты+Специальности].[Дата рождения],

'Возраст'=DATEDIFF(уу,[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 (уу), вычисленный исходя из его даты рождения и текущей даты (DATEDIFF(уу,[Запрос Студенты+Специальности].[Дата рождения], GETDATE())). Более того, выводятся студенты возраст которых больше определенного в параметре "Age" (DATEDIFF(уу,[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 нем наберем команду EXEC [Отображение студентов по возрасту] 26 и выполните ее. Должен появиться результат аналогичный результату, представленному на рис. 10.10.

St Microsoft SQL Server Management S	tudio											10 🕅
File Edit View Query Project Debug	Teol	s Window Commun	ity Help									
2 mm Query (3) (3) (3) (3) (3) (3)	1.	4 14 .										
1 2 2 Saderts . 1	Execut	* * = ~ ET -	10 2 4	0000	1 2 0 0	AC.						
Chest Explorer - 3 X	~	M Charger of and A Saver	ana (543)*	and the lines of a		1.040						
Connect - 3/ 3/ - 7 3 Connect - 3/ 3/ - 7 3 ACID-3A024-9513 (CQL Rever 33.0.1442 - 6C ACID-3A024-9513 (CQL Rever 33.0.1442 - 6C ACID-3A024 (CQ		DEC [Oridpase	nne crydesro	n no mospe	oty] 26							
Sustan Stored Production Sustan Stored Production Sustant Stored Production	-	Results () Messager	 Jara organization 	Report		4						3
in Types	1	Grappone E.K.	1981-09-27	27								
iii Defaults	2	Konstructs A.A.	1561-04-12	27								
🗰 🥁 Man Gudes	3	Lapertopoaire E.B.	1980-02-17	28								
in is Service Braker	4	Separces 7 8	1900-07-09	28								
in Ca Security	5	Depairs D.F.	1979-02-26	29								
 Sourty Sourty Replation Management SQL Server Agent (Agent NPs deabled) 	¢	Hoorees A.fl.	1979-03-17	29								
« (market a	3	Query executed execute	fully.				ACER-SADIE 40515 (10.0 CTP)	ACER-SADSE4	9515\Anexce	Students	00-00-00	6 rons
Ready								10.1	Col +4	0.4	1	145
📲 луск. 🖻 🗉 🖓 🛅 🖡 н	in the second	tisgi Server									RU (R.)	11-55

Рис. 10.10.

На этом мы заканчиваем описание хранимых процедур и переходим к рассмотрению пользовательских функций. В итоге, обозреватель объектов должен иметь вид как на рис. 10.11.



Рис. 10.11.

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. 12.1).



Рис. 12.1.

Начнем с создания скалярных пользовательских функций. Для создания новой скалярной пользовательской функции в обозревателе объектов, в БД "Students", в папке "Programmability", щелкните ПКМ по папке "Functions" и в появившемся меню выберите пункт "New/Scalar-valued Function". Появится окно новой скалярной пользовательской функции (рис. 12.2)



Рис. 12.2.

Синтаксис скалярной пользовательской функции похож на синтаксис хранимой процедуры (см. "Интерфейс информационных систем. Создание интерфейса пользователя"). Однако имеется ряд существенных отличий (рис. 12.2):

- 1. Область определения имени функции (Inline_Function_Name);
- 2. Параметры, передаваемые в процедуру (@**Param1**). Определение параметров аналогично определению параметров в хранимой процедуре (см. "Таблицы. Типы данных и свойства полей. Создание и заполнение таблиц");
- 3. Тип данных значения возвращаемого процедурой;
- Область объявления переменных, используемых внутри функции. Объявление переменных имеет следующий синтаксис: DECLARE @<Имя переменной> <Тип данных>
- 5. Тело самой пользовательской функции, содержит команды языка программирования запросов T-SQL;
- 6. 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 трех величин. В окне новой пользовательской функции наберите код представленный на рис. 12.3.



Рис. 12.3.

Рассмотрим более подробно код данной скалярной пользовательской функции (рис. 12.3):

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

Остальные фрагменты кода рассмотрены выше (рис. 12.2).

Для создания функции, выполним вышеописанный код, нажав кнопку

? Execute

(Выполнить) на панели инструментов. В нижней части окна с кодом появиться сообщение "Command(s) completed successfully.". Закройте окно с кодом, щелкнув 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 трех величин] (3, 5, 4) и нажмите кнопку

? Execute

на панели инструментов (рис. 12.4).



Рис. 12.4.

В нижней части окна с кодом появится результат выполнения новой скалярной пользовательской функции: **4** (рис. 12.4).

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. 12.5):


Рис. 12.5.

Перейдем к рассмотрению вышеприведенного кода (рис. 12.5). 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 длинной 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 переменныне @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 (уу год, mm месяц, dd день), "дата" это дата, части которой определяем.
- 6. IF @Month=12
- 7. BEGIN
- 8.SET @Month=19.SET @Year=@Year+1
- 10. END
- 11. ELSE

13.

- 12. BEGIN
 - SET @Month=@Month+1
- 14. END

Вышепреведенный 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.

15. 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 ее в переменную @TmpDate, затем переводит дату в строковом формате в тип данных даты и времени и помещает ее 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".
```

16. SET @Result=DateAdd(dd, -1, @Result) - из даты, хранимой в перменной @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").

17. RETURN @Result - возвращает значение, хранимое в переменной @Result. Для создания функции, выполним вышеописанный код, как и в случае с предыдущей функцией, нажав кнопку

🕴 Execute

После появления cooбщения "Command(s) completed successfully." закройте окно с кодом.

Проверим работу функции **"Последний день месяца"** выполнив ее. Создайте новый пустой запрос, затем в окне с пустым запросом наберите команду SELECT dbo.[Последний день месяца] ('12/07/08') и нажмите кнопку

? Execute

на панели инструментов (рис. 12.6).



увеличить изображение

Рис. 12.6.

Появится результат выполнения новой скалярной пользовательской функции: 2008-12-31 (рис. 12.6).

Теперь перейдем к созданию табличных пользовательских функций. Для создания табличной пользовательской функции в обозревателе объектов, в БД "Students", в папке "Programmability", щелкните ПКМ по папке "Functions" и в появившемся меню выберите пункт "New/Table-valued Function". Появится окно новой табличной пользовательской функции (рис. 12.7)



увеличить изображение

Рис. 12.7.

Рассмотрим структуру кода табличной пользовательской функции.

Табличная пользовательская функция состоит из следующих разделов:

- 1. Область определения имени функции (Inline_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. 12.7) наберите следующий код (рис. 12.8):

