

Задание

1. Изучить материал лекции, законспектировать.
2. Фотоотчет прислать на электронную почту

С уважением, Хвастова Светлана Ивановна

!!! Если возникнут вопросы обращаться по телефону 0721389311.

Электронная почта: xvsviv@rambler.ru

Лекция на тему:

«Механизм резервного копирования. Основы восстановления баз данных.»

План

1. Основы резервного копирования SQL SERVER 2005. Типы резервного копирования.
2. Типы резервного копирования SQL Server 2005
3. Основы восстановления баз данных. Проведение восстановления
4. Восстановление баз данных SQL Server 2005
5. Режимы восстановления баз данных. Оперативный режим восстановления. Восстановление отдельных страниц. Восстановление системных баз данных.
6. Восстановление базы данных в оперативном режиме
7. Восстановление отдельных страниц базы данных
8. Автоматическая доставка журналов

1. Основы резервного копирования SQL SERVER 2005.

Резервная копия – это копия данных, используемая для восстановления данных. Резервные копии позволяют восстанавливать данные после сбоя. При правильном создании резервных копий можно будет восстановить базу после многих сбоев, таких как: сбой носителя; ошибки пользователей; аппаратные сбои; стихийные бедствия.

Кроме того, резервные копии баз данных полезны и для выполнения повседневных административных задач, например для копирования базы данных с одного сервера на другой, настройки зеркального отображения баз данных и архивирования.

Первый вопрос, который нужно решить при реализации системы резервного копирования, – куда помещать резервные копии. В распоряжении администраторов на предприятиях есть три варианта:

- самый лучший с технической точки зрения, но самый дорогой вариант – это использование ленточных библиотек. Эти устройства отличаются высокой скоростью работы и надежностью. Часто с ними поставляется специальное программное обеспечение для проведения резервного копирования.

- проведение резервного копирования на стриммер (устройство записи с магнитной лентой), подключенный локально к компьютеру, на котором работает SQL Server 2005;

- проведение резервного копирования на жесткий диск или RAID-массив, подключенный либо локально к тому компьютеру, на котором работает SQLServer, либо к другому компьютеру. Во втором случае резервное копирование может производиться по сети.

После того, как вы определили, куда будете помещать резервные копии, следующее решение, которое вам нужно принять, – будет ли явно указываться путь для размещения резервной копии или для этой цели будут создаваться вспомогательные объекты, которые называются устройствами резервного копирования.

Устройства резервного копирования (backup devices) – это специальные объекты, которые хранятся в базе данных master. Их единственное назначение – хранить информацию о пути к физическому файлу в операционной системе или о стриммере. Создать такое устройство можно:

- на графическом интерфейсе – из контейнера Server Objects | Backup Devices (Объекты сервера | Устройства резервного копирования) в Management Studio;

- из кода Transact-SQL – при помощи хранимой процедуры sp_addumpdevice, например:

```
USE [master];  
GO  
EXEC sp_addumpdevice @devtype = 'disk', @logicalname =  
'BackupDevice1', @physicalname = 'D:\SQLBackups\BackupFile1.bak';
```

После создания логическое устройство можно использовать для резервного копирования. Например, команда на выполнение резервного копирования базы данных db1 без использования логического устройства может выглядеть так:

```
BACKUP DATABASE db1 TO DISK = 'D:\SQLBackups\BackupFile1.bak';
```

Если же вы создали логическое устройство резервного копирования, то можно использовать такую команду:

```
BACKUP DATABASE db1 TO BackupDevice1;
```

2. Типы резервного копирования SQL Server 2005

Первый тип – полное резервное копирование (full backup или base backup). В резервную копию записываются все данные, которые есть в базе данных. Пустые страницы при этом не копируются.

Полное резервное копирование, как и все другие типы резервного копирования, производится в оперативном режиме (-nline), без отключения пользователей. Стандартными средствами SQL Server 2005 нельзя произвести резервное копирование тех баз данных и файлов, которые находятся в автономном режиме (-offline). Их резервное копирование следует производить средствами операционной системы.

