


МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«БАЙКАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
ЧИТИНСКИЙ ИНСТИТУТ
КОЛЛЕДЖ

УТВЕРЖДАЮ:
Первый заместитель директора

Н.В. Раевский
«25» июня 2024 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

ОПЦ.08 Основы проектирования баз данных

Специальность 09.02.07 Информационные системы и программирование

Чита, 2024

Структура фонда оценочных средств
по дисциплине «Основы проектирования баз данных»
по специальности 09.02.07 Информационные системы и программирование

Результаты обучения	Критерии оценки	Формы и методы оценки
<p>Перечень умений, осваиваемых в рамках дисциплины:</p> <ul style="list-style-type: none"> - проектировать реляционную базу данных; 	<p>Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.</p> <p>«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.</p> <p>«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с основным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных</p>	<p>Подготовка выступлений</p> <p>выступления с сообщениями,</p> <p>Решение ситуационных заданий, оценка качества выполнения заданий;</p> <p>Наблюдение за выполнением практического задания, (деятельностью студента), оценка выполнения практического задания;</p> <p>Письменный опрос;</p> <p>Устный опрос,</p> <p>Самостоятельная работа,</p> <p>Дифференцированный зачет.</p>
<ul style="list-style-type: none"> - использовать язык запросов для программного извлечения сведений из баз данных 		
<p>Перечень знаний, осваиваемых в рамках дисциплины:</p> <ul style="list-style-type: none"> - основы теории баз данных; 		
<ul style="list-style-type: none"> - модели данных; 		
<ul style="list-style-type: none"> - особенности реляционной модели и проектирование баз данных; 		
<ul style="list-style-type: none"> - изобразительные средства, используемые в ER- моделировании; 		
<ul style="list-style-type: none"> - основы реляционной алгебры; 		
<ul style="list-style-type: none"> - принципы проектирования баз данных; 		
<ul style="list-style-type: none"> - обеспечение непротиворечивости 		
и целостности данных;	заданий содержат ошибки. «Неудовлетворительно» - теоретиче-	

- средства проектирования структур баз данных;	ское содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки	
- язык запросов SQL	<p>90-100% <i>правильных ответов</i> – «5»</p> <p>76-89% <i>правильных ответов</i> – «4»</p> <p>61-75% <i>правильных ответов</i> – «3»</p> <p>менее 61% - «2»</p>	Тестирование

№ п/п	Тема из рабочей программы	Проверяемые умения, знания, ОК, ПК	Наименование оценочного средства	Критерии оценивания
1	Тема 1.2. Классификация и основные понятия баз данных	ОК 1, ОК 2, ОК 4, ОК 5, ОК 9 ПК 11.1-11.6	Контрольная работа (3 вопроса)	100-95% - «5» 94-75% - «4» 74-50% - «3» Менее 50% - «2»
2	Тема 1.4. Основные функции СУБД	ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ПК 11.1-11.6	Лабораторная работа Контрольная работа (7 вопросов)	100-95% - «5» 94-75% - «4» 74-50% - «3» Менее 50% - «2»
3	Тема 1.5. Концепция базы данных и основные понятия теории баз данных	ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ПК 11.1-11.6	Лабораторные работы Контрольная работа (5 вопросов)	100-95% - «5» 94-75% - «4» 74-50% - «3» Менее 50% - «2»
4	Тема 2.4. Типовая организация СУБД и обработка SQL-запросов в серверах баз данных	ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ПК 11.1-11.6	Контрольная работа (5 вопросов) Лабораторные работы	100-95% - «5» 94-75% - «4» 74-50% - «3» Менее 50% - «2»
5	Тема 2.5. Проектирование и обработка внутренней схемы базы данных	ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ПК 11.1-11.6	Лабораторные работы	100-95% - «5» 94-75% - «4» 74-50% - «3» Менее 50% - «2»
7	Промежуточная аттестация		Экзамен (устный ответ на вопрос, выполнение практического задания)	Оценка «5» ставится, если студент: 1. полно излагает изученный материал, дает правильное определение языковых понятий; 2. обнаруживает понимание материала, может обосновать свои суждения, применить знания на практике, привести необходимые примеры не только из учебника, но и самостоятельно составленные; 3. излагает материал последовательно и правильно с точки зрения норм литературного языка; 4. в практической работе допущено не более 1 ошибки. Оценка «4» ставится, если студент:

				<p>1. дает ответ, удовлетворяющий тем же требованиям, что и для отметки «5», но допускает 1 - 2 ошибки, которые сам же исправляет, и 1 - 2 недочета в последовательности и языковом оформлении излагаемого.</p> <p>2. в практической работе допускает 2-3 ошибки.</p> <p>Оценка «3» ставится, если студент:</p> <p>1. обнаруживает знание и понимание основных положений данной темы, но:</p> <p>1) излагает материал неполно и допускает неточности в определении понятий или формулировке правил;</p> <p>2) не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры;</p> <p>3) излагает материал непоследовательно и допускает ошибки в языковом оформлении излагаемого;</p> <p>2. в практической работе допускает 4-5 ошибок.</p> <p>Оценка «2» ставится, если студент:</p> <p>1. обнаруживает незнание большей части соответствующего раздела изучаемого материала, допускает ошибки в формулировке определений и правил, искажающие их смысл, беспорядочно и неуверенно излагает материал.</p> <p>2. в практической работе допущено более 5 ошибок.</p> <p>Оценка «2» отмечает такие недостатки в подготовке ученика, которые являются серьезным препятствием к успешному овладению последующим материалом.</p>
--	--	--	--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Контрольная работа по теме
«Понятие базы данных, системы управления базами данных (СУБД)»**

Вариант 1

1. Определение СУБД
2. Функции информационных систем
3. История развития баз данных

Вариант 2

1. Определение базы данных
2. Файловые системы
3. История развития баз данных

Классификация баз данных и СУБД».

Вариант 1

1. Банк данных
2. Экземпляр сущности
3. Кардинальности связи
4. Состояние предметной области
5. Ограничения целостности
6. Функции СУБД
7. Классификация баз данных

Вариант 2

1. Информационная система
2. Сущность
3. Типы связей
4. Степень связи
5. Актуализация данных
6. Модели представления данных
7. Жизненный цикл базы данных

Лабораторная работа. «Объекты СУБД Access. Создание однотабличной БД»

Цель работы: Приобрести навыки и умения при работе с таблицами: создавать таблицы с помощью конструктора, задавать тип данных, задавать маску ввода для поля, использовать свойства поля, создавать ключи и индексы для полей таблицы, заполнять таблицы данными.

**З а д а н и е 1. С о з д а н и е т а б л и ц ы б а з ы д а н н ы х Э Т А П Ы
Р А Б О Т Ы**

1.1. Для создания новой базы данных:

- загрузите MS Access, в появившемся окне выберите пункт **Новая база данных**;
- Задайте имя вашей базы. По умолчанию MS Access предлагает вам имя базы *db1*(Access – *Database1.accdb*), а тип файла - *Базы данных Access*.
Имя задайте *Стоматология*.

1.2. Для создания таблицы базы данных:

- в окне базы данных выберите вкладку *Создание*, а затем щелкните по кнопке <Конструктор таблиц> (рис. 1).

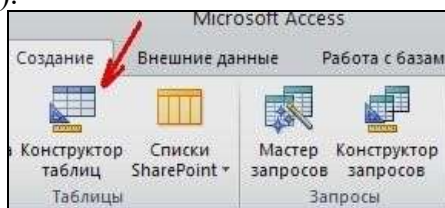


Рис. 1 Создание таблиц в режиме Конструктора

1.3. Для определения полей таблицы:

- введите в строку столбца **Имя поля** имя первого поля - *ID врача*. В
- строке столбца «Тип данных» щелкните по кнопке списка и выберите тип данных *Числовой*. Поля вкладки *Общие* оставьте такими, как предлагает Access (см.рис. 2).

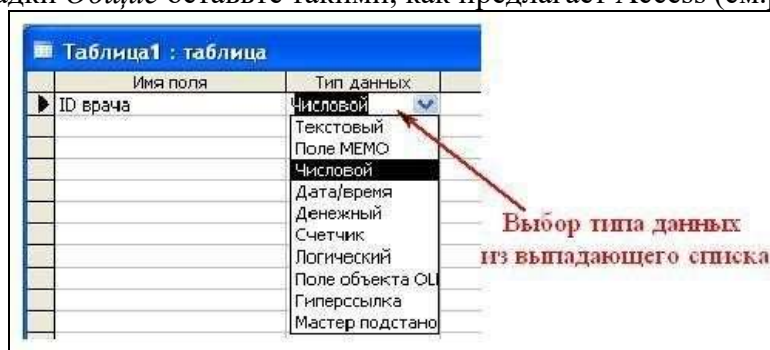


Рис. 2 Выбор типа данных

- Для определения всех остальных полей таблицы базы данных *Врачи* выполните действия, аналогичные указанным выше в соответствии с табл. 1.

Таблица 1. Описание свойств полей таблицы *Врачи*

Имя поля	Тип данных	Размер поля (вкладка <i>Общие</i>)
ID врача	Числовой	
ФИО	Текстовый	20
Должность	Текстовый	20
№ кабинета	Числовой	Длинное целое (по умолчанию)
Время приема	Текстовый	12
ВидОказываемыхУслуг	Текстовый	50 (Максимальное значение)



Рис. 3 Использование для поля дополнительных параметров с помощью вкладки «Общие»

1.4. Задайте условие для поля «№ кабинета».

Пусть по условию в поликлинике 2 этажа и значение номера кабинета находится в пределах от 100 до 130 и от 200 до 230. В строке «Условие на значение» перейдите в режим **Построителя выражений** (рис. 4) и введите условие: $\geq 100 \text{ And } \leq 130 \text{ Or } \geq 200 \text{ And } \leq 230$ (см. рис. 5). В строке «Сообщение об ошибке» введите: «Такого номера кабинета нет».



Рис. 4 Кнопка перехода в режим «Построителя выражений»

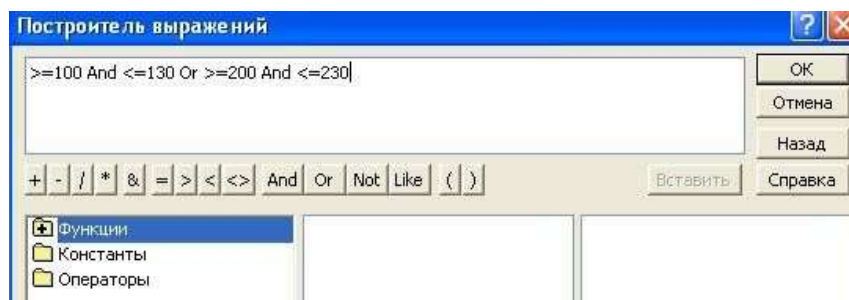


Рис. 5 Окно «Построителя выражений»

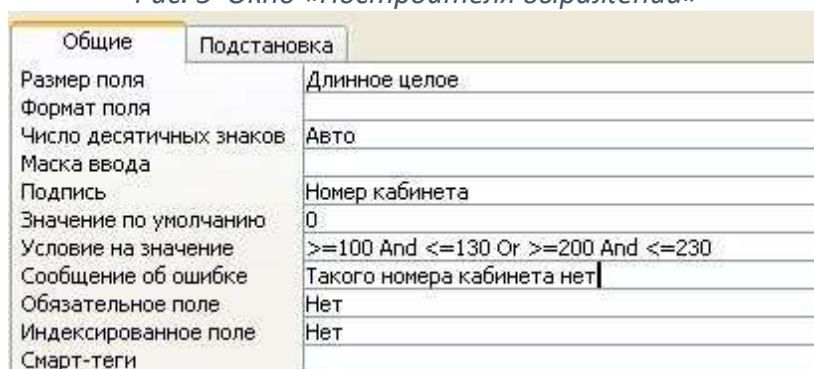
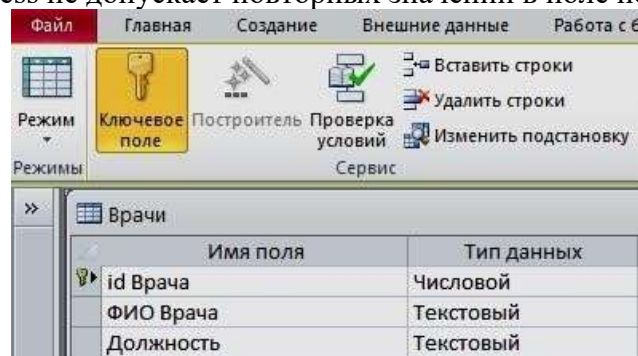



Рис. 6 Вкладка «Общие» с заданными условиями для поля «№ кабинета»

1.5. Создание ключевых полей.

Первичный ключ - это поле (или ряд полей), которое однозначно идентифицирует каждую запись в таблице. Access не допускает повторных значений в поле первичного ключа.



- Сделайте поле **ID врача** ключевым. Для этого откройте таблицу *Врачи* в режиме **Конструктора** и, поместив курсор на имя этого поля, щелкните по кнопке  - **Ключевое поле** (рис. 7):

Закройте таблицу. MS Access предложит сохранить таблицу (по умолчанию «Таблица1»). Задайте имя таблицы «Врачи».

Задание 2. Заполнение базы данных ЭТАПЫ РАБОТЫ

2.1. Создание таблиц-подстановок.

Для упрощения ввода данных в таблицу «Врачи» создайте таблицы подстановки для полей: «ВремяПриема» и «ВидОказываемыхУслуг». Для поля «Должность» данные будут подставляться из таблицы «Штатное расписание».

Подстановка упрощает режим ввода. При этом нет необходимости набирать длинные названия несколько раз. Достаточно будет выбрать необходимую запись из выпадающего списка.

2.1.1. Таблица «Смена-подстановка» (используется для поля «ВремяПриема»). Поля таблицы (см.табл.2):

Таблица 2.Описание свойств полей таблицы «Смена_подстановка»

Смена_подстановка : таблица	
Имя поля	Тип данных
НомерСмены	Текстовый
ВремяПриема	Текстовый

Примечание 1. В результате щелчка по кнопке <ОК> MS Access предложит вам задать ключевое поле. Для создаваемой таблицы щелкните по кнопке <Нет>.

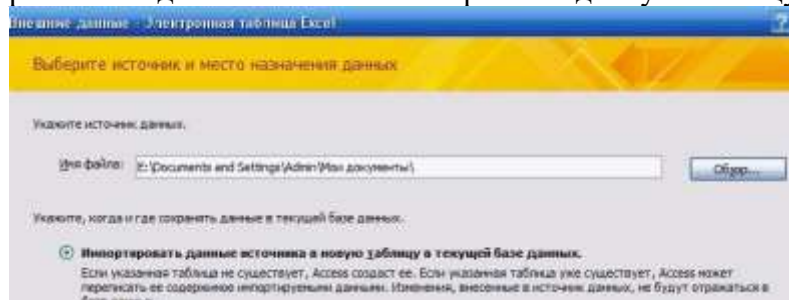
2.1.2. Таблицу «НаименованиеУслуги_подстановка» (используется для поля «Вид оказываемых услуг») создайте в Excel, чтобы импортировать её в нашу базу данных. Поля таблицы (см.табл.3):

Таблица 3. Таблица с данными «НаименованиеУслуги_подстановка»

Наименование направления услуги
Терапевтическая
Хирургическая
Ортопедическая

Импорт как самих таблиц, так и полей таблиц можно производить из разных источников и файлов разных форматов.

- Определите формат ячеек: для поля *НаименованиеУслуги* – текстовый.
- Сохраните таблицу в той же папке, где находится ваша БД *Стоматология*.
- В базе данных *Стоматология* на панели инструментов выберите **Внешние данные→Excel**.
- В открывшемся диалоговом окне выберите созданную таблицу:



- В следующем окне «Импорт электронной таблицы» выберите Лист 1, нажмите <Далее>.
- установите флажок «Первая строка содержит заголовки столбцов»,
- далее: «Данные необходимо сохранить в новой таблице»,
- далее подтвердите описание и название полей, □ далее выберите «Не создавать ключ», Готово.

Таблица «НаименованиеУслуги_подстановка» будет импортирована в вашу БД. Посмотрите ее в режиме **Конструктора**. Все ли поля соответствуют **типу данных**?

2.1.3. Таблица «Штатное расписание» (данные поля «НаименованиеДолжности» будут использоваться для подстановки в таблицу «Врачи» (см.табл. 4).

Таблица 4. Описание свойств полей таблицы «Штатное расписание»

ШтатноеРасписание : таблица	
Имя поля	Тип данных
Должность	Текстовый
Оклад	Денежный
Надбавка	Денежный

Для поля «Оклад» во вкладке **Общие** в строке «Формат поля» выбрать «Денежный».

Для поля «Должность» задать значение **Ключевое поле**.

2.2. Заполнение таблиц данными

Для заполнения таблиц данными необходимо перейти из режима **Конструктора** в режим **Таблицы** или открыть таблицу двойным щелчком.

Внимание! (данные в таблицу можно ввести только в режиме **Таблицы**).

- Заполните таблицу «Смена подстановка» данными (см.рис. 8).

Смена_подстановка	
НомерСме	ВремяПрие
1	08.00-12.00
2	13.00-19.00

Рис. 8. Таблица «Смена_подстановка» с данными

- Заполните таблицу «ШтатноеРасписание» данными (см.рис. 9).

ШтатноеРасписание : таблица			
	Должность	Оклад	Надбавка
+	врач-стоматолог	7 000,00р.	700,00р.
+	ортопед	14 000,00р.	1 200,00р.
+	хирург-терапевт	10 000,00р.	1 000,00р.

Рис. 9. Таблица «ШтатноеРасписание» с данными

2.3. Создание подстановочного поля

Для создания подстановочного поля используется два способа:

- режим **Конструктора**. Этот способ используется в том случае, если поле для подстановки стоит первым в подстановочной таблице.
- использование **Мастера подстановок**.

Откройте таблицу «Врачи» в режиме **Конструктора**. Для поля «Должность» во вкладке **Подстановка** выберите в строке **Тип элемента управления**: «Поле со списком»; в строке **Тип источника строк**: «Таблица или запрос» (задается по умолчанию); в строке **Источник строк** из выпадающего списка выбрать таблицу «Штатное расписание» - то есть таблицу, из которой данные будут подставляться в таблицу «Врачи» (см.рис. 10):

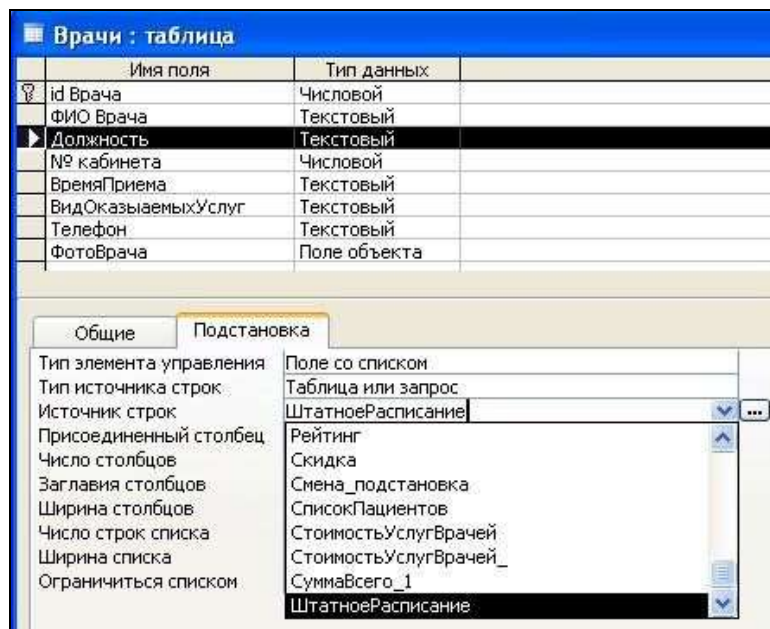



Рис. 10. Подстановка в режиме **Конструктора**

1) Для создания подстановки для поля «*ВремяПриема*» в столбце **Тип данных** выберите из выпадающего списка «**Мастер подстановки**». В следующем окне нажмите кнопку <Далее>, в открывшемся диалоговом окне выберите «*Таблица: Смена_подстановка*». Нажмите <Далее>.

В следующем окне «Создание подстановки» из списка «Доступные поля» выберите при помощи кнопки  поле «*ВремяПриема*» (данные из этого поля будут подставляться в нашу таблицу), нажмите кнопку <Далее>, затем <Готово> (рис. 11).

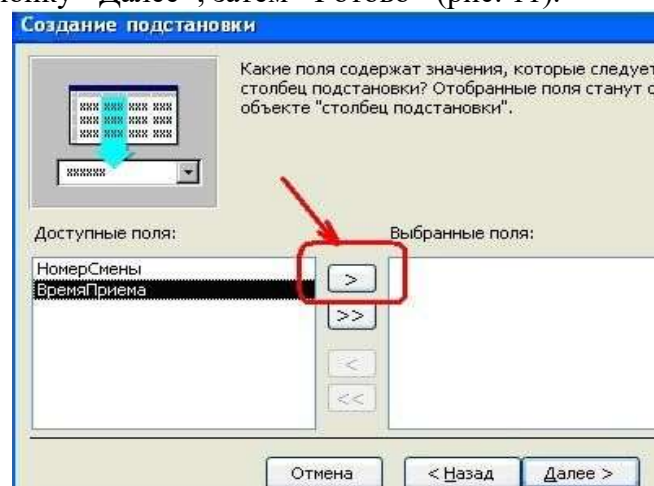


Рис 11 Использование Мастера подстановки

2.4. Ввод ограничения на данные в поле «ID врача»

Для поля «*ID врача*» в таблице «**Врачи**» выполним условия:

- коды врачей не должны повторяться,
- должна быть обеспечена возможность их изменения (из-за последнего условия в этом поле нельзя использовать тип данных **Счетчик**, в котором данные не повторяются).

Для выполнения этих условий сделайте следующее:

- установите курсор на строке «*ID врача*»;
- во вкладке **Общие** щелкните по строке параметра **Индексированное поле**;
- выберите в списке пункт «*Да (совпадения не допускаются)*»; ☐ перейдите в режим **Таблица**.

Примечание 2. Индекс - это средство MS Access, ускоряющее поиск и сортировку данных в таблице.

2.5. Заполните таблицу «Врачи» данными, используя подстановку

Использование режима "Подстановка"

Табельный ном	ФИО Врача	Должность	Номер кабин	ВремяПриема	ВидОказываемыхУслуг
100	Валиуллин Д.К.	ортопед	101	08.00-12.00	Ортопедическая
101	Галимянова О.А.	врач-стоматолог	102	08.00-12.00	Хирургическая
102	Дюкина Я.Б.	ортопед	203	08.00-12.00	Терапевтическая
103	Елизарова Е.П.	хирург-терапевт	115	08.00-12.00	Терапевтическая
104	Жуков А.Е.	врач-стоматолог	120	08.00-12.00	Терапевтическая
105	Карбышев Г.О.	ортопед	214	08.00-12.00	Ортопедическая
106	Леонидова М.Р.	врач-стоматолог	224	13.00-19.00	Терапевтическая
107	Накипов Н.Д.	хирург-терапевт	315	08.00-12.00	Хирургическая
108	Никоненко П.С.	ортопед	124	13.00-19.00	Ортопедическая

Рис 12 Пример заполнения таблицы «Врачи» с использованием режима Подстановки

Задание 3. Размещение новых объектов в таблице

3.1. Размещение объекта OLE

Рассмотрим размещение объекта OLE в таблице «Врачи» на примере поля «ФотоВрача». В нашем примере фотографии хранятся в формате графического редактора **Paint** в файлах с расширением *bmp*. (Создайте несколько таких объектов и сохраните их в той же папке, где хранится база данных).

- Откройте таблицу «Врачи» в режиме **Конструктора** и добавьте новое поле «ФотоВрача». Тип данных – **Поле объекта OLE**.
- Перейдите в режим **Таблицы**. В графе «Фото» выполните команду правой кнопкой мыши: **Добавить объект**.
- В окне **Вставка объекта** (рис. 13) отметьте переключатель «Создать из файла». Окно **Вставка объекта** позволяет ввести имя файла с фотографией. Для поиска файла воспользуйтесь кнопкой **Обзор**



Рис 13 Добавление объекта

Для введения в поле связанного объекта надо установить флажок «Связь», что позволит в дальнейшем отражать в документе изменения этого файла.

3.2. Добавление нового поля в таблицу. Создание маски ввода

1. Откройте таблицу «Врачи» в режиме **Конструктора**.
2. Добавьте новое поле «Телефон». Тип данных – **Текстовый**.
3. Задайте маску ввода для поля **Телефон** (в виде 22-22-222), для этого:
 - Щелкните на вкладке **Общие** в строке *Маска ввода* по кнопке
 - Подтвердите сохранение таблицы.
 - В открывшемся диалоговом окне «Создание масок ввода» щелкните по кнопке <Список>. В появившемся диалоговом окне «Настройка масок ввода» выполните действия по образцу (рис. 14):

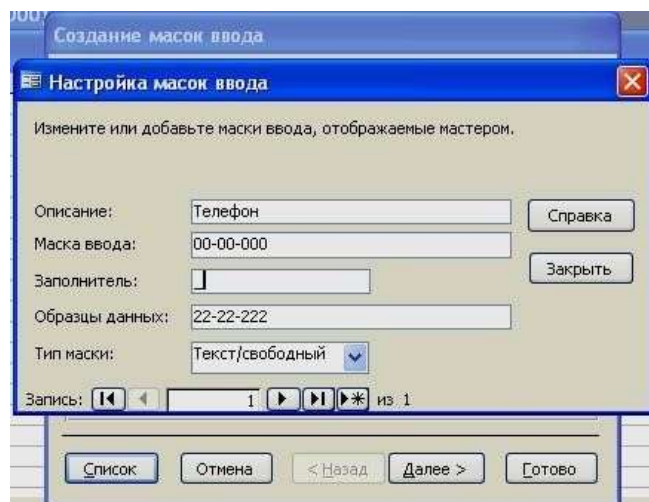


Рис 14 Создание маски ввода

- Нажмите кнопку <Закреть> и из появившегося списка выберите созданную маску ввода для поля *Телефон*.


Примечание 3. Маска ввода создается только для поля, имеющего тип данных – **текстовый!**

Таблица 5. Некоторые допустимые знаки масок ввода

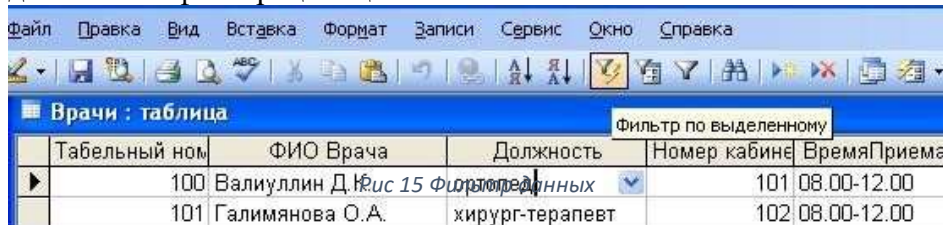
Знак	Описание
0	Цифра (от 0 до 9, ввод обязателен; знаки плюс [+] и минус [-] не допускаются).
9	Цифра или пробел (ввод не обязателен; знаки плюс и минус не допускаются).
#	Цифра или пробел (ввод не обязателен; пустые знаки преобразуются в пробелы, допускаются знаки плюс и минус).
L	Буква (от А до Z или от А до Я, ввод обязателен).
?	Буква (от А до Z или от А до Я, ввод не обязателен).
A	Буква или цифра (ввод обязателен).
&	Любой знак или пробел (ввод обязателен).

Фильтрация данных по полям

В режиме **Таблица** в таблице «*Врачи*» щелкните по записи *Ортопед* поля «Должность»;

- 1) щелкните по кнопке  или выполните команду Записи, Фильтр, Фильтр по выделенному. В таблице останутся только записи о врачах - ортопедах;

2) для отмены фильтрации щелкните по кнопке



The screenshot shows a software window with a menu bar (Файл, Правка, Вид, Вставка, Формат, Записи, Сервис, Окно, Справка) and a toolbar. Below the toolbar is a table titled "Врачи : таблица". The table has columns: Табельный ном, ФИО Врача, Должность, Номер кабинета, and Время Приема. The first row shows "100 Валиуллин Д.Р. 15 Февраль 2015" and "101 Галимянова О.А. хирург-терапевт". A filter button with a funnel icon is visible to the right of the table.

Табельный ном	ФИО Врача	Должность	Номер кабинета	Время Приема
100	Валиуллин Д.Р.	15 Февраль 2015	101	08.00-12.00
101	Галимянова О.А.	хирург-терапевт	102	08.00-12.00



на панели инструментов или выполните команду Записи, Удалить фильтр. В таблице появятся все данные.

Контрольная работа по теме «Нормализация базы данных».

Вариант 1.

1. Правила Армстронга
2. Нормализация
3. Требования второй нормальной формы
4. Функциональная зависимость
5. Аномалия обновления данных

Вариант 2.

1. Свойства математических зависимостей
2. Декомпозиция отношений
3. Требования третьей нормальной формы
4. Полная функциональная зависимость
5. Аномалия удаления данных

Лабораторная работа. «Проектирование и нормализация базы данных»

Цель практической работы

Получить теоретические знания и практические навыки реализации баз данных (БД). Осуществить анализ предметной области. Освоить концептуальное проектирование и научиться определять сущности и атрибуты БД. Научиться разрабатывать инфологическую модель БД в виде ER-диаграмм. Получить теоретические знания и практические навыки при физическом проектировании баз данных (БД). Научиться создавать даталогическую модель БД.

I. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ БАЗЫ ДАННЫХ

1.1. Понятие БД и СУБД

Информационная система - система, реализующая автоматизированный сбор, обработку и манипулирование данными и включающая технические средства обработки данных, программное обеспечение и соответствующий персонал.

Цель любой информационной системы - обработка данных об объектах реального мира. Основой информационной системы является **база данных**. В широком смысле слова **база данных** - это совокупность сведений о конкретных объектах реального мира в какой-либо предметной области.

Под **предметной областью** принято понимать часть реального мира, подлежащего изучению для организации управления и в конечном счете автоматизации, например, предприятие, вуз и т.д.

Создавая базу данных, пользователь стремится упорядочить информацию по различным признакам и быстро производить выборку с произвольным сочетанием признаков. Большое значение при этом приобретает структурирование данных.

Структурирование данных - это введение соглашений о способах представления данных.

Неструктурированными называют данные, записанные, например, в текстовом файле.

Ниже приведен пример неструктурированных и структурированных данных, содержащих сведения о студентах (номер личного дела, фамилию, имя, отчество и год рождения).

Неструктурированные данные:

Личное дело № 16493. Сергеев Петр Михайлович,
дата рождения 1 января 1976 г.; Л/д № 16593, Петрова
Анна
Владимировна, дата рожд. 15 марта 1975 г.; № личн.
дела 16693, д.р. 14.04.76, Анохин Андрей Борисович

Легко убедиться, что сложно организовать поиск необходимых данных, хранящихся в неструктурированном виде.

Структурированные данные:

№ личного	Фамилия	Имя	Отчество	Дата рождения
16493	Сергеев	Петр	Михайлович	01.01.76
16593	Петрова	Анна	Владимировна	15.03.75
16693	Анохин	Андрей	Борисович	14.04.76

В современной технологии баз данных предполагается, что создание базы данных, ее поддержка и обеспечение доступа пользователей к ней осуществляются централизованно с помощью специального программного инструментария - **системы управления базами данных (СУБД)**.

База данных (БД) - это поименованная совокупность данных, отражающая состояние объектов и их отношений в рассматриваемой предметной области.

Объектом называется элемент предметной области, информацию о котором мы сохраняем.

Объект может быть **реальным** (например, человек, изделие; или населенный пункт) и **абстрактным** (например, событие, счет покупателя или изучаемый студентами курс).

Так, в области продажи автомобилей примерами **объектов** могут служить **МОДЕЛЬ АВТОМОБИЛЯ, КЛИЕНТ и СЧЕТ**. На товарном складе - это **ПОСТАВЩИК, ТОВАР, ОТПРАВЛЕНИЕ** и т. д.

Понятие базы данных тесно связано с такими понятиями структурных элементов, как **поле, запись, файл (таблица)** (рис.1).

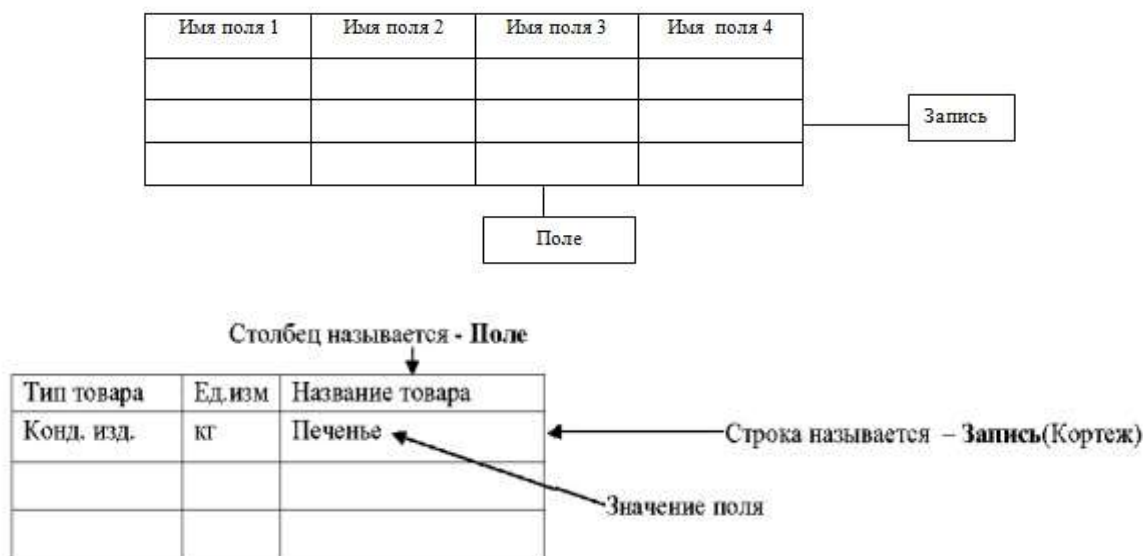


Рис.1 Основные структурные элементы БД

1.2. Структурные элементы базы данных

Поле - элементарная единица логической организации данных, которая соответствует неделимой единице информации - реквизиту. Для описания поля используются следующие характеристики:

- **имя**, например, **Фамилия, Имя, Отчество, Дата рождения**;
- **тип**, например, символьный, числовой, денежный;
- **длина**, например, 15 байт, причем будет определяться максимально возможным количеством символов;
- **точность** для числовых данных, например два десятичных знака для отображения дробной части числа,

Запись - совокупность логически связанных полей.

Экземпляр записи - отдельная реализация записи, содержащая конкретные значения ее полей.

Файл (таблица) - совокупность экземпляров записей одной структуры.

Описание логической структуры записи файла содержит последовательность расположения полей записи и их основные характеристики, как это показано на рис. 2 и 3.

Имя файла					
Поле		Признак ключа	Формат поля		
Имя (обозначение)	Полное наименование		Тип	Длина	Точность (для чисел)
имя 1					
имя N					

Рис. 2. Описание логической структуры записи файла

В структуре записи файла указываются поля, значения которых являются ключами:

первичными (ПК) и внешними (ВК),

Первичный ключ (ПК) - это одно или несколько полей, однозначно идентифицирующих запись. Если первичный ключ состоит из одного поля, он называется **простым**, если из нескольких полей - **составным** ключом.

Внешний ключ (ВК) - это одно или несколько полей, которые выполняют роль поисковых или группировочных признаков. В отличие от первичного, значение внешнего ключа может повторяться в нескольких записях файла, то есть он не является уникальным. Если по значению первичного ключа может быть найден один единственный экземпляр записи, то по внешнему - несколько.

Имя файла: СТУДЕНТ					
Поле		Признак ключа	Формат поля		
Обозначение	Наименование		Тип	Длина	Точность
Номер	№ личного дела	•	Симв	5	
Фамилия	Фамилия студента		Симв	15	
Имя	Имя студента		Симв	10	
Отчество	Отчество студента		Симв	15	
Дата	Дата рождения		Дата	8	

Рис. 3. Описание логической структуры записи файла **СТУДЕНТ**

1.3. Понятие модели данных

Для того, чтобы спроектировать структуру базы данных, необходима исходная информация о предметной области. Желательно, чтобы эта информация была представлена в формализованном виде.

Такое формализованное описание предметной области будем называть **инфологической (infological) моделью предметной области (ИЛМ)** или **концептуальной моделью (КМ)**.

Ядром любой базы данных является модель данных. **Модель данных** представляет собой множество структур данных, ограничений целостности и операций манипулирования данными. С помощью модели данных могут быть представлены объекты предметной области и взаимосвязи между ними.

Модель данных - совокупность структур данных и операций их обработки.

СУБД основывается на использовании **иерархической, сетевой или реляционной модели**, на комбинации этих моделей или на некотором их подмножестве. Самой распространенной моделью данных является - **реляционная**.

1.3.1. Иерархическая модель данных

Иерархическая модель организует данные в виде древовидной структуры

К основным понятиям иерархической структуры относятся: **уровень, элемент (узел), связь**. Дерево представляет собой иерархию элементов, называемых узлами. **Узел** - это совокупность атрибутов данных, описывающих некоторый объект. На самом верхнем уровне иерархии имеется один и только один узел - **корень**. Каждый узел, кроме корня, связан с одним узлом на более высоком уровне, называемым **исходным** для данного узла. Ни один элемент не имеет более одного исходного. Каждый элемент может быть связан с одним или несколькими элементами на более низком уровне. Они называются **порожденными** (рис. 4).

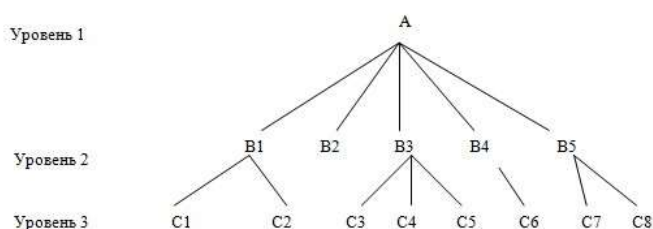


Рис. 4. Графическое изображение иерархической структуры БД

К каждой записи базы данных существует только один (иерархический) путь от корневой записи. Например, как видно из рис. 4, для записи C4 путь проходит через записи A и B3.

1.3.2. Сетевая модель данных

Сетевая модель организует данные в виде сетевой структуры.

Структура называется сетевой, если в отношениях между данными порожденный элемент имеет более одного исходного.

В сетевой структуре при тех же основных понятиях (уровень, узел, связь) каждый элемент может быть связан с любым другим элементом.

На рис. 5 изображена сетевая структура базы данных в виде графа.

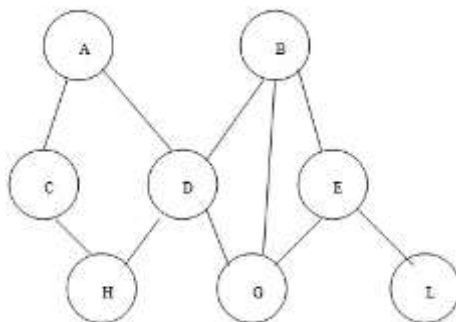


Рис. 5. Графическое изображение сетевой структуры

1.3.3. Реляционная модель данных

Реляционная модель данных является совокупностью взаимосвязанных двумерных таблиц объектов модели.

Например, реляционной таблицей можно представить информацию о студентах, обучающихся в вузе (рис. 6).

№ личного дела	Фамилия	Имя	Отчество	Дата рождения	Группа
16493	Сергеев	Петр	Михайлович	01.01.76	111
16593	Петрова	Анна	Владимировна	15.03.75	112

16693	Анохин	Андрей	Борисович	14.04.76	111
-------	--------	--------	-----------	----------	-----

Рис. 6. Пример реляционной таблицы

Связи между двумя логически связанными таблицами в реляционной модели устанавливаются по равенству значений одинаковых атрибутов этих таблиц.

Каждая реляционная таблица представляет собой двумерный массив и обладает следующими **свойствами**:

- - **каждый элемент таблицы - один элемент данных;**
- - **все столбцы в таблице однородные, т.е. все элементы в столбце имеют одинаковый тип (числовой, символьный и т.д.) и длину;**
- - **каждый столбец имеет уникальное имя;**
- - **одинаковые строки в таблице отсутствуют;**
- - **порядок следования строк и столбцов может быть произвольным.**

При описании реляционной модели часто используют следующие термины:

отношение, кортеж, домен.

Отношения представлены в виде таблиц, строки которых соответствуют записям (кортежам), а столбцы полям, атрибутам отношений (доменам).

Поле, каждое значение которого однозначно определяет соответствующую запись, называется простым ключом (ключевым полем). Если записи однозначно определяются значениями нескольких полей, то такая таблица базы данных имеет **составной ключ**.

В примере, показанном на рис.6, **ключевым полем** таблицы является "№ личного дела".

Между двумя реляционными таблицами могут быть сформированы **связи**. Различные таблицы, могут быть **связаны между собой через общее поле данных**.

На рис. 7 показан пример реляционной модели, построенной на основе отношений: **СТУДЕНТ, СЕССИЯ, СТИПЕНДИЯ**.



Рис.7. Пример реляционной модели

Таблица **СТУДЕНТ** имеет поля: Номер, Фамилия, Имя, Отчество, Дата рождения, Группа;

СЕССИЯ - Номер, Оценка 1, Оценка 2, Оценка 3, Оценка 4, Результат;

СТИПЕНДИЯ - Результат, Процент

Таблицы **СТУДЕНТ** и **СЕССИЯ** имеют совпадающие ключи (**Номер**), что дает возможность легко организовать связь между ними.

Таблица **СЕССИЯ** имеет **первичный ключ Номер** и содержит **внешний ключ Результат**, который обеспечивает ее связь с таблицей **СТИПЕНДИЯ**.

Благодаря имеющимся связям достигаются следующие преимущества:

1. Удастся избежать дублирования информации. Все необходимые данные можно хранить только в одной таблице. Так, например, нет необходимости в таблице **СЕССИЯ** хранить номер группы каждого студента, сдающего экзамены, достаточно задать связь с таблицей **СТУДЕНТ**.

2. В реляционных базах данных легко производить изменения. Если в таблице **СЕССИЯ** изменить какие-нибудь значения, то правильная информация автоматически будет связана с другими таблицами, ссылающимися на первую (например, таблица **СТИПЕНДИЯ**).

3. В нереляционных базах данных сложно передать все имеющиеся зависимости, т.е. связать друг с другом данные из различных таблиц. Реляционная база данных выполняет все эти действия автоматически.

4. В реляционных базах данных удастся легко избежать установления ошибочных связей между различными таблицами данных, а необходимый объем памяти сокращен до минимума.

2. ПРОЕКТИРОВАНИЕ БАЗ ДАННЫХ

2.1. Анализ и описание предметной области БД

Предметной областью называется фрагмент реальности, который описывается или моделируется с помощью БД и ее приложений. В предметной области выделяются информационные объекты – идентифицируемые объекты реального мира, процессы, системы, понятия и т.д., сведения о которых хранятся в БД.

Анализ предметной области выполняется в следующем порядке: название предприятия, цель деятельности предприятия, структура предприятия, информационные потребности пользователей. Для описания ПО следует привести основные решаемые задачи БД. Дать словесное описание процесса функционирования ПО, проанализировать основные хозяйственные операции, которые совершаются в ПО. Привести анализ структуры предприятия, перечислить задачи, решаемые отдельными подразделениями. ***Анализ предметной области целесообразно разбить на три фазы:***

1. анализ концептуальных требований и информационных потребностей;
2. выявление информационных объектов и связей между ними;
3. построение концептуальной модели предметной области и проектирование концептуальной схемы БД.

2.2. Анализ концептуальных требований и информационных потребностей

Требования пользователей к разрабатываемой БД представляют собой список запросов с указанием их интенсивности и объемов данных. Эти сведения разработчики БД получают в диалоге с ее будущими пользователями. Здесь же выясняются требования к вводу, обновлению и корректировке информации. Требования пользователей уточняются и дополняются при анализе имеющихся и перспективных задач.

Рассмотрим примерный состав вопросника при анализе различных предметных областей.

Пример. Предлагается разработать БД для учета студентов вуза.

Анализ предметной области:

1. Сколько студентов учится в вузе? 2. Сколько факультетов и отделений в вузе? 3. Как распределены студенты по факультетам отделений и курсам? 4. Сколько дисциплин читается на каждом курсе по каждой специальности? 5. Как часто обновляется информация в БД? 6. Сколько преподавателей в вузе? 7. Сколько иногородних студентов живет в общежитии, на частных квартирах? 8. Сколько лекционных аудиторий и аудиторий для проведения практических занятий, лабораторий? 9. Какая преемственность существует между читаемыми курсами? 10. Как информация, представленная в п.п. 1-9, используется в настоящее время (расписание занятий, экзаменов, зачетов и т.д.) и как ее собираются использовать? 11. Сколько раз в день, сколько человек и кто пользуются БД?

2.3. Выявление информационных объектов и связей между ними

Вторая фаза анализа предметной области состоит в выборе информационных объектов, задании необходимых свойств для каждого объекта, выявлении связей между объектами, определении ограничений, накладываемых на информационные объекты, типы связей между ними, характеристики информационных объектов- Проанализируем предметную область на примере БД "Видеомагнитофоны".

При выборе информационных объектов постараемся ответить на ряд вопросов:

1. На какие классы можно разбить данные, подлежащие хранению в БД?
2. Какое имя можно присвоить каждому классу данных?
3. Какие наиболее интересные характеристики (с точки зрения пользователя) каждого класса данных можно выделить?
4. Какие имена можно присвоить выбранным наборам характеристик?

Пример. БД "Видеомагнитофоны", рассчитанной на пользователей, которые хотят приобрести данный вид техники.

После беседы с различными пользователями и просмотра каталогов было выяснено, что интерес представляют три информационных объекта: видеомагнитофон, видеоплеер, видеокассета. Рассмотрим наиболее существенные характеристики каждого информационного объекта.

Объект - ВИДЕОМАГНИТОФОН.

Атрибуты - страна-изготовитель, фирма-изготовитель, № модели, телевизионные системы, число кассетных гнезд, ресурс непрерывной работы, система автопоиска, напряжение в сети, наличие таймера, число программ, габаритные размеры, масса, цена в долларах, год выпуска.

Объект - ВИДЕОПЛЕЙЕР.

Атрибуты - страна-изготовитель, фирма-изготовитель, № модели, телевизионные системы, число воспроизводящих головок, ресурс непрерывной работы, напряжение в сети, наличие таймера, габаритные размеры, масса, цена в долларах, год выпуска.

Объект - ВИДЕОКАССЕТА.

Атрибуты - наименование, страна-изготовитель, фирма-изготовитель, тип кассеты, время проигрывания, цена в долларах.

Далее выделим связи между информационными объектами. В ходе этого процесса постараемся ответить на следующие вопросы:

1. Какие типы связей между информационными объектами?
2. Какое имя можно присвоить каждому типу связей?
3. Каковы возможные типы связей, которые могут быть использованы впоследствии?
4. Имеют ли смысл какие-нибудь комбинации типов связей?

