

Министерство образования Белгородской области
Областное государственное автономное
профессиональное образовательное учреждение
«Белгородский индустриальный колледж»

Рассмотрено
Цикловой комиссией
Протокол заседания №1
от «31» августа 2022 г.
Председатель цикловой комиссии
_____ / Третьяк И.Ю.

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ**

МДК 11.01 Технология разработки и защиты баз данных

по специальности

09.02.07 Информационные системы и программирование

квалификация
Программист

Разработчик:
Преподаватель ОГАПОУ «БИК»
Внукова Н.В.

Белгород 2022 г.

Пояснительная записка

МДК 11.01 *Технология разработки и защиты баз данных* входит в состав ПМ.11 *Разработка, администрирование и защита баз данных*, формирует базовые умения для освоения выпускником профессиональных компетенций.

Методические указания по выполнению лабораторных работ составлены в соответствии с рабочей программой профессионального модуля ПМ.11 *Разработка, администрирование и защита баз данных*, МДК 11.01 *Технология разработки и защиты баз данных* для специальности 09.02.07 Информационные системы и программирование (квалификация - Программист) и рассчитаны на 58 часов.

Целью методических указаний по выполнению лабораторных работ является организация и управление работой обучающихся на лабораторных занятиях при изучении МДК 11.01 *Технология разработки и защиты баз данных*.

Методические указания по выполнению лабораторных работ содержат общие положения, тематику лабораторных работ, содержание лабораторных работ и требования к оформлению отчетов.

Методические указания к каждой лабораторной работе включают в себя следующие элементы: название темы, цель занятия, ход работы, теоретическую часть, практическую часть (указания по выполнению) и контрольные вопросы.

Методические указания содержат лабораторные работы, которые обеспечивают формирование базовых умений и навыков владения основными методологиями процессов разработки программного обеспечения, использования методов для получения кода с заданной функциональностью и степенью качества, раскрывают принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программного обеспечения.

В лабораторных работах, приведенных в данных методических рекомендациях, содержатся как задания с подробными указаниями к выполнению, так и задания без алгоритма работы.

Методические рекомендации предназначены для студентов очной формы обучения специальности 09.02.07 *Информационные системы и программирование*.

Методические рекомендации направлены на повышение мотивации учащихся к изучению МДК 11.01 *Технология разработки и защиты баз данных*, развитие гибкого логического и пространственного мышления обучающихся, развитие их профессиональных компетенций.

Тематика лабораторных работ

Номер лабораторной работы	Тема лабораторной работы	Кол-во часов
1-3	Сбор и анализ информации	6
4-6	Проектирование реляционной схемы базы данных в среде СУБД	6
7-9	Приведение базы данных к нормальной форме 3НФ	6
10-11	Создание базы данных в среде разработки	4
12-13	Организация локальной сети. Настройка локальной сети	4
14-15	Установка и настройка SQL-сервера	4
16-17	Экспорт данных базы в документы пользователя	4
18	Импорт данных пользователя в базу данных	2
19	Выполнение настроек для автоматизации обслуживания базы данных	2
20-21	Мониторинг работы сервера	4
22	Выполнение резервного копирования	2
23	Восстановление базы данных из резервной копии	2
24	Реализация доступа пользователей к базе данных	2
25	Мониторинг безопасности работы с базами данных	2
26	Установка приоритетов	2
27-28	Развертывание контроллеров домена	4
29	Мониторинг сетевого трафика	2
ИТОГО:		58

Лабораторная работа № 1-3

Тема: Сбор и анализ информации

Цель работы: изучение принципов разработки и формализации инфологической предметной области с целью построения базы данных.

Теоретическая часть

Важнейшая цель информационной модели заключается в выработке непротиворечивой интерпретации данных и взаимодействий между ними с тем, что необходимо для интеграции, совместного использования и управления целостностью данных. Появление понятий концептуальной схемы данных привело к методологии семантического моделирования данных, т.е. к определению значений данных в контексте их взаимосвязей с другими данными. Сущности. "Сущность" представляет множество реальных или абстрактных предметов (людей, объектов, мест, событий, состояний, идей, пар предметов и т.д.), обладающих общими атрибутами или характеристиками. Отдельный элемент этого множества называется "экземпляром сущности". Каждая сущность может обладать любым количеством отношений с другими сущностями. Сущность является "независимой", если каждый экземпляр сущности может быть однозначно идентифицирован без определения его отношений с другими сущностями.

Сущность называется "зависимой", если однозначная идентификация экземпляра сущности зависит от его отношения к другой сущности. Сущность обладает одним или несколькими атрибутами, которые либо принадлежат сущности, либо наследуются через отношение. Сущность обладает одним или несколькими атрибутами, которые однозначно идентифицируют каждый образец сущности.

Каждая сущность может обладать любым количеством отношений с другими сущностями модели. Если внешний ключ целиком используется в качестве первичного ключа сущности или его части, то сущность является зависимой от идентификатора. И наоборот, если используется только часть внешнего ключа или вообще не используются внешние ключи, то сущность является независимой от идентификатора.

Практическая часть

Требуется составить схему модели

1. Создать функциональную модель предметной области базы данных по варианту индивидуального задания.

2. Создать инфологическую модель предметной области базы данных по варианту индивидуального задания.

2.1 В Microsoft Office Visio 2003/2010 выбрать категорию (Category) FlowChart /Блок-схема/ и далее Template/другие шаблоны/ – IDEF1X Diagram /Схема модели базы данных/.

Вариант 1. Деятельность фирмы торговли ценными бумагами. Создана глобальная информационная система, автоматизирующая процесс учета сделок

купли-продажи ценных бумаг. Информационная система обеспечивает ввод, хранение и поиск информации о сделках, совершенных в фирме. Информационная система может выдавать котировки ценных бумаг, биржевые индексы, данные об уровне надежности компаний. Информация о сделке содержит сведения о дате, времени и сумме сделки, количестве и номинальной стоимости покупаемых и продаваемых ценных бумаг, фамилии, имени, отчестве и номере паспорта клиента, а также о фамилии, инициалах и учетном номере сотрудника фирмы в торговой системе. Система позволяет вычислять денежный оборот за один или несколько дней, а также осуществлять поиск информации о сделках по номеру паспорта клиента.

Вариант 2. Деятельность пункта обмена валюты. В пункте обмена валют создана локальная информационная система, автоматизирующая процесс учета сделок купли-продажи валюты. Информационная система обеспечивает ввод, хранение и поиск информации о сделках, совершенных в данном пункте обмена. Каждой сделке присваивается уникальный цифровой код. Информация о сделке содержит сведения о дате и времени сделки, суммах покупаемой и продаваемой валют, фамилии, имени, отчестве и номере паспорта клиента, а также о фамилии, инициалах и учетном номере личного дела кассира в отделе кадров. Система позволяет вычислять денежный оборот за один или несколько дней, а также осуществлять поиск информации о сделках по номеру паспорта клиента.

Вариант 3. Работа информационной системы коммерческого банка. Информационная система обеспечивает следующие виды работ: формирование уникального идентификационного номера клиента, счета клиента и кассира банка; формирование уникального номера ссуды клиенту в любом отделении банка (номер ссуды отличается от номера счета); формирование входных документов (приходный ордер, расходный ордер); формирование выходных документов (отчет управляющего отделением, отчет о состоянии ссуд по отделению, отчет кассира за текущий день); реализацию запросов (список клиентов, у которых остаток по счету превышает 100000 руб., в какие дни недели сумма выданных денег превышает сумму принятых денег от клиентов).

Вариант 4. Работа информационной подсистемы деканата факультета университета. Информационная система обеспечивает формирование: – входных документов (списки студентов по учебным группам и курсам, списки студентов, находящихся в академическом отпуске, списки студентов обучающихся по индивидуальным планам, списки студентов участвующих в выполнении НИР); – выходных документов (расписание занятий студентов по учебным группам на семестр, список студентов, слушающих заданный учебный курс, список учебных курсов, список студентов, не прошедших текущую аттестацию, списки отлично успевающих студентов, сведения о трудоустройстве выпускников, сведения о студентах, проживающих в общежитии университета и сведения о студентах нуждающихся в общежитии).

Вариант 5. Деятельность переговорного пункта. Информационная система пункта обеспечивает:

– ввод данных об авансовом взносе клиента при предоставлении ему междугородних и международных переговоров;

- ввод данных о тарифах за услуги связи, с учетом особенностей заказа клиента (льготное время, международный звонок, IP-телефония, факс и пр.).

- формирование отчетов о продолжительности разговора клиента, о полной стоимости услуги, предоставленной клиенту, о количестве услуг, предоставленных всем клиентам за указанный период времени (день, неделю, месяц) с разбивкой по видам услуг (междугородние переговоры, международные переговоры, факс, доступ в Интернет и др.).

Вариант 6. Работа информационной подсистемы приемной комиссии университета. Информационная подсистема обеспечивает формирование:

- перечня специальностей, по которым производится набор абитуриентов;
- личных дел абитуриентов;
- расписания тестирования по специальностям и абитуриентам;
- данных о результатах тестирования абитуриентов; – списка абитуриентов, принятых на первый курс.

- отчетов: о числе абитуриентов по различным специальностям, об абитуриентах не прошедших тестирование, об абитуриентах успешно прошедших тестирование по нескольким специальностям и др.

Вариант 7. Деятельность производственно-технического отдела фирмы. В производственно-техническом отделе гипотетической фирмы создана локальная информационная подсистема, автоматизирующая решение задач учета состояния и модернизации компьютерного парка и офисной техники.

Информационная подсистема обеспечивает:

- создание, корректировку и хранение данных о состоянии компьютерного парка и офисной техники с разбивкой по структурным подразделениям фирмы.

- создание, сохранение, корректировку и вывод на печать заявок на модернизацию компьютерного парка и офисной техники с разбивкой по структурным подразделениям фирмы.

- формирование отчетов о техническом состоянии и модернизации компьютерного парка и офисной техники фирмы за указанный период времени (месяц, квартал, полугодие и год).

Вариант 8. Работа информационной подсистемы городского бюро медико-социальной экспертизы. Информационная подсистема обеспечивает:

- формирование карточек учета пациентов;
- обновления данных о диагнозах пациентов, стоящих на учете;
- формирование данных о направлении пациентов на первичное и повторное медицинское освидетельствование;

- ввод данных о трудовой характеристике пациента;

- печать отчетов («О пациентах», «О пациентах и их инвалидности», «О направлении на обследование» и пр.);

- корректировку данных о пациенте;

- формирование отчетов о результатах деятельности бюро по медицинскому освидетельствованию граждан за указанный период времени (месяц, квартал, полугодие и год).

Вариант 9. Работа информационной подсистемы бюро туристической фирмы. Информационная подсистема обеспечивает:

- ввод данных о клиенте;
- формирование отчета обо всех услугах предоставленных выбранному клиенту за период туристического тура;
- формирование отчета обо всех услугах предоставленных фирмой всем клиентам за указанный период времени (за месяц, квартал и т. д.);
- формирование счета клиенту за период туристического тура, с разбивкой по услугам.
- список предоставляемых туристических маршрутов
- полная информация о содержании и условиях тура

Вариант 10. Деятельность офиса продаж оператора сотовой связи. Информационная система офиса обеспечивает:

- ввод, хранение и поиск информации о сделках, совершенных в данном офисе продаж, как по предоставляемым услугам, так и по проданным сотовым телефонам.
- вычисление денежного оборота офиса за указанный период времени (за день, неделю, месяц, квартал и т. д.);
- поиск информации о сделках по номеру паспорта клиента. Каждой сделке присваивается уникальный цифровой код. Информация о сделке содержит сведения о дате и времени сделки, стоимости сотового телефона (услуги), фамилии, имени, отчестве и номере паспорта клиента, а также о фамилии, инициалах и учетном номере менеджера, оформившего сделку.

При проектировании учесть следующие этапы проектирования ИС:

I. Исследование предметной области предусматривает следующие шаги: - Спецификацию деятельности в предметной области.

- Анализ деятельности в предметной области.
- Структурно-логический анализ деятельности:
 - Анализ путей.
 - Анализ связности (прочности и сцепления) компонентов предметной области.
- Анализ производительности.
- Экономический анализ.

II Разработка архитектуры системы включает в себя разработку следующих компонентов:

- Спецификации требований к проектируемой системе.
- Конструирование концептуальной модели предметной области.
- Спецификации обработки данных в проектируемой системе. - Спецификации пользовательского интерфейса системы.
- Спецификации деятельности в предметной области с учетом внедрения системы.

III. Реализация информационных систем на основе информационных технологий

- Разработка программных модулей;
- Восстановление информации о программной системе, ее документации и спецификаций;

- Обнаружение аномалий в архитектуре программной системы, моделях и исходном коде;
- Проверка соответствия исходного кода программы решениям, принятым на этапах анализа и проектирования;
- Перевод исходных кодов программ с одного языка программирования на другой.

IV. Внедрение системы в действие.

- подготовка объекта автоматизации;
- подготовка персонала;
- комплектация ИС поставляемыми изделиями (программными и техническими средствами, программно-техническими комплексами, информационными изделиями);
- строительные-монтажные работы;
- пусконаладочные работы;
- проведение предварительных испытаний;
- проведение опытной эксплуатации; - проведение приемочных испытаний.

V. Сопровождение ИС.

- выполнение работ в соответствии с гарантийными обязательствами;
- послегарантийное обслуживание.

Контрольные вопросы

1. Что такое сущность?
2. Что такое независимая сущность?
3. Что такое зависимая сущность?
4. Каковы правила формирования сущностей (типы, элементы, нумерация, обозначение)?

Лабораторная работа № 4-6

Тема: Проектирование реляционной схемы базы данных в среде СУБД

Цель работы: изучение принципов разработки и формализации даталогической модели предметной области с целью построения баз данных.

Теоретическая часть

Методология IDEF0 (более известная как методология SADT Structure Analysis and Design Technique) предназначена для представления функций системы и анализа требований к системам и является одной из самых известных и широко используемых методологий проектирования ИС.

В терминах IDEF0 система представляется в виде комбинации блоков и дуг. Блоки используются для представления функций системы и сопровождаются текстами на естественном языке. Дуги представляют множества объектов (как физических, так и информационных) или действия, которые образуют связи между функциональными блоками. Место соединения дуги с блоком определяет тип интерфейса.

Управляющие выполнением функции данные входят в блок сверху, в то время как информация, которая подвергается воздействию функции, показана с левой стороны блока; результаты выхода показаны с правой стороны. Механизм

(человек или автоматизированная система), который осуществляет функцию, представляется дугой, входящей в блок снизу. В основе методологии IDEF0 лежат следующие правила: Функциональный блок (или Функция) преобразует Входы в Выходы (т.е. входную информацию в выходную), Управление определяет, когда и как это преобразование может или должно произойти (рисунок 1). Исполнители непосредственно осуществляют это преобразование.

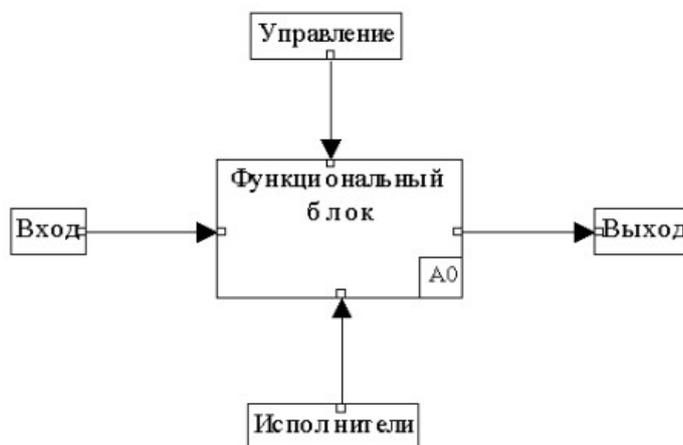


Рисунок 1 – Функциональный блок

С дугами связаны надписи (или метки) на естественном языке, описывающие данные, которые они представляют. Дуги показывают, как функции между собой взаимосвязаны, как они обмениваются данными и осуществляют управление друг другом.

Дуги могут разветвляться и соединяться. Выходы одной функции могут быть Входами, Управлением или Исполнителями для другой. Функциональный блок, который представляет систему в качестве единого модуля, детализируется на другой диаграмме с помощью нескольких блоков, соединенных между собой интерфейсными дугами.

Эти блоки представляют основные подфункции (подмодули) единого исходного модуля. Каждый из этих подмодулей может быть декомпозирован подобным же образом для более детального представления. Взаимодействие с окружающим миром описывается в терминах входа (данные или объекты, потребляемые или изменяемые функцией), выхода (основной результат деятельности функции, конечный продукт), управления (стратегии и процедуры, которыми руководствуется функция) и механизмов (необходимые ресурсы). Кроме того, при создании контекстной диаграммы формулируются цель моделирования, область (описание того, что будет рассматриваться как компонент системы, а что как внешнее воздействие) и точка зрения (позиция, с которой будет строиться модель). Обычно в качестве точки зрения выбирается точка зрения лица или объекта, ответственного за работу моделируемой системы в целом. Затем общая функция разбивается на крупные подфункции. Этот процесс называется функциональной декомпозицией. Затем каждая подфункция декомпозируется на более мелкие – и так далее до достижения необходимой детализации описания.

В процессе проектирования данных решаются следующие задачи:

Сбор и анализ входных данных - это начальный этап проектирования, на котором осуществляется сбор и контроль качества результатов анализа предметной области базы данных, готовится план проектирования базы данных.

Создание логической модели базы данных - это этап, на котором на основании информационной модели предметной области базы данных создается логическая структура базы данных, независимая от ее реализации.

Создание физической модели базы данных: внутренняя схема - это этап, на котором на основании логической модели базы данных создается физическая структура базы данных, зависящая от ее реализации. На этом этапе выполняется преобразование отношений логической модели реляционной базы данных в команды создания объектов физической базы данных, в результате чего создается так называемая внутренняя схема базы данных. Дополнительно может быть создана так называемая внешняя схема базы данных, которая отражает точку зрения пользователей на данные в базе данных. Полученный скрипт может быть применен для создания физической базы данных. Создание физической модели базы данных: учет влияния транзакций - это этап, на котором анализируются возможные транзакции системы, выполняется, в случае необходимости, денормализация отношений для обеспечения более высокой производительности базы данных. На этом этапе создается скрипт создания физической базы данных.

Создание серверного кода - это этап, на котором на основании функциональной модели предметной области базы данных создается серверный код базы данных в виде триггеров, хранимых процедур и пакетов. Эти модули создаются проектировщиком базы данных и выполняются сервером.

Проектирование модулей приложений - это этап, на котором создаются спецификации модулей приложений, разрабатываются стратегии тестирования базы данных и приложений, создается план тестирования приложений базы данных и готовятся тестовые данные. Контроль качества проектирования базы данных заключается в проверке качества результатов проектирования на каждом его этапе. Учет задач обратного влияния заключается в настройке некоторых транзакций к базе данных и локальном перепроектировании базы данных согласно требованиям, поступающим с других этапов создания базы данных. Как правило, последние четыре из сформулированных задач решаются в соответствии с правилами и стандартами, принятыми в конкретной организации.

Методология IDEF1X - один из подходов к семантическому моделированию данных, основанный на концепции Сущность - Отношение (Entity-Relationship), это инструмент для анализа информационной структуры систем различной природы. Информационная модель, построенная с помощью IDEF1X-методологии, представляет логическую структуру информации об объектах системы. Эта информация является необходимым дополнением функциональной IDEF0-модели, детализирует объекты, которыми манипулируют функции системы. Концептуально IDEF1X-модель можно рассматривать как проект логической схемы базы данных для проектируемой системы.

Практическая часть

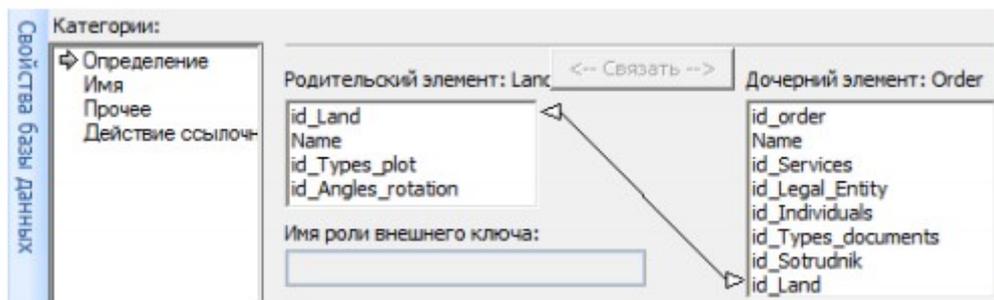
Требуется составить схему модели реляционной базы данных на основе разработанной в лабораторной работе №1-3 инфологической модели.

1. Выполнить преобразование инфологической модели: всех сущностей и их атрибутов в реляционные таблицы и поля с необходимыми типами данных. Например, на рисунке 2 отображены атрибуты сущности «Individuals».

Физическое имя	Тип данных	Индексы	PK	Заметки
id_Individuals	Large Signed Number	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	id_Individuals идентифицирует
Name	SBCS VarChar(10)	<input type="checkbox"/>	<input type="checkbox"/>	Name относится к Individuals
AdresNP	SBCS Char(10)	<input type="checkbox"/>	<input type="checkbox"/>	AdresNP относится к Individuals
AdresUl	SBCS Char(10)	<input type="checkbox"/>	<input type="checkbox"/>	AdresUl относится к Individuals
PasportS	SBCS VarChar(10)	<input type="checkbox"/>	<input type="checkbox"/>	PasportS относится к Individuals
PasportN	SBCS Char(10)	<input type="checkbox"/>	<input type="checkbox"/>	PasportN относится к Individuals
PasportV	SBCS VarChar(10)	<input type="checkbox"/>	<input type="checkbox"/>	PasportV относится к Individuals

Рисунок 2 – Атрибуты сущности «Individuals»

2. Определить первичные ключи реляционных таблиц. Задать внешние ключи для организации связей с соответствующими сущностями, например, на рисунке 3 отображена связь между сущностями «Order» и «Land».



Задание выполняется в выбранной вами системе управления базами данных

Контрольные вопросы

1. Что такое IDEF1X-модель?
2. Какие типы данных можно задать?
3. Как задать первичный ключ?
4. Что такое внешний ключ?
5. Как определить связь между таблицами?

Лабораторная работа № 7-9

Тема: Приведение базы данных к нормальной форме 3НФ

Цель работы: ознакомление с методами нормализации реляционных баз данных.

Теоретическая часть

Одной из проблем разработки базы данных является проблема группировки данных в базе данных. Существует много способов группировки элементов данных; одни из них лучше, другие — хуже, а третьи могут привести к сложным проблемам.

Большинство баз данных непрерывно изменяется. Добавляются новые элементы данных, между ними устанавливаются новые связи и определяются новые способы их использования. При изменении базы данных необходимо сохранять старое представление пользователя о данных для того, чтобы избежать

необходимости переписывать старые программы заново. Однако возможны и такие изменения связей между данными, которые требуют модификации программ. Например, иногда оказывается необходимым расщепить запись или сегмент на две части, или же изменить ключ некоторых элементов. Изменения такого рода являются крайне нежелательными, т.к. могут привести к разрушению структуры базы данных. В то же время такие разрушения можно сделать маловероятными, если группировка элементов данных и их ключи первоначально были хорошо продуманы.

Рассматриваемая методика рациональной группировки элементов данных, называемая нормализацией базы данных, является наиболее распространенной основой разработки структур баз данных, минимизирующей вероятность их разрушения.

Нормализация — это формальный аппарат ограничений на формирование таблиц, устраняющий дублирование, обеспечивающий непротиворечивость хранимых данных и уменьшающий трудозатраты на ведение базы данных.

Процесс нормализации заключается в разложении (декомпозиции) исходных отношений базы данных на более простые. При этом на каждой ступени этого процесса схемы отношений приводятся к нормальным формам. Для каждой ступени нормализации имеются наборы ограничений, которым должны удовлетворять отношения базы данных. Процесс нормализации имеет своей целью устранение избыточности данных и заключается в последовательном приведении базы данных к третьей нормальной форме (ЗНФ).

Первая нормальная форма (1НФ) требует, чтобы каждое поле таблицы базы данных было неделимым и не содержало повторяющихся групп.

Неделимость поля означает, что содержащиеся в нем значения не должны делиться на более мелкие поля. Например, если в поле «Подразделение» содержится название факультета и название кафедры, требование неделимости не соблюдается и необходимо выделить название факультета или кафедры в отдельное поле; поле, содержащее Фамилию, имя и отчество, следует разделить на три поля — отдельно для фамилии, имени и отчества и т.д.

Повторяющимися являются поля, содержащие одинаковые по смыслу значения, например, если требуется получить статистику продаж четырех товаров по месяцам, можно создать поля для хранения данных о продаже по каждому товару (таблица 1).

Таблица 1. Статистика продаж

Статистика продаж
Год
Месяц
Товар 1
Товар 2
Товар 3
Товар 4

Однако такой подход не годится, если количество товаров заранее не известно. Повторяющиеся группы следует устранить, сохранив в таблице

единственное поле «Товар» (таблица 2.) В результате получим запись, содержащую информацию о статистике продаж по одному товару, но этот товар может быть любым: для 4 товаров будем иметь 4 записи, для 104 товаров — 104 записи и т.д.

Таблица 2. Приведение к 1НФ

Статистика продаж
<p>Год Месяц Товар</p>

Рассмотрим пример приведения базы данных к 1НФ. Пусть необходимо автоматизировать процесс отпуска товаров со склада по накладной, примерный вид которой показан в таблице 3.

Таблица 3. Накладная

		Накладная 123		
Дата	Покупатель	Адрес		
10.01.2020	ООО «Кормилец»	г. Иркутск, ул. Тракторная, 20		
Отпущенный товар	Количество	Ед. измерения	Цена	Общая стоимость
Тушенка	10000	Банки	700	7000000
Сахар	200	кг	500	100000
Макароны	1000	кг	300	300000
Итого				7400000

Сведем имеющиеся данные в одну таблицу. Приводя ее к 1НФ, учтем, что впоследствии будет необходимо производить анализ продаж по городам. Поэтому из поля «Адрес» (допускающего толкование как делимого поля) выделим часть данных (город) в отдельное поле «Город».

Известно, что каждый покупатель может закупить в один день различное количество товаров, однако, чтобы не создавать повторяющихся групп, фиксируем факт отпуска каждого товара в отдельной записи. В результате получим таблицу, показанную в Таблице 4.

Вторая нормальная форма (2НФ) требует, чтобы все поля таблицы зависели от первичного ключа, т.е. чтобы первичный ключ однозначно определял запись и не был избыточен. Поля, которые зависят только от части первичного ключа, должны быть выделены в отдельные таблицы.

Таблица 4.

Отпуск товаров со склада
Дата

Покупатель
Город
Адрес
Товар
Ед_измерения
Цена_за_ед_измерения
Отпущено_ед
Общая_стоимость
№_накладной

Для приведения к 2НФ выделим поля, которые входят в первичный ключ. Дата накладной и номер накладной по отдельности не могут уникально определять запись, поскольку они будут одинаковы для всех записей, относящихся к одной и той же накладной (напомним, что одна накладная в таблице 4 представляется несколькими записями). Поэтому введем в первичный ключ поле «Товар». При этом исходим из предположения, что по одной накладной может быть отпущено одно наименование конкретного товара, т.е. не может быть ситуации, когда отпуск одного и того же товара оформляется в накладной двумя строками, что повлекло бы за собой две одинаковые записи в таблице «Отпуск товаров со склада».

Таблица 5.

Отпуск товаров со склада
Дата Покупатель Товар №_накладной Город Адрес Ед_измерения Цена_за_ед_измерения Отпущено_ед Общая_стоимость

В таблице 5. показана структура таблицы после выделения полей сцепленного первичного ключа (эти поля отчеркнуты от прочих полей линией и располагаются в верхней части структуры таблицы). Созданный нами первичный ключ является избыточным: поле «№ накладной» однозначно определяет дату и покупателя. Для данной накладной не может быть никакой иной даты и никакого иного покупателя. Поле «Товар» в комбинации с № накладной, напротив, однозначно идентифицирует запись. После устранения избыточности первичного ключа получаем таблицу 6.

Таблица 6.

Отпуск товаров со склада
Товар

№_накладной
Дата
Покупатель
Город
Адрес
Ед_измерения
Цена_за_ед_измерения
Отпущено_ед
Общая_стоимость

Первое требование 2НФ выполнено, чего не скажешь о втором, гласящем, что значения всех полей записи должны однозначно зависеть от совокупного значения первичного ключа и не должна иметь место ситуация, когда некоторые поля зависят от части первичного ключа. В Таблице 9.6 поля «Единица измерения», «Цена за единицу измерения» зависят от значения поля «Товар», входящего в первичный ключ. Поэтому выделяем эти поля в самостоятельную таблицу «Товары» и определяем связь: поскольку один товар может присутствовать во многих накладных, таблицы «Товары» и «Отпуск товаров со склада» находятся в связи 1:М (рисунок 3).

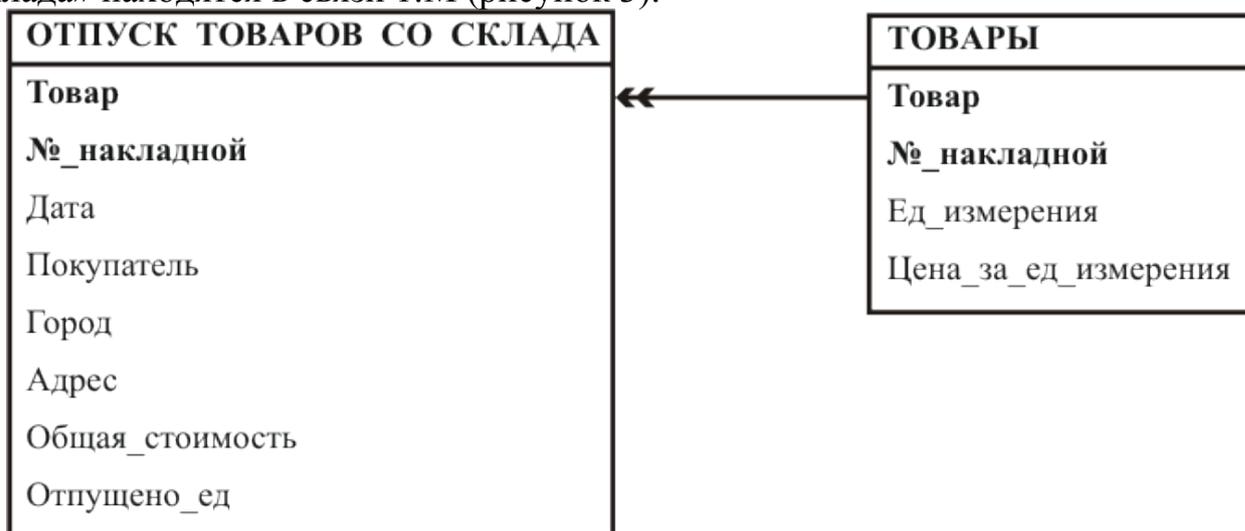


Рисунок 3. Связи

Можно заметить, что теперь значение поля «Покупатель» никоим образом не зависит от пары значений «№ накладной», «Товар», а «зависит только от значения поля «№ накладной». Поэтому данное поле и зависящие от его значения поля «Город», «Адрес» выделяем в таблицу «Покупатели». Анализируя далее структуру таблицы «Отпуск товаров со склада», обнаруживаем, что одно из оставшихся полей — «Дата» — зависит только от значения поля «№ накладной». Поэтому выделяем дату и номер накладной в самостоятельную таблицу «Накладные» (рисунок 4).

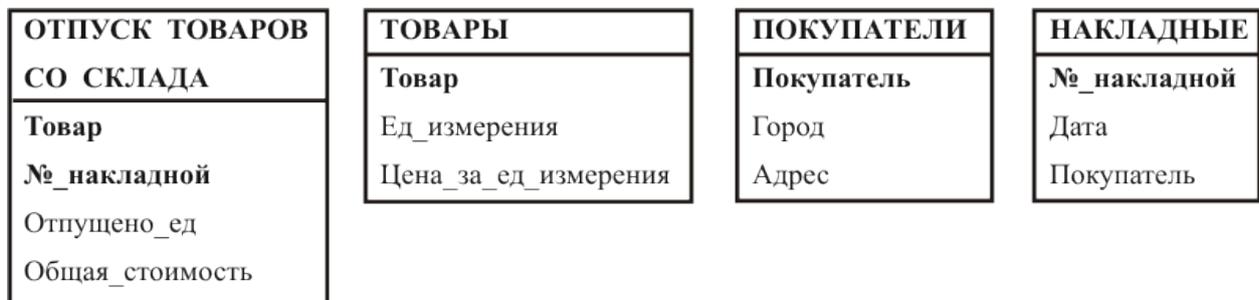


Рисунок 4. Таблица Накладные

Установим связи между таблицами. Один покупатель может встречаться во многих накладных. Поэтому между таблицами «Покупатели» и «Накладные» имеется связь 1:М по полю «Покупатель». Одной накладной может соответствовать несколько товаров. Поэтому между таблицами «Накладные» и «Отпуск товаров со склада» имеется связь 1:М по полю «№ накладной» (рисунок 5).

Обратите внимание, что в реляционной базе данных таблицы могут быть связаны только по одноименным полям, поэтому при выделении новых таблиц в них добавляют поля, по которым осуществляется связь с другими таблицами базы данных.



Рисунок 5. База данных во 2-й нормальной форме

Третья нормальная форма (3НФ) требует, чтобы в таблице не имелось транзитивных зависимостей между неключевыми полями, т.е. чтобы значение любого поля, не входящего в первичный ключ, не зависело от значения другого поля, также не входящего в первичный ключ.

В рассматриваемом примере можно увидеть, что в таблице «Отпуск товаров со склада» имеется зависимость значения поля «Общая стоимость» от значения поля количество т. е. «Отпущено_ед». По условию примера значение поля «Общая стоимость» может вычисляться как значение поля «Отпущено_ед», умноженное на значение поля «Цена за единицу измерения» из таблицы «Товары» (из записи с таким же значением поля «Товар»). Поэтому поле «Общая стоимость» из таблицы «Отпуск товаров со склада» удаляем.