Второй тип резервного копирования – разностный (differential backup). В этом случае на резервную копию записываются все изменения, которые были произведены с момента полного резервного копирования. Разностное резервное копирование можно использовать только в дополнение к полному.

Третий тип – резервное копирование журналов транзакций (transaction log backup). Если вы используете режим восстановления Full или Bulk-logged, то выполнение такого резервного копирования практически обязательно. Причина проста: если вы не будете производить резервное копирование журналов транзакций, то не будет производиться и их очистка. В результате место в файлах журналов транзакций может закончиться (а если для них установлен неограниченный размер, то закончится и место на диске).

Четвертый тип резервного копирования - копирующее резервное копирование (copy-only backups). Оно предназначено, в первую очередь, для переноса данных между компьютерами в виде резервных копий. Такой тип резервного копирования разделяется на полное (в резервную копию будут помещены те же данные, что и при обычном полном резервном копировании) и разностное (аналог обычного разностного копирования). Этот тип резервного копирования отличается только тем, что в столбце is_copy_only таблицы backupset базы данных msdb (в эту таблицу помещаются данные о всех созданных резервных копиях) такие резервные копии помечаются специальным флагом. За счет этого флага резервные копии, созданные в копирующем режиме, не учитываются в последовательности обычных резервных копий.

3. Основы восстановления баз данных.

Общий план восстановления выглядит следующим образом:

1. Вначале производится процедура restore – необходимая информация восстанавливается с носителя. Вы можете восстановить только полную

резервную копию, а уже после этого произвести восстановление разностной резервной копии и резервных копий журналов транзакций.

2. Если производится также восстановление журналов транзакций, то следующим действием SQL Server записывает в базу данных всю информацию о завершенных транзакциях из журнала транзакций. Эта операция называется r-lf-rward (завершение).

3. Открытие доступа на этом этапе – fast rec-very (быстрое восстановление). Если же пользователь попытается обратиться к данным, измененным незавершенными транзакциями, то доступ ему будет закрыт за счет механизма блокировок.

4. Затем SQL Server обнаруживает в журнале все незавершенные транзакции и отменяет их. Эта операция называется r-lback – откат транзакций, а сам этап называется этапом отката (r-lback phase).

5. После этого к базе данных открывается доступ в обычном режиме во всех версиях SQL Server.

4. Восстановление баз данных SQL Server 2005

Запустить восстановление можно при помощи графического интерфейса ManagementStudi- (контекстное меню Rest-re Database для контейнера Databases или контекстное меню Tasks | Rest-re для контейнера базы данных) или при помощи команды REST-RE. Возможности, которые представляет графический интерфейс:

- Destinati-n t- rest-re ... T- database (Назначение восстановления ... в базу данных) – это имя восстанавливаемой базы данных. Вместо выбора базы данных из списка вы можете ввести свое имя. В этом случае из резервной копии на сервере будет создана новая база данных. В некоторых случаях восстановить копию существующей базы данных под другим именем, а затем при необходимости старую базу данных удалить, а восстановленную переименовать, присвоив ей старое название.

Команда на восстановление базы: REST-RE DATABASE db2 FR-M
DISK = 'D:\SQLBackups\BackupFile1.bak';

- T- a p-int -f time (На момент времени) – позволяет задать восстановление на определенный момент времени. Обычно используется только в ситуации, когда пользователь совершил ошибку, и вы знаете примерно, когда это произошло. Используется только при восстановлении журналов транзакций.

- Fr-m database (Из базы данных) – для обнаружения резервных копий будет использоваться история резервного копирования из таблиц базы данных msdb. В списке можно выбрать не только текущую базу данных, но и другие базы данных, которые есть на этом сервере;

- Fr-m device (Из устройства) –потребуется указать местонахождение резервной копии. Эта возможность используется в тех ситуациях, когда вам нужно восстановить базу данных на другой сервер или местонахождение резервной копии изменилось. В любом потребуется выбрать логическое устройство резервного копирования, картридж стриммера или файл на диске.

- Select the backup sets t- rest-re (Выбрать резервную копию для восстановления) – в этом списке вам потребуется установить флажки напротив тех резервных копий, которые вы планируете восстановить. Обратите внимание, что флажки можно поставить напротив нескольких резервных копий. В этом случае для каждой выбранной резервной копии будет выполнена отдельная команда REST-RE.