Попытаемся задать ограничения на объекты и их характеристики.

Под **ограничением целостности** обычно понимают логические ограничения, накладываемые на данные. Ограничение целостности - это такое свойство, которое мы задаем для некоторого информационного объекта или его характеристики и которое должно сохраняться для каждого их состояния.

Введем следующие ограничения:

1. Значение атрибута "число кассетных гнезд" изменяется от 1 до 2.
2. Значение атрибута "ресурс непрерывной работы" изменяется от 4 до 24.
3. Значение атрибута "напряжение в сети" изменяется от 110 до 240 В.
4. Значение атрибута "число программ" изменяется от 1 до 20 и т.д.

Связи между различными классами объектов.

Помимо классов объектов в ИЛМ отображают связи между различными классами объектов. Такие связи моделируют отношения между объектами различных видов в реальном мире. При отборе связей помещаемых в ИЛМ следует руководствоваться информационными потребностями пользователей базы данных.

Каждая связь характеризуется именем, типом, классом принадлежности и направлением. Имя связи должно быть глагольным оборотом, например «**Принадлежит**», «**Закреплены за**», «**Входит в**» и т.д.

Все информационные объекты предметной области связаны между собой.

Соответствия, отношения, возникающие между объектами предметной области, называются связями. Различаются связи нескольких типов, для которых введены следующие обозначения:

Различают четыре типа связи:

- «один к одному» (1:1);
- «один ко многим» (1:M); ▪ «многие к одному» (M:1)
- «многие ко многим» (M:M).

Рассмотрим эти типы связей на примерах.

Пример. Дана совокупность информационных объектов, отражающих учебный процесс в вузе:

СТУДЕНТ (Номер, Фамилия, Имя, Отчество, Пол, Дата рождения, Группа)

СЕССИЯ (Номер, Оценка 1, Оценка 2, Оценка 3, Оценка 4, Результат)

СТИПЕНДИЯ (Результат, Процент)

ПРЕПОДАВАТЕЛЬ (Код преподавателя, Фамилия, Имя, Отчество)

Связь один к одному (1:1) предполагает, что в каждый момент времени одному экземпляру информационного объекта А соответствует не более одного экземпляра информационного объекта В и наоборот. Рис. 8 иллюстрирует указанный тип отношений.

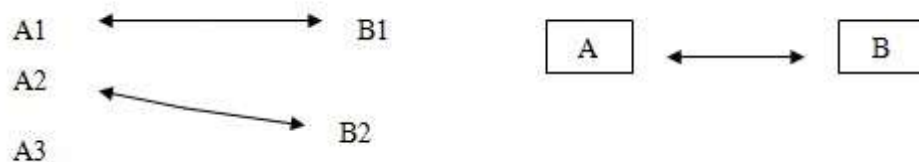


Рис. 8. Графическое изображение реального отношения 1:1

Примером связи 1:1 может служить связь между информационными объектами СТУДЕНТ и СЕССИЯ: СТУДЕНТ <-> СЕССИЯ

Каждый студент имеет определенный набор экзаменационных оценок в сессию.

При связи **один ко многим (1 : М)** одному экземпляру информационного объекта А соответствует 0, 1 или более экземпляров объекта В, но каждый экземпляр объекта В связан не более чем с 1 экземпляром объекта А. Графически данное соответствие имеет вид, представленный на рис. 9.

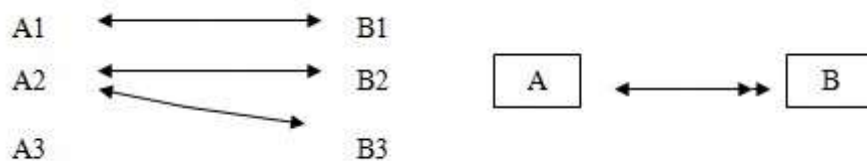


Рис. 9. Графическое изображение реального отношения 1:М

Примером связи 1: М служит связь между информационными объектами СТИПЕНДИЯ и СЕССИЯ: СТИПЕНДИЯ <-->> СЕССИЯ

Установленный размер стипендии по результатам сдачи сессии может повторяться многократно для различных студентов.

Связь **многие ко многим (М:М)** предполагает, что в каждый момент времени одному экземпляру информационного объекта А соответствует 0, 1 или более экземпляров объекта В и наоборот. На рис. 10 графически представлено указанное соответствие.

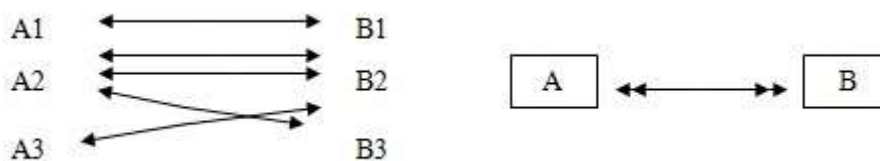


Рис. 10. Графическое изображение реального отношения М : М

Примером данного отношения служит связь между информационными объектами СТУДЕНТ и ПРЕПОДАВАТЕЛЬ: СТУДЕНТ <<-->>ПРЕПОДАВАТЕЛЬ

Один студент обучается у многих преподавателей, один преподаватель обучает многих студентов.

Связь «многие к одному» при создании БД физически обычно организуется путем введения дополнительного поля в таблицу со стороны «много». Это поле называется **внешний ключ**. На рис. 10. код группы - внешний ключ.

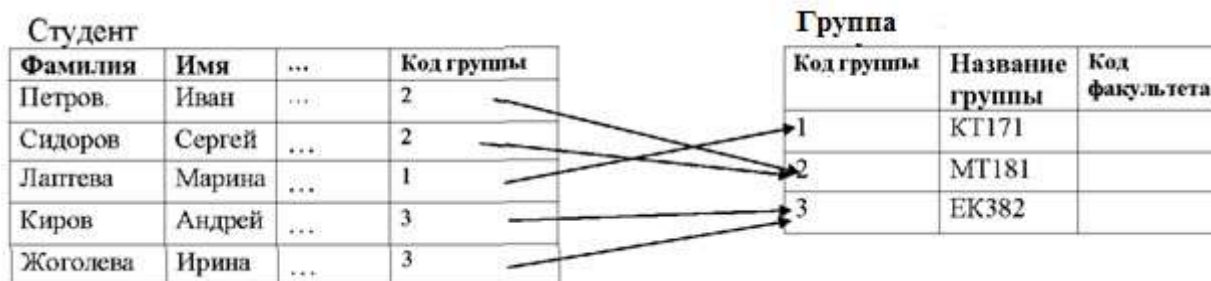


Рис. 10. Введение внешнего ключа в БД

2.4. Построение инфологической (концептуальной модели) предметной области

Заключительная фаза анализа предметной области состоит в проектировании ее инфологической структуры или концептуальной модели.

Концептуальная модель включает описания объектов и их взаимосвязей, представляющих интерес в рассматриваемой предметной области (ПО) и выявляемых в результате анализа данных.

Концептуальная модель применяется для структурирования предметной области с учетом информационных интересов пользователей системы. Она дает возможность систематизировать информационное содержание предметной области, позволяет как бы "подняться вверх" над ПО и увидеть ее отдельные элементы. При этом уровень детализации зависит от выбранной модели.

Концептуальная модель является представлением точки зрения пользователя на предметную область и не зависит ни от программного обеспечения СУБД, ни от технических решений.

Концептуальная модель должна быть стабильной. Могут меняться прикладные программы, обрабатывающие данные, может меняться организация их физического хранения, концептуальная модель остается неизменной или увеличивается с целью включения дополнительных данных.

Одной из распространенных моделей концептуальной схемы является модель «**сущность - связь**» (**ER-моделей** (или ER-диаграмм)). Основными конструкциями данной модели являются сущности и связи.

Под **сущностью** понимают основное содержание объекта ПО, о котором собирают информацию. В качестве сущности могут выступать место, вещь, личность, явление. Экземпляр **сущности** - конкретный объект.

Например:

сущность (объект) - служащий
 экземпляр сущности - Иванов А.В.;
 сущность (объект) - институт
 экземпляр сущности - МГУ.

Сущность принято определять **атрибутами** - поименованными характеристиками.

Например:

сущность - служащий атрибуты: ФИО, год рождения, адрес, образование и т.д.

Чтобы задать атрибут в модели, ему надо присвоить имя и определить область допустимых значений. Одно из назначений атрибута - идентифицировать сущность.

Связь определяет отношения между сущностями. **Типы связей: один к одному, один ко многим, многие ко многим.**

При построении модели «сущность - связь» используют **графические диаграммы**. При этом обозначают: сущности - прямоугольниками, атрибуты - овалами, связи - ромбами, см. рис.11.

На практике приходится строить несколько вариантов моделей, из которых выбирается одна, наиболее полно отображающая предметную область.

Классом объектов называют совокупность объектов, обладающих одинаковым набором свойств. Например, для объектов класса «СТУДЕНТ» таким набором свойств являются: «ГОД_РОЖДЕНИЯ», «ПОЛ» и др.

Объекты могут быть реальными, как названный выше объект «СТУДЕНТ», и абстрактными, как, например, «ПРЕДМЕТЫ», которые изучают студенты.

Пример. Спроектировать БД "Сессия". База данных должна выдавать оперативную информацию об успеваемости студентов на факультетах в семестре. Результатами сессии считать только экзамены.

По сути дела и БД исходя из формулировки задания можно выделить лишь одно приложение. Речь идет об успеваемости студентов разных факультетов по тем или иным дисциплинам. Более конкретно речь идет о выдаче справок по результатам сессии каждого студента, учебной группы, курса, факультета, а также об автоматизированном составлении ведомости

Выберем следующие сущности:

ИНСТИТУТ, ФАКУЛЬТЕТ, СТУДЕНТ, ПРЕПОДАВАТЕЛЬ, ДИСЦИПЛИНА.

В данном примере можно выделить сущность ЭКЗАМЕН или ВЕДОМОСТЬ, но можно не выделять, а сформировать ведомость из имеющихся данных посредством связей.

Зададим каждую сущность набором атрибутов:

ИНСТИТУТ (название, подчиненность, адрес, телефон, ФИО ректора)

ФАКУЛЬТЕТ (название, код специальности, данные о кафедрах, число выпускников, декан).

СТУДЕНТ (ФИО, группа, курс, номер текущего семестра, пол).

ПРЕПОДАВАТЕЛЬ (ФИО, должность, звание, кафедра, стаж).

ДИСЦИПЛИНА (название, число часов, код дисциплины, виды занятий, число читаемых семестров, номера текущих семестров, на каких курсах преподается)

В каждом наборе атрибутов, характеризующих сущность, необходимо выбрать ключевые атрибуты, т.е. атрибуты, делающие сущность уникальной. При задании атрибутов ключевые атрибуты подчеркивались.

Определим связи между сущностями.

Название связи **Связи между сущностями**

учится	студент, факультет
изучает	студент, дисциплина
имеет	институт, факультет
работает	преподаватель, факультет
преподает	преподаватель, дисциплина
экзамен	студент, дисциплина, преподаватель

После выбора сущностей, задания атрибутов и анализа связей можно перейти к проектированию информационной (концептуальной) схемы БД.

Концептуальная схема БД "Успеваемость» представлена на рис.11 (атрибуты сущностей на диаграмме не показаны).

Рассмотрим некоторые **ограничения** в рассматриваемом примере:

1. Значение атрибута "телефон" (сущность - ИНСТИТУТ) задается целым положительным шестизначным числом.
2. Значение атрибута "код факультета" (сущность - ФАКУЛЬТЕТ) лежит в интервале 1-10.
3. Значение атрибута "курс" (сущность - СТУДЕНТ) лежит в интервале 1 - 6
4. Значение атрибута "семестр" (сущность - СТУДЕНТ, ДИСЦИПЛИНА) лежит в интервале 1-12.
5. Значение атрибута "число часов" (сущность - ДИСЦИПЛИНА) лежит в интервале 1-300.
6. Одному студенту может быть приписана только одна группа.
7. Один студент может учиться только на одном факультете.
8. Один студент в семестре сдает от 3 до 5 дисциплин

9. Один студент изучает в семестре от 6 до 12 дисциплин.
10. Одному преподавателю приписывается только одна кафедра.
11. Один студент может пересдавать одну дисциплину не более трех раз.
12. Ключи: название института, название факультета, ФИО и группа студента, ФИО и кафедра преподавателя, название дисциплины.

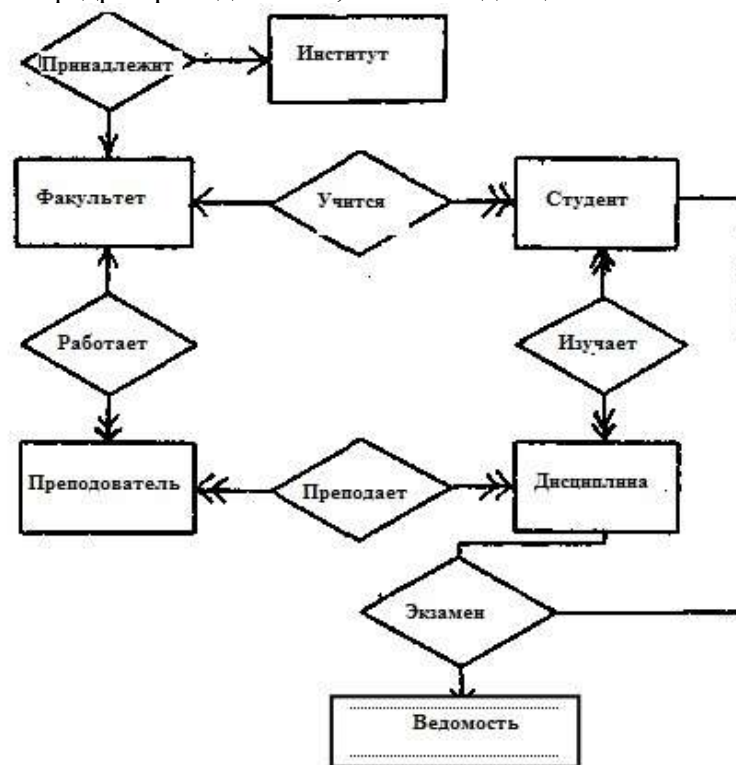


Рис 11. Концептуальная схема БД «Успеваемость»

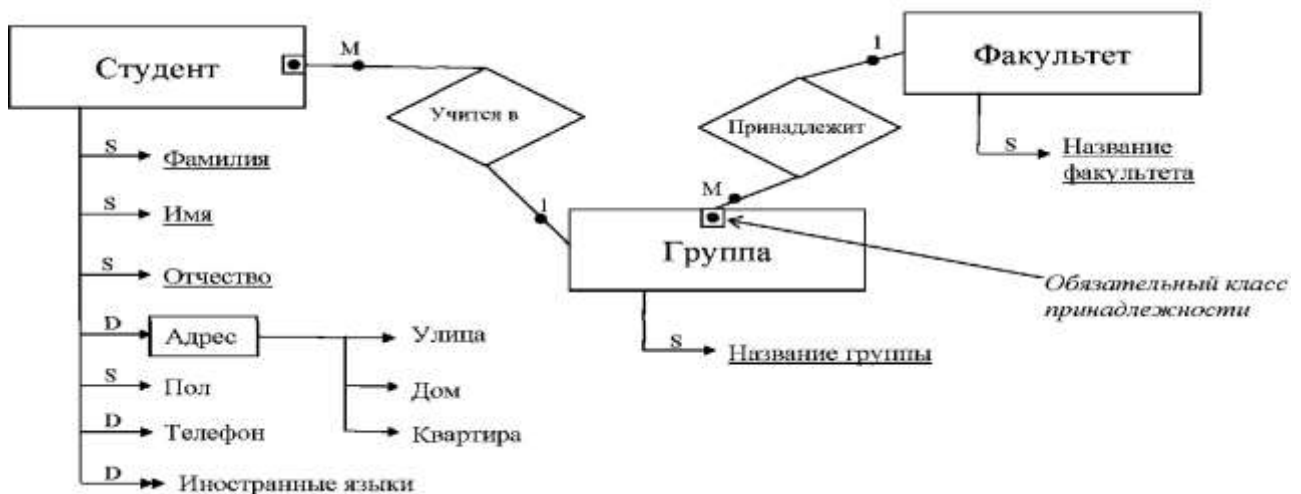


Рис. 12. Пример построения инфологической модели данных.

3. ФИЗИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

3.1. Даталогическое проектирование

Инфологическая модель является исходной для построения *даталогической* модели БД и служит промежуточной моделью для специалистов предметной области (для которой создается банк данных (БнД)) и администратора БД в процессе проектирования и разработки конкретной БД.

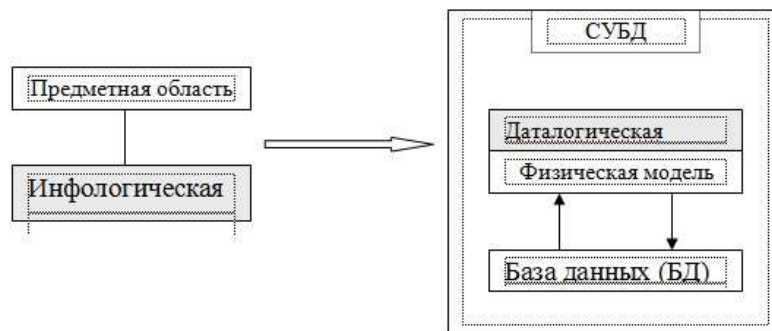


Рис. 13. Структура проектирования БД

Под *дательгической* понимается модель, отражающая логические взаимосвязи между элементами данных безотносительно их содержания и физической организации. При этом дательгическая модель разрабатывается с учетом конкретной реализации СУБД, также с учетом специфики конкретной предметной области на основе ее инфологической модели.

Инфологическая модель предметной области строится первой. Предварительная инфологическая модель строится еще на предпроектной стадии и затем уточняется на более поздних стадиях проектирования баз данных. Затем на ее основе строятся концептуальная (логическая), внутренняя (физическая) и внешняя модели.

Конечным результатом дательгического проектирования является описание логической структуры базы данных на языке программирования. Однако если проектирование выполняется «вручную», то для большей наглядности сначала строится схематическое графическое изображение структуры базы данных. При этом должно быть обеспечено однозначное соответствие между конструкциями языка описания данных и графическими обозначениями информационных единиц и связей между ними.

Графическое представление используется и при автоматизированном проектировании структуры базы данных как интерфейсное средство общения с проектировщиком, и при документировании проекта.

Спроектировать логическую структуру базы данных означает определить все информационные единицы и связи между ними, задать их имена; если для информационных единиц возможно использование разных типов, то определить их тип. Следует также задать некоторые количественные характеристики, например, длину поля.

3.2. Описание дательгической модели.

Дательгическая модель представляет собой описание базы данных, выполненное в терминах используемой СУБД. Наиболее часто при разработках баз данных применяют реляционные СУБД. Для СУБД этого типа дательгическая удобно представить в виде набора таблиц специальной формы (табл. 1.4.).

Такая таблица составляется для каждого отношения, используемого в базе данных. Отношения в базе соответствуют классам объектов из инфологической модели. Кроме того, отношения могут представлять некоторые связи предметной области.

Каждой таблице нужно поставить в соответствие ее **ключи**. Схема ключа представляет собой перечисление атрибутов отношения, составляющих ключ.

Различают простые и составные ключи. **Простой ключ** строится на основе одного атрибута. **Составной ключ** строится на базе использования нескольких атрибутов.

Ключи принято разделять на **первичные, внешние** и **вспомогательные**.

Первичный индекс должен быть только один для каждой таблицы. Значения атрибутов, используемых для формирования первичного ключа, должны быть уникальными для каждой записи в таблице. Значения первичного ключа уникально идентифицируют каждую запись. Не может быть двух записей в таблице с одинаковым значением первичного ключа. Например, в качестве первичного ключа для отношения «Сотрудники» можно выбрать атрибут «Табельный номер», значение которого является уникальным для каждой записи о сотруднике.

Внешние ключи используются для реализации связей типа **1:М** между отношениями. Внешний ключ строится для отношения, находящегося на стороне «много» связи 1:М. Для

каждого такого отношения на даталогической модели должен быть показан внешний ключ. Внешний ключ всегда должен иметь соответствующий ему первичный ключ отношения, находящегося на стороне «один» связи типа 1:M.

Для связанных ключа «внешний - первичный» соединяются на схеме даталогической модели ломаной линией. Например, если в таблице «Сотрудники» имеется атрибут «Шифр категории», то этот атрибут можно использовать в качестве внешнего ключа для связи с таблицей «Категории». В последней таблице должен быть первичный ключ по полю «Шифр категории». Внешних ключей может быть несколько для одной таблицы.

Следует отметить, что первичные и внешние ключи строятся как правило на основе целочисленных атрибутов, а не атрибутов, имеющих строковый или вещественный тип.

Кроме первичных и внешних ключей часто используют **вспомогательные индексы**. Они применяются для реализации связей, получения нужного упорядочения при выводе на экран и создании отчетов и т.д. В каждом случае использования вспомогательного индекса, его необходимость должна быть обоснована. Применение большого количества индексов замедляет работу СУБД, т.к. операции над записями отношения требуют корректировки всех индексов.

II. ЗАДАНИЕ ВЫПОЛНЕНИЯ ПРАКТИЧЕСКОЙ РАБОТЫ

Практическая работа выполняется письменно и в конце занятия сдается на проверку. После проверки будет выставлена оценка.

2.1. Выбор задания

Выбрать из таблицы «**Варианты заданий для лаб. работы №1.doc**» вариант задания, соответствующий **номеру студента в списке учебной группы**. Для всех последующих практических работ вариант остается неизменным. Каждому студенту предоставляется свой вариант предметной области (ПО), который он будет использовать в процессе выполнения всех практических работ.

2.2. Анализ предметной области.

На основании выбранного варианта привести: название предприятия, цель деятельности предприятия, структура предприятия, информационные потребности пользователей (кратко).

2.3. Описание основных сущностей ПО.

Здесь следует привести описание основных сущностей (объектов) ПО. Отбор сущностей производится на основе анализа информационных потребностей. Необходимо привести таблицы описания сущностей (сущностей должно быть не менее 3-х) Таблица 1.1.

Список сущностей предметной области.

N п.п.	Наименование сущности	Краткое описание

Здесь же приводится отбор атрибутов (не менее 5-ти) для каждого экземпляра сущности. Отбираются только те атрибуты сущностей, которые необходимы для формирования ответов на регламентированные и непредусмотренные запросы. Для каждого объекта следует привести таблицы его атрибутов.

Таблица 1.2. Список атрибутов.

N п.п.	Наименование атрибута	Краткое описание

На основе анализа информационных запросов следует выявить связи между сущностями. Для выявленных связей также нужно заполнить таблицу 1.3. Таблица 1.3. Список связей ПО.

N п.п.	Наименование связи	Сущности, участвующие в связи	Краткое описание
-----------	--------------------	-------------------------------	------------------

2.4. Построение инфологической модели.

На основании ранее выбранного варианта и таблиц 1.1-1.3:

- описать классы объектов (сущностей) и их свойства,
- расставить существующие связи между ними,
- на основании табл. 1.3. в письменной форме обосновать типы связей (1:1, 1:M и т.д.).

При графическом построении ИЛМ следует придерживаться единого масштаба для всей схемы. Все прямоугольники, обозначающие классы объектов, должны быть одного размера. Аналогично, все ромбы с именами связей также должны иметь одинаковый размер. **2.5.**

Построение даталогической модели.

На основании ранее выбранного варианта и таблиц 1.1-1.3, инфологической модели и нормализации БД необходимо:

- провести соответствие ключей для каждой таблицы 1.1-1.3,
- заполнить для каждой таблицы БД форму, согласно табл. 1.4.

Таблица 1.4. Структура таблицы для даталогической модели.

N п.п.	Наименование реквизита	Идентификатор	Тип	Длина	Формат изображения	Ограничения и комментарий

III. СОДЕРЖАНИЕ ОТЧЕТА

1. Название и цель работы.
2. Словесный и схематический анализ предметной области (ПО), включая схему структуры предприятия.
3. Заполненные таблицы 1.1 - 1.3. с описанием основных сущностей ПО.
4. Инфологическая модель БД, согласно варианту.
5. Обоснование типов связи в инфологической модели данных.
6. Даталогическая модель БД (табл. 1.4.).

IV. ПРИМЕР ОФОРМЛЕНИЯ

Пример. Разработать базу данных «Учеба студентов».

Решение.

Шаг первый. Анализ предметной области.

Студенты учатся на одном из факультетов, возглавляемом деканатом, в функции которого входит контроль за учебным процессом. В учебном процессе участвуют преподаватели кафедр, административно относящиеся к одному из факультетов. Каждому факультету могут принадлежать несколько кафедр. Студенты кафедр организованные в группы.

Преподаватели кафедр характеризуются фамилией именем и отчеством, должностью, научным званием, ставкой и стажем работы, адресом проживания, возрастом.

Каждая кафедра читает определенный набор закрепленных за ней дисциплин. Каждая дисциплина характеризуется своим полным названием, указанием общего количества часов и формы контроля (зачет, экзамен).

В конце каждого семестра составляется экзаменационно-зачетные ведомости, в которых указываются дисциплины и для каких групп проводится форма контроля, фамилия преподавателя и учебный год и семестр. В каждой такой ведомости составляется список студентов и выставляется оценка.

Шаг второй. Описание основных сущностей ПО.

В результате проведенного анализа предметной области базы данных «Учеба студентов» легко перечислить основные сущности этой БД. Так как на физическом уровне сущности соответствует таблица, то просто перечислим основные таблицы БД.

В реляционную модель проектированной БД будут входить следующие таблицы (сущности): Факультет, Кафедра, Преподаватели, Группы, Студенты, Дисциплины, Ведомости.

Список сущностей.

№	Название	Назначение
1	Факультет	Описание факультета и его деканата
2	Кафедра	Описание кафедры
3	Преподаватели	Описание состава сотрудников кафедр
4	Группы	Перечень групп, закрепленных за каждой кафедрой
5	Студенты	Перечень студентов каждой группы
6	Дисциплины	Перечень дисциплин, закрепленных за каждой кафедрой
7	Ведомости	Экзаменационно-зачетные ведомости с перечнем студентов и их оценками
8	Подчиненная ведомость	Это таблица внутри таблицы ведомости. Отражает связь одно-многим. Так как каждая ведомость выписывается каждой конкретной группе, а студентов в ней много.

Для каждой таблицы (сущности) приведем описание ее атрибутов. Атрибут на физическом уровне – это колонки таблицы и выражает определенное свойство объекта.

Список атрибутов таблицы «Факультеты»

Ключевое поле	Название	Назначение
ПК (первичный ключ)	Код факультета	Ключевое поле, предназначенное для однозначной идентификации каждой записи в таблице. Представляет собой первичный ключ. Это уникальное значение, соответствующее каждому факультету. Это целое число. Т.е. для идентификации каждого факультета будет применяться не названия самих факультетов, а определенный номер. Этот номер может быть случайным целым числом или счетчик по порядку.
	Название факультета	
	ФИО декана	
	Номер комнаты деканата	
	Телефон деканата	

Список атрибутов таблицы «Кафедра»

Ключевое поле	Название	Назначение
ПК (первичный ключ)	Код кафедры	Ключевое поле. Представляет собой первичный ключ. Это уникальное значение, соответствующее каждой кафедре. Однако для идентификации каждой кафедры первичного ключа недостаточно, так как каждая кафедра принадлежит определенному факультету. Для этого будем использовать внешний ключ.

ВК (внешний ключ)	Код факультета	Внешний ключ – это атрибут отношения, который является первичным ключом другого отношения. В нашем случае это атрибут таблицы факультеты. С помощью внешнего ключа будет определено к какому факультету принадлежит каждая кафедра.
	Название кафедры	
	ФИО заведующего	
	Номер комнаты кафедры	
	Телефон кафедры	

Список атрибутов таблицы «Преподаватели»

Ключевое поле	Название	Назначение
ПК (первичный ключ)	Код преподавателя	Ключевое поле. Представляет собой первичный ключ. Это уникальное значение, соответствующее каждому преподавателю. Это например, может быть его табельный номер. Однако для идентификации каждого преподавателя первичного ключа недостаточно, так как каждый сотрудник принадлежит определенной кафедре. Для этого будем использовать внешний ключ.
ВК (внешний ключ)	Код кафедры	С помощью данного внешнего ключа будет определено к какой кафедре принадлежит каждый преподаватель.
	ФИО	
	должность	Ассистент, доцент, профессор, ст. преподаватель
	научное звание	К.т.н., проф., магистр, ст.н.с., м.н.с.
	ставка	
	стаж работы,	
	адрес проживания	
	возраст	

Список атрибутов таблицы «Группы»

Ключевое поле	Название	Назначение
ПК (первичный ключ)	Код группы	Ключевое поле. Представляет собой первичный ключ. Это уникальное значение, соответствующее каждой группе. Однако для идентификации каждой группы первичного ключа недостаточно, так как каждая группа принадлежит определенной кафедре. Для этого будем использовать внешний ключ.
ВК (внешний ключ)	Код кафедры	С помощью данного внешнего ключа будет определено к какой кафедре принадлежит каждая группа.
	Номер группы	
	Год поступления	

	Курс обучения	
--	---------------	--

Список атрибутов таблицы «Студенты»

Ключевое поле	Название	Назначение
ПК (первичный ключ)	Код студента	Ключевое поле. Представляет собой первичный ключ. Это уникальное значение, соответствующее каждому студенту. Однако для идентификации каждого студента первичного ключа недостаточно, так как каждый студент принадлежит определенной группе. Для этого будем использовать внешний ключ.
ВК (внешний ключ)	Код группы	С помощью данного внешнего ключа будет определено к какой группе принадлежит каждый студент.
	ФИО	
	Год рождения	
	Адрес проживания	

Список атрибутов таблицы «Дисциплины»

Ключевое поле	Название	Назначение
ПК (первичный ключ)	Код дисциплины	Ключевое поле. Представляет собой первичный ключ. Это уникальное значение, соответствующее каждой дисциплине. Однако для идентификации каждой дисциплины первичного ключа недостаточно, так как каждая дисциплина принадлежит определенной кафедре. Для этого будем использовать внешний ключ.
ВК (внешний ключ)	Код кафедры	С помощью данного внешнего ключа будет определено к какой кафедре принадлежит каждая дисциплина.
	Название дисциплины	
	Расчасовка	
	Форма контроля	

Список атрибутов таблицы «Ведомости»

Ключевое поле	Название	Назначение
ПК (первичный ключ)	Код ведомости	Ключевое поле. Представляет собой первичный ключ. Это уникальное значение, соответствующее каждой учебной ведомости. Однако для идентификации каждой ведомости первичного ключа недостаточно, так как каждая ведомость выписывается для определенной учебной группы по определенной дисциплине и преподавателя. Для этого будем использовать внешние ключи.

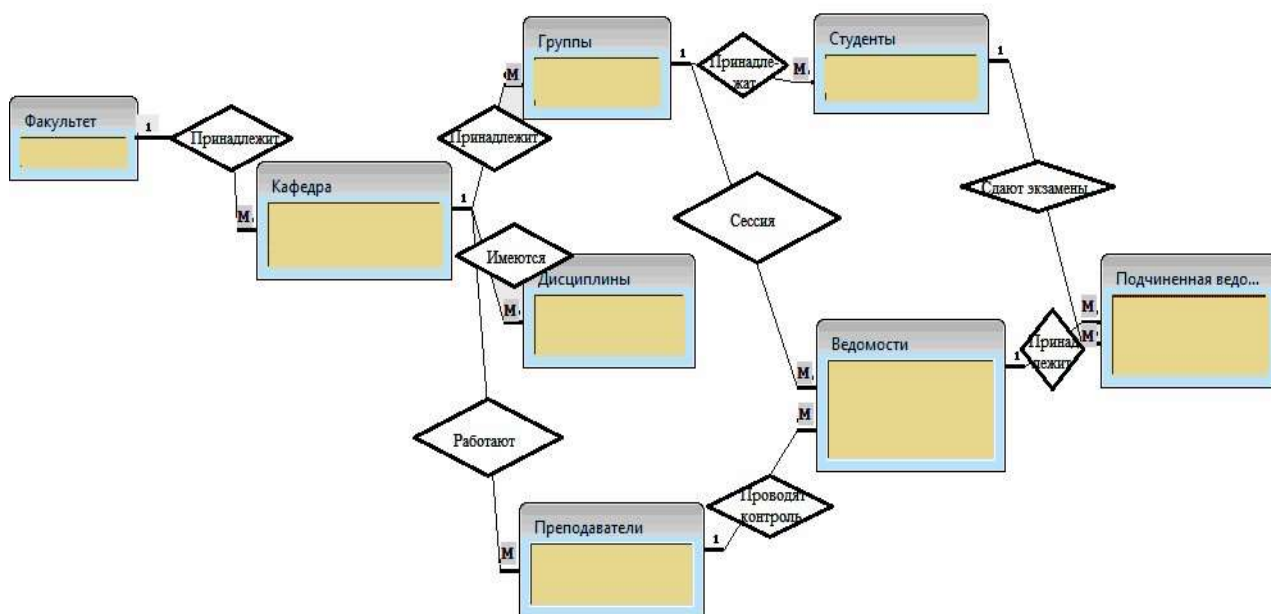
ВК (внешний ключ)	Код группы	С помощью данного внешнего ключа будет определено для какой группы выписывается ведомость.
ВК (внешний ключ)	Код дисциплины	С помощью данного внешнего ключа будет определено для какой дисциплины выписывается ведомость.
ВК (внешний ключ)	Код преподавателя	С помощью данного внешнего ключа будет определено какому преподавателю выписывается ведомость.
	Учебный год	
	Семестр	

Список атрибутов таблицы «Подчиненная таблица Ведомости»

Ключевое поле	Название	Назначение
ПК (первичный ключ)	Код под_ведомости	Ключевое поле. Представляет собой первичный ключ. Это уникальное значение, соответствующее каждой подведомости. Однако для идентификации каждой подведомости первичного ключа недостаточно, так как каждая подведомость принадлежит определенной ведомости. Для этого будем использовать внешний ключ.
ВК (внешний ключ)	Код ведомости	С помощью данного внешнего ключа будет осуществлена связь с таблицей ведомости.
ВК (внешний ключ)	Код студента	С помощью данного внешнего ключа будет определен студент
	Оценка	

Шаг третий. Построение инфологической модели.

Инфологическую модель лучше представить графически, где будут изображены все таблицы и связи между ними. В нашем случае схема связей представлена на рисунке.



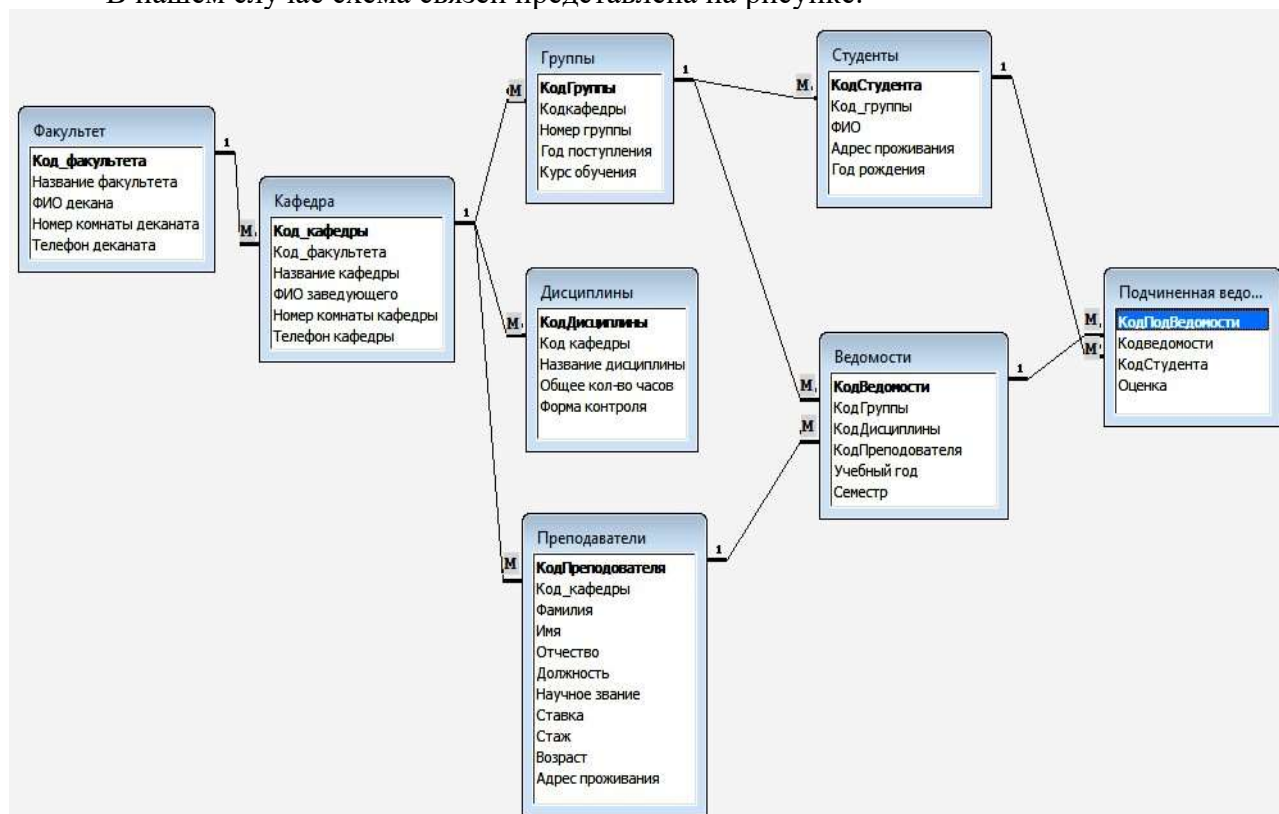
Для выявленных связей заполним таблицу **Список связей.**

№	Название связи	Сущности, участвующие в связи	Назначение
1	1:M	Факультет-Кафедра	Одному факультету могут принадлежать несколько кафедр
2	1:M	Кафедра - Группа	Одной кафедре может принадлежать несколько групп
3	1:M	Кафедра - Дисциплины	Одной кафедре могут принадлежать несколько читаемых дисциплин
4	1:M	Кафедра - Преподаватели	На одной кафедре работает более одного преподавателя
5	1:M	Группа-Студенты	В каждой группе учится множество студентов
6	1:M	Группа - Ведомость	Каждой группе выписывают несколько ведомостей
7	1:M	Дисциплины - Ведомость	Ведомость выписывается из множества дисциплин
8	1:M	Преподаватели - Ведомость	Ведомость выписывается конкретному преподавателю
9	1:M	Ведомость-Подчиненная ведомость	Подчиненная ведомость принадлежит одной конкретной ведомости
10	1:M	Студенты-Подчиненная ведомость	В подчиненной ведомости перечислены все студенты группы

Шаг четвертый. Построение даталогической модели БД.

Даталогическая модель отражается графически в виде схемы базы данных, где указываются имена сущностей, их атрибуты и связи между сущностями.

В нашем случае схема связей представлена на рисунке.



Даталогическая модель БД представляется в виде набора таблиц специальной формы, в которых указываются наименование атрибута, идентификатор, тип, длина, формат, ограничения.

Таблица «Факультеты»

№	Название	Идентификатор	Тип	Не пусто	Ограничение
1	Код факультета	Kod_fakulteta	Числовой	Да	ПК (первичный ключ)
2	Название факультета	Name_fakulteta	Текстовый	Нет	
3	ФИО декана	FIO	Текстовый	нет	
4	Номер комнаты деканата	N_komnatu_dekanata	Текстовый	Нет	Например, 123/a
5	Телефон деканата	Telefon_dekanata	Текстовый	Нет	Например, 41-69-99

Список атрибутов таблицы «Кафедра»

№	Название	Идентификатор	Тип	Не пусто	Ограничение
1	Код кафедры	Kod_kafedru	Числовой	Да	ПК (первичный ключ)
2	Код факультета	Kod_fakulteta	Числовой	Да	ВК (внешний ключ)
3	Название кафедры	Name_kafedru	Текстовый		
4	ФИО заведующего	FIO	Текстовый	нет	
5	Номер комнаты кафедры	N_komnatu_kafedru	Текстовый	Нет	Например, 123/a
6	Телефон кафедры	Telefon_kafedru	Текстовый	Нет	Например, 41-69-99

Список атрибутов таблицы «Преподаватели»

№	Название	Идентификатор	Тип	Не пусто	Ограничение
1	Код преподавателя	Kod_prepodavately	Числовой	Да	ПК (первичный ключ)
2	Код кафедры	Kod_kafedru	Числовой	Да	ВК (внешний ключ)
3	ФИО	FIO	Текстовый	Нет	
4	должность	Dolgnost	Текстовый	Нет	
5	научное звание	Zvanie	Текстовый	Нет	
6	ставка	Stavka	Числовой	Нет	Вещественное число Например, 0.5, 0.75, 1

7	стаж работы,	Stag	Числовой	Нет	Вещественное число
8	адрес проживания	Address	Текстовый	Нет	
9	возраст	Vozrast	Числовой	нет	

Список атрибутов таблицы «Группы»

№	Название	Идентификатор	Тип	Не пусто	Ограничение
1	Код группы	Kod_grupu	Числовой	Да	ПК (первичный ключ)
2	Код кафедры	Kod_kafedru	Числовой	Да	ВК (внешний ключ)
3	Номер группы	N_grupu	Текстовый	Нет	Например, МТ-461
4	Год поступления	God_post	Числовой	нет	
5	Курс обучения	Kurs	Числовой	Нет	Вычисляемое поле, как разность между текущей датой и годом поступления

Список атрибутов таблицы «Студенты»

№	Название	Идентификатор	Тип	Не пусто	Ограничение
1	Код студента	Kod_studenta	Числовой	Да	ПК (первичный ключ)
2	Код группы	Kod_grupu	Числовой	Да	ВК (внешний ключ)
3	ФИО	FIO	Текстовый	Нет	
4	Год рождения	God_rogdeniya	Числовой	нет	
5	Адрес проживания	Address	Текстовый	Нет	

Список атрибутов таблицы «Дисциплины»

№	Название	Идентификатор	Тип	Не пусто	Ограничение
1	Код дисциплины	Kod_disciplinu	Числовой	Да	ПК (первичный ключ)
2	Код кафедры	Kod_kafedru	Числовой	Да	ВК (внешний ключ)
3	Название дисциплины	Name_dis	Текстовый	Нет	
4	Расписание	Raschasovka	Числовой	нет	
5	Форма контроля	Kontrol	Текстовый	Нет	Два значения – экзамен или зачет

Список атрибутов таблицы «Ведомости»

№	Название	Идентификатор	Тип	Не пусто	Ограничение
1	Код ведомости	Kod_vedomosti	Числовой	Да	ПК (первичный ключ)
2	Код группы	Kod_grupu	Числовой	Да	ВК (внешний ключ)

3	Код дисциплины	Kod_disciplinu	Числовой	Да	ВК (внешний ключ)
4	Код преподавателя	Kod_prepodavately	Числовой	Да	ВК (внешний ключ)
5	Учебный год	God	Числовой	Нет	
6	Семестр	Semester	Числовой	Нет	Диапазон от 1-10

Список атрибутов таблицы «Подчиненная таблица Ведомости»

№	Название	Идентификатор	Тип	Не пусто	Ограничение
1	Код под_ведомости	Kod_pod_vedomopsti	Числовой	Да	ПК (первичный ключ)
2	Код ведомости	Kod_edomopsti	Числовой	Да	ВК (внешний ключ)
3	Код студента	Kod_studenta	Числовой	Да	ВК (внешний ключ)
4	Оценка	Osenka	Числовой	Нет	Диапазон от 0-12

Таблица 1. Варианты заданий для практической работы №1

№ варианта	Условие
Вариант №1	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. <p>БД – успеваемость студентов ВУЗА. БД состоит из следующих таблиц: факультеты, кафедры, учебные группы, студенты, ведомости успеваемости.</p> <p>Таблица факультеты имеет следующие атрибуты: название факультета, ФИО декана, номер комнаты, номер корпуса, телефон.</p> <p>Таблица кафедры имеет следующие атрибуты: название кафедры, факультет, ФИО заведующего, номер комнаты, номер корпуса, телефон, кол-во преподавателей.</p> <p>Таблица учебные группы имеет следующие атрибуты: название группы, год поступления, курс обучения, кол-во студентов в группе.</p> <p>Таблица студенты имеет следующие атрибуты: студента, фамилия, имя, отчество, группа, год рождения, пол, адрес, город, телефон.</p> <p>Таблица ведомости успеваемости имеет следующие атрибуты: группа, студент, предмет, оценка.</p>

Вариант №2	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. БД – информационная система супермаркета. БД состоит из следующих таблиц: отделы, сотрудники, товары, продажа товаров, должности. <p>Таблица отделы имеет следующие атрибуты: название отдела, кол-во прилавков, кол-во продавцов, номер зала.</p> <p>Таблица сотрудники имеет следующие атрибуты: фамилия, имя, отчество, отдел, год рождения, год поступления на работу, стаж, должность, пол, адрес, город, телефон.</p> <p>Таблица должности имеет следующие атрибуты: название должности, сумма ставки.</p> <p>Таблица товары имеет следующие атрибуты: название товара, отдел, страна производитель, условия хранения, сроки хранения .</p> <p>Таблица продажа товаров имеет следующие атрибуты: сотрудника являющегося продавцом, товара дата, время, кол-во, цена, сумма.</p>
Вариант №3	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. БД – информационная система военного округа. БД состоит из следующих таблиц: места дислокации, вид войск, части, роты, личный состав. <p>Таблица вид войск имеет следующие атрибуты: название.</p> <p>Таблица места дислокации имеет следующие атрибуты: страна, город, адрес, занимаемая площадь.</p> <p>Таблица части имеет следующие атрибуты: номер части, место дислокации, вид войск, кол-во рот.</p> <p>Таблица роты имеет следующие атрибуты: название роты, кол-во служащих.</p> <p>Таблица личный состав имеет следующие атрибуты: фамилия, рота, должность, год рождения, год поступления на службу, выслуга лет, награды, участие в военных мероприятиях.</p>
Вариант №4	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. <p>БД – информационная система библиотеки. БД состоит из следующих таблиц: библиотеки, фонд библиотеки, тип литературы, сотрудники, пополнение фонда.</p> <p>Таблица библиотеки имеет следующие атрибуты: название, адрес, город.</p>