Однако в случае, если «Цена за единицу измерения» зависит от количества отпущенного товара, что особенно распространено в наше рыночное время, следует создать отдельную таблицу, отображающую зависимость цены от количества отпущенного товара, после чего будет получена база данных в 3НФ.

Замечание. В таблице «Покупатели» значение поля «Адрес» зависит от значения поля «Город», поскольку в разных городах могут оказаться улицы с одинаковыми названиями и, соответственно, дома с одинаковыми номерами (вспомним известный кинофильм «Ирония судьбы, или с легким паром»). Однако такой зависимостью можно пренебречь, поскольку поле «Адрес» в нашем случае

носит чисто информационный характер и не должно входить в условия запросов самостоятельно. Вообще говоря, на практике не всегда возможно получить идеально нормализованную базу данных, да и не всегда это необходимо, т. е. бывает достаточно 2-й нормальной формы.

Практическая часть

1. На основании приведенных данных в таблице 7 необходимо получить базу данных в 1НФ.

Таблица 9.7

Ф.И.О	Должность	Оклад	Стаж	Надбавка	Кафедра	Дисциплина	Группа	Вид занятий
Волков И. С.	ассистент	1500	4	100	ГиМУ	СУБД	81	Практ.
						ИС	79	Практ.
Белкин А. М.	доцент	2000	8	300	ГиМУ	СУБД	81	Лекция
						ИТУ	81	Практ.
Зайцев С. С.	ассистент	1500	10	400	ГиМУ	ИС	79	Лекция
						ИТУ	81	Лекция
Медведев Е. В.	ассистент	1500	4	100	Э	Экономика	70	Лекция

2. Преобразовать схему базы данных в 1НФ ко 2НФ.
3. Преобразовать схему базы данных во 2НФ к 3НФ.
4. Результаты работы по всем этапам отобразите в отчете.
5. Приведение базы данных к более высоким нормальным формам поощряется.

Контрольные вопросы

1. Алгоритм приведения к 1НФ?
2. Алгоритм приведения к 2НФ?
3. Алгоритм приведения к 3НФ?

Лабораторная работа № 10-11

Тема: Создание базы данных в среде разработки

Цель работы: закрепить умения и навыки проектирования структуры базы данных, создания SQL баз данных в среде Visual Studio.Net.

Теоретическая часть

В процессе выполнения лабораторной работы необходимо сформировать умения:

- создавать базу данных с помощью инструментов Visual Studio.Net;
- заполнять таблицы базы данных;
- выводить данные на форму с помощью запросов;
- создавать хранимые процедуры с помощью инструментов Visual Studio.Net

Будем создавать SQL-базу данных *BDTur_firm.mdf* и потом создадим хранимые процедуры.

Хранимая процедура - это одна или несколько SQL-конструкций, которые записаны в базе данных. Задача администрирования базы данных включает в себя в первую очередь распределение уровней доступа к ней. Разрешение выполнения обычных SQL-запросов большому числу пользователей может стать причиной неисправностей из-за неверного запроса или их группы. Чтобы их избежать, разработчики базы данных могут создать ряд хранимых процедур для работы с данными и полностью запретить доступ для обычных запросов. Такой подход при прочих равных условиях обеспечивает большую стабильность и надежность работы. Это одна из главных причин создания собственных хранимых процедур. Другие причины - быстрое выполнение, разбиение больших задач на малые модули, уменьшение нагрузки на сеть - значительно облегчают процесс разработки и обслуживания архитектуры "клиент-сервер".

Операция *inner join* объединяет записи из двух таблиц, если поле (поля), по которому связаны эти таблицы, содержат одинаковые значения. Общий синтаксис выглядит следующим образом:

```

from таблица1 inner
join таблица2
on таблица1.поле1
оператор_сравнения
таблица2.поле2

```

Будем создавать базу данных *BDTur_firm.mdf*.

Имейте ввиду, что базу данных надо заполнить до того, как вы подключите к ней проект Windows Application. Фамилия первого туриста должна быть созвучна вашей фамилии, то есть фамилии автора проекта.

Структура создаваемой базы данных имеет следующий вид (при создании таблиц и полей обратите внимание на то, чтобы их имена состояли из одного слова, например, таблица *Информация_о_туристах* или столбец *КодТуриста*):

№	Название	Структура в режиме конструктора		Описание
		Имя поля	Тип данных	
1	Туристы	Код туриста	Счетчик	Содержит основные сведения о туристе
		Фамилия	Текстовый	
		Имя	Текстовый	
		Отчество	Текстовый	
2	Информация о туристах	Код туриста	Числовой	Содержит дополнительные сведения о туристе, которые были вынесены в отдельную таблицу - для избегания повторяющихся записей
		Серия паспорта	Текстовый	
		Город	Текстовый	
		Страна	Текстовый	
		Телефон	Текстовый	
		Индекс	Числовой	
3	Туры	Код тура	Счетчик	Содержит общие сведения о странах для туров
		Название	Текстовый	
		Цена	Денежный	
		Информация	Поле МЕМО	

4	Сезоны	Код сезона	Счетчик	Содержит сведения о сезонах - некоторые туры доступны лишь в определенный период
		Код тура	Числовой	
		Дата начала	Дата/время	
		Дата конца	Дата/время	
		Сезон закрыт	Логический	
		Количество мест	Числовой	
5	Путевки	Код путевки	Числовой	Содержит сведения о путевках, реализованных туристам.
		Код туриста	Числовой	
		Код сезона	Числовой	
6	Оплата	Код оплаты	Счетчик	Содержит сведения об оплате за путевки
		Код путевки	Числовой	
		Дата оплаты	Дата/время	
		Сумма	Денежный	

1. Создайте на рабочем столе папку с именем, созвучным вашей фамилии
2. После запуска Visual Studio.Net создайте новый проект Windows Application, назвав его, *Creation_data_base*
3. В Обозревателе решений вызовите контекстное меню узла *Creation_data_base* и выберите команду **Добавить – Компонент** (Add – New Item...)
4. В появившемся диалоговом окне **Добавление нового элемента** выберите пункт **База данных, основанная на службах**, введите имя базы данных *BDTur_firm.mdf*

Создание SQL-базы данных

5. Раскрыв узел созданного соединения в Обозревателе Серверов вы увидите, что ваша база данных еще не содержит таблиц
6. Вызовите контекстное меню на узле **Таблицы** (Tables) и выберите команду **Добавить новую таблицу** (Add New Table)
7. Создайте таблицу Туристы

Имя столбца	Тип данных	Разрешить значения null
КодТуриста	int	<input type="checkbox"/>
Фамилия	nchar(50)	<input checked="" type="checkbox"/>
Имя	nchar(50)	<input checked="" type="checkbox"/>
Отчество	nchar(50)	<input checked="" type="checkbox"/>

Обозначьте ключевое поле: **Конструктор таблиц – Задать первичный ключ** (Table Designer – Set Primary Key). Сохраните таблицу, вызвав контекстное меню на ее закладке и выбрав команду **Сохранить <Имя таблицы>**

8. Аналогично создайте остальные таблицы

Теперь в окне базы данных есть несколько таблиц.

Заполнение SQL-базы данных

9. Заполните таблицы. Для этого в Server Explorer вызовите контекстное меню на имени таблицы и выберите команду **Показать таблицу данных** (Show Table Data)

Аналогично заполните оставшиеся пять таблиц. При заполнении таблицы Путевки договоримся, что код зимы – 1, весны – 2, лета – 3, осени – 4.

10. При заполнении таблицы сезоны:

Код сезона	Код тура	Дата начала	Дата конца	Сезон закрыт	Количество мест
1	11	01.01.2008 0:0...	31.01.2008 0:0...	NULL	10
2	22	01.03.2008 0:0...	30.03.2008 0:0...	NULL	10
3	33	01.06.2008 0:0...	30.06.2008 0:0...	NULL	10
4	44	01.09.2008 0:0...	30.09.2008 0:0...	NULL	10

Создание хранимых процедур

11. Выделите папку **Хранимые процедуры (Stored Procedures)** и вызовите команду **Данные - Добавить новый – Хранимая процедура (Data-Add New-Stored Procedure)**

Создадим процедуру Fill для вывода всех записей таблицы Туристы (только помните, что для создания процедуры используется команда CREATE, а команда ALTER означает, что процедура уже была сохранена и ее изменяют)

```

ALTER PROCEDURE Fill
AS
select Кодтура, фамилия, Имя, Отчество
from Туристы
RETURN

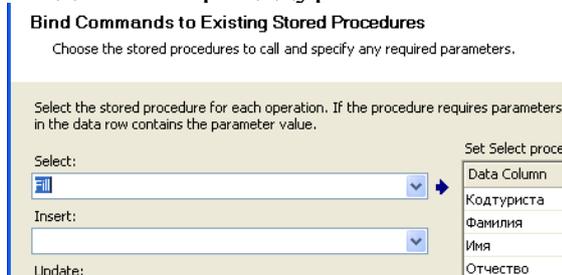
```

Теперь рассмотрим, как использовать хранимые процедуры в Windows-приложениях, связанных с базой данных.

12. Добавьте на форму объект DataGridView, установив его свойству Dock значение Fill

13. Добавьте на форму элемент управления SqlDataAdapter, настроив подключение к вашей базе данных, проверьте подключение.

14. Следуя указаниям Мастера настройки адаптера данных, в шаге **Выбор типа команды** выберите пункт **Использовать существующие хранимые процедуры (Use existing stored procedures)** и в раскрывающемся списке Select найдите имя созданной процедуры

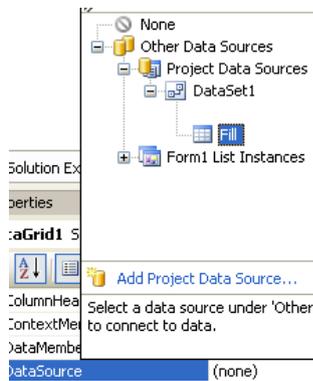


Завершите настройку адаптера.

15. Просмотрим данные, которые будут извлечены объектом DataAdapter. Выделяем sqlDataAdapter1, вызываем контекстное меню адаптера (правая кнопка мыши), переходим по ссылке **Просмотреть данные (Preview Data)**

В окне **Просмотр данных** выберите **Просмотр** и просмотрите результат

16. Создайте набор данных (**Данные – Создать набор данных**) и свяжите объект DataGridView с хранимой процедурой (см. screenshot)



17. В конструкторе формы вызываем метод Fill для события Load формы для заполнения DataSet (дважды кликнуть на заголовке формы). *Имейте в виду, что имя набора данных в проекте другое*

```
Students1.Clear()
OleDbDataAdapter1.Fill(Students1)
```

Запустите проект, протестируйте его работу.

18. Создадим еще несколько хранимых процедур. Выделите папку **Хранимые процедуры** и вызовите команду **Данные - Добавить новый - Хранимая процедура**

1) Процедура для подсчета числа записей таблицы Туристы

```
dbo.StoredPr...UR_FIRM.MDF)* DataSet1.xsd
CREATE PROCEDURE Число_записей
AS
select count (*)
from Туристы
RETURN
```

2) Подсчет значений поля Сумма таблицы Оплата

```
dbo.StoredPr...UR_FIRM.MDF)*
CREATE PROCEDURE Summa
AS
select sum(Сумма)
from Оплата
RETURN
```

3) Вывод максимального значения поля Цена таблицы Туры

```
dbo.StoredPr...UR_FIRM.MDF)* DataSe
CREATE PROCEDURE Максимум
AS
select max(Цена)
from Туры
RETURN
```

4) Для передачи изменений в базу данных

```
ALTER PROCEDURE myInsertSP
(@Кодтуриста int,
@фамилия NChar(50),
@Имя NChar(50),
@Отчество NChar(50))
AS
Insert Into Туристы
(Кодтуриста, фамилия, Имя, Отчество)
Values (@Кодтуриста, @фамилия, @Имя, @Отчество)
RETURN
```

5) Создаем процедуру типа UP_DATE

```
ALTER PROCEDURE Up_date
    (@Кодтуриста int,
     @Фамилия NVarChar (50),
     @Имя NVarChar (50),
     @Отчество NVarChar (50))
AS
UPDATE Туристы SET Кодтуриста = @Кодтуриста,
    Фамилия = @Фамилия, Имя = @Имя, Отчество = @Отчество
WHERE (Кодтуриста = @Кодтуриста)
RETURN
```

6) Для объединения записей двух таблиц (операция inner join)

```
CREATE PROCEDURE Объединение
AS
select *
from Туристы inner
join Информация_о_туристах
on Туристы.Кодтуриста=Информация_о_туристах.Кодтуриста
RETURN
```

19. Изменим настройку объекта DataAdapter. Выделяем SqlDataAdapter1, в окне Properties щелкаем по ссылке "Configure DataAdapter", при настройке адаптера оставляем имеющееся уже подключение, а на шаге выбора процедуры выбираем процедуру Объединение. Завершаем работу мастера настройки адаптера.

20. Изменим свойство DataSource объекта DataGrid: свяжем его с набором DataSet11.

21. Запускаем приложение. Теперь мы видим две ссылки - "Fill" и "Объединение". Переходя по последней, мы видим данные хранимой процедуры. Если мы перейдем по ссылке "Fill". Дело в том, что в структуре объекта DataSet остался "след" первой хранимой процедуры.

Сохраните файлы созданной вами базы данных, так как мы будем работать с этой базой на последующих лабораторных работах

Контрольные вопросы

1. Опишите процесс создания SQL базы данных в среде Visual Studio
2. Что такое хранимая процедура, каковы ее достоинства
3. Как создать хранимую процедуру в среде VS.NET, как к ней обратиться?
4. Какие свойства используются для связывания объекта сетки с набором данных?
5. Опишите синтаксис операции объединения записей двух таблиц.

Лабораторная работа № 12-13

Тема: Организация локальной сети. Настройка локальной сети

Цель работы: ознакомиться с моделированием организации и настройки локальной сети в программе 10-Strike LANState .

Теоретическая часть

Программа 10-Strike LANState , позволяет осуществлять мониторинг сетевых служб и устройств, устранять неполадки в их работе, и сокращать простои.

Внезапные сбои в работе ответственных служб и протоколов сервера или активного сетевого оборудования часто оборачивается для компании немалыми убытками и подорванным доверием клиентов. В обязанности системного

администратора входит задача своевременного обнаружения таких неполадок и их быстрого устранения. Но справиться с этой задачей без специальных программных инструментов подчас очень нелегко, и, можно сказать, невозможно. Решением проблемы автоматического мониторинга сети является программа 10-Strike LANState. Из под ее контроля не уйдет ни один сбой в работе сетевой службы или протокола. Программа вовремя обнаружит неполадку и сообщит о ней системному администратору.

В основе работы программы лежит механизм периодического выполнения заданных проверок контролируемых служб и протоколов на серверах и другом сетевом оборудовании. О результате проверок системный администратор оповещается несколькими альтернативными способами: электронной почтой, SMS, звуковым сигналом. Кроме этого, программой ведется фиксация всех событий в журналах с подробной расшифровкой неполадок и временем их происхождения.

10-Strike LANState обладает возможностями мониторинга работы серверов баз данных, систем управления базами данных, значений некоторых параметров производительности сетевого оборудования (например, трафик на коммутаторах), а также оперативного доведения информации до системного администратора о достижении критических значений этих параметров. Для устранения неполадок программа может автоматически выполнить заданные администратором действия: перезагрузку служб и компьютеров, запустить программу или скрипт. Кроме этого, отличительной особенностью 10-Strike LANState является то, что она наглядно отображает контролируемые устройства в виде графической карты сети со связями и условными обозначениями (имеется веб-интерфейс). Карта призвана визуализировать результаты мониторинга, и позволяет быстро определить местонахождение сбойного устройства.

В новой версии 10-Strike LANState реализована возможность отслеживания изменений в списке установленного программного обеспечения на серверах и рабочих станциях локальной сети. Системный администратор будет оповещен о фактах установки пользователями новых программ и удаления старых.

Практическая часть

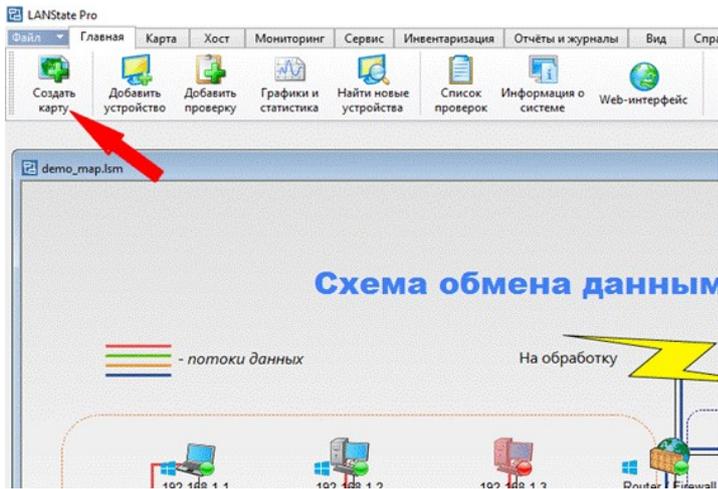
Часть I. Построение схемы сети

1. Установите на свой компьютер программу LANState
2. Запустите программу.
3. Создание схемы сети автоматически

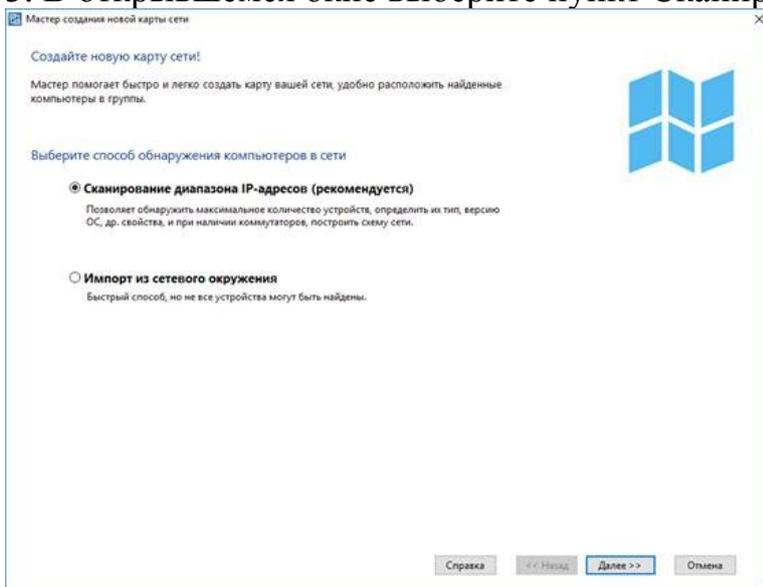
Начиная с версии 3.3, LANState поддерживает сканирование SNMP-устройств и может рисовать схему сети автоматически с созданием линий, соединяющих хосты. При этом номера портов коммутаторов проставляются в подписях к линиям.

Итак, как построим схему сети автоматически:

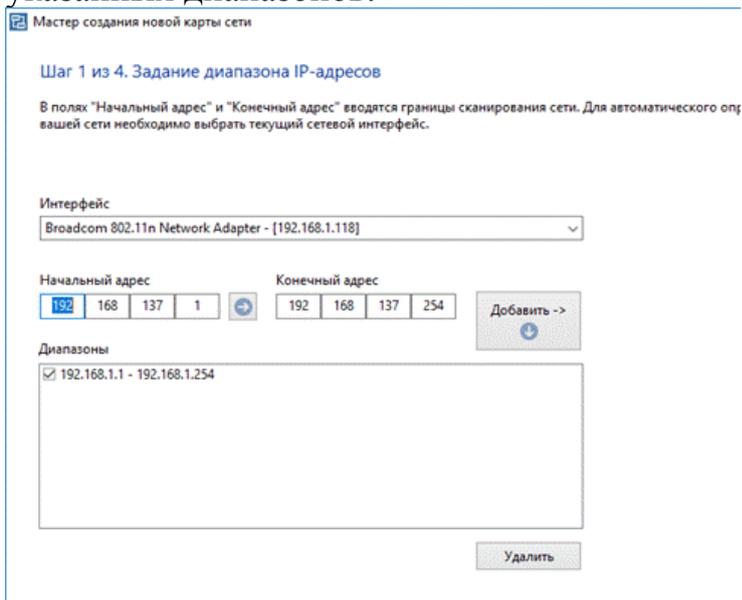
1. SNMP должен быть включен на коммутаторах. Программа должна быть разрешена в брандмауэре для успешной работы по протоколу SNMP.
2. Запустите Мастер Создания Карты Сети.



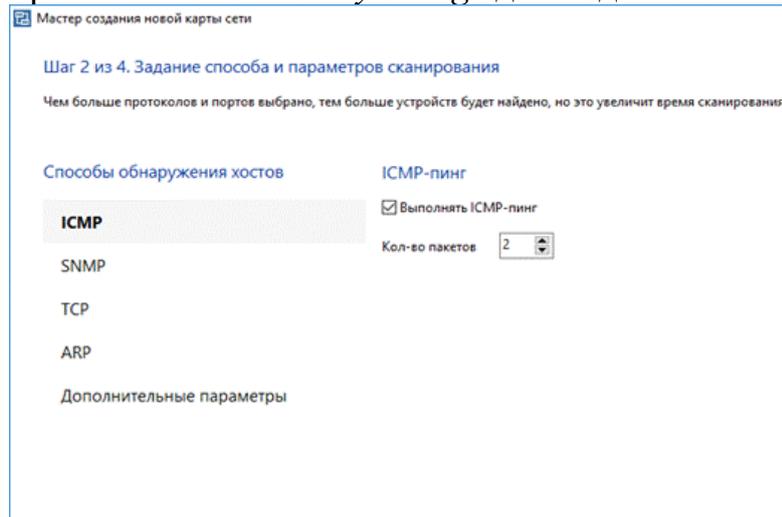
3. В открывшемся окне выберите пункт Сканирование диапазона IP-адресов



4. Выберите сканирование сети по диапазону IP-адресов. Укажите диапазоны (от 192.168.100.1 до 192.168.100.254) Устройства с SNMP должны находиться внутри указанных диапазонов.



4. Выберите методы сканирования и настройте их параметры. Не забудьте поставить галочку рядом с опцией "Поиск устройств с SNMP..." и укажите правильные *community strings* для подключения к коммутаторам.



5. После сканирования программа должна нарисовать схему сети. Если сканирование SNMP прошло успешно, соединения между сетевыми устройствами будут нарисованы автоматически. Передвиньте мышкой устройства для лучшего восприятия схемы.

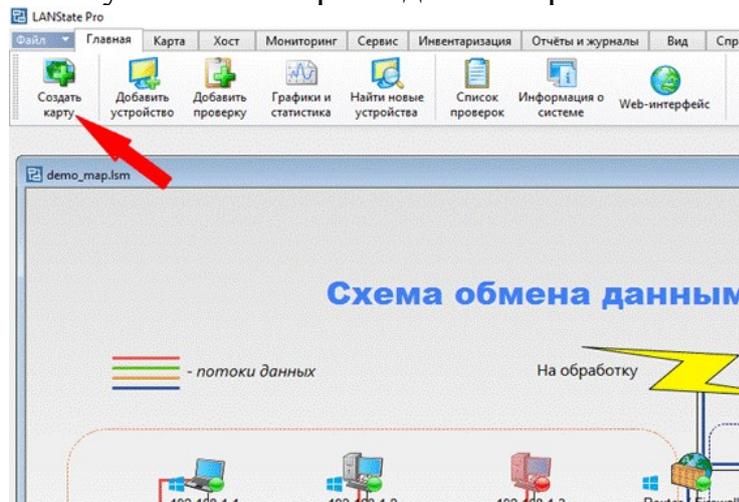
Часть I. Построение схемы сети

1. Установите на свой компьютер программу LANState
2. Запустите программу.
3. Создание схемы сети автоматически

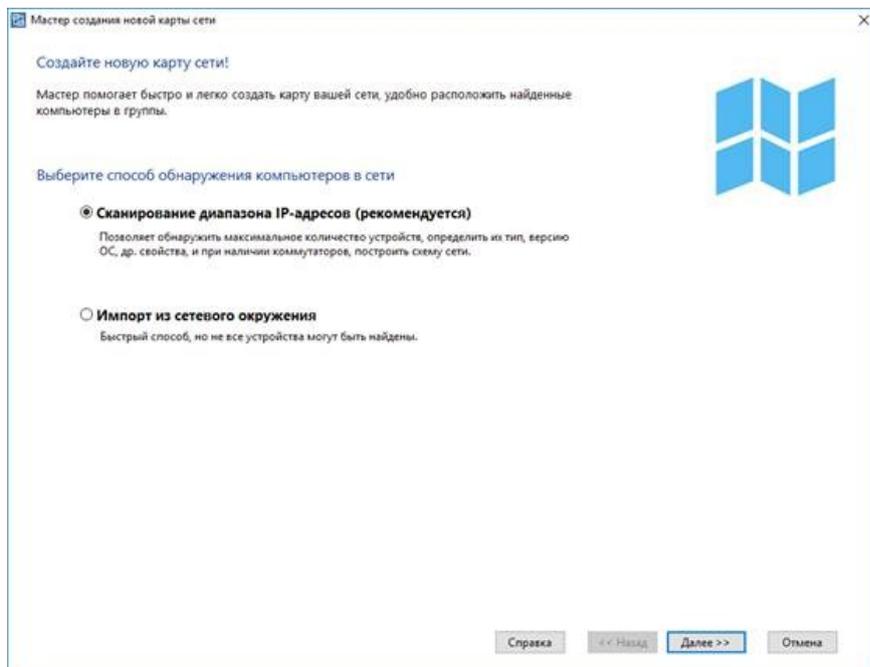
Начиная с версии 3.3, LANState поддерживает сканирование SNMP-устройств и может рисовать схему сети автоматически с созданием линий, соединяющих хосты. При этом номера портов коммутаторов проставляются в подписях к линиям.

Итак, как построим схему сети автоматически:

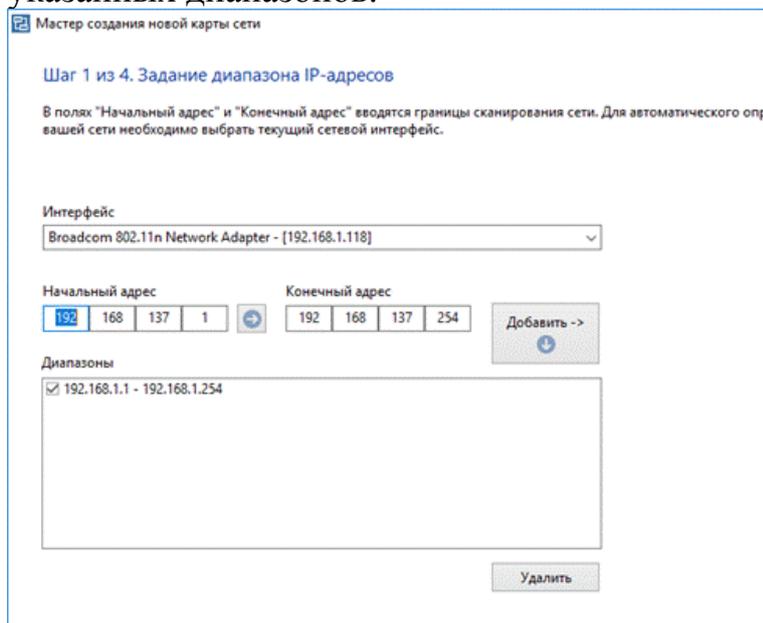
1. SNMP должен быть включен на коммутаторах. Программа должна быть разрешена в брандмауэре для успешной работы по протоколу SNMP.
2. Запустите Мастер Создания Карты Сети.



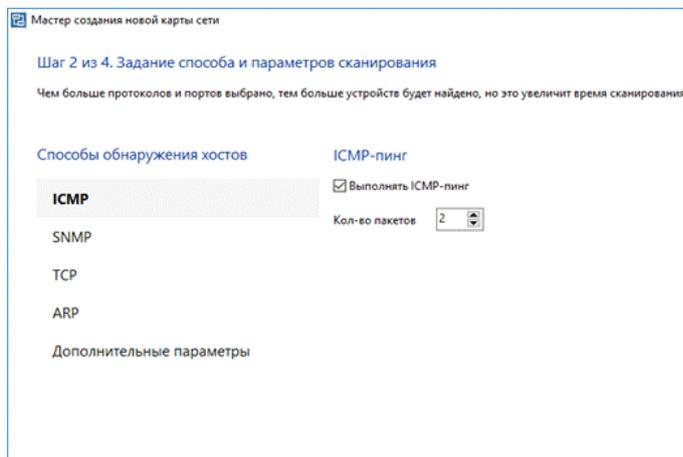
3. В открывшемся окне выберите пункт Сканирование диапазона IP-адресов



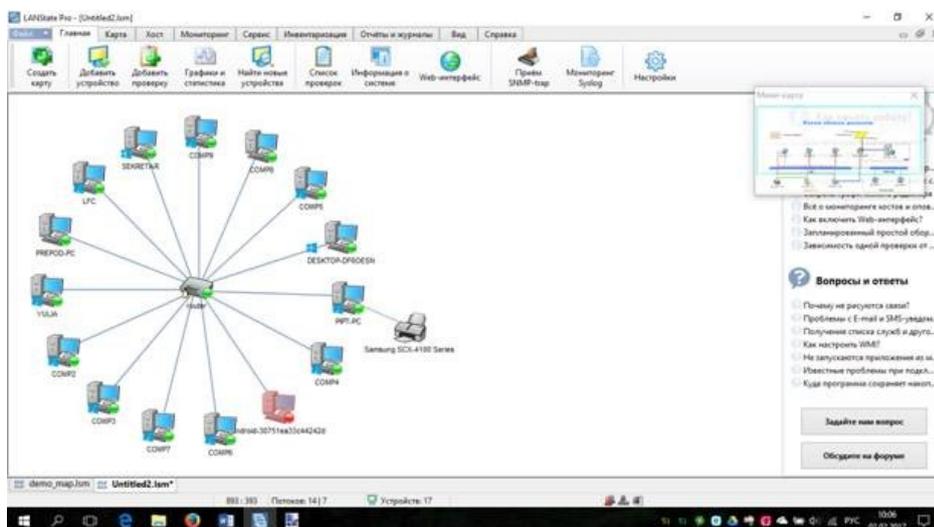
4. Выберите сканирование сети по диапазону IP-адресов. Укажите диапазоны (от 192.168.100.1 до 192.168.100.254) Устройства с SNMP должны находиться внутри указанных диапазонов.



4. Выберите методы сканирования и настройте их параметры. Не забудьте поставить галочку рядом с опцией "Поиск устройств с SNMP..." и укажите правильные *community strings* для подключения к коммутаторам.



5. После сканирования программа должна нарисовать схему сети. Если сканирование SNMP прошло успешно, соединения между сетевыми устройствами будут нарисованы автоматически. Передвиньте мышкой устройства для лучшего восприятия схемы.



6. Схема сети может быть выгружена в картинку, либо в схему Microsoft Visio (только в LANState Pro). Полученную схему сохраните в отдельный файл.

Часть II. Построение диаграмм сети

Краткие теоретические сведения

Программа построения диаграмм сети EDraw Network Diagrammer

При проектировании сетей иногда используется EDraw *Network Diagrammer* – программа создания диаграмм сети с большим количеством примеров и шаблонов.

Основные диаграммы:

Топологические схемы сети

Проектирование сетей Cisco

Диаграммы кабельных сетей

Диаграммы LAN (локальная компьютерная сеть)

Диаграммы сетей WAN (глобальная сеть)

Сетевая диаграмма (граф сети) - графическое отображение работ проекта сети и их взаимосвязей. Отличием от блок-схемы является то, что *сетевая*

диаграмма моделирует только логические зависимости между элементарными работами. Она не отображает входы, процессы и выходы.

Программа имеет как сходство с программой 10 Страйк: Схема Сети, так и принципиальные отличия. Например, в ней можно нарисовать не только изображение сети (рис.), но и изображение помещения, где эту сеть планируется установить.

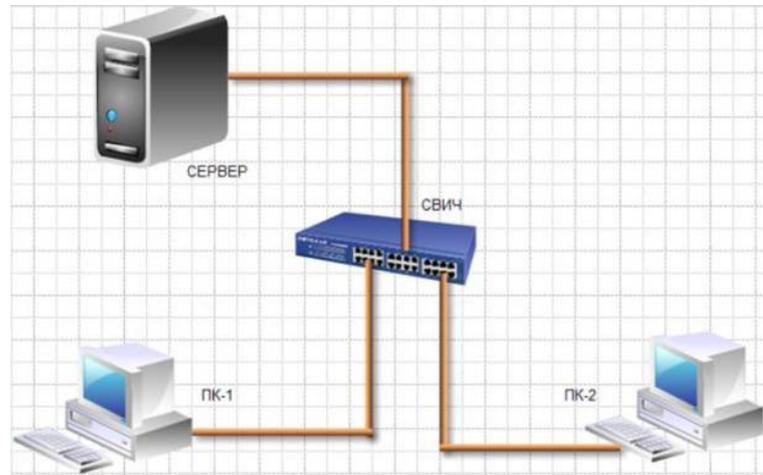


Рисунок 1. Пример элементарной схемы сети, выполненной в EDraw Network Diagrammer

Задание 1

1. Постройте схему, изображенную на рисунке 1.
2. Для выбора компьютеров и мониторов из библиотеки (Libraries) нужно выбрать команду Network-Computers and Monitors, а для выбора кабелей – команду Network and Peripherals.

Задание 2 Нарисуйте схему помещения, изображенного на рисунке 2.