```
SQLQuery11.sqL.. Бурков (61))*
```

```
1 ---
 SET ANSI NULLS ON
 GO
 SET QUOTED IDENTIFIER ON
 GO
F --- 1
CREATE FUNCTION [ФУНКЦИЯ отбора по возрасту]
  1
      -- Add the parameters for the function here
 RETURNS TABLE
 AS
 RETURN
 1
     -- Add the SELECT statement with parameter references here
     SELECT ФИО, [Дата рождения], Bospact = DateDiff (уу, [дата рождения], GetDate()))
     FROM Студенты
 GO
```

Рис. 12.8.

Из кода представленного на рис. 12.8 видно, что данная табличная функция не имеет параметров и реализуется командой

SELECT ФИО, [Дата рождения], Возраст = DateDiff(уу, [Дата рождения], GetDate()) FROM Студенты

Из вышепредставленной команды видно, что из таблицы "Студенты" отображаются поля "ФИО" и "Дата рождения", а также вычислимое поле "Возраст". Поле "Возраст" вычисляется при помощи встроенной функции DateDiff вычисляющей различие между датами в определенных 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) (уу - год, mm - месяц, dd - день), "начальная дата" - дата начала периода и "конечная дата" - дата конца периода. В нашем случае в качестве начальной даты берем дату рождения студента, а в качестве конечной даты берем 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

? Execute

на панели инструментов (рис. 12.9).

Se Microsoft SQL Server Management Studio										0
Ne Edit Ven Query Project Debug Taols Win	don Connunity Help									
Steeders D R R R D R A 4	1									
		an letter a	-	A.						
131 16 Arrest 1 Derve -	· · · · · · · · · · · · · · · · · · ·	63(6)(67) =	A 19 19	CR.M.						
Object Explore	X SQLQuery2.sql_ABy	pece (52))*		and the second s						* ×
Correct. 2/ 2/ = 1 3	JELEUS - FROM	10001-14/88	Dia orcopa	TO BOSPACTUIII						0
ACER-JACSERFELS BOD, Server 20.0.1442 - ACER-JACS ACER-JACSERFELS BOD, Server 20.0.1442 - ACER-JACS ACER-JACSER ACER-JACSERFELS ACER-JACERFELS ACER-	E-95.									
iai 🔤 Appreçate Functions	*									æ
 System Functions Database Trabase 	Tenuts 1 Messag	999								
a 🔁 Assembles	@V/O	Дате рождения	Bospace							
iai 🔁 Types	NA sceatt 1	1983-12-12	25							
ik 🛄 Rules	2 Temposa M.M.	1982-53-00	26							
iii 🥥 Defevits	3 Mjase M.A.	1002-05-14	-26							
W 🔄 Plan Guidea	4 Cranpose E K	1981-09-27	27							
a ja service proter	5 Konstructe A.A.	1901-04-12	27							
m Storage	6 Damandas H F	1925.09.02	25							
a Ca Seorth	7 Untertaine as Fil	1985/03.17	28							
a Ca Server Objects	a Francisco F.B.	1985,07.08	20							
In Cas Replication	States Diversion Dive	1478.03.54	14							
iki 🛄 Management	and the second second	10100000	1							
(B) SQLServer Apent (Apent XPs doubled)	T2 PENDINENA A.O.	18/942-12	9							
		cofully.			ACER-3405E49515 (10.0 CTF)	ACER-SADIE 49	S15 Anerice	Students	00.00.00	10 rows
Beaty						in t	Col 51	(2) 51		244
TIMOK 0 2 9 3 Boreact 50, Se	nier 🔛 Grown	(C) Tes		Totarrowski	Alexandre and a second s		- Contractor			* 15.45

увеличить изображение

Рис. 12.9.

В нижней части окна появиться таблица с фамилиями, датами рождения и возрастом студентов на данный момент времени (рис. 12.9).

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. 12.10):



Рис. 12.10.

Перейдем теперь к созданию диаграмм. В БД "Microsoft SQL Server 2008" все диаграммы находятся в папке "Database Diagrams" обозревателя объектов (рис. 14.1).



Рис. 14.1.

Создадим диаграмму, обеспечивающую целостность данных нашей БД "Students". Для создания новой диаграммы в БД "Students" щелкните **IIKM** 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. 14.2).

Add Table	? 🛛
Tables Оценки Предметы Специальности Студенты	
Refresh Add	Close

Рис. 14.2.

В окне добавления таблиц выделите все таблицы нашей БД и нажмите кнопку "Add" (рис. 14.2). 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. 14.3).

ables and Columns		? 🛚
Relationship name:		
FK_Студенты_Специальности		
Primary key table:	Foreign key table:	
Специальности	Студенты	
Код специальности	Код специальности	
	ОК	Cancel

Рис. 14.3.

В окне создания связи нажмите кнопку "**Ok**". Появится окно настройки свойств связи "**Foreign Key Relationship**" (рис. 14.4).

FK_Студенты_Специальности*	Editing properties for new relationship. The 'Tables property needs to be filled in before the new relation	And Columns Specification' onship will be accepted.
	🛛 (General)	
	Check Existing Data On Creation Or Re-Enabling	Yes
	Tables And Columns Specification	
	🖂 Database Designer	
	Enforce For Replication	Yes
	Enforce Foreign Key Constraint	Yes
	INSERT And UPDATE Specification	
	Identity	
	(Name)	FK_Студенты_Специальности
	Description	
		роц 141



Рис. 14.4.

Оставьте свойства связи без изменений и в окне свойств связи нажмите кнопку "**Ok**".В диаграмме между таблицами "**Студенты**" и "**Специальности**" появится связь в виде ломаной линии (рис. 14.5).

Аналогичным образом создайте связь таблицы "Студенты" с таблицей "Оценки", перетащив поле "Код студента" из таблицы "Студенты" на одноименное поле в таблице "Оценки". Затем, свяжите таблицы "Предметы" и "Оценки", перетащив поле "Код предмета" из таблицы "Предметы" на поля "Код предмета 1", "Код предмета 2" и "Код предмета 3" таблицы "Оценки". После выполнения вышеперечисленных действий диаграмма примет следующий вид (рис. 14.5).

ACER-3AD5E49... - Diagram_0* Оценки * Студенты * (Код студента) [Код студента] 040 [Дата экзанена 1] Пол [Код преднета 1] [Дата рождения] [Ouenca 1] Родители [Дата экзапена 2] Предметы * Appec (Код преднета 2) [Код преднета] Телефон [Querka 2] [Наитенование предлета] [Пастортные данные] [Дата экзанена 3] Юписание преднета] Нонер зачетки! Код преднета 31 [Оценка 3] [Дата поступления] [Cpeg-snit bann] Toynna Специальности * Kype (Код специальности) [Код специальности] (Наиченование специально... [Очная форна обучения] [Описание специальности]

увеличить изображение

Рис. 14.5.

Закройте окно с диаграммой, щелкнув мышью по кнопке закрытия

×

расположенной в верхнем правом углу окна с диаграммой. Появится окно с вопросом о сохранении новой диаграммы, где необходимо нажать кнопку "Yes" (рис. 14.6).

Aicrosoft SQL Server Management Studio	?
ACER-3AD5E49515.Students - Diagram_0	
Yes No	Cancel

Рис. 14.6.

Появится окно определения имени новой диаграммы "**Choose Name**". В окне определения имени, задайте имя диаграммы как "Диаграмма БД Студенты" и нажмите кнопку "**Ok**" (рис. 14.7).

Choose Name		
Enter a name for the di Диаграмма ЕД Студен	agram: ITBI	
	ОК	Cancel

Рис. 14.7.

Появится окно "Save" с запросом сохранения таблиц, входящих в диаграмму. В данном окне необходимо нажать кнопку "Yes" (рис. 14.8).

Специальности Студенты			~
Предметы			
ОЦЕНКИ			
			1
		1.1	

Рис. 14.8.

Перейдем 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. 14.9).



Рис. 14.9.

Для начала создадим триггер, выводящий сообщение "Запись добавлена" при добавлении записи в таблицу "Студенты". Создадим новый триггер, щелкнув ПКМ по папке "Triggers" в таблице "Студенты" и в появившемся меню выбрав пункт "New Trigger". Появится следующее окно с новым триггером (рис. 14.10):



Рис. 14.10.

Рассмотрим структуру триггеров:

- 1. Область определения имени функции (**Trigger_Name**);
- 2. Область, показывающая для какой таблицы создается триггер (Table_Name);
- Область, показывающая когда выполнять триггер (INSERT при создании записи в таблице, DELETE - при удалении и UPDATE - при изменении) и как его выполнять (AFTER - 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 рис. 14.11.

```
SQLQuery6.sql ... ii Бурков (52))*
                   ...
+ -- =
 SET ANSI NULLS ON
 GO
 SET QUOTED IDENTIFIER ON
 GO
CREATE TRIGGER [Индикатор добавления]
   ON dbo.CTygentu
    AFTER INSERT
 AS
 BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from ...
由
    SET NOCOUNT ON;
    -- Insert statements for trigger here
    PRINT 'Запись добавлена'
 END
 GO
```



Из рис. 14.11 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

? Execute

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 нем наберите следующую команду для добавления новой записи в таблицу "Студенты" (рис. 14.12):



увеличить изображение

Рис. 14.12.

Выполните набранную команду, нажав кнопку

🕴 Execute

на панели инструментов. В таблицу будет добавлена новая запись, и триггер выведет сообщение "Запись добавлена" (рис. 14.12).

Теперь создадим триггер отображающий сообщение **"Запись изменена"**. Создайте новый триггер, как в предыдущем случае. В окне нового триггера наберите следующий код (рис. 14.13):

```
SQLQuery8.sql ...й Бурков (51))*
```



Рис. 14.13.

Из рис. 14.13 видно, что новый триггер "Индикатор изменения" 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 нем наберите команду, представленную на рис. 14.14.



увеличить изображение

Рис. 14.14. Выполните набранную команду, нажав кнопку

? Execute

на панели инструментов. В таблицу будет добавлена новая запись, и триггер выведет сообщение "Запись изменена" (рис. 14.14).

Для полноты картины создадим триггер, выводящий сообщение при удалении записи из таблицы **"Студенты"**. Создайте новый триггер и в нем наберите код, показанный на рис. 14.15.



Рис. 14.15.

Создаваемый триггер "Индикатор удаления" выполняется после удаления записи (AFTER DELETE) из таблицы студенты (ON dbo.Студенты). После удаления записи триггер выводит сообщение "Запись удалена" (PRINT 'Запись удалена'). Выполните код, представленный рис. 14.15. В нижней части окна с кодом появится сообщение "Command(s) completed successfully.".

Проверим работу триггера "Индикатор удаления" удалив созданную ранее запись из таблицы "Студенты". Для этого создайте новый запрос и в нем наберите следующую команду (рис. 14.16):



Рис. 14.16.

Выполните вышеприведенную команду. После удаления записи триггер "Индикатор удаления" отобразит сообщение "Запись удалена" (рис. 14.16).

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 нем наберите следующий код (рис. 14.17):

