

# Теория Баз Данных



# Урок № 1

## Введение в теорию баз данных

## Содержание

<b>1. Введение в теорию баз данных.....</b>	<b>4</b>
История и этапы развития .....	4
Понятия база данных и система управления базами данных .....	5
Сравнение существующих моделей баз данных ...	6
Понятие реляционной модели баз данных .....	10
Двенадцать правил Кодда .....	12
Сравнительный анализ СУБД Microsoft SQL Server с существующими системами управления базами данных .....	14
<b>2. Основы взаимодействия с Microsoft SQL Server ..</b>	<b>16</b>
Версии и редакции Microsoft SQL Server .....	16

Инсталляция Microsoft SQL Server . . . . .	16
Инструменты управления и утилиты MS SQL Server. . . . .	30
Управление базой данных . . . . .	35
<b>3. Домашнее задание . . . . .</b>	<b>47</b>

# 1. Введение в теорию баз данных

---

Базы данных используются для хранения и обработки различной информации уже не один десяток лет и будут актуальны еще длительное время. Наверное, вам известна знаменитая фраза Натана Ротшильда: «Кто владеет информацией, тот владеет миром», а где же сейчас хранится эта информация? Правильно, в базах данных, поэтому специалисты в этой области будут востребованы всегда.

## История и этапы развития

Исследования по созданию баз данных стали возможны благодаря появлению программируемого оборудования обработки записей и начались в конце 50-х годов прошлого столетия. Основной идеей этих исследований была автоматизация офисной работы, связанная с хранением и учетом данных, которая выполнялась вручную и требовавшая больших затрат труда. Благодаря удешевлению вычислительных мощностей для решения этих задач стало возможным использование компьютеров.

Лидером в сфере исследований баз данных была компания IBM, которая в 1968 году выпустила свою первую систему управления базами данных IMS (IBM *Information Management System*), основанную на иерархической модели данных. В середине 70-х годов прошлого столетия компания IBM создала первую реляционную

систему управления базами данных (РСУБД) — IBM System R, в которой впервые был реализован язык запросов для реляционных баз данных — *Structured Query Language (SQL)*.

В 1979 году компания Oracle выпустила первую коммерческую РСУБД Oracle v2, вслед за этим различные компании выпустили свои коммерческие продукты. В 1982 году появилась РСУБД DB2 от IBM, в 1988 году компании Microsoft и Ashton-Tate выпустили совместный продукт Ashton-Tate/Microsoft SQL Server 1.0, в 1992 году появилась первая версия РСУБД Access от компании Microsoft.

## **Понятия база данных и система управления базами данных**

Вы, наверное, встречали в повседневной жизни сочетание слов «база данных», но это довольно широкое понятие, смысл которого сводится к организации хранения данных. Однако различаются способы хранения, например это может быть набор файлов.

В нашем случае, **база данных** (БД) — это коллекции связанных данных, сгруппированных в единый объект. Допустим, при создании базы данных какого-то учебного заведения, мы размещаем в ней необходимые наборы данных (информация о студентах, преподавателях, факультетах и т.д.), и впоследствии управляем полученной БД при помощи программного обеспечения как единым целым.

Специализированное программное обеспечение, позволяющее работать с базами данных (обновлять,

извлекать данные и т.д.), имеет общее название система управления базами данных (СУБД), например Microsoft SQL Server, Access, Oracle Database и т.д.

СУБД состоит из ряда серверных и клиентских средств, которые позволяют выполнять администрирование баз данных и различные действия, связанные с манипуляцией данными. Практически любая СУБД позволяет обрабатывать запросы на извлечение и изменение данных, предоставляет механизмы для резервного копирования и восстановления данных, оптимизирует производительность выполнения запросов, управляет памятью.

## **Сравнение существующих моделей баз данных**

Модели баз данных различаются правилами взаимосвязи основных типов структур данных и операциями над ними. Каждая СУБД использует логическую структуру хранения данных, которая зависит от конкретной модели хранения данных, рассмотрим далее наиболее часто используемые модели баз данных.

### **Файловая модель**

Файловая модель данных характеризуется определенным набором файлов, не связанных между собой, основными типами структур данных которой являются поле, запись, файл. Поле — это элементарная, неделимая единица данных. Запись — это совокупность логически связанных полей. Файл — это множество записей, одинаковых по структуре.

Файловая модель была первой моделью хранения данных, ее можно считать моделью без СУБД, ведь внутренняя структура файлов была известна только разработчику данного программного обеспечения, то есть была уникальна.

Файловая модель обладает рядом недостатков, перечислим основные из них:

- алгоритм управления базой данных полностью заложен в программном обеспечении, при изменении структуры данных необходимо вносить правки в программное обеспечение;
- сложности совместимости форматов файлов, созданных при помощи различных языков программирования, из-за отличающихся друг от друга структур;
- трудности при переносе данных из одной БД в другую, по причине несовпадения структур данных;
- отсутствие централизованного хранения приводит к необходимости дублирования одних и тех же данных.

### **Иерархическая модель**

Иерархическая модель представляет собой простую структуру, в которой отдельные записи организуются в отношения типа «родитель-потомок» и образовывают обращенное дерево. Данные в этой иерархической структуре делятся на логические категории и подкатегории, использующие записи для представления логических единиц данных.

Иерархическая модель довольно удобна для представления отношений между сущностями реального

мира, однако совершенно не подходит для многих современных приложений. Иерархическая структура не поддерживает сложные отношения между записями типа «многие ко многим», потому что дочерняя запись может ассоциироваться только с одной родительской записью. Еще одним недостатком является довольно громоздкий процесс навигации, ведь для того чтобы получить доступ к записям необходимо перемещаться вверх и вниз по уровням иерархии.

Наиболее известная СУБД, созданная на основе этой модели — это IMS от компании IBM. Иерархическая модель нашла свое применение не только в СУБД, такая модель применяется для систем управления файлами в операционных системах.

### **Сетевая модель**

Говоря о сетевой модели баз данных нельзя не упомянуть о Чарльзе Бахмане (*Charles Bachman*), который являлся один из основоположников сетевой модели баз данных. Именно он в 1963 году разработал одну из первых сетевых систем управления базами данных IDS (*Integrated Data Store*). За свои разработки, посвященные технологиям баз данных, в 1973 году он был удостоен премии Тьюринга.

Сетевая модель появилась как результат усовершенствования иерархической модели, и в отличие от последней позволяет записям принимать участие во множестве отношений типа «родитель-потомок».

Сетевая модель была стандартизована в 1975 году организацией CODASYL (*Conference of Data System Languages*)

ges), которая определила базовые понятия этой модели и язык описания.

Хотя сетевая модель является более гибкой по сравнению с иерархической, программирование навигации по записям должны осуществлять разработчики. Реализация этой модели подразумевает использование значительных ресурсов памяти, потому что каждый элемент должен хранить ссылки на другие элементы. Кроме того, внесение любых изменений может привести к сложным операциям обновления базы данных.

## **Реляционная модель**

Еще одним выдающимся ученым был Эдгар Кодд (*Edgar Codd*), будучи сотрудником кампании IBM, разработал реляционную модель данных, которую в 1970 году описал в статье «A Relational Model of Data for Large Shared Data Banks». В 1981 году за свои исследования в области баз данных он был награжден премией Тьюринга.

Реляционная модель данных не использует связь между родительскими и дочерними записями, а основана на взаимодействии строк и столбцов, которые образуют таблицы с данными, связанными между собой. Внутри базы данных, таблицы имеют уникальные названия и связываются между собой отношениями, которые позволяют осуществлять навигацию по записям. Такая структура данных является очень гибкой и удобной при извлечении и изменении информации.

В качестве примера РСУБД можно назвать: Microsoft SQL Server, Access, MySQL.

## Объектно-ориентированная модель

Объектно-ориентированная модель данных имеет древовидную структуру, узлами которой являются объекты. В этой модели технология объектно-ориентированного программирования (ООП) применяется к технологии баз данных. Каждая запись в базе данных является объектом, связи между записями осуществляются с помощью механизмов, которые используются в ООП. Поиск необходимой записи заключается в сравнении пользовательского объекта с объектами, которые хранятся в БД.

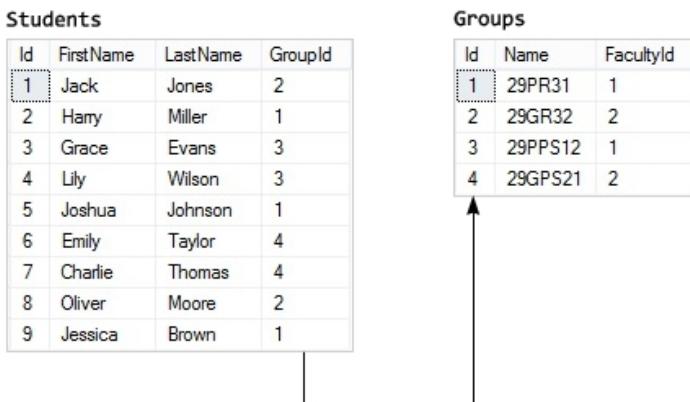
Базовые понятия объектно-ориентированной модели полностью повторяют основные понятия ООП: класс, объект, метод, инкапсуляция, наследование, полиморфизм, агрегация.

Основные недостатки модели объектно-ориентированной модели: сложность понимания ее структуры и относительно низкая скорость выполнения запросов. К достоинствам этой модели можно отнести возможность хранения и отображения информации о сложных объектах, с возможностью определения методов для работы с ними. Благодаря этим достоинствам некоторые РСУБД дополняются элементами объектно-ориентированного проектирования, например Oracle Database, которая является объектно-реляционной СУБД.

## Понятие реляционной модели баз данных

Начиная с конца 70-х годов прошлого столетия наиболее популярной моделью баз данных, является реляционная модель, в основе которой лежат математические принципы обработки данных.

В реляционной модели данные представлены в виде таблиц, которые состоят из полей. Например, на рисунке 1.1 показаны две таблицы, в первой из них хранится информация о студентах, а во второй о группах, в которых проходят обучение студенты.



**Рисунок 1.1.** Пример структуры реляционной базы данных

Каждое поле предназначено для хранения определенной информации, которая характеризует конкретную сущность. В нашем случае каждая запись в таблице **Students** представляет конкретного студента, в которой указывается уникальный идентификатор студента, имя, фамилия и идентификатор той группы, где он обучается. В таблице **Groups** записи содержат уникальную информацию по каждой группе, и состоят из идентификатора группы, имени группы и идентификатора факультета.

