



*Программирование
и администрирование
СУБД*

Microsoft®
SQL Server

Урок №1

Содержание

1. СУБД MS SQL Server 2005/2008. Преимущества и недостатки. Обзор версий и редакций	4
2. Учебный курс Microsoft по MS SQL Server	12
3. Установка MS SQL Server 2008 Express Edition / 2016.....	15
3.1. Установка MS SQL Server 2008 Enterprise.....	15
3.2. Установка Microsoft SQL Server 2016 Service Pack 1 Express	31
4. Утилиты SQL Server 2000/2005	40
4.1. Общая информация о утилитах	40
4.2. SQL Server Configuration Manager.....	41
4.3. SQL Server Management Studio.....	45
4.4. SQL Server Business Intelligence Development Studio	51

5. Работа с базой данных через утилиты и хранимые процедуры	54
5.1. Создание БД.....	54
5.2. Изменение свойств БД	62
5.3. Переименование БД.....	66
5.4. Удаление БД	67
6. Введение в SQL. Что это такое?	
Категории команд SQL: DDL, DML, DCL.....	68
7. Создание базы данных средствами SQL	72
8. Модификация и удаление базы данных.....	74
9. Импорт данных	77
10. Домашнее задание	85

1. СУБД MS SQL Server 2005/2008. Преимущества и недостатки. Обзор версий и редакций

Сегодня мы с вами начинаем изучение еще одной планеты из галактики "Базы Данных" – **MS SQL Server**. В настоящее время вы уже знаете, что такое база данных и СУБД, какие существуют модели баз данных, как осуществляется нормализация, что представляет собой первичный и внешний ключи и тому подобное. Итак, в данном курсе вам достаточно будет сконцентрировать свое внимание на особенностях работы SQL в среде СУБД MS SQL Server. Но прежде, чем перейти к изучению ее основ, рассмотрим вкратце историю ее создания.

Первоначально MS SQL Server разрабатывалась компанией **Microsoft** совместно с **Sybase Corporation** для использования на платформах IBM OS / 2. Весь код выпущенного около 1989 SQL Server 1.0 (до версии 7.0), был основан на коде Sybase SQL Server, что позволило ему выйти на рынок баз данных. Когда Microsoft и IBM разделились на отдельные компании, Microsoft отказалась от OS / 2 и создала свою собственную операционную систему – Windows NT Advanced Server 3.1. Ее релиз почти совпал с официальным релизом Microsoft SQL Server версии 4.21 для этой же ОС (1993 год). С этого момента было принято решение разрабатывать SQL Server только для среды Windows NT и в 1995 году. Microsoft выпускает **SQL Server 6.0**. После

этого Microsoft начали добиваться полных прав на все версии SQL Server для Windows. Такие шаги были вызваны тем, что в настоящее время корпорация Sybase, чтобы не стоять в стороне, выпустила собственную СУБД под Windows NT и назвала ее **Sybase Adaptive Server Enterprise** (для избежания путаницы с Microsoft SQL Server).

Развитие SQL Server не прекратилось и довольно скоро в свет выходят следующие версии:

- 1996 – SQL Server 6.5, кодовое название Hydra;
- 1999 – SQL Server 7.0, кодовое название Sphinx;
- 1999 – SQL Server 7.0 OLAP, кодовое название Plato;
- 2000 – SQL Server 2000 32-bit, кодовое название Shiloh (версия 8.0);
- 2003 – SQL Server 2000 64-bit, кодовое название Liberty;
- 2005 – SQL Server 2005, кодовое название Yukon (версия 9.0);
- 2008 – SQL Server 2008, кодовое название Katmai (версия 10.0);
- 2012 – SQL Server 2012, кодовое название Denali (версия 11.0);
- 2014 – SQL Server In-Memory OLTP, кодовое название Hekaton (версия 12.0).

Весь код SQL Server, начиная с версии 7.0, был переписан, чтобы избежать претензий со стороны Sybase на авторские права. Сам MS SQL Server 7.0 стал первым сервером баз данных с графическим интерфейсом пользователя для администрирования. Кроме того, в SQL Server 7.0 была улучшена система хранения данных, повышена

надежность, масштабируемость и доступность данных, а также добавлены оптимизатор запросов и целый ряд дополнительных средств, облегчающих работу как программиста баз данных, так и администратора.

SQL Server 2000 стал самой стабильной и надежной версией SQL Server. Поэтому эта версия остается популярной и сегодня. Он был реализован как служба Windows, но для Windows 98 и Windows ME остался выполняться как отдельное приложение в текущем сеансе пользователя. Все встроенные утилиты выполняются как дополнительные клиент-серверные приложения, позволяя тем самым управлять базой данных из любой точки сети.

Выпуск версии SQL Server 2005 проходил параллельно выпуску Visual Studio 2005 и поэтому его урезанную версию также включили в инсталляции последней. В новой версии, по сравнению с предыдущими, изменился интерфейс пользователя и администратора, часть утилит объединены. Но существенные изменения в основном касались реализации технологии ETL (извлечение, преобразование и загрузки данных), которые входят в состав компонента SQL Server Integration Services (SSIS), сервера оповещения, средств аналитической обработки многомерных моделей данных (OLAP) и сбора релевантной информации (обе службы входят в состав Microsoft Analysis Services), а также нескольких служб оповещения, а именно Service Broker и Notification Services. Кроме того, были сделаны существенные улучшения в производительности.

Следующая версия SQL Server – SQL Server 2008, которая была выпущена в августе 2008 года, направлена на то, чтобы сделать управление данными самонастраиваемым,

самоорганизованным и самообслуживаемым механизмом. Это позволит уменьшить до нуля время нахождения сервера в нерабочем состоянии (простои). Для реализации этих возможностей была создана технология SQL Server Always On.

Для повышения эффективности администрирования в SQL Server была включены новые инструменты, которые позволяют контролировать несколько серверов баз данных, а также библиотека Declarative Management Framework, которая позволяет распределять полномочия для баз данных или отдельных таблиц.

Кроме того, были добавлены **следующие возможности:**

- поддержка структурированных и частично структурированных данных, включая цифровые форматы для изображений, звуков, видео и других типов мультимедиа. Возможность использования мультимедиа форматов внутри СУБД позволила специализированным функциям взаимодействовать с этими типами данных;
- включены ряд дополнительных новых типов данных;
- улучшены методы компрессии данных;
- улучшена работа с полнотекстовыми индексами;
- улучшена многопоточность некоторых операций, например, создание больших индексов;
- встроены процедуры на языке .NET;
- улучшен интерфейс управления и отладки (Management Studio).

Корпорация Microsoft выпускает СУБД SQL Server в различных редакциях (реализациях). Это было сделано с целью удовлетворения требований широкого диапазона

потребителей. Например, для одной организации необходимо, чтобы база данных была дешевой или вообще бесплатной и удовлетворяла минимальные требования по работе. Для другой же организации нужно, чтобы аналитическая часть базы данных выполнялась как на больших машинах, так и на персональных или карманных компьютерах. В связи с этим основное отличие различных реализаций СУБД заключается в их эффективности, производительности и, конечно же, цене.

SQL Server 2005 представлена в таких редакциях:

1. **Enterprise Edition** (32- и 64-разрядная версия) – это наиболее полное издание, которое поддерживает до 32 процессоров и до 64 Гб ОЗУ. Основные характеристики:
 - масштабируемость до таких уровней эффективности, которые могут понадобиться при поддержке крупных Web-сайтов, промышленных OLTP и Data Warehousing систем;
 - эффективность;
 - высокий уровень доступности и готовности в критических условиях (например, при сбоях);
 - включает несколько дополнительных опций анализа, не включенные в SQL Server 2000 Standard Edition.
2. **Standard Edition** (32- и 64-разрядная версия) используется в основном на малых и средних предприятиях, которым не требуются высокие показатели масштабируемости, эффективности, аналитики или доступности. Поддерживает до 4-х процессоров и до 2-х Гб ОЗУ. Standard Edition включает в себя основные

функциональные возможности, которые необходимы для некритичных бизнес-приложений и приложений электронной коммерции, источников данных и тому подобное;

3. **Developer Edition** (32- и 64-разрядная версия) – аналог Enterprise Edition, но ориентирована на разработку (то есть на создание различного рода прикладных приложений) и запрещена для коммерческого использования;
4. **Workgroup Edition** (32-разрядная версия) – предназначена для управления данными в небольших организациях с ограниченными по размеру и по количеству пользователей базами данных, которые могут обслуживать небольшие веб-серверы, а также работу подразделений и филиалов. Фактически она предназначена для администраторов, работающих на небольших серверах с небольшими объемами данных.
5. **Express Edition** (32-разрядная версия) – урезанная и бесплатная версия Microsoft SQL Server. Простая в использовании и управлении, может использоваться и в качестве клиентской, и в качестве серверной базы данных. Работает с небольшими базами данных. SQL Server 2008 существует в редакциях:
 - 1) **Enterprise**;
 - 2) **Standard**;
 - 3) **Workgroup**;
 - 4) **Web** – специально предназначена для создания веб-служб с высоким уровнем доступности. Предназначена для крупных организаций,

которые в основном ориентированы на работу в Интернет-среде;

- 5) **Compact** – бесплатная и встроенная база данных SQL Server, которую можно использовать для разработки автономных или малосвязанных приложений для мобильных устройств (карманных ПК и смартфонов), ПК и веб-клиентов;
- 6) **Express** – урезанная бесплатная версия, которая доступна в SQL Server 2008 в трех вариантах:
 - **SQL Server 2008 Express** – позволяет создавать, хранить, обновлять и обрабатывать данные;
 - **SQL Server 2008 Express with Tools** – кроме основных действий с базой данных, включает в себя среду SQL Server Management Studio Basic – визуальное средство для создания, редактирования и управления базами данных;
 - **SQL Server 2008 Express with Advanced Services** – аналог SQL Server 2008 Express with Tools и дополнительно включает в себя:
 - полнотекстовый поиск – мощный скоростное средство для поиска данных в больших объемах текста;
 - службу отчетов, которая имеет свою среду конструирования.

Итак, **Microsoft SQL Server** – это система управления реляционными базами данных (СУБД), разработанная корпорацией Microsoft. Последние версии представлены расширенным набором реализаций, которые позволяют конечному пользователю выбрать необходимую,

сэкономив при этом средства. Кстати, MS SQL Server для запросов использует специальную разновидность **SQL – Transact-SQL (T-SQL)**, которая была создана совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI / ISO SQL с рядом расширений.

Поскольку версии MS SQL Server 2000 и 2005 все еще широко используются, так же как и MS SQL Server 2008, то рассмотрим основные принципы работы и инсталляции всех трех. Но больший акцент все же будем делать на новую версию.

2. Учебный курс Microsoft по MS SQL Server

Система сертификации Microsoft является общепринятым стандартом и является эффективным способом достижения карьерных целей в среде IT. Программы сертификации разработаны почти для каждого продукта Microsoft и для различных категорий пользователей: от IT профессионалов, разработчиков до офисных работников и домашних пользователей.

Прежде, чем перейти к системе сертификации, необходимо ввести ряд понятий предметной области. В частности, существуют такие понятия как **экзамены** и **статусы**. Как правило, для получения любого статуса необходимо сдать несколько экзаменов, но бывают и исключения, то есть для получения статуса нужно сдать только один экзамен.

Существует четко построенная иерархия статусов и поэтому довольно часто для получения определенного статуса необходимо иметь набор других. Базовый статус для IT специалиста сейчас MCTS (Microsoft Certified Technology Specialist). Данный статус можно получить по различным областям знаний, например, Windows Forms, Web based development и тому подобное.

Полный список сертификационных экзаменов Microsoft можно посмотреть по этой ссылке:

<https://www.microsoft.com/en-us/learning/exam-list.aspx>

Экзамены по MS SQL Server находятся под номерами от 70-461 до 70-467. С 18 февраля 2016 года экзамены с 461 по 467 включают материалы, охватывающие SQL Server 2012 и 2014. Сертификация MSCA теперь называется MCSA: SQL Server 2012/2014.

Чтобы получить сертификацию MCSA: SQL 2016 Database Development, нужно сдать следующие экзамены:

- Экзамен 70-761 Querying Data with Transact-SQL (beta);
- Экзамен 70-762 Developing SQL Databases.

Чтобы получить сертификацию MCSA: SQL 2016 Database Administration, нужно сдать следующие экзамены:

- Экзамен 70-764 Administering a SQL Database Infrastructure (beta);
- Экзамен 70-765 Provisioning SQL Databases.

Детали о новых сертификатах для MS SQL 2016 приведены здесь:

<https://www.microsoft.com/en-us/learning/mcsa-sql-2016-certification.aspx>

Курс, который мы сегодня с Вами начнем, подготовит Вас к сдаче экзаменов на получение статусов для MCTS, MCITP: Database Developer, а также поможет в получении начальных знаний для сертификационного экзамена на получение статуса MCITP: Database Administrator.

После статуса MCITP можно получить статус тренера **MCT (Microsoft Certified Trainer)**. Это человек, который имеет право вести авторизованные курсы Microsoft в центрах партнерах Microsoft. Для получения данного статуса необходимо прослушать специальный тренинг

и продемонстрировать педагогическое мастерство перед аудиторией.