```
SQLQuery10.sql... Бурков (56))*
SET ANSI NULLS ON
 GO
 SET QUOTED IDENTIFIER ON
 GO
CREATE TRIGGER [Удаление Студента]
    ON dbo. Cтуденты
    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. Ouenku
    FROM Deleted
    WHERE Deleted. [Код студента]=Оценки. [Код студента]
    DELETE dbo.Crygentu
    FROM Deleted
    WHERE Deleted. [Код студента]=Студенты. [Код студента]
 END
 GO
```

Рис. 14.17.

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

Замечание: При срабатывании триггера вместо удаления записи создается 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.Ouenku 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 рис. 14.17. 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 нем наберите следующий код (рис. 14.18):





Рис. 14.18.

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. 14.19):



Рис. 14.19.

Задание. В программе Microsoft SQL Server создать базу данных для учета успеваемости студентов, содержащую следующие объекты:

Таблицы:	 Специальности (Код специальности, Наименование специальности, Описание специальности)[5 записей]. Предметы (Код предмета, Наименование предмета, Описание предмета)[5 записей].
	 Студенты (Код студента, ФИО, Пол, Дата рождения, Родители, Телефон, Дата поступления, Паспортные данные, Группа, Курс, Код специальности, Очная форма обучения, Номер зачетки)[10 записей].
	 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)[10 записей].

Запросы:	 Студенты+Специальности (Связывает таблицы "Студенты" и "Специальности" по полю "Код специальности"). Студенты+Оценки (Связывает таблицы "Студенты" и "Оценки" по полю "Код студента", а также таблицу "Предметы" по полю "Код предмета" с таблицей "Оценки" по полям и "Код предмета 1" и "Код предмета 2" и "Код предмета 3").
Фильтры:	 Фильтры для отображения студентов отдельных специальностей (На основе запроса "Студенты+Специальности"). Фильтры для отображения студентов, не имеющих родителей или имеющих только одного родителя (На основе запроса "Студенты+Специальности"). Фильтры для отображения студентов заданной формы обучения (На основе запроса "Студенты+Специальности").
Хранимые процедуры	 Хранимая процедура для вычисления среднего арифметического трех величин. Хранимая процедура для отбора студентов из таблицы "Студенты" по их "ФИО" Хранимая процедура для отбора студентов, у которых средний балл выше заданного. Хранимая процедура для отображения студентов старше заданного возраста
Пользовательские функции	 Скалярная пользовательская функция, вычисляющая среднее трех величин. Табличная пользовательская функция, вычисляющих текущий возраст студентов в зависимости от их даты рождения.
Диаграммы	 "Диаграмма БД Студенты", отображающая связи между таблицами.
Триггеры	 Триггер, выводящий сообщение "Запись добавлена" при добавлении записи в таблицу "Студенты". Триггер, отображающий сообщение "Запись изменена" при обновлении записи в таблице "Студенты". Триггер "Удаление студента" для обеспечения целостности данных, 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".

Контрольные вопросы

- 1. Каким образом можно получить доступ к MS SQL Server?
- 2. С помощью каких средств можно создать таблицу для MS SQL Server?
- 3. Что такое первичный ключ?
- 4. Каким образом можно создать автоматическую нумерацию строк таблицы?

- 5. Что означают Not Null?
- 6. Как создать новый запрос?
- 7. Как создать новый фильтр?
- 8. Как использовать фильтр?
- 9. Что такое хранимая процедура?
- 10. Какова структура хранимой процедуры?
- 11. Назначение пользовательских функций.
- 12. Синтаксис скалярной пользовательской функции.
- 13. Табличные пользовательские функции.
- 14. 1. Структура триггеров.
- 15. 2. Для чего используется диаграмма.
- 16. З. Связи между таблицами.

Работа с литературой:

Рекомендуемые источники информации					
(№ источника)					
Дополнительная	Методическая	Интернет-ресурсы			
1-2	1-2	1-3			
	Рекомендуемые источ (№ источ Дополнительная 1-2	Рекомендуемые источники информации (№ источника) Дополнительная Методическая 1-2 1-2			

Оценочные средства: устный отчет к лабораторной работе (См.: Фонд оценочных средств)

Лабораторная работа № 2. «Выполнение индивидуальных заданий по проектированию информационных систем в Visual Studio 2012. Создание ленточных и табличных форм для работы с базами данных.»

Форма проведения лабораторная работа (3 часа)

Цель работы:

Научиться создавать пользовательский интерфейс (главная кнопочная форма, простые ленточные формы для работы с данными).

Перейдем теперь к созданию пользовательского интерфейса. Его создание начнем с создания главной кнопочной формы. Запустите "Microsoft Visual Studio 2008" и откройте созданный ранее проект "StudentsDB", щелкнув по его значку в области "Recent Projects" стартовой страницы "Start Page" (рис. 18.1).



Рис. 18.1.

После появления стандартного окна среды разработки в рабочей области на форму поместите надпись (Label) и четыре кнопки (Button) как показано на рис. 18.2.

🔚 Form1		
	Label1	
	Button 1	
	Button2	
	Button3	
	Button4	

Рис. 18.2.

Замечание: Для создания надписи на панели объектов необходимо нажать кнопку А Label

а затем нарисовать прямоугольник мышью на форме, удерживая ЛКМ. Кнопки создаются таким же образом, только на панели объектов нажмите кнопку в Button После создания объектов перейдем к настройке их свойств. Начнем с настройки свойств формы. Выделите форму, щелкнув **ЛКМ** в пустом месте формы. На панели свойств задайте свойства формы как представлено ниже:

- **FormBorderStyle** (Стиль границы формы): Fixed3D;
- **МахітігеВох** (Кнопка развертывания формы во весь экран): False;
- **МіпітіzeBox** (Кнопка свертывания формы на панель задач): False;
- **Техt** (Текст надписи в заголовке формы): База данных "Студент".

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

- AutoSize (Авторазмер): False;
- Font (Шрифт): Microsoft Sans Serif, размер 14;
- **ForeColor** (Цвет текста): Темно синий;
- **Техt** (Текст надписи): База данных "Студент";
- **TextAlign** (Выравнивание текста): MiddleCenter.

У кнопок задайте надписи (свойство "Text") как показано на рис. 18.3.

🔜 База данных "Студент"	
о База данных "Студент"	0.0
Таблица "Специальности"	
Таблица "Предметы"	
Таблица "Студенты"	
Таблица "Оценки"	

Рис. 18.3.

После настройки свойств вышеперечисленных объектов форма примет вид представленный на рис. 18.3.

Теперь перейдем 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. 18.4).

ategories:		Templates:	
Common Items		Visual Studio installed templates	1
Data Gener Web Windo	al ws Forms	About Box Custom Dialog Explorer Form Login Form	
Workf	ting fow	MDI Parent Splash Screen User Control Windows Form Inherited Form	in and a second s
		Inherited User Control	
A blank Windo	ws Form		
ame:	Form2.vb		

Рис. 18.4.

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 рис. 18.5.

🔜 Form2		
	Label1	

Рис. 18.5.

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

FormBorderStyle (Стиль границы формы): Fixed3D;

- **МахітігеВох** (Кнопка развертывания формы во весь экран): False;
- **MinimizeBox** (Кнопка свертывания формы на панель задач): False;
- **Техt** (Текст надписи в заголовке формы): Таблица "Специальности".

На форме выделите надпись, щелкнув 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): Темно синий;
- Техt (Текст надписи): Таблица "Специальности";
- **TextAlign** (Выравнивание текста): MiddleCenter.

После настройки всех вышеперечисленных свойств форма будет выглядеть следующим образом (рис. 18.6):

🔡 Таблица "Специальности"	
о О Таблица "Специальности"	0-0-0

Рис. 18.6.

Теперь поместим на форму поля таблицы "Специальности". Сначала откройте панель "Источники данных" (Data Sources), щелкнув по ее вкладке в правой части окна среды разработки (смотри рис. 18.6). На панели "Источники данных" отобразите поля таблицы "Специальности", щелкнув по значку "+", расположенному слева от имени таблицы (рис. 18.7).



Рис. 18.7.

Панель "Источники данных" примет вид, представленный на рис. 18.7.

Замечание: Под полями таблицы специальности в виде подтаблицы располагается таблица "Студенты" (рис. 18.7). 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. 18.7), позволяющего выбирать объект, отображающий содержимое выделенного поля (рис. 18.8).



Рис. 18.8.

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 рис. 18.9



Рис. 18.9.

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;
- Специальности TableAdapter (Адаптер таблиц для таблицы "Специальности") обеспечивает передачу данных с формы в таблицу и наоборот.
- **TableAdapterManager** (Менеджер адаптера таблиц) управляет работой объекта Специальности TableAdapter ;
 - 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" ее 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, щелкнув по вкладке "Form1.vb [Design]" в верхней части рабочей области. Для подключения новой формы "Специальности" к главной кнопочной форме дважды щелкните ЛКМ по кнопке "Таблица "Специальности", расположенной на главной кнопочной форме (рис. 18.3).

В появившемся окне кода формы в процедуре "**Button1_Click**" наберите команду "**Form2.Show**()", предназначенную для открытия формы "**Таблица** "Специальности"" (Form2), как это показано на рис. 18.10.



увеличить изображение

Рис. 18.10. 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. 18.11).

🔡 Таблиц	а "Специа	льнос	ти"				
I I I I I I	. of !	5 🕨	▶	⊕ 🗙			
Наиме	Габлиц; нование спе	а "Сг	пеци ости:	иаль	ност	'и"	
0	писание спе	циальн	ости:	Матем	атическ	ие м	

Рис. 18.11.

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. 18.12):

🔜 Таблица "Предметы"	
i 4 0 of {0} ▶ ▶ 🕂 🗙 🛃	
О Таблица "Предметы" Наименование предмета: Описание предмета:	 9

Рис. 18.12.

На главной кнопочной форме дважды щелкните **ЛКМ** по кнопке **"Таблица "Предметы"**" и в появившемся окне кода в процедуре **"Button2_Click"** наберите **"Form3.Show**()" (рис. 18.13).

the state	Private Sub Button2_Click(ByWal sender As System.Object, ByWal e As System.EventArgs) Mandles Button2.Click
	Form3.Show()
-	End Sub
End	(Class

увеличить изображение

Рис. 18.13.

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. 18.14):

📰 Таблица "Предметы" 🛛 🔀				
	\$ X 🖬			
Таблица "П Наименование предмета: Описание предмета:	РЕДМЕТЫ" Операционые сист Microsoft Windows '			
Описание предмета:	Microsoft Windows '			

Рис. 18.14.

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", щелкнув ЛКМ по знаку "+", расположенному слева от названия таблицы. Отобразятся все поля таблицы "Студенты" (рис. 18.15).



Рис. 18.15.

Замечание: Обратите внимание на тот факт, что поля "Дата рождения" и "Дата поступления" отображаются объектом "Выбор даты" (DataPicker), 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. 18.15).

Создайте новую форму и поместите в ее 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. 18.16):

🔡 Таблица "Студенти	61 ^{°°}	
) 🕨 🕅 🕂 🗙 📓	
о Табли	ца "Студенты"	
ФИU:		
Пол:		
Дата рождения:	4 ноября 2008 г.	~
Родители:		
Адрес:		
Телефон:		
Паспортные данные:		
Номер зачётки:		
Дата поступления:	4 ноября 2008 г.	~
Группа:		
Курс:		
Код специальности:		
Очная форма обучения:		

Рис. 18.16.

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. 18.17).

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

увеличить изображение

Рис. 18.17.

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. 18.18).

66

🔡 Таблица "Студент	ы"				
Таблица "Студенты"					
≏ИО:	Иванов А.И.				
Пол:	Мужской				
Дата рождения:	12 декабря 1983 г.	~			
Родители:	Отец и Мать				
Адрес:	Москва				
Телефон:	+74957895674				
Паспортные данные:	8567-567543				
Номер зачётки:	13245				
Дата поступления:	1 сентября 2007 г.	~			
Группа:	MM11				
Курс:	1				
Код специальности:	1				
Очная форма обучения:					

Рис. 18.18.

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. 18.19):

67

🔡 Таблица "Оценки"		×
. 14 4 0 of {	s 🕨 🛛 🖶 🗙 🖬	
о Таб	лица Оценки"	0-0-0
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. 18.19.

Подключите вновь созданную форму таблицы "**Оценки**" к главной кнопочной форме. Для этого отобразите главную кнопочную форму и на ней дважды щелкните **ЛКМ** по кнопке "**Таблица** "**Оценки**"". В появившемся окне с кодом, в процедуре "**Button4_Click**" наберите команду "**Form5.Show**" (рис. 18.20).

Private Suo Buttoni Click(ByVal sender As System.Object, ByVal e As System.Eventargs) Handles Buttoni.Click Form5.Show()
End Sub

увеличить изображение

Рис. 18.20.

Проверьте работу формы таблицы "**Оценки**", запустив проект, и на главной кнопочной форме нажмите кнопку "**Таблица** "**Оценки**"". Появится вновь созданная форма (рис. 18.21).

🔡 Таблица "Оценки'	P		X
. 📢 🔌 🕺 of 9	> > + × 🗐		
Tað	блица "Оценки"		
Код студента:	0		
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. 18.21.

В заключение, откройте обозреватель проекта (**Solution Explorer**) щелкнув по его вкладке в правой части окна среды разработки. На данной панели должны отобразиться все выше созданные формы (рис. 18.22).

Solution Explorer	→ ╄ ×
🔓 🚯 💽 🗉 🍇	
📴 StudentsDB	
- 📴 My Project	
📑 app.config	
🖭 Form1.vb	
E Form2.vb	
E Form3.vb	
E Form4.vb	
Form5.vb	
StudentsDataSet.xsd	
Solution Explorer	urces
Company and all hour pe	

Рис. 18.22.

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 рис. 20.1.

69

🔡 Таблица "Студенты" 🛛 🛛 🔀		
	0} 🕨 🕅 🕂 🗙 🔚	
ороди и стравити и стр общини и стравити и стр общини и стравити и стр общини и стравити и стр общини и стравити и с общини и стравити и стр общини и стравити и стр общини и стравити и стр общини и стравити и стравити и стравити и с и стравити и стр и стравити и		
ФИО: Пол:		
Дата рождения:	15 ноября 2008 г. 💌	
Родители:		
Адрес:		
Толофон:		
Паспортные данные:		
Номер зачётки:		
Дата поступления:	15 ноября 2008 г. 💌	
Группа:		
Курс:		
Код специально сти:		
Очная форма обучения:		
Button1	Button2 Button3	
Button4	Button5 Button6	
	Button 7	

Рис. 20.1.

В качестве надписей на созданных кнопках (Свойство "Caption") задайте как: "Первая", "Предыдущая", "Добавить", "Последняя", "Следующая", "Удалить" и "Сохранить" (рис. 20.2).

🔣 Таблица "Студенты" 🛛 🛛 🔀			
)} 🕨 🕅 🖶 >	< 🔒	
ородина и поредина и по Оставли и поредина и по Оставли и поредина и по	ца "Студент	гы" о	
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	Предидущая	Добавить	
Поспедняя	Следущая	Удалить	
	Сохрнить		

Рис. 20.2.

Дважды щелкните 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. 20.3).