В литературе, описывающей реляционную модель данных, вы можете встретить несколько другие определения: так таблицы называются отношениями (англ. *relation*), в общем, отсюда и пошло название самой модели — реляционная,

запись в таблице называются кортежами, а поля — атрибутами. Однако какие бы вы названия не использовали, суть реляционной модели останется прежней.

Как показано на рисунке 1.1 связь между таблицами обеспечивается благодаря наличию в одной таблице уникального идентификатора из другой таблицы, мы сейчас не будем останавливаться на этом подробно, более полную информацию вы получите в последующих уроках.

## Двенадцать правил Кодда

В процессе разработки реляционной модели баз данных, Эдгар Кодд сформулировал требования, которым должна соответствовать любая реляционная СУБД, он опубликовал их в 1985 году. На самом деле основное правило имеет номер 0, поэтому всего насчитывается 13 правил. Перечислим эти правила:

1. Основное правило: СУБД должна управлять базами данных, используя исключительно свои реляционные возможности;
2. Правило информации: вся информация в любой базе данных должна быть представлена исключительно значениями в таблицах;
3. Правило гарантированного доступа: каждое значение в любой таблице можно получить с помощью комбинации имени таблицы, значения первичного ключа и имени столбца (имя таблицы позволяет найти требуемую таблицу, имя столбца позволяет найти требуемый столбец, а первичный ключ позволяет найти строку, содержащую искомый элемент данных);

4. Правило поддержки недействительных значений: в СУБД должна быть реализована возможность поддержки неизвестных или отсутствующих значений (отсутствующие данные должны быть представлены с помощью недействительных значений (`NULL`));
5. Правило динамического каталога: информация о базах данных должна храниться в виде таблиц, и СУБД должна обеспечивать стандартный доступ к ней и пользовательским данным при помощи одних и тех же средств (СУБД должна содержать набор системных таблиц, описывающих структуру баз данных);
6. Правило исчерпывающего подъязыка данных: СУБД должна поддерживать хотя бы один язык, операторы которого обеспечивают все ее основные функции (создание базы данных, манипулирование данными, управление доступом и т.д.);
7. Правило обновления представлений: каждое представление должно поддерживать те же операции с данными, что и таблицы (чтение, вставка, изменение и удаление данных);
8. Правило добавления, обновления и удаления: операции связанные с изменением и удалением данных должны одинаково работать как с одной записью в таблице, так и с множеством записей;
9. Правило независимости физических данных: приложения, использующие любую базу данных, не должны зависеть от способа хранения информации и аппаратного обеспечения компьютеров;

10. Правило независимости логических данных: приложения не должны зависеть от структуры базы данных, любые изменения в структуре БД не должны влиять на работу приложения;
11. Правило независимости условий целостности: язык СУБД должен поддерживать проверку входных данных и обеспечивать их целостность;
12. Правило независимости распространения: база данных может находиться на разных компьютерах и язык СУБД должен поддерживать возможность работы с распределенными данными;
13. Правило единственности: не должно быть возможности нарушить безопасность и целостность данных в обход языка СУБД, то есть при работе с данными должен использоваться только язык СУБД.

## **Сравнительный анализ СУБД Microsoft SQL Server с существующими системами управления базами данных**

На сегодняшний день существует большое количество систем управления базами данных, некоторые из них уже упоминались нами в текущем уроке. В качестве сравниваемых СУБД мы возьмем MS SQL Server, Oracle, PostgreSQL и MySQL, у каждой из них есть как свои преимущества, так и недостатки. Сейчас вы, наверное, скажете: «Всяк кулик своё болото хвалит», однако мы все же попытаемся быть максимально объективными.

Если рассматривать Oracle и MS SQL Server, то по производственным характеристикам они приблизительно

одинаковы, может быть по ряду показателей Oracle лучше. Главное же отличие Oracle от MS SQL Server — это поддержка большого количества программно-аппаратных платформ (Linux, Windows, Mac OS и т.д.). В качестве отрицательных сторон Oracle можно назвать: высокую стоимость, высокие требования к аппаратному обеспечению и сложность администрирования. К тому же Oracle в основном предназначен для использования в больших промышленных проектах, и в относительно простых приложениях его использование будет не рационально.

**MySQL** — это РСУБД, предназначенная для использования в простых и средних приложениях и основное ее преимущество в том, что она бесплатная. К недостаткам можно отнести: ограниченную функциональность и некоторые проблемы с надежностью.

**PostgreSQL** — это свободно распространяемая объектно-реляционная СУБД, максимально соответствующая стандартам SQL, поддерживаемая ОС Windows и множеством UNIX-подобных платформ. И хотя PostgreSQL является довольно качественным программным продуктом, у нее тоже есть недостатки: производительность, сложность настройки и небольшая популярность (малое количество хостингов с поддержкой этой СУБД).

Из всего вышеперечисленного можно сделать вывод, что MS SQL Server является некой золотой серединой среди СУБД, хотя, конечно же, у нее есть и свои недостатки.

При изучении текущего предмета мы будем использовать СУБД Microsoft SQL Server, поэтому далее остановимся на ней более подробно.

## 2. Основы взаимодействия с Microsoft SQL Server

---

### Версии и редакции Microsoft SQL Server

Как мы уже говорили ранее, первая версия MS SQL Server (SQL Server 1.0) появилась в 1988 году. Следующая версия — SQL Server 1.1 — вышла в 1990 году. За прошедшие двадцать с лишним лет вышло 15 версий MS SQL Server. В 2016 году вышла последняя, на текущий момент времени, версия — SQL Server 2016, именно эту версию СУБД MS SQL Server мы и будем использовать при изучении текущего предмета, но прежде чем использовать тот или иной программный продукт его необходимо установить на компьютер.

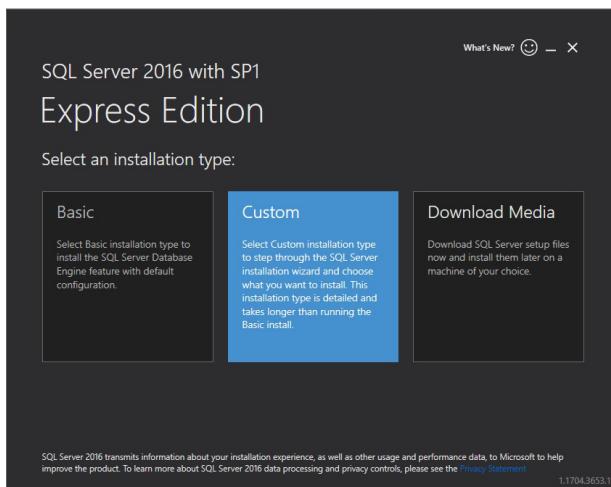
### Инсталляция Microsoft SQL Server

СУБД Microsoft SQL Server 2016 доступна в нескольких редакциях: Enterprise, Standard, Developer и Express. Для изучения материалов курса вам достаточно установить версию MS SQL Server 2016 Express, к тому же она бесплатная.

Прежде всего, вам необходимо скачать установщик программного продукта, для этого нужно перейти на сайт компании Microsoft по данному адресу: <https://www.microsoft.com/en-us/download/details.aspx?id=54284>. На этой же странице необходимо ознакомиться с системными требованиями, которым должен соответствовать ваш компьютер, для того чтобы установка MS SQL Server была успешной. Перечислим эти требования:

- поддерживаемые операционные системы: Windows 10, Windows 8, Windows 8.1, Windows Server 2012, Windows Server 2012 R2, Windows Server 2016;
- минимальная скорость процессора: 1 ГГц;
- минимальный объем оперативной памяти: 512 МБ;
- свободное место на жестком диске: 4,2 ГБ.

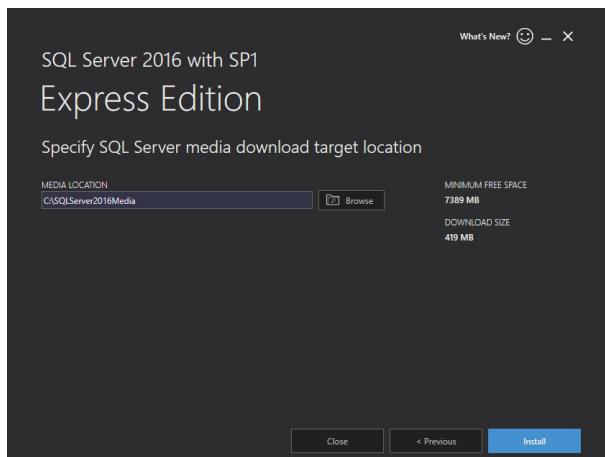
После того как вы скачаете и запустите установщик, появится окно, позволяющее выбрать тип установки. Если выбрать базовый тип установки, то ядро MS SQL Server будет сконфигурировано по умолчанию. Мы же выберем пользовательский тип установки — это расширенный тип установки, требующий больше времени (Рисунок 2.1).



**Рисунок 2.1.** Выбор пользовательского типа установки

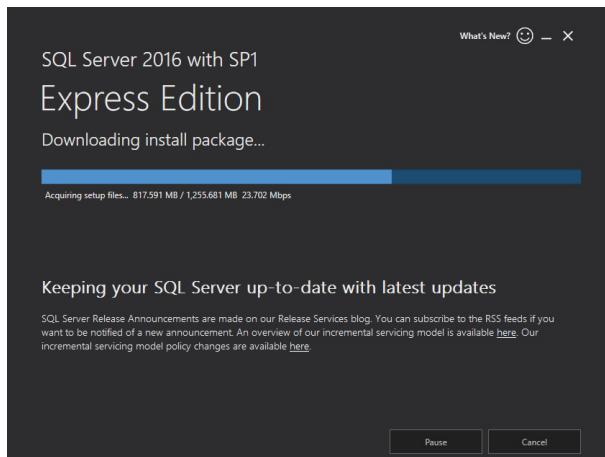
В следующем окне отображается необходимое свободное пространство на диске, размер скачивания и папка на компьютере, куда будет производиться установка, в случае необходимости ее можно изменить. Оставим

расположение установки по умолчанию и запустим установку, нажав кнопку **Install** (Рисунок 2.2).