	<p>Таблица фонд библиотеки имеет следующие атрибуты: название фонда, библиотека, кол-во книг, кол-во журналов, кол-во газет, кол-во сборников, кол-во диссертаций, кол-во рефератов.</p> <p>Таблица тип литературы имеет следующие атрибуты: название типа. Таблица сотрудники имеет следующие атрибуты: фамилия сотрудника, библиотека, должность, год рождения, год поступления на работу, образование, зарплата.</p> <p>Таблица пополнение фонда имеет следующие атрибуты: фонд, сотрудник, дата, название источника литературы, тип литературы, издательство, дата издания, кол-во экземпляров.</p>
Вариант №5	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. <p>БД – информационная система туристического агентства. БД состоит из следующих таблиц: пансионаты, туры, клиенты, путевки, вид жилья. Таблица пансионаты имеет следующие атрибуты: название пансионата, адрес, город, страна, телефон, описание территории, кол-во комнат, наличие бассейна, наличие медицинских услуг, наличие спа-салона, уровень пансионата, расстояние до моря.</p> <p>Таблица вид жилья имеет следующие атрибуты: название (дом, бунгало, квартира, 1-я комната, 2-я комната и т.д.), категория жилья (люкс, полулюкс, и т.д.), пансионат, описание условий проживания, цена за номер в сутки.</p> <p>Таблица туры имеет следующие атрибуты: название тура (Европа, средняя Азия, тибет и т.д.), вид транспорта, категория жилья на ночь (гостиница, отель, палатка и т.д.), вид питания (одноразовое, двухразовое, трехразовое, завтраки), цена тура в сутки.</p> <p>Таблица клиенты имеет следующие атрибуты: фамилия, имя, отчество, паспортные данные, дата рождения, адрес, город, телефон.</p> <p>Таблица путевки имеет следующие атрибуты: клиент, пансионата, вид жилья, дата заезда, дата отъезда, наличие детей, наличие мед. страховки, кол-во человек, цена, сумма.</p>
Вариант №6	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. <p>БД – информационная система автопредприятия города. БД состоит из следующих таблиц: автотранспорт, водители, маршруты, обслуживающий персонал, гаражное хозяйство.</p> <p>Таблица автотранспорт имеет следующие атрибуты: название транспорта (автобусы, такси, маршрутные такси, прочий легковой транспорт, грузовой транспорт и т.д.), кол-во наработки, пробег, кол-во ремонтов, характеристика.</p>

	<p>Таблица маршруты имеет следующие атрибуты: название маршрута, транспорт, водитель, график работы.</p> <p>Таблица водители имеет следующие атрибуты: фамилия, имя, отчество, год рождения, год поступления на работу, стаж, должность, пол, адрес, город, телефон.</p> <p>Таблица обслуживающий персонал имеет следующие атрибуты: должность (техники, сварщики, слесари, сборщики и др.), фамилия, имя, отчество, год рождения, год поступления на работу, стаж, пол, адрес, город, телефон.</p> <p>Таблица гаражное хозяйство имеет следующие атрибуты: название гаража, транспорт на ремонте, вид ремонта, дата поступления, дата выдачи после ремонта, результат ремонта, персонал, производящего ремонт.</p>
Вариант №7	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. БД – информационная система поликлиники. БД состоит из следующих таблиц: врачи, пациенты, история болезней, отделения, обслуживающий персонал. <p>Таблица отделения имеет следующие атрибуты: название отделения (хирургия, терапия, неврология и т.д.), этаж, номера комнат, ФИО заведующего.</p> <p>Таблица врачи имеет следующие атрибуты: фамилия, имя, отчество, должность, стаж работы, научное звание, адрес, номер отделения, в котором он работает.</p> <p>Таблица пациенты имеет следующие атрибуты: фамилия, имя, отчество, адрес, город, возраст, пол.</p> <p>Таблица диагнозы имеет следующие атрибуты: название диагноза, признаки болезни, период лечения, назначения.</p> <p>Таблица история болезни имеет следующие атрибуты: пациент, врач, диагноз, лечение, дата заболевания, дата вылечивания, вид лечения (амбулаторное, стационарное).</p>
Вариант №8	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. <p>БД – информационная система больницы. БД состоит из следующих таблиц: врачи, пациенты, история болезней, операции, лист лечения. Таблица врачи имеет следующие атрибуты: фамилия, имя, отчество, должность, стаж работы, научное звание, адрес.</p> <p>Таблица пациенты имеет следующие атрибуты: фамилия, имя, отчество, адрес, город, возраст, пол.</p> <p>Таблица история болезни имеет следующие атрибуты: пациента врач, диагноз, дата заболевания, дата вылечивания, вид лечения (амбулаторное, стационарное), код операции.</p> <p>Таблица лист лечения имеет следующие атрибуты: дата лечения, история бо-</p>

	<p>лезни, лекарства, температура, давление, состояние больного (тяжелое, среднее, и т.д.).</p> <p>Таблица операции имеет следующие атрибуты: описание операции (удаление аппендицита, пластическая операция и т.д.), врач, дата операции, пациент, результат операции.</p>
Вариант №9	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. БД – информационная система библиотек города. БД состоит из следующих таблиц: библиотеки, читальные залы, литература, читатели, выдача лит-ры. <p>Таблица библиотеки имеет следующие атрибуты: название, адрес, город.</p> <p>Таблица читальные залы имеет следующие атрибуты: название читального зала, библиотека, кол-во единиц лит-ры, кол-во посадочных мест, время работы, этаж, кол-во сотрудников.</p> <p>Таблица читатели имеет следующие атрибуты: фамилия, имя, отчество, категория читателя, место работы или обучения, возраст, дата регистрации в библиотеке.</p> <p>Таблица литература имеет следующие атрибуты: название, категория литературы, авторы, издательство, год издательства, кол-во страниц, читальный зал.</p> <p>Таблица выдача литературы имеет следующие атрибуты: читатель, литература, дата выдачи, срок выдачи, вид выдачи, наличие залога.</p>
Вариант №10	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. БД – информационная система автосалона. БД состоит из следующих таблиц: автомобили, марка автомобиля, сотрудники, продажа автомобилей, покупатели. <p>Таблица марка автомобиля имеет следующие атрибуты: название марки, страна производитель, завод производитель, адрес.</p> <p>Таблица автомобиля имеет следующие атрибуты: название автомобиля, марка, год производства, цвет, категория, цена.</p> <p>Таблица покупатели имеет следующие атрибуты: фамилия, имя, отчество, паспортные данные, адрес, город, возраст, пол.</p> <p>Таблица сотрудника имеет следующие атрибуты: фамилия, имя, отчество, стаж, зарплата.</p> <p>Таблица продажа автомобилей имеет следующие атрибуты: дата, сотрудник, автомобиль, покупатель.</p>

Вариант №11	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. БД – успеваемость студентов кафедры. БД состоит из следующих таблиц: кафедры, дисциплины, преподаватели, студенты, ведомости успеваемости. <p>Таблица кафедры имеет следующие атрибуты: название кафедры, факультет, ФИО заведующего, номер комнаты, номер корпуса, телефон, кол-во преподавателей.</p> <p>Таблица преподаватели имеет следующие атрибуты: фамилия, имя, отчество, кафедра, год рождения, год поступления на работу, стаж, должность, пол, адрес, город, телефон.</p> <p>Таблица студенты имеет следующие атрибуты: фамилия, имя, отчество, кафедра, год рождения, пол, адрес, город, телефон.</p> <p>Таблица дисциплины имеет следующие атрибуты: название дисциплины, кафедра, читаемой эту дисциплину, кол-во часов, вид итогового контроля.</p> <p>Таблица ведомости успеваемости имеет следующие атрибуты: преподаватель, дисциплина, студент, оценка</p>
Вариант №12	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. <p>БД – торговая организация. БД состоит из следующих таблиц: торговая организация, торговая точка, продавцы, поставщики, заказы поставщикам.</p> <p>Таблица торговая организация имеет следующие атрибуты: название торговой организации, адрес, ФИО директора, налоговый номер.</p> <p>Таблица торговая точка имеет следующие атрибуты: название торговой точки, тип (универмаги, магазины, киоски, лотки и т.д.), торговая организация, адрес, ФИО заведующего.</p> <p>Таблица продавцы имеет следующие атрибуты: фамилия, имя, отчество, торговая точка, должность, год рождения, пол, адрес проживания, город.</p> <p>Таблица поставщики имеет следующие атрибуты: название поставщика, тип деятельности, страна, город, адрес.</p> <p>Таблица заказы поставщикам имеет следующие атрибуты: дата заказа, торговая точка, поставщик, название товара, кол-во, цена.</p>

Вариант №13	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. БД – проектная организация. БД состоит из следующих таблиц: отделы, сотрудники, организации, договора, проектные работы. Таблица отделы имеет следующие атрибуты: название отдела, этаж, телефон, начальник отдела. Таблица сотрудники имеет следующие атрибуты: ФИО, должность (конструкторы, инженеры, техники, лаборанты, прочий обслуживающий персонал), номер отдела, в котором работает, пол, адрес, дата рождения. Таблица организации имеет следующие атрибуты: название организации, тип деятельности, страна, город, адрес, ФИО директора. Таблица договора имеет следующие атрибуты: номер договора, дата заключения договора, организация, стоимость договора. Таблица проектные работы имеет следующие атрибуты: дата начала проектной работы, дата завершения проектной работы, номер договора, отдел, осуществляющий разработку.
№14	<ol style="list-style-type: none"> 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. <p>БД – информационная система военно-морского флота. БД состоит из следующих таблиц: базы, части, личный состав, корабли, учения. Базы военно-морского флота имеет следующие атрибуты: название базы, географическое расположение, кол-во частей.</p> <p>Таблица части имеет следующие атрибуты: номер части, база флота, место базирования, вид войск (морская авиация, морская пехота и т.д.). Таблица личный состав имеет следующие атрибуты: фамилия, часть, должность, год рождения, год поступления на службу, выслуга лет, награды,</p> <p>Таблица корабли имеет следующие атрибуты: идентификационный номер корабля, название корабля, тип корабля, дата создания, наработка, кол-во посадочных мест, устройство двигателя (парусное, гребное, пароход, теплоход, турбоход, и т.д.), тип привода (самоходное, несамоходное), размещение корпуса (подводная лодка, ныряющее, полупогружное, и т.д.) Таблица учения: часть, корабль, дата учения, место проведения, оценка.</p>

Вариант №15	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. <p>БД – туристическая фирма. БД состоит из следующих таблиц: туристы, туристическая группа, состав групп, гостиницы, ведомости продаж. Таблица туристы имеет следующие атрибуты: ФИО, паспортные данные, пол, возраст, дети.</p> <p>Таблица туры имеет следующие атрибуты: название, страна, города, тип передвижения, тип питания, цена тура, тип проживания.</p> <p>Таблица туристическая группа имеет следующие атрибуты: название, дата отправления, дата прибытия, тур, кол-во туристов.</p> <p>Таблица состав групп имеет следующие атрибуты: дата продажи, турист, группа, цена билета.</p> <p>Таблица гостиницы имеет следующие атрибуты: название гостиницы, страна, город, адрес, кол-во мест, тип гостиницы.</p> <p>Таблица ведомость продаж имеет следующие атрибуты: дата, туристическая группа, гостиница, общая стоимость.</p>
Вариант №16	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. <p>БД – цирк. БД состоит из следующих таблиц: работники цирка, представления, расписание гастролей, труппа цирка, программа цирка. Таблица работники цирка имеет следующие атрибуты: фамилия, имя, отчество, год рождения, год поступления на работу, стаж, должность (акробат, клоун, гимнаст, музыкант, постановщик, служащий и т.д.), пол, адрес, город, телефон.</p> <p>Таблица представления имеет следующие атрибуты: название, режиссер-постановщик, художник-постановщик, дирижер-постановщик, автор, жанр, тип.</p> <p>Таблица расписание гастролей имеет следующие атрибуты: представление, дата начала, дата окончания, места проведения гастролей.</p> <p>Таблица труппа представления цирка имеет следующие атрибуты: представление, актер цирка, название роли.</p> <p>Таблица программа цирка имеет следующие атрибуты: представление, дата премьеры, период проведения, дни и время, цена билета.</p>

Вариант №17	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. <p>БД – аптека. БД состоит из следующих таблиц: лекарства, покупатель, продавец, рецепты, продажа лекарств.</p> <p>Таблица лекарства имеет следующие атрибуты: название, тип (готовое, изготавливаемое), вид (таблетки, мази, настойки), цена.</p> <p>Таблица покупатель имеет следующие атрибуты: фамилия, имя, отчество, адрес, город, телефон.</p> <p>Таблица продавец имеет следующие атрибуты: фамилия, имя, отчество, дата поступления, дата рождения, образование.</p> <p>Таблица рецепты имеет следующие атрибуты: номер рецепта, дата выдачи, ФИО больного (покупатель), ФИО врача, диагноз пациента. Таблица продажа лекарств имеет следующие атрибуты: дата, лекарство, кол-во, рецепт, продавец.</p>
Вариант №18	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; <p>Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями;</p> <ol style="list-style-type: none"> 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. <p>БД – городская телефонная сеть. БД состоит из следующих таблиц: АТС, абонент, ведомость звонков, прайс АТС, ведомость абонентской платы.</p> <p>Таблица АТС имеет следующие атрибуты: название АТС, вид (городские, ведомственные и учрежденческие), адрес, город, кол-во абонентов.</p> <p>Таблица абоненты имеет следующие атрибуты: фамилия, имя, отчество, вид телефона (основной, параллельный или спаренный), номер телефона, межгород (открыт/закрыт), льгота (да/нет), адрес: индекс, район, улица, дом, квартира.</p> <p>Таблица ведомость звонков имеет следующие атрибуты: абонент, дата звонка, время начала, время окончания, межгород (да/нет). Таблица прайс АТС имеет следующие атрибуты: АТС, цена на городские, цена на межгород.</p> <p>Таблица ведомость абонентской платы имеет следующие атрибуты: абонент, месяц, год, кол-во минут на городские, кол-во минут на межгород, стоимость, сумма льготы, общая стоимость.</p>
Вариант №19	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. <p>БД – аэропорт. БД состоит из следующих таблиц: работники аэропорта, расписание вылетов, самолеты, бригады самолетов, ведомость продаж билетов.</p> <p>Таблица работники аэропорта имеет следующие атрибуты: фамилия, имя, от-</p>

	<p>чество, год рождения, год поступления на работу, стаж, должность (пилотов, диспетчеров, техников, кассиров, работников службы безопасности, справочной службы и других.), пол, адрес, город, телефон. Таблица расписание вылетов имеет следующие атрибуты: самолет, дата вылета, время вылета, место выбытия, место прибытия, маршрут (начальный и конечный пункты назначения, пункт пересадки), стоимость билета.</p> <p>Таблица самолеты имеет следующие атрибуты: номер, год выпуска, кол-во посадочных место, грузоподъемность.</p> <p>Таблица бригады самолетов имеет следующие атрибуты: номер бригады, самолет, работник аэропорта (пилоты, техники и обслуживающий персонал)ю</p> <p>Таблица ведомость продажи билетов имеет следующие атрибуты: дата и время продажи, ФИО пассажира, паспортные данные, номер рейса, колво билетов, наличие льгот (пенсионеры, дети-сироты и т.д.), багаж (да/нет), стоимость.</p>
Вариант №20	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. БД – театр. БД состоит из следующих таблиц: работники театра, спектакли, расписание гастролей, труппа спектакля, репертуар театра. Таблица работники театра имеет следующие атрибуты: фамилия, имя, отчество, год рождения, год поступления на работу, стаж, должность (актеров, музыкантов, постановщиков и служащих), пол, адрес, город, телефон. <p>Таблица спектакли имеет следующие атрибуты: название, режиссерпостановщик, художник-постановщик, дирижер-постановщик, автор, жанр (музыкальная комедия, трагедия, оперетта и пр.), тип (детские, молодежные и пр.).</p> <p>Таблица расписание гастролей имеет следующие атрибуты: название, дата начала, дата окончания, места проведения гастроль, спектакль.</p> <p>Таблица труппа спектакля имеет следующие атрибуты: спектакль, актер, название роли.</p> <p>Таблица репертуар театра имеет следующие атрибуты: спектакль, дата премьеры, период проведения, дни и время, цена билета.</p>
Вариант №21	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. БД – железнодорожный вокзал. БД состоит из следующих таблиц: работники ж.д.вокзала, расписание движения поездов, поезда, бригады поездов, ведомость продаж билетов. <p>Таблица работники ж.д.вокзала имеет следующие атрибуты: фамилия, имя, отчество, год рождения, год поступления на работу, стаж, должность (машинист, диспетчеров, проводник, ремонтников подвижного состава, путей, кассиров, работников службы подготовки составов, справочной службы и других.), пол, адрес, город, телефон.</p> <p>Таблица расписание движения поездов имеет следующие атрибуты: поезд,</p>

	<p>дата отправления, время отправления, место отправления, дата прибытия, время прибытия, место прибытия, маршрут ((начальный и конечный пункты назначения, основные узловые станции), стоимость билета.</p> <p>Таблица поезда имеет следующие атрибуты: номер, год выпуска, кол-во вагонов, тип поезда (общий, скоростной, высокоскоростной).</p> <p>Таблица бригады поездов имеет следующие атрибуты: номер бригады, поезд, работник ж.д.вокзала (машинисты, техники, проводники и обслуживающий персонал).</p> <p>Таблица ведомость продажи билетов имеет следующие атрибуты: дата и время продажи, ФИО пассажира, паспортные данные, номер рейса, колво билетов, наличие льгот (пенсионеры, дети-сироты и т.д.), стоимость.</p>
Вариант №22	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. БД – информационная система ВУЗА. БД состоит из следующих таблиц: факультеты, кафедры, преподаватели, дисциплины, учебная нагрузка. <p>Таблица факультеты имеет следующие атрибуты: название факультета, ФИО декана, номер комнаты, номер корпуса, телефон.</p> <p>Таблица кафедры имеет следующие атрибуты: название кафедры, ФИО заведующего, номер комнаты, номер корпуса, телефон, кол-во преподавателей.</p> <p>Таблица дисциплины имеет следующие атрибуты: название дисциплины, кол-во часов, цикл дисциплин.</p> <p>Таблица преподаватели имеет следующие атрибуты: фамилия, имя, отчество, кафедра, год рождения, год поступления на работу, стаж, должность, пол, город.</p> <p>Таблица учебная нагрузка имеет следующие атрибуты: преподаватель, дисциплина, учебный год, семестр, группы, кол-во студентов, вид итогового контроля.</p>
Вариант №23	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. БД – информационная система военного округа. БД состоит из следующих таблиц: места дислокации, вид войск, части, роты, личный состав. <p>Таблица вид войск имеет следующие атрибуты: название вида войск. Таблица места дислокации имеет следующие атрибуты: страна, город, адрес, занимаемая площадь, кол-во сооружений.</p> <p>Таблица части имеет следующие атрибуты: номер части, место дислокации, вид войск, кол-во рот, кол-во техники, кол-во вооружений. Таблица техника имеет следующие атрибуты: название техники, часть, характеристики.</p> <p>Таблица вооружения имеет следующие атрибуты: название вооружения, часть, характеристики.</p>

Вариант №24	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. БД – информационная система супермаркета. БД состоит из следующих таблиц: отделы, клиенты, товары, продажа товаров, поставщики. <p>Таблица отделы имеет следующие атрибуты: название отдела, кол-во прилавков, кол-во продавцов, номер зала.</p> <p>Таблица клиенты имеет следующие атрибуты: название клиента, адрес, вид оплаты.</p> <p>Таблица поставщики имеет следующие атрибуты: название поставщика, адрес, страна, вид транспорта, вид оплаты.</p> <p>Таблица товары имеет следующие атрибуты: название товара, отдел, поставщик, условия хранения, сроки хранения .</p> <p>Таблица продажа товаров имеет следующие атрибуты: клиент, товар, дата, время, кол-во, цена, сумма.</p>
Вариант №25	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. <p>БД – информационная система больницы. БД состоит из следующих таблиц: врачи, пациенты, история болезней, отделения, лист лечения. Таблица отделения имеет следующие атрибуты: название отделения (хирургия, терапия, неврология и т.д.), этаж, номера комнат, ФИО заведующего.</p> <p>Таблица врачи имеет следующие атрибуты: фамилия, имя, отчество, должность, стаж работы, научное звание, адрес.</p> <p>Таблица пациенты имеет следующие атрибуты: фамилия, имя, отчество, адрес, город, возраст, пол.</p> <p>Таблица история болезни имеет следующие атрибуты: пациент, врач, диагноз, дата заболевания, дата вылечивания, вид лечения (амбулаторное, стационарное).</p> <p>Таблица лист лечения имеет следующие атрибуты: дата лечения, история болезни, лекарства, температура, давление, состояние больного (тяжелое, среднее, и т.д.).</p>

Лабораторная работа. «Создание базы данных в СУБД»

Цель: научиться устанавливать «Microsoft SQL Server 2008»

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

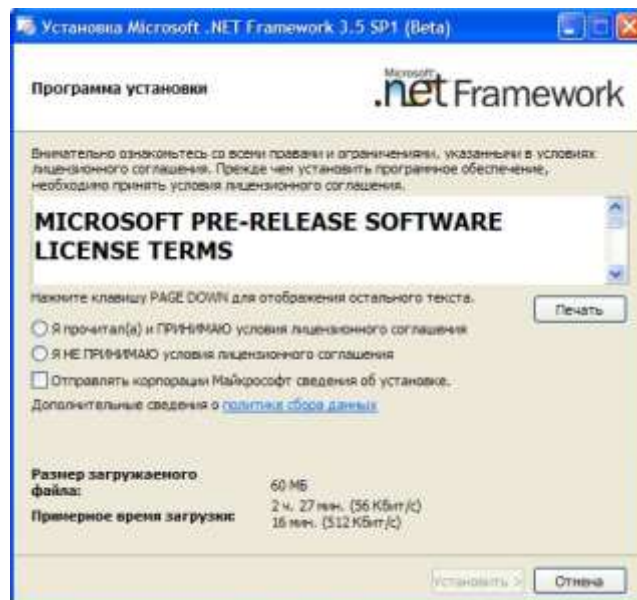


Рис.1.1

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

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

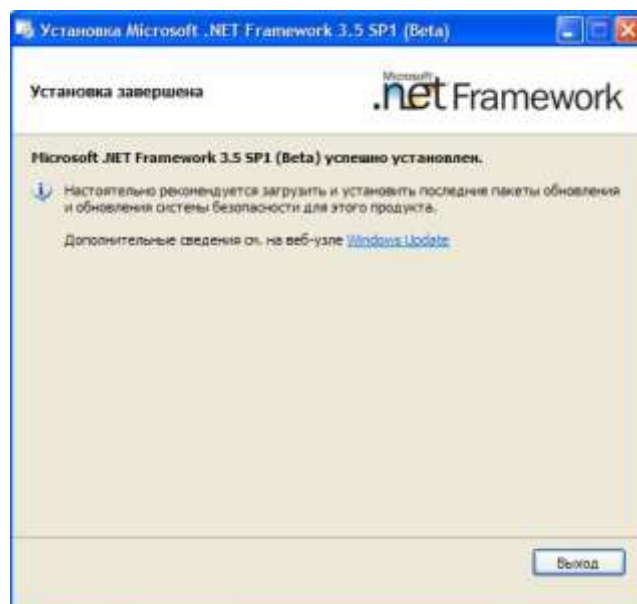


Рис.1.2

В данном окне нажмите кнопку «Выход».

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

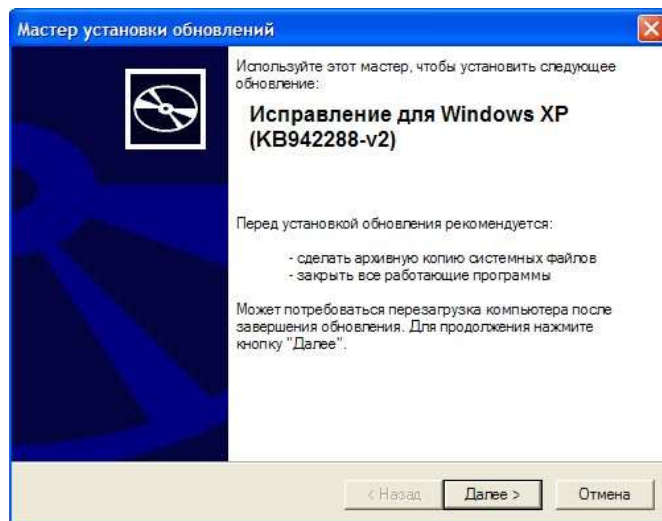


Рис.1.3

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



Рис.1.4

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

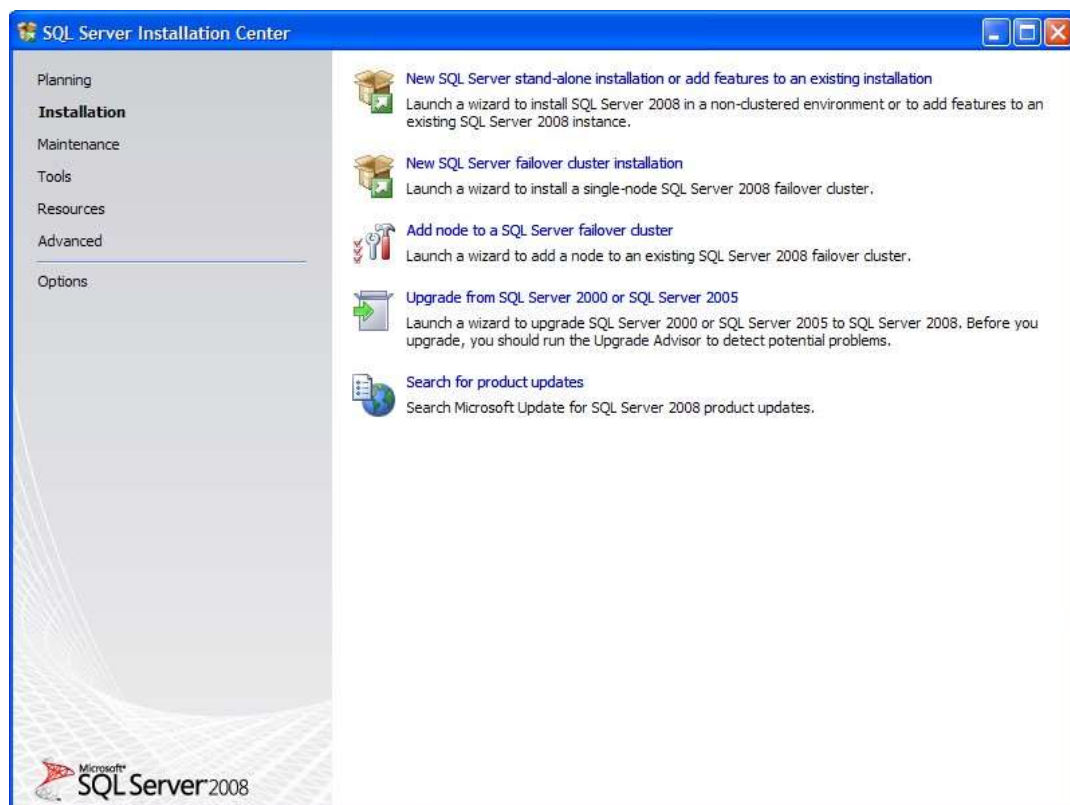


Рис.1.5

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

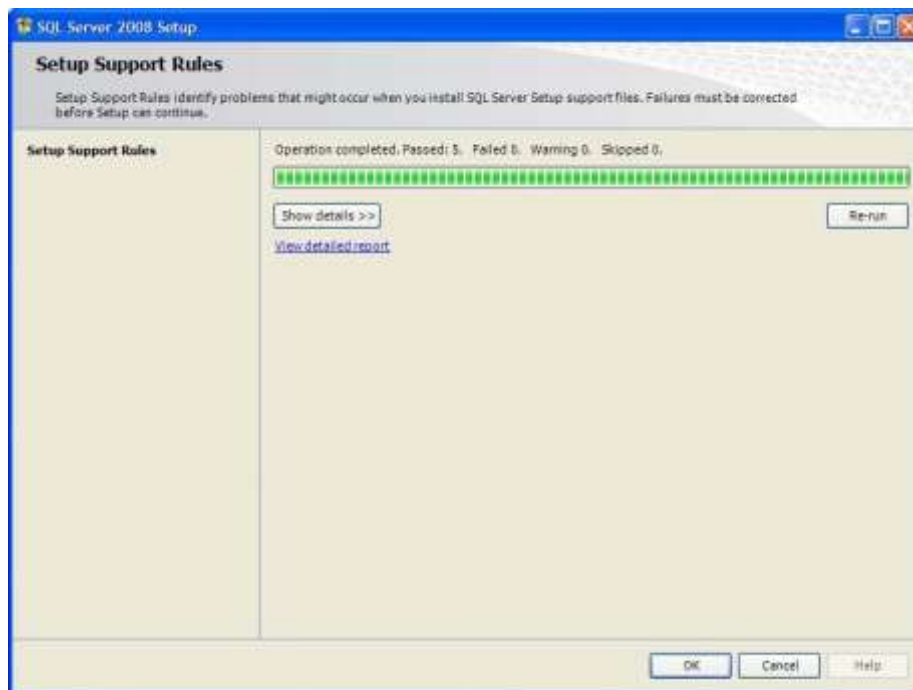


Рис.1.6

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

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

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



Рис.1.7

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



Рис.1.8

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

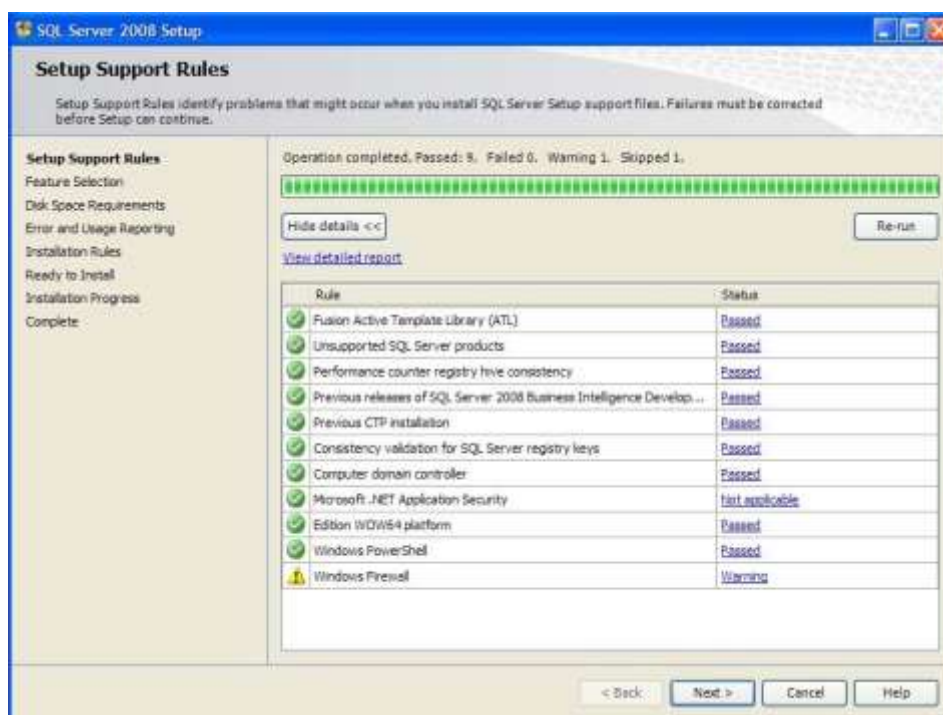


Рис.1.9

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

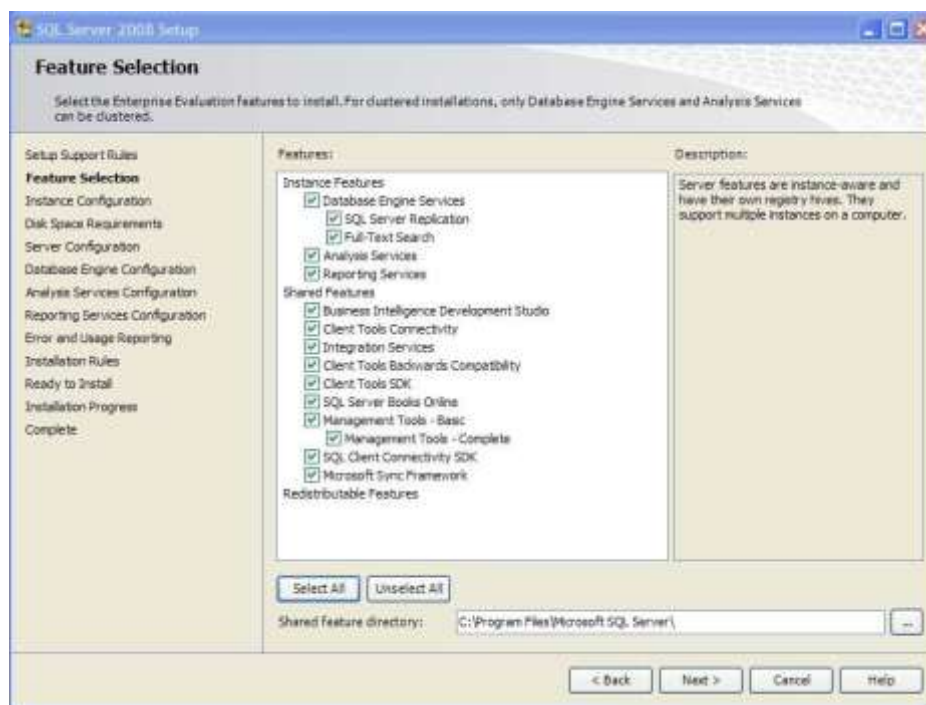


Рис.1.10

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

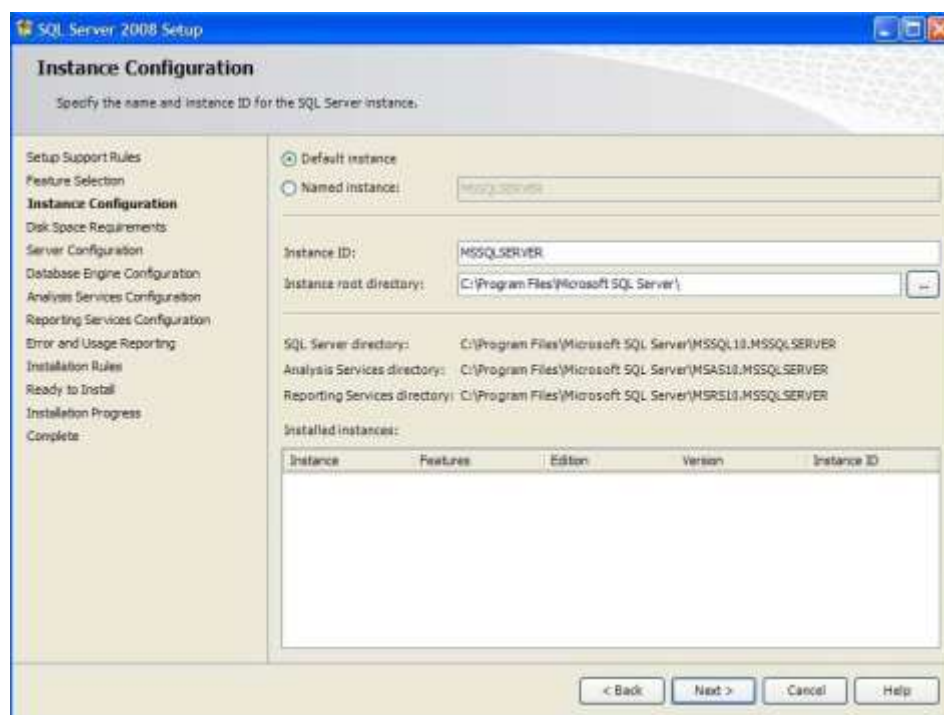


Рис.1.11

В данном окне определите следующие настройки:

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

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

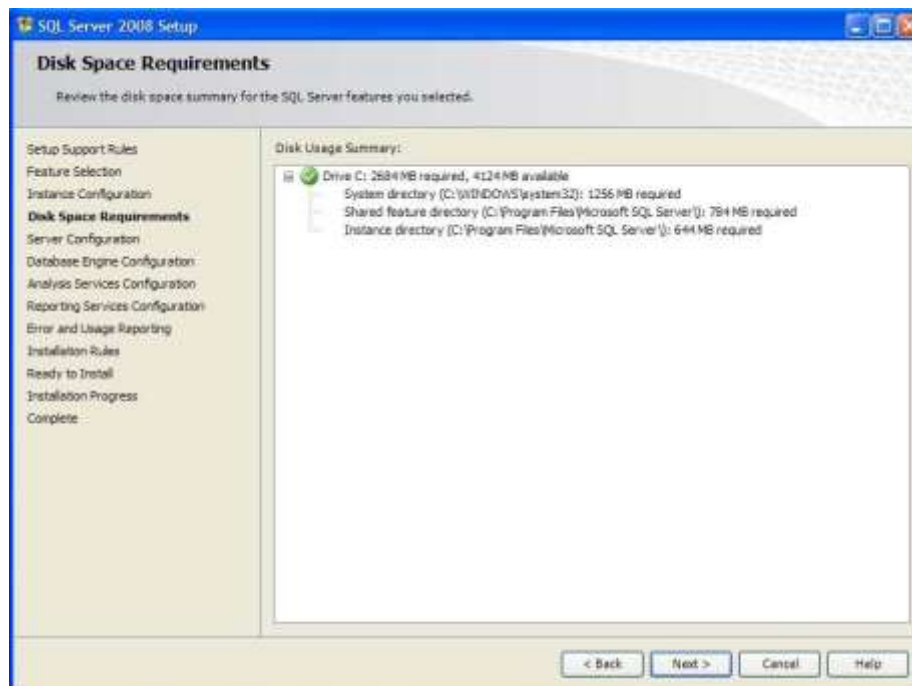


Рис.1.12

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

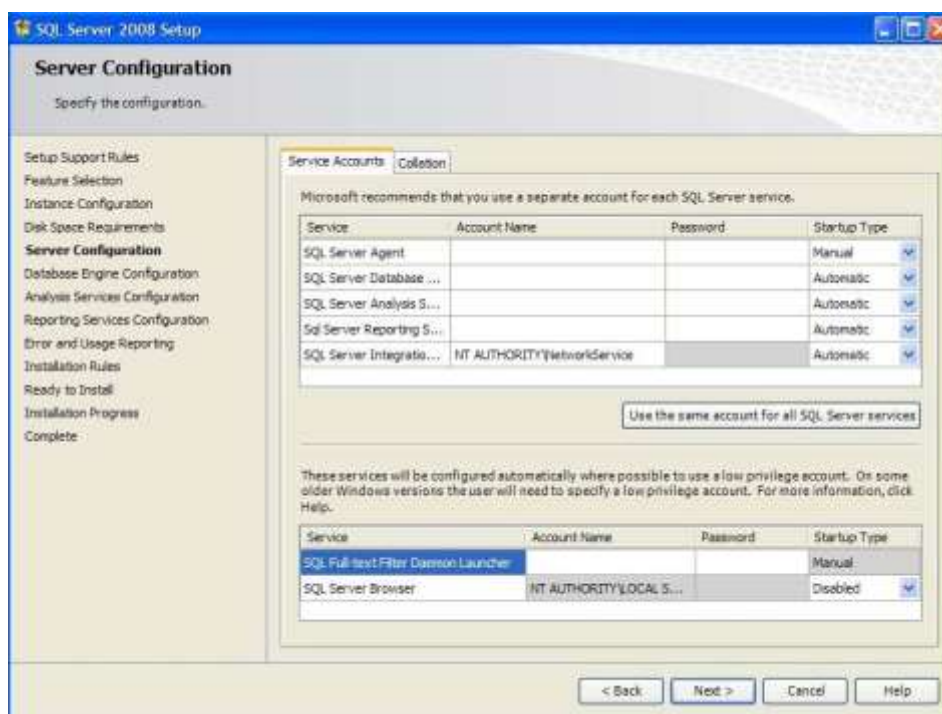


Рис.1.13

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



Рис.1.14

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

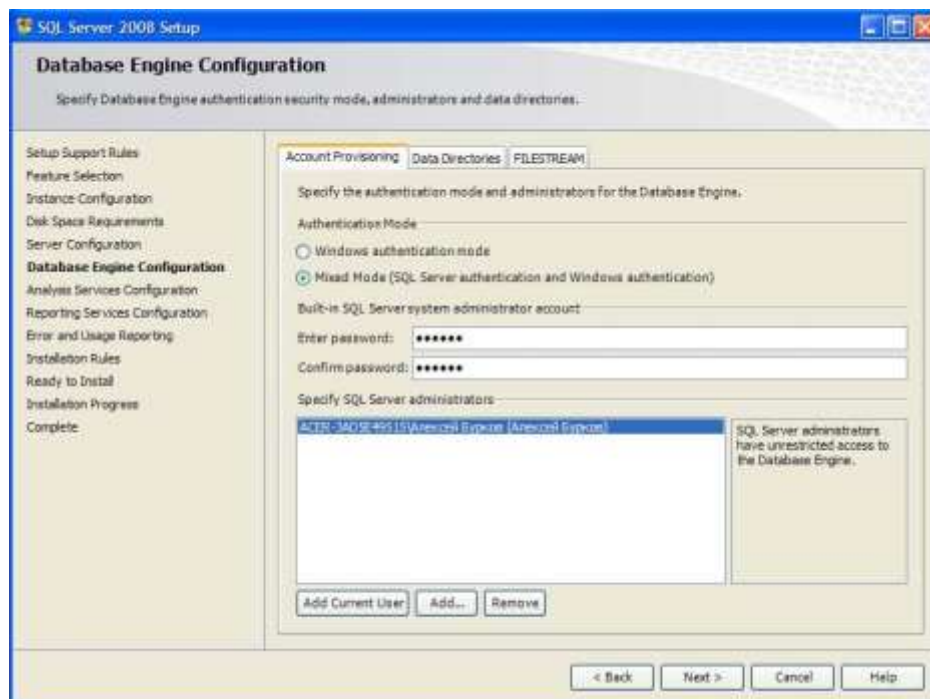


Рис.1.15

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

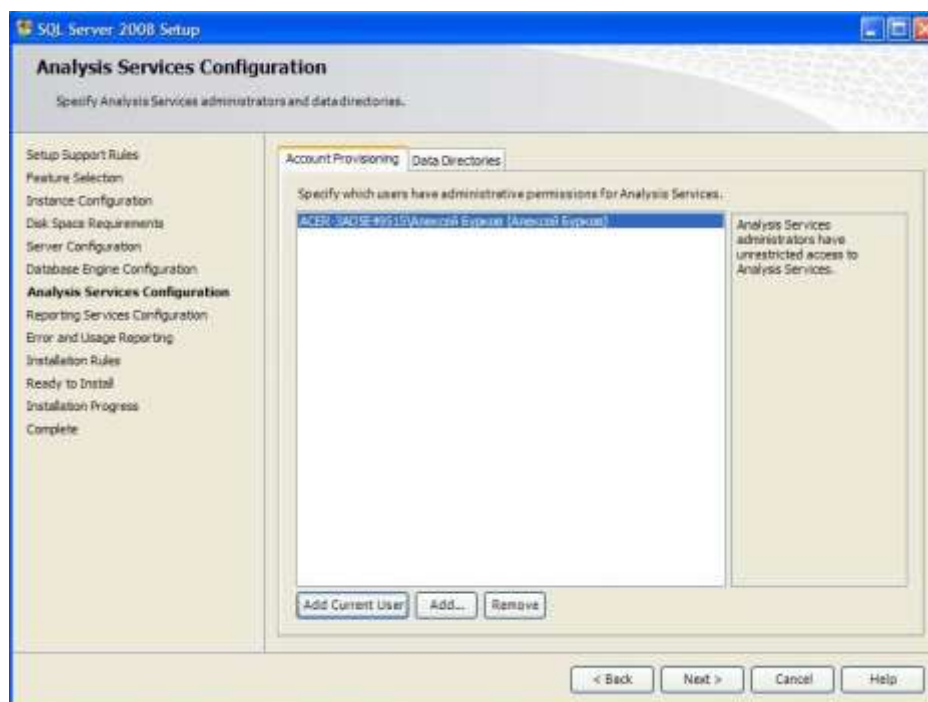


Рис.1.16

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

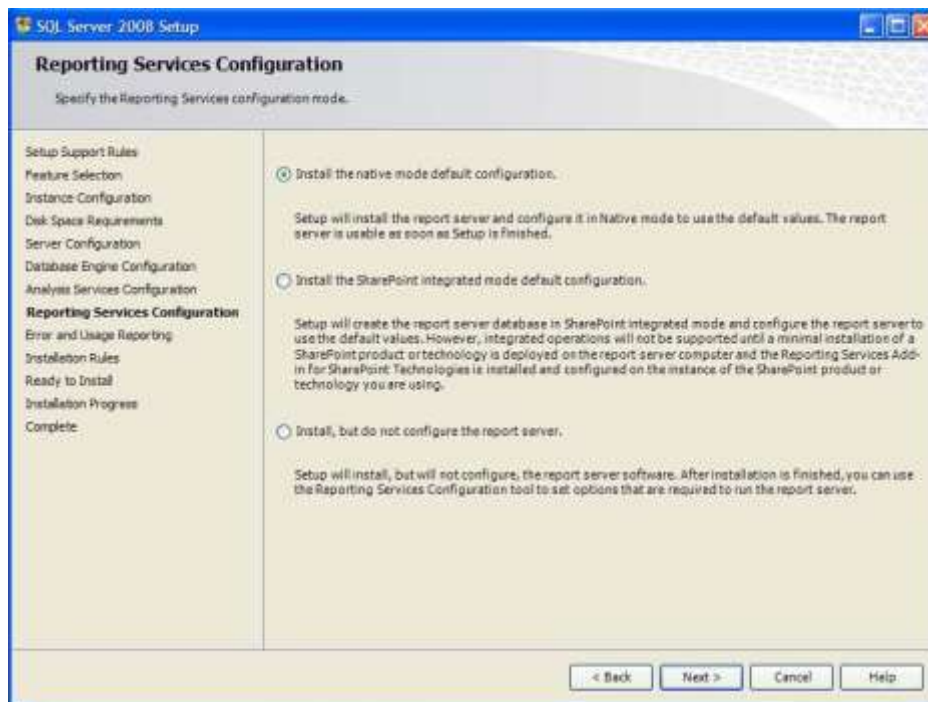


Рис.1.17

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

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

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

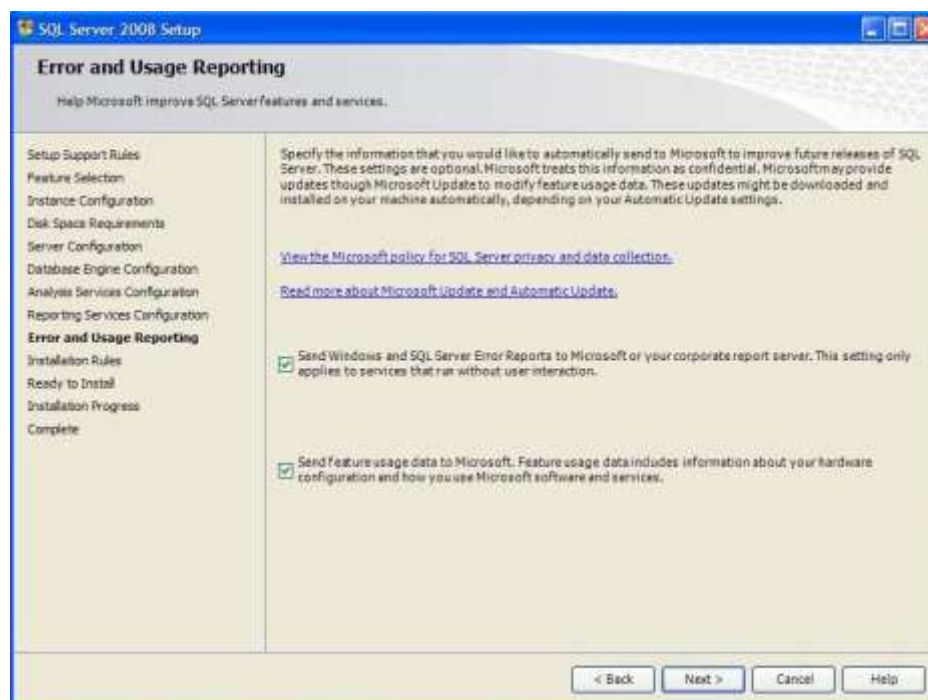


Рис.1.18

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

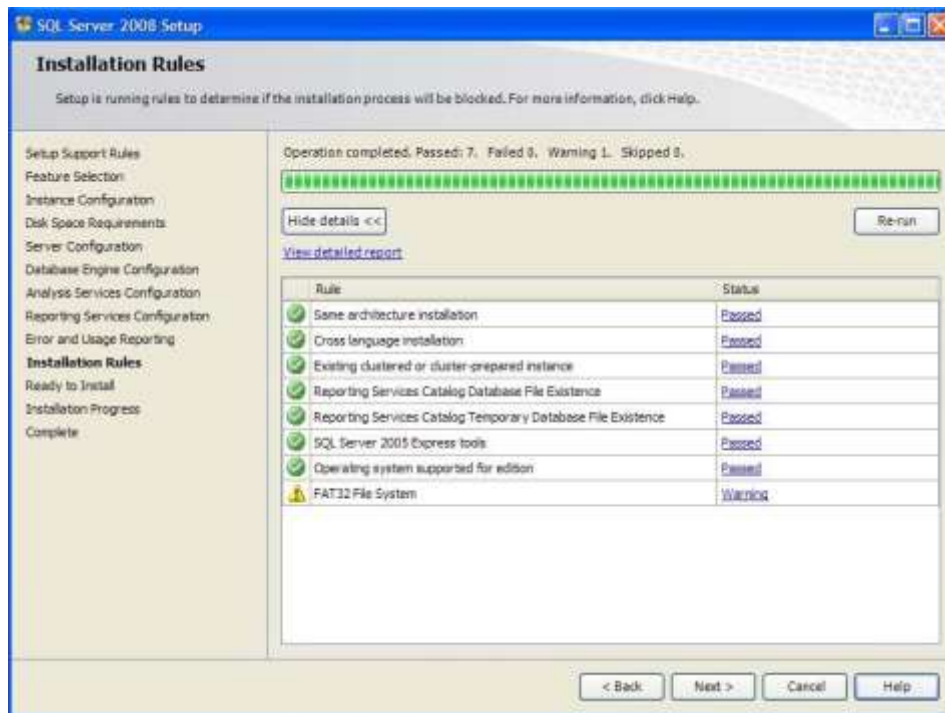


Рис.1.19

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

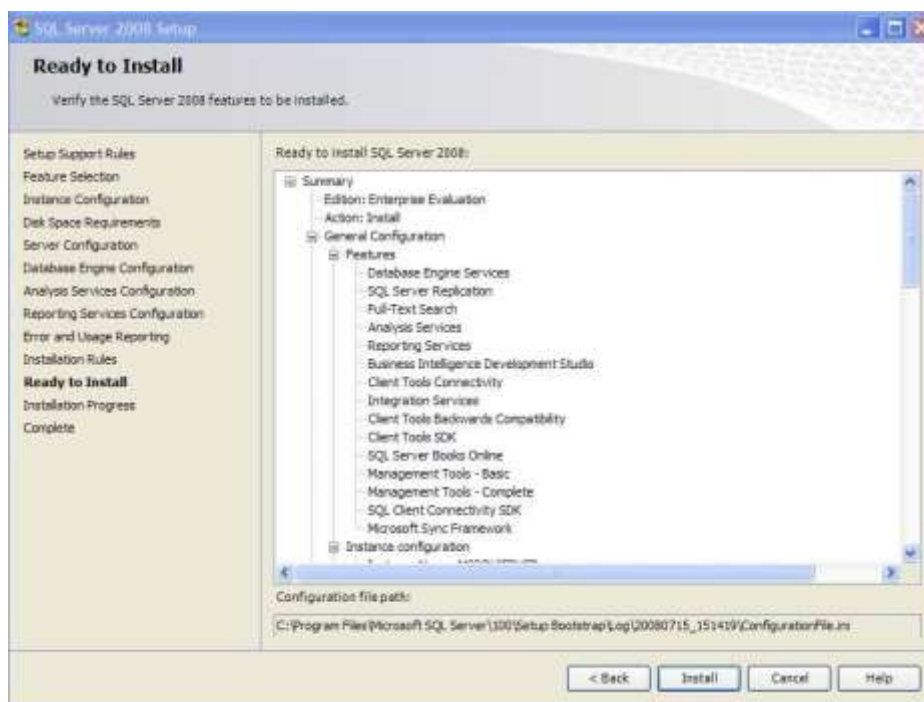


Рис.1.20

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

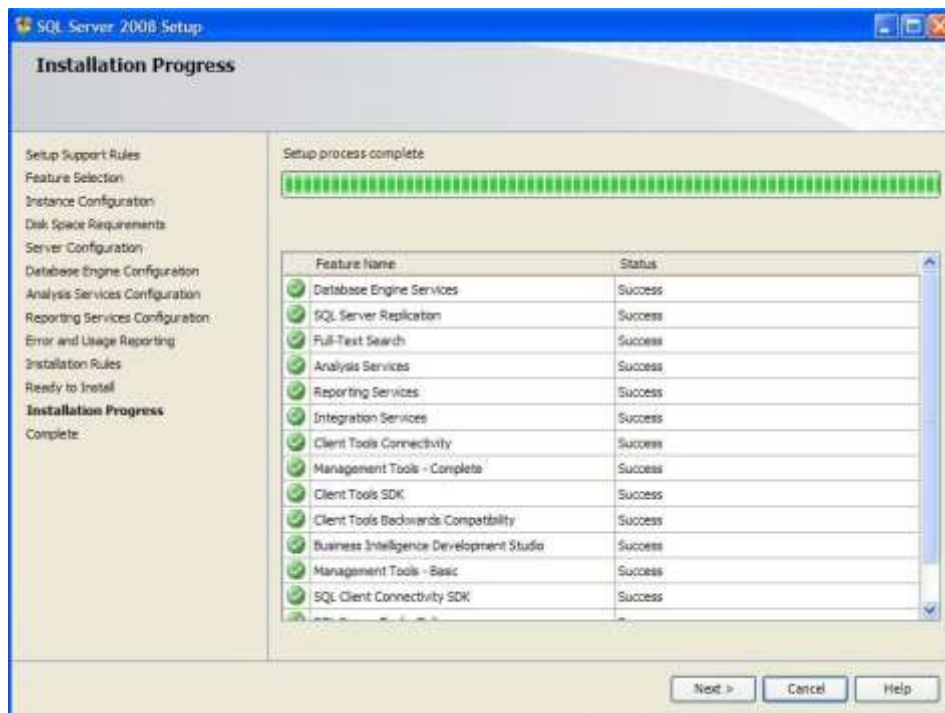


Рис.1.21

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

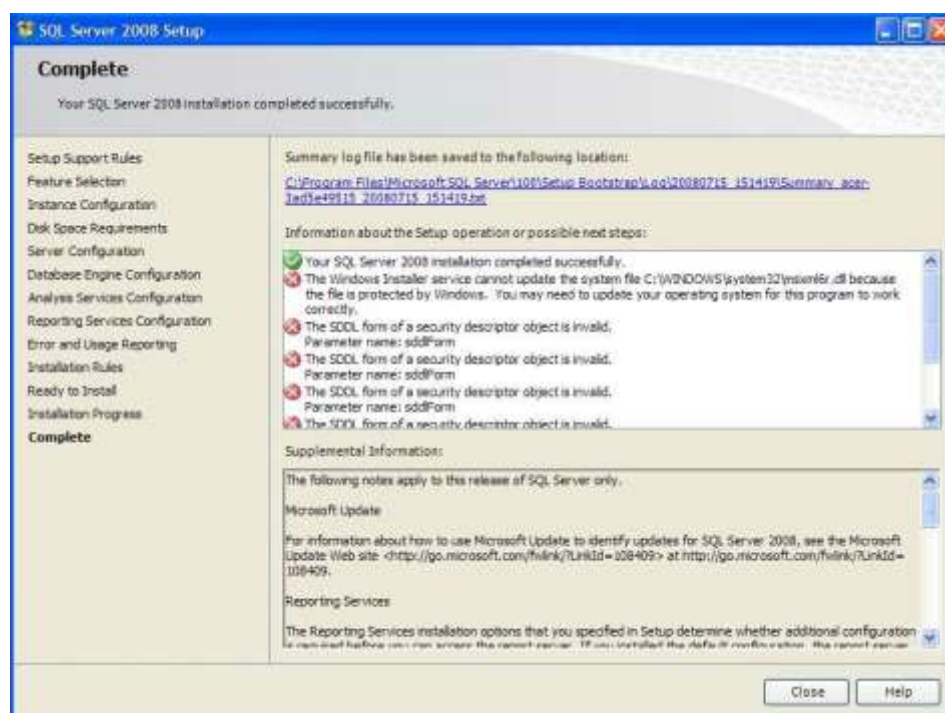


Рис.1.22

Для завершения установки сервера в данном окне нажмите кнопку «Close» (Закреть). На этом мы заканчиваем описание установки «Microsoft SQL Server 2008» и переходим к созданию баз данных.

Лабораторная работа. «Создание файла данных и журнала транзакций»

Цель: научиться создавать файла данных и журнал транзакций

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

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

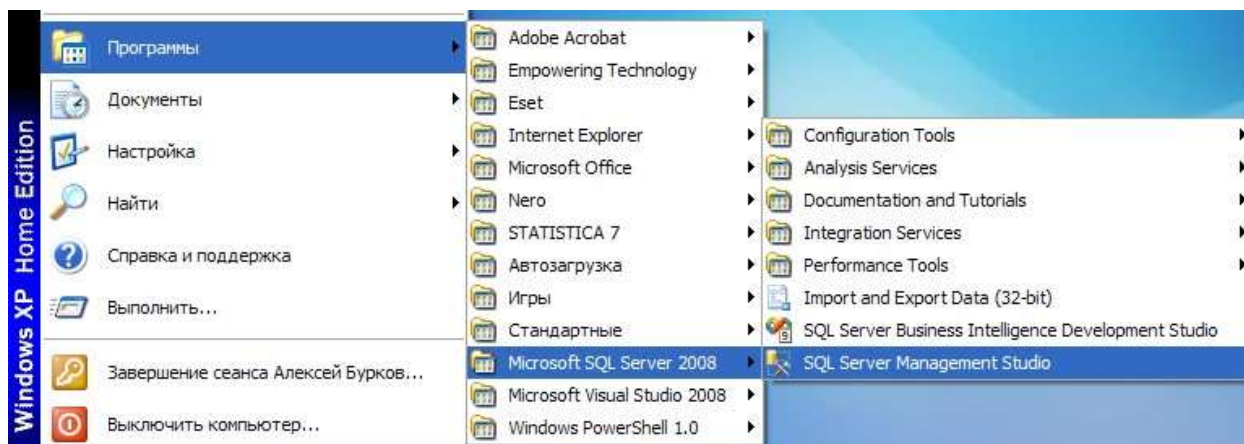


Рис.2.1

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



Рис.2.2

В этом окне необходимо нажать кнопку «Connect»

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

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

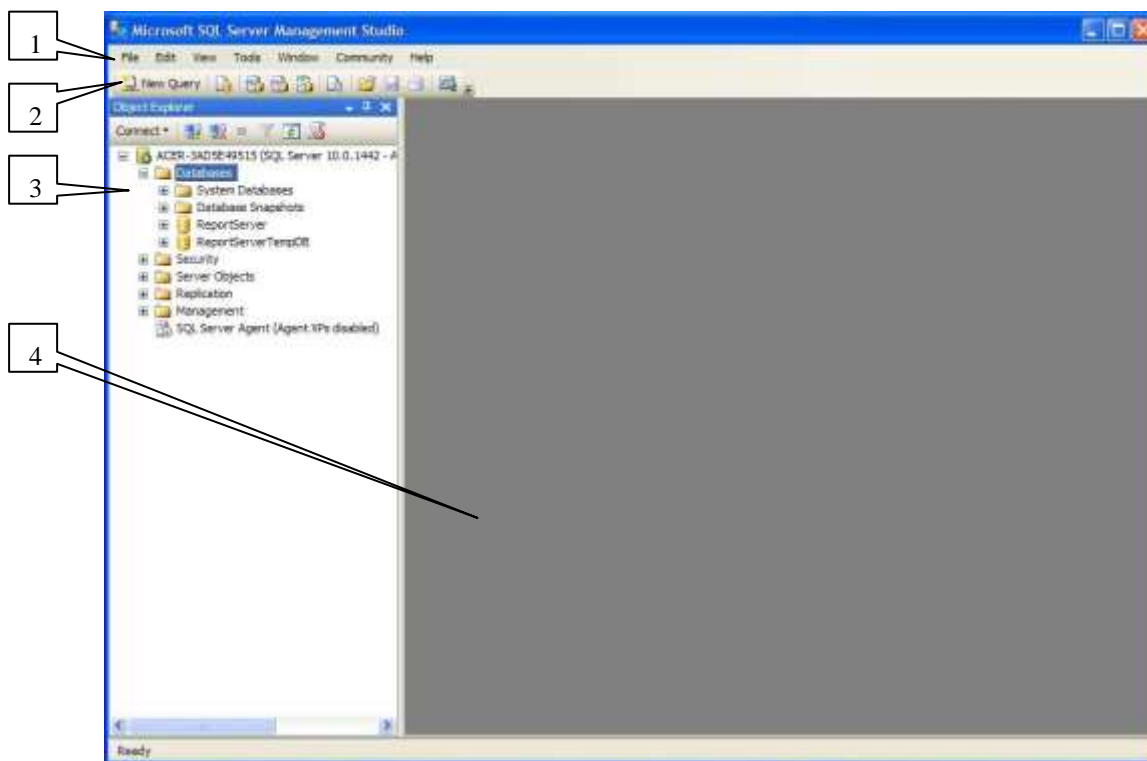


Рис.2.3

Данное окно имеет следующую структуру (Рис.2.3):

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

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

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

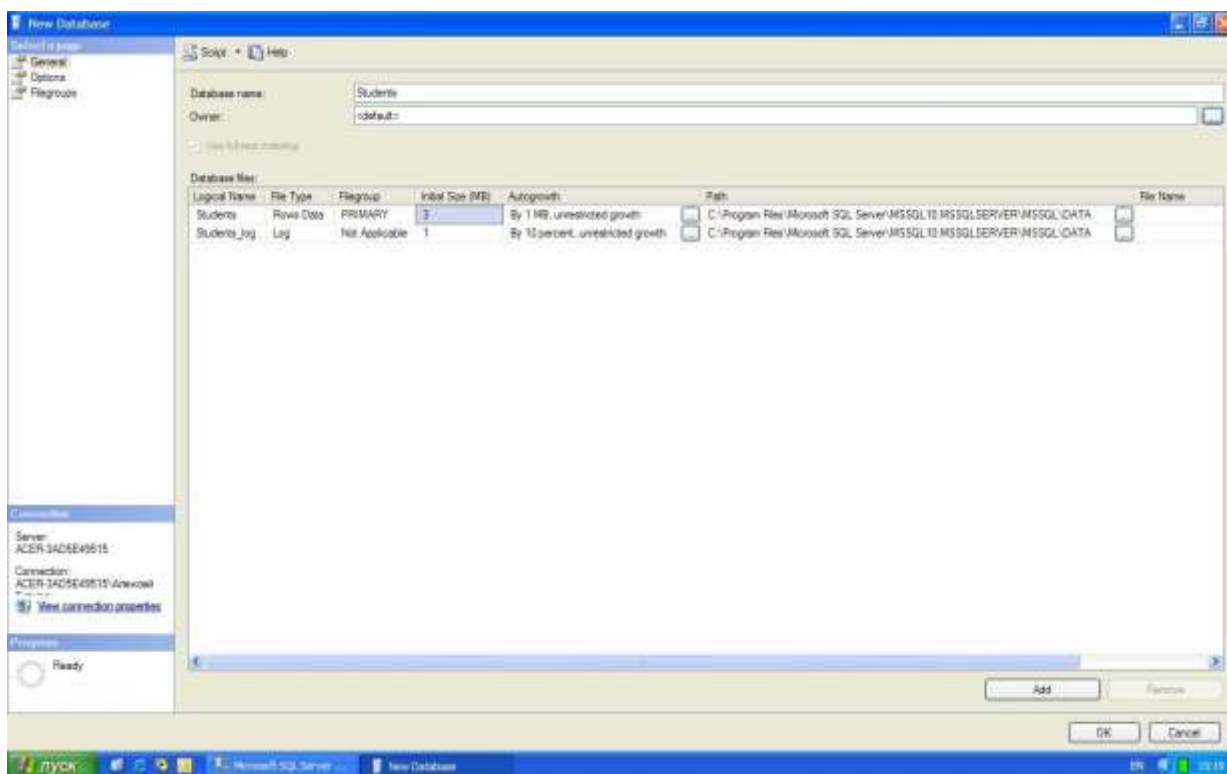


Рис.2.4

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

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

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

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

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

В нашем случае мы оставим все основные настройки без изменений.

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

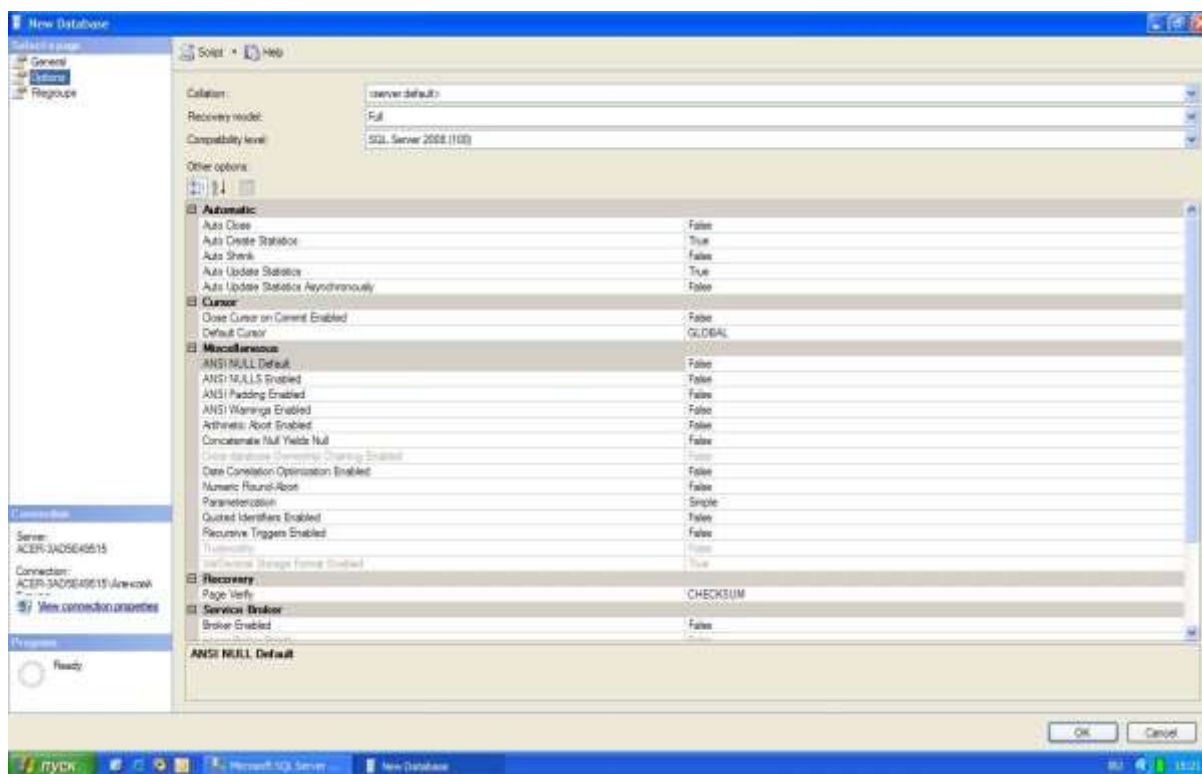


Рис.2.5

В правой части окна мы видим следующие настройки:

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

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

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

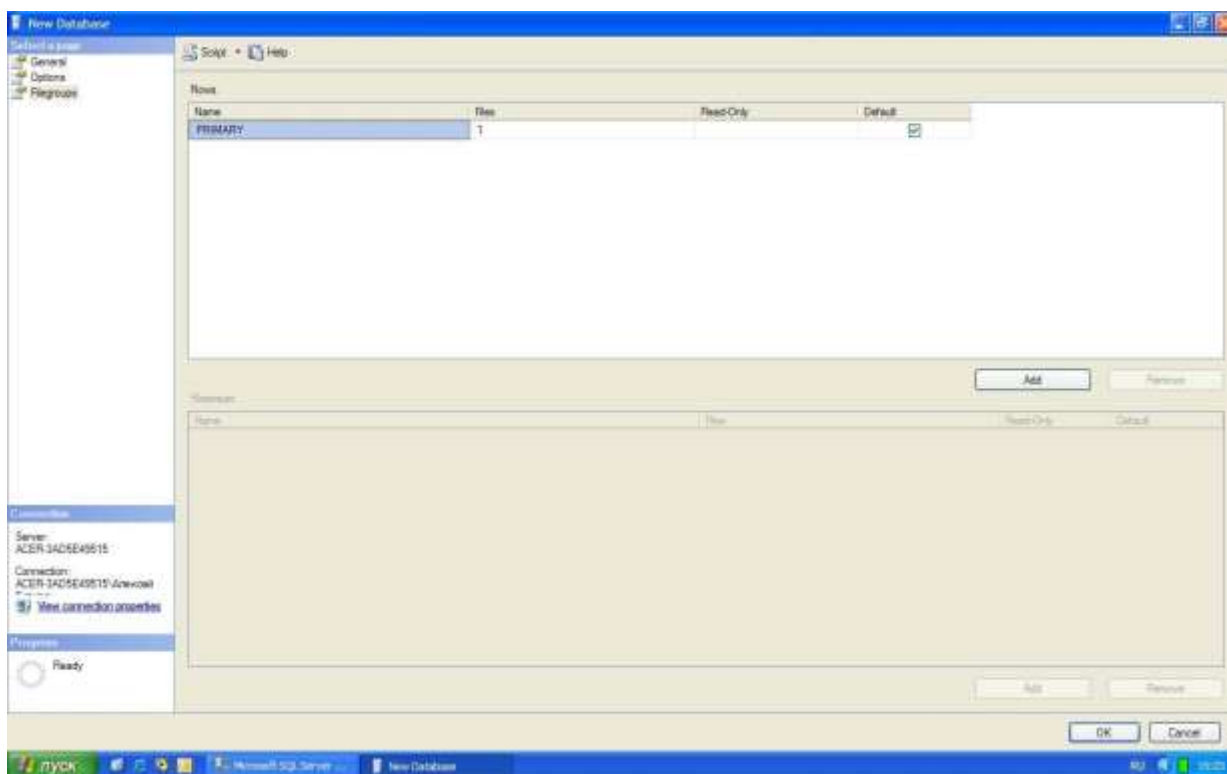


Рис.2.6

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

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

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

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

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

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

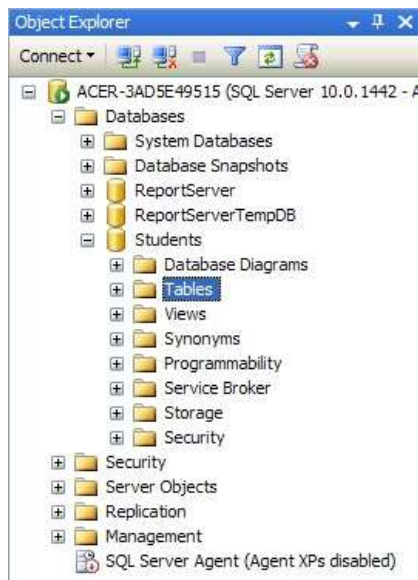


Рис.2.7

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

Контрольная работа на тему «Язык структурированных запросов SQL».

Вариант 1

1. Типы данных
2. Язык DDL
3. Возможности команды SELECT
4. Создание хранимых процедур
5. Создание ограничений

Вариант 2

1. Ограничения
2. Язык DML
3. Возможности команды CREATE TABLE
4. Создание триггеров
5. Снятие ограничений

Лабораторная работа. MS SQL Server. Описание данных. Команды изменения содержания таблиц.

Цель: научиться создавать и заполнять таблицы

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

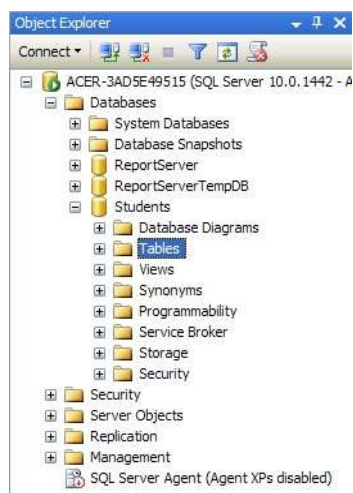


Рис.3.1

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

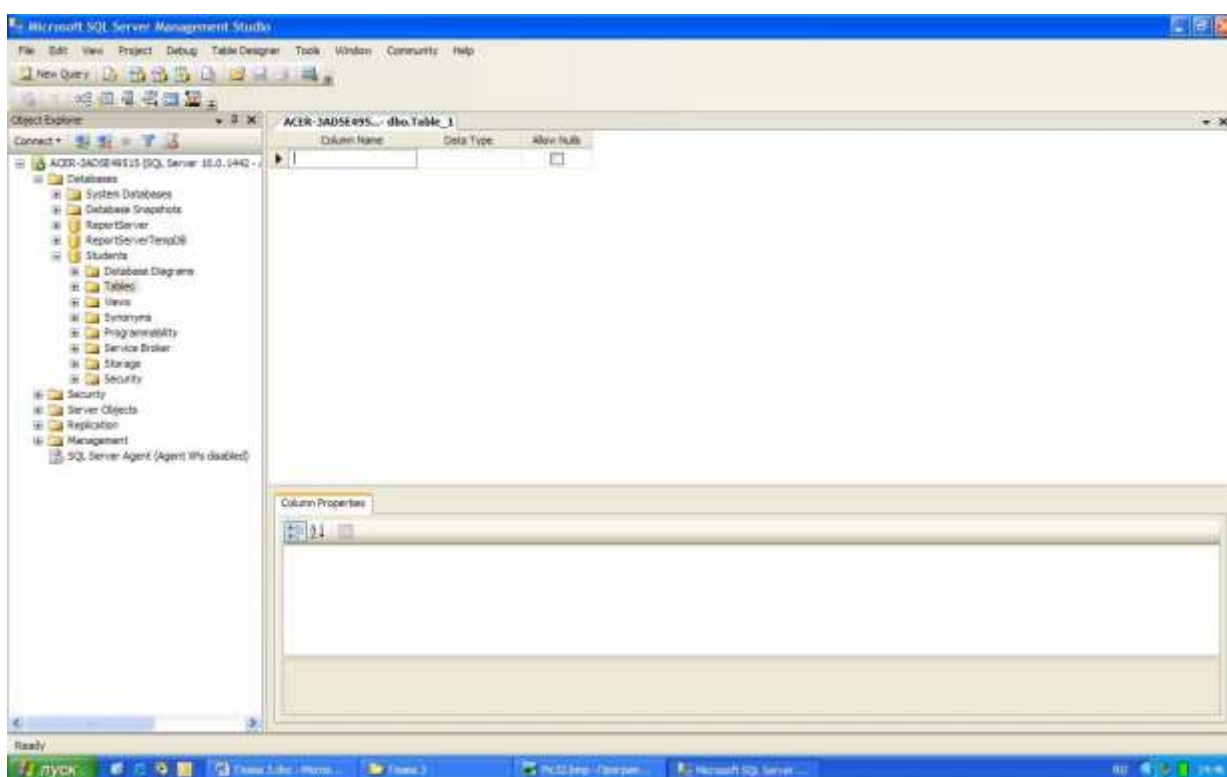


Рис.3.2

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

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

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

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

Рис.3.3

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

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

На этом настройку таблицы «Специальности» можно считать завершённой.



Рис.3.4

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



Рис.3.5

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

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

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

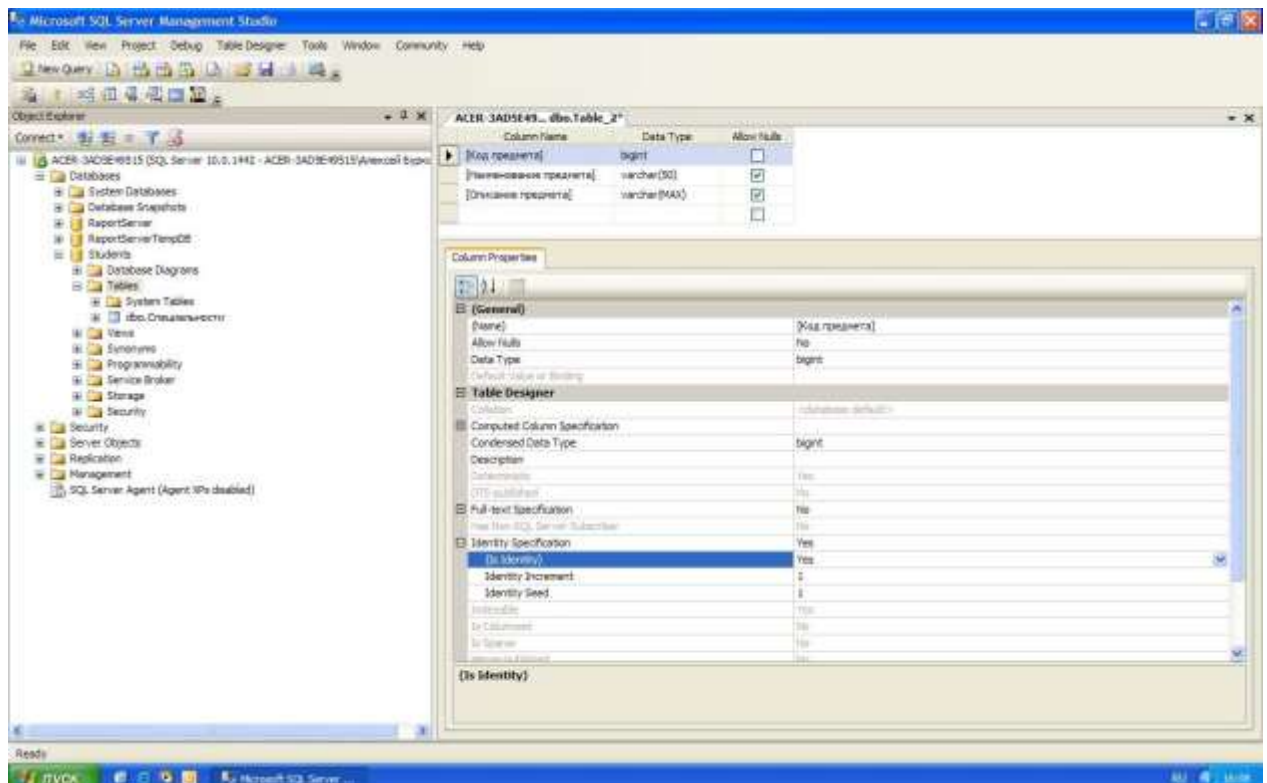


Рис.3.6

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



Рис.3.7

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

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

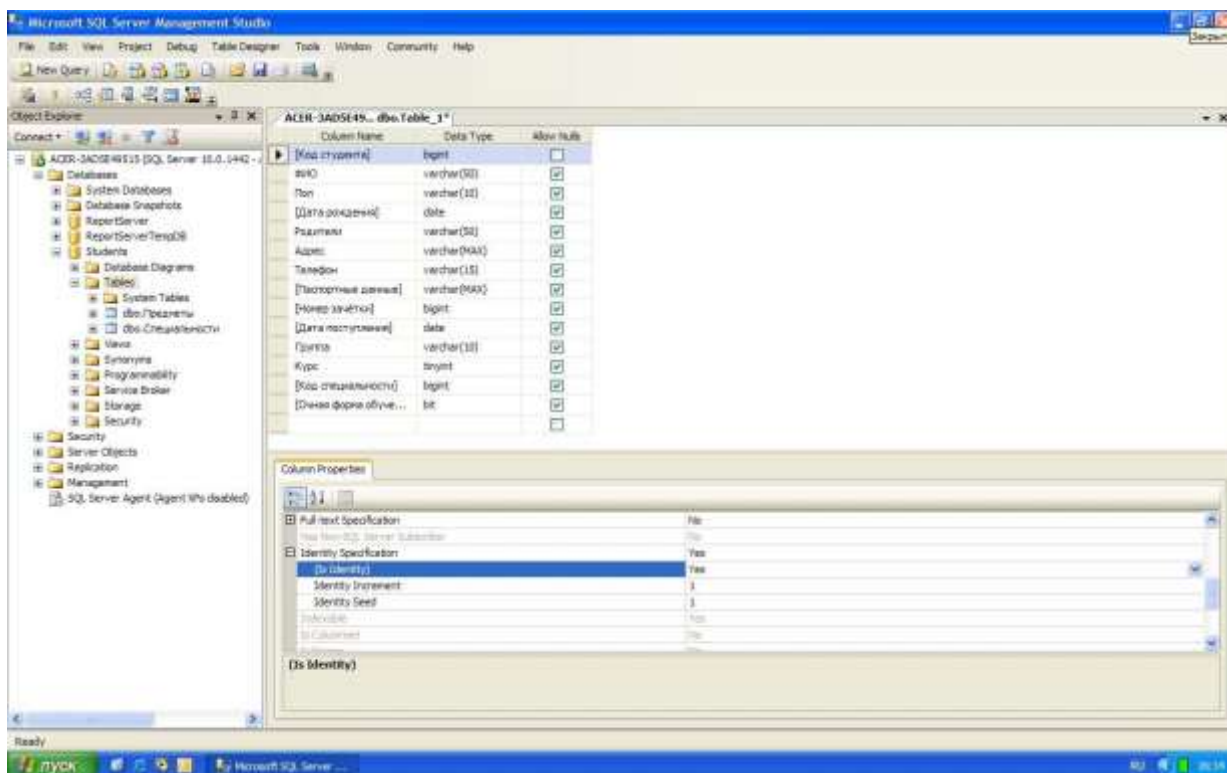


Рис.3.8

Рассматривая поля новой таблицы можно прийти к следующим выводам:

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

предназначено для хранения целых чисел в диапазоне 0...255 (тип данных «tinyint»);

- Поле «Код специальности» - это поле связи с таблицей «Специальности». Однако, данное поле связи является вторичным, поэтому его можно сделать просто целочисленным, то есть, «bigint».

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

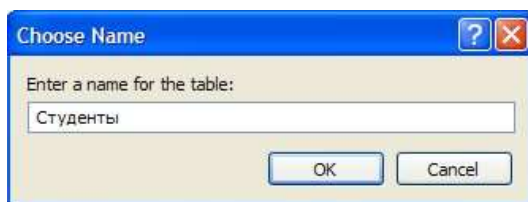


Рис.3.9

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

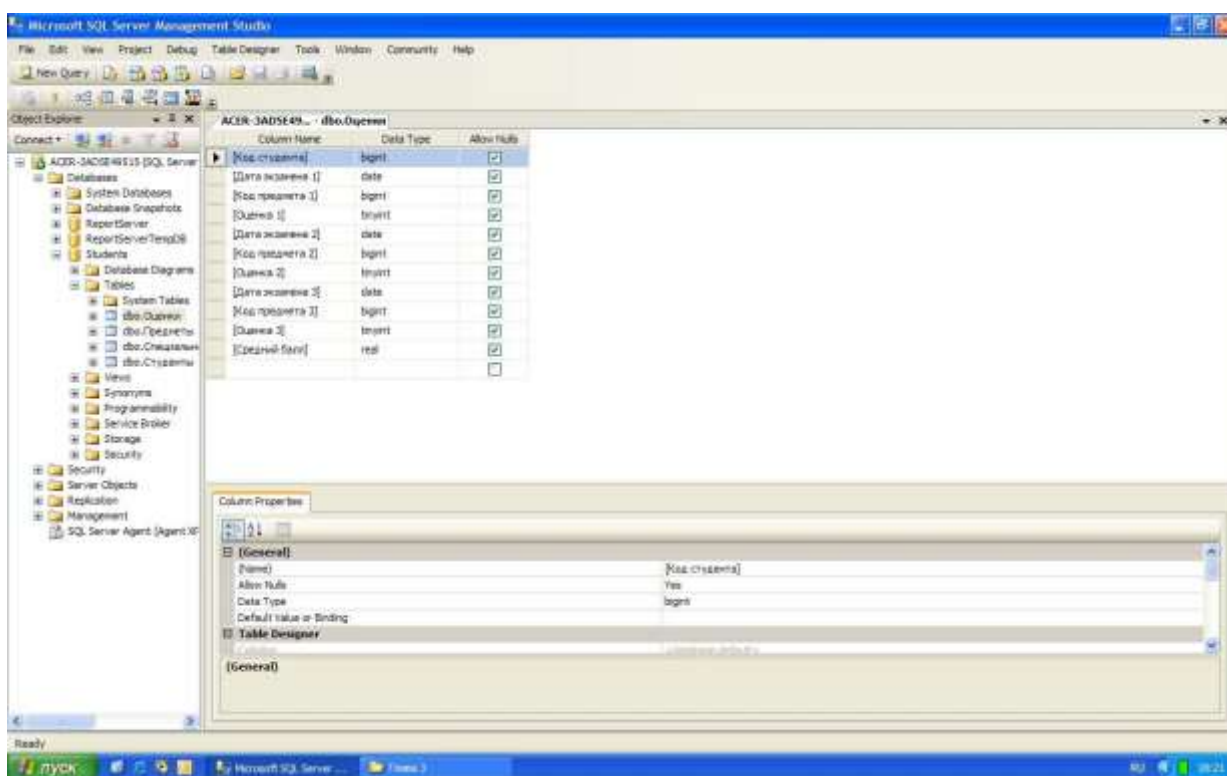


Рис.3.10

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

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



Рис.3.11

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

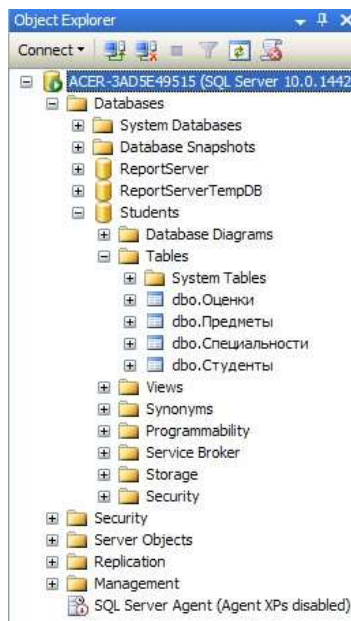


Рис.3.12


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

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

Рис.3.13

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

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

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

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

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

Рис.3.14

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

ACER-3AD5E49...dbо.Студенты														
	Код сту...	ФИО	Пол	Дата рожд...	Родители	Адрес	Телефон	Паспортные д...	Номер за...	Дата пост...	Группа	Курс	Код спец...	Очная форма
	1	Иванов А.И.	Мужской	1983-12-12	Отец, Мать	Москва	+74957895674	8567-567543	13245	2007-09-01	ММ11	1	1	True
	2	Петрова И.И.	Женский	1982-11-01	Мать	Москва	+74957889876	4567-765432	34563	2006-08-01	ПМ21	2	2	False
	3	Мухомов М.А.	Мужской	1982-05-14	Отец	Самара	+78462875690	5438-098787	56732	2006-07-05	СТ22	2	3	False
	4	Сидоров В.К.	Женский	1981-09-27	Нет	Саратов	+79027868909	1287-987509	27543	2005-06-23	МО31	3	4	True
	5	Кокеев-Кокеев А.А.	Мужской	1981-04-12	Мать	Казань	+79168863476	2312-675468	34217	2005-07-21	БМ33	3	5	True
	6	Пальчикова Н.Е.	Женский	1983-09-02	Отец, Мать	Челябинск	+74969090723	8743-856780	43278	2007-08-01	ММ12	1	1	False
	7	Царегородцев Е.В.	Мужской	1980-02-17	Отец	Самара	+78462234769	6543-834521	43765	2004-07-04	ПМ41	4	2	True
	8	Баранова Г.В.	Женский	1980-07-09	Отец, Мать	Чебоксары	+79027874638	2133-896567	10387	2004-08-09	СТ42	4	3	False
	9	Леукин П.Г.	Мужской	1979-02-26	Нет	Казань	+79067453678	2769-634904	67348	2003-07-23	МО51	5	4	True
▶	10	Николаева А.П.	Женский	1979-03-17	Мать	Саратов	+78546456432	3256-490932	45287	2003-06-21	БМ53	5	5	False
✖	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рис.3.15

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

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

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

Код студента	Дата экзамена 1	Код предмета 1	Оценка 1	Дата экзамена 2	Код предмета 2	Оценка 2	Дата экзамена 3	Код предмета 3	Оценка 3	Средний балл
1	2008-02-01	1	5	2008-02-09	4	3	2008-02-14	2	4	0
2	2008-01-30	5	4	2008-02-23	3	5	2008-02-27	1	5	0
3	2008-01-26	3	5	2008-02-05	1	3	2008-02-15	5	3	0
4	2007-12-26	2	3	2008-01-05	4	4	2008-01-21	3	4	0
5	2008-01-12	4	4	2008-01-18	5	4	2008-01-25	1	4	0
6	2007-12-17	2	4	2007-12-26	4	5	2008-01-05	1	3	0
7	2008-02-21	5	2	2008-02-29	1	2	2008-02-27	2	4	0
8	2008-02-03	3	3	2008-02-12	5	3	2008-02-20	4	5	0
9	2008-01-25	1	5	2008-02-02	3	5	2008-02-14	5	5	0
►	10	2007-12-28	4	2008-01-11	1	4	2008-01-23	2	3	0
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рис.3.16

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

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

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

Лабораторная работа. MS SQL Server. Построение запросов на выборку. Использование оператора SELECT.

Цель: научиться создавать запросы и фильтры

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

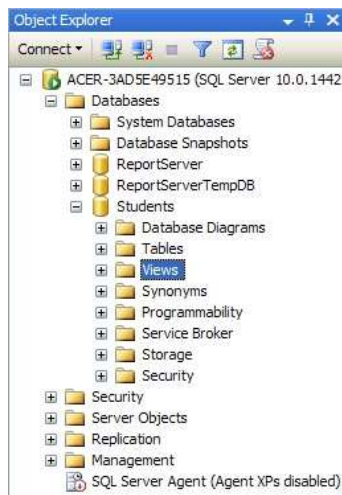


Рис.4.1

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



Рис.4.2

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

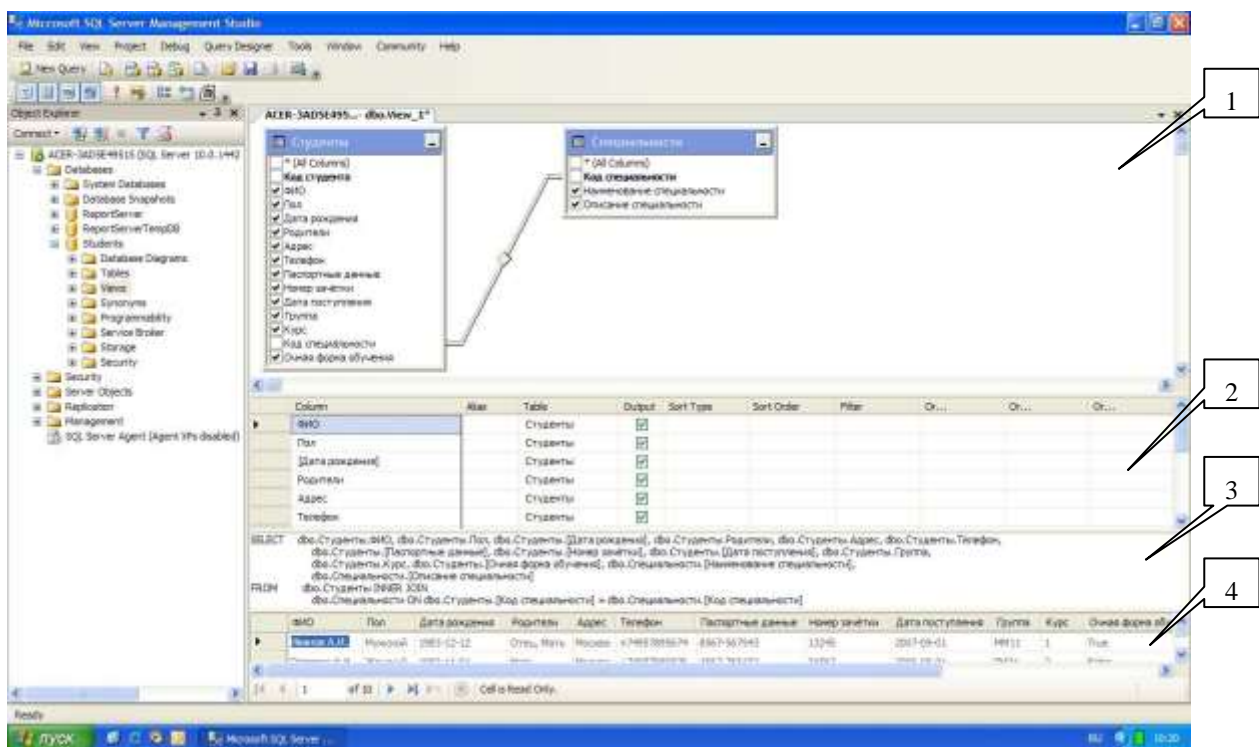







Рис.4.3

Замечание: Окно конструктора запросов состоит из следующих панелей:

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

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

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

Теперь перейдём к связыванию таблиц «Студенты» и «Специальности» по полям связи «Код специальности». Чтобы создать связь необходимо в схеме данных перетащить мышью поле «Код специальности» таблицы «Специальности» на такое же поле таблицы «Студенты». Связь отобразится в виде ломаной линии соединяющей эти два поля связи (Рис.4.3).