```
Frivate Sub Buttonl_Click(ByVal sender As System.Cbject, ByVal e As System.EventArgs) Handles Buttonl.Click
CtypextuBindingSource.MoveFirst()
End Sub
```

увеличить изображение

Рис. 20.3.

Дважды щелкните **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. 20.4).

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

увеличить изображение Рис. 20.4.

Дважды щелкните ЛКМ по кнопке "Добавить" и в появившемся окне кода формы "Form4" в процедуре "Button3_Click" наберите команду для добавления новой записи "СтудентыBindingSource.AddNew()" (рис. 20.5).

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

увеличить изображение

Рис. 20.5.

Дважды щелкните **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. 20.6).

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

увеличить изображение

Рис. 20.6.

Дважды щелкните **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. 20.7).

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

увеличить изображение

Рис. 20.7.

Дважды щелкните ЛКМ по кнопке "Удалить" и в появившемся окне кода формы "Form4" в процедуре "Button6_Click" наберите команду для удаления текущей записи "СтудентыBindingSource.RemoveCurrent()" (рис. 20.8).

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

увеличить изображение

Рис. 20.8.

Дважды щелкните **ЛКМ** по кнопке "**Сохранить**" и в появившемся окне кода формы "**Form4**" в процедуре "**Button7_Click**" наберите команду для сохранения изменений, отображенную на рис. 20.9.

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

увеличить изображение

Рис. 20.9.

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 зачетки", "Курс"** и **"Код специальности"**. После удаления, перечисленных полей форма, отображающая таблицу **"Студенты"** примет следующий вид (рис. 20.10):

😸 Таблица "Студенты" 🛛 🛛 🔀		
	D} ▶ ▶ 🕂 🕽	× 🖫
о о Табли	ца "Студен	ты" о
ФИО: Пол:		
Дата рождения:	15 ноября 2008 г	· · · · · · · · · · · · · · · · · · ·
Адрес:		
Телефон: Паспортные данные:		
Номер зачётки: Дата поступления:	15 ноября 2008 г	· · · · · · · · · · · · · · · · · · ·
Группа: Курс:		
Код специальности: Очная форма обучения:		
Первая	Предидущая	Добавить
Последняя	Следущая	Удалить
	Сохрнить	

Рис. 20.10.

Для отображения полей "**Телефон**", "**Паспортные данные**" и "**Номер зачетки**" 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 зачетки", как это показано на рис. 20.11.
📰 Таблица "Студенты" 🛛 🛛 🔀				
)} ▶ ▶ ⊕ 🕽	× 🖬 🚽		
о Табли	ца "Студен	ты" о		
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	Предидущая	Добавить		
Поспедняя	Следущая	Удалить		
	Сохрнить			

Рис. 20.11.

Теперь у созданных объектов настроим маски ввода. Начнем с объекта, отображающего номер зачетки. На форме выделите соответствующее полю **"Номер зачетки"** текстовое поле ввода по маске. Для задания маски в меню действий с объектом выберите пункт **"Set Mask..."** (Установить маску...) (рис. 20.12).

💀 Таблица "Студент	ol"		×	
14 4 0 of {	} ▶ ▶ ♣	X		
Табли	ца "Студен	нты"		
ФИО:				
Пол:				
Дата рождения:	15 ноября 2008	} r.	¥	
Родители:				
Адрес:	3	14 24		
Телефон:				
Паспортные данные:			Mack	adTaxtBox Tacks
Номер зачётки: (Set Ma	sk
Дата поступления:	15 ноября 2008	3 r		
Группа:				
Курс: Код специальности:				
Очная форма обучения:				
Первая	Предидущая	Добавить		
Последняя	Следущая	Удалить		
	Сохрнить			

Рис. 20.12.

Замечание: Для отображения меню действий в верхнем правом углу объекта необходимо нажать кнопку

4

(рис. 20.12).

После выбора пункта "Set Mask..." на экране появится окно задания маски "Input Mask" (Введите маску) (рис. 20.13).

Mask Des	scription	Data Format	Validating Type
Nurreric ((5-digits)	12345	Int32
Phone nu	mber	(574) 555-0123	(none)
Phone nu	mber no area code	555-0123	(none)
Short dat	e	12/11/2003	DateTime
Short date and time (US)		12/11/2003 11:20	DateTime
Social sec	curity number	000-00-1234	(none)
Time (Eur	opean/Military)	23:20	DateTime
Time (US)		11:20 D	DateTime
Zip Code		98052-6399	(none)
<custom< td=""><td>></td><td></td><td>(none)</td></custom<>	>		(none)
	T		

Рис. 20.13.

В окне "**Input Mask**" выберите маску "**Numeric (5-digits**)" (Числовое (5-цифр)) и нажмите кнопку "**Ok**" (рис. 20.13).

Для текстового поля ввода по маске для поля **"Паспортные данные"** задайте маску как показано на рис. 20.14.

Mask Description		Data Format	Validating Type
Numeric Phone nu Phone nu Short da Short da Social se Time (Eu Time (US Zip Code	(5-dgits) umber te te and time (US) curity number ropean/Military))	12345 (574) 555-0123 555-0123 12/11/2003 12/11/2003 11:20 000-00-1234 23:20 11:20 98052-6399	Int32 (none) (none) DateTime DateTime (none) DateTime DateTime (none)
<custor< td=""><td>1></td><td></td><td>(none)</td></custor<>	1>		(none)
/lask:	0000-000000		Use ValidatingType
Mask:	0000-000000		Use ValidatingTyp

Рис. 20.14.

Замечание: Обратите внимание, что паспортные данные отображаются как четыре числа, тире, шесть чисел. То есть в поле "Mask" (Macka) нужно задать "0000-0000000". 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 рис. 20.15.

Mask Description		Data Format	Validating Type
Numeric (5-digits) Phone number Phone number no area code Short date Short date and time (US) Social security number Time (European/Military) Time (US) Zip Code		12345 (574) 555-0123 555-0123 12/11/2003 12/11/2003 11:20 000-00-1234 23:20 11:20 98052-6399	Int32 (none) (none) DateTime DateTime (none) DateTime DateTime (none)
<custom< td=""><td></td><td></td><td>(none)</td></custom<>			(none)
lask:	+7 (999) 000-0000		Use ValidatingType

Рис. 20.15.

Теперь нам необходимо подключить созданные текстовые поля ввода по маске к соответствующим полям. Для этого с панели "Источники данных" (DataSources) перетащите поле "Номер зачетки" на текстовое поле ввода по маске, расположенное справа от надписи "Номер зачетки". Проделайте 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 счетчика (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. 20.16).



Рис. 20.16.

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

Замечание: После перетаскивания поля "Курс" на форму слева от него появится еще одна надпись "Курс". Удалите ее, щелкнув 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. 20.17).

abl	TextBox
₽ \$	ComboBox
Α	Label
A	LinkLabel
•	ListBox
0	[None]
	Customize

Рис. 20.17.

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. 20.18).

String Collection Editor	? 🔀
Enter the strings in the colection (one per line):	
Мужосой Женский	
	OK Cancel

Рис. 20.18.

В появившемся окне в отдельных строках наберите элементы выпадающего списка: "Мужской" и "Женский" (рис. 20.18). 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. 20.19).



Рис. 20.19.

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 ее код будет автоматически подставляться в поле **"Код специальности"** таблицы **"Студенты"**.

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

E 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. 20.20).

ComboBox Tas	sks	
Use data bou	nd items	
Data Binding M	1ode	
Data Source	(none)	~
Display Member	-	~
Value Member		×
Selected Value	(none)	~

Рис. 20.20.

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

внутрии поля параметра. Появится древовидная структура выбора источника данных (рис. 20.21).

Đ	🚫 None 🍟 СтудентыBindingSource
-	🗊 Other Data Sources
	😑 🦣 Project Data Sources
	🕒 🗗 StudentsDataSet
	🗈 🦙 Form4 List Instances
0	Add Project Data Source
Sole	act a PindingSource to bind to
Jelt	eet a bindingsource to bind to.

Рис. 20.21.

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. 20.22):

ComboBox Tasks			
🕑 Use data bou	nd items		
Data Binding M	1ode		
Data Source	СпециальностиBindingS	¥	
Display Member	Наименование специальн	~	
Value Member	Код специальности	~	
Selected Value	СтудентыBindingSource -	~	
Add Query			
Preview Data			

Рис. 20.22.

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. 20.23).

🛃 StudentsDataSet	😲 Сту	дентыBindingSource	🔀 Студенты	TableAdapter	TableAdapterManager
🚰 СтудентыBindingNat	vigator	🖓 Специально стиВ	indingSource	🔀 Специаль	ностиTableAdapter

увеличить изображение

Рис. 20.23.

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 рис. 20.24.

🔜 Таблица "Студент	📰 Таблица "Студенты" 🛛 🛛 🔀			
	o} 🕨 🕅 🕂 🗶 📓 🔡			
Таблица "Студенты"				
ФИО:				
Пол:				
Дата рождения:	15 ноября 2008 г. 💌			
Родители:				
Адрес:				
Телефон:	+7()			
Паспортные данные:	<u></u>			
Номер <mark>зачётк</mark> и:				
Дата поступления:	15 ноября 2008 г. 💽			
Группа:				
Курс:	0			
Код специальности: (
Очная форма обучения:				
Первая	Предидущая Добавить			
Поспедняя	Следущая Удалить			
	Сохрнить			

Рис. 20.24.

Проверим работу формы, отображающей таблицу "Студенты". Запустите проект и на главной кнопочной форме нажмите кнопку "Таблица "Студенты"". Появится форма, имеющая следующий вид (рис. 20.25):

🛃 Таблица "Студент	ы	
Табли	іца "Студенты"	
ФИ <mark>О</mark> :	Иванов А.И.	
Пол:	Мужской	~
Дата рождения:	12 декабря 1983 г.	*
Родители:	Отеци Мать	~
Адрес:	Москва	
Талефон:	+7 (495) 789-5674	
Паспортные данные:	8567-567543	
Номер зачётки:	13245	
Дата поступления:	1 сентября 2007 г.	~
Группа:	MM11	
Курс:	1	
Код специальности:	MM	~
Очная форма обучения:		
Первая	Предидущая Добав	ить
Поспедняя	Следущая Удали	ть
	Сохрнить	

Рис. 20.25.

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

Отобразите форму для таблицы "**Оценки**", щелкнув **ЛКМ** по ее вкладке в верхней части рабочей области среды разработки. На форму, справа от поля "**Средний балл**" поместите кнопку (рис. 20.26).

🔡 Таблица "Оценки'		$\mathbf{\times}$
	0} 🕨 🕅 🕂 🛠 🔚	
o o Tab	о блица "Оценки"	0
Код студента:		
Дата экзамена 1:	15 ноября 2008 г. 💌	
Код предмета 1:		
Оценка 1:		
Д <mark>ата экза</mark> мена 2:	15 ноября 2008 г. 💌	
Код предмета 2:		
Оценка 2:		
Д <mark>ата экзамена</mark> 3:	15 ноября 2008 г. 💌	
Код предмета 3:		
Оценка 3:		
Средний балл:	Button 1	

Рис. 20.26.

Задайте свойство "Text" у вновь созданной кнопки как "Вычислить" (рис. 20.27).

🔡 Таблица "Оценки'		×
	0} ▶ ▶ 🕂 🛠 🔚	
o Tac	о блица "Оценки"	0
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:		
Оценкз 3:		
Средний балл:	Вычислить	

Рис. 20.27.

Теперь дважды щелкните **ЛКМ** по кнопке "Вычислить" и в появившемся коде процедуры "Button1_Click" наберите код, представленный на рис. 20.28, вычисляющий среднее оценок.