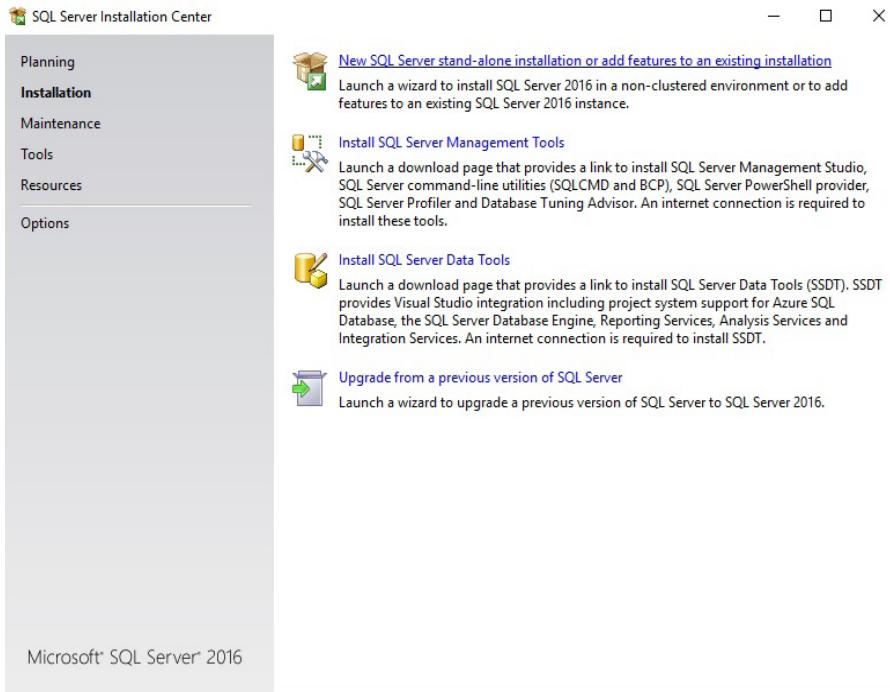
**Рисунок 2.2.** Начало установки

После этого начинается скачивание пакета установки, процесс скачивания отображается при помощи индикатора (Рисунок 2.3).



**Рисунок 2.3.** Скачивание пакета установки

После того как будет скачан пакет установки, появиться окно установки MS SQL Server 2016 с выбранным пунктом **Installation**. В списке возможных установок необходимо выбрать первый пункт — **New SQL Server stand-alone installation...** (Рисунок 2.4).



**Рисунок 2.4.** Начало новой установки экземпляра  
MS SQL Server

В следующем окне предлагается ознакомиться с лицензионным соглашением от корпорации Microsoft по использованию Microsoft SQL Server 2016 Express. После внимательного ознакомления, его следует принять и нажать кнопку **Next** (Рисунок 2.5).

## Урок № 1

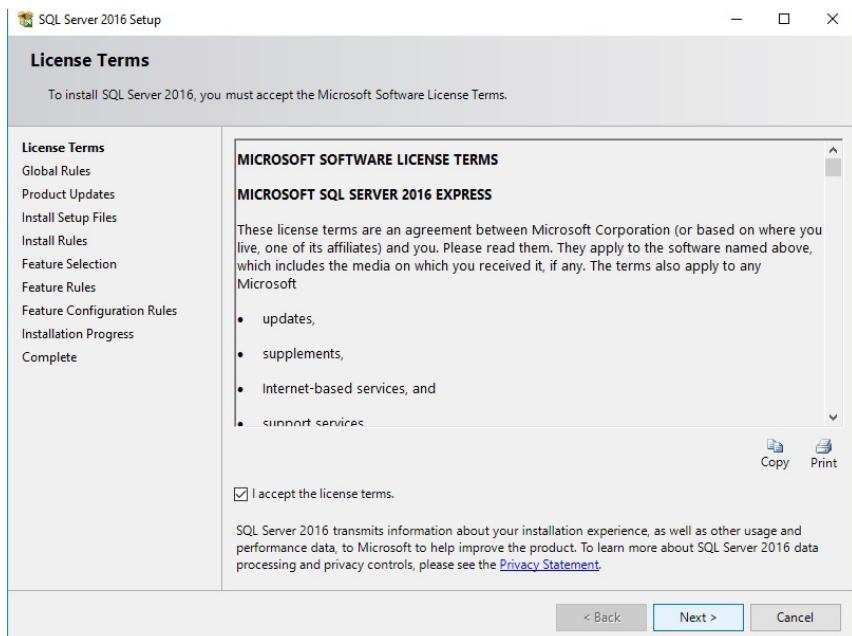


Рисунок 2.5. Лицензионное соглашение от корпорации Microsoft

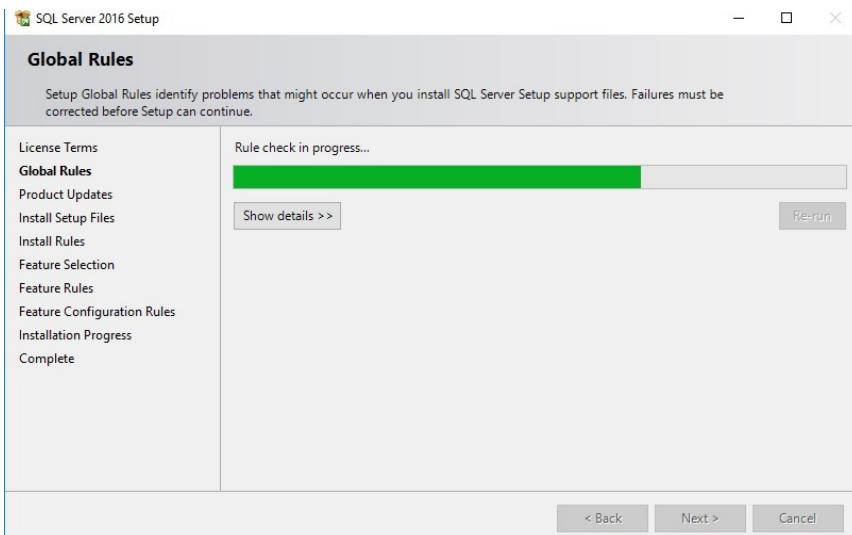
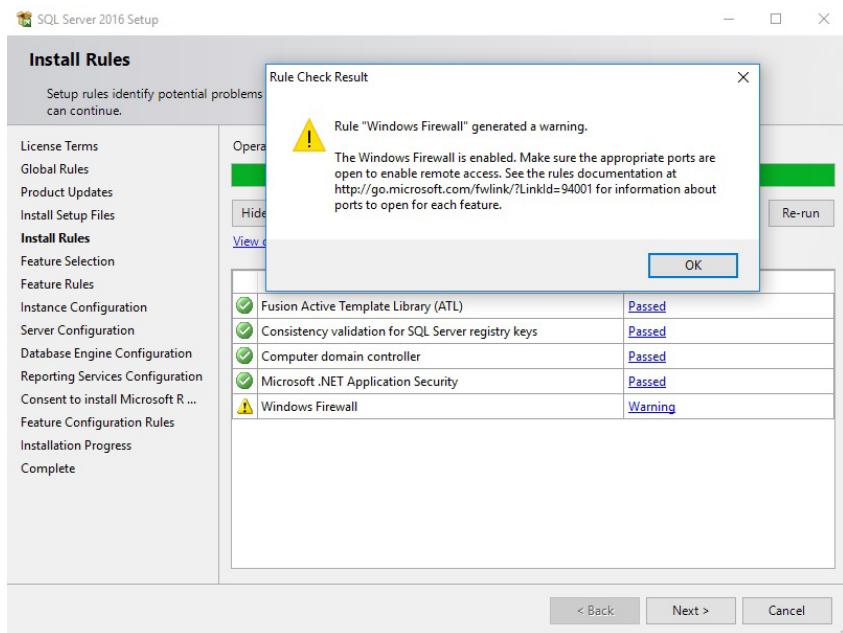


Рисунок 2.6. Установка файлов

После этого осуществляется поиск обновлений, загрузка и установка файлов установки (Рисунок 2.6).

Затем программа установки определяет потенциальные проблемы, которые могут возникнуть при выполнении установки MS SQL Server. В нашем случае появилось предупреждение, в котором говорится о необходимости открытия соответствующих портов для обеспечения удаленного доступа. Мы учтем это предупреждение при настройке удаленного доступа, а пока что нажмем кнопку Next (Рисунок 2.7).



**Рисунок 2.7. Правила установки**

В следующем окне установки необходимо выбрать компоненты для установки и перейти на следующее окно (Рисунок 2.8).