Дополнительные и очень важные параметры восстановления представлены на вкладке -pti-ns окна восстановления базы данных Management Studi-:

- -verwrite the existing database (Перезаписывать существующую базу данных) – установленный флажок позволяет перезаписать существующую базу данных. Фактически он отменяет проверки, которые призваны не допустить потери данных в случае ошибочного восстановления.

- Preserve the replicati-n settings (Сохранить настройки репликации) – сохранить настройки репликации при восстановлении. Соответствует параметру KEEP_REPLICATI-N команды REST-RE. Обычно используется только тогда, когда база данных одновременно участвует и в репликации, и в автоматической доставке журналов (l-g shipping).

- Pr-mpt bef-re rest-ring each backup (Выводить приглашение перед каждым восстановлением) – выводить приглашение перед восстановлением каждой следующей резервной копии из выбранного вами списка. Обычно этот параметр используется только тогда, когда каждая копия лежит на своем картридже стриммера, и вам нужно их менять.

- Restrict access t- the rest-red database (Ограничить доступ к восстанавливаемой базе данных) – после восстановления доступ будет открыт только членам роли базы данных db_-wner и членам серверных ролей dbcreat-r и sysadmin. Этот параметр обычно применяется в тех случаях, когда после восстановления базы данных вам необходимо произвести дополнительные проверки или внести исправления. Ему соответствует параметр команды REST-RE WITH RESTRICTED_USER;

- Rest-re the database files as (Восстановить файлы базы данных как) – очень важный параметр, который позволяет определить новый путь для восстанавливаемых файлов баз данных. Этому флажку в команде REST-RE соответствует параметр M-VE.

Rec-very state (Состояние восстановления) – еще один важнейший параметр, который определяет, будет ли база данных открыта для пользователей по.

5. Режимы восстановления баз данных. Оперативный режим восстановления. Восстановление отдельных страниц. Восстановление системных баз данных.

Одно из важных решений, которые нужно принять при создании базы данных – в каком режиме восстановления будет работать база. Этот параметр выбирается на вкладке `-rti-ns` свойств базы данных в строке `Rec-very M-del` (Режим восстановления) (над списком остальных параметров). Изменить режим восстановления базы данных можно также при помощи команды `ALTER DATABASE`.

Всего предусмотрено три режима восстановления базы данных.

- Full (режим полного протоколирования) – в этом режиме максимальное количество операций записывается в журнал транзакций. Журнал транзакций автоматически не обрезается. Этот режим обеспечивает максимальные возможности восстановления (за счет снижения производительности). Только в этом режиме вы можете использовать зеркальное отображение баз данных и автоматическую доставку журналов (`l-g shipping`). Именно этот режим выбирается по умолчанию для пользовательских баз данных, поскольку он настроен для базы данных `m-del`. Если изменить режим восстановления для базы данных `m-del`, то для создаваемых баз данных по умолчанию будет выбираться новый режим.

- Bulk-l-gged (режим неполного протоколирования) – это компромисс между требованиями производительности и возможностями восстановления. При использовании этого режима запись в журнал практически отключается (в терминологии `Micr-s-ft` – проводится минимальное протоколирование) для операций следующих типов:

- массовой вставки (команды `BULK INSERT`, `SELECT INTO`, загрузка средствами `bcpr` и т. п.);
- вставка/изменение больших двоичных данных (`text`, `ntext`, `image`);
- операции по созданию, перестроению и удалению индексов.

Автоматическая перезапись журналов транзакций при этом не производится, работа с транзакциями, не включающими в себя перечисленные операции, производится как обычно.

При работе в этом режиме вы лишаетесь возможности использовать журнал транзакций для восстановления (при утрате файлов данных, на момент времени или на метку транзакции), если в нем была хотя бы одна запись о перечисленных ранее операциях. Micr-s-ft рекомендует не использовать этот режим восстановления на постоянной основе, а переключаться в него из режима Full на время выполнения больших операций массовой вставки, а потом возвращаться обратно.

