

**Практическая работа 51**  
**Проектирование структуры базы данных.**  
**Создание в Microsoft SQL Server пустого файла данных**  
**и журнала транзакций**

**Цель работы:** Получить практический опыт проектирования структуры базы данных.

### **Перечень оборудования и программного обеспечения**

Персональный компьютер  
Microsoft Office (Word, Visio, Access)  
MS SQL Server

### **Краткие теоретические сведения**

Базу данных часто отождествляют с набором таблиц, которые хранят данные. Но это не совсем так. Лучше сказать, что база данных представляет хранилище объектов. Основные из них:

- **Таблицы:** хранят собственно данные
- **Представления (Views):** выражения языка SQL, которые возвращают набор данных в виде таблицы
- **Хранимые процедуры:** выполняют код на языке SQL по отношению к данным к БД (например, получает данные или изменяет их)
- **Функции:** также код SQL, который выполняет определенную задачу.

В SQL Server используется два типа баз данных: системные и пользовательские. Системные базы данных необходимы серверу SQL для корректной работы. А пользовательские базы данных создаются пользователями сервера и могут хранить любую произвольную информацию. Их можно изменять и удалять, создавать заново. Собственно это те базы данных, которые мы будем создавать и с которыми мы будем работать.

#### **Системные базы данных**

В MS SQL Server по умолчанию создается четыре системных баз данных:

- **master:** эта главная база данных сервера, в случае ее отсутствия или повреждения сервер не сможет работать. Она хранит все используемые логины пользователей сервера, их роли, различные конфигурационные настройки, имена и информацию о базах данных, которые хранятся на сервере, а также ряд другой информации.
- **model:** эта база данных представляет шаблон, на основе которого создаются другие базы данных. То есть когда мы создаем через SSMS свою бд, она создается как копия базы model.

- msdb: хранит информацию о работе, выполняемой таким компонентом как планировщик SQL. Также она хранит информацию о бекапах баз данных.

- tempdb: эта база данных используется как хранилище для временных объектов. Она заново пересоздается при каждом запуске сервера.

Все эти базы можно увидеть через SQL Server Management Studio в узле Databases -> System Databases:

Эти базы данных не следует изменять, за исключением бд model.

Если на этапе установки сервера был выбран и установлен компонент PolyBase, то также на сервере по умолчанию будут расположены еще три базы данных, которые используются этим компонентом: DWConfiguration, DWDiagnostics, DWQueue.

Каждая база данных SQL Server имеет как минимум два файла, с ней ассоциирующихся: один файл данных, в котором непосредственно хранятся данные и как минимум один файл журнала транзакций. Журнал транзакций это основной компонент системы управления базами данных (СУБД). Все изменения в базе данных записываются в журнал транзакций. Используя эту информацию, СУБД может определить какая транзакция какие изменения внесла в данные SQL Server.

Оператор CREATE DATABASE используется для создания базы данных Microsoft SQL Server. Опция этой команды LOG ON используется для определения журнала транзакций создаваемой базы данных. Впервые созданные данные помещаются в файл данных, а запись изменений этих данных помещается в файле журнала транзакций. Как только делаются изменения в базе, журнал транзакций растет. Поскольку большинство изменений вносимых в базу, журналируются, нужно будет отслеживать размер журнала транзакций, потому что, если данные постоянно меняются, журнал соответственно вырастает. Каждая контрольная точка Microsoft SQL Server гарантирует что все записи в журнале и все модифицированные страницы данных корректно записаны на диск. Файл журнала транзакций используется Microsoft SQL Server в процессе операции восстановления базы данных, чтобы зафиксировать завершенные транзакции и откатить незавершенные. Информация, записываемая в журнал транзакций, включает:

- время начала каждой транзакции;
- изменения внутри каждой транзакции и информацию для их отката (для этого используются снимки страниц данных до, и после транзакции);
- информация о распределении памяти для страниц БД (выделении и изъятии экстенда);
- информация о завершении или откате каждой транзакции.

