

## Лабораторная работа № 1

Тема: " Ознакомление с интерфейсом СУБД MS SQL Server. Утилиты, входящие в состав СУБД "

**Цель работы:** приобрести практические навыки работы в среде разработки баз данных MS SQL Server и усвоить способы управления базами данных;

### Краткие теоретические сведения

#### *Версии MS SQL Server*

Microsoft SQL Server – это система управления клиент-серверными реляционными базами данных, ориентированная на работу под управлением операционных систем Microsoft Windows.

Требования для установки сильно зависят от потребностей приложения. Различные выпуски SQL Server удовлетворяют индивидуальным требованиям каждой организации или отдельного лица к производительности, среде выполнения и цене. Набор устанавливаемых компонентов SQL Server зависит от потребностей конкретного пользователя. В следующих разделах содержатся сведения, на основе которых из множества выпусков и компонентов, доступных в SQL Server, можно сделать наилучший выбор.

MS SQL Server включает в себя как серверную, так и клиентскую часть. Однако состав служб, включенных в поставку сервера, зависит от версии.

MS SQL Server доступен в следующих версиях (редакциях):

Выпуск SQL Server	Определение
Enterprise	Выпуск SQL Server Enterprise Edition является предложением высшего класса, обеспечивающим полный набор возможностей ЦОД с исключительно высокой производительностью, неограниченными возможностями виртуализации и исчерпывающими средствами бизнес-аналитики, что позволяет добиться высокого уровня обслуживания важнейших рабочих нагрузок и предоставить конечным пользователям доступ к анализу данных.
Standard Edition	Выпуск SQL Server Standard обеспечивает основные функции управления данными и предоставляет базу данных бизнес-аналитики для приложений, работающих в отделах и небольших организациях. Поддерживаются распространенные средства разработки в локальных системах и вычислительных облаках, что делает возможным эффективное управление базами данных с минимальными затратами ИТ-ресурсов.
Web Edition	Выпуск SQL Server Web Edition — это вариант с низкой совокупной стоимостью владения, предназначенный для размещения веб-сайтов и дополнительных веб-услуг, который по доступной цене обеспечивает масштабируемость и функции управления для небольших и крупномасштабных веб-проектов.
Разработчик	Выпуск SQL Server Developer Edition позволяет разработчикам создавать приложения любого типа на базе SQL Server. Он включает все функциональные возможности выпуска Enterprise Edition, однако лицензируется как система для разработки и тестирования, а не для применения в качестве рабочего сервера. Выпуск SQL Server Developer Edition является идеальным выбором для тех, кто создает SQL Server и тестирует приложения.
Экспресс-выпуски	Выпуск Express является бесплатной базой данных начального уровня и идеально подходит для обучения, а также для создания управляемых данными приложений, работающих на рабочих станциях и небольших серверах. Этот выпуск — лучший выбор для независимых поставщиков программного обеспечения, непрофессиональных разработчиков и любителей, создающих клиентские приложения. Если необходимы дополнительные функции базы данных, выпуск SQL Server Express можно легко обновить до версий SQL Server более высокого класса. SQL Server Express LocalDB, облегченная версия Express, которая имеет все программные функции, запускается в пользовательском режиме, быстро устанавливается, не требует настройки; количество предварительных условий для ее установки невелико.

На веб-сервере (например, под управлением служб IIS) обычно устанавливают клиентские средства SQL Server . Клиентские средства включают в себя клиентские компоненты соединения, которые используются приложениями, соединяющимися с экземпляром SQL Server. Хотя возможна установка экземпляра SQL Server на тот же компьютер, где работают службы IIS, обычно это делается только для небольших веб-сайтов, состоящих из одиночного серверного компьютера. У большинства веб-сайтов их системы IIS среднего уровня расположены на одном сервере или серверном кластере, а базы данных — на отдельном сервере или федерации серверов.

На компьютер, где работают клиент-серверные приложения, которые подключаются непосредственно к экземпляру SQL Server , можно установить только клиентские компоненты SQL Server. Установка клиентских компонентов будет хорошим выбором также и в том случае, если администрируется экземпляр SQL Server на сервере базы данных или планируется разработка приложений SQL Server .

При выборе установки клиентских средств будут установлены следующие компоненты SQL Server : компоненты обеспечения обратной совместимости, SQL Server Data Tools, компоненты подключения, средства управления, пакет средств разработки программного обеспечения и компоненты электронной документации по SQL Server .

Ограничения редакций MS SQL серверов:

Компонент	Enterprise	Standard Edition	Web Edition	Express с дополнительными службами	Express
Максимальная вычислительная мощность, используемая одним экземпляром, — Компонент SQL Server Database Engine <sup>1</sup>	Максимум, поддерживаемый операционной системой	Ограничение: меньше из 4 процессоров и 24 ядер	Ограничение: меньше из 4 процессоров и 16 ядер	Ограничение: меньше из 1 процессора и 4 ядер	Ограничение: меньше из 1 процессора и 4 ядер
Максимальная вычислительная мощность, используемая одним экземпляром, — Службы Analysis Services или Службы Reporting Services	Максимум, поддерживаемый операционной системой	Ограничение: меньше из 4 процессоров и 24 ядер	Ограничение: меньше из 4 процессоров и 16 ядер	Ограничение: меньше из 1 процессора и 4 ядер	Ограничение: меньше из 1 процессора и 4 ядер
Максимальный объем памяти для буферного пула на экземпляр Компонент SQL Server Database Engine	Максимум, поддерживаемый операционной системой	128 ГБ	64 ГБ	1410 МБ	1410 МБ
Максимальный объем памяти для кэша сегмента Columnstore на экземпляр Компонент SQL Server Database Engine	Неограниченная память	32 ГБ <sup>2</sup>	16 ГБ <sup>2</sup>	352 ГБ <sup>2</sup>	352 ГБ <sup>2</sup>
Максимальный размер данных, оптимизированных для памяти, на базу данных в Компонент SQL Server Database Engine	Неограниченная память	32 ГБ <sup>2</sup>	16 ГБ <sup>2</sup>	352 ГБ <sup>2</sup>	352 ГБ <sup>2</sup>
Максимальный объем используемой памяти на экземпляр Службы Analysis Services	Максимум, поддерживаемый операционной системой	Табличный: 16 ГБ MOLAP: 64 ГБ	Недоступно	Недоступно	Недоступно
Максимальный объем используемой памяти на экземпляр Службы Reporting Services	Максимум, поддерживаемый операционной системой	64 ГБ	64 ГБ	4 ГБ	Недоступно
Максимальный размер реляционной базы данных	524 ПБ	524 ПБ	524 ПБ	10 ГБ	10 ГБ

### *Управление MS SQL Server*

На странице «Выбор компонентов» мастера установки SQL Server выберите компоненты, которые должны быть включены в установку SQL Server. По умолчанию в дереве не выбран ни один из компонентов.

По следующим таблицам определите набор компонентов, лучше всего соответствующий вашим потребностям.