## Урок № 1

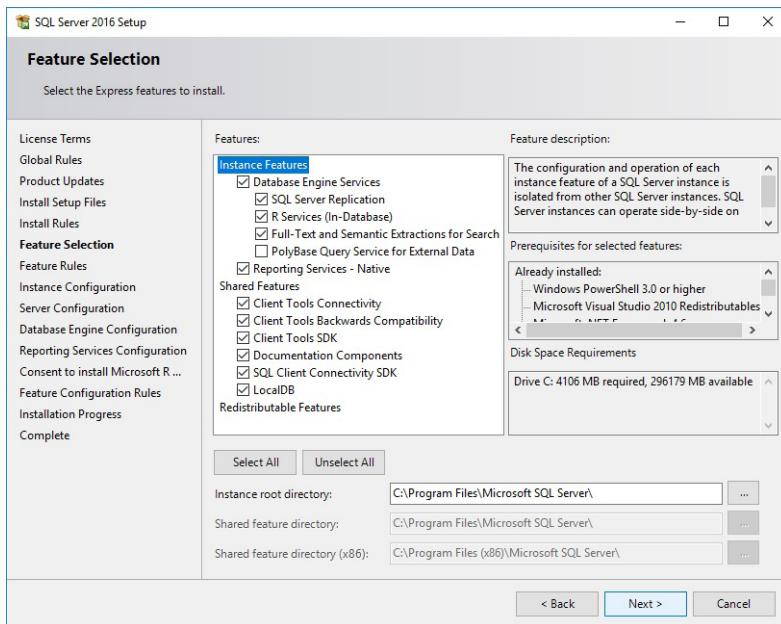


Рисунок 2.8. Выбор компонентов

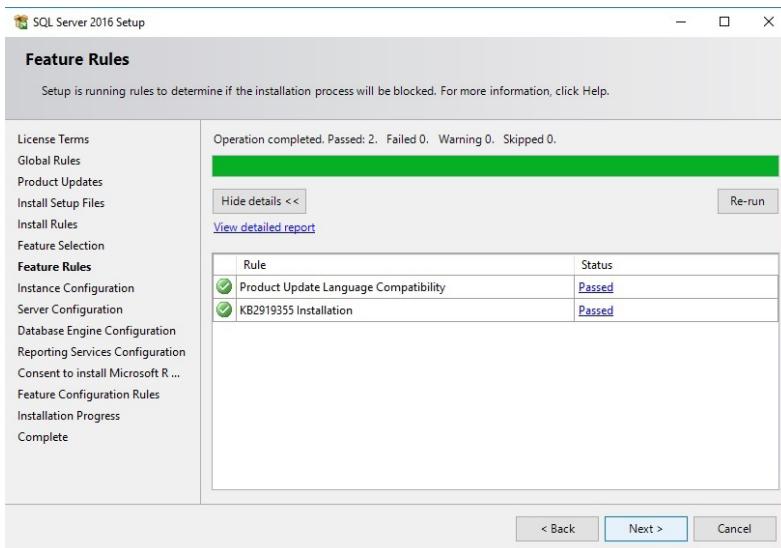
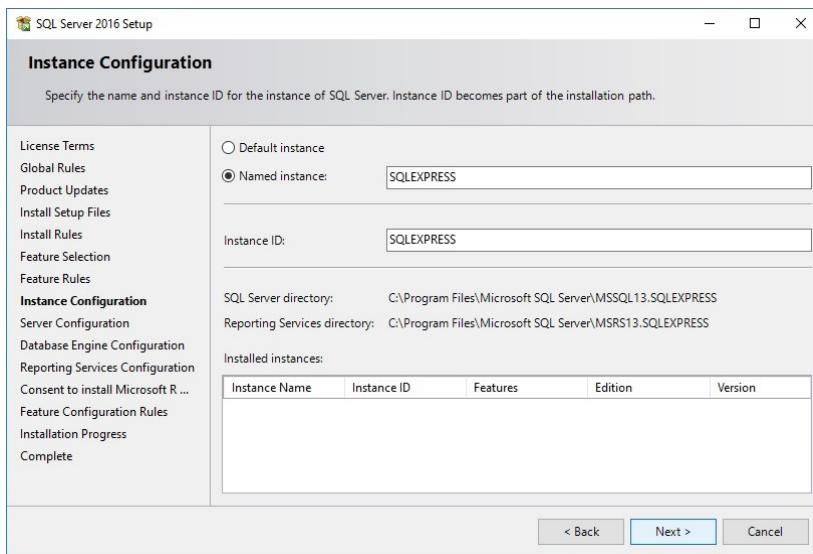


Рисунок 2.9. Установка компонентов

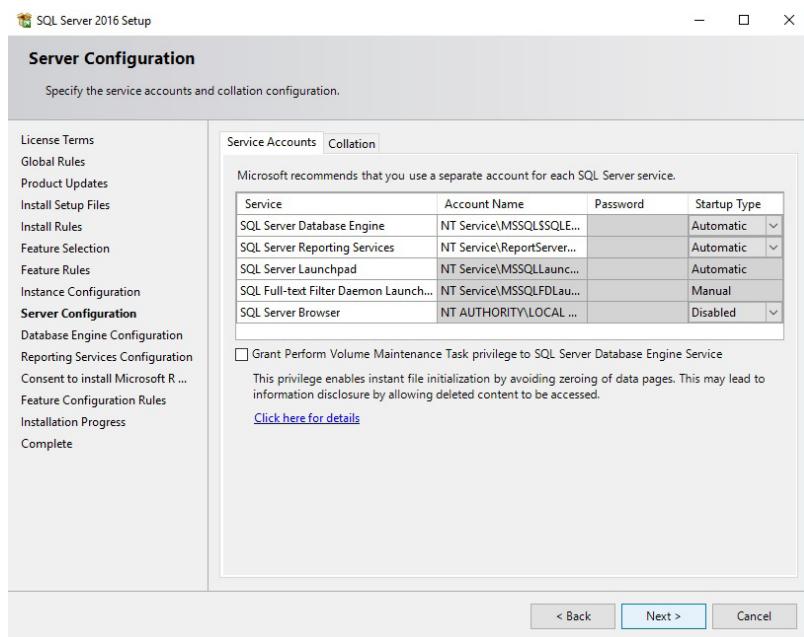
В следующем окне демонстрируется процесс и результат установки выбранных ранее компонентов, убедившись, что установка прошла успешно нужно нажать кнопку **Next** (Рисунок 2.9).

Далее вы увидите окно, в котором необходимо указать имя и идентификатор для экземпляра SQL Server, выберем пункт **Named instance** (*Именованный экземпляр*) и перейдем далее (Рисунок 2.10).



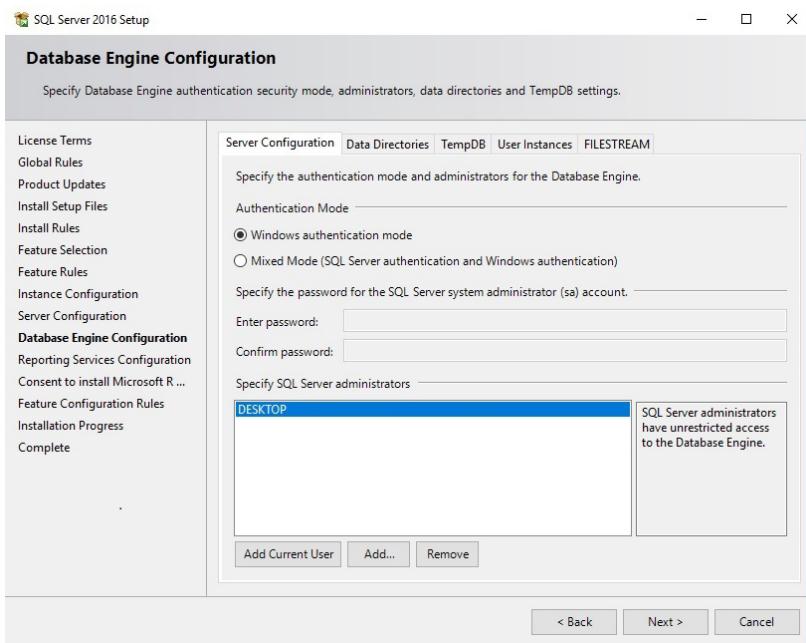
**Рисунок 2.10.** Настройка экземпляра

Следующее окно предназначено для настройки учетных записей и параметров сортировки, оставим эти пункты без изменений и нажмем кнопку **Next** (Рисунок 2.11).



**Рисунок 2.11. Конфигурация сервера**

Следующее окно позволяет настроить компонент **Database Engine**. На вкладке **Server Configuration** нужно выбрать один из двух режимов проверки подлинности, либо **Windows authentication mode** (*режим проверки подлинности Windows*), либо **Mixed Mode** (*смешанный режим*), при выборе которого необходимо указать пароль для встроенной учетной записи администратора SQL Server (sa), оставим здесь значение по умолчанию. Также необходимо указать администраторов SQL Server, по умолчанию в списке будет находиться имя текущего пользователя. При помощи соответствующих кнопок вы можете, как добавить пользователя или текущего пользователя, так и удалить пользователя из списка (Рисунок 2.12).



**Рисунок 2.12.** Вкладка «Конфигурация сервера»

На вкладке **Data Directories** зададим корневой каталог данных. Можно оставить значение по умолчанию, однако рекомендуется указывать каталог на другом физическом диске, в крайнем случае, на другом логическом диске, что повышает вероятность сохранения данных в случае отказа системы. В нашем случае, за неимением еще одного физического диска, укажем заранее созданный каталог SQLDB на логическом диске D, для чего нажмем кнопку с тремя точками в пункте **Data root directory** (Рисунок 2.13).

После нажатия кнопки **OK** в окне **Browse For Folder** (Рисунок 2.13), пути ко всем каталогам данных обновятся автоматически. Можно изменить расположение каталогов пользовательской базы данных и каталога резервного

## Урок № 1

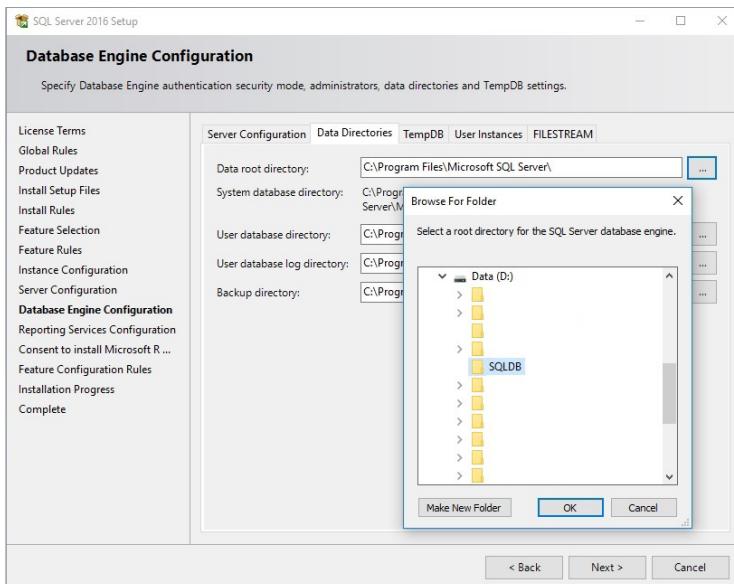


Рисунок 2.13. Установка корневого каталога данных

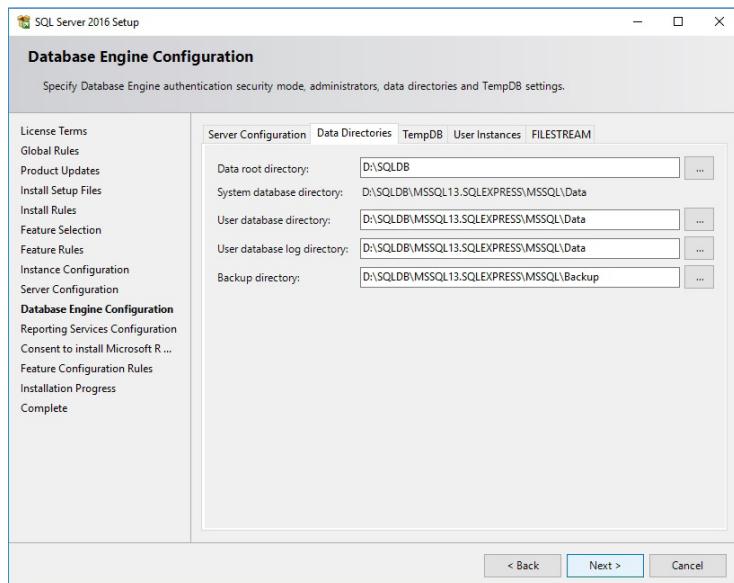
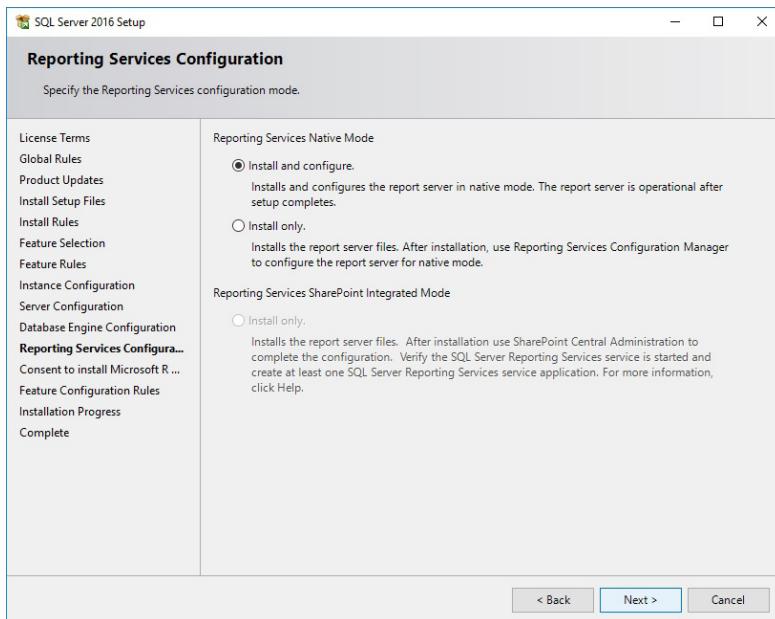


Рисунок 2.14. Вкладка «Каталоги данных»

копирования, проделав аналогичные операции, но мы остановимся на полученном результате (Рисунок 2.14).

