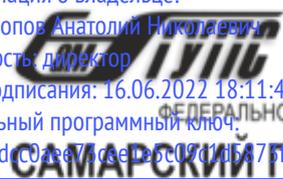


Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Попов Анатолий Николаевич
Должность: директор
Дата подписания: 16.06.2022 18:11:44
Уникальный программный ключ:
1e0c38dccc0aee71dce1e6c09d1d5875tc7497bc8



МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ АГЕНТСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
САМАРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ПУТЕЙ СООБЩЕНИЯ

Приложение 2
к рабочей программе дисциплины

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

Современные СУБД

(наименование дисциплины(модуля))

Направление подготовки / специальность

09.03.03 Прикладная информатика
(код и наименование)

Направленность (профиль)/специализация

Прикладная информатика на железнодорожном транспорте
(наименование)

Содержание

1. Пояснительная записка.
2. Типовые контрольные задания или иные материалы для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих уровень сформированности компетенций.
3. Методические материалы, определяющие процедуру и критерии оценивания сформированности компетенций при проведении промежуточной аттестации.

1. Пояснительная записка

Цель промежуточной аттестации – оценивание промежуточных и окончательных результатов обучения по дисциплине, обеспечивающих достижение планируемых результатов освоения образовательной программы.

Перечень компетенций, формируемых в процессе освоения дисциплины

Код и наименование компетенции	Код индикатора достижения компетенции
ПК-2.1 Использует существующие типовые решения и шаблоны проектирования программного обеспечения	Знает: возможности существующей программно-технической архитектуры, современных средств разработки программных продуктов; методологии разработки программного обеспечения и технологии программирования, проектирования и использования баз данных; языки формализации функциональных спецификаций, методы и приемы формализации задач; типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения;
	Умеет: выбирать средства реализации требований к программному обеспечению; выработать варианты реализации программного обеспечения; использовать существующие типовые решения и шаблоны проектирования программного обеспечения;
	Владеет: способами проектирования структур данных, баз данных, программных интерфейсов
ПК-2.2 Применяет методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов	Знает: принципы проектирования структуры базы данных, которая удовлетворяет требованиям функциональности АИС современные технологии разработки приложений базы данных, программных интерфейсов, структур данных
	Умеет: использовать современные инструментальные средства и технологии программирования разрабатывать функциональность автоматизированной информационной системы разрабатывать программные компоненты для работы с базами данных разрабатывать пользовательский интерфейс автоматизированной информационной системы
	Владеет: навыками работы с различными СУБД и их администрирования методами проектирования структуры базы данных технологией ADO .NET и EntityFramework для доступа к базе данных различных СУБД навыками создания web-приложений базы данных на основе шаблона проектирования MVC.

Результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы

Код и наименование индикатора достижения компетенции	Результаты обучения по дисциплине	Оценочные материалы
ПК-2.1 Использует существующие типовые решения и шаблоны проектирования программного обеспечения	Знает: возможности существующей программно-технической архитектуры, современных средств разработки программных продуктов; методологии разработки программного обеспечения и технологии	

	программирования, проектирования и использования баз данных; языки формализации функциональных спецификаций, методы и приемы формализации задач; типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения;	
	Умеет: выбирать средства реализации требований к программному обеспечению; выработать варианты реализации программного обеспечения; использовать существующие типовые решения и шаблоны проектирования программного обеспечения;	
	Владеет: способами проектирования структур данных, баз данных, программных интерфейсов	
ПК-2.2 Применяет методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов	Знает: принципы проектирования структуры базы данных, которая удовлетворяет требованиям функциональности АИС современные технологии разработки приложений базы данных, программных интерфейсов, структур данных	
	Умеет: использовать современные инструментальные средства и технологии программирования разрабатывать функциональность автоматизированной информационной системы разрабатывать программные компоненты для работы с базами данных разрабатывать пользовательский интерфейс автоматизированной информационной системы	
	Владеет: навыками работы с различными СУБД и их администрирования методами проектирования структуры базы данных технологией ADO .NET и EntityFramework для доступа к базе данных различных СУБД навыками создания web-приложений базы данных на основе шаблона проектирования MVC.	

Промежуточная аттестация (зачет) проводится в одной из следующих форм:

- 1) собеседование;
- 2) выполнение заданий в ЭИОС СамГУПС.

2. Типовые¹ контрольные задания или иные материалы для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих уровень сформированности компетенций

2.1 Типовые задания для оценки навыкового образовательного результата

Проверяемый образовательный результат

Код и наименование индикатора достижения компетенции	Образовательный результат
--	---------------------------

¹Приводятся типовые вопросы и задания. Оценочные средства, предназначенные для проведения аттестационного мероприятия, хранятся на кафедре в достаточном для проведения оценочных процедур количестве вариантов. Оценочные средства подлежат актуализации с учетом развития науки, образования, культуры, экономики, техники, технологий и социальной сферы. Ответственность за нераспространение содержания оценочных средств среди обучающихся университета несут заведующий кафедрой и преподаватель – разработчик оценочных средств.

I. ЗАДАНИЕ ВЫПОЛНЕНИЯ ПРАКТИЧЕСКОЙ РАБОТЫ

Практическая работа №1 выполняется письменно и в конце занятия сдается на проверку. После проверки будет выставлена оценка.

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.

Анализ предметной области.

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

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 связь между сущностями. Для выявленных связей так же нужно заполнить таблицу 1.3.

Таблица 1.3. Список связей ПО.

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

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:Мит.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.

Построение даталогической модели.

На основании ранее выбранного варианта и таблиц 1.1-1.3, инфологической модели инормализации 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.

№ п. п.	Наименование реквизита	Идентификатор	Тип	Длина	Формат изображения	Ограничения и комментарий

ПК-2.2

Применяет методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов

УСТАНОВКА СОЕДИНЕНИЯ С СЕРВЕРОМ MICROSOFT SQL SERVER И ПРИНЦИПЫ СОЗДАНИЯ БАЗ ДАННЫХ

Цель практической работы

Познакомиться с основными принципами создания базы данных в MS SQL Server. Изучить операции, проводимые с базами данных в целом. Получить навыки использования программы "SQL Server Management Studio" для создания, удаления, регистрации, подключения, извлечения метаданных, резервного копирования и восстановления базы данных. Изучить SQL-операторы для создания, подключения и удаления базы данных. Познакомиться с основными принципами управления учетными записями и ролями.

Исходные данные

Студент получает индивидуальный вариант исходных данных кратким описанием предметной области, который используется при выполнении всех описанных в данном пособии практических работ. При этом каждая очередная Практическая работа является продолжением выполненной ранее и поэтому они должны обязательно выполняться последовательно.

Используемые программы

1. Работающий на компьютере сервер "Microsoft SQL Server 2008 R2".
2. Установленная платформа .NET Framework 2.0, 3.0, 3.5 или 4.0.
3. Операционная система Microsoft Windows 2000/XP/2003/Vista/Windows 7/Windows 8/Windows 10.
4. Приложение "SQL Server Management Studio 2008rus", установленное на локальном компьютере.

Теоретические сведения

На сегодняшний день известно более двух десятков серверных СУБД, из которых наиболее популярными являются Oracle, Microsoft SQL Server, Informix, DB2, Sybase, InterBase, MySQL.

Для выполнения практических работ будет использоваться сервер "Microsoft SQL Server 2008", установленный на сервере кафедры компьютерных систем и сетей (компьютер pi_srv).

Microsoft® SQL Server™

— это система анализа и управления реляционными базами данных в решениях электронной коммерции, производственных отраслей и хранилищ данных.

Microsoft SQL Server —

система управления реляционными базами данных (СУБД), разработанная корпорацией Microsoft. Основным используемым языком запросов — **Transact-SQL**, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта

ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия; конкурирует с другими СУБД в этом сегменте рынка.

В SQL Server 2008 имеется большой набор интегрированных служб, расширяющих возможности использования данных: вы можете составлять запросы, выполнять поиск, проводить синхронизацию, делать отчеты, анализировать данные. Все данные хранятся на основных серверах, входящих в состав центра обработки данных. К ним осуществляется доступ на столевых компьютерах, мобильных устройствах. Таким образом, полностью контролируются данные независимо от того, где они хранятся.

Система SQL Server 2008 позволяет обращаться к данным из любого приложения, разработанного

с применением технологий Microsoft .NET и VisualStudio, а также в пределах сервисно-ориентированной архитектуры и бизнес-процессов — через MicrosoftBizTalkServer. Сотрудники, отвечающие за сбор и анализ информации, могут работать с данными, не покидая привычных приложений, которыми они пользуются каждый день, например приложений выпуска 2007 системы MicrosoftOffice.

ВMicrosoftSQLбазы данных хранятся в виде обычных файлов на диске. Как минимум на одну БД приходится таких файлов 2: *.mdf и *.ldf. В первом хранятся сами данные, таблицы, индексы и пр., а во втором находится т.н. transactionlog, в котором находится информация необходимая для восстановления БД.

Файл базой данных представляет собой набор страниц одинакового размера. Размер страницы задается при создании базы данных и может быть изменен только при ее восстановлении из резервной копии. Чтение и запись данных в базе данных осуществляется постранично.

Все операции с базой данных должны производиться только посредством команд SQL-серверу. Для клиентских приложений эти файлы абсолютно бесполезны при правильной организации доступа папользователей к файлам в сети, вообще не должны быть доступны.

Сервер СУБД не имеет интерфейса пользователя для выполнения операций с базой данных ему необходимо посылать команды либо с помощью командной строки или с помощью какой-либо прикладной программы.

Для выполнения операций с базой данных при проведении практических работ предлагается использовать программу "SQLServerManagementStudio2008Rus" (рис. 1), представляющую собой наиболее распространенное и удобное средство администрирования баз данных под управлением MSSQLServer (Среда ManagementStudio Express доступна для свободной загрузки из центра загрузки Майкрософт-http://download.microsoft.com/download/5/C/0/5C0C5CE4-10EB-4623-A63E-8D850D55D8EF/SQLEXPRESS_x86_RUS.exe).