Вот и все. Наконец, можно привести несколько полезных ссылок:

- Базовая страница с информацией о сертификации:
<https://www.microsoft.com/en-us/learning/certification-overview.aspx>
- Информация о сертификации:
<https://www.microsoft.com/en-us/learning/certification-exams.aspx>
- Древовидная диаграмма информации о статусах:
http://download.microsoft.com/download/a/0/6/a0675804-40a3-4f34-ac64-6de190da822d/ICT_Curriculum_Roadmap.pdf

3. Установка MS SQL Server 2008 Express Edition / 2016

3.1. Установка MS SQL Server 2008 Enterprise

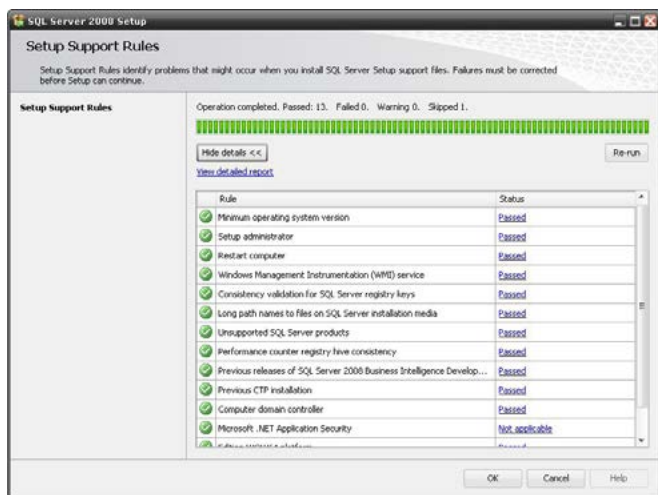
Для успешной установки MS SQL Server 2008 Enterprise необходимо наличие установленных пакетов: Microsoft .NET Framework 3.5 SP1, Windows Installer 4.5 и Windows PowerShell 1.0. Но если у Вас нет необходимых компонентов – не следует разочаровываться, программа-инсталлятор их установит сразу после запуска.

Сама установка MS SQL Server 2008 включает в себя следующие шаги:

1. Запустить программу установки.
2. В разделе **"Planning"** (Планирование) выбрать пункт **"System Configuration Checker"** (Средство проверки конфигурации):



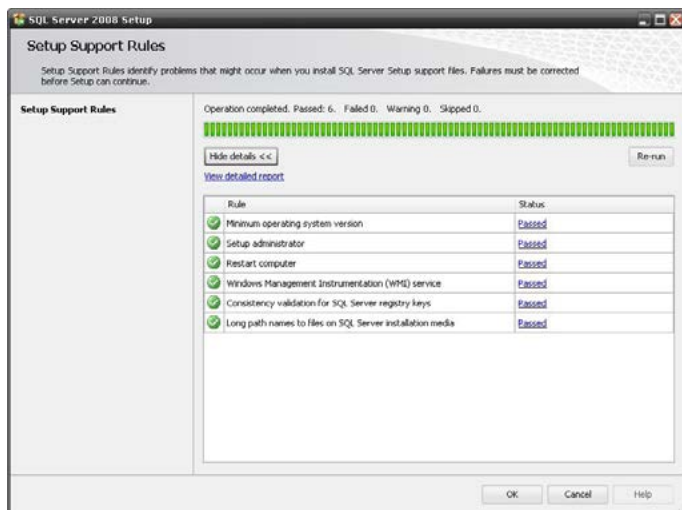
3. Нажать кнопку "Show details" (Показать детально) и убедиться, что все проверки успешно пройдены. Если будут обнаружены ошибки, их необходимо исправить и запустить повторную проверку кнопкой "Re-run" (Повторить). После этого закрыть окно с помощью кнопки "ОК":



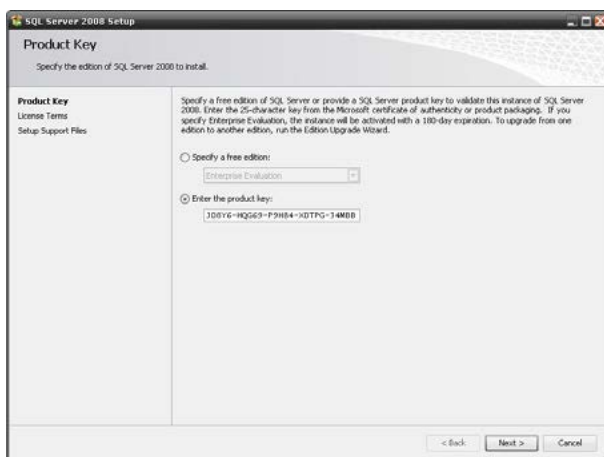
4. После этого следует в окне инсталлятора выбрать раздел "Installing" (Установка) и пункт "New SQL Server stand-alone installation or add features to an existing installation" (Новая установка изолированного SQL Server или добавление компонентов к существующему экземпляру). Если у вас установлена предыдущая версия, то вы можете ее обновить до необходимой с помощью пункта "Upgrade from SQL Server 2000 or SQL Server 2005" (Обновление с SQL Server 2000 или SQL Server 2005):



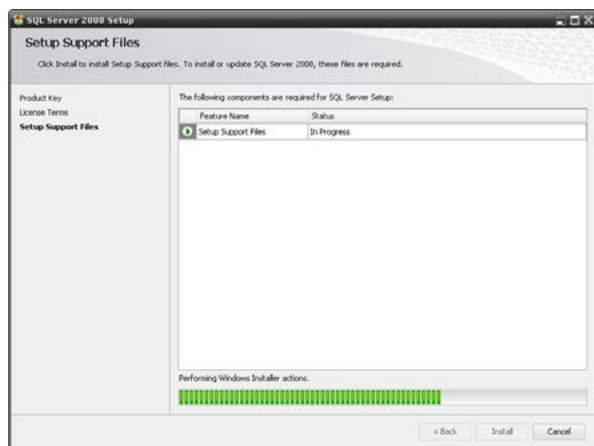
5. После этого запустится процесс проверки системы перед установкой. Аналогично с пунктом 3, вы можете просмотреть подробный отчет этой проверки с помощью кнопки "Show details" (Показать подробно), а для завершения нажать кнопку "OK":



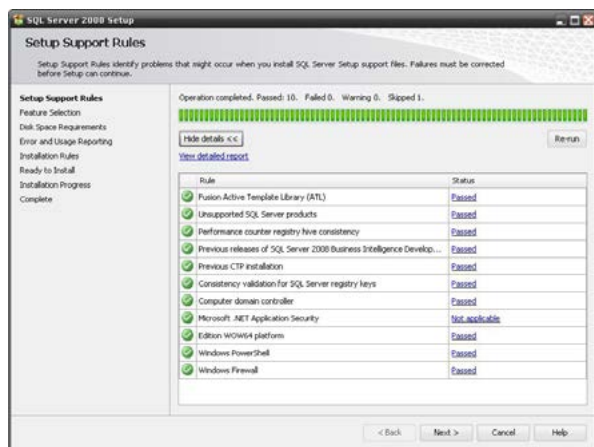
6. Вводим лицензионный ключ или выбираем бесплатную редакцию и нажимаем кнопку "Next":



7. Далее необходимо прочесть лицензию. Если после всего прочитанного, вы решитесь с ними согласиться, тогда ставите соответствующую галочку и переходите на следующий шаг (кнопка "Next"). Иначе процесс инсталляции для вас завершился.
8. В следующем окне запускаем инсталляцию с помощью кнопки "Install" (Установить):



9. Проверку для дальнейшей работы осуществляем аналогично пункту 3 и 5 инсталляции. После этого нажимаем кнопку "Next":



Примечание! Предупреждение в пункте "Windows Firewall" (Брандмауэр Windows), если оно возникло, можно проигнорировать – оно акцентирует внимание на том, что нужно его (брандмауэра) дополнительная настройка для доступа к SQL Server с других компьютеров.

10. Выбираем необходимые компоненты для установки. Их количество и состав в зависимости от редакции SQL Server 2008 будет отличаться. Например, для редакции SQL Server 2008 Express Enterprise он будет следующим:



Коротко опишем список **компонентов**:

1. **Компоненты экземпляра (Instance Features)** – это набор серверных компонентов для отдельных экземпляров. Для каждого из этих компонентов существуют отдельные ветви реестра. К ним относятся:
 - 1.1. **Служба компонента Database Engine (Database Engine Services)** – включают компонент Database Engine, который представляет собой основную службу для хранения, защиты и обработки данных. По сути, это ядро SQL Server, которое использует .NET. Эти службы обеспечивают управляемый доступ, быструю

обработку транзакций и поддержание высокого уровня доступности. Думаем, что вы и сами понимаете, что установление службы Database Engine является обязательным.

- репликация SQL Server (SQL Server Replication) – отвечает за возможность репликации данных.
- полнотекстовый поиск (Full-Text Search) – отвечает за быстрый поиск текста в существующих данных, то есть за работу с индексами, сохраняют значение данных и указатели на строки, где эти данные встречаются. То есть отвечает за технологию сервера Microsoft Index Server.

1.2. **Службы Analysis Services** – это набор средств для поддержки аналитической обработки (OLAP) и интеллектуального анализа данных. Поддерживается в редакции MS SQL Server 2008 Enterprise.

1.3. **Службы Reporting Services** – это набор компонентов, средств и программных интерфейсов для разработки и использования встроенных отчетов в управляемой среде. В набор средств входят средства разработки, настройки и администрирования, а также средства просмотра отчетов. К программным интерфейсам относятся протокол доступа к объектам (Simple Object Access Protocol, SOAP), конечные точки URL-адреса и инструментарий

управления Windows (WMI), который обеспечивает простую интеграцию с Web-приложениями и порталами.

2. **Совместные функции (Shared Features)** – используются всеми экземплярами сервера:

2.1. **Средства Business Intelligence Development Studio** – новая версия среды Microsoft Visual Studio 2008 с дополнительными типами проектов, которые характерны для работы с бизнес-аналитикой в SQL Server. Это основная среда для разработки бизнес-решений, в состав которых входят проекты служб Analysis Services, Integration Services и Reporting Services. Подробнее смотрите раздел описания данной утилиты.

2.2. **Средства связи клиентских средств (Client Tools Connectivity)** – средства для связи клиентов и сервера.

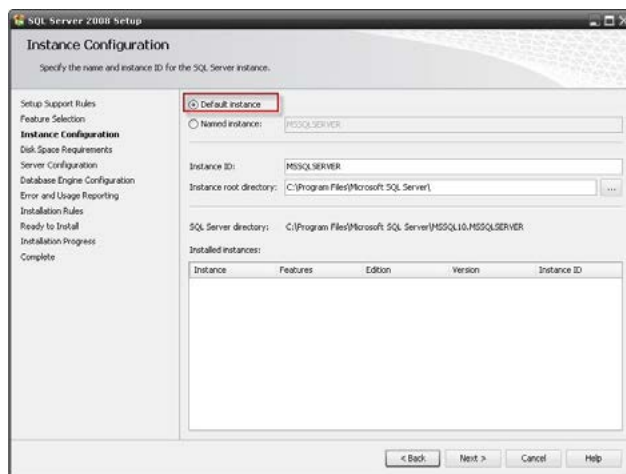
2.3. **Службы Integration Services** – эти службы используются для осуществления преобразования, интеграции и перемещения данных между разного рода бизнес-приложениями. Не написав ни строчки кода, можно создавать решения служб Integration Services, которые позволяют решать сложные бизнес-проблемы, управлять базами данных SQL Server, а также копировать объекты SQL Server с одного экземпляра в другой.

- 2.4. **Обратная совместимость клиентских средств (Client Tools Backwards Compatibility)** – предусматривает включение клиентских средств, которые будут обеспечивать совместимость с младшими версиями SQL Server, например, консоль управления Microsoft, разного рода APE-интерфейсы и тому подобное.
- 2.5. **Пакет SDK клиентских средств (Client Tools SDK)** – включает пакет программного обеспечения с ресурсами для программистов.
- 2.6. **Электронная документация по SQL Server (SQL Server Books Online)** – электронный справочник по SQL Server.
- 2.7. **Средства управления – основные (Management Tools – Basic)** – включает поддержку среды Management Studio для компонента Database Engine, утилиту командной строки SQLCMD и поставщик SQL Server PowerShell.
- 2.8. **Пакет SDK средств связи клиента SQL (SQL Client Connectivity SDK).**
- 2.9. **Microsoft Sync Framework** – комплексная платформа синхронизации, которая поддерживает совместную работу и автономный доступ для приложений, служб и устройств. В платформе применяются технологии и средства, которые поддерживают перемещение, совместное использование данных и перевода их в автономный режим. С помощью

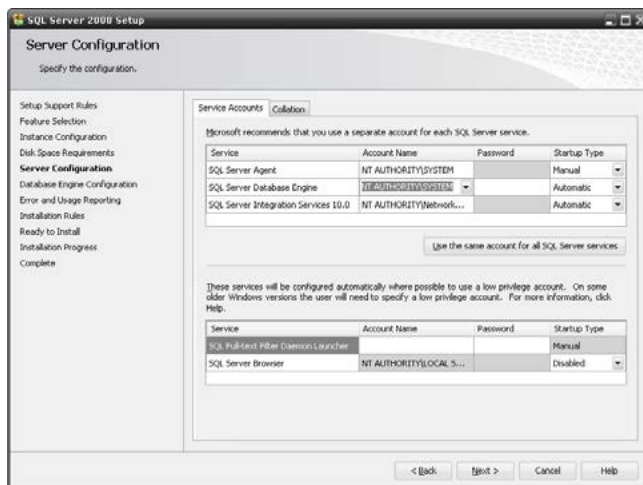
платформы Sync Framework разработчики могут создавать экосистемы синхронизации, которые позволяют интегрировать произвольные приложения с данными любого источника данных, используя любой сетевой протокол.

3. Распространяемые компоненты (Redistributable Features) – набор компонентов, которые устанавливаются в случае необходимости: отчеты об ошибках, собственный клиент SQL Server, MSXML версии 6.0, SQL Server Compact Workbench, среда выполнения SQL Server Compact, Sync Services для ADO.NET и обозреватель SQL Server.

11. Выбираем **вариант установки** – экземпляр по умолчанию (Default instance) или именованный экземпляр (Named instance). Более подробно эта информация содержится в предыдущем разделе. Мы выбираем стандартный экземпляр (Default instance) и все опции оставляем по умолчанию и переходим на следующий шаг.



