

Министерство образования Республики Беларусь
Учреждение образования «Витебский государственный
университет имени П.М. Машерова»
Кафедра информатики и информационных технологий

Н.Д. Адаменко

Модели данных и СУБД

Методические рекомендации

В 3 частях

ЧАСТЬ 1

*Витебск
ВГУ имени П.М. Машерова
2015*

УДК 004.65(075.8)
ББК 32.972.32я73
А28

Печатается по решению научно-методического совета учреждения образования «Витебский государственный университет имени П.М. Машерова». Протокол № 1 от 23.10.2015 г.

Автор: доцент кафедры информатики и информационных технологий ВГУ имени П.М. Машерова, кандидат педагогических наук **Н.Д. Адаменко**

Р е ц е н з е н т ы :

кафедра математики и информационных технологий УО «ВГТУ»;
доцент кафедры математики и информационных технологий
УО «ВГТУ», кандидат физико-математических наук *Т.В. Никонова*

Адаменко, Н.Д.

А28 Модели данных и СУБД : методические рекомендации : в 3 ч. / Н.Д. Адаменко. – Витебск : ВГУ имени П.М. Машерова, 2015. – Ч. 1. – 36 с.

Учебное издание содержит описание технологии проектирования баз данных с помощью СУБД MS SQL Server, а также методические материалы для проведения лабораторных занятий, последовательно формирующих основные умения, необходимые для эффективной работы с СУБД MS SQL Server. Материалы предлагаемого издания могут найти применение при изучении дисциплин, связанных с освоением способов разработки многопользовательских систем баз данных: «Модели данных и СУБД» для специальностей «Программное обеспечение компьютерных систем», «Прикладная информатика (веб-программирование и компьютерный дизайн)», «Компьютерная безопасность (радиофизические методы и программно-технические средства)», «Программное обеспечение информационных технологий», «Информационные системы» для специальности «Математика и информатика».

УДК 004.65(075.8)
ББК 32.972.32я73

© Адаменко Н.Д., 2015
© ВГУ имени П.М. Машерова, 2015

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
ЛАБОРАТОРНАЯ РАБОТА № 1 Тема: Установка «Microsoft SQL Server 2008»	5
ЛАБОРАТОРНАЯ РАБОТА № 2 Тема: Проектирование базы данных	16
ЛАБОРАТОРНАЯ РАБОТА № 3 Тема: Создание базы данных. Типы данных, используемые в SQL-сервере. Создание пользовательских типов данных	27

ВВЕДЕНИЕ

В связи с широким распространением корпоративных информационных систем актуальной становится задача подготовки специалистов в области распределенной обработки данных. В данном издании рассмотрены теоретические основы и методы, на которых базируется практика разработки современных многопользовательских систем обработки информации на базе технологии клиент-сервер. Основное внимание уделено технологии проектирования баз данных с помощью ER-диаграмм и способам создания объектов базы – таблиц, запросов, хранимых процедур, триггеров. В процессе выполнения заданий, приведенных в издании, у обучаемых формируется представление о способах поддержки целостности данных. Рассматриваются программные средства автоматизированного контроля соответствия базы данных постоянно изменяющемуся состоянию предметной области.

Методические рекомендации содержат систему лабораторных работ, последовательное выполнение которых обеспечивает формирование устойчивых умений проектирования, построения баз данных с распределенной обработкой. В технологии разработки многопользовательских информационных систем значительное внимание уделяется безопасности данных при их совместной обработке баз данных, поэтому в предлагаемом издании рассматриваются вопросы администрирования баз данных. В частности – управление правами доступа и привилегиями, а также резервное копирование данных и восстановление после сбоев.

Каждая работа включает ряд заданий, выполнение которых способствует освоению основных возможностей СУБД MS SQL Server. Данное учебное издание может быть использовано для самостоятельного освоения приложения. В этом случае необходимо проработать теоретический материал, а затем выполнить один или несколько вариантов заданий.

ЛАБОРАТОРНАЯ РАБОТА № 1

ТЕМА: УСТАНОВКА «MICROSOFT SQL SERVER 2008»

Цель работы: Изучение основных этапов установки и настройки «Microsoft SQL Server».

ОСНОВНЫЕ СВЕДЕНИЯ

Для установки MS SQL Server необходимо наличие установленных пакетов: Microsoft .NET Framework 3.5 SP1, Windows Installer 4.5 и Windows PowerShell 1.0.

1. Запустить программу-установщик (в бесплатной версии Express Edition обычно называется SQLEXP_x86_RUS.exe для 32-разрядной архитектуры или SQLEXP_x64_RUS для 64-разрядной архитектуры процессора) с правами администратора на данном компьютере.

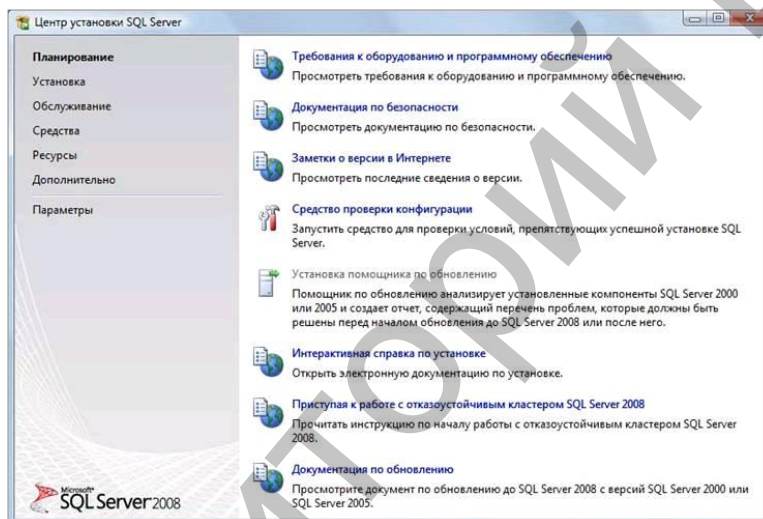


Рисунок 1

2. В разделе «Планирование» нажать пункт «Средство проверки конфигурации»:

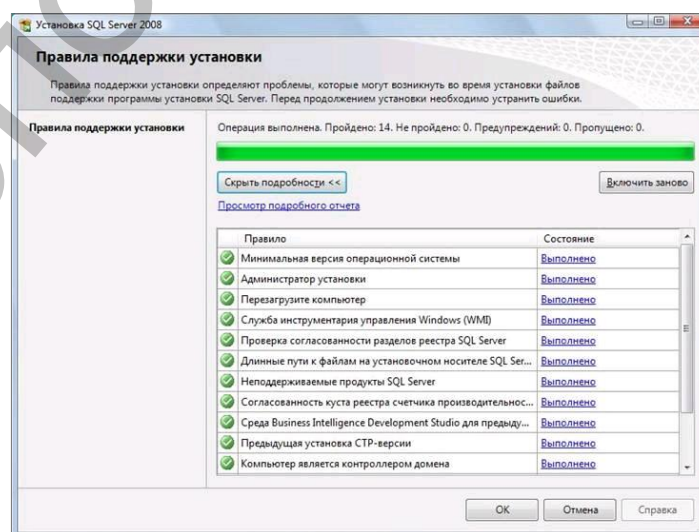


Рисунок 2

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»:

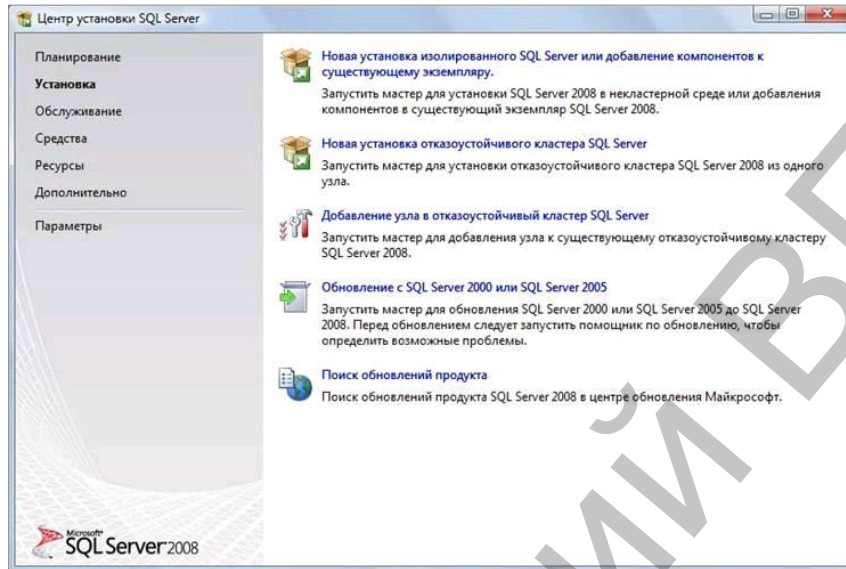


Рисунок 3

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

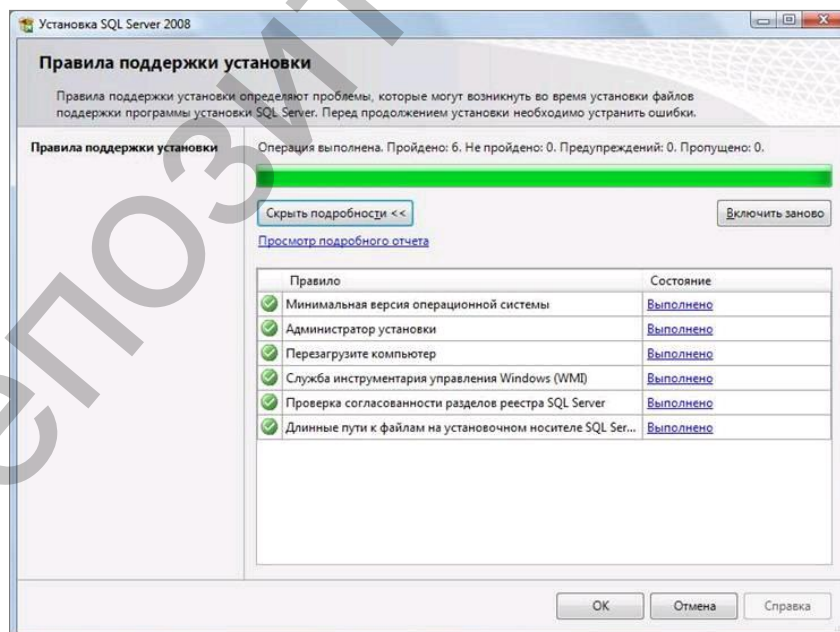


Рисунок 4

5. Нажать кнопку «Показать подробности» и убедиться, что все проверки успешно пройдены. Если будут обнаружены какие-то проблемы,

то необходимо их устранить и запустить повторно проверку кнопкой «включить заново».

