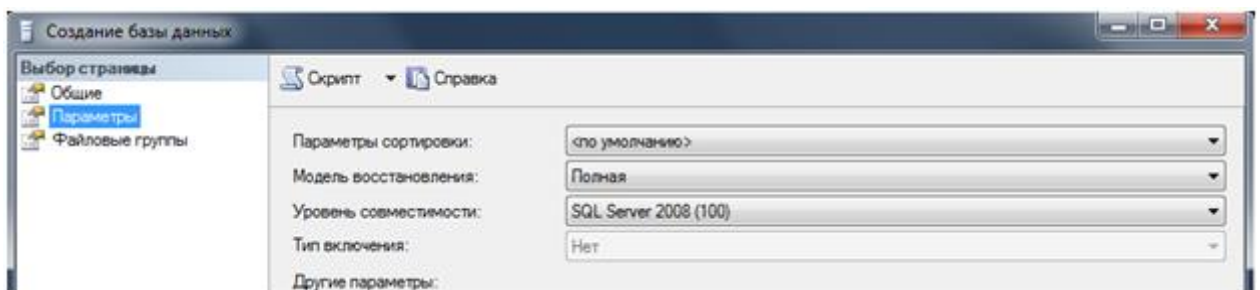
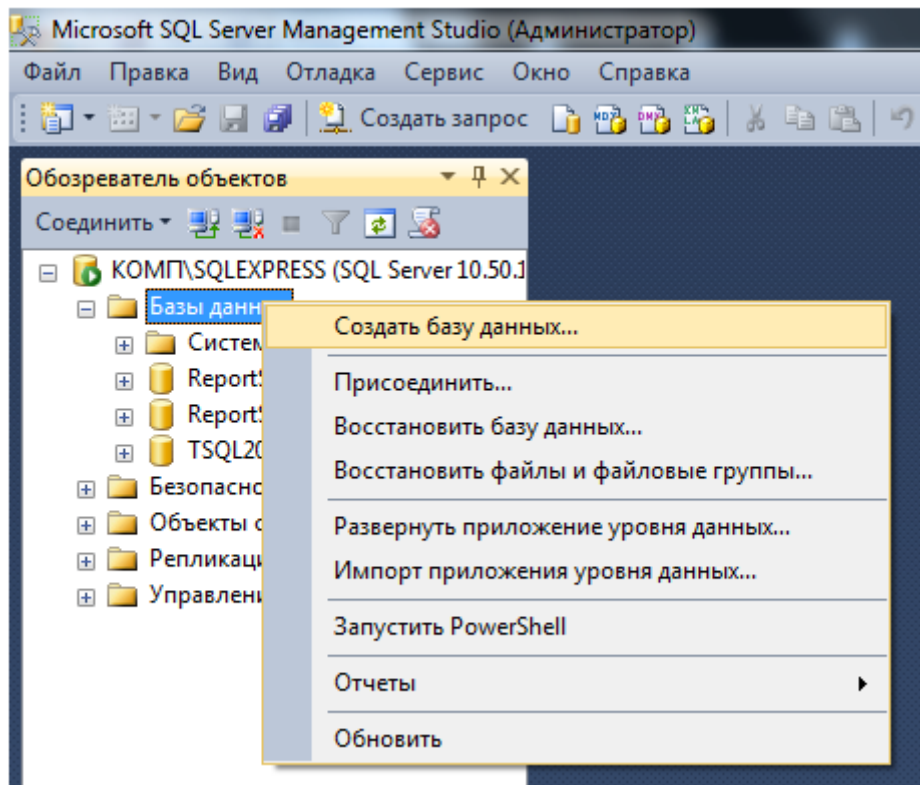


## Создание резервных копий и восстановление в MS SQL Server

Создадим тестовую базу данных “**sbase**”, модель восстановления - полная:



### Создание резервных копий

#### 1 Резервное копирование системных баз данных.

Список системных баз: **master, model, msdb, tempdb**.

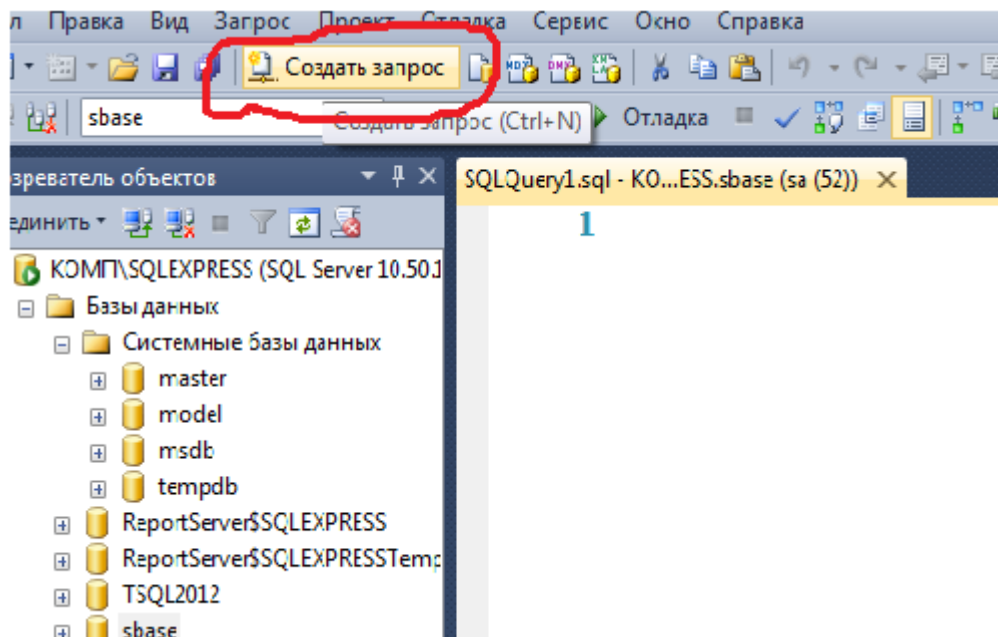
**Master:** содержит сведения обо всех базах данных на сервере. Резервное копирование необходимо делать каждый раз, когда создаются, удаляются или изменяются пользовательские базы данных.