Вкладка **TempDB** предназначена для настройки базы данных, хранящей временные данные, на следующей вкладке **User Interfaces** существует возможность установить разрешение обычным пользователям запускать экземпляр компонента **Database Engine** (по умолчанию установлено) и последняя вкладка позволяет включить **FILESTREAM** для интеграции Database Engine с файловой системой NTFS (по умолчанию отключено). Оставим эти три вкладки без изменений и перейдем далее.

Следующее окно позволяет задать настройку служб **Reporting Services**, выберем пункт **Install and configure** и нажмем кнопку **Next** (Рисунок 2.15).



**Рисунок 2.15.** Настройка служб Reporting Services

## Урок № 1

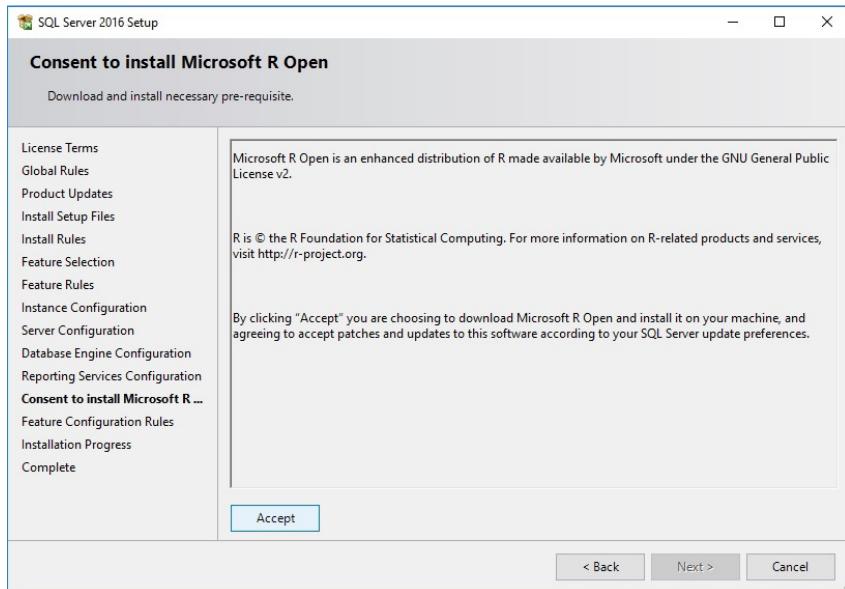


Рисунок 2.16. Установка Microsoft R Open

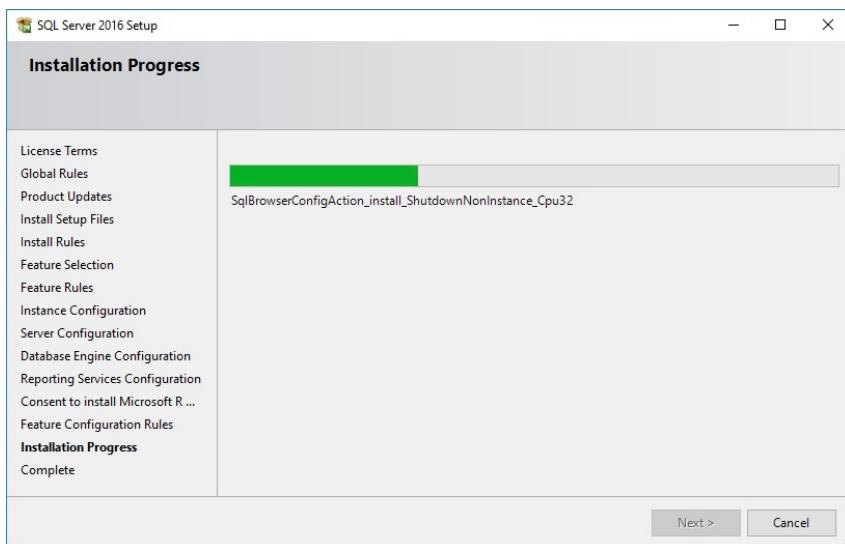
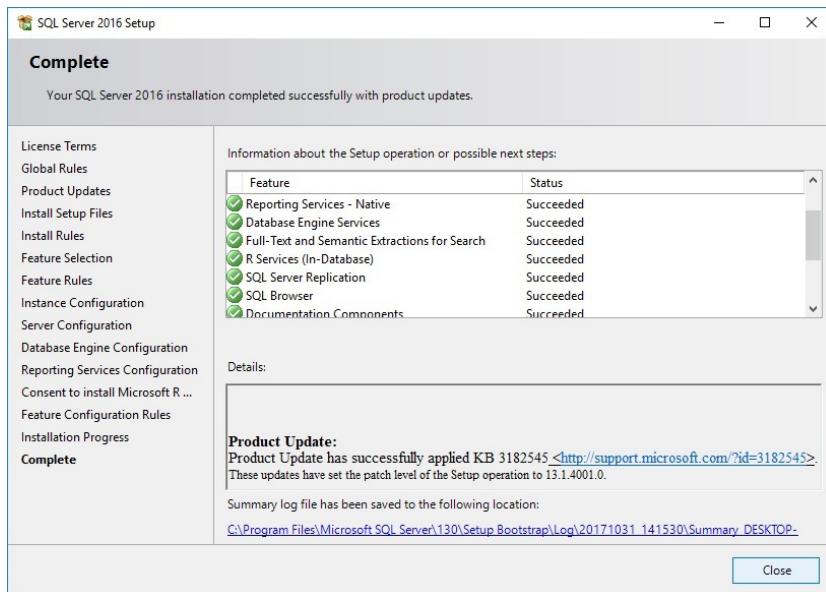


Рисунок 2.17. Ход выполнения установки

Для продолжения установки SQL Server, в следующем окне необходимо нажать кнопку **Accept**, чтобы установить на ваш компьютер расширенное распределение R от корпорации Microsoft — Microsoft R Open, которое поддерживает интеллектуальное моделирование и работу со статистическими данными (Рисунок 2.16).

Нажав кнопку **Next**, после успешной установки Microsoft R Open начнется процесс установки SQL Server на ваш компьютер, ход выполнения которой вы увидите в следующем окне (Рисунок 2.17).

По прошествии определенного времени, процесс установки завершится, и вы увидите окно завершения установки SQL Server, которое содержит сведения об установленных компонентах (Рисунок 2.18).



**Рисунок 2.18.** Завершение установки

По завершению процесса установки, откроем список установленных программ и убедимся, что SQL Server 2016 установлен на ваш компьютер (Рисунок 2.19).

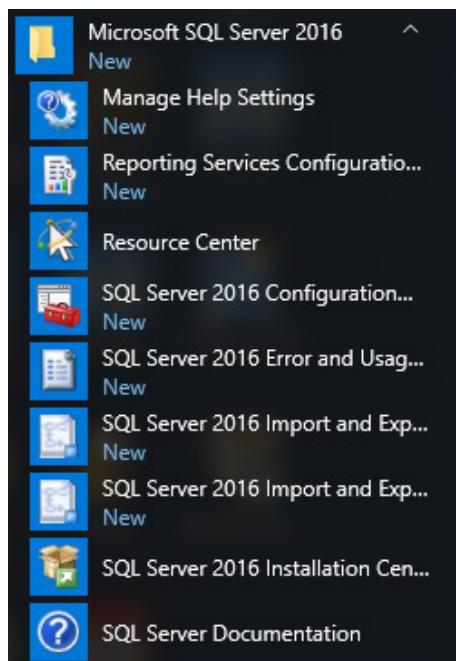


Рисунок 2.19. Программа Microsoft SQL Server 2016

## Инструменты управления и утилиты MS SQL Server

В списке установленных компонентов Microsoft SQL Server 2016 вы заметили ряд утилит, кратко остановимся на каждой из них.

Утилита **Reporting Services Configuration Manager** используется для настройки сервера отчетов: учетной записи, базы данных, электронной почты и т.д.

Компонент **SQL Server 2016 Configuration Manager** предназначен для управления службами, связанными с SQL Server (запуск, приостановка, возобновление и остановка служб). При помощи этой утилиты осуществляется управление конфигурацией сетевых подключений с клиентских компьютеров и настройки сетевых протоколов, используемых SQL Server.