Затем нажать кнопку «ОК»

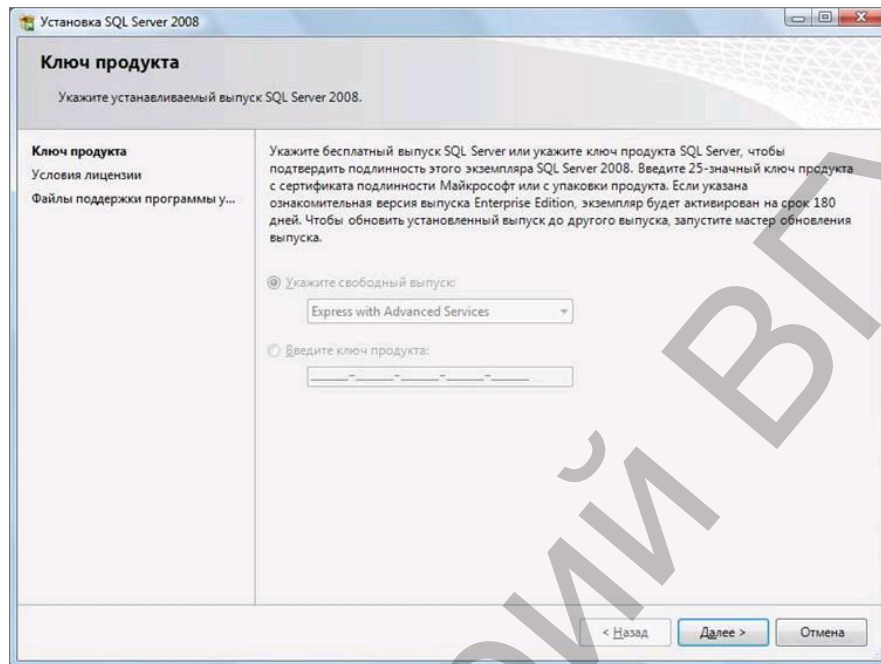


Рисунок 5

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

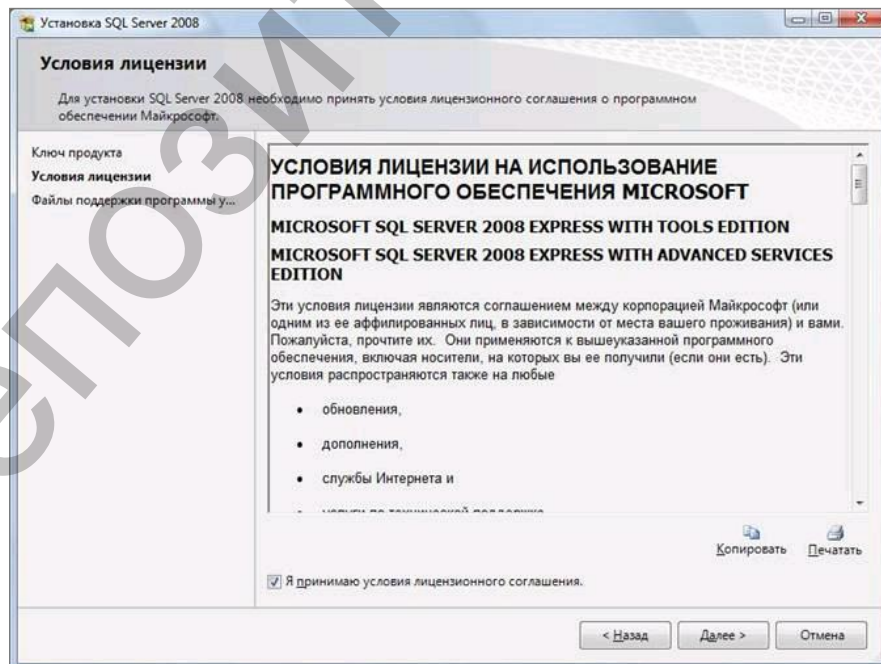


Рисунок 6

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

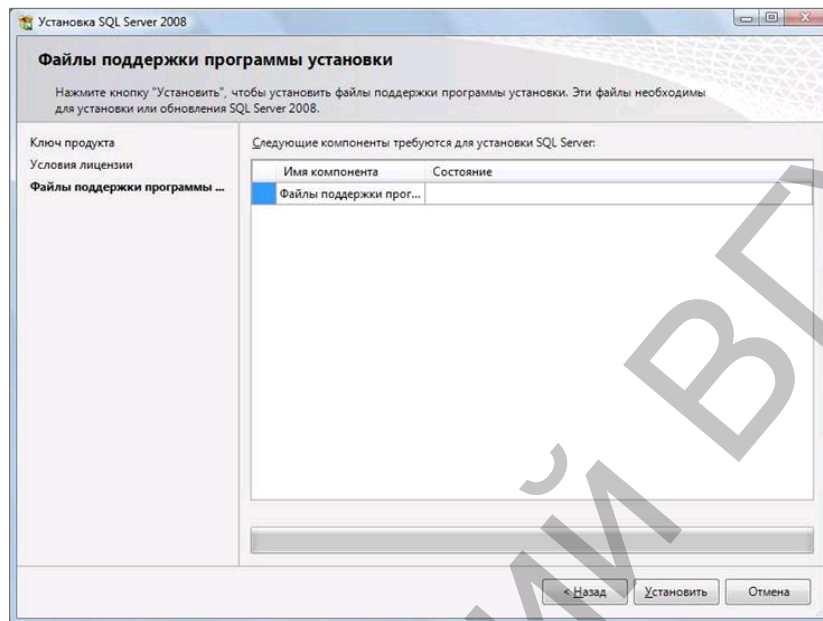


Рисунок 7

8. Нажать кнопку «Установить»:

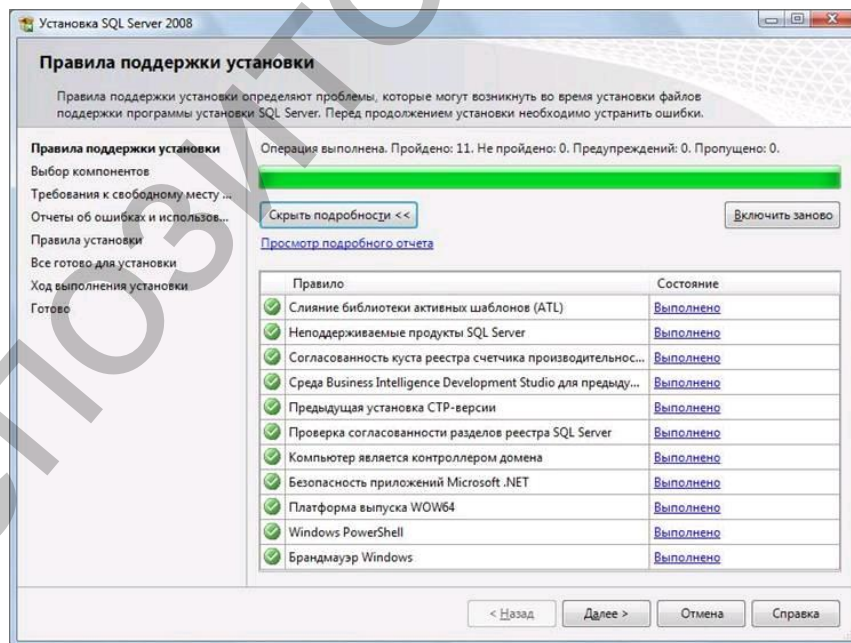


Рисунок 8

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»:
Если появится предупреждение в строке «Брандмауэр Windows», то его можно проигнорировать.

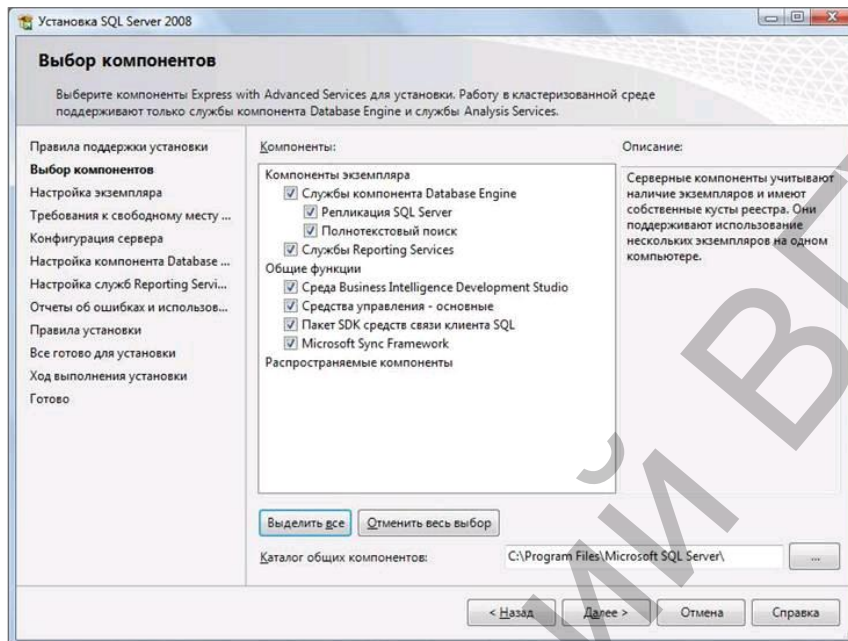
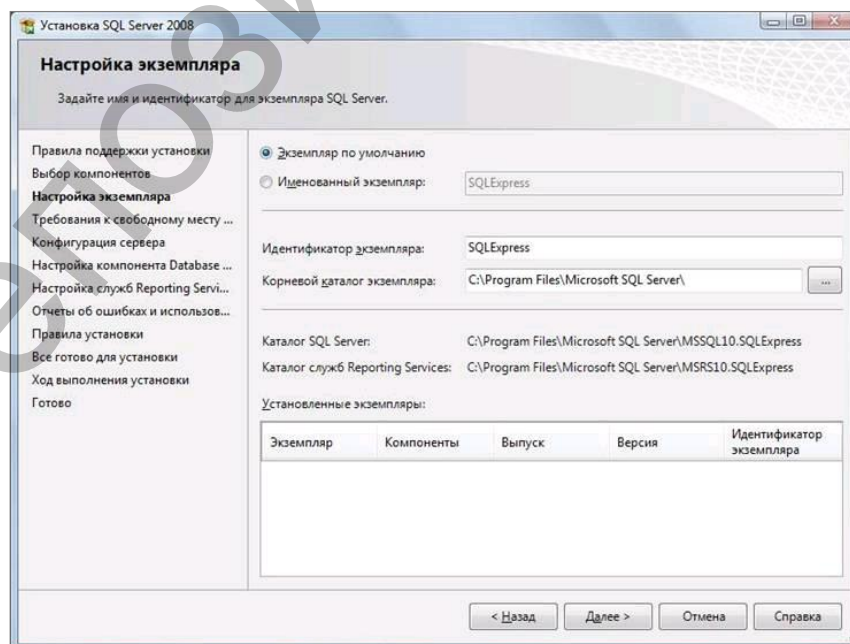


Рисунок 9

10. Выбрать компоненты для установки (можно воспользоваться кнопкой «Выделить все»), и нажать кнопку «Далее»: для управления самим SQL Server на этом этапе следует установить компонент «Средства



управления – основные».

