

Лабораторная работа № 15

Управление базами данных

Цель: научить использовать SQL-команды для создания баз данных и изменения их структуры, а также выполнять резервное копирование и восстанавливать базы данных из резервных копий.

Перед выполнением лабораторной работы рекомендуется изучить **Теоретический материал:**

1. Создание и настройка базы данных

Планирование физической организации базы данных при ее создании – важнейшая часть административной работы с базами данных, файлами и файловыми группами. Плохо физически организованная будет работать с меньшей производительностью.

СУБД MS SQL Server предлагает возможных вариантов создания базы данных:

1. Использование графического интерфейса *Management Studio*.
2. Использование команд SQL.

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 расположение всех остальных файлов БД.

2. *Вторичные файлы данных*. Как правило, используется расширение NDF. Вторичным является любой файл кроме первичного и файлов журналов. БД могут не содержать ни одного вторичного файла.

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.

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*.

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:

- *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;
- *NAME* – определяет логическое имя файла. По умолчанию совпадает с физическим именем файла, определенном в параметре *FILENAME*;
- *FILENAME* – указывает полный путь и имя физического файла;
- *SIZE* – указывает размер файла: в мегабайтах, килобайтах. 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;
- *MAXSIZE* – указывает максимальный размер, до которого может увеличиваться файл. Если этот параметр не указан, то устанавливается значение *UNLIMITED*, позволяющее увеличивать файлам размер без ограничений;
- *FILEGROWTH* – задает шаг увеличения файла, причем ноль означает запрет увеличения размера. Значение указывается в мегабайтах, килобайтах или процентах. По умолчанию приращение – 10%, если не указаны единицы, то цифра воспринимается в мегабайтах;
- *FILEGROUP* – определяет имя группы файлов, в которую помещается файл.

Для просмотра информации о базе данных, файлах и группах файлов используются следующие хранимые процедуры:

- **sp_helpdb** [*база_данных*] – информация о базе данных и ее настройках. Если база данных не указана, то отображается отчет по всем базам данных, поддерживаемых данным MS SQL Server.

- **sp_helpfile** [*имя*] – информация о файлах, относящихся к текущей базе данных. Если имя файла не указано, то отображается информация обо всех файлах этой базы данных.

- **sp_helpfilegroup** [*имя*] – информация обо всех файловых группах в текущей базе данных. Если указано имя файловой группы, то выводится информация по каждому файлу указанной группы.

- **sp_spaceused** [*объект*] – сведения о дисковом пространстве, используемом указанным объектом.

Помимо перечисленных выше физических параметров база данных имеет еще и логические параметры. Только владелец и системный администратор может изменить эти параметры. Управление параметрами осуществляется с помощью системной хранимой процедуры **sp_dboption**:

```
sp_dboption [ [@dbname=] 'имя_базы' ] [, [@option=] '' ]  
[, [@value=] ON | OFF]
```

2. Изменение базы данных

Удаление базы данных осуществляется с помощью оператора:

```
DROP DATABASE имя_базы_данных [, ...n]
```

В результате удаляются все файлы, используемые базой данных. Правом на удаление обладает владелец базы и пользователи роли *sysadmin*, это право не может быть передано другим учетным записям.

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

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

Переименование базы данных:

```
sp_renamedb [@old_name=] 'старое_имя', [@new_name=] '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 будут записываться файлы, у которых в параметрах не определена принадлежность к группе; установленной по умолчанию изначально считается первичная файловая группа).

3. 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 ('имя_БД', ['процент'] [, NOTRUNCATE | TRUNCATEONLY])
```

Описание параметров:

- *имя_БД* – имя базы данных, которую необходимо сжать;
- *процент* – количество процентов свободного пространства, которое желательно оставить после сжатия;
- *NOTRUNCATE* – свободное пространство не возвращается операционной системе, а резервируется в файлах, т.е. физически уменьшения размера базы данных не происходит;
- *TRUNCATEONLY* – свободное пространство удаляется за последним используемым в файле экстендом¹, при этом данные не перемещаются, а параметр *процент* игнорируется.