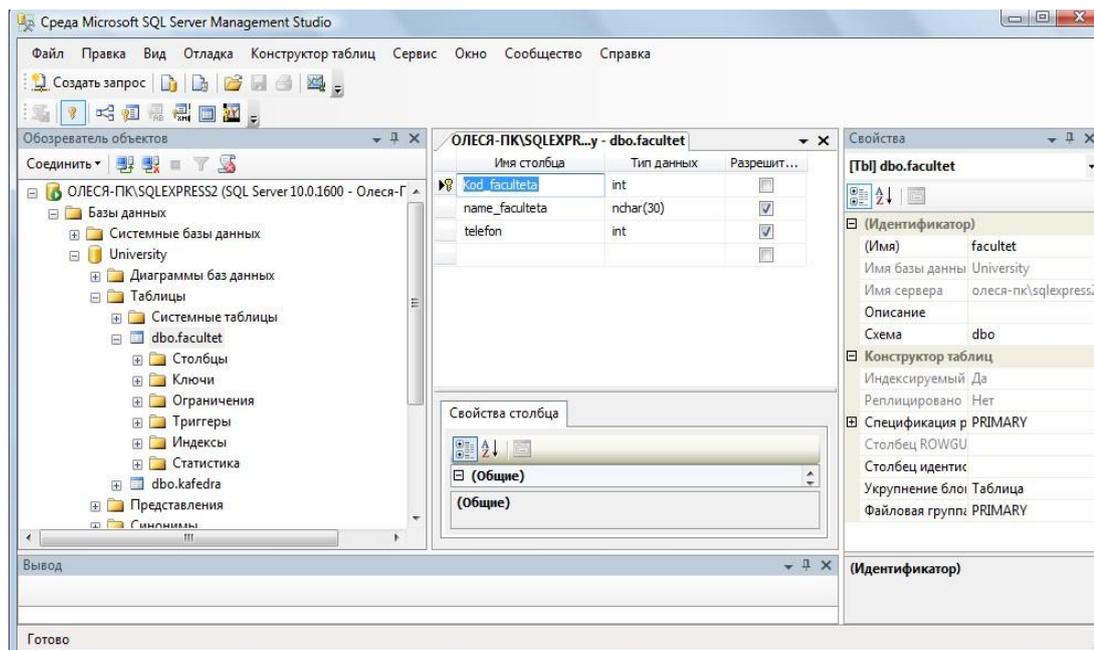


Рис. 1. Программа SQLServerManagementStudio

Среда SQL ServerManagementStudio — это интегрированная среда для доступа, настройки, управления, администрирования и разработки всех компонентов SQL Server. Среда SQL ServerManagementStudio объединяет большое число графических средств с набором полнофункциональных редакторов сценариев для доступа к SQL Server разработчиков и администраторов в любой момент работы.

Среда SQLServerManagementStudio обеспечивает следующие основные возможности:

- поддерживает большинство административных задач для SQL Server;
- единая интегрированная среда для управления SQL Server Database Engine и разработки;
- новые управляющие диалоговые окна для управления объектами в компоненте SQL Server Database Engine, службах Analysis Services, Reporting Services, Notification Services и выпуске SQL Server Compact 3.5 с пакетом обновления 1 (SP1), позволяющие выполнять действия немедленно, направляя их в редактор кода или включать эти действия в сценарий для последующего выполнения;
- экспорт и импорт регистрации сервера среды SQL ServerManagementStudio из одной среды ManagementStudio в другую;

- сохранение и печать XML-файлов плана выполнения и взаимоблокировок, созданных приложением SQL Server Profiler, просмотр их в любое время и отправка для анализа администратору;
- новые окна сообщений об ошибках и информационных сообщениях, предоставляющие гораздо больше сведений и позволяющие отправлять в Майкрософт комментарии о сообщениях, копировать сообщения в буфер обмена и отправлять их по электронной почте в службу поддержки;
- встроенный веб-обозреватель для быстрого обращения к библиотеке MSDN или получения интерактивной справки;
- встроенная справка от сообщества Интернет и т.д.

Большинство действий с базой данной MS SQL Server в среде Среда SQL Server Management Studio может быть осуществлено двумя способами: **либо выполнением операторов языка SQL** в окнах "Script Execute" (подключение баз данных необязательно) и "SQL Editor" (требуется подключение баз данных), либо с использованием меню и диалоговых окон. В последнем случае операторы SQL, которые требуются для выполнения данного действия, будут сгенерированы и выполнены средой SQL Server Management Studio автоматически.

Задание

Практическую работу следует выполнять в следующем порядке:

1. Создать на сервере pi_srv (или на локальном компьютере, если нет сервера) рабочую папку для хранения файлов, получаемых при выполнении практической работы. Эта папка должна располагаться в папке **Базы данных\Группа\Студент** и соответствовать номеру выполняемой практической работы.
2. На основании индивидуального задания выбрать имя файла создаваемой базы данных. Для имени лучше всего выбрать одно или несколько английских слов, соответствующих наименованию предметной области. Использование для имени русских слов, записанных латинскими буквами, недопускается.
3. Открыть приложение "Среда SQL Server Management Studio". Для этого можно либо воспользоваться меню Пуск (**Пуск/Программы/Microsoft SQL Server 2008 /Среда SQL Server Management Studio**).
4. Создать соединение с локальным или удаленным сервером.
5. Создать базу данных для своей предметной области с помощью диалога, выбрав сервер "pi_srv" или локальный сервер "**Имя_компьютера\SQLEXPRESS**".
6. Создать базу данных и указать в качестве имени файла "**Базы данных\Группа\ФИО_студента\Название_БД**".
7. Извлечь метаданные для автоматической генерации команды создания базы данных.
8. Удалить базу данных, выполнив команду "**Database/Drop Database**" (База данных/Удалить базу данных).
9. Создать базу данных вторым способом, выполнив в окне "Script Executive" операторы, полученные при извлечении метаданных перед предыдущим удалением.
10. Создать резервную копию базы данных.
11. Удалить базу данных.
12. Восстановить базу данных из резервной копии.
13. Сохранить файл сценария на сервере в папке "Студент", дав ему имя «лаб.№1» и стандартное расширение "***.sql**".

Ход работы

Создание соединения с сервером

Выполните следующие инструкции:

Работа с приложением **SQL Server Management Studio** начинается с создания соединения с установленным сервером. Убедитесь вначале, что сервер Microsoft SQL Server (2008) на локальной машине или на сервере компьютерного класса установлен и работает.

Откройте приложение "SQL Server Management Studio". Для этого можно либо воспользоваться меню **Пуск(Пуск/Программы/ Microsoft SQL Server 2008/ Среда SQL Server Management Studio)**.

В диалогом окне **Соединение с сервером** подтвердите заданные по умолчанию параметры и нажмите кнопку **Соединить**, см. рис. 2.

Для соединения необходимо, чтобы поле **Имя сервера** содержало имя компьютера, на котором установлен SQL Server.

Если компонент **Database Engine** является именованным экземпляром, то поле **Имя сервера** должно

также содержать имя экземпляра в формате
<имя_компьютера>\<имя_экземпляра>.

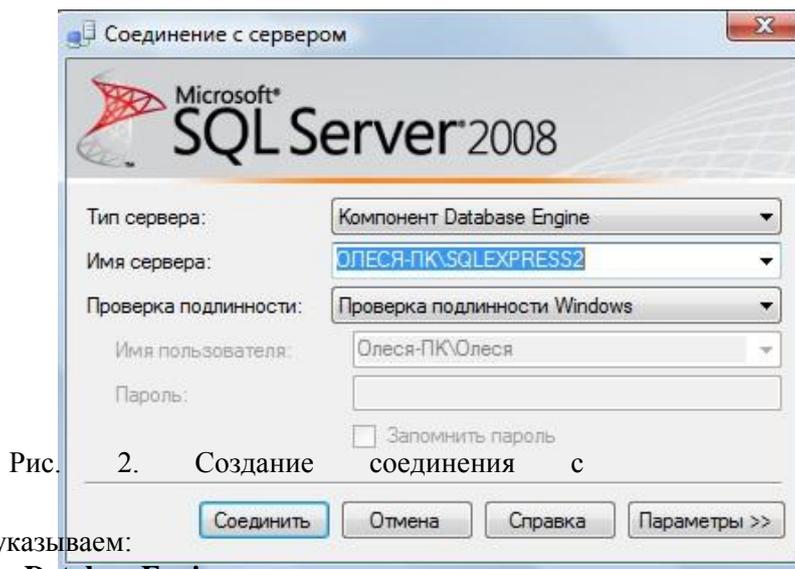


Рис. 2. Создание соединения с

сервером. В параметрах указываем:

Тип сервера—Компонент Database Engine.

Имя сервера. Подключение может быть *локальным* или *удаленным*. Представляет собой название компьютера в сети, на котором установлен сервер СУБД. Если сервер установлен на том же компьютере, где сейчас работает пользователь, то в качестве имени используется имя компьютера и идентификатор сервера;

проверка подлинности—Windows (по умолчанию),

имя пользователя—

имя пользователя по умолчанию, зарегистрированного на сервере MSSQL Server (задается при установке сервера),

пароль—пусто или пароль для пользователя, заданного для сервера MSSQL Server;

Нажмите кнопку **Соединить**. Если соединение будет совершенно успешно, то на экране появятся данные сервера.

Среда Management Studio представляет данные в виде окон, выделенных для отдельных типов данных. Сведения о базе данных отображаются в обзорном объекте в окнах документов.

Обозреватель объектов является представлением в виде дерева, в котором отображаются все объекты базы данных на сервере. Он может содержать базы данных компонента SQL Server Database Engine, служб Analysis Services, служб Reporting Services, служб Integration Services и SQL Server Compact 3.5 с пакетом обновления 1 (SP1).

Обозреватель объектов включает сведения по всем серверам, к которым он подключен. При открытии среды Management Studio пользователю предлагается применить при подключении обозревателя объектов параметры, которые использовались в прошлый раз. Чтобы подключиться к любому из серверов, следует дважды щелкнуть его в компоненте «**Зарегистрированные серверы**», однако регистрировать его не обязательно, см. рис. 1.

Окно документов представляет собой наиболее крупную часть среды Management Studio. В окнах документов могут размещаться редакторы запросов и окна обзора. По умолчанию отображается страница «**Сводка**», подключенная к экземпляру компонента Database Engine на текущем компьютере.

Общие сведения о базах данных MSSQL Server

Кроме четырех системных баз, SQL Server может обрабатывать до **32734** баз данных, определяемых пользователем.

База данных представляет собой:

- набор взаимосвязанных таблиц;
- связанный набор страниц, выделенных для хранения данных MSSQL Server;
- совокупность данных архивации;
- два или более файла;
- важную совокупность данных для целей защиты и управления.

Файлывазыданных

Базаданныхсостоитиздвухиболеефайлов,каждыйизкоторыхможетиспользоватьсялишьодной базой.

Уфайловсуществуютдваимени:**логическоеифизическое**.**Логическое**имяподчиняется стандартным правилам выбора имен объектов SQL Server. **Физическое имя**представляет собой полное имя любого локального или сетевого файла. Максимальноечислофайловвбазеданных— 32768.**Файлыделятсянатри типа:**

- **Первичныефайлы**.Используютсядляхраненияданныхиинформации,определяющихначальные действиябазой.Базаданныхсодержитлишьодинпервичныйфайл.Стандартноерасширение—**.mdf**.
- **Вторичные файлы**. Одна или несколько вспомогательных областей для храненияданных.Могутиспользоватьсядляраспределенияоперацийчтения/записипонесколькимдискам.С тандартноерасширение—**.ndf**.
- **Файлыжурналов**.Содержатжурналытранзакцийбазыданных.Базаданныхсодержит по крайней мере один файл журнала. Стандартное расширение — **.ldf**. Переднепосредственнойзаписьютранзакцийвфайлданныхвсевносимыеизменениязаписываютсявжурнал.