Рисунок 10

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

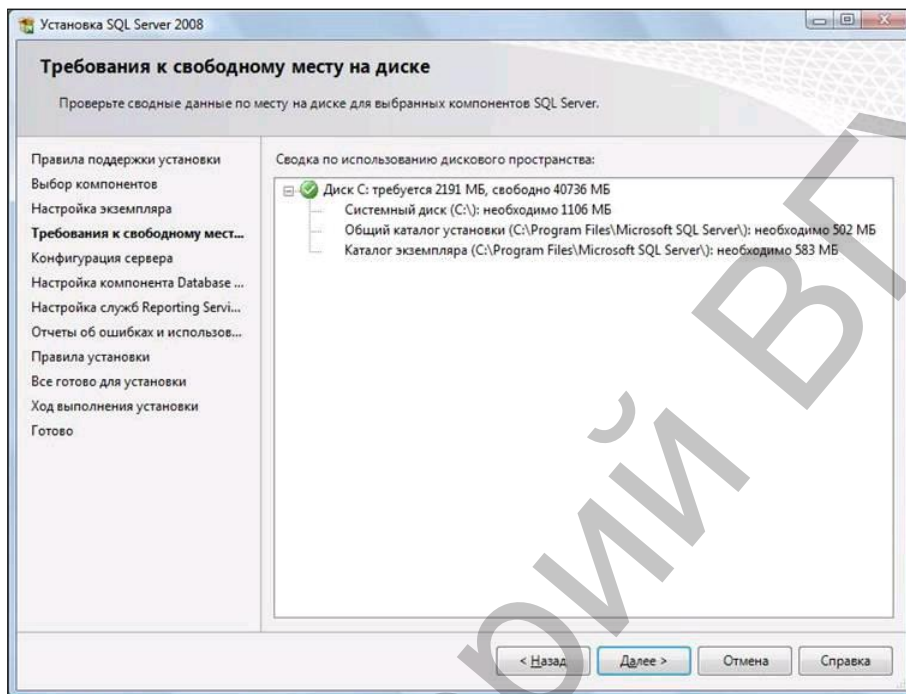


Рисунок 11

12. Нажать кнопку «Далее»:

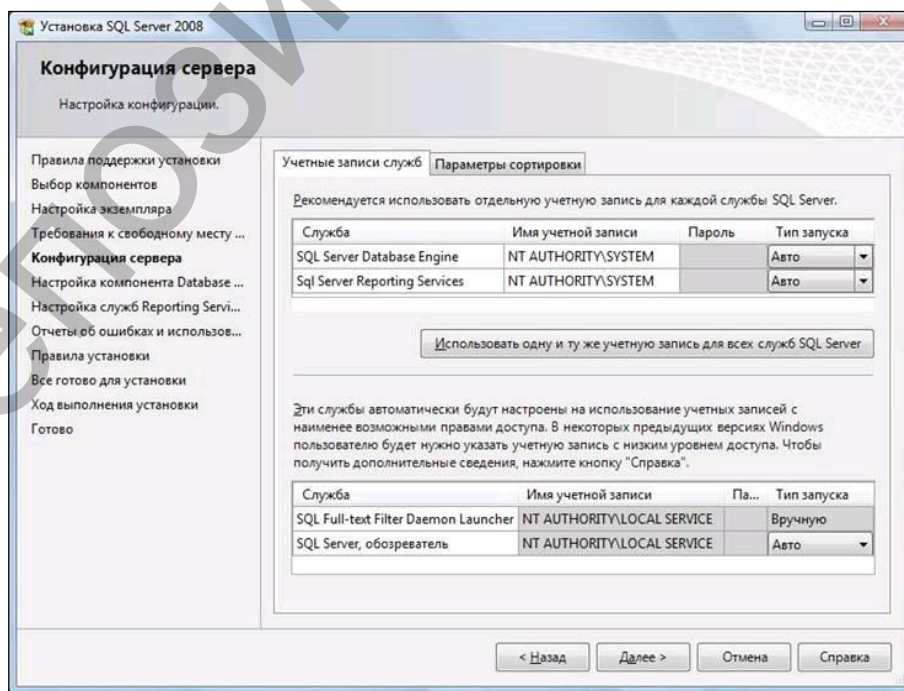


Рисунок 12

13. Выбрать опции, как показано на рисунке 12, и перейти на закладку «Параметры сортировки»:

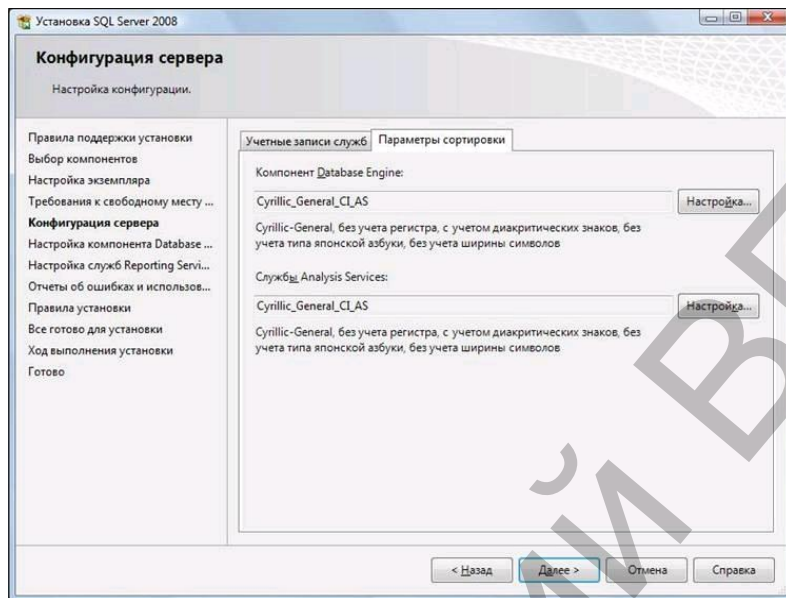


Рисунок 13

14. Выбрать опции, как показано на рисунке 13, и нажать кнопку «Далее». Чтобы изменить опцию нажать расположенную рядом кнопку «Настройка» и установите параметры, как показано на рисунке 14:

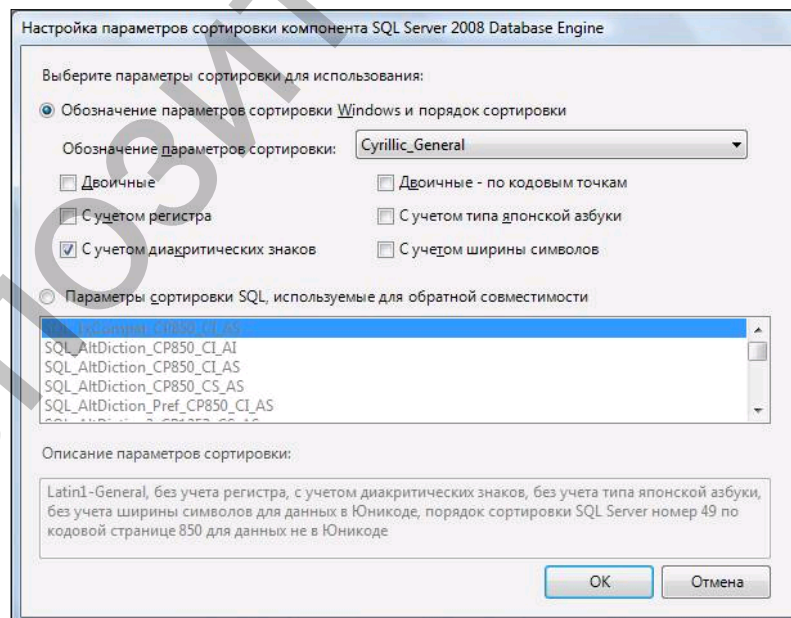
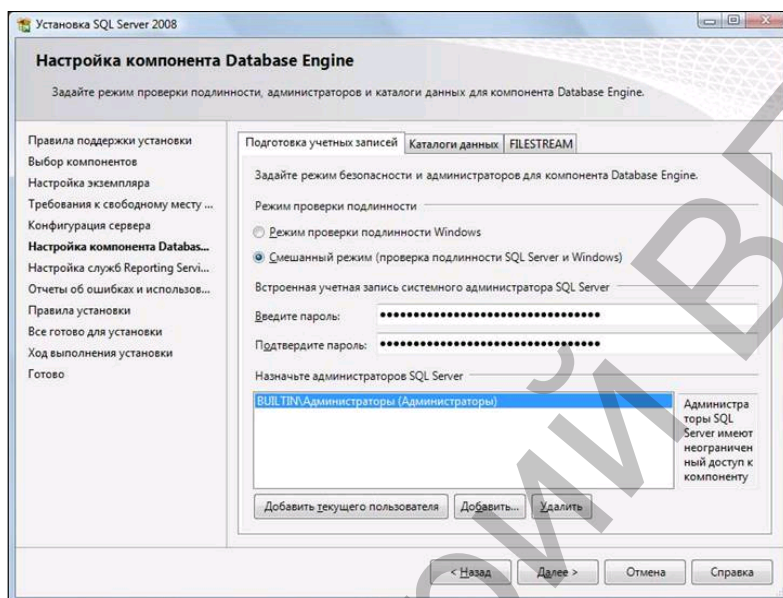


Рисунок 14

Данную настройку нельзя будет изменить после установки.

15. Выбрать опцию «Смешанный режим» и задать пароль (12345) для

встроенной учетной записи администратора «sa» (эта учетная запись обладает максимальными правами доступа ко всем функциям и объектам на SQL-сервере). Дополнительно можно задать учётные записи пользователей Windows или группы пользователей Windows, которые должны обладать максимальными правами доступа к SQL Server. Затем



перейти на закладку «Каталоги данных»:

Рисунок 15

16. В поле «Корневой каталог данных» ввести путь к папке, где будут размещаться файлы баз данных (**Ввести путь D:\SQLDB**), и нажать кнопку «Далее»:

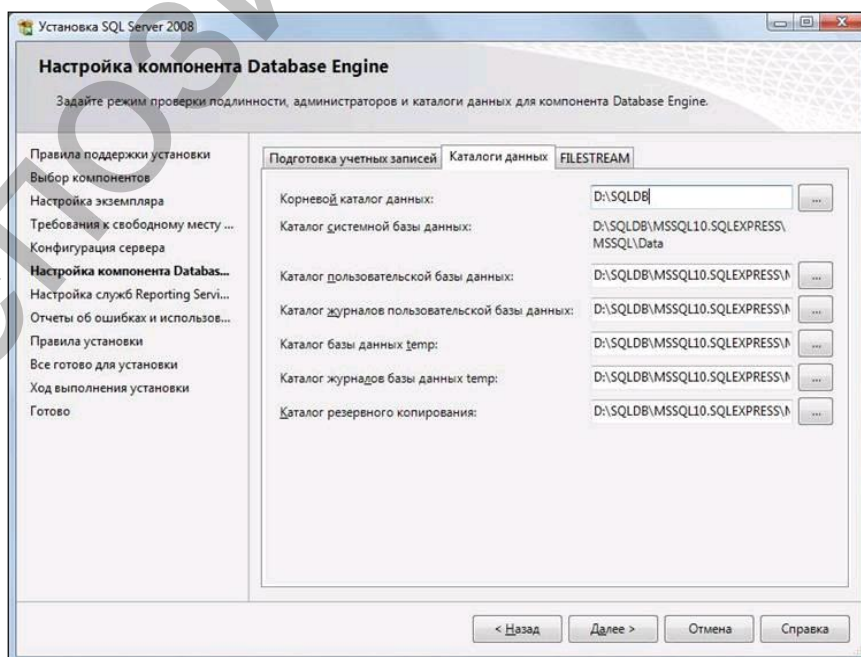


Рисунок 16

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

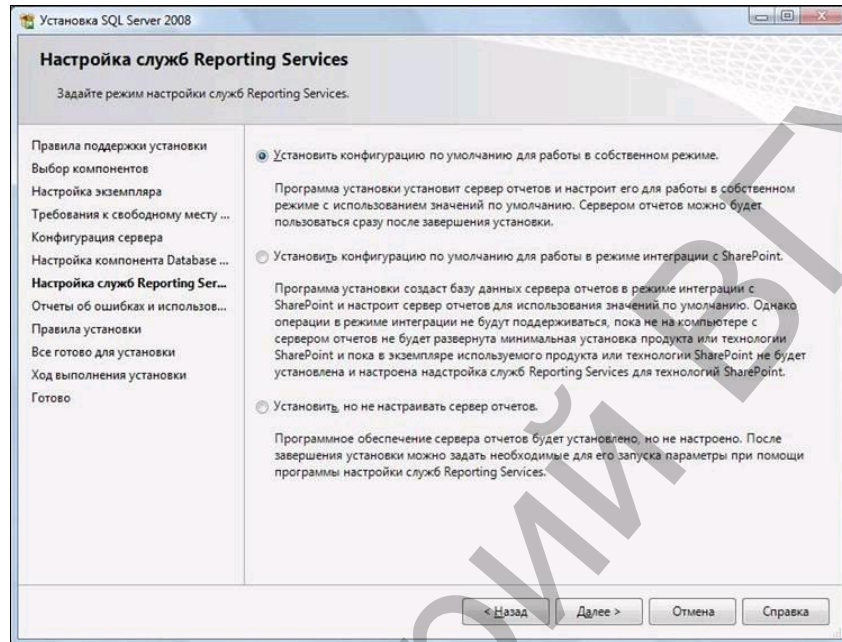


Рисунок 17

18. Нажать кнопку «Далее»:

