Архитектура СУБД SQL Server. Знакомство со средой SQL Server Management Studio

Основы безопасности СУБД SQL Server

Цели работы:

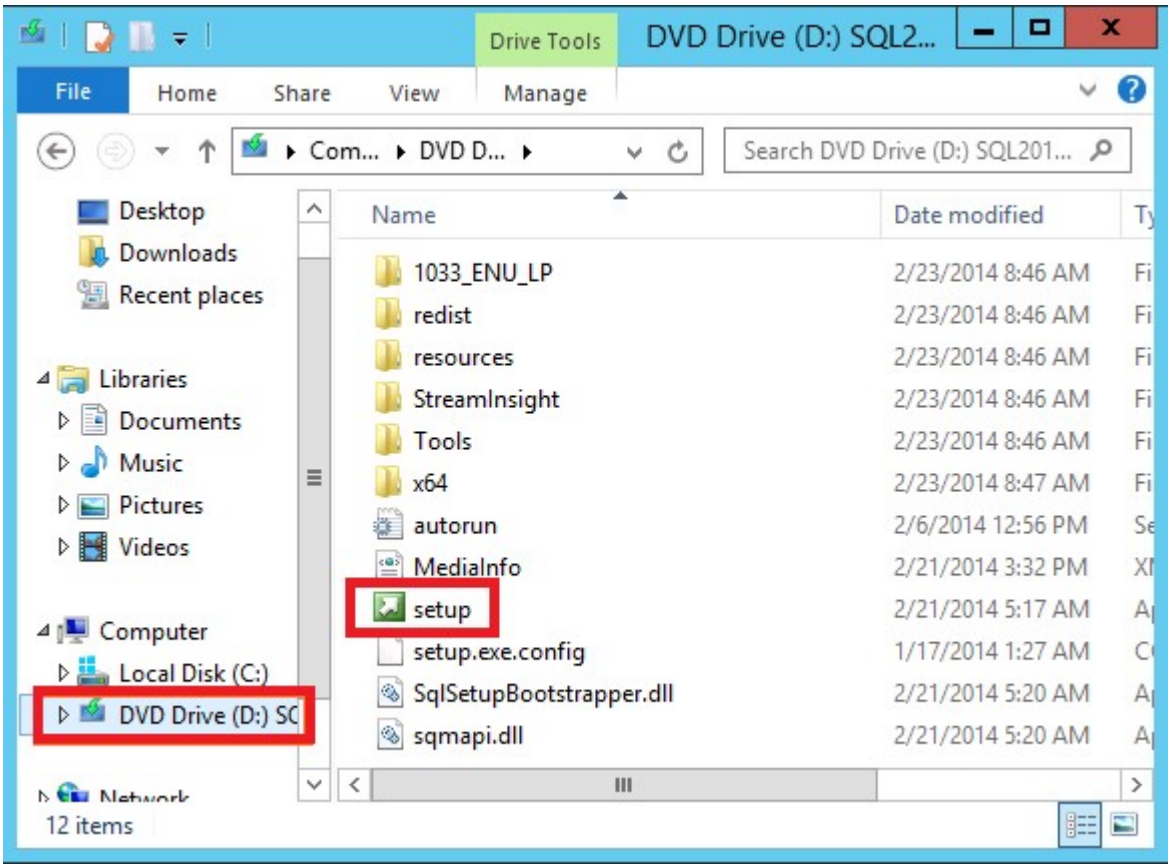
1. Научиться устанавливать SQL Server. Изучить основные принципы клиент-серверной архитектуры СУБД SQL Server.
2. Научиться использовать среду SQL Server Management Studio для подключения к серверу БД, а также для управления несколькими экземплярами сервера БД.
3. Научиться создавать и администрировать БД в середе SQL Server Management Studio.
4. Изучить на практике основы безопасности SQL Server.

Теоретические указания:

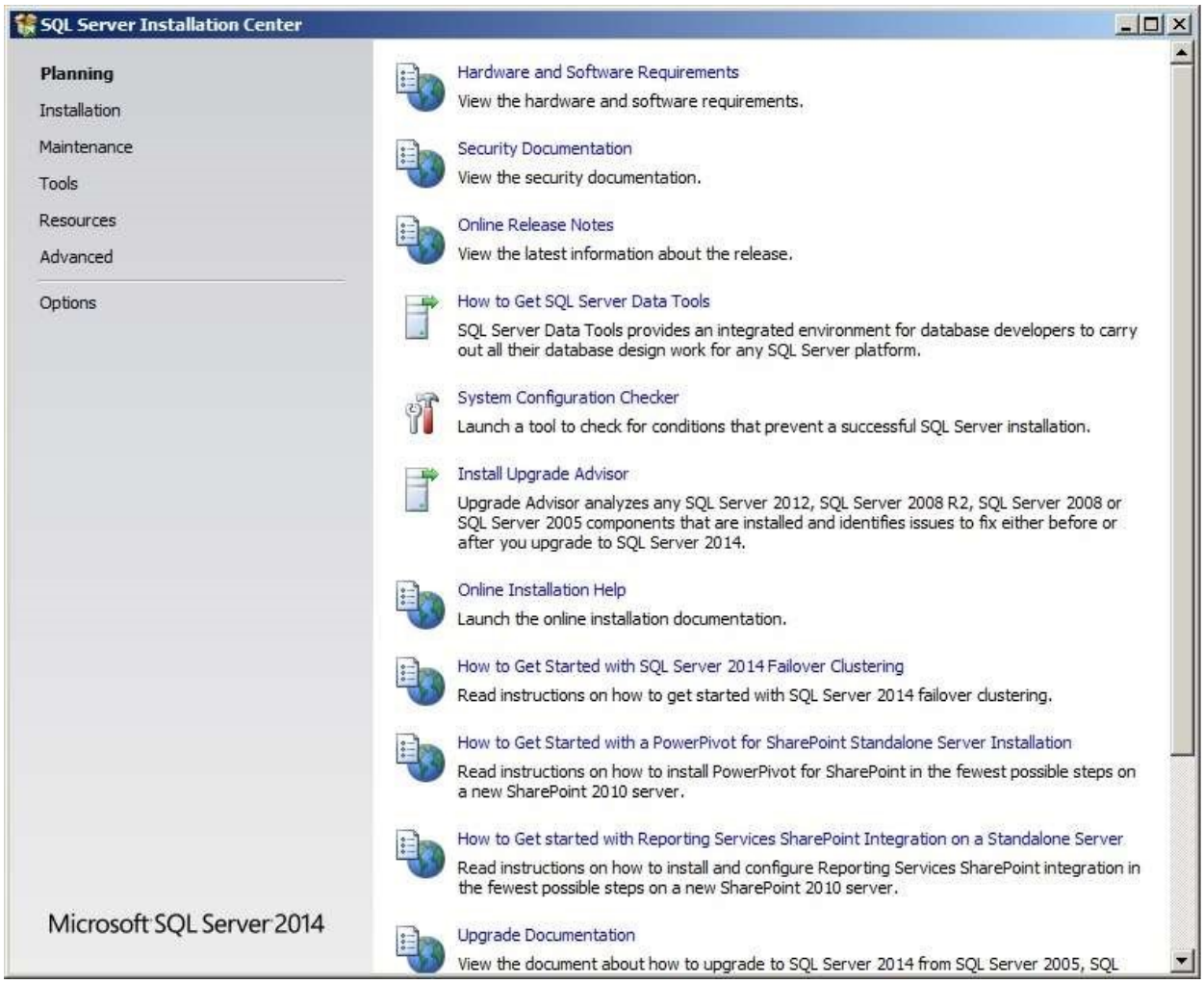
Установка СУБД Microsoft SQL Server

Версию Microsoft SQL Server Developer Edition можно скачать на *сайте корпорации Microsoft*:

Монтируем образ, запускаем процесс установки.



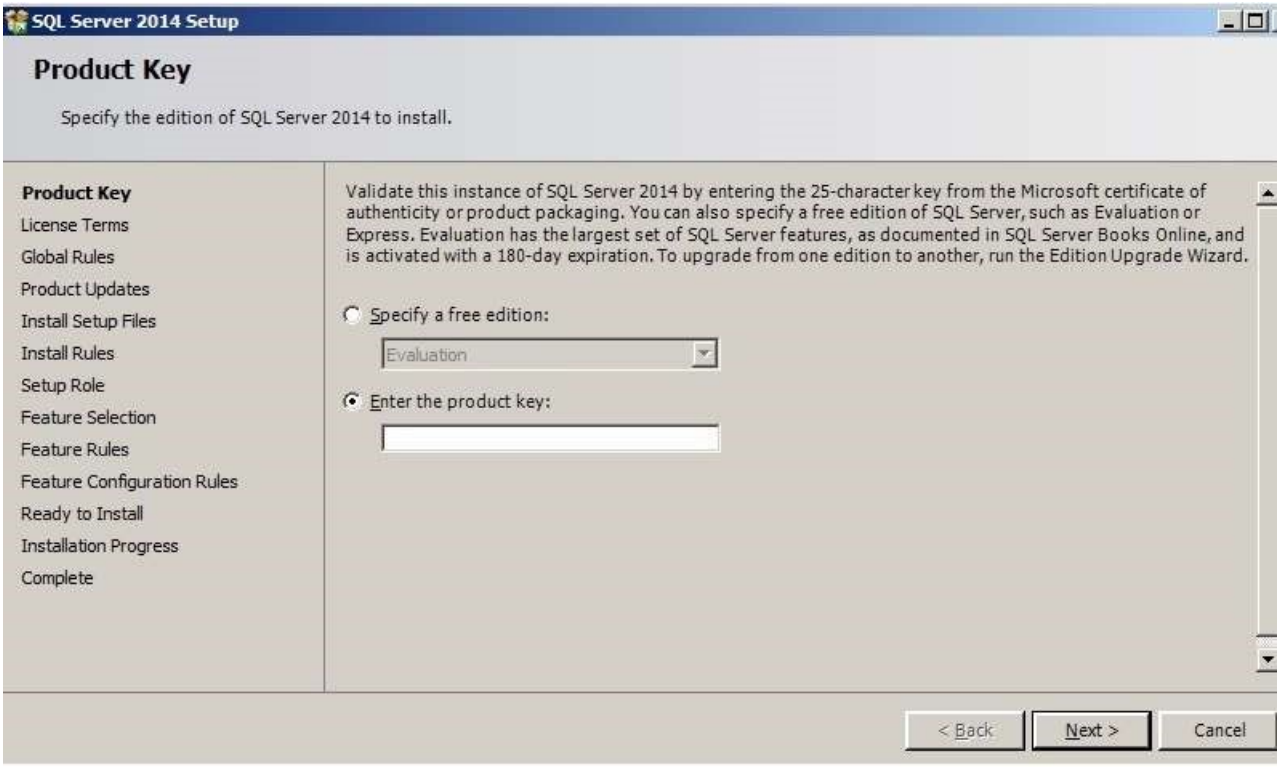
Появляется окно центра установки



Перейдя к разделу «Installation» переходим к списку опций установки. Выбираем опцию «New SQL Server stand-alone installation or add features to an existing installation»

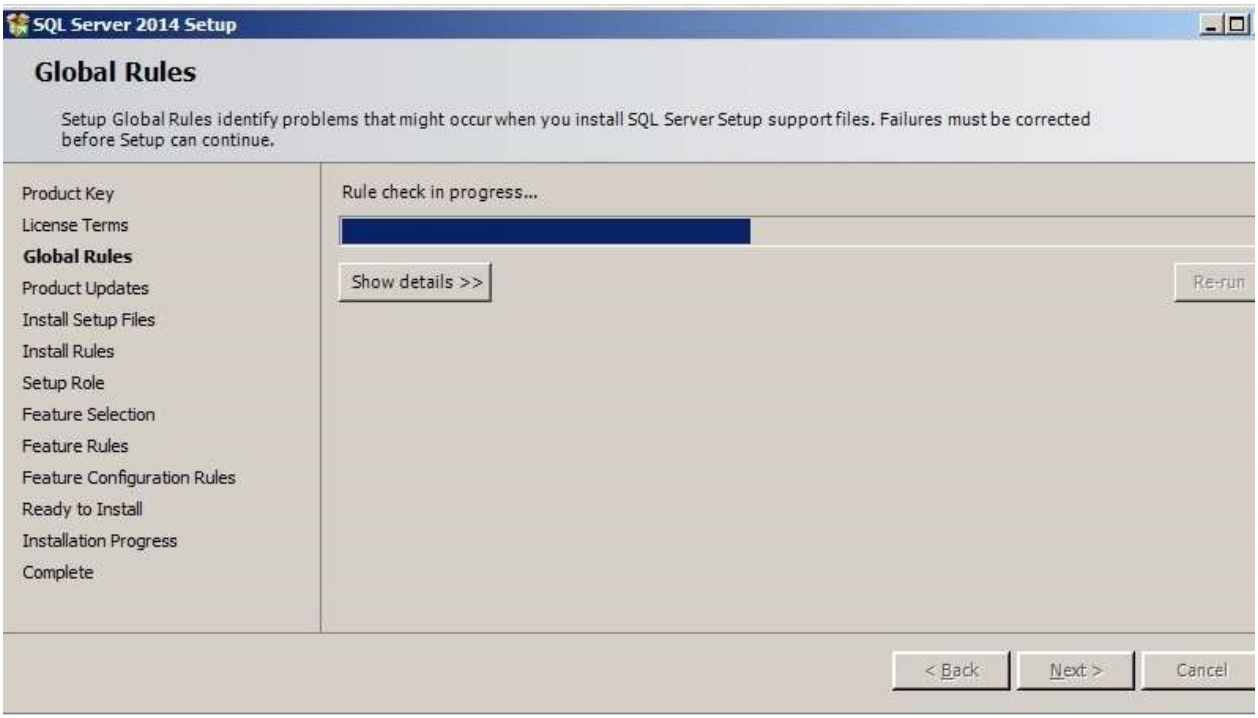
Установщик SQL Server выведет слева шаги, которые он выполнит в процессе установки.

Введите ключ продукта или выберите версию SQL Server Developer Edition (free) и нажмите «Next >»



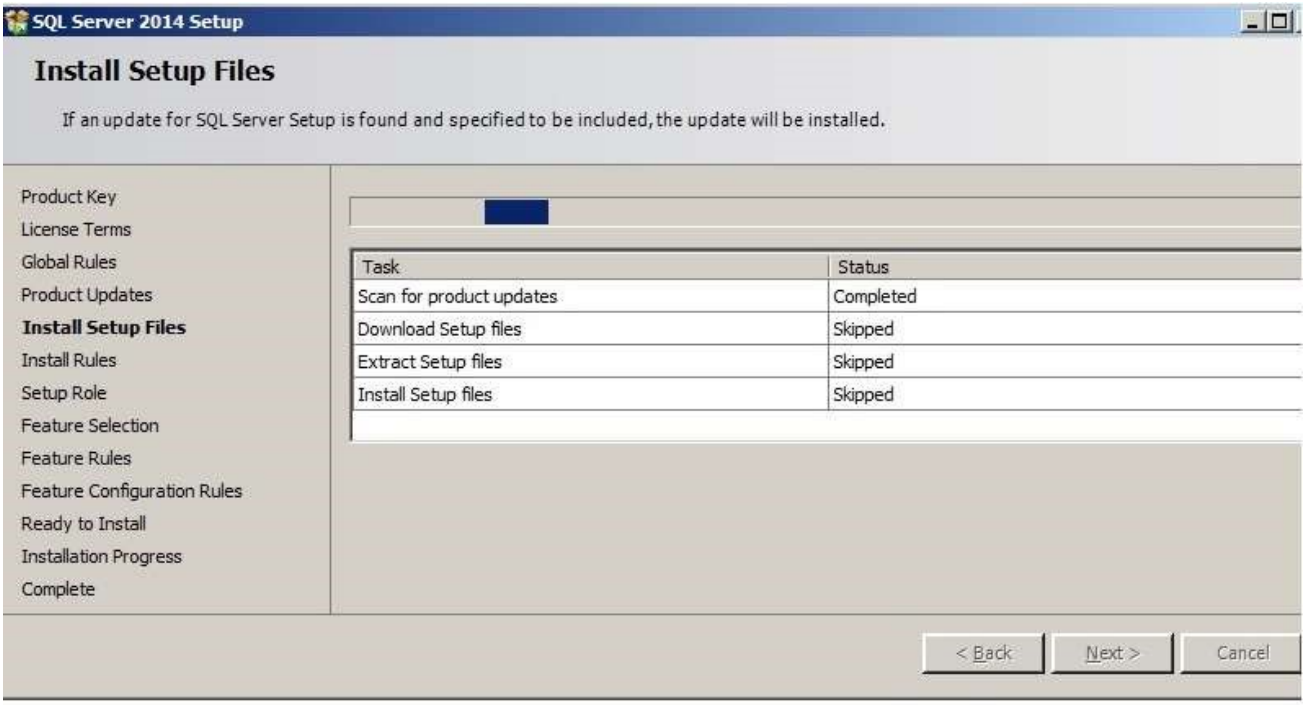
Согласитесь с лицензионным соглашением.

Установщик SQL Server запустит множество проверок. Если какая-либо из проверок завершается неудачей, программа установки предложит меры по устранению причин провала проверки. Программа установки также попытается найти обновления для продукта.