```
Private Sub Buttoni_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Buttoni.Click

Cpegnukk_GannTextBox.Text = (Val(Ouenna_ITextBox.Text) + Val(Ouenna_2TextBox.Text) + Val(Ouenna_STextBox.Text)) / 3

End Sub
```

увеличить изображение

Рис. 20.28.

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. 20.29).

🔡 Таблица "Оценки'		$\mathbf{\times}$
. 📢 🖣 🛛 1. of 9) • • • • • • • • • • • • • • • • • •	
Tað	блица "Оценки"	
Код студента:	1	
Дата экзамена 1:	1 февраля 2008 г. 💌	
Код предмета 1:	1	
Оценка 1:	5	
Д <mark>ата экзамена</mark> 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 Вычислить	

Рис. 20.29.

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

Перейдем 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 нее 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 рис. 22.1.

Form6			
	Label1	GroupBox1 Label2 ListBox1	
		RadoButton1 RadoButton2 But	ton 1
Label3	8	Button2	Button 3
Label4		Button4	Button5

увеличить изображение

Рис. 22.1.

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

v

расположенную справа от таблицы "Студенты". В появившемся списке объектов для отображения всей таблицы выберите "DataGridView" (рис. 22.2).



Рис. 22.2.

Перетащите таблицу "Студенты" из панели "Источники данных" на форму. Форма примет следующий вид (рис. 22.3):

			Label1		12-2-2	
	Код студента	ФИО	Пол	Дата	GroupBox1 Label2	
*				poxdenna	ListBox1	
					RadioButton1 RadioButton2 Button1	

увеличить изображение

Рис. 22.3.

Обратите внимание на то, что на форме появилась таблица для отображения данных, подключенная к таблице "Студенты". Также появились объекты связи и панель навигации (рис. 22.4).

StudentsDataSet	😲 СтудентыBindingSource	🔄 Студенты TableAdapter	🔂 TableAdapterManager	📅 СтудентыBindingNavigator
-----------------	-------------------------	-------------------------	-----------------------	----------------------------

увеличить изображение

Рис. 22.4.

Теперь перейдем к настройке свойств объектов. Начнем с настройки свойств формы. Задайте свойства формы следующим образом:

- FormBorderStyle (Стиль границы формы): Fixed3D;
- MaximizeBox (Кнопка развертывания формы во весь экран): False;
- MinimizeBox (Кнопка свертывания формы на панель задач): False;
- **Техt (Текст надписи в заголовке формы)**: Таблица "Студенты" (Табличный вид).
- Задайте свойства надписей (Label1, Label2, Label3 и Label4) как:
- AutoSize (Авторазмер): False;
- Техt (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)**: Темно 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 (Включен) равное True (Истина).

Заполните список (ListBox1) значениями, представленными на рис. 22.5, а затем нажмите кнопку "Ok".

String Collection Editor	? 🛛
Enter the strings in the collection (one per lin	e):
ФИО Пол Дата рождения Родители Адресс Телефон Паспортные данные Помер зачётки Дата поступления Группа Курс Очная форма обучения	
8	<u>></u>
	OK Cancel

Рис. 22.5.

Настроим таблицу для отображения данных, удалив из нее поля с кодами. Выделите таблицу на форме и отобразите ее меню действий, щелкнув **ЛКМ** по кнопке

расположенной в верхнем правом углу таблицы. В меню действий выберите пункт **''Edit columns...''** (рис. 22.6).

DataGridView Task	s	
Choose Data Source	СтудентыBindingSource	~
Edit Columns Add Column		
Enable Adding		
Enable Editing		
Enable Deleting		
Enable Column Rec	ordering	
Dock in parent contain	her	
Add Query Preview Data		

Рис. 22.6.

Появится окно настройки свойств полей таблицы "Edit Columns" (рис. 22.7).



Рис. 22.7.

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 рис. 22.7. 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\Cтуденты", а параметр "Display Member" равным "ФИО". Остальные параметры оставьте без изменений (рис. 22.8).

ComboBox Tas	iks	
🕑 Use data bou	nd items	
Data Binding M	1ode	
Data Source	СтудентыBindingSource	¥
Display Member	ΦИΟ	~
Value Member		~
Selected Value	(none)	۷
Add Query Preview Data		

Рис. 22.8.

Закройте окно действий выпадающего списка. На панели невидимых объектов появится дополнительный объект связи "**СтудентыBindingSource1**", предназначенный для заполнения выпадающего списка (рис. 22.9).

88



увеличить изображение

Рис. 22.9.

После настройки всех вышеперечисленных свойств объектов новая форма примет вид (рис. 22.10):

		о Таблиц	о а "Студенты	" (Таблич	ный вид)	3
		6	0		Сортировка	5
	ФИ0	Гол	Дата рождения	Родители	Поле для сортир	OBKM:
*					Пол Дата рождения Родители Адресс Телефон Паспортные дан Номер зачётки Дата поступлени Группа	ные
2	ci.			2	 Сортировка п Сортировка п Сорти 	о возрастанию о убыванию ровать
ФИО:				~	Фильтровать	Показать воё
						-

увеличить изображение

Рис. 22.10.

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

Работу с кодом начнем с написания кода для разблокирования кнопки "Сортировать", при выборе пункта списка (ListBox1). Для создания процедуры события дважды щелкните ЛКМ по списку. Появится процедура обработки события, происходящего при выборе пункта списка (ListBox1_SelectedIndexChanged). В процедуре наберите команду разблокировки кнопки "Copтировать" (Button1): Button1.Enabled = True (рис. 22.11).

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

End Sub

увеличить изображение

Рис. 22.11.

Теперь перейдем к созданию кода сортирующего нашу таблицу в зависимости от выбранного поля и порядка сортировки при нажатии кнопки "Сортировать". Дважды щелкните ЛКМ по кнопке "Сортировать". Появится процедура "Button1_Click", выполняемая при щелчке ЛКМ по кнопке. В процедуре наберите код, представленный на рис. 22.12.

```
Private Sub Buttoni_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim Col As System.Windows.Forms.DataGricViewColumn
    Select Case ListBox1.Selectedindex
       Case 0
            Col = DataGridViewTextBoxColumn2
       CARP 1
            Col = DataGridViewTextBoxColumn3
       Case 2
            Col = DataGridViewTextBoxColumn4
       Case 3
            Col = DataGridViewTextBoxColumn5
        Case 4
            Col = DataGridViewTextBoxColumné
       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
       CTygeHTHDataGridView.Sort(Col, System.ComponentModel.ListSortDirection.Ascending)
       CTVDextmDataGridView.Sort(Col. Svstem.ComponertModel.ListSortDirection.Descending)
   End If
```

```
End Sub
```

увеличить изображение

Рис. 22.12.

Рассмотрим код более подробно:

- Команда Dim Col As System.Windows.Forms.DataGridViewColumn создает 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 включен 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 щелкните по кнопке **"Фильтровать"** и в процедуре обработки события **"Button2_Click"** наберите код: СтудентыBindingSource.Filter = "ФИО="" & ComboBox1.Text & """ (рис. 22.13).

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
CTyDextmBindingSource.Filter = "4000='" & ComboBox1.Text & "'"
```

End Sub

увеличить изображение Рис. 22.13.

Замечание: У объекта СтудентыBindingSource имеется текстовое свойство Filter (рис. 22.13), 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. 22.13).

Теперь перейдем к кнопке **"Показать все"**, отменяющей фильтрацию записей. Дважды щелкните по вышеперечисленной кнопке. Появится процедура **Button3_Click**. В появившейся процедуре наберите команду СтудентыBindingSource.Filter = "" (рис. 22.14).

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

End Sub

увеличить изображение

Рис. 22.14.

Заметим, что если присвоить свойству "Filter" значение пустой строки (""), то его действие будет отменено (рис. 22.14).

Далее рассмотрим реализацию поиска информации в таблице. Дважды щелкните по кнопке **"Найти"**. В появившейся процедуре обработки нажатия кнопки **"Button4_Click"** наберите следующий код (рис. 22.15).

```
Private Sub Buttoni_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Buttoni.Click
    For 1 = 0 To CrygestsDataGridView.ColumnCount - 1
        For j = 0 To CrygesrsDataGridView.RowCount - 1
            CTygehTwDataGridView.Item(i, j).Style.BackColor = Color.White
           CTynextsDataGridView.Item(i, j).Style.ForeColor = Color.Black
       Next 1
    Next 1
   For 1 = 0 To CrynesruDataGridView.ColumnCount - 1
       For j = 0 To CTYMENTEDataGridView.RowCount - 1
            If InStr(CrynexrsDataGridView.Item(i, j).Value, TextBox1.Text) Then
                CTYDENTEDAtaGridView.Item(i, j).Style.BackColor = Color.AliceBlue
                CTYMEHTEDataGridView.Item(1, j).Style.ForeColor = Color.Blue
           End If
       Next 1
   Next 1
End Sub
```

увеличить изображение

Рис. 22.15.

Рассмотрим более подробно код вышеприведенной 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 черный цвет текста. То есть, отменяет результаты предыдущего поиска;

Второй блок For i=0.....Next i. перебирает все ячейки таблицы и если они содержат текст, введенный в поле ввода (TextBox1), то устанавливает в них голубой цвет фона и синий цвет текста, чем выделяет искомые ячейки.

Наконец рассмотрим код для кнопки "Закрыть". Дважды щелкните ЛКМ по этой кнопке и в появившейся процедуре "Button5_Click" наберите команду

"Me.Close()", закрывающую выше рассматриваемую форму (рис. 22.16).