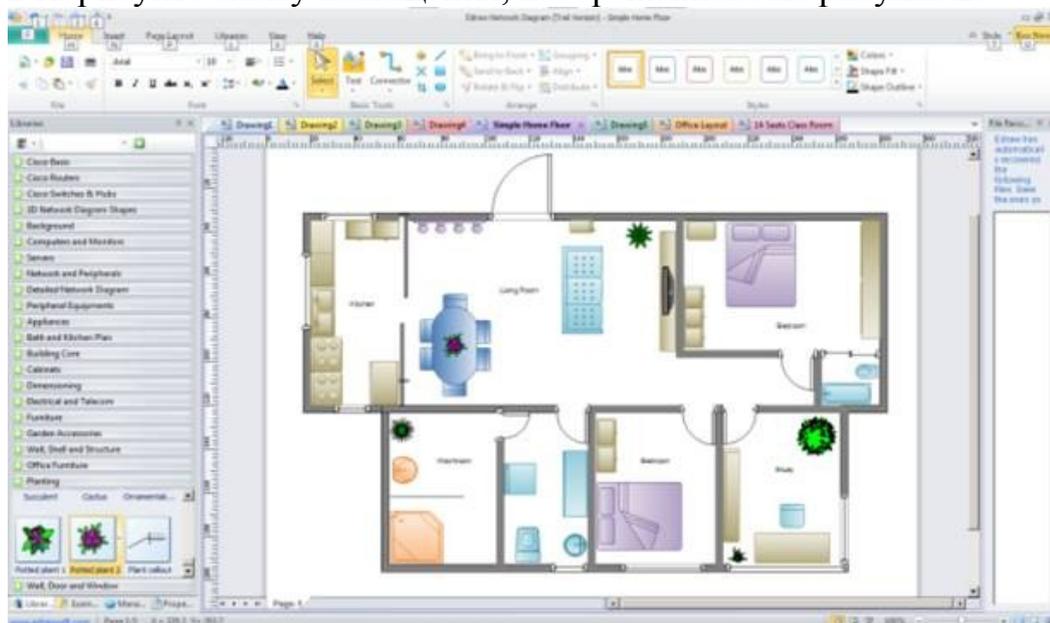


Рисунок 2. Изображение офисного помещения, нарисованного в EDraw Network Diagrammer

В этом случае из библиотеки нужно выбрать вариант Floor Plans (рисунок 3).



Рисунок 3. Различные схемы офисов, для размещения в них ПК
 Задание 3. В программе EDraw Network Diagrammer повторите схему, показанную на рисунке 4. Поясните, что за устройства присутствуют в данной сети и как они работают.



Рисунок 4. Схема сети небольшого офиса
 Задание 3. Повторите рисунок, изображающий расположение компьютеров в компьютерном классе (рисунок 5).

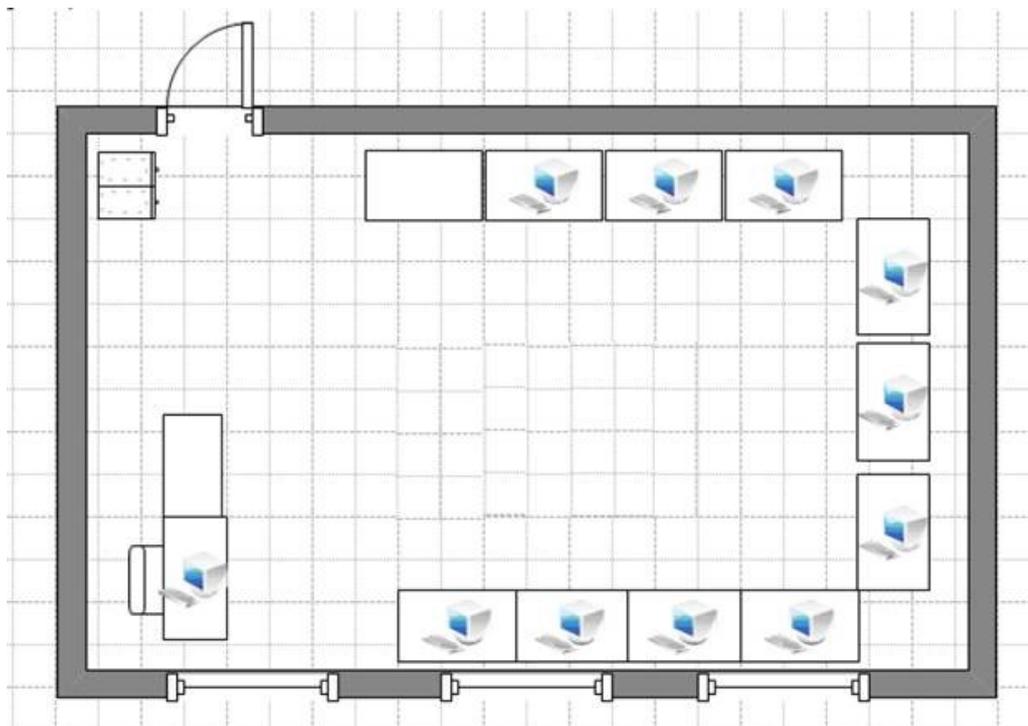


Рисунок 5. Расположение компьютеров в некотором ВЦ

Контрольное задание :

Используя возможности программы EDraw Network Diagrammer создайте схему помещения и расположения компьютерной техники в нашем кабинете (по аналогии с рисунком 5)

Контрольные вопросы

1. Назовите основную функцию программы 10-Strike LANState.
2. Перечислите сетевые функции программы 10-Strike LANState, применимые к удаленным компьютерам.
3. Какие в программе 10-Strike LANState реализованы полезные сервисные функции?
4. Вы просканировали сеть программой 10-Strike LANState , нашлись компьютеры, но связи не прорисованы. Почему?

Лабораторная работа № 14-15

Тема: Установка и настройка SQL-сервера

Цель работы: приобретение навыков установки и первоначальной настройки Microsoft SQL Server

Теоретическая часть:

Родоначальником серии SQL Server и его основой является язык запросов SQL. Данный язык был создан компанией IBM в начале 1970г. Изначально он назывался SEQVEL (Structured English Query Language) В основу языка SQL, используемого в SQL Server, легла разновидность языка T-SQL (Transact – SQL).

В начале 80 г. фирма IBM и ее подрядчики Microsoft и Sybase создают первую версию сетевой СУБД, которая называлась SQL Server версия 1.0, для операционной системы IBM OS/2. После этого под эту операционную систему

было выпущено еще 3 версии SQL Server. В середине 80-х г. компания Microsoft и Sybase отделяются от фирмы IBM, и Microsoft начинает работу над своей операционной системой Windows и вместе с компанией Sybase начинает развитие SQL Server.

В середине 90-х г. (в частности в 1995г) Microsoft создала операционную систему Windows NT и вместе с компанией Sybase выпускает первую версию SQL Server для Windows версии 4.1.

После этого компания Sybase разрывает свои отношения с Microsoft и Microsoft создает Microsoft SQL Server 6.0. Данная версия была предназначена для работы в операционной системе Windows NT, 95 и 98. В 1999г. выходит версия Microsoft SQL Server 7.0, которая стала одной из самых популярных серверных СУБД в мире. В 2000г. выходит 8-я версия Microsoft SQL Server 2000. В 2005 году выходит новая версия сервера, основанная на новой технологии NET, а в 2008 году выходит её улучшенная версия Microsoft SQL Server 2008.

Требования к аппаратному обеспечению

Минимальные:

- процессор: Intel (или совместимый) Pentium III 1000 МГц или выше;
- память: 512 МБ или более;
- жесткий диск: 20 ГБ или более.

Рекомендуемые:

- ~ процессор: Intel Pentium4 3000 МГц или выше;
- ~ память: 2 ГБ или более;
- ~ жесткий диск: 100 ГБ или более.

Требования к программному обеспечению

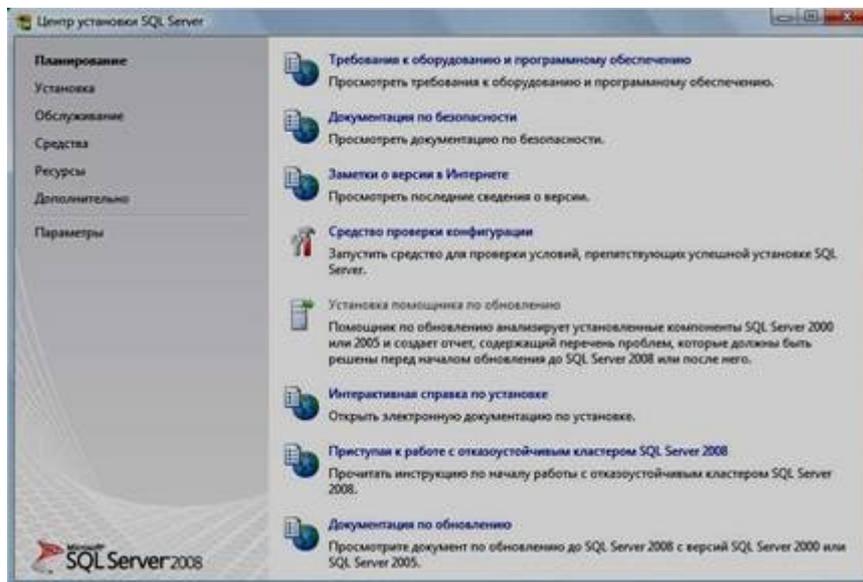
Необходимо наличие установленных пакетов: **Microsoft dotNET Framework 3.5 SP1, Windows Installer 4.5 и Windows PowerShell 1.0.**

Операционная система: MS Windows 2003 Server SP2 (Standard Edition, Enterprise Edition, Data Center Edition), MS Windows 2003 Small Business Server SP2 (Standard Edition, Premium Edition), MS Windows 2008 Server (Standard Edition, Enterprise Edition, Data Center Edition, Web Edition). Кроме того, некоторые редакции SQL Server 2008, в том числе бесплатную редакцию (Express Edition), можно устанавливать на следующие ОС: MS Windows XP SP2 (Home Edition, Professional Edition, Media Center Edition, Tablet Edition), MS Windows Vista (Home Basic Edition, Home Premium Edition, Business Edition, Enterprise Edition, Ultimate Edition).

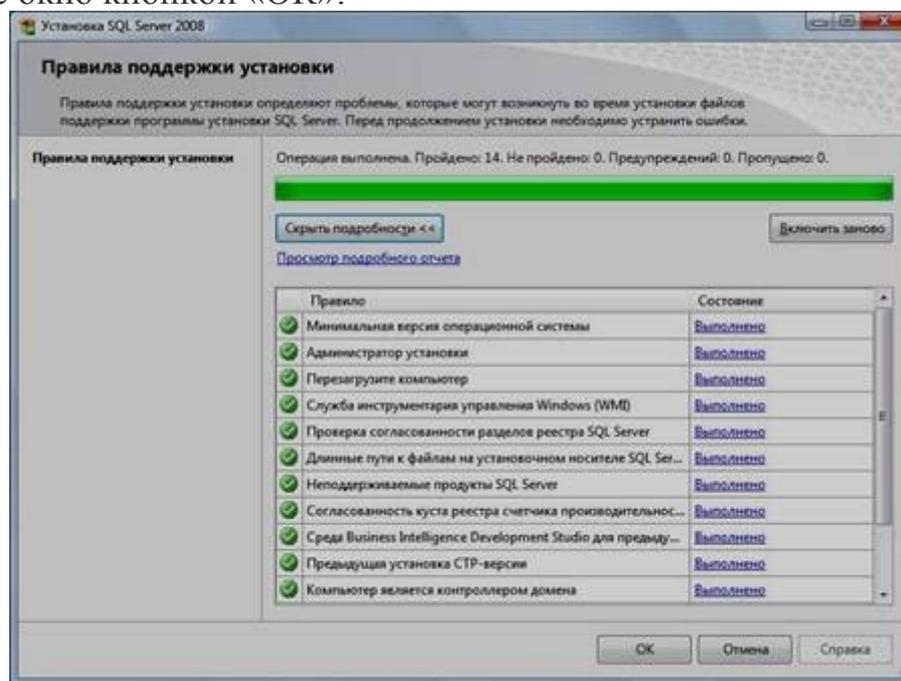
Примечание. 64-разрядные редакции SQL Server 2008 предъявляют другие требования к аппаратно-программному обеспечению.

Установка MS SQL Server 2008

1. Запустить программу-установщик (в бесплатной версии Express Edition обычно называется SQLEXPADV_x86_RUS.exe) с правами администратора на данном компьютере.
2. В разделе «Планирование» нажать пункт «Средство проверки конфигурации»:



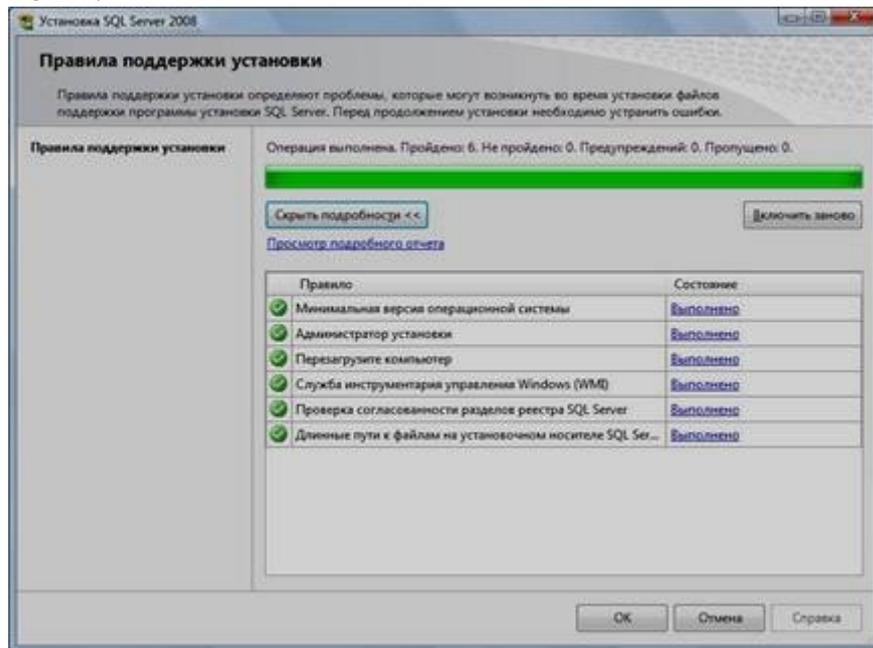
3. Нажать кнопку «Показать подробности» и убедиться, что все проверки успешно пройдены. Если будут обнаружены какие-то проблемы, то необходимо их устранить и запустить повторную проверку кнопкой «Включить заново». Затем закрыть данное окно кнопкой «ОК»:



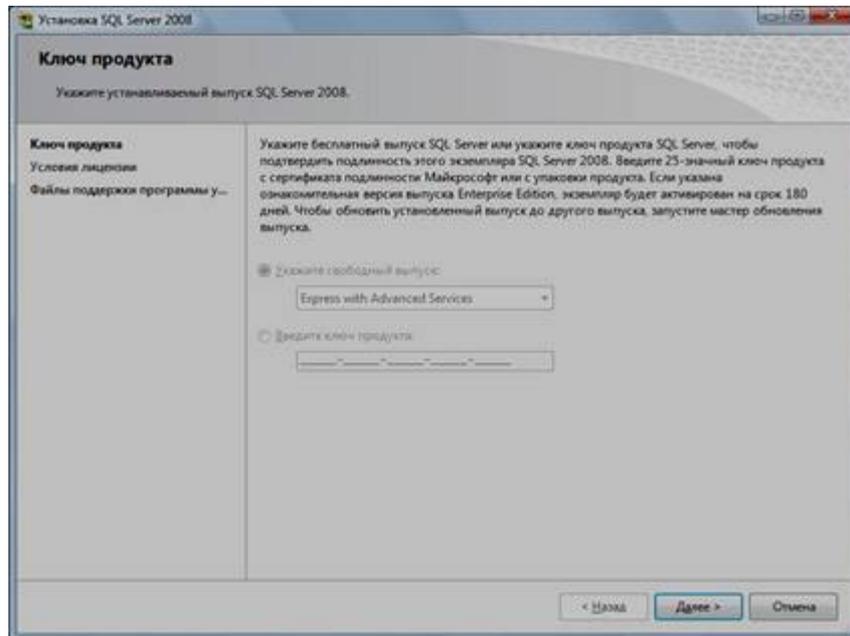
4. Нажать на раздел «Установка» и затем пункт «Новая установка изолированного SQL Server или добавление компонентов ...»:



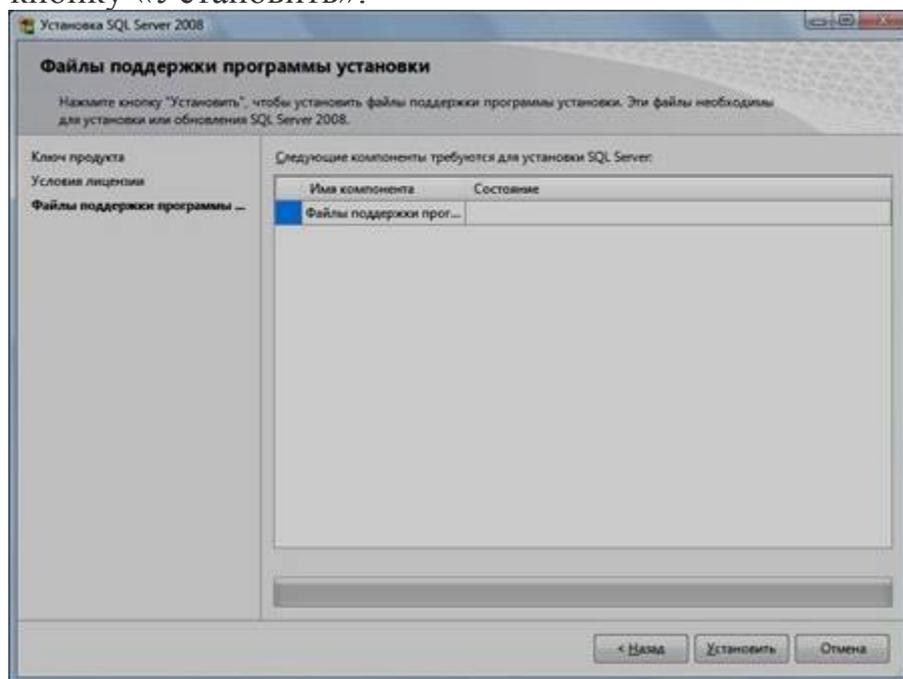
5. Нажать кнопку «Показать подробности» и убедиться, что все проверки успешно пройдены. Если будут обнаружены какие-то проблемы, то необходимо их устранить и запустить повторную проверку кнопкой «Включить заново». Затем нажать кнопку «ОК»:



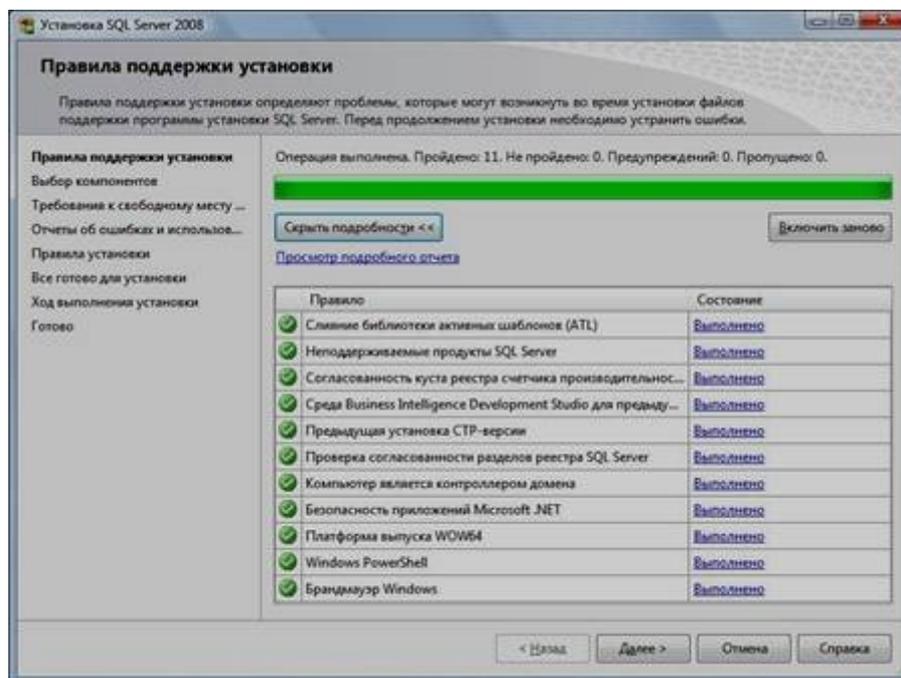
6. Ввести приобретенный ключ продукта (для бесплатной версии не требуется) и нажать кнопку «Далее»:



7. Прочитать лицензию, установить галочку и нажать кнопку «Далее»:
8. Нажать кнопку «Установить»:

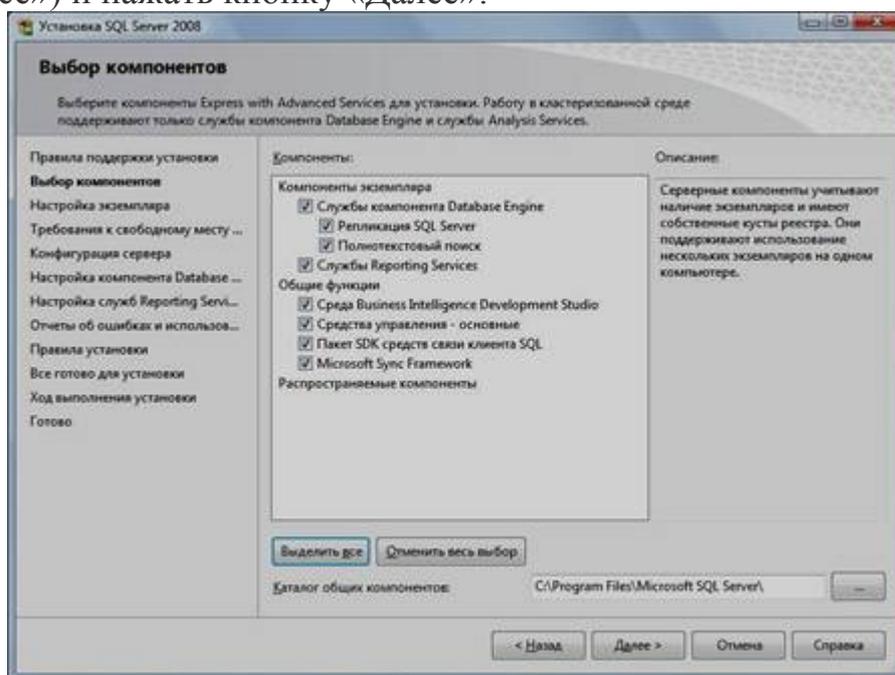


9. Нажать кнопку «Показать подробности» и убедиться, что все проверки успешно пройдены. Если будут обнаружены какие-то проблемы, то необходимо их устранить и запустить повторную проверку кнопкой «Включить заново». Затем нажать кнопку «Далее»:



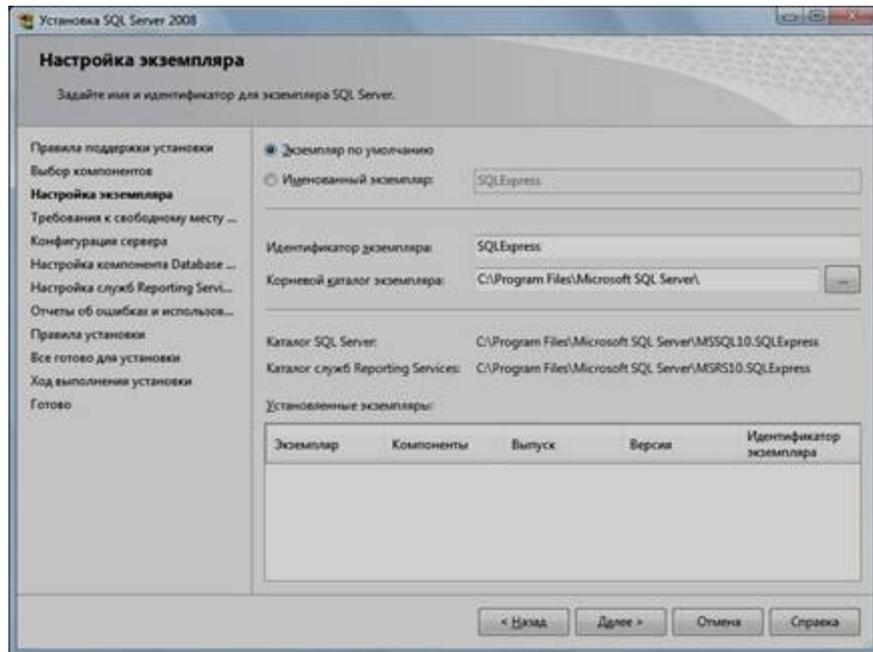
Примечание. Если появится предупреждение в строке «Брандмауэр Windows», то его можно проигнорировать – оно просто акцентирует Ваше внимание на том, что потребуется дополнительная настройка «Брандмауэра Windows» для доступа к SQL Server с других компьютеров (см. ниже).

10. Выбрать компоненты для установки (можно воспользоваться кнопкой «Выделить все») и нажать кнопку «Далее»:

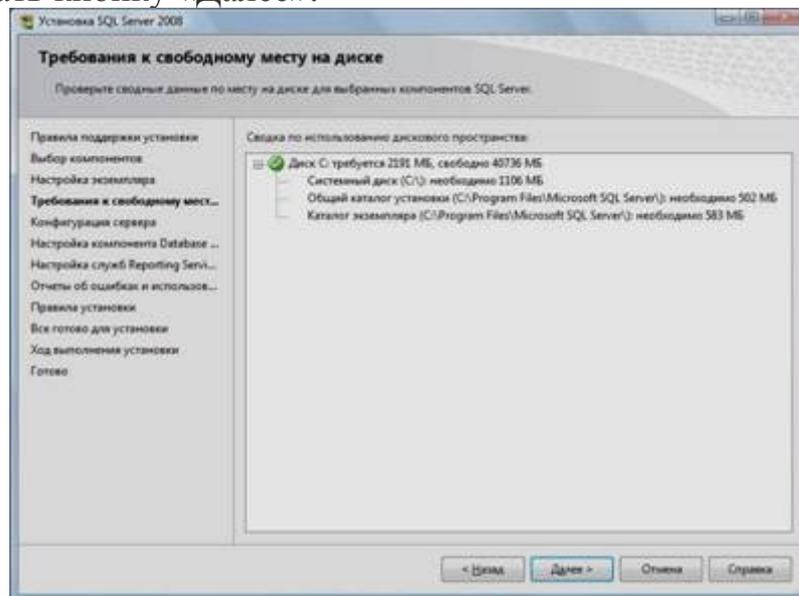


Внимание! Для нормального функционирования ПО «Альта ГТД-ПРО» необходимо обязательно установить компонент «**Полнотекстовый поиск**»! Кроме того, для управления самим SQL Server необходимо установить компонент «**Средства управления – основные**»!

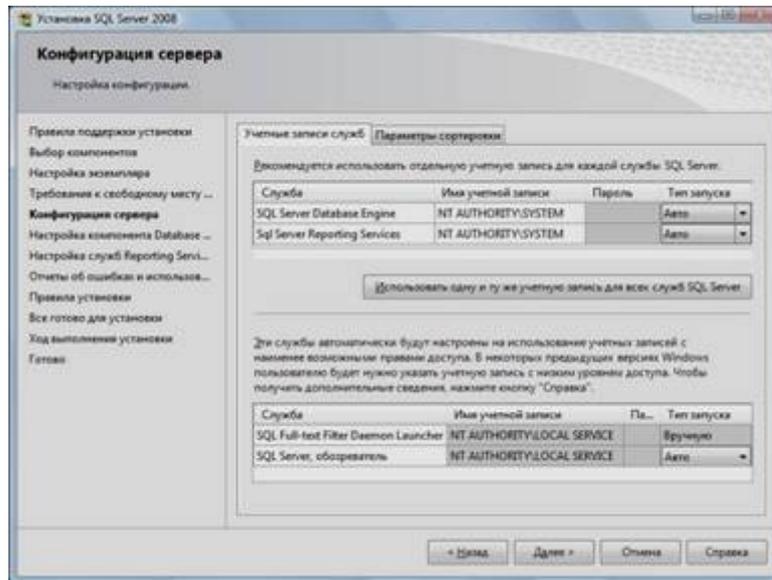
11. Выбрать опцию «Экземпляр по умолчанию» и нажать кнопку «Далее»:



12. Нажать кнопку «Далее»:

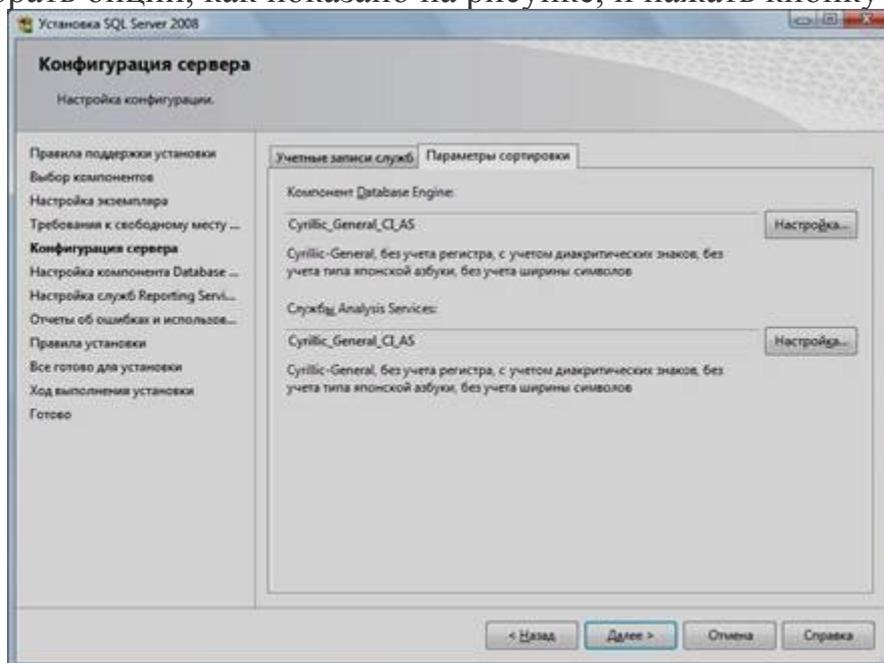


13. Выбрать опции, как показано на рисунке, и перейти на закладку «Параметры сортировки»:

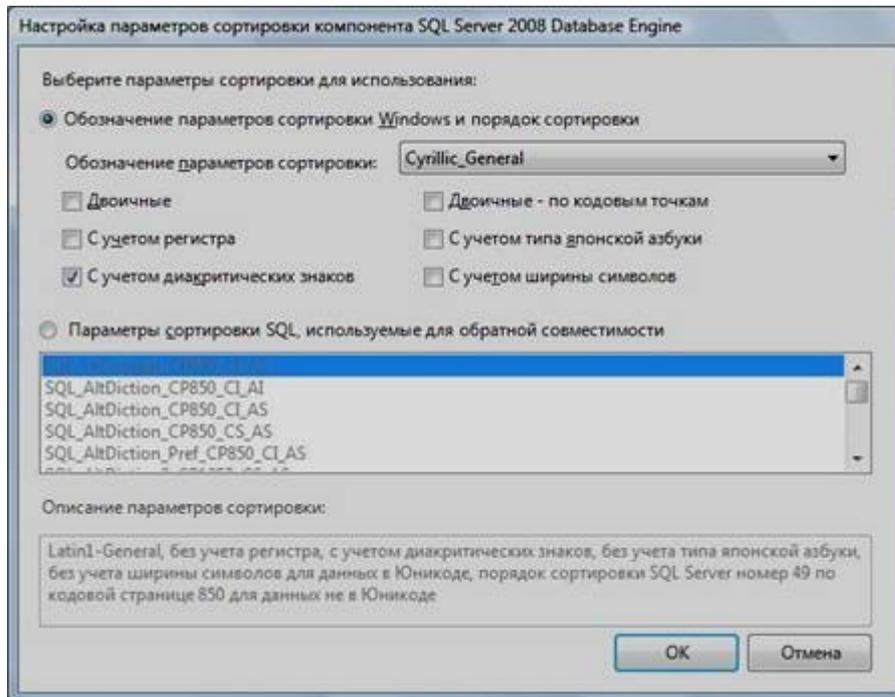


Примечание. Если Вы хотите использовать «SQL Server Agent» (встроенный планировщик заданий, в бесплатную версию не входит) для выполнения регулярного резервного копирования файлов БД на другой компьютер в своей локальной сети (рекомендуется) и Ваша сеть построена с использованием домена Windows NT, то необходимо завести в Вашем домене отдельную учетную запись для SQL Server Agent и предоставить ей права на соответствующие ресурсы (более подробную информацию можно найти в справочной системе SQL Server). При такой конфигурации в этом окне необходимо в поля «Имя учетной записи» и «Пароль» ввести имя созданной учетной записи (вида ДОМЕН\ИМЯ) и ее пароль, как минимум, для службы «SQL Server Agent».