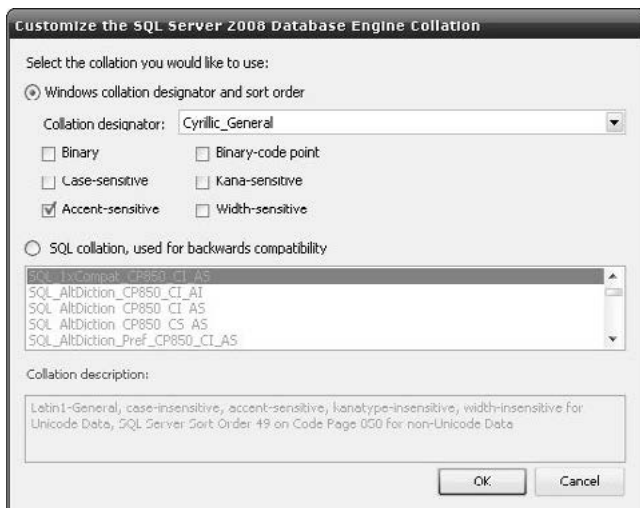
12. Следующий шаг знакомит нас с тем, сколько места будет занимать наш сервер баз данных, и по какому адресу он будет размещен. Нажимаем кнопку "Next".
13. Настраиваем конфигурацию будущего сервера. Для этого выбираем опции как показано на рисунке и переходим на закладку "Collation" (Параметры сортировки):



Примечание! Если Вы хотите использовать SQL Server Agent 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 отдельный аккаунт для SQL Server Agent и предоставить ему права на соответствующие ресурсы. При такой конфигурации, в этом же окне необходимо в поле "Account Name" (Имя учетной записи) и "Password" (Пароль) ввести имя созданной учетной записи (вида Домен \ Имя) и ее пароль.

14. В закладке **"Collation" (Параметры сортировки)** необходимо указать требуемые значения параметров сортировки SQL Server и службы аналитики Analysis Services. Мы оставляем все без изменений:

Примечание! Чтобы изменить данные опции следует нажать соответствующую кнопку "Customize" (Настройка) и установить необходимые параметры.

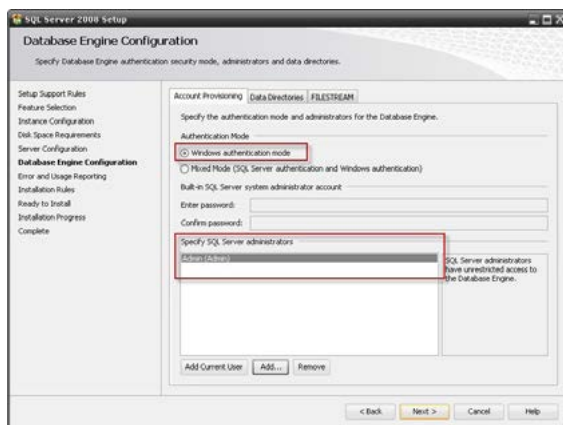


Если Вам необходимо обеспечить обратную совместимость с ранними версиями SQL Server, тогда следует выбрать переключатель с опцией "SQL Collation" (Параметры сортировки SQL). Но если вас волнует обратная совместимость, тогда выберите параметр сортировки Cyrillic_General или удобный для вас и необходимый метод сортировки:

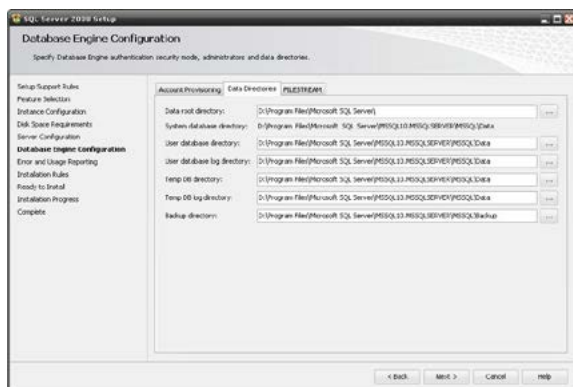
- 1) Binary (двоичный) – сортировка по ASCII кодам;
- 2) Case sensitive – сортировка с учетом регистра;
- 3) Accent sensitive – сортировка с учетом диакритических значений, то есть как с учетом регистра, так и с учетом ударения;
- 4) Binary code point – сортировка по ASCII кодам, с учетом кодовых точек;
- 5) Kana sensitive – с учетом типов японского алфавита хирагана от Катаканы (алфавиты, в которых одинаковые символы пишутся по-разному);
- 6) Width sensitive – для языков с двухбайтовыми символами при операции сортировки каждый байт должен отличаться друг от друга.

Кстати, данные настройки после установки изменить нельзя, поэтому будьте внимательны.

15. Выберите способ авторизации пользователя на сервере (подробнее можно прочитать в предыдущем разделе). Мы выбираем режим авторизации Windows и назначаем аккаунт, который будет соответствовать администратору.



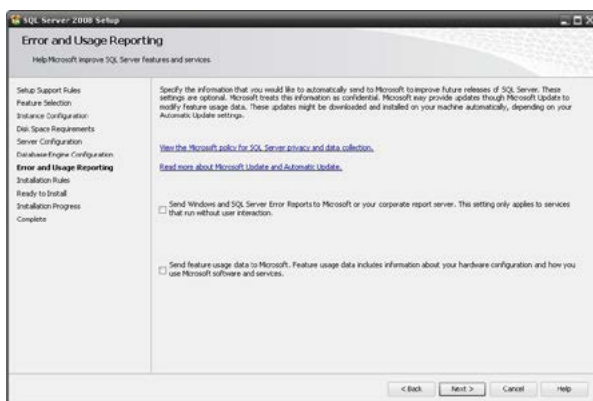
16. После этого переходим на **закладку "Data Directories" (Каталоги данных)** и настраиваем необходимые каталоги: корневой каталог данных, системный и пользовательский каталог базы данных и тому подобное.



17. Устанавливаем службу Analysis Services и Reporting Services, если вы их выбрали и нажимаем на кнопку "Next":

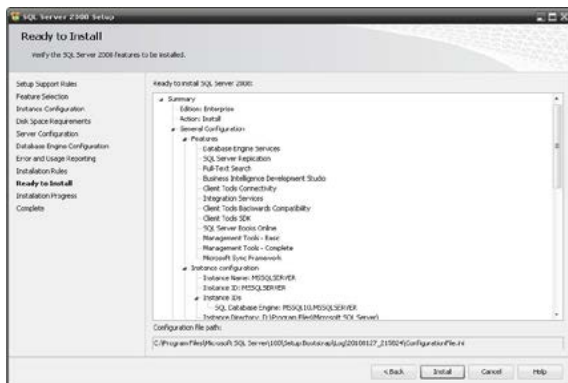


18. Настраиваем механизм отправки ошибок MS SQL Server (подробнее в предыдущем разделе).

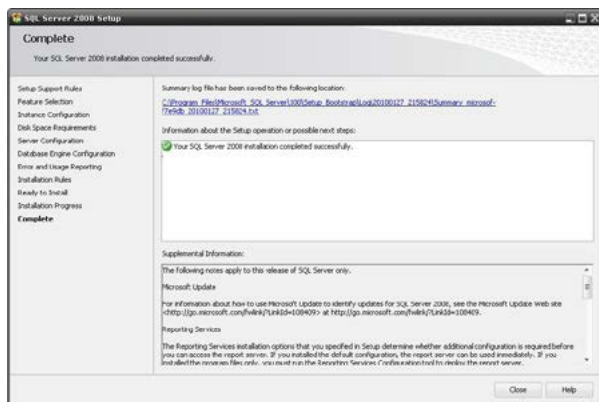


19. Проходим очередную проверку и если все в порядке, переходим на следующий шаг инсталляции.

20. Проверяем, правильно мы выбрали необходимые компоненты для установки и нажимаем кнопку "Install" (Установить):



21. После завершения работы инсталлятора и просмотра самого процесса инсталляции переходим на следующий шаг и завершаем работу, нажав кнопку "Close" (Закреть):



Вот и все. Поздравляем с завершением процесса инсталляции MS SQL Server 2008 Enterprise.

3.2. Установка Microsoft SQL Server 2016 Service Pack 1 Express

Прежде всего, вы должны знать, что MS SQL Server 2016 существует только в версии для 64-разрядных операционных систем. Если у вас операционная система не 64-разрядная, вам надо выбрать сервер предыдущей версии.

Затем надо скачать небольшой загрузчик дистрибутива MS SQL Server 2016 по ссылке:

<https://www.microsoft.com/en-us/download/details.aspx?id=54284>

Перед скачиванием проверьте на этой же странице системные требования к компьютеру и операционной системе, чтобы понять, возможна ли установка в вашем случае. Если с этим у вас порядок, переходите к следующим действиям. Запустите программу загрузчик, и она скопирует на ваш компьютер файл с именем SQLEXP_x64_ENU.exe размером более 430 МБ. Этот файл является дистрибутивом MS SQL Server 2016. После его запуска он распакует требуемые файлы в указанную вами папку и вы сможете начать процесс установки.

Внимание! Запомните папку, в которую дистрибутив распаковывает свои файлы. Эта папка вам понадобится в будущем.

1. Окно установки выглядит таким образом.

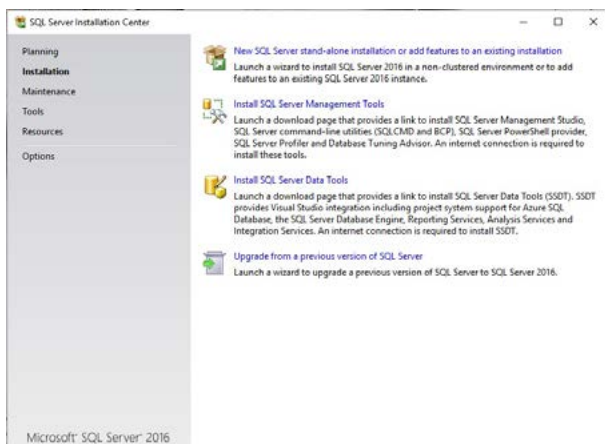


Рис. 1. Окно установки MS SQL Server 2016 SP1

- Выберите опцию New SQL Server stand-alone installation or add features to existing installation. Эта опция позволит вам либо установить новую версию сервера, либо выполнить апгрейд предыдущей версии, если она у вас была установлена. После активации этой опции начнется процесс детализации установки, который займет какое-то время.

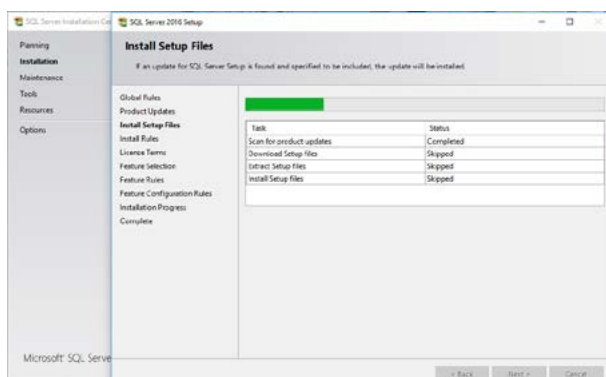


Рис. 2. Процесс установки MS SQL Server 2016 SP1

- Сначала дистрибутив автоматически выполнит проверку среды на возможность установки MS SQL Server 2016. Если возникнут какие-либо проблемы, программа выведет вам их описание. Если все в порядке, вы увидите такое (или подобное) окно:

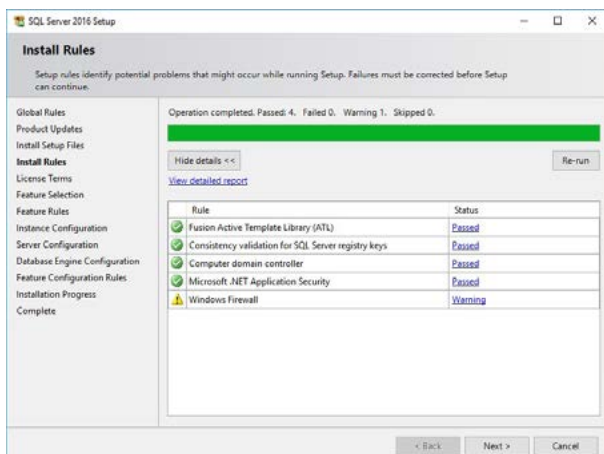


Рис. 3. Результат проверки правил установки

- Теперь надо уточнить некоторые параметры установки MS SQL Server 2016 на ваш компьютер. Сначала вам предложат ознакомиться и согласиться с лицензионным соглашением (см. Рис.4).
- Теперь вы сможете определиться с набором устанавливаемых опций MS SQL Server 2016 и выбрать папку на своем диске, в которую сервер будет установлен. Желательно отметить все доступные опции. Не ищите на этом этапе возможность указать, что вы хотите установить вместе с сервером SQL Server Management Studio (SSMS). Дело в том, что установка SQL Server

Management Studio не включается в установку самого сервера. После завершения установки сервера, мы рассмотрим установку SQL Server Management Studio отдельно (см. Рис.5).

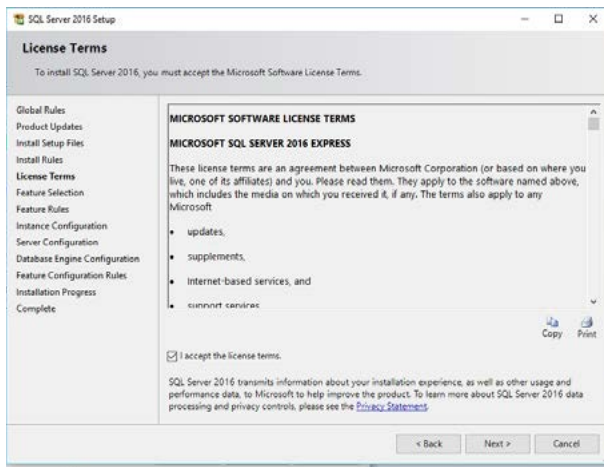


Рис. 4. Лицензионное соглашение

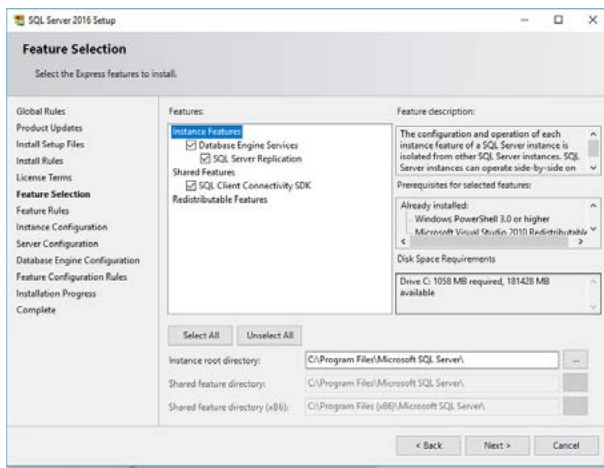


Рис. 5. Выбор устанавливаемых компонент

6. На этом этапе вы можете указать, какой экземпляр сервера хотите установить. Предпочтительным является выбор значения по умолчанию. Обратите внимание на имя экземпляра сервера, приведенное здесь. Вы будете указывать это имя при подключении к серверу в своих приложениях или в SQL Server Management Studio.