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

```
End Sub
```

увеличить изображение

Рис. 22.16.

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 нее новую кнопку, как это показано на рис. 22.17.

📕 Таблица "Студент	ы"	
14 4 0 ur {	0} 🕨 🕅 🖶 🕽	× 🖬
Табли	ца "Студен	ты"
ФИО:		
Пол:		~
Дата рождения:	23 ноября 2003 г	
Родители:		~
Адрес:		
Телефон:	+7()	
Паспортные данные:		
Номер зачётки:		
Дата поступления:	23 ноября 2003 г	- 💌
Группа:		
Курс:	0	
Код специальности:	-	*
Очная форма обучения:		
Первая	Предидущая	Добавить
Последняя	Следущая	Удалить
Button 8	Сохрнить	

Рис. 22.17.

Задайте надпись у новой кнопки (свойство **Text**), как "**Таблица**". Форма примет следующий вид (рис. 22.18):

📰 Таблица "Студенты" 🛛 🚺				
14 4 0 of {	0} 🕨 🕅 🕂 🗶 📓			
Табли	іца "Студенты"			
ФИО;				
Пол:	~			
Дата рождения:	23 ноября 2008 г. 💌			
Родители:	✓			
Адрес:				
Телефон:	+7(
Паспортные данные:	<u></u>			
Номер зачётки:				
Дата поступления:	23 ноября 2008 г. 💌			
Группа:				
Курс:	0			
Код специальности:	×			
Очная форма обучения:				
Первая	Предидущая Добавить			
Последняя	Следущая Удалить			
С Таблица	Сохрнить			

Рис. 22.18.

Подключим к кнопке **"Таблица"** созданную ранее табличную форму (**Form6**). Для этого дважды щелкните **ЛКМ** по кнопке **"Таблица"** и в появившейся процедуре **"Button8_Click"** наберите команду "Form6.Show" (рис. 22.19).

Private Sub ButtonS_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Form6.Show()
Ted event

```
End Sub
```

увеличить изображение

Рис. 22.19.

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. 22.20).

		Таблица	"Студенты	" (Таблич	ный вид)		
	ФИО	Пол	Дата рождения	Родите.	Поле для сортир	08191:	
•	Иванов А.И.	Мужской	12.12.1983	Отеци І	Пол		
	Петрова И.И.	Женский	01.11.1982	Мать	Родители		
	Мужин М.А.	Мужской	14.05.1982	Отец	Адресс Телефон		
	Сидорова В.К.	Женский	27.09.1981	Нот	Паспортные данные Номер зачётки Дата поступления	ые	
	Кожевников А.А.	Мужской	12.04.1981	Мать		19	
	Пальчикова Н.Е.	Женский	02.09.1983	Отеци І	Ipynna	Y	
	Царегородцев Е	Мужской	17.02.1980	Отец	 Сортировка по возрастанию Сортировка по убыванию 	о возрастанию	
	Баранова Г.В.	Женский	09.07.1980	Отеци 1			
<		Maxama	26.02.1979	Цат 🎽	Сорти	ровать	
РИO:	Mesune A.M.				(Dun mana	Devenerational	

увеличить изображение

Рис. 22.20.

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. 22.21):



Рис. 22.21.

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

Задание.

В программе Microsoft Visual Studio создать приложение для работы с базой данных по учету успеваемости студентов, созданной в предыдущей лабораторной работе. Создать главную кнопочную форму, простые и сложные ленточные формы для работы с данными, табличные формы и отчет «Студенты».

Контрольные вопросы

- 1. Настройки свойств формы.
- 2. Главная кнопочная форма.
- 3. Как настроить маски ввода.
- 4. Как создать выпадающий список
- 5. Как создать новую табличную форму.
- 6. Как обеспечить фильтрацию и сортировку данных.
- 7. Как реализуется поиск информации в таблице.

Работа с литературой:

	Рекомендуемые источники информации					
	(№ источника)					
Основная	Дополнительная	Методическая	Интернет- ресурсы			
1-2	1-2	1-2	1-3			

Оценочные средства: устный отчет к лабораторной работе (См.: Фонд оценочных средств)

Лабораторная работа № 3. «Выполнение индивидуальных заданий по проектированию информационных систем в Visual Studio 2012. Создание отчетов и диаграмм.»

Форма проведения лабораторная работа (З часа)

Цель работы:

Научиться создавать отчеты.

Начнем рассмотрение отчетов с создания ленточного отчета, отображающего таблицу "Студенты". Для начала добавим в проект новый пустой отчет. Для этого в оконном меню выберите пункт "Project\Add New Item..." (рис. 24.1).



Рис. 24.1.

Появится окно "Add New Item-StudentsDB" (Добавить новый элемент - StudentsDB). В данном окне в списке "Categories" (Категории) выберите пункт "Reporting" (Отчеты), затем в области "Templates" (Шаблоны) выберите шаблон "Report" (Отчет) и нажмите кнопку "Add" (Добавить) (рис. 24.2).

Add New Ite	m - StudentsDB		? 🛛
Categories:		Templates:	
Common 1 Code Data Gener Web Windo Repor Work	Items ral ows Forms rting flow	Visual Studio installed templates	
An empty Rep	port file for creating	a report using Microsoft reporting technology	
Name:	Report1.rdl	D:	
			Add Cancel

увеличить изображение

Рис. 24.2.

В рабочей области среды разработки появится пустой отчет. Новый отчет также отобразится и на панели обозревателя проекта (**Solution Explorer**) (рис. 24.3).



Рис. 24.3.

Для того чтобы в отчет поместить поля таблицы "Студенты" в него необходимо добавить объект "Table" (Таблица). Для этого на панели объектов (Toolbox) нажмите кнопку Таble

а затем в отчете нарисуйте прямоугольник. Отчет примет вид, представленный на рис. 24.4.

Header	
Detail	
Footer	

увеличить изображение

Рис. 24.4.

=

Замечание: Объект таблица имеет три строки:

- **Header** (заголовок) верхняя часть первой страницы отчета, содержит заголовок отчета;
- **Detail** (область данных) средняя часть каждой страницы отчета, содержит поля отображаемой таблицы;
- **Footer** (примечание) нижняя часть последней страницы отчета, содержит итоговую информацию по отчету.
 - Добавим в таблицу в область данных дополнительные строки для отображения полей таблицы "Студенты". Выделите область данных, как это показано на рис. 24.5, щелкнув **ЛКМ** по заголовку строки области данных

	Header	
=	Detail	
	Footer	

Рис. 24.5.

Для вставки новой строки щелкните **ПКМ** по заголовку выделенной строки

и в появившемся меню выберите пункт "Insert Row Below" (Вставить строку ниже) (рис. 24.6).



Рис. 24.6.

Проделайте эту операцию одиннадцать раз. Таблица примет вид (рис. 24.7)

	Header	
=		
=		
=		
=		
=		
=		
=		
=		
=		
=		
=		
=		
····	Footer	

Рис. 24.7.

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. Щелкните **ПКМ** по заголовку первого столбца таблицы и в появившемся меню выберите пункт "Delete Columns" (Удалить строки) (рис. 24.8).

0	Insert Column to the Left
0Ţ	Insert Column to the Right
Ψ.	Delete Columns
X	Cut
	Сору
	Paste
7	Properties

Рис. 24.8.

Теперь объединим ячейки в заголовке и примечании таблицы. Выделите ячейки заголовка таблицы, обведя их курсором мыши удерживая **ЛКМ**. Затем щелкните по выделению **ПКМ** и в появившемся меню выберите пункт "**Merge Cells**" (Объединить ячейки) (рис. 24.9).

99



Рис. 24.9.

Проделайте аналогичную операцию с ячейками примечания. После выполнения вышеперечисленных операций таблица примет вид (рис. 24.10):



Рис. 24.10.

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 рис. 24.11. В одну ячейку перетаскивается одно поле Поля связи, имеющие в своем имени слово "Код" перетаскивать не нужно.

Hea	der
1	=Fields!ΦИO.Value
	=Fields!Flon.Value
	=Fields!Дата_рождения Value
	=Fields!Родители Value
	=Fields!Agpec.Value
	=Fields/Teneфон.Value
	=Fields!Паспортные_данные.Value
	=Fields!Номер_зачётки.Value
1	=Fields!Дата_поступления.Value
	=Fields!/Fpynna.Value
	=FieldslKypc.Value
E .	=Fields!Очная форма обучения Value

Рис. 24.11.

В левом столбце таблицы наберите имена полей и установите их выравнивание по правому краю (Свойство **TextAlign**). В заголовке наберите заголовок отчета **"Отчет таблицы "Студенты"** и сделайте выравнивание текста в нем по центру (рис. 24.12).



Рис. 24.12.

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. 24.13).

	Отчёт таблиц	ы "Студенты"
-	ΦИΟ	=Fields!ФИО.Value
	Пол	=Fields!Non.Value
	Дата рождения	=Fields!Дата_рождения.Value
	Родители:	=Fields!Родители Value
•	Адрес	=Fields!Agpec.Value
	Телефон:	=Fields!Телефон Value
	Паспортные данные:	=Fields!Паспортные_данные.Value
-	Номер зачётки:	=Fields!Номер_зачётки.Value
	Дата поступления:	=FieldslДата_поступления.Value
	Группа	=Fields![pynna.Value
-	Kypc:	=Fields!Kypc Value
=	Очная форма обучения:	=Fields!Очная_форма_обучения Value

Рис. 24.13.

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. 24.14). Замечание: Если кнопка

⊞ -

отсутствует на панели инструментов, то необходимо включить панель редактирования границ отчетов (**Report borders**). Для этого щелкните **ПКМ** по панели инструментов и в появившемся меню выберите пункт ''**Report borders**''.

ФИО	=Fields!@I/O Value
Пол	=Fields!Пon Value
Дата рождения	=Fields!Дата_рождения Value
Родители	=Fields!Родители.Value
Адрес	=Fields!Agpec.Value
Телефон	=Fields!Teneфoн Value
Паспортные данные:	=Fields!Паспортные_данные Value
Номер зачётки:	=Fields!Homep_зачётки Value
Дата поступления	=Fields:Дата_поступления.Value
Группа	=Fields!//pynna.Value
Kypc:	=Fields!Kypc.Value
Очная форма обучения:	=Fields!Очная форма обучения.Value

Рис. 24.14.

Теперь создадим форму отображающую созданный отчет. Добавьте в проект новую форму (Form7). Определите заголовок формы (Свойство Text) как "Отчет таблицы "Студенты"".