Эти данные Microsoft SQL Server использует в целях повышения целостности данных. Журнал транзакций используется при старте SQL Server, для того чтобы отменить сделанные изменения и установить состояние базы данных на момент, предшествующий началу изменений. При запуске SQL Server для каждой БД начинается процесс регенерации (recovery). SQL Server определяет те транзакции, которые необходимо откатить. Это происходит в том случае, когда неизвестно все ли изменения из кэша записаны на диск. Поскольку при выполнении контрольной точки все изменения сбрасываются на диск, то с нее и стартует процесс регенерации, который производит фиксацию транзакций на диск. Все изменения на страницах, сделанные до контрольной точки, уже записаны на диск, поэтому нет смысла для сброса их на диск еще раз и изменения, выполненные до контрольной точки, не берутся к рассмотрению. При необходимости отката транзакции SQL Server копирует снимки страниц данных до изменений, сделанных с момента запуска оператора BEGIN TRANSACTION. Вы можете использовать журнал транзакций при восстановлении базы данных. В этом случае журналируется фиксация транзакций. В процессе фиксации транзакций SQL Server сохраняет все сделанные изменения в базе данных на диске. Журнал транзакций полезен для устранения ошибок в базе данных, ошибок транзакций и позволяет обеспечить целостность данных.

Поэтому одно из самых важных решений, которое необходимо принять при создании базы данных — это решение о структуре и размещении файлов данных и журналов транзакций. Неверное решение может существенно снизить производительность и надежность работы вашего приложения.

Для любой базы данных создаются файлы самой базы данных и файлы журналов транзакций. В файлах базы данных хранится вся информация о самой базе данных. В файлы журналов транзакций производится последовательная запись всех изменений, которые вносятся в базу данных. Минимальный набор файлов для любой базы данных (он же используется по умолчанию) содержит один файл для самой базы данных и один файл для журнала транзакций.

В каждой базе данных обязательно есть один главный (primary) файл. По умолчанию для него используется расширение mdf (хотя использовать именно такое расширение не обязательно — для любых файлов баз данных и журналов транзакций расширения могут быть любыми, а могут и отсутствовать). Удалять этот файл нельзя. Для базы данных можно создать и дополнительные файлы (secondary), для которых по умолчанию используется расширение ndf. Точно так же есть главный и дополнительные файлы у журналов транзакций, для них по умолчанию используется расширение ldf.

В принципе, база данных может вообще обходиться без файлов. Вся необходимая информация при этом будет храниться на неформатированном диске. Такой вариант называется использованием неформатированных разделов (rawpartitions). Однако, в отличие от связки Unix/Oracle, связка Windows/SQL Server ничего не выигрывает от применения неразмеченных разделов, и поэтому этот вариант используется редко.

**Размещение файлов баз данных и журналов транзакций.** По умолчанию и файлы баз данных, и файлы журналов помещаются в каталог C:\Program Files\Microsoft SQL Server\MSSQL.X\MSSQL\Data (где X — номер экземпляра SQL Server 2008). Такое размещение, конечно, не оптимально.

Идеальный вариант, с точки зрения размещения файлов баз данных, — поместить их на отдельный внешний аппаратный RAID-массив. Причем на этом RAID-массиве не должно быть ничего, кроме файлов базы данных, и, кроме того, на нем должно быть как минимум 50% пустого пространства. Такой вариант обеспечивает ряд преимуществ:

- RAID-массив (в зависимости от выбранного уровня) обеспечивает высокую производительность и отказоустойчивость;
- внешний RAID-массив делает процесс восстановления больших баз данных предельно простым и быстрым. Если на сервере возникли какие-то проблемы, а с файлами базы данных все в порядке, достаточно просто подключить RAID-массив к другому серверу и присоединить базу данных;
- если вы выберете внешний RAID-массив, который входит в список совместимого оборудования (HardwareCompatibilityList) для кластеров WindowsServer, то при необходимости вы сможете еще больше повысить отказоустойчивость за счет создания кластера;
- если на диске будет не менее половины пространства свободно, то этим вы обеспечите себе отсутствие проблем при выполнении различных служебных операций, например, при перестроении кластеризованных индексов для больших таблиц.

Внешний RAID-массив — это идеальный вариант. Иначе рекомендуется использовать для файлов баз данных отдельный быстрый жесткий диск. Категорически не рекомендуется помещать на тот же диск, где находятся файлы баз данных, программные файлы операционной системы и SQL Server. Помните также, что на контроллерах доменов для разделов, на которые помещается база данных ActiveDirectory (по умолчанию она находится в каталоге C:\Windows\NTDS), отключается кэширование на запись: падение производительности может быть просто устрашающим.