Компоненты сервера	Описание
Компонент SQL Server Database Engine	Компонент SQL Server Database Engine включает компонент Компонент Database Engine, основную службу для хранения, обработки и обеспечения безопасности данных, репликации, полнотекстового поиска, средств управления реляционными и XML-данными, интеграции аналитики с базами данных и интеграции Polybase для доступа к Hadoop и другим разнородным источникам данных, а также сервер Службы Data Quality Services (DQS).
Службы Analysis Services	Службы Analysis Services содержит средства создания приложений оперативной аналитической обработки (OLAP) и приложений интеллектуального анализа данных, а также средства управления ими.
Службы Reporting Services	Службы Службы Reporting Services включают в себя серверные и клиентские компоненты для создания, управления и развертывания табличных, матричных и графических отчетов, а также отчетов в свободной форме. Службы Службы Reporting Services являются расширяемой платформой, которую можно использовать для разработки приложений отчетов.
Службы Integration Services	Службы Integration Services представляют собой набор графических средств и программируемых объектов для перемещения, копирования и преобразования данных. Они также включают компонент Службы Data Quality Services (DQS) для служб Службы Integration Services.
Службы Master Data Services	Службы Master Data Services (MDS) — это решение SQL Server по управлению основными данными. MDS можно настроить для управления любой структурой (товары, заказчики, счета). Поддерживаются иерархии, детальная настройка безопасности, транзакции, управление версиями данных и бизнес-правила, а также использование Надстройка для Excel для управления данными.
Службы R Services (в базе данных)	Службы R Services (в базе данных) поддерживает распределенные и масштабируемые решения R на нескольких платформах с использованием нескольких корпоративных источников данных, включая Hadoop, Teradata и Linux.
Средства управления	Описание
SQL Server Management Studio	Среда SQL Server Management Studio — это интегрированная среда для доступа, настройки, управления, администрирования и разработки всех компонентов SQL Server. Среда Среда Management Studio позволяет разработчикам и администраторам, обладающим различными уровнями навыков, использовать SQL Server.
SQL Server Configuration Manager	Диспетчер конфигурации SQL Server обеспечивает базовые возможности управления конфигурациями для служб, серверных протоколов, клиентских протоколов и псевдонимов клиентов SQL Server.
Приложение SQL Server Profiler	Приложение Приложение SQL Server Profiler предоставляет графический пользовательский интерфейс для наблюдения за экземпляром компонента Компонент Database Engine или служб Службы Analysis Services.
Помощник по настройке Компонент Database Engine	Помощник по настройке ядра компонента Компонент Database Engine помогает создавать оптимальные наборы индексов, индексируемых представлений и секций.
Клиент Data Quality	Предоставляет очень простой и понятный графический пользовательский интерфейс для подключения к серверу DQS и выполнения операций очистки данных. Он также позволяет централизованно отслеживать различные действия, выполняемые во время операции очистки данных.
SQL Server Data Tools	SQL Server Data Tools содержат интегрированную среду разработки, предназначенную для создания решений для следующих компонентов бизнес-аналитики: Службы Analysis Services, Службы Reporting Services и Службы Integration Services. (Панель — среда Business Intelligence Development Studio.) SQL Server Data Tools также содержит компонент «Проекты баз данных», который предоставляет интегрированную среду для разработчиков, предназначенную для выполнения всех работ по разработке баз данных для любой платформы SQL Server (на самом предприятии и за его пределами) в Visual Studio. Разработчикам баз данных предлагается расширенный обозреватель серверов, который является компонентом Visual Studio, предназначенным для облегчения процессов создания и изменения объектов баз данных и данных в них, а также для выполнения запросов.
Компоненты связи	Устанавливает компоненты для связи между клиентами и серверами и сетевые библиотеки для DB-библиотеки, ODBC и OLE DB.

К стандартным утилитам администрирования относятся следующие приложения.

SQL Server Configuration Manager

Предоставляет следующие возможности:

- по управлению работой всех служб MS SQL Server, рассмотренных выше. Можно запустить, приостановить или полностью остановить любую из описанных выше служб, а также указать, от имени какого пользователя ее следует запустить.
- По определению параметров сетевых библиотек, которые обеспечивают взаимодействие с MS SQL Server. Можно выбрать один или сразу несколько методов доступа к серверу:
  - *именованные каналы (Named Pipes)* – технология похожа на использование сокетов, применяется в случае недоступности протоколов TCP/IP;
  - *стек протоколов TCP/IP* (используется по умолчанию) – подходит для использования через сеть Интернет;
  - *разделяемая память (Shared Memory)* – подходит для локального использования, например, веб-приложение и MS SQL Server находятся на одном компьютере. Обеспечивает максимальную скорость работу;
  - *виртуальный интерфейсный адаптер (Virtual Interface Adapter, VIA)* – используется для подключений типа сервер-сервер с применением специализированного оборудования.
- По конфигурированию сетевых библиотек клиента, используемых для доступа к MS SQL Server. После настройки методов доступа к серверу, можно произвести конфигурацию клиентских протоколов. Узел *SQL Native Client 10.0 Configuration* содержит два раздела: *Client protocols* и *Aliases* (рис. 1).

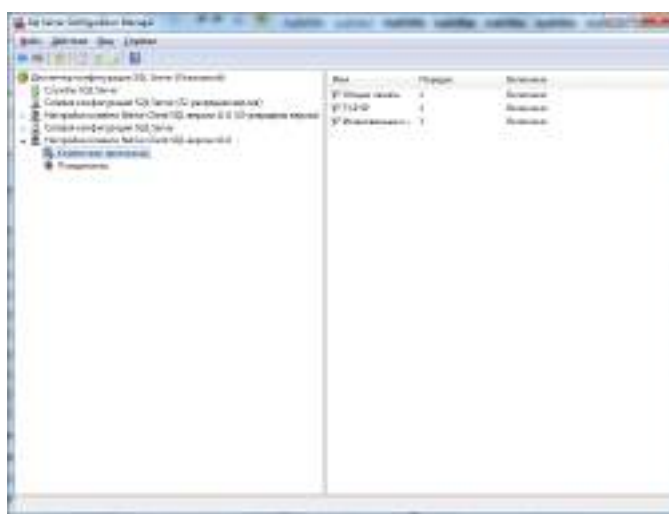


Рис. 1. Окно утилиты *SQL Server Configuration Manager*

Начиная с MS SQL Server 2000, появилась возможность подключения к серверу при помощи нескольких протоколов, например, сначала пытаемся подключиться через *Shared Memory*, если не получилось, то через TCP/IP, и в последнюю очередь через *Named Pipes*. Для определения порядка использования протоколов используется свойство *Order*.

Узел *Aliases* позволяет создавать псевдонимы для подключения к серверу. *Псевдоним (Alias)* – это альтернативное имя соединения, которое может отличаться от имени сервера. При создании псевдонима можно выбрать протокол и порт, через которые следует подключаться к серверу.

#### SQL Server Management Studio

Утилита *Management Studio* позволяет выполнять следующее:

- управлять настройками MS SQL Server;
- конфигурировать систему безопасности: управление ролями, учетными записями, удаленными серверами;
- работать со структурой баз данных: создавать, редактировать и удалять БД и элементы БД;
- управлять выполнением заданий по расписанию;
- показывать текущую активность: текущие пользователи, какие объекты заблокированы, информацию о производительности.

Перед началом работы с сервером необходимо подключиться к нему, указав следующую информацию:

- *Server Type*. Здесь следует выбрать, к какой именно службе необходимо подключиться: *Database Engine*, *Analysis Services*, *Report Server* или *Integration Services*.
- *SQL Server*. Позволяет указать, к какому серверу будет осуществляться подключение. По умолчанию имя *SQL Server* совпадает с именем компьютера.
- *Authentication Type* – способ аутентификации, можно выбрать *Windows Authentication* или *SQL Server Authentication*. Способ *Windows Authentication* использует учетную запись, под которой текущий пользователь

осуществил вход в Windows (рис. 2). *SQL Server Authentication* использует свою собственную систему безопасности.



Рис. 2. Окно соединения с SQL-сервером

#### Редактор запросов (Query Editor)

Для того чтобы написать новый запрос к базе данных, необходимо выполнить команду **New Query**, расположенную на панели инструментов *Management Studio*. В результате откроется новая вкладка, в которой можно писать SQL-код (см. рис. 3).

*Замечание:* Для выполнения запроса необходимо выполнить команду **Query – Execute (F5)**. Чтобы просто проверить правильность синтаксической записи можно воспользоваться командой **Query – Parse (Ctrl+F5)**, при этом сам запрос не будет выполнен.