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


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

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


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

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

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

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

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

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

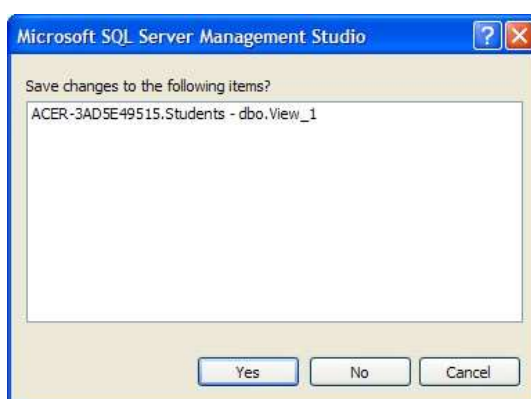


Рис.4.4

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

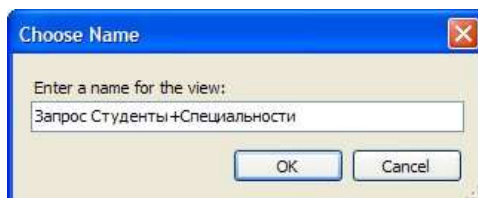


Рис.4.5

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

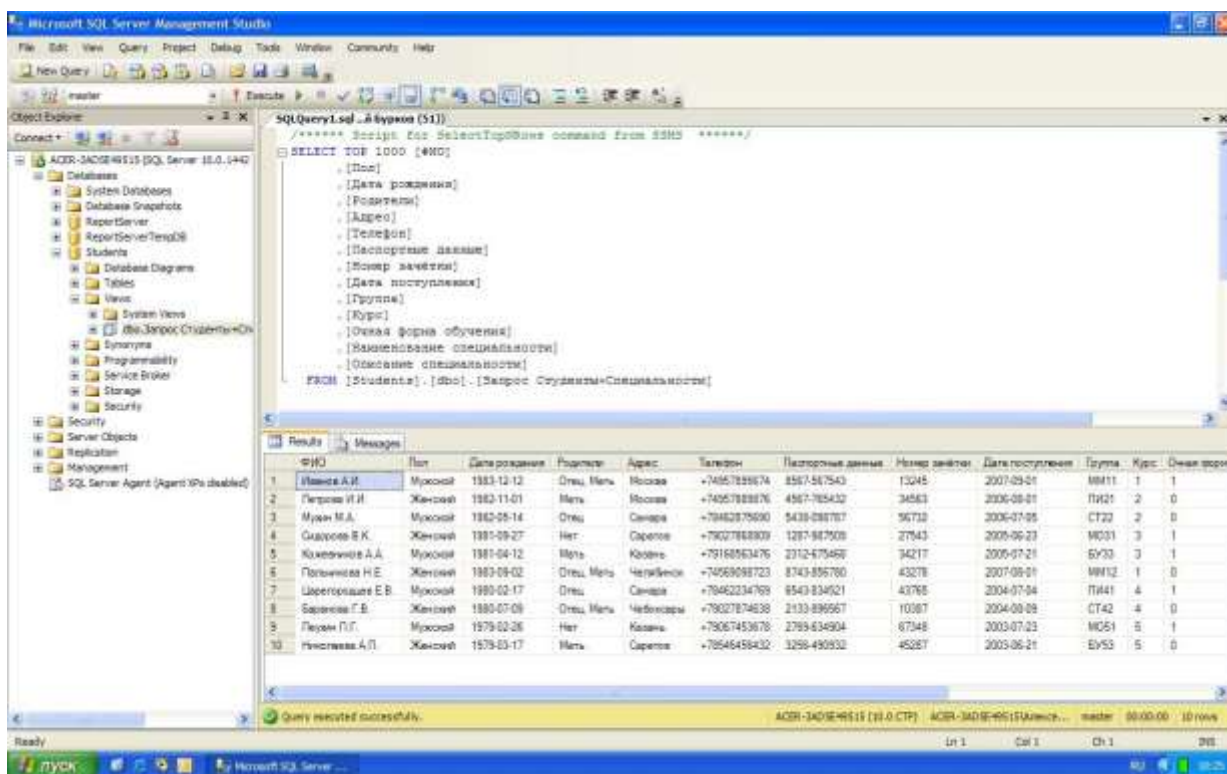


Рис.4.6

Проверим работоспособность созданного запроса вне конструктора запросов. Запустим вновь созданный запрос «Запрос Студенты+Специальности» без использования конструктора запросов. Для выполнения уже сохранённого запроса необходимо щёлкнуть 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 рисунке 4.6.

Перейдём 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». Появится окно «Add Table» (Рис.4.2).

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 трём полям. То есть поля «Код предмета 1», «Код предмета 2» и «Код предмета 3» таблицы «Оценки» связаны с полем «Код предмета» таблицы «Предметы». По этому 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 рисунке 4.7.

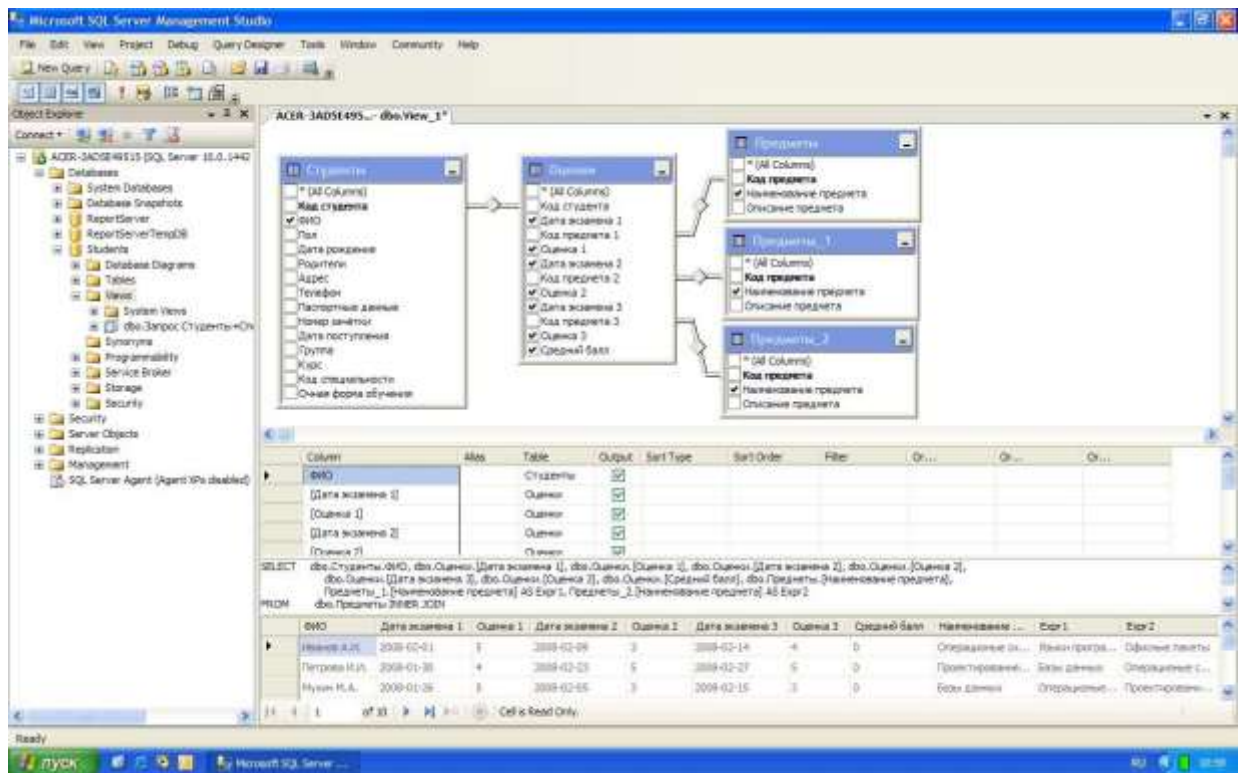


Рис.4.7

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

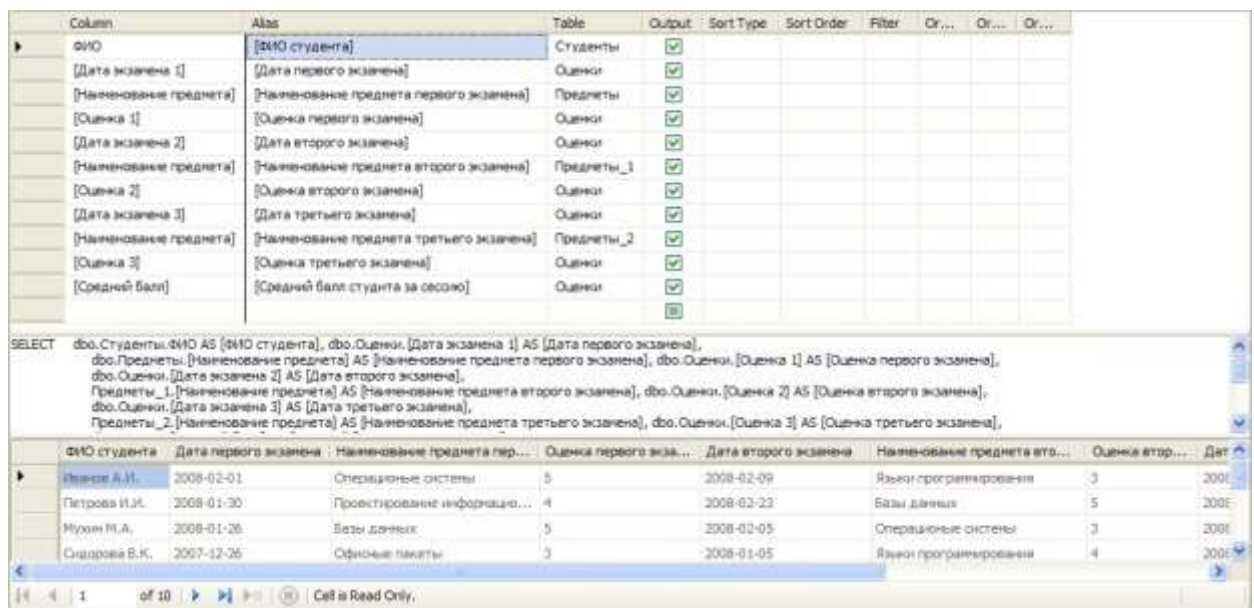


Рис.4.8

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

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



Рис.4.9

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

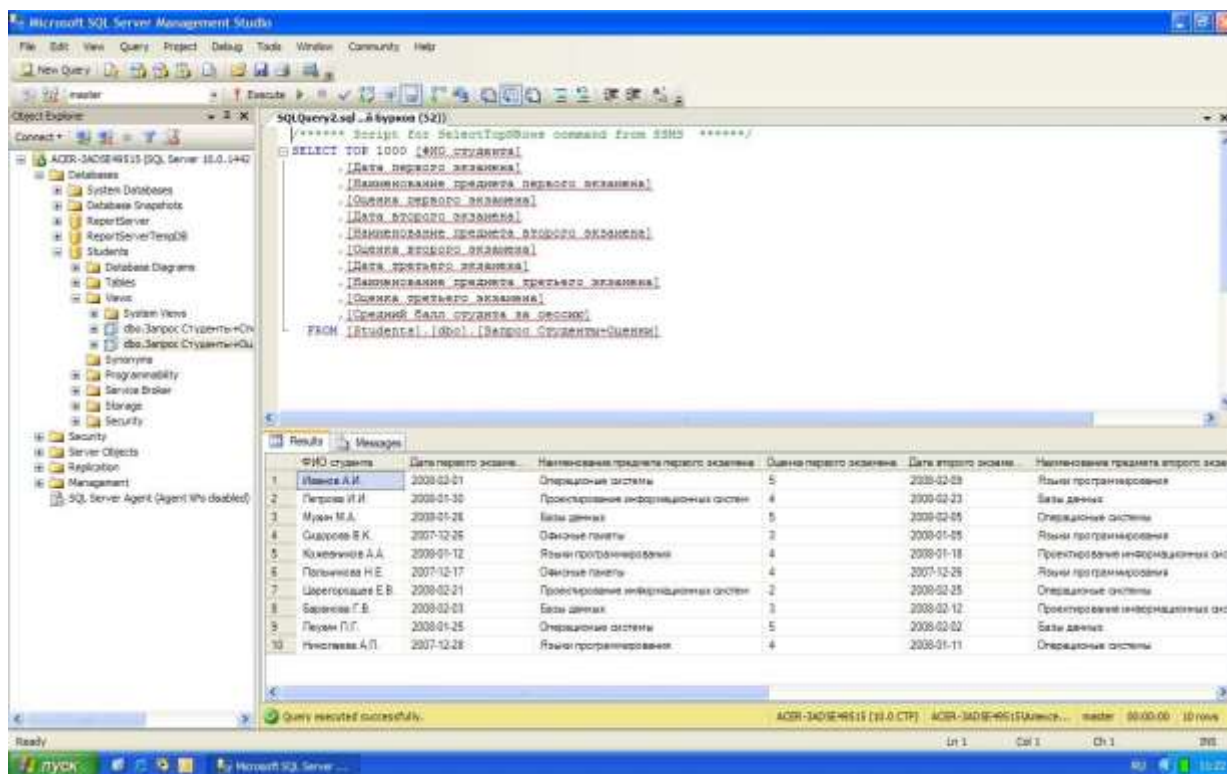


Рис.4.10

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

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



Рис.4.11

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

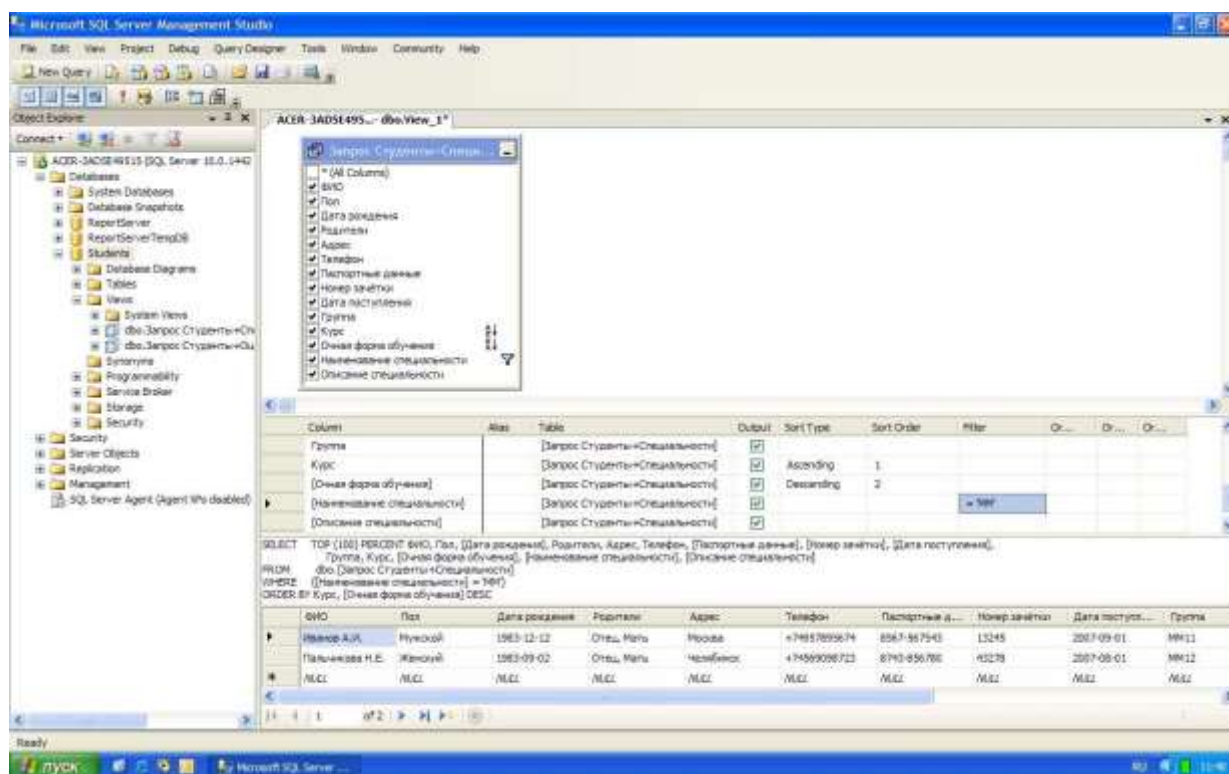


Рис.4.12

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

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

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

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

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

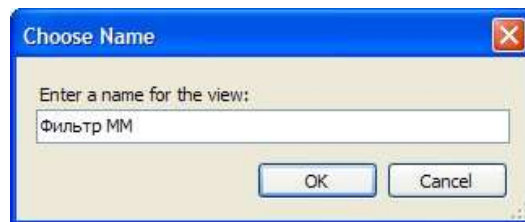


Рис.4.13

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

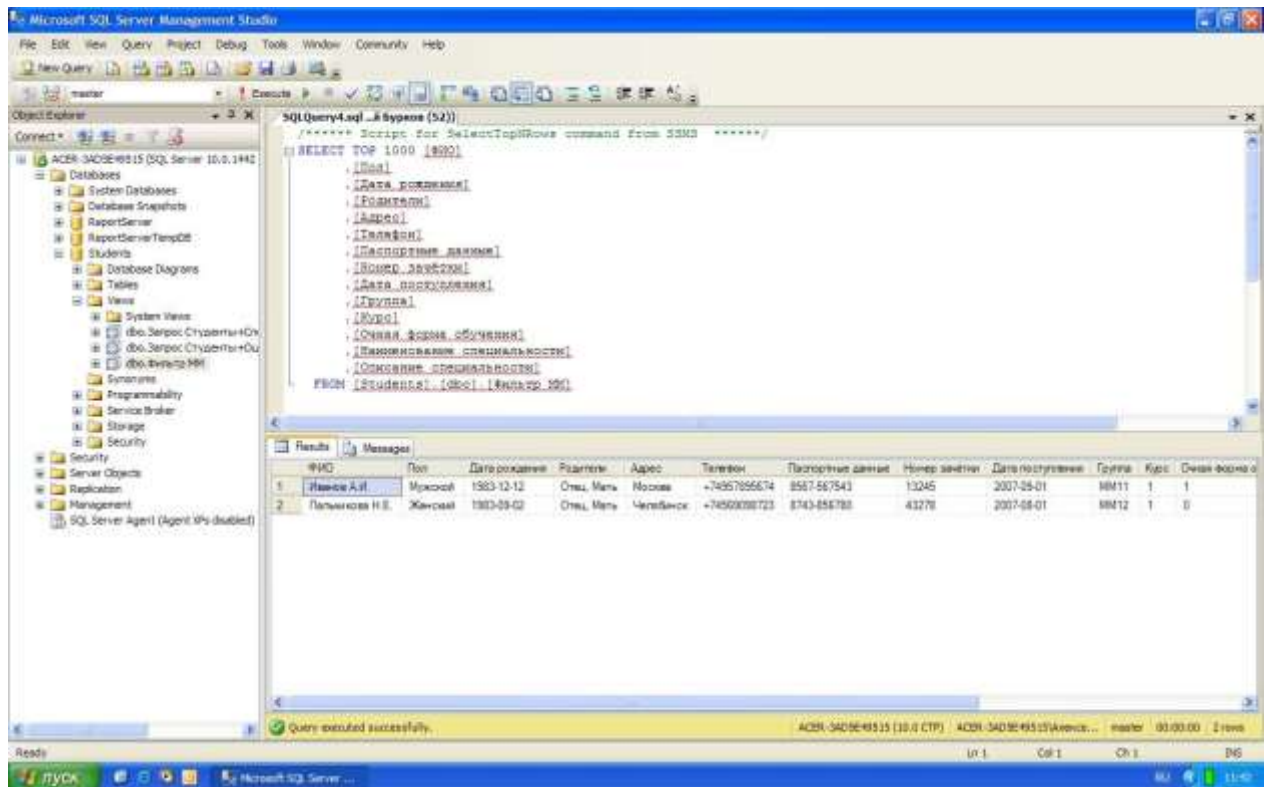


Рис.4.14

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

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

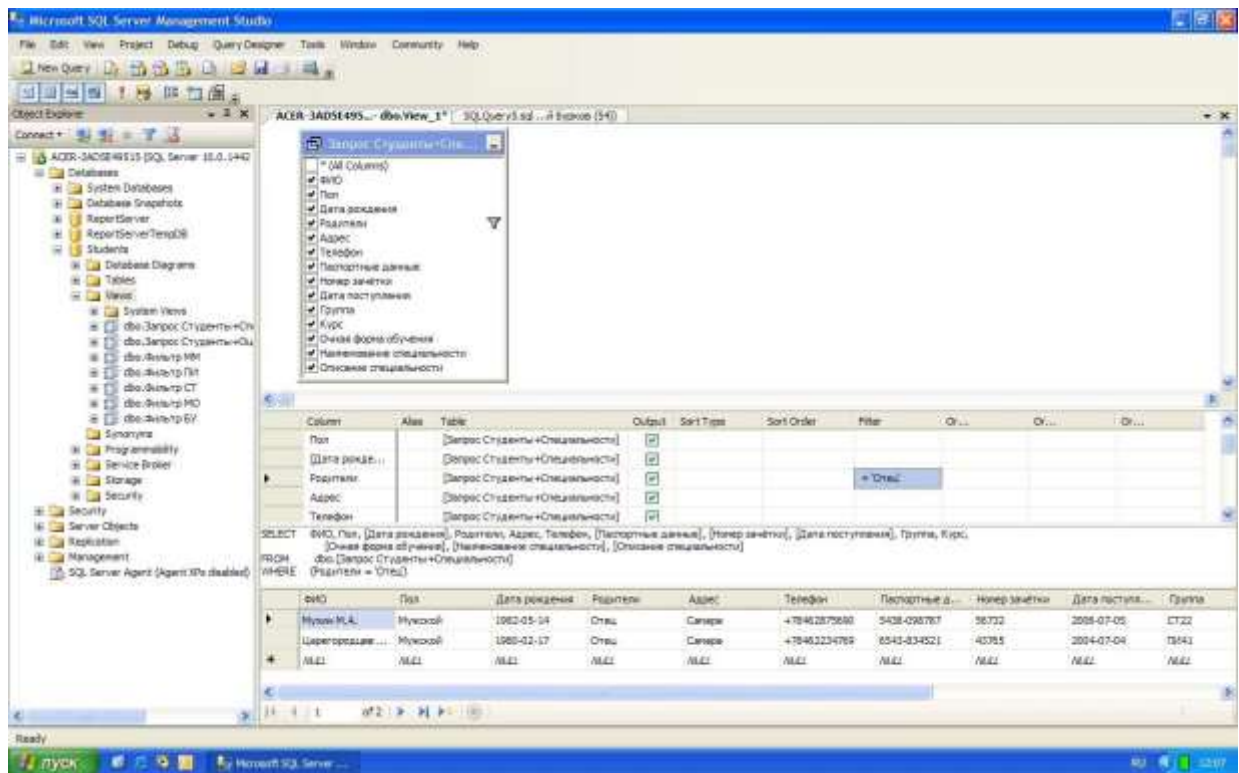


Рис.4.15

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

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

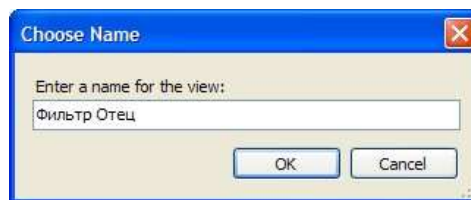


Рис.4.16

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

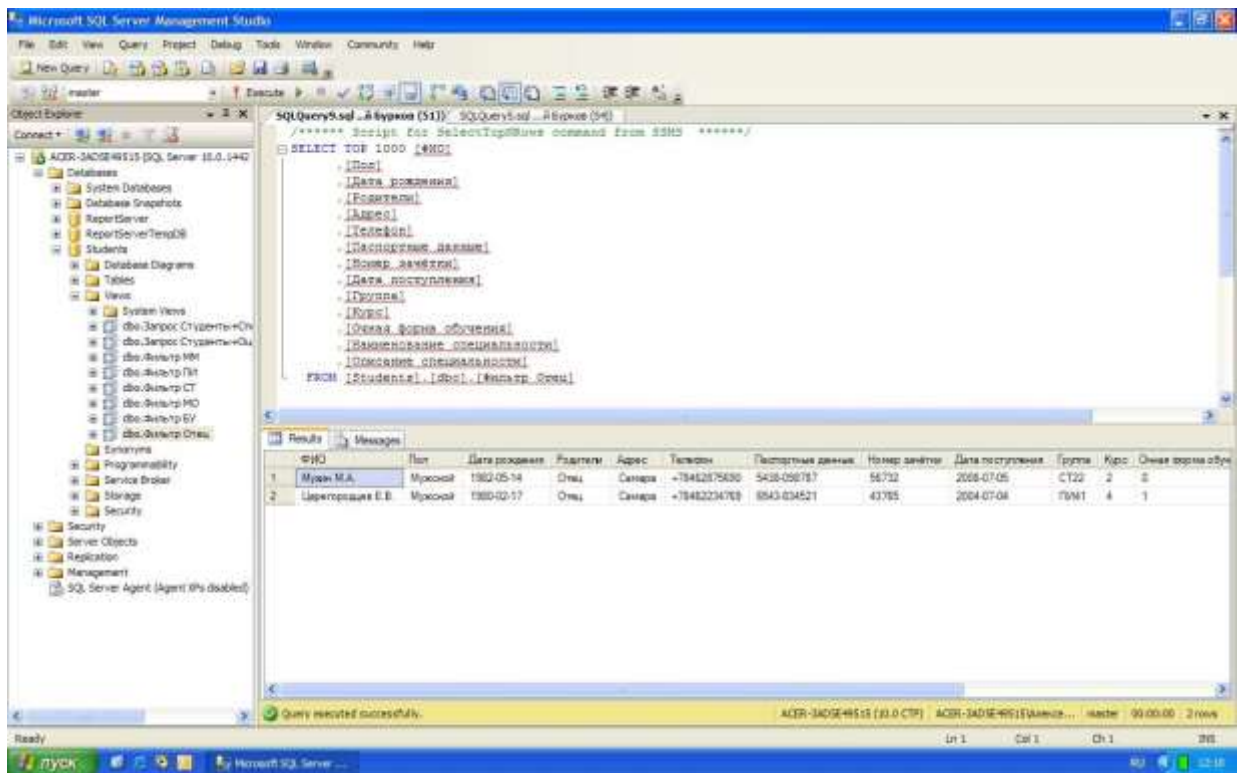


Рис.4.17

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

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

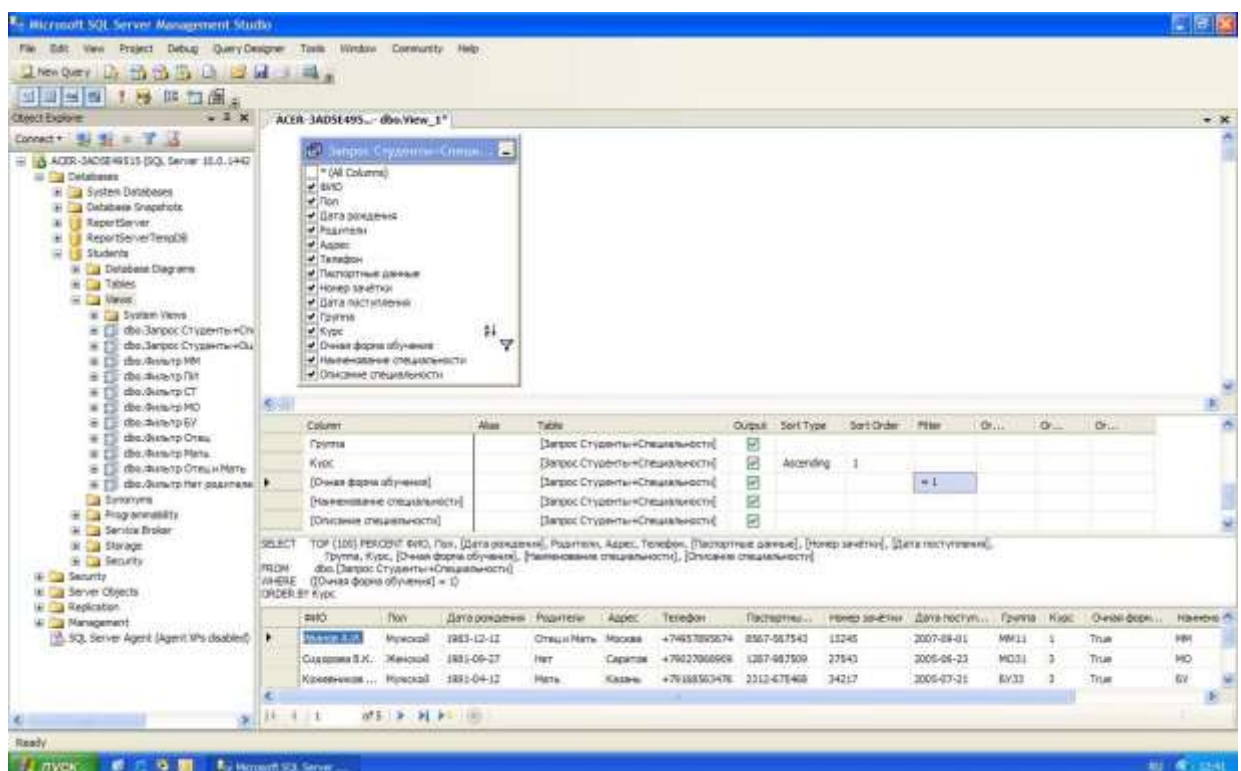


Рис.4.18

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

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

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

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

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

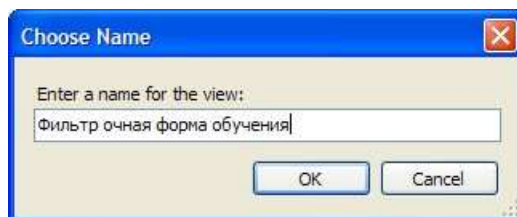


Рис.4.19

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

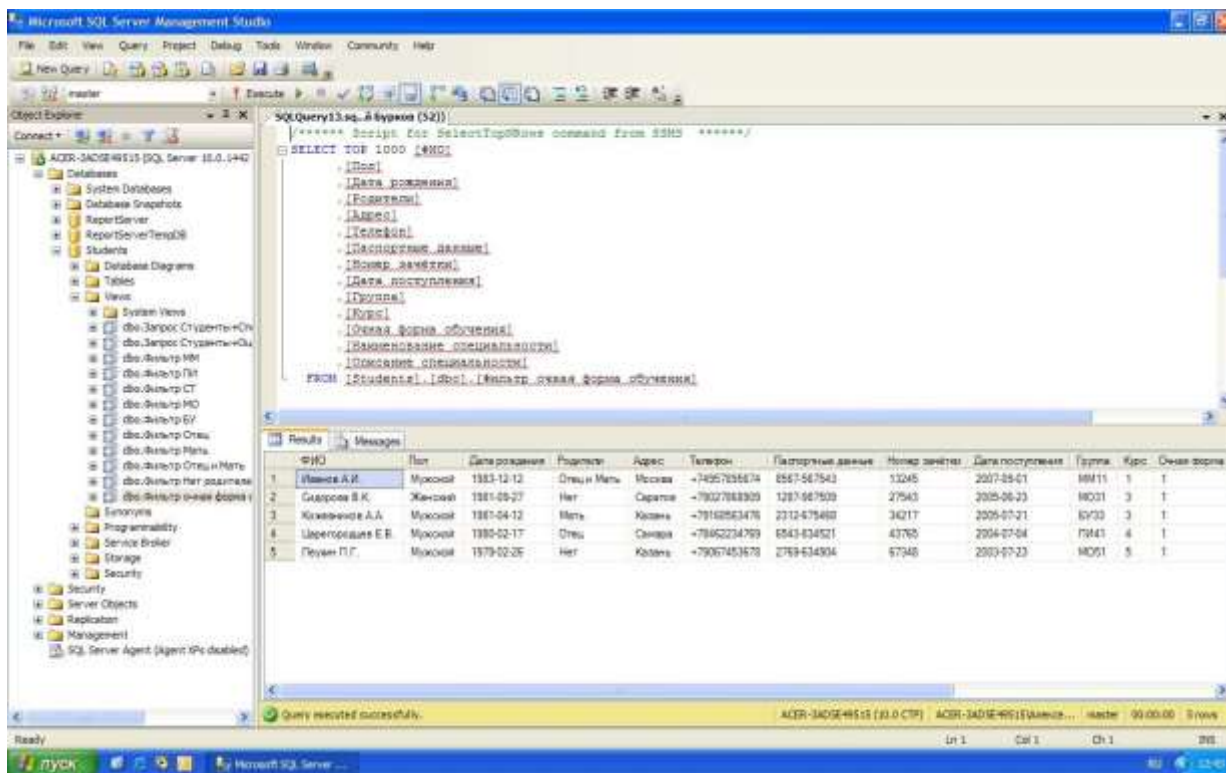


Рис.4.20

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

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

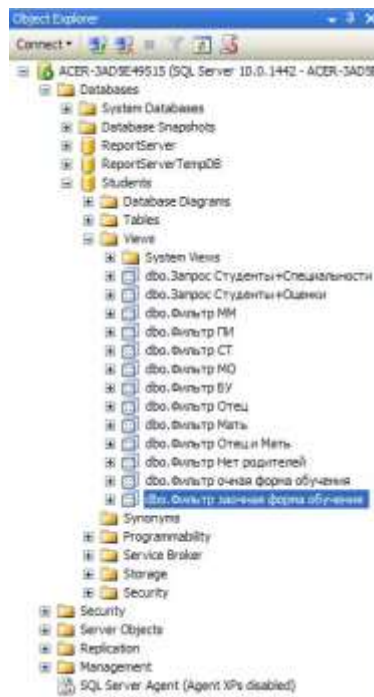


Рис.4.21

Лабораторная работа «Хранимые процедуры»

Цель: научиться работать с хранимыми процедурами

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

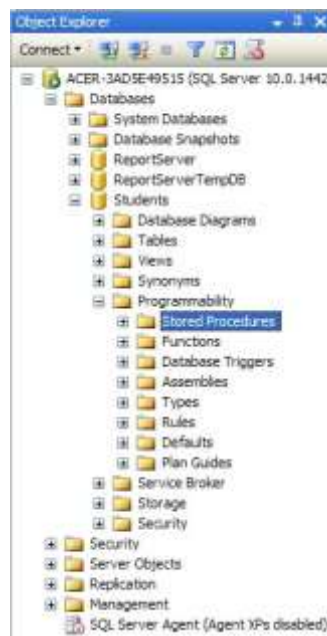


Рис.5.1

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

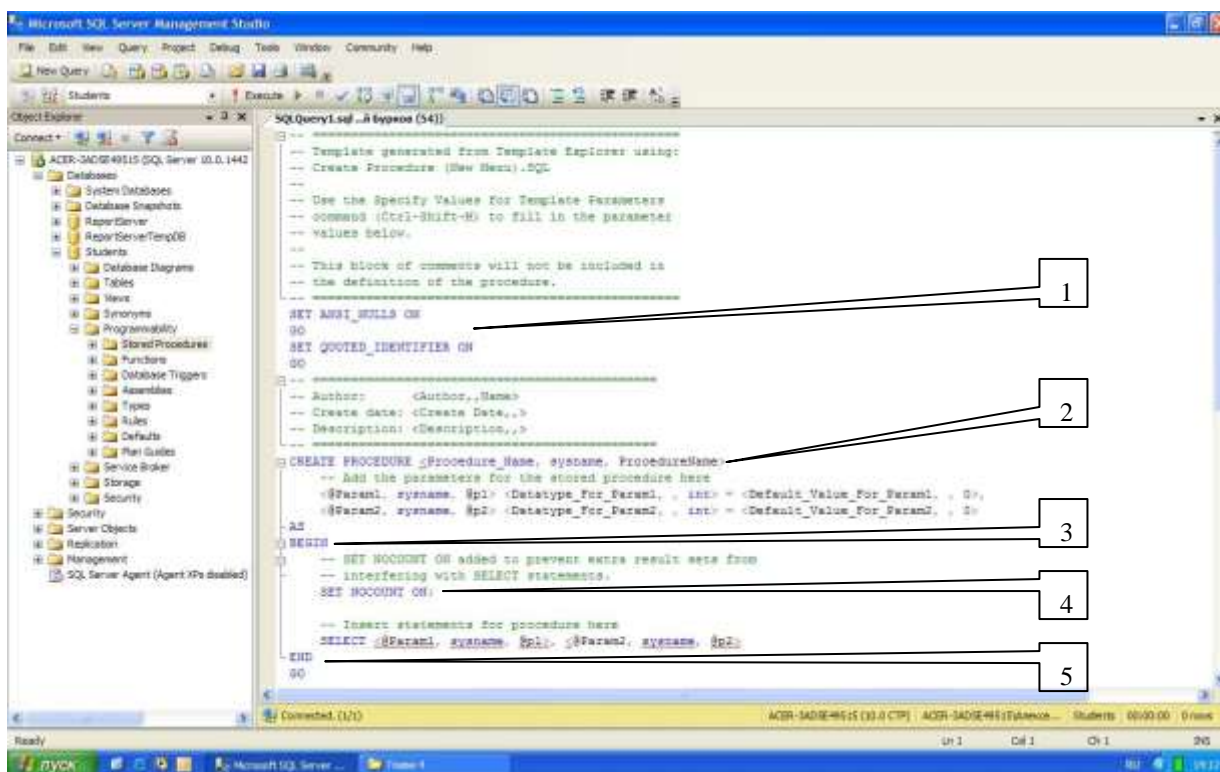


Рис.5.2

Хранимая процедура имеет следующую структуру (Рис.5.2):

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

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

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

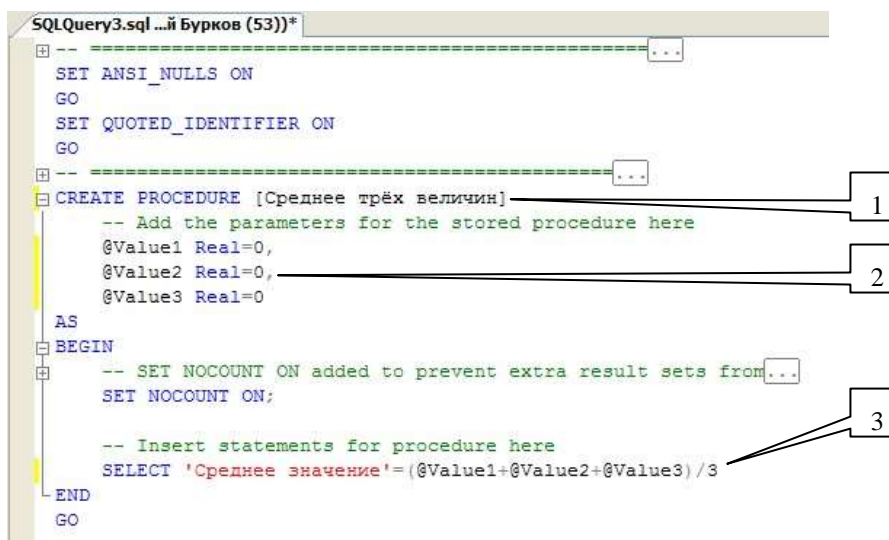
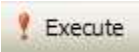



Рис.5.3

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


1. CREATE PROCEDURE [Среднее трёх величин] – определяет имя создаваемой процедуры как «Среднее трёх величин»;
2. @Value1 Real = 0, @Value2 Real = 0, @Value3 Real = 0 - определяют три параметра процедуры Value1, Value2 и Value3. Данным параметрам можно присвоить дробные числа (Тип данных Real), значения по умолчанию равны 0;
3. SELECT 'Среднее значение'=(@Value1+@Value2+@Value3)/3 – вычисляет среднее и выводит результат с подписью «Среднее значение». Остальные фрагменты кода рассмотрены выше (Рис.5.2).

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

Проверим работоспособность созданной хранимой процедуры. Для запуска хранимой процедуры необходимо создать новый пустой запрос, нажав на кнопку



(Новый запрос) на панели инструментов. В появившемся окне с пустым

запросом наберите команду EXEC [Среднее трёх величин] 1, 7, 9 и нажмите кнопку  Execute на панели инструментов (Рис.5.4).

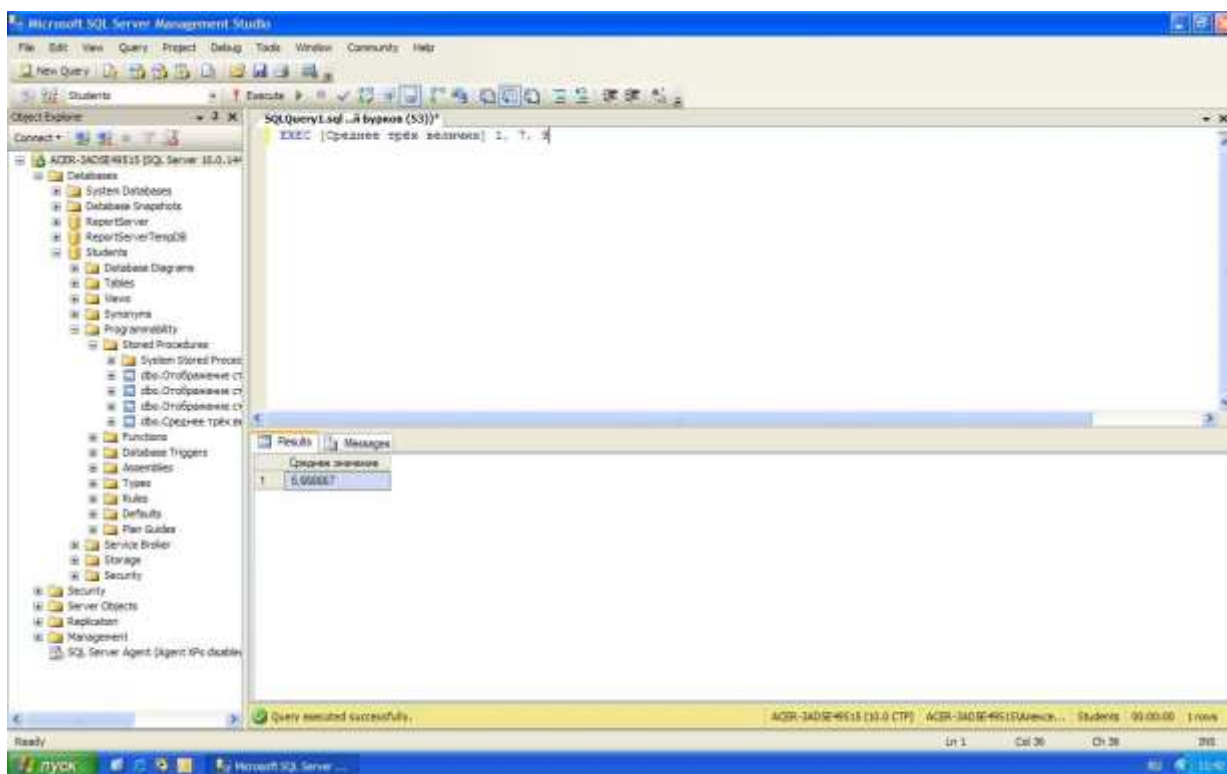


Рис.5.4

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

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

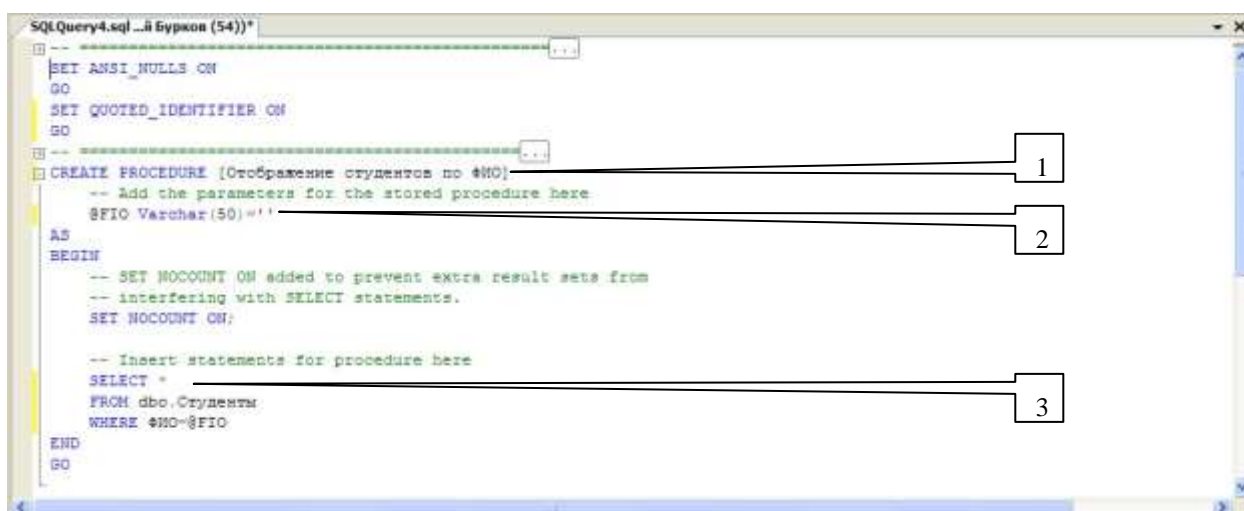
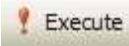


Рис.5.5

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

1. CREATE PROCEDURE [Отображение студентов по ФИО] – определяет имя создаваемой процедуры как «Отображение студентов по ФИО»;
2. @FIO Varchar(50)='' - определяют единственный параметр процедуры FIO. Параметру можно присвоить текстовые строки переменной длины, длиной 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  на панели инструментов (Рис.5.6).

