

Занятие 49. Резервное копирование

Основные понятия

В данной главе будут рассмотрены вопросы выполнения резервного копирования средствами MS SQL Server.

Зачастую многие специалисты делают резервное копирование с помощью выгрузки информационной базы в dt файл, что является в корне неверным. Выгрузка предназначена для переноса информационных баз между различными средами, например, при смене СУБД, но никак не для резервных копий.

Еще одно неоптимальное решение для резервного копирования – это копирование файлов базы данных MS SQL Server (mdf и ldf), что также является неправильным.

В обоих описанных случаях нельзя сделать копирование, не прерывая работы пользователей. При этом получившийся файл копии, даже с учетом сжатия будет довольно большим, что увеличивает стоимость хранения данных, к тому же нет возможности восстановить состояние системы на произвольный момент времени.

В данном разделе будет рассмотрен способ, который позволяет выполнять резервное копирование, не прерывая работу пользователей, получать файлы резервных копий меньшего размера, тратить меньше времени, а также автоматизировать процесс копирования.

Тема резервного копирования довольно обширна, в ней есть масса нюансов и тонкостей, подробное рассмотрение которых легко может вместиться в отдельную книгу. В данной главе будет рассмотрен необходимый минимум информации, достаточный для того, чтобы настроить резервное копирование, подходящее для подавляющего большинства информационных систем. Будут разобраны способы, позволяющие организовать восстановление системы практически на любой момент времени, в том числе и на момент сбоя.

Типы резервных копий

MS SQL Server поддерживает 3 основных типа резервных копий: полная, разностная и журнал транзакций.

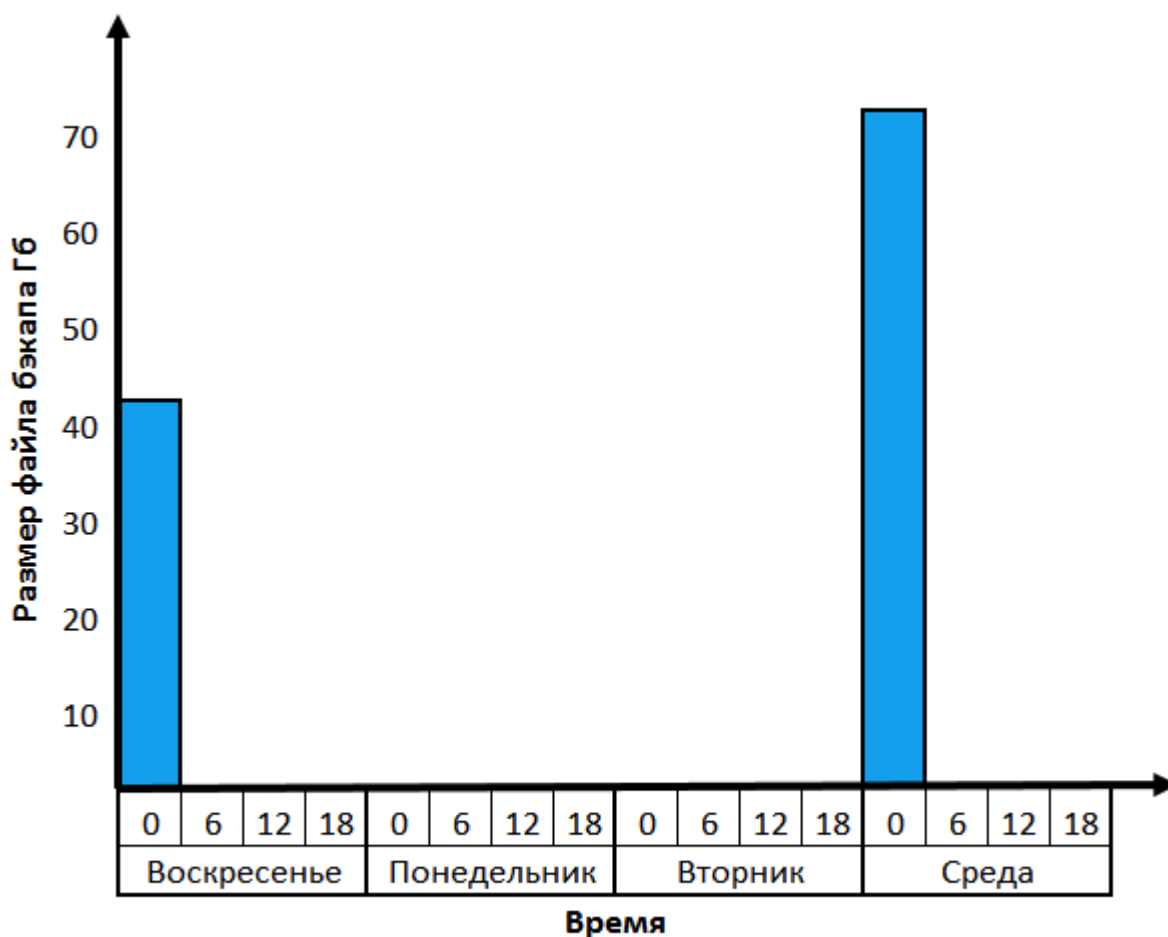
Рассмотрим поочередно каждый из этих типов.

