



Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Магнитогорский государственный технический университет им. Г.И. Носова»

**И.И. Баранкова**  
**У.В. Михайлова**  
**Г.И. Лукьянов**

**РАЗРАБОТКА БД В MS SQL SERVER  
С ИСПОЛЬЗОВАНИЕМ SSMS**

*Утверждено Редакционно-издательским советом университета  
в качестве учебного пособия*

Магнитогорск  
2018

**Рецензенты:**

кандидат технических наук,  
заместитель директора  
СЦ ООО «ТЕХНОАП Инжиниринг» в г. Магнитогорск  
**Д.В. Швидченко**

кандидат физико-математических наук,  
заведующий кафедрой физики,  
ФГБОУ ВО «Магнитогорский государственный технический  
университет им. Г.И. Носова»  
**Ю.И. Савченко**

**Баранкова И.И., Михайлова У.В., Лукьянов Г.И.**

**Разработка БД В MS SQL Server с использованием SSMS** [Электронный ресурс] : учебное пособие / Инна Ильинична Баранкова, Ульяна Владимировна Михайлова, Георгий Игоревич Лукьянов ; ФГБОУ ВО «Магнитогорский государственный технический университет им. Г.И. Носова». – Электрон. текстовые дан. (1,74 Мб). – Магнитогорск : ФГБОУ ВО «МГТУ им. Г.И. Носова», 2018. – 1 электрон. опт. диск (CD-R). – Систем. требования : IBM PC, любой, более 1 GHz ; 512 Мб RAM ; 10 Мб HDD ; MS Windows XP и выше ; Adobe Reader 8.0 и выше ; CD/DVD-ROM дисковод ; мышь. – Загл. с титул. экрана.

ISBN 978-5-9967-1207-6

Разработанное электронное издание «Разработка БД В MS SQL Server с использованием SSMS» соответствует требованиям ФГОС ВО и образовательной программе по специальности 10.05.03 «Информационная безопасность автоматизированных систем», специализация «Обеспечение информационной безопасности распределенных информационных систем». Данное электронное издание полностью соответствует рабочей программе по дисциплине «Информационные технологии. Базы данных», а так же может использоваться в рамках изучения дисциплин: «Информатика», «Технологии и методы программирования».

В разработанном электронном издании рассмотрены основы языка SQL на примере MS SQL Server. Данное электронное издание освещает достаточный минимум, который необходим при работе с базами данных.

УДК 004.658

## Содержание

Введение .....	4
1. Установка и настройка MS SQL SERVER 2014.....	4
1.1. Планирование установки SQL Server .....	4
1.2. Установка Microsoft SQL Server 2014.....	8
1.3. Конфигурирование SQL Server для сетевого доступа .....	16
2. SQL Server Management Studio.....	20
3. Язык описания данных - DDL.....	23
3.1. Создание БД.....	24
3.2. Создание таблиц.....	26
3.3. Команды ALTER и INSERT .....	28
3.4. Первичный ключ и внешний ключ .....	29
3.5. Дополнительные опции .....	31
3.6. Ограничения UNIQUE, DEFAULT, CHECK.....	33
3.7. Имена входа и пользователи.....	34
3.7.1. Имена входа .....	34
3.7.2. Создание пользователей .....	38
Библиографический список.....	42
Приложение. Функции и возможности SQL ServerExpress 2017 .....	43

## **ВВЕДЕНИЕ**

В данном пособии рассмотрены основы языка SQL на примере MSSQL Server. Рассмотрен достаточный минимум, который необходим при работе с базами данных. Если встанет необходимость применять более полные конструкции SQL, то найти информацию о них можно в библиотеке MSDN расположенную в сети интернет.

Язык манипулирования базами данных SQL подразделяется на несколько частей. Рассмотрим две наиболее важные его части:

1. DDL – Data Definition Language (язык описания данных);
2. DML – Data Manipulation Language (язык манипулирования данными).

Для работы с базами данных будем использовать SQL Server Management Studio. SQL Server Management Studio (SSMS) -утилита для Microsoft SQL Server, которая используется для конфигурирования, управления и администрирования компонентов базы данных. Данная утилита содержит редактор скриптов (который в основном и будет нами использоваться) и графическую оболочку, которая работает с объектами и настройками сервера. Главным инструментом SQL Server Management Studio является Object Explorer, который позволяет пользователю просматривать, извлекать объекты сервера, а также управлять ими.

В данном пособии рассмотрено установка и настройка MS SQL Server, конфигурирование SQL Server для сетевого доступа, а так же основы языка описания данных – DDL.

## **1. УСТАНОВКА И НАСТРОЙКА MS SQL SERVER 2014**

### **1.1. Планирование установки SQL Server**

Microsoft SQL Server - система управления реляционными базами данных (СУРБД), разработанная корпорацией Microsoft. Основным используемый язык запросов -Transact-SQL (T-SQL). Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия, успешно конкурирует с другими СУБД в этом сегменте рынка. Решение о том, какой выпуск SQL Server следует использовать, достаточно сильно зависит от того, какие приложения планируется использовать. Выбор выпуска зависит от количества и размера обслуживаемых баз данных, а так же от того, какие клиентские приложения планируется использовать для работы (см. таблицу 1, 2, 3).

## Выпуски Microsoft SQL Server

Характеристика выпуска	Выпуск SQL Server	Определение
Основной	Enterprise Edition	Предложение высшего класса, обеспечивающее полный набор возможностей центра обработки данных (ЦОД) с высочайшей производительностью, неограниченными возможностями виртуализации и развитыми средствами бизнес-аналитики, позволяющими добиться высокого уровня обслуживания важнейших рабочих нагрузок, предоставлять конечным пользователям доступ к анализу данных.
Основной	Business Intelligence	Комплексная платформа, дающая организациям возможность строить и развертывать надежные, масштабируемые и управляемые решения бизнес-аналитики. В выпуск включены функции просмотра и представления данных в браузере, мощные возможности комбинирования данных и расширенное управление интеграцией.
Основной	Express	Бесплатная база данных начального уровня, которая идеально подходит для создания небольших приложений для обработки данных, занимающих до 10 ГБ места на диске, и развертывания небольших баз данных в производственных средах.

Характеристика выпуска	Выпуск SQL Server	Определение
Основной	Developer	Бесплатный, обладающий полным набором функций выпуск SQL Server 2016 SP1 Enterprise для создания, тестирования и демонстрации приложений в непроизводственной среде.

Таблица 2

## Компоненты SQL Server

Компоненты сервера	Описание
SQL Server Database Engine	В состав компонента SQL Server Database Engine входит компонент Database Engine, основная служба для хранения, обработки и обеспечения безопасности данных, репликации, полнотекстового поиска, средств управления реляционными и XML-данными и сервера служб Data Quality Services (DQS).
Analysis Services	содержат средства создания и управления приложениями оперативной аналитической обработки (OLAP) и приложениями интеллектуального анализа данных
Службы Reporting Services	включают в себя серверные и клиентские компоненты для создания, управления и развертывания табличных, матричных и графических отчетов, а также отчетов в свободной форме. Службы Reporting Services являются расширяемой платформой, которую можно использовать для разработки приложений отчетов.
Службы Integration Services	представляют собой набор графических средств и программируемых объектов для перемещения, копирования и преобразования данных. Они так же включают компонент DQS для служб Службы Integration Services.
Master Data Services (MDS)	решение SQL Server по управлению основными данными. MDS можно настроить для управления любой структурой (товары, заказчики, счета). Поддерживаются иерархии, детальная настройка безопасности, транзакции, управление версиями данных и бизнес-правила, а также использование надстройка для Excel для управления данными

## Средства управления SQL Server

Средства управления	Описание
SQL Management Studio Serv	интегрированная среда для доступа, настройки, управления, администрирования и разработки всех компонентов SQL Server. Среда Management Studio позволяет разработчикам и администраторам, обладающим различными уровнями навыков, использовать SQL Server. Обозреватель Internet Explorer 6 с пакетом обновления 1 или более поздней версии необходим для любой установки среды Среда Management Studio.
Диспетчер конфигурации SQL Server	обеспечивает базовые возможности управления конфигурациями для служб, серверных протоколов, клиентских протоколов и псевдонимов клиентов SQL Server.
Приложение SQL Server Profiler	Приложение SQL Server Profiler предоставляет графический пользовательский интерфейс для наблюдения за экземпляром компонента Компонент Database Engine или служб Analysis Services.
Помощник по настройке ядра СУБД (Компонент Database Engine)	помогает создавать оптимальные наборы индексов, индексированных представлений и секций.
Клиент Data Quality	простой и понятный графический пользовательский интерфейс для подключения к серверу DQS и выполнения операций очистки данных. Он также позволяет централизованно отслеживать различные действия, выполняемые во время операции очистки данных. Для установки клиента Data Quality необходим обозреватель Internet Explorer 6 с пакетом обновления 1 (SP1) или более поздней версии.

Средства управления	Описание
SQL Server Data Tools	содержат интегрированную среду разработки, предназначенную для создания решений для следующих компонентов бизнес-аналитики. Analysis Services, Службы Reporting Services и Службы Integration Services. (Панель — среда Business Intelligence Development Studio.) SSDT также содержит компонент «проекты баз данных», который предоставляет интегрированную среду для разработчиков, предназначенную для выполнения всех работ по разработке баз данных для любой платформы SQL Server (как в основном помещении, так и вне его) в пределах Visual Studio. Разработчикам баз данных предлагается расширенный обозреватель серверов, который является компонентом Visual Studio, предназначенным для облегчения процессов создания и изменения объектов баз данных и данных в них, а также для выполнения запросов. Для установки SQL Server Data Tools необходим браузер Internet Explorer 6 с пакетом обновления 1 (SP1) или более поздней версии.
Компоненты связи	Устанавливает компоненты для связи между клиентами и серверами и сетевые библиотеки для DBбиблиотеки, ODBC и OLEDB.

Для бесплатного, лицензионного использования БД MSSQL Server можно использовать выпуск Expressверсии 2014 и выше (доступен выпуск 2017 года)[1]. Краткий обзор возможностей MSSQL Server 2017Expressприведен в приложении 1.

## 1.2. Установка Microsoft SQL Server 2014

Рассмотрим этапы установки SQL Server.

Сначала необходимо запустить *программу-установщик* с правами администратора на данном компьютере. В разделе «Планирование» нажать пункт «Средство проверки конфигурации» и убедиться, что все проверки будут успешно пройдены (рис. 1):

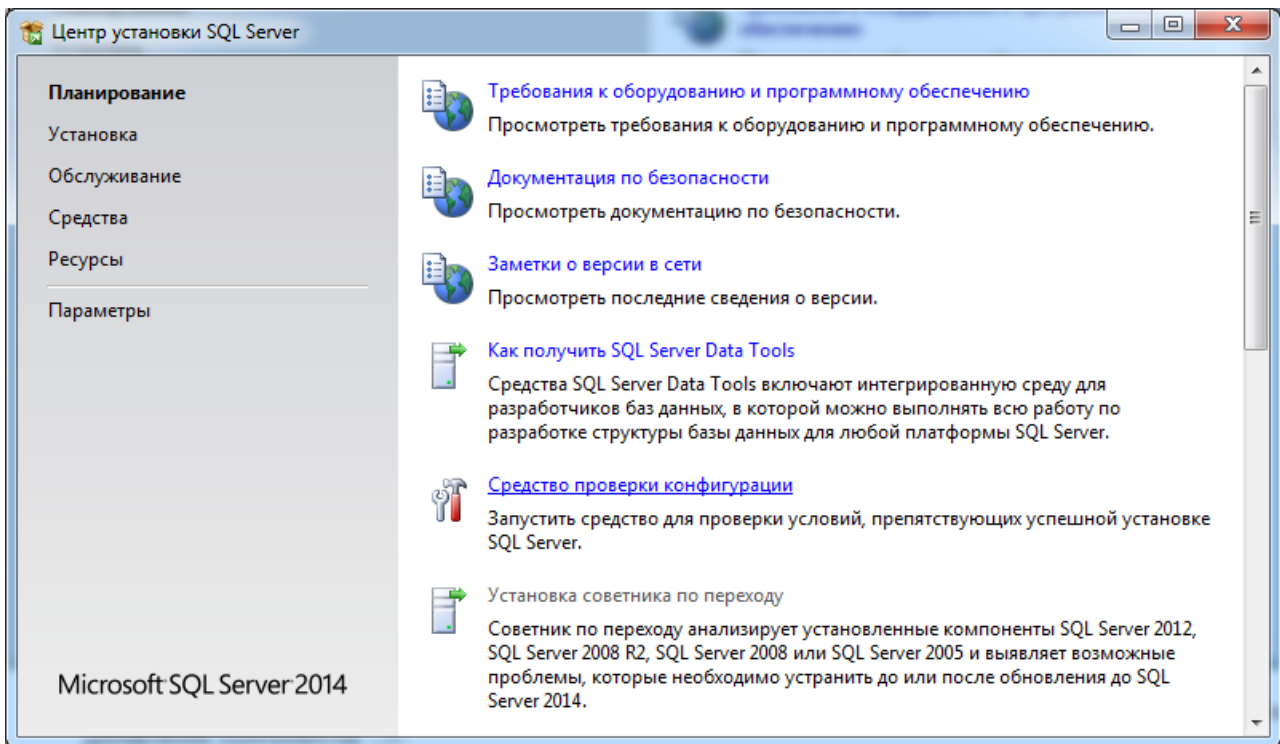


Рис.1. Центр установки

Затем необходимо перейти на вкладку «Установка» (Installation) и нажать на «Новая установка изолированного экземпляра SQL Server» или «Добавление компонентов к существующей установке» (рис. 2):

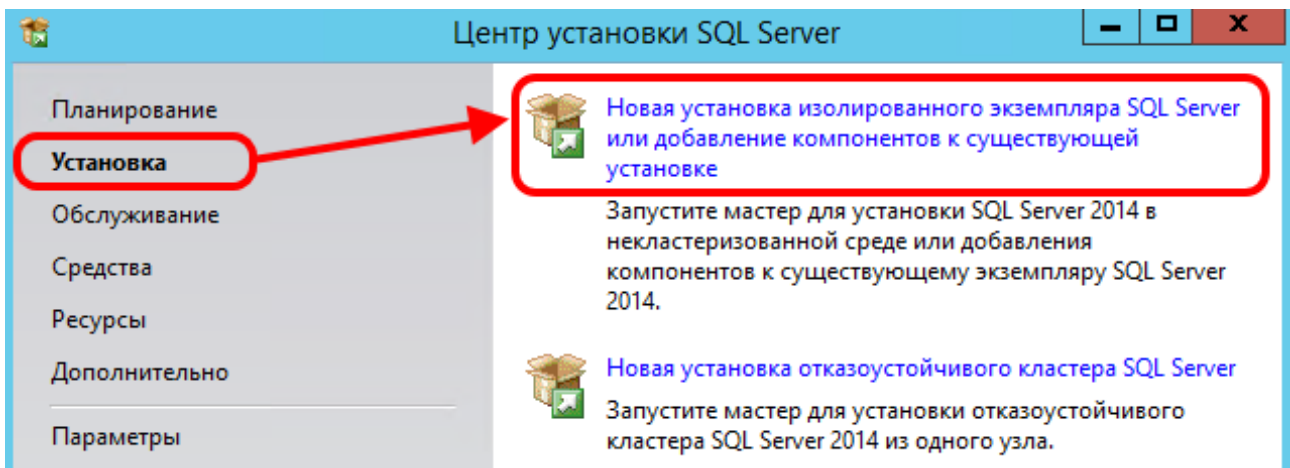


Рис.2. Диалоговое окно установки

Затем принимаем условия лицензионного соглашения, соглашаемся (или нет) на передачу данных об использовании компонентов в корпорацию Microsoft. Жмем «Далее» (Next). Убеждаемся в том, что нет проблем, которые могут возникнуть во время установки и, если все нормально, жмем «Далее» (Next). Если все же имеются какие-либо проблемы, необходимо устранить их и повторить процедуру проверки нажав кнопку «Включить заново» (Re-run).

Затем выбираем компоненты для установки (рекомендуется воспользоваться кнопкой «Выделить все») и нажать кнопку «Далее» (рис. 3):

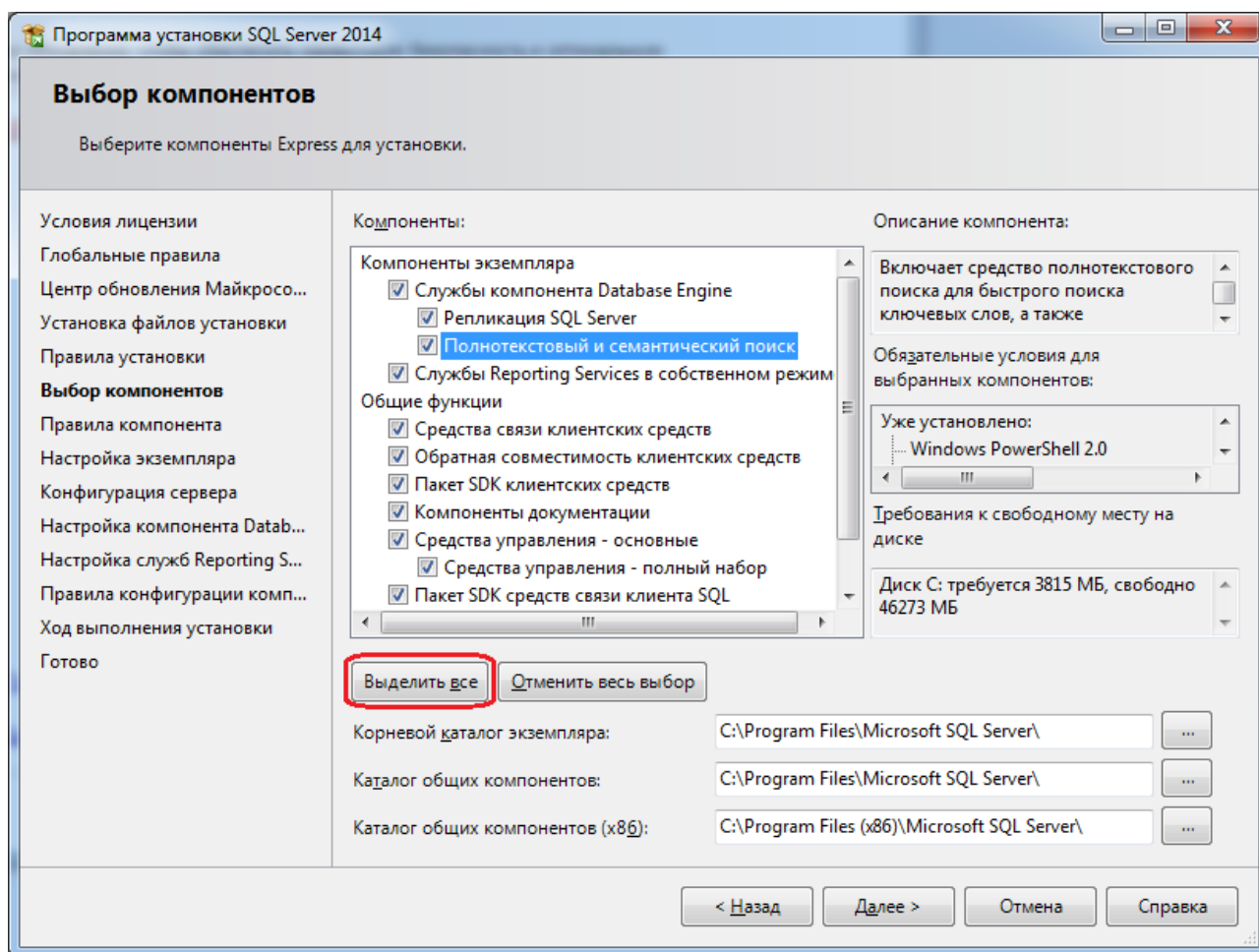


Рис.3. Выбор компонентов

В очередной раз запускается проверка требований для установки выбранных на предыдущем шаге компонент (набор требований зависит от устанавливаемых компонент). Например, если заранее не установить платформу Microsoft .NET Framework 3.5, то появится соответствующая ошибка. Запуск проверки снова осуществляется с помощью кнопки «Включить заново» (Re-run). После удовлетворения всех требований системы нажимаем кнопку «Далее» (Next) (рис. 4):