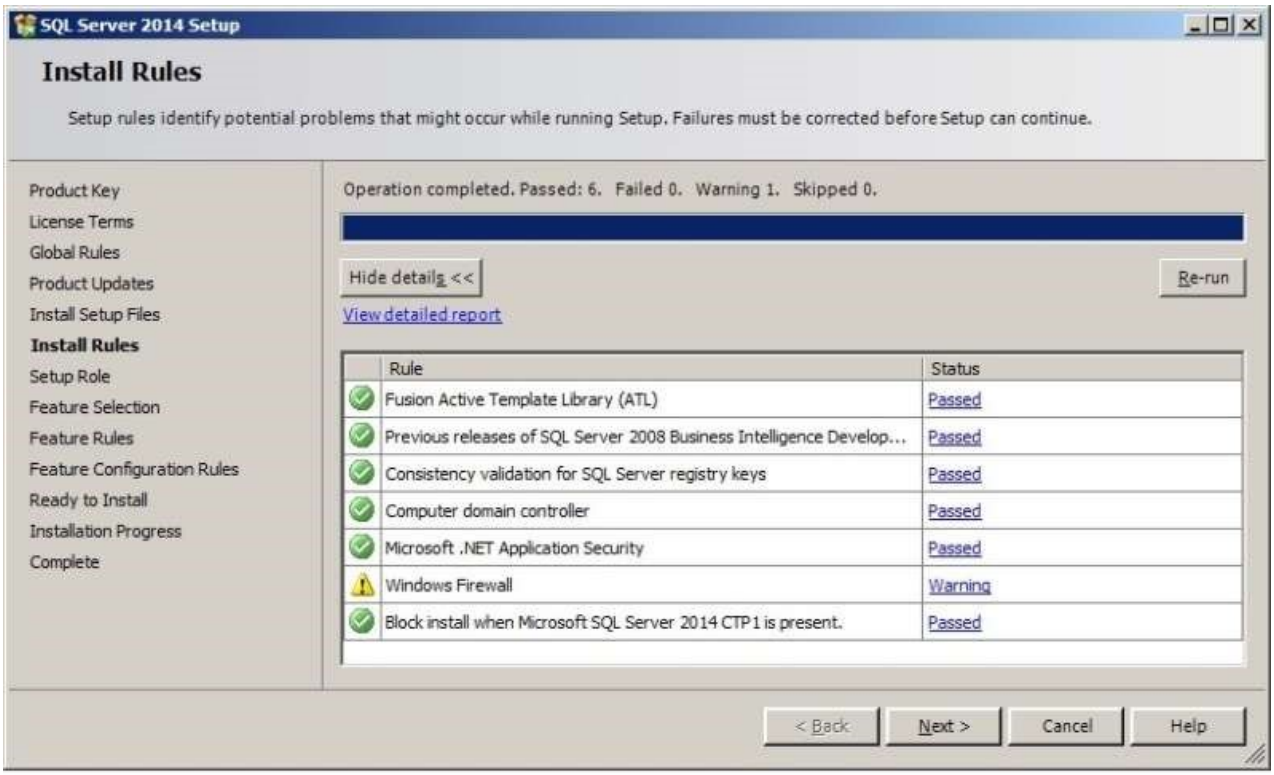


Программа установки загрузит, извлечёт и установит файлы необходимые для установки.

Нажмите «Next >» для продолжения установки.

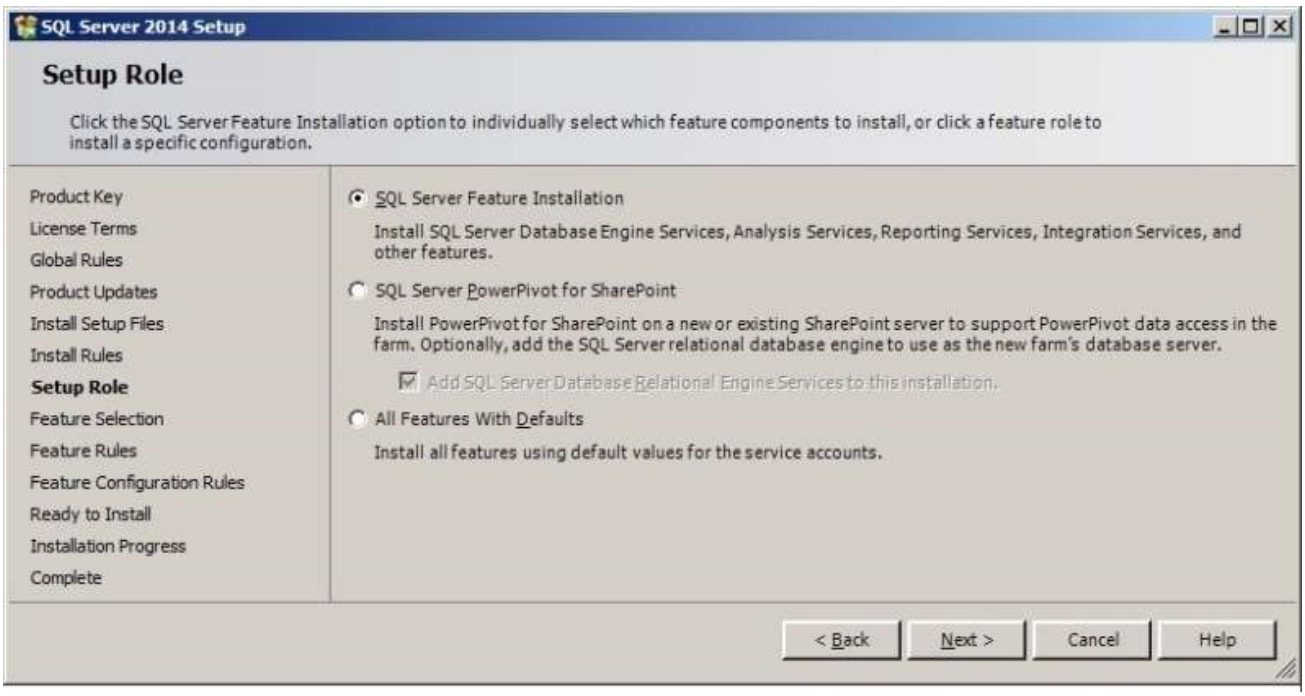


Другой набор проверок будет запущен чтобы убедится, что всё готово ддля установки.

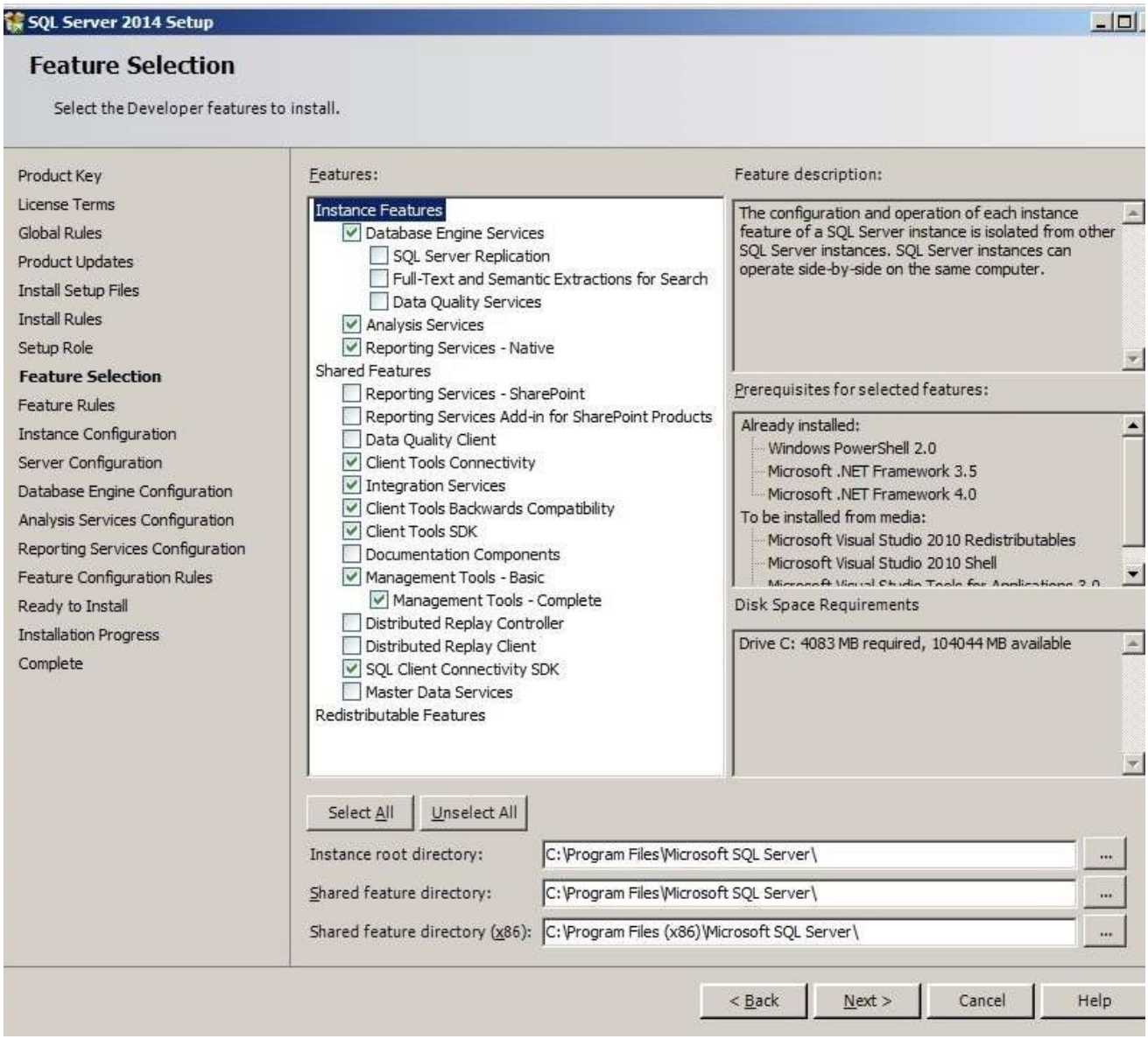


Нажмите «Next >» для продолжения установки.

На странице Setup Role выберем опцию «SQL Server Feature Installation». Это поможет нам установить движок базы данных, Analysis Services, Reporting Services, Integration Services и другие компоненты SQL Server. Нажмите «Next >» для продолжения установки.

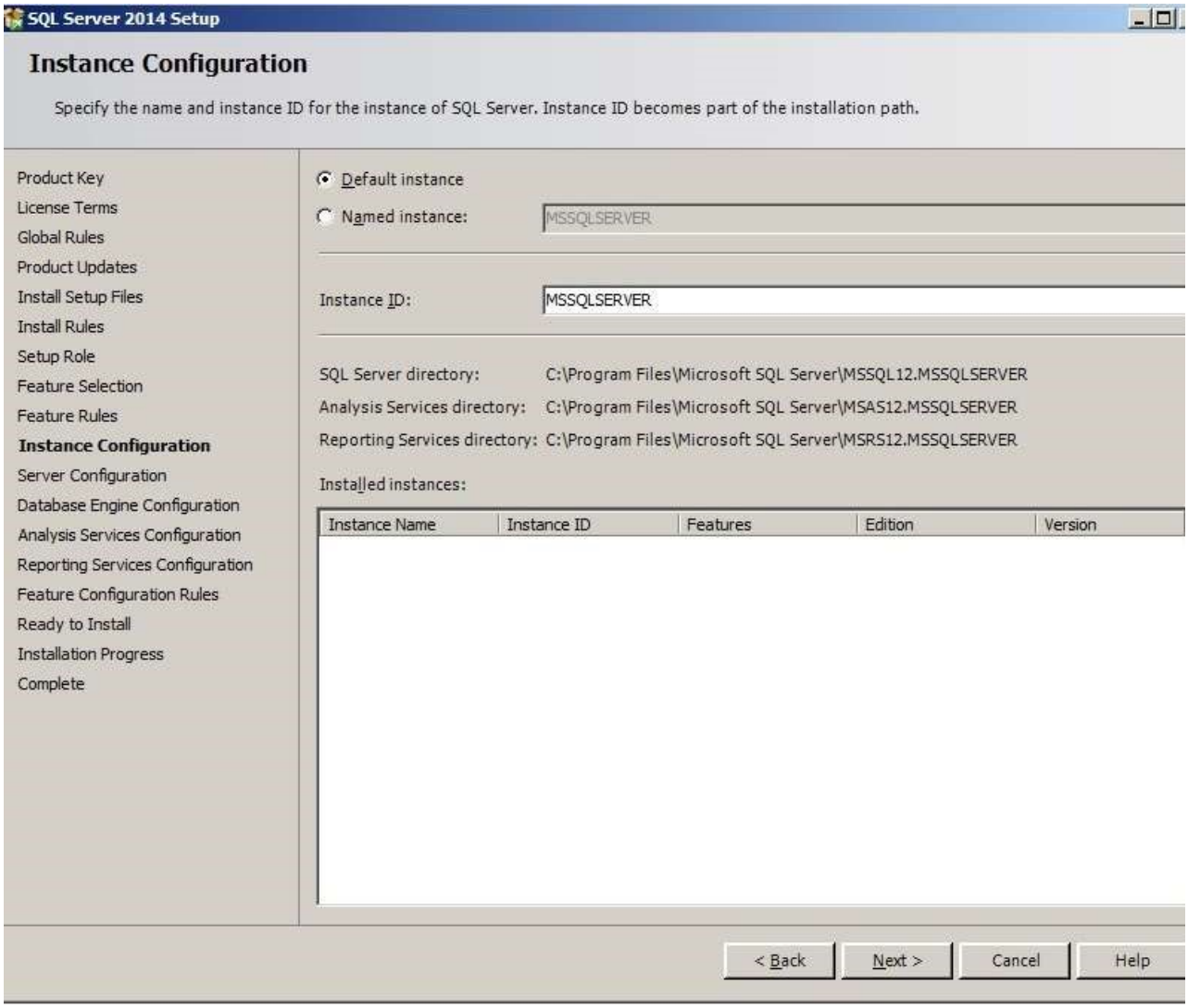


Следующий шаг – это выбор компонентов. Как правило, устанавливаем все компоненты, нажав кнопку «Select All». В качестве примера приводим минимально необходимый набор компонентов.

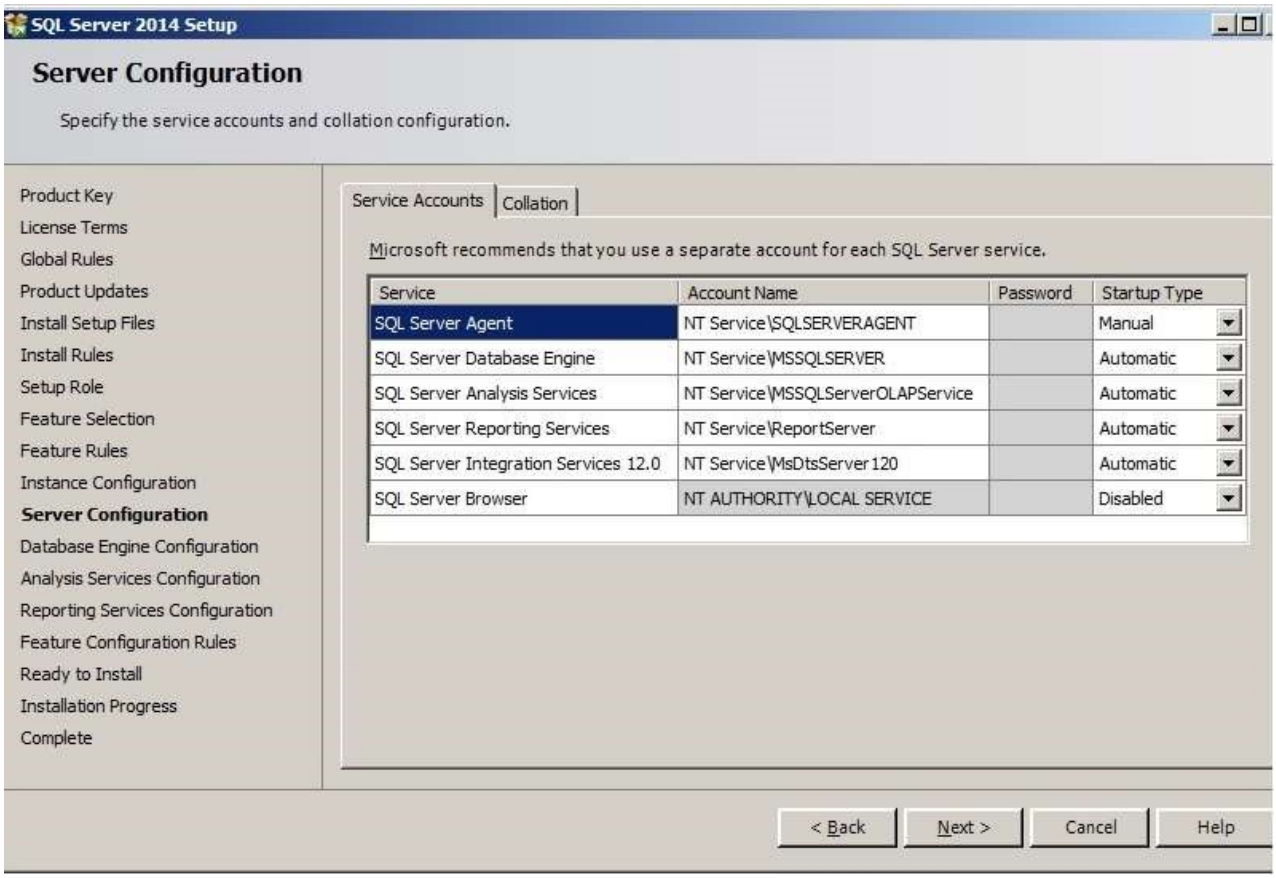


Иногда в процессе установки может появиться сообщение

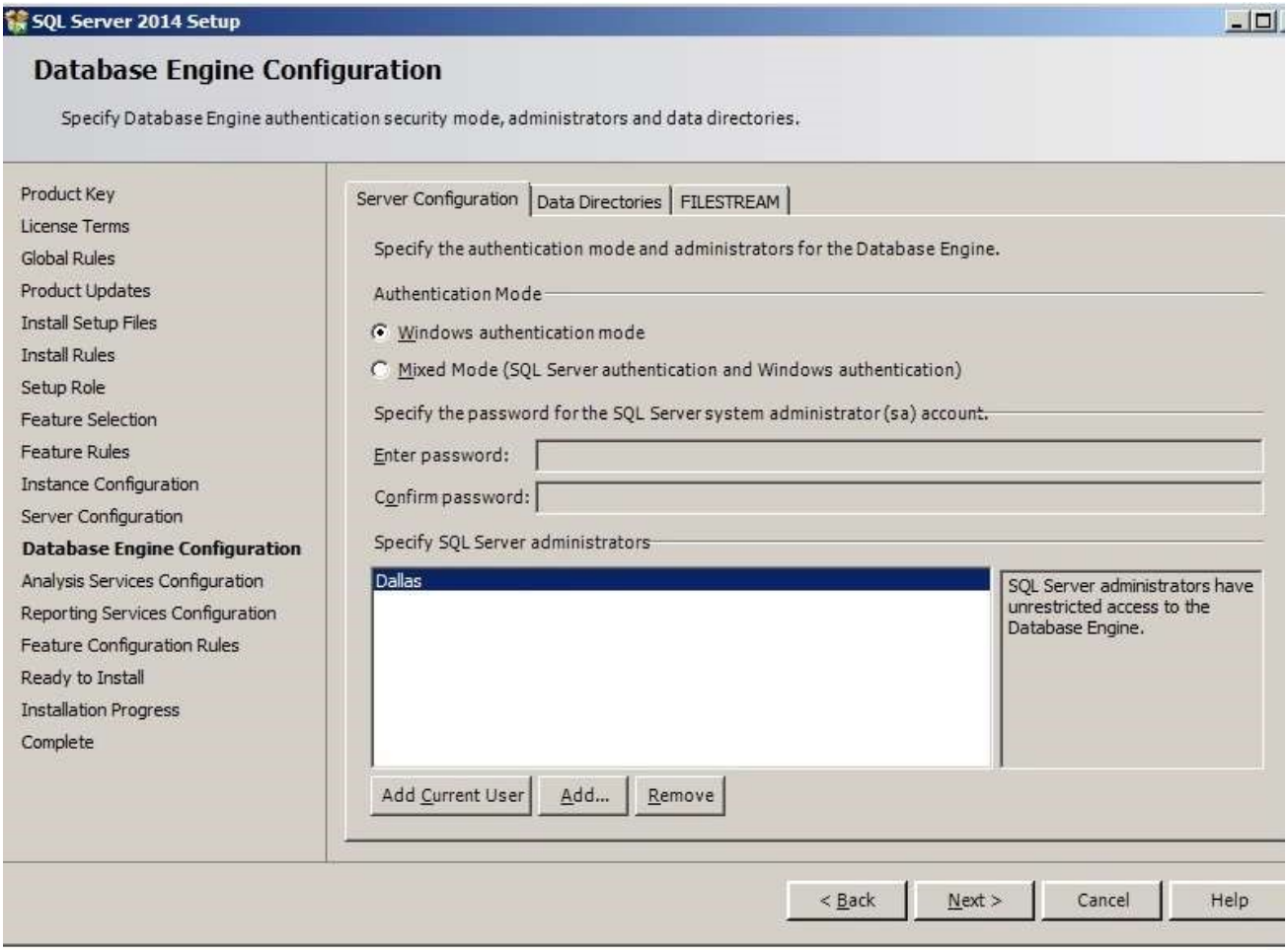
Этап Instance Configuration позволяет установщику указать ID экземпляра установки SQL Server. Здесь так же будут показаны другие установленные экземпляры SQL Server, если такие имеются в системе.