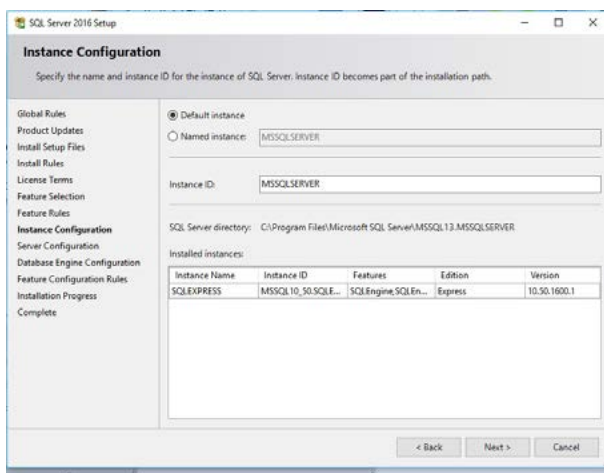


Рис. 6. Выбор экземпляра сервера

7. Теперь вам предложат выбрать способы аутентификации при подключении к серверу. Полезно будет выбрать опцию Mixed Mode, что позволит вам использовать как Windows аутентификацию, так и аутентификацию MS SQL Server. При этом вам надо будет придумать и указать в этом окне пароль для доступа с MS SQL Server аутентификацией.

Внимание! Запомните указанный здесь пароль для доступа с MS SQL Server аутентификацией.

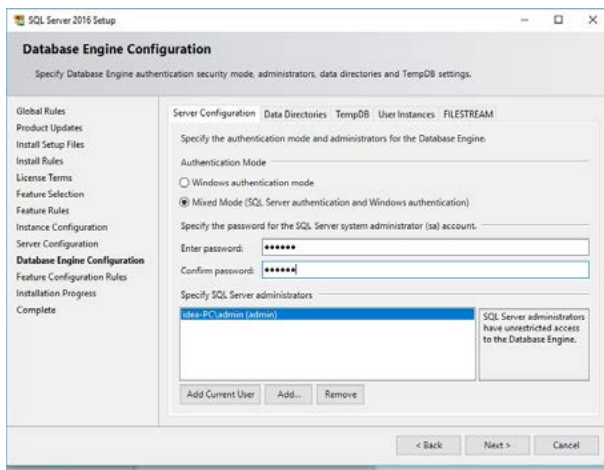


Рис. 7. Выбор способов аутентификации

8. Теперь будет выполняться установка в соответствии с выбранными вами опциями. Этот процесс займет какое-то время и будет отображаться в окне прогресс баром.

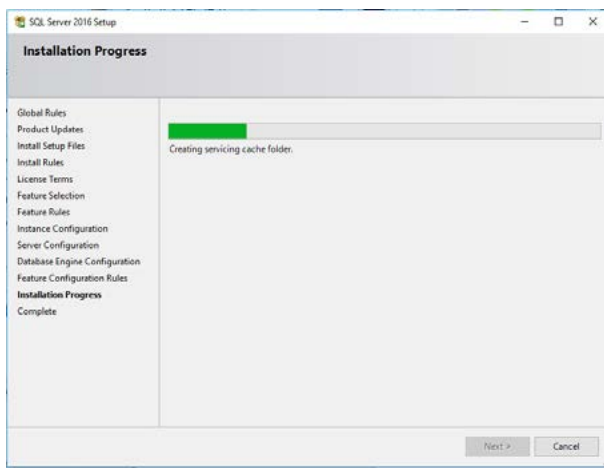


Рис. 8. Выполнение установки

9. При успешном завершении установки вы увидите такое окно:

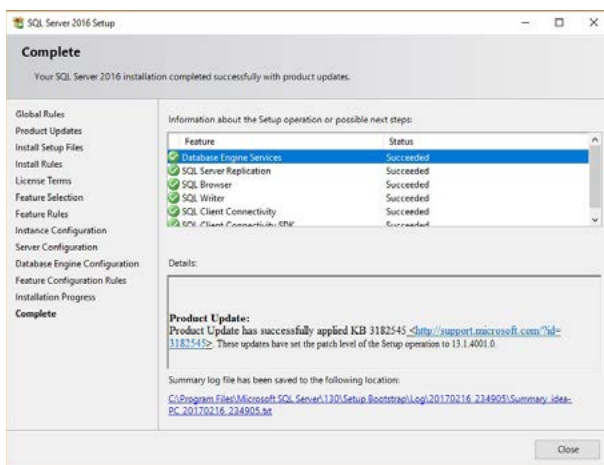


Рис. 9. Успешное завершение установки

10. Вот теперь пришло время для установки SQL Server Management Studio. Все время, пока выполнялась установка, начальное окно дистрибутива оставалось открытым. Если вы его закрыли, не расстраивайтесь. Вам надо зайти в папку, куда был распакован дистрибутив и запустить там файл setup.exe. В любом случае, вы снова увидите окно, приведенное на Рис. 1. Но в этот раз вам надо выбрать опцию Install SQL Server Management Tools. После активации этой опции установщик запустит браузер и переадресует вас на страницу загрузки дистрибутива SQL Server Management Studio. Начните загрузку. После ее завершения на вашем компьютере появится дистрибутив SQL Server Management Studio — файл с именем

SSMS-Setup-ENU.exe и с размером более 919 МБ. Запустите этот файл, чтобы начать установку SQL Server Management Studio. Вы увидите такое окно:

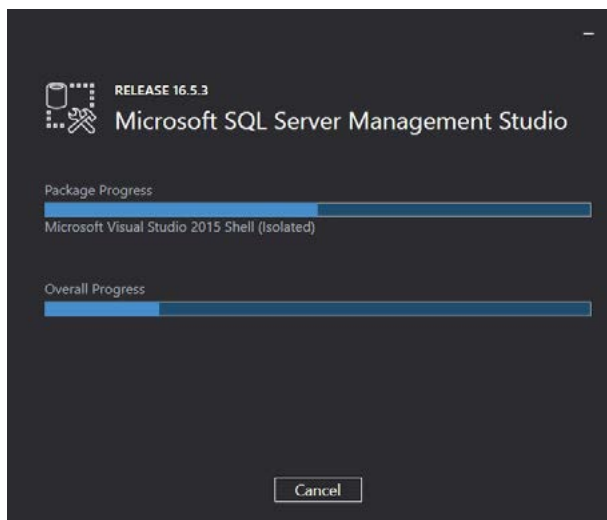


Рис. 10. Установка SQL Server Management Studio

В какой-то момент вам может показаться, что процесс установки остановился: прогресс бар будет оставаться на месте, и комментарии не будут изменяться. Это может продолжаться достаточно долго. Пожалуйста, не паникуйте и не прерывайте процесс установки. После паузы он продолжится далее. И скоро вы увидите такое окно:

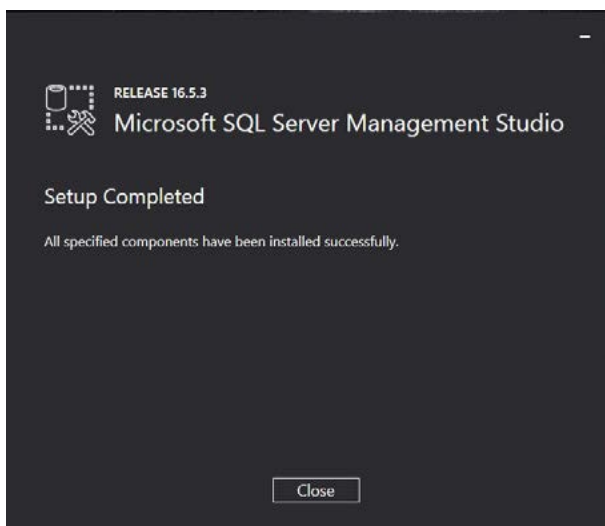


Рис. 11. Успешное завершение установки SQL Server Management Studio

4. Утилиты SQL Server 2000/2005

4.1. Общая информация о утилитах

SQL Server 2005 и 2008 предоставляет ряд **средств управления SQL Server**, основными из которых являются следующие:

- инструмент управления базами данных – SQL Server 2005 Management Studio, содержащий в себе инструменты для работы с запросами SQL, а также включает в себя средства администрирования и импорта / экспорта базы данных;
- инструменты разработки баз данных – SQL Server 2005 Business Intelligence Development Studio – это новая версия среды Microsoft Visual Studio с дополнительными типами проектов, которые характерны для работы с бизнес-аналитикой в SQL Server;
- диспетчер конфигурации SQL Server – SQL Server Configuration Manager, который позволяет контролировать SQL серверы, работающие на машине. Точнее он отвечает за управление службами SQL Server на компьютере, где установлен сервер баз данных, конфигурирование сетевых протоколов серверной и клиентской частей и тому подобное;
- sqlcmd – новая утилита командной строки для замены isql и osql.

4.2. SQL Server Configuration Manager

SQL Server Configuration Manager – это средство, которое предназначено для:

- управления службами, связанными с SQL Server;
- настройки сетевых протоколов, которые используются SQL Server;
- управления конфигурацией подключений с клиентских компьютеров SQL Server.

SQL Server Configuration Manager представляет собой оснастку консоли управления (mmc.exe) для открытия которого **используется файл SQLServerManager.msc в папке Windows-> System32**. Как уже было сказано, данная утилита объединяет в себе функциональные возможности таких средств SQL Server 2000 как Server Network Utility, Client Network Utility и диспетчер служб (Services Manager).

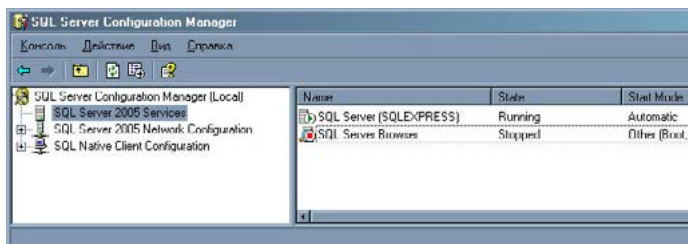
Вид и основные принципы работы данной утилиты в версиях SQL Server 2005 и SQL Server 2008 не отличается.

На закладке **"SQL Server Services" ("Службы SQL Server")** вы можете просмотреть список и текущее состояние сервисов, которыми Вы можете управлять.

В SQL Server 2005 Express Edition или SQL Server 2008 Express их всего две:

- SQL Server – это ядро SQL Server. Она обслуживает ключевые задачи как выполнение запросов, управление доступом к данным и распределение системных ресурсов (оперативная память и мощность процессора);
- SQL Browser. Эта служба прослушивает в сети входящие запросы к ресурсам SQL Server, а также

обеспечивает пользователя информацией об экземплярах SQL Server, установленные на сервере.



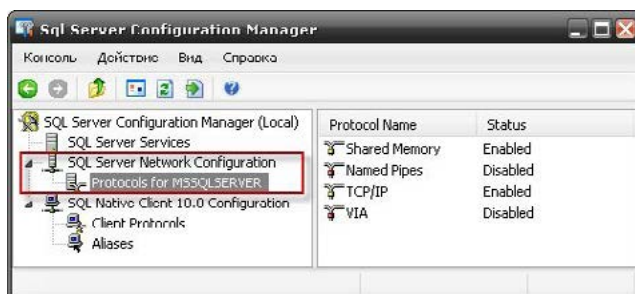
В SQL Server 2005/2008 Enterprise (или другой распространенной редакции сервера) этот список может быть несколько шире и зависит от того, какие компоненты служб Вы выбрали при инсталляции сервера. Например, на рисунке, который представлен ниже, включена дополнительно служба управления автоматизацией администрирования SQL Agent, SQL Server Integration Services и др.



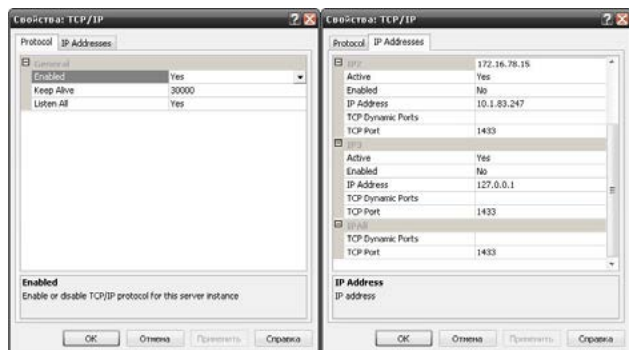
Чтобы изменить конфигурацию определенной службы, достаточно открыть окно ее свойств. Например, вы можете запускать или останавливать службу, указывать режим ее автозапуска, изменить имя пользователя и пароль учетной записи данной службы и тому подобное.



Чтобы клиент имел возможность подключаться к SQL Server по сети, на обоих концах соединения должна быть установлена общая сетевая библиотека. В разделе **"SQL Server Network Configuration" (сетевая конфигурация SQL Server)** вы сможете просмотреть список сетевых библиотек, установленных на сервере и получить возможность их настроить.



Если раскрыть группу **"Protocols for MSSQLSERVER"**, вы увидите список протоколов, которые может "прослушивать" сервер. Для настройки свойств любой библиотеки, выберите в контекстном меню пункт **"Properties"** (Свойства).



На закладке **"IP Address" (IP-адрес)** можно настроить протокол на одних сетевых адаптерах и отключить на других, сконфигурировать отдельно каждый адаптер или использовать специальную опцию **"IP All"** и настроить все адаптеры как единое устройство.