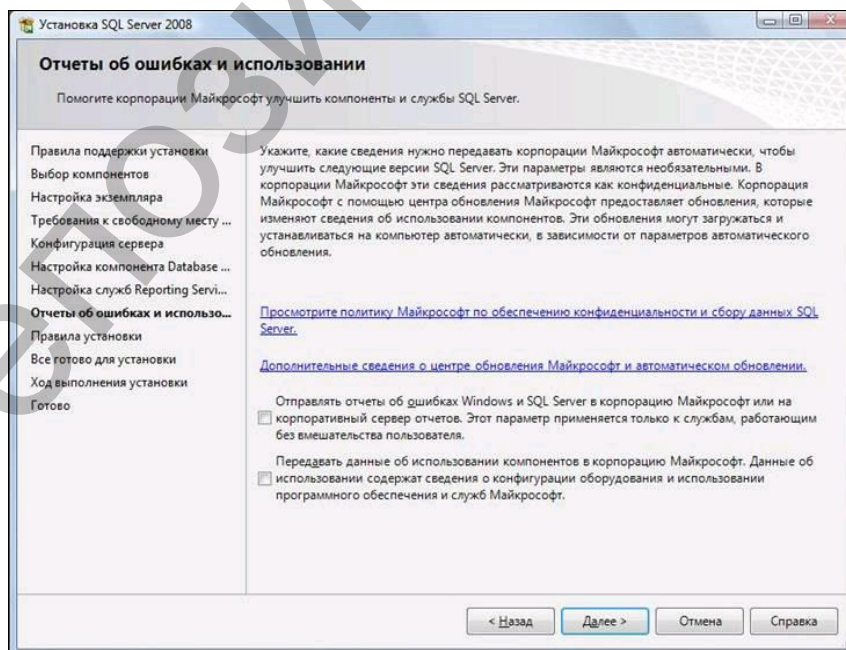


Рисунок 18

19. 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»:

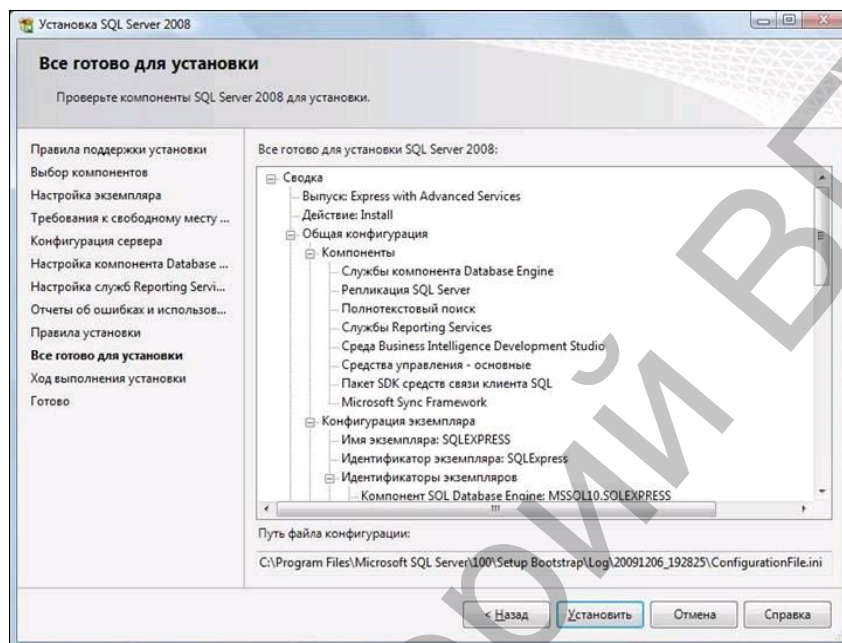


Рисунок 19

20. Нажать кнопку «Установить»:

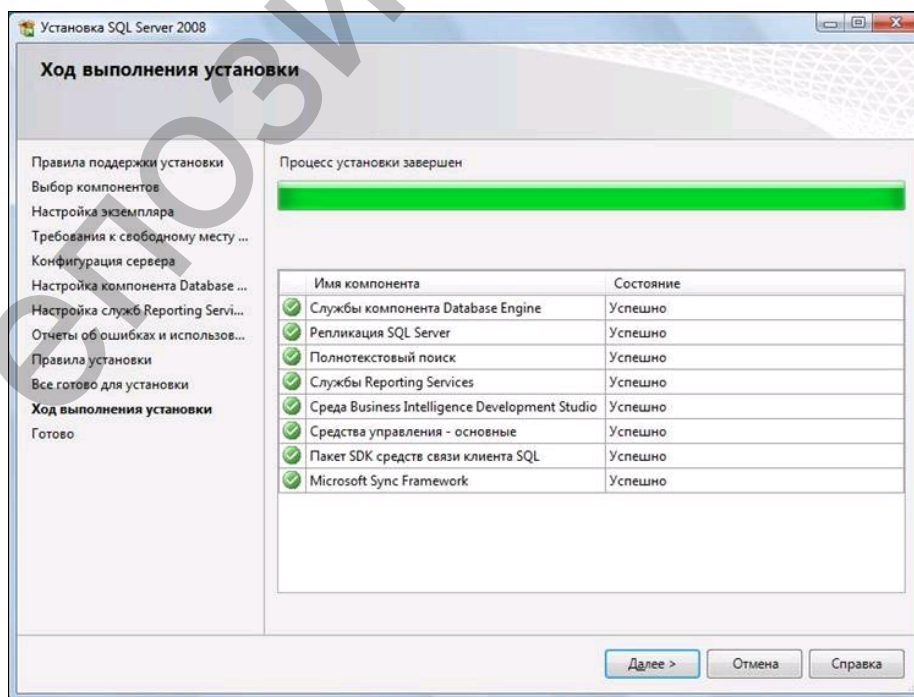


Рисунок 20

21. После завершения установки нажать кнопку «заккрыть»:
После завершения установки и настройки рекомендуется перезагрузить компьютер.

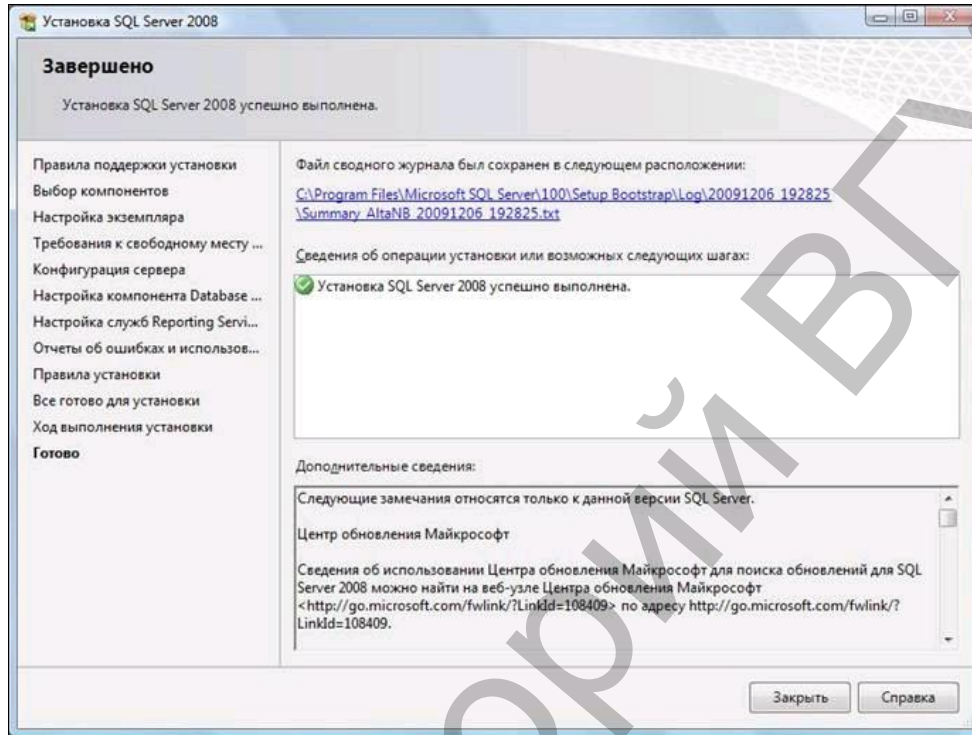


Рисунок 21

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Выполните установку SQL Server в соответствии с инструкцией.
2. Выполните установку MS SQL Management Studio (Запустить программу-установщик SQLManagementStudio_x86Rus.exe или QLManagementStudio_x64_RUS).

ЛАБОРАТОРНАЯ РАБОТА № 2

ТЕМА: ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ.

Цель работы: Изучить методы проектирования БД на основе инфологического моделирования. Ознакомиться с основными этапами разработки концептуального представления и логической структуры базы данных.

ОСНОВНЫЕ СВЕДЕНИЯ

Создание инфологической модели методом «сущность-связь».

Создание ER-диаграмм

Предварительный этап создания инфологической модели предусматривает выполнение системного анализа и словесного описания информационных объектов предметной области. На первом этапе проектирования создаётся концептуальная схема БД, которая затем преобразуется к реляционной схеме. В результате создаётся реляционная БД в 3-ей нормальной форме. Рассмотрим одну из наиболее важных и распространенных семантических моделей данных – модель «Сущность-связь» (ER-модель). Основные понятия ER- модели: сущность, связь и атрибут.

Сущность – это реальный или представляемый объект, информация о котором должна сохраняться и быть доступна. В диаграммах ER- модели сущность представляется в виде прямоугольника, содержащего имя сущности. При этом имя сущности (в виде существительных) – это имя типа, а не некоторого конкретного элемента этого типа. Каждый элемент сущности должен быть отличим от любого другого элемента этой же сущности.