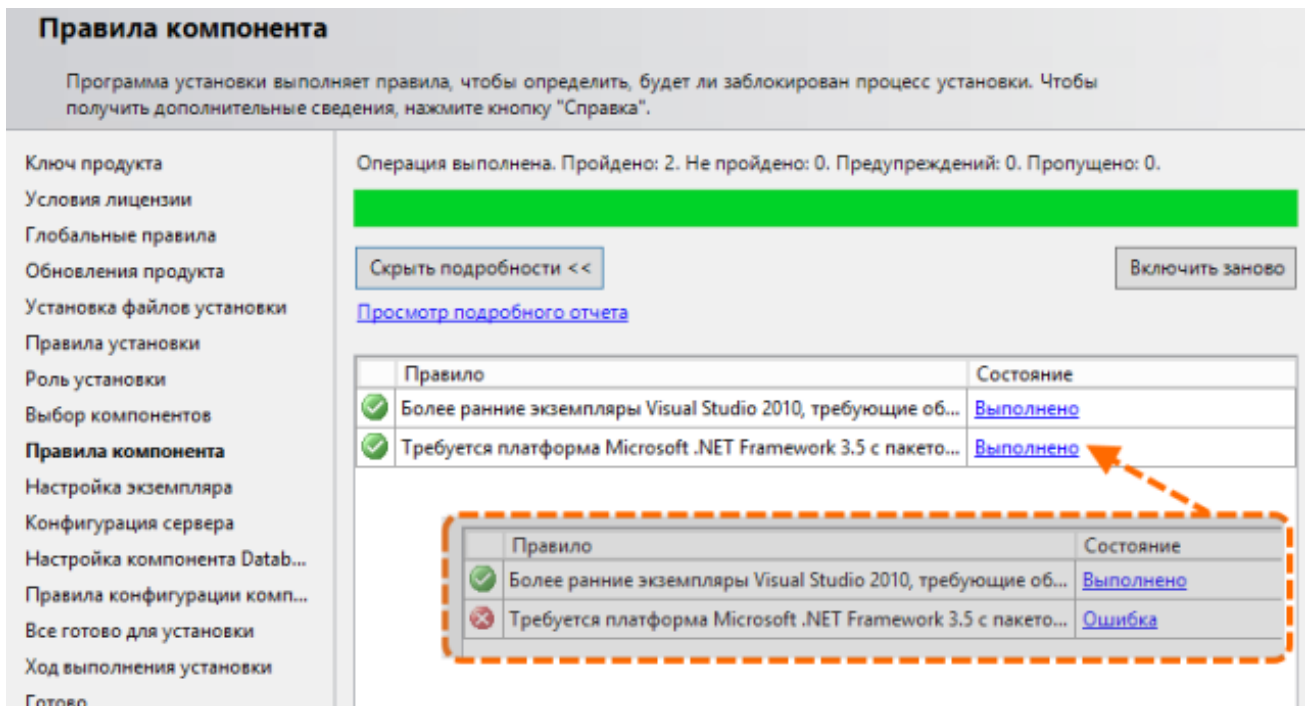


Рис.4. Правила компонентов

Теперь необходимо ввести имя именованного экземпляра SQL Server или оставить имя экземпляра по умолчанию. На одном изолированном сервере возможно устанавливать до 50 именованных экземпляров MS SQL Server 2014 и до 25 в отказоустойчивом кластере при использовании общего диска кластера. Экземпляр по умолчанию (идентификатор MSSQL Server) может быть только один. Определившись с именем и идентификатором экземпляра SQL Server или выбрав экземпляр по умолчанию нажимаем кнопку «Далее» (рис. 5).

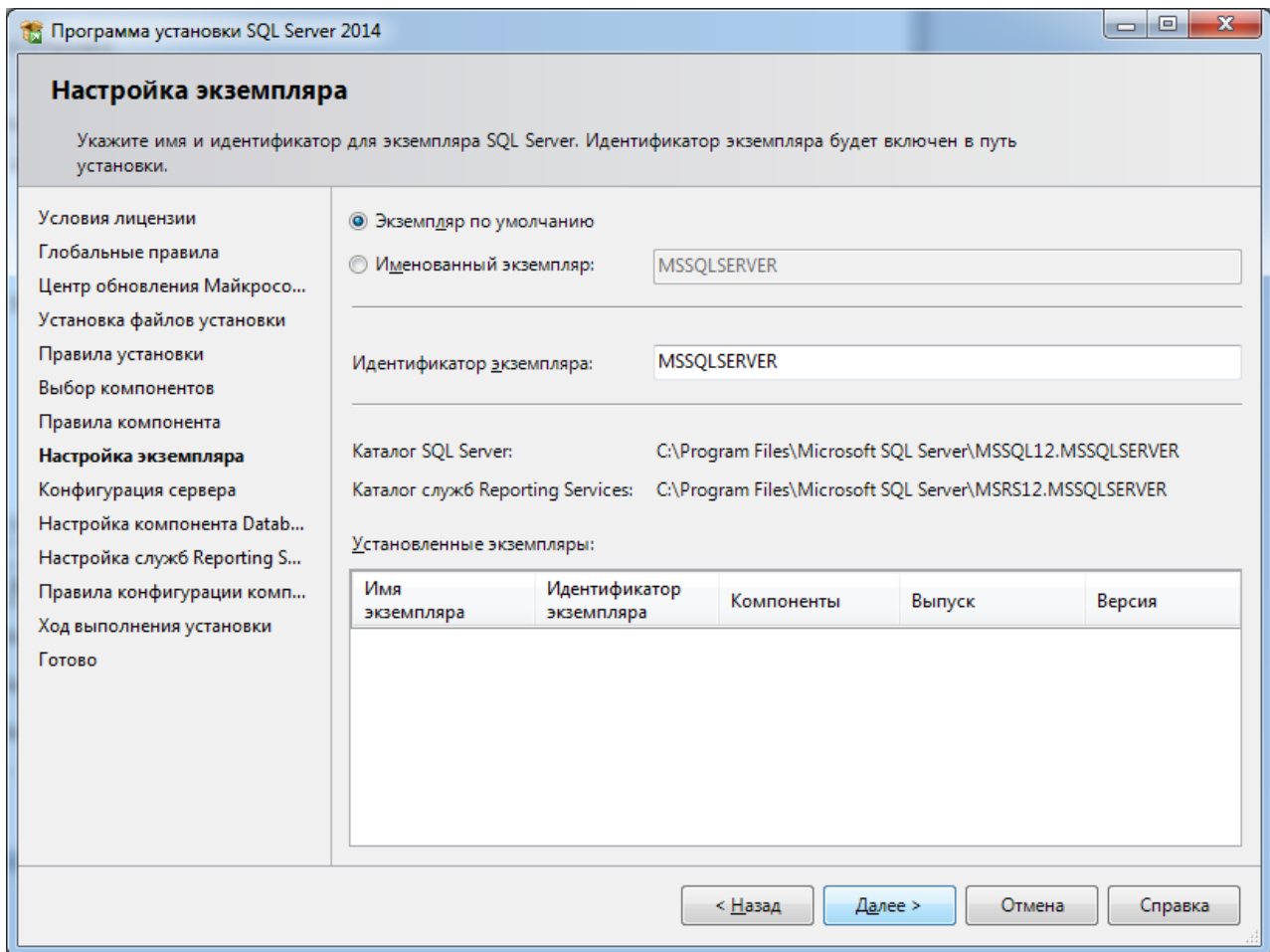


Рис.5. Настройка экземпляра

Теперь выберем режим запуска служб SQL Server. Выбрать опции, как показано на рис. 6:

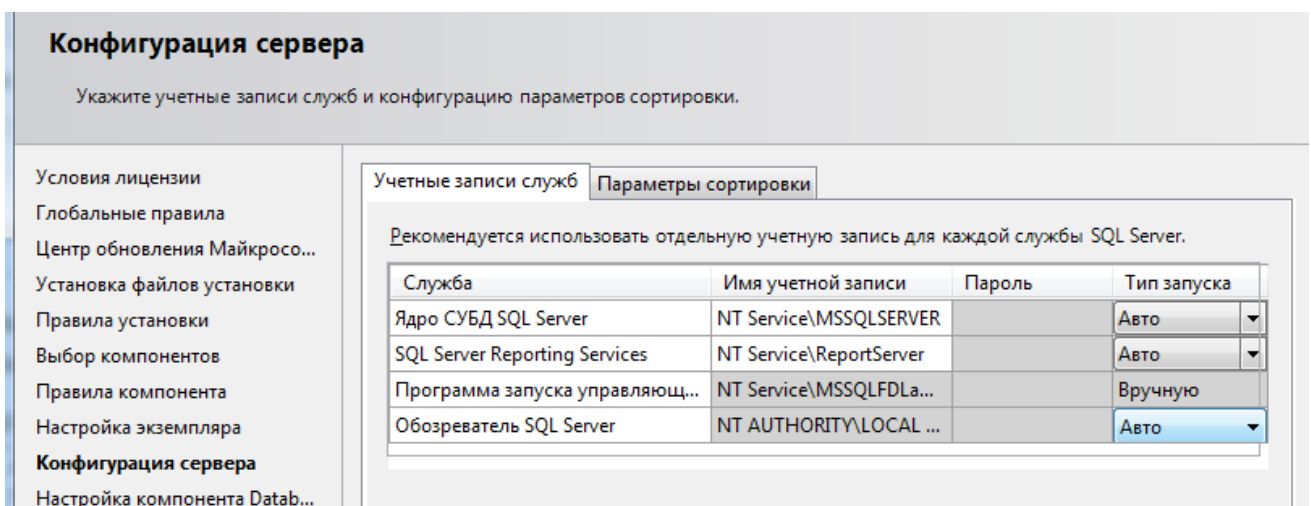


Рис. 6. Режим SQL Server

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

Далее перейдем на вкладку «Параметры сортировки». Здесь необходимо определиться с параметрами сортировки баз данных по умолчанию. Для большинства программ, которые будут работать с SQL сервером достаточно выбрать «Cyrillic\_General\_CI\_AS» (рис. 7). Указав параметры сортировки нажимаем кнопку «Далее».

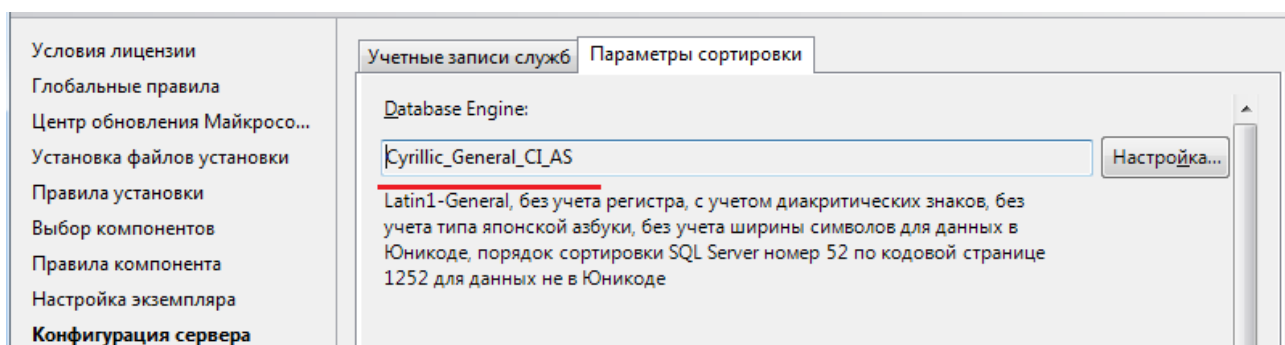


Рис.7. Параметры сортировки

Теперь следует выбрать режим проверки подлинности. В зависимости от способа подключения к SQL Server можно выбрать «Режим проверки подлинности Windows» (Windows authentication mode) или «Смешанный режим» (Mixed Mode) - в этом случае будут использоваться оба режима.

В случае выбора «Смешанного режима» (Mixed Mode) потребуется вести пароль на встроенную учетную запись «sa» (эта учетная запись обладает максимальными правами доступа ко всем функциям и объектам на SQL-сервере). Дополнительно можно указать учетные записи пользователей Windows или целые группы пользователей Windows, которые должны обладать максимальными правами доступа к SQL Server (например, встроенную группу «Администраторы») (рис. 8).

**Примечание. Безопасность.** Учетная запись «sa» — хорошо известная учетная запись SQL Server, поэтому она часто становится мишенью злоумышленников. Не включайте учетную запись «sa», если это не требуется для работы приложения. Для имени входа «sa» очень важно использовать надежный пароль.

Затем необходимо перейти на закладку «Каталоги данных» (рис. 8):

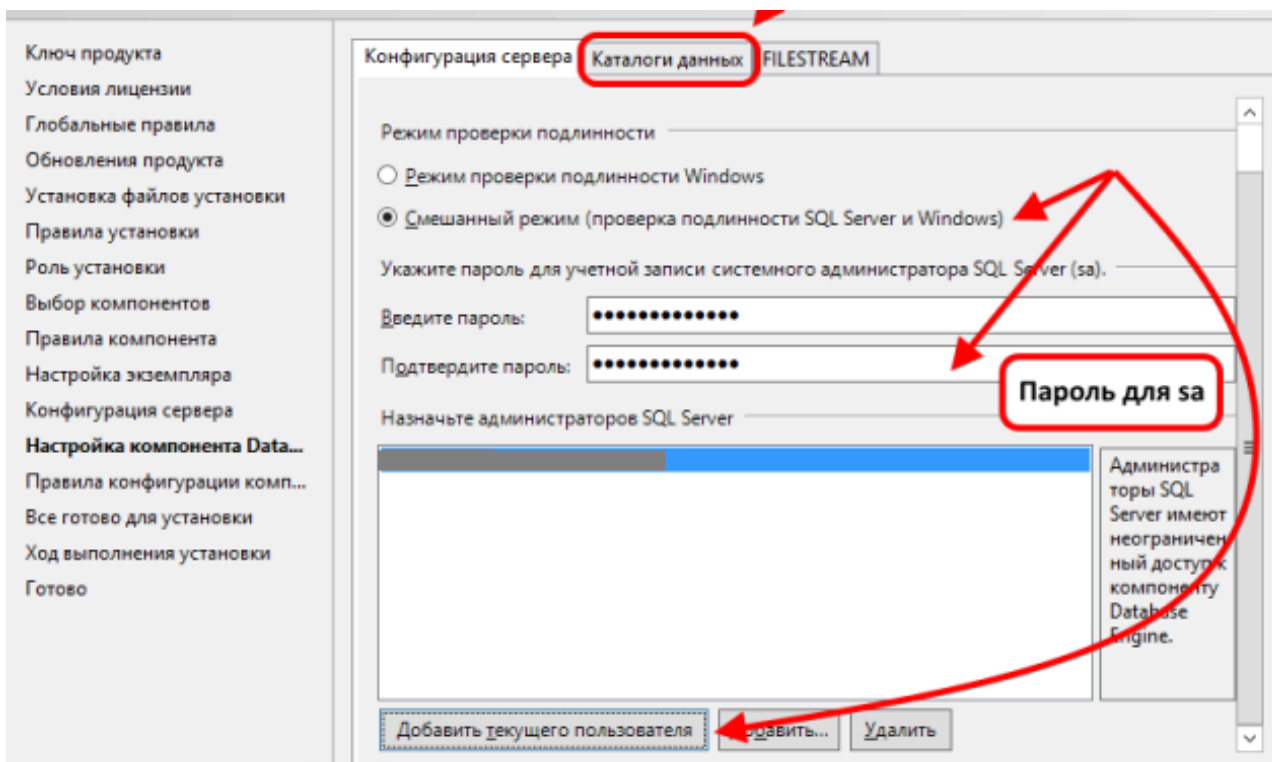


Рис.8. Добавление пользователя БД

На данной закладке можно выбрать каталог хранения баз данных (Data root directory). Для большинства случаев рекомендуется придерживаться следующих правил:

- Если в конфигурации сервера присутствует RAID-массив из жестких дисков (что крайне рекомендуется для сервера баз данных), разумно расположить файлы данных именно на нем.
- Для увеличения производительности рекомендуется разнести файлы баз данных и файлы журналов баз данных на разные диски (если есть такая возможность).
- Каталог системной базы данных temp и каталог журналов базы temp желательно вынести на быстрый SSD диск, это даст ощутимый прирост производительности SQL Server.
- Есть также смысл изменить каталог резервного копирования, выбрав физический диск, отличный от диска с файлами баз данных.