14. Выбрать опции, как показано на рисунке, и нажать кнопку «Далее»:

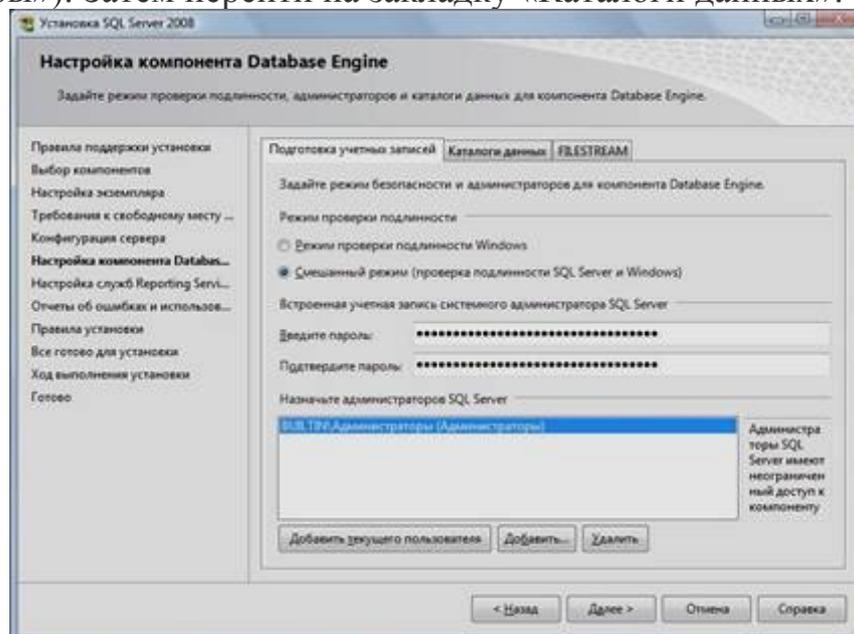


Примечание. Чтобы изменить опцию нажмите расположенную рядом кнопку «Настройка» и установите параметры, как показано на следующем рисунке:

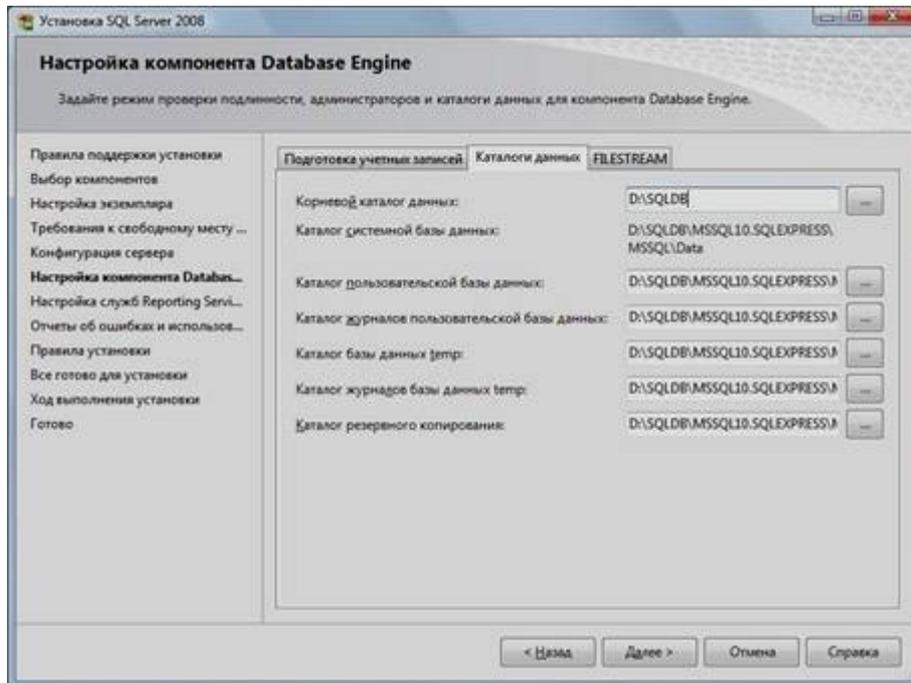


Внимание. Данную настройку нельзя будет изменить после установки. **Будьте внимательны!**

15. Выбрать опцию «Смешанный режим» и задать пароль для встроенной учетной записи администратора «sa» (эта учетная запись обладает максимальными правами доступа ко всем функциям и объектам на SQL-сервере). Дополнительно можно указать учетные записи пользователей Windows или целые группы пользователей Windows, которые должны обладать максимальными правами доступа к SQL Server (например, встроенную группу «Администраторы»). Затем перейти на закладку «Каталоги данных»:



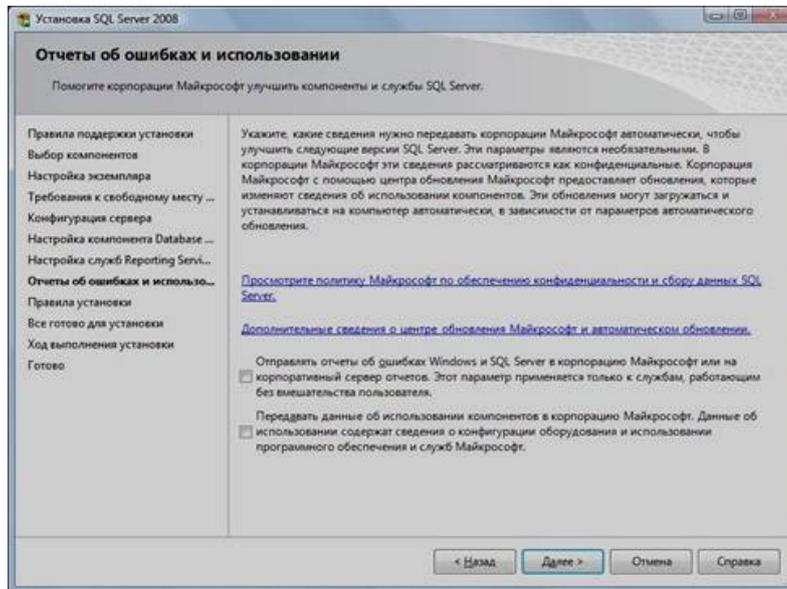
16. В поле «Корневой каталог данных» ввести путь к папке, где будут размещаться файлы баз данных (рекомендуется использовать отдельный от ОС физический диск), и нажать кнопку «Далее»:



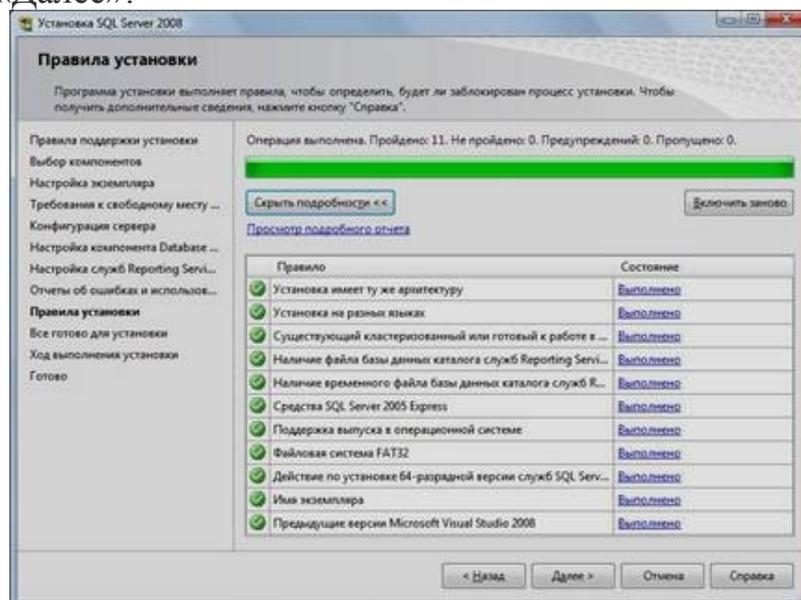
17. Выбрать опции, как показано на рисунке, и нажать кнопку «Далее»:



18. Выбрать опции, как показано на рисунке, и нажать кнопку «Далее»:

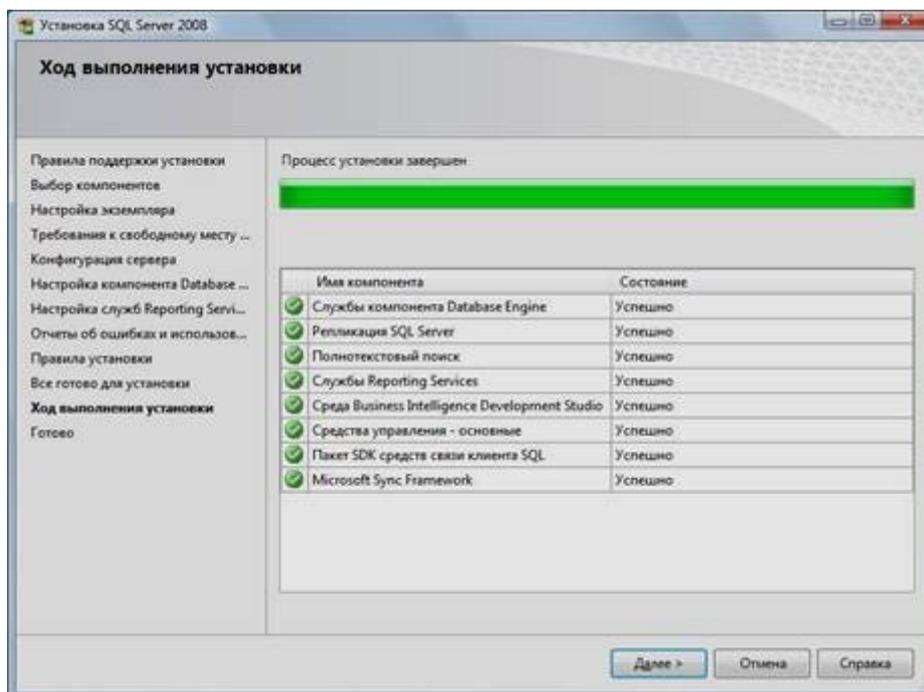


19. Нажать кнопку «Показать подробности» и убедиться, что все проверки успешно пройдены. Если будут обнаружены какие-то проблемы, то необходимо их устранить и запустить повторную проверку кнопкой «Включить заново». Затем нажать кнопку «Далее»:



20. Нажать кнопку «Установить».

21. После завершения установки нажать кнопку «Далее»:



22. Нажать кнопку «Закреть». На экране появится окно, сообщающее об успешной установке Microsoft SQL Server .

Практическая часть:

- 1 Установить в Windows виртуальную машину.
- 2 Установить Microsoft SQL Server на виртуальную машину.

Контрольные вопросы

1. Для чего предназначено программное обеспечение Microsoft SQL Server?
2. Как настраивать пользователей в среде Microsoft SQL Server?

Лабораторная работа № 16-17

Тема: Экспорт данных базы в документы пользователя

Цель работы: закрепить знания и умения по созданию базы данных, заполнению таблиц различными способами, научиться экспортировать информацию из базы данных.

Теоретическая часть

Импорт и присоединение таблиц из текстовых файлов

Access позволяет импортировать в базы данных табличные данные из обычных текстовых файлов. Эти файлы могут быть созданы либо одним из текстовых редакторов (например, приложением "Блокнот" в Windows), либо путем экспорта из другого приложения. Такая возможность особенно удобна в тех случаях, когда вам необходимы данные из приложения, которое не позволяет экспортировать данные из внутреннего представления в форматы DBF, WKS или XLS.

Существуют два основных формата текстовых файлов, в которых содержатся табличные данные:

- с полями фиксированной ширины — строки состоят из полей фиксированной ширины, пустые позиции при этом заполняются пробелами;

- с полями переменной ширины — со специальными символами — разделителями полей.

Для файлов с фиксированной шириной полей начало каждого поля определяется по его позиции относительно начала строки. Строки имеют одинаковую длину и разделяются символами новой строки (обычно это пара символов — возврат каретки и перевод строки).

Для файлов с разделителями полей существует несколько стандартных вариантов разделения полей. Они приведены в табл. 1.1.

Таблица 1.1. Форматы текстовых файлов, поддерживаемые Access

Формат	Описание
Записи в одну строку, поля разделены запятыми	Разделителем записей является символ новой строки . Некоторые приложения заключают все значения полей в двойные кавычки, другие приложения ограничиваются заключением в кавычки лишь текстовых значений, чтобы отличить их от числовых
Записи в одну строку, поля разделены знаками табуляции	Разделителем записей является символ новой строки. Предполагается, что все значения полей текстовые, а сами поля разделены знаками табуляции. Этот формат поддерживается большей частью текстовых процессоров
Записи в одну строку, поля разделены пробелами	Access позволяет использовать пробел в качестве символа разделителя полей, однако использование такого разделителя очень неудобно для файлов, содержащих текстовые данные (в которые также может входить пробел)

Процесс импорта данных, хранящихся в текстовом файле, выполняется с помощью Мастера импорта текстов. Выберите команду **Файл, Внешние данные, Импорт** и следуйте указаниям Мастера. Если файл содержит разделители полей, то укажите символ разделителя на втором шаге Мастера импорта текстов или выберите переключатель **фиксированная ширина полей**. Если в первой строке импортируемого файла содержатся заголовки полей, установите флажок **Первая строка содержит имена полей**

Практическая часть:

ИМЯ БД «ВЕДОМОСТЬ УСПЕВАЕМОСТИ»

Создать макеты таблиц, определив важные характеристики: (ключевое поле выделено полужирным начертанием)

Факультеты

Характеристики поля

Поле

Тип

поля

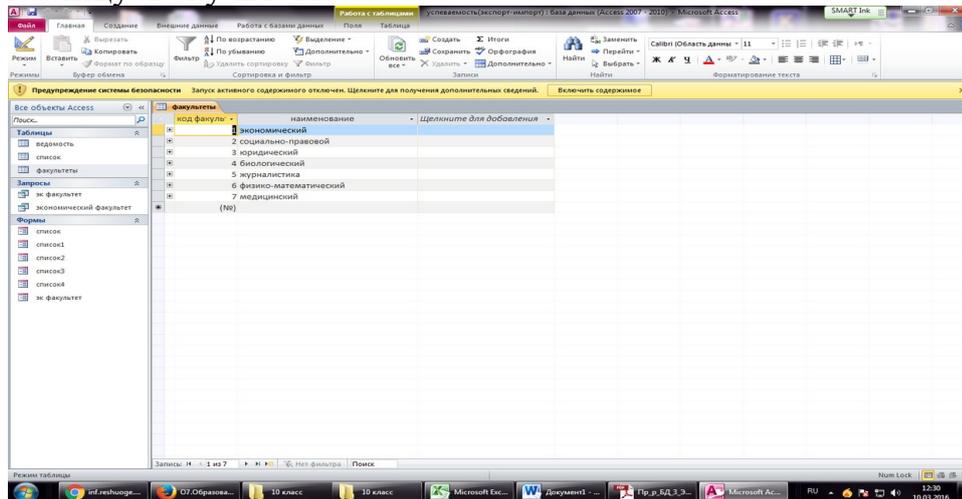
Код факультета

Счетчик

Наименование

Текстовый

Заполнить таблицу Факультеты



Создать макет таблицы список

Характеристики поля

Поле

Тип

поля

Списочный характер

Условие на значение

Сообщение об ошибке

№ студенческого билета

Текстовой

-

-

Фамилия

Текстовый

-

-

Имя

Текстовый

-

-

Отчество

Текстовый

-

-

Пол

Текстовый

-

-

Год рождения

Дата/время

-

-

Факультет

Мастер подстановок

Использовать поле «наименование» из таблицы Факультеты

-

Заполнить таблицу «Список»

№ студенческого билета

Фамилия

Имя

Отчество

пол

дата рождения

факультет

123-ж

Абакумова

Ирина

Александровна

ж

02.02.2000

журналистика

022-м

Авдеева

Юлия

Павловна

ж

01.10.2001

медицинский

025-м

Алексеев

Степан

Сергеевич

м

03.03.2000

медицинский

106-э

Афанасьева

Екатерина

Викторовна

ж

10.10.1999
экономический
152-ф
Богданова
Диана
Ильинична

ж
18.08.2002
физико-математический
122-ю
Воронов
Юрий
Николаевич

м
14.07.2001
юридический
100-ф
Григорьева
Елена
Николаевна

ж
25.07.2001
физико-математический
153-ф
Гурьева
Галина
Романовна

ж
27.06.1998
физико-математический
104-э
Жуков
Илья
Владимирович

м
11.02.2000
экономический
175-б
Иванова
Оксана
Олеговна

ж
24.05.1998
биологический
122-ж

Кондратьев
Иван
Александрович
м

12.01.2001
журналистика
122-б

Ледаков
Александр
Иванович
м

13.01.2001
биологический
147-с

Никитина
Дарья
Николаевна

ж
27.12.1998
социально-правовой
125-ю

Николаев
Алексей
Владимирович
м

04.05.1999
юридический
105-э

Новиков
Роман
Алексеевич

м
14.02.2000
экономический
125-б

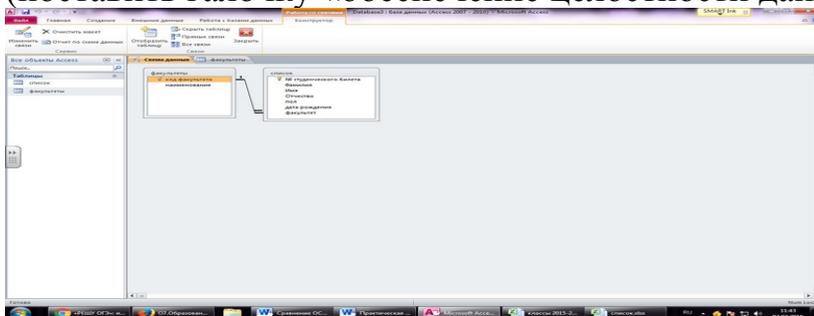
Орлов
Михаил
Иванович

м
12.12.1998
биологический
141-с

Осипов
Михаил
Юрьевич

М
14.03.2002
социально-правовой
024-м
Осипова
Наталья
Павловна
Ж
17.07.1999
медицинский
144-с
Павлова
Светлана
Владимировна
Ж
25.12.1998
социально-правовой
124-ж
Тимофеева
Александра
Игоревна
Ж
29.09.1999
журналистика
123-ю
Филиппова
Ольга
Евгеньевна
Ж
23.05.2001
юридический

Изменить в схеме данных связь между таблицами на «один-ко-многим»
(поставить галочку «обеспечение целостности данных»)



Открыть таблицу «список», далее вкладка «Внешние данные», нажать кнопку
«Экспорт в Excel»

№	Фамилия	Имя	Отчество	пол	дата рожде-	факультет
001-м	Абакумова	Ирина	Александровна	ж	02.02.2000	юридический
002-м	Алексеев	Олег	Павлович	м	03.03.2000	юридический
003-м	Алексеев	Степан	Сергеевич	м	03.03.2000	юридический
004-ф	Александрова	Виктория	Ильинична	ж	10.03.1999	юридический
005-ф	Волганова	Дина	Ильинична	ж	18.09.2002	Физико-математический
006-ф	Волгарева	Вера	Николаевна	ж	18.07.2003	юридический
007-ф	Григорьева	Алина	Николаевна	ж	25.07.2003	Физико-математический
008-ф	Гурова	Галина	Романовна	ж	27.06.1998	Физико-математический
009-ф	Иванова	Елена	Владимировна	ж	11.02.2000	юридический
010-ф	Иванова	Олеся	Олеговна	ж	24.05.1998	Биологический
011-ф	Исмаилов	Иван	Александрович	м	12.01.1999	юридический
012-ф	Иванов	Александр	Иванович	м	13.03.2003	Биологический
013-ф	Иванов	Дарья	Николаевна	ж	27.12.1999	социально-правовой
014-ф	Иванов	Владимир	Владимирович	м	04.05.1999	юридический
015-ф	Иванов	Роман	Александрович	м	14.02.2000	социально-правовой
016-ф	Орлов	Михаил	Иванович	м	12.12.1998	Биологический
017-ф	Ситников	Роман	Иванович	м	14.02.2000	социально-правовой
018-ф	Степанов	Наталья	Павловна	ж	17.07.1999	юридический
019-ф	Павлова	Светлана	Владимировна	ж	25.12.1998	социально-правовой
020-ф	Тимофеева	Александра	Игоревна	ж	29.09.1999	юридический
021-ф	Филиппова	Ольга	Евгеньевна	ж	23.05.2003	юридический

Выбрать необходимые параметры для экспорта документа: указать свою папку в качестве места, куда будет экспортирован файл.

Поставить галочку «Экспортировать данные с макетом и форматированием»

Выборите место назначения для экспортируемых данных

Укажите имя и формат целевого файла.

Имя файла: C:\Users\JaravinaAG\Desktop\список.xlsx

Формат файла: Книга Excel (*.xlsx)

Укажите параметры экспорта.

Экспортировать данные с макетом и форматированием.
Выберите этот параметр для сохранения сведений о форматировании и макете при экспорте таблицы, запроса, формы или отчета.

Открыть целевой файл после завершения операции экспорта.
Выберите этот параметр для просмотра результатов операции экспорта. Параметр доступен только при экспорте форматированных данных.

Экспортировать только выбранные записи.
Выберите этот параметр для экспорта только выбранных записей. Параметр доступен только при экспорте форматированных данных и наличии выбранных записей.

OK Отмена

Открыть экспортированный документ

Удалить столбцы «Пол», «дата рождения», «Фамилия», «Имя», «Отчество»

Добавить столбцы и заполнить таблицу по образцу, представленному ниже (менять направление текста для наименования дисциплины не нужно!)

№ студенческого билета

математика

физика

русский язык

литература

экономика

юриспруденция

биология

химия

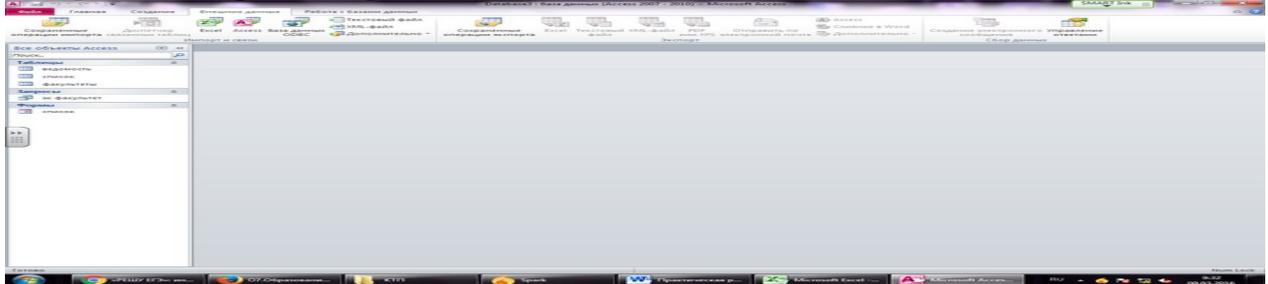
право

Сохранить документ в своей папке под именем «Ведомость»

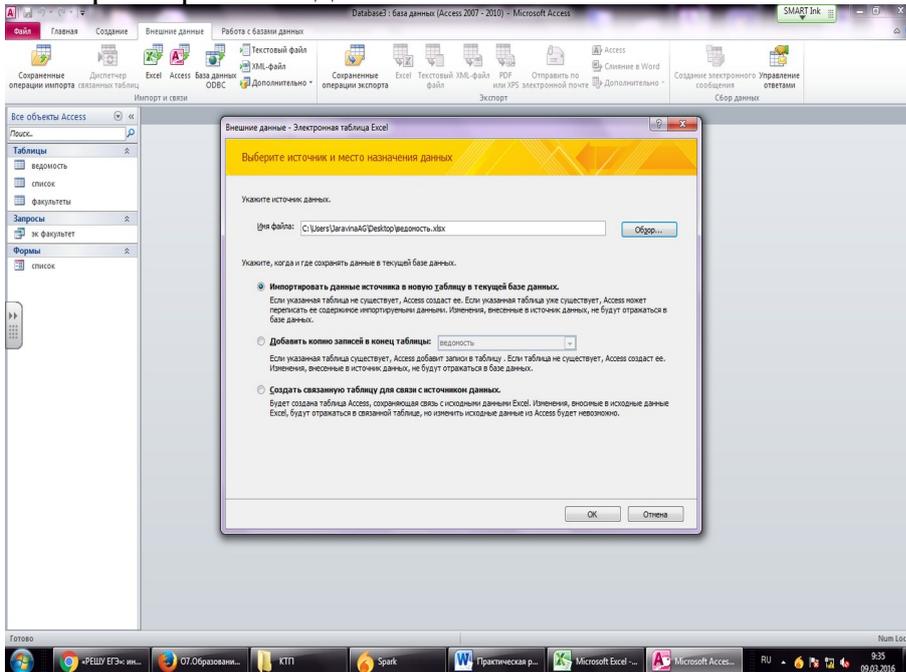
Вернуться в БД

Перейти на вкладку «Внешние данные»

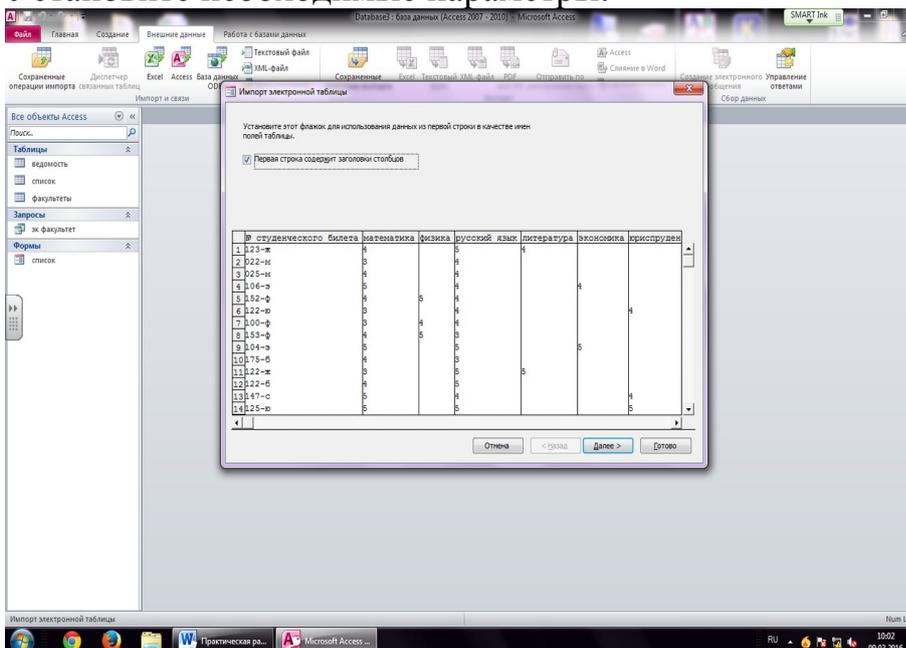
Выбрать импорт электронной таблицы



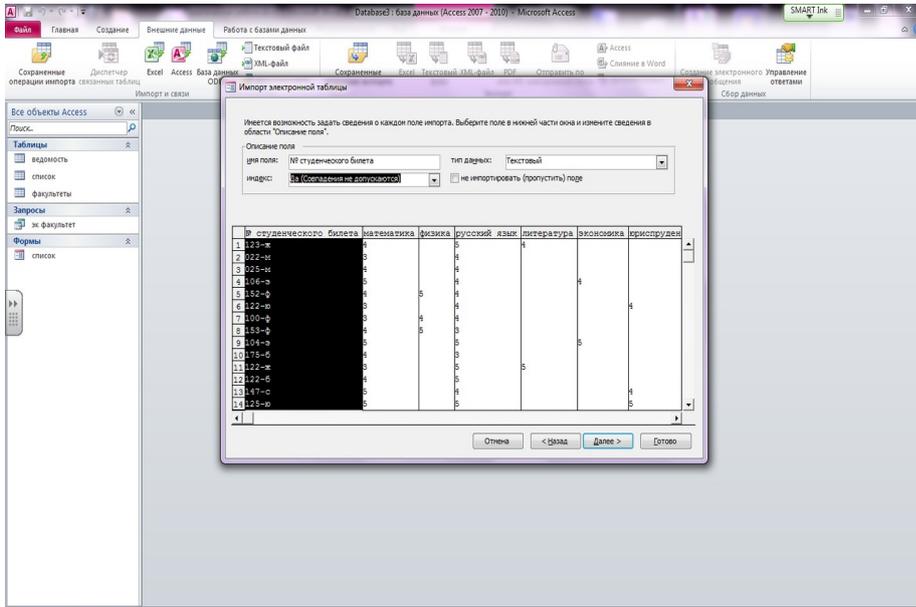
Выберите файл «ведомость» из своей папки



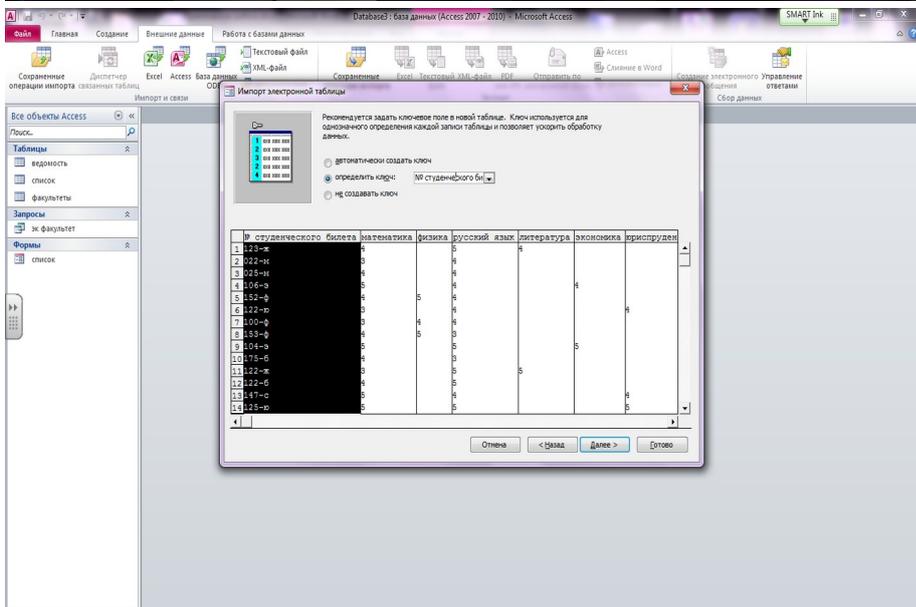
Установите необходимые параметры:



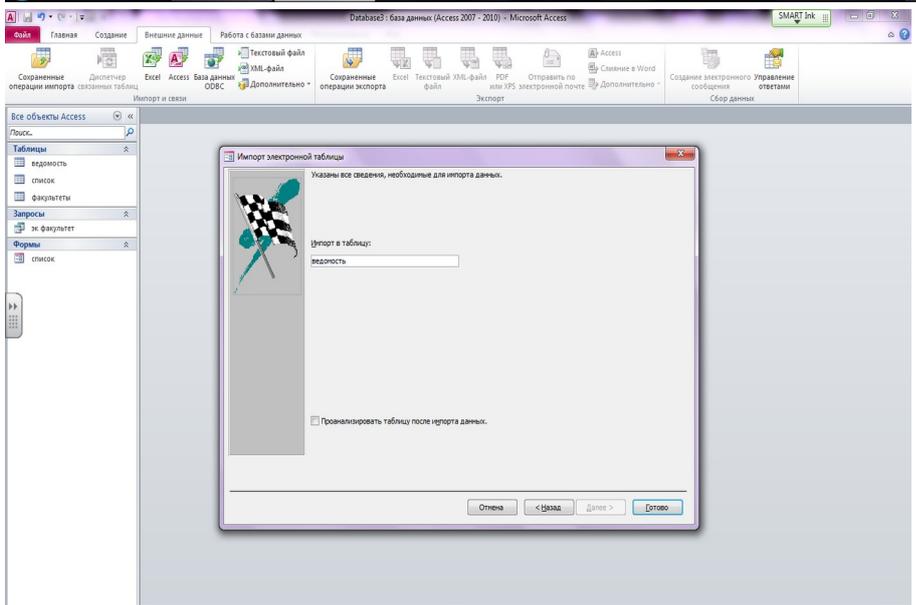
1.



2.

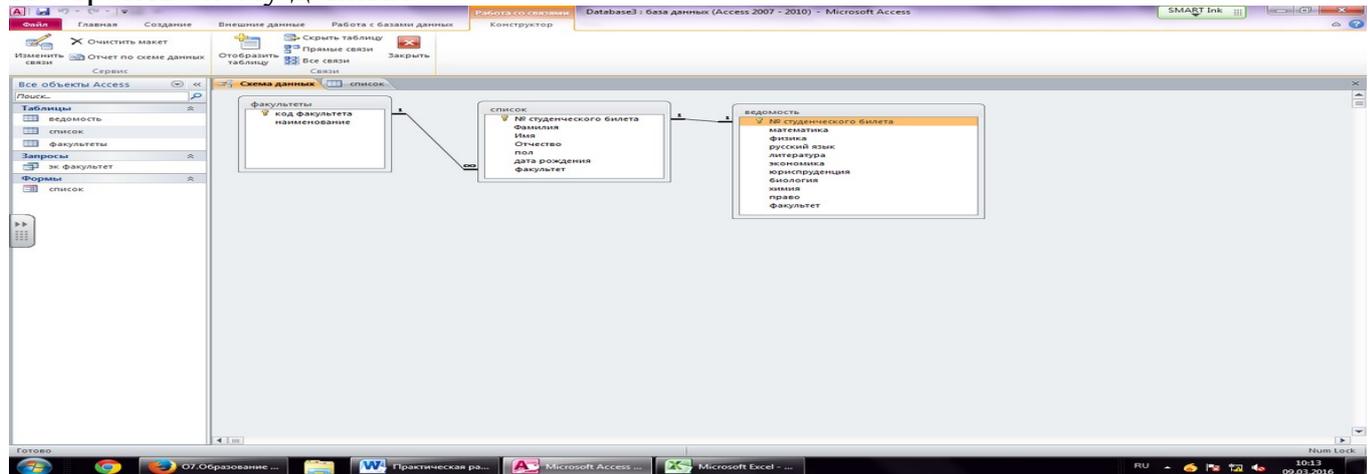


3.

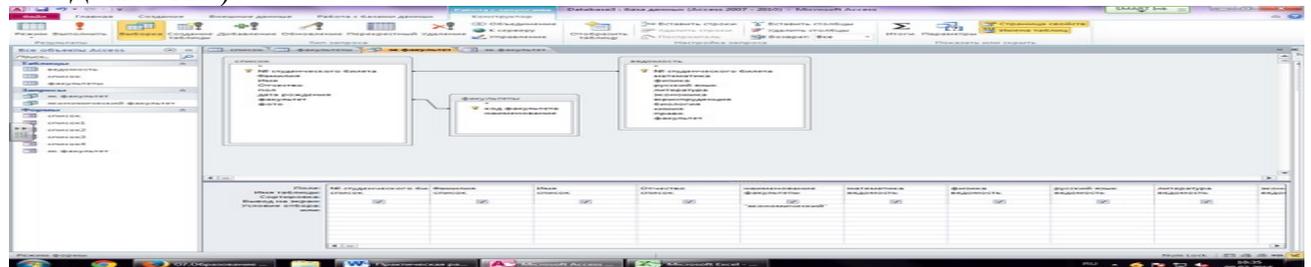


4.