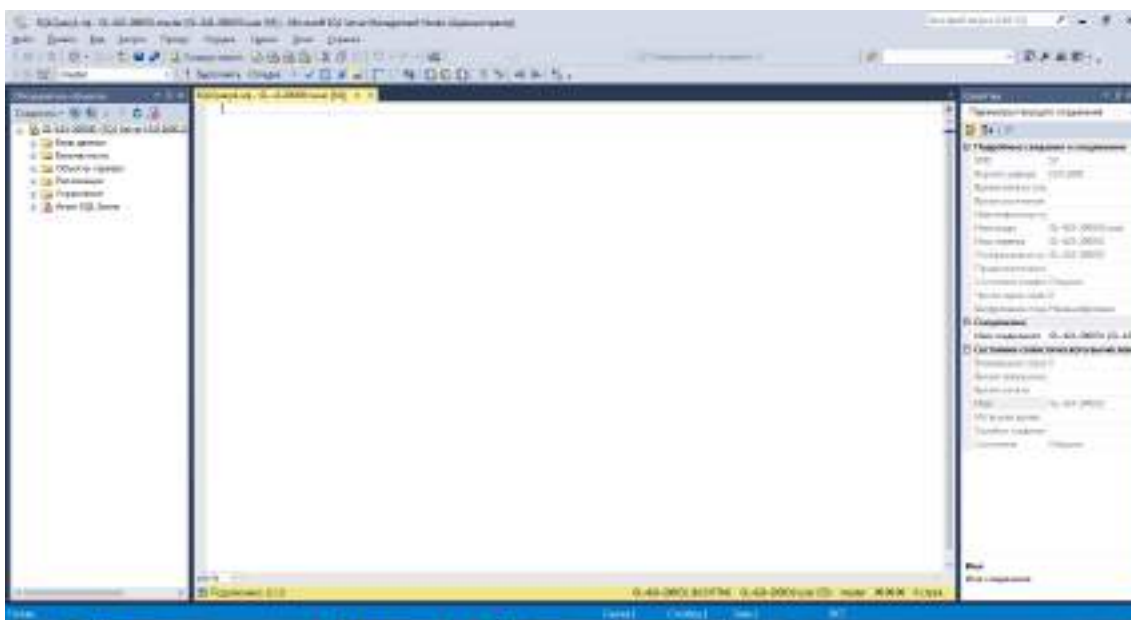


Рис. 3. Главное окно приложения *Management Studio* с окном *Редактора запросов*

Теперь будет доступно следующее:

- заголовок, в котором указывается логическое имя сервера, текущая база данных и имя пользователя, установившего соединение;
- область запроса, используемая для ввода запросов, передаваемых MS SQL Server;
- область результатов, в которой отображаются результаты выполнения запроса. Способы отображения результатов могут быть следующими:
  - *Results in Text* – результаты выводятся в виде обычного текста.
  - *Results in Grid* – результат выводится в виде таблицы, в которой можно изменять ширину столбцов, выделять нужные ячейки/строки/столбцы.
  - *Results to File* – аналогично *Results in Text*, только вывод осуществляется не на экран, а в файл.

*Management Studio* позволяет открывать несколько окон запросов и работать с несколькими базами данных одновременно. В каждом окне устанавливается собственное соединение с MS SQL Server, которое описано в *SQL Server Configuration Manager*, на основе различных учетных записей пользователей и их паролей. Для создания нового подключения используется команда **File – New – Database Engine Query**.

Содержимое области запроса текущего подключения может быть сохранено в файле на внешнем носителе командой **File – Save**.

#### Object Explorer

Позволяет осуществлять навигацию по базе данных: просматривать доступные объекты, выполнять запросы на просмотр содержимого таблиц, создавать скрипты для объектов и т.д. (рис. 4).

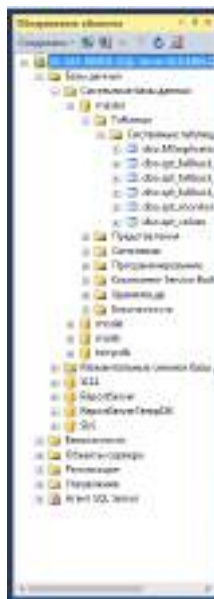


Рис. 4. Панель *Object Explorer*

## Выпадающий список баз данных

База данных, выбранная в этом списке, используется в редакторе запросов как база данных по умолчанию (см. рис. 5). Поэтому важно перед выполнением запросов, убедиться, что выбрана нужная БД. Это можно сделать либо через выпадающий список, либо при помощи команды SQL

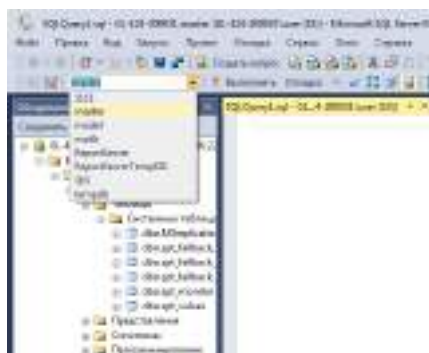


Рис. 2.5. Окно выбора текущей базы данных

### Reporting Services Configuration

Используется для конфигурации служб отчетов. MS SQL Server включает в себя встроенный web-сервер, поэтому нет необходимости в установке и настройке служб интернет-серверов IIS (*Internet Information Services*). Для создания отчетов используется Report Definition Language (RDL) – язык, основанный на XML.

## Bulk Copy Program

Утилита командной строки, предназначенная для переноса форматированных данных большого объема в MS SQL Server или из него. Например, отформатированные данные могут быть автоматически перенесены из обычного текстового файла в таблицу MS SQL Server.

## SQL Server Profiler

Позволяет в реальном времени отслеживать выполнение всех команд. Профайлер может находить «узкие» места в базе данных, определять запросы, которые долго выполняются, и наиболее часто выполняемые запросы.

Утилита sqlcmd

Утилита командной строки, которая позволяет выполнять SQL-скрипты. Данная утилита может оказаться намного эффективнее, чем *Management Studio*, когда не требуется графический пользовательский интерфейс.

## SQL Server Integration Services (SSIS)

Позволяет легко извлекать данные из любых источников через механизм OLE DB или провайдеров данных .NET и помещать их в таблицы MS SQL Server. Во время переноса данных к ним может быть применена трансформация.

#### *SQL Server Business Intelligence Development Studio*

Представляет особую версию Visual Studio и позволяет создавать пакеты для *Integration Services*, отчеты для *Reporting Services* и работать с проектами *Analysis Services*.

### **Конфигурация MS SQL Server**

Конфигурирование работы службы *MSSQLServer* может быть выполнено либо специальной хранимой процедурой, выполняемой в утилите *Management Studio*, либо графическим способом средствами этой же утилиты. Выбор способа не имеет значения, т.к. графический способ осуществляет доступ к системным данным с помощью этой же хранимой процедуры, только в более наглядной форме.

Для изменения параметров службы с помощью *Management Studio* необходимо выбрать нужный сервер и в контекстном меню выбрать команду **Properties**, и в появившемся диалоговом окне выполнить конфигурирование сервера.

На вкладке *General* отображаются основные сведения о системе: версия операционной системы, объем памяти, количество процессоров и др., а также параметры запуска служб сервера.

Вкладка *Memory* позволяет управлять выделением памяти для выполнения действий MS SQL Server: либо динамическое управление памятью, либо установка фиксированного размера.

Вкладка *Processors* позволяет управлять тем, на каких процессорах можно выполнять запросы SQL.

С помощью вкладки *Security* определяется тип аутентификации пользователей, также определяются параметры аудита доступа к серверу. Можно настроить сервер на использование определенной учетной записи, под которой будет запускаться служба *MSSQLServer*.

Вкладка *Connections* позволяет конфигурировать клиентские подключения к серверу. Если параметр равен 0, то разрешается подключение максимального количества пользователей – 32767 подключений.