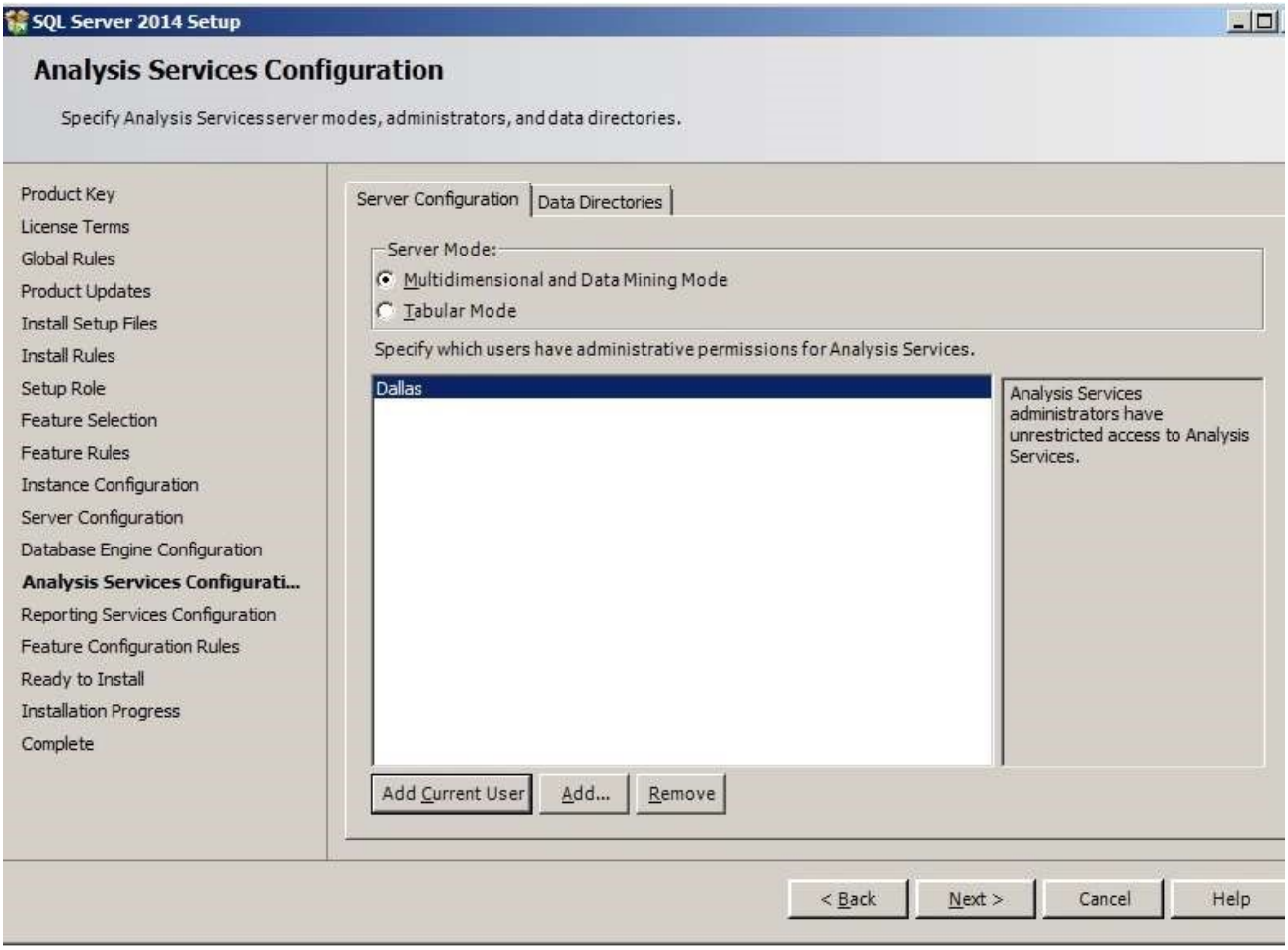
Окно Server Configuration показывает сервисы, которые будут установлены. Оставляем настройки по-умолчанию.



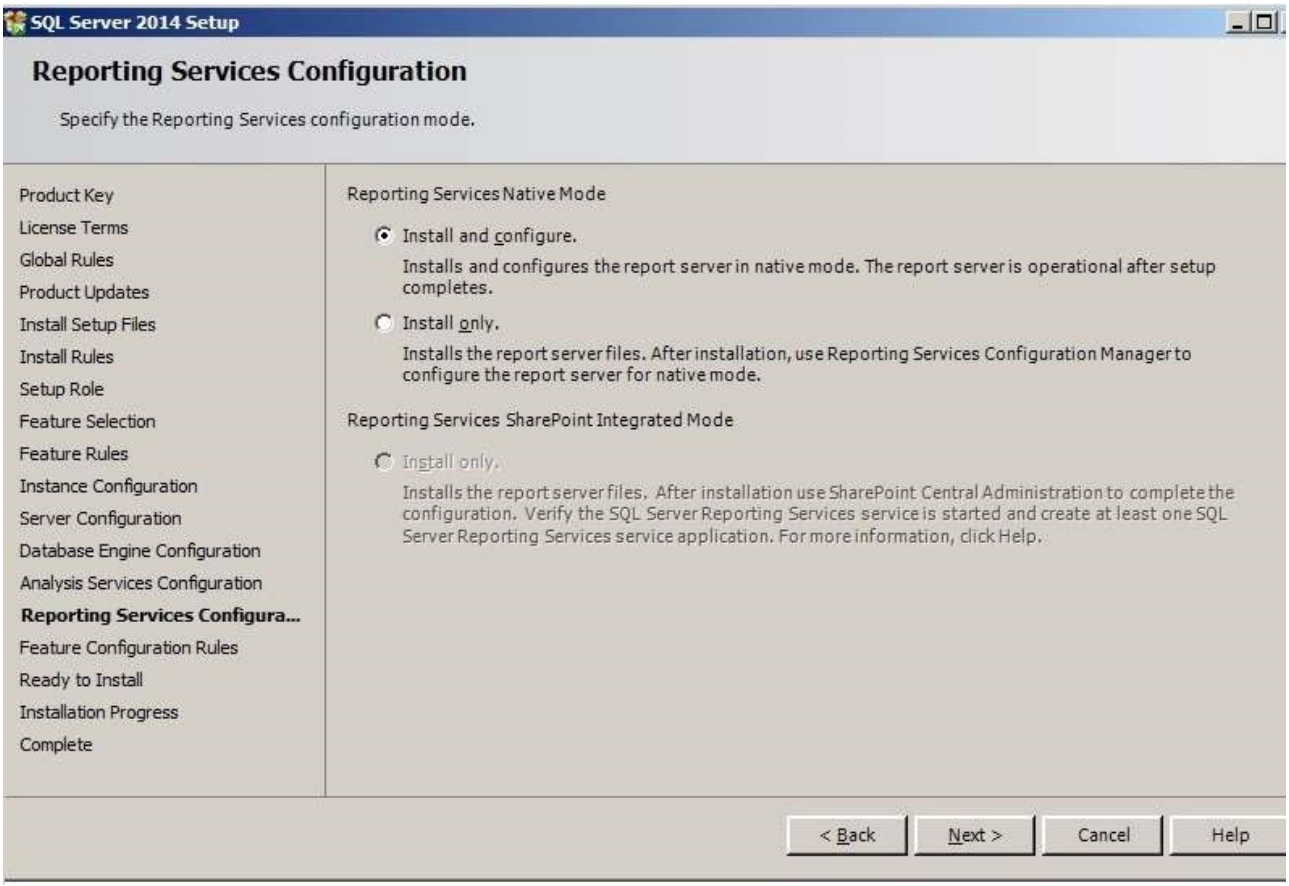
Следует этап Database Engine Configuration. Во вкладке Server Configuration выберите способ аутентификации (обычно выбираем Windows Authentication Mode). Так же нажмите кнопку «Add Current User» для добавления текущего пользователя системы в качестве администратора SQL Server.



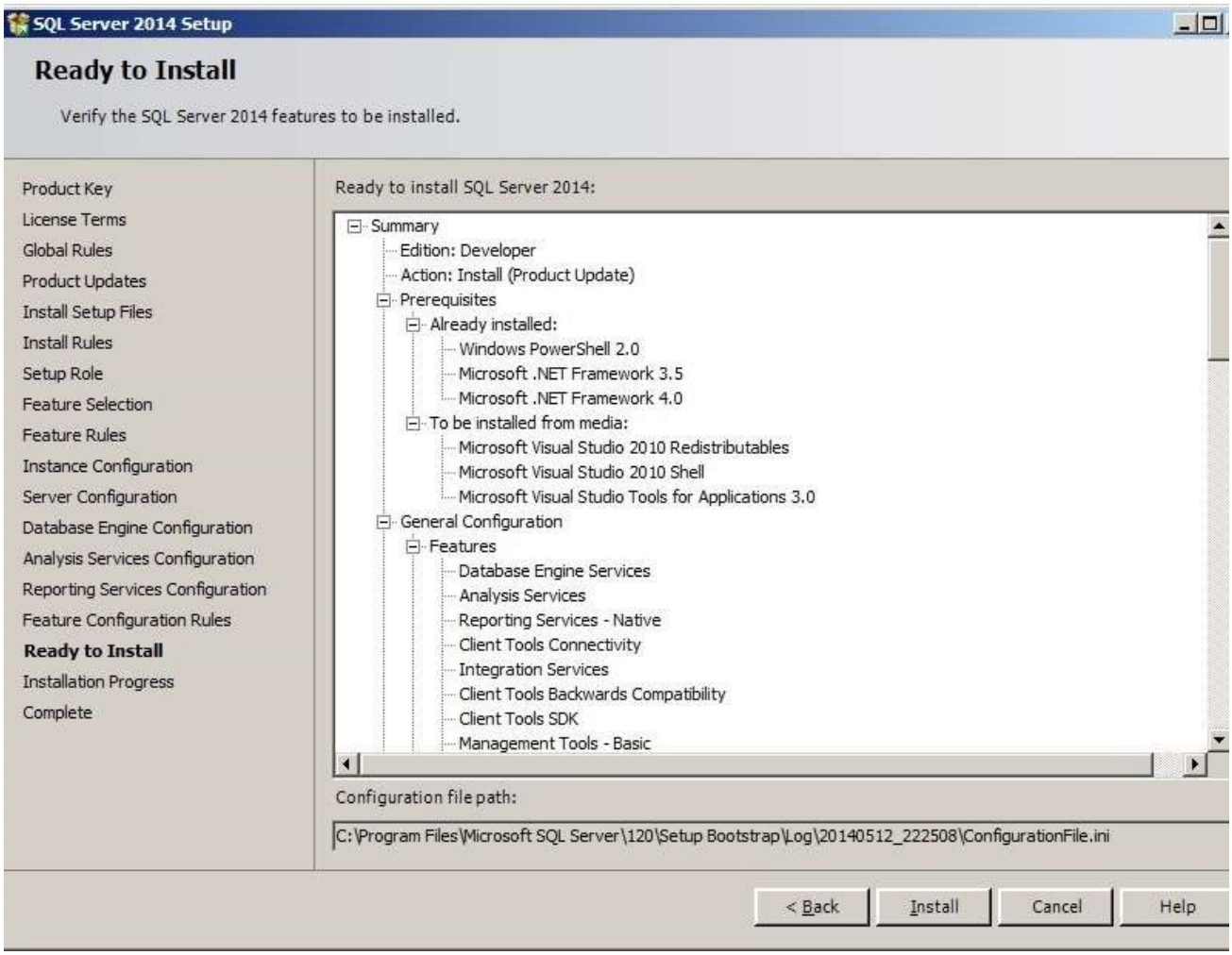
Переходим к шагу Analysis Services Configuration. Выбираем Multidimensional and Data Mining Mode в разделе Server Mode. Так же нажмите кнопку «Add Current User» для добавления текущего пользователя системы для предоставления ему административных полномочий для Analysis Services.



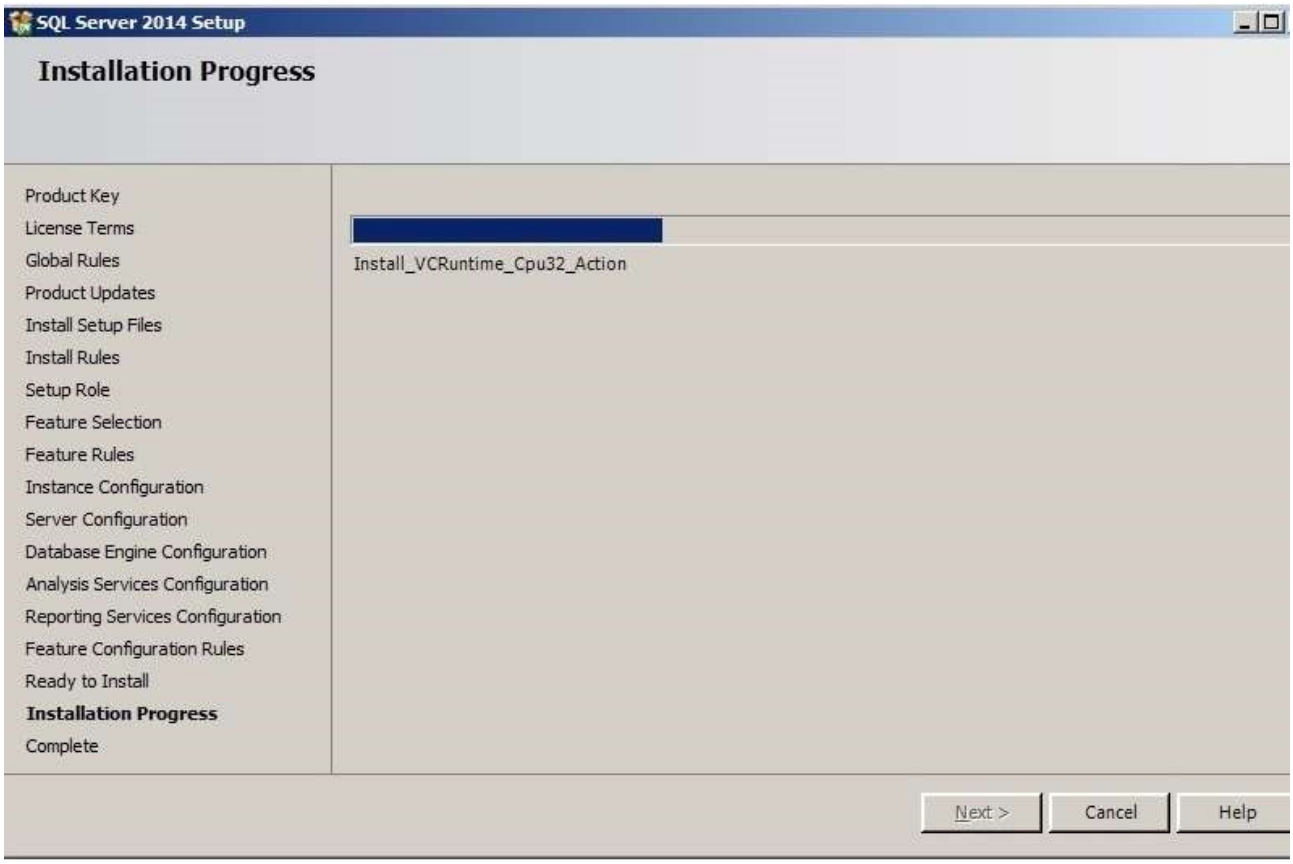
На этапе Reporting Services Configuration выбираем опцию Install and configure.