Группыфайлов

Группы файлов предназначены для объединения нескольких файлов. Каждый файлможет входить не более чем в одну группу. Файлы журналов не могут принадлежатьникакимгруппам.Группыфайловиспользуютсядляраспределенияоперацийчтения/записип онесколькимдискам.Еслигруппасодержитболееодногофайла,операции записи распределяются между файлами группы. Базы данных могут содержатьдо32768групп файлов.

Укаждойбазыданныхимеется**первичнаягруппафайлов**.Онасодержитпервичный файл данных и все файлы, которые не были явно назначены в другую группуфайлов.Имяпервичнойгруппы файлов— **PRIMARY**.

Созданиеирегистрациябазыданных

Длясозданиябазыданныхможноиспользоватьодиниздвухспособов:

Первый способ создания БД. Выполнить команду "База данных/Создать базуданных..." в программе SQL ServerManagementStudio,ввестипараметрысоздаваемойбазыданныхвдиалоговомокне "Создание базы данных" (рис. 3) и нажать кнопку[OK].

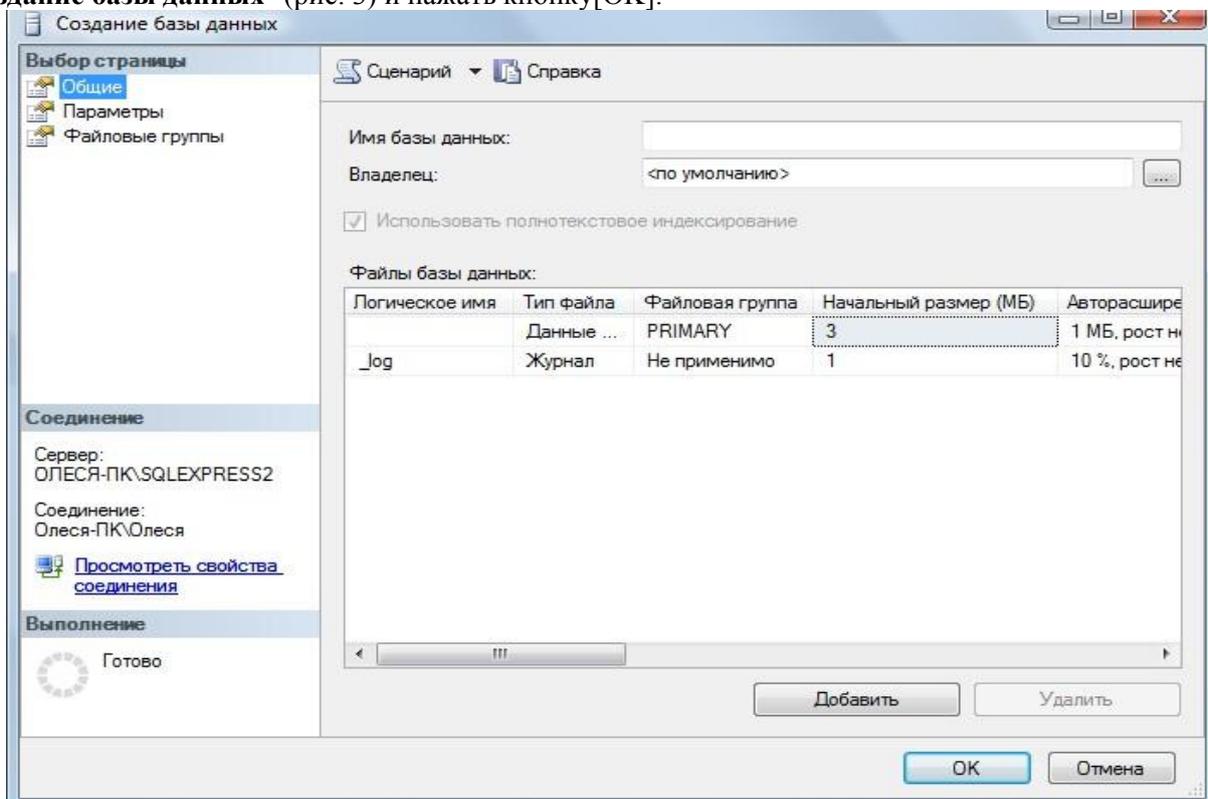


Рис.3.Диалоговое окносозданиябазыданных

Вполе**Имябазыданных**введитеимянашейбудущейбазыданных,например— **University**.
Поле**Владелец**-заданпоумолчанию,взависимостиотнастройкисервера.

Files\MicrosoftSQLServer\MSSQL10.SQLEXPRESS2\MSSQL\DATA\.

Прежде чем нажать кнопку **Добавить**, просмотрите **Параметры** и **Файловые группы** для создаваемой базы данных.

После нажатия на кнопку **[OK]** программа "SQL Server Management Studio" создаст базу данных, имя которой вы увидите в обозревателе объектов, а также сгенерирует необходимый SQL-код для создания базы данных с теми свойствами, которые указаны в этом диалоговом окне и передаст его серверу СУБД для выполнения.

Пример этих операторов приведен на рис. 4. (нажмите на имени базы данных **University** правой клавишей и из контекстного меню выберите **Создать скрипт как..CREATE**). Если параметры введены правильно, база данных будет создана.

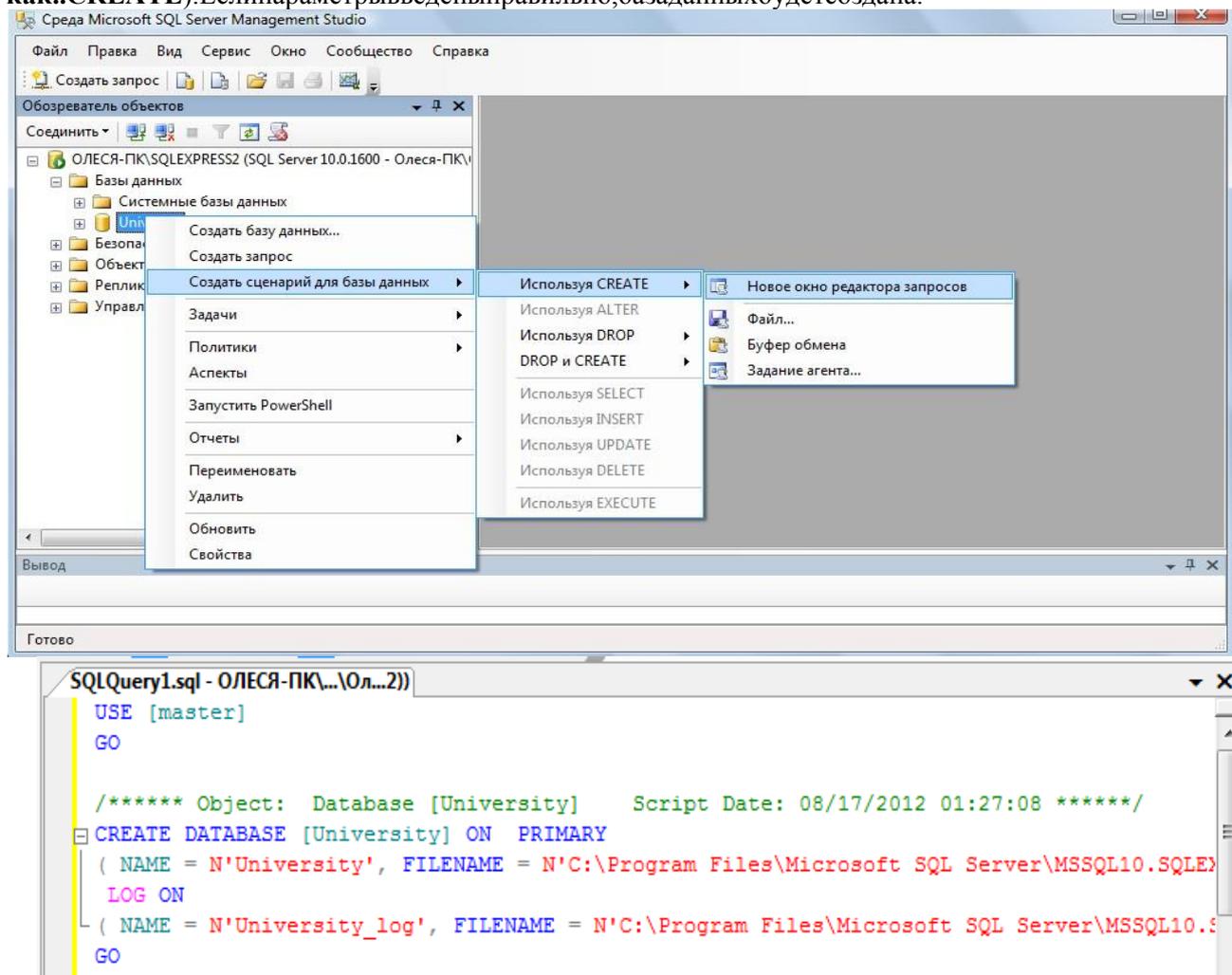


Рис.4. Сгенерированный sql-код созданной базы данных

Содержащиеся в сценарии операторы отделяются друг от друга символом ";". Сценарий может содержать поясняющие комментарии двух видов: многострочный комментарий (начинается символами "/*" и заканчивается символами "*/") и однострочный комментарий, который начинается символами "--" и продолжается до конца строки.

При создании базы данных возможны следующие типичные ошибки:

1. На целевом компьютере не запущены или не установлены сервер СУБД – т.е. выполнять команду создания базы данных просто некому.
2. На целевом компьютере нет каталога, в котором предполагается создать базу данных.
3. Файл, в котором должна быть найдена база данных на сервере, уже существует.

После создания базы данных вся введенная информация запоминается программой SQL Server Management Studio и в окне редактора в дереве навкладки "Проводник" добавляется узел зарегистрированной базы данных (рис.5).

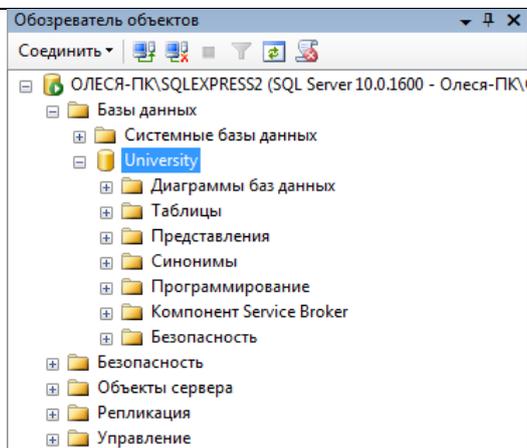


Рис.5.ПереченьзарегистрированныхбазданныхвSQLServerManagementStudio

Второй способ создания БД. Выполнить в программе SQL Server Management Studio команду "Создать запрос"  Создать запрос на панели инструментов, затем ввести команду, создающую базу данных в окне "Script Execute" (рис.3) и нажать кнопку  Выполнить.