В поле «Корневой каталог данных» ввести путь к папке, где будут размещаться файлы баз данных (рекомендуется использовать отдельный от ОС физический диск), и перейти на закладку «Пользовательские экземпляры» (рис. 9):

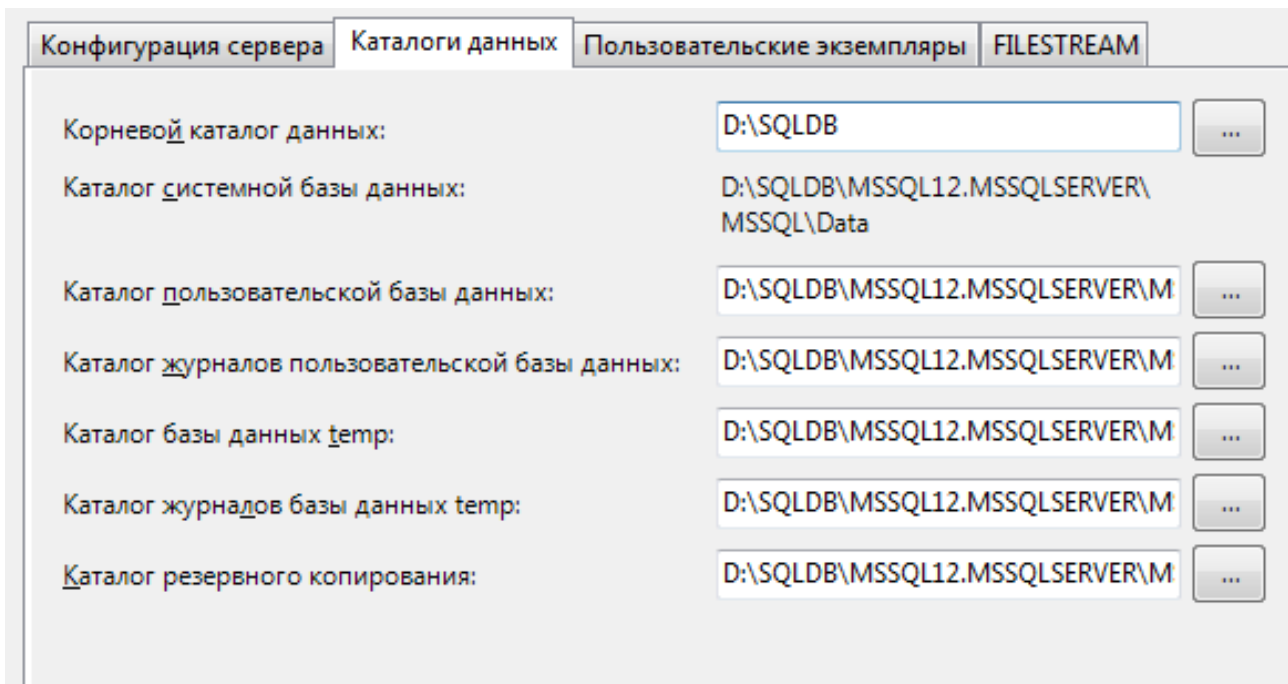


Рис.9. Каталоги данных

Затем выбрать опции, как показано на рис. 10:

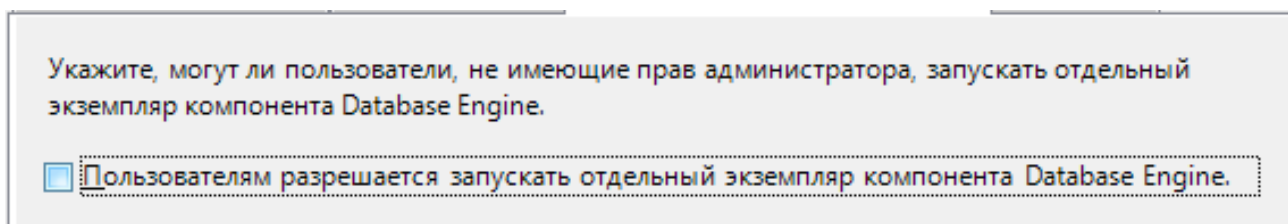


Рис.10. Пользовательский экземпляр

Затем переходим на вкладку «FILESTREAM». Если на данном SQL-сервере в будущем планируется хранить неструктурированные данные, например большие документы, изображения, видеофайлы и пр., и для хранения таких файлов будет использоваться хранилище FILESTREAM, то необходимо также настроить соответствующие параметры на данной вкладке. В противном случае, оставляем настройки без изменений и нажимаем «Далее».

Выбрать опции, как показано на рисунке 11, и нажать кнопку «Далее»:



Рис.11. Reporting Services

Дожидаемся окончания установки, убеждаемся, что все компоненты установлены успешно и завершаем работу мастера, нажав на кнопку «Заккрыть».

### 1.3. Конфигурирование SQL Server для сетевого доступа

Примечание. Дальнейшие шаги следует выполнять только, если требуется обеспечить доступ к SQL Server с других компьютеров в сети!

Запустить утилиту «Диспетчер конфигурации SQL Server» (Пуск → Все программы → Microsoft SQL Server 2014 → Средства настройки → Диспетчер конфигурации SQL Server). В разделе «Сетевая конфигурация SQL Server»→«Протоколы для ...» нажать правой кнопкой мыши на строке «TCP/IP» и в контекстном меню выбрать пункт «Свойства» (рис. 12):

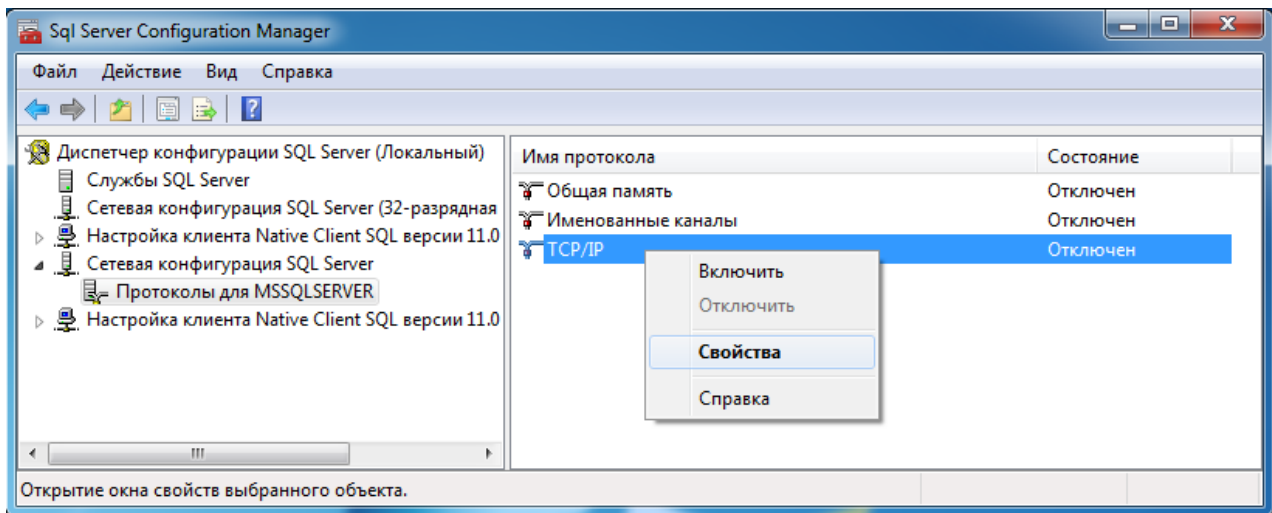


Рис.12. Менеджер конфигурации

На закладке «Протокол» установить параметр «Включено» = «Да», а на закладке «IP-адреса» в разделе «IPAll» параметр «TCP-порт» = «1433». Затем нажать кнопку «ОК» (рис. 13):

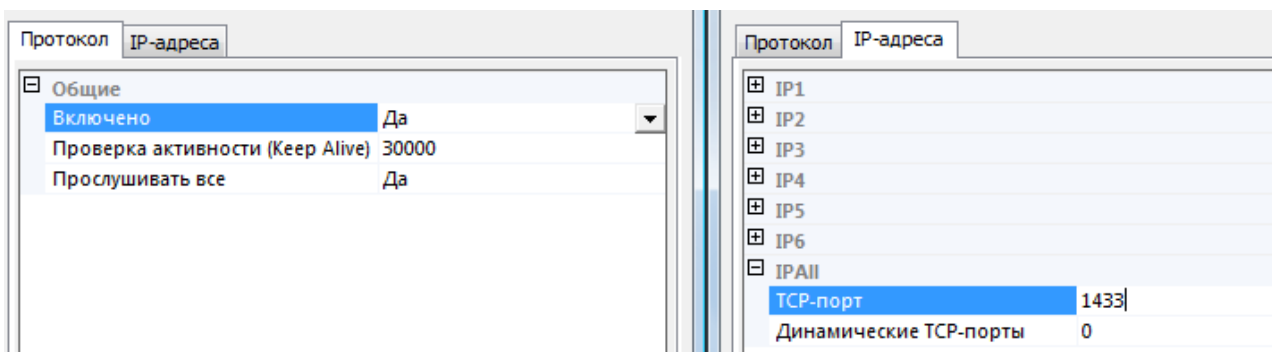


Рис.13. Свойства IP/TCP

В разделе «Службы SQL Server» нажать правой кнопкой мыши на строке «SQL Server (...)» и в контекстном меню выбрать пункт «Перезапустить» (рис. 14):

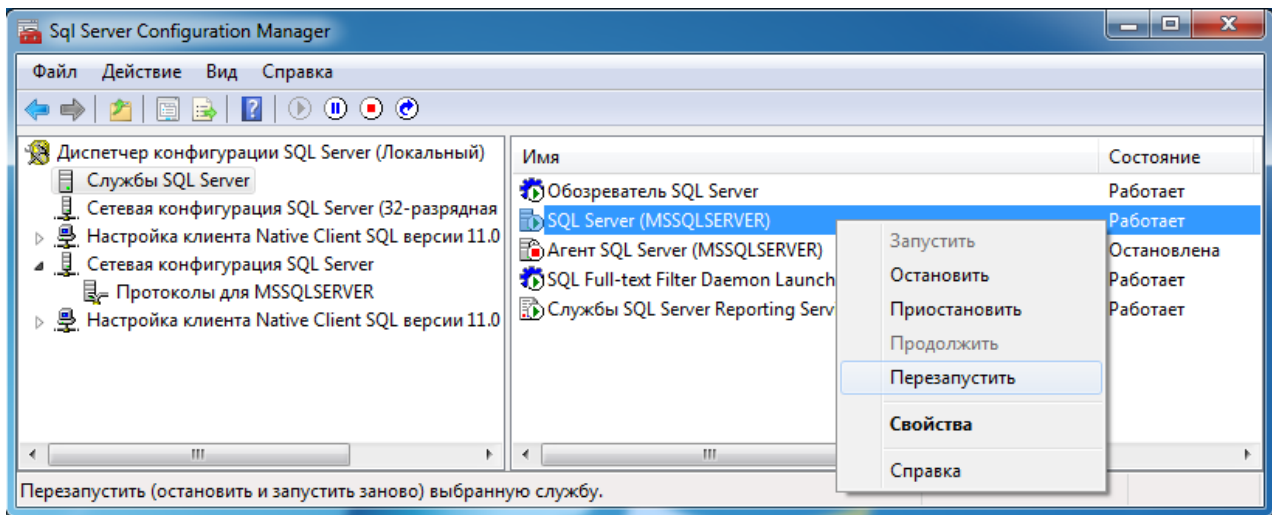


Рис.14. Службы SQL Server

Если на компьютере установлен какой-либо FireWall, то в нем необходимо разрешить входящие соединения по протоколу TCP/IP на порт 1433 (для подключения к SQL Server с других компьютеров), а также по протоколу UDP на порт 1434 (для того чтобы другие компьютеры могли обнаруживать данный экземпляр SQL Server). В стандартном «Брандмауэре Windows» (Панель управления→Брандмауэр Windows) для этого необходимо в разделе «Дополнительные параметры→Правила для входящих подключений» дважды «Создать правило»:

- Выбрать вариант «Для порта» и нажать кнопку «Далее» (рис. 15):

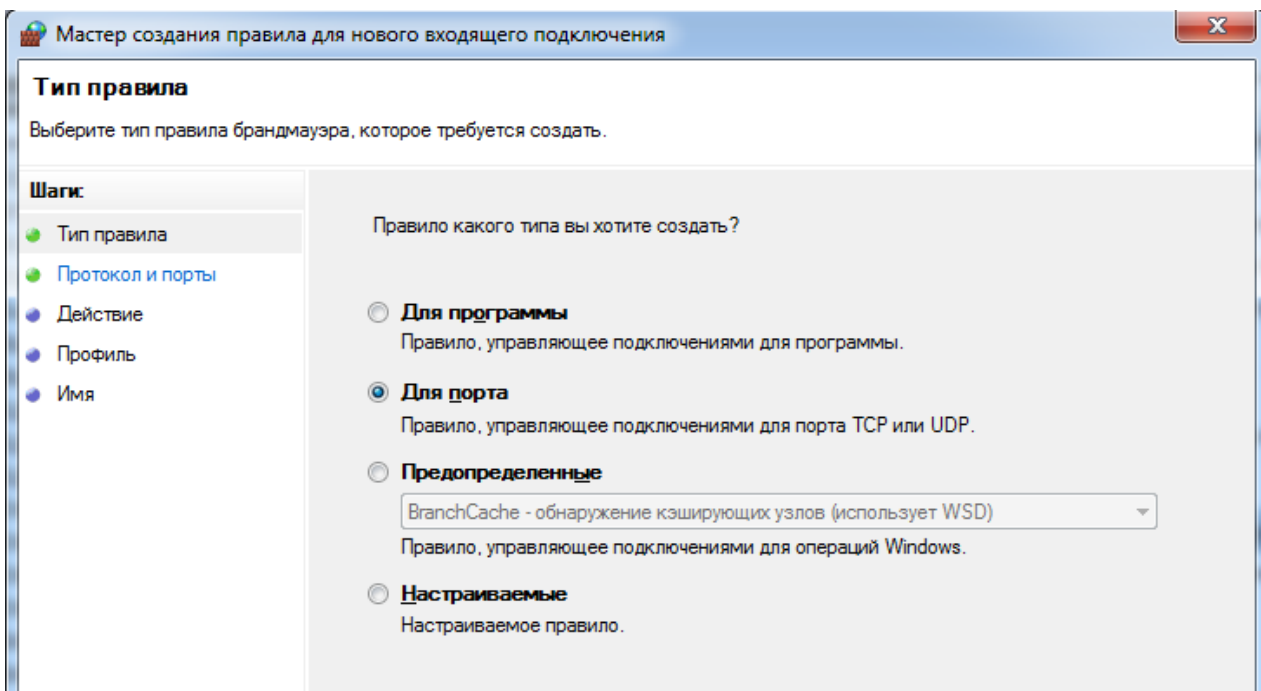


Рис.15. Создание правила

- Выбрать протокол TCP, ввести номер порта 1433 и нажать кнопку «Далее» (рис. 16):

Укажите протокол, к которому будет применяться это правило.

**Протокол TCP**

**Протокол UDP**

Укажите порты, к которым будет применяться это правило.

**Все локальные порты**

**Определенные локальные порты:**

Пример: 80, 443, 5000-5010

Рис. 16. Протокол TCP

- Выбрать «Разрешить подключение» и нажать кнопку «Далее» (рис. 17):

Укажите действие, которое должно выполняться, когда подключение удовлетворяет указанным условиям.

**Разрешить подключение**  
Включая как подключения, защищенные IPSec, так и подключения без защиты.

**Разрешить безопасное подключение**  
Включая только подключения с проверкой подлинности с помощью IPSec. Подключения будут защищены с помощью параметров IPSec и правил, заданных в разделе правил безопасности подключений.

**Блокировать подключение**

Рис. 16. Разрешение подключения

- Ввести любое имя и нажать кнопку «Готово».
- После завершения установки и настройки рекомендуется перезагрузить компьютер.

## 2. SQL SERVER MANAGEMENT STUDIO

При запуске SSMS появляется окно «Соединение с сервером» (рис. 18), через него подключаемся к ядру БД и получаем доступ к базе данных. Подключить можно удаленно или локально.

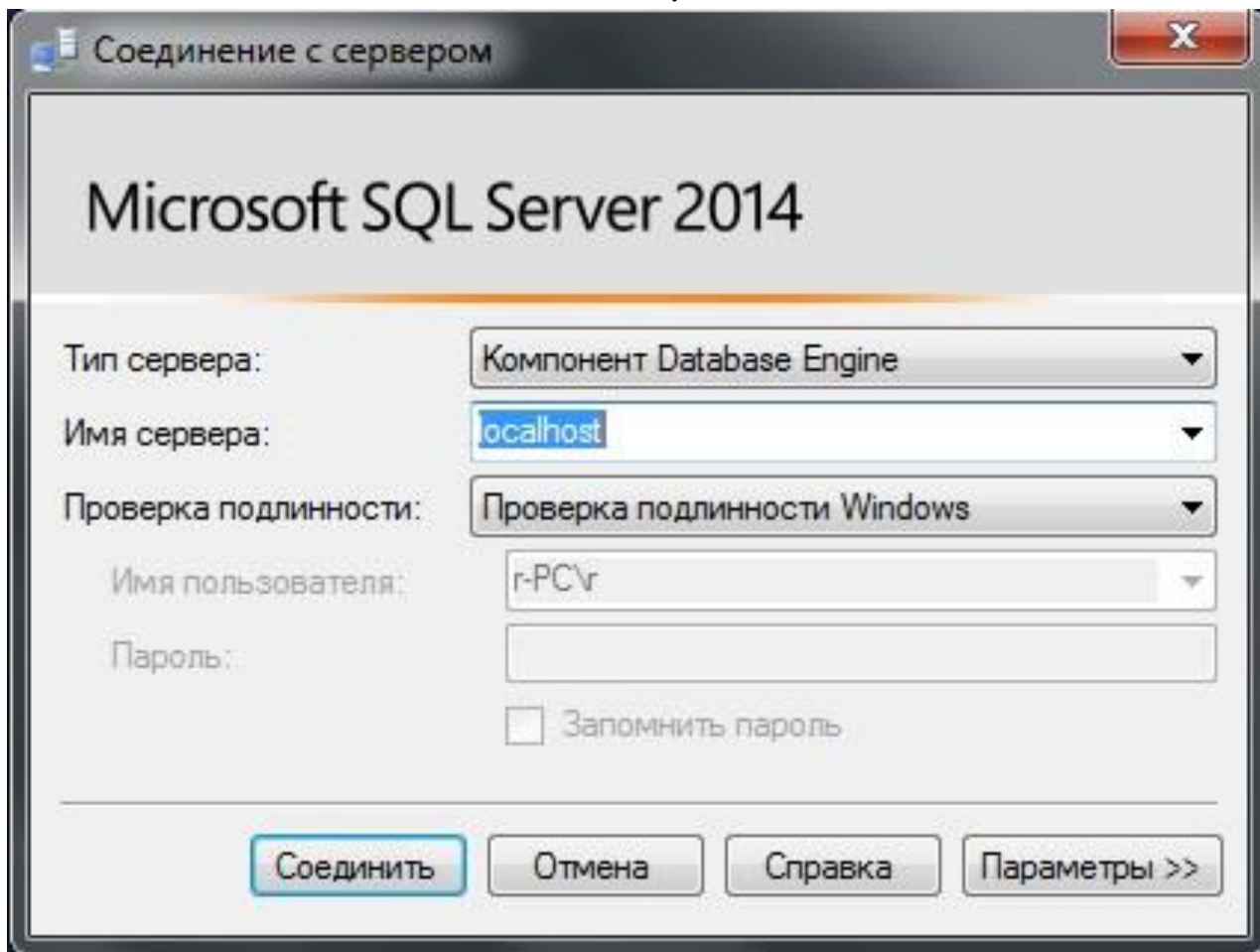


Рис.178. Соединение с сервером

Для работы с БД требуется сервер Database Engine (основная служба для хранения, обработки и защиты данных). Компонент Database Engine используется для создания реляционных баз данных для обработки транзакций в сети или интерактивной аналитической обработки. Сюда входит создание таблиц для хранения данных и объектов баз данных, таких как индексы, представления и хранимые процедуры для просмотра и защиты данных и для управления ими.

При соединении с экземпляром компонента Компонент Database Engine необходимо указать имя экземпляра SQL Server. Если экземпляра SQL Server является экземпляром по умолчанию (неименованным экземпляром), то укажите имя компьютера, на котором установлен SQL Server, или IP-адрес этого компьютера. Если экземпляр SQL Server является именованным (например, SQLEXPRESS), то укажите имя компьютера, на котором установлен

SQL Server, или IP-адрес этого компьютера и добавьте косую черту и имя экземпляра.