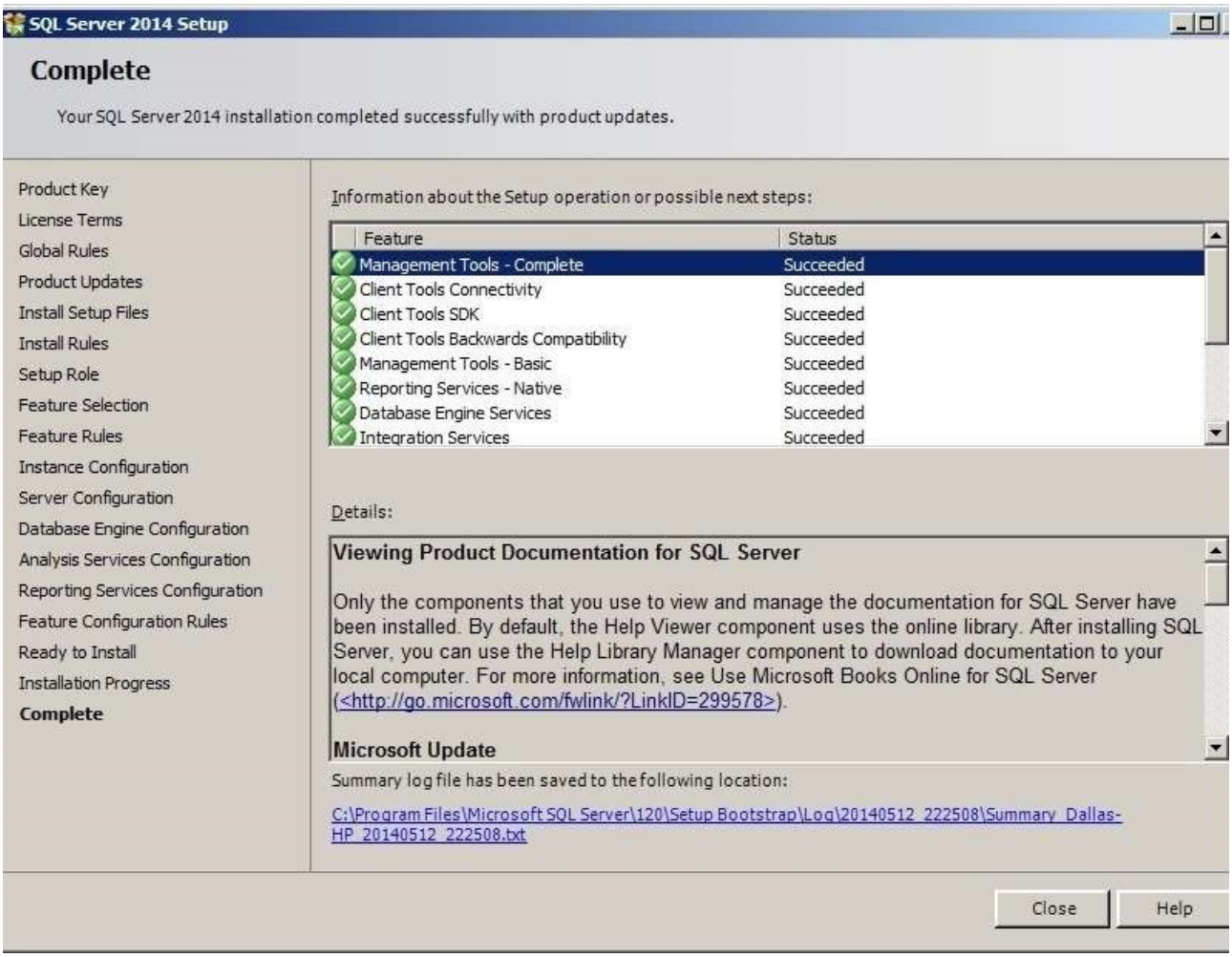
Этап Ready to Install показывает подробную информацию о том, как будет проходить установка и какие дополнительные компоненты будут установлены.



Ждём завершения установки



По окончанию установки, видим подробный отчёт о том какие компоненты SQL Server были установлены.



Теперь SQL Server Installation Center может быть закрыт.

SQL Server готов для работы.

1.1 Введение в SQL Server Management Studio

Основной интерфейс работы с SQL Server реализован с помощью утилиты Management Studio. Она вобрала в себя мощный набор инструментов, подобный Visual Studio, позволяющий разработчикам и администраторам баз данных создавать проекты баз данных и управлять СУБД SQL Server с помощью графического интерфейса пользователя либо с помощью инструкций T- SQL. Для задач бизнес аналитики и обеспечения работы со службами анализа, отчетности и интеграции в пакете SQL Server содержится специализированная среда разработки — SQL Server Business Intelligence Development Studio. SQL Server предоставляет множество возможностей, которые используются для решения различных задач, таких как установка, конфигурирование, аудит и тестирование производительности. Management Studio – необходимое средство администрирования баз данных и взаимодействия с SQL Server. Как администраторы, так и конечные пользователи могут использовать данную среду для быстрой разработки баз данных.

Чтобы открыть Management Studio, выберите пункт стартового меню **Start menu**, **All Programs**, **Microsoft SQL Server**, затем **SQL Server Management Studio.**

1.2 Подключение к серверу

При открытии SQL Server Management Studio появляется диалоговое окно **Connect to Server** (рисунок 1.1), где необходимо специфицировать параметры подключения к серверу БД.

Рисунок 1.1 – Диалоговое окно Connect to Server

Server Type: тип подключаемого сервера. С помощью среды Management Studio можно подключиться как к ядру Database Engine SQL Server, так и к серверу Analysis Server. В нашем случае мы подключаемся к ядру SQL Server.

Server name: имя подключаемого сервера. На одном компьютере можно установить несколько экземпляров SQL Server. Каждому экземпляру назначается имя, по которому можно идентифицировать экземпляр сервера. При установке SQL Server в системе появляется один экземпляр ядра БД. В случае установки версии SQL Server Express данный экземпляр получает имя **<Имя компьютера>\SQLExpress**. В данном случае (рисунок 1.1), мы подключаемся к серверу **ANATOLIIPC\MYSQL**, не указывая имя экземпляра и, таким образом, подключаемся к экземпляру по умолчанию (без указания имени экземпляра). Следует отметить, что каждый экземпляр SQL Server содержит собственные объекты БД и не разделяет их с другими экземплярами.

Authentication: тип аутентификации. Возможны два типа аутентификации: Windows Authentication и SQL Server Authentication. В случае аутентификации Windows подключение к серверу происходит с использованием аккаунта Windows (т.е. используя учетную запись Windows, осуществившую вход в систему). В случае аутентификации SQL Server Authentication необходимо задать имя пользователя и пароль для подключения к серверу.

После нажатия кнопки **Connect,** служба Database Engine подключается к указанному серверу. В результате появляется окно среды SQL Server Management Studio.

1.3 Администрирование сервера баз данных

Задачи администрирования, которые можно выполнить с помощью Management Studio:

- Регистрация сервера.
- Подключение к серверу.
- Создание серверных групп. Запуск и остановка сервера.

Подключение к серверу

SQL Server Management Studio также разделяет задачи регистрации сервера и подключение к нему. В результате для подключения к серверу из окна **Object Explorer** необходимо щелкнуть правой кнопкой мыши на имени сервера и выбрать пункт **Connect** (подключение) из контекстного меню (рисунок 1.2).

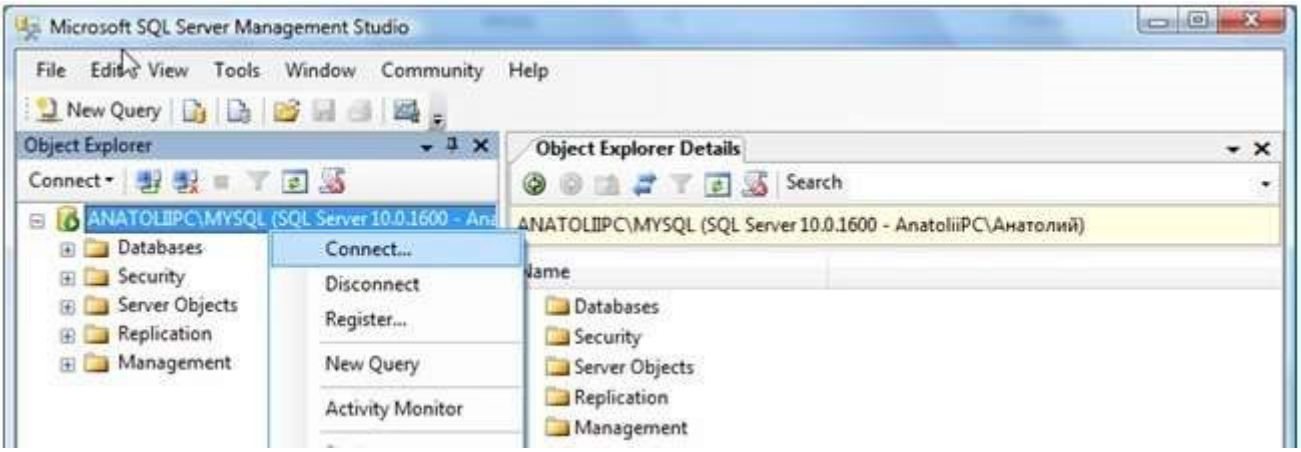


Рисунок 1.2 – Подключение к серверу

Свойства сервера

Для управления свойствами сервера и его конфигурацией необходимо выбрать правым кликом на имени сервера в окне **Object Explorer** пункт меню **Properties** (свойства). В результате появится диалоговое окно **Server Properties** (рисунок 1.3), содержащее несколько вкладок: **General** (Общие свойства), **Security** (Безопасность), **Permissions** (Разрешения) и другие. Вкладка **General** содержит общие свойства сервера; вкладка **Security** содержит информацию об аутентификации и аудите пользователей; вкладка **Permissions** показывает всех пользователей и роли, которые открывают доступ к серверу, также здесь можно определить различные разрешения на пользование базой данных и соотнести их с конкретным пользователем (именем входа) или ролью (пользовательской или системной).

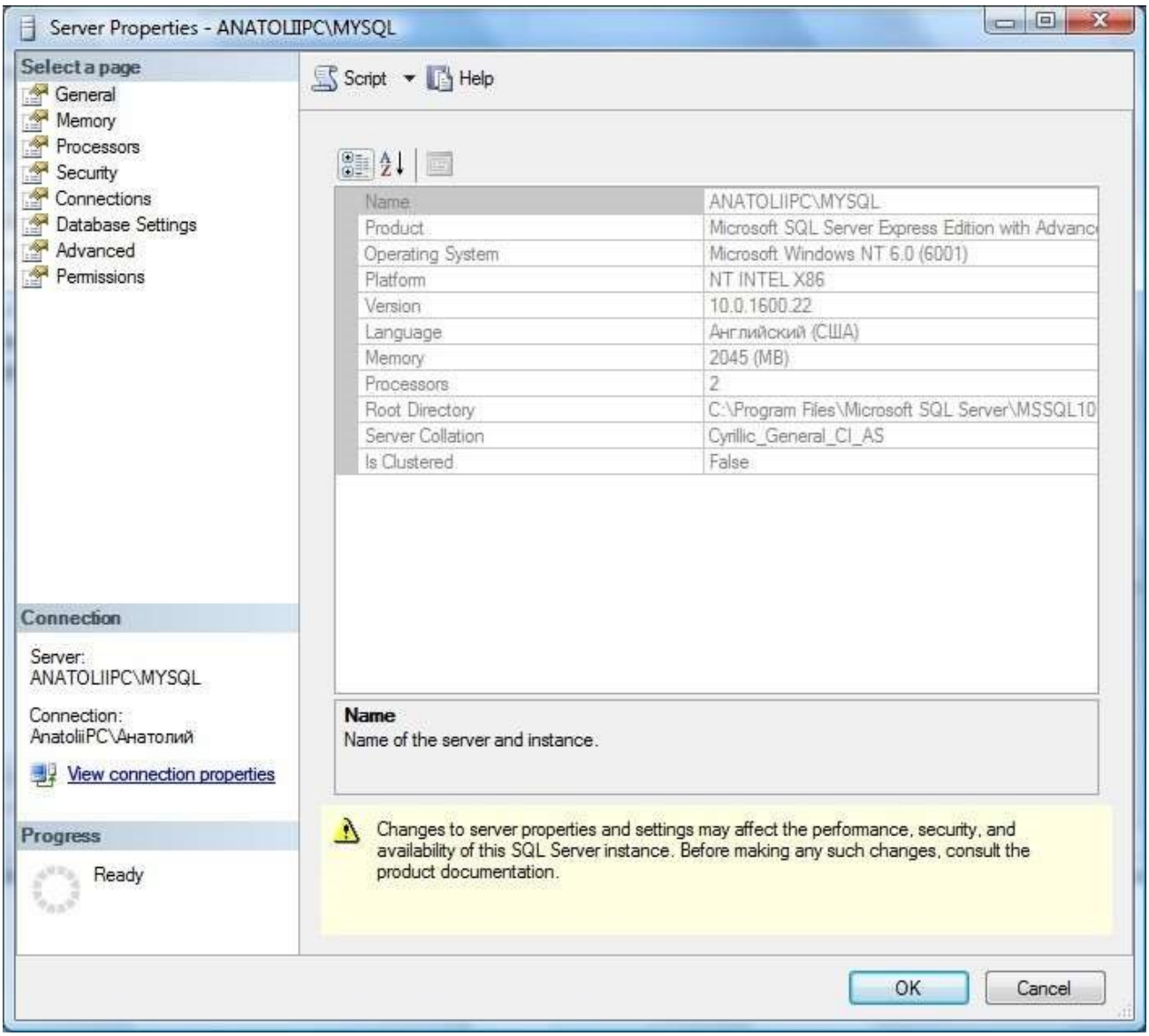


Рисунок 1.3 – Диалоговое окно Server Properties

Запуск и остановка сервера

Экземпляр БД может запускаться автоматически каждый раз при старте операционной системы или используя SQL Server Management Studio. Для того чтобы запустить сервер из Management Studio необходимо выбрать пункт контекстного меню **Start** (Пуск) выбранного в окне **Object Explorer** сервера. Контекстное меню также содержит пункты **Stop** (Остановить) и **Pause** (Приостановить), которые позволяют останавливать работу сервера или прекратить на время соответственно.

1.4 Управление базами данных в SQL Server Management Studio

Используя SQL Server Management Studio можно выполнить следующие задачи:

- Создание БД (без использования Transact-SQL).
- Изменение БД (без использования Transact-SQL).
- Управление объектами БД и их использованием.
- Генерация и выполнение SQL команд.

Создание БД без использования Transact-SQL

Создать новую БД можно с помощью окна **Object Explorer** или используя язык **Transact-SQL**. С помощью окна Object Explorer мы можем управлять всеми объектами БД сервера. Дерево папок окна Object Explorer содержит несколько объектов, включая **Databases**. Эта папка содержит несколько подпапок: одна для системных БД – System Databases, другие папки представляют отдельные БД, созданные пользователем.

Для того, чтобы создать новую БД, используя Object Explorer, щелкните правой кнопкой мыши на объекте Databases и выберите пункт меню **New Database**. В появившемся диалоговом окне **New Database** задайте имя создаваемой БД в поле **Database Name** и нажмите кнопку **OK**.

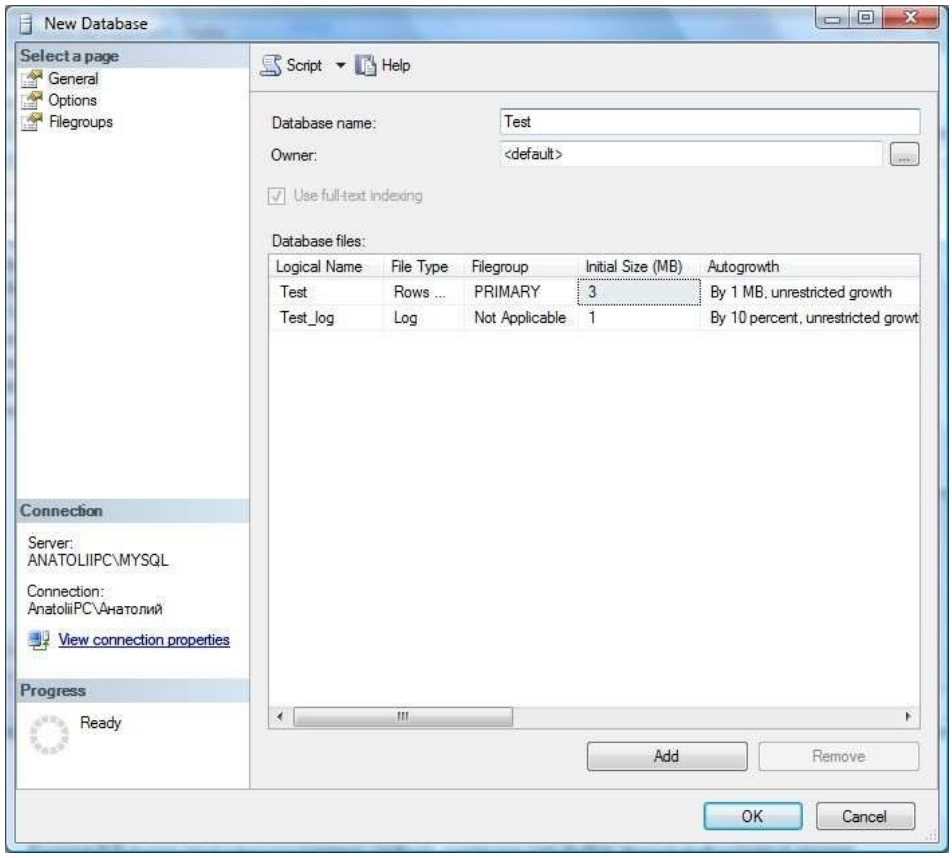


Рисунок 1.4 – Диалоговое окно **New Database**

Для удаления БД из системы необходимо выбрать пункт Delete из контекстного меню окна Object Explorer, предварительно выбрав удаляемую БД из списка.

Каждая БД имеет несколько различных свойств, такие как тип файла, начальный размер и другие. Свойства БД могут быть выбраны в левой панели диалогового окна **New Database,** где расположены несколько страниц или групп свойств (рисунок 1.4):

- **General**
- **Files** (появляется только в уже созданной БД)
- **Filegroups Options**
- **Permissions** (появляется только в уже созданной БД)
- **Extended Properties** (появляется только в уже созданной БД)
- **Mirroring** (появляется только в уже созданной БД)
- **Transaction Log Shipping** (появляется только в уже созданной БД)

Группа свойств **General** диалогового окна **New Database** содержит, кроме всех прочих, имя БД, владельца БД, а также свойства файлов БД. Свойства файлов БД включают в себя имя (**FileName**), путь (**Path**) к файлу (то место, где будет храниться БД), начальный размер (**Initial size**), тип файла (**Type**) – Data (файл данных) или Log (файл журнала транзакций), файловую группу (**FileGroup**), а также возможность задания автоматического увеличения размера файла данных или журнала транзакций при достижении лимита использования (**Autogrowth**).

База данных может состоять из нескольких файлов, а файлы, в свою очередь, могут принадлежать различным файловым группам (или файловой группе по умолчанию – **PRIMARY**).

Конфигурирование файлов журнала и файлов данных

В базах данных SQL Server используется два типа файлов операционной системы: файлы данных и файлы журнала. *Файлы данных* (data files) содержат данные и объекты, такие как таблицы и индексы, а в *файлах журнала* (log files) хранится журнал транзакций для восстановления транзакций базы данных. Для упрощения администрирования и повышения производительности файлы данных можно объединять в файловые группы.

Файлы данных. В базе данных SQL Server можно создавать файлы данных двух типов.

- *Первичный файл данных* (primary data file) является обязательным. В нем хранится загрузочная информация каталога базы данных и указатели на другие файлы базы данных.

Первичный файл данных может также содержать объекты и пользовательские данные. Для имени первичного файла рекомендуется расширение mdf.

- *Вторичные файлы данных* (secondary data files) не являются обязательными и определяются пользователем. В них содержатся объекты и пользовательские данные. Для повышения производительности вторичные файлы рекомендуется размещать на разных дисках. В базе данных может быть не более 32 766 вторичных файлов данных. Для имени вторичного файла данных рекомендуется расширение ndf.