Конечно же, никогда нельзя сжимать рабочие файлы баз данных средствами NTFS или помещать их на сжатые диски.

**Размещение файла журнала транзакций.** Требования по производительности к ним намного меньше, чем к файлам баз данных (одна из причин связана с тем, что в файлы журналов транзакций записываются только изменения, которые вносятся в базу данных — команды SELECT на них, за исключением специальных случаев, влияния не оказывают). Как размещать эти файлы, зависит от требований к базе данных и бюджета предприятия. Далее представлены разные варианты, начиная от наиболее желательного и заканчивая наименее удачным:

- второй RAID-массив;

- отдельный набор дисков на том же RAID-массиве, что и файлы баз данных;
- два обычных диска, которые зазеркалированы по отношению друг к другу;
- просто обычный отдельный диск;
- размещение на том же диске, на котором размещены файлы баз данных (этот вариант наименее желателен, но именно он по умолчанию выбирается SQL Server 2008 в расчете на однодисковые системы).

Главное, что нужно постараться обеспечить, — это размещение журналов транзакций на другом физическом диске по отношению к файлам баз данных. Если вы помните это условие, то в случае отказа жесткого диска сможете провести восстановление на момент сбоя (конечно, при условии, что резервное копирование все-таки производилось). Если же файлы баз данных и журналы транзакций находились на одном диске, и этот диск отказал, восстановить базу данных удастся только на момент создания последней резервной копии. В этом случае пользователям придется заново вносить данные, например, за последний день или за последнюю неделю — в зависимости от того, когда последний раз производилось резервное копирование.

Второе решение, которое нужно принять при создании базы данных, — выбор размера файлов баз данных и журналов транзакций.

Размер файлов баз данных полностью зависит от задачи, для которой используется эта база данных. Можно сразу создать большие файлы баз данных или настроить для них режим автоматического приращения, когда файлы при необходимости будут автоматически увеличиваться. Если такая возможность имеется, всегда нужно с самого начала создавать файлы максимального размера (или, по крайней мере, настраивать автоприращение сразу большими частями, например, в несколько Гбайт), даже несмотря на то, что в течение продолжительного времени значительная часть этих файлов использоваться не будет. Аргументация здесь проста — таким образом, вы снижаете фрагментацию файлов баз данных, повышая производительность. По умолчанию для файлов баз данных настраивается худший вариант — автоприращение маленькими "порциями": 1 Мбайт для файлов базы данных и 10% от существующего размера для файлов журналов транзакций.

Настройку режима автоприращения при помощи графического интерфейса можно выполнить в окне NewDatabase при создании новой базы данных или на вкладке General свойств базы данных в SQL Server Management Studio. Режим автоприращения устанавливается в соответствующей строке для каждого файла базы данных нажатием на кнопку в столбце Autogrowth (Автоматический рост) свойств данного файла.

В SQL Server 2008 есть возможность, которая называется немедленной инициализацией файлов (instantfileinitialization). Она позволяет не заполнять файлы данных нулями, что резко сокращает время, требуемое для создания файлов баз данных или их увеличения.

Эти же самые принципы относятся и к настройке автоприращения файлов журналов транзакций (заметим только, что немедленная инициализация файлов при создании файлов журналов не используется). Однако для файлов журналов транзакций возникает еще один вопрос: а какой размер журнала транзакций нужен для конкретной базы данных? Ответ на этот вопрос зависит от множества факторов:

- первый фактор — в каком режиме используется база данных. Если это база данных OLTP (т. е. данные в ней изменяются постоянно, чаще всего пользователями при помощи клиентских приложений), к которой относится абсолютное большинство используемых на предприятиях баз данных, то Microsoft рекомендует устанавливать для журналов транзакций размер от 10 до 25% от общего размера файлов баз данных. Для баз данных DataWarehouse (архивные хранилища, которые в обычном режиме используются только на чтение и пополняются, как правило, средствами массовой загрузки данных или пакетами SSIS/DTS) достаточно будет и нескольких процентов от объема файлов баз данных;

- второй фактор — какой режим восстановления настроен для базы данных. Режим восстановления (recovery model) настраивается на вкладке Options свойств базы данных. Режим восстановления Simple предъявляет минимальные требования к размеру файлов журнала (поскольку старые записи в журнале сразу же перезаписываются), а режим восстановления Full требует файлов журнала намного большего размера;

- если база данных работает в режиме восстановления Full, то записи в журнале транзакций будут копиться до бесконечности, пока не будет произведено резервное копирование журнала транзакций или журнал транзакций не будет очищен вручную. Поэтому при определении необходимого размера файлов журналов транзакций нужно учитывать и частоту резервного копирования.