**Model:** Используется в качестве шаблона для создаваемых баз данных. Резервное копирование необходимо при изменении настройки самой базы **model**.

**Msdb:** Содержит сведения о заданиях и для агента сервера **MS SQL Server**. копирование необходимо делать каждый раз при добавлении задания для агента сервера **MS SQL Server**.

**Tempdb:** Хранит временные данные например для транзакций. Уничтожается и создается при перезапуске экземпляра **MS SQL Server**. Резервное копирование делать нет смысла.

Создадим новый запрос:



Выполним следующий запрос:

BACKUP DATABASE master

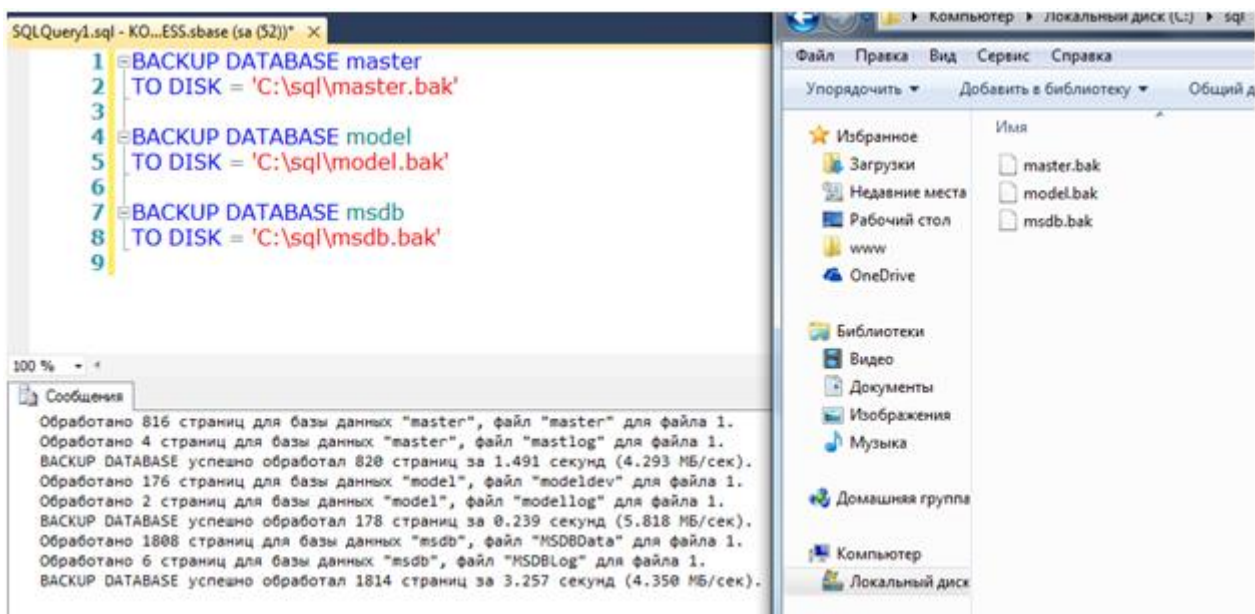
TO DISK = 'C:\sql\master.bak'

BACKUP DATABASE model

TO DISK = 'C:\sql\model.bak'

BACKUP DATABASE msdb

TO DISK = 'C:\sql\msdb.bak'



Как видим, на диск 'C' было произведено успешное резервное копирование системных баз данных.

## 2 Полное резервное копирование базы данных.

Включает в себя файлы данных и журнал транзакций. По сути является базой данных на момент создания резервной копии базы данных MS SQL Server.

Включает в себя:

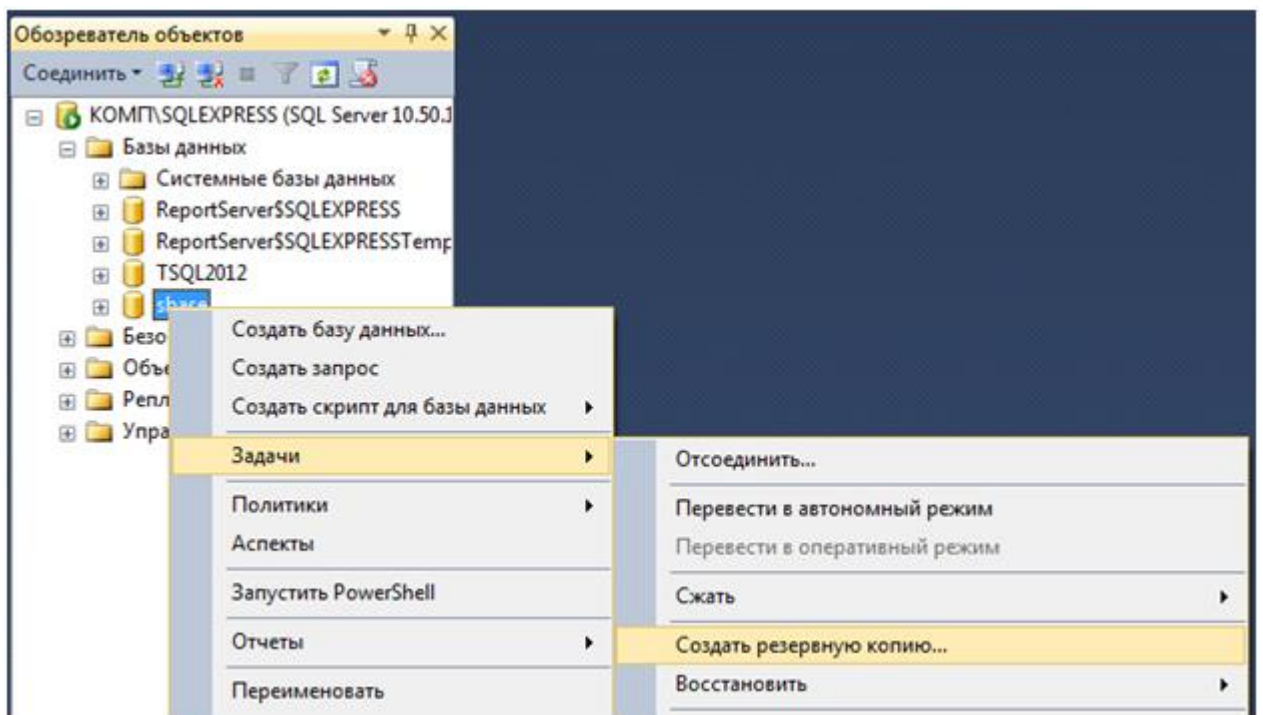
Резервное копирование данных в базе.

Резервное копирование изменений, возникающих во время резервного копирования

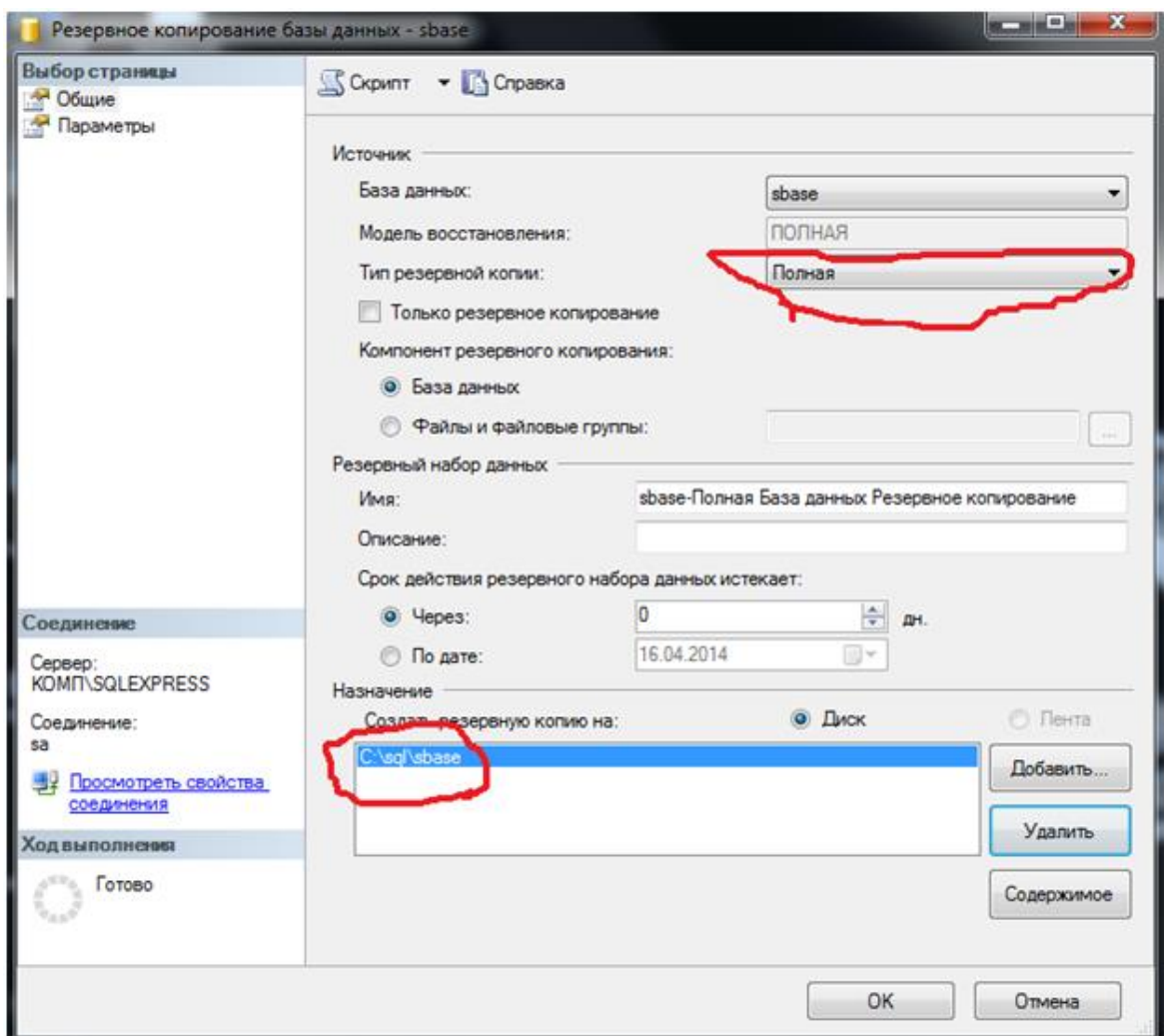
Резервное копирование транзакций, не зафиксированных в журнале транзакций.