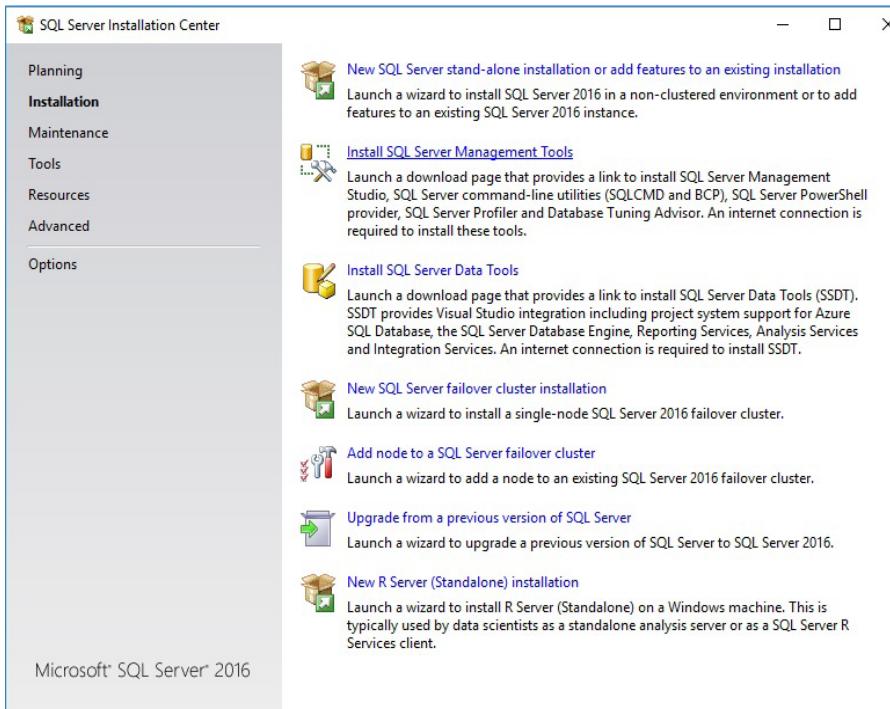
Для того чтобы улучшить функциональность SQL Server, корпорация Microsoft использует отчеты об ошибках, за сбор и отправку этой информации отвечает утилита **SQL Server 2016 Error and Usage Reporting**.

Утилита **SQL Server 2016 Import and Export Data** обеспечивает удобный способ копирования данных между источниками данных.

В списке установленных компонентов SQL Server 2016 отсутствует один очень важный инструмент управления — **Microsoft SQL Server Management Studio**, именно этот компонент обеспечивает наличие визуальной среды для работы с SQL Server. Следующее наши действия будут направлены на установку этого компонента, но перед этим необходимо перезагрузить ваш компьютер.

Для установки Management Studio в списке компонентов SQL Server 2016 (Рисунок 2.19) необходимо выбрать утилиту **SQL Server 2016 Installation Center**, которая обеспечивает установку необходимых компонентов в SQL Server, нажав соответствующий пункт. После этого откроется одноименное окно (Рисунок 2.20).

В открывшемся окне необходимо выбрать пункт **Install SQL Server Management Tools**, после чего в окне

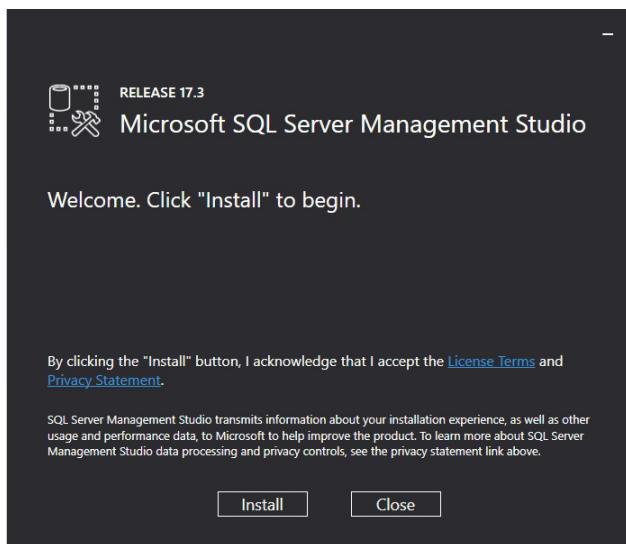


**Рисунок 2.20. Центр установки SQL Server**

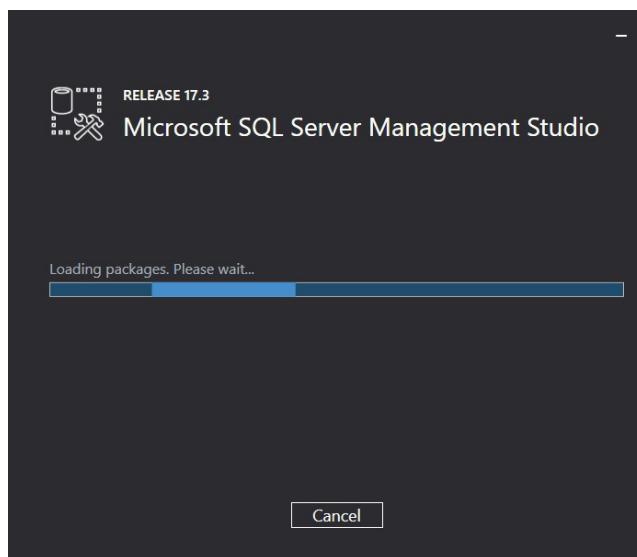
браузера откроется страница по следующему пути <https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms>, на этой странице необходимо выбрать пункт **Download SQL Server Management Studio 17.3**, после чего начнется загрузка установщика Management Studio на ваш компьютер.

После запуска установщика, появится начальное окно установки, чтобы начать установку нужно нажать кнопку **Install** (Рисунок 2.21).

Первоначально появится окно загрузки пакетов установки (Рисунок 2.22).

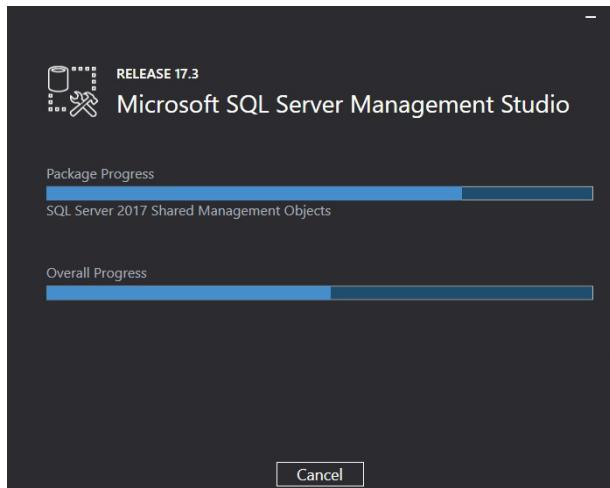


**Рисунок 2.21.** Начало установки SQL Server Management Studio

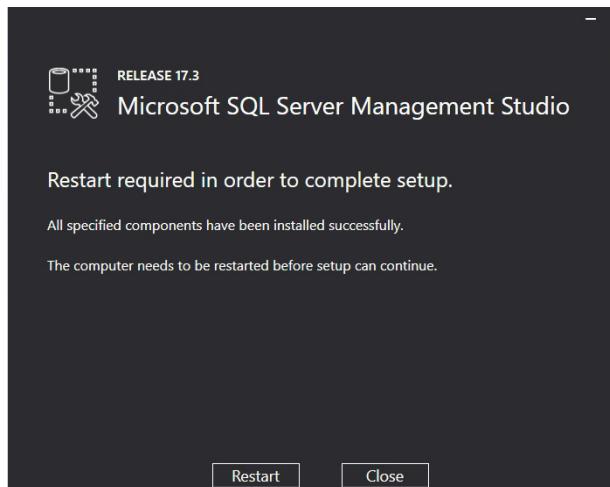


**Рисунок 2.22.** Загрузка пакетов установки

После того как пакеты будут загружены начнется сам процесс установки SQL Server Management Studio (Рисунок 2.23).



**Рисунок 2.23.** Процесс установки SQL Server Management Studio



**Рисунок 2.24.** Окончание установки SQL Server Management Studio

По окончании установки SQL Server Management Studio появится окно, в котором предлагается перезагрузить ваш компьютер, для чего необходимо нажать кнопку **Restart** (Рисунок 2.24).

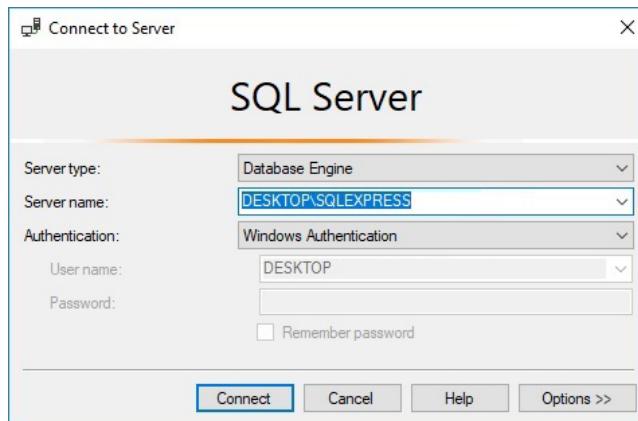
После перезагрузки компьютера в списке установленных программ появится пункт **Microsoft SQL Server Tools 17**, раскрыв который вы увидите пункт **Microsoft SQL Server Management Studio 17** (Рисунок 2.25).



**Рисунок 2.25.** Microsoft SQL Server Management Studio

## Управление базой данных

Чтобы создать базу данных, запускаем Management Studio, нажав соответствующий пункт (Рисунок 2.25).



**Рисунок 2.26.** Подключение к серверу

При первом запуске вы увидите окно загрузки пользовательских настроек, после чего запустится оболочка и появится окно подключения к серверу (Рисунок 2.26).

В открывшемся окне можно сделать несколько дополнительных настроек, кратко остановимся на каждой из них.

При помощи первой настройки (*Server type*) можно задать одну из технологий управления и анализа данных:

- **Database Engine** — основная служба, обеспечивающая безопасность данных, их хранение и обработку (значение по умолчанию);
- **Analysis Services** — служба, предназначенная для обработки аналитических данных и включает набор средств для бизнес-аналитики;
- **Reporting Services** — служба, предоставляющая средства для создания различного рода отчетов, с возможностью их публикации в разнообразных форматах;
- **Integration Services** — служба, которая включает различного рода решения по интеграции данных (извлечение, преобразование, загрузка) (Рисунок 2.27).

Имя сервера (*Server name*) прописывается автоматически, но в случаях, когда требуется указать его явно, необходимо открыть выпадающий список и выбрать требуемый сервер (Рисунок 2.28).