Настроить схему данных

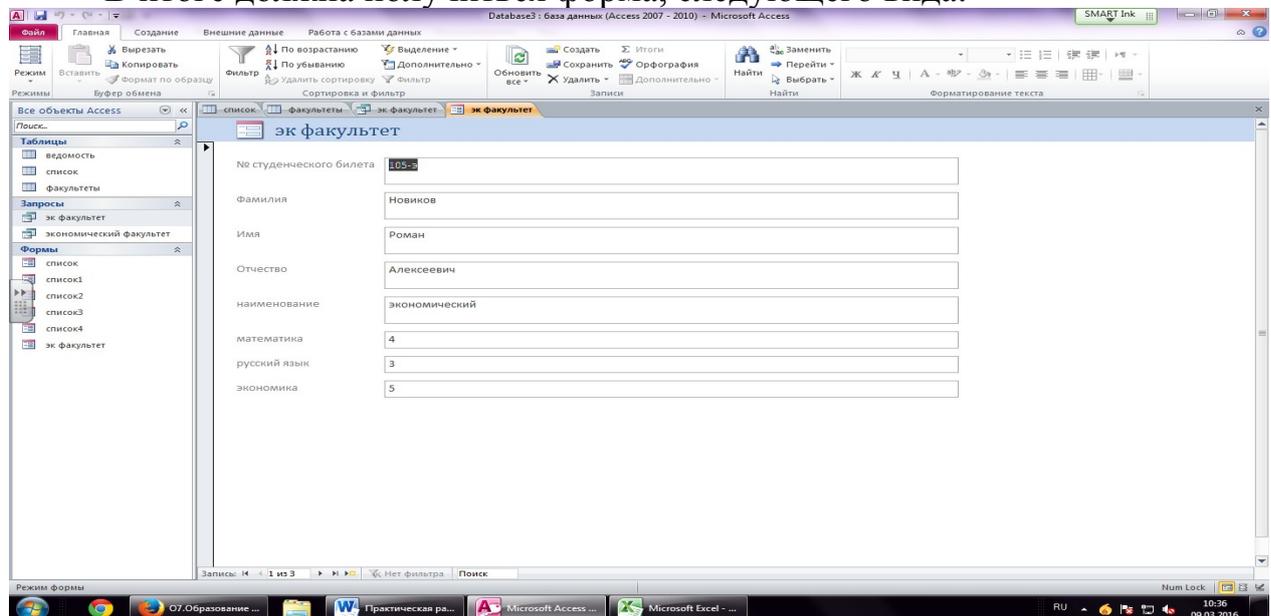


Создать запрос на отбор всех студентов экономического факультета. Запрос содержит наименование факультета, фамилию, имя, отчество студента, оценки за экзамены (в запрос добавляются все наименования дисциплин из таблицы «Ведомость»).



По созданному запросу создать отчет. В режиме Конструктора удалить лишние объекты.

В итоге должна получиться форма, следующего вида:



Лабораторная работа № 18

Тема: Импорт данных пользователя в базу данных

Цель работы: закрепить знания и умения по созданию базы данных, заполнению таблиц различными способами, научиться импортировать информацию в базы данных.

Теоретическая часть:

Импорт и присоединение таблиц из текстовых файлов

Access позволяет импортировать в базы данных табличные данные из обычных текстовых файлов. Эти файлы могут быть созданы либо одним из текстовых редакторов (например, приложением "Блокнот" в Windows), либо путем экспорта из другого приложения. Такая возможность особенно удобна в тех случаях, когда вам необходимы данные из приложения, которое не позволяет экспортировать данные из внутреннего представления в форматы DBF, WKS или XLS.

Существуют два основных формата текстовых файлов, в которых содержатся табличные данные:

- с полями фиксированной ширины — строки состоят из полей фиксированной ширины, пустые позиции при этом заполняются пробелами;
- с полями переменной ширины — со специальными символами — разделителями полей.

Для файлов с фиксированной шириной полей начало каждого поля определяется по его позиции относительно начала строки. Строки имеют одинаковую длину и разделяются символами новой строки (обычно это пара символов — возврат каретки и перевод строки).

Для файлов с разделителями полей существует несколько стандартных вариантов разделения полей. Они приведены в табл. 1.1.

Таблица 1.1. Форматы текстовых файлов, поддерживаемые Access

Формат	Описание
Записи в одну строку, поле разделены запятыми	Разделителем записей является символ новой строки . Некоторые приложения включают все значения полей в двойные кавычки, другие приложения ограничиваются заключением в кавычки лишь текстовых значений, чтобы отличить их от числовых
Записи в одну строку, поле разделены знаками табуляции	Разделителем записей является символ новой строки. Предполагается, что все значения полей текстовые, а сами поля разделены знаками табуляции. Этот формат поддерживается большей частью текстовых процессоров
Записи в одну строку, поле разделены пробелами	Access позволяет использовать пробел в качестве символа разделителя полей, однако использование такого разделителя очень неудобно для файлов, содержащих текстовые данные (в которые также может входить пробел)

Процесс импорта данных, хранящихся в текстовом файле, выполняется с помощью Мастера импорта текстов. Выберите команду **Файл, Внешние данные, Импорт** и следуйте указаниям Мастера. Если файл содержит разделители полей, то укажите символ разделителя на втором шаге Мастера импорта текстов или выберите переключатель **фиксированная ширина полей**. Если в первой строке

импортируемого файла содержатся заголовки полей, установите флажок **Первая строка содержит имена полей**

Практическая часть

Задание 1.

Создайте следующие таблицы. Таблица Покупатели – текстовый файл **таблица1.txt** (в качестве разделителя можно выбрать знак табуляции), таблица Заказы – электронная таблица **Заказы**. Импортируйте созданные таблицы (команда Внешние данные/Текстовый файл и Внешние данные/Excel) в базу данных **База5** создайте в ней таблицу **Продавцы**. Задайте ключевые поля (если они не заданы), установите связи, обеспечивающие целостность данных.

Таблица 1.1 Продавцы

Номер	Имя	Город	Комиссионные
1001	Дугинов	Липецк	0,12
1002	Ольшевский	Саратов	0,13
1004	Помякушина	Липецк	0,11
1007	Шикин	Бронницы	0,15
1003	Борескова	Новгород	0,1

Таблица 1.2 Покупатели

Номер	Имя	Город	Рейтинг	Номер_продавца
2001	Голубев	Липецк	100	1001
2002	Дмитриева	Архангельск	200	1003
2003	Кустов	Саратов	200	1002
2004	Кузьминов	Подольск	300	1002
2006	Кречко	Липецк	100	1001
2008	Фролов	Саратов	300	1007
2007	Каратыгин	Архангельск	100	1004

Таблица 1.3 Заказы

Номер	Стоимость	Дата_покупки	Номер_покупателя	Номер_продавца
3001	18,69	10/03/1996	2008	1007
3003	767,19	10/03/1996	2001	1001
3002	1900,10	10/03/1996	2007	1004
3005	5160,45	10/03/1996	2003	1002
3006	1098,16	10/03/1996	2008	1007
3009	1713,23	10/04/1996	2002	1003
3007	75,75	10/04/1996	2004	1002
3008	4723,00	10/05/1996	2006	1001
3010	1309,95	10/06/1996	2004	1002
3011	9891,88	10/06/1996	2006	1001

Задание 2.

Создайте еще три произвольных таблицы в Access, Excel и текстовом файле и присоедините их к существующей базе используя связь с таблицами.

Контрольные вопросы

1. Особенности форматов текстовых файлов при присоединении.
2. Особенности присоединения и импорта таблиц баз данных Access.
3. Что будет отображено в таблицах после присоединения электронной таблицы, содержащей формулы.
4. Как происходит определение типов полей импортируемой таблицы.

Лабораторная работа № 19

Тема: Выполнение настроек для автоматизации обслуживания базы данных

Цель работы: изучить систему основных компонентов Microsoft SQL Server 2008, понять процесс создания файла данных, освоить управление базами данных при помощи команд языка T-SQL.

Теоретическая часть

Основные компоненты Microsoft SQL Server 2008

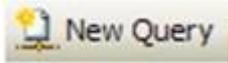
Все компоненты Microsoft SQL Server 2008 запускаются из меню «Пуск \ Программы \ Microsoft SQL Server 2008». В Microsoft SQL Server 2008 входят следующие компоненты:

- 1) Deployment Wizard – мастер по выводу информации хранимой на сервере;
- 2) SQL Server Installation Center – центр установки SQL Server 2008;
- 3) Reporting Services Configuration Manager – менеджер службы настройки отчётов;
- 4) SQL Server Configuration Manager – менеджер настройки сервера;
- 5) SQL Server Error and Usage Reporting – служба протоколирования работы сервера и служба отчётов об ошибках;
- 6) Microsoft Samples Overview – ссылка на сайт корпорации Microsoft, где можно просмотреть примеры работы с сервером;
- 7) SQL Server Books Online – полная справочная система по Microsoft SQL Server 2008. Она содержит справки, как по программированию, так и по администрированию сервера;
- 8) SQL Server Tutorials – учебники по работе с сервером;
- 9) Data Profile Viewer – просмотр профилей по работе с данными;
- 10) Execute Package Utility – инструменты по сжатию данных;
- 11) Database Engine Tuning Advisor – мастер настройки ядра базы данных;
- 12) SQL Server Profiler – настройка профилей по работе с данными;
- 13) Import and Export Data – импорт и экспорт данных;
- 14) SQL Server Business Intelligence Development Studio – интегрированная среда разработки Business Intelligence Development Studio;
- 15) SQL Server Management Studio – графическая оболочка для управления сервером и разработки баз данных.

Создание файла данных

Новую БД можно создать, используя стандартные команды языка T-SQL. Для создания новой БД необходимо сделать активную БД «Master». Это можно сделать либо выбором ее из выпадающего списка БД на панели инструментов, либо набором команды USE Master на вкладке нового запроса.

Замечание: все команды языка T-SQL набираются на вкладке нового запроса (SQLQuery). Для того чтобы создать новый запрос на панели инструментов необходимо нажать кнопку



Для выполнения команд языка T-SQL на панели инструментов необходимо нажать кнопку



или на вкладке нового запроса набрать команду GO.

Замечание: В Microsoft SQL Server БД состоит из двух частей:

- файл данных – файл, имеющий расширение mdf и где находятся все таблицы и запросы;

- файл журнала транзакций – файл, имеющий расширение ldf, содержит журнал, где фиксируются все действия с БД. Данный файл предназначен для восстановления БД в случае её выхода из строя.

Для создания нового файла данных используется команда CREATE DATABASE, которая имеет следующий синтаксис:

```
CREATE DATABASE <Имя БД>  
(Name=<Логическое имя>, FileName=<Имя файла>  
[Size=<Нач.размер>],[Maxsize=<Макс.размер>],[FileGrowth=<Шаг>])  
[LOG ON  
(Name=<Логическое имя>, FileName=<Имя файла>  
[Size=<Нач.размер>],[Maxsize=<Макс.размер >],[FileGrowth=<Шаг>])
```

Здесь:

- Имя БД – имя создаваемой БД,
- Логическое имя – определяет логическое имя файла данных БД, по которому происходит обращение к файлу данных,
- Имя файла – определяет полный путь к файлу данных,
- Нач.размер – начальный размер файла данных в Мб,
- Макс.размер – максимальный размер файла данных в Мб,
- Шаг – шаг увеличения файла данных, либо в Мб либо в %.

Параметры в разделе LOG ON аналогичны параметрам в разделе CREATE DATABASE. Однако они определяют параметры журнала транзакций.

Управление базами данных при помощи команд языка T-SQL

В языке запросов T-SQL с БД возможны следующие действия:

- 1) отображение сведений о БД: EXEC sp_helpdb <Имя БД>;
- 2) изменение параметров БД: EXEC sp_dboption <Имя БД>, <Параметр>, <Значение>;
- 3) добавления новых файлов, удаление файлов и переименования файлов, входящих в БД:

```
ALTER DATABASE <Имя БД>
```

ADD FILE (<Параметры>)|
REMOVE FILE <Логическое имя файла>|
MODIFY FILE (<Параметры>)

где раздел ADD FILE добавляет файл, REMOVE FILE удаляет, а раздел MODIFY FILE изменяет параметры файла;

- 4) сжатие всей БД: DBCC SHRINKDATABASE <Имя БД>;
- 5) сжатие конкретного файла БД: DBCC SHRINKFILE <Логическое имя файла>;
- 6) переименование БД: EXEC SP_RENAMEDB <Имя БД>, <Новое имя БД>;
- 7) удаление БД: DROP DATABASE <Имя БД>.

Замечание: вышеперечисленные команды используют следующие параметры:

- <Имя БД> - имя БД с которой производится действие;
- <Параметр> - изменяемый параметр;
- <Значение> - новое значение изменяемого параметра;
- <Параметры> - параметры файла БД, аналогичные параметрам, используемым в команде CREATE DATABASE;
- <Логическое имя файла> - логическое имя файла, входящего в БД;
- <Новое имя БД> - новое имя БД.

Для запуска среды разработки «SQL Server Management Studio» в меню «Пуск» выбираем пункт «Программы\Microsoft SQL Server 2008\SQL Server Management Studio».

После запуска среды разработки появится окно подключения к серверу «Connect to Server».



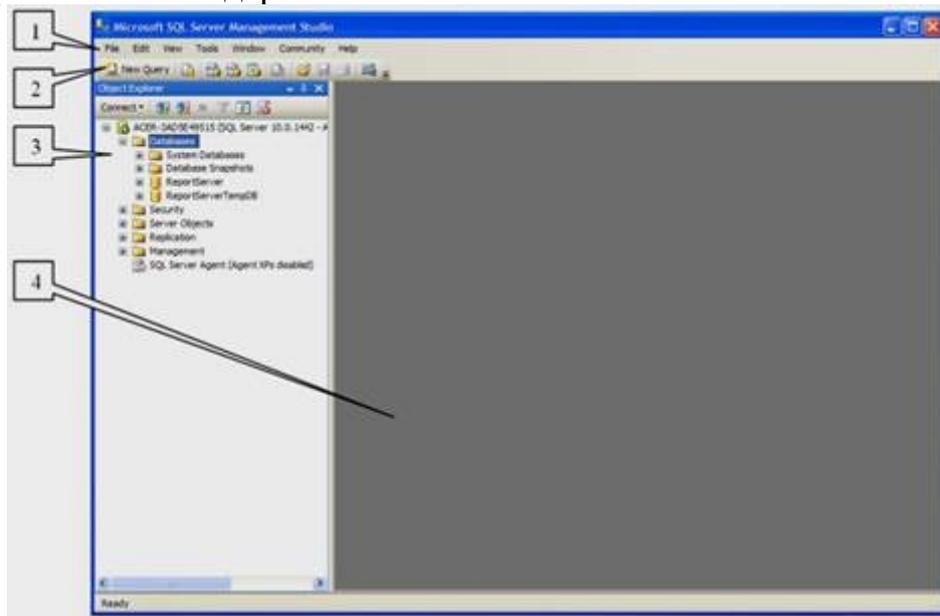
В этом окне необходимо нажать кнопку «Connect»

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» (см. рисунок 2.2).

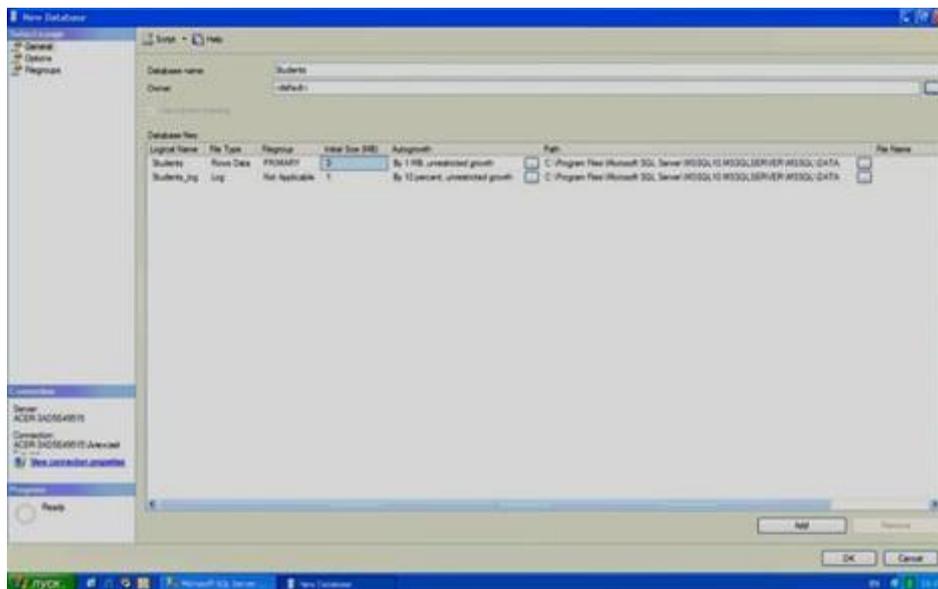
Данное окно имеет следующую структуру:

- 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 ее содержимое.



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

Перейдем непосредственно к созданию файла данных. Для этого в обозревателе объектов щелкните ПКМ на папке «Databases» (Базы данных) (см. рисунок. 2.2) и в появившемся меню выберите пункт «New Database» (Новая БД). Появится окно настроек параметров файла данных новой БД «New Database» (см. рисунок 2.3). В левой части окна настроек имеется список «Select a page». Этот список позволяет переключаться между группами настроек.



Настроим основные настройки «General». Для выбора основных настроек нужно просто щелкнуть мышью по пункту «General» в списке «Select a page». В правой части окна «New Database» появятся основные настройки.

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

1) Logical Name – логическое имя файла данных и журнала транзакций. По этим именам будет происходить обращение к вышеприведенным 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»;

2) File Type – тип файла. Этот параметр показывает, является ли файл файлом данных или журналом транзакций;

3) Filegroup – группа файлов, показывает к какой группе файлов относится файл. Группы файлов настраиваются в группе настроек «Filegroups»;

4) Initial Size (MB) – начальный размер файла данных и журнала транзакций в мегабайтах;

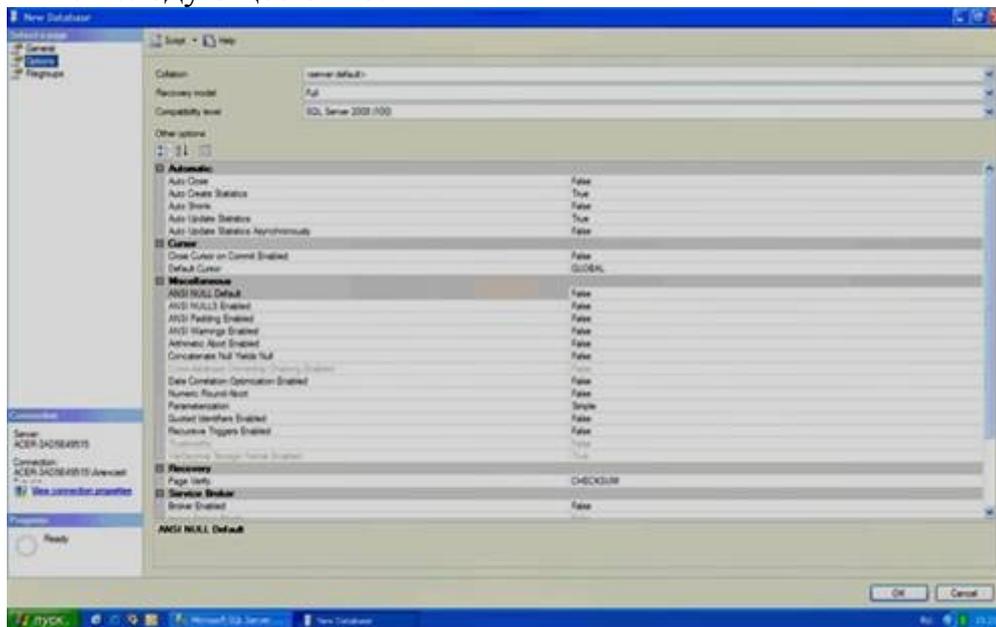
5) Autogrowth – автоувеличение размера файла. Как только файл заполняется информацией его размер автоматически увеличивается на величину, указанную в параметре «Autogrowth». Увеличение можно задавать как в мегабайтах так и в процентах. Здесь же можно задать максимальный размер файлов. Для изменения этого параметра надо нажать кнопку «...»;

6) Path – путь к папке, где хранятся файлы. Для изменения этого параметра также надо нажать кнопку «...»;

7) 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».

Рассмотрим второстепенные настройки файла данных. Для доступа к этим настройкам необходимо щелкнуть мышью по пункту «Options» в списке «Select a page». Появится следующее окно.



В правой части окна мы видим следующие настройки:

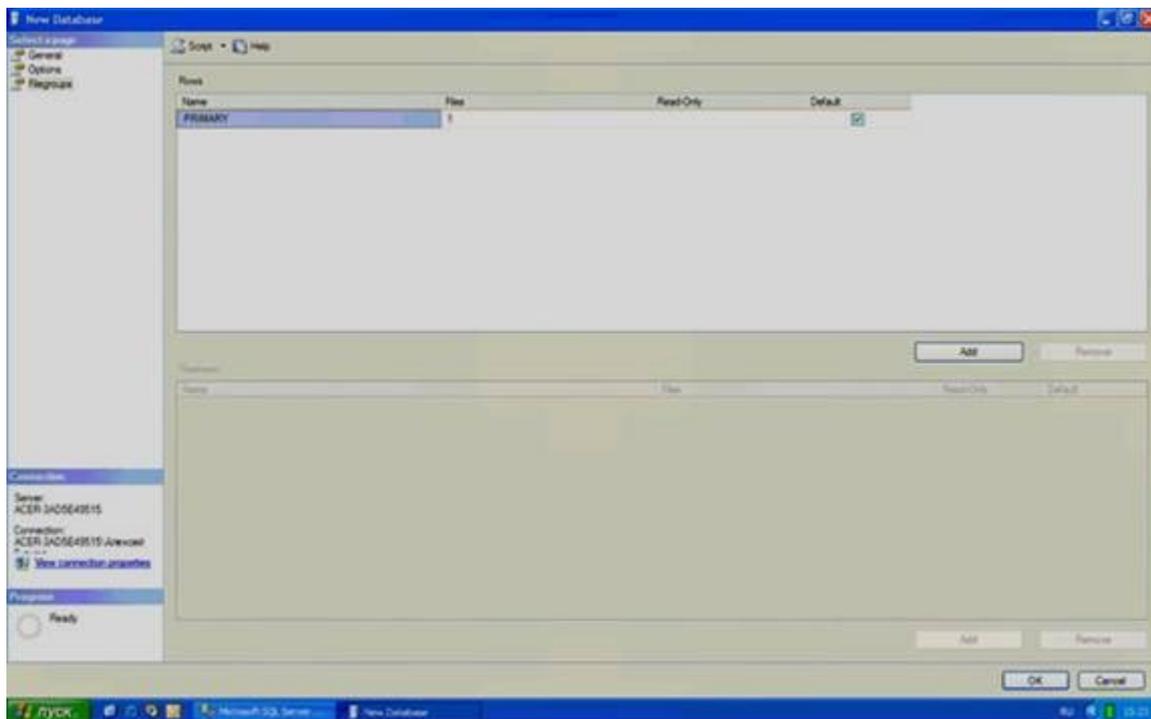
1) Collation – этот параметр отвечает за обработку текстовых строк, их сравнение, текстовый поиск и т.д. Рекомендуется оставить его как «<server default>». При этом данный параметр будет равен значению, заданному на вкладке «Collation», при установке сервера;

2) Recovery Model – модель восстановления. Данный параметр отвечает за информацию, предназначенную для восстановления БД, хранящуюся в файле транзакций. При наличии места на диске рекомендуется оставить этот параметр в значении «Full»;

3) Compatibility level – уровень совместимости, определяет совместимость файла данных с более ранними версиями сервера. Если планируется перенос данных на другую, более раннюю версию сервера, то ее необходимо указать в этом параметре;

4) Other options – второстепенные параметры. Данные параметры являются необязательными для изменения.

Рассмотрим последнюю группу настроек «Filegroups». Данная группа настроек отвечает за группы файлов. Для ее отображения в списке «Select a page» необходимо щелкнуть мышью по пункту «Options». Отобразятся настройки групп файлов.



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

- 1) Name – имя группы файлов.
- 2) Files – количество файлов входящих в группу.
- 3) Read only – файлы в группе будут только для чтения, их можно только просматривать, но нельзя изменять.
- 4) 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».

Для переименования БД необходимо в обозревателе объектов щелкнуть 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».

Практическая часть:

Создать БД «Students», расположенную в файле «D:\Students.mdf» и имеющую начальный размер файла данных 1мб., максимальный размер файла данных 100мб. И шаг увеличения файла данных, равный 1мб. Файл журнала транзакций данной БД имеет имя «StudentsLog» и расположен в файле «D:\Students.ldf». Данный файл имеет начальный размер, равный 1мб., максимальный размер равный 100мб. И шаг увеличения, равный 1мб.

Контрольные вопросы

1. Перечислите основные компоненты Microsoft SQL Server
2. Как установить первоначальный размер файла базы данных?

Лабораторная работа № 20-21

Тема: Мониторинг работы сервера

Цель работы: формирование умений:

- Контролировать производительность сервера с помощью программы WindowsSystemMonitor;
- Контролировать производительность сервера с помощью SQLProfiler;
- Наблюдать план выполнения запроса, используя SQL Server Query Analyzer ;
- Контролировать блокировки, используя системные хранимые процедуры и программу SQL Server Enterprise Manager .

Практическая часть

1. Использование программы Windows SystemMonitor

Создайте базу данных NorthwindCopy. Для этого восстановите резервную копию базы данных NorthwindCopy из файла C:\MOC\2072a\Labfiles\L08\NorthwindCopy.bak.

Сконфигурируйте Windows System Monitor. Для этого выполните следующие действия:

- запустите программу «Системный монитор» (Мой компьютер /Администрирование /Системный монитор);
- на панели инструментов щелкните по кнопке «Добавить» (+);
- в окне диалога добавьте счетчики, используя информацию табл.1. В конце закройте окно кнопкой «Закреть».

Примечание. Основной набор счетчиков программы «Системный монитор» приведен в прилож.3.

Таблица 1

Набор счетчиков программы Системный монитор для первого упражнения

Объект	Счетчик	Выбрать вхождение из списка
SQL Server:Access Methods (Методы доступа)	FullScans/sec(Полных сканирований в сек)	
SQL Server:Access Methods	IndexSearch/sec(Индексных поисков в сек)	
SQL Server:Buffer Manager (Диспетчер буферов)	Buffer Cashe Hit Ratio (коэффициент кэшированных операций)	
SQL Server:Databases	Active Transactions (Активные транзакции)	NorthwindCopy
SQL Server:Databases	PercentLogUsed(% использования журнала транзакций)	NorthwindCopy

SQL Server:Databases	Transactions/sec (Транзакций/сек)	NorthwindCopy
SQL Server:Memory Manager (Диспетчер памяти)	LockBlocks(Препятствующие блокировки)	
SQL Server:SQL Statistics (Статистика SQL Server)	BatchRequests/sec(Пакетных запросов в секунду)	

Проведите имитацию деятельности сервера

Деятельность сервера будет имитировать программа C:\MOC\2072a\Labfiles\L08\Monitor.bat, которую следует вызвать из командной строки (Пуск/Выполнить).

Наблюдайте окно «Просмотр диаграммы» во время выполнения командных файлов. Запишите значения счетчиков. Опишите в отчете, какие тенденции Вы отметили.

Отслеживание использования памяти и процессора.

В окне системного монитора щелкните по кнопке «Новый набор счетчиков» и добавьте набор счетчиков в соответствии с табл.2.

Таблица 2

Набор счетчиков для мониторинга использования памяти и процессора

Объект	Счетчик	Выбрать вхождение из списка
Память	Обмен страниц/сек	
Память	Ошибок страниц/сек	
Процесс	% загрузки процессора	Sqlserver
Процесс	Ошибок страниц/сек	Sqlserver
SQL Server:Cache Manager (Диспетчер КЕШ-памяти)	CacheHitRatio(Коэффициент успешного обращения к КЕШ памяти)	Adhoc SQL Plans
SQL Server:Memory Manager	Connection Memory(КВ)	
SQL Server:Memory Manager	Total Server Memory (КВ)	

Наблюдайте за окном «Просмотр диаграммы» во время выполнения программы Monitor.bat. Какие тенденции вы наблюдаете? Кнопкой на панели инструментов перейдите в режим отображения значений счетчиков. Скопируйте это окно в отчет, сравните значения счетчиков с допустимыми, сделайте выводы.

Закройте все окна командной строки на панели задач. Счетчики программы Системный монитор должны отразить снижение активности на сервере.

Создайте журнал для записи показаний системного монитора через каждые 20 сек в течение 5-10 мин. Остановите запись и просмотрите журнал.

2. Использование утилиты sql Profiler

Утилита SQLProfilerиспользуется для регистрации и проверки активности SQL Server . Ваша задача изучить способы фиксации событий SQL Server и запросов, а так же как организовывать и фильтровать полученную информацию. С помощью утилитыSQLProfilerможно обнаружить запросы, выполнение которых занимает много времени, отследить попытки незарегистрированных пользователей получить доступ к вашему серверу и многое другое.

Примечание. Перечень основных категорий событий и их основных свойств указаны в прилож.4.

Конфигурирование утилиты sqlProfilerдля обнаружения попыток несанкционированного подсоединения к серверу

Запустите программу SQLProfiler(Пуск/Программы/MicrosoftSQL Server /Profiler.

В окне утилиты щелкните по кнопке «NewTrace» (Новая трассировка), после чего соединитесь с вашим сервером, используя аутентификацию Windows.

Установите параметры трассировки, используя данные табл.3.

Таблица 3

Параметры трассировки для проверки несанкционированных подключений

Параметр	Значение
Tracename(Введите имя трассировки)	Аудит_безопасности
Template name (наименование шаблона)	SQLProfilerStandart
Установите флажок Save to file	Имя файла трассировки SecurityAudit, сохраните на дискете
На вкладке Events	Добавьте AuditLoginFailedк событиям по умолчанию в данной трассировке
На вкладке Datacolumns(столбцы данных для вывода в трассировке)	Переместите NTUserNameв секциюGroups(кнопкаUP)
На вкладке Filters(фильтры)	NTUserName not like SQLService (имя NTUserName не должно совпадать с SQLService)

- Щелкните RUN, чтобы запустить трассировку.
- Запустите SQL Server Query Analyzer .
- Попробуйте установить соединения с SQL Server с помощью SQLServerAuthentication, используя учетную записьGLORIAбез пароля.
- Сделайте еще одну попытку, задав другую учетную запись без пароля.
- Переключитесь в окно утилиты SQLProfiler. Опишите в отчете, какая информация появилась в окне трассировки?
- Переключитесь в SQL Server Query Analyzer и установите соединения, используя Windows Authentication.
- Переключитесь в окно утилиты SQLProfilerи исследуйте трассировочную информацию. Какая информация зарегистрирована?
- Остановите и закройте активную трассировку.

Конфигурирование трассировки для регистрации долго выполняющихся запросов

В этой процедуре вы настроите трассировку на отслеживание времени выполнения запросов и отфильтруете информацию таким образом, чтобы не отображать активность SQLServerAgentи некоторых системных процедур.

- В окне SQLProfilerщелкните по кнопкеNewTrace, после чего установите соединение с сервером, используя аутентификацию Windows.
- Настройте параметры трассировки в соответствии с табл.4.

Таблица 4

Параметры трассировки для регистрации времени выполнения запросов

Параметр	Значение
Trace name	Долго_выполняющийся_запрос
Template name	SQLProfilerTSQLDuration
Save to file	Долго_выполняющийся_запрос
Events Stored procedures TSQL	RPC:Completed SQL:BatchCompleted
Data columns	Groups:Duration, TextData Columns:NTUserName,CPU,EventClass, SPID
Filters	NTUserName not like SQLService

- Щелкните RUN, чтобы начать трассировку.
- Переключитесь в SQL Server Query Analyzer.
- Откройте файл Labfiles\L08\OrderWorkload.sql, просмотрите его содержимое, а затем выполните. Этот файл моделирует активность SQL Server с помощью ряда запросов к базе данныхNorthwindcopy.
- После завершения запросов переключитесь в SQLProfilerи остановите трассировку.

Обратите внимание, что самая долго выполняемая команда SELECT имеет вид:
 SELECT c.CompanyName, p.ProductName, SUM(d.Quantity * d.UnitPrice)
 FROM Customers c JOIN Orders o ON c.CustomerID = o.CustomerID
 JOIN [Order Details] d ON o.OrderID = d.OrderID
 JOIN Products p ON d.ProductID = p.ProductID
 WHERE p.ProductID in (5, 11, 12)
 GROUP BY c.CompanyName, p.ProductName

- Скопируйте команду запроса из нижней части окна трассировки (эта команда вам пригодится в следующем приложении).
- Закройте SQL Profiler.

3. Использование sql Server Query Analyzer для анализа долго выполняющихся запросов

Формирование статистических характеристик производительности и графического плана выполнения запроса

- Переключитесь в SQL Server Query Analyzer.
- Откройте новое окно запроса и вставьте в него текст сохраненной в буфере команды долго выполняющегося запроса.
- В списке DB откройте базу данных Northwindcopy/
- В меню Query выберите пункт Current Connection Properties (свойства текущего соединения). В открывшемся окне установите флажки для настройки статистики Set statistics Time (статистика по времени) Set Statistics IO (статистика по количеству операций ввода-вывода).
- Выполните запрос.
- На панели результатов щелкните по вкладке Messages. Скопируйте результаты с нее в ваш отчет. Просмотрите их. Обратите внимание на использование времени процессора и счетчики операций ввода-вывода.
- В меню Query выберите пункт Display Estimated Execution Plan (показать план оценки выполнения запроса).

Сколько сканирований таблиц и индексов использовалось для обработки этого запроса? Что можно сделать для повышения производительности?