### Способ 1(Графический интерфейс):

Выберем «создать резервную копию»



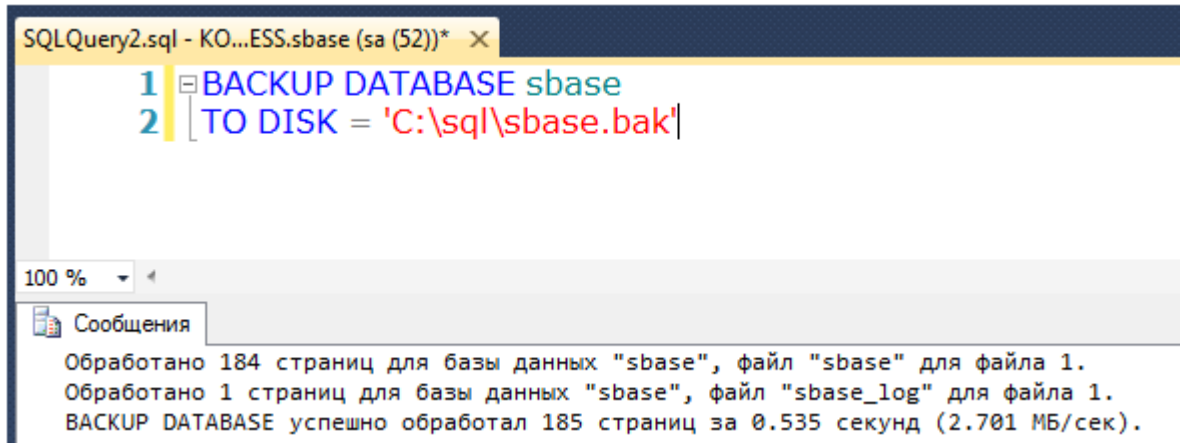
Указываем куда копировать и модель – полная.



## Способ 2(Запрос SQL):

BACKUP DATABASE sbase

TO DISK = 'C:\sql\sbase.bak'



## 3 Разностное резервное копирование базы данных.

Включает в себя все изменения базы данных с момента последнего полного резервного копирования.

Нельзя восстановить без полной резервной копии. После каждого запуска разностного копирования, размер резервной копии возрастает из-за количества транзакций с момента полного резервного копирования.

При создании разностного резервного копирования выполняются следующие действия:

Создание резервных копий баз данных, которые изменились с момента полного резервного копирования.

Создание резервных копий всех операций, выполняющихся во время разностного резервного копирования и всех транзакций не зафиксированных в журнале транзакций.

Добавим данные:

use sbase;

--Создадим таблицу test

CREATE TABLE test(

id INT,

name VARCHAR(MAX)

);