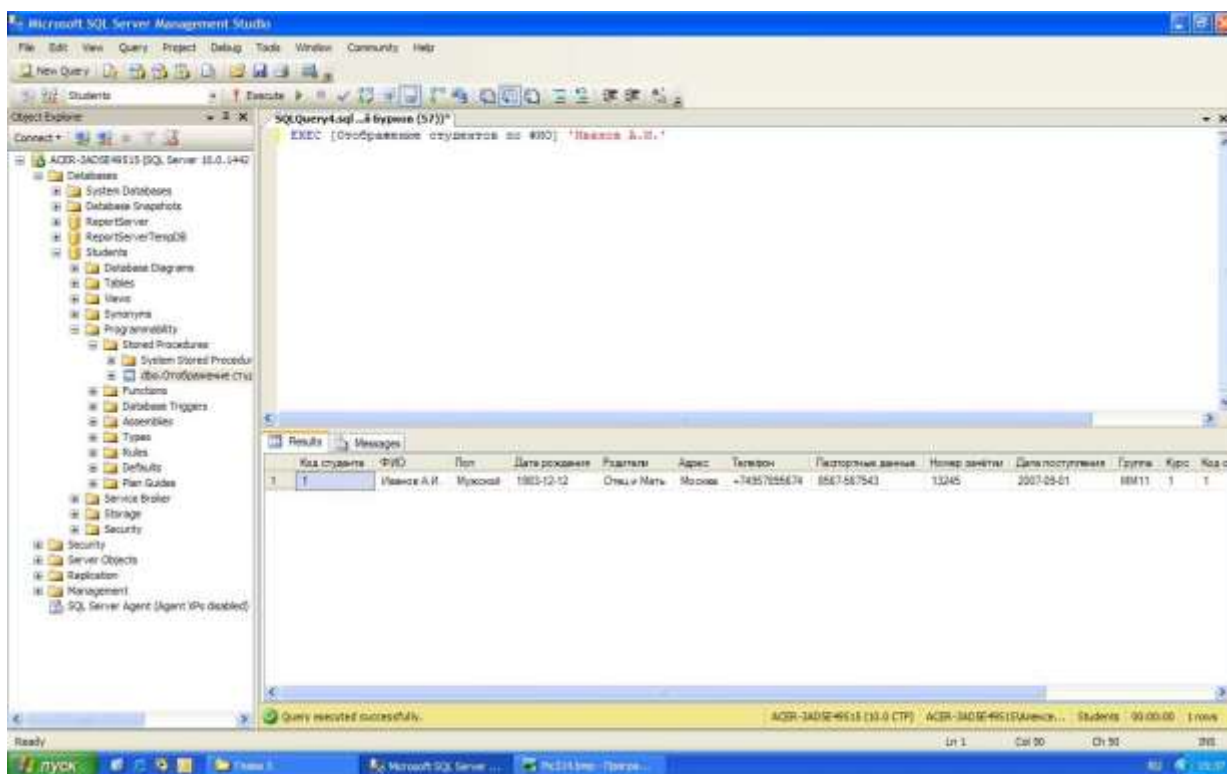


Рис.5.6

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

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

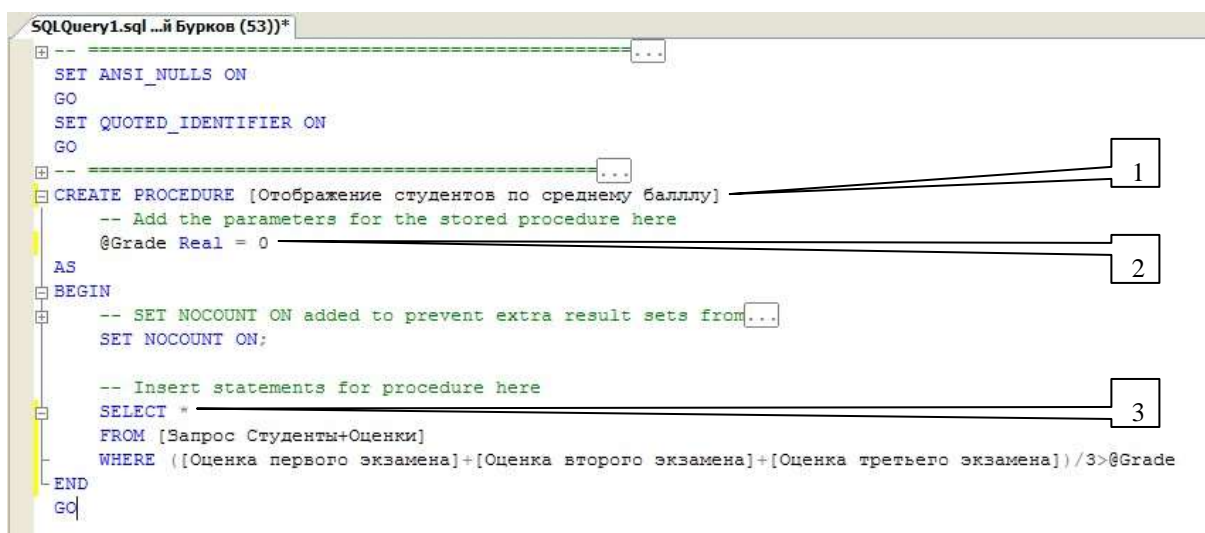


Рис.5.7

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

1. CREATE PROCEDURE [Отображение студентов по среднему баллу] – определяет имя создаваемой процедуры как «Отображение студентов по среднему баллу»;

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

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

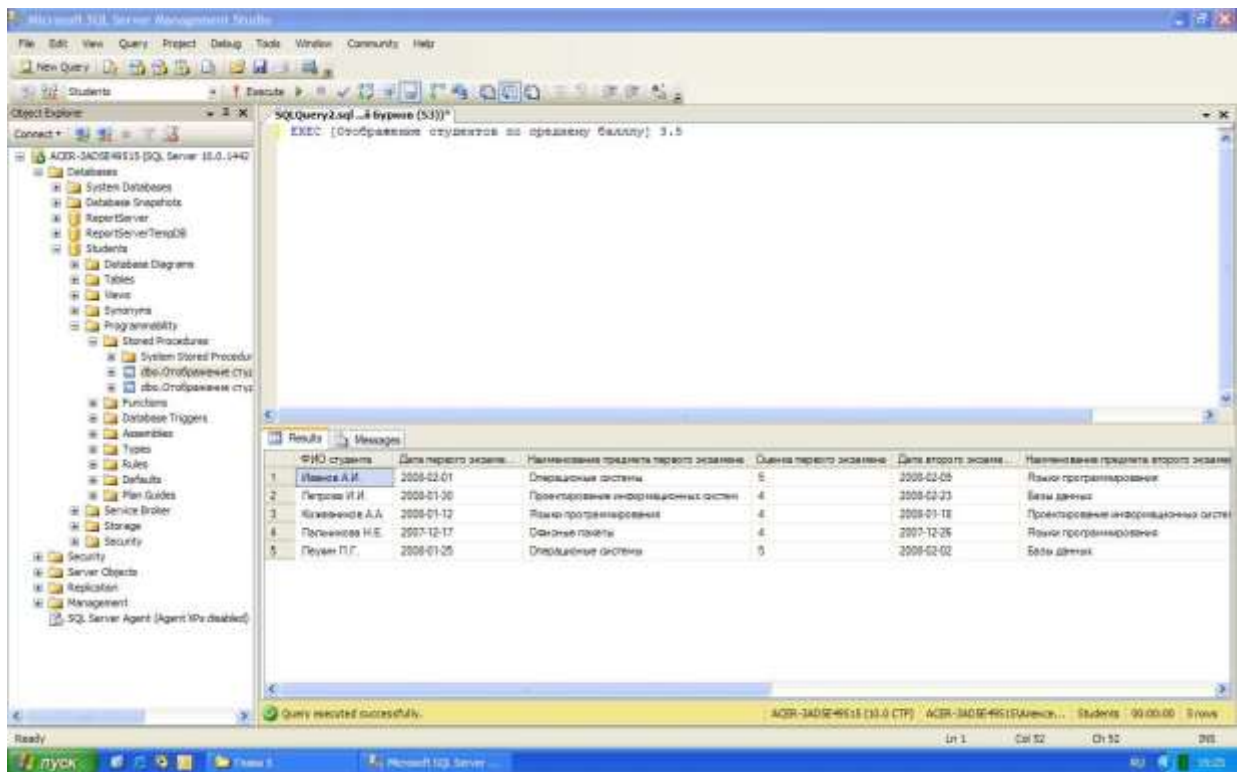


Рис.5.8

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

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

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

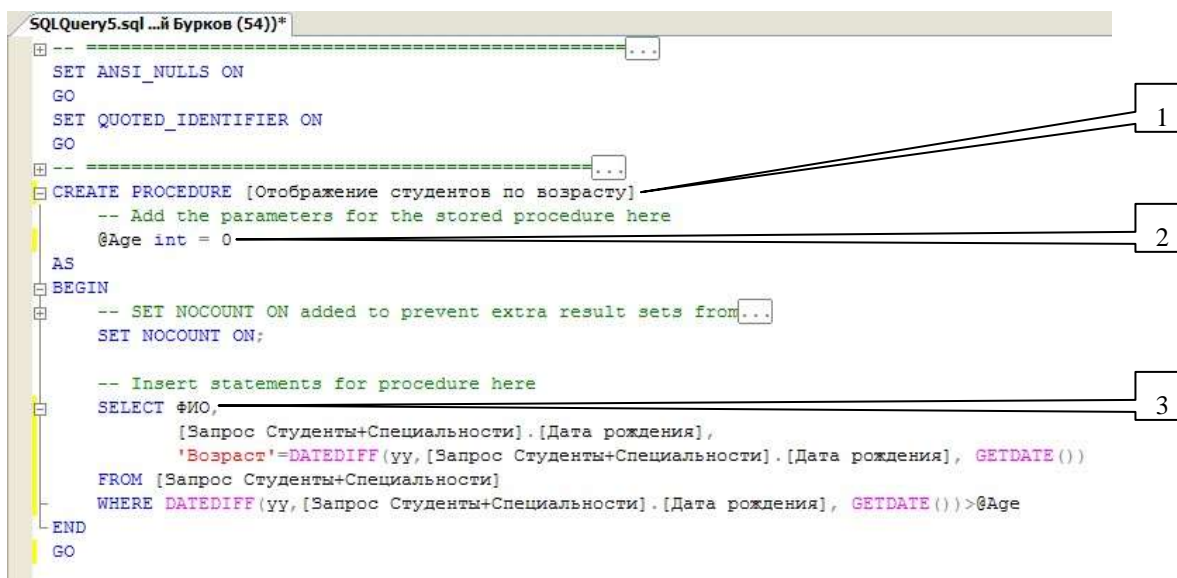


Рис.5.9

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

1. CREATE PROCEDURE [Отображение студентов по возрасту] – определяет имя создаваемой процедуры как «Отображение студентов по возрасту»;
2. @Age int=0 - определяют параметр процедуры Grade. Параметру можно присвоить целые числа (Тип данных int), значения по умолчанию равны 0;
3. ФИО, [Запрос Студенты+Специальности].[Дата рождения], 'Возраст'=DATEDIFF(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 определённого 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 нём наберём команду EXEC [Отображение студентов по возрасту] 26 и выполните её. Должен появиться результат аналогичный результату, представленному на рисунке 5.10.

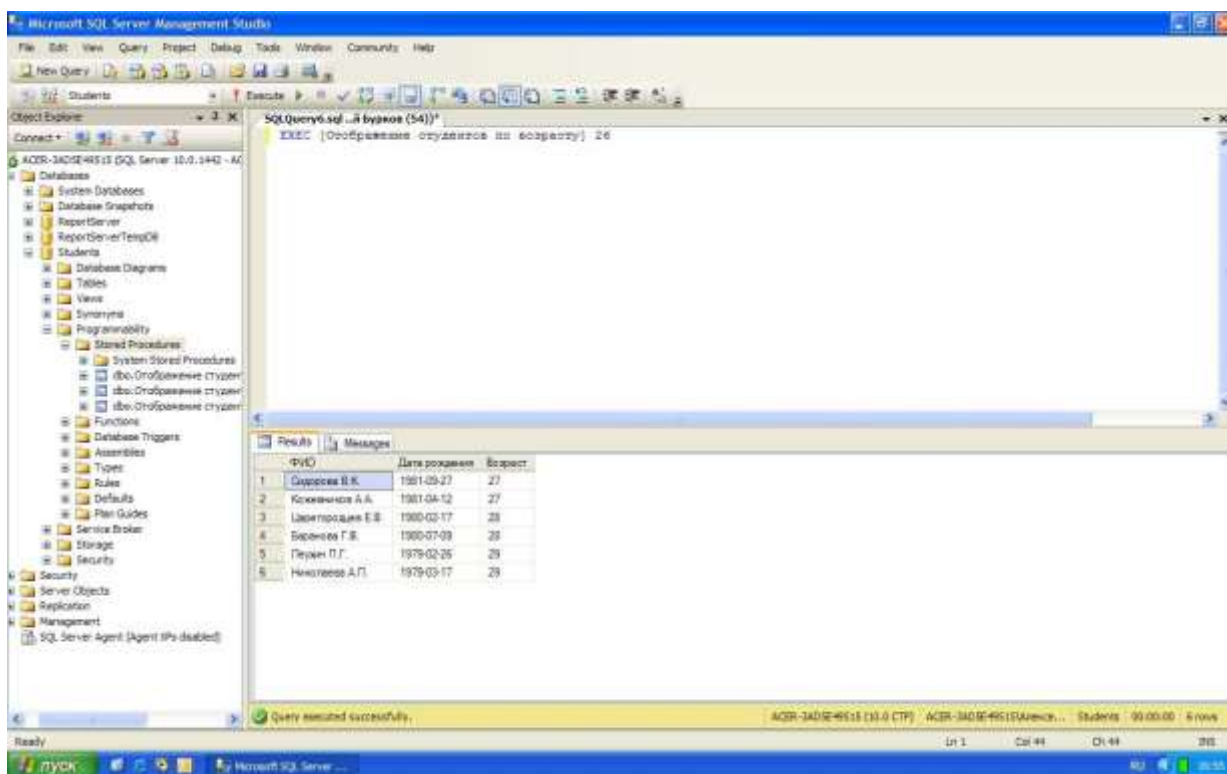


Рис.5.10

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

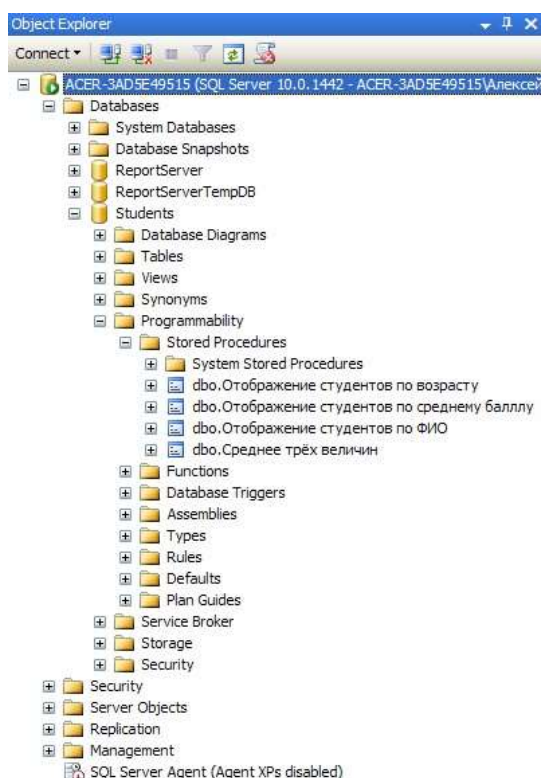


Рис.5.11

Лабораторная работа. «Пользовательские функции»

Цель: научиться работать с пользовательскими функциями

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

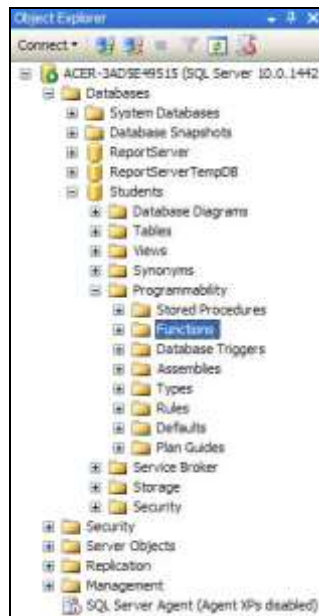


Рис.6.1

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

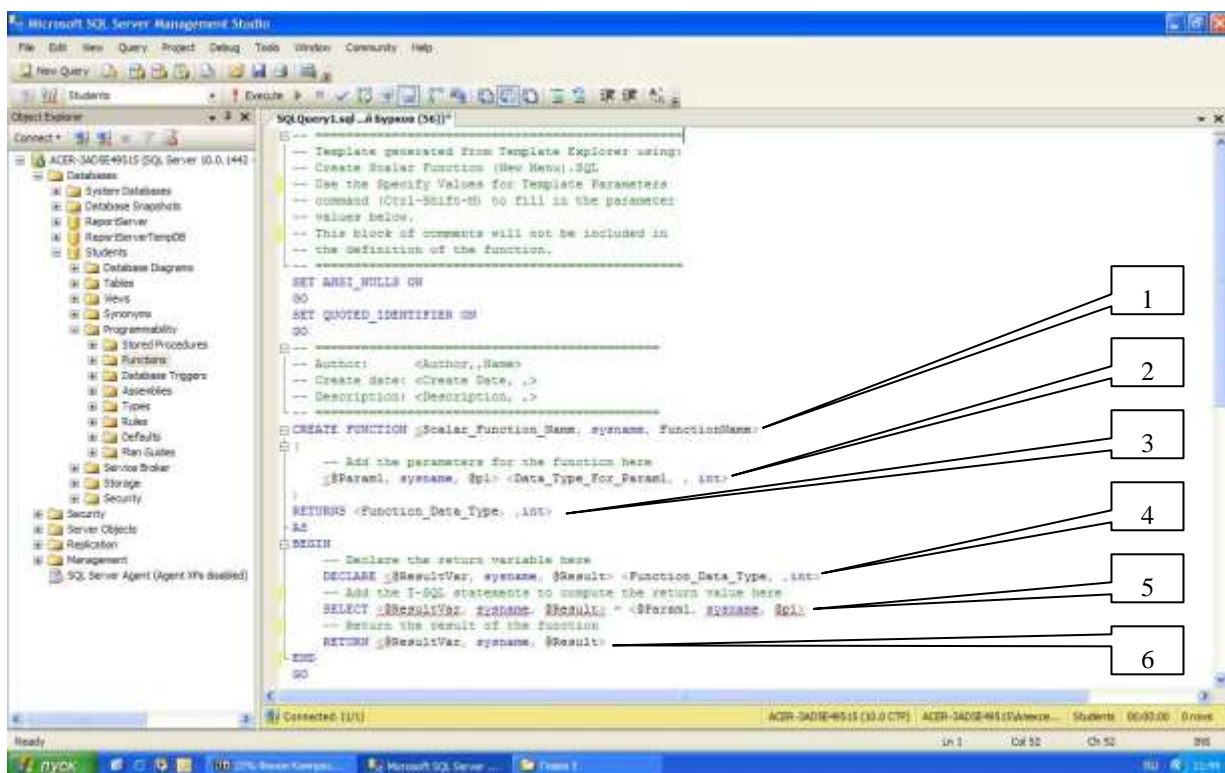


Рис.6.2

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

6. Область определения имени функции (Scalar_Function_Name);
7. Параметры, передаваемые в процедуру (@Param1). Определение параметров аналогично определению параметров в хранимой процедуре (см. занятие 5);
8. Тип данных значения возвращаемого процедурой;
9. Область объявления переменных, используемых внутри функции.

Объявление переменных имеет следующий синтаксис:

DECLARE @<Имя переменной> <Тип данных>

10. Тело самой пользовательской функции, содержит команды языка программирования запросов T-SQL;

11. Команда RETURN возвращающая результат выполнения функции. Имеет следующий синтаксис:

RETURN @<Имя переменной с результатом>

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

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

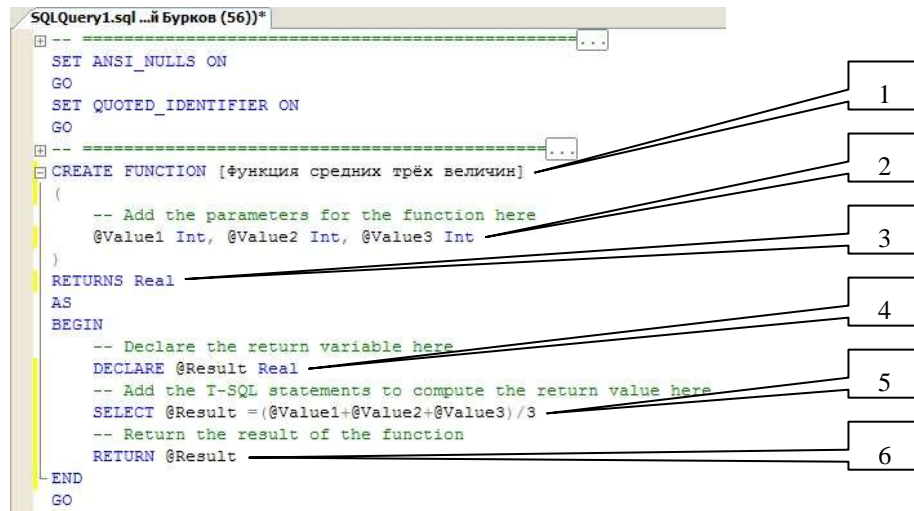
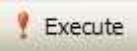




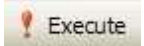
Рис.6.3

Рассмотрим более подробно код данной скалярной пользовательской функции

(Рис.6.3):

4. CREATE FUNCTION [Функция средних трёх величин] – определяет имя создаваемой функции как «Функция средних трёх величин»;
5. @Value1 Real, @Value2, @Value3 - определяют три параметра процедуры Value1, Value2 и Value3. Данным параметрам можно присвоить целые числа (Тип данных Int);
6. RETURNS Real – показывает, что функция возвращает дробные числа (Тип данных Real);
7. DECLARE @Result Real – объявляется переменная @Result для хранения результата работы функции, то есть дробного числа (Тип данных Real);
8. SELECT @Result=(@Value1+@Value2+@Value3)/3 – вычисляет среднее и помещает результат в переменную @Result;
9. RETURN @Result – возвращает значение переменной @Result. Остальные фрагменты кода рассмотрены выше (Рис.6.2).

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

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

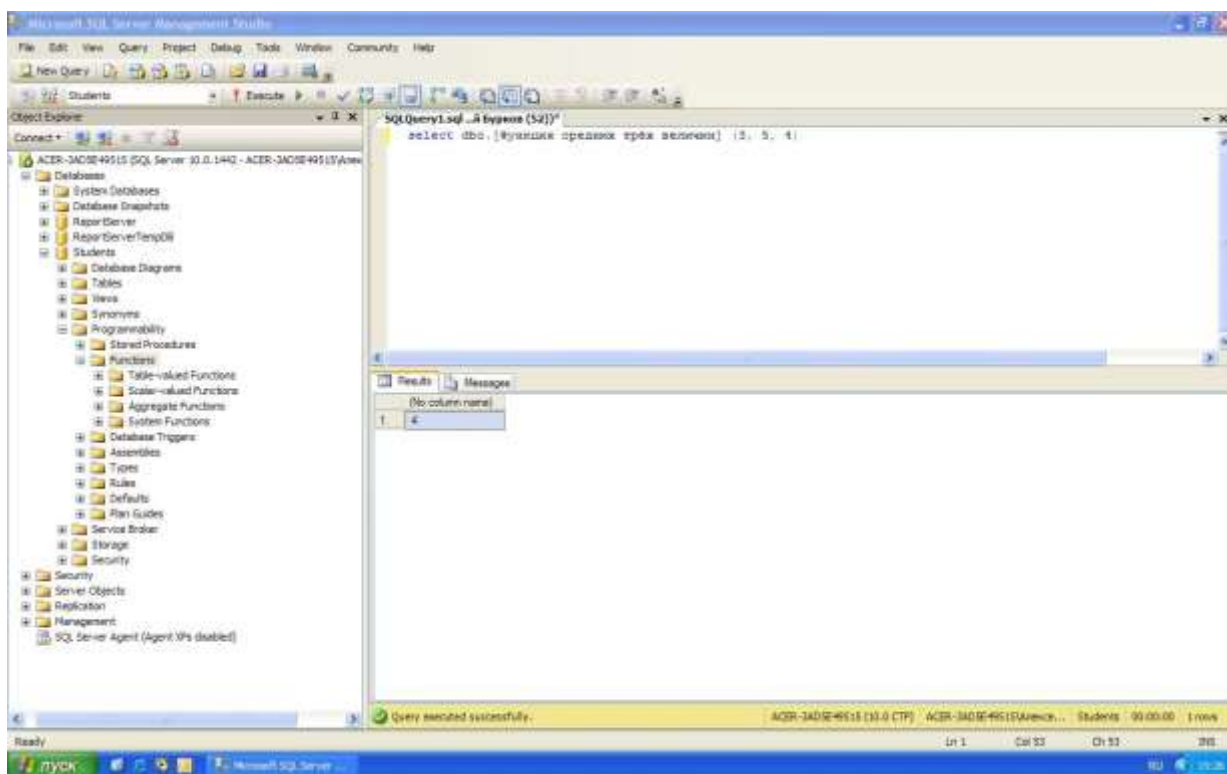


Рис.6.4

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

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

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

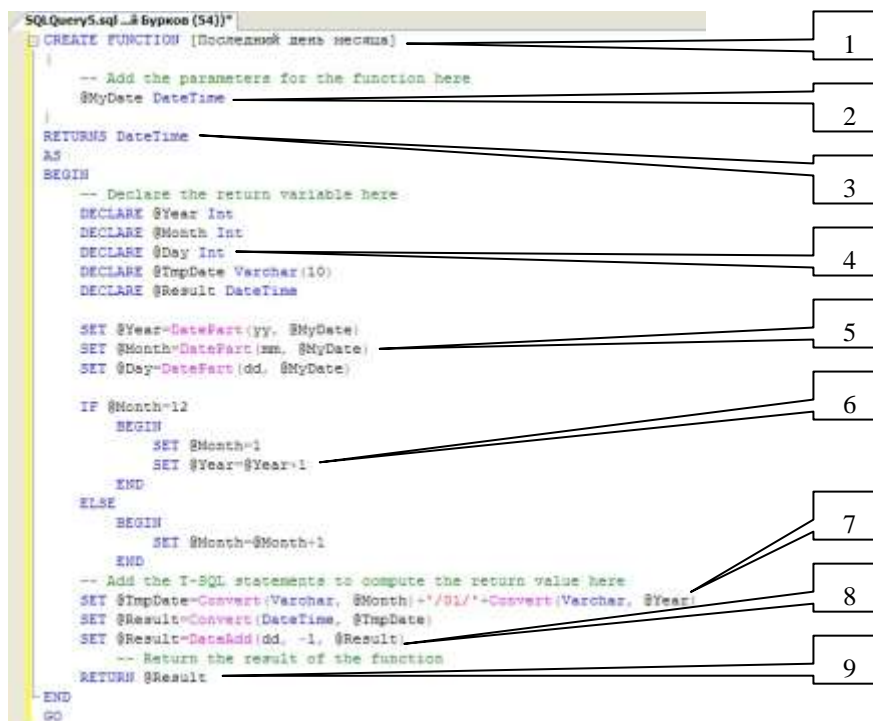


Рис.6.5

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

1. `CREATE FUNCTION [Последний день месяца]` – определяет имя создаваемой функции как «Последний день месяца»;

2. @MyDate - определяют параметр процедуры MyDate. Параметру можно присвоить значения дат или времени (Тип данных DateTime);
3. RETURNS DateTime – показывает, что функция возвращает дату или время (Тип данных DateTime);
4. DECLARE @Year Int, DECLARE @Month Int, DECLARE @Day Int– объявляются переменные @Year, @Month и @Day для хранения целочисленных значений года, месяца и дня введенной даты (Тип данных Int).

DECLARE @TmpDate VarChar(10) объявляет переменную «TmpDate» для хранения промежуточного значения даты в строке длиной до 10 символов (Тип данных VarChar(10)).

DECLARE @Result DateTime объявляет переменную «Result» для хранения результата – даты последнего дня месяца (Тип данных DateTime).

5. SET @Year=DatePart(yy, @MyDate), SET @Month=DatePart(mm, @MyDate), SET @Day=DatePart(dd, @MyDate) – определяются части введенной даты и помещаются в переменные @Year, @Month и @Day. Для определения частей даты используется функция DatePart, имеющая следующий синтаксис: DatePart(<часть даты>, <дата>). Здесь «часть даты» - это закодированная специальными символами определяемая часть даты (yy – год, mm – месяц, dd - день), «дата» - это дата, части которой определяем.
6. IF @Month=12

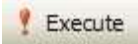

```

      BEGIN
          SET @Month=1
          SET @Year=@Year+1
      END
      ELSE
          BEGIN
              SET @Month=@Month+1
          END
      
```

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

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

Для создания функции, выполним вышеописанный код, как и в случае с предыдущей

функцией, нажав кнопку  Execute. После появления сообщения «Command(s) completed successfully.» закройте окно с кодом.

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

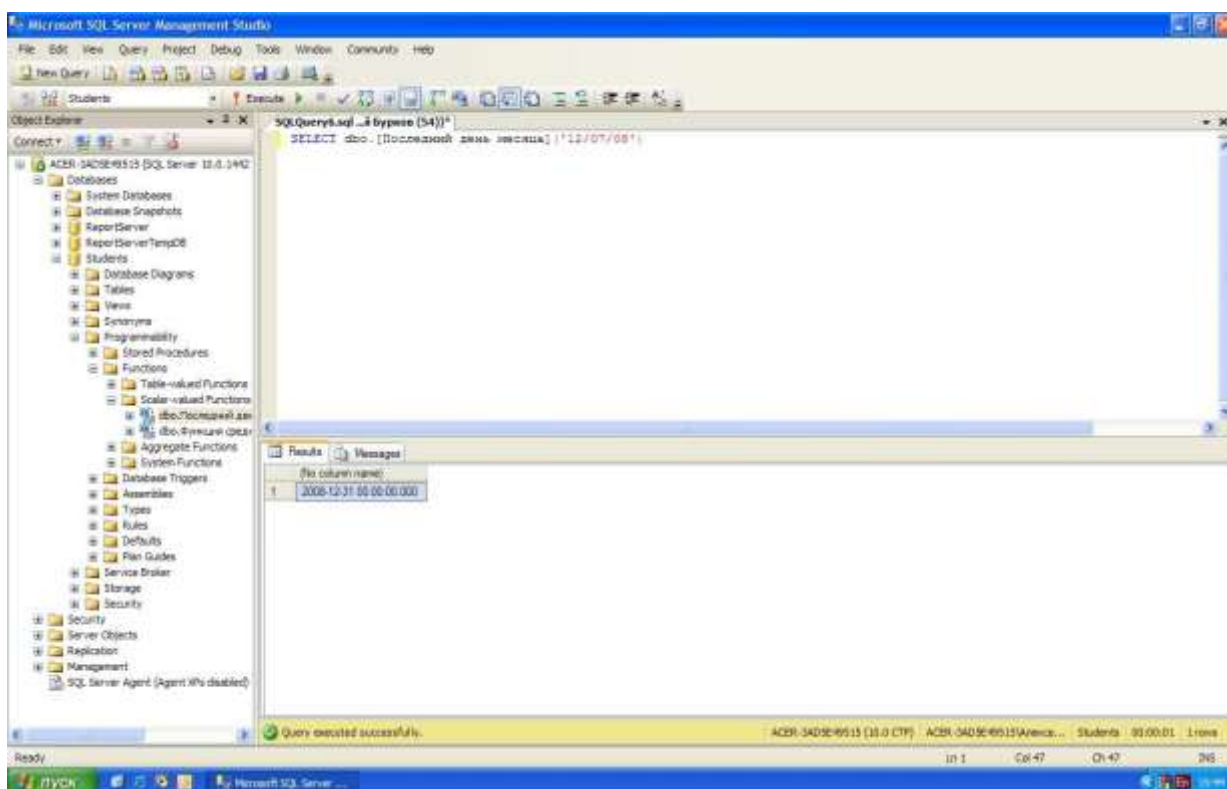


Рис.6.6

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

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

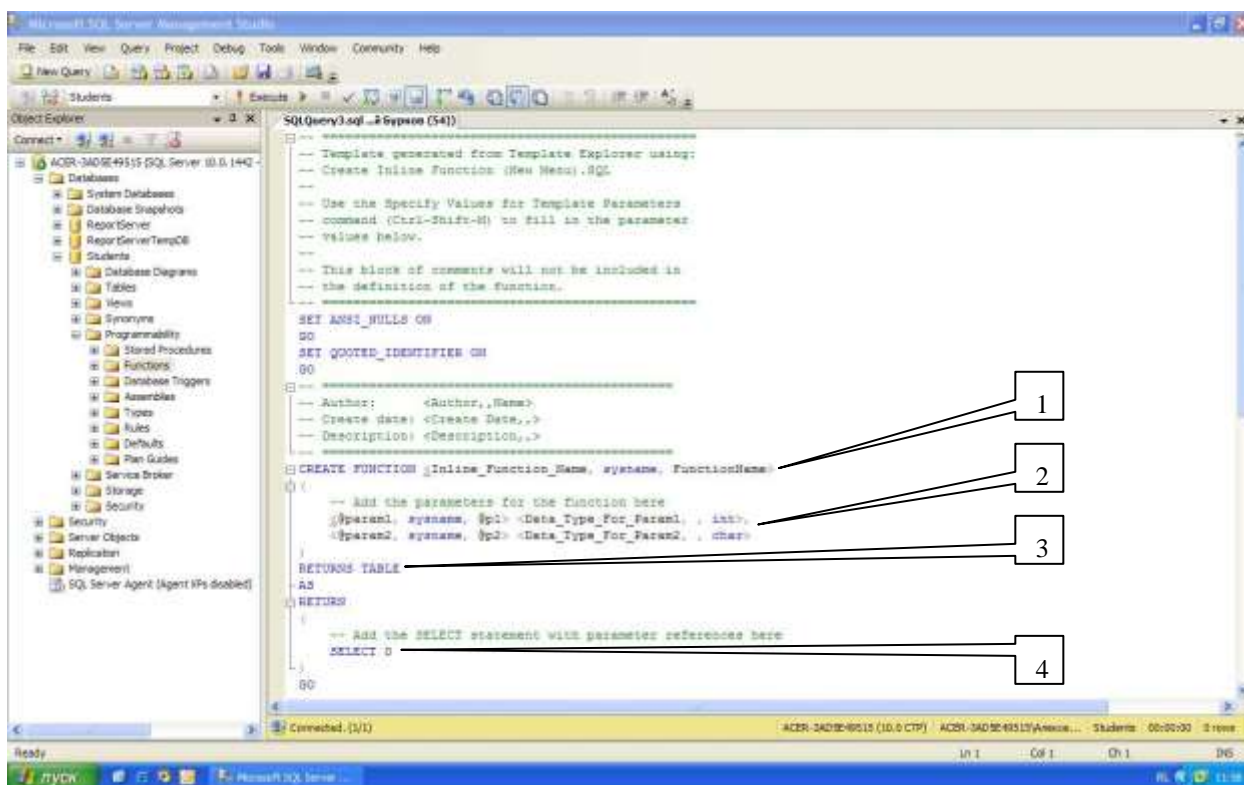


Рис.6.7

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

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

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

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

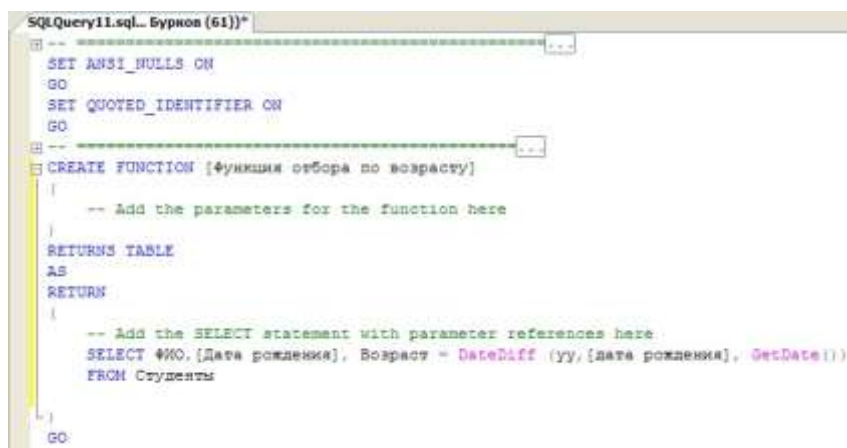


Рис.6.8

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

```

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

```

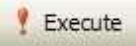

Из вышепредставленной команды видно, что из таблицы «Студенты» отображаются поля «ФИО» и «Дата рождения», а также вычисляемое поле «Возраст». Поле «Возраст» вычисляется при помощи встроенной функции DateDiff вычисляющей различие между датами в определённых единицах измерения (частях даты) и имеющей следующий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

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

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

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

Проверим работоспособность новой табличной пользовательской функции. Создайте новый пустой запрос, затем в окне с пустым запросом наберите команду SELECT * FROM

dbo.[Функция отбора по возрасту]() и нажмите кнопку  на панели инструментов (Рис.6.9).

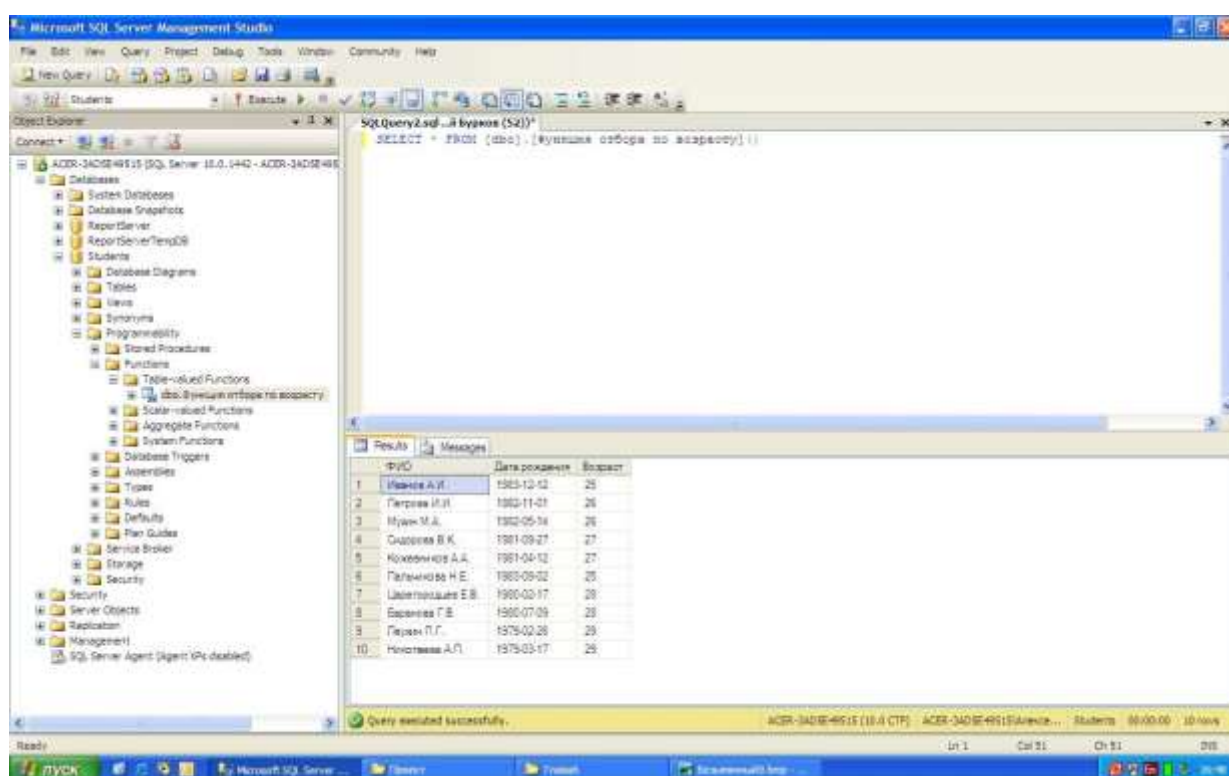


Рис.6.9

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

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

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

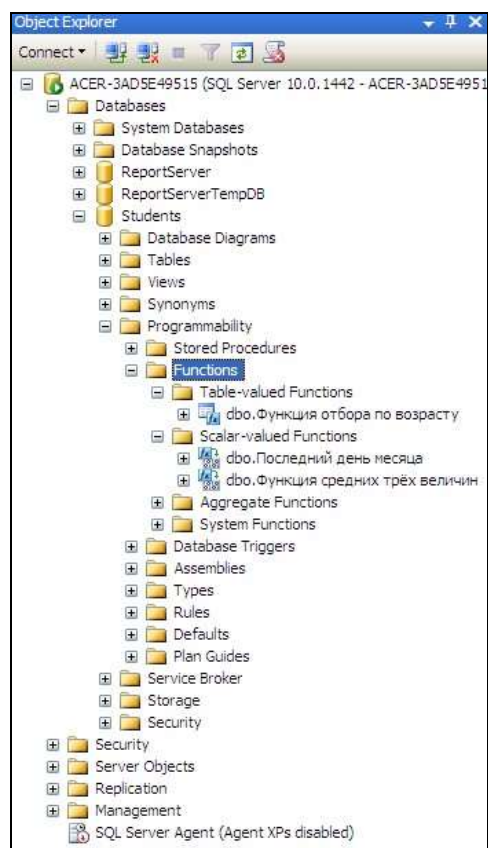


Рис.6.10

Лабораторная работа. «Диаграммы и триггеры»

Цель: научиться создавать диаграммы и триггеры

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

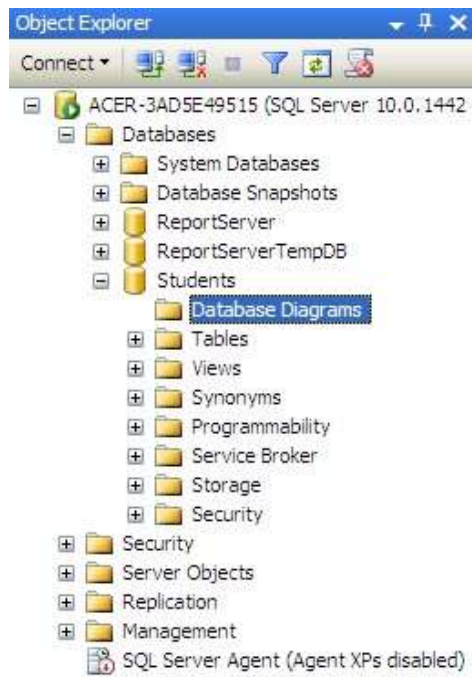


Рис.7.1

Создадим диаграмму, обеспечивающую целостность данных нашей БД «Students». Для создания новой диаграммы в БД «Students» щёлкните ПКМ по папке «Database Diagrams» и в появившемся меню выберем пункт «New Database Diagram». Сначала появится окно с вопросом о добавлении нового объекта «Диаграмма». В этом окне нужно нажать кнопку «Yes».

Затем появится окно «Add Table» предназначенное для добавления таблиц в новую диаграмму (Рис.7.2).

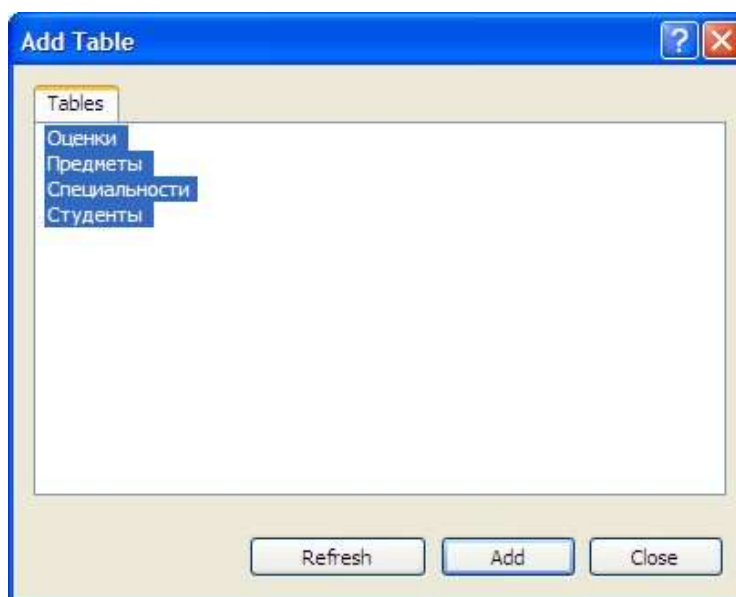


Рис.7.2

В окне добавления таблиц выделите все таблицы нашей БД и нажмите кнопку «Add» (Рис.7.2). Закройте окно «Add Table» нажатием на кнопку «Close».

Появится окно диаграммы, где будут отображены отобранные таблицы. Теперь необходимо определить связи между таблицами. Перетащите поле «Код специальности» из таблицы «Специальности» на такое же поле в таблице «Студенты». Появится окно создания связи между таблицами «Tables and Columns» (Рис.7.3).

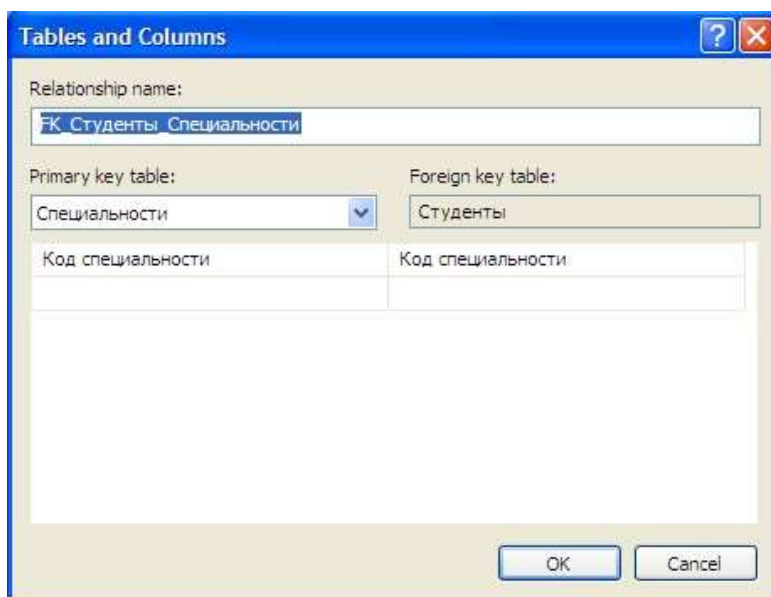


Рис.7.3

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

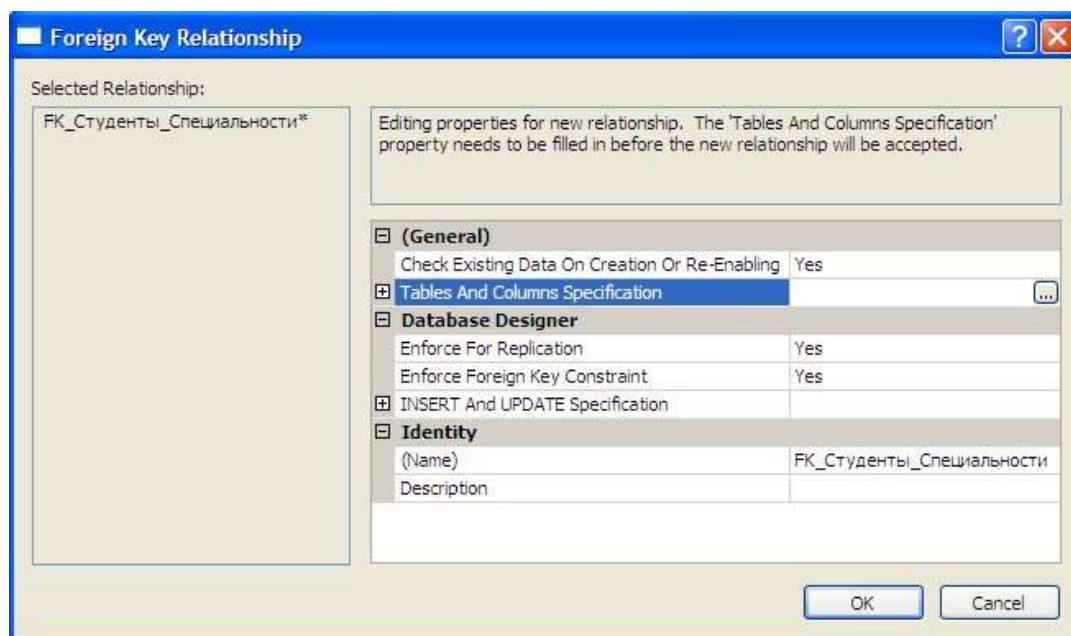


Рис.7.4

Оставьте свойства связи без изменений и в окне свойств связи нажмите кнопку «Ok». В диаграмме между таблицами «Студенты» и «Специальности» появится связь в виде ломанной линии (Рис.7.5).

Аналогичным образом создайте связь таблицы «Студенты» с таблицей «Оценки», перетащив поле «Код студента» из таблицы «Студенты» на одноимённое поле в таблице «Оценки». Затем, свяжите таблицы «Предметы» и «Оценки», перетащив поле «Код предмета» из таблицы «Предметы» на поля «Код предмета 1», «Код предмета 2» и «Код предмета 3» таблицы «Оценки». После выполнения вышеперечисленных действий диаграмма примет следующий вид (Рис.7.5).