Команда **CREATEDATABASE**-Создание базы данных MSSQL Server

Базы данных создаются командой **CREATEDATABASE**. Создание баз данных разрешается любому пользователю с ролью системного администратора или всем, кому системный администратор предоставил такое право. Команда **CREATEDATABASE** имеет следующий синтаксис:

```

01. CREATE DATABASE имя_базы
02. [ ON [PRIMARY] ([ NAME = логическое_имя_файла, ]
03. FILENAME = 'имя_файла_ОС'
04. [, SIZE = размер]
05. [, MAXSIZE = { максимальный_размер | UNLIMITED } ]
06. [, FILEGROWTH = приращение] )
07. | {FILEGROUP имя_группы_файлов FILEDEFINITIONS}
08. [,...n] ]
09. [LOG ON {[ NAME = логическое_имя_файла, ]
10. [FILENAME = 'имя_файла_ОС'
11. [, SIZE = размер]
12. [, MAXSIZE = { максимальный_размер | UNLIMITED } ]
13. [, FILEGROWTH = приращение] } [,...n]
14. [FOR LOAD | FOR ATTACH]

```

Если при создании базы не указан первичный файл данных и/или файл журнала, то отсутствующий файл (или файлы) создается с именем по умолчанию.

Физические файлы будут находиться в стандартном каталоге.

Первичному файлу присваивается имя **имя_базы.mdf**, а файлу журнала — **имя_базы_log.ldf**.

Если размер файлов не задан, то при создании размер первичного файла совпадает с размером первичного устройства базы **model**, а размер файла журнала и вторичных файлов данных равен 1 Мбайт. Он может быть и больше, если размер первичного файла базы данных **model** превышает 1 Мбайт. Хотя имена и размеры файлов указывать необязательно, на практике это всегда следует делать. SQL Server создает базу данных в два этапа. На первом этапе база **model** копируется в новую базу данных, а на втором этапе инициализируется ее неиспользуемое пространство.

Команда **CREATEDATABASE** имеет следующие параметры:

- **PRIMARY**—файл определяется как первичное устройство.
- **NAME**—логическое имя; по умолчанию совпадает с именем файла.
- **FILENAME**—полное имя файла на диске.
- **SIZE**—исходный размер файла. Минимальный размер файла журнала равен 512 Кбайт.
- **MAXSIZE** — максимальный размер файла.
- **UNLIMITED** — размер файла не ограничивается.
- **FILEGROWTH**—

приращение размера в мегабайтах (MB), килобайтах (KB) или процентах (%). По умолчанию

приращениеравно10%.

- **FORLOAD**—

обеспечиваетобратнуюсовместимостьсосценариямиSQL,написаннымидляпредыдущихверсийSQL Server.

- **FORATTACH**—указывает,чтофайлыбазыданныхуже существуют.

Пользователь,создавшийбазуданных,являетсяеевладельцем.Все параметры конфигурации базы копируются из базы model, если только при создании базы не был указан параметр **FOR ATTACH**. В этом случае параметры конфигурации читаются из существующейбазыданных.Рассмотримнекоторыепримерыкоманды**CREATEDATABASE**:

/*Базаданныхсостандартнымразмероми именами файлов*/

```
01. CREATE DATABASE test1
02. /* Данные – 2 Мбайт, файл журнала – по умолчанию */
03. CREATE DATABASE test2
04. ON (FILENAME = 'c:\d1.mdf', SIZE = 2, NAME = 'd1')
05. /* Первичный файл – 10 Мбайт, одна группа файлов
06. g1 и журнал размером 10 Мбайт */
07. CREATE DATABASE test3
08. ON PRIMARY (FILENAME = 'c:\test3.mdf',
09. SIZE = 10 , NAME = 'd1'),
10. FILEGROUP g1 (FILENAME = 'c:\g1.mdf',
11. SIZE = 10 , NAME = 'g1')
12. LOG ON (FILENAME = 'c:\test3.ldf',
13. SIZE = 10, NAME = 'log1')
```

Задача 1. Создайте sql-скрипт создания новой базы данных под именем **Educator** на "D:\Базыданных\Группа\ФИО_студента\Название_БД.mdf", с первичным устройством, с исходным размером файла в 10 Мбайти запустите на выполнение скрипт (кнопка  на панели инструментов). Выполните в окне обозревателя объектов **Обновление**. Сохраните созданный скрипт в текущую папку под именем **1.sql**.

После успешного выполнения и обновления проводника у вас должна появиться новая база данных.

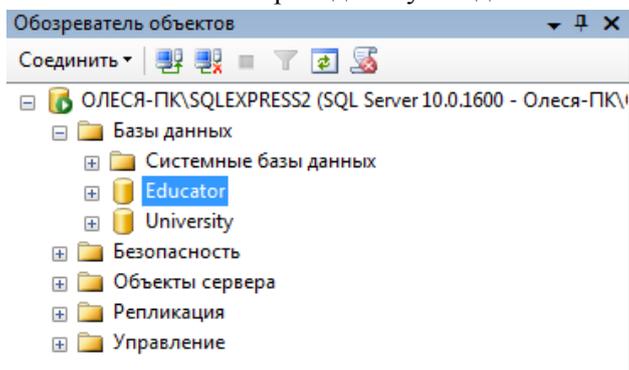


Рис. 6. Окно проводника после выполнения сценария создания базы данных

Подключение к базе данных

Чтобы подключиться к зарегистрированной базе данных, надо выбрать нужную базу данных в списке (рис. 5) и сделать двойной щелчок мышкой на выбранной базе данных.

Если все параметры подключения были введены правильно, то произойдет подключение к базе данных, название подключенной базы данных в окне "Обозревателя объектов" будет выделено жирным шрифтом, а также появятся вложенные узлы объектами, содержащимися в подключенной базе данных (рис. 7).

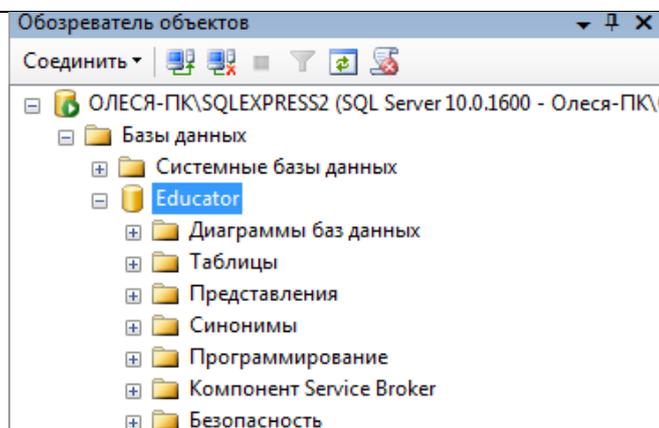


Рис.7. Зарегистрированные базы данных в SQL Server Management Studio

После подключения баз данных можно просматривать имеющиеся объекты, создавать новые, вносить и просматривать данные, а также проводить операции с имеющимися объектами.

После создания БД в окне **Обозревателя объектов** (его можно вызвать по <F8>) выбираем **Data Bases (Базы данных)** и открывается список БД, в котором откроем созданную БД (если она не появилась, то в окне **Object Explorer** нажать <F5> для обновления списков), которая состоит из восьми вложенных разделов (некоторые содержат еще дополнительные разделы), соответствующих объектам СУБД SQL Server:

Database Diagrams (Диаграммы БД)	Views (Представления)	Programmability (Объекты программирования)
Tables (Таблицы)	Synonyms (Синонимы)	Security (Безопасность)
Service Broker	Storage	

На начальном этапе раздел созданной БД пуст, за исключением некоторых объектов, которые создаются по умолчанию, например в разделе **Security/Users** создаются пользователи, которые имеют права доступа к объектам БД, их можно изменить.

Удаление базы данных

Для удаления базы данных можно использовать один из трех способов:

1. Выполнить в программе "SQL Server Enterprise Manager" команду контекстного меню "Удалить", выбрав перед этим в списке базу данных, а затем подтвердить свое желание в диалоговом окне.

2. Выполнить оператор **DROP DATABASE** в SQL-редакторе.

3. Удалить файл с базой данных. Синтаксис оператора **DROP DATABASE**:

DROP DATABASE database_name;

Резервное копирование и восстановление

Резервное копирование (backup) базы данных и **восстановление** из резервной копии (**restore**) – два важнейших и наиболее частых процесса, осуществляемых администраторами баз данных.

Резервное копирование базы данных – единственный надежный способ предотвратить потерю данных в результате поломки диска, сбоя электропитания, действий злоумышленников и ошибок в программах. В процессе резервного копирования создается независимый от платформы "снимок" базы данных, с помощью которого можно перенести данные на другую операционную систему или даже другую платформу. **Полный цикл:** резервное копирование и восстановление из резервной копии приводит к корректировке статистической информации, является средством от излишнего "разбухания" базы данных и необходимой операцией обслуживания базы данных. Кроме того, миграция от одной версии сервера к другой также происходит при помощи процесса **backup/restore**.

Для создания резервной копии базы данных с помощью программы "SQL ServerManagementStudio" необходимо подключиться к базе данных, выбрать из контекстного меню базы данных **Задачи/Создать резервную копию**. В открывшемся диалоговом окне **"Мастер резервного копирования"** задать несколько параметров и нажать кнопку **[Выполнить]**, см. рис. 8.

После выбора пути файла для резервной копии в окне **BackUp Database** нажать на **ОК** запускаем процесс создания резервной копии. В случае успешной работы появится сообщение.

В результате будет создан файл с резервной копией. Стандартным расширением таких файлов для "SQL ServerManagementStudio" является **"*.bak"**. Файл с резервной копией базы данных обычно на порядок меньше оригинала.