--Добавим данные

```
INSERT INTO test (id,name)
```

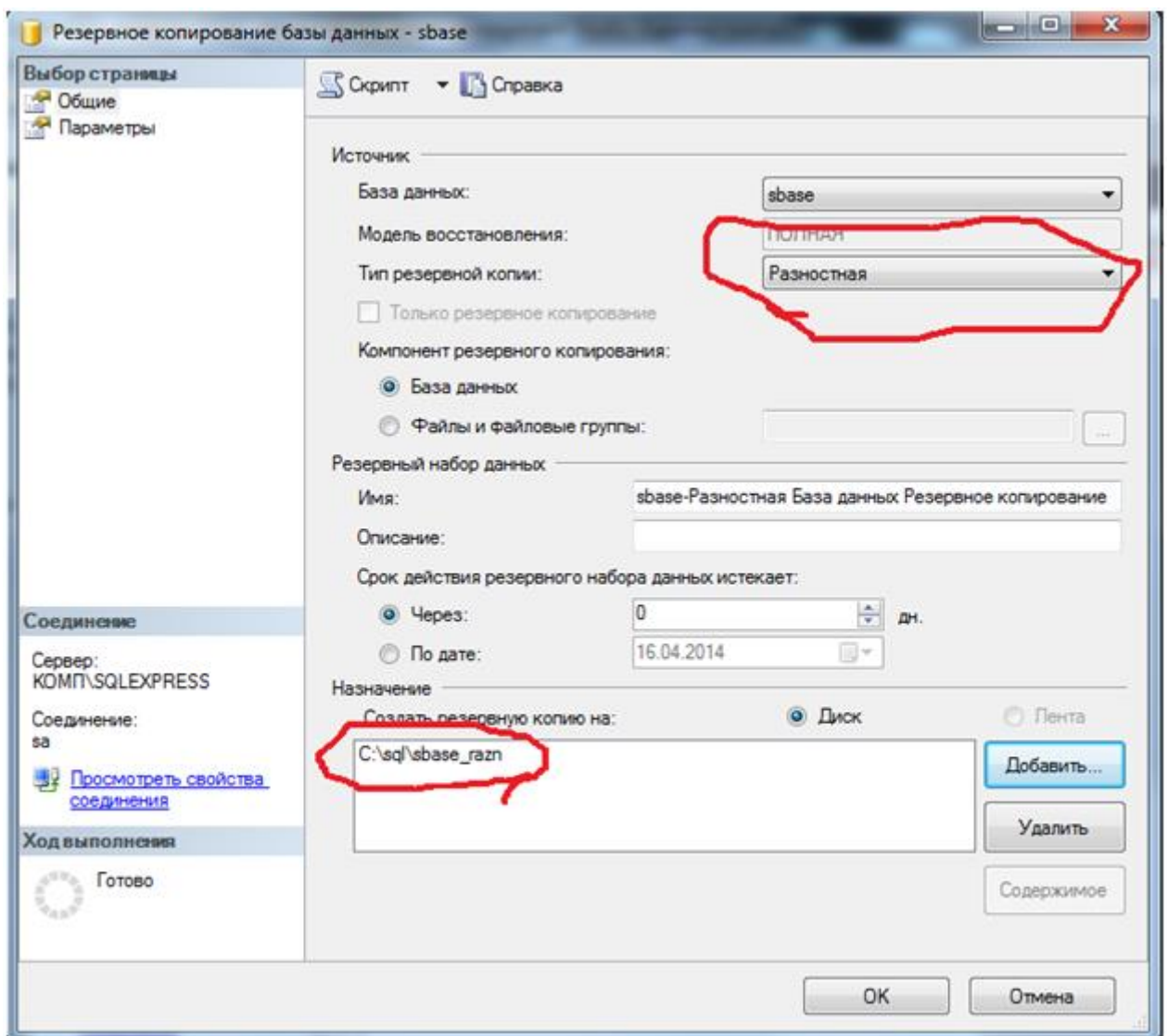
```
VALUES
```

```
(1, 'Миша'),
```




```
(2, 'Саша'),
```

```
(3, 'Маша');
```

Далее по аналогии с полным запусим задачу резервного копирования, но модель выберем – разностную:



Вот результат:

|  |                  |            |          |
|--|------------------|------------|----------|
|  sbase      | 16.04.2014 21:59 | Файл       | 1 557 КБ |
|  sbase.bak  | 16.04.2014 22:00 | Файл "БАК" | 1 557 КБ |
|  sbase_razn | 16.04.2014 22:16 | Файл       | 917 КБ   |

Проведем полный бэкап, добавим еще данных, проведем разностный бэкап:

--Делаем полный бэкап

```
BACKUP DATABASE sbase
```

```
TO DISK = 'C:\sql\sbase_razh2'
```

--Добавим еще данные

```
INSERT INTO test (id,name)
```

```
VALUES
```

```
(4, 'Миша'),
```

```
(5, 'Саша'),
```

```
(6, 'Миша'),
```

```
(7, 'Саша'),
```

```
(8, 'Миша'),
```

```
(9, 'Саша'),
```

```
(10, 'Маша');
```




```
BACKUP DATABASE sbase
```

```
TO DISK = 'C:\sql\sbase_razh3'
```

```
WITH DIFFERENTIAL;
```

А вот и результат:



|   |                  |      |               |
|---|------------------|------|---------------|
|  sbase_razh2 | 16.04.2014 22:24 | Файл | 1 557 КБ      |
|  sbase_razh3 | 16.04.2014 22:24 | Файл | <u>533 КБ</u> |
|  sbase_razn  | 16.04.2014 22:16 | Файл | <u>917 КБ</u> |

Вывод, не надо доводить разностное копирование до больших объемов, иначе оно теряет свой смысл быстрого восстановления данных.

#### **4 Резервное копирование журнала транзакций базы данных.**

Содержат все изменения базы данных при первичном резервном копировании лога транзакций, или изменения с последней успешной резервной копии журнала транзакций.

Не имеет смысла, если хоть раз не выполнялось полное резервное копирование, т.к. резервную копию лога невозможно будет восстановить при отсутствии полной резервной копии.