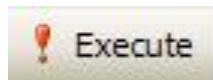
## Создание файла данных

Новую БД можно создать, используя стандартные команды языка T-SQL. Для создания новой БД необходимо сделать активную БД "Master". Это можно сделать либо выбором ее из выпадающего списка БД на панели инструментов, либо набором команды USE Master на вкладке нового запроса.



Замечание: Все команды языка T-SQL набираются на вкладке нового запроса (SQLQuery). Для того чтобы создать новый запрос на панели инструментов необходимо нажать кнопку

Для выполнения команд языка T-SQL на панели инструментов необходимо нажать кнопку



или на вкладке нового запроса набрать команду GO.

Замечание: В Microsoft SQL Server БД состоит из двух частей:

Файл данных - файл, имеющий расширение mdf и где находятся все таблицы и запросы;

Файл журнала транзакций - файл, имеющий расширение ldf, содержит журнал, где фиксируются все действия с БД. Данный файл предназначен для восстановления БД в случае её выхода из строя.

Для создания нового файла данных используется команда CREATE DATABASE, которая имеет следующий синтаксис:

```
CREATE DATABASE <ИмяБД>  
ON (Name=<Логическое имя>,  
FileName=<Имя файла>  
[Size=<Нач.размер>],  
[Maxsize=<Макс.размер>],  
[FileGrowth=<Шаг>])  
[LOG ON  
(Name=<Логическое имя>,  
FileName=<Имя файла>  
[Size=<Нач.размер>],  
[Maxsize=<Макс.размер>],  
[FileGrowth=<Шаг>])
```

Здесь:

Имя БД - имя создаваемой БД

Логическое имя - определяет логическое имя файла данных БД, по которому происходит обращение к файлу данных.

Имя файла - определяет полный путь к файлу данных.

Нач.размер - начальный размер файла данных в Мб.

Макс.размер - максимальный размер файла данных в Мб.

Шаг - шаг увеличения файла данных, либо в Мб либо в %.

Параметры в разделе LOG ON аналогичны параметрам в разделе CREATE DATABASE. Однако они определяют параметры журнала транзакций.

**Пример:** Создать БД "Students", расположенную в файле "D:\Students.mdf" и имеющую начальный размер файла данных 1мб., максимальный размер файла данных 100мб. и шаг увеличения файла данных равный 1мб. Файл журнала транзакций данной БД имеет имя "StudentsLog" и расположен в файле "D:\Students.ldf". Данный файл имеет начальный размер

равный 1мб., максимальный размер равный 100мб. и шаг увеличения равный 1мб.

```
CREATE DATABASE Students
ON (Name = Students,
FileName = 'D:\Students.mdf',
Size = 1Mb,
Maxsize = 100Mb,
FileGrowth= 1Mb)
LOG ON
(Name = StudentsLog,
FileName = 'D:\Students.ldf',
Size = 1Mb,
Maxsize = 100Mb,
FileGrowth = 1Mb)
```

### **Управление базами данных при помощи команд языка T-SQL**

В языке запросов T-SQL с БД возможны следующие действия:

Отображение сведений о БД: EXEC sp\_helpdb<Имя БД> ;

Изменение параметров БД: EXEC sp\_dboption<Имя БД>, <Параметр>, <Значение> ;

Добавления новых файлов, удаление файлов и переименования файлов, входящих в БД:

```
ALTER DATABASE <Имя БД>
```

```
ADD FILE (<Параметры>)|
```

```
REMOVE FILE <Логическое имя файла>|
```

```
MODIFY FILE (<Параметры>)
```

где, раздел ADD FILE - добавляет файл, REMOVE FILE - удаляет, а раздел MODIFY FILE - изменяет параметры файла;

Сжатие всей БД: DBCC SHRINKDATABASE <Имя БД> ;

Сжатие конкретного файла БД: DBCC SHRINKFILE <Логическое имя файла> ;

Переименование БД: EXEC SP\_RENAMEDB <Имя БД>, <Новое имя БД> ;

Удаление БД: DROP DATABASE <Имя БД>.