Связь – это графически изображаемая ассоциация, устанавливаемая между двумя сущностями. Эта ассоциация всегда является бинарной и может существовать между 2-мя сущностями или между сущностью и ею же самой (рекурсивная связь). В любой связи выделяются 2 конца, на каждом из которых указывается имя конца связи (в виде глагола), степень конца связи (сколько элементов данной сущности связывается) и обязательность связи (то есть, любой ли элемент данной сущности должен участвовать в этой связи).

Связь представляется в виде линии, соединяющей 2 сущности, при этом в месте соприкосновения связи с сущностью используется множественный вход в прямоугольник, если для связи могут использовать несколько элементов, и единичный – если в связи может участвовать только один элемент сущности. Обязательность связи изображается перпендикуляром, а необязательность – окружностью.

Примеры:



Конец связи **имеет**, означает, что в каждом отделении компании работает 1-5 и более сотрудников. Конец связи с именем **работает** позволяет соединить с одним сотрудником не более 1-го отделения, то есть сотрудник не может работать одновременно в нескольких отделениях. Трактовка изображенной диаграммы следующая: каждый сотрудник может работать в одном из отделений компании (не все сотрудники обязательно принадлежат отделениям, так как у компании может быть свой офис и административный аппарат), каждое отделение обязательно имеет 1–5 и более сотрудников.

Атрибутом сущности является любой элемент, который служит для уточнения, идентификации, классификации, числовой характеристики или выражения состояния сущности. Имена атрибутов заносятся в прямоугольник, записываются малыми буквами под именем сущности, возможно с примерами.

Некоторый набор атрибутов назначается уникальным идентификатором (**ключом**).

Как и в реляционных схемах БД, так и в ER-схемах вводится понятие нормальных форм, причём их смысл очень близко соответствует смыслу реляционных нормальных форм.

Определения 3-х первых нормальных форм (НФ):

В 1 НФ ER-схемы устраняются повторяющиеся атрибуты или группы атрибутов. Во 2 НФ устраняются атрибуты, зависящие от части уникального идентификатора. Эта часть уникального идентификатора определяет отдельную сущность.

В 3 НФ устраняются атрибуты, зависящие от атрибутов, не входящих в уникальный идентификатор. Они являются основой отдельной сущности.

В ER-модели допускается принцип **категоризации** сущностей.

Это значит, что, как и в объектно-ориентированных языках программирования, вводится понятие подтипа сущности (категории). Все подтипы сущностей рассматриваются как взаимоисключающие. Сущность, на основе которой строятся подтипы, называется супертипом. Как в языках объектно-ориентированного программирования, вводится возможность наследования типа сущности, исходя из одного или нескольких супертипов.

Эти элементы ER-модели делают её более мощной, но существенно усложняют её использование. Например, сущность может быть расщеплена на 2 или более взаимоисключающих подтипа, каждый из которых включает общие атрибуты. В подтипах могут определяться собственные атрибуты и связи.

Преобразование ER-модели в реляционную схему осуществляется в соответствии со следующими правилами:

- каждая простая сущность превращается в отношение. Имена отношений

могут отличаться от имен сущностей, так как могут быть ограничены требованиями конкретной СУБД;

- каждый атрибут становится возможным столбцом с тем же именем, для каждого атрибута задается допустимый тип данных и обязательность или необязательность этого атрибута;
- компоненты уникального идентификатора сущности превращаются в первичный ключ отношения;
- в каждое отношение, соответствующее подчиненной сущности, добавляется набор атрибутов основной сущности, являющейся первичным ключом основной сущности. В отношении, соответствующем подчиненной сущности этот набор атрибутов становится внешним ключом.
- Для связи М:М используется специальный механизм преобразований, который позволяет отразить множественные связи, неспецифичные для реляционной модели. Это делается введением дополнительного связующего отношения, которое связано с каждым исходным связью 1:М, атрибутами этого связующего отношения являются первичные ключи связываемых отношений. При этом каждый из атрибутов нового отношения является *внешним ключом*, а вместе они образуют первичный ключ новой связующей сущности.

Спроектируем инфологическую модель системы, предназначенной для компании, которая занимается продажей объектов недвижимости. Компания предлагает следующие услуги:

- сбор информации об объектах, выставляемых на продажу;
- представление данных в общую БД;
- организация просмотра объектов потенциальными покупателями;
- составление договоров на продажу недвижимости.

Компания имеет несколько отделений (агентств), расположенных в разных районах города и районных центрах. При этом компания состоит не только из отделений, так как у компании может быть свой офис и административный аппарат, следовательно, не все сотрудники обязательно принадлежат отделениям. В каждом отделении компании есть (но не весь) персонал, отвечающий за работу с выставленными на продажу объектами недвижимости.

Описание предметной области

Выделим объекты предметной области:

- Каждое отделение компании может быть описано с помощью следующих характеристик: уникальный номер отделения, его адрес (почтовый индекс, город, район, улица, дом), номер телефона и номер факса.
- В штат входят работники, называемые торговыми агентами. Торговые агенты занимаются продажей недвижимости. Информация, описывающая каждого сотрудника компании, включает персональный

номер, полное имя (имя и фамилию), адрес проживания, номер телефона, пол, дату рождения, занимаемую должность, а также номер и адрес отделения компании, в котором он работает. Личный номер каждого работника является уникальным в пределах всех отделений компании.

- Данные обо всей выставленной на продажу недвижимости можно получить в любом отделении компании. Информация, описывающая каждый объект недвижимости, включает номер объекта, адрес его местонахождения (почтовый индекс, город, район, улица, дом и квартира), тип объекта, количество комнат в нём, отпускную цену, а также имя и адрес владельца этого объекта. Каждый объект недвижимости имеет единственного владельца.
- Компания управляет недвижимостью частных лиц. Частный владелец идентифицируется собственным номером, уникальным для всех отделений компании. Дополнительная информация о владельцах включает фамилию, имя, адрес и номер телефона. Каждому владельцу принадлежит, по крайней мере, один объект недвижимости.
- Потенциальный покупатель обращается в отделение компании, в котором ему могут предложить осмотреть разные объекты недвижимости. Информация, сохраняемая по каждому проведённому осмотру объекта, включает имя и адрес клиента, номер и адрес осмотренного объекта недвижимости, дату осмотра, а также комментарии по результатам осмотра. Клиент может осматривать любое количество объектов недвижимости.
- О каждом клиенте хранится следующая информация: фамилия и имя, адрес, номер телефона, предпочтительное количество комнат в покупаемой квартире, а также максимальная цена, которую клиент согласен уплатить. Каждый клиент получает личный номер, уникальный для всех отделений компании.
- При покупке некоторого объекта покупатель заключает с компанией договор на покупку выбранного им объекта (квартиры). Подробная информация о договоре на покупку включает: номер соглашения, дату заключения договора, личный номер покупателя, его имя и адрес, номер покупаемой квартиры и адрес её местонахождения, стоимость квартиры с учётом комиссионных, а также сведения о сотруднике, который составил данный договор. Клиент может купить как один, так и сразу несколько объектов недвижимости.

Необходимо предусмотреть следующие ограничения на информацию в системе:

1. В каждом отделении компании работает, по крайней мере, 5 сотрудников, а максимальное их количество не ограничено.
2. Каждый сотрудник может отвечать не более чем за 10 объектов недвижимости одновременно.

3. Компания требует сохранять данные об уволившемся сотруднике в течение года.
4. Каждый покупатель при обращении в агентство должен оставить свой номер телефона (рабочий или домашний) для быстрой с ним связи.

В данной информационной системе должны реализовываться определенные задачи, за выполнение которых несут ответственность сотрудники компании. А именно:

- Создание и корректировка записей с данными о сотрудниках каждого отделения.
- Создание отчёта со сведениями о сотрудниках каждого отделения.
- Удаление сведений об уволившемся сотруднике из базы данных и передача ответственности за все курируемые им объекты недвижимости другому сотруднику.
- Создание и корректировка записей с данными о выставленных на продажу объектах недвижимости в конкретном отделении компании.
- Создание отчета с данными о выставленных на продажу объектах недвижимости в данном отделении компании.
- Создание и корректировка записей с описанием потенциальных покупателей и их требований.
- Поиск всех объектов недвижимости, удовлетворяющих требованиям покупателя.
- Поиск возможного покупателя для вносимого в базу данных объекта недвижимости.
- Создание и корректировка записей со сведениями об осмотре объектов недвижимости.
- Создание и корректировка записей со сведениями о заключённых договорах.
- Распечатка договора.
- При заключении договора на объект он должен автоматически удаляться из списка объектов недвижимости, выставленных на продажу.

Описание сущностей и типов связей

Определим основные типы сущностей исходя из описания предметной области.

1. Отделение (BRANCH)

Каждое отделение имеет следующий набор атрибутов: номер отделения, адрес (почтовый индекс, город, улица, дом), номер телефона и номер факса.

2. Сотрудник (STAFF)

Каждый сотрудник характеризуется следующими атрибутами: номер сотрудника, фамилия, имя сотрудника, дата рождения, пол, адрес (город, улица, дом, квартира), номер домашнего телефона, дата зачисления в штат, должность и зарплата.

3. Объект недвижимости для продажи (PROPERTY)

Характеризуется такими атрибутами как: номер объекта недвижимости, дата регистрации, полный адрес (почтовый индекс, город, улица, дом и квартира), тип объекта (N-этажный панельный или кирпичный), этаж, количество комнат, площадь (общая, жилая, площадь кухни), балкон (балкон, лоджия, застеклен, их количество или отсутствие), наличие телефона (есть, нет) и отпускная цена.

4. Владелец (OWNER)

Владелец имеет атрибуты: номер владельца, фамилия, имя, адрес и номер телефона.

5. Покупатель (BUYER)

Каждый покупатель характеризуется следующими атрибутами: код покупателя, фамилия и имя, адрес (город, улица, дом, квартира) номер телефона (рабочий и домашний), предпочтительное количество комнат, максимальная цена.

Определим типы связей, которые существуют между основными сущностями.

Между сущностями **Отделение** и **Сотрудник** существует связь 1:M, обязательная только с одной стороны. Связь обязательна только с одной стороны, т.к. каждое из отделений компании **имеет** несколько штатных сотрудников, но не все сотрудники компании работают в отделениях. В обратном направлении, каждый из сотрудников отделений **работает** только в одном из них. Ключевой атрибут сущности **Отделение**: **Номер отделения (Branch_no)**. Ключевой атрибут сущности **Сотрудник**: **Номер сотрудника (Staff_no)**.