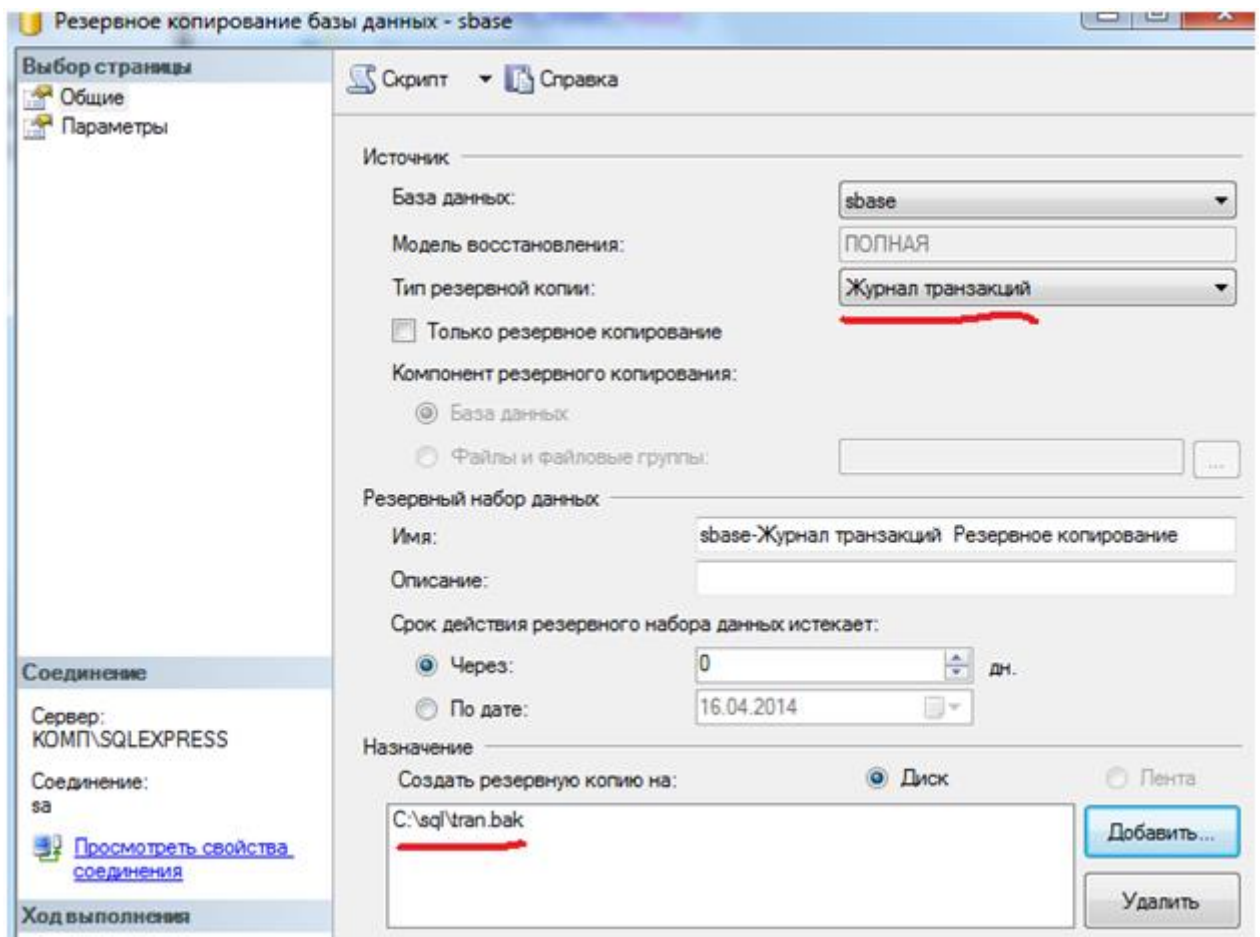
В процессе выполняются следующие действия:

Создается копия журнала транзакций от последнего резервного копирования лога до конца текущего.

Очищаются части журнала транзакций до начала активной части и отбрасываются сведения в неактивной части.

По аналогии с полным и разностным копированием запускаем задачу, но тип выбираем – лог транзакций:





Или с помощью запроса:

BACKUP LOG sbase

TO DISK = 'C:\sql\sbase\_tran.bak'

## 5 Резервное копирование файловых групп базы данных.

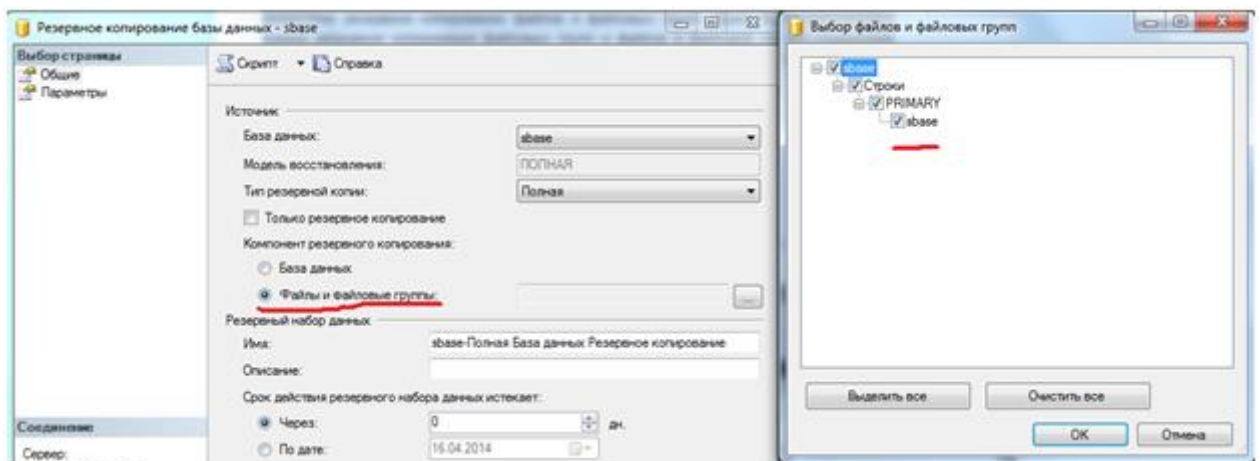
По сути являются именованными коллекциями файлов и используются для упрощения размещения данных и выполнения задач администрирования.