Раздел **"SQL Native Client 10.0 Configuration" (Настройка собственного клиента SQL)** является партнером утилиты SQL Server 2000 Network Configuration. Его используют для настройки протоколов, которые используются клиентской стороной для взаимодействия с сервером. В данном разделе существует только два пункта:

- **Client Protocols** – список доступных протоколов. Отличие данного списка от списка протоколов SQLSERVER (из раздела SQL Server Network Configuration) состоит лишь в том, что каждый протокол имеет свой порядковый номер. Эти порядковые номера

указывают на порядок, в котором клиент будет подключаться к серверу;

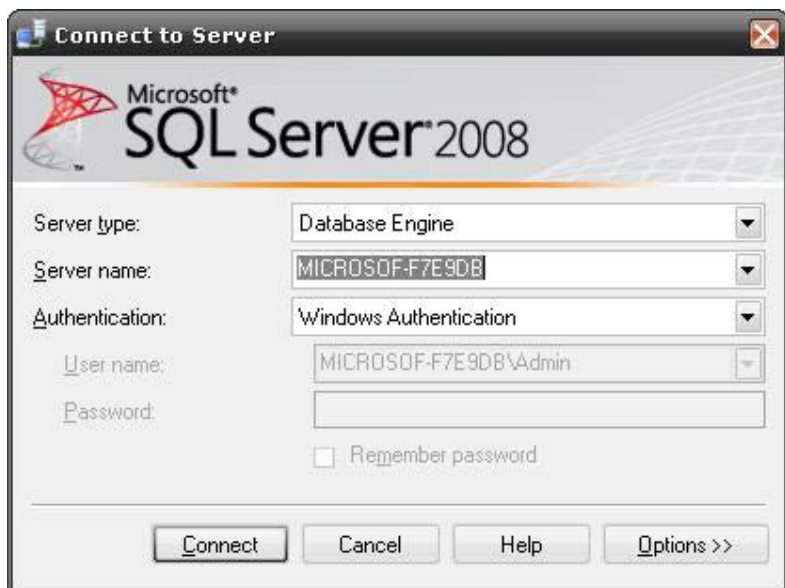
- **Aliases.** Данное подразделение используется только в исключительных ситуациях, и поэтому разберем его на примере. Многие компании имеют несколько экземпляров SQL Server, которые работают параллельно и имеют различные настройки. Например, один сервер может работать с включенной сетевой библиотекой TCP / IP и прослушивать порт 1433, а другой – порт TCP 37337 (обычно используется системой безопасности). Другие серверы могут иметь другие настройки, которые позволяют различным клиентам подключаться к серверу. Если в Вашей компании используется такой подход, то для каждого из серверов организации, который не использует значение по умолчанию для сетевой библиотеки, можно создавать серверный псевдоним.

4.3. SQL Server Management Studio

Среда SQL Server Management Studio – это интегрированная среда для доступа, настройке, управлению и администрированию всех компонентов SQL Server. Она объединяет в единую среду возможности программ Enterprise Manager и Query Analyzer, которые входили в состав предыдущих версий SQL Server. То есть SQL Server Management Studio является хорошим помощником в администрировании, создании и разработке баз данных, таблиц, связей и тому подобное. Кроме того она предоставляет среду для управления службами Analysis Services, Integration Services, Reporting Services и запросами на XQuery.

Внешний вид и основные принципы работы в SQL Server 2005 и SQL Server 2008 не отличаются.

При входе в SQL Server Management Studio вам предложат пройти аутентификацию. На экране появится следующее окно:



Для начала вам необходимо выбрать тип сервера, а в поле **"SQL Server" (Имя сервера)** выбрать собственно сам сервер (при необходимости с указанием его IP-адреса), с которым вы будете работать. После этого нужно указать режим авторизации: Windows authentication или SQL Server authentication. При выборе последнего вам будет предоставлена возможность ввести логин и пароль для доступа.

После ввода данных для подключения перед вами появится собственная среда Management Studio.



В левой панели этой утилиты размещается панель **Object Browser**, в выпадающем древовидном списке которой отражаются все активные SQL Server. Для каждого зарегистрированного сервера отводится своя ветка, название которой равно имени сервера, к которому осуществляется подключение, и логин, введенный при регистрации.

Каждый сервер имеет набор стандартных групп, которые размещаются в отдельных папках:

1. В папке **Databases** указаны все базы данных, которые на данном сервере. В СУБД существует группа уже заранее установленных баз данных, к которым относятся:

- **база master** – одна из важных баз данных SQL сервера. Она содержит информацию о пользователях, хранимые процедуры, сообщения об ошибках и другую служебную информацию;
- **база model** – на ее основе создаются все новые базы, то есть она является шаблоном для других (пользовательских) баз данных. При этом в новую базу данных копируются все объекты model. Если вы имеете права на ее модификацию, то можете поместить в нее новые, например, пользовательские

типы данных (домены) или таблицы. После этого, при создании пользовательских баз данных эти типы данных или таблицы всегда будут в них существовать;

- **база msdb** – это база данных поддерживает службу SQL Server Agent, включая данные о задачах, оповещения, события и репликации. В ней также хранятся сведения о всех резервных копиях и процедурах восстановления. Она занимает ориентировочно 12 Мб;
- **база tempdb** – в этой базе выполняется сортировка, объединение, сохранение временных результатов и другие изменения с данными, которые требуют дополнительного места на жестком диске. Эта база данных после инсталляции занимает примерно 2Мб, но обладает способностью разрастаться. При каждом перезапуске SQL Server база данных заново переинициализуется.

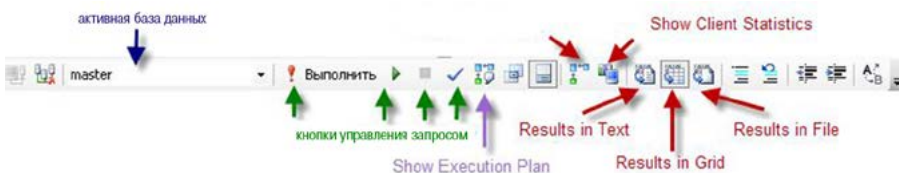
2. Папка **Security** отвечает за настройку системы безопасности данных, хранящихся в SQL Server.
3. Папка **Server Objects** содержит средства управления другими службами компьютера, для создания связанных серверов и тому подобное.
4. Папка **Replication** сохраняет структуру системы репликации данных. При необходимости можно вносить изменения в эту структуру.
5. В папке **Management** находятся средства администрирования SQL Server.



6. В SQL Server 2005/2008 служба **SQL Server Agent** имеет собственную ветку.

По центру располагается область, в которой с помощью кнопки на панели инструментов **"New Query"** (**"Создать запрос"**) или пункта меню **"File-> New-> Blank Query Window"** открывается дочернее окно, в котором можно писать SQL запросы. Таких окон можно создавать множество.

Панель инструментов, которая расположена ниже, предназначена для работы с редактором SQL и имеет следующий вид.



Первые две кнопки управляют соединением с сервером баз данных ("Connect" и "Disconnect"). Далее, в выпадающем списке указывается, какая база данных является активной. Все запросы и действия, которые вы осуществляете, будут относиться именно к ней. По умолчанию – это база данных master. Чтобы изменить активную базу данных, ее достаточно выбрать в этом списке.

Далее размещается 4 кнопки для управления запросом. Первая кнопка (**с восклицательным знаком**) позволяет проверить Ваш запрос на наличие ошибок. Причем проверяются также порядок ключевых слов в запросе. При нажатии кнопки в виде **зеленой стрелочки**, написанный SQL-запрос запустится на выполнение. В случае ошибки, на экране появится соответствующее сообщение. Кнопка с **красным квадратом** – это кнопка "СТОП", которая позволяет критически завершить работу запроса. Последняя кнопка в виде **синей галочки** предназначена для проверки запроса на наличие синтаксических ошибок.

К **важным кнопкам**, которые еще размещены на этой панели инструментов можно отнести:

- **Show Execution Plan** (показать возможный план выполнения) – позволяет перед выполнением запроса пересмотреть процедуру его выполнения, включая индексы, которые принимают участие в поиске данных;
- **Show Server Trace** (показать действительный план выполнения) – показывает статистическую информацию о выполнении запроса, то есть время его выполнения, количество полученных строк, количество ошибок и предупреждений и тому подобное;

- **Show Client Statistics** (показать статистику клиента) – просмотр статистики проведенных запросов;
- **Results in Text** (в виде текста). Такой вывод результата нужен, если необходимо выводить строки, в которых более 128 символов;
- **Results in Grid** (в виде сетки). Определяет, что результат запроса будет выведен в виде таблицы. Пункт по умолчанию;
- **Results in File** (в файл). Позволяет занести результат запроса автоматически в файл с расширением *.rpt.

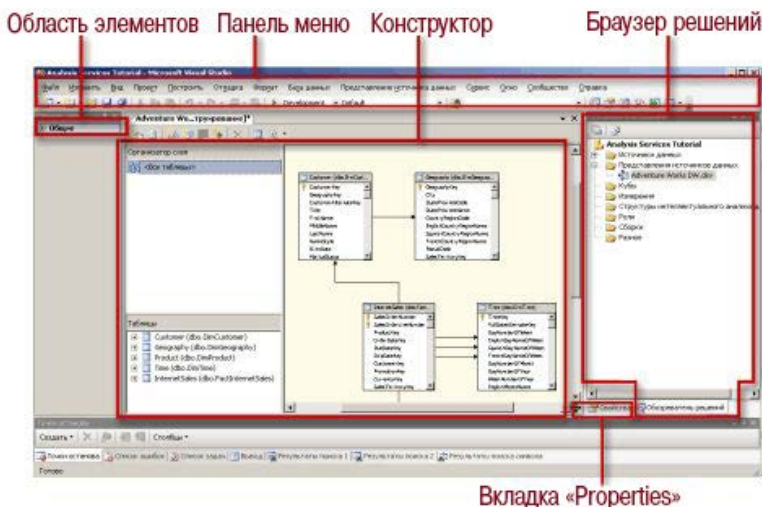
4.4. SQL Server Business Intelligence Development Studio

Business Intelligence Development Studio (BIDS) – новая версия среды Microsoft Visual Studio 2008 с дополнительными типами проектов, которые характерны для работы с бизнес-аналитикой (business intelligence – BI) в SQL Server, например: кубов, источников данных, отчетов и пакетов служб Integration Services. Это основная среда для разработки бизнес-решений, в состав которых входят проекты для служб Analysis Services, Integration Services и Reporting Services Services в программах, написанных на C # и VB.NET. Каждый тип проектов содержит шаблоны для создания объектов, необходимых для решений бизнес-аналитики, и предоставляет различные типы конструкторов, средств и мастеров для работы с такими объектами.

Среда Business Intelligence Development Studio включает ряд окон для всех этапов разработки решений и управления проектами. Например, в состав среды BIDS входят окна, которые позволяют несколькими проектами как

единым целым, а также просматривать свойства объектов в проектах и изменять их. Эти окна доступны для всех видов проектов.

На рисунке показаны окна среды Business Intelligence Development Studio в конфигурации по умолчанию.



Среда BIDS состоит из **4-х основных окон**:

- 1) **браузер решений** – отражает все решения по их проектам, включая все элементы, которые с ними связаны. С его помощью можно создавать пустые решения, куда затем добавляются новые или существующие проекты;
- 2) **окно "Properties" (Свойства)** отражает все свойства объекта. Его можно использовать для просмотра и изменения свойств файлов, проектов и решений;

- 3) **окно конструктора** представляет собой окно средств, в котором можно создавать или изменять объекты бизнес-аналитики. Конструктор предоставляет два представления одного объекта: коды и схемы. Стоит отметить, что окно конструктора недоступно до тех пор, пока в решение не будет добавлен проект, а в этом проекте не будет открыт какой-либо объект;
- 4) **область элементов** содержит элементы, которые могут использоваться в проектах бизнес-аналитики. Вкладки и элементы на этой области меняются в зависимости от того, какой конструктор и редактор используется в данный момент времени.

Другие окна среды BIDS позволяют просматривать результаты поиска, получать сведения о сообщениях ошибок и сведения, которые выводятся отладчик или конструкторами проектов. **Браузер серверов** показывает список соединений с базами данных. **Браузер объектов** показывает символы, которые могут быть использованы в проекте.

5. Работа с базой данных через утилиты и хранимые процедуры

5.1. Создание БД

Работу с базой данных, то есть ее создание, модификации и удаление можно осуществлять как средствами SQL Management Studio, так и вызывая определенный набор хранимых процедур или SQL команд. Рассмотрим все эти случаи. И начнем с самого простого, то есть с работы со средой SQL Management Studio.

Итак, запускаем SQL Server Management Studio, авторизуемся на сервере и вызываем контекстное меню папки Databases, в котором выбираем пункт **"New Database"** (**"Создать базу данных"**). Диалоговое окно, которое появится после этого, позволит создать новую базу данных и настроить все ее параметры. Осуществляется это с помощью **трех закладок**: General, Options и Filegroups.