Из описания предметной области известно, что каждый объект недвижимости, выставленный на продажу, **закрепляется** за конкретным отделением компании. В каждом отделении компании есть сотрудник, отвечающий за работу с выставленными на продажу объектами недвижимости. Для отражения этой ситуации необходимо провести связь между сущностями **Объект недвижимости для продажи** и **Отделение** (ключевым атрибутом сущности **Объект недвижимости для продажи** является **Номер объекта недвижимости (Property_no)**). Для того чтобы узнать, какой объект недвижимости **обслуживается** каким сотрудником и, с другой стороны, какой сотрудник **отвечает** за данный объект, вводится дополнительная связь между сущностями **Объект недвижимости для продажи** и **Сотрудник**. Между сущностями **Отделение** и **Объект недвижимости для продажи** установлена связь 1:M, обязательная с 2-х сторон. Между сущностями **Сотрудник** и **Объект недвижимости для продажи** – связь 1:M, необязательная с 2-х сторон, так как из всех сотрудников компании только торговые агенты занимаются продажей недвижимости и отвечают за работу с ними. В обратном направлении, объект может быть не связан ни с одним из сотрудников. Например, когда

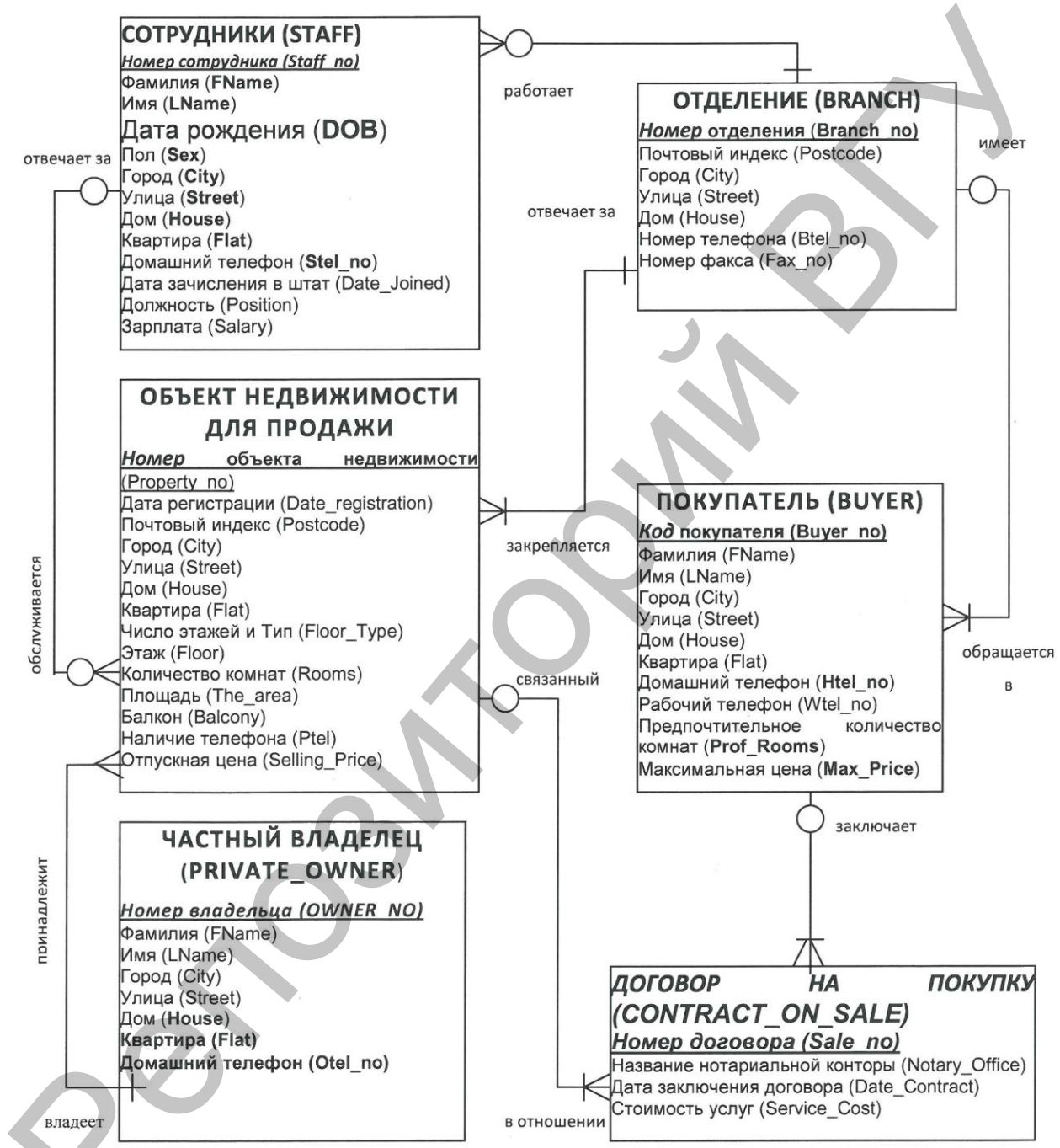
объект впервые регистрируется в компании.

Теперь необходимо отразить связь между сущностями **Владелец** (атрибут *Номер владельца (Owner_no)* является ключевым) и **Объект недвижимости для продажи**. Если рассмотреть эту связь с одной стороны, то можно заметить, что один владелец может владеть несколькими объектами недвижимости. С другой стороны, каждый объект **принадлежит** только одному владельцу. Следовательно, связь между сущностями - 1:М. Поскольку каждый владелец **владеет**, по крайней мере, одним объектом недвижимости, а каждый объект должен иметь одного владельца, связь является обязательной с обеих сторон.

Из описания предметной области известно, что потенциальный покупатель **обращается в** одно из отделений компании (только когда клиент становится потенциальным покупателем агентства, данные о нём, заносятся в базу данных), в котором ему могут предложить осмотреть разные объекты недвижимости. Клиент, как правило, желает осмотреть один или несколько, предлагаемых ему объектов недвижимости. Сведения о таком просмотре включают дату осмотра объекта и комментарии потенциального покупателя (согласен он или нет купить данную квартиру и др.). Образуется необязательная с двух сторон связь М:М между сущностями **Покупатель** и **Объект недвижимости для продажи**. Отдельный клиент может осмотреть несколько выставленных на продажу объектов (1:М), а каждый объект может быть **осмотрен** несколькими клиентами (1:М). Связь необязательна со стороны клиента из-за возможного отсутствия объекта, отвечающего его требованиям. С другой стороны, сведения о некоторых объектах просто регистрируются в компании, а осмотр их клиентами не производится.

Если клиент согласен купить некоторый объект, то он заключает с компанией договор на покупку выбранного им объекта. Сотрудник компании должен оформить это соглашение. Каждый объект может быть продан единственному клиенту, и каждый клиент может купить один или более объектов в одно и то же время. Образуется необязательная с двух сторон связь 1:М между сущностями **Покупатель** и **Объект недвижимости для продажи**. Но так как, всякий раз, при покупке клиент **заключает** договор с компанией, мы определим две связи. Связь 1:М между сущностями **Покупатель** и **Договор на покупку**, а также связь 1:М между сущностями **Объект недвижимости для продажи** и **Договор на покупку**. Связи обязательны со стороны сущности **Договор на покупку**. Ключевым атрибутом для сущности **Покупатель** является *Код покупателя (Buyer_no)*, а для сущности **Договор на покупку** – атрибут *Номер договора (Sale_no)*. Кроме этого сущность **Договор на покупку** (CONTRACT_ON_SALE) имеет атрибуты: название нотариальной конторы, дата заключения договора, стоимость услуг.

Инфологическая модель предметной области



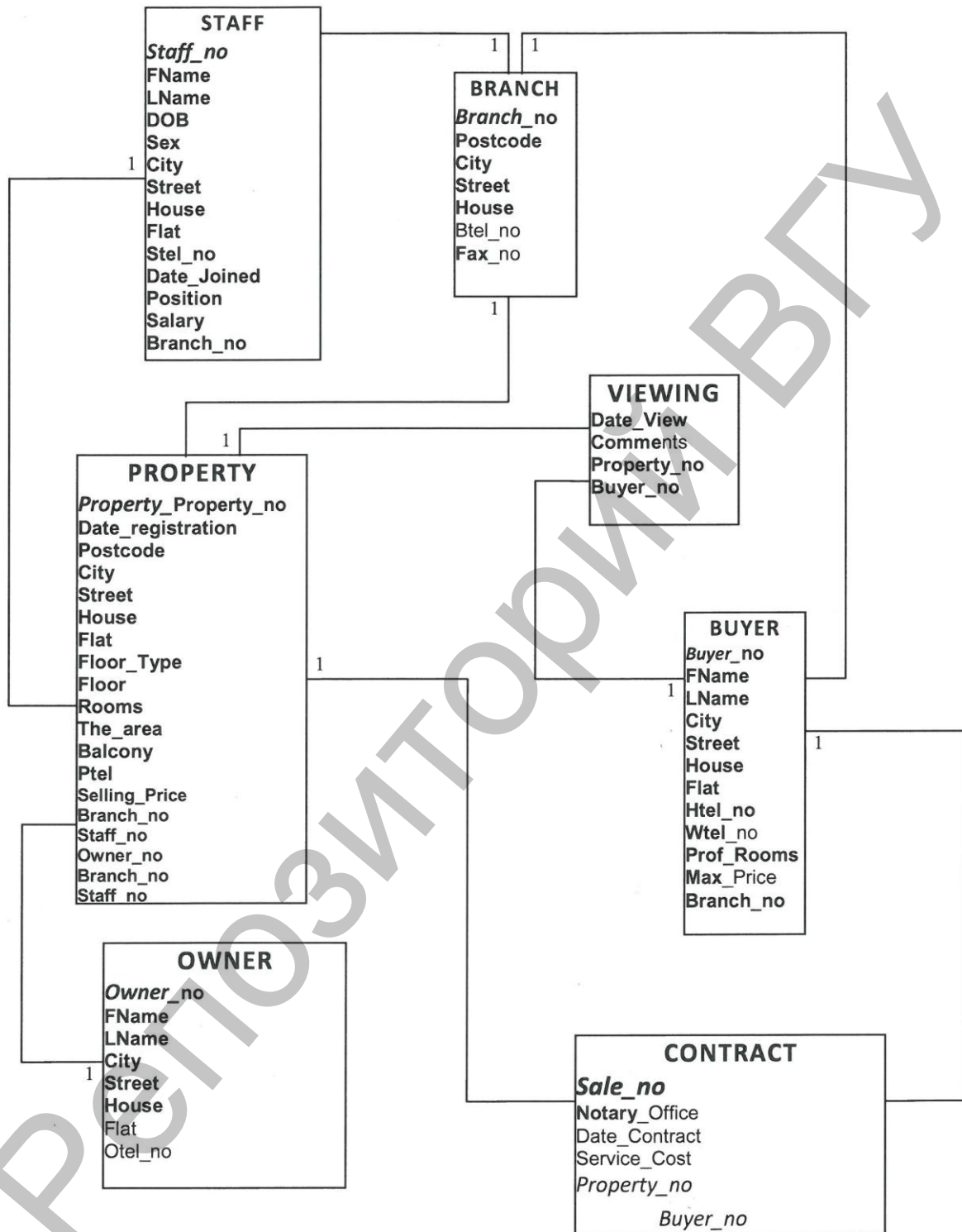
Переход к реляционной модели

В реляционной модели связи явным образом не отображаются, однако между отношениями поддерживаются иерархические связи (в каждой связи одно отношение выступает как основное, а другое как подчинённое). Это значит, что один кортеж основного отношения может быть связан с несколькими кортежами подчинённого отношения. Для поддержки этих связей оба отношения должны содержать наборы атрибутов, по которым они связаны. В основном отношении это первичный ключ отношения. В подчинённом отношении для моделирования связи должен присутствовать набор атрибутов, соответствующий первичному ключу основного отношения. Данный набор атрибутов в подчинённом отношении принято называть внешним ключом. Согласно правилу 4 перехода к реляционной модели (в каждое отношение, соответствующее подчинённой сущности, добавляется набор атрибутов основной сущности, являющейся ею первичным ключом), введём в дополнительное отношение **Договор на покупку** ключи отношений **Покупатель** и **Объект недвижимости для продажи**.