Замечание: Вышеперечисленные команды используют следующие параметры:

<Имя БД> - имя БД с которой производится действие;

<Параметр> - изменяемый параметр;

<Значение> - новое значение изменяемого параметра;

<Параметры> - параметры файла БД, аналогичные параметрам, используемым в команде CREATE DATABASE ;

<Логическое имя файла> - логическое имя файла, входящего в БД;

<Новое имя БД> - новое имя БД.



## Задания

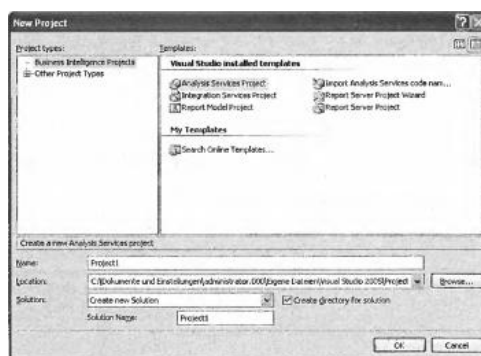
- 1 Изучить теоретические сведения.
- 2 В соответствии с вариантом задания выполнить проектирование базы данных.

## Порядок выполнения работы

Первым шагом в создании аналитического приложения является создание нового проекта. Для создания проекта в меню File выберите New | Project. В диалоговом окне New Projects в панели Project types щелкните по папке Business Intelligence Projects. В панели Templates выберите Analysis Services Project. Наберите имя проекта и его местоположение в текстовых полях Name и Location соответственно. Новый проект будет создан после того, как вы щелкнете мышью по кнопке ОК.

Новый проект всегда создается в новом решении. Следовательно, решение является наибольшей единицей управления в Business Intelligence Development Studio и всегда содержит один или более проектов. Если панель Solution

Explorer, которая позволяет вам просматривать и управлять объектами решения или проекта, невидима, вы можете ее увидеть, выбрав в меню View команду Solution Explorer.



Посмотрите на панель Solution Explorer с новым созданным проектом. Ниже узла проекта помимо других присутствуют следующие папки:

- Data Sources хранит информацию о соединении с исходной базой данных;
- Data Source Views содержит информацию, связанную с подмножеством таблиц в исходной базе данных;
- Cubes содержит все кубы, которые принадлежат данному проекту;
- Dimensions содержит все измерения. Analysis Services поддерживает три типа измерений. Разделяемые измерения (также называемые согласованными измерениями) - это измерения, которые

разделяются между двумя или более кубами базы данных. Типичными разделяемыми измерениями являются Time, Product и Customer. Личные измерения создаются для индивидуального куба. Связанное измерение основано на измерении, которое хранится в другой базе данных Analysis Services. Исходная база данных для связанного измерения может быть на том же самом или на другом сервере;

– Mining Structures позволяет вам создавать модель извлекаемых данных с использованием Data Mining Wizard. Эти модели основываются на существующей в кубе информации.

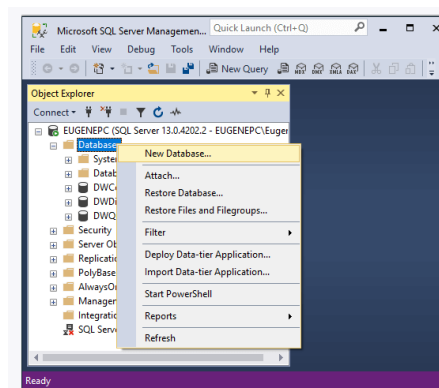
Помимо этих папок существует еще и несколько других папок (Roles, Assemblies и Miscellaneous).