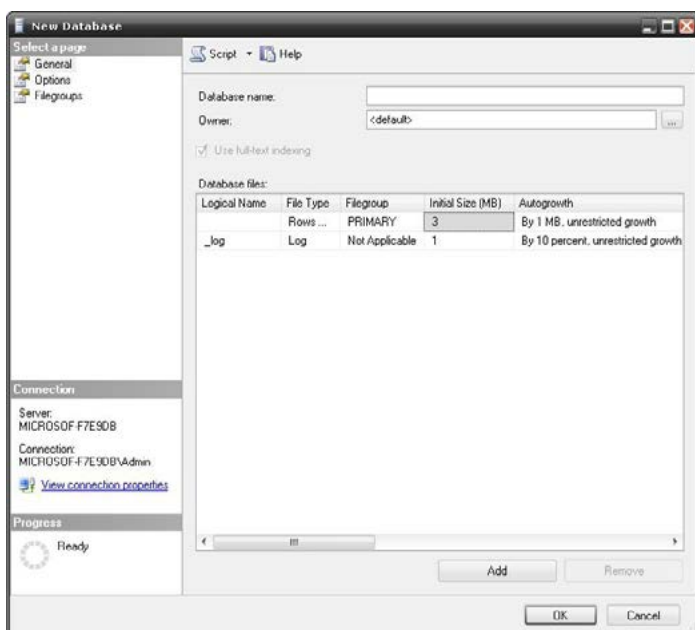
Закладка **General** позволяет указать:

- имя будущей базы данных (поле Database name);
- владельца новой базы данных (поле Owner). По умолчанию – это dbo (Data Base Owner), то есть текущий пользователь. Чтобы его изменить, нужно нажать на кнопку рядом и выбрать необходимого владельца из зарегистрированного списка пользователей данного сервера;

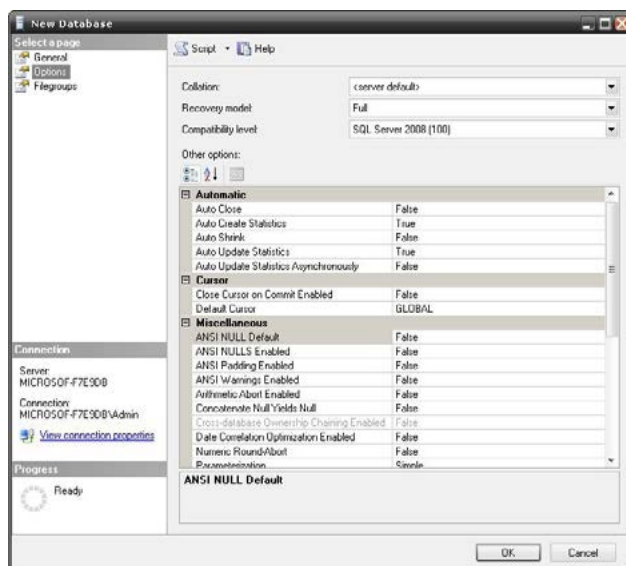
- таблица Database files позволяет создать для новой базы данных несколько файлов, указав их характеристики:
 - **Logical Name** определяет имя будущего файла базы данных;
 - **File Type** – тип файла: файл с данными (Data) или файл журнала транзакций (Log). Файл журнала транзакций по умолчанию носит имя в формате "Имя_создаваемой_базы_Log.ldf". Его параметры настройки такие же, что и для файлов базы данных;
 - **Filegroup (группа файлов)**. При создании файла необходимо указать, к какой группе файлов он принадлежит. По умолчанию выбирается группа файлов **PRIMARY**. Это говорит о том, что этот файл является первичным. **В первичном файле обычно размещаются все системные таблицы.** Если база данных состоит из нескольких файлов, один из них все равно будет относиться к группе файлов PRIMARY и имеет расширение **.mdf**. Остальным можно указать другую, собственную группу файлов (создается на закладке Filegroups).
 - **Initial Size** – размер файла, который по умолчанию равен 3 Мб для файла базы данных и 1 Мб для файла журнала транзакций.
 - **Autogrowth (авторасширение)** – определяет как будет меняться размер файла, если изначально установленного размера не хватит. Настроить эти параметры можно с помощью диалогового

окна, которое вызывается с помощью кнопки, расположенной в этом поле. При этом, прирост можно запретить, задать максимальный размер файла (поле **Maximum file size**) или указать темп роста файла:

- **In Megabytes** – файл будет увеличиваться до указанного справа количества мегабайт;
 - **By Percent** – увеличение файла будет осуществляться в процентном соотношении относительно первоначально указанного размера.
- Location отвечает за месторасположение файлов. По умолчанию это папка Data, которая находится в % папка установки% \ Microsoft SQL Server \ MSSQL.



Следующая вкладка **Options** – это дополнительные характеристики (параметры) базы данных.



Поле **Collation (параметры сортировки)** определяет способ сортировки данных в базе данных, а **Recovery model (модель обновления)** определяет режим обновления базы данных:

- full – полное восстановление;
- simple – простое (обычное) восстановление;
- bulk-logged – массовое восстановление или с неполным протоколированием.

Поле **Compatibility Level** указывает на уровень совместимости с другими версиями SQL Server. Уровень 100 говорит о совместимости только версии SQL Server 2008, 90 – с версии SQL Server 2005; 80 – версии SQL Server 2000. В SQL Server 2005 существовал уровень совместимости 70, который использовался для совместимости с SQL Server 7.0.

Далее следует список других дополнительных опций, которые условно разделены на группы.

1. **Automatic** (автоматические настройки):

- Auto close – автоматическое завершение работы SQL Server, когда отсоединился последний пользователь;
- Auto create statistics – статистические сведения будут создаваться автоматически;
- Auto shrink – отвечает за сжатие данных. Если опцию установить, то как только неиспользуемая часть общего объема базы данных будет меньше 25%, будет осуществлено автоматическое сжатие. Но оно никак не повлияет на размер базы данных, указанного при ее создании;
- Auto update statistics – определяет автоматическое обновление статистики;
- Auto update statistics asynchronously – указывает на то, будет ли осуществляться автоматическое асинхронное обновление статистики.

2. **Cursor** (курсоры):

- Close Cursor on Commit Enabled – при завершении транзакции все открытые курсоры автоматически закрываются;
- Default Cursor – область видения курсоров: глобальная (Global) или локальная (Local).

3. **Miscellaneous** (разное):

- ANSI NULL default – определяет, что значение полей новой таблицы будут иметь по умолчанию значение NULL, если параметр NOT NULL не установлен;

- **ANSI NULLS Enabled** – разрешает или запрещает использование NULL значений;
- **ANSI Padding Enabled** – разрешить или запретить ANSI заполнения;
- **ANSI Warnings Enabled** – разрешить или запретить ANSI предупреждения;
- **Arithmetic Abort Enabled** – разрешить прерывание операции в случае возникновения ситуации деления на ноль;
- **Concatenate Null Yields Null (автоокругления цифр)** – эта опция логично сопоставима с операцией умножения на ноль. Результат операции конкатенации любой величины со значением NULL всегда будет равен NULL. Например, "Hello, World" + NULL = NULL;
- **Cross-database Ownership Chaining Enabled** – включить межбазовую цепь владения всеми базами данных. Такая цепь возникает, когда один и тот же логин владеет объектами в различных базах данных. В случае включения опции его права не будут проверяться при обращении к цепи объектов. Если же данная опция выключена, то сервер баз данных сгенерирует ошибку, если, например, приложение обращается к данным базы по этой цепи владения. По умолчанию опция Cross-database ownership chaining выключена, позволяет сделать сервер базы данных более безопасным. Изменить ее значение можно с помощью системной хранимой процедуры `sp_configure`.
- **Date Correlation Optimization Enabled** – разрешить оптимизацию корреляции дат. Если данная опция

включена, SQL Server будет поддерживать статистику корреляции между любыми двумя таблицами в базе данных, которые связаны между собой ограничением внешнего ключа и столбиком типа DateTime;

- Parametrization – определение параметров: простое или принудительное;
- Quoted Identifiers Enabled – дает возможность указать название в двойных кавычках. Это необходимо, если название поля совпадает с ключевым словом языка SQL или название состоит из нескольких частей, через пробел или в названии содержатся кавычки;
- Recursive Triggers Enabled – указывает на возможность создавать рекурсивные триггеры;
- Torn page detection – возможность проверки, осуществилась ли запись разбитой на части таблицы на жесткий диск.

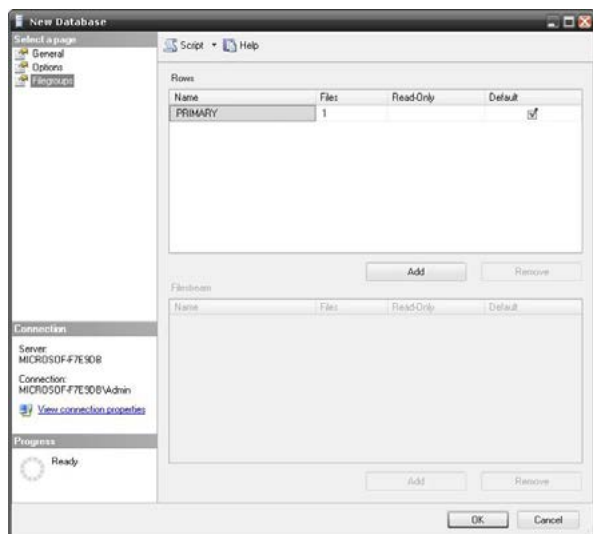
4. **Recovery (восстановление)**, в котором существует только одна опция **Page Verify (проверка страниц)**, которая определяет, как будет осуществляться проверка страницы базы данных:

- TornPageDetection – поиск частично сохраненной страницы, то есть позволяет определить, состоялась ли запись на диск разбитой на части страницы (форма повреждения данных);
- Checksum – поиск поврежденной страницы по контрольной сумме;
- None – отсутствует проверка.

5. **State** (состояние):

- Restrict Access (ограничение доступа) – если данная опция установлена, то в БД получают доступ в определенный момент времени все пользователи (MultiUser), только один (SingleUser) или доступ пользователей будет ограничен (RestrictedUser);
- Database Read-only укажет на то, что БД будет доступна только для чтения;
- Database State – состояние базы данных;
- Encryption Enabled – разрешить или запретить шифрования данных.

Вкладка **Filegroups** предоставляет информацию об использованные группы файлов.



По умолчанию, существует только одна группа файлов – PRIMARY, но вы можете добавить свою группу, указав ее название, сколько максимально файлов в него

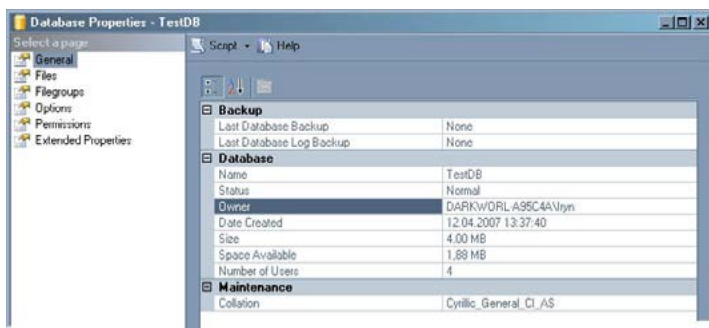
может входить, или она только для чтения и группой по умолчанию для новых файлов баз данных. Если же определенная группа файлов перестала играть функциональную роль, ее можно удалить в этом же окне.

5.2. Изменение свойств БД

Если Вам необходимо изменить параметры базы данных после создания или изменить настройки определенного файла уже созданной базы данных, тогда следует в контекстном меню необходимой базы данных (в папке Databases) выбрать пункт **"Properties" (Свойства)**.

Это окно свойств несколько схоже с окном создания базы данных, но имеет немного больше вкладок:

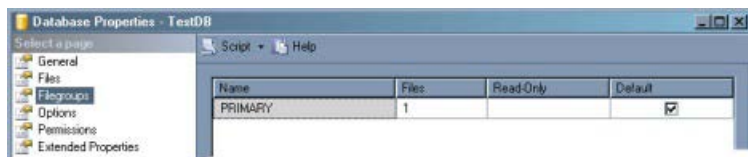
- а) вкладка **General** содержит лишь описание характеристик БД и изменить здесь ничего нельзя;



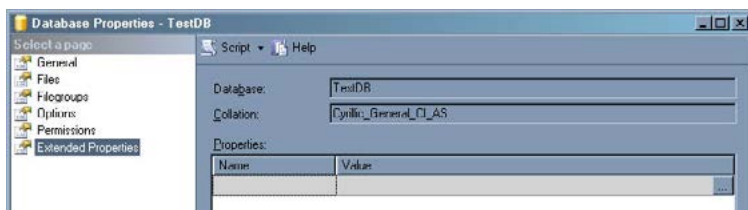
- б) вкладка **Files** содержит информацию о владельце текущей БД, из которых файлов она состоит, где они размещаются, который имеют размер и величину прироста и тому подобное;



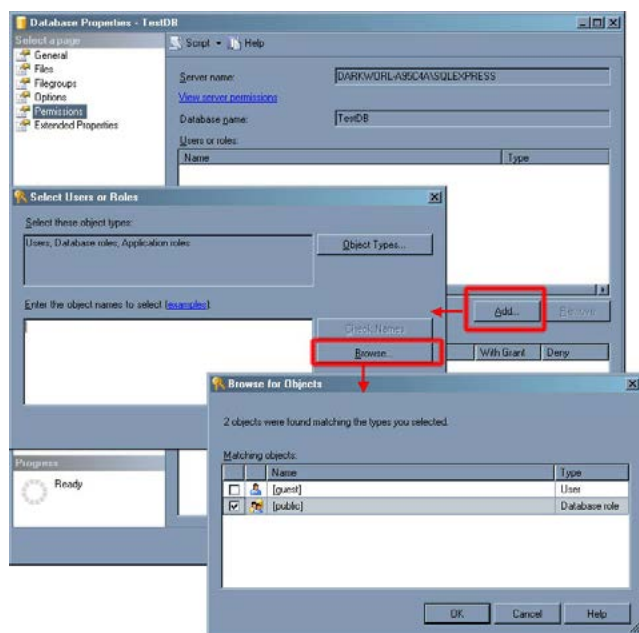
- в) вкладка **Filegroups** предоставляет информацию о группе файлов с возможностью изменения данной информации;



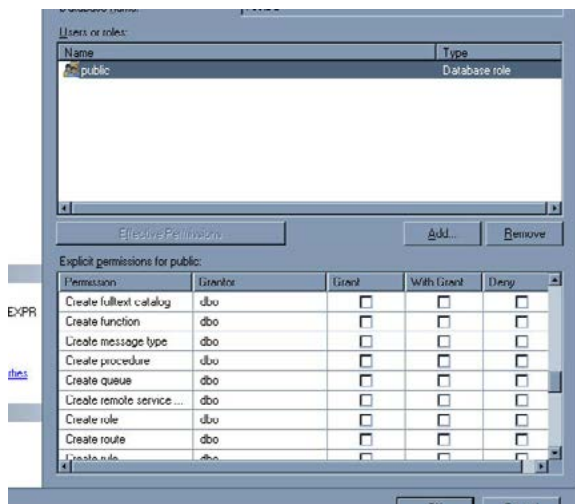
- г) четвертая вкладка **Options** отвечает за дополнительные настройки выбранной базы данных (см. Создание БД в SQL Server – Закладка Options);
- д) вкладка **Extended Properties** содержит данные о названии базы данных, ее параметры сортировки и дополнительные (пользовательские) свойства;



- е) вкладка **Permission** определяет настройки ролей базы данных.