Формирование статистики производительности для запроса при наличии индексов

В этой процедуре вы должны проанализировать, как влияют индексы на производительность запросов. В принципе, вы должны повторить все действия предыдущего раздела, но с базой данных Northwind, которая имеет подходящие индексы

- Откройте новое окно запроса в Query Analyzer.
- Вставьте в него текст команды запроса и в списке баз данных установите базу данных Northwind.
- Выполните запрос.
- Просмотрите вкладку Message, Скопируйте текст сообщения в ваш отчет. Сравните его с предыдущим сообщением. Сделайте выводы.
- Просмотрите план выполнения запроса (Query|Display Estimated Execution Plan). Сколько сканирований таблиц и индексов использовались для

Удержание блокировки на сервере

- Переключитесь в Query Analyzer, откройте новое соединение (File|Connect...), используя аутентификацию Windows. **Первое соединение не закрывайте!**
- Используя второе соединение, запишите в нем текст приведенной выше программы и выполните ее. Просмотрите результаты работы системной хранимой процедуры sp_lock в блокировках. Запишите идентификатор последнего процесса (SPID).

Какие типы блокировок связаны с этим процессом? Каков режим блокировки каждого ресурса?

Переключитесь в SQL Server Enterprise Manager, щелкните правой кнопкой мыши по рубрике CurrentActivity, выполните команду «Обновить». Раскройте рубрику Locks/ProcessID, чтобы просмотреть блокировки второго соединения.

Примечание. Описание основных типов блокировок и механизмов их действия находится в прилож.5.

1. *Обнаружение блокирование блокировок*

Переключитесь в Query Analyzer и откройте еще одно новое соединения, не закрывая два предыдущих.

В третьем соединении скопируйте в окно Query Analyzer программу из второго соединения и запустите программу.

Завершился ли запрос?

- Переключитесь в окно первого соединения и выполните в нем процедуру sp_lock. Просмотрите результаты. По колонки статуса просмотрите, ожидает ли последний процесс какие-либо ресурсы?
- Переключитесь в Enterprise Manager и обновите окно Current Activity.
- Раскройте папку Locks/Process ID. Просмотрите информацию о блокировках третьего соединения. Почему запрос в третьем соединении не может быть завершен?
- В дереве консоли щелкните по блокирующему соединению.
- В окне детализации выполните двойной щелчок по строке блокировки, и в открывшемся окне завершите этот процесс (KillProcess)|Yes|Close.
- Переключитесь в Query Analyzer, в окне первого соединения выполните процедуру для показа блокировок sp_lock. Вы увидите, что блокировки, которые были установлены от выполнения транзакции во втором соединении, теперь отсутствуют.
- Переключитесь в третье соединение. Обратите внимание, что запрос теперь выполнен. Наберите в окне команду Rollback tran, выделите ее и выполните, чтобы завершить транзакцию откатом.
- Переключитесь в Enterprise Manager, обновите окно Current Activity, просмотрите информацию о блокировках, и убедитесь, что теперь третье соединение удерживает только разделяемые (S - Shared) блокировки.
- Закройте все соединения Query Analyzer.

Контрольные вопросы

Лабораторная работа № 22

Тема: Выполнение резервного копирования

Цель работы: формирование умений создания резервной копии базы данных

Практическая часть

1. Резервное копирование с помощью Enterprise Manager

1. Разверните папку Databases, выберите базу данных Northwind с помощью команды Свойства контекстного меню, откройте вкладку Options и отмените все ранее установленные опции.
2. Затем перейдите к рубрике Backup из папки Management и выполните команду контекстного меню Backup a database.
3. В окне SQL Server Backup заполните опции в соответствии с табл. 1.

Таблица 1

Опции для резервного копирования БД

Опция	Значение
Database(База данных)	Northwind
Name(Имя)	NorthwindFull
Description(Описание)	Первая полная резервная копия
Backup Type(Тип резервного копирования)	Database complete(Полное резервное копирование БД)
Destination(Место назначения копии) ¹	Nw1 (C:\Backup\Nw1.bak)
Overwrite(Перезапись)	Overwrite existing media(Перезаписать существующие носители)

Щелчок по кнопке ОК запустит процесс резервного копирования.

2. Добавление резервных копий в один и тот же файл с помощью языка Transact-sql

1. Запустите Query Analyzer.
2. Наберите и выполните следующие команды для создания новой резервной копии базы данных Northwind и добавления в нее в логическое устройство Nw1.

/*

**Создание еще одной полной резервной копии

*/

```
BACKUP DATABASE Northwind to Nw1
```

```
WITH NOINIT,
```

```
DESCRIPTION= 'Вторая полная резервная копия'
```

```
Сохраните сценарий в папке C:\backup под именем back_nw.sql
```

Что задается с помощью опции WITH NOINIT?

В результате выполнения данного сценария Вы должны получить сообщения примерно такого вида:

Processed 400 pages for database 'Northwind', file 'Northwind' on file 1.

Processed 1 pages for database 'Northwind', file 'Northwind_log' on file 1.

BACKUP DATABASE successfully processed 401 pages in 0.359 seconds (9.130 MB/sec).

Что это означает? Какой тип резервной копии был создан?

Напишите самостоятельно:

- Команду для создания логического устройства для резервного копирования базы данных с именем `kadru1` в папке `C:\backup`;
- Команду для создания полной резервной копии базы данных `kadru1` созданное логическое устройство

3. Просмотр содержания устройства резервного копирования

1. Переключитесь в программу SQL Server Enterprise Manager .
2. раскройте содержимое рубрики Backupи посмотрите свойства логических устройств `Nw1`, `kadru1` (команда Свойства контекстного меню).
3. В появившемся меню щелкните по кнопке ViewContents. Отобразите информацию окна в своем отчете. Объясните ее.

4. Перезапись существующего файла резервных копий с помощью языка Transact-SQL

В этой процедуре вы должны создать полную резервную копию средствами языка Transact-SQL, полностью перезаписав файл резервной копии `Nw1`. Для этого следует задать опцию WITHINи дать описание (Description) «Третья резервная копия». Внесите соответствующие исправления в файл сценария `C:\backup\back_nw.sql` и выполните его в Query Analyzer.

Просмотрите заново свойства логического устройства `Nw1`. Убедитесь, что оно содержит только одну последнюю копию. Запишите команду и результат просмотра в отчет.

5. Резервное копирование одной базы данных в несколько файлов резервных копий

Порядок выполнения данного раздела работы такой же, как в п. 1, только в области Destination проставьте два файла `Nwstripe1` и `Nwstripe1` и дайте другое описание «Параллельное резервное копирование базы данных Northwind».

Просмотрите содержимое файлов `Nwstripe1` и `Nwstrip2`. Что в них содержится?

6. Резервное копирование базы данных с созданием временного устройства резервного копирования

Выполните следующую программу для создания резервной копии в файле с именем `C:\Backup\MyNewBackup.bak`:

```
/*
```

```
**Выполнение резервного копирования.
```

```
**Создание нового файла копии .
```

```
**
```

```
*/
```

```
BACKUP DATABASE Northwind
```

```
TO DISK = 'C:\Backup\MyNewBackup.bak'
```

WITHFORMAT,

DESCRIPTION= ' Создание полной резервной копии без лог.устройства '

Откройте папку и убедитесь, что в ней создан файл MyNewBackup.bak.

Сохраните сценарий в папке C:\Backup

7. Резервное копирование журнала транзакций

Журнал транзакций также должен иметь резервные копии, чтобы фиксировать изменения в базе данных. Копирование журнала транзакций производится так же, как и базы данных с помощью Enterprise Manager или языка Transact-SQL.

1. Резервное копирование журнала транзакций с помощью Enterprise Manager

1. Раскройте ваш сервер и вызовите окно свойств базы данных Northwind.
2. На вкладке Options измените модель восстановления (Recovery Model) на FULL.
3. Выполните команду Backup database из контекстного меню, вызванного на рубрике Backup Management, и в окне Backup установите параметры в соответствии с табл. 2.

Таблица 2

Опция	Значение
Database	Northwind
Name	Nwindlog
Description	Копия журнала транзакций
Backup	Transaction Log
Destination	C:\Backup\Nwlog.bak
Overwrite	Перезапись существующего носителя

2. Резервное копирование журнала транзакций в Query Analyzer

Сделайте еще одну дополнительную копию журнала транзакций на устройство Nwlog с помощью следующей программы:

```
/*
```

```
** Добавление новой резервной копии журнала транзакций
```

```
*/
```

```
BACKUP LOG Northwind TO Nwlog
```

```
WITH NOINIT,
```

```
DESCRIPTION='ЕЩЕ ОДНА КОПИЯ ЖУРНАЛА'
```

Просмотрите свойства файла Nwlog и убедитесь, что там находятся две копии журнала транзакций.

8. Создание дифференциальной резервной копии

Дифференциальное (разностное) резервное копирование фиксирует самые последние изменения во всех записях базы данных Northwind с момента создания полной резервной копии.

1. Выполните дифференциальное резервное копирование с помощью следующей команды:

```
BACKUP DATABASE Northwind TO DISK = 'C:\Backup\Nwdiff.bak'  
WITH NOINIT, DIFFERENTIAL
```

Запишите команду в отчет, проанализируйте ее параметры.

В Enterprise Manager вызовите команду Backup database и в окне SQL Server Backup заполните опции в соответствии с табл.3.

Таблица 3

Опция	Значение
Database	Northwind
Description(Описание)	Изменения после полной копии базы данных
Backup Type	Database– differential(дифференциальная копия базы данных)
Destination(место расположения копии)	C:\Backup\Nwdiff.bak
Overwrite (перезапись)	Append to media (Добавить)

После завершения операции создания дифференциальной копии базы данных, щелкните правой кнопкой мыши по Northwind, выберите пункт Все задачи, а затем Backup Database. В окне диалога в списке Destination щелкните по Northwind, после чего щелкните по кнопке Contents, чтобы просмотреть содержимое временного устройства. Затем нажмите на кнопку Cancel, чтобы не выполнять резервное копирование еще раз. Запишите в отчет.

копирования

Напишите команды для создания дифференциальной резервной копии базы данных kadru и резервной копии журнала транзакций этой базы на логическое устройство kadru1.

Каждой копии дайте понятное название и снабдите ее описанием. Затем просмотрите содержимое устройства kadru1 в Enterprise Manager.

Отобразите результат в отчете.

Лабораторная работа № 23

Тема: Восстановление базы данных из резервной копии

Цель работы: обучение административным действиям для выполнения следующих работ:

- восстановление базы данных из полной резервной копии;
- выбор стратегии восстановления базы данных;
- восстановление баз данных с использованием дифференциальной резервной копии и копии журнала транзакций.

Практическая часть

1. Восстановление базы данных из полной резервной копии

Создание базы данных NWCOPY

1. Скопируйте файл NWC1.bak из папки C:\Moc\2072\Labfiles\L07 в папку C:\Backup.
2. Откройте SQL Server Query Analyzer .
3. Выполните программу SetupNWC.sql

Модификация базы данных NWCOPY

Запустите процедуру, которая добавит новую строку в таблицу Products. Затем для проверки запустите запрос, который возвратит эту строку. Для этого:

1. Откройте SQL Server Query Analyzer , откройте файл сценария Labfiles\L07\Addprod.sql
2. Просмотрите сценарий, выделите сначала команды, не включив последнюю команду запроса, и выполните их.
3. Выделите последнюю команду, запустите ее, просмотрите результат и убедитесь, что строка добавлена.

Создание резервной копии базы данных NWCOPY

1. Откройте файл C:\MOC\2072\Labfiles\L07\MakeBack.sql. Просмотрите его содержание, после чего выполните его.

USEMASTER

GO

*создание устройства для резервной копии

```
sp_addumpdevice 'disk', 'NWC2', 'c:\backup\NWC2.bak'
```

*выполнение резервного копирования

```
BACKUP DATABASE NWCOPY TO NWC2
```

```
WITH FORMAT, NAME = 'NWCOPY_Full',
```

```
DESCRIPTION = ' Полная копия NWCOPY'
```

Моделирование случайного изменения данных

*повреждение базы данных в результате выполнения

* команды обновления без условия

USENWCOPY

GO

```
UPDATE products SET productname = 'Nut Crunch Cookies'
```

*проверка с помощью запроса

```
SELECT * FROM products WHERE productname = 'Maple Flavor Pancake Mix'
```

2. Закройте SQL Server Query Analyzer.

Восстановление базы данных из полной резервной копии

1. Откройте SQL Server Enterprise Manager .
2. Разверните свой сервер. Разверните рубрику Databases, откройте окно свойств базы данных NWCOPY.
3. На вкладке Options перечеркните флажок Restrict Access, выберите опцию Members of db_owner, dbcreator, or sysadmin, чтобы ограничить доступ к базе данных в процессе ее восстановления.
4. В контекстном меню, вызванном на имени БД NWCOPY вызвать команду Все задачи | Restore Database в окне Restoredatabases

вкладке **General** выберите имя базы данных, тип **Database**, в списке **First backup to restore** - имя **NWCOPY_FULL**.

5. На вкладке **Options** выберите опцию **Leave database operational. No additional transaction logs can be restored**

Проверка восстановления базы данных

1. Выполните запрос к таблице **Products**.

```
use NWCOPY
```

```
select * from products
```

2. Убедитесь, что в поле **ProductName** находятся различные наименования продуктов, а в последней записи имя продукта **Maple Flavor Pancake Mix**.

Моделирование изменений в базе данных и сбор сведений о них

Целью данной работы является внесение изменений в базу данных **NWCOPY**, и выполнение следующих видов резервного копирования: полного, дифференциального и журнала транзакций. Затем Вы смоделируете повреждение устройства, которое содержит базу данных **NWCOPY** и исследуете сообщение об ошибке в журнале **Application Log** операционной системы **Windows 2000**.

Выполнение полного резервного копирования базы данных **NWCOPY**

1. Откройте **Query Analyzer**.
2. Создайте устройство для копирования логическим именем **nwc3**, с именем файла **C:\backup\nwc3.bak**.

Создайте полную резервную копию базы данных **Nwcopy** именем **'NwcopyFull'**, описанием = **' Полная резервная копия Nwcopy'**

```
USE Master
```

```
GO
```

```
sp_addumpdevice 'disk','NWC3','C:\BACKUP\NWC3.bak'
```

```
GO
```

****выполняется полное резервное копирование**

```
BACKUP DATABASE to NWC3
```

```
WITH FORMAT, NAME = 'Nwcopy Full', Description = ' Полная резервная копия Nwcopy '
```

Изменение базы данных **NWCOPY** и создание резервной копии журнала транзакций

1. Выполните сценарий **C:\MOC\2072\Labfiles\L07\AddCust1.sql**, который добавит клиента в таблицу **Customers**. Убедитесь, что операция прошла успешно.
2. Выполните сценарий **C:\MOC\2072\Labfiles\L07\LogBack1.sql**, с помощью которого создается резервная копия журнала транзакций на дисковое устройство **NWCHANGE**.

Изменение базы данных и создание дифференциальной резервной копии

1. Выполните сценарий **C:\MOC\2072\Labfiles\L07\AddCust2.sql**, который добавит еще одного клиента в таблицу **Customers** и с помощью запроса подтвердит, что добавление прошло успешно. Запишите в отчет название компании в добавленной записи.
2. Создайте дифференциальную резервную копию с помощью процедуры **C:\MOC\2072\Labfiles\L07\DiffBack.sql**, которая должна зафиксировать все

изменения в базе данных с момента создания полной резервной копии. Дифференциальная копия добавляется на дисковое устройство NWCHANGE.

Изменение базы данных nwcorY и создание еще одной резервной копии журнала транзакций

1. Выполните сценарий C:\MOC\2072\Labfiles\L07\AddCust3.sql, с помощью которого добавится еще одна запись в таблицу Customers, а затем подтвердите результат выполнением команды запроса. Запишите название компании в добавленной записи.
2. Откройте сценарий C:\MOC\2072\Labfiles\L07\LogBack2.sql, просмотрите его команды, а затем выполните их. При этом создастся еще одна копия журнала транзакций на устройство NWCHANGE.

Моделирование повреждения базы данных

1. Откройте на панели задач окно SQL Server Service Manager и остановите службу SQL Server .
2. Закройте программу Enterprise Manager
3. Переименуйте файл, содержащий базу данных NWCOPY (C:\Program Files\Microsoft SQL Server\Mssql\Data\Nwcopy_data.mdf), на Nwcopy_data.bad.
4. Запустите службу SQL Server.
3. Откройте Enterprise Manager, разверните свой сервер, рубрику Databases и убедитесь, что база данных NWCOPY стала подозрительной (Suspect).
4. Откройте программу Просмотр событий (C:\WINNT\system32\eventvwr.exe) и изучите содержимое журнала приложений. В нем должно присутствовать сообщение об ошибке, утверждающее, что файл NWCOPY_Data.mdf не найден. (Двойной щелчок по красному значку Error открывает окно с сообщением об ошибке).

Вопрос: Что вы должны сделать, чтобы восстановить базу данных?

Восстановление полной резервной копии базы данных, дифференциальной копии и копии журнала транзакций
Проверка имеющихся резервных копий
Вы должны исследовать дату и содержимое имеющихся резервных копий базы данных NWCOPY. Для этого выполните следующие шаги:

1. Запустите или переключитесь в программу Enterprise Manager.
2. Разверните свой сервер, разверните ветвь Management, после чего щелкните по команде Backup.
3. В окне детализации вызовите контекстное меню на файле NWC3 и вызовите команду «Свойства».
4. Нажмите в открывшемся окне кнопку View Contents (Просмотреть содержимое). Запишите в свой отчет тип, описание, а также дату и время каждой резервной копии на устройстве.

Вопрос: что содержит устройство NWC3?

5. Исследуйте аналогичным образом устройство NWCHANGE.

Вопрос: что содержит устройство NWCHANGE?

Просмотр выбранной стратегии восстановления

Вы должны рассмотреть стратегию восстановления, которую предлагает Вам по умолчанию Enterprise Manager, и определить, подходит ли она в данном случае. Для этого следует выполнить следующие действия:

1. Раскройте Ваш сервер, затем рубрику Databases, выберите испорченную базу данных NWCOPY.
2. В контекстном меню выполните команду Все задачи | Restore Database. Откроется окно Restore database.

Enterprise Manager по умолчанию выберет все необходимые параметры восстановления для того, чтобы привести базу данных в непротиворечивое состояние. Проанализируйте этот выбор. Обратите внимание на то, что из четырех резервных копий Enterprise Manager выбирает три (полную резервную копию, дифференциальную и последнюю копию журнала транзакций).

Вопрос 1: Вы согласны с этим планом восстановления?

Вопрос 2: Почему не выбрана первая резервная копия журнала транзакций?

3. Отмените восстановление.

Восстановление полной и дифференциальной резервных копий

Вы выполните восстановление только полной и дифференциальной резервных копий и затем организуете доступ к базе данных только для чтения.

1. Раскройте Ваш сервер, затем рубрику Databases, выберите испорченную базу данных NWCOPY.
2. В контекстном меню выполните команду Все задачи | Restore Database. Откроется окно Restore database.
3. В области Restore as Database выберите NWCOPY.
4. В области Restore установите переключатель Database
5. В области Parameters в списке Show backup of database выберите NWCOPY, в списке First backup to restore (Первая резервная копия для восстановления) выберите опцию восстановления полной резервной копии базы данных.
6. В списке Restore выберите файлы полной и дифференциальной резервных копий. (нажатие на кнопку Properties поможет Вам сделать этот выбор).
7. На вкладке Options выберите режим Leave database read-only and able to restore additional transaction logs (Оставить базу данных в режиме только для чтения с возможностью восстановления дополнительных журналов транзакций).
8. Щелкните по ОК, чтобы начать процесс восстановления.
9. Щелкните по ОК, когда процесс восстановления завершится.

Проверка содержимого базы данных

1. Откройте Query Analyzer. В окне запроса откройте программу C:\MOC\2072\Labfiles\L07>ListCust.sql, просмотрите текст программы и выполните ее.
2. Проверьте, восстановились ли данные о всех трех записях, добавленных Вами ранее?
3. Закройте окно запроса.

Восстановление журнала транзакций из резервной копии

Восстановите полную резервную копию базы данных (см. п. 1-5 из раздела «Восстановление полной дифференциальной резервных копий»)

1. В списке Restore выберите самый последний журнал транзакций.
2. На вкладке Options выберите режим, позволяющий оставить базу данных в рабочем состоянии (Leavedatabaseoperational.No additional transaction logs can be restored).
3. Щелкните по ОК, чтобы начать процесс восстановления.
4. Щелкните по ОК, когда процесс восстановления завершится.
5. Отмените режим ограничения доступа к базе данных NWCOPY.

Просмотр содержимого базы данных

Выполните все действия упражнения «Проверка содержимого базы данных». Восстановились ли данные о всех трех новых клиентах?

Лабораторная работа № 24

Тема: Реализация доступа пользователей к базе данных

Цель работы: изучить основные принципы управления учетными записями и ролями в MS SQL Server

Теоретическая часть

Список системных процедур и команд, которые позволяют реализовать политику разделения прав между пользователями баз данных.

Название встроенной процедуры	Описание	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_addlogin	Создание новой учетной записи в SQL Server в разделе Security: sp_addlogin [@loginame =] 'login' [, [@passwd =] 'password'] [, [@defdb =] 'database'] [, [@deflanguage =] 'language'] [, [@sid =] sid] [, [@encryptopt =] 'encryption_option'] Например: sp_addlogin 'login1',sysname, 'DB_Books' Создает пользователя в SQL Server без PUBLIC в БД 'DB_Books'. Нужно еще использовать sp_adduser [@loginame =] 'login' [, [@name_in_db =] 'user'] [, [@grpname =] 'group'], Пример: Создана база данных DB_Books. В ней создан пользователь Admin_DB с серверной ролью sysadmin, с ролью в БД db_owner. Создать в QueryAnalyzer нового пользователя с именем Public
sp_defaultdb	Изменяет для пользователя БД по умолчанию Этот пример устанавливает БД по умолчанию pubs для пользователя Victoria. Например, EXEC sp_defaultdb 'Victoria', 'pubs'		и паролем Public_1 (пароль не должен совпадать с именем пользователя) с помощью следующих команд (не забудьте нажать F5 для запуска команд на выполнение): EXEC sp_addlogin 'Public_', 'Public_1', 'DB_Books' use DB_Books EXEC sp_adduser 'Public_', 'Public_' В БД DB_Books создан пользователь Public_ с ролью в БД DB_Books public.
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 [@loginame =] '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'	Deny (отрицание)	Этот пример запрещает несколько системных привилегий для нескольких пользователей. Пользователи не могут использовать системные привилегии CREATE DATABASE or CREATE TABLE , если они не наделены ими через команду GRANT . Пример: DENY CREATE DATABASE, CREATE TABLE TO Mary, John, [Corporate\BobJ] DENY SELECT, INSERT, UPDATE, DELETE ON authors TO Mary, John, Tom
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'	Grant (предоставлять)	This example grants multiple statement permissions to the users Mary and John, and the Corporate\BobJ Windows NT group. GRANT CREATE DATABASE, CREATE TABLE TO Mary, John, [Corporate\BobJ] Назначение разрешения на выборку (SELECT) для роли PUBLIC в таблице Authors: GRANT SELECT ON Authors TO public
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	В текущей БД назначает роль конкретному пользователю. Пример А. Этот пример добавляет учетную запись Corporate\JeffL из Windows NT в БД Sales как пользователя Jeff. Jeff затем получает роль Sales_Managers в БД Sales. USE Sales --сделать текущей БД Sales GO –выполнить команду, а потом запустить следующую EXEC sp_grantdbaccess 'Corporate\JeffL', 'Jeff' GO EXEC sp_addrolemember 'Sales_Managers', 'Jeff' Пример В. Этот пример добавляет пользователя SQL Server с именем Michael к роли Engineering в текущей БД. EXEC sp_addrolemember 'Engineering', 'Michael'		
sp_helpprotect	Показывает список привилегий, ассоциированных с ролью.		
sp_helprolemember	Показывает список пользователей БД, входящих в указанную роль		
sp_addsrvrolemember	Присвоение встроенной серверной роли для существующей учетной записи sp_addsrvrolemember [@loginame =] 'login' ., [@rolename =		

	'role' Например: <code>sp_addsrvrolemember 'Admin_DB', 'sysadmin'</code>		Revoke (отменять)	This example revokes multiple statement permissions from multiple users. REVOKE CREATE TABLE, CREATE DEFAULT FROM Mary, John
sp_dropsrvrolemember	Удаление встроенной серверной роли для учетной записи или группы <code>sp_dropsrvrolemember [@loginame =] 'login', [@rolename =] 'role'</code> Например: <code>sp_dropsrvrolemember 'Admin_DB', 'sysadmin'</code>			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
sp_helpsrvrole	Описание только встроенных ролей в SQL Server <code>sp_helpsrvrole [[@srvrolename =] 'role']</code> Например: <code>sp_helpsrvrole 'sysadmin'</code>			
sp_helpsrvrolemember	Возвращает список ролей и учетных записей, которым присвоены эти роли <code>sp_helpsrvrolemember [[@srvrolename =] 'role']</code> Например: <code>sp_helpsrvrolemember 'sysadmin'</code>			
sp_srvrolepermission	Возвращает список ролей и разрешений, которые присвоены этим ролям <code>sp_srvrolepermission [[@srvrolename =] 'role']</code> Например: <code>sp_srvrolepermission 'sysadmin'</code>			

Практическая часть:

Задание 1. Создание базы данных *УчетВыданныхПодарков*

- Запустите утилиту SQL Server Management Studio (Пуск/Все программы/SQL Server 2014 Management Studio). Настройте подключение к серверу:
 - компонент DatabaseEngine
 - локальный компьютер (localhost\SQLEXPRESS)
 - проверки подлинности Windows
- Создайте базу данных *УчетВыданныхПодарков* с помощью скрипта или графической утилиты SQL Server Management Studio.

Скрипт

```
SQLQuery1.sql - loc...(BIK\user2413 (51))* ×
USE [master]
CREATE DATABASE [UchetPodarkov]
ON PRIMARY
( NAME = 'UchetPodarkov', FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL12.SQLEXPRESS\MSSQL\DATA\UchetPodarkov.mdf',
SIZE = 5120KB, MAXSIZE = UNLIMITED, FILEGROWTH = 1024KB )
LOG ON
( NAME = 'UchetPodarkov_log', FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL12.SQLEXPRESS\MSSQL\DATA\UchetPodarkov_log.ldf',
SIZE = 1024KB, MAXSIZE = 2048GB, FILEGROWTH = 10%)
```

- Создайте и выполните запросы для создания таблиц *Сотрудники*, *Дети*, *Подарки* базы данных.

Код сотрудника	Код сотрудника	Код ребенка
Фамилия	Имя ребенка	Стоимость подарка
Имя	Дата рождения	Дата выдачи подарка
Отчество	Код ребенка	Код выдачи
Должность		
Подразделение		
Дата приема на работу		

Запрос 1 (таблица *Сотрудники*)

```
USE [УчетВыданныхПодарков]
CREATE TABLE [dbo].[Сотрудники](
    [КодСотрудника] [int] NOT NULL,
    [Фамилия] [nvarchar](10) NOT NULL,
    [Имя] [nvarchar](10) NOT NULL,
    [Отчество] [nvarchar](15) NOT NULL,
    [Должность] [nvarchar](10) NOT NULL,
    [Подразделение] [nvarchar](10) NOT NULL,
    [ДатаПриемаНаРаботу] [date] NOT NULL,
    CONSTRAINT [PK_Сотрудники] PRIMARY KEY CLUSTERED
(
    [КодСотрудника] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
) ON [PRIMARY]
```

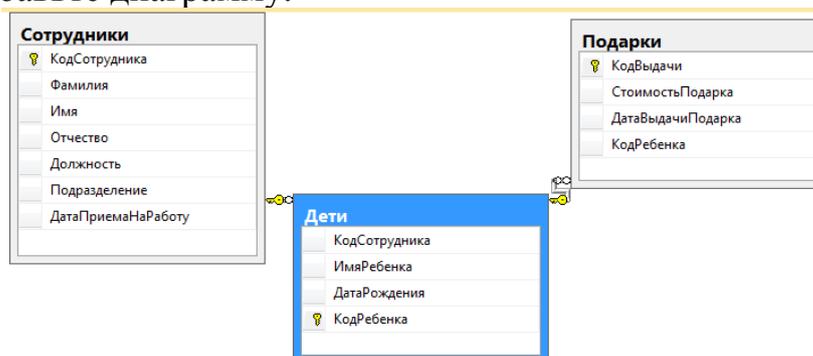
Запрос 2 (таблица Дети)

```
USE [УчетВыданныхПодарков]
CREATE TABLE [dbo].[Дети](
    [КодСотрудника] [int] NOT NULL,
    [ИмяРебенка] [nvarchar](10) NOT NULL,
    [ДатаРождения] [date] NOT NULL,
    [КодРебенка] [int] NOT NULL,
    CONSTRAINT [PK_Дети] PRIMARY KEY CLUSTERED
(
    [КодРебенка] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
) ON [PRIMARY]
```

Запрос 3 (таблица Подарки)

```
USE [УчетВыданныхПодарков]
CREATE TABLE [dbo].[Подарки](
    [КодВыдачи] [int] NOT NULL,
    [СтоимостьПодарка] [money] NOT NULL,
    [ДатаВыдачиПодарка] [date] NOT NULL,
    [КодРебенка] [int] NOT NULL,
    CONSTRAINT [PK_Подарки] PRIMARY KEY CLUSTERED
(
    [КодВыдачи] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
) ON [PRIMARY]
```

4. В Обозревателе объектов вызовите контекстное меню узла Диаграммы баз данных, добавьте диаграмму.



Либо создайте и выполните запросы:

```
ALTER TABLE [dbo].[Дети] WITH CHECK ADD CONSTRAINT [FK_Дети_Сотрудники] FOREIGN KEY([КодСотрудника])
REFERENCES [dbo].[Сотрудники] ([КодСотрудника])
GO

ALTER TABLE [dbo].[Дети] CHECK CONSTRAINT [FK_Дети_Сотрудники]
GO

ALTER TABLE [dbo].[Подарки] WITH CHECK ADD CONSTRAINT [FK_Подарки_Дети] FOREIGN KEY([КодРебенка])
REFERENCES [dbo].[Дети] ([КодРебенка])
GO

ALTER TABLE [dbo].[Подарки] CHECK CONSTRAINT [FK_Подарки_Дети]
GO
```

5. Заполните таблицы базы данных пятью записями

Задание 2. Создание пользователей для доступа к серверу через утилиту Microsoft SQL Server Management Studio

1. Создадим новую учетную запись для нашей базы данных *УчетВыданныхПодарков*. Для этого выберите в Обозревателе объектов раздел Безопасность/Имена входа. Добавьте новое имя входа – ProbaFirstName (созвучно **вашей** фамилии, например, ProbaIvanov), установите опцию Проверка

подлинности SQL Server, присвойте свой пароль, примените к **созданной** базе данных, установите язык по умолчанию – русский.

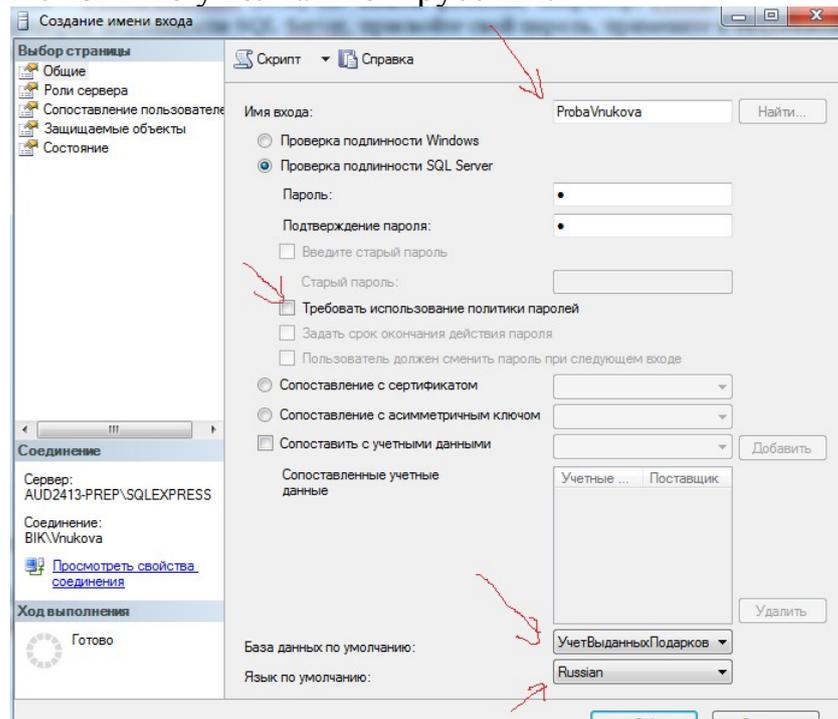


Рисунок 1 - Раздел Безопасность (Security) для работы с пользователями и создание нового пользователя (при SQL Server аутентификации нужно снять галочки с **Enforce password policy**)

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

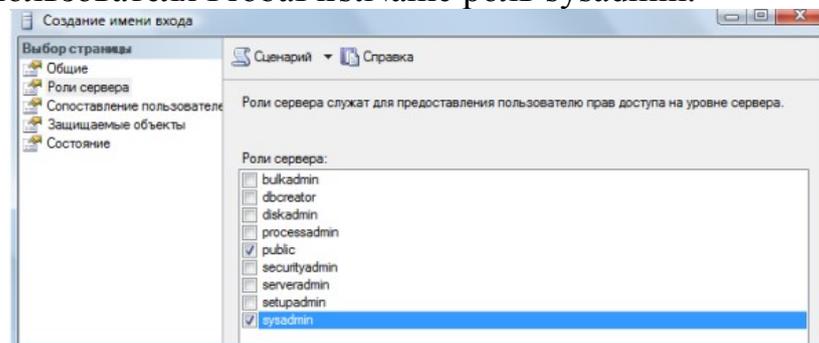


Рисунок 2. Настройка серверной роли для нового пользователя

Предопределенная роль уровня сервера	Описание
sysadmin	Члены предопределенной роли сервера sysadmin могут выполнять любые действия на сервере.
serveradmin	Элементы предопределенной роли сервера serveradmin могут изменять параметры конфигурации на уровне сервера, а также выключать сервер.
securityadmin	Элементы предопределенной роли сервера securityadmin управляют именами входа и их

Предопределенная роль уровня сервера	Описание
	<p>свойствами. Это могут быть разрешения на уровне сервера GRANT, DENY и REVOKE. Они также могут предоставлять, запрещать и отменять разрешения на уровне базы данных (инструкции GRANT, DENY и REVOKE) при наличии доступа к базе данных. Кроме того, они могут сбрасывать пароли для имен входа SQL Server.</p> <p>ВНИМАНИЕ! Возможность предоставления доступа к компоненту Компонент Database Engine и настройки разрешений пользователей позволяет администратору безопасности назначать большинство разрешений сервера. Роль securityadmin должна считаться эквивалентной роли sysadmin.</p>
processadmin	Члены предопределенной роли сервера processadmin могут завершать процессы, работающие на экземпляре SQL Server.
setupadmin	Члены предопределенной роли сервера setupadmin могут добавлять или удалять связанные серверы с помощью инструкций Transact-SQL.(Членство в роли sysadmin необходимо при использовании службы Среда Management Studio.)
bulkadmin	Элементы предопределенной роли сервера bulkadmin могут выполнять инструкцию BULK INSERT.
diskadmin	Предопределенная роль сервера diskadmin используется для управления файлами на диске.
dbcreator	Члены предопределенной роли сервера dbcreator могут создавать, изменять, удалять и восстанавливать любые базы данных.
public	<p>Каждое имя входа SQL Server принадлежит к роли сервера public. Если для участника на уровне сервера не были предоставлены или запрещены конкретные разрешения на защищаемый объект, он наследует разрешения роли public на этот объект. Разрешения роли public следует назначать только тому объекту, который будет доступен всем пользователям. Нельзя изменить членство в роли public.</p> <p>Примечание. Роль public реализуется не так, как другие роли; разрешения могут предоставляться, запрещаться или отменяться из открытых предопределенных ролей сервера.</p>

3. Далее просмотрите раздел Сопоставление пользователей. Установите для базы данных *УчетВыданныхПодарков* у пользователя `ProbaFirstName` права

доступа **Db_owner**, означающие, что пользователь может выполнять любые действия с базой данных.