С помощью вкладки *Database Settings* указываются настройки вновь создаваемых баз данных: параметры индексов и работы с устройствами резервного копирования, время восстановления базы данных.

Вкладка *Advanced* содержит некоторые общие установки сервера. Например, определяется язык по умолчанию для сообщений сервера или регулируется поддержка 2000 года, которая определяет, как будут интерпретироваться две последние цифры года.

Вкладка *Permissions* позволяет управлять именами входа и ролями, а также управлять правами на выполнение действий в MS SQL Server.

### **Структура базы данных MS SQL Server**

Каждая база данных SQL Server состоит из набора файлов операционной системы. *Создание базы данных* – это процесс указания имени файла, определения размеров и размещения файлов базы данных, а также определение параметров файла журнала транзакций.

Можно выделить три типа файлов в базах данных MS SQL Server:

1. *Первичные файлы данных.* Как правило, используется расширение MDF. В любой базе данных есть один первичный файл, который содержит данные и описание расположения всех остальных файлов БД.
2. *Вторичные файлы данных.* Как правило, используется расширение NDF. Вторичным является любой файл кроме первичного и файлов журналов. Можно иметь ноль, один или несколько вторичных файлов. Для некоторых баз данных требуется иметь несколько вторичных файлов, чтобы размещать данные по нескольким отдельным дискам.
3. *Файлы журналов.* Как правило, используется расширение LDF. В каждой БД существует по меньшей мере один файл журнала. *Журнал транзакций* содержит сведения об изменениях, происходящих в БД, т.е. при совершении некоторой транзакции (операции) в этот журнал заносятся сведения. Со временем этот журнал становится все больше, поэтому требуется следить за его размером. Основное назначение журнала транзакций – это обеспечение целостности данных. Он позволяет отменять сделанные изменения в БД. Место на диске, отводимое для файлов журналов всегда должно администрироваться отдельно от места, отводимого для данных, и никогда не должно быть частью файла данных.

Для удобства администрирования и распределения нагрузки файлы можно объединять в файловые группы, которые делятся на два вида.

1. *Первичные файловые группы.* Сюда входят первичный файл и все файлы, которые явным образом не были помещены в другую группу.
2. *Пользовательские файловые группы* – это любая группа создаваемая пользователем в БД. Файлы журналов не входят ни в одну файловую группу, они обрабатываются отдельно от обычных файлов. Новая база данных представляет собой копию базы данных *model*, все параметры которой копируются в новую базу данных. По умолчанию базы данных имеют создавать только те пользователи, которым назначены роли *sysadmin* и *dbcreator*.

База данных SQL Server – это контейнер объектов, предназначенных для хранения, записи и извлечения данных в безопасном режиме – содержит следующие компоненты:

- Таблицы. Это объекты, внутри которых хранятся данные БД. Информация хранится в таблицах в форме строк и столбцов. Столбцы таблицы определяют простейшие элементы информации, определяют назначение и тип

данных, которые в нем хранятся и объем элемента этих данных. Таблица должна состоять минимум из одного столбца. Строка образуется одним элементом информации от каждого столбца. Таблица может содержать любое число строк – оно ограничивается местом свободного пространства на диске. Строка определяет блок информации, состоящий из элементов, хранящихся в столбцах. Другое название строки – запись.

- Ограничения целостности - объект, который существует в пределах инфраструктуры отдельной таблицы, позволяют регламентировать ввод данных в таблицу в соответствии с заданными критериями.

- Индексы – объект, который существует в пределах инфраструктуры отдельной таблицы или представления и представляет собой набор поисковых (или ключевых) значений, отсортированных определенным образом. Индексы предназначены для ускорения поиска информации в таблице

- Диаграмма – визуальное представление проекта базы данных, включающая таблицы БД с именами столбцов в каждой из таблиц и связей между этими таблицами.

- Хранимые процедуры. Программный код, состоящий из одного или нескольких операторов T-SQL, реализующие те или иные действия, которые уже откомпилированы, хранятся в БД и может быть запущена в любой момент.

- Триггер - объект, который существует в пределах инфраструктуры отдельной таблицы, представляет собой фрагменты алгоритмического кода, автоматически вызываемые на выполнение в связи с тем, что к таблице, на которой они заданы, применяются операции вставки, удаления или обновления.

- Пользовательские функции по функциональности похожи на хранимые процедуры. Разница заключается в том, что возможностей у них меньше (в частности, они должны возвращать только одно значение, например, скалярное или табличное), но их удобнее использовать с точки зрения синтаксиса.

- Представления. Особый способ просмотра данных, хранящихся в БД.

- Сборка. Является файлами динамической библиотеки, которые используются в экземпляре SQL Server для развертывания функций, хранимых процедур, триггеров, определяемых пользователем статистических вычислений и определяемых пользователем типов.

#### Правила именования объектов

1. имя объекта должно начинаться с любого символа, определенного как буква. Должны ли прописные и строчные буквы рассматриваться как различные зависит от настройки сервера.

2. имена обычных объектов могут содержать 128 символов, а временных объектов – 116

3. любые имена, совпадающие с ключевыми словами СУБД или содержащие пробелы, должны быть заключены в двойные кавычки (" ") или в квадратные скобки [ ]

В SQL Server есть несколько БД, создаваемых в процессе его установки. К системным БД относят: master (главная), tempdb (временная), model (модель) и msdb.

*master.* Хранит информацию уровня всей системы, информацию инициализации SQL Server и настройки конфигурации SQL Server. Эта база данных также хранит все учетные записи для входа в систему, информацию о наличии всех остальных баз данных и о местоположении первичного файла для всех пользовательских баз данных.

*tempdb.* Хранит временные таблицы и временные хранимые процедуры. Эта база данных используется также для хранения прочей временной информации, нужной для работы SQL Server, например, для сортировки данных. При каждом запуске SQL Server создается новая чистая копия базы данных tempdb. Затем, если нужно, эта база данных растет автоматически. Если для хранения ваших временных данных требуется много места, то можно увеличить стандартный размер этой базы данных, применив команду ALTER DATABASE.

*model.* Служит образцом (шаблоном) для всех остальных баз данных, создаваемых на данной системе, в том числе и для tempdb. При создании базы данных ее начало создается как копия содержимого базы данных model, а всё остальное заполняется пустыми страницами. База данных model обязательно должна иметься в системе, потому что она применяется для воссоздания базы данных tempdb при каждом запуске SQL Server. Вы можете изменять базу данных model, добавляя туда пользовательские (определяемые пользователем) типы данных, таблицы и т.д. Если вы измените базу данных model, то каждая созданная вами база данных будет иметь измененные атрибуты.

*msdb.* Содержит таблицы, которые SQL Server Agent применяет для планирования заданий и оповещений и для записи операторов (здесь *операторы* - это люди, которые отвечают за работу заданий и оповещений). Эта база данных также хранит таблицы, применяемые для репликации.

Каждая из этих системных баз данных имеет свои собственные первичный файл данных и файл журнала. Системные базы данных хранятся в папке для хранения системных файлов.

#### **Создание базы данных**

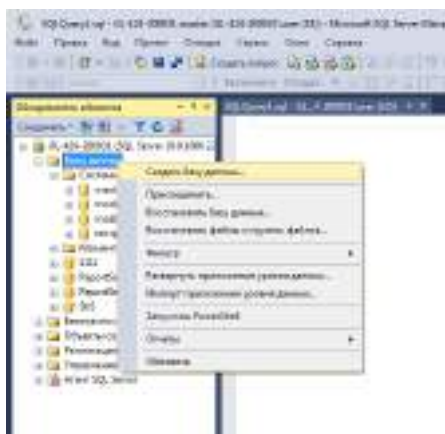
SQL Server предлагает два метода создания баз данных: средствами *SQL Server Management Studio* или применением команды CREATE DATABASE в среде *Query Editor*, которую можно сохранить в файле и запускать как сценарий.