Пример приведен в таблице 4:

Таблица 4

Пример элементов для имени сервера

Тип экземпляра	Элемент для имени сервера
Соединение с экземпляром по умолчанию на том же компьютере при помощи localhost для указания, что экземпляр выполняется на локальном компьютере.	localhost
Соединение с именованным экземпляром на том же компьютере с помощью localhost, указывающее, что экземпляр выполняется на локальном компьютере.	localhost\SQLEXPRESS
Соединение с экземпляром по умолчанию, с прослушиванием TCP-адреса 192.168.17.28 с помощью IP-адреса.	192.168.17.28
Соединение с именованным экземпляром, с прослушиванием TCP-адреса 192.168.17.28 с помощью IP-адреса.	192.168.17.28\SQLEXPRESS
Соединение с экземпляром по умолчанию, который не прослушивает TCP-порт по умолчанию, с указанием используемого порта, в данном случае 2828. (Это не является обязательным, если компонент Компонент Database Engine прослушивает порт по умолчанию 1433.)	192.168.17.28,2828

SQL Server поддерживает два режима проверки подлинности: режим проверки подлинности Windows и режим смешанной проверки подлинности.

Режим проверки подлинности Windows является режимом по умолчанию. Поскольку эта модель безопасности SQL Server тесно интегрирована с Windows, часто ее называют встроенной функцией безопасности. Определенным учетным записям пользователей и групп Windows разрешается входить в SQL Server. Пользователи Windows, прошедшие проверку подлинности, не должны предъявлять дополнительные учетные данные.

Режим смешанной аутентификации поддерживает проверку подлинности как средствами Windows, так и средствами SQL Server. Пары имен пользователей и паролей ведутся в SQL Server.

Обычно проверка подлинности Windows является наилучшим вариантом в следующих ситуациях:

- Имеется контроллер домена.

- Приложение и база данных находятся на одном компьютере.
- Используется экземпляр SQL Server Express или LocalDB.

Имена входа SQL Server часто используются в следующих ситуациях.

- При наличии рабочей группы.
- Пользователи подключаются из разных, не доверенных доменов.
- Интернет-приложения, например ASP.NET (для сторонних клиентов).

В SQL Server существует три типа имен входа:

1. Учетная запись локального пользователя Windows или учетная запись доверенного домена. SQL Server доверяет проверку подлинности учетных записей пользователей Windows самой системе Windows.
2. Группа пользователей Windows. В случае предоставления доступа группе пользователей Windows право доступа получают все имена пользователей, входящие в эту группу.
3. Имя входа SQL Server. SQL Server хранит в базе данных master имя пользователя и хэш пароля путем использования внутренних методов проверки подлинности при попытке входа в базу данных.

В «Обозревателе объектов» (рис.19) будут показываться к каким серверам подключились, а так же их содержимое.

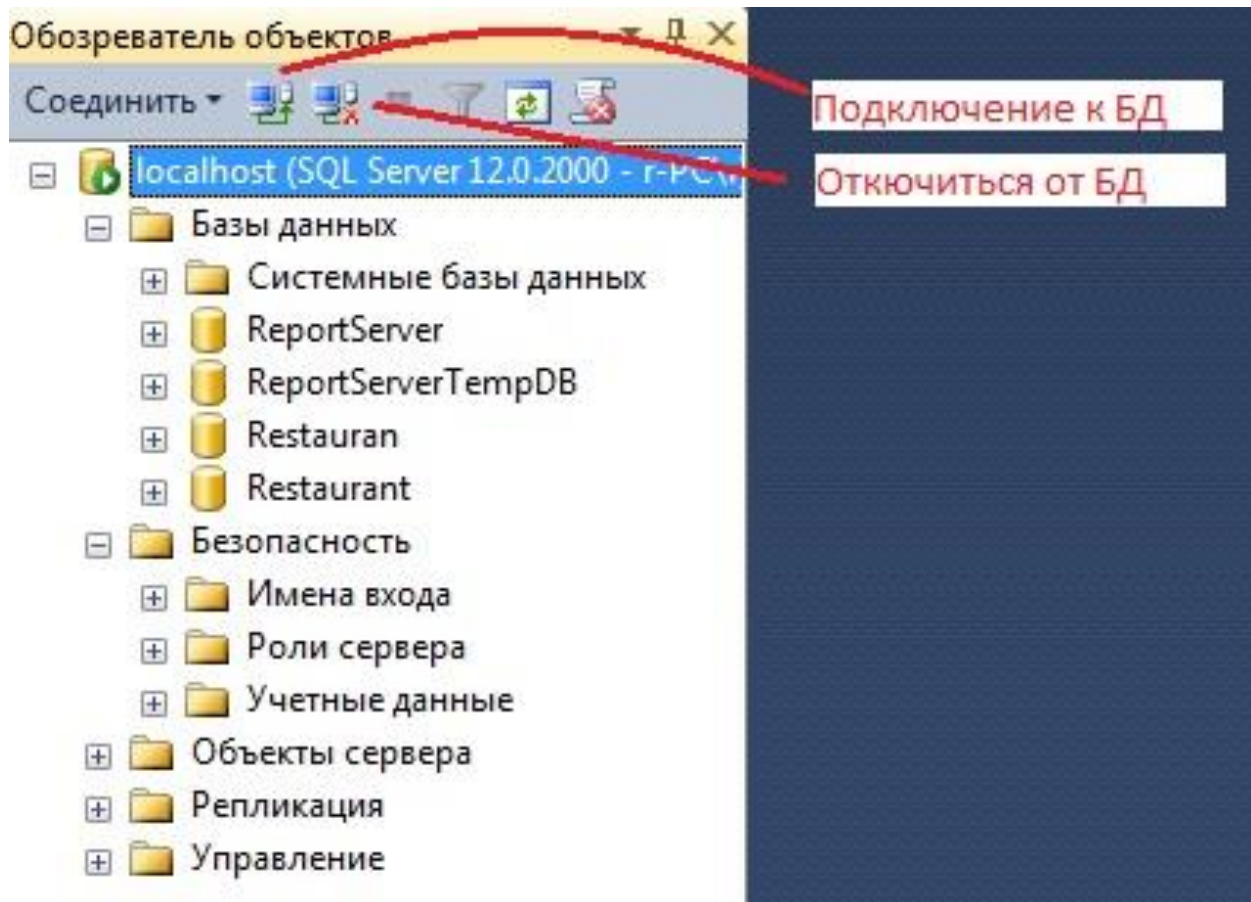


Рис.19. Обозреватель объектов

Для создания нового редактора скрипта используйте кнопку «Создать запрос» (рис.20). Для смены текущей базы данных можно использовать выпадающий список (рис 3 синий квадрат). Для выполнения определенной команды (или группы команд) выделите ее и нажмите кнопку «Выполнить» (рис.20) или же клавишу «F5». Если в редакторе в текущий момент находится только одна команда, или же вам необходимо выполнить все команды, то ничего выделять не нужно.

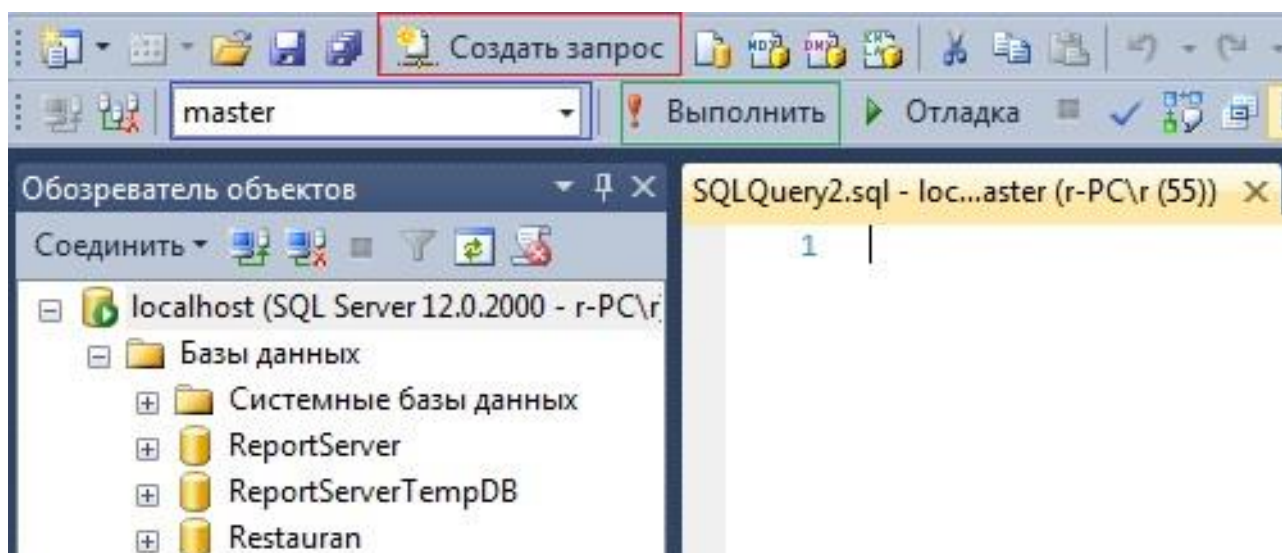


Рис. 20. Вкладка создания нового редактора скрипта

После выполнения скриптов, в особенности создающих объекты (таблицы, столбцы, индексы), чтобы увидеть изменения, используйте обновление из контекстного меню, выделив соответствующую группу (например, Таблицы), саму таблицу или группу «Столбцы» в ней.

### 3. ЯЗЫК ОПИСАНИЯ ДАННЫХ - DDL

Язык DDL служит для создания и модификации структуры БД, т.е. для создания/изменения/удаления таблиц и связей. К примеру, рассмотрим данные о студентах (таблица 5).

Таблица 5

Пример БД «Студенты»

ID	ФИО	ДР	Группа	Институт
101	Иванов И.И	15.10.1989	ЭНБ	Энергетика
102	Абрамов А.А	17.11.1995	АИБ	Автоматизация
103	Петров П.П	02.05.1996	КАиБ	Кибернетика
104	Андреев А.А	25.02.1995	ААБ	Автоматизация

Каждый из столбцов БД можно охарактеризовать по типу содержащихся в нем данных:

- ID (номер студента) – числовой;
- ФИО – строка;
- ДР (день рождения) – дата;
- Группа – строка;
- Институт – строка;

Основные типы данных языка DDL:

1. **varchar(N)** и **nvarchar(N)** - При помощи числа N, мы можем указать максимально возможную длину строки для соответствующего столбца. Например, если мы хотим сказать, что значение столбца «ФИО» может содержать максимум 30 символов, то необходимо задать ей тип **nvarchar(30)**. Отличие **varchar** от **nvarchar** заключается в том, что **varchar** позволяет хранить строки в формате ASCII, где один символ занимает 1 байт, а **nvarchar** хранит строки в формате Unicode, где каждый символ занимает 2 байта. Тип **varchar** стоит использовать только в том случае, если вы на 100% уверены, что в данном поле не потребуется хранить Unicode символы.

2. **int**- Данный тип позволяет нам использовать в столбце только целые числа, как положительные, так и отрицательные.

3. **float** -Если говорить простым языком, то это числа, в которых может присутствовать десятичная точка (запятая).

4. **date** - Если в столбце необходимо хранить только Дату, которая состоит из трех составляющих: Числа, Месяца и Года.

5. **time** - Данный тип можно использовать, если в столбце необходимо хранить только данные о времени, т.е. Часы, Минуты, Секунды и Миллисекунды.

6. **datetime** - Данный тип позволяет одновременно сохранить и Дату, и Время. Например, 15.02.2014 17:38:31.323.

### 3.1. Создание БД

Для создания базы данных можно использовать конструктор или SQL–запрос.

Что бы создать БД через конструктор в обозревателе нажимаем правой кнопкой мыши по папке «База данных» и выбираем «Создать БД» (рис.21).

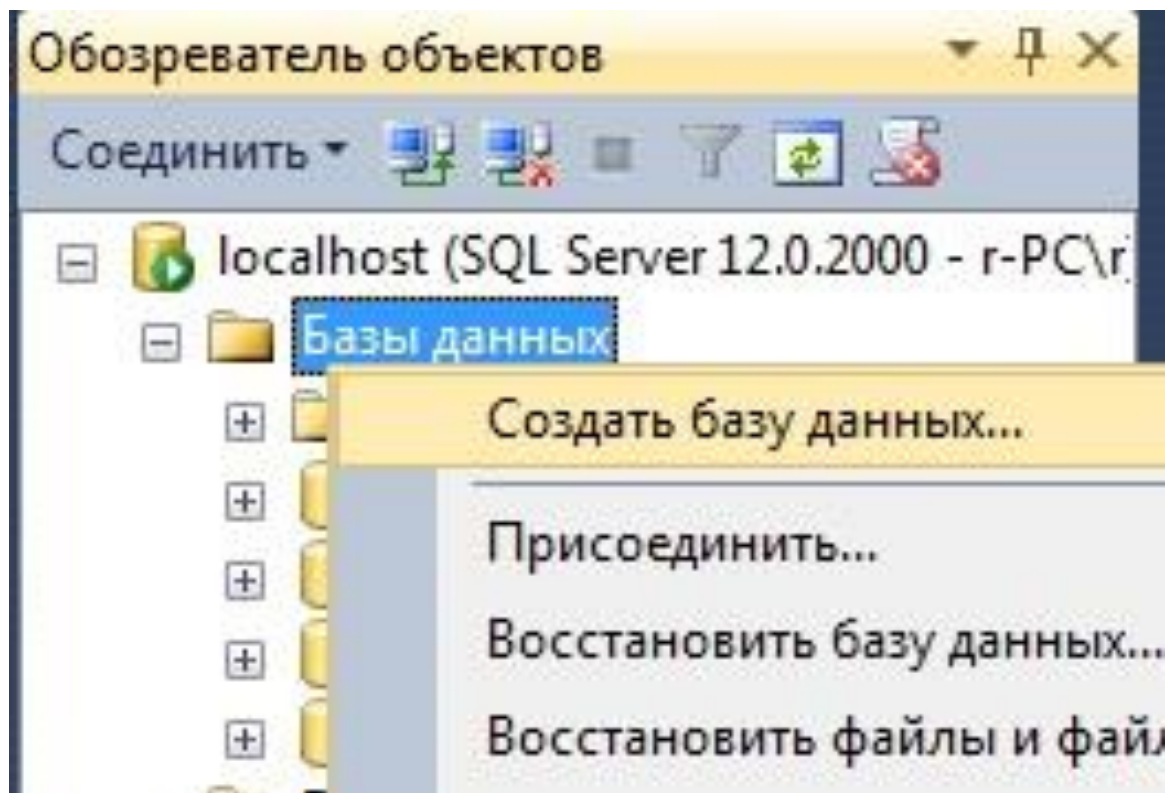


Рис. 21. Создание БД

В появившемся окне можно задать, кроме стандартных настроек, владельца, путь, правила, имена входа и т.д.

Для выполнения дальнейших примеров создадим простую тестовую базу данных без указания дополнительных параметров через запрос:

```
CREATEDATABASE Test
```

После выполнения запроса внизу появится сообщение (успешно, ошибка и т.д.) (рис.22):

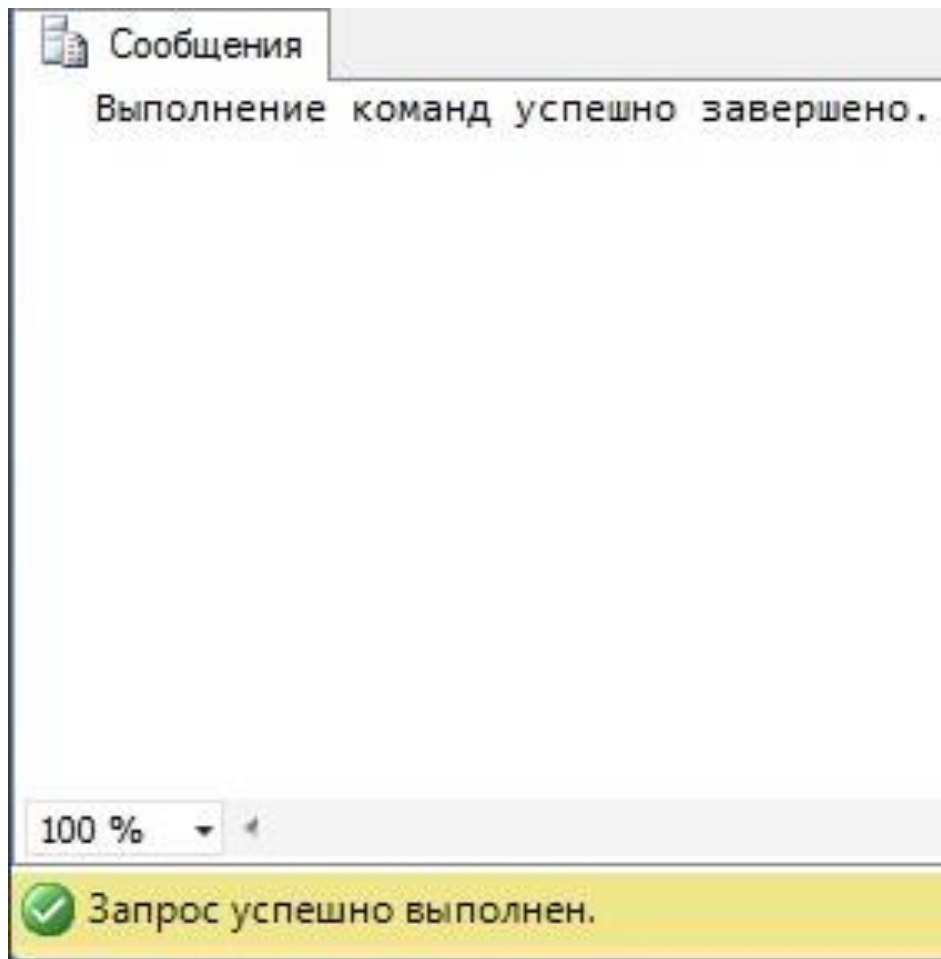


Рис.22. Сообщение по выполнению запроса на создание БД

Для удаление БД создается следующий запрос:

```
DROPDATABASE Test
```

### 3.2. Создание таблиц

Для переключения БД создается следующий запрос:

```
USE Test
```

Или можно выбрать базу данных Test в выпадающем списке в области меню SSMS.

Теперь в нашей БД можно создавать таблицу, используя описания в том виде как они есть, используя пробелы и символы кириллицы:

```
CREATETABLE [Студенты]  
(  
  ID int,  
  ФИО nvarchar(30),  
  ДР date,  
  Группа nvarchar(20),
```

Институт `nvarchar` (30)  
)

Если в имени есть пробелы то нужно использовать квадратные скобки, например, [ИД Студента].

**Примечание.** Для большего удобства в базе данных все наименования объектов лучше задавать на латинице и не использовать в именах пробелы.