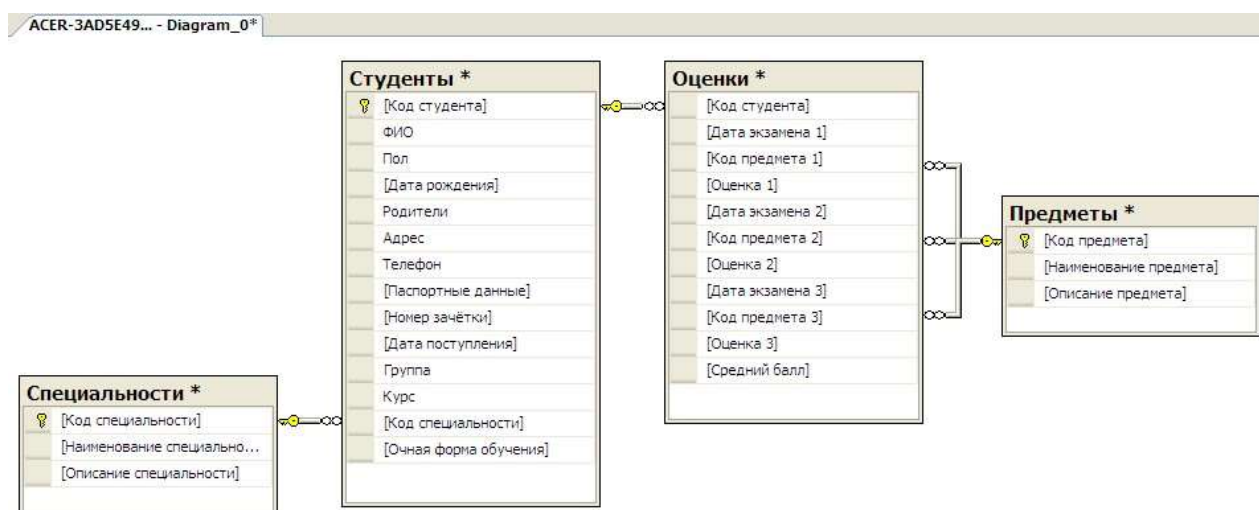


Рис.7.5


Закройте окно с диаграммой, щёлкнув мышью по кнопке закрытия , расположенной в верхнем правом углу окна с диаграммой. Появится окно с вопросом о сохранении новой диаграммы, где необходимо нажать кнопку «Yes» (Рис.7.6).



Рис.7.6

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

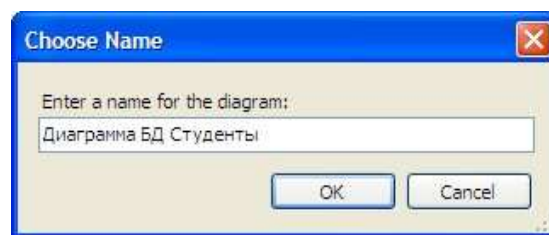


Рис.7.7

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

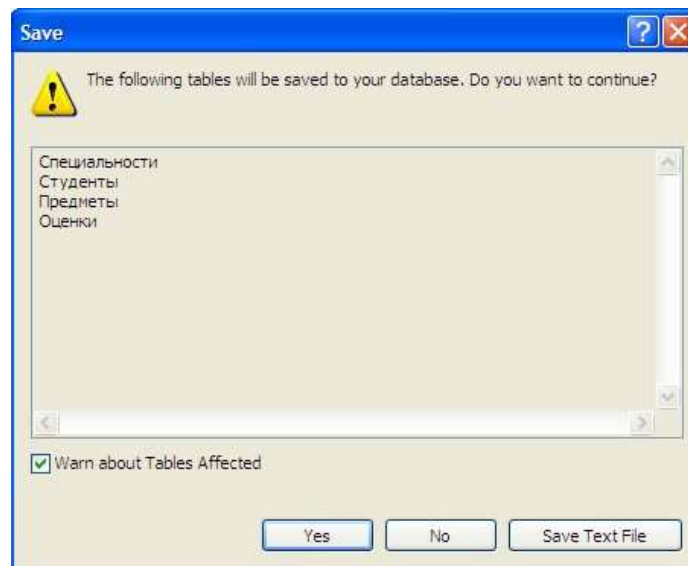


Рис.7.8

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

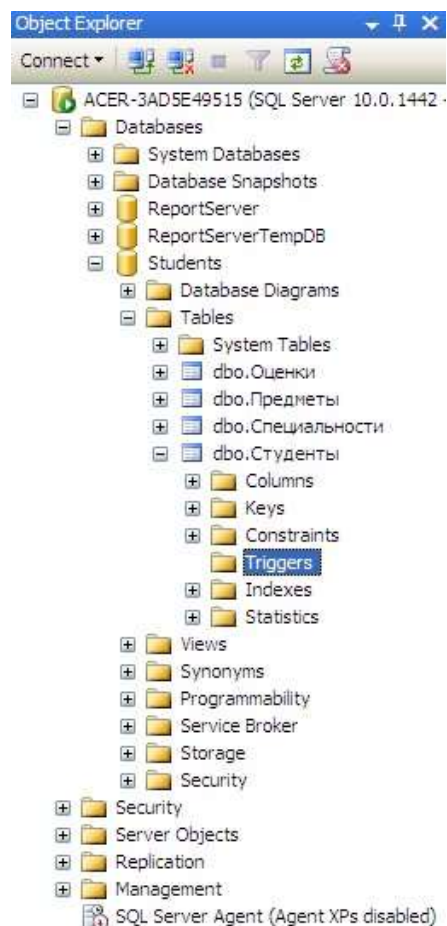


Рис.7.9

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

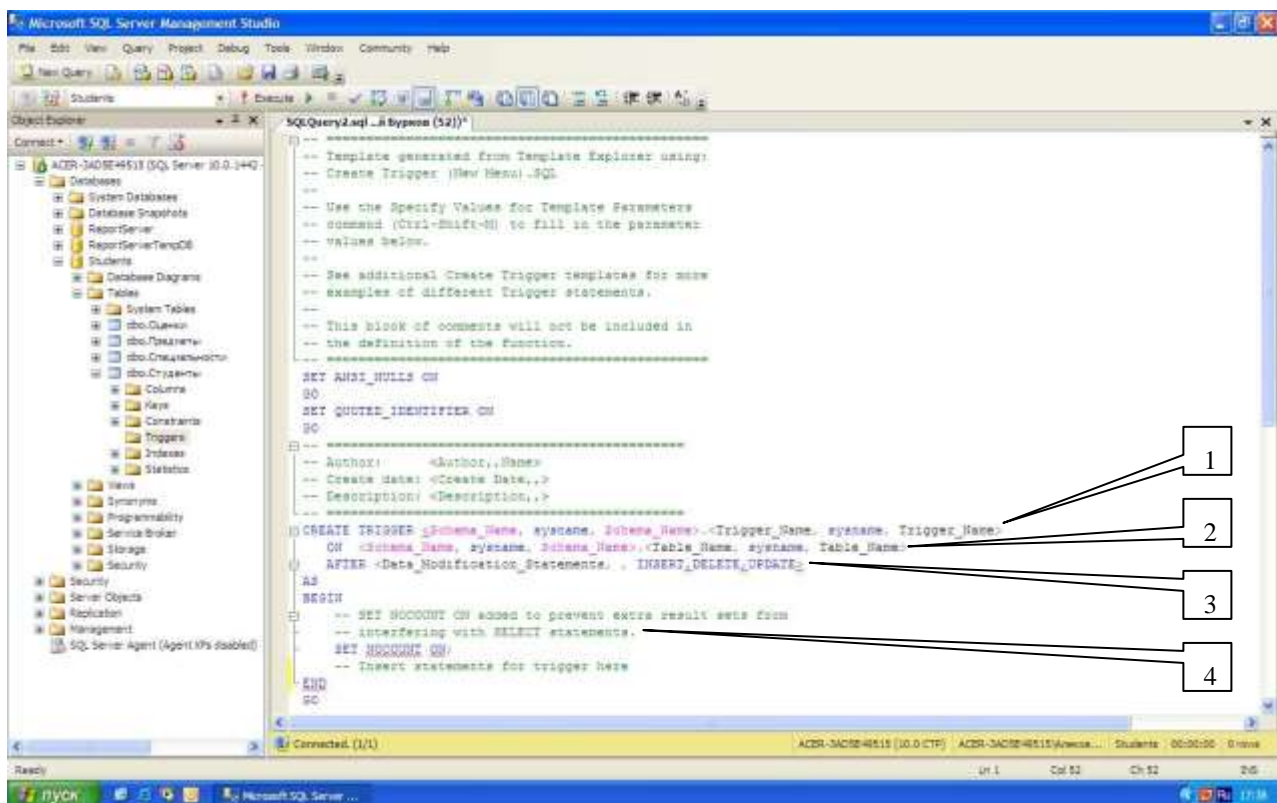


Рис.7.10

Рассмотрим структуру триггеров:

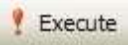
- 1) Область определения имени функции (Trigger_Name);
- 2) Область, показывающая для какой таблицы создаётся триггер (Table_Name);
- 3) Область, показывающая когда выполнять триггер (INSERT – при создании записи в таблице, DELETE – при удалении и UPDATE – при изменении) и как его выполнять (ALTER –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

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

```
SQLQuery6.sql ...и Бурков (52))
--
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
--
CREATE TRIGGER [Индикатор добавления]
ON dbo.Студенты
AFTER INSERT
AS
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from...
SET NOCOUNT ON;
-- Insert statements for trigger here
PRINT 'Запись добавлена'
END
GO
```

Рис.7.11

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

Выполните набранный код, нажав кнопку  на панели инструментов. В нижней части окна с кодом появится сообщение «Command(s) completed successfully.».

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

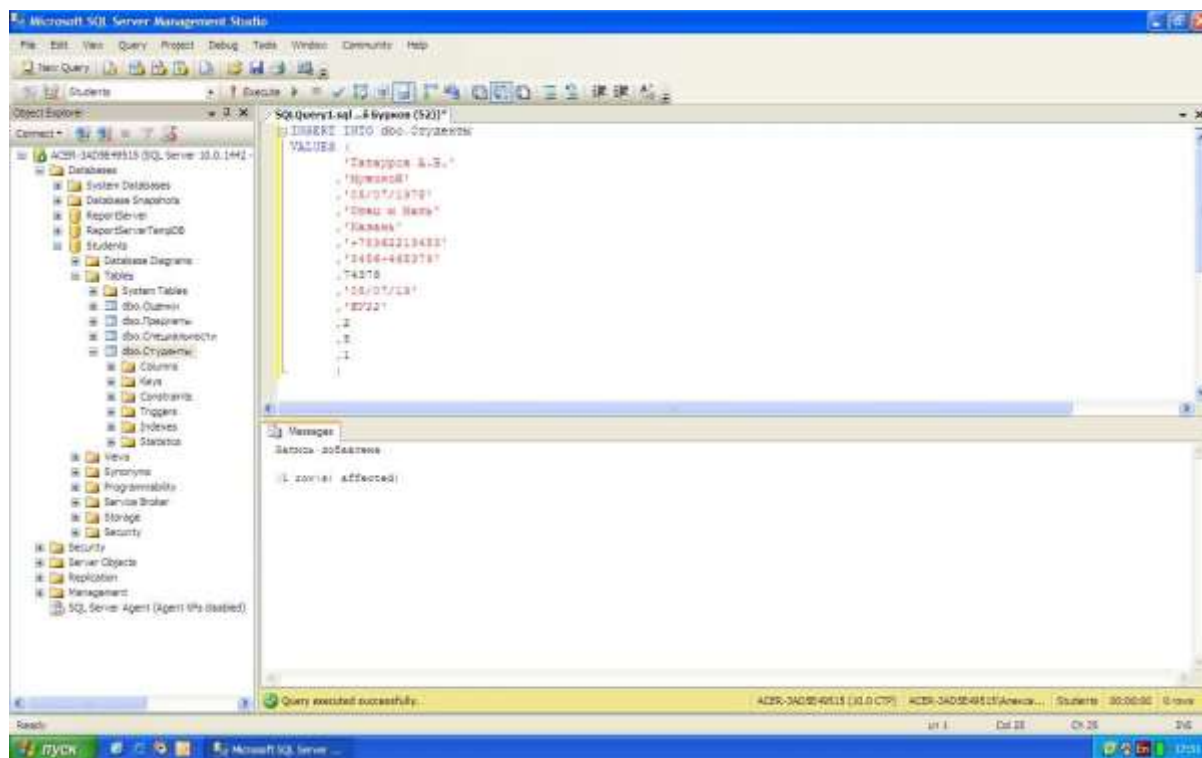
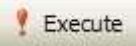
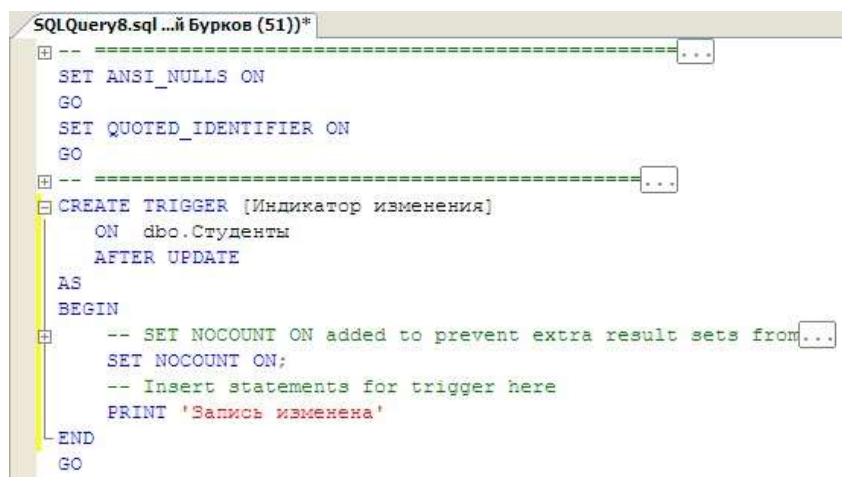


Рис.7.12

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

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



```
SQLQuery8.sql ...й Бурков (51))*
--
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
--
CREATE TRIGGER [Индикатор изменения]
ON dbo.Студенты
AFTER UPDATE
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from...
    SET NOCOUNT ON;
    -- Insert statements for trigger here
    PRINT 'Запись изменена'
END
GO
```

Рис.7.13

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

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

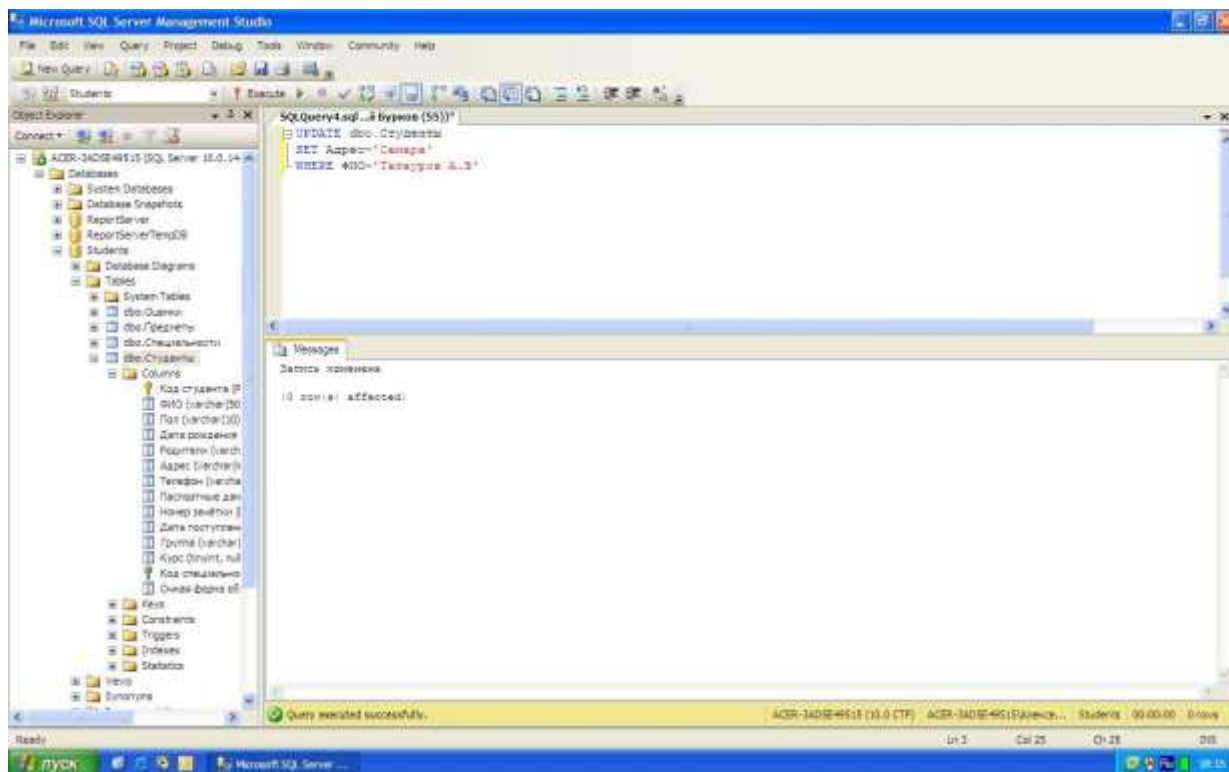
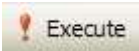
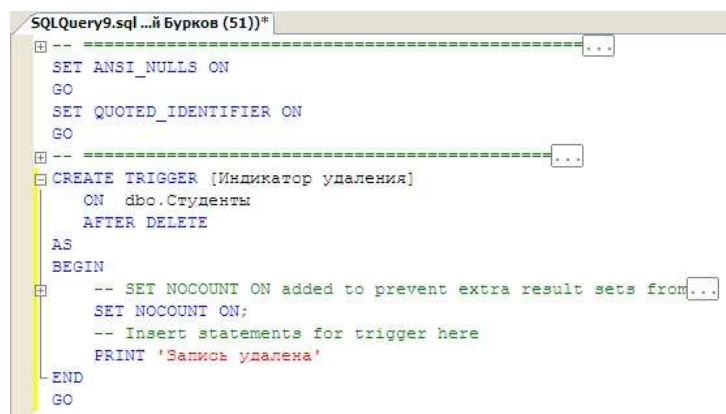


Рис.7.14

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

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



```
SQLQuery9.sql ...й Бурков (51))*
--
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
--
CREATE TRIGGER [Индикатор удаления]
ON dbo.Студенты
AFTER DELETE
AS
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from
SET NOCOUNT ON;
-- Insert statements for trigger here
PRINT 'Запись удалена'
END
GO
```

Рис.7.15

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

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

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

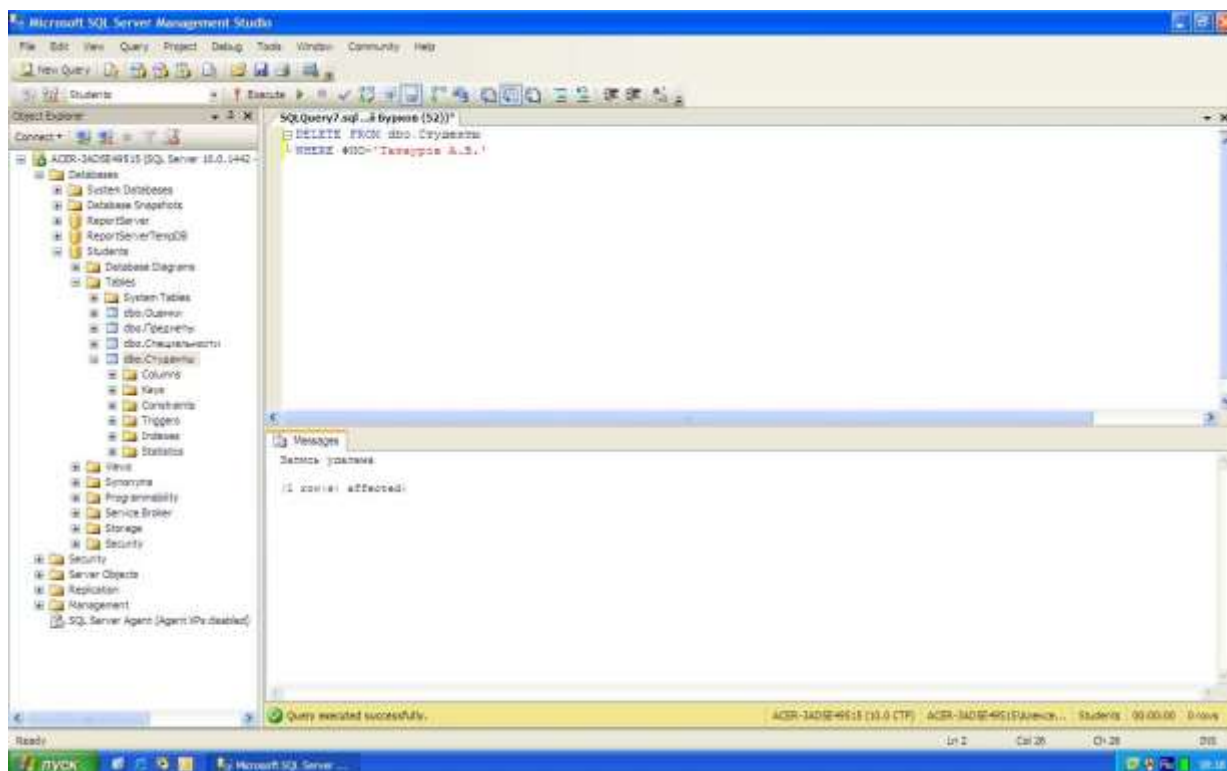
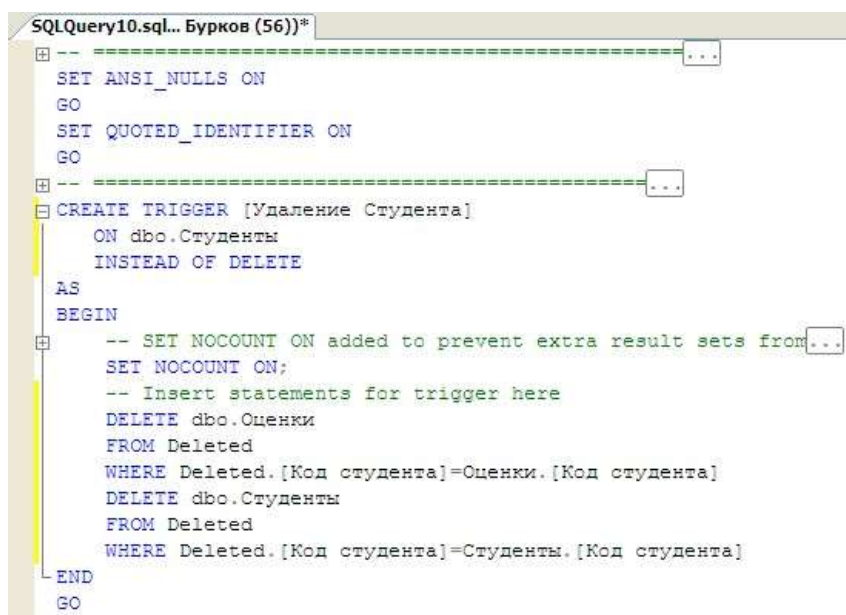


Рис.7.16

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

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

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



```
SQLQuery10.sql... Бурков (56)) *
-- =====
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
CREATE TRIGGER [Удаление Студента]
ON dbo.Студенты
INSTEAD OF DELETE
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from...
    SET NOCOUNT ON;
    -- Insert statements for trigger here
    DELETE dbo.Оценки
    FROM Deleted
    WHERE Deleted.[Код студента]=Оценки.[Код студента]
    DELETE dbo.Студенты
    FROM Deleted
    WHERE Deleted.[Код студента]=Студенты.[Код студента]
END
GO
```

Рис.7.17

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

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

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

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

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

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

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

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

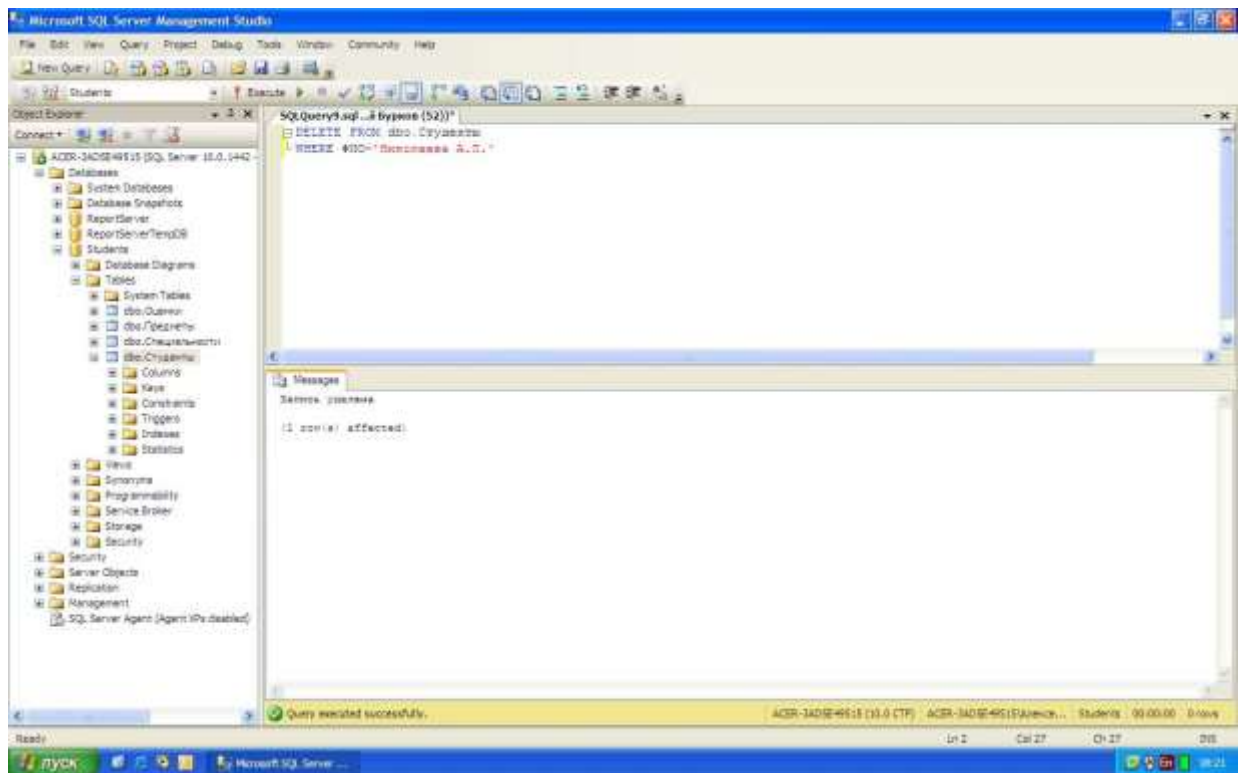
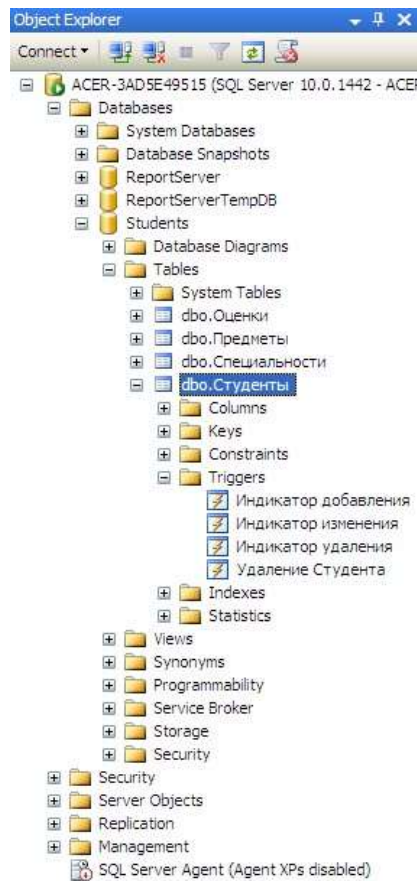


Рис.7.18

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

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

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



Контрольная работа №2 на тему

«Хранимые процедуры и триггеры. Использование основных операторов».

1. Пользовательские функции в Microsoft SQL Server 2008, в отличие от хранимых процедур, ... (Отметьте один правильный вариант ответа.)
 - a) Не возвращают результат
 - b) Могут как возвращать, так и не возвращать результат
 - c) **Возвращают результат**
2. Обработчики событий, которые выполняют команды SQL, если происходят какие либо действия с таблицей: (Отметьте один правильный вариант ответа.)
 - a) Диаграммы
 - b) **Триггеры**
 - c) Хранимые процедуры
3. На какие виды делятся пользовательские функции в Microsoft SQL Server 2008? (Ответ считается верным, если отмечены все правильные варианты ответов.)
 - a) Векторные функции
 - b) **Табличные функции**
 - c) **Скалярные функции**
4. При создании новой пользовательской функции после служебного слова return в табличных функциях ставится: (Отметьте один правильный вариант ответа.)
 - a) тип данных результата, который возвращает функция
 - b) sql команда функции
 - c) **служебное слово table**
5. При создании новой пользовательской функции после служебного слова return в скалярных функциях ставится: (Отметьте один правильный вариант ответа.)
 - a) **служебное слово table**

- b) sql команда функции
 - c) тип данных результата, который возвращает функция
6. Если при создании новой пользовательской функции используется одна SQL команда после служебного слова return, то в этом случае: (Отметьте один правильный вариант ответа.)
- a) **служебные слова begin и end не используются**
 - b) служебное слово begin используется, а служебное слово end не используется
 - c) служебные слова begin и end используются
7. Какой компонент базы данных в Microsoft SQL Server 2008 используется для обеспечения целостности данных? (Отметьте один правильный вариант ответа.)
- a) пользовательские функции
 - b) хранимые процедуры
 - c) **диаграммы**
8. Организовать автоматическое удаление записей из вторичной таблицы при удалении связанной с ними записи из первичной таблицы можно при помощи: (Отметьте один правильный вариант ответа.)
- a) **триггеров**
 - b) хранимых процедур
 - c) диаграмм
9. Какие в Microsoft SQL Server 2008 существуют виды триггеров: (Ответ считается верным, если отмечены все правильные варианты ответов.)
- a) **триггеры, выполняемые после события, произошедшего с таблицей**
 - b) триггеры, выполняемые до события, произошедшего с таблицей
 - c) **триггеры, выполняемые вместо события, происходящего с таблицей**
10. Команда create trigger в Microsoft SQL Server 2008 используется для: (Отметьте один правильный вариант ответа.)
- a) активации триггера
 - b) **создания триггера**
 - c) запуска триггера
11. Какой параметр необходимо установить при создании триггера, для того, чтобы триггер выполнялся вместо события, происходящего с таблицей? (Отметьте один правильный вариант ответа.)
- a) **instead of**
 - b) alter
 - c) with encryption
12. Триггеры создаются для: (Отметьте один правильный вариант ответа.)
- a) **конкретной таблицы базы данных**
 - b) всех таблиц базы данных
 - c) не менее, чем двух таблиц базы данных
13. Компоненты баз данных, которые блокируют удаление записей из первичных таблиц, если существуют связанные с ними записи во вторичных таблицах: (Отметьте один правильный вариант ответа.)
- a) Диаграммы
 - b) **Триггеры**
 - c) Хранимые процедуры
14. Табличные пользовательские функции - это ... (Отметьте один правильный вариант ответа.)

- a) **функции, которые выводят результат в виде таблицы**
 - b) функции, которые не возвращают результат
 - c) функции, которые возвращают число или текст
15. Для создания новой пользовательской функции используется команда: (Отметьте один правильный вариант ответа.)
- a) **create function**
 - b) update
 - c) select
16. Для обеспечения целостности данных при работе базы данных необходимо, чтобы: (Отметьте один правильный вариант ответа.)
- a) **при удалении записей из первичных таблиц автоматически удалялись связанные с ними записи из вторичных таблиц**
 - b) записи из вторичных таблиц не удалялись
 - c) записи из первичных таблиц не удалялись
17. Диаграммы - это ... (Отметьте один правильный вариант ответа.)
- a) пользовательские функции, которые возвращают число или текст
 - b) обработчики событий, которые выполняют команды SQL, если происходят какие либо действия с таблицей
 - c) **компоненты баз данных, которые блокируют удаление записей из первичных таблиц, если существуют связанные с ними записи во вторичных таблицах**
18. Для обеспечения целостности данных используются: (Отметьте один правильный вариант ответа.)
- a) **триггеры, выполняемые вместо события, происходящего с таблицей**
 - b) все виды триггеров
 - c) триггеры, выполняемые после события, произошедшего с таблицей
19. Какой параметр необходимо установить при создании триггера, для того, чтобы триггер выполнялся после события, происходящего с таблицей? (Отметьте один правильный вариант ответа.)
- a) **with encryption**
 - b) **after**
 - c) **instead of**
20. Если при создании триггера используется параметр instead of, то: (Отметьте один правильный вариант ответа.)
- a) **триггер выполняется вместо события**
 - b) триггер выполняется после события
 - c) триггер не выполняется
21. Какой параметр необходимо установить при создании триггера, для того, чтобы произошло включение шифрования данных при выполнении триггера? (Отметьте один правильный вариант ответа.)
- a) **with encryption**
 - b) alter
 - c) instead of
22. Скалярные пользовательские функции - это ... (Отметьте один правильный вариант ответа.)
- a) **функции, которые возвращают число или текст**
 - b) функции, которые не возвращают результат
 - c) функции, которые выводят результат в виде таблицы

Лабораторная работа «Администрирование SQL-сервера. Назначение и ликвидация прав. Создание группы управления правами роли. Формирование списка прав, связанных с ролью. Формирование прав пользователей на основе ролей. Связывание пользователей с ролями»

Цель: приобретение навыков регистрации удаленных серверов с помощью утилиты Enterprise Manager, мастера Register Server Wizard, а также команд Transact-SQL и системной хранимой процедуры sp-addserver. Приобретение навыков управления основной службой MSSQLServer и вспомогательными службами сервера MS SQL Server: задание режима автоматического запуска службы, ручной запуск службы, запуск сервера в однопользовательском режиме, с минимальными требованиями и нестандартной конфигурации, приостановка службы и остановка служб и сервера. Ознакомление с основными концепциями и технологиями, лежащими в основе функционирования сервера, и реализующими и их компонентами: средствами администрирования, сетевыми библиотеками, службами, интерфейсами для создания клиентских приложений

Методические рекомендации для выполнения лабораторной работы

Регистрация удаленных серверов

Перед использованием локального или удаленного сервера в среде Enterprise Manager его необходимо зарегистрировать.

При запуске Enterprise Manager первый раз регистрация локального сервера происходит автоматически. После установки дополнительных копий они также регистрируются в Enterprise Manager автоматически. И только при работе с удаленными серверами их необходимо регистрировать, используя среду Enterprise Manager для выполнения команд меню, запуска Register Server Wizard или интерпретации команд Transact-SQL и системной хранимой процедуры sp-addserver.

При регистрации сервера необходимо указать следующую информацию:

1. Имя сервера.
2. Тип безопасности, используемый для входа на сервер.
3. Имя учетной записи и пароль, используемые для входа на сервер.
4. Имя группы в иерархии групп, в которой необходимо зарегистрировать сервер.

Утилита Enterprise Manager представляет собой всего лишь графический интерфейс для выполнения системных хранимых процедур SQL Server 2000. Таким образом, она является клиентским средством, устанавливающим соединение с SQL Server и выполняющим те или иные процедуры. Поэтому, прежде чем эта программа сможет выполнить какие-либо операции с сервером, она должна получить соответствующие права доступа, т.е. пройти аутентификацию.

Запуск, остановка и приостановка служб сервера

До выполнения каких-либо работ по администрированию сервера MS SQL Server или баз данных, а также манипулированию данными необходимо запустить сервер. Точнее говоря, запустить его основную службу MSSQLServer. Только после запуска этой службы и проверки прав доступа пользователя, пользователь сможет выполнять функции, определенные его правами и разрешениями. Остальные службы являются вспомогательными, и их работа строится на фундаменте, обеспечиваемом службой MSSQLServer. Например, служба SQLServerAgent запускается лишь тогда, когда требуется автоматическое администрирование и управление системой на базе SQL Server. Служба MSSearch используется для работы с электронными документами, обеспечивает полнотекстовый поиск информации и, как правило, используется автономно. Служба MSDTC позволяет организовать доступ к распределенным источникам информации и управлять распределенными транзакциями.

Дополнительные службы запускаются отдельно и устанавливают соединение с сервером, подобно обычным клиентам. Каждая такая служба самостоятельно подключается к основной службе MSSQLServer, используя определенные учетные записи с соответствующими правами доступа.

Для сетевого варианта установки управлять службами можно как локально, так и удаленно даже средствами операционной системы. Для операционной системы Windows 98 можно запустить только один экземпляр сервера в качестве приложения, так как в Windows 98 нет служб, и управлять этим приложением локально. Запускать, останавливать и приостанавливать сервер можно также при отсутствии сети.

SQL Server имеет множество инструментов для импорта и экспорта данных. Лучшим является служба преобразования данных Data Transformation Services (DTS), которая предоставляет набор инструментальных средств. Она также позволяет извлекать, преобразовывать и объединять данные из источников данных разной природы, расположенных как в одном, так и в разных местах. Можно управлять данными, используя инструментальные средства DTS, для графического построения пакетов DTS или создавая объектно-ориентированные пакеты DTS.

Пакет DTS – это объект, в котором хранится описание выполняемых в ходе импорта, экспорта и трансформации данных действий. Каждый пакет реализует один или большее количество шагов, которые выполняются последовательно или параллельно. С помощью пакета может выполняться копирование и преобразование данных и объектов баз данных. Пакеты можно редактировать, защищать паролем, конфигурировать для автоматического выполнения по расписанию.

Задания для самостоятельного выполнения

Задание 1. Произвести регистрацию удаленного сервера с помощью окна параметров регистрации сервера Register SQL Server Properties утилиты Enterprise Manager, выполнив действия:

1. На дереве объектов консоли выбрать одну из групп серверов, где будет зарегистрирован удаленный сервер.
2. Открыть контекстное меню группы серверов и выполнить команду New SQL Server Registration.
3. В открывшемся окне Register SQL Server Properties задать следующие параметры:
 - а) Имя удаленного сервера в виде следующей записи: сетевое имя NetBios соответствующего компьютера, косая черта «\», имя копии сервера (для сервера по умолчанию это имя копии можно не задавать);
 - б) Учетную запись, которая будет использоваться для установления соединения с соответствующим сервером: либо учетная запись домена Windows NT и ее набор прав в SQL Server, либо учетная запись сервера, созданная на регистрируемом сервере и включающая входное имя пользователя Login Name и его пароль Password, с указанием режима подключения с вводом пароля при каждом соединении или без ввода пароля;
 - в) Имя группы серверов из числа имеющихся или имя новой группы, которую можно создать, щелкнув по кнопке с многоточием в том же окне в области Options;
 - д) Установить, если необходимо, следующие переключатели: Display SQL Server state in console – показывать состояние сервера в окне объектов Enterprise Manager; Automatically start SQL Server when connecting – автоматически запускать сервер при соединении; Show system database and system table – отображать системные базы данных и таблиц.

Задание 2. Произвести регистрацию удаленного сервера с помощью мастера Register Server Wizard, выполнив следующие действия:

1. Щелкнуть на кнопке Run a Wizard панели инструментов Enterprise Manager.
2. В открывшемся окне, вид которого зависит от левого окна Enterprise Manager (выбран или не выбран сервер или папка группы серверов), выбрать мастер регистрации сервера Register Server Wizard.
3. Щелкнуть по кнопке ОК.
4. В первом окне мастера предлагается следующий порядок работы:
 - а) выбрать SQL сервер;
 - б) выбрать режим аутентификации;
 - в) определить группу SQL серверов.
5. В этом же окне сбросить флажок From now on, I want to perform this task without using a wizard (теперь я хотел бы выполнить задачу без использования мастера), иначе мастер

закончит свою работу, открыв окно Register SQL Server Properties для ручной регистрации сервера

6. Щелкнуть по кнопке Next.
7. Во втором окне выбрать или ввести имя регистрируемого сервера в левой части окна.
8. Щелкнуть по кнопке добавить Add. В случае ошибки использовать кнопку удалить Remove. Если регистрируется сразу несколько серверов, то они будут включены в одну и ту же группу с одинаковыми параметрами и с одной и той же учетной записью для установления соединения.
9. Щелкнуть по кнопке Next.
10. В третьем окне необходимо выбрать режим аутентификации при подключении к регистрируемому серверу.
11. Если использовать аутентификацию Windows NT, то возможность подключения к серверу будет зависеть от учетной записи, под которой работает SQL Server.
12. Если выбрать аутентификацию SQL Server, то для установления соединения потребуется имя и пароль учетной записи, предварительно созданной на регистрируемом сервере SQL Server. В этом случае открывается окно, в котором надо сделать выбор режим подключения к регистрируемому серверу:
 - a) с использованием сохраняемой одной и той же учетной записи, для которой
 - b) надо в этом же окне ввести имя и пароль;
 - c) с использованием учетной записи, имя и пароль который надо вводить каждый
 - d) раз при подключении к удаленному серверу.
13. Щелкнуть по кнопке Next и перейти к следующему окну мастера.
14. Выбрать или создать новую группу, в которую включить регистрируемый сервер.
15. Щелкнуть по кнопке Next и перейти в последнее окно мастера со списком регистрируемых серверов.
16. Щелкнуть по кнопке Finish.
17. Если регистрируемый сервер найден, то произойдет подключение к нему.
18. Если регистрируемый сервер не найден. То Enterprise Manager выдаст запрос на повторную регистрацию сервера.

19. Возможные ошибки при регистрации:

- a) регистрируемый сервер был остановлен;
- b) на компьютере, с которого выполняется регистрация используются иные сетевые библиотеки и протоколы, чем на регистрируемом сервере;
- c) если сервер зарегистрирован с использованием аутентификации Windows NT, а для пользователей не создано соответствующей учетной записи на SQL Server (Login failed);
- d) если используется аутентификация SQL Server и имя пароль заданы неверно.

Задание 3. Произвести регистрацию удаленного сервера, выполняя команду:

```
EXEC sp.addserver @server = 'server',
```

```
@local = 'local',
```

```
@duplicate.ok = 'duplicate.ok'
```

Здесь параметры имеют следующее назначение:

Server – имя регистрируемого сервера, которое состоит из сетевого имени NetBios соответствующего компьютера и имени копии сервера, между которыми ставиться разделитель ”\”, при этом для копии сервера по умолчанию второе имя задавать не требуется;

Local – сервер для регистрации является локальным, иначе – сервер удаленный;

duplicate.ok – разрешает дублирование имен серверов, что приводит к тому, что информация о новом сервере будет записана поверх старой информации.

Задание 4. Произвести настройку конфигурации утилиты Enterprise Manager, выполнив действия:

1. исполнить команду Tools/Options утилиты;

2. в открывшемся окне SQL Server Enterprise Manager Properties выбрать вкладку General.
3. в группе Server state pooling (опрос состояния сервера) выбрать службу Service и задать количество секунд, через которое будет проводиться опрос состояния соответствующей службы, и отображаться это состояние в виде соответствующего значка.
4. для конфигурирования одного из серверов в качестве центрального хранилища информации необходимо снять флажок Read/Store User Independent (независимое считывание/хранение пользователей), а на локальном сервере установить переключатель Read from remote (считывать с удаленного сервера) и указать имя удаленного сервера, с которого будет считываться информация о конфигурации.
5. убедиться, что установленный флажок Read/Store User Independent означает коллективное использование информации о конфигурации, а сброшенный – личное, когда информация для каждого пользователя сохраняется отдельно.

Задание 5. Установить режим автоматического запуска служб SQL Server 2000, который производится автоматически операционной системой при ее запуске, выполнив следующие действия:

1. При установке сервера MS SQL Server 2000 задать режим автоматического запуска служб сервера. В этом случае сразу же после установки и каждый раз при запуске операционной системы все установленные на компьютере службы сервера будут запускаться автоматически.
2. Если режим автоматического запуска не был задан при установке или по каким-либо причинам был отключен в дальнейшем, то его можно задать следующими действиями (три варианта):

2.1 Войти в Enterprise Manager и выполнить команды:

- a) в его левом окне выбрать требуемый сервер, так как для каждого экземпляра, или копии сервера имеются отдельные экземпляры, или копии служб MSSQLServer, SQLServerAgent и MSDTC;
- b) щелкнуть правой клавишей мыши, чтобы открылось контекстное меню сервера;
- c) щелкнуть левой клавишей по элементу Properties (свойства);
- d) в открывшемся окне SQL Server Properties (свойства SQL Server) на вкладке General (общие) установить флажок для требуемых служб:
Autostart SQL Server;
Autostart SQL Server Agent;
Autostart MSDTC;
- e) щелкнуть по кнопке ОК;
- f) перезагрузить операционную систему и убедиться, что нужные службы запущены.

2.2 Войти в • утилиту Services (Службы) операционной системы Windows NT или Windows, исполнив команду Пуск/Настройка/Панель управления/Службы (Start/.../Control panel/Services) и выполнить команды:

- a) в открывшемся окне служб Services выбрать требуемую службу;
- b) дважды щелкнуть по выбранной службе;
- c) в открывшемся окне свойств выбранной службы Properties на вкладке General (общие) раскрыть список Start type (тип запуска);
- d) в списке выбрать режим Automatic и щелкнуть по нему;
- e) щелкнуть по кнопке ОК;
- f) закрыть все окна операционной системы;
- g) перезагрузить операционную систему и убедиться, что все нужные службы запущены.

2.3 Войти в утилиту SQL Server Services Manager и в открывшемся окне с таким же названием выполнить команды:

- a) раскрыть список Server (сервер);
- b) щелкнуть по требуемому серверу;

- с) раскрыть список Services (службы) для этого сервера;
- д) щелкнуть по рассматриваемой службе;
- е) в открывшемся окне установить флажок Autostart service when OS start (автоматический старт при запуске операционной системы);
- ф) закрыть окна утилиты Services Manager;
- г) перезагрузить операционную систему и убедиться, что все нужные службы запущены.

Задание 6. Произвести ручной запуск службы SQL Server 2000 одним из следующих четырёх способов:

1. Войти в Enterprise Manager и выполнить действия:
 - а) выбрать требуемый сервер;
 - б) открыть его контекстное меню;
 - с) щелкнуть по команде Start для запуска службы MSSQLServer;
 - д) для запуска службы SQLServerAgent надо открыть папу Management сервера и щелкнуть по команде Start;
 - е) для запуска служб MSDTC и SQLMail надо открыть папку Support Services и щелкнуть по команде Start для соответствующей службы.
2. Войти в утилиту SQL Server Service Manager, выбрать требуемый сервер и службу и щелкнуть по кнопке Start.
3. В командной строке запустить утилиту командной строки net start, указав в качестве параметра имя требуемой службы или экземпляра сервера:


```
net start mssqlserver
net start sqlserveragent
net start MSSQL$TRELON net
start SQLAgent$TRELON
net start
```

 для выдачи списка запущенных в ОС служб
4. Установить режим работы операционной системы сеанс DOS и в командной строке исполнить команду sqlserver для запуска сервера, как отдельного приложения операционной системы. В этом случае все средства администрирования система SQL Server такие, как Service Manager, Enterprise Manager, Service (для панели управления) будут показывать, что сервер остановлен, и все системные сообщения будут появляться в консольном окне, в котором выполнена команда sqlserver. Сервер будет запущен под учетной записью пользователя, и если необходимо завершить сеанс работы ОС, то сначала надо завершить работу SQL Server.