- Simple (простая модель восстановления) – максимальный выигрыш в производительности и удобстве работы за счет возможностей восстановления. Минимально протоколируются те же операции, что и в режиме восстановления Bulk-l-gged, а кроме этого, журнал транзакций автоматически очищается (блоками, размер которых изначально равен 256 Кбайт, но при необходимости он может быть автоматически увеличен). В результате вы получаете максимальную производительность и возможность не думать о потенциальной нехватке места в журнале транзакций. Но в этом режиме использовать журнал транзакций для восстановления уже удастся. Вы не сможем даже выполнить резервное копирование журнала транзакций: команда BACKUP L-G в этом режиме сразу вернет ошибку.

Какой же режим восстановления выбрать? Micr-s-ft (в своих учебных курсах) рекомендует для рабочих баз данных выбирать только режим Full. Однако из опыта проведения автором этих самых учебных курсов и общения со слушателями можно сказать, что очень многие опытные администраторы сознательно настраивают для своих баз данных режим восстановления Simple. Значительное повышение производительности при операциях массовой вставки и при работе с большими двоичными данными вполне оправдывает некоторое снижение возможностей резервного копирования и восстановления. Что важнее для вашей задачи – дополнительные возможности восстановления или максимальная производительность, решать вам.

6. Восстановление базы данных в оперативном режиме

Во всех предыдущих версиях SQL Server можно было выполнять восстановление базы данных, только отключив от нее всех пользователей. В SQLServer 2005 появилась новая возможность – восстановление на работающей базе данных. Другое название такого типа восстановления – оперативное восстановление (-nline rest-re).

Конечно, на практике обойтись совсем без ограничения доступа пользователей не удастся. При восстановлении на работающей базе данных вам в любом случае придется перевести в автономный режим (-ffline) тот файл или файловую группу, восстановление которого вы производите в данный момент. Остальные файлы или файловые группы могут оставаться в рабочем режиме.

Для восстановления на открытой базе данных предусмотрены и другие ограничения:

- резервное копирование на работающей базе данных может использоваться только для баз данных, которые работают в режиме восстановления Full или Bulk-l-gged;

- оперативное восстановление первого файла базы данных или первичной файловой группы (в которых находятся системные таблицы и карта размещения данных) производить нельзя.

Если это возможно, SQL Server автоматически применяет режим оперативного восстановления при восстановлении отдельных файлов, файловых групп и страничном восстановлении (но не при обычном восстановлении всей базы данных). Если вы хотите запретить применение оперативного восстановления и производить восстановление файлов, файловых групп и отдельных страниц в обычном автономном режиме, то можно перед восстановлением выполнить команду BACKUP L-G WITH N-REC-VERY. Эта команда, которая обычно применяется только при использовании автоматической доставки журналов (l-g shipping), позволяет создать резервную копию журнала транзакций и перевести базу данных в

специальное состояние REST-RING. В этом состоянии доступ к базе данных пользователей будет закрыт, а восстановление будет производиться только в автономном режиме.

Синтаксис команд для выполнения оперативного восстановления ничем не отличается от обычного. Например, чтобы в оперативном режиме восстановить файл db1file2, уже переведенный в автономный режим, можно использовать следующие команды:

```
REST-RE DATABASE db1 FILE = 'db1file2' FR-M DISK =  
'D:\SQLBackups\BackupFile1.bak' WITH N-REC-VERY;
```

```
REST-RE L-G db1 FR-M DISK = 'D:\SQLBackups\BackupL-gFile1.bak';
```

7. Восстановление отдельных страниц базы данных

Еще одна новая возможность SQL Server 2005, связанная с восстановлением, – восстановление отдельных страниц данных (page rest-re). Теперь в некоторых ситуациях можно вместо восстановления всей базы данных или каких-то файлов, ограничиться восстановлением лишь отдельных страниц. Это позволит:

- сэкономить время;
- произвести восстановление в оперативном режиме, без отключения пользователей от базы данных. Недоступными для пользователей будут только восстанавливаемые страницы.

Чаще всего ошибки на страницах баз данных возникают из-за сбоев дисков или дисковых контроллеров. Поэтому перед таким восстановлением лучше убедиться, что с дисковой подсистемой сервера у вас все в порядке.