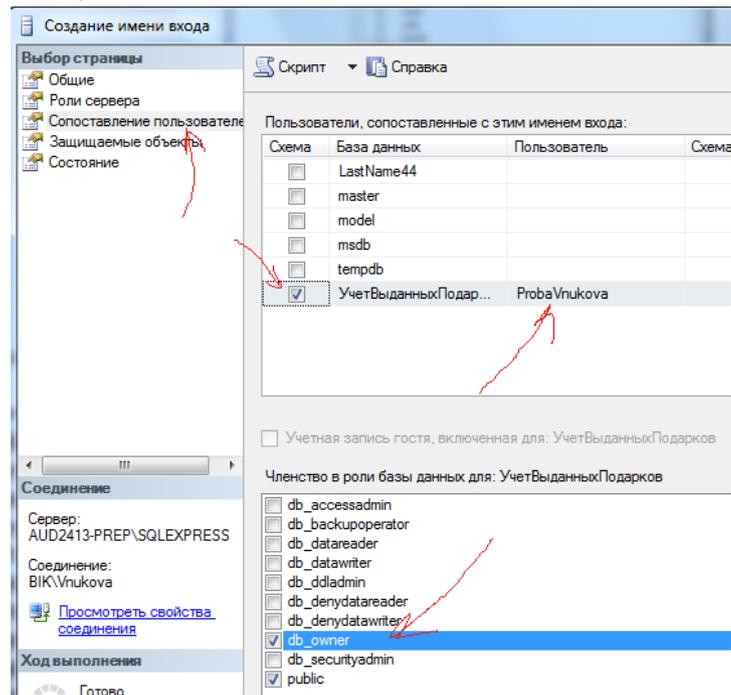


Рисунок 3. Настройка роли базы данных для нового пользователя (весь список ролей баз данных с их привилегиями ниже)

Перечень ролей БД:

Public – минимальные права доступа к базе данных (на просмотр)

Db_owner – может выполнять любые действия с базой данных

Db_accessadmin – добавляет и удаляет пользователей в базе данных

Db_securityadmin – управляет ролями в базе данных и разрешениями на запуск команд и работу с объектами базы данных

Db_ddladmin – добавляет, изменяет и удаляет объекты базы данных

Db_backupoperator – осуществляет резервное копирования базы данных

Db_dataSTUDENT – может просматривать все данные в каждой таблице в базе данных

Db_datawriter - может добавлять, удалять и изменять данные в каждой таблице в базе данных

Db_denydataSTUDENT – запрет на просмотр всех данных в каждой таблице в базе данных

Db_denydatawriter - запрет на добавление, удаление и изменение всех данных в каждой таблице в базе данных

4. Далее перейдите на раздел Состояние. Установите опции Разрешение на подключение к ядру СУБД – предоставить и Имя входа - включено.

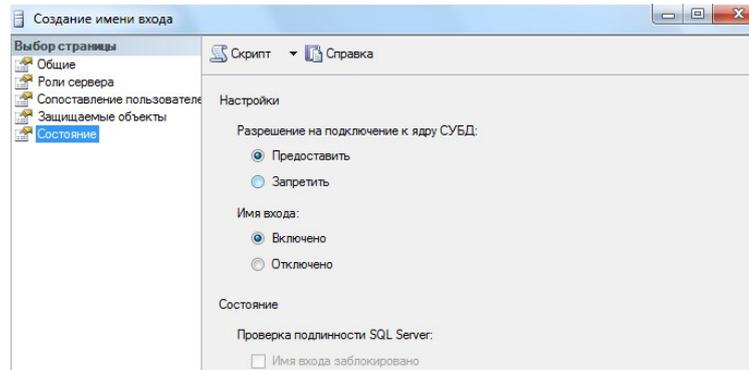
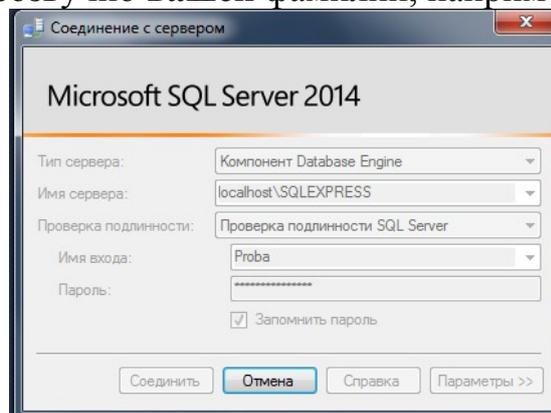


Рисунок 4. Разблокирование создаваемой учетной записи

После нажатия Ок в базе данных появится пользователь ProbaFirstName с правами собственника базы данных, который может выполнять все манипуляции с базой данных *УчетВыданныхПодарков*. Откройте в окне обозревателя объектов базу данных *УчетВыданныхПодарков* и перейдите на вкладку Безопасность, там вы найдете только что созданного пользователя.

Задание 3. Закройте Microsoft SQL Server Management Studio, зайдите снова под именем ProbaFirstName (созвучно **вашей** фамилии, например, ProbaIvanov)



Внесите изменения в таблицы базы данных *УчетВыданныхПодарков*

Контрольные вопросы

1. Какая системная хранимая процедура MS SQL Server предназначена для создания новой учетной записи?
2. Какая системная хранимая процедура MS SQL Server предназначена для запрета нескольких системных привилегий для нескольких пользователей?
3. Что дает роль Public в среде MS SQL Server?
4. Какая роль в среде MS SQL Server позволяет добавлять и удалять пользователей в базе данных?

Лабораторная работа № 25

Тема: Мониторинг безопасности работы с базами данных

Цель работы: научиться конфигурировать режим аутентификации для доступа к SQL Server; создавать пользователей и группы пользователей Windows; разрешать доступ группам и пользователям к SQL Server; аннулировать и запрещать доступ для пользователей и групп пользователей Windows

Практическая часть

1.1. Проверка способа аутентификации

1. Запустите программу SQLServerEnterpriseManager от имени администратора.
2. Разверните Microsoft SQL Server , а затем SQL Server Group. В контекстном меню на имени Вашего сервера вызовите команду Properties и откройте вкладку Security. Убедитесь, что возможна аутентификация Windows.

1.2. Создание пользователей и групп пользователей в Windows

1. Откройте с помощью «Мой компьютер» или через главное меню Windows (Пуск/Программы/Настройка) окно панели управления, раскройте в нем папку «Администрирование», а в ней выполните двойной щелчок по ярлыку «Управление компьютером».
2. Вызовите контекстное меню на папке «Пользователи», создайте новых пользователей, включив некоторых из них в новую группу с именем Нов_группа (см. табл.1).

Таблица 1

Пользователи и группы

Группа Windows 2000	Пользователи	Пароль пользователя
Нов_группа	Юрий Михаил Евгений Ирина	Юрий Михаил Евгений Ирина
	Дмитрий Павел Олег	Дмитрий Павел Олег

3. Создайте в Enterprise Manager Logins для доступа в SQL Server для новой группы (Нов_группа) и пользователей: Юрия, Михаила, Павла и Олега. Для этого разверните ветвь Security, выполните команду контекстного меню New Logins на рубрике Logins. Обеспечьте всем полный доступ к базе данных Northwind с помощью переключателя «Grant access» и выбора базы данных по умолчанию в области «Database». Отметьте в отчете сформированные компьютером имена пользователей и группы.

4. Перезагрузите компьютер, для входа в Windows используйте имя Юрий и пароль Юрий. Запустите программу Query Analyzer в режиме аутентификации Windows. Смог ли пользователь с именем Юрий подключиться к базе данных Northwind? Аналогичные действия выполните для Евгения, Олега и Дмитрия. По результатам заполните табл. 2.

Таблица 2

Результаты выполнения задания

Имя пользователя	Удалось подключение (Да/нет)?	Почему?
Юрий		

Евгений		
Олег		
Дмитрия		

5. Перезагрузите компьютер, войдя в Windows под именем системного администратора. Запретите Юрию доступ к SQL Server (переключателем DenyAccess). Сможет ли Юрий получить доступ к SQL Server как член группы, у которой есть доступ?

6. Удалите учетную запись группы Нов_группа. Проверьте, кто из членов удаленной группы сможет соединиться с SQL Server? Почему?

2. Управление разрешениями.

Цель данного раздела лабораторной работы состоит в получении знаний и навыков для выполнения следующих административных действий:

1. создания ролей базы данных;
2. включения учетных записей в роли;
3. назначения командных и объектных разрешений.

2.1. Создание ролей базы данных

1. Перезагрузите компьютер, войдите в Windows под именем системного администратора.
2. Запустите SQL Server Enterprise Manager. Разверните Ваш сервер и раскройте базу данных Northwind.
3. Добавьте пользователей базы данных с именами Михаил и Олег в рубрику Users.
4. Создайте роль с именем Нов_роль в базе данных Northwind, командой контекстного меню New Database Role, вызванного на рубрике Roles. Добавьте в данную роль пользователей, созданных в предыдущем пункте.

2.2. Назначение разрешений на операторы

1. Раскройте окно свойств базы данных Northwind и перейдите в нем на вкладку Permissions (разрешения). Пользователю с именем Олег дайте разрешение на создание представлений (Create View) и хранимых процедур (Create SP). Отразите в отчете: разрешение на создание каких объектов можно получить?
2. Перезагрузите компьютер, войдите в Windows под именем Олег с паролем Олег.
3. Запустите SQL Query Analyzer с аутентификацией Windows, создав представление следующими командами:

Use Northwind

Go

Create View v1 as

Select FirstName, LastName from Employees

Получилось?

4. Создайте новую таблицу следующими командами

Use Northwind

Go

Create table t1 (col1 int not null, col2 char(10))

Получилось? Почему да или почему нет?

2.3. Назначение разрешений объектам

1. Перезагрузите компьютер, войдя в Windows как системный администратор.
2. Запустите SQL Server Enterprise Manager, разверните свой сервер, раскройте базу данных Northwind и выполните двойной щелчок по рубрике Tables.
3. В окне со списком таблиц вызовите контекстное меню на таблице Categories, в нем команду «Все задания», а в подменю этой команды – команду Manage Permissions. В появившемся окне выберите объекты и назначьте им разрешения в соответствии с табл. 3. Для каждого объекта нажимайте на кнопку «Применить». Для назначения разрешения (Grant) в соответствующей клеточке должна быть проставлена «галочка» зеленого цвета, для отзыва разрешения (Revoke) клеточка должна оставаться пустой, для запрещения доступа (Deny) в клеточке должен появиться «крестик» красного цвета.

Таблица 3

Назначение разрешений объектам

Объект	Имя пользователя, группы или роли	Разрешение
Categories	Public	Grant all
Customers	Олег	Deny all
Customers	Нов_роль	Grant all
Customers	Public	Revoke all
Employees	Public	Revoke all
OrderDetails	Public	Revoke all
Orders	Public	Revoke all
Products	Public	Revoke all

4. Перезагрузите компьютер и войдите в Windows под именем Михаил с паролем Михаил. Запустите Query Analyzer, и соединитесь с SQL Server, используя Windows Authentication. (Напоминаем Вам, что Михаил является членом роли Нов_роль).

5. Для проверки разрешения для пользователя Михаила, попробуйте создать запрос для всех таблиц, результаты занесите в табл.4.

Результаты выполнения задания

Объект	Удалось создать запис (Да/нет)?	Почему?
Categories		
Customers		
Employees		

OrderDetails		
Orders		
Products		

6. Перегрузив компьютер, войдите в Windows под именем Олег с паролем Олег. Выполните аналогичные действия для пользователя Олег.

7. Какие разрешения могут быть заданы для таблицы?

3. Управление безопасностью на уровне приложений

Цель данного раздела лабораторной работы состоит в обучении следующим административным действиям:

1. созданию и активированию роли приложения;
2. использованию представлений и хранимых процедур для упрощения процесса управления разрешениями.

3.1. Создание роли приложения

1. Перегрузите компьютер, войдя в Windows как системный администратор.
2. Запустите SQL Server Enterprise Manager, разверните свой сервер, раскройте базу данных Northwind и выполните команду New Database Role контекстного меню на рубрике Roles. Создайте роль приложения с именем Прил_роль с паролем Прил_роль. В окне создания роли следует установить переключатель Application role.
3. Назначьте разрешения данной роли в соответствии с табл.5. (Properties/Permissions).

Таблица 5

Список разрешений для роли приложения Прил_роль

Таблица	Разрешения
Categories	SELECT
Customers	SELECT, INSERT, UPDATE
OrderDetails	SELECT, INSERT, UPDATE
Orders	SELECT, INSERT, UPDATE
Products	SELECT

4. Перегрузите компьютер, войдите в Windows под именем Олег с паролем Олег. Напоминаем Вам, что Олег является членом роли Нов_роль. Данной роли в свое время не был дан доступ к таблице Products. Также к данной таблице не был дан доступ и роли public, в которую по умолчанию попадают все пользователи базы данных.

5. Запустите Query Analyzer, используя аутентификацию Windows.

6. Выполните системную процедуру sp_setapprole, чтобы активизировать роль с помощью следующих команд:

Use Northwind

EXEC sp_setapprole 'Прил_роль', 'Прил_роль'

7. Выполните запрос из таблицы Products:

Выполнился ли данный запрос? Если да, то почему?

Попробуйте что-нибудь исправить в таблице Orders(например, значение 'Reims' поляShipcityзамените на 'reims').

Как долго будет активирована роль Прил_роль для пользователя Олег? Закройте QueryAnalyzer, а затем запустите его заново и попробуйте повторить п.7.

Получилось? Почему да или почему нет?

3.2. Назначение разрешений при помощи представлений и хранимых процедур

Перезагрузите компьютер, войдите в Windows под именем системного администратора.

1. Запустите SQLServerEnterpriseManager, разверните папкуDatabasesи с помощью мастера построений представлений (Tools/Wizards/CreateViewWizard) постройте представление к таблицеEmployees, выбирая последовательно в окнах мастера базу данныхNorthwind, таблицуEmployees, колонкиFirstName,LastNameиTitle, и задав имя представленияEmployees_View1.
2. Запустите программу Query Analyzer и создайте хранимую процедуру с именем employee_proc1 для выполнения запроса к таблицеEmployees с помощью следующих команд:

```
use northwind
```

```
go
```

```
create procedure employee_proc1 as
```

```
select firstname,lastname,title from employees
```

3. Выполните команды для назначения роли publicвыполнять представлениеEmployees_View1 и хранимую процедуруEmployee_proc1.

```
Grant select on Employees_view1 to public
```

```
Grant exec on Employee_proc1 to public
```

4. Выполните следующую команду для отзыва права выборки из таблицы Employeesу роли public:

```
revoke select on employees from public
```

3.3. Проверка разрешений на представление и хранимые процедуры

1. Перезагрузите компьютер, войдите в Windows под именем Олег с паролем Олег.
2. Запустите Query Analyzer с аутентификацией Windows.
3. Выполните запрос из представления Employees_view1

Удалось выполнить эту команду? Почему?

4. Выполните хранимую процедуру Employee_proc1 командой EXEC Employee_proc1

Получилось?

5. А теперь попробуйте выполнить запрос к таблице Employees
Получилось? Почему да или почему нет?

5. Лабораторная работа по теме «Резервное копирование баз данных»

Целью данной лабораторной работы является обучение административным действиям для выполнения следующих работ:

- создания постоянных файлов для хранения резервных копий;
- резервного копирования баз данных;
- резервного копирования журнала транзакций;
- выполнения дифференциального (разностного) резервного копирования.

1. Создание устройств резервного копирования с помощью SQL Server Enterprise Manager

1. Создайте на диске С новую папку с именем Backup.
2. Запустите программу SQL Server Enterprise Manager, разверните свой сервер.
3. Раскройте папку Management, вызовите контекстное меню на рубрике Backup и выполните команду NewBackupDevice. Создайте файл для резервной копии базы данных и назовите его Nw1.bak, разместите этот файл в папке C:\Backup.
4. Создайте еще одно устройство для хранения копии журнала транзакций с именем Nwlog.bak в папке C:\Backup.

2. Создание устройств резервного копирования с помощью команд языка Transact-SQL

1. Запустите Query Analyzer.
2. Выполните программу со сценарием для создания резервных устройств, содержащую следующие команды:

USE master

EXEC sp_addumpdevice 'disk', 'Nwstripe1', 'C:\Backup\Nwstripe1.bak'

EXEC sp_addumpdevice 'disk', 'Nwstripe2', 'C:\Backup\Nwstripe2.bak'

3. Откройте SQL Server Enterprise Manager, разверните свой сервер, в папке Management раскройте рубрику Backup и убедитесь, что все устройства Вами созданы.

Лабораторная работа № 26

Тема: Установка приоритетов

Цель работы: изучить основные функции и механизмы, заложенные в СУБД на примере MS SQL Server, позволяющие поддерживать БД в работоспособном состоянии.

Теоретическая часть:

Возможность создания *контрольных точек* была включена в SQL Server с целью фиксации изменений в базе данных или параметрах настройки в заранее заданный удобный момент времени. Если производилась настройка сервера или в его функционирование вносились изменения, потребовавшие перезагрузки управляющей программы, это значит, наступил момент для ручного запуска процедуры создания контрольной точки.

После запуска процедуры создания контрольной точки (независимо от способа ее вызова – вручную или посредством автоматического процесса) все модифицированные страницы памяти сбрасываются на диск. *Модифицированная страница* – это такая страница памяти, которая содержит изменения в данных, еще не зафиксированные в образе базы данных на жестком диске. По умолчанию

процедура создания контрольной точки автоматически запускается приблизительно каждые 60 секунд. Реальный промежуток времени между двумя контрольными точками определяется текущей загрузкой сервера, заданными параметрами защиты от сбоев и текущими установками общей настройки производительности SQL Server. Однако этот реальный промежуток времени будет достаточно близок к 60 секундам.

Вероятно, вы уже заметили, что если SQL Server неожиданно прекратил работу в результате случайного сбоя, то на его повторный запуск может потребоваться достаточно много времени. Это происходит по той причине, что SQL Server осуществляет откат и повторное выполнение всех транзакций, которые происходили после создания последней контрольной точки. При этом база данных предварительно приводится в последнее, известное системе, корректное состояние. Оно соответствует именно тому моменту, когда был запущен и успешно завершён процесс создания последней контрольной точки.

Работа SQL Server может быть в любой момент остановлена с помощью выдачи команды shutdown. Если перед выдачей этой команды ввести команду checkpoint, то все внесенные в результате последних транзакций изменения будут корректно зафиксированы в базе данных до остановки ее работы.

Если по какой-либо причине предстоит остановить работу SQL Server (особенно если требуется указать параметр WITH NOWAIT), то можно избежать продолжительной процедуры последующего запуска системы, предварительно вручную выдав команду CHECKPOINT. Выполнение этой команды ни чем не отличается от процедуры создания контрольной точки, вызванной автоматически. Вся информация будет сохранена на диске и запуск системы будет сведен к простой загрузке программ ядра базы данных и предоставления доступа к данным всем клиентским приложениям. Это замечание особенно полезно в тех случаях, когда необходима экстренная остановка сервера – например, при отказе системы электропитания и наличии источника бесперебойного питания, способного обеспечить функционирование сервера на время, достаточное для выполнения цикла создания контрольной точки.

Команда checkpoint выполняется на уровне базы данных и всегда применяется по отношению к текущей базе данных. Если в системе имеется несколько баз данных, эту команду следует выдать для каждой из баз данных в отдельности. Чтобы иметь право выдать команду checkpoint для некоторой базы данных, необходимо иметь статус ее владельца.

Кроме этого, следует упомянуть параметр Truncate log on checkpoint, который также может оказаться довольно полезным. Установка этого параметра вызывает автоматическую очистку журнала транзакций после завершения создания очередной контрольной точки. Таким образом, журнал транзакций будет вестись только в промежутках между созданием контрольных точек. Для установки этого параметра в окне SQL Server Enterprise Manager щелкните правой кнопкой мыши на пиктограмме базы данных и выберите в контекстном меню команду Properties. В раскрывшемся диалоговом окне свойств базы данных перейдите во вкладку Options, показанную на рис. 19.1.

При установленном параметре Truncate log on checkpoint невозможно сделать резервную копию журнала транзакций в процессе выполнения стандартной процедуры резервного копирования базы данных. Обычно это не является проблемой, поскольку стандартное копирование базы данных по-прежнему будет выполняться. Однако данное замечание следует учитывать при планировании процедур копирования. Процесс резервного копирования и восстановления системы подробно обсуждается в последующих разделах этой главы.

Если для некоторой базы данных установлен параметр Truncate log on checkpoint, то ее журнал транзакций будет каждый раз очищаться в момент успешной фиксации последней транзакции (при отсутствии в системе репликации данных). Если в системе используется репликация данных, информация в журнале очищается в момент успешного завершения репликации и фиксации в базе данных последней транзакции. Поскольку процедуры репликации данных используют данные журнала транзакций, информация о проведенной транзакции не удаляется из журнала до тех пор, пока сведения о ней не будут доставлены всем подписчикам данного источника.

Проверка целостности базы данных

```
USE master;
```

```
GO
```

```
EXEC sp_dboption 'Demo_DataBase', 'read only', 'TRUE';
```

Режимы выполнения команды dbcc

Команда DBCC поддерживает несколько режимов выполнения. В последующих разделах будут рассмотрены те из них, которые используются чаще всего. Кроме того, мы проанализируем, чем они могут помочь в управлении системой SQL Server.

Dbcc checkalloc

```
DBCC CHECKALLOC [(имя_базы_данных'[, NOINDEX])] [WITH  
NO_INFOMSGS]
```

Если при вызове команды опустить имя базы данных, SQL Server осуществит проверку текущей базы данных. При выполнении команды с указанием режима CHECKALLOC в отчет помещается детальная информация о системе и существующих в ней объектах базы данных. Эту информацию можно использовать для поиска возможных проблем в системе. Обнаруженные проблемы следует изучать и устранять по отдельности. Помещаемая в отчет информация – это сведения о существующей структуре таблиц, распределении страниц памяти и т.п. Каждое сообщение об обнаруженной ошибке обязательно сопровождается конкретными инструкциями по ее устранению.

Практическая часть:

Пример:

```
USE Demo_DataBase
```

```
GO
```

```
DBCC CHECKALLOC
```

Результат (фрагмент):

DBCC results for 'Demo_DataBase'.

Table sys.sysrowsetcolumns Object ID 4.

Index ID 1, partition ID 262144, alloc unit ID 262144 (type In-row data). FirstIAM (1:139). Root (1:66). Dpages 5.

Index ID 1, partition ID 262144, alloc unit ID 262144 (type In-row data). 7 pages used in 0 dedicated extents.

Total number of extents is 0.

...

CHECKALLOC found 0 allocation errors and 0 consistency errors in database 'Demo_DataBase'.

DBCC execution completed. If DBCC printed error messages, contact your system administrator.

Dbcc checkdb

При вызове команды DBCC в режиме CHECKDB, на наличие ошибок проверяется каждая таблица и связанные с ней страницы данных, индексы и указатели. Индексы и страницы данных контролируются на предмет корректности установленных связей. Кроме того, индексы дополнительно анализируются на соответствие заданному порядку сортировки. Для каждой страницы проверяется обоснованность данных, целостность указателей и корректность ее формата. Дополнительно в режиме CHECKDB контролируется правильность связей между текстом, графикой и страницами типа NTEXT, а также корректность их размера. Если выполняется вызов команды DBCC в режиме CHECKDB, то повторно запускать ее в режимах CHECKALLOC или CHECKTABLE не потребуется. Команда DBCC в режиме CHECKDB имеет следующий синтаксис:

```
DBCC CHECKDB [(имя_базы_данных['], NOINDEX)] [WITH NO_INFOMSGS]
```

Если опустить параметр имени базы данных, будет выполнена проверка текущей базы данных.

Результат (фрагмент) выполнения команды DBCC CHECKDB без параметров для текущей БД Demo_DataBase:

DBCC results for 'Demo_DataBase'.

Service Broker Msg 9675, State 1: Message Types analyzed: 14.

Service Broker Msg 9676, State 1: Service Contracts analyzed: 6.

Service Broker Msg 9667, State 1: Services analyzed: 3.

Service Broker Msg 9668, State 1: Service Queues analyzed: 3.

Service Broker Msg 9669, State 1: Conversation Endpoints analyzed: 0.

Service Broker Msg 9674, State 1: Conversation Groups analyzed: 0.

Service Broker Msg 9670, State 1: Remote Service Bindings analyzed: 0.

DBCC results for 'sys.sysrowsetcolumns'.

There are 557 rows in 5 pages for object "sys.sysrowsetcolumns".

DBCC results for 'sys.sysrowsets'.

There are 83 rows in 1 pages for object "sys.sysrowsets".

DBCC results for 'sysallocunits'.

There are 96 rows in 1 pages for object "sysallocunits".

DBCC results for 'sys.sysfiles1'.

There are 2 rows in 1 pages for object "sys.sysfiles1".

DBCC results for 'sys.syshobtcolumns'.

There are 557 rows in 5 pages for object "sys.syshobtcolumns".

DBCC results for 'sys.syshobts'.

There are 83 rows in 1 pages for object "sys.syshobts".

...

CHECKDB found 0 allocation errors and 0 consistency errors in database 'Demo_DataBase'.

DBCC execution completed. If DBCC printed error messages, contact your system administrator.

Dbcc checktable

Если проверка всей базы данных в целом занимает слишком много времени, можно воспользоваться режимом checktable. В этом случае в команде dbcc задается режим checktable и указывается перечень имен таблиц базы данных, которые следует проверить. Будет выполнен анализ состояния только указанных таблиц, что позволит сократить общее время проверки.

Dbcc checkfilegroup

При выполнении команды DBCC в режиме CHECKFILEGROUP, проверяется каждая таблица и связанные с ней страницы данных, индексы и указатели. Индексы и страницы данных контролируются на корректность установленных связей. Кроме того, индексы дополнительно анализируются на соответствие заданному порядку сортировки. Для каждой страницы проверяется обоснованность данных, целостность указателей и корректность ее формата. Дополнительно в режиме CHECKFILEGROUP контролируется правильность связей между текстом, графикой и страницами типа NTEXT, а также корректность их размера.

Команда DBCC в режиме CHECKFILEGROUP имеет следующий синтаксис:

```
DBCC CHECKFILEGROUP [({'filegroup_name' | filegroup_id | 0 } ]  
[ , NOINDEX ]]  
[ WITH { [ ALL_ERRORMSG ] [ NO_INFOMSGS ] } [ , [ TABLOCK ] ]  
[ , [ ESTIMATEONLY ] ] }
```

Операторы, выводящие информацию о БД

DBCC CONCURRENCYVIOLATION	DBCC SHOW_STATISTICS
DBCC INPUTBUFFER	DBCC SHOWCONTIG
DBCC OPENTDBCC	DBCC SQLPERF
INPUTBUFFERRAN	
DBCC OUTPUTBUFFER	DBCC TRACESTATUS
DBCC PROCCACHE	DBCC USEROPTIONS

Операторы проверки целостности и корректности данных объектов БД

DBCC CHECKALLOC	DBCC CHECKFILEGROUP
DBCC CHECKCATALOG	DBCC CHECKIDENT
DBCC CHECKCONSTRAINTS	DBCC CHECKTABLE

DBCC CHECKDB	
Операторы очистки и восстановления объектов БД	
DBCC CLEANTABLE	DBCC INDEXDEFRAG
DBCC DBREINDEX	DBCC SHRINKDATABASE
DBCC DROPCLEANBUFFERS	DBCC SHRINKFILE
DBCC FREEPROCCACHE	DBCC UPDATEUSAGE
Miscellaneous Statements	
DBCC dllname (FREE)	DBCC TRACEOFF
DBCC HELP	DBCC TRACEON

Команды update statistics и recompile

Высокая производительность системы SQL Server в значительной степени обеспечивается за счет интеллектуальных процессов обработки сохраняемых в таблицах данных. Необходимый анализ информации осуществляется различными методами, однако самым рациональным из них является изучение сохраняемых в системе реальных данных с целью определения оптимальных путей доступа к ним.

Чтобы лучше уяснить, о чем идет речь, давайте в качестве примера рассмотрим маршрут следования к вашему собственному дому. Весьма вероятно, что вы уже нашли самый оптимальный и быстрый путь между местом работы и местом жительства. И раз этот маршрут уже найден, то чаще всего вы будете пользоваться именно им, даже если кто-то предложит вам воспользоваться иным путем. Ведь вы уже потратили достаточно времени на анализ различных возможных маршрутов и выбрали из них самый оптимальный.

Для того чтобы в сложившийся маршрут были внесены изменения, необходимы достаточно серьезные причины. Это может быть новая дорога, реконструкция уже существующих путей или что-либо в этом роде, способное оказать влияние на время, за которое вы добираетесь домой.

В системе SQL Server подобная аналогия вполне справедлива. При создании хранимой процедуры SQL Server анализирует логику ее оператора SELECT совместно с любыми прочими условиями, зависящими от обрабатываемых данных. На основе этого анализа выбирается оптимальный маршрут обращения к данным при выполнении хранимой процедуры. После того как маршрут определен, информация о нем запоминается, поэтому при каждом следующем выполнении хранимой процедуры используемый в ней алгоритм доступа к данным будет оптимальным. "Маршруты" к данным в SQL Server прокладываются на основе существующих в системе индексов таблиц. Именно эти элементы анализирует система, определяя наиболее эффективный способ доступа к требуемой информации.

Если в процессе работы в таблицу будет добавлено достаточно большое количество строк данных (превышающее 20% ее исходного размера), следует выполнить обновление статистических данных об этой таблице. Используемая для этой цели команда имеет следующий синтаксис:

```
UPDATE STATISTICS table | view
[
{
```

```

{ index | statistics_name }
| ( { index | statistics_name } [ ,...n ] )
}
]
[ WITH
[
[ FULLSCAN ]
| SAMPLE number { PERCENT | ROWS } ]
| RESAMPLE
| <update_stats_stream_option> [ ,...n ]
]
[ [ , ] [ ALL | COLUMNS | INDEX ]
[ [ , ] NORECOMPUTE ]
];
<update_stats_stream_option> ::=
[ STATS_STREAM = stats_stream ]
[ ROWCOUNT = numeric_constant ]
[ PAGECOUNT = numeric_constant ]

```

Ниже приведен пример команды обновления статистики для таблицы Authors:
update statistics Authors

В команде можно потребовать выполнить пересчет статистики как для одного определенного индекса или столбца данных, так и для всей таблицы в целом. Если указывается отдельный индекс или столбец данных, то это делается так, как показано выше, с одновременным указанием имени таблицы, в которой этот индекс или столбец располагается.

Периодически SQL Server по собственной инициативе использует промежутки времени, когда процессор простаивает, для обработки таблиц с целью поддержания актуальности статистических данных, что способствует повышению эффективности обработки информации в системе.

Режим ручного обновления статистики также сохранен, причем соответствующая команда дополнена новыми параметрами. Так, при указании в команде обновления статистики параметра FULLSCAN SQL Server выполнит полное сканирование указанного индекса или таблицы. Наличие в команде параметра SAMPLE потребует от SQL Server использовать технологию выборки данных на основе указанного количества или процента строк. В этом случае SQL Server должен будет убедиться, что в качестве образца выбрано статистически значимое число значений. Если указанное число или процент строк недостаточно велики для выполнения этого требования, система автоматически увеличит заданное значение. Последний из параметров INDEX, COLUMN или ALL указывает, для каких объектов должна собираться статистика – индексов, столбцов или и тех, и других. При опускании этого параметра статистика собирается только для индексов.

Для прекращения автоматического сбора статистики (хотя я и не знаю, зачем это может кому-либо потребоваться) следует ввести команду UPDATE STATISTICS с параметром NORECOMPUTE. Указание этого параметра

блокирует выполнение процедур автоматического сбора статистики в системе SQL Server.