При помощи последней настройки (*Authentication*) можно задать режим проверки подлинности пользователя:

- **Windows Authentication** — этот режим задан по умолчанию и позволяет пользователю при подключении использовать учетную запись ОС Windows;

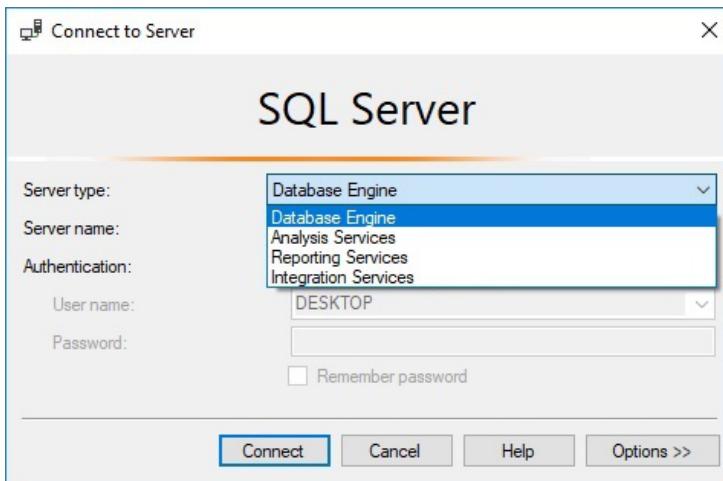


Рисунок 2.27. Технологии управления и анализа данных

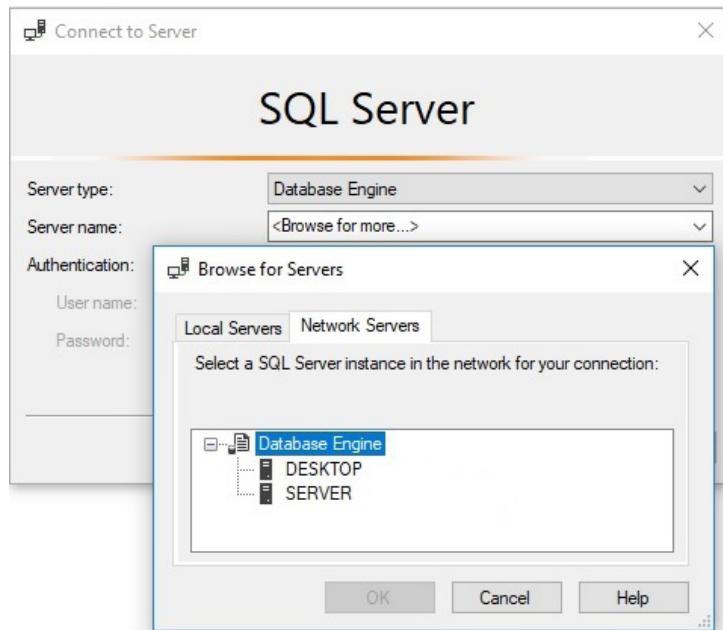


Рисунок 2.28. Выбор сервера

- **SQL Server Authentication** — режим, при котором подлинность пользователя определяется средствами SQL Server, в этом случае проверяется соответствие имени пользователя и пароля;
- **Active Directory – Universal with MFA support** — режим, который обеспечивает аутентификацию пользователя при помощи различных вариантов проверки (телефонный звонок, текстовое сообщение и т.д.);
- **Active Directory – Password** — режим аутентификации, который позволяет подключиться к базе данных Microsoft Azure SQL, используя идентификаторы в Azure Active Directory, если текущая учетная запись не связана с Azure;
- **Active Directory – Integrated** — режим аутентификации, который подобен предыдущему, но применяется, если при входе в OC Windows использовалась учетная запись Azure Active Directory (Рисунок 2.29).

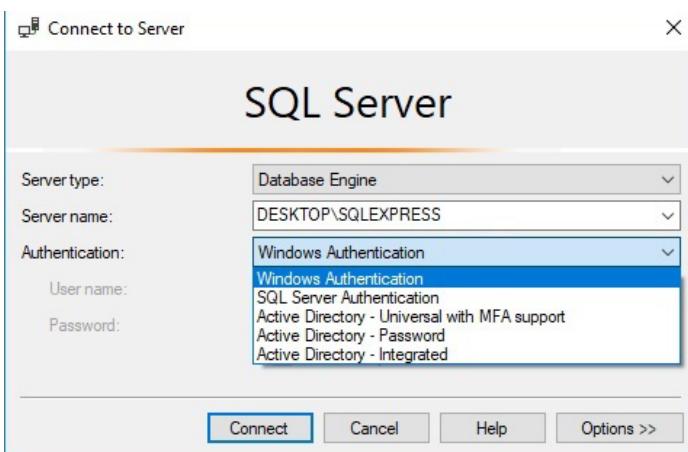


Рисунок 2.29. Выбор режима аутентификации

Оставьте все настройки по умолчанию (Рисунок 2.26) и нажмите кнопку **Connect**. После этого откроется Management Studio, в левой части которой вы увидите окно **Object Explorer**. В этом окне находится список папок, в которых содержатся различные настройки SQL Server, сейчас нас интересует папка **Databases**, с остальными папками ознакомимся по мере необходимости (Рисунок 2.30).

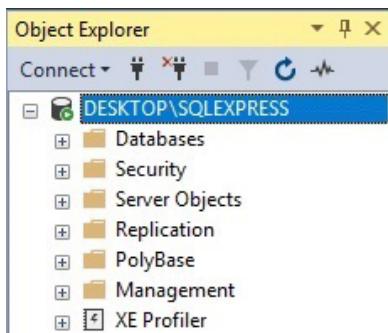


Рисунок 2.30. Окно Object Explorer

## *Создание базы данных*

Для того чтобы создать новую базу данных необходимо на папке **Databases** нажать правой клавишей мыши и в контекстном меню выбрать пункт **New Database...** (Рисунок 2.31).

После этих действий появится окно **New Database**, в котором необходимо указать имя базы данных, в нашем случае **University**, при этом автоматически изменяются логические имена файлов БД (*таблица Database files*). Также можно указать владельца базы данных, здесь мы оставим значение по умолчанию (*default*) (Рисунок 2.32).

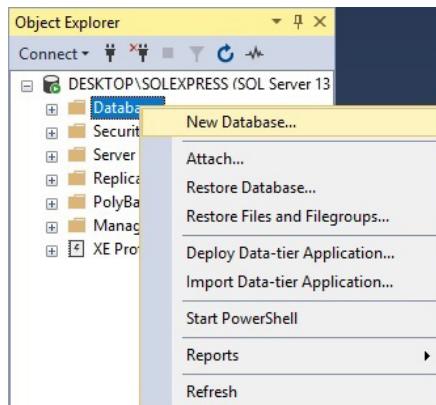


Рисунок 2.31. Начало создания новой базы данных

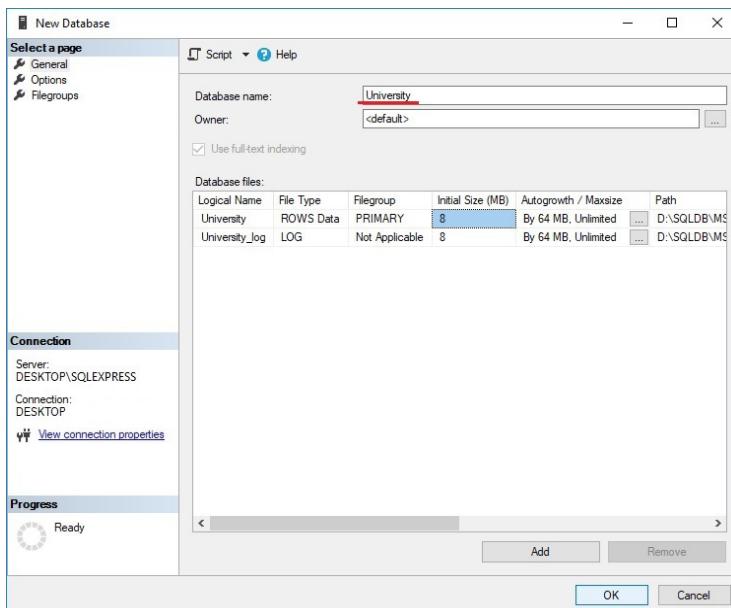


Рисунок 2.32. Имя новой базы данных

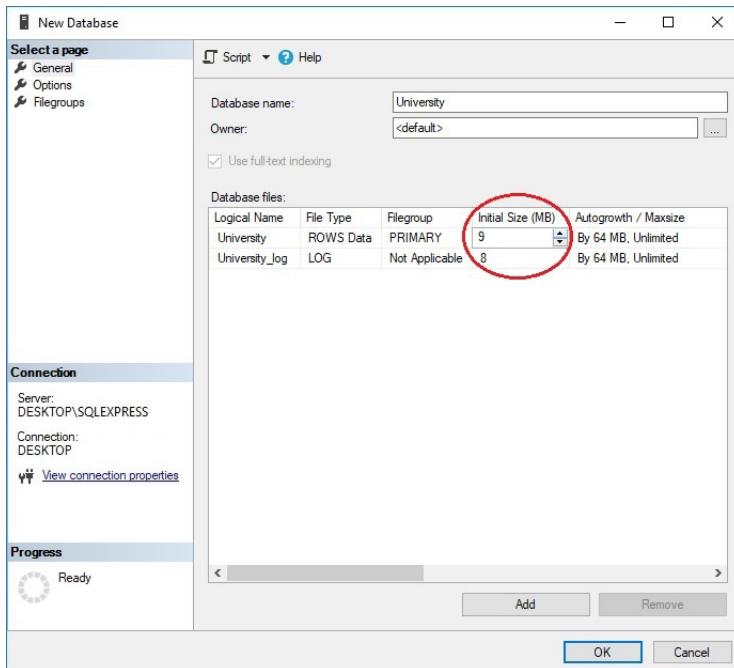
Хотя для создания базы данных изменять остальные настройки не обязательно, мы остановимся на них более подробно.

## Настройка параметров базы данных

Любая база данных в Microsoft SQL Server состоит из двух физических файлов. Файл с расширением **.mdf** — это основной файл данных, в нем содержится информация обо всех объектах БД. Файл с расширением **.ldf** — это файл журнала транзакций, он играет роль некоего предварительного хранилища данных. Параметры этих файлов находятся в таблице **Database files**.