Восстановление отдельных страниц базы данных можно производить только при соблюдении следующих условий:

- вы используете редакцию Enterprise Editi-n;
- восстанавливаемые страницы не относятся к журналу транзакций, к служебным страницам базы данных и к полнотекстовым каталогам;

- база данных работает в режиме Full или Bulk-logged;
- файловые группы, к которым относятся восстанавливаемые страницы, доступны и на чтение, и на запись.

Как выглядит процедура восстановления отдельных страниц базы данных? Порядок действий обычно такой:

1. Вначале вы обнаруживаете, что некоторые страницы в базе данных повреждены. Такую информацию можно получить при просмотре журналов событий SQL Server, при помощи команд DBCC (например, DBCC CHECKDB) и просто при помощи анализа сообщений, которые возвращаются клиентскому приложению. Сам SQL Server выявляет поврежденные страницы при помощи анализа контрольных сумм или контрольных бит.

2. Перед восстановлением вам нужно получить информацию о номерах поврежденных страниц и номерах файлов, в которых эти страницы находятся. Эта информация хранится в таблице suspect_pages базы данных msdb (она заносится в эту таблицу автоматически). Номера страниц находятся в столбце page_id, а номера файлов – в столбце file_id. Надо отметить, что в таблице suspect_pages не может быть более 1000 записей. По достижении этого предела запись в таблицу просто прекращается. Поэтому рекомендуется в случае физического повреждения баз данных после восстановления очистить эту таблицу.

3. Затем запускаете команду на восстановление базы данных, например:
RESTORE DATABASE db1 PAGE = '1:51, 1:52, 1:55' FROM DISK = 'D:\SQLBackups\BackupFile1.bak';

По умолчанию восстановление запускается в оперативном режиме, без отключения пользователей от базы данных. Больше 1000 поврежденных страниц восстанавливать нельзя.

Восстановление системных баз данных

В некоторых ситуациях может потребоваться произвести восстановление системной базы данных master, в которой хранится служебная

информация всего сервера (например, информация о логинах, пользовательских базах данных, настройках сервера и т. п.).

Перед тем, как производить восстановление базы данных master, подумайте об альтернативных возможностях. Если пострадала не только эта база данных, но и пользовательские базы данных, то возможно легче и надежнее будет просто переустановить весь сервер, а затем восстановить пользовательские базы данных с резервных копий. Если повреждена база данных master, а пользовательские базы данных не пострадали, то можно думать о том, чтобы переустановить сервер или перестроить базу данных master, а пользовательские базы данных присоединить. Такой вариант будет наиболее надежным.

Восстановление базы данных master отличается от восстановления обычных баз данных некоторыми особенностями:

- производить восстановление базы данных master можно только после перезапуска сервера в однопользовательском режиме. Проще всего сделать это, запустив SQL Server из командной строки. Для этого нужно перейти в каталог, в котором находится файл sqlservr.exe (по умолчанию это C:\Program Files\Micro-soft SQL Server\MSSQL.1\MSSQL\Binn), а затем выполнить команду:

```
sqlservr.exe -m
```

- если база данных master повреждена, то сервер вполне может не запуститься. В этом случае, чтобы все-таки можно было запустить сервер и произвести процедуру восстановления, нужно перестроить базу данных master. При перестроении база данных master возвращается к своему исходному состоянию (когда сервер был только что установлен). В предыдущих версиях SQL Server для перестроения базы данных master использовалась специальная утилита rebuildm. В SQL Server 2005 для этой цели используется программа установки SQL Server;

- для базы данных master доступен только один тип резервного копирования – полное резервное копирование всей базы данных. Поэтому

восстановить вы можете только всю базу данных master целиком. Резервное копирование и восстановление журналов транзакций, а также любые другие операции восстановления (файлов, файловых групп, отдельных страниц и т. п.) для этой базы данных не предусмотрены;

- после восстановления базы данных master сервер автоматически перезагрузится.