Все данные и объекты рекомендуется хранить во вторичных файлах, а каталог базы данных - в первичном файле. Такая конфигурации помогает снизить вероятность конфликтов доступа к дискам.

Файловые группы

Файловая группа (filegroup) — это логическая структура, с помощью которой администратор баз данных может группировать файлы данных и управлять ими как логической единицей. Чтобы повысить производительность, объекты базы данных (например, таблицы) можно выделить в отдельные файловые группы. Распределение объектов базы данных по разным файловым группам позволяет реализовать преимущества разных дисковых подсистем и разрешить SQL Server выполнять параллельные дисковые операции. Кроме того, создав несколько файловых групп, вы сможете выполнять резервное копирование и восстановление файлов независимо друг от друга.

В SQL Server поддерживается два типа файловых групп: первичные и пользовательские.

- *Первичная файловая группа* (primary filegroup) содержит первичный файл данных и все вторичные файлы данных, не входящие ни в какую другую файловую группу. Все системные таблицы помещаются в первичную файловую группу.
- В каждой базе данных есть *файловая группа по умолчанию* (default filegroup). Если при создании объекта базы данных не указать файловую группу. SQL Server поместит его в файловую группу по умолчанию.
- Файловой группе можно присвоить атрибут «только для чтения». *Файловые группы только для чтения* (read-only filegroups) можно использовать для объектов базы данных, которые нельзя изменять, например хронологических таблиц. Все файловые группы, за исключением первичной, можно отметить как поддерживающие только чтение.

Процесс конфигурирования файлов данных и файлов журнала

С помощью среды SQL Server Management Studio можно добавлять файлы данных и файлы журнала транзакций, а также задавать их свойства и изменять динамические параметры роста размера файлов при достижении ими предельных размеров. Для добавления нового файла данных (журнала транзакций), выберите БД в окне **Object Explorer** и откройте диалоговое окно свойств (пункт **Properties** из контекстного меню), перейдите на страницу свойств **Files** (рисунок 1.5). Для добавления нового файла в БД необходимо нажать кнопку **Add** в правом нижнем углу.

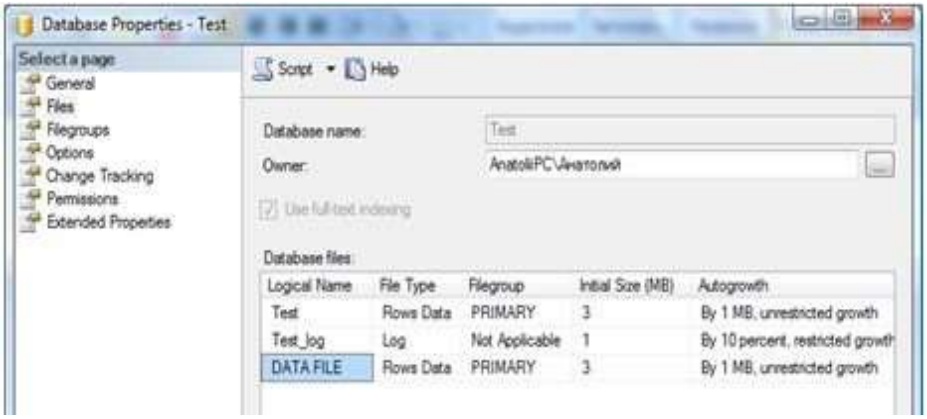


Рисунок 1.5 – Создание нового файла данных

После добавления файла можно назначить ему файловую группу из списка имеющихся (выпадающий список в столбце **Filegroup**). Здесь же можно определить первоначальный размер файла в столбце **Initial Size** (в MB). Для управления свойствами роста размера файла при его заполнении необходимо открыть диалоговое окно нажатием кнопки в столбце **Autogrowth** (рисунок 1.6).

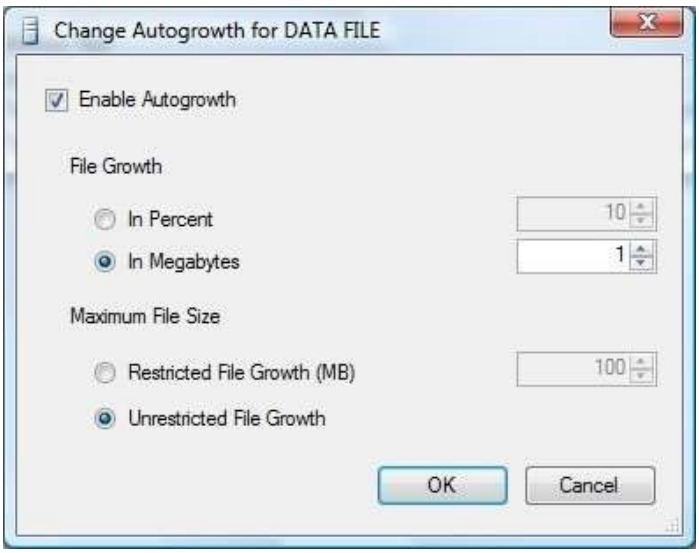


Рисунок 1.6 – Изменение свойств файла данных

Отключение и подключение баз данных

Если дисковое пространство и доступность исходной базы данных не критичны, для переноса базы данных с одного сервера на другой её обычно отключают на старом экземпляре и подключают на новом. И хотя при этом пользователи не смогут работать с базой данных, такой способ переноса имеет преимущество по надежности: при возникновении непредвиденных проблем администратор баз данных всегда сможет заново подключить копию файла базы данных к исходному экземпляру SQL Server.

Этот метод часто обходят вниманием, несмотря на то, что он является простейшим. Он предполагает отключение базы данных, копирование файлов на сервер назначения и подключение на этом сервере скопированных файлов базы данных.

Практический опыт подсказывает, что прежде чем подключать файл базы данных на новом экземпляре, следует создать его копию (на случай восстановления). После подключения файла базы данных его нельзя будет использовать в предыдущих версиях SQL Server.

Отключение и последующее подключение баз данных приводит к потере всех учетных записей пользователей, ролей и разрешений системы безопасности — на сервере назначения их придется создавать заново. Лучше всего скоординировать систему безопасности с администратором локальной сети и использовать сконфигурированные им группы. Если серверы источника и назначения принадлежат к одной и той же группе системы безопасности сети, это позволит в большинстве случаев обойти вопросы, связанные с регистрацией.

В Management Studio щелкните правой кнопкой мыши на копируемой базе данных и выберите в контекстном меню пункт **Tasks -> Detach** для отсоединения БД.

После отключения файла базы данных он исчезает из списка баз данных Management Studio. Теперь файлы базы данных можно копировать и перемещать так же, как обычные файлы.

Чтобы снова подключить базу данных, выделите в дереве консоли Management Studio узел **Databases** и выберите пункт **Tasks -> Attach** из контекстного меню или из меню **Action** программы. Диалоговое окно подключения базы данных предлагает выбрать файл данных и проверить его место размещения и имя.

1.5 Конфигурирование участников системы безопасности сервера

SQL Server поддерживает строгую модель безопасности, позволяющую предотвратить неавторизованный доступ к важным источникам данных. Эта модель основана на разрешениях,

которые назначаются участникам системы безопасности — пользователям, группам и процессам, запрашивающим ресурсы SQL Server. SQL Server проверяет подлинность разрешений всех пользовательских соединений, поэтому для каждого такого соединения должны быть указаны режим проверки подлинности и учетные данные. Есть два режима проверки подлинности: *проверка подлинности Windows* (Windows authentication) и *смешанный режим* (Mixed Mode). Эти режимы управляют тем, как пользователи приложений устанавливают соединение с SQL Server. Также есть два типа имен входа SQL Server: *имена входа Windows* (Windows logins) и *имена входа SQL Server* (SQL Server logins), управляющие доступом к экземпляру SQL Server. Чтобы упростить управление именами входа участников системы безопасности, имеющих права администратора SQL Server, их можно объединить в *фиксированные серверные роли* (fixed server roles). Режим проверки подлинности и имена входа являются первым уровнем защиты SQL Server, поэтому следует позаботиться о выборе наиболее безопасных параметров для вашей среды.

Система безопасности SQL Server основана на концепции *защищаемых объектов (*securables*),* т.е. объектов, на которые можно назначать разрешения, и *принципалов (*principles*),* т.е. объектов, которым можно назначать разрешения. Принципалами могут быть пользователи и роли. Пользователи назначаются ролям; при этом как пользователям, так и ролям могут предоставляться разрешения на доступ к объектам (рисунок 1.7). Каждый объект имеет своего владельца, и права собственности также влияют на разрешения.

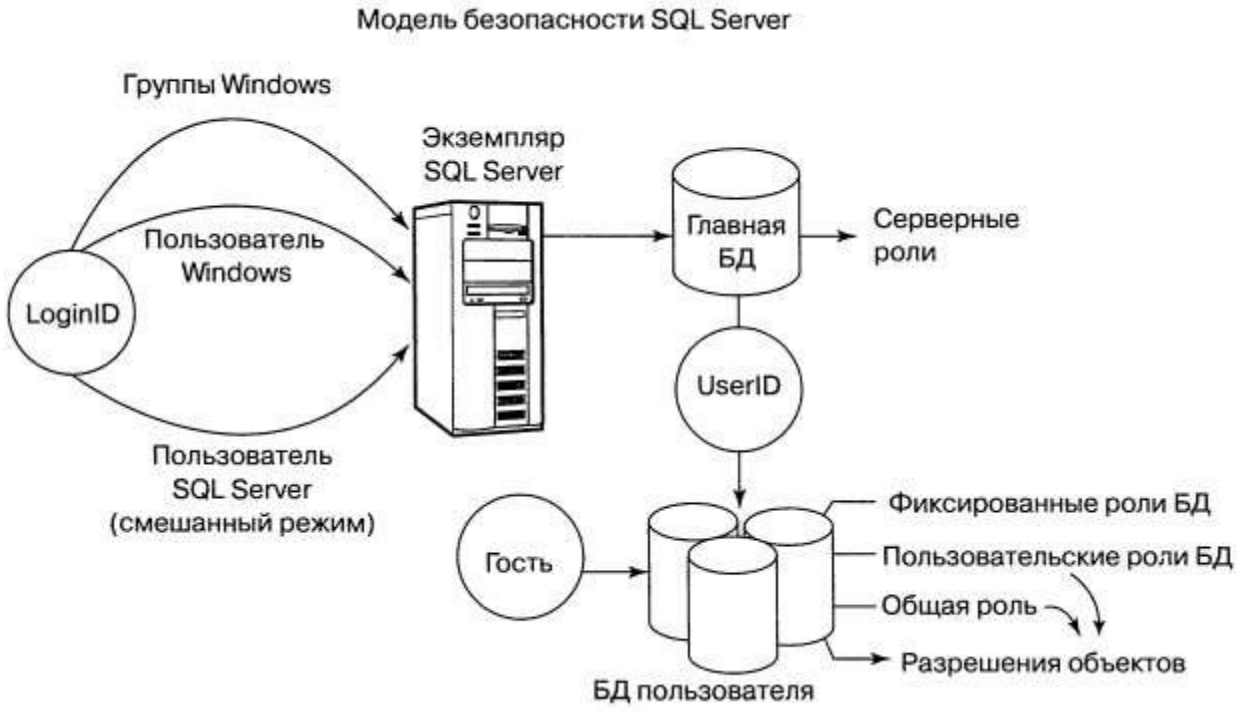


Рисунок 1.7 – Обобщенная схема системы безопасности SQL Server

Как только пользователь был представлен серверу и идентифицирован, он получает те права, которые были назначены его фиксированной серверной роли администратором. Если пользователь принадлежит роли **sysadmin**, то он имеет полный доступ ко всем функциям сервера, а также ко всем базам данных и объектам на сервере. Пользователю может быть разрешен доступ к базе данных, и его идентификатор отображается на специфический для базы данных идентификатор пользователя (**UserlD**). Если пользователю доступ к базе данных явно не разрешен, он может обратиться к ней как гость (если в конфигурацию сервера предварительно внести некоторые изменения).

Выбор режима проверки подлинности

Как уже было отмечено, в SQL Server поддерживается два режима проверки подлинности для доступа к ресурсам базы данных: проверка подлинности Windows и смешанный режим:

- **Проверка подлинности Windows.** В этом режиме проверки подлинности только пользователи, прошедшие проверку подлинности Windows, могут получить доступ к экземпляру SQL Server. Для каждого пользователя (или группы) Windows, которому нужен доступ к экземпляру SQL Server, необходимо добавить имя входа Windows.
  Проверка подлинности Windows является режимом по умолчанию.
- **Смешанный режим.** В этом режиме проверки подлинности доступ к экземпляру SQL Server могут получить как имена входа Windows, так и имена входа SQL Server (ни один из которых не связан с пользователем операционной системы). Смешанный режим используется в том случае, когда необходимо предоставить доступ к SQL Server пользователям клиентской операционной системы, отличной от Windows.

Режим проверки подлинности можно изменить в диалоговом окне Свойства сервера (**Server Properties –** рисунок 1.8) в SQL Server Management Studio, выполнив следующие действия:

a) В SQL Server Management Studio выберите правым кликом нужный сервер и выберите Свойства (**Properties**).
b) Выберите вкладку Безопасность (**Security**).
c) В разделе Серверная проверка подлинности (**Server Authentication**) выберите режим проверки подлинности для сервера: Проверка подлинности Windows (**Windows authentication mode**) или Проверка подлинности SQL Server и Windows (**SQL Server And Windows authentication mode**).
d) Сохраните изменения, щелкнув **ОК**.

e) Щелкните **OK**, чтобы закрыть окно с сообщением о том, что изменения вступят в силу только после перезапуска SQL Server.

f) Чтобы перезапустить сервер, в обозревателе объектов (**Object Explorer**) щелкните правой кнопкой ваш сервер и выберите Перезапустить (**Restart**).

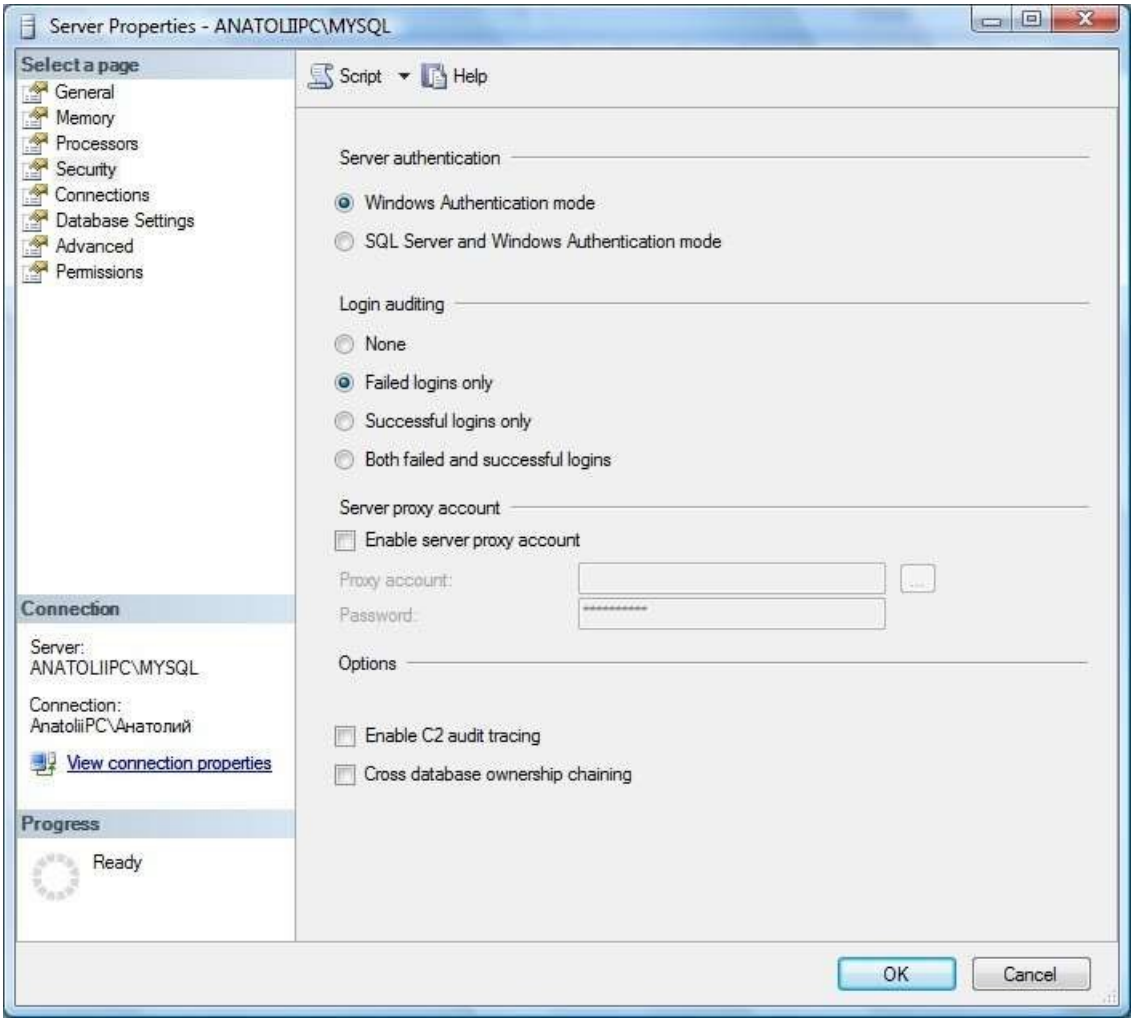


Рисунок 1.8 – Окно выбора аутентификации SQL Server

Конфигурирование имён входа SQL Server

Создание имен входа Windows. Если учетная запись пользователя уже существует в списке пользователей компьютера или домена Windows, то SQL Server может его распознать. Чтобы добавить новую учетную запись для SQL Server с помощью Object Explorer, выполните следующие действия:

a) Откройте узел **Security -> Logins** сервера и выберите в контекстном меню пункт **New Login**.

b) На странице **General** диалогового окна Login - New (рисунок 1.9) с помощью кнопки **Search** найдите пользователя Windows.

c) Вы также имеете возможность ввести имя пользователя или группы вручную (используя формат [Домен\Пользователь]) или воспользоваться кнопкой **Advanced** для выполнения расширенного поиска.

Пользователю может быть назначена база данных, к которой он подключается по умолчанию. Ее имя выбирается в раскрывающемся списке **Default Database** в нижней части окна. Следует отметить, что назначение базы данных по умолчанию никак не связано с назначением для доступа к ней определенных прав. Права доступа к базам данных назначаются во вкладке **Database Access**.