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;

¹ Экстенд – непрерывная область памяти на накопителе.

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

4. Резервное копирование данных

Необходимо уделять особое внимание целостности информации, с которой работает пользователь. 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);
- *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.

5. 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.

Требования к отчету: по результатам работы представить отчёт со скриншотами, содержащими SQL-команды и результаты их выполнения для каждого задания из разделов «Подготовительные задания» и «Самостоятельная работа».

Подготовительные задания

Задание 1. Создайте базу данных *Sales* с журналом.

Указания к выполнению:

1. Воспользуемся функцией **CREATE DATABASE:**

```
CREATE DATABASE Sales  
ON  
( NAME = Sales_dat,  
  FILENAME = 'e:\saledat.mdf',  
  SIZE = 10,  
  MAXSIZE = 50,
```

```

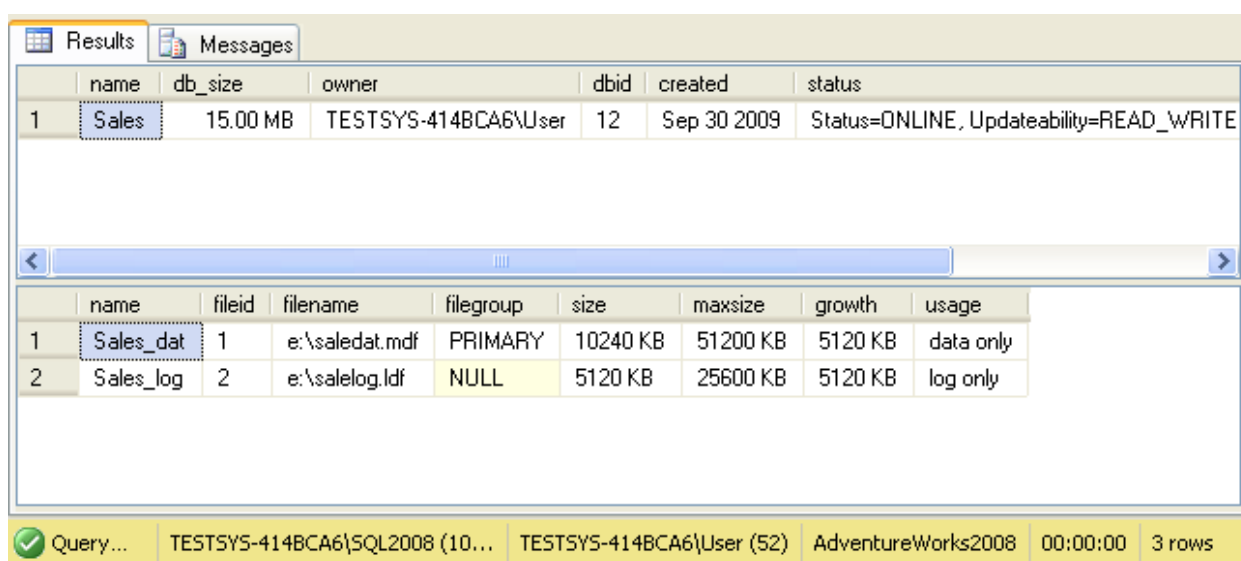
FILEGROWTH = 5 )
LOG ON
( NAME = Sales_log,
  FILENAME = 'e:\salelog.ldf',
  SIZE = 5MB,
  MAXSIZE = 25MB,
  FILEGROWTH = 5MB )

```

2. Выполните следующий код, чтобы убедиться, что БД была создана с нужными параметрами (рис. 6.1):

```
sp_helpdb Sales
```

Результат выполнения показан на рис. 6.1.



	name	db_size	owner	dbid	created	status
1	Sales	15.00 MB	TESTSYS-414BCA6\User	12	Sep 30 2009	Status=ONLINE, Updateability=READ_WRITE

	name	fileid	filename	filegroup	size	maxsize	growth	usage
1	Sales_dat	1	e:\saledat.mdf	PRIMARY	10240 KB	51200 KB	5120 KB	data only
2	Sales_log	2	e:\salelog.ldf	NULL	5120 KB	25600 KB	5120 KB	log only

Рис. 6.1. Сведения о базе данных

Задание 2. Создайте резервную копию базы данных *Sales*.