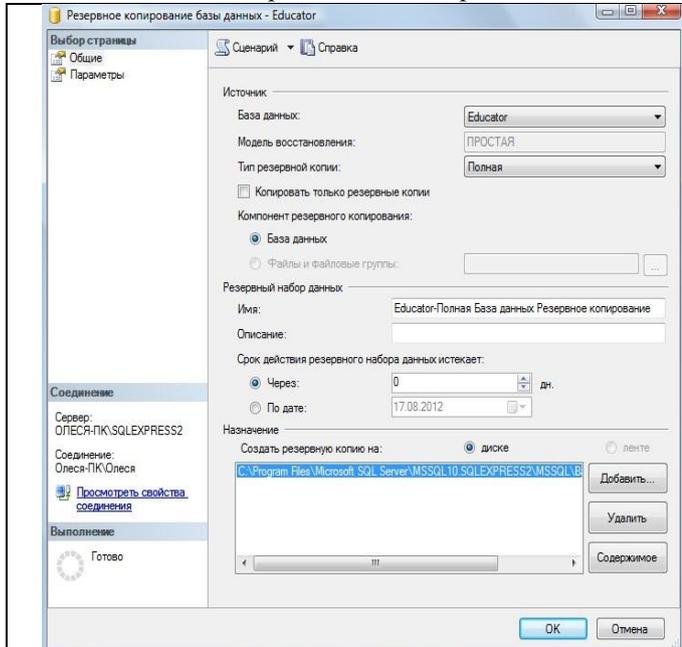


Рис. 8. Создание резервной копии базы данных

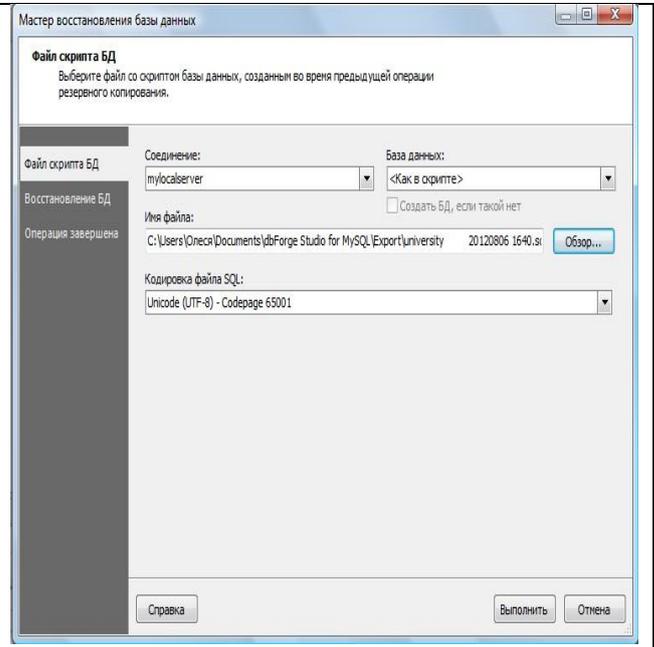


Рис. 9. Восстановление базы данных

Для восстановления базы данных из резервной копии используется команда **"База данных/ Восстановление базы данных"**. В результате откроется диалоговое окно **"Мастер восстановления баз данных"**, в котором надо выбрать имя БД куда будет восстанавливаться база данных, в которую будет помещен результат, способ восстановления, файл, из которого будет восстанавливаться база данных, отметить выбранную резервную копию, и нажать кнопку **[Восстановить]**, см. рис. 9. Запускаем процесс восстановления. В случае успешного выполнения получим сообщение.

Резервное копирование и восстановление базы данных, наряду с процессом извлечения метаданных и последующего выполнения полученного

сценария, можно использовать при переносе разрабатываемой базы данных между различными компьютерами для обеспечения самостоятельной работы студентов над практическими работами и курсовым проектом.

Самостоятельно Выполните вначале резервное копирование, а затем восстановление базы данных. Удалите базу данных **Educator** с помощью скрипта сохраните sql-запрос.

Копирование и перенос на другой сервер БД

Для просмотра, запуска, остановки служб MS SQL Server необходимо запустить утилиту **SQL Server Configuration Manager** (рис. 10).

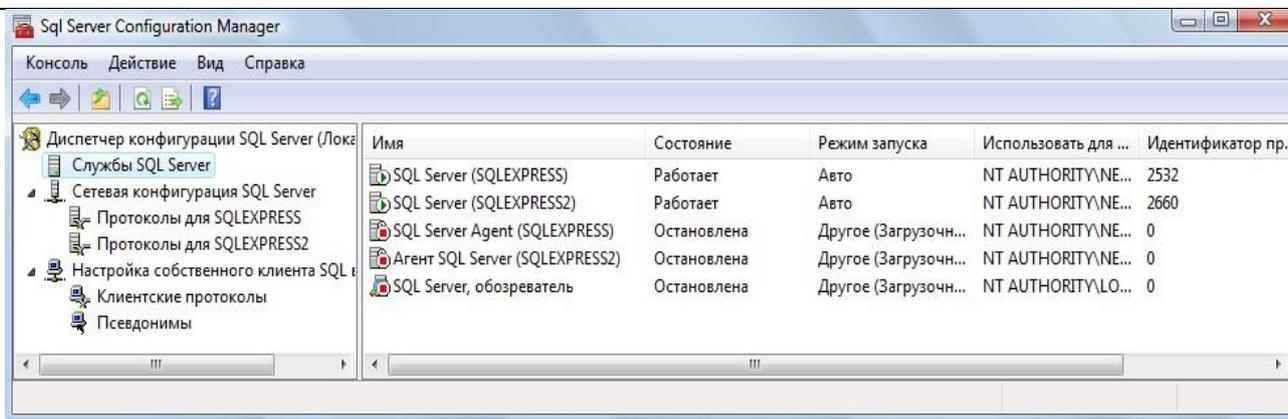


Рис. 10. Список служб сервера БД

Для того чтобы скопировать БД необходимо остановить службу SQL Server (в ее контекстном меню выбрать **Stop**). Далее в подпапке... \MSSQL.1\MSSQL\Data\ скопировать файлы с вашим названием БД (по умолчанию их два). Не забудьте потом снова запустить службу SQL Server (в ее контекстном меню выбрать **Start**).

Для того чтобы подключить скопированную БД на другом сервере, нужно предварительно скопировать ваши файлы в папку... \MSSQL.1\MSSQL\Data\ соответствующего сервера. Далее запустить утилиту SQL Server Management Studio. В появившемся окне с названием Object Explorer Проводник объектов (его можно вызвать по <F8>) выбираем Data Bases (Базы данных) и по <правой кнопке мыши> в контекстном меню (рис. 5) выбираем **Attach...** (**Присоединить...**). В появившемся окне Attach Data Bases (Присоединение базы данных) нажать <Add> и выбрать ваш файл БД с расширением **.mdf**.

Системные базы данных

Системные базы данных сервера, создаваемые при установке, их файлы представлены в таблице 1.

Название	Назначение	Размещение
Master	Хранит всю информацию сервера, включая учетные записи и параметры, сведения о всех базах и нахождении их первичных файлов с данными об инициализации баз данных пользователя.	Master.mdb – файл данных (75 mb) Mastlog.ldf – журнал транзакций (1 mb)
TempDB	Хранит все временные системные и пользовательские объекты: таблицы, переменные, хранимые процедуры, курсоры и т.п.	Tempdb.mdf – файл данных (8 mb) Templog.ldf – журнал транзакций (0.5 mb)
Model	Является шаблоном, задаваемых администратором и используемым для создания любых пользовательских баз данных. Содержит параметры по умолчанию, которые можно переопределять при создании соответствующей базы данных пользователя.	Model.mdf – файл данных (0.75 mb) Model.ldf – журнал транзакций (0.75 mb)
MSDB	Хранит информацию, относящуюся к автоматизации администрирования и управления сервером.	Msdbdata – файл данных (3.5 mb) Msdblog – журнал транзакций (0.75 mb)
		Всего – 22.75 mb

Все системные и пользовательские базы данных содержат в обязательном порядке 18 системных таблиц, которые хранят информацию, определяющую структуру и организацию соответствующей базы данных.

MSSQL Server поддерживает два основных класса приложений клиентского типа

:

1. приложения реляционных баз данных, использующие команды Transact-SQL

расширениями ODBC и набор стандартных функций и объектно-ориентированных методов;
 2. web-приложения, использующие команды Transact-SQL и запросы на языке XPath и документы XML.
 Оба класса приложений используют API интерфейса баз данных типа OLEDB или ODBC.

Основные принципы управления учетными записями и ролями в MSSQL Server
Список системных процедур команд, которые позволяют реализовать политику делегирования прав между пользователями БД.

Название встроенной процедуры	Описание
sp_grantlogin	– позволяет использовать пользователей или группы ОС для соединения с Microsoft SQL Server™, используя Windows Authentication . Этот пример позволяет пользователю Windows NT Corporate\BobJ соединиться с SQL Server. Например, EXEC sp_grantlogin 'Corporate\BobJ'
sp_defaultdb	Изменяет для пользователя БД по умолчанию. Этот пример устанавливает БД по умолчанию pubs для пользователя Victoria. Например, EXEC sp_defaultdb 'Victoria', 'pubs'
sp_grantdbaccess	Добавляет учетную запись из раздела security в текущую БД для учетных записей Microsoft Windows также дает разрешение на доступ к текущей БД. Синтаксис: EXEC sp_grantdbaccess [@loginname =] 'login' [, [@name_in_db =] 'name_in_db'] [OUTPUT] Этот пример добавляет учетную запись Corporate\GeorgeW в текущую БД и присваивает псевдоним внутри БД Georgie. Например, EXEC sp_grantdbaccess 'Corporate\GeorgeW', 'Georgie'
sp_revokedbaccess	Удаляет информацию об учетной записи из текущей БД. Синтаксис: EXEC sp_revokedbaccess [@name_in_db =] 'name' Этот пример удаляет учетную запись Corporate\GeorgeW из текущей БД. EXEC sp_revokedbaccess 'Corporate\GeorgeW'
sp_addrole	Создает новую роль в текущей БД. Этот пример создает новую роль в текущей БД с названием Managers. EXEC sp_addrole 'Managers'

sp_addrolemember	<p>ВтекущейБДназначаетрольконкретномупользователю.</p> <p>ПримерА. Этот пример добавляет учетную запись Corporate\JeffL изWindowsNT в БД Sales как пользователя Jeff. Jeff затемполучает рольSales_ManagersвБД Sales.</p> <p>USESales--сделатьтекущейБД Sales GO--выполнитькоманду,апотомзапуститьследующую EXECsp_grantdbaccess'Corporate\JeffL','Jeff'GO EXECsp_addrolemember'Sales_Managers','Jeff'</p> <p>ПримерВ. Этот пример добавляет пользователя SQLServer с именемMichaelкролиEngineeringвтекущейБД. EXECsp_addrolemember'Engineering','Michael'</p>
sp_helpprotect	Показываетсписокпривилегий,ассоциированныхсролью.
sp_helprolemember	ПоказываетсписокпользователейБД,входящихвуказанную роль
sp_addsrvrolemember	Присвоениевстроеннойсервернойролидлясуществующейучетнойзаписи sp_addsrvrolemember[@loginname=']login',[@rolename=
]role' Например: sp_addsrvrolemember'Admin_DB','sysadmin'
sp_dropsrvrolemember	Удалениевстроеннойсервернойроли дляучетнойзаписиилигруппы sp_dropsrvrolemember[@loginname=']login',[@rolename=]role' Например: sp_dropsrvrolemember'Admin_DB','sysadmin'
sp_helpsrvrole	ОписаниетольковстроенныхролейвSQLServer sp_helpsrvrol e[[@srvrolename =] 'role'] Например: sp_helpsrvrole'sysadmin'
sp_helpsrvrolemember	Возвращаетсписокролейиучетныхзаписей,которымприсвоены этироли sp_helpsrvrolemember[[@srvrolename=']role'] Например: sp_helpsrvrolemember'sysadmin'
sp_srvrolepermission	Возвращаетсписокролейиразрешений,которыеприсвоеныэтим ролям sp_srvrolepermission[[@srvrolename=']role'] Например: sp_srvrolepermission'sysadmin'