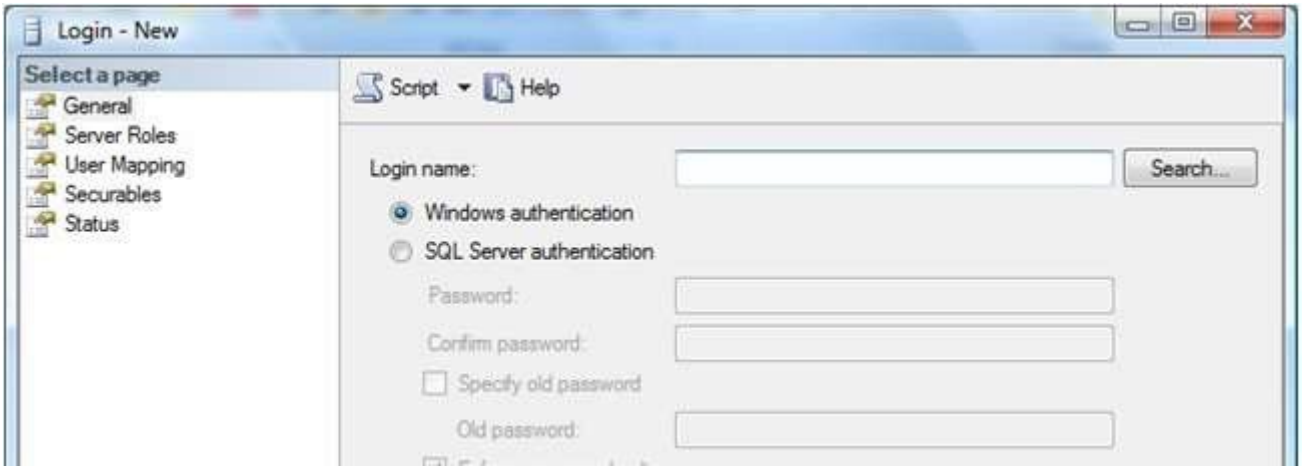


Рисунок 1.9 – Вкладка General диалогового окна Login - New

Окно **Login - New** также используется для управления существующими пользователями. Для открытия версии этого окна, регулирующего разрешения существующего пользователя, дважды щелкните на его имени в узле **Security -> Logins** или щелкните правой кнопкой мыши и выберите в контекстном меню пункт **Properties**.

Удаление учетной записи Windows

Удаление учетной записи Windows из SQL Server достаточно просто выполняется в Management Studio. Выберите эту учетную запись в Object Browser и с помощью контекстного меню удалите ее (т.е. выполните команду Delete). Разумеется, эта операция не удаляет данную учетную запись из операционной системы — только из сервера баз данных.

Серверные роли

SQL Server содержит только фиксированные, предопределенные серверные роли. В основном эти роли открывают доступ к административным задачам, связанным с сервером баз данных. Один пользователь может принадлежать к нескольким ролям. Следующие роли лучше всего использовать для выполнения конкретных задач администрирования:

- Члены роли **Bulk admin** могут выполнять операции массовой вставки.
- Члены роли **Dbcreators** могут создавать, изменять, удалять и восстанавливать базы данных.
- Члены роли **Diskadmin** могут создавать, изменять и удалять файлы на диске.
- Члены роли **Processadmin** могут прерывать запущенный процесс SQL Server.
- Члены роли **Securityadmin** могут управлять регистрационными записями сервера.
- Члены роли **Serveradmin** могут конфигурировать параметры сервера, включая настройки полнотекстового поиска и останов сервера.
- Члены роли **Setupadmin** могут конфигурировать связанные серверы, расширенные хранимые процедуры и процедуры автозапуска.
- Члены роли **Sysadmin** могут осуществлять любую деятельность в инсталляции SQL Server, независимо от наличия других разрешений. Права роли sysadmin даже замещают собой запрет доступа к отдельным объектам.

SQL Server автоматически создает пользователя **BUILTINS/Administrators**, который включает всех пользователей Windows в группу Windows Admins и назначает этой группе роль на сервере баз данных **sysadmin**. Пользователя BUILTINS/Administrators лучше удалить или изменить по своему усмотрению. Серверные роли настраиваются в Management Studio на странице **Server Roles** диалогового окна **Login Properties** (рисунок 1.10).

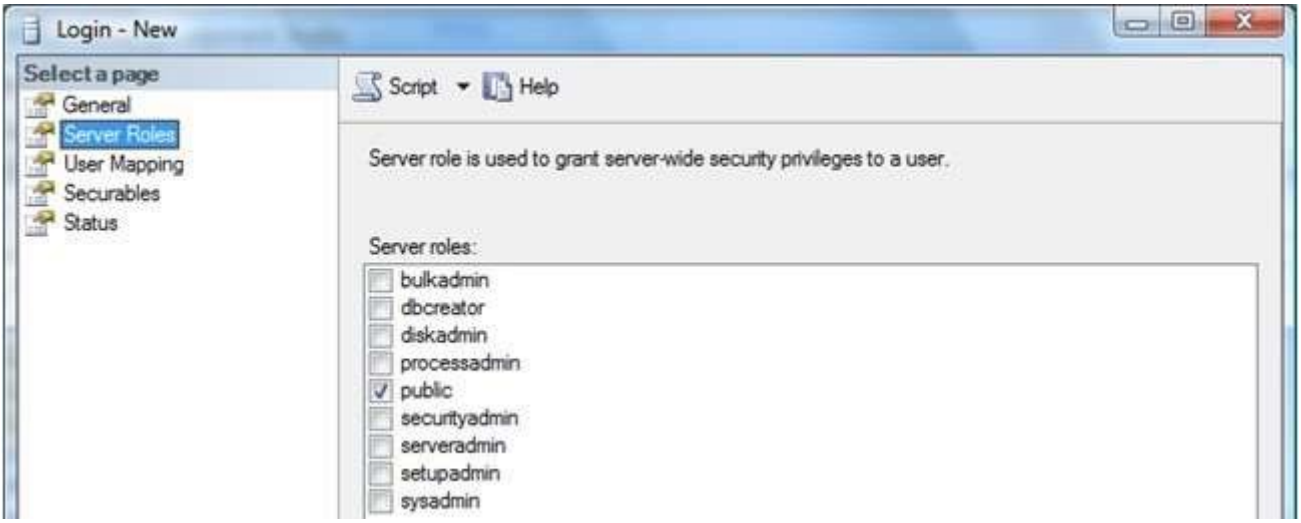


Рисунок 1.10 – Страница Server Roles.

Безопасность на уровне баз данных

После получения доступа к серверу пользователь может получить доступ к отдельным базам данных. Система безопасности баз данных относительно сложная.

Изначально доступ к базе данных устанавливается либо путем добавления базы данных пользователю, либо добавления пользователя базе данных.

Предоставление доступа к базе данных

Доступ пользователям к любой базе данных должен открываться в явном виде. Так как эта схема представляет собой отношение «многие ко многим», доступом к базе можно управлять как на уровне регистрационных записей, так и на уровне баз данных. Если некоторой учетной записи открыт доступ к базе данных, то ей может быть назначена учетная запись базы. Если её имя совпадает с именем регистрационной записи сервера, то ей можно назначить другое имя — оно будет служить чем-то вроде псевдонима регистрационной записи, который распознает база данных.

Для предоставления доступа к базе данных на уровне регистрационных записей в Object Explorer используется страница **User Mapping** диалогового окна **Login Properties** (рисунок 1.11).

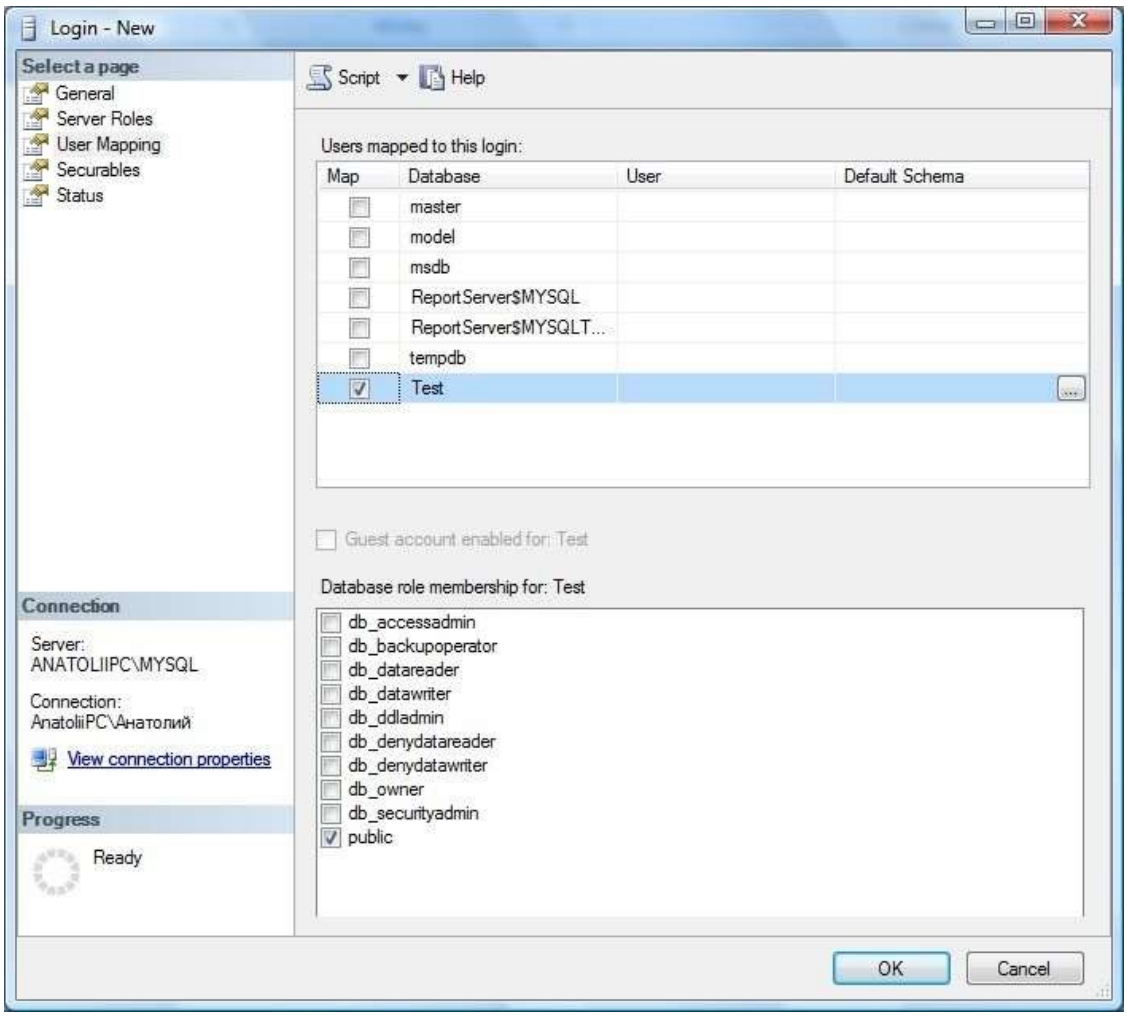


Рисунок 1.11 – Диалоговое окно Login – New, вкладка User Mapping

Для предоставления доступа пользователям со стороны базы данных используйте команду **New User** контекстного меню в узле **Database- > [Database Name] -> Security -> Users**. В поле Login Name открывшегося диалогового окна **Database User - New** (рисунок 1.12) введите имя добавленной регистрационной записи. В поле **User Name** вводится имя пользователя, под которым он будет известен базе данных.

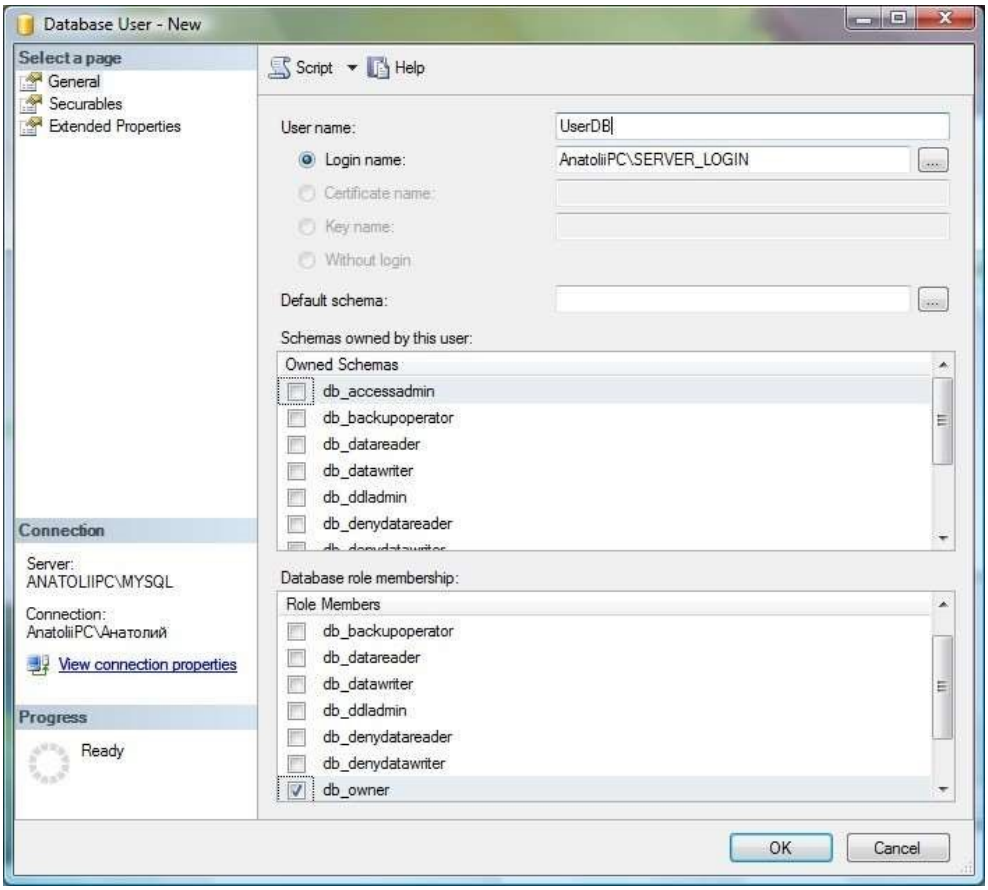


Рисунок 1.12 – Диалоговое окно Database User - New

Стандартные роли базы данных

Стандартные роли базы данных иногда называют пользовательскими. Они могут быть соз - даны любым пользователем сервера, имеющим серверную роль **sysadmin** или роли базы данных **db_owner** или **securityadmins**. Эти роли функционально идентичны группам пользователей в операционной системе Windows. Стандартной роли могут быть предоставлены разрешения и принадлежности другим ролям, и этой роли могут принадлежать пользователи.

Роль public. Роль public является фиксированной, однако, подобно стандартной роли, ей можно назначать разрешения к объектам. Каждый пользователь автоматически становится членом

роли public, и эту связь невозможно удалить. Таким образом, роль public служит своеобразным мерилом минимально допустимых разрешений.

При назначении разрешений роли **public** будьте особо осторожны, поскольку они будут применены ко всем пользователям, за исключением тех, кто принадлежит серверной роли **sysadmin**.

Создание стандартных ролей. Для создания новой стандартной (пользовательской) роли выберите БД из окна **Object Explorer**, перейдите в раздел **Security -> Roles -> Database Roles** и выберите пункт **New Database Role** из контекстного меню. В результате появится диалоговое окно (рисунок 1.13). В поле **Role name** задается имя новой пользовательской роли (**Manager**).

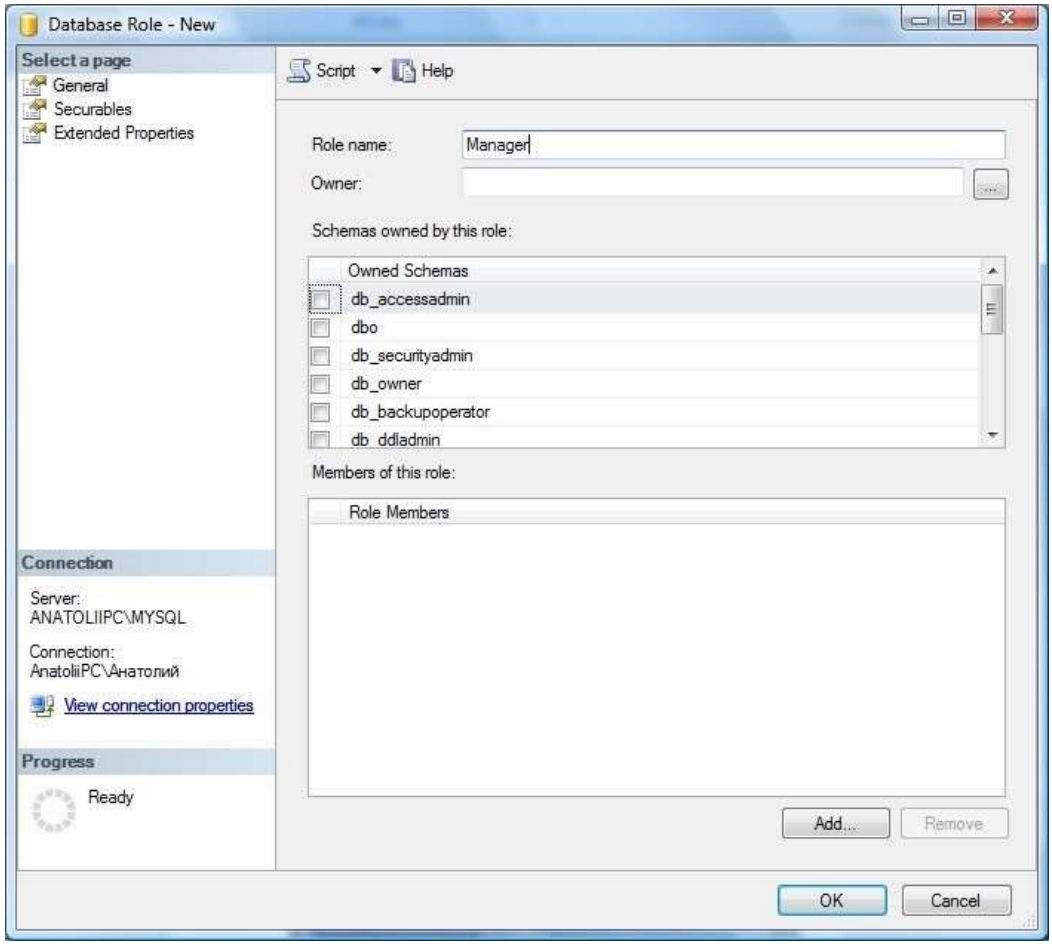


Рисунок 1.13 – Диалоговое окно Database Role – New

Для того чтобы назначить пользователей данной роли в разделе **Members of this role** нажмите кнопку **Add**. В результате нажатия кнопки появиться диалоговое окно **Select Database User or Role** (рисунок 1.14).