Для связи М: М между сущностями **Покупатель** и **Объект недвижимости для продажи** введём дополнительное связующее отношение, которое связано с каждым исходным связью 1:М.. Атрибутами этого связующего отношения, помимо даты осмотра и комментарии, будут первичные ключи связываемых отношений, т.е. *Property_no* и *Buyer_no*. Для нового отношения они являются внешними ключами, а вместе они образуют первичный ключ новой связующей сущности **Осмотр (VIEWING)**.

ОСМОТР (VIEWING) Дата осмотра (Date_View) Комментарии (Comments) Property_no Buyer_no
--

Реляционная модель



КОНТРОЛЬНЫЕ ВОПРОСЫ

1. С чего начинается разработка базы данных?
2. Охарактеризуйте основные понятия ER- модели: сущность, связь и атрибут.
3. Каким образом на ER диаграмме отображается обязательность и необязательность связи.
4. Перечислите правила преобразования ER-модели в реляционную схему.

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Выполните проектирование базы данных (ER-модель, реляционную схему).
2. Исходные данные в соответствии с вариантом задания находятся в Приложении 1 части 3.

ЛАБОРАТОРНАЯ РАБОТА № 3

ТЕМА: СОЗДАНИЕ БАЗЫ ДАННЫХ. ТИПЫ ДАННЫХ, ИСПОЛЬЗУЕМЫЕ В SQL-СЕРВЕРЕ. СОЗДАНИЕ ПОЛЬЗОВАТЕЛЬСКИХ ТИПОВ ДАННЫХ

Цель работы: Изучить различные способы создания баз данных, рассмотреть основные типы данных и способы создания пользовательских типов данных.

ОСНОВНЫЕ СВЕДЕНИЯ

Создание базы данных

База данных (БД) – некоторый набор таблиц, связанных между собой определенной связью для выполнения различного рода задач.

Создание базы данных в системе SQL-сервер может осуществляться следующими способами: используя стандартные команды языка T-SQL и MS SQL Server Management Studio.

Первый способ: С помощью оператора CREATE DATABASE. В процессе создания базы также создается журнал транзакций (с помощью оператора LOG ON).

Создание базы данных – это процесс указания имени базы и определения размеров и размещения файлов базы данных (первичного и вторичных файлов базы данных, файла журнала транзакций). В primary файле базы данных (расширение .mdf) записывается информация об основных её объектах – таблицах, индексах и т.д., а в файл журнала транзакций (расширение .ldf) информация о процессе работы с транзакциями (контроль целостности данных, состояние базы данных до и после выполнения транзакции). Если в процессе использования базы данных планируется размещение её на нескольких дисках, то в этом случае создаются secondary файлы (расширение .ndf). По умолчанию базы данных имеют право создавать только те пользователи, которым назначены роли *sysadmin* и *dbcreator*.

Синтаксис:

```
CREATE DATABASE имя_базы_данных
[ON
[PRIMARY] (NAME=логическое_имя_файла,
FILENAME='физическое_имя_файла'
[, SIZE=размер]
[, MAXSIZE=максимальный_размер]
[, FILEGROWTH=шаг_приращения_размера])
[, ...n]
]
[LOG ON
(NAME=логическое_имя_файла_журнала,
```

FILENAME='физическое_имя_файла_журнала'
[, SIZE=размер_журнала])

[, ...n]

]

[FOR RESTORE]

При этом следует помнить, что пробелы используются для разделения элементов команды SQL, и поэтому они не могут быть частью имени базы данных, таблицы или какого-либо другого создаваемого объекта.

При создании базы данных можно указать следующие параметры:

- PRIMARY указывает файлы основной группы файлов, которая содержит все системные таблицы базы данных. Кроме того, здесь также содержатся объекты, не привязанные к пользовательским группам файлов. В любой базе данных должен быть лишь один основной (первичный) файл данных. Он служит отправной точкой базы данных и указывает на все прочие её файлы. Стандартное расширение имени основного файла данных – .mdf. Если ключевое слово PRIMARY опущено, основным файлом становится первый файл в операторе;
- FILENAME – задаёт физическое имя и путь к файлу. В пути (*физическое_имя_файла*) необходимо указывать папку локального диска сервера, на котором установлен SQL Server;
- 1. SIZE – указывает размер файла: в мегабайтах, тогда используется суффикс MB (по умолчанию), или в килобайтах – в этом случае применяется суффикс KB. Минимально возможное значение – 512 кб. Параметр *size* задает минимальный (начальный) размер файла. Файл может увеличиваться, однако его нельзя сжать так, чтобы его объем стал меньше заданного минимального размера;
- MAXSIZE – указывает максимальный размер файла. Если размер не указан, то файл будет увеличиваться до полного заполнения диска;

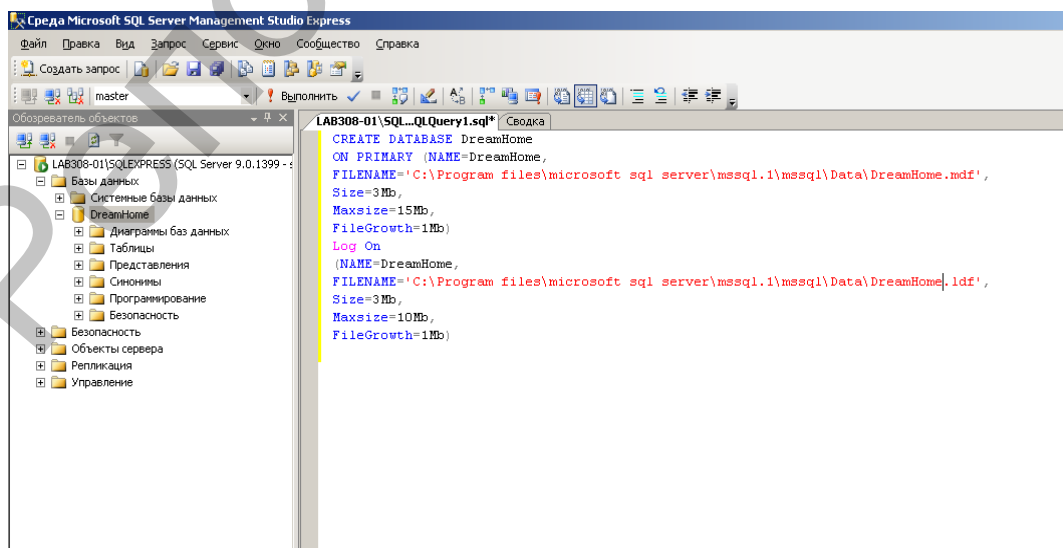


Рисунок 22

- FILEGROWTH – задаёт шаг приращения размера файла. Если SQL Server необходимо увеличить размер файла, он увеличит его на значение, заданное параметром FILEGROWTH, причём ноль означает запрет увеличения размера. По умолчанию (если параметр FILEGROWTH не определён) – шаг приращения равен 10%, а его минимальное значение – 64 Кб. Указанный вами размер округляется до ближайшего числа, кратного 64 Кб;
- FOR RESTORE – задает восстановление системы по журналу транзакций в случае её сбоя. Имеется в виду сбой системы, нарушающий все выполняемые в данный момент транзакции, но не нарушающий базу данных физически. При сбое носителей, который представляет собой физическую угрозу для данных, восстановление осуществляется с резервной копии БД.

Кроме перечисленных, при создании базы данных можно указать параметры настройки и параметры доступа к базе данных. Например, можно сделать базу данных доступной только для чтения или указать, чтобы регистрационные записи удалялись из журнала транзакций после записи изменённых страниц данных на диск. Можно также осуществить настройку доступа пользователей к базе данных. По умолчанию свойства доступа могут устанавливаться для пользователей, осуществляющих подключение, используя роль *public*. Наиболее часто используемые параметры смотрите в Приложении 2.

Упражнение: При помощи оператора CREATE DATABASE создайте базу данных DreamHome. Параметры базы данных указаны в таблице:

Файл	Логическое имя	Физическое имя	Начальный размер	Шаг приращения размера	Максимальный размер
База данных	DreamHome	C:\Program files\Microsoft Sql server\mssql.1\mssql \Data\DreamHome.mdf	3 Мб	1 Мб	15 Мб
Журнал	DreamHome_log	c:\program files\microsoft sql server\mssql.1\mssql \data\r.ldf	1 Мб	1 Мб	10 Мб

1. Для начала необходимо запустить среду разработки «MS SQL Server Management Studio». Для этого в меню «Пуск» выбрать пункт «Программы\Microsoft SQL Server 2008\SQL Server Management Studio».
2. Создайте новый запрос, нажав на панели инструментов кнопку «Создать запрос».
3. Выполните оператор CREATE DATABASE, чтобы создать базу данных (см. рисунок 22).

4. Просмотрите свойства базы данных с помощью группы процедур *sp_helpdb*, чтобы убедиться, что база создана должным образом:

EXEC sp_helpdb DreamHome

1. Второй способ:

В обозревателе объектов откройте контекстное меню папки Базы данных и в появившемся меню выберите пункт «Создать базу данных». Появится окно настроек параметров файла данных новой БД (см. рисунок 23)

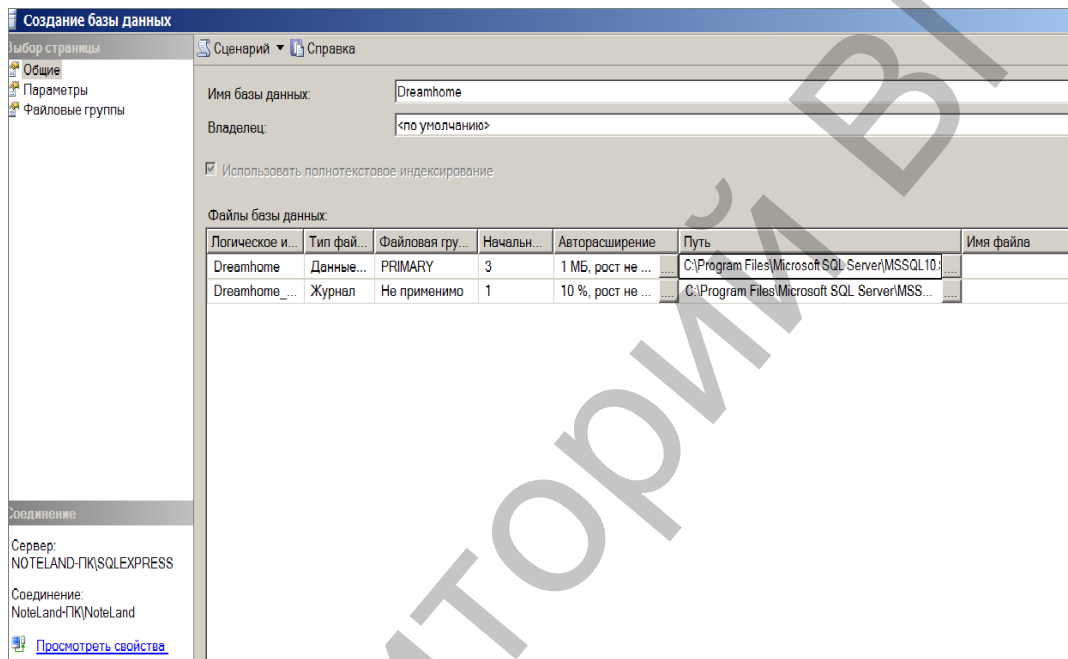


Рисунок 23

На панели **Выбор страницы** выбрать **Общие**.

