***Типы резервного копирования SQL Server***

***Полное резервное копирование (full backup), разностное резервное копирование (differential backup), резервное копирование журналов транзакций (transaction log backup), копирующее резервное копирование (copy-only backups)***

В SQL Server 2005 предусмотрено четыре главных типа резервного копирования.

**Первый тип** — **полное резервное копирование** (*full backup* или *base backup*). Это самый очевидный тип резервного копирования. В резервную копию записываются все данные, которые есть в базе данных. Пустые страницы при этом не копируются, поэтому если, например, файлы вашей базы данных имеют размер в 1 Гбайт, а реально данные в них занимают всего лишь 200 Мбайт, то получится резервная копия размером 200 Мбайт.

Конечно, полное резервное копирование, как и все другие типы резервного копирования, производится в оперативном режиме (*online*), 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.

Обратите внимание на один момент, с которым часто возникает путаница. Предположим, что полное резервное копирование базы данных началось в 7 часов, а закончилось в 8. Данные по состоянию на какое время оказались помещены в резервную копию — на 7 или 8 часов?

Правильный ответ — на 8 часов.

Действительно, в момент начала резервного копирования база данных стабилизируется (т. е. никакие изменения в ее файлы не вносятся для обеспечения целостности резервной копии). Однако после того, как перенос самой базы данных завершен, к резервной копии дописывается информация о всех изменениях, которые были внесены в базу данных во время резервного копирования, т. е. с 7 до 8 часов. При восстановлении резервной копии эта информация используется в автоматическом режиме.

**Второй тип резервного копирования** — **разностный** (*differential backup*, другое название, которое появилось в SQL Server 2005, — *full differential backup*). В этом случае на резервную копию записываются все изменения, которые были произведены с момента полного резервного копирования. Обратите внимание, что именно с момента последнего **полного** резервного копирования, а не другого разностного! Если вы производите полное резервное копирование по понедельникам, а во все остальные дни недели — разностное, то во вторник в резервную копию будут помещены изменения, которые произошли с понедельника по вторник, в среду — с понедельника по среду (а не со вторника по среду) и т. п.

Конечно же, разностное резервное копирование можно использовать только в дополнение к полному.

**Третий тип резервного копирования** — **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).

Это резервное копирование можно использовать как аналог добавочного (*incremental*) резервного копирования, которое предусмотрено в операционной системе. Например, можно производить полное резервное копирование по понедельникам, а в каждый другой день недели — резервное копирование журналов транзакций. В результате во вторник в резервную копию будут записаны те изменения, которые произошли с понедельника по вторник, в среду будут записаны изменения со вторника по среду и т. п. Такое резервное копирование будет выполняться быстрее, чем разностное, но придется пожертвовать скоростью восстановления. Если сбой произойдет, например, в четверг, то при использовании разностного резервного копирования вам потребуются две резервные копии: за понедельник (полная) и за четверг (разностная). В той же ситуации при использовании резервного копирования только журналов транзакций вам потребуются: полная копия за понедельник и копии журналов транзакций за вторник, среду и четверг.

**\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/**

**Четвертый тип резервного копирования** появился только в SQL Server 2005. Это так называемое **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. Например, если после обычной разностной копии будет создана полная копирующая, то при следующем разностном резервном копировании эта полная копия, созданная в режиме COPY\_ONLY, будет проигнорирована. И в разностную копию будет помещена информация, измененная с момента обычного полного резервного копирования.

Резервное копирование в режиме COPY\_ONLY (а также восстановление созданных этим способом копий) невозможно произвести при помощи графического интерфейса Management Studio. Вместо этого вам нужно будет использовать ключевое слово COPY\_ONLY в командах BACKUP и RESTORE.

В качестве дополнительного типа резервного копирования можно рассматривать **резервное копирование файлов**(*file backup*) и **файловых групп** (*filegroup backup*). Применяется оно достаточно редко и обычно только в двух ситуациях:

      когда резервная копия всей базы данных не помещается на картридж стриммера. В этом случае размер файлов или файловых групп подбирается под размер картриджа;

      когда в базе данных таблицы можно условно поделить на две группы: таблицы-справочники, в которые изменения практически не вносятся, и пользовательские таблицы, которые изменяются активно. В этом случае каждая группа таблиц помещается в свою файловую группу, а затем проводится резервное копирование каждой файловой группы со своей частотой.

Для обеспечения целостности информации при проведении резервного копирования файлов и файловых групп SQL Server 2005 автоматически определяет поколения резервных копий. Пока не будет завершено резервное копирование всех файлов или файловых групп в рамках одного поколения, журнал транзакций очищаться не будет. Поэтому администраторы применяют этот тип резервного копирования редко.

Необходимо также учитывать, что не все типы резервного копирования баз данных всегда доступны. Выбор типа резервного копирования зависит от режима восстановления базы данных. Подробно про режимы восстановления рассказывалось в [*разд. 4.5*](http://www.askit.ru/custom/sql2005_admin/m4/04_05_recovery_model.htm). Здесь мы отметим только, что поскольку при использовании режима восстановления **Simple** журнал транзакций очищается автоматически, то в этом режиме резервное копирование журнала транзакций проводить нельзя. Кроме того, в этом режиме можно производить резервное копирование только тех файловых групп, которые доступны только на чтение.

В документации SQL Server 2005 упоминается также концепция **резервных копий-снимков баз данных** (*snapshot backups*). Создание таких резервных копий производится при помощи службы Volume Shadow Copy (Теневое копирование тома) в Windows Server 2003, но стандартными средствами SQL Server 2005/Windows Server 2003 создать такие резервные копии не удастся. Для этого требуются специальные программные и/или аппаратные средства третьих фирм.

Резервные копии-снимки баз данных (*snapshot backups*) не следует путать со снимками баз данных на определенный момент времени (*database snapshots*), которые создаются обычными средствами SQL Server 2005 и также могут использоваться для восстановления (точнее, для отката базы данных к состоянию на определенное время в прошлом).