Рисунок 1.14 – Диалоговое окно Select Database User or Role

Как упоминалось выше членами пользовательской роли могут быть как пользователи БД так и другие (ранее созданные) роли. Выбрать, какой тип объектов добавлять к роли, можно нажав кнопу **Object Types** (рисунок 1.15).

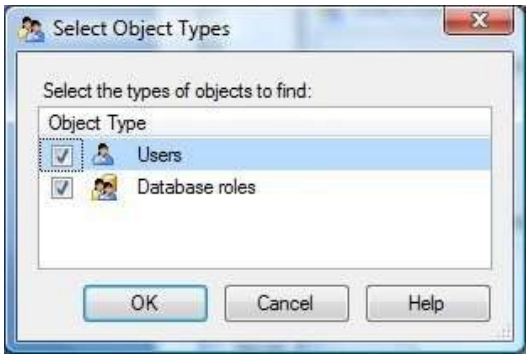


Рисунок 1.15 – Диалоговое окно Select Object Types

После того как тип добавляемых объектов выбран, необходимо нажать кнопку **Browse** (рисунок 1.16). В результате будет получен список, из которого можно выбрать те объекты (роли или пользователи), которые будут входить в данную роль.

Рисунок 1.16 – Диалоговое окно Browse for Objects

Разрешения защищаемых объектов

Пользователю базы данных могут быть предоставлены конкретные *разрешения* к определенным защищаемым объектам. Для этого используется страница **Securables** диалогового окна **Database User**. Способ конфигурирования разрешений зависит от типа объекта. Каждая база данных представляет собой защищаемый объект и имеет ряд специфичных только для неё разрешений. Если у вас нет особых оснований регулировать доступ на уровне отдельных инструкций, лучше всего управлять задачами администрирования базы данных с помощью фиксированных ролей базы данных.

Представим следующий сценарий: членам пользовательской группы Manager запрещено изменять (конфигурировать) параметры пользователя TestUser1.

Для реализации этого сценария нажмите кнопку **Search** из раздела **Securables**. В результате появится диалоговое окно **Add Objects**, с помощью которого можно выбрать типы добавляемых объектов. Выберите пункт **All objects of the types**, чтобы критерий выбора соответствовал объектам БД (рисунок 1.17).

Рисунок 1.17 – Диалоговое окно Add Objects

При нажатии кнопки **OK** отобразиться диалоговое окно **Select Object Types**, в котором перечислен список объектов. Выберите пункт **Users** из списка, т.к. мы собираемся «защитить» именно объекты данного типа.

После подтверждения нашего выбора в диалоговом окне **Database Role Properties** появиться список пользователей БД и возможность назначения отдельных видов разрешений для каждого из них. В нашем случае, мы выбираем пользователя **TestUser1** из списка и в разделе разрешений **Alter** отмечаем пункт разрешения **Deny**, тем самым запрещая членам группы Manager изменять конфигурацию пользователя БД Test **TestUser1** (рисунок 1.18).

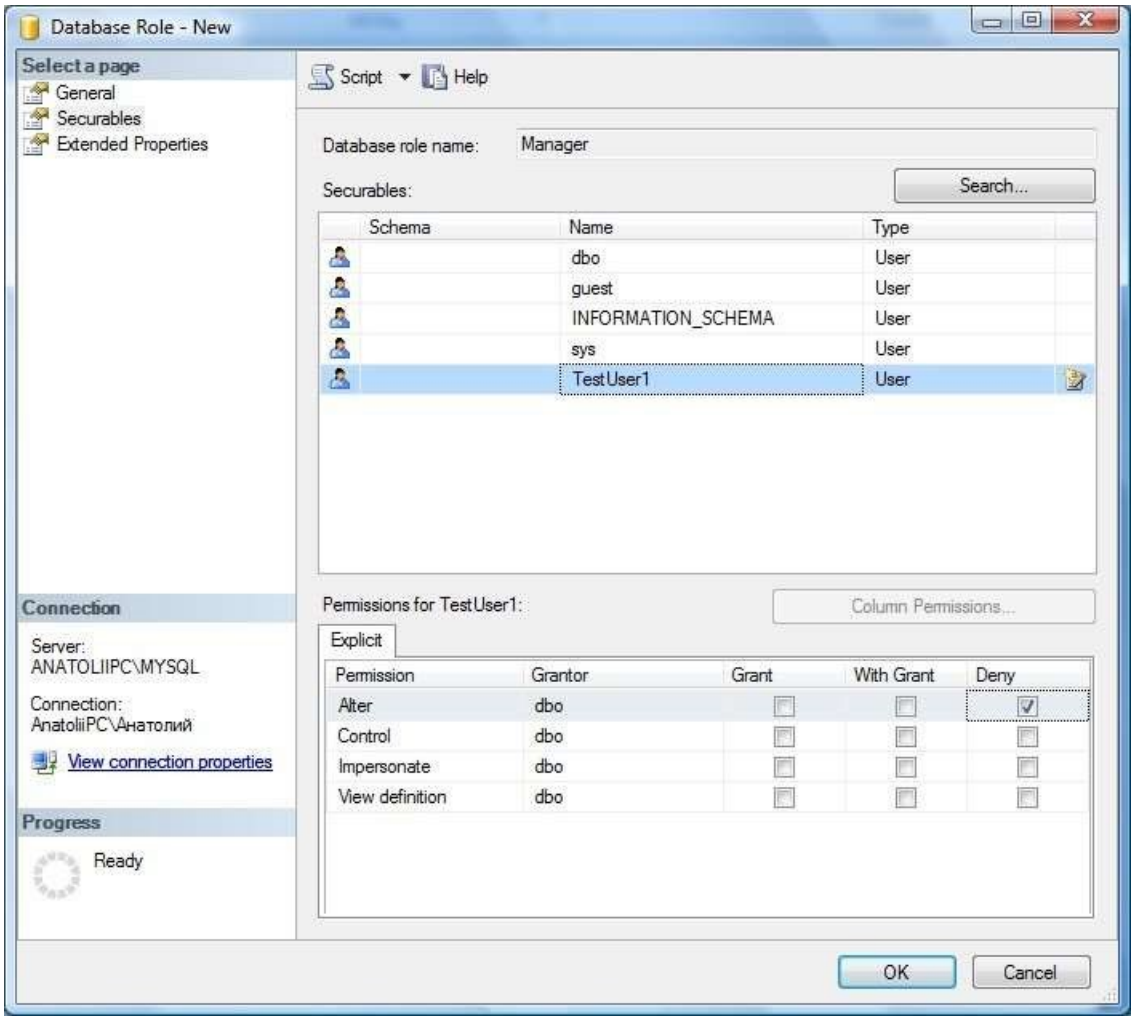


Рисунок 1.18 – Диалоговое окно со свойствами ролей базы данных

1.6 Конфигурирование участников системы безопасности сервера

Для переноса базы данных с одного экземпляра SQL Server на другой в процессе обновления можно создать резервную копию базы данных — это более безопасный метод, позволяющий избежать потери файлов базы данных. Процесс переноса БД:

- Создайте простой *файл резервной копии* (backup file) обычным способом резервного копирования базы данных.
- Переместите файл резервной копии на новый экземпляр SQL Server.
- Восстановите файл резервной копии с помощью процедуры *восстановления базы дан- ных SQL Server,* при необходимости изменив его размещение.

Этот процесс не влияет на доступность исходной базы данных и не подвергает опасности возможность использования её файлов в предыдущей версии SQL Server. Ещё одно преимущество резервного копирования и восстановления состоит в том, что резервные копии обычно меньше исходных файлов базы данных, поскольку резервное копирование затрагивает только данные из базы данных, а не зарезервированное, неиспользуемое пространство. Благодаря уменьшению размера файла передача резервной копии выполняется быстрее, чем исходного файла базы данных.

Однако в процессе обновления необходимо обеспечить дисковое пространство для исходных файлов базы данных, файлов с резервными копиями и файлов новой базы данных.

Для создания полной резервной копии выберите пункт **Tasks -> Back Up…** из контекстного меню раздела **Databses -> [имя БД]**.

В результате появиться диалоговое окно **Back Up Database** (рисунок 1.19), в котором можно выбрать тип резервного копирования **Full** (полное) или **Differential** (выборочное). Также можно выбрать размещение копии (путь на диске) в разделе **Destination**, наименование копии (**Name**), описание (**Description**), количество дней после которых созданная копия данных станет неактуальной (**Back Up Set will expire**). После нажатия кнопки OK резервная копия будет создана в месте, указанном в разделе **Destination**.

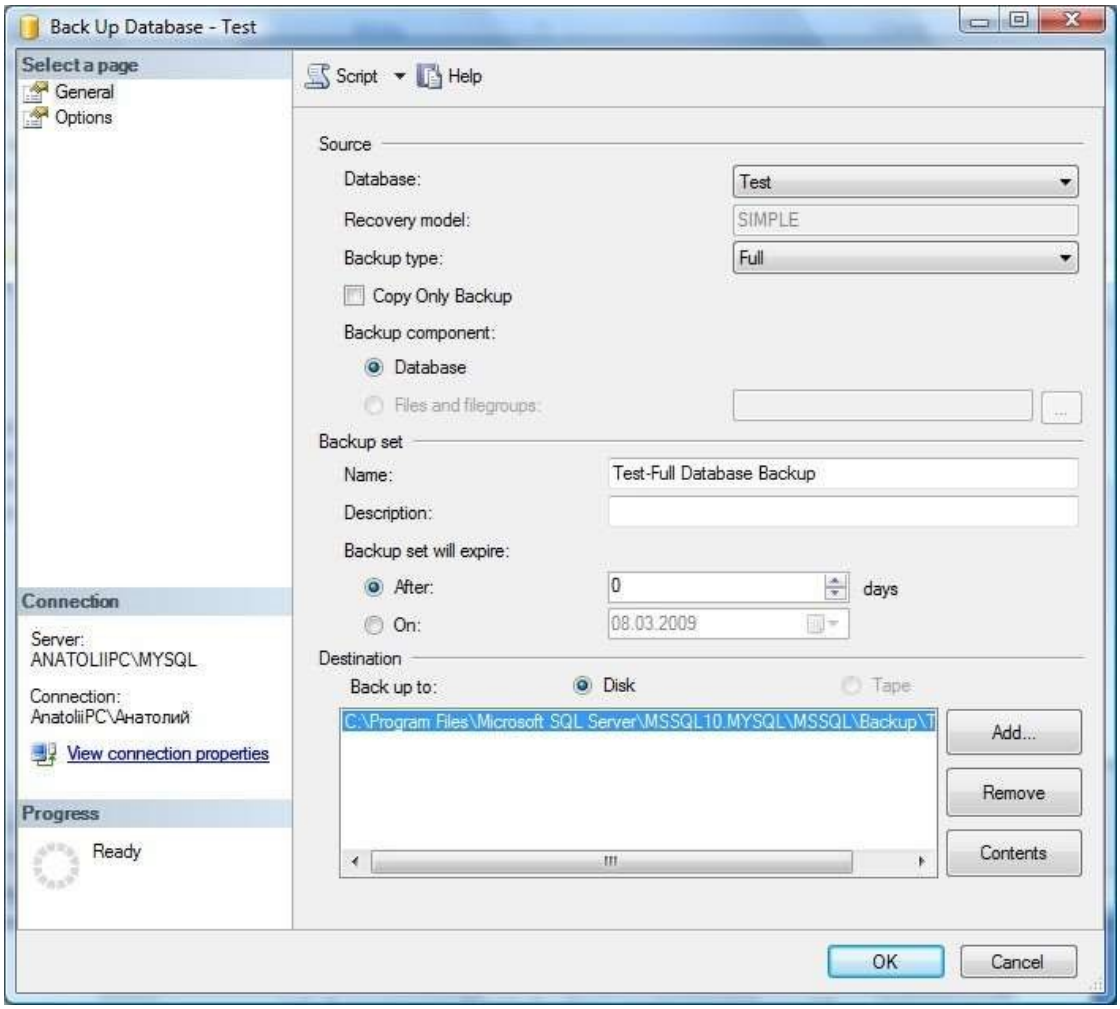


Рисунок 1.19 – Диалоговое окно Back Up Database

Для восстановления резервной копии выберите пункт контекстного меню **Tasks -> Restore -> Database…** из раздела **Databses -> [имя БД]**. В результате появиться диалоговое окно **Restore Database** (рисунок 1.20), где можно указать резервную копию для восстановления (в случае присутствия нескольких), а также БД, куда копия резервных данных будет восстановлена.

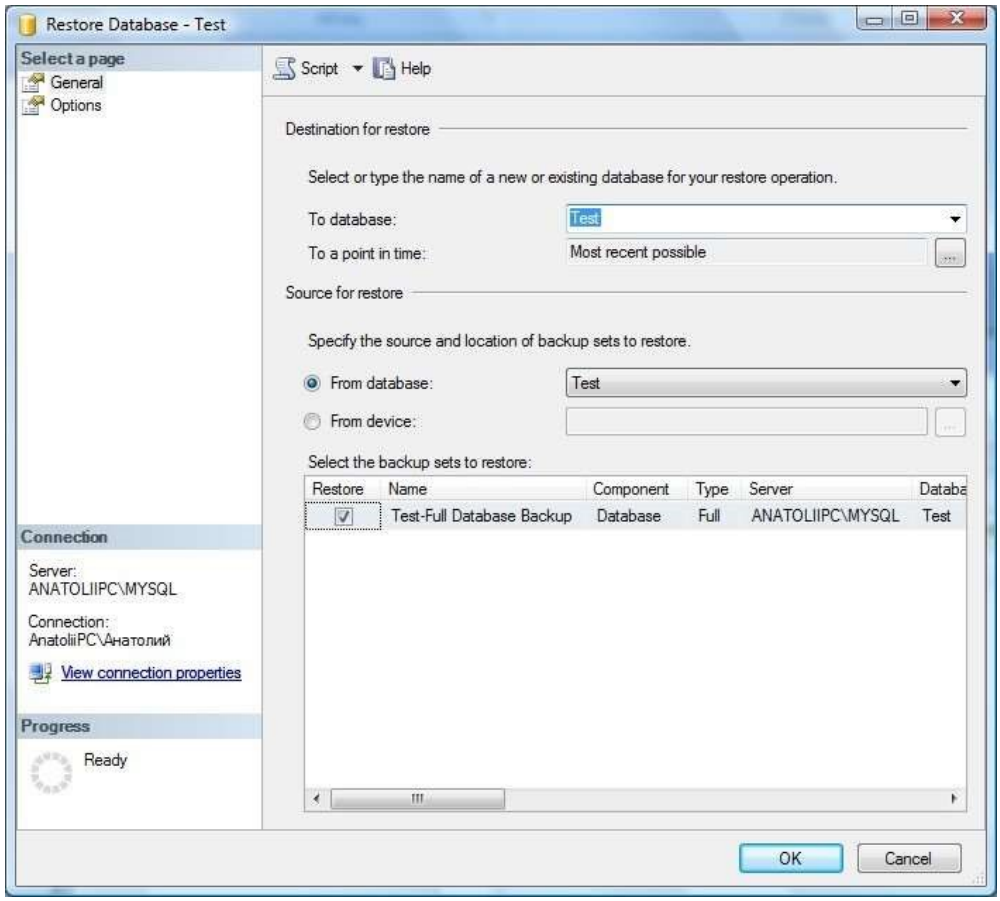


Рисунок 1.20 – Диалоговое окно Restore Database

В общем случае, операция резервного копирования является специфической, так как план резервного копирования зависит от конкретных условий, например, от нагрузки на сервер БД в рабочие дни (в обеденный перерыв) конкретного предприятия. Поэтому резервное копирование тщательно планируется, должно отвечать политике предприятия и обеспечивать надежность БД к сбоям и потерям данных.

Создание файла резервной копии по расписанию

SQL Server (в версии Standard и выше) позволяет осуществлять различные операции над базами данных по расписанию. Данную работу осуществляет компонент SQL Server Agent.

Создадим план по автоматическому созданию резервной копии базы данных ShopBDC каждую неделю (предварительно запустим сервис SQL Server Agent). Для этого открываем папку Management, на папке Maintenance Plans правым кликом выбираем опцию New Maintenance Plan, вводим имя («ShopBDC_backup»).

Появляется окно дизайнера плана (рисунок 1.21).

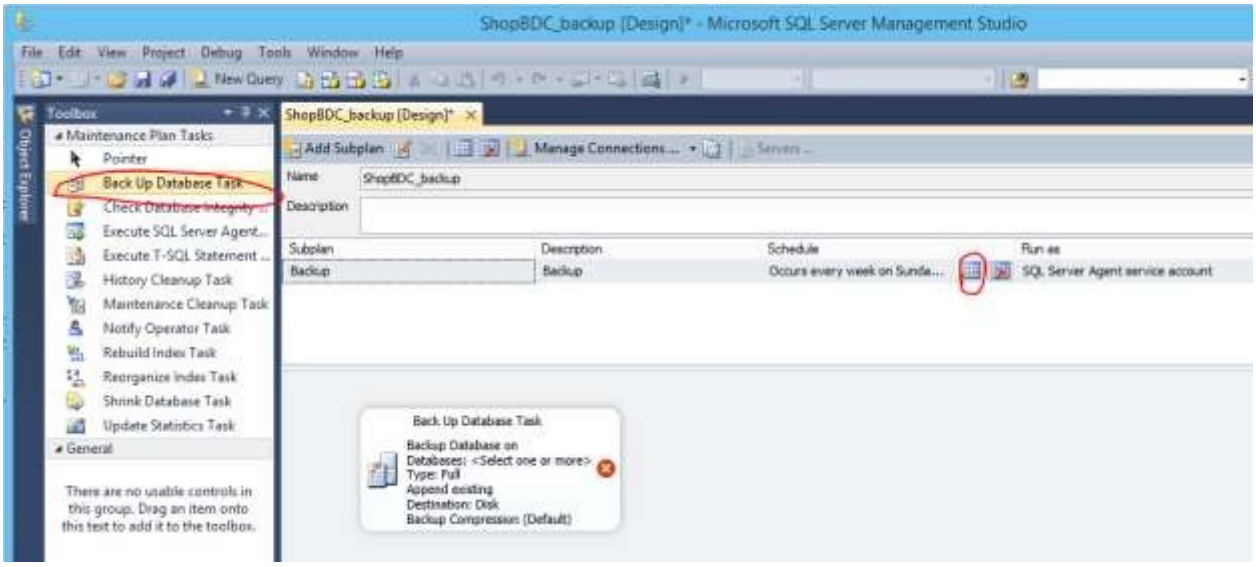


Рисунок 1.21 – Окно дизайнера плана

Далее, как показано на рисунке 1.21 выбираем компонент Back Up Database Task и перетаскиваем его на рабочую поверхность дизайнера. Выше в списке под-планов указываем имя под-плана, а также расписание (время) его выполнения (рисунок 1.22).