В верхней части окна расположено два параметра: **Имя Базы данных** и **Владелец**). Задайте **Имя Базы данных** - **DreamHome**. Параметр **Владелец** " оставить без изменений.

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

- Логическое имя файла данных и журнала транзакций. По этим именам будет происходить обращение к вышеприведённым файлам в БД. Можно заметить, что файл данных имеет то же имя что и БД, а имя файла журнала транзакций составлено из имени БД и суффикса "_log".
- Тип файла. Этот параметр показывает, является ли файл файлом данных или журналом транзакций.
- Файловая группа показывает, к какой группе файлов относится файл. **Группы файлов** настраиваются в группе настроек **Файловые группы**.

- Начальный размер файла данных и журнала транзакций в мегабайтах.
- Авторасширение размера файла. Как только файл заполняется информацией, его размер автоматически увеличивается на величину, указанную в параметре **Авторасширение**. Увеличение можно задавать как в мегабайтах, так и в процентах. Здесь же можно задать максимальный размер файлов. Для изменения этого параметра надо нажать кнопку "...".
- Путь к папке, где хранятся файлы. Для изменения этого параметра также следует нажать кнопку "...", указать путь к папке, где будет храниться база данных.
- имя файла. По умолчанию имена файлов аналогичны логическим именам. **Замечание:** Для добавления новых файлов данных или журналов транзакций используется кнопка **Добавить**, а для удаления кнопка **Удалить**.

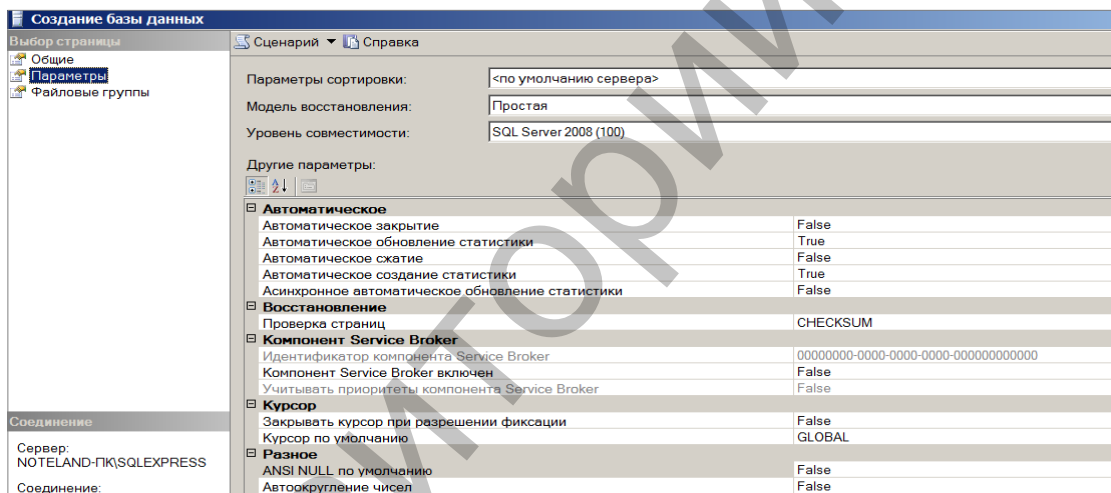


Рисунок 24

Перейдём к второстепенным настройкам файла данных. Для доступа к этим настройкам необходимо щёлкнуть мышью по пункту **Параметры**. В правой части окна находятся следующие настройки:

- **Сортировка** – этот параметр отвечает за обработку текстовых строк, их сравнение, текстовый поиск и т.д. Рекомендуется оставить его как **по умолчанию сервера**.
- **Модель восстановления**. Данный параметр отвечает за информацию, предназначенную для восстановления БД, хранящуюся в файле транзакций. Чем полнее модель восстановления, тем больше вероятность восстановления данных при сбое системы или ошибках пользователей, но и больше размер файла журнала транзакций.
- **Уровень совместимости**, определяет совместимость файла данных с более ранними версиями сервера. Если планируется перенос данных

на другую, более раннюю версию сервера, то ею необходимо указать в этом параметре.

Рассмотрим последнюю группу настроек **Файловые группы**.

Таблица имеет следующие столбцы:

- **Имя** группы файлов.
- **Файлы** - количество файлов входящих в группу.
- **Только для чтения**. Файлы нельзя изменять.
- **По умолчанию**. Все новые файлы данных будут входить в эту группу.

Эту группу настроек оставляем без изменений.

Для принятия всех настроек и создание файла данных и журнала транзакций нашей БД в окне **Новая база данных** нажать кнопку "Ok".

Произойдёт возврат в окно среды разработки "SQL Server Management Studio". На панели обозревателя объектов в папке **Базы данных** появится новая БД **DreamHome**.

Упражнение: просмотрите информацию о базе данных **DreamHome**.

Щёлкните правой кнопкой базу данных **DreamHome**.

Выберите в контекстном меню команду *Свойства* и просмотрите сведения о пространстве базы данных и журнала, параметры настройки (закладка *Параметры*) и параметры доступа к базе данных (закладка *Разрешения*).

Удаление базы данных

По умолчанию право на удаление базы данных принадлежит его владельцу и пользователям – членам постоянной роли *sysadmin*.

Удалить базу данных можно с помощью команды **Удалить** среды MS SQL Server Management Studio или оператора:

DROP DATABASE имя_базы_данных.

Упражнение: Удалите базу данных **DreamHome**.

Выполните оператор **DROP DATABASE**, чтобы удалить базу данных **DreamHome**:

DROP DATABASE DreamHome.

Отправьте операторы серверу с помощью команды **Go**;

4. Выполните системную процедуру *sp_helpdb*, чтобы создать список баз данных. Убедитесь, что вы удалили БД **DreamHome_db**:

EXEC sp_helpdb

Типы данных, используемые в SQL-сервере

После создания логической структуры базы данных и самой БД можно приступать к созданию в ней объектов: пользовательских типов данных и таблиц.

Одним из основных этапов в процессе создания таблиц является определение типов данных его полей. Тип данных поля таблицы определяет тип информации, которая будет размещаться в этом поле. SQL-сервер поддерживает большое число различных типов данных: текстовые, числовые, двоичные и т.д. (см. Приложение 3).

Пользовательский тип данных

В системе SQL-сервер имеется поддержка пользовательских типов данных. Они могут использоваться при определении или какого-либо специфического формата, или наоборот, часто употребляемого. Например, когда в нескольких таблицах базы данных имеется поле с одинаковыми свойствами (типом, размером или ограничениями на значение). Создание пользовательских типов данных в системе SQL-сервер может осуществляться следующими способами¹:

Первый способ: При помощи процедуры *sp_addtype*

Синтаксис

sp_addtype тип, системный_тип_данных [, 'NULL' | 'NOT NULL']

Упражнение: С помощью процедуры *sp_addtype* создайте пользовательские типы данных для **Почтового индекса (Postcode)** и **Номера отделения (объекта, владельца и т.д.)** в базе данных DreamHome. Типы данных указаны в таблице:

Имя пользовательского типа	Стандартный тип данных	Описание данных
Postcode	char	Строка из 6 символов, допускающая значения NULL
Member_no	smallint	Целое число, не превышающее 30000

1. Введите и выполните следующие операторы:

EXEC sp_addtype postcode, 'char (6)', NULL

EXEC sp_addtype member_no, 'smallint'

Удаление пользовательского типа данных осуществляется при помощи процедуры *sp_droptype*. Если на пользовательский тип ссылаются таблицы или объекты базы данных, его нельзя удалить.

Второй способ: С помощью обозревателя MS SQL Server Management Studio.

Для этого следует:

1. Выбрать в базе данных **DreamHome** группу **Программирование/Типы/Определяемые пользователем типы данных**. В результате выполнения этой операции на экран будет выведено диалоговое окно настройки нового типа данных.

¹ Перед созданием пользовательских типов данных, выполните п. 1 в разделе Порядок выполнения работы.

2. В поле *Имя* этого диалогового окна следует указать его имя; выпадающий список *Тип данных* определяет стандартный тип данных, относительно которого будет создан пользовательский. Размерность нового типа данных указывается в поле *Точность*, а проверку наличия NULL-значений можно определить, воспользовавшись флажком *Разрешить значения NULL*.

Теперь при указании типа данных поля таблицы в списке стандартных типов данных будет присутствовать вновь созданный пользовательский тип, при выборе которого будут установлены описанные выше параметры.

При создании пользовательских типов данных следует придерживаться следующих правил:

1. Если длина поля варьируется, использовать типы данных переменной длины. Например, список имен лучше хранить в поле с типом *varchar*, чем *char* (имеющим фиксированную длину). Типы данных переменной длины помогают экономить дисковое пространство.
2. Целые и денежные типы данных, а также типы даты и времени поддерживают разные диапазоны значений в зависимости от объема отведённой им памяти. Например, если для хранения уникальных идентификаторов объектов недвижимости используется тип *tinyint*, число объектов ограничивается 255.
3. Выбор числовых типов данных зависит от требуемой точности; обычно используется тип *decimal*.
4. Если объём информации превышает 8000 байт, следует использовать типы *text* или *image*. В противном случае – *binary*, *char* или *varchar*. Типы *char* или *varchar* более функциональны по сравнению с *text* и *image*.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Перечислите основные способы создания базы данных в SQL Server.
2. В каких случаях создаются secondary файлы? Какой тип они имеют?
3. Какая команда служит для удаления базы данных?
4. Кому принадлежит право на удаление базы данных?
5. Какие ограничения существуют на удаление базы данных?
6. Чем отличаются типы данных Char и Varchar?
7. В каких случаях используются пользовательские типы данных?
8. Как создать пользовательский тип данных в SQL Server?

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Создать базу данных DreamHome (написать запрос).

В таблице перечислены ее свойства:

Файл	Начальный размер, Мб	Шаг приращения размера, Мб	Максимальный размер файла, Мб
База данных	15	1	30
Журнал	2	1	7

2. Создайте с помощью команды `sp_addtype` пользовательский тип данных для **Номера телефона** (обычного, мобильного, пейджера, с междугородным кодом):

Имя пользовательского типа	Стандартный тип данных	Описание данных
Phonenumber	char	Строка из 17 символов

3. Создайте с помощью настроек MS SQL Server Management Studio пользовательский тип данных:

Имя пользовательского типа	Стандартный тип данных	Length	Precision	Scale
shortstring	varchar	20	0	0

4. Создайте базу данных в соответствии с вариантом индивидуального задания.

Учебное издание

АДАМЕНКО Наталья Дмитриевна

МОДЕЛИ ДАННЫХ И СУБД

Методические рекомендации

В 3 частях

Часть 1

Технический редактор

Г.В. Разбоева

Компьютерный дизайн

Т.Е. Сафранкова

Подписано в печать 10.12.2015. Формат 60x84^{1/16}. Бумага офсетная.

Усл. печ. л. 2,09. Уч.-изд. л. 1,88. Тираж 80 экз. Заказ 181.

Издатель и полиграфическое исполнение – учреждение образования

«Витебский государственный университет имени П.М. Машерова».

Свидетельство о государственной регистрации в качестве издателя,

изготовителя, распространителя печатных изданий

№ 1/255 от 31.03.2014 г.

Отпечатано на ризографе учреждения образования

«Витебский государственный университет имени П.М. Машерова».

210038, г. Витебск, Московский проспект, 33.