<p>sp_addlogin</p> <p>sp_adduser</p>	<p>СозданиеновойучетнойзаписивSQLServerвразделеSecurity:</p> <pre> sp_addlogin @loginname=]'login',[@passwd=]'pa ssword' [,[@defdb=]'database' [, [@deflanguage =] 'language'],[@sid=]sid] [,[@encryptopt=]'encryption_option'] </pre> <p>Например:</p> <pre> sp_addlogin'login1',sysname,'DB_Books' </pre> <p>Создает пользователя в SQLServer без PUBLIC в БД 'DB_Books'.</p> <p>Нужноещеиспользовать</p> <pre> sp_adduser [@loginname =] 'login',[@name_in_db =]'user' [,[@grpname=]'group'], </pre> <p>Пример: Создана база данных DB_Books. В ней создан пользовательAdmin_DBссервернойрольюsysadmin, с рольювБДdb_owner.</p> <p>СоздатьвQueryAnalyzerновогопользователясименемPublic_и паролем Public_1 (пароль не должен совпадать с именемпользователя)спомощьюследующихкоманд(незабудьтенажатьF5для запускакоманднавыполнение):</p> <pre> EXECsp_addlogin'Public_','Public_1','DB_Books'useDB_ Books EXECsp_adduser'Public_','Public_' </pre> <p>ВБДDB_BooksсозданпользовательPublic_срольювБДDB_Bookspublic.</p>
<p>Deny(отрицание)</p>	<p>Этотпримерзапрещаетнесколькосистемныхпривилегийдлянесколькихпользователей.</p> <p>Пользователинемогутиспользоватьсистемныепривилегии CREATEDATABASEorCREATETABLE,еслиониене наделеныимимчерезкомандуGRANT.Пример:</p> <pre> DENYCREATEDATABASE,CREATE TABLE TOMary,John, [Corporate\BobJ] </pre> <pre> DENYSELECT,INSERT,UPDATE, DELETEON authorsTO Mary,John, Tom </pre>
<p>Grant(предоставлять)</p>	<p>ThisexamplegrantsmultiplestatementpermissionstotheusersMaryandJohn,andthe Corporate\BobJWindowsNTgroup.</p> <pre> GRANTCREATEDATABASE,CREATE TABLE TOMary,John, [Corporate\BobJ] </pre> <p>Назначение разрешения на выборку (SELECT) для ролиPUBLICвтаблицеAuthors:</p> <pre> GRANT SELECTONAuthorsTO public </pre>

Revoke(отменять)

This example revokes multiple statement permissions from multiple users.

```
REVOKE CREATE TABLE, CREATE DEFAULT  
FROM Mary, John
```

This example removes the denied permission from Mary and, through the SELECT permissions applied to the Budget role, allows Mary to use the SELECT statement on the table.

```
REVOKE SELECT ON Budget_Data TO Mary
```

Создание пользователей для доступа к серверу через утилиту Microsoft SQL Server Management Studio
Создадим новую учетную запись для нашей базы данных **University**. Для этого выберите в Обозревателе объектов раздел **Безопасность/Имена входа**. Добавьте новое имя входа – **Proba**, установите опцию **Проверка подлинности SQL Server**, присвойте свой пароль, примените к выбранной базе данных, установите язык по умолчанию – русский.

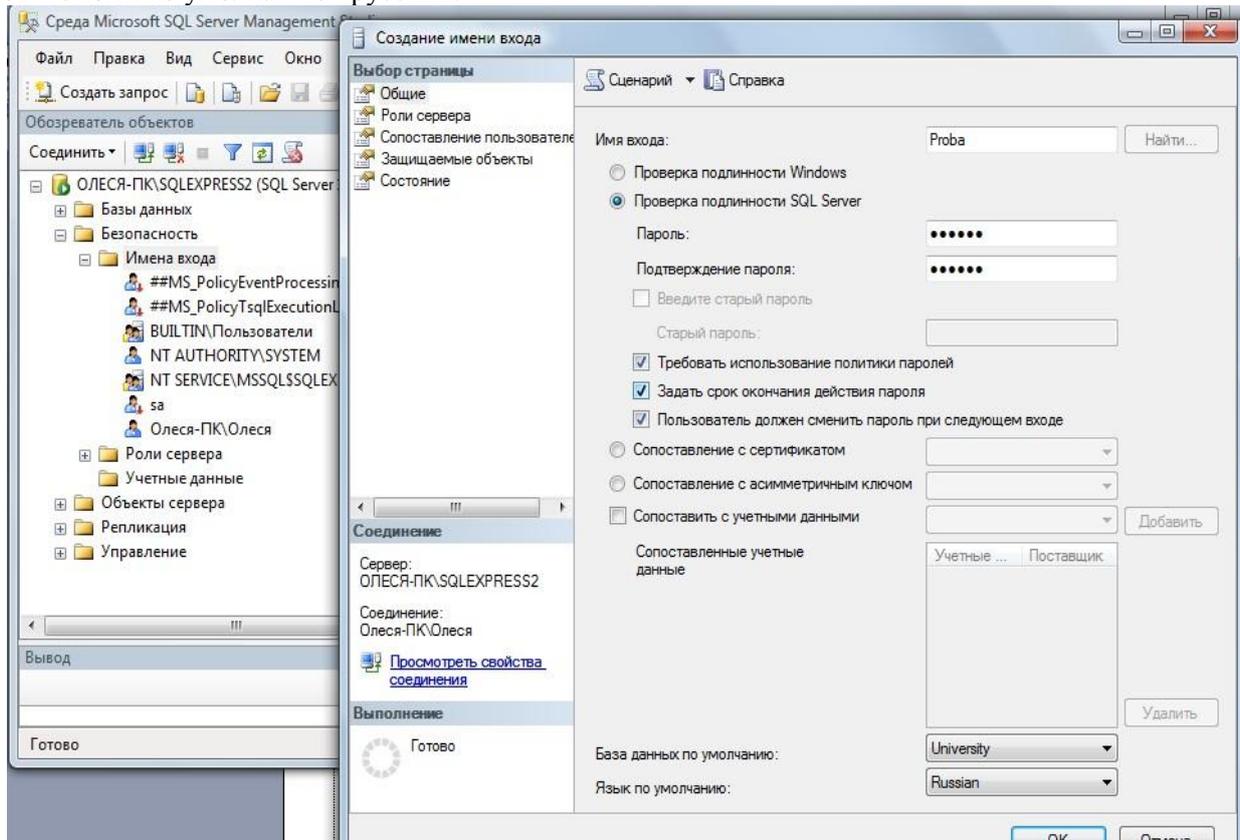


Рис. 2.1. Раздел Безопасность (Security) для работы с пользователями и создание нового пользователя (при SQL Server аутентификации нужно снять галочки с **Enforce password policy**)

Прежде чем добавлять нового пользователя просмотрите его назначенные серверные роли. Для этого в этом же окне выберите раздел **Роли сервера**. Установите для пользователя **Proba** роль **sysadmin**.

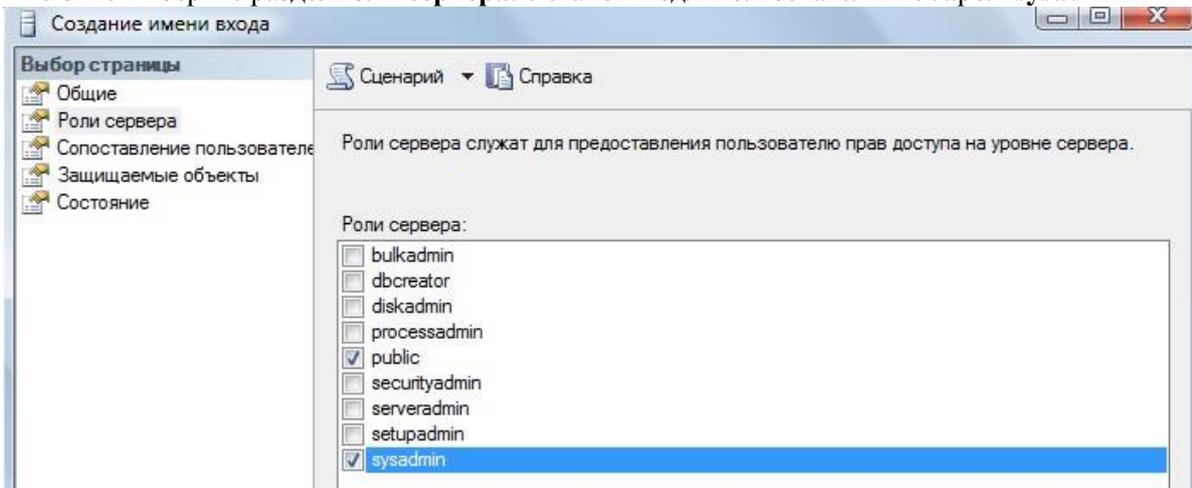


Рис. 2.2. Настройка серверной роли для нового пользователя (весь список

серверных ролей с их привилегиями в концерте работы)
 Далее просмотрите раздел **Сопоставление пользователя**. Установите для базы данных **University** пользователя **Проба** права доступа **Db_owner**, означающие, что пользователь может выполнять любые действия с БД. Ниже перечислены все возможные варианты прав доступа.