В итоге получим таблицу (рис.23):

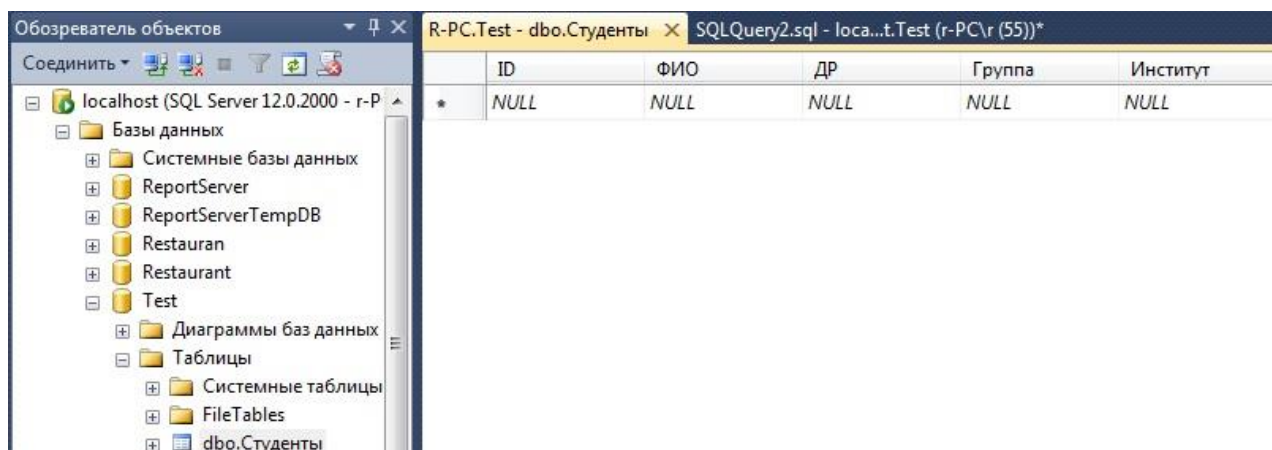


Рис. 23. Конструктор таблиц

Что бы посмотреть и редактировать содержимое таблицы через конструктор таблиц, необходимо в «Таблице» нажать на правую кнопку мыши и выбрать: «Изменить первые 200 строк».

Для создания таблицы в конструкторе через ПКМ необходимо выбрать «Таблица...». Там можно настроить строки или оставить по умолчанию.

Для того, чтобы задать обязательные для заполнения столбцы, можно использовать опцию NOT NULL, Primary Key, identity и т.д. Так же можно их сразу использовать при создании таблицы. Например, создадим таблицу продукты:

```
createtableПродукты  
(  
[IDПродукта] intidentity(1,1) primarykeynotnull,  
Названиенvarchar(50) notnull,  
Категорияnvarchar(30),  
Стоимостьrealnotnull,  
[Срок хранения] intnotnull  
)
```

Для удаления таблицы или строк используют команду DROP.

### 3.3. Команды ALTER и INSERT

Для изменения свойств, удаление строк или добавление используется команда ALTER. Для изменения в таблице «Студенты», добавим ограничитель NOTNULL для ID и ФИО:

```
ALTERTABLE [Студенты] ALTERCOLUMN ID int NOTNULL
ALTERTABLE [Студенты] ALTERCOLUMN ФИО nvarchar(30) NOTNULL
```

**Примечание.** После выполнения команды, при заполнении таблицы, эти две столбца не могут быть пустыми иначе получим сообщение об ошибке.

Если требуется наоборот сделать существующий столбец необязательным для заполнения, то используем следующий синтаксис команды:

```
ALTERTABLE [Студенты] ALTERCOLUMN ФИО nvarchar(30) NULL
```

Или вообще не используют ограничитель NULL.

Для заполнения таблицы используют команду INSERT:

```
INSERT [Студенты] (ID, ФИО, Группа, Институт) VALUES
(101, N'Иванов И.И', N'ЭНБ', N'Энергетика'),
(102, N'Абрамов А.А', N'АИБ', N'Автоматизация'),
(103, N'Петров П.П', N'КАИБ', N'Кибернетика'),
(104, N'Андреев А.А', N'ААБ', N'Автоматизация')
```

**Примечание.** Иногда в БД есть столбцы (обычно идентификаторы, первичные ключи и т.д.), которые требуют автоматического заполнения. В этом случае для них используют ограничитель IDENTITY.

Ограничитель IDENTITY указывает, что новый столбец является столбцом идентификаторов. При добавлении в таблицу новой строки компонент Database Engine формирует для этого столбца уникальное последовательное значение. Столбцы идентификаторов обычно используются с ограничением PRIMARY KEY для поддержания уникальности идентификаторов строк в таблице. Свойство IDENTITY присвоено столбцам типа tinyint, smallint, int, bigint, decimal(p,0) или numeric(p,0). Для каждой таблицы можно создать только один столбец идентификаторов. Необходимо указать как начальное значение, так и приращение, или же не указывать ничего. Если ничего не указано, применяется значение по умолчанию (1,1).

В таблице, оптимизированной для памяти, единственное допустимое значение для seed и increment - 1; значение по умолчанию для свойств seed и increment - (1,1). Seed - Значение, используемое для самой первой строки, загружаемой в таблицу. Increment - Значение приращения, добавляемое к значению идентификатора предыдущей загруженной строки.

Например, можно добавить ограничение для столбца ID:

```
CREATETABLE [Студенты]
```

```

(
  ID intidentity(100,1) not null,
  ФИО nvarchar(30) not null,
  ДР date,
  Группa nvarchar(20),
  Институт nvarchar(30)
)
-- То в случае заполнения нам бы не потребовалась
заполнять
-- столбец ID

INSERT [Студенты] (ФИО, Группa, Институт) VALUES
(N'Иванов И.И', N'ЭНБ', N'Энергетика'),

```

В итоге при заполнение ID автоматически будут формироваться значения, начиная от 100 с приростом +1 для следующих значений: 101, 102 и т.д.

### 3.4. Первичный ключ и внешний ключ

При создании таблицы желательно, чтобы она имела уникальный столбец или же совокупность столбцов, которая уникальна для каждой ее строки – по данному уникальному значению можно однозначно идентифицировать запись. Такое значение называется первичным ключом (PRIMARY KEY) таблицы. Для нашей таблицы Студенты таким уникальным значением может быть столбец ID:

```

ALTER TABLE [Студенты] ADD CONSTRAINT
PK_Студенты PRIMARY KEY (ID)
--Если первичный ключ состоит из нескольких полей, то эти
поля
--необходимо перечислить в скобках через запятую:
ALTER TABLE имя_таблицы ADD CONSTRAINT имя_ограничения PRIMARY KEY
(поле1, поле2, ...)

```

Где «PK\_Студенты» это имя ограничения, отвечающего за первичный ключ. Обычно для наименования первичного ключа используется префикс «PK\_», после которого идет имя таблицы.

Так же можно ПК создать во время создание ключа, используя просто PRIMARY KEY или через CONSTRAINT:

```

CREATE TABLE [Студенты]
(
  ID int primarykey not null,
  ФИО nvarchar(30),
  ДР date,

```

```
Группа nvarchar (20) ,  
Институт nvarchar (30)
```

```
)
```

Ограничение, которое обеспечивает ссылочную целостность данных в этом столбце или столбцах. Ограничения FOREIGN KEY требуют, чтобы каждое значение в столбце существовало в соответствующем связанном столбце или столбцах в связанной таблице. Ограничения FOREIGN KEY могут ссылаться только на столбцы, являющиеся ограничениями PRIMARY KEY или UNIQUE в связанной таблице или на столбцы, на которые имеются ссылки в индексе UNIQUE INDEX связанной таблицы. Внешние ключи в вычисляемых столбцах должны быть также помечены как PERSISTED.

Для примера создадим таблицу «Институты», она потребуется, что бы в таблице «Студенты» в столбце «Институт» пользователи не могли ввести любой текст. Так же оно уменьшит объем хранимой информации, исключит ее дублирование и упростит ее обновление полей, то есть, если потребуется наименование института, то нам не придется изменять каждое поле. Таким образом, мы избежим недостатки и нормализуем базу данных – дробление ее на подтаблицы, таблицы справочники.

Создадим таблицу «Институт»:

```
CREATETABLEИнституты  
(  
IDintIDENTITY (1, 1) NOTNULLCONSTRAINTПК_ИнститутыPRIMARYKEY  
,  
Наименование nvarchar (30) NOTNULL  
)  
-- Для удобства заполним его через инструкцию Insert  
--  
INSERT Институты (Наименование)  
SELECTDISTINCT Институт  
FROMСтуденты  
WHEREИнститут ISNOTNULL
```

В итоге получим (рис.24):

	ID	Наименование
▶	1	Автоматизация
	2	Кибернетика
	3	Энергетика

Рис. 24. Таблица Институты

Перед тем, как поставим ограничение FOREIGN KEY, очистим данные из столбца «Институт» и изменим тип столбца:

```
Altertable Студенты ALTER COLUMN Институт int
```

Добавим ссылочное ограничение FOREIGN KEY:

```
Altertable Студенты ADDCONSTRAINT FK_Студенты_Институты FOREIGN KEY (Институт) REFERENCES Институты (ID)
```

Если столбец не имеет пустое значение, выдаст ошибку, поэтому перед тем как ставить ограничение стоит очистить столбец.

Имя ссылочного ограничения, обычно является составным, оно состоит из префикса «FK\_», затем идет имя таблицы и после знака подчеркивания идет имя поля, которое ссылается на идентификатор таблицы-справочника.

Идентификатор (ID) обычно является внутренним значением, которое используется только для связей и какое значение там хранится, в большинстве случаев абсолютно безразлично, поэтому не нужно пытаться избавиться от дырок в последовательности чисел, которые возникают по ходу работы с таблицей, например, после удаления записей из таблицы. FK можно поставить так же при создании таблицы. Все это можно проделать через конструктор. Так же стоит отметить, что таблица может ссылаться сама на себя, т.е. можно создать рекурсивную ссылку.

### 3.5. Дополнительные опции

Для ссылочных ключей могут включать дополнительные опции ON DELETE и ON UPDATE, которые говорят о том, как вести себя при удалении или обновлении записи, на которую есть ссылки в других таблицах. Если эти опции не указаны, то мы не можем изменить ID в таблице у той записи, на которую есть ссылки из другой таблицы. Так же мы не сможем удалить такую запись из таблицы, пока не удалим все строки, ссылающиеся на эту запись или, же обновим в этих строках ссылки на другое значение.

ON DELETE { NO ACTION | CASCADE | SET NULL | SET DEFAULT }

Определяет операцию, которая производится над строками создаваемой таблицы, если эти строки имеют ссылочную связь, а строка, на которую имеются ссылки, удаляется из родительской таблицы. Параметр по умолчанию - NO ACTION.

NO ACTION - Компонент Database Engine формирует ошибку, и выполняется откат операции удаления строки из родительской таблицы.

CASCADE - Если из родительской таблицы удаляется строка, соответствующие ей строки удаляются и из ссылающейся таблицы.

SET NULL - Все значения, составляющие внешний ключ, при удалении соответствующей строки родительской таблицы устанавливаются в NULL. Для

выполнения этого ограничения внешние ключевые столбцы должны допускать значения NULL.

SET DEFAULT - Все значения, составляющие внешний ключ, при удалении соответствующей строки родительской таблицы устанавливаются в значение по умолчанию. Для выполнения этого ограничения все внешние ключевые столбцы должны иметь определения по умолчанию. Если столбец допускает значения NULL и значение по умолчанию явно не определено, значением столбца по умолчанию становится NULL.

Не следует использовать параметр CASCADE, если таблица будет включена в публикацию слиянием, в которой используются логические записи.

ON UPDATE { NO ACTION | CASCADE | SET NULL | SET DEFAULT }

Указывает, какое действие совершается над строками в изменяемой таблице, когда эти строки имеют ссылочную связь и строка родительской таблицы, на которую указывает ссылка, обновляется. Параметр по умолчанию — NO ACTION. Все параметры соответственно как ONDELETE.

Обычно использую параметр CASCADE, если, к примеру, мы удалим один из институтов в таблице «Институты», то в таблице «Студент» ячейка под этим институтом примет значение «NULL». Аналогично и при обновлении. Рекомендую хорошо подумать, прежде чем указывать их в ссылочном ограничении, т.к. при нечаянном удалении записи из таблицы это может привести к большим проблемам и создать цепную реакцию.

Пример:

```
Altertable Студенты ADDCONSTRAINTFK_Студенты_Институты
FOREIGNKEY (Институт) REFERENCESИнституты (ID)
ondeleteCascadeonupdatecascade;
```

Теперь рассмотрим диаграмму того как будут выглядеть наши таблицы и как они связаны. Для этого в БД находим «Диаграмма база данных», нажимаем ПКМ, выбираем «Создать диаграмму БД». В появившемся окне добавляем таблицы (рис.25):

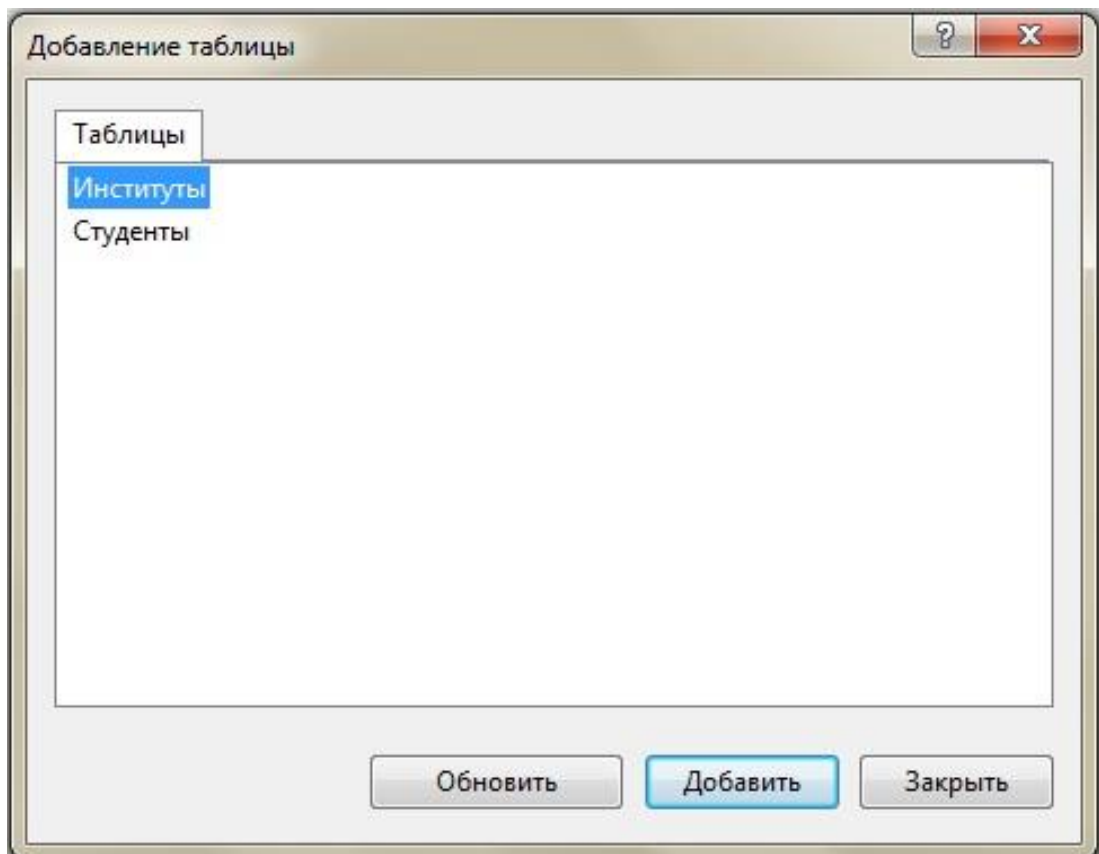


Рис. 25. Создание диаграммы  
Получим диаграмму (рис.26):

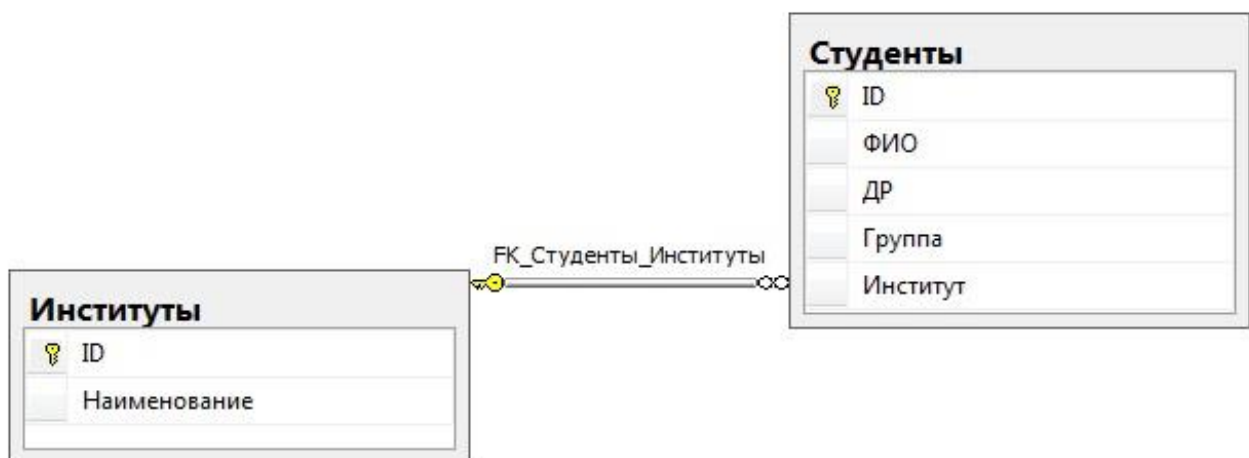


Рис.26. Диаграмма БД

### 3.6. Ограничения UNIQUE, DEFAULT, CHECK

Данные ограничения не обязательно используются при работе с БД, их использование называют ситуативным. Ограничение UNIQUE обозначает уникальность полей, на которые оно накладывается, т.е. значения для каждой строки в данном поле или в наборе полей должно быть уникальным.

Пример. Если мы добавим столбец «Паспорт» в таблице «Студенты» и наложим на него ограничения, то пользователь не сможет добавить одинаковые паспортные данные на несколько студентов.

Ограничение уникальности обычно именуется следующим образом – сначала идет префикс «UQ\_», далее название таблицы и после знака подчеркивания идет имя поля, на которое накладывается данное ограничение:

```
ALTERTABLE Студенты ADDCONSTRAINT UQ_Студенты_Паспорт  
UNIQUE (Паспорт)
```

При помощи добавления к полю ограничения DEFAULT можно задавать значение по умолчанию, которое будет подставляться в случае, если при вставке новой записи данное поле не будет перечислено в списке полей команды INSERT.

Проверочное ограничение CHECK используется в том случае, когда необходимо осуществить проверку вставляемых в поле значений.

Пример. Добавим таблицу «Курс» и у нас она ограничена от 1 по 5 курс. Ограничение обычно именуется так же, сначала идет префикс «СК\_», затем имя таблицы и имя поля, на которое наложено это ограничение:

```
ALTERTABLE Студенты ADDCONSTRAINT  
СК_Студенты_Курс CHECK (Курс BETWEEN 1 AND 5)
```

Если при добавлении ввести значение 6, то выведется сообщение об ошибке.