После того, как восстановление базы данных master завершится, очень рекомендуется проверить, не возникло ли каких-то проблем на SQL Server. Могут обнаружиться проблемы:

- с логинами. Для проверки можно использовать хранимую процедуру `sp_validatelogins`;

- с пользователями баз данных. Проверку можно произвести при помощи команды:

```
sp_change_users_login @action = 'Rep-rt';
```

- со списком баз данных на сервере. Если какой-то базы данных в списке нет, но файлы ее остались на диске, эту базу данных можно заново присоединить к серверу.

Если вы произвели перестроение базы данных master, то после завершения восстановления этой базы данных обязательно нужно произвести восстановление баз данных m-del и msdb. В остальном, резервное копирование и восстановление этих баз производится так же, как и пользовательских.

Произвести резервное копирование базы данных tempdb невозможно. Поскольку эта база данных создается заново при каждом запуске SQL Server, то восстанавливать ее не нужно.

8. Автоматическая доставка журналов

Автоматическая доставка журналов (log shipping) – это еще одна технология, которая призвана повысить отказоустойчивость вашего приложения и скорость восстановления. Принцип ее работы очень прост: на

одном сервере регулярно (например, с интервалом в несколько минут) производится резервное копирование журналов транзакций. Затем эти копии журналов автоматически передаются на другой сервер, где существует копия этой базы данных, и автоматически там восстанавливаются. В результате на втором сервере у вас создается копия рабочей базы данных, которая будет синхронизироваться с рабочей базой с разницей в несколько минут.

Главное назначение данной резервной копии – это, конечно, обеспечение отказоустойчивости. В случае отказа рабочего сервера в вашем распоряжении будет резервная копия, отстающая всего на несколько минут. Правда, в отличие от кластера, автоматической смены ролей не произойдет. Вам придется вручную открыть доступ пользователям на сервер с копией базы данных. При этом еще потребуются "объяснить" клиентским приложениям, что теперь они должны обращаться на новый сервер. Это можно сделать множеством разных способов (в зависимости от текущей ситуации): поменять IP-адрес или имя запасного сервера, изменить записи на сервере DNS, использовать псевдонимы на компьютерах пользователей, перенастроить источники данных -DBС и т. п.

Вполне можно представить себе ситуацию, когда на вашем предприятии работает несколько рабочих серверов, и при этом один резервный сервер при помощи автоматической доставки журналов поддерживает резервные копии баз данных каждого из этих серверов. Любой рабочий сервер в случае выхода его из строя можно будет заменить на резервный.

Кроме отказоустойчивости, Micr-s-ft предлагает использовать резервный сервер в такой конфигурации также и для снятия нагрузки с основного сервера. Дело в том, что резервные копии журналов транзакций можно восстанавливать на сервер в так называемом режиме STANDBY(см. разд. 6.3.2). В этом случае после восстановления каждой копии журнала транзакций резервная база данных будет автоматически открываться пользователям на чтение. Теоретически ее при этом можно использовать для обслуживания запросов пользователей, которые не изменяют данные

(например, для генерации отчетов). На практике же использовать эту возможность вряд ли удастся. Причина проста: для восстановления журналов транзакций необходимо, чтобы в базе данных не было пользовательских подключений. Это ставит вас перед двумя не самыми лучшими вариантами: либо разрешить пользователям работать в базе данных постоянно (и в это время восстановление журналов транзакций производиться не будет, т. е. расхождение между рабочей и резервной базами данных будет накапливаться), либо принудительно отключать пользователей для восстановления журналов.

Несмотря на то, что графический интерфейс для настройки автоматической доставки журналов появился еще в SQL Server 2000 (а в предыдущих версиях можно было настроить ее вручную при помощи пакетов SQL Server Agent), автор еще не встречал предприятия, где автоматическая доставка журналов применялась бы для рабочих серверов. Если возникает такая потребность, то администраторы стараются использовать вместо доставки журналов репликацию. В основном это связано с тем, что репликация:

- более привычна;
- более функциональна (можно настроить больше параметров, чем для доставки журналов);
- меньше влияет на работу пользователей на резервном сервере.

В то же время доставка журналов значительно проще в настройке и в администрировании, чем репликация, и ее вполне можно рассматривать в качестве одного из возможных вариантов повышения отказоустойчивости сервера.