Поместите на форму специальный объект, отображающий отчеты "MicrosoftReportViewer", используя кнопку

MicrosoftReportViewer

расположенную на панели объектов (**Toolbox**). К объекту, отображающему отчеты подключите, созданный ранее отчет. Для этого в меню действий в выпадающем списке **"Choose report"** (Выберите отчет) выберите отчет **"StudentsDB.Report1.rdlc"**. Разверните объект, отображающий отчеты во всю форму. Для этого в меню действий объекта выберите пункт **"Dock in Parent Container"** (Развернуть в родительский контейнер). Меню действий примет вид (рис. 24.15):

ReportViewer	Tasks	
Choose Report	StudentsDB.Report1.rdlc	~
Choose Data So	urces	
Rebind Data Sou	irces	
Design a new rej	port	
Dock in parent o	ontainer	

Рис. 24.15.

Замечание: Обратите внимание на тот факт, что после подключения отчета к объекту, отображающему отчеты, на панели невидимых объектов появились объекты связи, подключающие отчет к таблице "Студенты" (рис. 24.16).

StudentsDataSet	பி СтудентыBindingSource	🔄 Студенты Table Adapter	
-----------------	--------------------------	--------------------------	--

Рис. 24.16.

После выполнения всех вышеперечисленных действий форма, отображающая отчет примет вид, представленный на рис. 24.17.



увеличить изображение Рис. 24.17.

Проверим работоспособность нового отчета, подключив форму для его отображения к кнопке на форме "Таблица "Студенты"". На форме, отображающей таблицу "Студенты" создайте кнопку (Button9) (рис. 24.18).

😸 Таблица "Студент	ы"	
4 0of {	0} 🕨 🕅 🖶 🕽	× 🖫
Табли	ца "Студен	ты"
ФИО:		
Пол:		*
Дата рождения:	27 ноября 2003 г	. 💌
Родители:		~
Адрес:		
Телефон:	+7()	
Пастортные данные:		
Номер зачётки:		
Дата поступления:	27 ноября 2003 г	. 💌
Группа:		
Курс:	0	
Код специальности:		~
Очная форма обучения:		
Первая	Предидущая	Добавить
Последняя	Следущая	Удалить
Таблица	Сохрнить С	Button9

Рис. 24.18.

Задайте надпись на кнопке (Свойство Text) равную "Отчет" (рис. 24.19).

105

🛃 Таблица "Студен	ты?	
14 4 0 of	{0} • • 💠	×
Табл	ица "Студе	нты"
ФИО:		
Non:		~
Дата рождения:	27 ноября 200	8 r. 💌
Родители		~
Адрес:		
Телефон:	+7(
Паспортные данные:		
Номер зачётки:		
Дата поступления:	27 ноября 200	Br. 💌
Spynna:		
Курс:	0	
Код специальности:		~
Очная форма обучения:		
Переая	Предидущая	Добавить
Поспедняя	Спедушая	Удалить
Таблица	Сохрнить	

Рис. 24.19.

Теперь определим код обработчика события нажатия кнопки. Дважды щелкните ЛКМ по кнопке "Отчет" и в появившейся процедуре "Button9_Click" наберите команду "Form7.Show()" (рис. 24.20).

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

увеличить изображение

Рис. 24.20.

Запустите проект и на главной кнопочной форме нажмите кнопку "Таблица "Студенты"". На появившейся ленточной форме, отображающей таблицу "Студенты" нажмите кнопку "Отчет". Появится новая форма с отчетом, построенным по таблице "Студенты" (рис. 24.21).

📕 Отчёт таблицы "Студенты"	
🔟	2 2 2 100% • Find J. Next
Отчёт таблиц	ы "Студенты"
ФИО:	Иванов А.И.
	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:	MM11
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

увеличить изображение

Рис. 24.21.

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

На этом мы завершаем разработку нашей БД "Студент".

Задание.

В программе Microsoft Visual Studio для работы с базой данных по учету успеваемости студентов, созданной в лабораторной работе № 1. Создать отчет «Студенты».

Контрольные вопросы

1. Как создать ленточный отчет.

2. Как создать форму отображающую созданный отчет.

Работа с литературой:

Рекомендуемые источники информации (№ источника)

Основная	Дополнительная	Методическая	Интернет- ресурсы
1-2	1-2	1-2	1-3

Оценочные средства: устный отчет к лабораторной работе (См.: Фонд оценочных средств)

Лабораторная работа № 4. «Подготовка документации IT

проекта.» Форма проведения лабораторная работа (3 часа)

Цель работы:

Научиться готовить необходимую организационно-техническую и эксплуатационную документацию IT проекта.

Общие требования к документированию

Документы должны быть представлены на бумажном виде (оригинал) и на магнитном носителе (копия). Исходные тексты программ - только на магнитном носителе (оригинал). Возможно предоставление комплекта документации и текстов программ на компактдисках.

Все документы должны быть оформлены на русском языке. Состав документов на общее программное обеспечение, поставляемое в составе АИС, должен соответствовать комплекту поставки компании - изготовителя.

Перечень подлежащих разработке документов

В ходе создания Подсистемы должен быть подготовлен и передан Заказчику комплект документации в составе:

• проектная документация и материалы техно-рабочего проекта на разработку Подсистемы;

•конструкторская, программная и эксплуатационная документация на Подсистему;

• сопроводительная документация на поставляемые программно-аппаратные средства в комплектности поставки заводом-изготовителем;

• предложения по организации системно-технической поддержки функционирования Подсистемы.

Состав и содержание комплекта документации на Подсистему может быть уточнен на стадии проектирования.

Подготовленные документы должны удовлетворять требованиям государственных стандартов и рекомендаций по оформлению, содержанию, форматированию, использованию терминов, определений и надписей, обозначений программ и программных документов.

Порядок выполнения лабораторной работы

По заданному преподавателем описанию предметной области разработать организационно-техническую и эксплуатационную документацию (использовать результаты предыдущих работ).

Контрольные вопросы

1. Каковы общие требования к документированию.

2. Каков перечень подлежащих разработке документов.

1 a001 a C J	niepaiypon.		
	Рекомендуемые источ (№ источ	ники информации ника)	
Основная	Дополнительная	Методическая	Интернет- ресурсы
1-2	1-2	1-2	1-3

Работа с литературой:

Оценочные средства: устный отчет к лабораторной работе (См.: Фонд оценочных средств)

Лабораторная работа № 5. «Расчет экономической эффективности проекта.» Форма проведения: лабораторная работа (1,5 часа). Цель работы:

Научиться проводить стоимостный анализ.

Стоимостный анализ представляет собой соглашение об учете, используемое для сбора затрат, связанных с работами, с целью определить общую стоимость процесса. Стоимостный анализ основан на модели работ, потому что количественная оценка невозможна без детального понимания функциональности предприятия. Обычно АВС применяется для того, чтобы понять происхождение выходных затрат и облегчить выбор нужной модели работ при реорганизации деятельности предприятия (Business Process Reengineering, BPR). С помощью стоимостного анализа можно решить такие задачи, как определение действительной стоимости производства продукта, определение действительной стоимости поддержки клиента, идентификация наиболее дорогостоящих работ (тех, которые должны быть улучшены в первую очередь), обеспечение менеджеров финансовой мерой предлагаемых изменений и т.д.

ABC-анализ может проводиться только тогда, когда модель работы последовательная (следует синтаксическим правилам IDEF0), корректная (отражает бизнес), полная (охватывает всю рассматриваемую область) и стабильная (проходит цикл экспертизы без изменений), другими словами, когда создание модели работы закончено.

АВС включает следующие основные понятия:

- объект затрат причина, по которой работа выполняется, обычно основной выход работы. Стоимость работ есть суммарная стоимость объектов затрат ("Сборка и тестирование компьютеров", рис. 1);
- двигатель затрат характеристики входов и управлений работы ("Заказы клиентов", "Правила сборки и тестирования", "Персонал производственного отдела", рис. 1), которые влияют на то, как выполняется и как долго длится работа;
- центры затрат, которые можно трактовать как статьи расхода.



Рисунок 1 - Иллюстрация терминов АВС

При проведении стоимостного анализа в BPwin сначала задаются единицы измерения времени и денег. Для задания единиц измерения следует вызвать диалог Model Properties (меню Model), закладка ABC Units (рис. 2).

General Purpose Definition	on Source Statu	us Numbering Display
Layout ABC Units Page	Setup Header/Foote	er Shapes Draw Style
Model Name:		
Currency description:	<u>Symbol</u> p	placement:
<u>\$U.S.</u> ▼	1p.	•
p. 0 ÷	2 🕂	
Ti <u>m</u> e Unit:	Decimals in <u>f</u> requency values:	Decimals in duration values:
Days 💌	2 🕂	2 ÷

Рисунок 2 - Настройка единиц измерения валюты и времени

Если в списке выбора отсутствует необходимая валюта (например, рубль), ее можно добавить. Диапазон измерения времени в списке Time Unit достаточен для большинства случаев — от секунд до лет.

Затем описываются центры затрат (cost centers). Для внесения центров затрат необходимо вызвать диалог Cost Center Editor из меню Model (рис. 3).

Управление		<u>\</u> dd
Cost centers:	Uo	date
Компоненты Рабочая сила		-1-+-
Управление		elere
	C	lose
	<u>I</u>	<u>l</u> elp
<u>D</u> efinition		
Затраты на управление, связанные с составлением графика работ, формированием партий компьютеров, контролем за сборкой и тестированием	^ Q	
	0	

Рисунок 3 - Диалог Cost Center Editor
Каждому центру затрат следует дать подробное описание в окне Definition. Для задания стоимости работы (для каждой работы на диаграмме декомпозиции) следует щелкнуть правой кнопкой мыши по работе и на всплывающем меню выбрать Cost (рис. 4). В диалоге Activity Cost указывается частота проведения данной работы в рамках общего процесса (окно Frequency) и продолжительность (Duration). Затем следует выбрать в списке один из центров затрат и в окне Cost задать его стоимость. Аналогично назначаются суммы по каждому центру затрат, т. е. задается стоимость каждой работы по каждой статье расхода. Если в процессе назначения стоимости возникает необходимость внесения дополнительных центров затрат, диалог Cost Center Editor вызывается прямо из диалога Activity Properties/Cost соответствующей кнопкой.

ACDY	ity Properties	
UDP Values UOW Name Definition Sta Activity Name: Сборка настольных компьютеров	Source Roles stus Font Color s	Box Style Costs
Cost Center	1	yónь
Компоненты		16 000.00
Visional Citila Visional Citila		0.00
Data is from this level. Queride decompositions C. Compute from decompositions	Total cost: Total cost x Frequency:	16 100,0 193 200,0
Data is from this level. Quentide decompositions Compute from decompositions Final accor.	Total cost: Total cost x Frequency: Cost Ce	16 100,0 193 200,0 nter Editor
Data is from this level.	Total cost: Total cost x Frequency: Cost Ce	16 100,(193 200,(nter Editor
Data is from this level. Qvemide decompositions Compute from decompositions Brequency: 12.00 Duration: 1.00 Hours Duration x Frequency	Total cost Total cost x Frequency: Cost Ce 12.00 Houre	16 100,0 193 200,0 nter Editor

Рисунок 4 - Задание стоимости работ в диалоге Activity Properties/Cost

Общие затраты по работе рассчитываются как сумма по всем центрам затрат. При вычислении затрат вышестоящей (родительской) работы сначала вычисляется произведение затрат дочерней работы на частоту работы (число раз, которое работа выполняется в рамках проведения родительской работы), затем результаты складываются. Если во всех работах модели включен режим Compute from Decompositions (рис. 4), подобные вычисления автоматически проводятся по всей иерархии работ снизу вверх (рис. 5).