В результате мы предоставляем всем (public) права на текущую базу данных:



Более подробно про права будет рассмотрено в конце курса MS SQL Server.

Просмотреть и изменить свойства БД можно также с помощью системных хранимых процедур (напомним, что запуск хранимых процедур нужно писать в редакторе SQL запросов). Для просмотра используется системная хранимая процедура **sp_helpdb**, а для изменения свойств – **sp_dboption**. Кстати, все системные хранимые процедуры в SQL Server имеют префикс **sp_xxx (stored procedure)**, а для запуска процедуры используется оператор **EXEC** (поскольку все хранимой процедуры являются выполняемыми).

```
/*EXEC sp_helpdb [имя_базы_данных]*/
EXEC sp_helpdb          /*просмотр свойств всех баз
данных*/
EXEC sp_helpdb TestDB   / просмотр свойств базы данных*/
TestDB*/
```

Результат:

DARKWORL-A95C4...SQLQuery1.sql Summary

exec sp_helpdb

	name	db_size	owner	dbid	created	status	compatibility le
1	master	5.25 MB	sa	1	Apr 8 2003	Status=ONLINE, Updateability=READ_WRITE, UserAcc...	90
2	model	3.19 MB	sa	3	Apr 8 2003	Status=ONLINE, Updateability=READ_WRITE, UserAcc...	90
3	msdb	7.44 MB	sa	4	Oct 14 2005	Status=ONLINE, Updateability=READ_WRITE, UserAcc...	90
4	MyTestDB	14.00 MB	DARKWORL-A95C4\Uym	8	Apr 12 2007	Status=ONLINE, Updateability=READ_WRITE, UserAcc...	90
5	Press	4.00 MB	DARKWORL-A95C4\Uym	7	Apr 4 2007	Status=ONLINE, Updateability=READ_WRITE, UserAcc...	90
6	pubs	5.50 MB	DARKWORL-A95C4\Uym	6	Apr 1 2007	NULL	80
7	tempdb	2.69 MB	sa	2	Apr 12 2007	Status=ONLINE, Updateability=READ_WRITE, UserAcc...	90
8	TestDB	4.00 MB	DARKWORL-A95C4\Uym	5	Apr 12 2007	Status=ONLINE, Updateability=READ_WRITE, UserAcc...	90

DARKWORL-A95C4...SQLQuery1.sql Summary

exec sp_helpdb TestDB

	name	db_size	owner	dbid	created	status	compatibility le
1	TestDB	4.00 MB	DARKWORL-A95C4\Uym	5	Apr 12 2007	Status=ONLINE, Updateability=READ_WRITE, UserAcc...	90

	name	fileid	filename	filegroup	size	maxsize	growth	usage
1	Test	1	c:\Program Files\Microsoft SQL Server\MSSQL1\M...	PRIMARY	3072 KB	Unlimited	1024 KB	data only
2	Test_log	2	c:\Program Files\Microsoft SQL Server\MSSQL1\M...	NULL	1024 KB	2147483648 KB	10%	log only

Синтаксис и пример на изменение свойств:

```
/*EXEC sp_dboption [имя_базы_данных [, 'название_свойства', ['TRUE' | 'FALSE']]*/
exec sp_dboption TestDB, 'Auto update statistics', 'TRUE'
```

В данном примере было изменено свойство базы данных "Auto update statistics" (автоматическое обновление статистики) на значение "true", то есть мы его включили.

5.3. Переименование БД

Чтобы переименовать БД средствами SQL Server Management Studio, необходимо в контекстном меню необходимой базы данных вызвать пункт контекстного меню **"Rename"** (переименовать).

Для переименования базы данных также предназначена системная хранимая процедура **sp_renamedb**. Для выполнения, запускаем редактор SQL запросов и напишем следующий запрос на переименование базы данных TestDB в NewTestDB:

```
/*EXEC sp_renamedb 'старое_имя_базы_данных', 'новое_имя_базы_данных'*/  
EXEC sp_renamedb 'TestDB', 'NewTestDB'
```

Но не следует забывать, что для того, чтобы иметь возможность переименовать базы данных, вы должны быть либо членом группы системных администраторов SQL Server, или ее владельцем (owner).

5.4. Удаление БД

Пункт **"Delete"** (удалить) находится в контекстном меню каждой БД. Достаточно выбрать его и подтвердить удаление. Следует отметить, что восстановлению удаленная БД не подлежит, ее возможно восстановить только из резервной копии.

Чтобы удалить определенную базу данных, вы также должны быть членом группы системных администраторов SQL Server или ее владельцем (owner). Кроме того, при удалении нужно находиться в базе данных master.

6. Введение в SQL. Что это такое? Категории команд SQL: DDL, DML, DCL

SQL (Structured Query Language) – это универсальный компьютерный язык, который используется для создания, модификации и управления данными в реляционных базах данных. Стоит отметить, что SQL не является языком программирования. Уникальность данного языка заключается в том, что он является единственным стандартным языком баз данных. Язык SQL поддерживают более сотни СУБД. И самое главное: несмотря на то, что каждая СУБД по-своему интерпретирует стандарт (вносит свои дополнения), набор стандартных инструкций SQL поддерживают все базы данных.

Что касается истории, то начало языка SQL было заложено в 1974 году, когда компанией IBM для экспериментальной реляционной СУБД **System R** был разработан специальный язык **SEQUEL (Structured English QUery Language, структурированный английский язык запросов)**, который позволял очень просто управлять данными в базе данных. Разработкой занимались Дональд Чемберлин и Рей Бойс. Целью разработки SEQUEL было создание языка, на котором мог бы пользоваться обычный пользователь, не имеющий опыта программирования.

Позже язык SEQUEL был переименован в SQL и в 1986 г. вышел ее первый стандарт ANSI (American National Standards Institute), а в 1987 г. стандарт ISO (International Organization for Standardization, Международная организация по стандартизации). Но он не был единственным языком для баз данных. В то же время Калифорнийский университет Беркли разработал собственный язык QUEL для некоммерческой реляционной СУБД Ingres (прародитель сегодняшней PostgreSQL). Но, к сожалению, она не выдержала конкуренции с SQL.

Таким образом, в 1987 г. SQL стал международным языком баз данных (SQL-86), в 1992 г. Вышла вторая расширенная версия данного стандарта (ANSI SQL-92 или SQL2), а в 1999 г. Третья версия (SQL 1999, SQL-99 или SQL3). Сегодня действует стандарт, принятый в 2003 г. (SQL 2003) с небольшими изменениями, которые были внесены в 2006 и 2008 годах. Кстати, в новый стандарт SQL была добавлена поддержка работы и с XML данными.

Работа с SQL несложная. В случае, если пользователю необходимо получить данные из базы данных, он обращается с запросом к СУБД на получение этой информации. В результате, СУБД обрабатывает этот запрос и возвращает пользователю результирующие данные. Фактически, запрос – это команда, написанная на языке SQL.

С помощью SQL можно:

- создавать, модифицировать или удалять объекты базы данных и саму базу данных;
- устанавливать связи между объектами базы данных;
- осуществлять выборку данных;

- вставлять, модифицировать и выделять данные из базы данных;
- обеспечивать целостность данных;
- другое.

Вышеописанные действия и еще много других, осуществляются с помощью инструкций (команд) SQL, которые мы будем с Вами изучать в течение следующих уроков.

Все команды языка SQL делятся на **четыре категории**:

- 1) **DDL (Data Definition Language** – Язык Определения Данных) – включает в себя инструкции, которые предназначены для создания, модификации и выделения объектов базы данных (таблицы, представления, индексы и т.д.);
- 2) **DML (Data Manipulation Language** – язык манипулирования данными) – это набор инструкций, которые предназначены для работы с данными в базе данных, то есть их вставкой, удалением, изменением и выборкой;
- 3) **DCL (Data Control Language** – Язык Управления Данными) – содержит набор инструкций, которые управляют правами доступа пользователей к объектам базы данных;
- 4) **TCL (Transaction Control Language** – Язык Управления транзакциями). В нее вносят набор инструкций, позволяющих управлять транзакциями.

Как уже было сегодня сказано, MS SQL Server для запросов использует специальную разновидность

SQL – **Transact-SQL (T-SQL)**, которая была создана совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI / ISO SQL с рядом расширений, изучению которых мы посвятим отдельную пару.

7. Создание базы данных средствами SQL

Для создания базы данных в SQL существует оператор **CREATE DATABASE**, синтаксис которого имеет следующий вид:

```
CREATE DATABASE имя_БД
                                [ON PRIMARY] /*группа файлов*/
( NAME = логичное_имя_для_файла_на_диске,
  FILENAME = 'адрес_и_имя_файла_БД'
  [, SIZE = размер_файла_БД ]
  [, MAXSIZE = максимальный_размер_файла | UNLIMITED ]
  [, FILEGROWTH = инкремент_автоматического_увеличения_
файла ]
)
[, ... n ] /*описание следующего файла базы данных*/
LOG ON      /*файл журнала транзакций*/
( NAME = логичное_имя_для_файла_журнала_транзакций_на_
диске,
  FILENAME = 'адрес_и_имя_файла_журнала_транзакций'
  [, SIZE = размер_файла_журнала_транзакций ]
  [, MAXSIZE = максимальный_размер_файла | UNLIMITED ]
  [, FILEGROWTH = инкремент_автоматического_увеличения_
файла ]
)
[COLLATE способ_сортировки_данных_в_БД ]
[FOR LOAD ] /*используется только для совместимости с SQL
Server 6.5*/
| [FOR ATTACH] /*позволяет перекомпановать набор уже
созданных файлов в единую БД*/
```

Начальный, максимальный размер файла базы данных и инкремент автоматического увеличения базы данных можно задавать как в мегабайтах, так и в килобайтах. Если

начальный размер не указывать, тогда по умолчанию он устанавливается равный размеру БД **model**. Если не задать максимальный размер, то по умолчанию принимается параметр **UNLIMITED**. Размер автоматического увеличения размера файла по умолчанию равен 1 Мб.

Например, создадим базу данных с именем MyTestDB, которая имеет аналогичное название файла, в котором хранятся ее данные. База данных состоит из одного файла, которые входит в группу файлов PRIMARY и имеет начальный размер файла – 12 Мб, максимальный – 20 Мб, а величину приращения при недостатке размера 2 Мб. Начальная величина журнала транзакций – 2 Мб, а максимальная – 12 Мб.

Создаваемая база данных использует кодовую страницу Latin1 или 1251 и имеет алфавитный порядок сортировка данных (General) – регистрозависимый (CI) и индексозависимый (AI).

```
create database MyTestDB on primary
(name = MyTestDB,
 filename = 'C:\Program Files\Microsoft SQL Server\
MSSQL.1\MSSQL\Data\MyTestDB.mdf',
 size = 12MB,
 maxsize = 20MB,
 filegrowth = 2MB )
LOG ON
(name = MyTestDB_Log,
 filename = 'C:\Program Files\Microsoft SQL Server\
MSSQL.1\MSSQL\Data\MyTestDB_Log.ldf',
 size = 2MB,
 maxsize = 12MB,
 filegrowth = 1MB)
COLLATE Latin1_General_CI_AI;
```

После запуска данного запроса, создается новая база данных с вышеописанными параметрами.

8. Модификация и удаление базы данных

Для модификации уже созданной базы данных (изменение размеров, добавление новых файлов в БД и т.д.) используется оператор SQL **ALTER DATABASE**:

```
ALTER DATABASE имя_БД
ADD FILE описание_файла [, ... n]           /*добавить новый
файл*/
[TO FILEGROUP имя_группы_файлов_в_которую_добавить_
новый_файл]
| ADD LOG FILE описание_файла [, ... n]      /*добавляет
новый файл журнала транзакций в БД*/
| REMOVE FILE логическое_имя_файла           /*удалить файл*/
| ADD FILEGROUP имя_группы_файлов            /*добавить новую
группу файлов*/
| REMOVE FILEGROUP имя_группы_файлов         /*удалить группу
файлов и все файлы, которые входят в ее состав*/
| MODIFY FILE описание_файла                /*сменить
свойства файла БД*/
| MODIFY NAME = новое_имя                   /*переименовать
БД*/
| MODIFY FILEGROUP имя_группы_файлов, свойство_группы_
файлов /*сменить свойства группы файлов, включая
параметры READONLY, READWRITE и DEFAULT*/
| SET доступ, ...n WITH завершение           /*задает
параметры READONLY, READWRITE и DEFAULT для БД. Параметр
завершения определяет момент перехода к начальному
состоянию и может принимать два значения:
* ROLLBACK AFTER n SECONDS
* ROLLBACK IMMEDIATE */
| COLLATE способ_сортировки_данных
;
/*=====*/
```

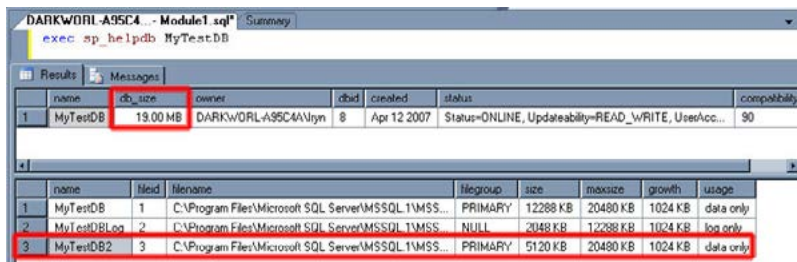
описание_файла:

```
( NAME = логическое_имя_для_файла_на_диске,
  FILENAME = 'адрес_и_имя_файла_БД'
[, SIZE = размер_файла_БД ]
[, MAXSIZE = максимальный_размер_файла | UNLIMITED ]
[, FILEGROWTH = инкремент_автоматического_увеличения_
файла ]
)
```

Например, добавим к существующей базе данных MyTestDB еще один файл с именем MyTestDB2, который имеет начальный размер – 5 Мб, максимальный – 20 Мб, а величину приращения при недостатке размера 1 Мб.

```
alter database MyTestDB
add file
( name = MyTestDB2,
  filename = 'C:\Program Files\Microsoft SQL Server\
MSSQL.1\MSSQL\Data\MyTestDB2.mdf',
  size = 5,
  maxsize = 20MB,
  filegrowth = 1MB
);
```

Чтобы убедиться, что в базу данных TestDB был добавлен еще один файл, пересмотрим ее свойства:



name	db_size	owner	dbid	created	status	compatibility
MyTestDB	19.00 MB	DARKWORLD-A95C4\Nym	8	Apr 12 2007	Status=ONLINE, Updateability=READ_WRITE, UserAcc...	90

	name	fileid	filename	filegroup	size	maxsize	growth	usage
1	MyTestDB	1	C:\Program Files\Microsoft SQL Server\MSSQL.1\MSS...	PRIMARY	12288 KB	20480 KB	1024 KB	data only
2	MyTestDBLog	2	C:\Program Files\Microsoft SQL Server\MSSQL.1\MSS...	NULL	2048 KB	12288 KB	1024 KB	log only
3	MyTestDB2	3	C:\Program Files\Microsoft SQL Server\MSSQL.1\MSS...	PRIMARY	5120 KB	20480 KB	1024 KB	data only

Как видно из рисунка, база данных MyTestDB действительно имеет новый файл, в связи с чем и размер базы данных увеличился с 14 до 19 Мб.