Задание 7. Запустить SQL Server в однопользовательском режиме, выполнив действия:

1. Убедиться, что все службы рассматриваемого сервера остановлены.
2. В командной строке исполнить команду: sqlserver.exe -m.
3. Приступить к конфигурированию характеристик сервера или восстановлению поврежденной системной базы, учитывая, что:
 - а) модифицированные страницы сразу записываются на диск, а не остаются, как обычно в кэш-памяти;
 - б) разрешен прямой доступ к системным таблицам с помощью команд INSERT< DELETE и UPDATE.

Задание 8. Произвести аварийный запуск сервера с минимальными требованиями для проведения восстановительных работ из-за неправильного конфигурирования:

1. Для запуска SQL Server 2000 как службы с минимальными требованиями исполнить команду в командной строке: sqlserver.exe -f.
2. Для запуска SQL Server 2000 как приложения с минимальными требованиями исполнить команду в командной строке: sqlserver.exe -f -c.
3. Для первого случая убедиться, что:
 - а) количество открытых баз данных, таблиц, открытых объектов, размер КЭШа процедур минимальны;
 - б) запрещено исполнение хранимых процедур;
 - с) установлен однопользовательский режим;

- d) удаленный доступ запрещен;
- e) разрешен прямой доступ к таблицам.

Задание 9. Приостановите, а затем и остановите работу служб сервера. Запустите их вновь.

Просмотрите параметры запуска в реестре по адресу
HKEY_LOCAL_MACHINE\SOFTWARE\MICROSOFT\MSSQLSERVER
\PARAMETERS.

Задание 10. Осуществить передачу данных с помощью мастера Data Transformation Services(DTS), используя способ Copy table(s) and view(s) from the source database(копировать таблицу(таблицы) и представление(представления) из источника), выполнив следующие действия:

1. Запустить мастер: Пуск \ Программы \ Microsoft SQL Server \ Import and Export Data.
2. В первом открывшемся окне, которое содержит общую информацию о работе мастера, щёлкнуть по кнопке Next.
3. Во втором окне в раскрывающемся списке Source(источник) необходимо выбрать тип источника данных; в списке Server(сервер) выбрать сервера-источника; указать список аутентификации; в списке Database выбрать базу данных, в которую будет осуществляться взаимодействие. После этого щёлкнуть по кнопке Next.
4. Для редактирования, по необходимости, параметров конфигурации щёлкнуть на кнопке Advanced(дополнительно).
5. В третьем окне сконфигурировать получатель: в раскрывающемся списке Database(база данных) выбрать пункт New(создать) и создать новую базу данных.
6. В четвёртом окне DTS Wizard выбрать способ передачи данных Copy table(s) and view(s) from the source database.
7. В пятом окне в столбце Source Table(таблица источник) выбрать одну или более таблиц или представлений для копирования.
8. Для того, чтобы увидеть содержание исходной таблицы, щёлкните на кнопке Preview(просмотр).
9. В столбце Destination(получатель) указать имя таблицы-получателя.
10. Если необходимо выполнить преобразование данных, то в столбце Transform(преобразование) для соответствующей таблицы щёлкните на кнопке с многоточием. В открывшемся окне можно настроить процесс трансформации не только самих данных, но и их типов.
11. Следующее окно мастера DTS Wizard (рис. 24.22) будет общим для всех способов переноса. В этом окне для созданного пакета DTS указать способ его сохранения.
12. Если выбрали вариант SQL Server, то необходимо установить параметры:
 - a) в поле Name(имя) указывается имя, под которым пакет DTS будет сохранен в системной базе данных msdb;
 - b) в поле Description (описание) можно ввести описание объекта в произвольной форме;
 - c) в поле Owner Password (пароль владельца), чтобы скрыть информацию, указанную при создании пакета, от просмотра неавторизованными пользователями, можно установить пароль владельца;
 - d) установив в поле User Password (пароль пользователя) пароль пользователя, можно запретить выполнение пакета пользователями, которые не имеют на это права. Только те пользователи, которые знают пароль, смогут выполнить пакет DTS.
 - e) в списке Server name (имя сервера) выбирается имя сервера, на котором будет сохранен пакет DTS.
13. При выборе режима хранения SQL Server Meta Data Services (службы метаданных SQL Server) мастер выведет окно, во многом напоминающее окно при режиме хранения SQL Server. Добавлена лишь кнопка Scanning (сканирование), с помощью которой можно установить взаимосвязи между объектами в источнике и получателе данных, сохраняемые в хранилище(первичный и внешний ключи, индексы, столбцы, типы данных и т. д.).

14. При выборе двух оставшихся режимов хранения мастер выведет окно, в котором помимо имени, описания, пароля пользователя и пароля владельца необходимо указать только имя файла, в который будет записан пакет.

На этом работа с мастером DTS Wizard по созданию пакетов для импорта экспорта данных заканчивается. В последнем окне (рис. 24.28) приведена сводная информация о созданном пакете. После щелчка на кнопке Finish(готово) будет создан сам пакет.

15. Если при создании пакета было задано его незамедлительное выполнение, то мастер откроет окно Executing Package (выполнение пакета), позволяющее следить за процессом выполнения пакета.

Задание 2. Осуществить передачу данных с помощью мастера Data Transformation Services(DTS), используя способ Use a query to specify the data to transfer(использовать запрос для выборки данных), выполнив следующие действия:

1. Выполнить с первого по пятый пункты задания 1.
2. В четвёртом окне мастера DTS Wizard установить переключатель Use a query to specify the data to transfer.
3. В открывшемся окне ввести SQL-код запроса; если имеется готовый код, сохранённый на диске, его можно подключить, воспользовавшись кнопкой

Browse(обзор).

4. Если необходимо написать сложный запрос с перечислением множества таблиц и столбцов и при этом гарантировать, что указаны правильные имена объектов, можно воспользоваться встроенным в мастер конструктором запросов. Для вызова конструктора запроса щёлкнуть на кнопке Query Builder (конструктор запросов).

Откроется окно, в котором нужно выбрать, какие столбцы, из каких таблиц будут включены в запрос.

5. Щёлкнуть по кнопке Next.
6. В открывшемся окне, перенося имена столбцов из левой части окна в правую, задать порядок сортировки, которая ведётся по столбцам, указанным в самом верху списка.
7. Щёлкнуть по кнопке Next.
8. В следующем окне указать критерии для выборки данных: установить указатель Only Rows meeting criteria(только строки, соответствующие критерию).
9. Если необходимости в фильтрации нет, установите переключатель All rows(все строки). Нажмите на кнопку ОК.
10. После того как редактирование запроса закончено, мастер откроет окно, в котором можно настроить трансформацию данных. Работа с этим окном практически ничем не отличается от работы по настройке трансформации данных при копировании между таблицами, описанной в задании 1.
11. После настройки трансформации данных необходимо сохранить пакет DTS одним из способов, которые указаны в предыдущем задании.

Задание 3. Осуществить передачу данных с помощью мастера Data Transformation Services(DTS), используя способ Copy objects and data between SQL Server databases(копировать объекты и данные между базами данных SQL Server), выполнив следующие действия:

1. Выполнить с первого по пятый пункты задания 1.
2. В четвёртом окне мастера DTS Wizard установить переключатель Copy objects and data between SQL Server databases.
3. Щёлкнуть на кнопке Next.
4. В открывшемся окне указать, какие объекты и данные будут копироваться:
 - установка флажка Create destination objects – создание переносимых объектов;
 - установка флажка Drop destination objects first – удаление всех одноимённых объектов из конечной базы данных;
 - установка флажка Include all dependent objects – включение всех зависимых объектов;

- установка флажка Copy data – копирование только структуры объектов.

5. Установив флажок Copy all objects, выполняется копирование всех объектов.
6. Если необходимо скопировать только часть объектов, сбросьте флажок Copy all objects и выберите нужные объекты, щёлкнув на кнопке Select Objects(выбор объектов).
7. Чтобы выбрать только некоторые их них, в окне мастера сбросьте флажок Use default options(использовать параметры по умолчанию). После щелчка на кнопке Options(параметры) в открывшемся окне укажите объекты, которые необходимо скопировать.
8. После указания объектов необходимо сохранить пакет DTS одним из способов, которые указаны в первом задании. После будет создан сам пакет.

Контрольные вопросы:

1. На каких критериях следует основываться при выборе метода импорта или экспорта данных.
2. Что собой представляет служба преобразования данных Data Transformation Serviced(DTS).
3. Какие способы передачи данных можно выделить, используя мастер Data Transformation Serviced(DTS) Import and Export Wizard.
4. Какими способами можно осуществить хранение пакета DTS.
5. Что собой представляет внутренняя структура пакета DTS.

Задания к экзамену

Вариант 1

1 задание. Выберите правильный ответ:

1. Структура данных, для которой характерна подчиненность объектов нижнего уровня объектам верхнего уровня, называется а) табличной;
б) реляционной;
в) иерархической;
г) линейная.
2. Для организации учета выдачи книг в библиотеке создается база данных, состоящая из двух таблиц, между которыми установлены отношения подчиненности: · *таблица карточек читателей*, содержащая информацию о читателе;
· *таблица выдачи книг*, в которую заносится информация о выдаче книги читателю и о возврате книги.
Таблица карточек читателей является ...
а) подчиненной;
б) главной;
в) вторичной;
г) необязательной.
3. Атрибут поля «Обязательное поле» означает:
а) обязательность заполнения поля в каждой записи;
б) обязательность наличия поля в таблице;
в) обязательность отображения поля при открытии таблицы в режиме таблицы;
г) обязательность отображения поля при открытии таблицы в режиме конструктора.
4. На рисунке представлена связь _____ между таблицами «товары» и «прайс-лист»:



- а) один-ко-многим;
- б) многие-ко-многим;
- в) многие-к-одному;
- г) один-к-одному.

5. Перекрестным запросом является:

- а) запрос, в котором один или несколько столбцов является заголовком строк, один столбец – заголовком столбцов и один – столбцом значений;
- б) запрос, в котором строки становятся столбцами, а столбцы – строками;
- в) запрос из нескольких несвязанных таблиц;
- г) запрос, в котором данные одной выборки добавляются данными другой.

6. В поле «Наименование имущества» таблицы MS Access установлен тип данных...

Имущество : таблица			
Номер	Код клиента	Наименование имущест	Дата страхов
7	104	Городская квартира	05.05.2001
5	103	Городская квартира	10.11.2000
8	105	Домовладение	12.07.2001
3	102	Домовладение	20.06.2000
1	101	Домовладение	01.01.2000
6	103	Загородная дача	10.11.2000

- а) текстовый;
- б) счетчик;
- в) поле объекта OLE;
- г) числовой.

7. Стандартное имя документа, созданного в MS Access называется:

- а) Db1;
- б) Книга1;
- в) Документ1;
- г) Презентация1.

8. На рисунке представлен объект базы данных MS Access...

- а) отчеты;
- б) формы;
- в) запросы;
- г) таблицы.

9. СУБД - это:

- а) оболочка операционной системы, позволяющая более комфортно работать с файлами;

- б) набор программ, обеспечивающих работу всех аппаратных устройств компьютера и доступ пользователя к ним;
 - в) программы, которые позволяют создавать БД, хранить их, модифицировать и т.д.;
 - г) программы по обслуживанию БД.
10. Таблицы в базах данных предназначены для...
- а) хранения данных базы;
 - б) отбора и обработки данных базы;
 - в) ввода данных базы и их просмотра;
 - г) автоматического выполнения группы команд.

2 задание.

Необходимо разработать проект базы данных по заданной предметной области и реализовать его средствами системы управления базами данных Microsoft Access. А именно разработать и создать базу данных «Библиотека», учитывая следующие требования:

1. База данных должна содержать 3 таблицы;
2. Все таблицы в проекте должны быть связаны между собой (рекомендуется использовать связь 1: М);
3. База данных должна быть нормализована, то есть приведена к 3-ей нормальной форме;
4. Каждая таблица должна содержать как минимум 5 записей;
5. Для каждой таблицы разработать форму ввода (для основных рабочих таблиц рекомендуется использование формы ленточного типа);
6. Создать 2 запроса объединяющих несколько таблиц. Наименование запроса должно отражать его содержание
7. Создать 2 отчета по основным таблицам с группировкой.

Данная база данных должна реализовать потребности будущего пользователя (например: библиотекаря).

Вариант 2

1 задание. Выберите правильный ответ:

1. Файл СУБД Microsoft Access представляет собой:
 - а) файл с расширением DBF;
 - б) файл с расширением MDB;
 - в) файл с расширением MDE;
 - г) файл с расширением MDB или два файла с расширениями MDB и MDE.
2. Тип данных «Поле объекта OLE» означает:
 - а) тест произвольной длины;
 - б) хранилище рисунков или фотографий;
 - в) хранилище графических объектов;
 - г) хранилище мультимедиа-объектов.
3. Атрибут поля «Условие на значение» является:
 - а) логическим условием, которому должны удовлетворять значения в поле;
 - б) условием проверки соответствия значения типу поля;
 - в) арифметическим выражением;
 - г) логическим оператором.
4. Для организации учета выдачи книг в библиотеке создается база данных, состоящая из двух таблиц, между которыми установлены отношения подчиненности: · *таблица карточек читателей*, содержащая информацию о читателе;

· *таблица выдачи книг*, в которую заносится информация о выдаче книги читателю и о возврате книги.

Таблица карточек читателей является ... а) подчиненной;
б) главной;
в) вторичной;
г) необязательной.

5. Обновление данных возможно в следующем виде запросов:

- а) запрос на выборку из одной таблицы;
- б) запрос с использованием групповых операций;
- в) перекрестный запрос;
- г) запрос на объединение данных.

6. В поле «Дата страховки» таблицы MS Access установлен тип данных...

Имущество : таблица			
Номер	Код клиент	Наименование имущест	Дата страхов
7	104	Городская квартира	05.05.2001
5	103	Городская квартира	10.11.2000
8	105	Домовладение	12.07.2001
3	102	Домовладение	20.06.2000
1	101	Домовладение	01.01.2000
6	103	Загородная дача	10.11.2000

- а) текстовый;
- б) дата/время;
- в) поле объекта OLE;
- г) числовой.

7. База данных - это:

- а) совокупность данных со строгой внутренней организацией;
- б) набор данных большого объема;
- в) совокупность таблиц;
- г) информация доступная для использования группой людей.

8. Что из перечисленного не является объектом Access?

- а) таблицы;
- б) ключи;
- в) формы;
- г) запросы.

9. Какую строку будет занимать запись 386DX после проведения сортировки по возрастанию в поле ОП?

Компьютер	ОП	Винчестер
Pentium	16	800 Мб
386DX	4	300 Мб
486DX	8	500 Мб
Pentium Pro	32	2 Гб

10. Выберите режим, в котором можно изменить тип данных таблицы базы данных:

- а) режим Конструктора;
- б) режим Мастера;
- в) режим Таблицы;
- г) режим Формы.

2 задание.

Необходимо разработать проект базы данных по заданной предметной области и реализовать его средствами системы управления базами данных Microsoft Access. А именно разработать и создать базу данных «Кадры предприятия», учитывая следующие требования:

1. База данных должна содержать 3 таблицы;
 2. Все таблицы в проекте должны быть связаны между собой (рекомендуется использовать связь 1: M);
 3. База данных должна быть нормализована, то есть приведена к 3-ей нормальной форме;
 4. Каждая таблица должна содержать как минимум 5 записей;
 5. Для каждой таблицы разработать форму ввода (для основных рабочих таблиц рекомендуется использование формы ленточного типа);
 6. Создать 2 запроса объединяющих несколько таблиц. Наименование запроса должно отражать его содержание
 7. Создать 2 отчета по основным таблицам с группировкой.
- Данная база данных должна реализовать потребности будущего пользователя (например: работника кадров).

Вариант 3

1 задание. Выберите правильный ответ:

1. Полем базы данных Access является:
 - а) ячейка таблицы для ввода данных;
 - б) экран монитора;
 - в) столбец таблицы базы данных;
 - г) строка таблицы базы данных.
2. Тип данных «Счетчик» означает:
 - а) поле, не редактируемое пользователем;
 - б) числовое поле, редактируемое пользователем;
 - в) числовое поле с размером «длинное целое», редактируемое пользователем с помощью элемента управления «Счетчик»;
 - г) числовое поле с размером «длинное целое», автоматически заполняемое в каждой новой записи программой Access случайным или последовательным образом, не редактируемое пользователем
3. Атрибут поля «Сообщение об ошибке» служит для:
 - а) вывода сообщений при ошибках пользователя;
 - б) вывода сообщений при ошибках открытия таблицы;
 - в) задания строки, которая выводится в окне сообщений при нарушении условия, заданного атрибутом «Условие на значение»;
 - г) задания строки, которая выводится в окне сообщений при нарушении значения атрибута «Значение по умолчанию».
4. На рисунке представлена связь ...

Код	Фамилия	Имя
1	Иванов	Кузьма
2	Петров	Василий
...		

Код	Год рождения	Адрес
1	1992	Суворовский, д. 20, кв. 6
2	1993	Кирочная ул., д. 30, кв 18
...		

- а) один-ко-многим;
- б) многие-ко-многим;
- в) многие-к-одному;

г) один-к-одному.

5. Формы в СУБД Access имеют следующие области размещения данных:
- а) заголовок;
 - б) верхний и нижний колонтитул;
 - в) область данных;
 - г) примечания;
 - д) заголовок группы;
 - е) примечания группы;
 - ж) подчиненную форму;
 - з) постскриптум.
6. Поле содержит информацию об...
- а) всех свойствах одного объекта таблицы;
 - б) макетах объекта;
 - в) одном свойстве всех объектов таблицы;
 - г) поле, по которому можно однозначно идентифицировать любую запись таблицы.
7. Ключевое поле базы данных - это:
- а) поле типа "Счетчик";
 - б) поле, значения которого могут быть типа "Счетчик" или "Числовой";
 - в) поле, значения которого уникальны;
 - г) поле, значения которого логического типа.
8. Для чего предназначены запросы?
- а) хранения данных базы;
 - б) ввода данных базы и их просмотра;
 - в) автоматического выполнения группы команд;
 - г) отбора и обработки данных базы.
9. Какую строку будет занимать запись 386DX после проведения сортировки по убыванию в поле ОП?

Компьютер	ОП	Винчестер
Pentium	16	800 Мб
386DX	4	300 Мб
486DX	8	500 Мб
Pentium Pro	32	2 Гб

- а) 1;
- б) 2;
- в) 3;
- г) 4.

10. Наиболее распространенными в практике являются:
- а) распределенные базы данных;
 - б) иерархические базы данных;
 - в) сетевые базы данных;
 - г) реляционные базы данных.

2 задание.

Необходимо разработать проект базы данных по заданной предметной области и реализовать его средствами системы управления базами данных Microsoft Access. А именно раз-

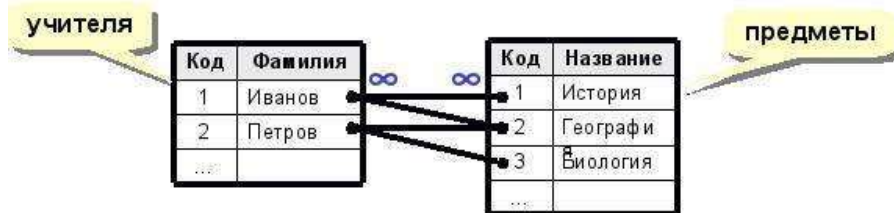
работать и создать базу данных «Телефонный справочник», учитывая следующие требования:

1. База данных должна содержать 3 таблицы;
 2. Все таблицы в проекте должны быть связаны между собой (рекомендуется использовать связь 1: M);
 3. База данных должна быть нормализована, то есть приведена к 3-ей нормальной форме;
 4. Каждая таблица должна содержать как минимум 5 записей;
 5. Для каждой таблицы разработать форму ввода (для основных рабочих таблиц рекомендуется использование формы ленточного типа);
 6. Создать 2 запроса объединяющих несколько таблиц. Наименование запроса должно отражать его содержание
 7. Создать 2 отчета по основным таблицам с группировкой.
- Данная база данных должна реализовать потребности будущего пользователя.

Вариант 4

1 задание. Выберите правильный ответ:

1. Записью базы данных Access является:
 - а) строка в таблице базы данных;
 - б) столбец в таблице базы данных;
 - в) любая текстовая строка;
 - г) любая текстовая строка длиной до 255 символов.
2. На рисунке представлена связь _____ между таблицами «учителя» и «предметы»



- а) один-ко-многим;
 - б) многие-ко-многим;
 - в) многие-к-одному;
 - г) один-к-одному.
3. В таблице «Студенты» произведена сортировка по полю _____

zach	Group	Fam	Year_b	Pregmet	Ozenka
115261	B2201	Иванов	1982	Информатика	4
234582	Э3514	Петров	1984	Информатика	5
365484	Э3514	Сидоров	1983	Математика	3
498759	A1501	Егорова	1982	Математика	5

Рисунок 1 - Исходная таблица «Студенты»

zach	Group	Fam	Year_b	Pregmet	Ozenka
498759	A1501	Егорова	1982	Математика	5
115261	B2201	Иванов	1982	Информатика	4
234582	Э3514	Петров	1984	Информатика	5
365484	Э3514	Сидоров	1983	Математика	3

Рисунок 2 - Таблица «Студенты» после сортировки»

- а) Fam;
- б) Zach;
- в) Year_b;
- г) Ozenka.

4. Установление связи между таблицами служит для:

- а) обеспечения целостности данных;
- б) обеспечения репликации данных;
- в) обеспечения копирования данных;
- г) обеспечения актуальности данных.

5. Отчеты в СУБД Access имеют следующие области размещения данных:

- а) заголовок;
- б) верхний и нижний колонтитул;
- в) область данных;
- г) примечания;
- д) заголовок группы;
- е) примечания группы.

6. Ключевое поле может быть в

- а) всех свойствах одного объекта таблицы;
- б) изменять макет объекта;
- в) одном свойстве всех объектов таблицы;
- г) поле, по которому можно однозначно идентифицировать любую запись таблицы.

7. Дана следующая база данных:

№ п/п	Фамилия	Имя	Отчество	Год рождения	Должность	Оклад
1.	Иванов	Федор	Сергеевич	1955	Переводчик	7500 руб.
2.	Петров	Виктор	Николаевич	1964	Консультант	8800 руб.
3.	Попов	Вадим	Валерьевич	1981	Кассир	5500 руб.
4.	Румянцева	Валентина	Рудольфовна	1958	Бухгалтер	11200 руб.
5.	Бабенко	Григорий	Иванович	1955	Директор	12000 руб.

Какое поле может являться ключевым?

- а) Оклад;
- б) Должность;
- в) № п/п;
- г) Год рождения.

8. Для чего предназначены формы?

- а) хранения данных базы;
- б) ввода данных базы и их просмотра;
- в) отбора и обработки данных базы;
- г) автоматического выполнения группы команд.

9. Какую строку будет занимать запись Pentium Pro после проведения сортировки по возрастанию в поле ОП?

Компьютер	ОП	Винчестер
Pentium	16	800 Мб
386DX	4	300 Мб

486DX	8	500 Мб
Pentium Pro	32	2 Гб

- а) 1;
- б) 2;
- в) 3;
- г) 4.

10. Наиболее точным аналогом реляционной базы данных может служить:

- а) неупорядоченное множество данных;
- б) вектор;
- в) генеалогическое дерево;
- г) двумерная таблица.

2 задание.

Необходимо разработать проект базы данных по заданной предметной области и реализовать его средствами системы управления базами данных Microsoft Access. А именно разработать и создать базу данных «Студенты», учитывая следующие требования:

1. База данных должна содержать 3 таблицы;
2. Все таблицы в проекте должны быть связаны между собой (рекомендуется использовать связь 1: М);
3. База данных должна быть нормализована, то есть приведена к 3-ей нормальной форме;
4. Каждая таблица должна содержать как минимум 5 записей;
5. Для каждой таблицы разработать форму ввода (для основных рабочих таблиц рекомендуется использование формы ленточного типа);
6. Создать 2 запроса объединяющих несколько таблиц. Наименование запроса должно отражать его содержание
7. Создать 2 отчета по основным таблицам с группировкой.

Данная база данных должна реализовать потребности будущего пользователя (например: классного руководителя).

Вариант 5

1 задание. Выберите правильный ответ:

1. Типами полей СУБД Access являются:
 - а) текстовый;
 - б) числовой;
 - в) дата и время;
 - г) формула.
2. Тип данных «Гиперссылка» означает:
 - а) ссылку по другой ссылке;
 - б) ссылку на любую другую ссылку;
 - в) ссылку на любой другой объект базы данных;
 - г) ссылку на любой документ в Интернет.
3. На рисунке представлена _____ для ввода и редактирования информации

Комплектующие

Компонент: Процессор

Модель: 888

Основной параметр: 7889

Цена: 500

Компонент	Модель	Основной л.	Цена
Жесткий диск	тп	10	180
Клавиатура	ооо	136	150
Мышь	мм	234	100
Материнская плата	ппп		430
Монитор	ккк	58	80
Оперативная память	ччч	864	270
Процессор	888	7889	500

- а) таблица;
- б) форма;
- в) отчет;
- г) запрос.

4. В СУБД MS Access не существует запрос на _____ данных. а) создание;
б) обновление;
в) удаление;
г) добавление.
5. Инфологическая модель представляет собой:
 - а) описание предметной области в понятиях ER-диаграмм;
 - б) описание предметной области в понятиях конкретной СУБД;
 - в) схему данных;
 - г) описание прав пользователей при организации многопользовательского доступа к базе данных.
6. В режиме конструктора можно:
 - а) узнать все свойства одного объекта таблицы;
 - б) изменять макет объекта;
 - в) узнать об одном свойстве всех объектов таблицы;
 - г) изменить поле, по которому можно однозначно идентифицировать любую запись таблицы.
7. Запись табличной базы данных - это:
 - а) Совокупность столбцов таблицы;
 - б) Столбец таблицы;
 - в) Строка таблицы;
 - г) Совокупность строк таблицы.
8. В каком диалоговом окне создают связи между полями таблиц базы данных?
 - а) схема данных;
 - б) таблица связей;
 - в) схема связей;
 - г) таблица данных.
9. Какую строку будет занимать запись Pentium Pro после проведения сортировки по убыванию в поле ОП?

Компьютер	ОП	Винчестер
Pentium	16	800 Мб
386DX	4	300 Мб

486DX	8	500 Мб
Pentium Pro	32	2 Гб

- а) 1;
- б) 2;
- в) 3;
- г) 4.

10. Для чего предназначены макросы:
- а) для хранения данных базы;
 - б) для отбора и обработки данных базы;
 - в) для ввода данных базы и их просмотра;
 - г) для автоматического выполнения команд.

2 задание.

Необходимо разработать проект базы данных по заданной предметной области и реализовать его средствами системы управления базами данных Microsoft Access. А именно разработать и создать базу данных «Продажа товара», учитывая следующие требования:

1. База данных должна содержать 3 таблицы;
2. Все таблицы в проекте должны быть связаны между собой (рекомендуется использовать связь 1: М);
3. База данных должна быть нормализована, то есть приведена к 3-ей нормальной форме;
4. Каждая таблица должна содержать как минимум 5 записей;
5. Для каждой таблицы разработать форму ввода (для основных рабочих таблиц рекомендуется использование формы ленточного типа);
6. Создать 2 запроса объединяющих несколько таблиц. Наименование запроса должно отражать его содержание
7. Создать 2 отчета по основным таблицам с группировкой.

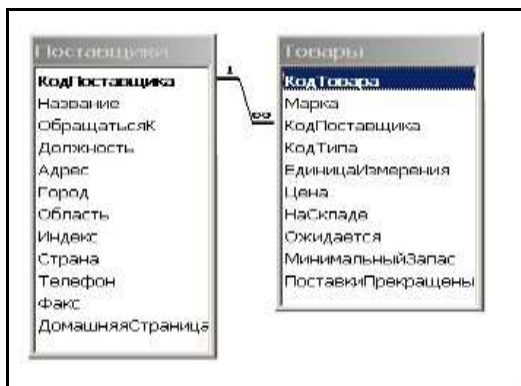
Данная база данных должна реализовать потребности будущего пользователя (например: продавца).

6 вариант

1 задание. Выберите правильный ответ:

1. СУБД Microsoft Access относится:
 - а) к настольным;
 - б) к персональным;
 - в) с серверным;
 - г) к индивидуальным.
2. Тип данных поля «Логическое» означает:
 - а) поле, значение в котором может принимать значение «Да» или «Нет»;
 - б) поле, значение в котором может принимать значение «Вкл» или «Выкл»;
 - в) поле, значение в котором может принимать значение «Истина» или «Ложь»;
 - г) поле, значение в котором может принимать значение «0» или «-1».
3. Общий формат для типа поля «ДАТА/ВРЕМЯ» задается в виде:
 - а) dd.mm.yyyy hh:nn:ss;
 - б) dd.mm.yy;
 - в) «Полный формат даты»;
 - г) dd.mm.yyyy hh:mm:ss.

4. На рисунке изображена следующий вид связи:
- «Один-ко-многим», обеспечение целостности данных включено;
 - «Один-к-одному», обеспечение целостности данных включено;
 - «Один-к-одному», обеспечение целостности данных выключено;
 - «Много-ко-многим», обеспечение целостности данных включено.



5. Дatalogическая модель представляет собой:
- описание предметной области в понятиях ER-диаграмм;
 - описание предметной области в понятиях конкретной СУБД;
 - схему данных;
 - описание прав пользователей при организации многопользовательского доступа к базе данных.
6. Запись содержит информацию о...
- всех свойствах одного объекта таблицы;
 - макетах объекта;
 - одном свойстве всех объектов таблицы;
 - поле, по которому можно однозначно идентифицировать любую запись таблицы.
7. Поле табличной базы данных - это:
- совокупность столбцов таблицы;
 - столбец таблицы;
 - совокупность строк таблицы;
 - строка таблицы.
8. Без каких объектов не может существовать база данных?
- без отчетов;
 - без форм;
 - без запросов;
 - без таблиц.
9. Какую строку будет занимать запись 486DX после проведения сортировки по возрастанию в поле ОП?

Компьютер	ОП	Винчестер
Pentium	16	800 Мб
386DX	4	300 Мб
486DX	8	500 Мб
Pentium Pro	32	2 Гб

10. В каком режиме работает с базой данных пользователь:

- а) в проектировочном;
- б) в любительском;
- в) в заданном;
- г) в эксплуатационном.

2 задание.

Необходимо разработать проект базы данных по заданной предметной области и реализовать его средствами системы управления базами данных Microsoft Access. А именно разработать и создать базу данных «Касса аэрофлота», учитывая следующие требования:

1. База данных должна содержать 3 таблицы;
2. Все таблицы в проекте должны быть связаны между собой (рекомендуется использовать связь 1: М);
3. База данных должна быть нормализована, то есть приведена к 3-ей нормальной форме;
4. Каждая таблица должна содержать как минимум 5 записей;
5. Для каждой таблицы разработать форму ввода (для основных рабочих таблиц рекомендуется использование формы ленточного типа);
6. Создать 2 запроса объединяющих несколько таблиц. Наименование запроса должно отражать его содержание
7. Создать 2 отчета по основным таблицам с группировкой.

Данная база данных должна реализовать потребности будущего пользователя (например: кассира). Кассир, который работает в аэрофлоте, должен всего лишь открыть данную базу данных, заполнить ее необходимыми данными, и при необходимости воспользоваться этими данными. То есть пользователь (например: кассир) должен работать с ней, как с готовым программным продуктом, не задумываясь о ее разработке и редактировании.

База данных «Касса аэрофлота» должна решать следующие потребности пользователя (кассира): вести учет пассажиров, продаваемых билетов, расписание рейсов. Помочь легко и быстро заполнить имеющиеся справочники данными и работать с ними.

7 вариант

1 задание. Выберите правильный ответ:

1. Сетевая база данных – это...
 - а) БД, в которой информация организована в виде прямоугольных таблиц;
 - б) БД, в которой элементы в записи упорядочены, т.е. один элемент считается главным, остальные подчиненными;
 - в) БД, в которой записи расположены в произвольном порядке;
 - г) БД, в которой принята свободная связь между элементами разных уровней.
2. Мастер – это...
 - а) программный модуль для вывода операций;
 - б) программный модуль для выполнения, каких либо операций;
 - в) режим, в котором осуществляется построение таблицы или формы;
 - г) режим, в котором осуществляется вывод таблицы или формы.
3. Тип данных «Поле-МЕМО» означает:
 - а) Любую дату после 2000 года;
 - б) Строку произвольной длины;
 - в) Текст длиной до 65 535 символов;
 - г) Число в диапазоне от -1E108 до 1E108.

4. БД содержит информацию об учениках школы: фамилия, класс, балл за тест, балл за практическое задание, общее количество баллов. Какого типа должно быть поле ОБЩЕЕ КОЛИЧЕСТВО БАЛЛОВ?

- а) символьное;
- б) логическое;
- в) числовое;
- г) любого типа.

5. Объект «Таблица»:

- а) предназначен для хранения информации;
- б) предназначен для ввода и просмотра данных;
- в) предназначен для выдачи данных по определенным параметрам;
- г) предназначен для выдачи данных на экран или печать.

6. В записи реляционной базы данных может содержаться:

- а) неоднородная информация (данные разных типов);
- б) исключительно однородная информация (данные только одного типа);
- в) исключительно числовая информация;
- г) только текстовая информация.

7. Дана следующая база данных:

№ п/п	Фамилия	Имя	Отчество	Год рождения	Должность	Оклад
1.	Иванов	Федор	Сергеевич	1955	Переводчик	7500 руб.
2.	Петров	Виктор	Николаевич	1964	Консультант	8800 руб.
3.	Попов	Вадим	Валерьевич	1981	Кассир	5500 руб.
4.	Румянцева	Валентина	Рудольфовна	1958	Бухгалтер	11200 руб.
5.	Бабенко	Григорий	Иванович	1955	Директор	12000 руб.

Количество полей в ней равно?

- а) 5;
- б) 6;
- в) 7;
- г) 4.

8. В чем состоит особенность поля типа «Счетчик»?

- а) служит для ввода числовых данных;
- б) имеет ограниченный размер;
- в) имеет свойство автоматического наращивания;
- г) служит для ввода действительных чисел.

9. Какую строку будет занимать запись 486DX после проведения сортировки по убыванию в поле ОП?

Компьютер	ОП	Винчестер
Pentium	16	800 Мб
386DX	4	300 Мб
486DX	8	500 Мб
Pentium Pro	32	2 Гб

10. Почему при закрытии таблицы программа Access не предлагает выполнить сохранение внесенных данных:

- а) недоработка программы;
- б) потому что данные сохраняются сразу после ввода в таблицу;
- в) потому что данные сохраняются только после закрытия всей базы данных;
- г) потому что необходимо перед закрытием программы сохранить самому.

2 задание.

Необходимо разработать проект базы данных по заданной предметной области и реализовать его средствами системы управления базами данных Microsoft Access. А именно разработать и создать базу данных «Аптека», учитывая следующие требования:

- 1. База данных должна содержать 3 таблицы;
- 2. Все таблицы в проекте должны быть связаны между собой (рекомендуется использовать связь 1: М);
- 3. База данных должна быть нормализована, то есть приведена к 3-ей нормальной форме;
- 4. Каждая таблица должна содержать как минимум 5 записей;
- 5. Для каждой таблицы разработать форму ввода (для основных рабочих таблиц рекомендуется использование формы ленточного типа);
- 6. Создать 2 запроса объединяющих несколько таблиц. Наименование запроса должно отражать его содержание
- 7. Создать 2 отчета по основным таблицам с группировкой.

Данная база данных должна реализовать потребности будущего пользователя (например: фармацевта).

8 вариант

1 задание. Выберите правильный ответ:

- 1. Конструктор – это...
 - а) программный модуль для вывода операций;
 - б) программный модуль для выполнения, каких либо операций;
 - в) режим, в котором осуществляется построение таблицы или формы;
 - г) режим, в котором осуществляется вывод таблицы или формы.
- 2. Атрибут поля «Подпись» служит для:
 - а) указания заголовка для поля в режиме таблицы;
 - б) указание заголовка для поля в режиме формы;
 - в) ввода электронной подписи;
 - г) задания значения, присваемого полю в каждой новой записи.
- 3. Запрос, позволяющий определить одно или несколько условий отбора во время выполнения запроса:
 - а) выборка;
 - б) перекрестный;
 - в) изменение;
 - г) с параметром.
- 4. Структура базы данных изменится, если
 - а) добавить/удалить запись;
 - б) добавить/удалить поле;
 - в) отредактировать запись;
 - г) поменять местами записи.
- 5. Объект «Форма»:
 - а) предназначен для хранения информации;

- б) предназначен для ввода и просмотра данных;
- в) предназначен для выдачи данных по определенным параметрам;
- г) предназначен для выдачи данных на экран или печать.

6. Дана следующая база данных:

№ п/п	Фамилия	Имя	Отчество	Год рождения	Должность	Оклад
1.	Иванов	Федор	Сергеевич	1955	Переводчик	7500 руб.
2.	Петров	Виктор	Николаевич	1964	Консультант	8800 руб.
3.	Попов	Вадим	Валерьевич	1981	Кассир	5500 руб.
4.	Румянцева	Валентина	Рудольфовна	1958	Бухгалтер	11200 руб.
5.	Бабенко	Григорий	Иванович	1955	Директор	12000 руб.

Количество записей в ней равно?

- а) 4;
- б) 6;
- в) 7;
- г) 5.

7. Какую строку будет занимать запись Pentium после проведения сортировки по возрастанию в поле ОП?

Компьютер	ОП	Винчестер
Pentium	16	800 Мб
386DX	4	300 Мб
486DX	8	500 Мб
Pentium Pro	32	2 Гб

8. Чтобы изменить имя поля базы данных, надо:

- а) в режиме Конструктора выделить исходное имя поля и ввести новое;
- б) воспользоваться меню Главная, вкладка Создание;
- в) в режиме Конструктора установить Маску ввода;
- г) в режиме Конструктора поменять тип данных нужного поля.

9. В классификации типов СУБД присутствуют ...

- а) реляционные;
- б) модемные;
- в) иерархические;
- г) сетевые.

10. Атрибут поля «Размер поля» имеет смысл для:

- а) для Мемо-полей;
- б) для полей типа «Денежное»;
- в) для текстовых полей;
- г) для типа «Счетчик».

2 задание.

Необходимо разработать проект базы данных по заданной предметной области и реализовать его средствами системы управления базами данных Microsoft Access. А именно разработать и создать базу данных «Детский сад», учитывая следующие требования:

1. База данных должна содержать 3 таблицы;
2. Все таблицы в проекте должны быть связаны между собой (рекомендуется использовать связь 1: М);

3. База данных должна быть нормализована, то есть приведена к 3-ей нормальной форме;
4. Каждая таблица должна содержать как минимум 5 записей;
5. Для каждой таблицы разработать форму ввода (для основных рабочих таблиц рекомендуется использование формы ленточного типа);
6. Создать 2 запроса объединяющих несколько таблиц. Наименование запроса должно отражать его содержание
7. Создать 2 отчета по основным таблицам с группировкой.
Данная база данных должна реализовать потребности будущего пользователя (например: воспитателя).

9 вариант

1 задание. Выберите правильный ответ:

1. БД содержит информацию о собаках из клуба собаководства: кличка, порода, дата рождения, наличие родословной, количество медалей. Какого типа должны быть поля?
 - а) текстовое, текстовое, числовое, текстовое, числовое;
 - б) текстовое, текстовое, дата, текстовое, числовое;
 - в) текстовое, текстовое, дата, логическое, числовое;
 - г) текстовое, текстовое, дата, логическое, текстовое.
2. Сколько записей в базе данных?
 - а) 2;
 - б) 4;
 - в) 3;
 - г) 5.

3. Реляционная задана таблицей:

№	Компьютер	ОЗУ	Винчестер
1	Pentium	16	800Мб
2	386DX	4	300 Мб
3	486DX	8	500Мб
4	Pentium Pro	32	2Гб

	Ф.И.О.	пол	возраст	клуб	спорт
1	Панько Л.П.	жен	22	Спарта	футбол
2	Арбузов А.А.	муж	20	Динамо	лыжи
3	Жиганова П.Н.	жен	19	Ротор	футбол
4	Иванов О.Г.	муж	21	Звезда	лыжи
5	Седова О.Л.	жен	18	Спарта	биатлон
6	Багаева С.И.	жен	23	Звезда	лыжи

база данных

- Какие записи будут выбраны по условию: спорт = «лыжи» И пол = «жен» ИЛИ возраст < 20
- а) 2, 3, 4, 5, 6;
 - б) 3, 5, 6;
 - в) 1, 3, 5, 6;
 - г) таких записей нет;
 - д) 2, 3, 5, 6.
4. Сортировка в запросе невозможна по следующему типу полей:
 - а) Счетчик;
 - б) Логическое поле;
 - в) Гиперссылка;

г) Поле-МЕМО.

5. Объект «Запрос»:

- а) предназначен для хранения информации;
- б) предназначен для ввода и просмотра данных;
- в) предназначен для выдачи данных по определенным параметрам;
- г) предназначен для выдачи данных на экран или печать.

6. Какого типа данных в MS Access не существует...

- а) логический;
- б) финансовый;
- в) текстовый;
- г) счетчик.

7. Предположим, что некоторая база данных содержит поля "Фамилия", "Год рождения", "Доход". При поиске по условию (Год рождения>1958 and Доход<3500) будут найдены фамилии лиц:

- а) имеющих доход менее 3500 и тех, кто родился в 1958 году и позже;
- б) имеющих доход менее 3500 или тех, кто родился в 1958 и позже;
- в) имеющих доход менее 3500 или тех кто родился позже 1958 года;
- г) имеющих доход менее 3500 и тех кто родился позже 1958 года.

8. Какую строку будет занимать запись Pentium после проведения сортировки по убыванию в поле ОП?

Компьютер	ОП	Винчестер
Pentium	16	800 Мб
386DX	4	300 Мб
486DX	8	500 Мб
Pentium Pro	32	2 Гб

- а) 1;
- б) 2;
- в) 3;
- г) 4.

9. Чтобы установить связи между таблицами, надо воспользоваться:

- а) вкладка Работа с базами данных, Схема данных;
- б) вкладка Создание, меню Формы;
- в) Схема данных, вкладка Главная;
- г) вкладка Внешние данные, Схема данных.

10. Реляционная база данных – это хранилище данных

- а) в структуре файловой системы;
- б) в структуре связанных страниц;
- в) в структуре связанных таблиц;
- г) произвольной структуры.

2 задание.

Необходимо разработать проект базы данных по заданной предметной области и реализовать его средствами системы управления базами данных Microsoft Access. А именно разработать и создать базу данных «Вокзал», учитывая следующие требования:

1. База данных должна содержать 3 таблицы;
2. Все таблицы в проекте должны быть связаны между собой (рекомендуется использовать связь 1: М);
3. База данных должна быть нормализована, то есть приведена к 3-ей нормальной форме;
4. Каждая таблица должна содержать как минимум 5 записей;
5. Для каждой таблицы разработать форму ввода (для основных рабочих таблиц рекомендуется использование формы ленточного типа);
6. Создать 2 запроса объединяющих несколько таблиц. Наименование запроса должно отражать его содержание
7. Создать 2 отчета по основным таблицам с группировкой.

Данная база данных должна реализовать потребности будущего пользователя (например: кассира). Кассир, который работает в вокзале, должен всего лишь открыть данную базу данных, заполнить ее необходимыми данными, и при необходимости воспользоваться этими данными. То есть пользователь (например: кассир) должен работать с ней, как с готовым программным продуктом, не задумываясь о ее разработке и редактировании.

База данных «Вокзал» должна решать следующие потребности пользователя (кассира):

вести учет пассажиров, продаваемых билетов, расписание рейсов. Помочь легко и быстро заполнить имеющиеся справочники данными и работать с ними.

10 вариант

1 задание. Выберите правильный ответ:

1. Определите тип базы данных:

№	класс	фамилия	имя	«5»	«4»	«3»	«2»
1	10 а	Иванов	Петя	7	12	1	1
2	10 б	Петров	Юра	8	14	5	0
3	11 а	Сидорова	Катя	1	17	2	0

- а) реляционная;
- б) иерархическая;
- в) сетевая;
- г) модемная.

2. База данных «Ученики» выглядит так:

Фамилия	Класс	Адрес	Школа	Оценка
Иванов	10а	Уфа	345	5

Количество полей в этой базе данных равно

- а) 10;
- б) 5;
- в) 2;
- г) 1.

3. Запрос, предназначенный для группирования данных и представления их в компактном виде: а) выборка;
б) перекрестный;
в) изменение;
г) с параметром.

4. Кнопка  обозначает

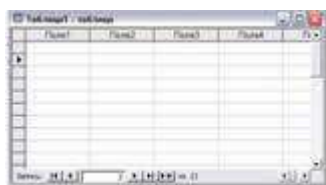
- а) сортировку записей по возрастанию;

- б) сортировку записей по убыванию;
- в) вывод на экран записей начинающихся с эти букв;
- г) подсчет количества записей в БД

5. Объект «Отчет»:

- а) предназначен для хранения информации;
- б) предназначен для ввода и просмотра данных;
- в) предназначен для выдачи данных по определенным параметрам;
- г) предназначен для выдачи данных на экран или печать.

6. На рисунке представлено окно для создания таблицы базы данных MS Access в режиме...



- а) создание таблицы с помощью мастера;
- б) создание таблицы в режиме конструктора;
- в) создание таблицы путем ввода данных;
- г) создание пустой таблицы.

7. Содержит ли какую-либо информацию таблица, в которой нет полей?

- а) содержит информацию о структуре базы данных;
- б) не содержит никакой информации;
- в) таблица без полей существовать не может;
- г) содержит информацию о будущих записях.

8. Как называется категория запросов, предназначенная для выбора данных из таблиц базы данных:

- а) запросы на сортировку;
- б) запросы на выборку;
- в) запросы с параметром;
- г) запросы итоговые.

9. Сколько баз данных MS Access может быть открыто одновременно а) 1;

- б) 2;
- в) неограниченное количество;
- г) 3.

10. Тип данных «Текстовое поле» означает:

- а) любую строку длиной до 50 символов;
- б) любую строку длиной до 255 символов;
- в) строку любой длины без пробелов и специальных символов;
- г) строку любой длины, включая пробелы и специальные символы.

2 задание.

Необходимо разработать проект базы данных по заданной предметной области и реализовать его средствами системы управления базами данных Microsoft Access. А именно разработать и создать базу данных «Сотрудники фирмы», учитывая следующие требования:

1. База данных должна содержать 3 таблицы;
2. Все таблицы в проекте должны быть связаны между собой (рекомендуется использовать связь 1: М);

3. База данных должна быть нормализована, то есть приведена к 3-ей нормальной форме;
4. Каждая таблица должна содержать как минимум 5 записей;
5. Для каждой таблицы разработать форму ввода (для основных рабочих таблиц рекомендуется использование формы ленточного типа);
6. Создать 2 запроса объединяющих несколько таблиц. Наименование запроса должно отражать его содержание
7. Создать 2 отчета по основным таблицам с группировкой.
Данная база данных должна реализовать потребности будущего пользователя (например: директора).