## **3.7. Имена входа и пользователи**

### ***3.7.1. Имена входа***

Рассмотрим создание имен входа в SQL Server или Базу данных SQL с помощью среды SQL Server Management Studio или Transact-SQL. Имя входа представляет собой идентификатор пользователя или процесса, выполняющего соединение с экземпляром SQL Server. С помощью него происходит подключение к БД для дальнейшей работы или для подключения других пользователей к БД.

Имя входа - это субъект безопасности, с помощью которого система безопасности может проверить подлинность лица или сущности. Имя входа необходимо пользователю для соединения с SQL Server. Можно создавать имя входа на основе участника Windows (например, пользователя или группы домена Windows), либо на основе пользователя, не являющегося участником Windows (например, имени входа SQL Server).

Как субъекты безопасности, разрешения могут быть предоставлены именам входов. Областью имени входа является весь компонент Компонент

Database Engine. Для подключения к определенной базе данных на экземпляре SQL Server имя входа должно быть сопоставлено с пользователем базы данных, о них поговорим позже.

Примечание. Разрешения внутри базы данных предоставляются и запрещаются для **пользователя** базы данных, **а не имени входа**.

Для этого SQL Server требуется наличие разрешения ALTER ANY LOGIN или ALTER LOGIN на сервере. Базе данных SQL требуется членство в роли login manager.

Прежде чем приступать к созданию имени входа на SQL сервер необходимо определиться с методом аутентификации. Существует два варианта:

1. Проверка подлинности Windows – это когда имя входа может идентифицировать пользователя как учетную запись Windows или как члена группы Windows (в том числе и доменные учетные записи, и группы);

2. Проверка подлинности SQL Server. В данном случае имя входа существует только в SQL Server.

Для создания имени входа через конструктор, нужно в обозревателе объектов найти пункт «Безопасность», раскрыть его, кликаем правой кнопкой мыши по пункту «Имена входа» и выбираем пункт «Создать имя входа» (рис 27.).

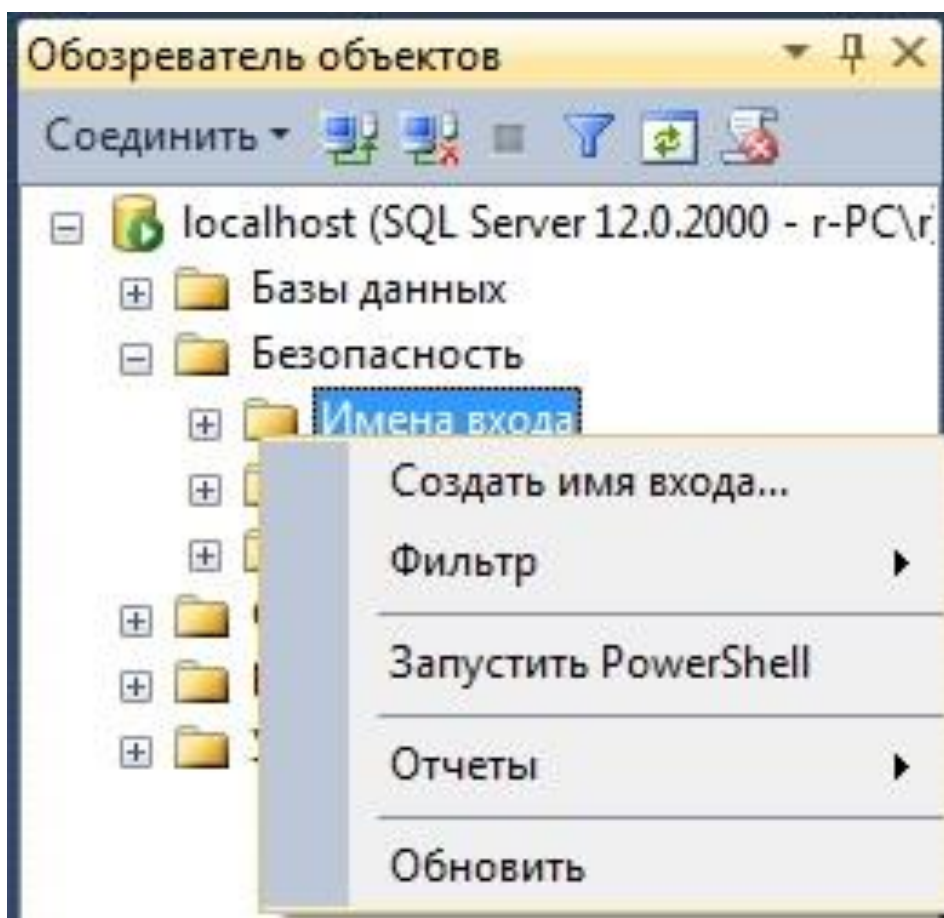


Рис.27. Окно обозревателя

В диалоговом окне «Имя входа - создание» на странице **Общие** введите имя пользователя в поле **Имя входа** пользователя. Либо нажмите кнопку «Поиск...», чтобы открыть диалоговое окно «Выбор пользователя или группы» (рис 28).

Так же можно установить политику паролей (только для смешенного), где «БД по умолчанию» выбираем нашу БД (наш текущий БД «Test»), а язык русский.

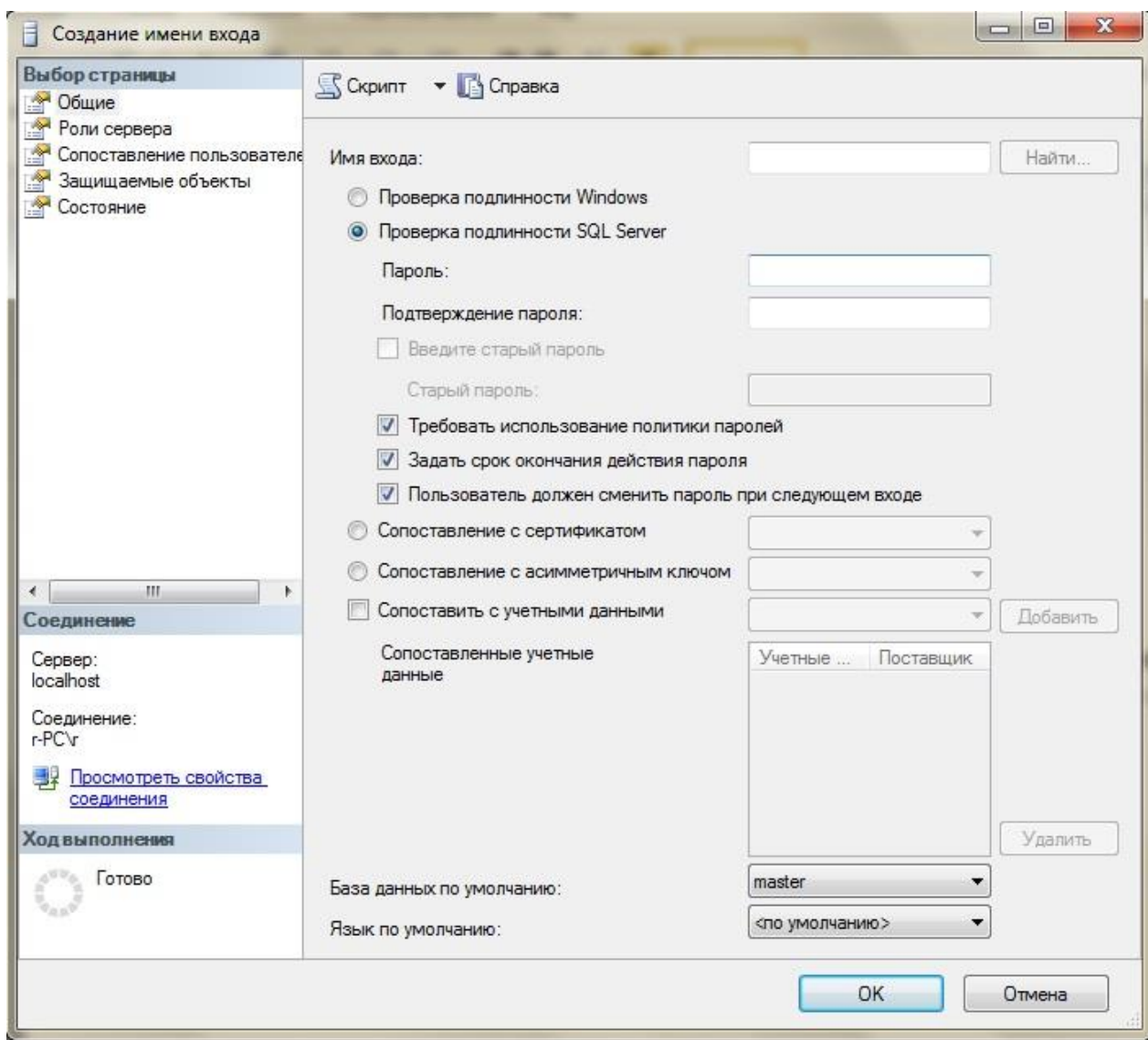


Рис.28. Создание имени входа

Диалоговое окно **Создание имени входа** также предлагает варианты на четырех дополнительных страницах: **Роли сервера**, **Сопоставление пользователей**, **Защищаемые объекты** и **Состояние**. Рассмотрим коротко, для чего они нужны:

На странице **Роли сервера** перечислены все возможные роли, которые могут быть назначены с новым именем входа. Доступны следующие параметры:

**bulkadmin** - Элементы предопределенной роли сервера выполнять инструкцию BULK INSERT.

**dbcreator** - Члены предопределенной роли сервера могут создавать, изменять, удалять и восстанавливать любые базы данных.

**diskadmin** - Элементы предопределенной роли сервера могут управлять дисковыми файлами.

**processadmin** - Элементы предопределенной роли сервера компонент DatabaseEngine могут завершать процессы, запущенные в экземпляре компонента.

**public** - Все пользователи, группы и роли SQL Server по умолчанию принадлежат предопределенной роли сервера.

**securityadmin** - Элементы предопределенной роли сервера управляют именами входа и их свойствами. Они могут предоставлять, запрещать и отменять разрешения на уровне сервера (инструкции GRANT, DENY и REVOKE). Они также могут предоставлять, запрещать и отменять разрешения на уровне базы данных (инструкции GRANT, DENY и REVOKE). Кроме того, они могут сбрасывать пароли для имен входа SQL Server.

**Serveradmin** - Элементы предопределенной роли сервера могут изменять параметры конфигурации на уровне сервера, а также выключать сервер.

**setupadmin** - Элементы предопределенной роли сервера могут добавлять и удалять связанные серверы, а также выполнять некоторые системные хранимые процедуры.

**sysadmin** - Элементы предопределенной роли сервера могут выполнять любые действия в компоненте Компонент Database Engine.

По сути я пользовался только **sysadmin** и **public**, и этого мне хватало, особенно при удаленном работе.

На странице **Сопоставление пользователей** приведен список всех возможных баз данных и ролей баз данных, которые могут применяться в момент входа. Выбранные базы данных определяют членство в ролях, которые доступны для этого имени входа. На этой странице доступны следующие параметры.

На странице **Защищаемые объекты** перечислены все возможные защищаемые объекты и разрешения на эти объекты, которые могут быть предоставлены для имени входа. Этой вкладкой я не пользовался, так как мне оно не нужно было, если хотите подробно рассмотреть ссылка на [MSDN](#).

На странице **Состояние** приведен список некоторых параметров проверки подлинности и авторизации, которые можно настроить для выбранного имени входа SQL Server . Так же можно пользователя заблокировать.

Теперь создаем имя входа через запрос:

```
CREATELOGINTest_login
```

```
WITH PASSWORD=N'Qwerty123',  
DEFAULT_DATABASE=[Test],
```

```
DEFAULT_LANGUAGE=[русский],  
CHECK_EXPIRATION=OFF,  
CHECK_POLICY=ON
```

Test\_login – это имя входа, параметр WITHPASSWORD – требуется только для смешенного режима. CHECK\_EXPIRATION - Применяется только к именам входа SQL Server . Указывает, должна ли политика истечения срока действия паролей принудительно применяться к этому имени входа. Значение по умолчанию - OFF.

CHECK\_POLICY - Применяется только к именам входа SQL Server . Указывает, что политики паролей Windows компьютера, на котором работает SQL Server , должны принудительно применяться к этому имени входа. Значение по умолчанию - ON.

Для того чтобы создать имя входа с проверкой подлинности Windows выполните следующую SQL инструкцию:

```
CREATELOGIN [ComputerName\NameUser]  
FROM WINDOWS  
WITH DEFAULT_DATABASE=[Test],  
DEFAULT_LANGUAGE=[русский];
```

Для удаление использую инструкцию DROPLOGIN.

Что бы присоединить имя входа к роли используем процедуру **sp\_addsrvrolemember**

Пример:

```
EXECsp_addsrvrolemember'Test_login', 'sysadmin';
```

Вводим «Имя входа», потом какой роль хотим дать. Для удаления используем инструкцию sp\_dropsvrolemember, она аналогична предыдущей инструкции. Все роли были описаны выше.

Или использовать инструкцию ALTER SERVER ROLE.

Пример:

```
ALTER SERVER ROLEsysadminADD MEMBER Test_login;
```

Для включения и выключения имя входа:

```
ALTERLOGIN Test_login DISABLE/ENABLE
```

### ***3.7.2. Создание пользователей***

После того как имя входа создано, можно переходить к созданию пользователя базы данных, т.е. сопоставлению пользователя с именем входа.

Пользователь базы данных - это идентификатор имени входа при подключении к базе данных. Имя пользователя базы данных может совпадать с именем входа, но это не является обязательным требованием. Пользователь является субъектом безопасности уровня базы данных. Для соединения с базой данных имя входа должно быть сопоставлено с пользователем базы данных. Имя входа может быть сопоставлено с различными базами данных в качестве разных пользователей, но в каждой базе данных ему может быть сопоставлен только один пользователь. В частично автономной базе данных можно создать пользователя, не имеющего имени входа.

В обозревателе объектов находим нужную базу данных («Test») и открываем ее. Затем открываем пункт «Безопасность» и кликаем по папке «Пользователи» правой кнопкой мыши и выбираем пункт «Создать пользователя» (рис. 29).

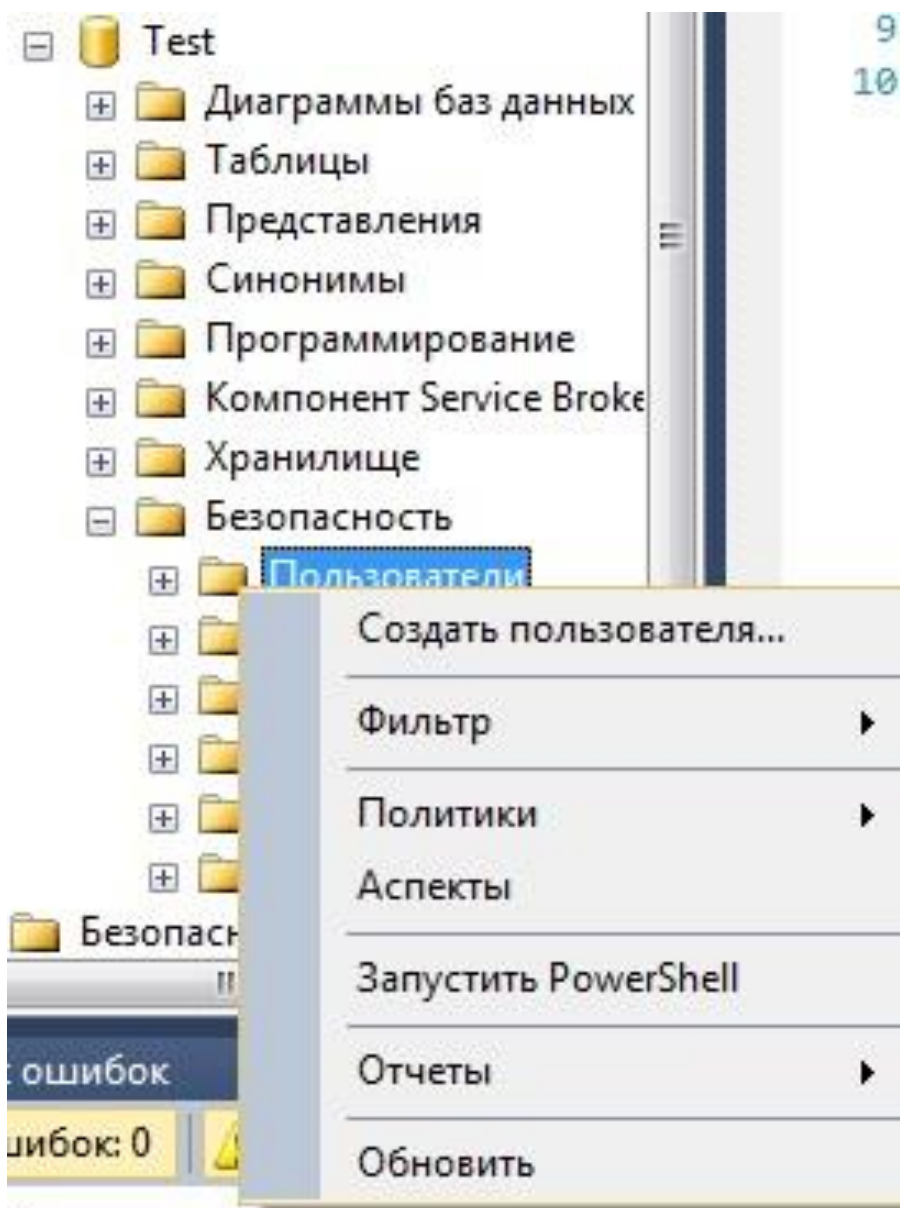


Рис.29. Создание пользователя

Далее вводим имя пользователя и имя входа, которое соответствует данному пользователю, а также указываем схему по умолчанию (если не указывать, то присвоится схема dbo) (рис. 30).