#### Применение SQL Server Management Studio

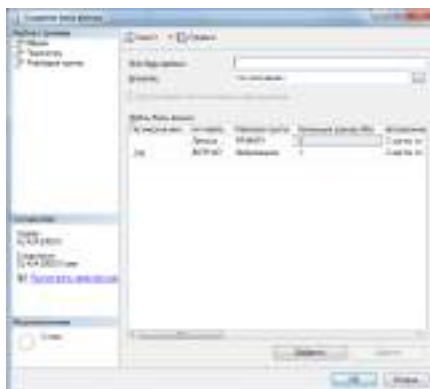
Для создания базы данных при помощи SQL Server Management Studio выполните следующую последовательность шагов:

1. Запустите на выполнение SQL Server Management Studio. В левой панели раскройте группу SQL Server, в которой находится имя сервера, на котором вы хотите создать базу данных, а затем раскройте узел самого этого сервера. Затем нажмите правой кнопкой мыши на папку Database и выберите New Database.





2. Откроется диалоговое окно свойств базы данных (Database Properties) с открытой вкладкой General (Общие).



Введите с клавиатуры имя базы данных (логическое имя) в поле Database Name. Затем нужно ввести информацию о размере файла БД и параметрах его роста. Management Studio автоматически создает первичный файл данных, с именем вашей базы данных в качестве префикса и primary в качестве имени группы файлов.

Вы можете изменить имя, местоположение и размер первичного файла, но вы не сможете изменить группу файлов для первичного файла данных. Для каждого файла данных, кроме первичного файла, вы можете задать имя пользовательской группы файлов, и, в соответствии с вашим желанием, эта группа файлов будет создана. По умолчанию, каждый файл располагается на диске в папке, в которой установлен SQL Server.

Вы можете изменить эту настройку, задав другой путь с клавиатуры или при помощи экранной кнопки для его поиска («. . »).



Нажав на селективную кнопку («. . ») вы можете задать приращение файла, выраженное в мегабайтах или в процентах от свободного места, оставшегося в файле. Вы также можете задать максимальный размер файла, указав предел роста, выраженный в мегабайтах, а можете не ограничивать рост файла. Эти настройки можно задавать при создании каждого из файлов, а можете оставить настройки, применяемые по умолчанию, и задать их позднее при помощи окна Database Properties.



Файлы журнала конфигурируются точно так же, как и файлы данных, за исключением того, что вы не сможете задать для них группу файлов, потому что они не принадлежат ни одной из групп файлов. Задайте с клавиатуры местоположение (физическое имя) и начальный размер одного или нескольких файлов журнала. Кроме того, задайте настройки автоматического роста файлов журнала, так же как это было описано для файлов данных.

3. Теперь можно перейти к вкладке Options, на которой доступно большое количество настроек.

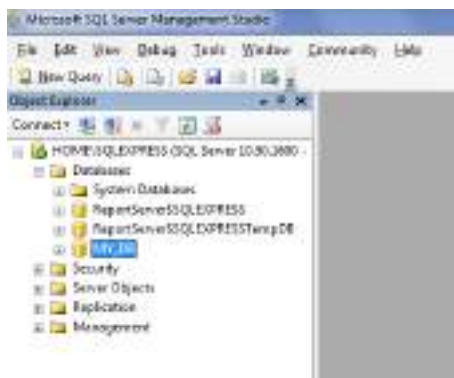


Наибольшего внимания заслуживает параметр с обозначением способа упорядочивания (Collation).

Следующим важным параметром является уровень совместимости (Compatibilty level). От него зависит то, будут ли поддерживаться некоторые синтаксические конструкции и ключевые слова, предусмотренные в версии SQL Server. С его помощью можно вернуться к использованию ключевых слов, принятых в предыдущих версиях SQL Server.

Выбор остальных параметров настройки зависит от требований к конкретной установке.

4. После того как вы настроите все файлы так, как вам это нужно, нажмите на OK, и SQL Server создаст базу данных. Вернитесь в Enterprise Manager и нажмите на папку Database сервера, в который вы только что добавили новую базу данных. Вы увидите в правой панели Management Studio, что SQL Server добавил значок-иконку для этой базы данных.



### Применение операторов T-SQL в Query Editor

Создавать или изменять свои базы данных можно при помощи операторов T-SQL, не пользуясь графическим пользовательским интерфейсом (GUI). Можно создавать свои собственные сценарии создания баз данных. Сценарий, вводимый в Query Editor, представляет собой набор SQL команд. Создание базы данных осуществляется с помощью команды:

```
CREATE DATABASE имя_базы_данных
[ON [PRIMARY] (NAME = 'логическое_имя_файла',
FILENAME = 'физическое_имя_файла'
[, SIZE = размер]
[, MAXSIZE = {максимальный_размер | UNLIMITED} ]
[, FILEGROWTH = шаг_приращения_размера [Mb | Kb | %] )
[ {FILEGROUP имя_файловой_группы} ]
[, ...n ]
[LOG ON (NAME = 'логическое_имя_файла',
```

```

FILENAME = 'физическое_имя_файла'
[, SIZE = размер]
[, MAXSIZE = {максимальный_размер | UNLIMITED} ]
[, FILEGROWTH = шаг_приращения_размера [Mb | Kb | %] )
[, ...n ]

```

Описание параметров оператора CREATE DATABASE:

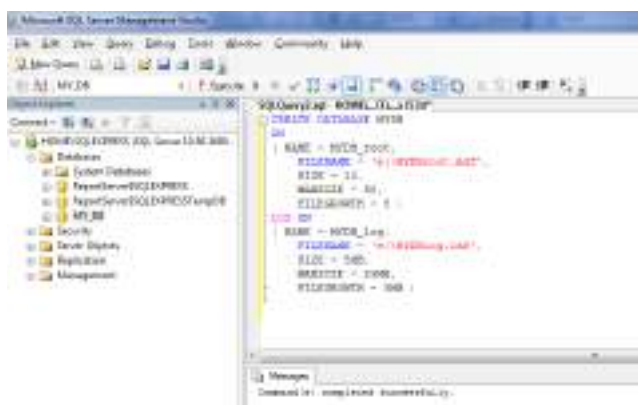
- ON – ключевое слово, указывает службе SQL Server, что в команде должны быть заданы расположение файлов данных, их имена, объем и величина объема приращения;
- NAME – логическое имя файла, по которому происходит обращение к этому файлу со стороны SQL Server. По умолчанию совпадает с физическим именем файла, определенном в параметре FILENAME;;
- FILENAME – физическое имя файла с указанием полного пути с обязательным указанием расширения файла;
- SIZE – исходный объем в мегабайтах. Параметр не обязателен, его можно опустить. Минимально возможное значение 512 Кб. Размер основного файла по умолчанию равен размеру БД **model**. По умолчанию размер дополнительных файлов данных и журнала равен 1 Мб;
- FILEGROWTH – приращение объема файла после его заполнения. Приращение можно указать в мегабайтах или процентах от текущего объема. Если этот параметр не указан, то устанавливается значение UNLIMITED, позволяющее увеличивать файлам размер без ограничений. По умолчанию приращение – 10%, если не указаны единицы, то цифра воспринимается в мегабайтах;
- LOG ON – ключевое слово, указывает службе SQL Server, что в команде должны быть заданы расположение файлов журнала, их имена, объем и величина объема приращения;
- COLLETE – необязательный параметр, указывает схему сортировки, применяемой в БД.
- PRIMARY – определяет файл как первичный или как член первичной файловой группы, если опущено, то основным файлом становится первый файл в операторе и для хранения используется первичная файловая группа;
- MAXSIZE – указывает максимальный размер, до которого может увеличиваться файл.
- FILEGROUP – определяет имя группы файлов, в которую помещается файл.