Статистика по определенной таблице, индексу или столбцу может быть удалена, для чего предназначена команда `DROP STATISTICS`, имеющая приведенный ниже синтаксис:

DROP STATISTICS *таблица.столбец* [...*таблица.столбец*]

Последним шагом на пути обновления индексов с целью использования новых статистических данных в хранимых процедурах является уведомление SQL Server о необходимости пересчитать маршруты, используемые для извлечения информации из базы данных. Это достигается посредством перекомпиляции хранимых процедур, работающих с данными, статистические сведения о которых были обновлены только что выполненными командами `UPDATE STATISTICS`.

Обычно хранимые процедуры компилируются в момент первого их вызова после перезапуска системы SQL Server. Существуют и другие ситуации, в которых выполняется автоматическая перекомпиляция хранимых процедур, однако простая выдача команды `UPDATE STATISTICS` к ним не относится. В качестве примера можно указать ситуацию, когда ликвидируется индекс, ранее использовавшийся в некоторой хранимой процедуре. Иногда можно заметить, что при первом вызове некоторой хранимой процедуры время реакции системы оказывается существенно большим, чем обычно. Причем все последующие вызовы этой процедуры выполняются уже с нормальной скоростью. Это происходит по той причине, что оптимизатор выполнил перекомпиляцию хранимой процедуры с целью учета новых статистических данных, полученных для используемых в запросе таблиц или индексов.

Чтобы перекомпилировать хранимые процедуры, необходимо установить для таблицы соответствующий флажок, предназначенный для извещения системы о том, что любые работающие с этой таблицей и находящиеся в процедурном кэше копии хранимых процедур следует считать недействительными. В результате SQL Server перезагрузит и перекомпилирует все соответствующие процедуры при первом же их вызове. Для установки данного флажка используется следующая команда:

`sp_recompile <имя_таблицы>`

Здесь параметр *имя_таблицы* определяет имя той таблицы, для которой следует перекомпилировать все обращающиеся к ней хранимые процедуры. В случае успешного выполнения команда выводит простое сообщение, уведомляющее, что все связанные с указанной таблицей хранимые процедуры будут перезагружены и перекомпилированы.

Пример:

```
EXEC sp_recompile 'authors'  
GO
```

Результат:

Object 'authors' was successfully marked for recompilation.

Если не перекомпилировать те хранимые процедуры, которые связаны с результатами выполнения команды `update statistics`, то ожидаемое улучшение производительности достигнуто не будет вплоть до очередной остановки и

перезагрузки сервера. Только в этом случае все хранимые процедуры будут автоматически заново перекомпилированы.

Выполнение команд UPDATE STATISTICS и RECOMPILE не может нанести системе никакого вреда, однако делать это рекомендуется в то время, когда активность пользователей в системе относительно невысока. Особенно важно придерживаться этого правила при работе с крупными базами данных. Время, которое будет затрачено на получение новых статистических данных, может существенно отразиться на производительности системы по обработке запросов пользователей. Лучше всего выполнять процедуры обновления статистики как часть регулярных манипуляций, связанных с плановым обслуживанием системы. Например, имеет смысл выполнять эту процедуру не реже, чем раз в месяц для каждой из интенсивно используемых в системе таблиц с невысоким уровнем изменения данных. В случае, когда система только запускается в работу и производится интенсивное заполнение данными ее таблиц, может потребоваться выполнять процедуры обновления статистики гораздо чаще – возможно, даже ежедневно, – если поток новых данных достаточно велик.

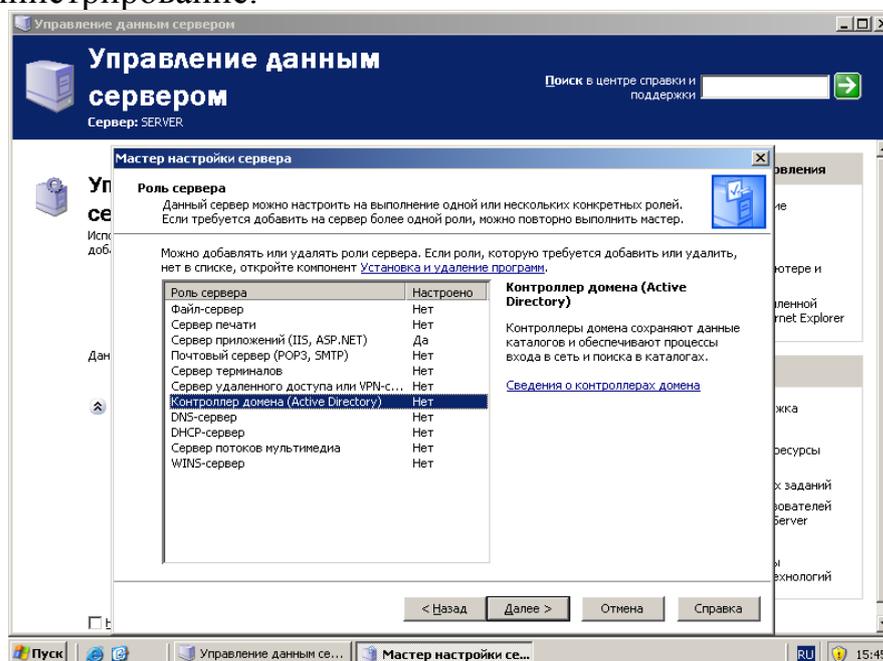
Лабораторная работа № 27-28

Тема: Развертывание контроллеров домена

Цель работы: научить студентов устанавливать и настраивать домен на основе операционной системы WindowsServer.

Теоретическая часть

Установка контроллера домена выполняется на компьютере с ОС WindowsServer, для чего можно запустить консоль Управление данным сервером в разделе Администрирование.



Выберем пункт Добавление/Удаление роли сервера для того, чтобы увидеть перечень возможных ролей сервера с указанием их наличия (Да) или отсутствия (Нет). Как видно из представленной мастером настройки информации контроллер домена (ActiveDirectory) не установлен, не настроены также необходимые в

домене службы имен (DNS-сервер) и служба присвоения IP-адресов (DHCP-сервер). Для установки контроллера домена выберем эту роль и нажмем кнопку Далее. Другой путь установки домена - запуск программы установки домена dsrptom.exe командой «Выполнить», доступной после активации кнопки Пуск. Мастер настройки сервера последовательно открывает окна для выполнения требуемых настроек. Необходимо выполнить следующие операции.

В окне Выбор типа контроллера домена – Контроллер домена в новом домене. В окне Создать новый домен – Новый домен в новом лесу. Новое имя домена – введите имя домена, например class.pi. Далее следовать указаниям мастера установки, оставляя предлагаемые настройки до пункта настройки службы имен на основе DNS– сервера. В окне диагностика регистрации DNSсервера выбрать Установить и настроить DNSсервер на этом компьютере.

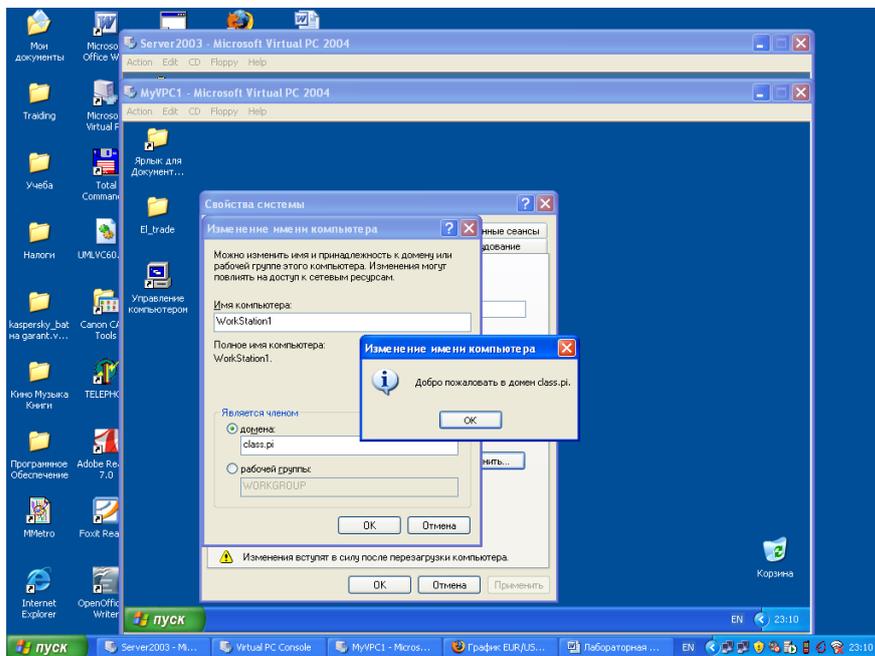
На этом этапе установки потребуется диск с ServicePack1. Для дальнейшей установки - с помощью кнопки обзор укажите на диск с дистрибутивом или подключите виртуальную машину через ее менюCDк образу ISO. После завершения установки потребуется перезагрузка, после чего компьютер готов к выполнению роли контроллера домена. На рисунке 2.2 видно, что установлены роли Контроллера домена (ActiveDirectory), DNS– сервера и DHCP– сервера..

Основные средства администрирования домена доступны через главное меню сервера в разделе Администрирование. В повседневной работе администратора наиболее часто используется пункт ActiveDirectory – пользователи и компьютер.

Следующей задачей создания сети предприятия, позволяющей изучать операционные системы в среде, моделирующей реальную сеть предприятия с правами администратора масштаба предприятия, является подключение компьютера клиента к домену.

Для этого на машине клиента необходимо раскрыть Мой компьютер/Свойства и выбрать раздел «Имя компьютера/Изменение имени компьютера». В этом пункте необходимо выбрать подключение к домену и набрать имя домена и ввести учетную запись администратора домена.

Если все настройки были верными, через некоторое время последует сообщение «Добро пожаловать в домен class.pi».



После перезагрузки компьютера становится возможным вход в домен с учетной записью администратора домена, после чего необходимо средствами администрирования сервера приступить к созданию рабочей среды домена

Для достижения поставленной цели рассматриваются следующие объекты домена и операции управления ими:

- Учетные записи пользователей и компьютеров
- Управление пользователями
- Управление группами
- Управление компьютерами

Учетные записи пользователя и компьютера

Учетные записи пользователей и учетные записи компьютеров в Active Directory представляют собой физические объекты, как компьютер или пользователь.

Группы, а также учетные записи пользователей и компьютеров называются участниками безопасности. Участники безопасности являются объектами каталогов, которым автоматически назначают коды безопасности (SID) для доступа к ресурсам домена.

Учетная запись пользователя или компьютера используется для следующих целей:

- Проверка подлинности пользователя или компьютера.
- Разрешение или запрещение доступа к ресурсам домена.
- Аудит действий, выполняемых с использованием учетной записи пользователя или компьютера. Аудит помогает при наблюдении за безопасностью учетных записей.

Учетные записи пользователей

Учетные записи пользователей, созданные автоматически при установке домена, называются встроенными учетными записями пользователей. После установки Active Directory контейнер Users, расположенный в оснастке Active Directory — пользователи и компьютеры, отображает встроенные учетные записи пользователей: «Администратор», «Гость» .

Учетная запись администратора имеет самые большие права и разрешения в домене, а учетная запись гостя — ограниченные права и разрешения.

- **Учетная запись «Администратор»**

При помощи учетной записи «Администратор» можно полностью контролировать домен, назначать права пользователей и разрешения управления доступом для пользователей домена. Эта запись должна использоваться только для задач, выполнение которых требует учетных данных администратора. По умолчанию учетная запись «Администратор» назначается членом групп Администраторы, Администраторы домена, Администраторы предприятия, Владельцы-создатели групповой политики и Администраторы схемы в Active Directory. Она не может быть удалена или перемещена из группы «Администраторы», но ее настоятельно рекомендуется переименовать. Поскольку коды безопасности (SID) учетных записей сохраняются, переименованная учетная запись сохраняет все остальные свойства, в том числе описание, пароль, принадлежность к группам, профиль пользователя, учетную информацию, а также любые разрешения и права пользователя.

- **Учетная запись «Гость»**

Учетная запись Гость используется теми, кто не имеет действительной учетной записи в домене. Если учетная запись пользователя отключена (но не удалена), он также может воспользоваться учетной записью «Гость». Учетная запись «Гость» не требует пароля.

Учетной записи «Гость», как и любой другой учетной записи, можно предоставлять права и разрешения на доступ к объектам. По умолчанию учетная запись «Гость» является членом встроенной группы Гости и глобальной группы Гости домена, позволяющих пользователям входить в домен. По умолчанию она отключена, и рекомендуется оставить ее в этом положении.

Защита учетных записей пользователей

Для обеспечения безопасности проверки подлинности пользователя следует создавать отдельные учетные записи для каждого пользователя сети, применяя для этого оснастку «Active Directory — пользователи и компьютеры». Каждая учетная запись пользователя (включая учетные записи администратора и гостя) может быть добавлена в группу для управления правами и разрешениями, назначенными этой учетной записи. Использование соответствующих этой сети учетных записей и групп позволяет проверить подлинность входящего в сеть пользователя и возможность предоставления ему разрешенных ресурсов.

Повысить защиту домена от атак можно с помощью надежных паролей и политики блокировки учетных записей. Применение политики блокировки учетных записей снижает вероятность проникновения злоумышленника в домен путем повторных попыток. Политика блокировки учетных записей позволяет установить число неудачных попыток входа в систему, после которых учетная запись отключается.

Параметры учетных записей

Каждая учетная запись пользователя Active Directory имеет ряд параметров, относящихся к безопасности и определяющих, как производится проверка подлинности данной учетной записи при входе в сеть. Параметры пароля и

безопасности для учетных записей можно настроить указанными ниже способами.

- Потребовать смену пароля при следующем входе в систему

Задаёт требование смены пользователем пароля при следующем входе в сеть. Параметр используется, когда необходима уверенность в том, что никто, кроме пользователя, не знает его пароля.

- Запретить смену пароля пользователем

Не разрешает пользователю менять пароль. Параметр используется при необходимости контролировать учетную запись пользователя, например учетную запись гостя или временную учетную запись.

- Срок действия пароля не ограничен

Снимает временные ограничения на использование пароля.

- Хранить пароль, используя обратимое шифрование

Позволяет пользователю входить в сеть Windows с компьютеров Apple.

- Отключить учетную запись

Не разрешает использовать данную учетную запись для входа в сеть. Многие администраторы используют отключенные учетные записи в качестве шаблонов для часто употребляемых учетных записей пользователей.

- Требовать смарт-карту для интерактивного входа в сеть

Требует наличия смарт-карты для входа в сеть в интерактивном режиме.

- Учетная запись доверена для делегирования

Позволяет службе использовать данную учетную запись для выполнения операций от имени других пользователей в сети.

- Учетная запись важна и не может быть делегирована

- Использовать для данной учетной записи тип шифрования DES

Обеспечивает поддержку шифрования с алгоритмом DES (Data Encryption Standard).

Управление пользователями:

Управление пользователями домена включает в себя следующие операции:

- Создание новой учетной записи пользователя
- Смена пароля пользователя
- Копирование учетной записи пользователя
- Перемещение учетной записи пользователя
- Установка времени входа
- Отключение или включение учетной записи пользователя
- Сопоставление сертификата учетной записи пользователя
- Изменение основной группы пользователя
- Удаление учетной записи пользователя

Создание новой учетной записи пользователя

Чтобы создать новую учетную запись пользователя, используя интерфейс Windows, откройте оснастку Active Directory — пользователи и компьютеры.

В дереве консоли щелкните правой кнопкой мыши папку, в которую добавляется учетная запись пользователя, выделите пункт Создать, а затем выберите команду Пользователь.

В поле Имя введите имя пользователя. В поле Инициалы введите инициалы пользователя. В поле Фамилия введите фамилию пользователя.

В поле Имя входа пользователя введите имя входа пользователя, к которому добавляется суффикс UPN(@ имя домена), а если доменов в сети несколько то выберите суффикс UPN в раскрывающемся списке.

В полях Пароль и Подтверждение введите пароль пользователя, а затем выберите соответствующие параметры пароля.

Смена пароля пользователя, копирование или перемещение учетной записи пользователя, отключение или включение учетной записи, смена основной группы пользователя, а также ее удаление осуществляется выбором соответствующей операции, доступной после щелчка правой кнопкой по учетной записи пользователя.

Чтобы установить время входа, щелкните учетную запись пользователя правой кнопкой мыши и выберите команду Свойства.

На вкладке Учетная запись выберите команду Время входа и установите часы, когда пользователю разрешен или запрещен вход в систему.

Управление компьютерами

Каждый компьютер, который присоединяется к домену, имеет учетную запись компьютера. Так же, как и учетные записи пользователей, учетные записи компьютеров предоставляют возможность проверки подлинности и аудита доступа компьютера к сети и к ресурсам домена. Учетная запись компьютера должна быть уникальной. Учетная запись компьютера создается при подключении компьютера к домену.

Учетные записи компьютеров и пользователей добавляются, отключаются, восстанавливаются и удаляются с помощью оснастки Active Directory — пользователи и компьютеры.

Чтобы выполнить операцию с учетной записью компьютера используя интерфейс Windows, откройте оснастку Active Directory — пользователи и компьютеры, выберите компьютер и соответствующую команду, доступную при щелчке правой кнопкой мыши.

Управление группами

Группы используются для объединения учетных записей пользователей, учетных записей компьютеров и учетных записей групп в управляемые элементы. Использование групп позволяет упростить обслуживание и администрирование сети.

В Active Directory существует два типа групп: группы распространения и группы безопасности. Группы распространения используются только приложениями электронной почты (например, Exchange) для отправки сообщений электронной почты группам пользователей. Группы распространения могут быть использованы для создания списков рассылки электронной почты, а группы безопасности для задания разрешений на использование общих ресурсов.

Группы безопасности обеспечивают эффективное управление доступом к ресурсам сети. Использование групп безопасности позволяет выполнять следующие действия:

- Назначать права пользователя группе безопасности в Active Directory.

- Назначать разрешения на использование ресурсов для групп безопасности.

Не следует путать разрешения с правами пользователей. Разрешения задаются для групп безопасности, использующих общие ресурсы. Разрешения определяют, кто может получить доступ к данному ресурсу и уровень доступа. Права пользователя, заданные для групп безопасности, определяют, что может делать член данной группы в области действий домена или на локальном компьютере рабочей группы.

Группы безопасности перечислены в избирательных таблицах управления доступом, которые определяют разрешения на ресурсы и объекты. Администраторам следует назначать разрешения для ресурсов (общих файлов, принтеров, и т. д.) группам безопасности, а не отдельным пользователям. Разрешения назначаются группе один раз, вместо назначения прав каждому отдельному пользователю. Каждая учетная запись при добавлении к группе получает права, заданные данной группе в Active Directory, и разрешения, определенные для данной группы на ресурсе.

Группы безопасности могут использоваться в качестве адресатов электронной почты, как и группы распространения. Сообщение электронной почты, отправленное группе, отправляется всем членам группы.

Управление группами включает в себя следующие операции:

- Создание новой группы
- Добавление участника группы
- Преобразование типа группы
- Изменение области действия группы
- Удаление группы
- Поиск групп, в которые входит пользователь

Создание новой группы

Чтобы создать новую группу щелкните правой кнопкой мыши папку, в которую добавляется новая группа.

Выделите пункт Создать, а затем выберите команду Группа. Введите имя создаваемой группы. Задавая тип группы, выберите один из предложенных вариантов: группа безопасности или группа распространения.

Задавая область действия группы, выберите один из предложенных вариантов: локальная в домене, глобальная или универсальная (только для групп распространения).

Группы с локальной доменной областью действия могут включать учетные записи и группы с глобальной областью действия, а также группы с универсальной областью действия и группы с локальной доменной областью действия только из данного домена.

Группы с глобальной областью действия могут включать учетные записи и группы с глобальной областью действия из данного домена.

Группы с универсальной областью действия могут включать учетные записи и группы с глобальной областью действия, а также группы с универсальной областью из любого домена.

Добавление участника группы

Чтобы добавить участника группы выберите команду Свойства данной группы и на вкладке «Члены групп» нажмите кнопку Добавить.

В поле Введите имена выбираемых объектов введите имя пользователя, группы или компьютера, добавляемых к выбранной группе и нажмите кнопку ОК. Можно воспользоваться кнопками «Дополнительно» и «Поиск», а затем выбрать добавляемый объект из списка доступных в домене.

Преобразование типа группы

Чтобы преобразовать тип группы выберите папку, включающую группу, тип которой необходимо изменить. В области сведений щелкните правой кнопкой мыши необходимую группу и выберите команду Свойства.

На вкладке Общие в группе Тип группы выберите тип группы.

Изменение области действия группы

Чтобы изменить область действия группы выберите папку, включающую группу, для которой необходимо изменить область действия. В области сведений щелкните правой кнопкой мыши необходимую группу и выберите команду Свойства.

На вкладке Общие в разделе Область действия группы выберите область действия группы.

Удаление группы

Чтобы удалить группу выберите папку, включающую группу, которую необходимо удалить.

На панели сведений щелкните правой кнопкой мыши необходимую группу и выберите команду Удалить.

Поиск групп, в которые входит пользователь

Чтобы найти группы, в которые входит пользователь, в дереве консоли Active Directory щелкните узел Пользователи.

Можно также выбрать папку, содержащую учетную запись пользователя. На панели сведений щелкните правой кнопкой мыши учетную запись пользователя и выберите команду Свойства. Выберите вкладку Член групп и определите, в какие группы входит данный пользователь.

Практическая часть

1. Установить службу ActiveDirectory на виртуальном компьютере с операционной системой WindowsServer с интегрированной службой DNS. С целью коллективного использования системы в учебном процессе строго выполнить следующее указание: при создании домена установить учетную запись администратора со следующими параметрами: имя – администратор, пароль – p@ssw0rd. Создав домен, учащиеся могут создать удобную для себя дополнительную учетную запись с правами администратора.

2. Подключить рабочую станцию к домену, используя учетную запись администратора домена.

3. Продемонстрировать преподавателю полученные результаты и сохранить снимки экранов с основными этапами работы для включения их в отчет о лабораторной работе.

Создание групп

Создайте две группы безопасности с локальной доменной областью действия, например группу Разработчики и группу Менеджеры.

Создайте две группы безопасности с глобальной областью действия, например группу Инженеры и группу Экономисты.

Создание учетных записей пользователей и помещение их в группы

Создайте по одной учетной записи для каждой из созданных групп, задавая в качестве параметра человеческие имена.

Создайте одну учетную запись, которую поместите в каждую из созданных групп.

Включение групп в другие группы

Поместите глобальную группу Инженеры в локальную группу Разработчики, а глобальную группу Экономисты в локальную группу Менеджеры.

Создайте учетную запись для нового компьютера WorkStation1, который предполагается подключить к домену.

Создайте произвольную группу, учетную запись пользователя и учетную запись компьютера. Выполните с ними все описанные в теоретическом разделе работы операции по управлению ими и после показа результатов преподавателю, удалите эти временные объекты.

Контрольные вопросы

1. С какой целью создается домен?
2. Укажите основные этапы установки ActiveDirectory.
3. Какая ОС должна быть установлена на компьютере, что бы была возможна установка ActiveDirectory?
4. Кто имеет право подсоединить компьютер к домену?

Лабораторная работа № 29

Тема: Мониторинг сетевого трафика

Цель работы: ознакомиться с возможностью защиты сетевого трафика SQL Server 2000 при помощи сертификатов и технологии SSL.

Теоретическая часть

Одна из проблем при работе SQL Server заключается в том, что данные передаются по сети в пакетах TDS в полностью открытом виде. Любой пользователь, который сможет перехватить эти пакеты, сможет прочитать как текст запроса, так и возвращаемые сервером данные. В этом легко убедиться на практике при помощи, к примеру, стандартного средства Microsoft для перехвата сетевого трафика - Network Monitor или средств третьих фирм (общее название средств для прослушивания сети - Sniffer).

Защиту от перехвата данных по сети можно реализовать разными способами и на разных уровнях:

- использовать в качестве элементов сетевой инфраструктуры свитчи вместо хабов - поскольку при использовании свитчей между передающими компьютерами создается виртуальный канал и другие пользователи прослушать передаваемую по нему информацию не смогут. Проблемы: свитчи намного дороже хабов, защита, предлагаемая свитчами, легко обходится при помощи специальных утилит;

- использовать стандартные средства шифрования сетевого трафика в Windows (например, IPSec). Проблемы: по IPSec не умеют работать старые клиенты, шифроваться будет не только трафик SQL Server, но и любой другой, что может привести к проблемам с производительностью при передаче данных по сети;

- использовать средство, которое применялось в предыдущих версиях SQL Server (начиная с 6.0) - встроенное шифрование сетевой библиотеки Multiprotocol. Проблемы: средство устарело и не будет поддерживаться в следующих версиях SQL Server;

- использовать шифрование при помощи SSL. Наиболее рекомендованное средство, которое и будет рассмотрено в этом модуле.

Для использования шифрования средствами SSL необходимо вначале получить сертификат от Certificate Services - специального программного компонента Windows, который по умолчанию не устанавливается и его необходимо доустановить вручную. Установка производится стандартными способами через Control Panel -> Add/Remove Programs -> Add/Remove Windows Components. Рекомендуется при установке выбрать режим Stand-Alone Root Certificate Authority.

Устанавливать Certificate Services можно как на компьютер, на котором работает SQL Server, так и на другой компьютер, но в этом случае SQL Server должен иметь возможность постоянно обращаться к этому компьютеру.

Затем вам необходимо определиться с типом шифрования, которое будет использовано для вашей системы - будет ли шифрование реализовано на сервере или оно будет реализовано на клиенте. Если оно будет реализовано на сервере, то:

- будет шифроваться трафик от всех клиентов, обращающихся на этот сервер, без исключения;

- никакие дополнительные настройки на клиентах производить будет не нужно.

Если шифрование будет реализовано на клиенте, то:

- будет шифроваться трафик, передаваемый на сервер, только с этого клиента;

- потребуется установить сертификаты и произвести дополнительные настройки как на сервере, так и на клиентах;

- необходимо будет обеспечить возможность обращения к Certificate Authority как сервера, так и клиентов.

С точки зрения производительности применение SSL будет заметно только при обмене между клиентом и сервером большими объемами данных. При обычной работе применение шифрования практически не скажется на загрузке сервера, поэтому обычно рекомендуется применять шифрование на сервере.

Настройка шифрования по SSL на сервере выглядит так:

1) войти локально от имени учетной записи, от имени которой работает SQL Server. Это должна быть именно доменная учетная запись (или обычная локальная) - локальная системная не подойдет;

2) запросить, получить и установить сертификат типа Server Authentication Certificate с Certificate Services (если Certificate Services находятся на одном компьютере с SQL Server, то необходимо установить на SQL Server SP1 минимум);

3) если установленных сертификатов типа Server Authentication Certificate на компьютере несколько, то необходимо также выбрать нужный сертификат путем правки реестра. Подробно - в статье Knowledge Base 276553 (находится на компакт-диске, прилагаемом к курсу);

4) в Server Network Utility установить флажок Force protocol encryption и перезапустить сервер. Если с сертификатом возникли какие-то проблемы, то сервер просто не запустится.

Настройка шифрования на клиенте - сложнее, описана в той же самой статье Knowledge Base 276553.

Проверить, применяется ли шифрование, можно только при помощи Network Monitor или аналогичных средств.

Практическая часть

Эта лабораторная работа выполняется в паре. Договоритесь с партнером по лабораторной и при необходимости попросите его выполнить запрос на ваш компьютер.

Задание:

1) перехватите при помощи Network Monitor запросы, которые отправлены на ваш сервер, и ответы вашего сервера на них и убедитесь, что обмен информацией с SQL Server по умолчанию проходит открытым текстом;

2) настройте шифрование при помощи SSL на вашем сервере;

3) еще раз перехватите трафик обмена информацией с вашим SQL-сервером и убедитесь, что вся передача информации теперь происходит в зашифрованном виде.

Примечание:

если SQL Server и Certificate Services будут стоять на одном компьютере, убедитесь перед началом выполнения лабораторной, что на SQL Server установлен как минимум SP1.

Решение:

1) Установите Network Monitor на своем компьютере. Для этого откройте в Панели управления консоль Add/Remove Programs, откройте список Add/Remove Windows Components, зайдите в список компонентов Management and Monitoring Tools, установите флажок напротив Network Monitor Tools и нажмите ОК. В случае появления приглашения введите путь к дистрибутиву Windows 2000 на компьютере преподавателя.

2) Запустите и настройте Network Monitor. Запустить Network Monitor можно из меню Start -> Programs -

> Administrative Tools. В ответ на приглашение Select a network выполните в командной строке команду `ipconfig /all`, определите MAC-адрес для сетевого адаптера Local Area Connection и выберите адаптер с этим MAC-адресом в окне Network Monitor.

3) В окне Network Monitor в меню Capture дайте команду Start и постарайтесь сразу же после этого (чтобы не ловить лишний трафик) выполнить с другого компьютера запрос к вашему SQL Server (например, `SELECT * FROM CUSTOMERS` в базе данных Northwind). Просмотрите пакеты, которыми обменялись два компьютера, и убедитесь, что в коде пакетов можно прочесть сам запрос и ответ на него, возвращенный SQL Server.

4) Убедитесь, что SQL Server работает от доменной учетной записи. Если нет, то в этом случае при помощи Enterprise Manager или консоли Services выберите для него доменную учетную запись (например, ту, от которой вы работаете сами) и перезапустите SQL Server.

5) Установите Certificate Services. Для этого еще раз откройте в Панели управления консоль Add/Remove Programs, откройте список Add/Remove Windows Components, установите флажок напротив Certificate Services и нажмите OK. В случае появления приглашения введите путь к дистрибутиву Windows 2000 на компьютере преподавателя. На первом экране Windows Components Wizard выберите Stand-Alone Root SA, далее заполняйте поля в соответствии с указаниями мастера, другие значения оставляйте по умолчанию.

6) После установки Certificate Services войдите на компьютер локально от имени той учетной записи, от которой работает SQL Server, откройте Internet Explorer и перейдите на страницу <http://localhost/certsrv>. На странице Welcome убедитесь, что переключатель стоит в положении Request a certificate и нажмите Next.

7) На странице Choose Request Type переведите переключатель в положение Advanced Request и еще раз нажмите Next.

8) На странице Advanced Certificate Requests убедитесь, что переключатель стоит в положении Submit a certificate request using a form и нажмите Next.

9) На следующей странице в поле Name введите полное имя вашего компьютера в формате FQDN (например, `vancouver.nwtraders.msft`), в списке Intended Purpose выберите Server Authentication Certificate и нажмите Submit.

10) В меню Start->Programs->Administrative Tools откройте консоль Certification Authority, в нем раскройте узел для созданного вами Certification Authority, раскройте контейнер Pending Requests, правой кнопкой мыши щелкните по иконке запрошенного вами сертификата и в меню All Tasks выберите команду Issue. Убедитесь, что сертификат появился в контейнере Issued Certificates и закройте консоль Certificate Authority.

11) В окне Internet Explorer еще раз откройте страницу <http://localhost/certsrv>, переведите переключатель в

положение Check on a pending certificate и нажмите Next. На следующем экране выберите запрошенный вами сертификат и еще раз нажмите Next. В окне Certificate Issued щелкните по ссылке Install this Certificate и нажмите Yes. После появления сообщения о том, что сертификат был успешно установлен, закройте окно Internet Explorer.

12) В меню Start -> Programs -> Microsoft SQL Server откройте Server Network Utility и установите флажок Force protocol encryption. Нажмите на кнопку ОК и перезапустите SQL Server.

13) Еще раз выполните действия, описанные в п. 3, и убедитесь, что просмотреть текст запроса и возвращаемые данные теперь невозможно.

Контрольные вопросы

1. В чем заключается основная проблема при работе SQL Server с позиции защиты данных?
2. Опишите способы реализации защиты от перехвата данных по сети

Список использованных источников

1. Кумскова, И. А. Базы данных: учебник для СПО / И. А. Кумскова.- М.: КНОРУС, 2016.-488 с.
2. Федорова Г.Н. Основы проектирования баз данных: учебное пособие для студ. учреждений сред. проф. образования / Г.Н. Федорова – М.: Издательский центр «Академия», 2016
3. Базы данных. В 2-х кн. Кн. 2. Распределенные и удаленные базы данных [Электронный ресурс]: учебник / В.П. Агальцов. - М.: ИД ФОРУМ: НИЦ Инфра-М, 2013. - 272 с. - Режим доступа: <http://znanium.com/catalog.php?bookin>
4. Шустова Л.И. Базы данных : учебник / Л.И. Шустова, О.В. Тараканов. — М. : ИНФРА-М, 2017. — 304 с. + Доп. материалы [Электронный ресурс; Режим доступа <http://www.znanium.com>]. ——— www.dx.doi.org/10.12737/11549.
5. Мартишин С.А., Симонов В.Л., Храпченко М.В Базы данных. Практическое применение СУБД SQL и NoSQL-типа для применения проектирования информационных систем: Учебное пособие. - М.:ИД ФОРУМ, НИЦ ИНФРА-М, 2017. - 368 с.:
6. Дейт К. Дж. Введение в системы баз данных, 7 – е издание. : Пер. с англ. — М. : Издательский дом «Вильямс», 2016
7. Воробьёв Л.В., Давыдов А.В., Щербина Л.П. Системы и сети передачи информации: учеб. пособие / Л.В.Воробьёв, А.В.Давыдов, Л.П.Щербина – М.: Издательский центр «Академия», 2016
8. Гусева А.И. Вычислительные системы, сети и телекоммуникации: учебник для студентов. / А.И.Гусева, В.С.Киреев – М.: Издательский центр «Академия», 2016
9. Советов Б.Я. Базы данных: теория и практика: учебник / Б.Я.Советов, В.В. Цехановский, В.Д.Чертовской. – 2-е изд. - М.: Издательство Юрайт, 2017
- 10.Фуфаев Э.В. Базы данных: учеб. пособие для студ. учреждений сред. проф. образования / Э.В.Фуфаев, Д.Э.Фуфаев. – 8-е изд., стер. – М.: Издательский центр «Академия», 2016