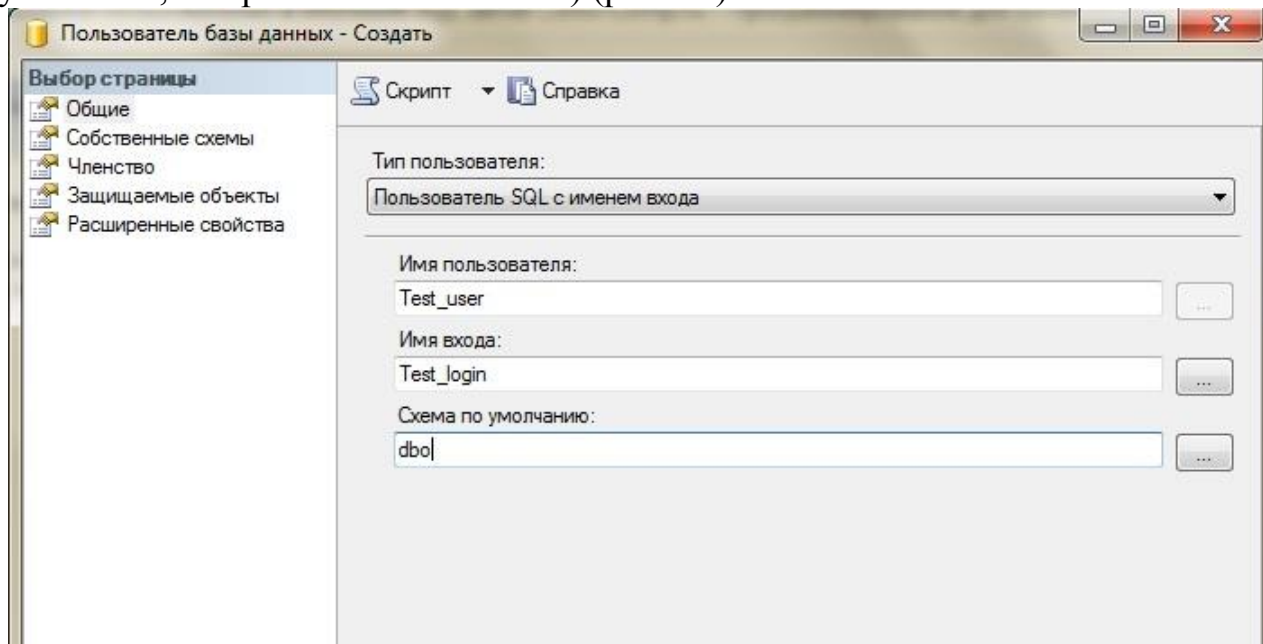


Рис.30. Диалоговое окно создания пользователя

Когда вы выбираете вариант типа пользователя, остальные параметры в диалоговом окне могут поменяться. Некоторые варианты применимы только к определенным типам пользователей базы данных. Некоторые параметры можно не заполнять. В результате будут использоваться значения по умолчанию.

Диалоговое окно также предлагает варианты на четырех дополнительных страницах: **Собственные схемы**, **Членство**, **Защищаемые объекты** и **Расширенные свойства**.

На странице **Собственные схемы** перечислены все возможные схемы, которые принадлежат новому пользователю базы данных.

На странице **Членство** приведен список всех ролей, к которым может принадлежать новый пользователь базы данных. Рассмотрим уровни роля БД:

- Члены предопределенной роли базы данных **db\_owner** могут выполнять все действия по настройке и обслуживанию базы данных, а также удалять базу данных в SQL Server.
- Элементы предопределенной роли базы данных **db\_securityadmin** могут изменять членство в роли и управлять разрешениями. Добавление участников к этой роли может привести к непреднамеренному повышению прав доступа.
- Члены предопределенной роли базы данных **db\_accessadmin** могут добавлять или удалять права удаленного доступа к базе данных для имен входа и групп Windows, а также имен входа SQL Server .

- Члены предопределенной роли базы данных **db\_backupoperator** могут создавать резервные копии базы данных.
- Члены предопределенной роли базы данных **db\_ddladmin** могут выполнять любые команды языка определения данных (DDL) в базе данных.
- Члены предопределенной роли базы данных **db\_datawriter** могут добавлять, удалять или изменять данные во всех пользовательских таблицах.
- Элементы предопределенной роли базы данных **db\_datareader** могут считывать все данные из всех пользовательских таблиц.
- Члены предопределенной роли базы данных **db\_denydatawriter** не могут добавлять, изменять или удалять данные в пользовательских таблицах базы данных.
- Члены предопределенной роли базы данных **db\_denydatareader** не могут считывать данные из пользовательских таблиц базы данных.

На странице Защищаемые объекты перечислены все возможные защищаемые объекты и разрешения на эти объекты, которые могут быть предоставлены для имени входа.

Страница Расширенные свойства позволяет добавлять пользовательские свойства пользователям базы данных. На этой странице доступны следующие параметры.

Теперь создадим через запросы, используем инструкцию CREATE USER:

```
CREATEUSER[Test_user]FOR LOGIN[Test_login]WITH
DEFAULT_SCHEMA=[dbo]
```

Теперь добавим членство, как и в предыдущем можно использовать 2 метода:

```
ALTERROLE[db_datareader]ADD MEMBER [Test_user];
---libo
EXECsp_addrolemember'db_datareader', 'Test_user';
```

Для удаления пользователя или роли используют инструкцию DROPвместоCREATEиALTER.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Выпуск SQL Server 2017 Express [Электронный ресурс] – Режим доступа: <https://www.microsoft.com/ru-ru/sql-server/sql-server-editions-express> свободный. – Загл. с экрана. – Яз. рус.

2. Михайлова У.В. Защита баз данных ORACLE / У.В. Михайлова, Т.Н. Носова, Т.В. Быкова, Р.Р. Булатов // Актуальные проблемы современной науки, техники и образования: материалы 73-й междунар. науч.-техн. конф. – Магнитогорск: изд-во Магнитогорск. гос. техн. ун-та им. Г.И. Носова. - 2015. – Т.2. – С. 158-162.

2. Баранкова И.И. Особенности формирования оценочных средств для оценки уровня сформированности компетенций специалиста по информационной безопасности / И.И. Баранкова, У.В. Михайлова // Информационное противодействие угрозам терроризма. - 2015. - Т. 2. № 25. - С. 26-30.

3. Ranga Rengarajan T.K. SQL Server 2016 publicpreviewcomingthissummer [Электронный ресурс] – Режим доступа: <https://blogs.technet.microsoft.com/dataplatforminsider/2015/05/04/sql-server-2016-public-preview-coming-this-summer/> свободный. – Загл. с экрана. – Яз. англ.

## ПРИЛОЖЕНИЕ. ФУНКЦИИ И ВОЗМОЖНОСТИ SQL SERVER EXPRESS 2017

SQL Server Express 2017 - бесплатный выпуск SQL Server, который идеально подходит разработчикам приложений для использования на настольных компьютерах, веб-серверах и других небольших серверах. Данный выпуск позволяет создавать небольшие, управляемые данными мобильные и веб-приложения размером до 10 ГБ.

Преимущества выпуска SQL Server Express:

1. Возможность создания мобильных и веб-приложений для различных типов исходных данных. Реализованы возможности хранения и поддержки бизнес-данных - структурированных или неструктурированных и с собственной поддержкой реляционных данных, XML и пространственных данных - плюс, увеличение детализации временных данных с типами данных даты и времени.

2. Возможность встраивания в готовые приложения упрощенной базы данных. Реализованы возможности создания небольших баз данных в базовых классических приложениях или средствах с помощью SQL Server Express LocalDB- упрощенного варианта развертывания, при котором используется меньше обязательных требований и процесс выполняется с использованием приложений, а не средств, предоставляемых как услуга.

3. Возможность масштабирования по выпускам. Реализованы возможности масштабирования своих приложений по выпускам SQL Server по мере вашего роста - без необходимости изменять код приложений благодаря поверхности общего программирования, предоставляемой SQL Server 2017 с пакетом обновления.

4. Усиленная безопасность данных. Применение функции Always Encrypted реализует дополнительный уровень защиты к безопасности SQL Server Express, за счет реализации возможности защиты данных в состоянии покоя (хранения), при их использовании и обработке запросов. Данная функция реализована на основе алгоритмов шифрования. Иерархия шифрования, реализованная в SQL Server, приведена на рис.31.Повышенная безопасность: шифрование применяется всегда, безопасность на уровне строк, маскирование данных, детальный аудит.

5. Доступные бесплатные пакеты обновлений. А именно базовый экспресс-выпуск, включающий функциональный модуль базы данных, который идеально подходит для принятия удаленных подключений или удаленного администрирования.

6. Наличие бесплатных дополнительных сервисов. А именно дополнительные возможности, имеющиеся в экспресс-выпуске, включая модуль базы данных, сервисы отчетности и полнотекстовый поиск. Доступен бесплатно при присоединении к программе Visual Studios Dev Essentials.

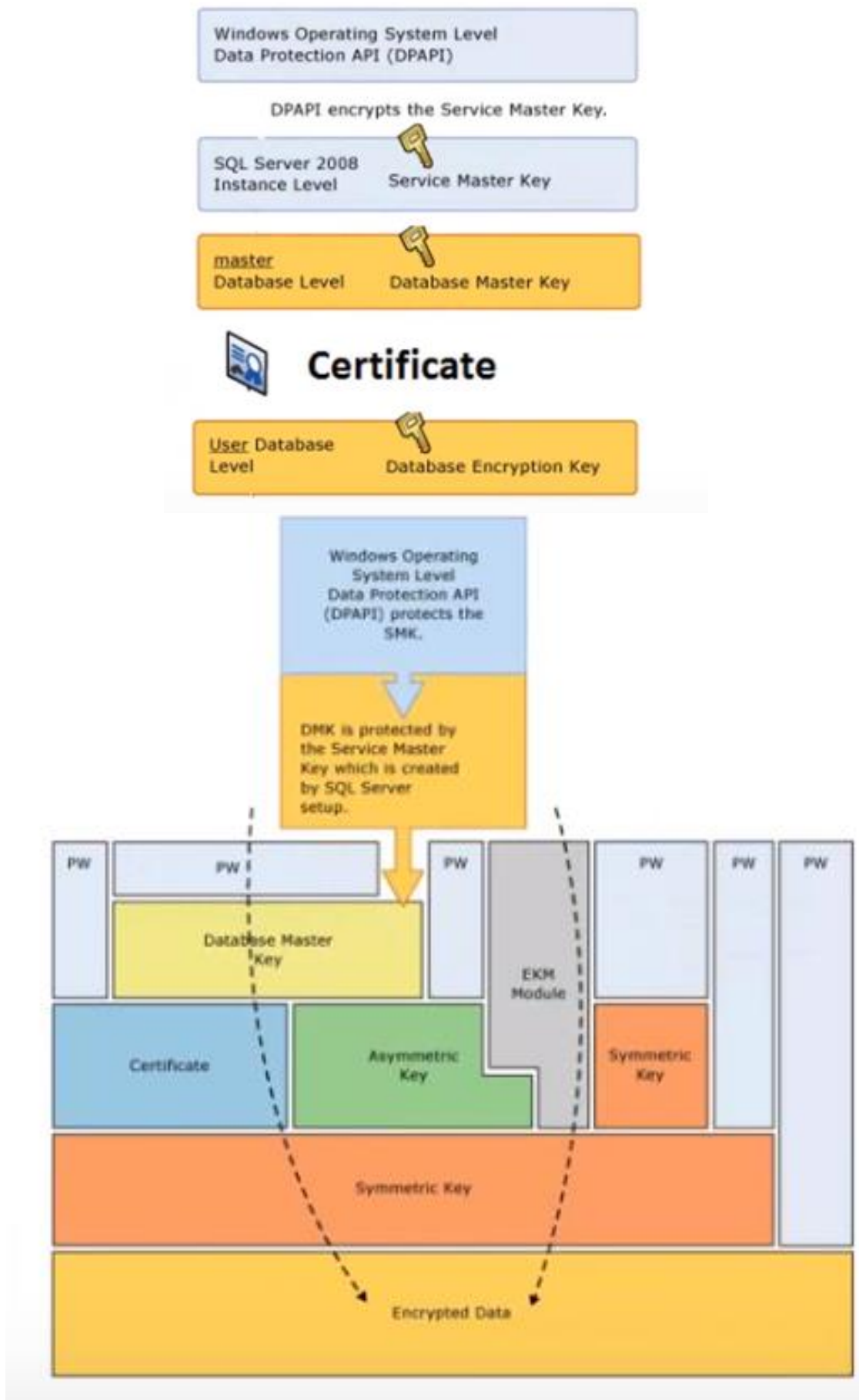


Рис. 31. Иерархия шифрования, реализованная в SQL Server

Рассмотрим сравнительный анализ выпуска SQL Server Express с другими выпусками SQL Server по уровню доступности реляционной системы управления базами данных (РСУБД), который приведен в таблице 4.

Таблица 6

Сравнение доступности РСУБД SQL Server в зависимости от выпуска

Компонент	Enterprise	Standard Edition	Web Edition	Express с доп. службами	Express
Поддержка Server Core	Есть	Есть	Есть	Есть	Есть
Доставка журналов;	Есть	Есть	Есть	Нет	Нет
Зеркальное отображение базы данных	Есть	Есть Только полная безопасность	Только следящий сервер	Только следящий сервер	Только следящий сервер
Сжатие резервных копий	Есть	Есть	Нет	Нет	Нет
Моментальный снимок базы данных	Есть	Есть	Есть	Есть	Есть
Экземпляры отказоустойчивого кластера AlwaysOn <sup>1</sup>	Есть	Есть	Нет	Нет	Нет
Группы доступности AlwaysOn <sup>2</sup>	Есть	Нет	Нет	Нет	Нет

Продолжение табл. 6

Компонент	Enterprise	StandardEdition	WebEdition	Express с доп. службами	Express
Базовые группы доступности <sup>3</sup>	Нет	Есть	Нет	Нет	Нет
Восстановление страниц и файлов в режиме «в сети»	Есть	Нет	Нет	Нет	Нет
Индексирование в сети	Есть	Нет	Нет	Нет	Нет
Возобновляемая перестройка индексов в подключенном режиме	Есть	Нет	Нет	Нет	Нет
Изменение схемы в режиме «в сети»	Есть	Нет	Нет	Нет	Нет
Быстрое восстановление	Есть	Нет	Нет	Нет	Нет
Зеркальные резервные копии	Есть	Нет	Нет	Нет	Нет

Окончание табл. 6

Компонент	Enterprise	Standard Edition	Web Edition	Express с доп. службами	Express
Поддержка памяти и ЦП с «горячей» заменой	Есть	Нет	Нет	Нет	Нет
Помощник по восстановлению базы данных	Есть	Есть	Есть	Есть	Есть
Зашифрованная резервная копия	Есть	Есть	Нет	Нет	Нет
Гибридное резервное копирование в Microsoft Azure (резервное копирование на URL-адрес)	Есть	Есть	Нет	Нет	Нет
Группа доступности без кластеров	Есть	Есть	Нет	Нет	Нет
Группа доступности с минимальным числом реплик для фиксации	Есть	Есть	Есть	Нет	Нет

\*1 В выпуске Enterprise количество узлов равно максимуму, поддерживаемому операционной системой. В выпуске Standard поддерживается два узла.

\*2 В выпуске Enterprise поддерживается до 8 вторичных реплик, включая 2 синхронные вторичные реплики.

\*3 В выпуске Standard поддерживаются базовые группы доступности. Базовая группа доступности поддерживает две реплики с одной базой данных.

### **Установка SQL Server 2017 Express и Management Studio.**

Для установки выпуска SQL Server 2017 Express сначала необходимо скачать SQL Server 2017 Express. Сделать это можно с официального сайта компании Microsoft (рис. 32).

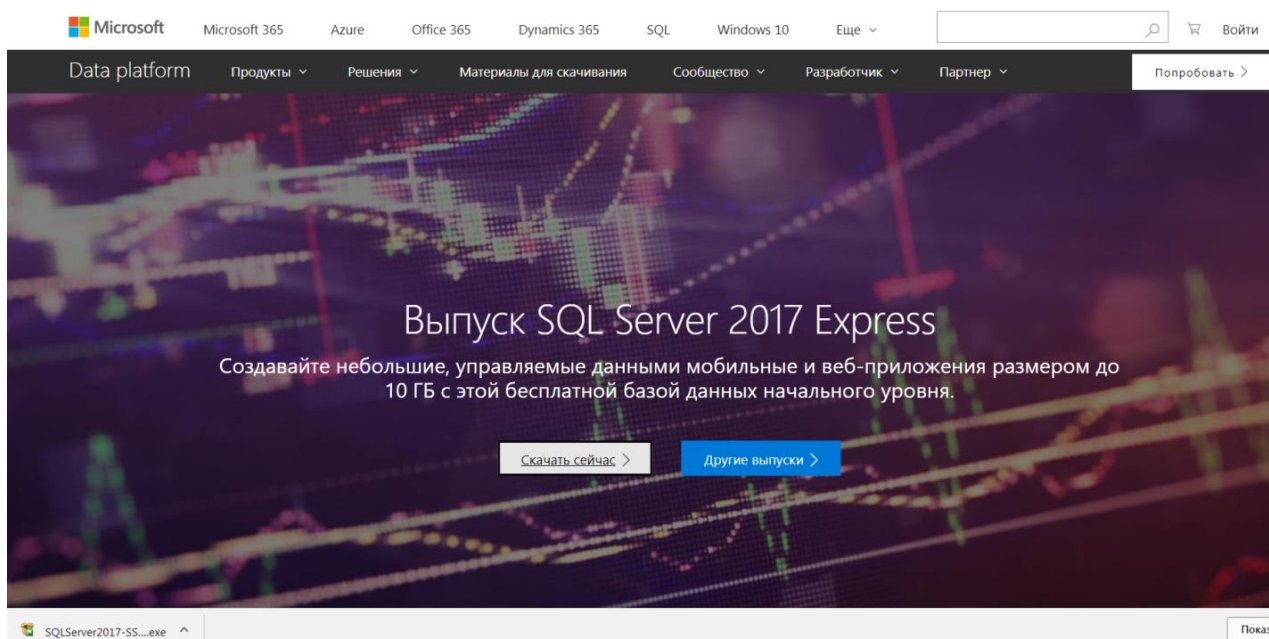


Рис. 32. Диалоговое окно для скачивания SQL Server 2017 Express

В результате на компьютер загрузится файл SQLServer2017-SSEI-Expr.exe размером 4,76 МБ, который затем необходимо запустить для установки выпуска. Запуск данного файла инициирует вызов программы web-установщика данного выпуска. Дальнейшая установка аналогична установке Microsoft SQL Server 2014, рассмотренной во второй главе данного пособия.

Затем необходимо установить Management Studio. Для этого сначала необходимо скачать ее дистрибутив. Это можно сделать, например, запустив «Центр установки SQL Server» из меню Пуск и выбрать пункт «Установка средств управления SQL Server» (рис. 37).