Файлы журналов не входят в состав файловых групп.

Управление пространством журнала отделено от управления пространством данных, возможно только полное и разностное резервное копирование файлов и файловых групп.

Пример полного копирования:

По аналогии с другими видами копирования запускаем мастер:



Тоже, только запросом:

BACKUP DATABASE sbase

FILEGROUP = 'PRIMARY'

TO DISK = 'C:\sql\primary.bak'

### **Стадии восстановления:**

Стадия копирования данных: копирование всех страниц данных, журнала и индекса с резервного носителя в файлы базы данных.

Стадия повтора: журнальные транзакции применяются к данным, скопированным из резервной копии, чтобы произвести их накат до точки восстановления. В этой точке базы данных обычно имеются незафиксированные транзакции, и потому база находится в непригодном для работы состоянии. В этом случае следует производить в процесс восстановления базы данных стадию отмена.

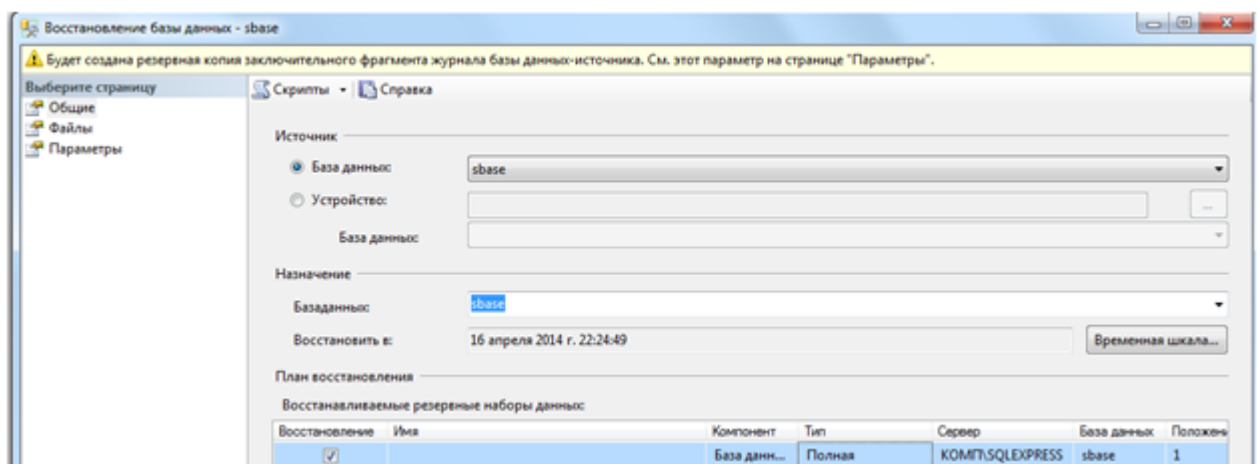
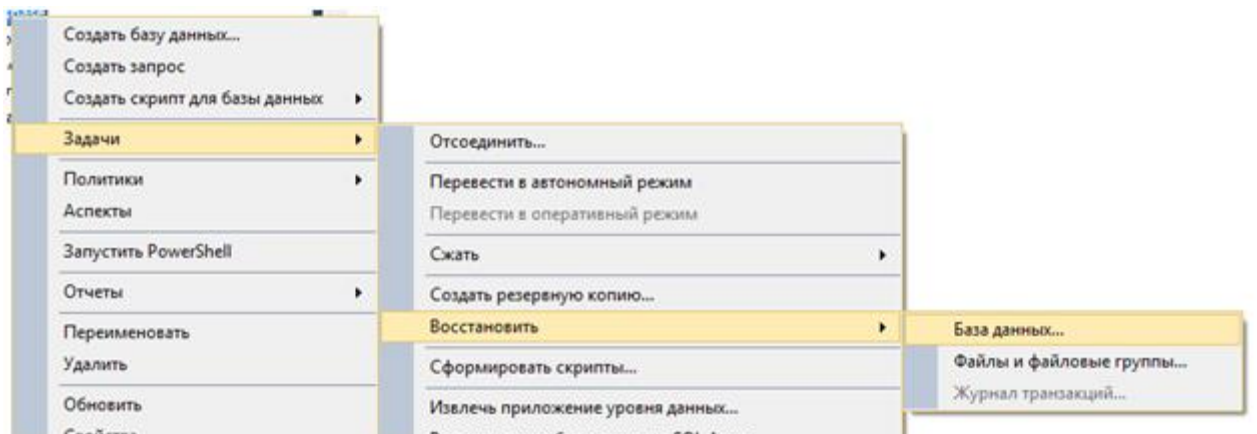
Стадия отката: производит откат незафиксированных транзакций и делает базу данных доступной для пользователей. После завершения стадии отката восстановление последующих резервных копий становится невозможным. Затем в процессе восстановления база данных переводится в активный режим.

Режим WITH RECOVERY включает и стадию повтора, и стадию отката и восстанавливает базу данных. Более поздние резервные копии восстановить невозможно. Это значение по умолчанию. Если набор данных наката не был восстановлен в достаточной степени, чтобы обеспечить согласованность с базой данных, стадия отката выполнена быть не может. Компонент Database Engine выдает ошибку и прекращает восстановление. Если весь набор данных наката согласован с базой данных, то выполняется восстановление, после чего базу данных можно перевести в режим в сети.