В приведенном ниже тексте дан пример создания базы данных с именем MyDB, содержащей: первичный файл данных (MyDB\_root) и один файл журнала транзакций (MYDB\_log). Наберите данный текст в панели запросов и запустите этот сценарий нажатием клавиш *F5* или *Ctrl-E*

```

CREATE DATABASE MYDB
ON
( NAME = MYDB_root,
  FILENAME = 'e:\MYDBroot.mdf',
  SIZE = 10,
  MAXSIZE = 50,
  FILEGROWTH = 5 )
LOG ON
( NAME = MYDB_log,
  FILENAME = 'e:\MYDBlog.ldf',
  SIZE = 5MB,
  MAXSIZE = 25MB,
  FILEGROWTH = 5MB )

```



Выполните следующий код, чтобы убедиться, что БД была создана с нужными параметрами:

```
sp_helpdb MYDB
```

Результат выполнения показан на рисунке



Можно создать и SQL-скрипт для создания нашей базы данных на, например, другом экземпляре SQL сервера. Для этого в окне Object Explorer в контекстном меню базы данных выбираем Script Database as -> CREATE To -> New Query Editor Window:



и в новом окне SQL редактора появится скрипт для создания нашей базы данных. Это скрипт можно сохранить и запустить потом на другом SQL сервере для создания базы данных с такой же структурой.

### **Удаление баз данных**

Базы данных можно удалять как при помощи Management Studio, так и командами T-SQL.

#### **Применение Management Studio**

1. Находясь в Management Studio, раскройте группу SQL Server, а затем раскройте имя сервера, на котором установлена база данных.
2. Раскройте папку Databases, чтобы стали видны имеющиеся базы данных.
3. Нажмите правой кнопкой мыши на имя удаляемой базы данных, а затем выберите Delete в контекстном меню.

#### **Применение команд SQL**

Помните, что удаление базы данных является неотменяемым действием. Для удаления базы данных применяется T-SQL команда DROP DATABASE. Ниже приведены примеры команд, которые удалят базу данных MyDB и все ее файлы.

Use master

(для запуска команды DROP DATABASE вы должны применять базу данных master)

GO

DROP DATABASE MyDB

(Единственным параметром этой команды является имя удаляемой базы)

GO

После удаления базы данных следует обновить резервную копию базы данных master, чтобы в ней содержалась свежая информация о пользовательских базах данных и не содержалась информация о только что удаленной базе данных, также обратите внимание, что база данных не может быть удалена, когда к ней имеют доступ пользователи. Перед удалением базы данных нужно отсоединить от нее всех пользователей.

### **Отсоединение базы данных от сервера**

Отключение базы данных означает, что она становится недоступной и может быть перенесена на другой носитель. Для того чтобы отключить базу данных, необходимо вначале отключить всех пользователей. Процедура отключения БД заключается в следующем:

1. Откройте окно Enterprise Manager и найдите узел вашей базы данных.
2. Щелкните на значке базы данных правой кнопкой мыши и выберите в контекстном меню команду Tasks (Задачи), а затем, в развернувшемся подменю, щелкните на Detach Database (Отсоединить базу данных). Щелкните на Ok.

Так как ваша база данных более недоступна ни для каких обновлений или модификаций, то ее можно копировать любым способом, применимым для копирования обычных файлов.

### **Подключение базы данных**

Когда база отключена, ее можно скопировать на несколько разных носителей, совершив, таким образом, своеобразное "клонирование". Перед подключением базы данных, убедитесь, что копия базы находится там же, где и всегда, если это не так, то при попытке подключения возникнет ошибка.

Подключение базы данных выполняется следующим образом:

1. Убедитесь, что окно Management Studio открыто, и что выделен узел вашей базы данных.
2. Щелкните на нем правой кнопкой мыши и выберите в контекстном меню команду Attach (Присоединить).
3. Вы войдете в диалоговое окно, где нужно найти местоположение вашей присоединяемой БД. Кроме того, для этой операции пользователь должен принадлежать к группе администраторов. После того, как все настройки сделаны, нажмите Ok.

**Изменение владельца базы данных** производится с помощью специальной хранимой процедуры. Владелец можно сделать любую учетную запись, которая в настоящий момент не является пользователем базы, следующим образом:

```
sp_changedbowner [ (@loginname=] 'имя_пользователя'
```

### **Переименование базы данных:**

```
sp_renamedb [ @dbname=] 'старое_имя', [ @newname=] 'новое_имя'
```

Для переименования базы данных ее необходимо перевести в однопользовательский режим работы.

Для **управления** уже существующими файлами журнала и файлами данных, добавления дополнительных файлов данных или журнала, удаления файлов, а также для работы с файловыми группами используется команда:

```
ALTER DATABASE база_данных
{ ADD FILE <указание_на_файл> [TO FILEGROUP наименование]
| ADD LOG FILE <указание_на_файл>
| REMOVE FILE логическое_имя_файла
| ADD FILEGROUP имя_группы
| REMOVE FILEGROUP имя_группы
| MODIFY FILE <указание_на_файл>
| MODIFY FILEGROUP имя_группы свойство_группы }
где <указание_на_файл> =
(NAME = 'логическое_имя_файла',
FILENAME = 'физическое_имя_файла'
[, SIZE = размер]
[, MAXSIZE = {максимальный_размер | UNLIMITED} ]
[, FILEGROWTH = шаг_приращения_размера [Mb | Kb | %] )
```

Данная команда позволяет добавлять файл в существующую файловую группу, удалять файлы (при этом удаляется и физический файл), добавлять и удалять файловые группы, изменять физические параметры уже существующих файлов, а также изменять свойства файловых групп: *READONLY*, *READWRITE*, *DEFAULT* (при определении этого свойства, в эту группу будет заноситься файлы, у которых в параметрах не определена принадлежность к группе; установленной по умолчанию изначально считается первичная файловая группа).

### **Сжатие базы данных**

Сжатие базы данных – это процесс уменьшения размеров файлов базы данных за счет удаления неиспользуемых частей файла. Существует три способа сжатия базы данных:

- автоматическое сжатие при установке соответствующего параметра в настройках базы данных;
- удаление свободного пространства из файлов базы данных с помощью утилит администрирования MS SQL Server;
- уменьшение размера указанных файлов (или файловых групп), а также очистка содержимого файлов для их последующего удаления.

**Автоматическое сжатие** данных выполняется постоянно с определенными интервалами, если установлен параметр базы данных *autoshrink*. При операциях автоматического сжатия нельзя определить, какую часть базы данных необходимо сжать. MS SQL Server пытается освободить значительную часть базы данных самостоятельно. Эти операции выполняются в период наименьшей активности пользователей.

**Сжатие всей базы данных** вручную осуществляется с использованием следующей команды:

```
DBCC SHRINKDATABASE
( database_name | database_id | 0
[ , target_percent ]
```

```
[ , { NOTRUNCATE | TRUNCATEONLY } ]
)
[ WITH NO_INFOMSGS ]
```

Описание параметров:

- *database\_name* | *database\_id* | 0 - Имя или идентификатор базы данных, которая должна быть сжата. Если указано значение 0, используется текущая база данных.

- *target\_percent* - процент свободного пространства, которое должно остаться в базе данных после сжатия.

- *NOTRUNCATE* - сжимает данные в файлах с помощью перемещения распределенных страниц из конца файла на место нераспределенных страниц в начале файла. Аргумент *target\_percent* является необязательным. Свободное место в конце файла операционной системе не возвращается, и физический размер файла не изменяется. Следовательно, если указан аргумент *NOTRUNCATE*, сжатие файлов данных незначительно. Аргумент *NOTRUNCATE* применим только к файлам данных. Файл журнала не затрагивается.