Рисунок 5 - Вычисление затрат родительской работы

Этот достаточно упрощенный принцип подсчета справедлив, если работы выполняются последовательно. Встроенные возможности ВРwin позволяют разрабатывать упрощенные модели стоимости, которые, тем не менее, оказываются чрезвычайно полезными при предварительной оценке затрат. Если схема выполнения более сложная (например, работы производятся альтернативно), можно отказаться от подсчета и задать итоговые суммы для каждой работы вручную (Override Decompositions). В этом случае результаты расчетов с нижних уровней декомпозиции будут игнорироваться, и при расчетах на верхних уровнях будет учитываться сумма, заданная вручную. На любом уровне результаты расчетов сохраняются независимо от выбранного режима, поэтому при выключении опции Override Decompositions расчет снизу вверх производится обычным образом.

Результаты стоимостного анализа могут существенно повлиять на очередность выполнения работ. Предположим, что для оценки качества изделия необходимо провести три работы:

- внешний осмотр стоимость 50 руб.;
- пробное включение стоимость 150 руб.;
- испытание на стенде стоимость 300 руб.

Предположим также, что с точки зрения технологии очередность проведения работ несущественна, а вероятность выявления брака одинакова (50%). Пусть необходимо проверить восемь изделий. Если проводить работы в убывающем по стоимости порядке, то затраты на получение готового изделия составят:

300 руб. (испытание на стенде)*8 +150 руб. (пробное включение) *4 + 50 руб. (внешний осмотр) *2 = 3100 руб.

Если проводить работы в возрастающем по стоимости порядке, то на получение готового изделия будет затрачено:

50 руб. (внешний осмотр) *8 +150 руб. (пробное включение) *4 + 300 руб. (испытание на стенде) *2 = 1600 руб.

Следовательно, с целью минимизации затрат первой должна быть выполнена наиболее дешевая работа, затем — средняя по стоимости и в конце — наиболее дорогая.

Результаты стоимостного анализа наглядно представляются на специальном отчете BPwin, настройка которого производится в диалоговом окне Activity Cost Report (меню Tools/Reports/Activity Cost Report) (рис. 6). Отчет позволяет документировать имя, номер, определение и стоимость работ, как суммарную, так и раздельно по центрам затрат.

Standard reports: Model: Деятельность к Start from activity:	омпании	<u> </u>	Update	New Delet
Activity Options 2 Activity name 1 Activity number Definition 3 Activity costs	Number of Levels: Cost Center Options 4 Cost center name ☐ Cost center definition 5 Cost center costs Time Period Options ☐ Activity duration ☐ Activity frequency		C Fixed column C Tab delimited C Comma delimited C DDE table C <u>R</u> PTwin	
Activity Ordering C Alphabetical Hierarchical C Breadth First			C Re C <u>Fille</u> C He	 C Repeating group C Filled C Header I ▲ Merger I ■ Remove special charge
ABC Options	f node costs nter totals	Format Option C Report co Report co C Report ac	st centers o st centers o st centers withi	mn headings nly vithin activity n cost centers

Рисунок 6 - Диалог настройки отчета по стоимости работ

Результаты отображаются и непосредственно на диаграммах. В левом нижнем углу прямоугольника работы может показываться либо стоимость (по умолчанию), либо продолжительность, либо частота проведения работы. Настройка отображения осуществляется в диалоге Model Properties (меню Model/Model Properties), закладка Display (ABC Data, ABC Units).

Контрольные вопросы

1. Каковы цели стоимостного анализа.

2. АВС-анализ.

Работа с литературой:

Рекомендуемые источники информации (№ источника)					
Основная	Дополнительная	Методическая	Интернет- ресурсы		
1-2	1-2	1-2	1-3		

Оценочные средства: устный отчет к лабораторной работе (См.: Фонд оценочных средств)

6 КРИТЕРИИ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ

Оценка «отлично» выставляется студенту, если он продемонстрировал глубокие, исчерпывающие знания и творческие способности в понимании, изложении и использовании учебно-программного материала; логически последовательные, содержательные, полные, правильные и конкретные ответы на все поставленные вопросы и дополнительные вопросы преподавателя; свободное владение основной и дополнительной литературой, рекомендованной учебной программой.

Оценка «хорошо» выставляется студенту, если он продемонстрировал твердые и достаточно полные знания всего программного материала, правильное понимание сущности и взаимосвязи рассматриваемых процессов и явлений; последовательные, правильные, конкретные ответы на поставленные вопросы при свободном устранении

замечаний по отдельным вопросам; достаточное владение литературой, рекомендованной учебной программой.

Оценка «удовлетворительно» выставляется студенту, если он продемонстрировал твердые знания и понимание основного программного материала; правильные, без грубых ошибок ответы на поставленные вопросы при устранении неточностей и несущественных ошибок в освещении отдельных положений при наводящих вопросах преподавателя; недостаточное владение литературой, рекомендованной учебной программой.

Оценка «неудовлетворительно» выставляется студенту, если он продемонстрировал неправильные ответы на основные вопросы, допущены грубые ошибки в ответах, непонимание сущности излагаемых вопросов; неуверенные и неточные ответы на дополнительные вопросы.

7 МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРЫ ОЦЕНИВАНИЯ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ И (ИЛИ) ОПЫТА ДЕЯТЕЛЬНОСТИ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ

Текущая аттестация студентов проводится преподавателями, ведущими лабораторные занятия по дисциплине, в следующей форме: отчет письменный по заданию преподавателя.

Допуск к лабораторным работам происходит при наличии у студентов печатного варианта отчета. Защита отчета проходит в форме доклада студента по выполненной работе и ответов на вопросы преподавателя.

Отчет включает в себя следующие разделы: титульный лист с названием работы; цель работы; краткие теоретические сведения; описание результатов лабораторной работы (скриншоты); вывод из работы, включающий в себя описание проделанной работы.

Оценку «отлично» студент получает, если оформление отчета соответсвует установленным требованиям, правильно отвечает на предложенные преподавателем контрольные вопросы, правильно отвечает на дополнительные вопросы по теме лабораторной работы.

Оценку «хорошо» студент получает, если оформление отчета соответсвует установленным требованиям, правильно отвечает на предложенные преподавателем контрольные вопросы.

Оценку «удовлетворительно» студент получает без беседы с преподавателем, если оформление отчета соответсвует установленным требованиям.

Отчет может быть отправлен на доработку в следующих

случаях: - польностью не соовествует установленным

требованиям; - не раскрыта суть работы.

8 УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

8.1. Рекомендуемая литература 8.1.1. Основная литература

1. Проектирование информационных систем. Проектный практикум / А.В. Платёнкин, И.П. Рак, А.В. Терехов, В.Н. Чернышов ; Министерство образования и науки Российской Федерации, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Тамбовский государственный технический университет». – Тамбов : Издательство ФГБОУ ВПО «ТГТУ», 2015. – 81 с. : ил., схем. – Режим доступа: по подписке. – URL: <u>http://biblioclub.ru/index.php?page=book&id=444966</u> (дата обращения: 21.10.2019). – Библиогр. в кн. – ISBN 978-5-8265-1409-2. – Текст : электронный.

2. Бурков, А.В. Проектирование информационных систем в Microsoft SQL Server 2008 и Visual Studio 2008 / А.В. Бурков. – Москва : Интернет-Университет Информационных Технологий, 2010. – 273 с. – Режим доступа: по подписке. – URL: <u>http://biblioclub.ru/index.php?page=book&id=233750 (д</u>ата обращения: 21.10.2019). – Текст : электронный.

3. Лазицкас, Е.А. Базы данных и системы управления базами данных : [12+] / Е.А. Лазицкас, И.Н. Загумённикова, П.Г. Гилевский. – Минск : РИПО, 2016. – 267 с. : ил. – Режим доступа: по подписке. – URL: <u>http://biblioclub.ru/index.php?page=book&id=463305</u> (дата обращения: 21.10.2019). – Библиогр. в кн. – ISBN 978-985-503-558-0. – Текст : электронный.

8.1.2. Дополнительная литература:

1. Соболева, М.Л. Информационные системы. Лабораторный практикум / М.Л. Соболева, А.С. Алфимова. – Москва : Прометей, 2011. – 88 с. – Режим доступа: по подписке. – URL: http://biblioclub.ru/index.php?page=book&id=212836 (дата обращения: 21.10.2019). – Библиогр. в кн. – ISBN 978-5-4263-0025-5. – Текст : электронный.

2. Щелоков, С.А. Разработка и создание баз данных средствами СУБД Access и SQL Server / С.А. Щелоков ; Министерство образования и науки Российской Федерации, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Оренбургский государственный университет», Кафедра программного обеспечения вычислительной техники и автоматизированных систем. – Оренбург : Оренбургский государственный университет, 2014. – 109 с. – Режим доступа: по подписке. – URL: http://biblioclub.ru/index.php?page=book&id=260754 (дата обращения: 21.10.2019). – Текст : электронный.

3. Сенченко, П.В. Организация баз данных / П.В. Сенченко ; Министерство образования и науки Российской Федерации, Томский Государственный Университет Систем Управления и Радиоэлектроники (ТУСУР), Факультет дистанционного обучения. – Томск : ТУСУР, 2015. – 170 с. : схем., табл., ил. – Режим доступа: по подписке. – URL: http://biblioclub.ru/index.php?page=book&id=480906 (дата обращения: 21.10.2019). – Библиогр.: с. 163-164. – Текст : электронный.

8.1.3. Методическая литература:

1. Методические указания по выполнению лабораторных работ по дисциплине «Проектный практикум».

2. Методические рекомендации для студентов по организации самостоятельной работы по дисциплине «Проектный практикум».

8.1.4. Интернет-ресурсы:

1. http://www.intuit.ru – сайт дистанционного образования в области информационных технологий

2. http://window.edu.ru – образовательные ресурсы ведущих вузов

3. http:// www.informika.ru – сервер Министерства образования РФ и ГосНИИ Информационных технологий и телекоммуникаций. На сервере представлена разнообразная информация по всем аспектам образования (нормативная и законодательная база, обучающие ресурсы, информационные технологии).