## Создание базы данных в SQL Management Studio

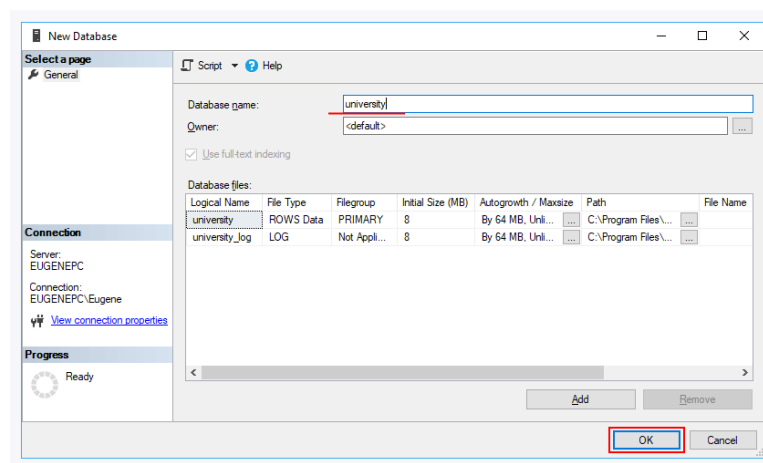
Теперь создадим свою базу данных. Для этого мы можем использовать скрипт на языке SQL, либо все сделать с помощью графических средств в SQL Management Studio. Попробуем оба способа.

## Создание базы данных с помощью графических средств

Для этого откроем SQL Server Management Studio и нажмем правой кнопкой мыши на узел Databases. Затем в появившемся контекстном меню выберем пункт New Database:



После этого нам открывается окно для создания базы данных:



В поле Database необходимо ввести название новой бд. Пусть у нас база данных называется university.

Следующее поле Owner задает владельца базы данных. По умолчанию оно имеет значение <default>, то есть владельцем будет тот, кто создает эту базу данных. Оставим это поле без изменений.

Далее идет таблица для установки общих настроек базы данных. Она содержит две строки - первая для установки настроек для главного файла, где будут храниться данные, и вторая строка для конфигурации файла логирования. В частности, мы можем установить следующие настройки:

Logical Name: логическое имя, которое присваивается файлу базы данных.

File Type: есть несколько типов файлов, но, как правило, основная работа ведется с файлами данных (ROWS Data) и файлом лога (LOG)

Filegroup: обозначает группу файлов. Группа файлов может хранить множество файлов и может использоваться для разбиения базы данных на части для размещения в разных местах.

Initial Size (MB): устанавливает начальный размер файлов при создании (фактический размер может отличаться от этого значения).

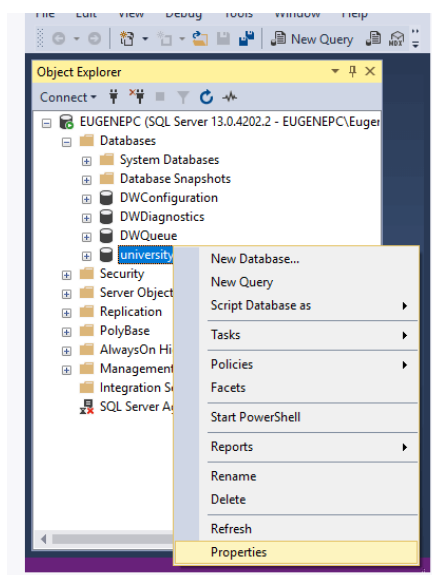
Autogrowth/Maxsize: при достижении базой данных начального размера SQL Server использует это значение для увеличения файла.

Path: каталог, где будут храниться базы данных.

File Name: непосредственное имя физического файла. Если оно не указано, то применяется логическое имя.

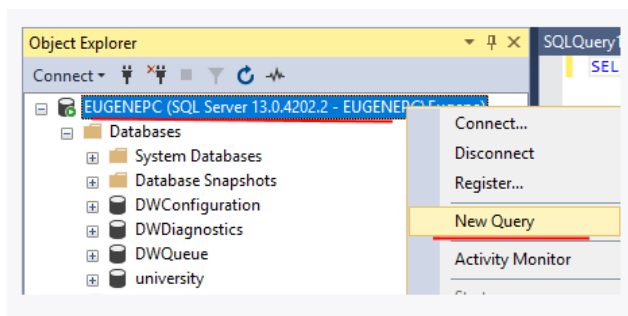
После ввода названия базы данных нажмем на кнопку ОК, и бд будет создана.