### Изменение размера базы данных

Для того чтобы изменить размер базы данных необходимо настроить три параметра: начальный размер,

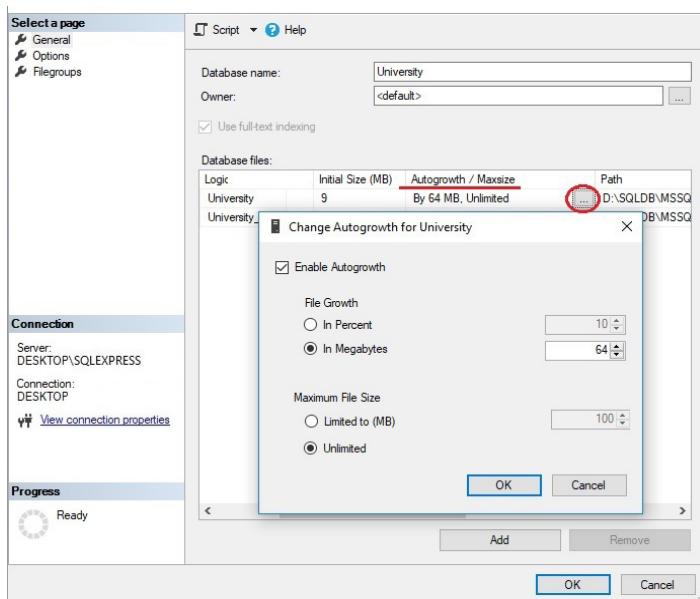


**Рисунок 2.33.** Изменение начального размера файлов базы данных

автоматическое приращение размеров и максимальный размер файлов БД.

Начальный размер файлов базы данных изменяется указанием целочисленного значения размера в поле **Initial Size** таблицы **Database files** (Рисунок 2.33).

Максимальный размер и автоматическое приращение размера конкретного файла базы данных можно настроить, нажав кнопку с тремя точками в соответствующей ячейке поля **Autogrowth/Maxsize**, после чего появиться окно **Change Autogrowth for University** (Рисунок 2.34).



**Рисунок 2.34.** Изменение максимального размера и приращения размера файла БД

В окне **Change Autogrowth for University** вы можете задать автоматическое приращение размера файла либо в процентах, либо в мегабайтах, указав необходимое

значение. Аналогичным образом можно задать максимальный размер файла базы данных, либо указав размер в мегабайтах, либо выбрав пункт **Unlimited**.

## Управление группами файлов

Файлы данных базы данных подразделяются на файловые группы, которые используются для объединения файлов с целью повышения производительности. В любой базе данных существует файловая группа **PRIMARY**, в ней находится первичный файл данных. Также существует возможность создания пользовательских файловых групп, для этого необходимо в списке **Select a page** выбрать пункт **Filegroups**, после чего появится соответствующий интерфейс (Рисунок 2.35).

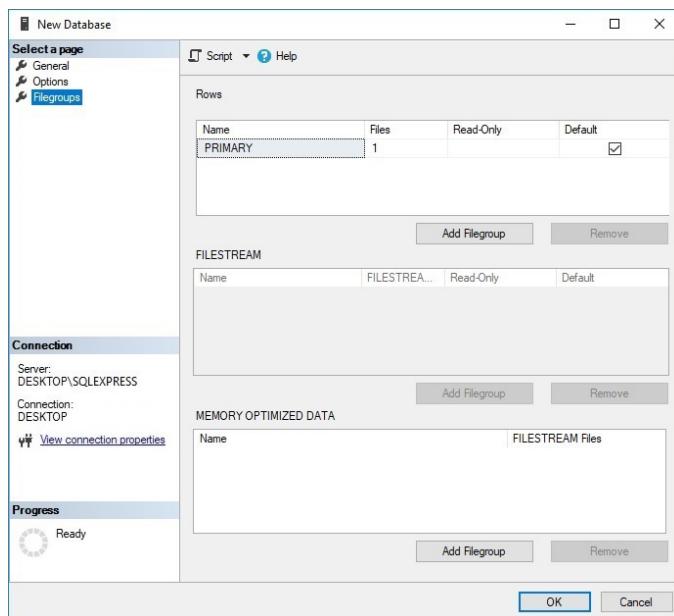


Рисунок 2.35. Управление группами файлов

После того как вы сделаете все необходимые настройки вашей базы данных и нажмете кнопку **OK** в окне **New Database**, вы увидите индикатор создания БД (Рисунок 2.36).

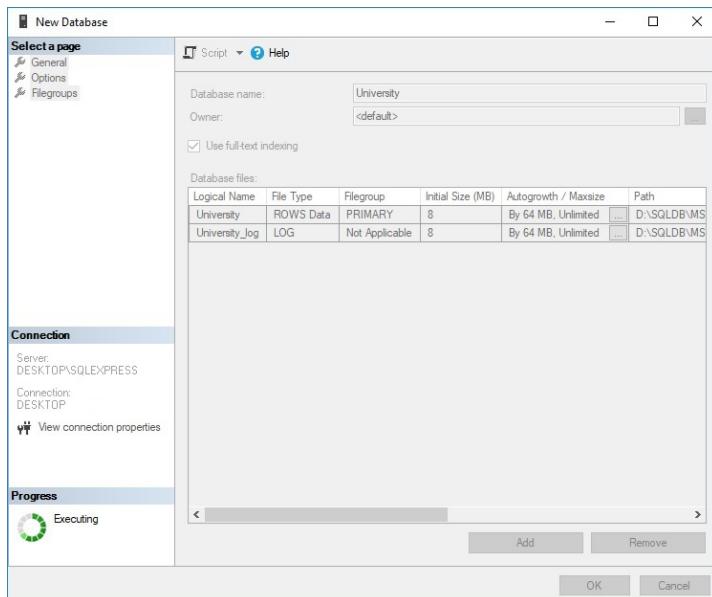


Рисунок 2.36. Процесс создания базы данных

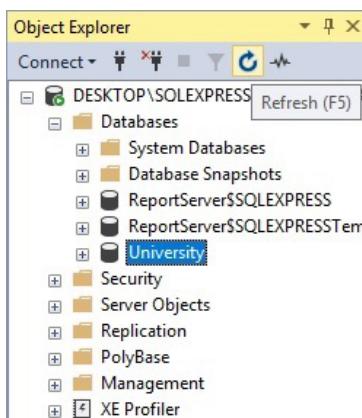


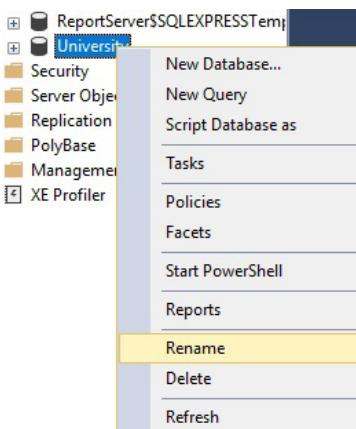
Рисунок 2.37. Результат создания новой базы данных

После закрытия окна, ваша база данных появится в списке **Databases** окна **Object Explorer** (Рисунок 2.37).

Если по какой-то причине, созданная вами база данных не отобразилась в списке, нажмите кнопку **Refresh** (Рисунок 2.37) и список баз данных обновится.

### **Переименование базы данных**

После того как база данных создана, существует возможность внесения в нее ряда изменений, например ее можно переименовать. Для этого необходимо щелкнуть правой клавишей мыши на названии редактируемой базы данных и в контекстном меню выбрать пункт **Rename**, а затем ввести новое имя (Рисунок 2.38).



**Рисунок 2.38.** Переименование базы данных

### **Удаление базы данных**

Существует также возможность удаления существующей базы данных, для этого необходимо вызвать контекстное меню на названии конкретной базы данных и выбрать пункт **Delete** (Рисунок 2.39).

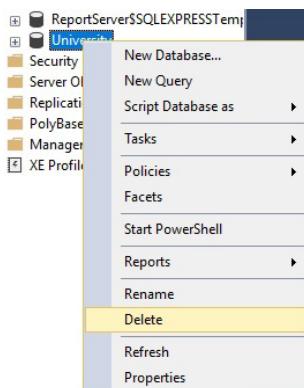


Рисунок 2.39. Начало удаления базы данных

После этого появится окно **Delete Object**, в котором после нажатия кнопки **OK** вы увидите индикатор процесса удаления (Рисунок 2.40).

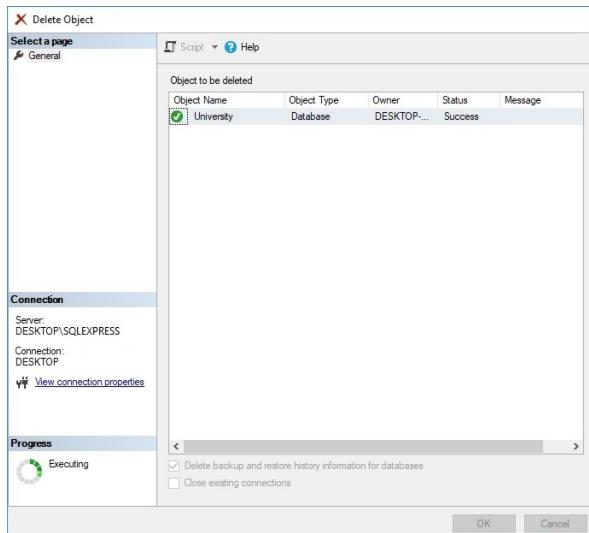


Рисунок 2.40. Завершение удаления базы данных

После того как окно закроется, ваша база данных исчезнет из списка **Databases** окна **Object Explorer**.

## 3. Домашнее задание

---

1. Установить Microsoft SQL Server 2016 Express и SQL Server Management Studio на свой компьютер.
2. Создать собственную базу данных. Задать: файлу данных начальный размер 15 МВ, приращение размера файла 15% и неограниченный максимальный размер; файлу транзакций — начальный размер 10 МВ, автоматическое приращение размера в 5 МВ и максимальный размер в 60 МВ.



## Урок № 1

# Введение в теорию баз данных

© Юрий Задерей.  
© Компьютерная Академия «Шаг».  
[www.itstep.org](http://www.itstep.org)

Все права на охраняемые авторским правом фото-, аудио- и видеопрограммные средства, фрагменты которых использованы в материале, принадлежат их законным владельцам. Фрагменты произведений используются в иллюстративных целях в объеме, оправданном поставленной задачей, в рамках учебного процесса и в учебных целях, в соответствии со ст. 1274 ч. 4 ГК РФ и ст. 21 и 23 Закона Украины «Про авторське право і суміжні права». Объем и способ цитируемых произведений соответствует принятым нормам, не наносит ущерба нормальному использованию объектов авторского права и не ущемляет законные интересы автора и правообладателей. Цитируемые фрагменты произведений на момент использования не могут быть заменены альтернативными, не охраняемыми авторским правом аналогами, и как таковые соответствуют критериям добросовестного использования и честного использования.

Все права защищены. Полное или частичное копирование материалов запрещено. Согласование использования произведений или их фрагментов производится с авторами и правообладателями. Согласованное использование материалов возможно только при указании источника.

Ответственность за несанкционированное копирование и коммерческое использование материалов определяется действующим законодательством Украины.