- *TRUNCATEONLY* - освобождает все свободное пространство в конце файла операционной системе, но не перемещает страницы внутри файла. Файл данных сокращается только до последнего выделенного экстенда. Аргумент *target\_percent* не обрабатывается, если указан аргумент *TRUNCATEONLY*. Аргумент *TRUNCATEONLY* оказывает влияние на файл журнала. Для усечения только файла данных используйте инструкцию *DBCC SHRINKFILE*.

- *WITH NO\_INFOMSGS* - добавляет все информационные сообщения со степенями серьезности от 0 до 10.

Права на сжатие базы данных выданы только членам роли *sysadmin* и владельцам базы данных. После сжатия базы данных выводится отчет, в котором указывается:

- количество страниц, до которых сжимается файл;
- расчетное число страниц, в которые могут быть помещены все данные файла;
- количество страниц, содержащих данные;
- количество страниц, на которые файл может быть еще сжат.

Нельзя сжать базу данных до размера меньше первоначального.

Сжатие базы данных можно осуществить также и путем *сжатия каждого ее файла* с помощью следующей команды:

```
DBCC SHRINKFILE ('имя_файла', ['конечный_размер'] [, EMPTYFILE | NOTRUNCATE | TRUNCATEONLY ])
```

Описание параметров:

- *имя\_файла* – логическое имя файла, который необходимо сжать;
- *конечный\_размер* – желательный размер (целое число в мегабайтах), который должен иметь файл после выполнения сжатия. Если этот параметр не указан или меньше минимально допустимого размера, то файл сжимается до минимально возможного размера;

- *EMPTYFILE* – выполняется перенос данных из файла в другие файлы файловой группы;
- *NOTRUNCATE* – освободившееся место не возвращается операционной системе, т.е. размер файла не уменьшается на самом деле. При этом данные располагаются более компактно и смещаются к началу файла;

- *TRUNCATEONLY* – происходит обрезание файла, начиная с последней используемой страницы. Никакого перемещения данных не происходит.

### **Резервное копирование данных**

MS SQL Server предлагает следующие типы резервного копирования информации:

- *полная копия* базы данных, которая является отправной точкой при восстановлении базы данных после сбоя, однако в зависимости от объема данных этот процесс может занимать много времени, поэтому не рекомендуется выполнять его слишком часто. Полная копия содержит все данные, содержащиеся в базе данных на момент окончания резервирования;

- *копия журнала транзакций*, необходима для фиксирования всех изменений данных, произошедших в системе с момента последнего резервного копирования. Сама копия журнала содержит сведения о транзакциях и лишь только вместе с копией базы данных позволяет вернуться к состоянию, предшествующему сбою;

- *дифференциальная копия данных* содержит изменения данных, произошедшие с момента последнего создания полной копии базы данных. При этом сохраняются только страницы подвергшиеся изменениям. Таким образом, для восстановления базы данных достаточно самой последней дифференциальной копии.

Для выполнения резервного копирования необходимо выбрать носитель, т.е. определить устройство, которое будет использоваться для создания копий. Для добавления устройства используется хранимая процедура:

```
sp_addumpdevice 'тип_устройства', 'логическое_имя', 'физическое_имя'
```

Описание параметров:

- *тип\_устройства* – тип устройства резервного копирования. Допустимые значения: *TAPE* (магнитная лента), *DISK* (магнитный диск), *Virtual\_Device* (виртуальное хранилище);

- *логическое\_имя*, *физическое\_имя* – логическое и физическое имя устройства резервного копирования соответственно.

Для создания резервной копии базы данных, журнала транзакций, файлов и файловых групп необходимо воспользоваться командой:

```

BACKUP {LOG | DATABASE } имя_БД
[ FILE = 'логическое_имя_файла', ...]
[ FILEGROUP = 'имя_группы' ]
TO логическое_имя_устройства
[ WITH
[ DESCRIPTION = 'комментарий' ]
[ DIFFERENTIAL ]
[ EXPIREDATE = 'дата' ]
[ INIT | NOINIT ] ... ]

```

Описание параметров:

- *DIFFERENTIAL* – создается дифференциальная копия базы данных;
- *EXPIREDATE* – определяется дата, после которой резервная копия считается устаревшей и может быть перезаписана;
- *INIT | NOINIT* – система осуществляет или нет инициализацию устройства.

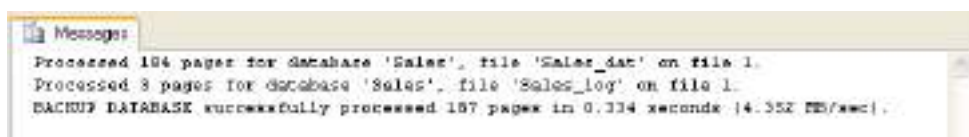
*Пример.* Создадим устройство для резервирования:

```
sp_addumpdevice 'disk', 'backupdisk', 'e:\MYDB.bak'
```

Воспользуемся функцией BACKUP для выполнения резервного копирования:

```
BACKUP DATABASE MYDB TO backupdisk
```

Результат выполнения показан на рисунке



### **Восстановление базы данных**

При восстановлении базы данных из резервной копии существующая база данных будет перезаписана. Для восстановления базы данных используется команда:

```

RESTORE {LOG | DATABASE } имя_БД
'файл_или_файловая_группа'
[ FROM логическое_имя_устройства ]
[ WITH
[ DBO_ONLY ]
[ MOVE 'логическое_имя_файла' TO 'физическое_имя' ] ... ]

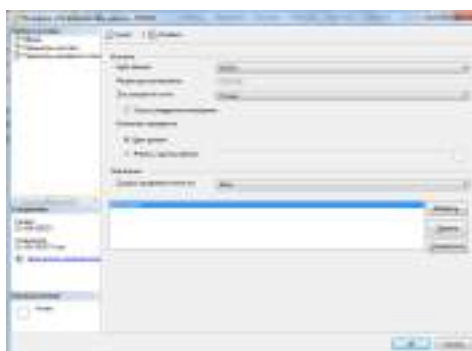
```

Описание параметров:

- *DBO\_ONLY* – разрешается доступ к восстановленной базе только владельцам;
- *MOVE* – указывает, какое физическое имя будет соответствовать восстанавливаемому файлу. По умолчанию файл восстанавливается с тем же физическим именем, которое было определено при резервном копировании.

### **Резервное копирование базы данных и создание расписаний автоматического резервного копирования**

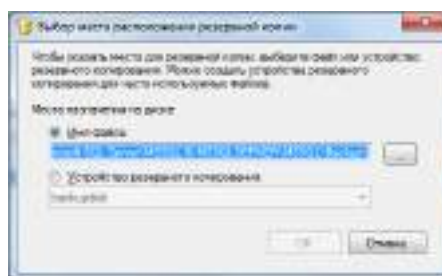
Для резервного копирования базы данных следует щелкнуть правой клавишей мыши на элементе, представляющем выбранную базу данных в правой части фрейма SQL Server Management Studio, и во всплывающем меню выбрать пункт **Tasks > Backup...** При этом открывается диалоговое окно для управления резервным копированием выбранной базы данных:



В этом окне в **Backup Type** необходимо выбрать пункт **Full** (полное копирование базы данных). После этого необходимо добавить одно или несколько устройств резервного копирования, которыми могут быть локальные или сетевые жесткие диски и другие типы носителей информации. Для добавления нового устройства резервного



копирования необходимо воспользоваться кнопкой **Add** диалога, после чего откроется выбора места расположения резервной копии базы данных:



В этом окне надо указать путь к файлу backup-a

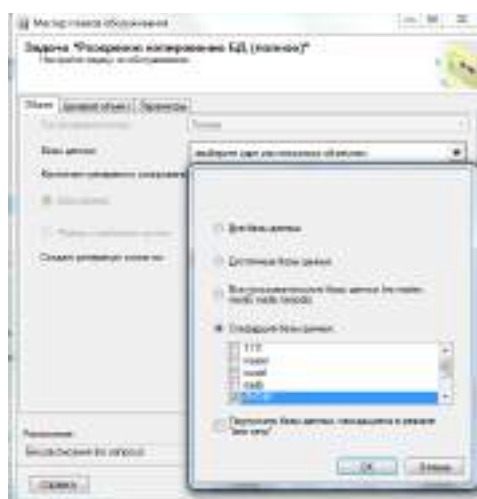
Зачастую требуется периодическое резервное копирование базы данных, для этого необходимо вызвать визард Maintenance Plan Wizard. Для этого следует щелкнуть правой клавишей мыши на элементе Management > Maintenance Plans, в правой части фрейма SQL Server Management Studio (При этом в левой части фрейма необходимо выбрать корневой элемент, представляющий подсоединенный сервер), и во всплывшем меню выбрать пункт Maintenance Plan Wizard, появится окно изображенное на рисунке



В этом окне необходимо ввести название плана и его описание, а также выбрать способ соединения с сервером и нажать кнопку Next. Окно сменится на изображенное на рисунке.