Для удаления базы данных используется SQL оператор **DROP DATABASE:**

```
DROP DATABASE имя_БД, имя_БД2 , ...
```

Например, удалим подопытную базу данных TestDB:

```
DROP DATABASE MyTestDB;
```

9. Импорт данных

Для импортирования и экспортирования данных в MS SQL Server 2000 использовалась утилита **Enterprise Manager**, в главном меню которой нужно выбрать пункт **Tools \ Data Transformation Services \ Import Data**. При этом должен быть выбран тот сервер, в который Вы собираетесь импортировать данные.

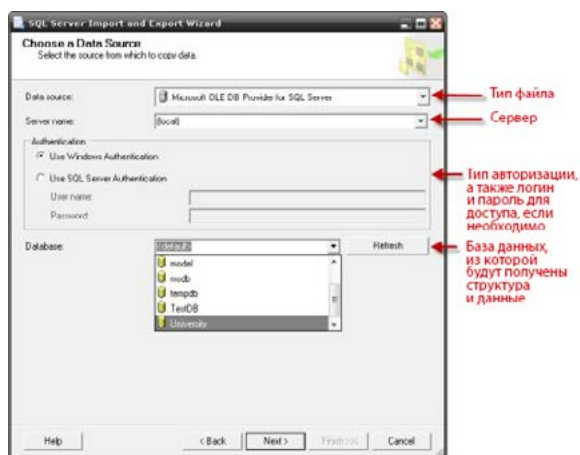
Начиная с SQL Server 2005 за импорт и экспорт данных отвечает пакет служб **SQL Server Integration Services**, а все настройки этой службы перемещения данных (extract, transformation and load – ETL) определены в папке **Data Transformation Services**. Запустить эту службу можно одним из пяти способов:

- 1) В меню "Пуск> Программы-> Microsoft SQL Server 2008-> Import and Export Data";
- 2) В окне командной строки запустить программу **DTSWizard.exe**, которая находится в папке "C: \ Program Files \ Microsoft SQL Server \ 100 \ DTS \ Binn";
- 3) В среде Business Intelligence Development Studio нужно вызвать контекстное меню папки **"SSIS Packages"** (пакеты служб SSIS), а затем выбрать пункт **"SSIS Import and Export Wizard"** (мастер импорта и экспорта служб SSIS);

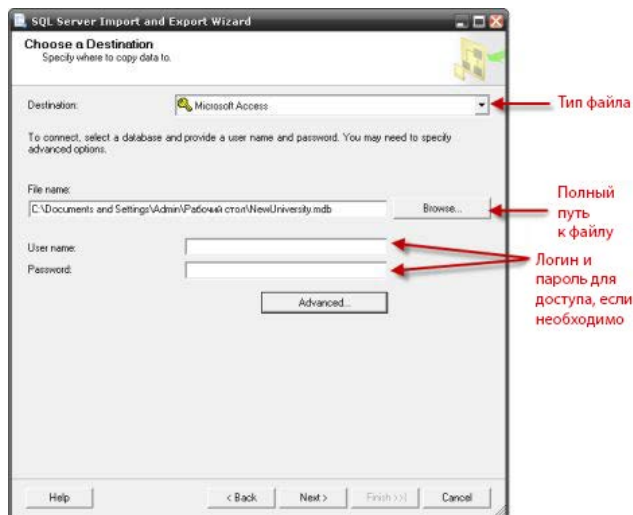
- 4) В среде Business Intelligence Development Studio в меню **"Project"** (проекты) выберите пункт **"SSIS Import and Export Wizard"** (мастер импорта и экспорта служб SSIS);
- 5) В среде SQL Server Management Studio подключитесь к серверу типа Database Engine, разверните базы данных, вызовите пункт контекстного меню необходимой базы данных **"Tasks"** (задачи), затем выберите пункт **"Import Data"** (импорт данных) или **"Export data"** (экспорт данных).

Итак, попробуем экспортировать данные некоторой базы данных University:

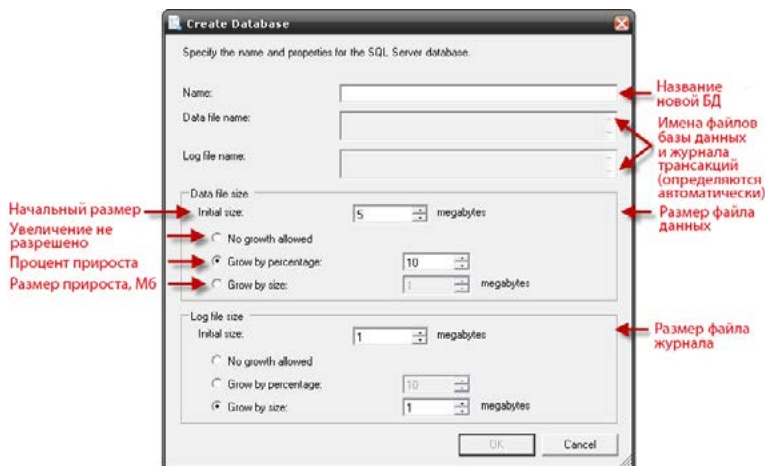
1. Запускаем одним из вышеупомянутых способов мастер импорта и экспорта данных.
2. Выбираем источник, откуда будут получены данные, то есть выбираем тип сервера SQL Server и нашу базу данных.



3. Указываем, куда будут экспортированы данные. Например, экспортируем нашу базу данных в формат MS Access, а именно в базу данных NewUniversity.mdb (предварительно ее необходимо создать).



При импорте данных в базу данных MS SQL Server окно несколько изменится: появится дополнительная кнопка "Create" (создать), которая позволит создать новую базу данных, в которую будут помещены данные. Диалоговое окно создания новой базы данных будет иметь следующий вид:

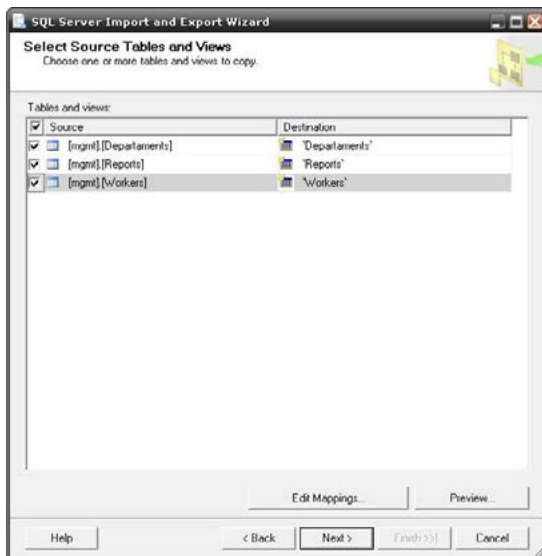


4. На следующем шаге Вы должны определить, что именно должно быть скопировано:

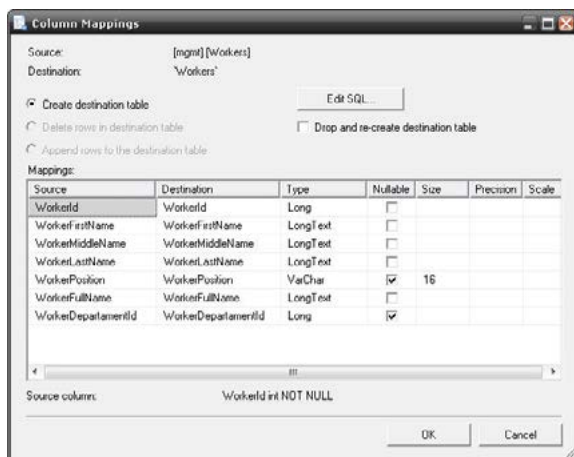
- Copy data from one or more tables or views – скопировать таблицы и представления с БД, указанной в ресурсах;
- Write a query to specify the data to transfer – написать запрос, который определяет процесс копирования.



5. Если вы выбрали первый пункт (а именно так мы и делаем), тогда на следующем шаге вам предложат список экспортируемой таблиц и представлений. Не-
нужные таблицы можно отключить, сняв соответ-
ствующий флажок в списке.

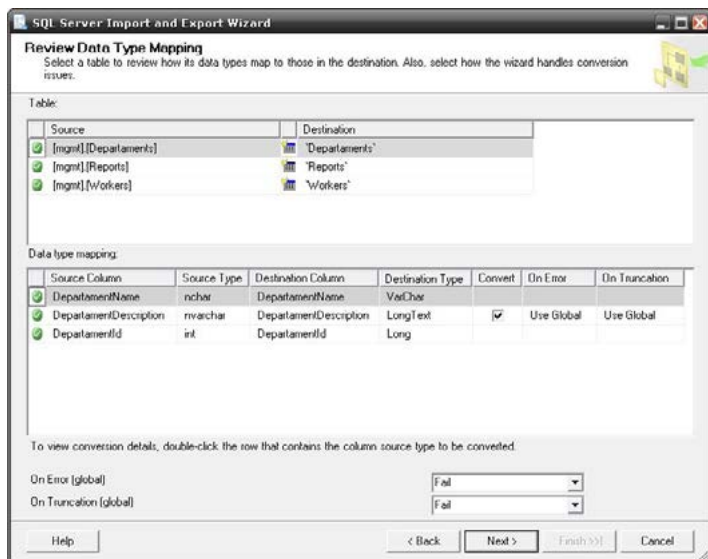


При этом, если вас не устраивает имя таблицы, которое было в экспортируемой базе данных, вы можете его изменить вместе с названием полей или предлагаемыми другими характеристиками. Для этого необходимо нажать на кнопку **"Transform..."** (изменить).

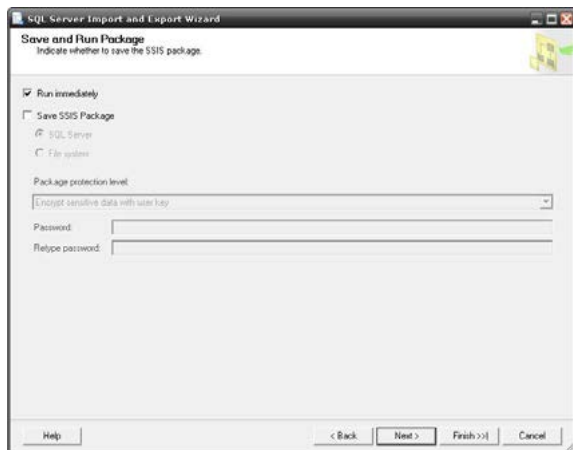


Кнопка **"Preview..."** (просмотр) позволит просмотреть содержимое экспортируемой таблицы.

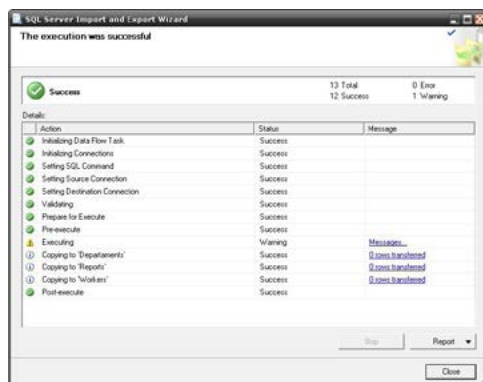
6. Посмотреть, как будут конвертированы типы данных из таблиц экспортируемой базы данных. Если Вас не устраивает предложенный вариант, можно выбрать тип, по вашему мнению, лучше подходит для определенного поля (-ей).



7. Наконец, последний существенный шаг, в котором необходимо сказать, хотите ли вы осуществить экспорт немедленно (**"Run immediately"**) и необходимо ли сохранять пакет служб SQL Server Intergration Services (**"Save SSIS Package"**) в SQL Server или в файловой системе. Здесь вы также можете указать на уровень защиты хранимого пакета, указав пароль для доступа к нему.



8. Далее вам предложат посмотреть значения, выбранные с помощью Визард, и запустится процесс экспорта.
9. В конце экспорта будет выведен подробный отчет с указанием ошибок, если таковые возникли:



Вот и все. Если после импорта базы данных вы ее не видите на сервере, тогда следует просто обновить список баз данных на сервере путем вызова контекстного меню папки Databases и выбора пункта "Refresh" (Обновить).

10. Домашнее задание

1. Установить SQL Server 2008 произвольной редакции.
2. Создать базу данных TestDatabase, которая состоит из двух файлов, размером 2 и 3 Mb соответственно.
3. Переименовать базу данных в FirstDatabase.
4. Попробовать импортировать и экспортировать произвольные базы данных.



Урок №1

Программирование и администрирование СУБД MS SQL Server

© Компьютерная Академия «Шаг»

www.itstep.org

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

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

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