Предложение WITH NORECOVERY позволяет пропустить стадию отката, чтобы сохранить незафиксированные транзакции. Пропуск стадии отката позволяет восстановить другие резервные копии, чтобы выполнить накат базы данных на более поздний момент времени. Иногда инструкция RESTORE WITH NORECOVERY выполняет накат данных до момента, пока они не будут согласованы с базой данных. В таких случаях компонент Database Engine выдает информационное сообщение, указывающее, что набор данных наката теперь можно восстановить при помощи параметра RECOVERY. Другими словами, параметр NORECOVERY нужно использовать, когда для восстановления базы используются несколько восстанавливаемых резервных копий, за исключением последней восстанавливаемой резервной копии. После применения параметра NORECOVERY, база данных переходит в состояние восстановления.

## 6 Восстановление из полной резервной копии.

Графический интерфейс:

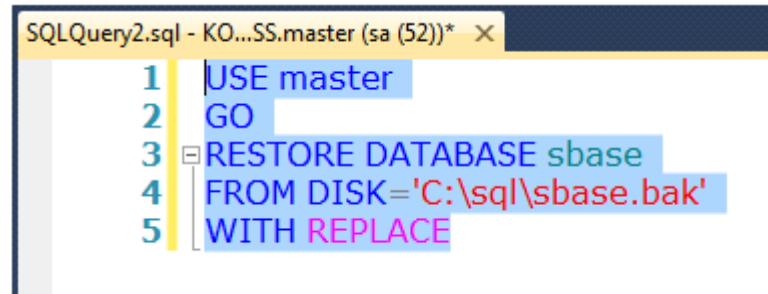


Или с помощью запроса:

USE master

GO

```
RESTORE DATABASE sbase  
  
FROM DISK='C:\sql\sbase.bak'  
  
WITH REPLACE
```



### **7 Восстановление из разностной резервной копии.**

В начале восстанавливается полная копия(например в прошлом шаге мы это уже сделали), а далее восстановим разностную копию.

Графический интерфейс аналогичен с предыдущим примером за исключением типа выбираемой копии, а запрос будет таков на примере наших разностных копий:

```
USE master
```

```
RESTORE DATABASE sbase
```

```
FROM DISK = 'C:\sql\sbase_razh2'
```

```
WITH FILE = 1, NORECOVERY, REPLACE
```

```
USE master
```

```
RESTORE DATABASE sbase
```

```
FROM DISK = 'C:\sql\sbase_razh3'
```

```
WITH FILE = 1, RECOVERY
```

```
GO
```

### **8 Восстановление журнала транзакций.**

В начале следует восстановить базу данных из полной резервной копии, затем накатить на базу последовательно резервные копии журнала транзакций.

Графический вариант интуитивно понятен, будет продемонстрирован только SQL запрос:

Для того, чтоб все отработало корректно, вернемся разностному бэкапу 2, и после него накатим журнал транзакций:

```
USE master
```

```
RESTORE DATABASE sbase
```

```
FROM DISK = 'C:\sql\sbase'
```

```
WITH FILE = 1, NORECOVERY, REPLACE
```

```
RESTORE DATABASE sbase
```

```
FROM DISK = 'C:\sql\sbase_razh2'
```

```
WITH FILE = 1, NORECOVERY, REPLACE
```

```
RESTORE LOG sbase
```

```
FROM DISK = 'C:\sql\tran.bak'
```

```
WITH FILE = 1, NORECOVERY
```

## **9 Восстановление файловых групп.**

Графический вариант показан не будет, он довольно интуитивен, запрос SQL:

```
RESTORE DATABASE sbase FILEGROUP = 'PRIMARY'
```

```
FROM DISK = 'C:\sql\primary.bak'
```

```
WITH PARTIAL, RECOVERY, REPLACE
```

Так как мы восстанавливали только часть базы – файловую группу, то мы использовали параметр «PARTIAL».

## **10 Восстановление системных баз данных.**

Если экземпляр SQL сервера доступен, то системные базы восстанавливаются согласно приведенной таблице:

| Системная база данных | Способ восстановления   |
|-----------------------|---|
| master                | Запускаем экземпляр сервера в однопользовательском режиме. Восстановление базы осуществляется так же, как полное восстановление пользовательской базы данных. После восстановления следует перезапустить экземпляр SQL сервера. |
| msdb                  | Восстановление базы осуществляется так же, как полное восстановление пользовательской базы данных.  |
| model                 | Восстановление базы осуществляется так же, как полное восстановление пользовательской базы данных.  |

Запускаем экземпляр сервера в однопользовательском режиме: выключим и включим экземпляр сервера с параметром запуска /m, введя в командной строке Windows (CMD):

**net stop MSSQLSERVER**

**net start MSSQLSERVER /m**

Подключимся к серверу и запустим процесс восстановления базы.

**sqlcmd**

**RESTORE DATABASE master FROM DISK = 'C:\sql\sbase.bak' WITH REPLACE;**

**GO**

Вернем экземпляр SQL в состояние «в сети».

Стартуем сервер в многопользовательском режиме:

**net start MSSQLSERVER**