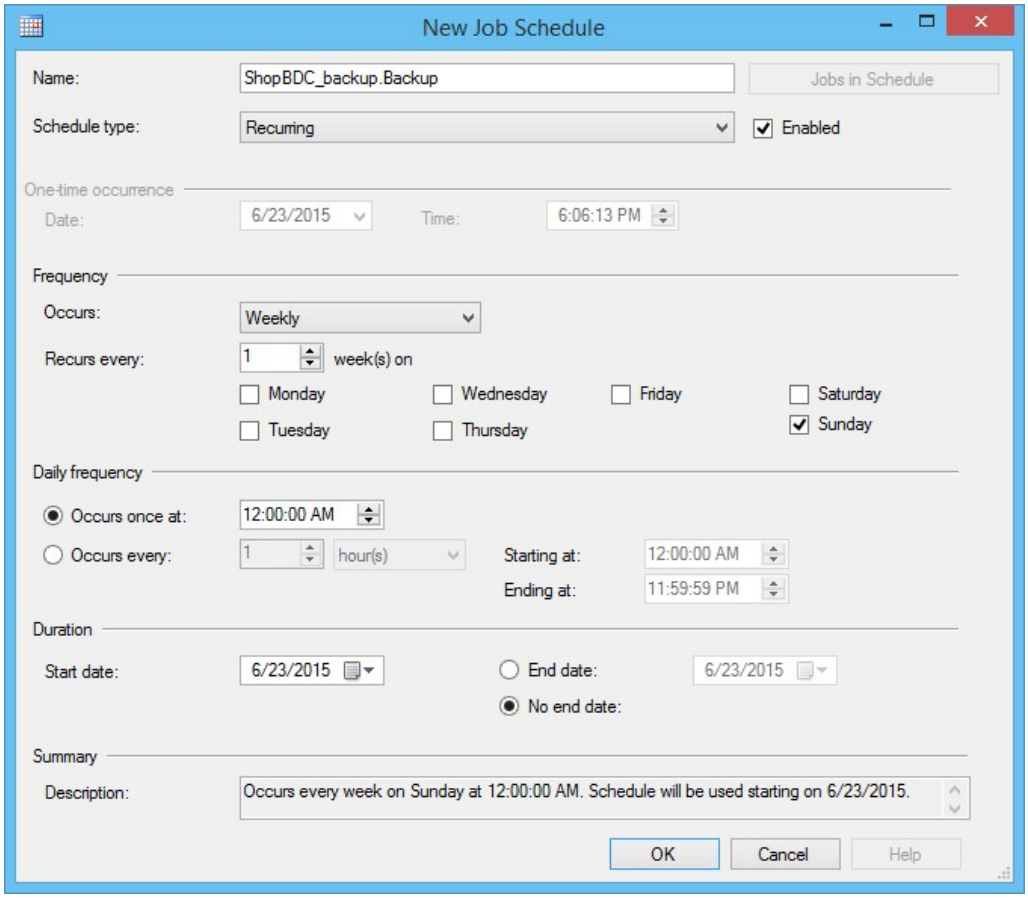


Рисунок 1.22 – Окно задания расписания

Переходим на рабочую поверхность дизайнера и правым кликом на объекте Back Up Database Task выбираем опцию Edit. Во вкладке General указываем базу данных («ShopBDC») как показано на рисунке 1.23.

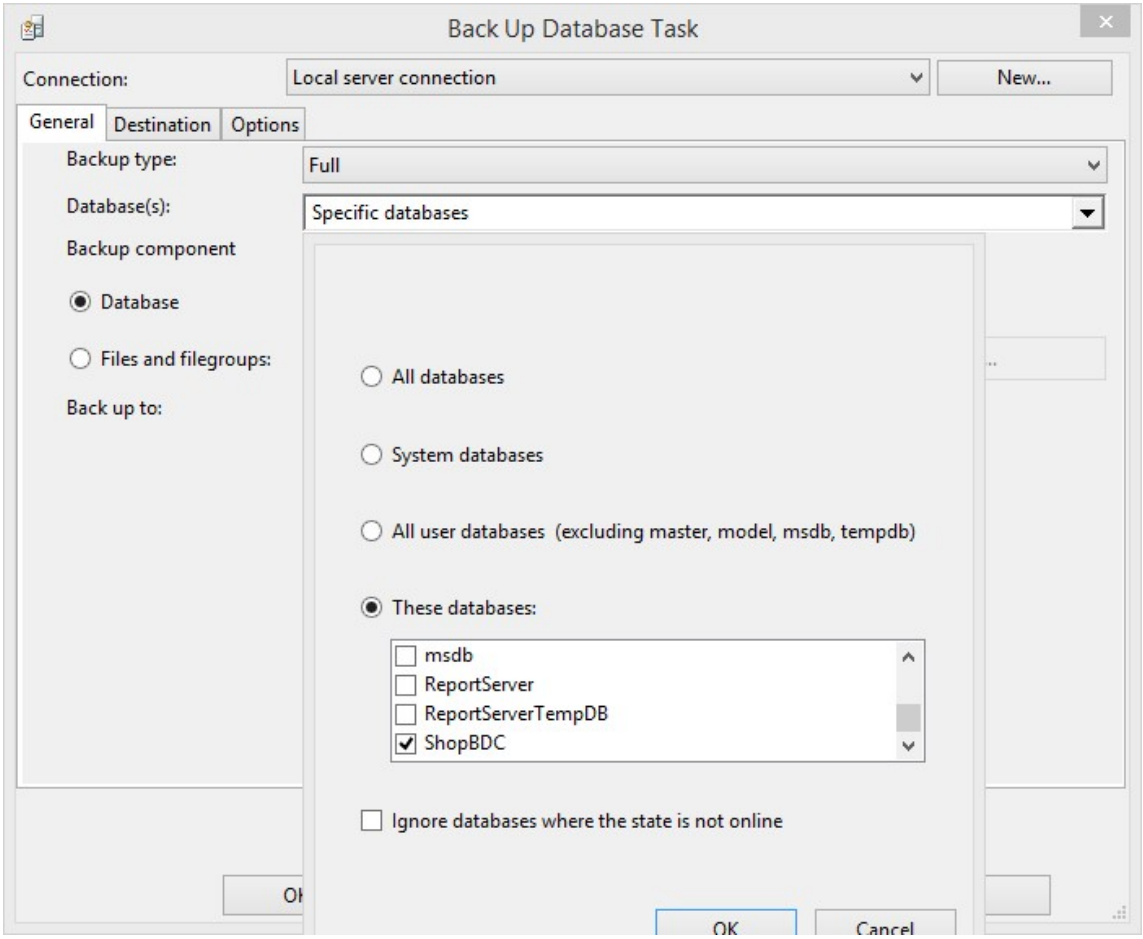


Рисунок 1.23 – Задание опций для плана по резервированию базы данных

Во вкладке Destination указываем путь к папке куда будет сохранён резервный файл базы данных. Нажимаем OK. Сохраняем план. Для немедленного запуска плана на выполнение выбираем опцию ***Execute***. (рисунок 1.24)

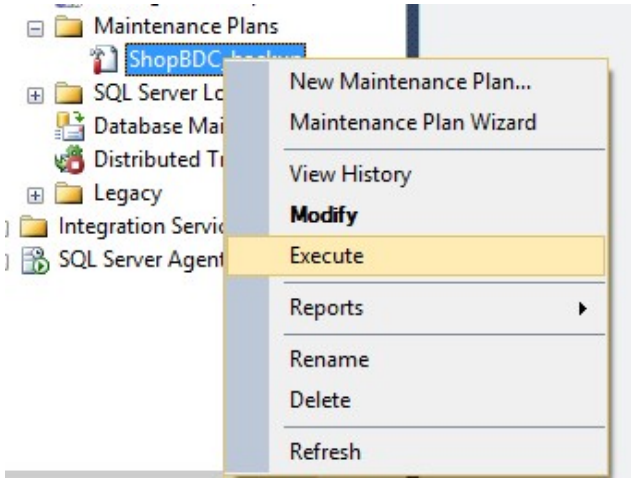


Рисунок 1.24 – Выполнение плана

Отметим, что при каждом запуске плана, старые файлы резервных копий удаляться не будут. Поэтому модифицируем план. Добавим на рабочую поверхность компонент Maintenance Cleanup Task. (Рисунок 1.25)

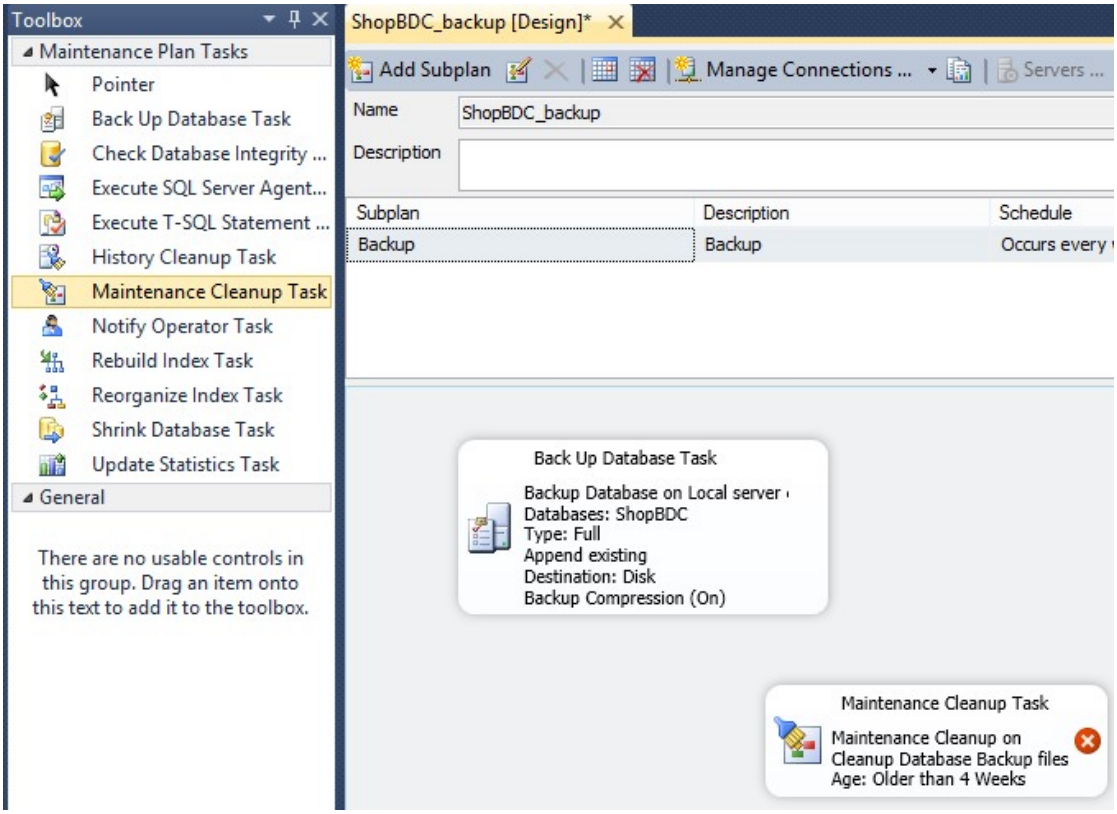


Рисунок 1.25 – Компонент Maintenance Cleanup Task

Настраиваем компонент Maintenance Cleanup Task: указываем папку в которой следует искать резервные файлы для удаления, а также указываем их расширение. Дополнительно указываем: удалять файлы, которые старше, чем 4 недели (рисунок 1.26)

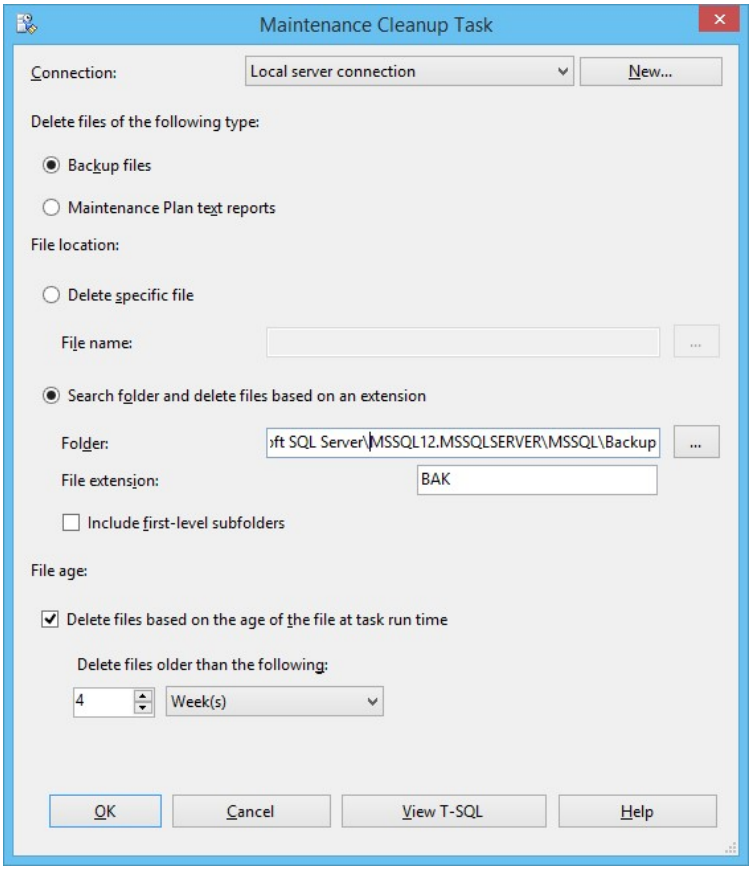


Рисунок 1.26 – Настройка компонента Maintenance Cleanup Task

Соединяем оба компонента зелёной стрелкой, что показывает, что данные компоненты будут задействованы последовательно при выполнении плана (рисунок 1.27). Сохраняем план. Теперь каждую неделю компонент SQL Server Agent будет осуществлять резервное хранение базы данных «ShopBDC» и удалять резервные файлы, которые старше 4х недель.

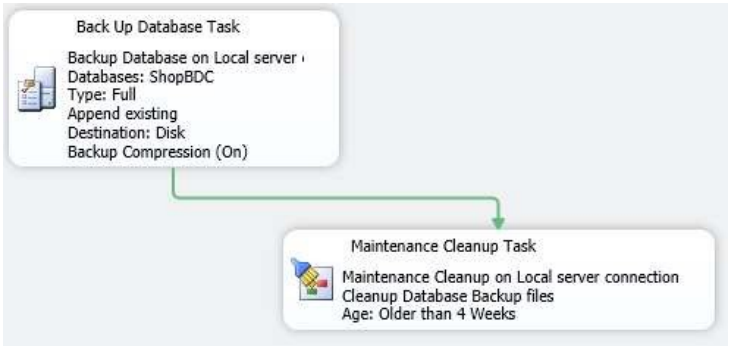


Рисунок 1.27 – Соединение компонентов

Контрольные вопросы:

1. В чем основные преимущества архитектуры «клиент-сервер»?
2. Что такое T-SQL? Чем отличается T-SQL от стандарта ANSI SQL? Каковы его преимущества?
3. Какие основные службы предоставляет SQL Server?

4. Какие режимы проверки подлинности для доступа к данным существуют в SQL Server? Опишите их.
5. Чем отличается имя входа (Login) от пользователя (User) БД?
6. Какой режим проверки подлинности SQL Server является предпочтительным?
7. Каким образом в СУБД SQL Server обеспечивается безопасность БД?
8. В чем преимущество объединения серверов в серверные группы? Как это отражается на производительности серверов БД?
9. Какие возможности предоставляет SQL Server для конфигурирования физического представления БД? Как это может повлиять на производительность и надежность сервера?
10. Какими способами можно осуществить перенос базы данных в SQL Server?

Задания к лабораторной работе:

Задание 0. Установите Microsoft SQL Server Developer Edition

Задание 1. Создание и конфигурация базы данных

Используя SQL Server Management Studio, создайте БД под названием *Test*. Сохраните БД в файле *testdata_a* и выделите для хранения данных в этом файле 4MB. Сконфигурируйте БД Test для автоматического роста размера файла при достижении лимита на 2MB с максимально возможным размером в 10MB.

Задание 2. Создание и конфигурация файловой группы

Создайте новую файловую группу под названием *TestFileGroup*. Создайте вторичный файл данных (с расширением ndf) БД Test с размером 5MB под названием *testdata_b*.

Сконфигурируйте созданный вторичный файл данных для автоматического роста размера, при достижении лимита, на 2MB с неограниченным размером (Unrestricted File Growth). Добавьте его в созданную ранее файловую группу *TestFileGroup*.

Задание 3. Режим проверки подлинности и настройка имени входа

Измените режим проверки подлинности сервера на смешанный (если до этого был режим Windows Authentication). Создайте имя входа *TestLogin1*, укажите для него пароль (выберите аутентификацию SQL Server Authentication). Добавьте созданное имя входа (или регистрационную запись) в фиксированную серверную роль sysadmin. Установите в качестве используемой по умолчанию БД для имени входа *TestLogin1* базу данных *Test*. Создайте имя входа *TestLogin2*, задайте для него пароль. Создайте пользователей БД *Test* с именами *TestUser1*, *TestUser2*.

Пользователь *TestUser1* должен соответствовать имени входа *TestLogin1*, а *TestUser2 – TestLogin2*.

Задание 4. Настройка пользовательских ролей

Создайте две пользовательские роли *Manager* и *Employee* в БД *Test*. Задайте для пользователя *TestUser1* роль *Manager*, а для пользователя *TestUser2* – роль *Employee*. Запретите для роли *Employee* изменение пользователя guest. Для этого в свойствах роли *Employee* (Data Role Properties) базы данных Test в разделе Securables выберите защищаемый объект Users. В списке разрешений (Explicit permissions) в пункте Alter (Изменение) выберите опцию Deny.

Задание 5. Резервное копирование и восстановление базы данных

1. Создайте файл резервной копии БД Test. Восстановите файл резервной копии с помощью процедуры восстановления (Restore) базы данных SQL Server.
2. Реализуйте план создания резервной копии базы данных Test через заданный интервал времени.

Так же реализуйте удаление файлов резервных копий старше 2-х недель в этом же плане.

Дополнительная информация:

1. Vitalie Cotelea, Marian Cotelea. Microsoft SQL Server 2019: Pas cu pas. – Universitatea Tehnică a Moldovei, Departamentul “Ingineria Software si Automatică”. - Chişinău: S.n., 2020 (Tipografia “Foxtrot”). -474 p.