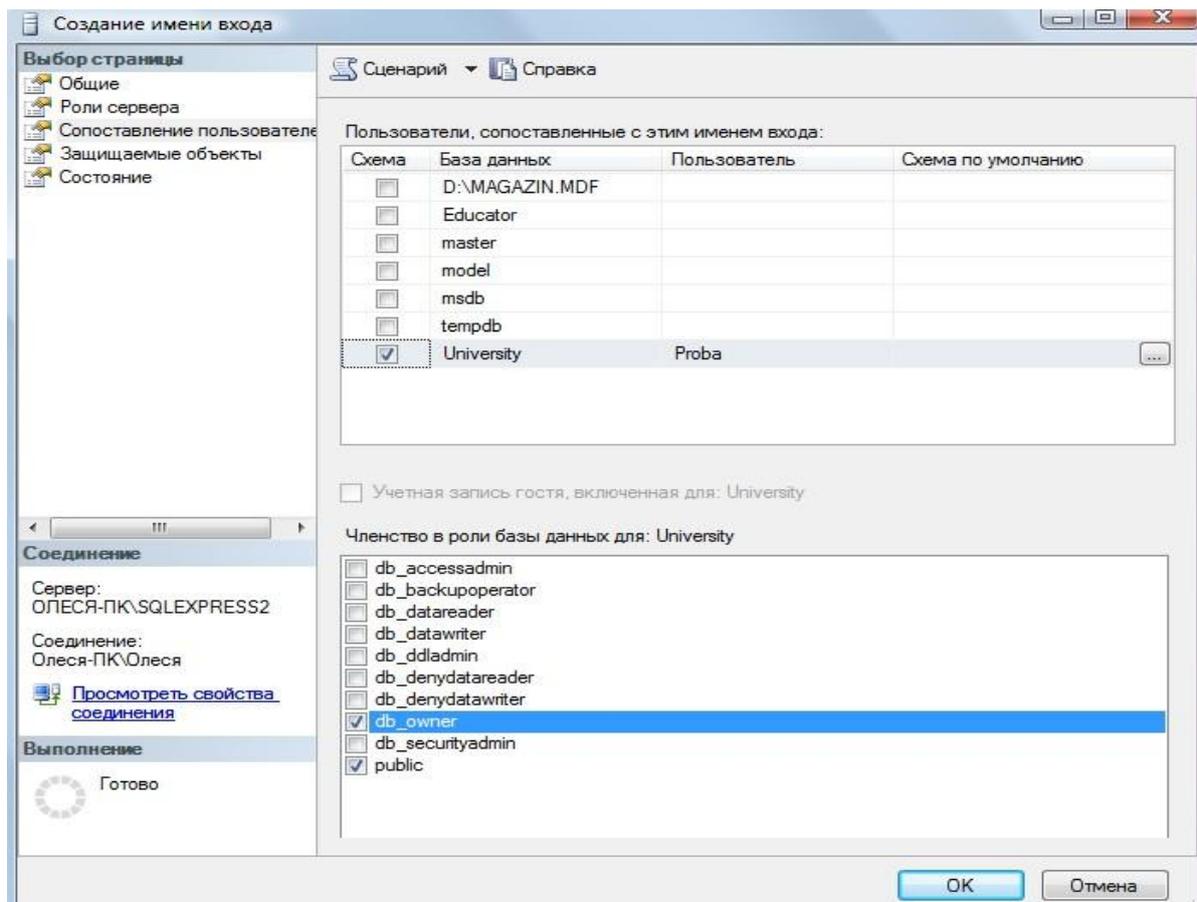


Рис.2.3. Настройка ролей базы данных для нового пользователя (весь список ролей баз данных с их привилегиями ниже)

Перечень ролей БД:

Public – минимальные права доступа к БД (на просмотр) **Db_owner** –

может выполнять любые действия с БД **Db_accessadmin** – добавляет и удаляет пользователей БД

Db_securityadmin – управляет ролями в БД и разрешениями на запуск команд и работу с объектами БД

Db_ddladmin – добавляет, изменяет и удаляет объекты БД **Db_backupoperator** –

осуществляет резервное копирование БД **Db_dataSTUDENT** –

может просматривать все данные в каждой таблице БД

Db_datawriter – может добавлять, удалять и изменять данные в каждой таблице

БД

Db_denydataSTUDENT – запрет на просмотр всех данных в каждой таблице БД

Db_denydatawriter – запрет на добавление, удаление и изменение всех данных в каждой таблице БД

Далее перейдите на раздел **Состояние**. Установите опции **Разрешение** **ключ к ядру СУБД** – предоставить имя входа включить.

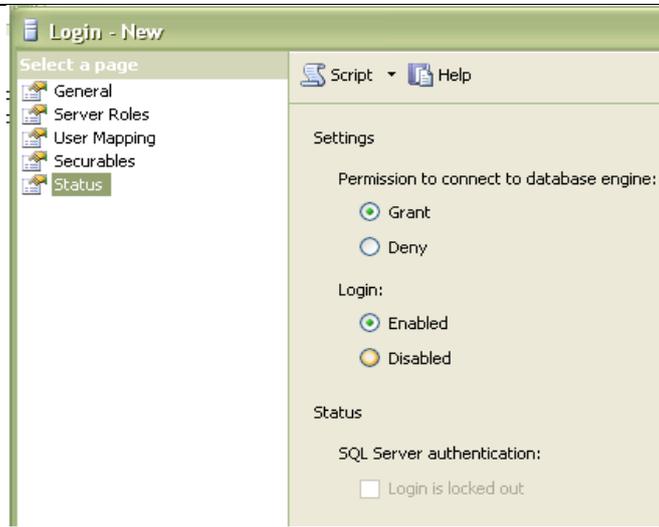


Рис.4.4.Разблокирование создаваемой учетной записи

После нажатия на <ОК> в БД появится пользователь **Proba** с правами собственника БД, который может выполнять все манипуляции с БД **University**.

Откройте окно обозревателя объектов БД **University** и перейдите на вкладку **Безопасность**, там вы найдете только что созданного пользователя.

даниеролейпрограммно

Для упрощения управления правами доступа в системе создаются **роли**, которые затем можно назначать группам пользователей.

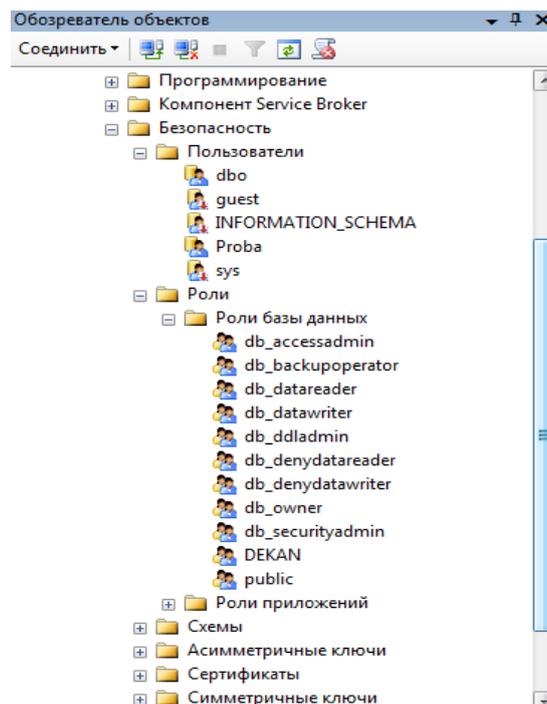
Создадим для нашего примера роли декана (**DEKAN**) и студента (**STUDENT**).

Пример создания роли декана:

**USE University--сделать текущей БД
UniversityEXECsp_addrole 'DEKAN'**

Эти операторы набрать на странице, вызванной нажатием кнопки <Создать запрос>.

Для запуска команд на выполнение нажать **Выполнить**.
Сохраните запрос.



Повторный запуск тех же команд сгенерирует ошибки типа «**В БД уже существует роль DEKAN**».

Чтобы посмотреть, что роль добавлена, откройте вкладку

Безопасность/Роли/Ролибазыданных.

Пример создания роли студента:
USE University--сделать текущей БД
universityEXECsp_addrole'STUDENT'

Декан должен обладать правами на **чтение, удаление, изменение, добавление** в **овсетаблицы** БД **University**, а также должен иметь возможность запускать на исполнение процедуры и функции БД **University**. Поэтому роли декана из системных привилегий назначаем **EXECUTE**, а из привилегий доступа к объектам назначаем **DELETE, INSERT, UPDATE, SELECT**.

Студент должен обладать правами на **чтение** из таблиц. Поэтому роли читателя из привилегий доступа к объектам назначаем **SELECT**.

Оператор представления привилегий

Синтаксис:
GRANT <привилегия>,
...ON<объект>,...
TO<имя>
[WITHgrantoption];

Атрибут **WITHGRANTOPTION** дает право пользователю самому раздавать права, которые он получил.

С помощью оператора **GRANT** для каждого пользователя формируется список привилегий, привилегии управляют работой сервера данных с точки зрения защиты данных. Выполнению каждой транзакции предшествует проверка привилегий пользователя, сеанс которого породил транзакцию.

Например (невыполнять):

```
GRANT select, update (Sales, num) ON Sales_data TO  
user1 WITHGRANTOPTION
```

Пользователь, предоставивший привилегию другому, называется **грантор** (grantor — поставитель). Привилегия является предоставляемой, если право на нее можно предоставить другим пользователям.

PUBLIC — имя роли, которую получает пользователь при добавлении в список пользователей конкретной БД, включает все минимальный набор прав на чтение данных из таблиц и представлений в БД.

Для примера (немного забегаю вперед) создадим таблицу **Discuplinu**. Без объяснения синтаксиса выполните следующий sql-запрос:

```
USE University--сделать текущей БД  
universitycreatetableDiscuplinu (  
    Kod_Discuplinu int NOT NULL primary  
    key, name_Discuplinu char(30)  
    NULL, kol_chasov int NULL  
);
```

Выполните код и обновите вкладку **Таблицы**. Вы должны увидеть созданную таблицу для сохранения данных о всех дисциплинах. Эта таблица пока пустая с тремя столбцами **Kod_Discuplinu, name_Discuplinu, kol_chasov**.

Роль декана названа **DEKAN**. Операторы назначения прав доступа для этой роли представлены ниже:

```
GRANT DELETE, INSERT, UPDATE, SELECT ON Discuplinu TO DEKAN  
GRANT EXECUTE TO DEKAN
```

Роль студента названа **STUDENT**. Операторы назначения прав доступа для этой роли представлены ниже:

```
GRANT SELECT ON Discuplinu TO STUDENT
```

Примените роли декана и студента к созданной таблице.