Указания к выполнению:

1. Создадим устройство для резервирования:

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

2. Воспользуемся функцией **BACKUP** для выполнения резервного копирования:

```
BACKUP DATABASE Sales TO backupdisk
```

Результат выполнения показан на рис. 6.2.

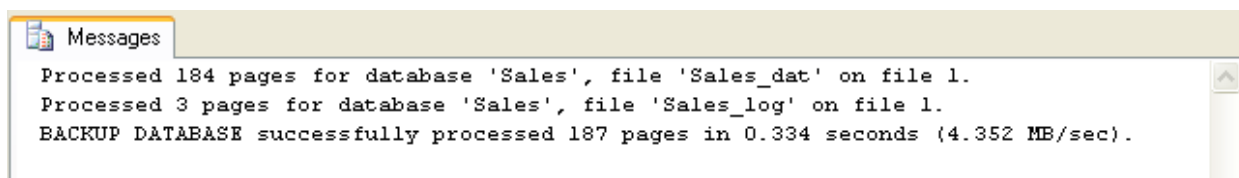


Рис. 6.2. Результат выполнения оператора BACKUP

Самостоятельная работа

Указание. Для доказательства правильности выполнения действий можно использовать процедуры **sp_helpdb** до выполнения некоторой операции и после.

Задание 1.

1. Создайте пользователя, имеющего право создания и удаления баз данных.
2. Создайте **новую собственную базу данных**, размер основного файла которой 5 Мб, размер журнала – 2 Мб. Основной файл может увеличиваться до 10 Мб с шагом 20%. Объем файла журнала увеличивается до 5 Мб с шагом 1 Мб. Расположение файлов на диске – *D:\MSSQL\DATA*.
3. Убедитесь, что база данных создана должным образом.

Задание 2.

1. Просмотрите список параметров базы данных, которые могут быть установлены.
2. Просмотрите список установленных параметров созданной Вами базы данных.
3. Определите использование базы данных только владельцем и в режиме поддержки одного пользователя.
4. Убедитесь в изменении параметров базы данных.

Задание 3.

1. Переименуйте созданную Вами базу данных.

Задание 4.

1. Создайте собственную файловую группу.
2. Для Вашей базы данных определите еще один файл размером 2 Мб и поместите его в созданную рабочую группу.
3. Докажите правильность выполненных Вами действий.

Задание 5.

1. Для Вашей созданной базы данных установите возможность автоматического сжатия данных.

Задание 6.

1. Определите сведения о дисковом пространстве, занимаемом базой данных, созданной в результате выполнения ЛР-11.

2. Сожмите базу данных так, чтобы она содержала только 25% пространства, доступного ей на текущий момент.
3. Докажите правильность выполненного действия.

Задание 7.

1. Для базы данных, созданной в *ЛР-11* добавьте еще один файл к файловой группе PRIMARY.
2. Выполните сжатие базы данных созданной в *ЛР-11*. Обоснуйте выбранные Вами параметры.
3. Докажите, что сжатие базы данных действительно выполнено.

Задание 8.

1. Создайте резервную копию базы данных созданной в *ЛР-11*. Физическое имя устройства резервного копирования: *D:\MSSQL\BACKUP\ЛР-11_backup.bak*.
2. Выполните резервное копирование журнала транзакций базы данных созданной в *ЛР-11*.

Задание 9.

1. Удалите базу данных, созданную в *ЛР-11*.
2. Восстановите удаленную базу с устройства с физическим именем *ЛР-11_backup.bak*.
3. Докажите правильность выполненного действия.