После этого она появится среди баз данных сервера. Если эта бд впоследствии не потребуется, то ее можно удалить, нажав на нее правой кнопкой мыши и выбрав в контекстном меню пункт Delete:



## Создание базы данных с помощью скрипта на языке SQL

Чтобы создать новую базу данных откроем SQL Server Management Studio. Нажмем на назначение сервера в окне Object Explorer и в появившемся меню выберем пункт New Query.



В центральное поле для ввода выражений sql введем следующий код:

```
CREATE DATABASE usersdb
```

Тем самым мы создаем базу данных, которая будет называться "usersdb":

Для выполнения команды нажмем на панели инструментов на кнопку Execute или на клавишу F5. И на сервере появится новая база данных.

Но лучше создавать базу данных вместе с журналом транзакций:

```
CREATE DATABASE usersdb
ON (Name = usersdb,
FileName = 'C:\Program Files\Microsoft SQL
Server\MSSQL13.MSSQLSERVER\MSSQL\DATA\usersdb.mdf',
Size = 1Mb,
Maxsize = 100Mb,
FileGrowth= 1Mb)
LOG ON
(Name = usersdbLog,
FileName = 'C:\Program Files\Microsoft SQL
Server\MSSQL13.MSSQLSERVER\MSSQL\DATA\usersdb.ldf',
Size = 1Mb,
Maxsize = 100Mb,
FileGrowth = 1Mb)
```

После создания базы данных, мы можем установить ее в качестве текущей с помощью команды USE:

```
USE usersdb;
```

## **Прикрепление базы данных**

Возможна ситуация, что у нас уже есть файл базы данных, который, к примеру, создан на другом компьютере. Файл базы данных представляет файл с расширением mdf, и этот файл в принципе мы можем переносить. Однако даже если мы скопируем его компьютер с установленным MS SQL Server, просто так скопированная база данных на сервере не появится. Для этого необходимо выполнить прикрепление базы данных к серверу. В этом случае применяется выражение:

```
CREATE DATABASE название_базы_данных  
ON PRIMARY(FILENAME='путь_к_файлу_mdf_на_локальном_компьютере')  
FOR ATTACH;
```

В качестве каталога для базы данных лучше использовать каталог, где хранятся остальные базы данных сервера. На Windows 10 по умолчанию это каталог C:\Program Files\Microsoft SQL Server\MSSQL13.MSSQLSERVER\MSSQL\DATA. Например, пусть файл с данными называется userstoredb.mdf. И надо этот файл добавить на сервер как базу данных. Вначале его надо скопировать в выше указанный каталог. Затем для прикрепления базы к серверу надо использовать следующую команду:

```
CREATE DATABASE contactodb  
ON PRIMARY(FILENAME='C:\Program Files\Microsoft SQL  
Server\MSSQL13.MSSQLSERVER\MSSQL\DATA\userstoredb.mdf')  
FOR ATTACH;
```

После выполнения команды на сервере появится база данных contactodb.

## **Удаление базы данных**

Для удаления базы данных применяется команда DROP DATABASE, которая имеет следующий синтаксис:

```
DROP DATABASE database_name1 [, database_name2]...
```

После команды через запятую можно перечислить все удаляемые базы данных. Стоит учитывать, что даже если удаляемая база данных была прикреплена, то все равно будут удалены все файлы базы данных.

## **Содержание отчета**

- 1 Название работы
- 2 Цель работы
- 3 Перечень технических средств обучения

- 4 Порядок выполнения работы
- 5 Ответы на контрольные вопросы
- 6 Вывод

### **Варианты заданий**

Варианты заданий представлены в практической работе № 39

### **Контрольные вопросы:**

1. Какие типы баз данных используется в SQL Server?
2. Для чего необходимы системные базы данных?
3. Что такое транзакция в SQL Server?
4. Для чего предназначен журнал транзакций?
5. Какая информация содержится в журнале транзакций?
6. Способы создания базы данных.
7. Как удалить базу данных?

### **Используемая литература**

- Г.Н.Федорова Основы проектирования баз данных. М.: Академия, 2020
- Г.Н.Федорова Разработка, администрирование и защита баз данных. М.: Академия, 2018
- <https://www.delphiplus.org/microsoft-sql-server/sozdanie-novogo-proekta.html> <https://studfile.net/preview/5917121/page:2>
- <https://metanit.com/sql/sqlserver/3.1.php>