Создание пользователей с определенной ролью Пример создания декана Ivanov_Dek и присвоения ему роли:

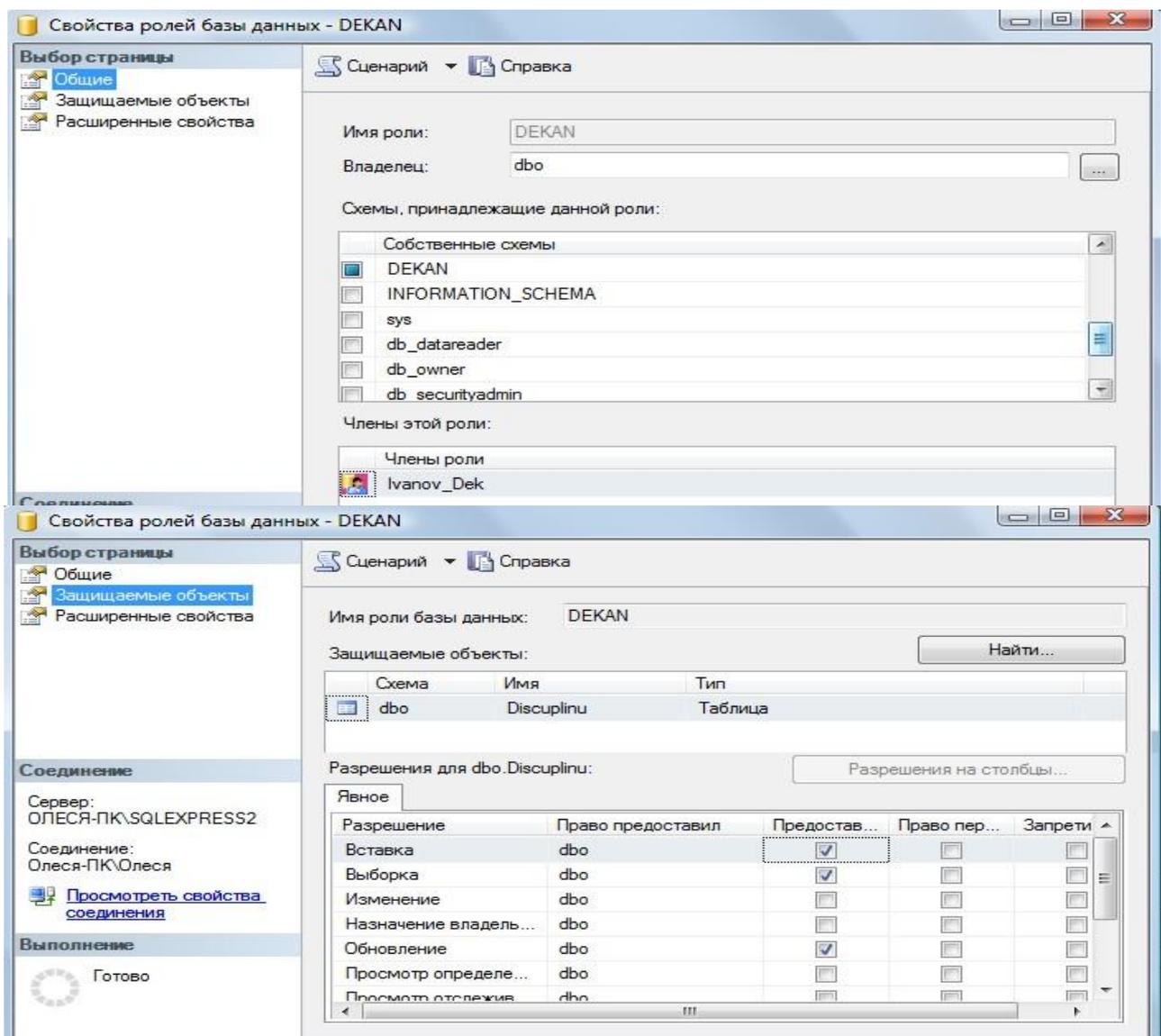
```
EXEC sp_addlogin 'Ivanov_Dek','Ivanov',  
'University' use University  
EXEC sp_adduser  
'Ivanov_Dek','Ivanov_Dek' EXEC sp_addrolemember 'DEK  
AN','Ivanov_Dek'
```

Пример создания студента Petrov_Stud и присвоения роли: EXEC sp_ad

```
dlogin 'Petrov_Stud','Petrov','University'  
use University  
EXEC sp_adduser 'Petrov_Stud','Petrov_Stud'  
EXEC sp_addrolemember 'STUDENT','Petrov_Stud'
```

Выполните команды. Перейдите в окне Обозреватель объектов на **Роли/Роли базы данных/DEKAN** и просмотрите его свойства. Просмотрите назначенные общие свойства, защищаемые объектами и расширенные свойства.

Самостоятельно просмотрите свойства роли базы данных **Student**. Просмотрите назначенные общие свойства, защищаемые объектами и расширенные свойства.



Оператор отмены привилегий

Синтаксис отмены привилегий:
REVOKE[with grant option]

<привилегии>,...O

N<объект>,...

FROM<имя_пользователя>;

Предложение **with grant option** сохраняет за пользователем перечисленные привилегии, но отменяет его право передавать их кому-либо другому.

Пример:

REVOKE SELECT ON Disciplina FROM STUDENT

Выполните команду.

**Оператор изымания роли у
пользователя Revoke<список ролей>from<список пользователей>.**

Пример:

use University

EXEC sp_droprole member 'STUDENT', 'Petrov_Stud'

Выполните команду и просмотрите результат.

2.2. Перечень вопросов для подготовки обучающихся к промежуточной аттестации ЗАДАНИЕ (практическое) к зачету:

1. Предмет и содержание дисциплины, ее связь с другими дисциплинами.
2. Файловая организация данных, ее недостатки. Причины, обусловившие появление баз данных.
3. Понятие базы данных. Преимущества базы данных. Приложения базы данных. Компоненты базы данных.
4. Понятие модели данных. Классификация моделей данных.
5. Трехуровневая модель организации баз данных. Внешний, концептуальный, внутренний уровни.
6. Иерархическая модель. Сетевая модель. Понятие, достоинства и недостатки.
7. Реляционная модель. Базовые понятия модели. Структура данных, виды отношений. Первичный и внешний ключи. Достоинства и недостатки модели.
8. Реляционная целостность (целостность отношений, ссылочная целостность). Индексирование.
9. Постреляционная модель. Ее понятие, достоинства и недостатки. Объектно-ориентированная модель.
10. Объектно-реляционная модель. Ее понятие и отличие от объектно-ориентированной модели. Достоинства и недостатки модели.
11. Многомерная модель. Базовые понятия модели: измерение, ячейка. Поликубическая и гиперкубическая организация данных. Достоинства и недостатки модели.
12. Требования, предъявляемые к базе данных. Этапы жизненного цикла базы данных.
13. Этапы проектирования базы данных и их процедуры.
14. Нормализация таблиц. Назначение нормализации.
15. Модель "сущность-связь". Сущность, атрибут, экземпляр сущности, связь, мощность связи, показатель кардинальности, класс принадлежности сущности.
16. ER-диаграммы. Общие сведения о CASE-средствах для создания ER-диаграмм. Преобразование ER-модели в реляционную модель данных.
17. Категории языков манипулирования реляционными данными.
18. Реляционная алгебра. Реляционные исчисления.
19. Понятие СУБД. Трехуровневая архитектура СУБД. Архитектура типичной СУБД.
20. Классификация и функциональные возможности СУБД. Производительность СУБД. Режимы работы пользователя с СУБД.
21. Общая характеристика СУБД MS Access. Объекты базы данных MS Access.
22. Язык SQL в СУБД. Назначение, стандарты, достоинства.
23. Создание баз данных. Язык DDL
24. Манипуляция данными, язык DML.
25. Запросы на выборку данных. Язык DQL
26. Пользователи БД. Функции администратора БД. Методы защиты БД.
27. Создание резервных копий БД. Сжатие БД. Восстановление БД. Оптимизация работы БД. Установка параметров запуска.
28. Характеристики СУБД PostgreSQL. Архитектурные концепции.
29. Инструментальные средства в СУБД для создания базы данных.
30. Характеристика базы данных и ее приложений, создаваемых в СУБД PostgreSQL.

3. Методические материалы, определяющие процедуру и критерии оценивания сформированности компетенций при проведении промежуточной аттестации

Критерии формирования оценок по ответам на вопросы, выполнению тестовых заданий

- оценка **«отлично»** выставляется обучающемуся, если количество правильных ответов на вопросы составляет 100 – 90% от общего объема заданных вопросов;

- оценка **«хорошо»** выставляется обучающемуся, если количество правильных ответов на вопросы – 89 – 76% от общего объема заданных вопросов;

- оценка **«удовлетворительно»** выставляется обучающемуся, если количество правильных ответов на тестовые вопросы – 75–60 % от общего объема заданных вопросов;

- оценка **«неудовлетворительно»** выставляется обучающемуся, если количество правильных ответов – менее 60% от общего объема заданных вопросов.

Критерии формирования оценок по результатам выполнения заданий

«Отлично/зачтено» – ставится за работу, выполненную полностью без ошибок и недочетов.

«Хорошо/зачтено» – ставится за работу, выполненную полностью, но при наличии в ней не более одной негрубой ошибки и одного недочета, не более трех недочетов.

«Удовлетворительно/зачтено» – ставится за работу, если обучающийся правильно выполнил не менее 2/3 всей работы или допустил не более одной грубой ошибки и двух недочетов, не более одной грубой и одной негрубой ошибки, не более трех негрубых ошибок, одной негрубой ошибки и двух недочетов.

«Неудовлетворительно/не зачтено» – ставится за работу, если число ошибок и недочетов превысило норму для оценки «удовлетворительно» или правильно выполнено менее 2/3 всей работы.

Виды ошибок:

- *грубые ошибки: незнание основных понятий, правил, норм; незнание приемов решения задач; ошибки, показывающие неправильное понимание условия предложенного задания.*

- *негрубые ошибки: неточности формулировок, определений; нерациональный выбор хода решения.*

- *недочеты: нерациональные приемы выполнения задания; отдельные погрешности в формулировке выводов; небрежное выполнение задания.*

Критерии формирования оценок по зачету с оценкой

«Отлично/зачтено» – студент приобрел необходимые умения и навыки, продемонстрировал навык практического применения полученных знаний, не допустил логических и фактических ошибок

«Хорошо/зачтено» – студент приобрел необходимые умения и навыки, продемонстрировал навык практического применения полученных знаний; допустил незначительные ошибки и неточности.

«Удовлетворительно/зачтено» – студент допустил существенные ошибки.

«Неудовлетворительно/не зачтено» – студент демонстрирует фрагментарные знания изучаемого курса; отсутствуют необходимые умения и навыки, допущены грубые ошибки.

Экспертный лист
оценочных материалов для проведения промежуточной аттестации по
дисциплине «Современные СУБД»

Направление подготовки / специальность

09.03.01. «Информатика и вычислительная техника»
(код и наименование)

Направленность (профиль)/специализация

(наименование)

Бакалавр
квалификация выпускника

1. Формальное оценивание			
Показатели	Присутствуют	Отсутствуют	
Наличие обязательных структурных элементов:	+		
– титульный лист	+		
– пояснительная записка	+		
– типовые оценочные материалы	+		
– методические материалы, определяющие процедуру и критерии оценивания	+		
Содержательное оценивание			
Показатели	Соответствует	Соответствует частично	Не соответствует
Соответствие требованиям ФГОС ВО к результатам освоения программы	+		
Соответствие требованиям ОПОП ВО к результатам освоения программы	+		
Ориентация на требования к трудовым функциям ПС (при наличии утвержденного ПС)	+		
Соответствует формируемым компетенциям, индикаторам достижения компетенций	+		

Заключение: ФОС рекомендуется/ не рекомендуется к внедрению; обеспечивает/ не обеспечивает объективность и достоверность результатов при проведении оценивания результатов обучения; критерии и показатели оценивания компетенций, шкалы оценивания обеспечивают/ не обеспечивают проведение всесторонней оценки результатов обучения.