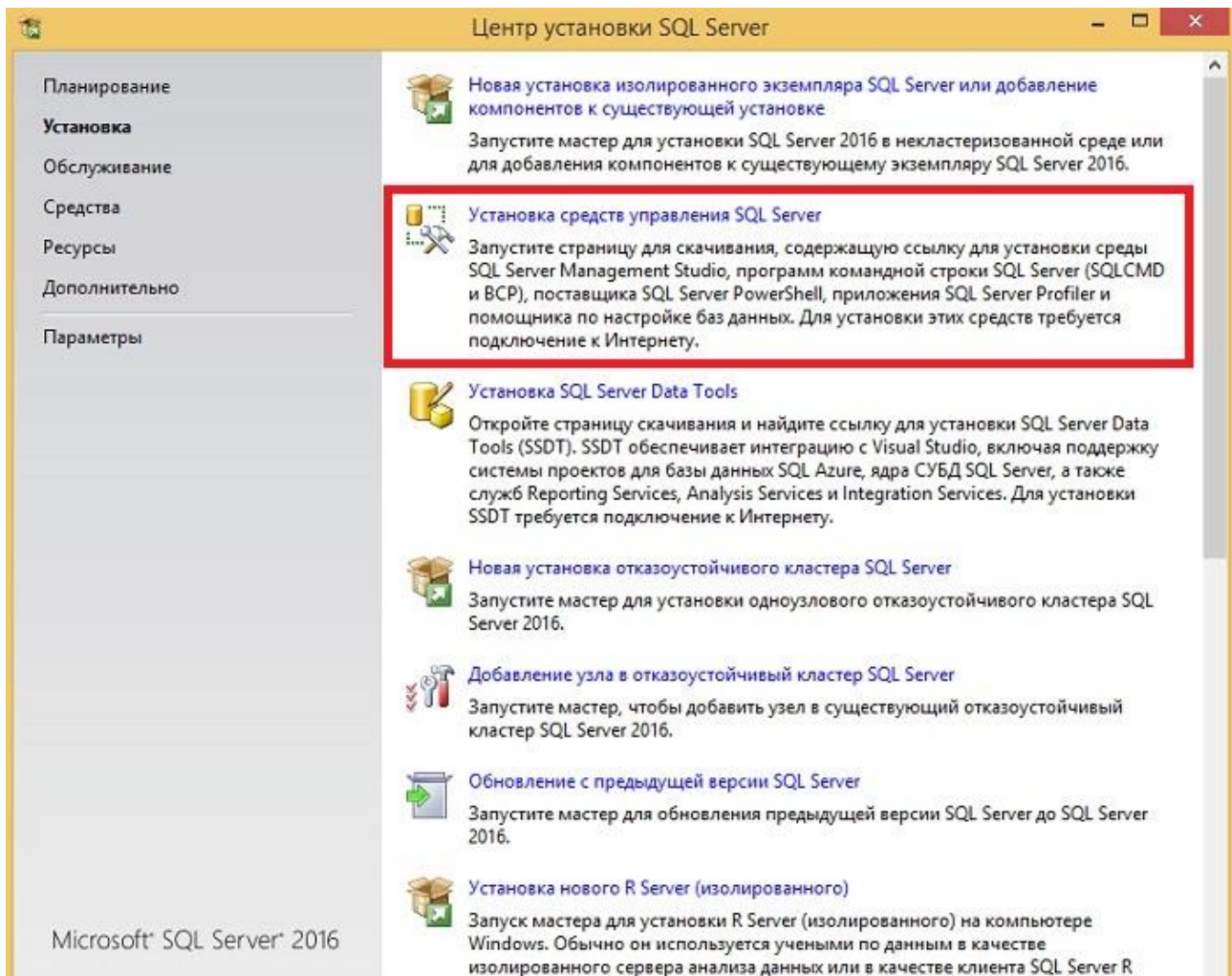


Рис.37. Окно центра установки SQL Server

Так же установить данное средство управления можно, скачав его с официального сайта Microsoft (рис. 38).

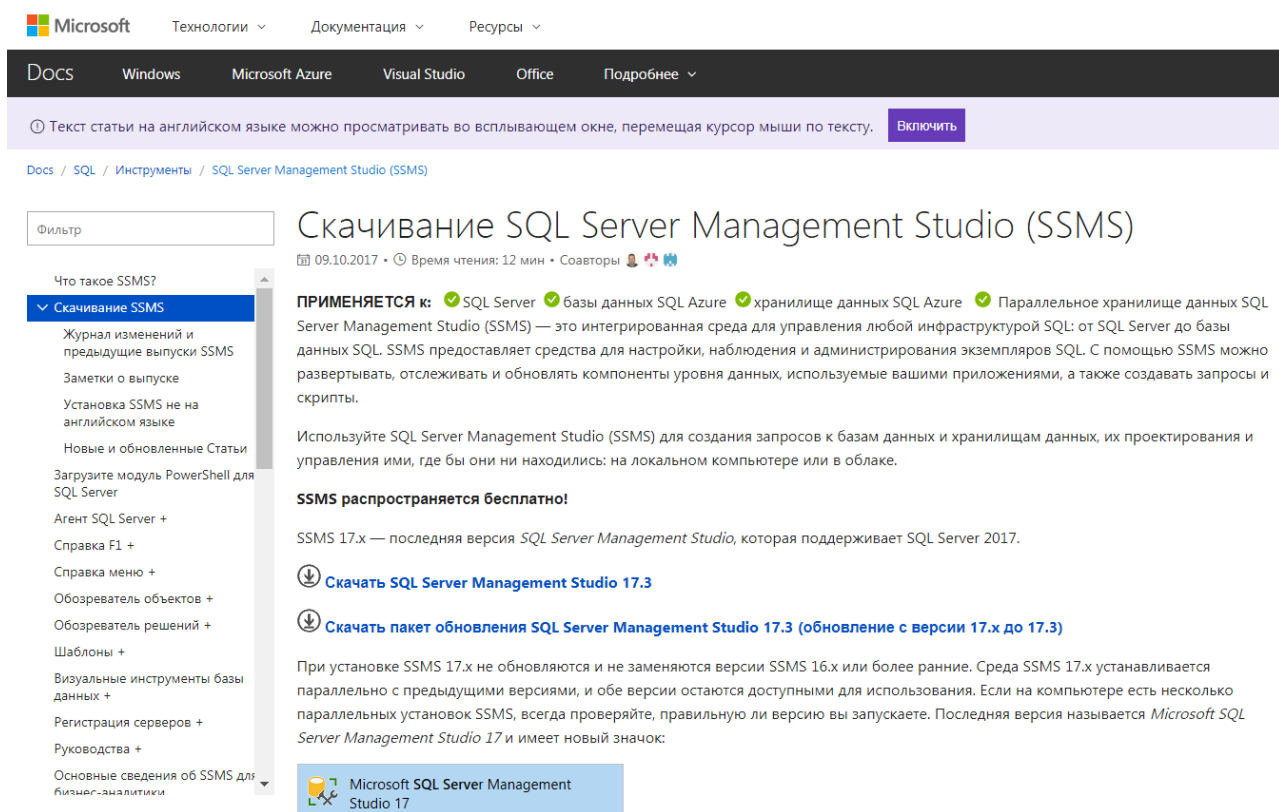


Рис.38. Сайт для скачивания SQL Server Management Studio (SSMS)

В обоих случаях откроется страница, с которой можно будет загрузить дистрибутив Management Studio. В результате загрузится файл SSMS-Setup-RUS.exe размером примерно 816 МБ. Затем запускаем скаченный файл SSMS-Setup-RUS.exe и попадаем в меню программы установки. Дальнейшая работа с Management Studio описана в 4 разделе данного пособия.

### Новое в выпуске SSMS 17.3

SSMS 17.3 - это новейшая версия SQL Server Management Studio. Поколение 17.x среды SSMS поддерживает почти все функциональные возможности выпусков SQL Server 2008 - SQL Server 2017. Версия 17.x также поддерживает SQL Analysis Services в режиме PaaS.

Версия 17.3 включает следующее:

1. Добавлен новый мастер импорта неструктурированных файлов, работающий на базе интеллектуальной платформы, которая упрощает процесс импорта CSV-файлов и не требует от пользователей многочисленных действий или специализированных знаний. Дополнительные сведения см. в разделе Мастер импорта неструктурированных файлов в SQL.
2. В обозреватель объектов добавлен узел XEventProfiler. Дополнительные сведения см. в разделе Использование SSMS XEventProfiler.
3. Обновлена фильтрация и классификация ожиданий в отчете об истории ожиданий из панели мониторинга производительности.
4. Добавлена проверка синтаксиса функции Predict.
5. Добавлена проверка синтаксиса запросов в функции управления внешними библиотеками.

6. Добавлена поддержка SMO для управления внешними библиотеками.
7. Добавлена поддержка запуска PowerShell в окне «Зарегистрированные серверы» (требуется новый модуль PowerShell для SQL).
8. AlwaysOn: добавлена поддержка маршрутизации запросов только для чтения для групп доступности.
9. Добавлена возможность отправлять в окно вывода данные трассировки по входам с использованием проверки подлинности "ActiveDirectory — универсальная с поддержкой MFA". По умолчанию эта функция отключена, и ее нужно включить в параметрах пользователя, последовательно выбрав «Сервис» > «Параметры» > «Службы Azure» > «Облако Azure» > «Уровень трассировки для окна вывода ADAL».
10. Хранилище запросов
11. Пользовательский интерфейс хранилища запросов доступен, даже когда служба QDS отключена, при условии, что она записала какие-либо данные.
12. Пользовательский интерфейс хранилища запросов теперь предоставляет классификацию ожиданий во всех существующих отчетах. За счет этого пользователям будут доступны сценарии "Топ ожидающих запросов" и многие другие.
13. Включать заголовки параметров сценариев стало необязательно. По умолчанию эта возможность отключена, и ее можно включить в параметрах пользователя, последовательно выбрав «Сервис» > «Параметры» > «Обозреватель объектов SQL Server» > «Сценарии» > «Включить заголовок с параметрами сценариев». Это связано с элементом Connect 3139199.

14. Удалена фирменная символика "RC".

Поддерживаемые предложения SQL:

1. Эта версия SSMS работает со всеми поддерживаемыми версиями SQL Server (SQL Server 2008 — SQL Server 2017) и предоставляет превосходную поддержку новейших облачных функций базы данных SQL Azure и хранилища данных SQL Azure.
2. SQL Server 2000 или SQL Server 2005 напрямую не блокируются, однако некоторые функции могут работать неправильно.
3. Кроме того, SSMS 17.X можно установить одновременно с SSMS 16.X или SQL Server 2014 SSMS и более ранними версиями.

При использовании последнего пакета обновления данный выпуск SSMS поддерживает следующие 64-разрядные платформы:

1. Windows 10 (64-разрядная);
2. Windows 8.1 (64-разрядная);
3. Windows 8 (64-разрядная);
4. Windows 7 с пакетом обновления 1 (64-разрядная);
5. Windows Server 2016 \*;
6. Windows Server 2012 R2 (64-разрядная версия);
7. Windows Server 2012 (64-разрядная версия);
8. Windows Server 2008 R2 (64-разрядная версия).

\* Среда SSMS 17.X основана на изолированной оболочке Visual Studio 2015, которая была выпущена до Windows Server 2016. Корпорация Майкрософт уделяет большое внимание совместимости приложений и гарантирует, что уже выпущенные приложения продолжат работать в последних выпусках Windows. Чтобы минимизировать проблемы с запуском SSMS в Windows Server 2016, убедитесь, что для SSMS установлены все последние обновления. При возникновении каких-либо проблем с SSMS в Windows Server 2016, обратитесь в службу поддержки. Служба поддержки определит, связана ли проблема с SSMS, с Visual Studio или с совместимостью SSMS и Windows. Затем ваш запрос будет перенаправлен соответствующей группе для дальнейшего изучения.

**Примечание.** Советы по установке SSMS и связанные с ней проблемы.

Минимизация перезагрузок при установке. Выполните следующие действия, чтобы уменьшить вероятность того, что программе установки SSMS потребуется перезагрузка после установки:

1. Убедитесь, что используется последняя версия распространяемого пакета Visual C++ 2013. Требуется версия 12.00.40649.5 (или более поздняя). Нужна только 64-разрядная версия.

2. Убедитесь, что на компьютере используется платформа .NET Framework 4.6.1 (или более поздней версии).

3. Закройте все остальные экземпляры Visual Studio, открытые на компьютере.

4. Убедитесь, что на компьютере установлены все последние обновления операционной системы.

Указанные действия обычно требуется выполнить всего один раз. В некоторых ситуациях перезагрузка необходима для дополнительных обновлений до того же основного номера версии SSMS. Все необходимые компоненты SSMS для промежуточных обновлений уже будут установлены на компьютере.

**Рассмотрим проблемы и ограничения, имеющиеся в выпуске 17.3.**

Общие ошибки SSMS:

1. При использовании универсальной многофакторной проверки подлинности в Azure AD следующие функциональные возможности SSMS не поддерживаются.

2. Для проверки подлинности Azure AD не поддерживается помощник по настройке ядра СУБД. Имеется известная проблема, когда пользователь получает довольно непонятное сообщение об ошибке «Не удалось загрузить файл или сборку Microsoft.IdentityModel.Clients.ActiveDirectory...» вместо ожидаемого сообщения «Помощник по настройке ядра СУБД не поддерживает базу данных SQL Microsoft Azure. (DTAClient)».

3. При попытке проанализировать запрос в DTA появляется сообщение об ошибке «Объект должен реализовать IConvertible. (mscorlib)».

4. В списке отчетов хранилища запросов из обозревателя объектов отсутствуют Регрессионные запросы. Обходное решение: щелкните правой

кнопкой мыши узел Хранилище запросов и выберите Просмотр регрессионных запросов.(Integration Services (IS)).

5. Для выполнения пакетов в ScaleOut указан некорректный [путь\_выполнения] в элементе [каталог].[сообщения\_событий]. Путь [execution\_path] начинается с "\Package", а не с имени объекта исполняемого файла пакета. При просмотре обзорного отчета по выполнению пакетов в SSMS ссылка «Путь выполнения» в обзоре выполнения не работает. Обходное решение: щелкните «Просмотреть сообщения» в обзорном отчете, чтобы проверить все сообщения событий.

### Ограничения вычислительной емкости для разных выпусков SQL Server.

На рис. 39 приведена схема работы в физических и виртуальных средах, содержащих процессоры с технологией Hyper-Threading.

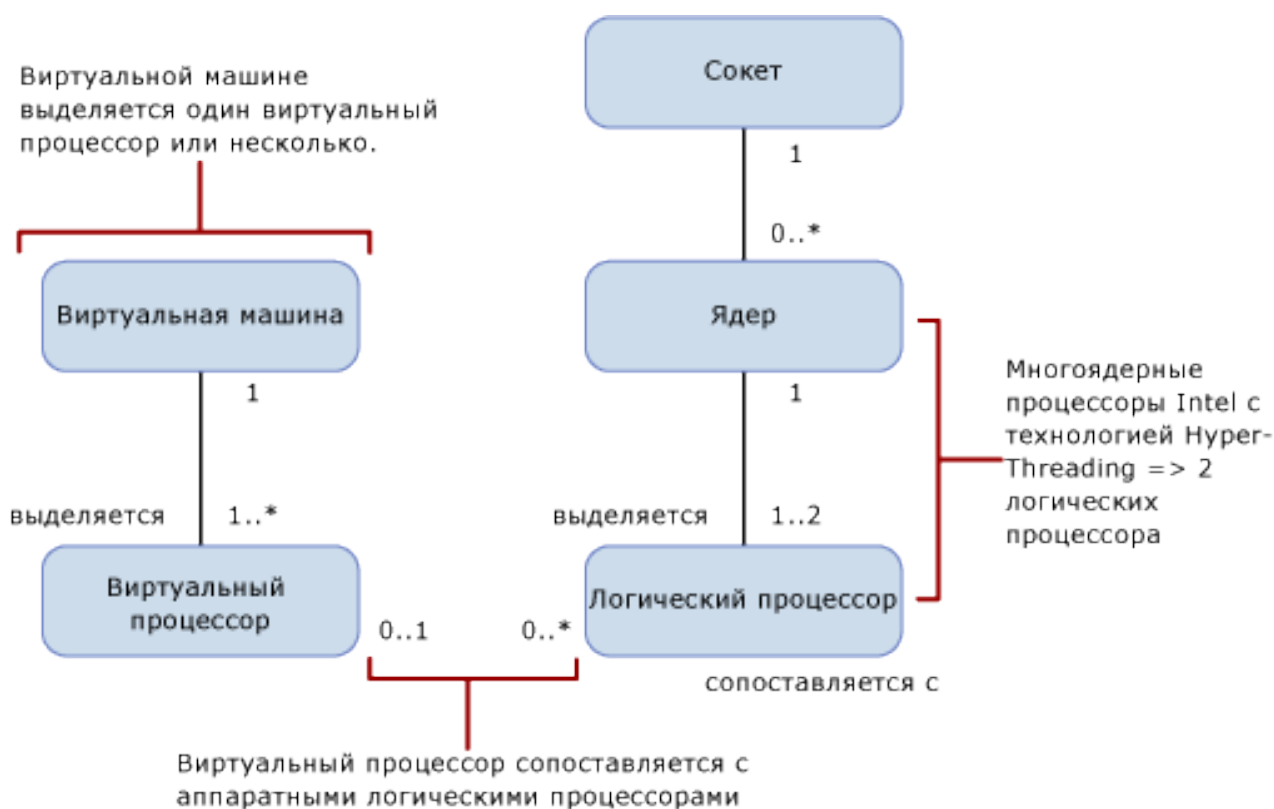


Рис. 39. Схема работы в физических и виртуальных средах, содержащих процессоры с технологией Hyper-Threading

Условные обозначения на схеме (рис. 39):

Значение Описание

0..1 - Ноль или один;

1 - Ровно один;

1..\* - Один или более;

0..\* - Ноль или более;

1..2 - Один или два.

### Примечания:

1. Поток или логический процессор - одна логическая вычислительная подсистема с точки зрения SQL Server, операционной системы, приложения или драйвера.

2. Ядро- единица процессора. Оно может состоять из одного логического процессора или нескольких.

3. Физический процессор часто еще называют процессорным пакетом или сокетом. Физический процессор может содержать одно или несколько ядер.

Системы с несколькими физическими процессорами, а также системы, где физические процессоры имеют несколько ядер или поддерживают технологию Hyper-Threading, позволяют операционной системе одновременно выполнять несколько задач. Каждый поток выполнения представляется как логический процессор. Например, если на компьютере установлено два четырехъядерных процессора с включенной технологией Hyper-Threading и двумя потоками на ядро, то получится 16 логических процессоров: 2 процессора с 4 ядрами в каждом процессоре и по 2 потока в каждом ядре.

**Примечания:**

1. Вычислительная мощность логического процессора на один поток ядра с технологией Hyper-Threading меньше, чем мощность логического процессора с таким же ядром при отключенной технологии Hyper-Threading.

2. Емкость вычислительных ресурсов двух логических процессоров в ядре с технологией Hyper-Threading больше, чем у того же ядра без этой технологии.

3. Каждый выпуск SQL Server обладает двумя ограничениями вычислительной мощности: максимальное число сокетов (физических процессоров или процессорных пакетов); максимальное число ядер, обнаруженных операционной системой.

Эти ограничения относятся к отдельному экземпляру SQL Server. Они представляют максимальную вычислительную мощность, которую будет использовать отдельный экземпляр. Ограничения не распространяются на сервер, где может развертываться экземпляр. На практике развертывание нескольких экземпляров SQL Server на одном физическом сервере является эффективным способом использовать вычислительную емкость физического сервера, на котором процессоров или ядер больше, чем допускают ограничения.

Учебное текстовое электронное издание

**Баранкова Инна Ильинична  
Михайлова Ульяна Владимировна  
Лукьянов Георгий Игоревич**

**РАЗРАБОТКА БД В MS SQL SERVER  
С ИСПОЛЬЗОВАНИЕМ SSMS**

Учебное пособие

1,74 Мб

1 электрон. опт. диск

г. Магнитогорск, 2018 год  
ФГБОУ ВО «МГТУ им. Г.И. Носова»  
Адрес: 455000, Россия, Челябинская область, г. Магнитогорск,  
пр. Ленина 38

ФГБОУ ВО «Магнитогорский государственный  
технический университет им. Г.И. Носова»  
Кафедра информатики и информационной безопасности  
Центр электронных образовательных ресурсов и  
дистанционных образовательных технологий  
e-mail: ceor\_dot@mail.ru