В нем надо выбрать пункт Backup Database(Full) и нажать Next.Окно сменится на изображенное на рисунке.





В нем необходимо указать базу данных для резервного копирования и нажать ОК. После необходимо указать, куда необходимо создать резервную копию (Disk, Tape).

Следующий шаг создание расписания резервного копирования. Нажав кнопку Change появится диалоговое окно настройки расписания изображенное на рисунке

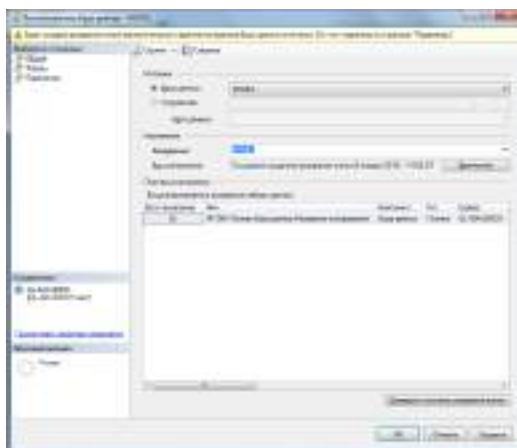


В этом окне с помощью панели **Occurs** можно определять периодичность запуска процесса резервного копирования (без задания периодичности копирования в пределах одного дня), с помощью панели **Daily frequency** определять частоту копирования в пределах одного дня, с помощью панели **Duration** определить временные рамки действия создаваемого расписания.

После подтверждения создания нового расписания кнопками **OK** на соответствующих формах, оно активируется и становится доступным, как задача планировщика заданий Microsoft SQL Server. Просмотреть эту задачу можно воспользовавшись вкладкой **Management > Maintenance Plans** в левой части главного фрейма SQL Server Management Studio и два раза щелкнув мышкой на соответствующую задачу планировщика в списке, появившемся в правой части главного фрейма SQL Server Management Studio.

### **Восстановление базы данных из резервной копии**

Для восстановления базы данных из резервной копии следует щелкнуть правой клавишей мыши по элементу, представляющему выбранную базу данных в правой части фрейма SQL Server Management Studio, и во всплывающем меню выбрать пункт **Tasks > Restore > Database....** При этом открывается диалоговое окно для управления восстановлением выбранной базы данных:



**Лабораторная работа** рассчитана на 3 часа аудиторных занятий и состоит в изучении теоретического материала и получении практических навыков по конфигурированию| созданию| просмотру| удалению |отключению|подключению базы данных. Сдача лабораторной работы заключается в ответах на контрольные вопросы и демонстрации индивидуального задания.

### **Содержание отчета:**

- 1.Название и цель работы
- 2.Задания
- 3.Результаты выполнения заданий

### **Задания**

1. Изучите утилиту SQL Server Configuration.

1.1 Запустите утилиту SQL Server Configuration Manager и с ее помощью определите список запущенных на сервере служб. Запишите этот список в отчет.

1.2 На сервере с установленным MS SQL Server 2008 с помощью утилиты Services определите параметры запуска служб MS SQL Server и запишите их в отчет. (Если нет доступа к утилите Services, то при помощи SQL Server Configuration Manager).

1.3 Определите, с помощью каких сетевых библиотек может быть установлено соединение с MS SQL Server (см. пример рис). Какие библиотеки являются активными в момент запуска? Запишите эту информацию в отчет.



Рис. Протоколы на стороне сервера, через которые к нему можно подключаться

1.4 При помощи SQL Server Configuration Manager определите, на основе каких сетевых библиотек клиент может подключаться к MS SQL Server (см. пример рис). Запишите список библиотек в отчет.

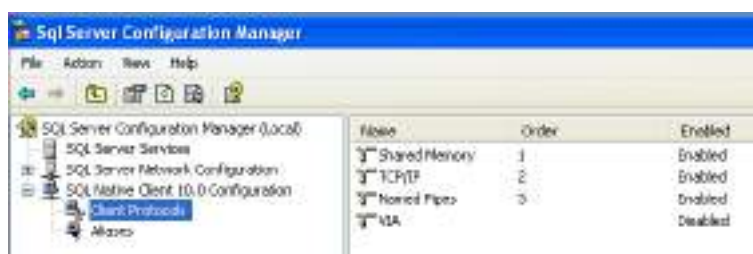


Рис. Протоколы на стороне клиента, через которые он может подключаться к серверу

## 2. Установите соединение с SQL сервером.

2.1 На рабочей станции запустите SQL Server Management Studio и выберите из списка логическое имя сервера, запущенного на вашем компьютере. Если нужного сервера нет в списке, то можно выбрать <Browse for more...> и найти требуемый сервер в списке серверов, к которым может быть выполнено подключение.

2.2 Подключитесь к серверу с использованием средств аутентификации MS SQL Server.

2.3 Для того чтобы написать новый запрос необходимо выполнить команду New Query расположенную на панели инструментов *SQL Server Management Studio*. В результате откроется новая вкладка, которая предоставляет следующие возможности:

- заголовок, в котором указывается логическое имя сервера, текущая база данных и имя пользователя, установившего соединение;
- область запроса, используемая для ввода запросов, передаваемых MS SQL Server;
- область результатов, в которой отображаются результаты выполнения запроса, а способ отображения задается кнопками Messages (в виде текста) и Results (в виде таблицы) соответственно.

2.4 С помощью команды SELECT @@version определите и запишите в отчет информацию об используемой версии MS SQL Server и операционной системы (результат запроса должен быть отображен в текстовом виде).

*Примечание:* Для выполнения запроса необходимо выполнить команду Query – Execute (F5), а для анализа правильности его синтаксической записи можно воспользоваться командой Query – Parse (Ctrl+F5).

SQL Server Management Studio позволяет открывать несколько окон запросов и работать с несколькими базами данных одновременно. В каждом окне устанавливается собственное соединение с MS SQL Server на основе различных учетных записей пользователей и их паролей. Для создания нового подключения используется команда File – New – Database Engine Query.

Содержимое области запроса текущего подключения может быть сохранено в файле на внешнем носителе командой File – Save.



8. Удалите созданную вами базу данных с именем Stud\_<фio студента>\_2 и отобразите в отчете результат выполнения оператора удаления

9. Отключить/подключить созданную вами БД Stud\_<фio студента> от сервера. Если БД создавалась на жестком диске, то переместить ее на резервный носитель и отобразите в отчете результат выполнения оператора

### **Контрольные вопросы**

1. Характеристика версий СУБД MS SQL Server
2. Характеристики инструментальных средств MS SQL Server
3. Что представляет собой БД SQL Server?
4. Объекты базы данных SQL Server
5. Способы создания баз данных в СУБД MS SQL Server?