Полная резервная копия (Full backup)

В данном случае будет создана полная копия базы данных, которая включает в себя и файл данных, и файл журнала транзакций. Это самый простой и понятный тип бэкапа. Обычно полный бэкап выполняется 1 раз в неделю, и чем больше размер базы, тем дольше он выполняется и тем больше размер файла бэкапа.

Допустим, есть база размером 40 ГБ, и каждый день, включая выходные дни, база прирастает на 10 ГБ. Для простоты будем считать, что старые данные в базе не меняются, а только записываются новые. При этом два раза в неделю, в воскресенье и в среду в полночь, выполняется полный бэкап без сжатия.

Эту ситуацию можно отразить на графике, где на одной оси показан размер файла бэкапа, а на второй – время.



Полная резервная копия

На оси времени, одна клетка означает 6 часов, цифрой обозначается первый час шестичасового интервала.

Следует помнить, что в бэкап попадают данные, которые были на момент окончания бэкапа. Если во время создания бэкапа в базе было что-то изменено, то это изменение в данный бэкап все равно попадет. Тут надо учитывать, что попадет только то изменение, которое успело успешно завершиться пока бэкап формировался.

Допустим, полный бэкап формируется один час с 20.00 до 21.00, тогда возможны следующие варианты:

- Транзакция началась 19.30 и закончилась 20.50 – изменение попадет в бэкап
- Транзакция началась в 20.30 и закончилась в 20.55 – изменение попадет в бэкап.
- Транзакция началась в 19.30 и закончилась в 21.05 – изменение не попадет бэкап
- Транзакция началась в 20.30 и закончилась в 21.05 – изменение не попадет бэкап

Неважно, когда началась транзакция, важно, чтобы она успела закончиться до того, как сформируется бэкап.

Если произойдет сбой, то для восстановления базы данных потребуется только загрузить один из файлов полного бэкапа и восстановить состояние базы либо на начало воскресенья, либо на начало среды.

Если сбой произойдет в промежутке между бэкапами, то данные будут утеряны. Получается, что интервал потери данных равен нескольким дням, это очень много.

Конечно, можно делать полный бэкап каждый день и даже чаще. Это применимо, если размер баз небольшой и достаточно дискового пространства. Но если базы очень большие, то это вызывает целый ряд проблем:

- Размер полного бэкапа сопоставим с размером базы (если не включено сжатие), что требует много дискового пространства для его хранения
- Создание полного бэкапа отнимает достаточно много времени: чем больше база, тем дольше создается бэкап
- Создание полного бэкапа сильнее нагружает оборудование, чем другие виды резервного копирования, т.к. обрабатывается большой объем данных.

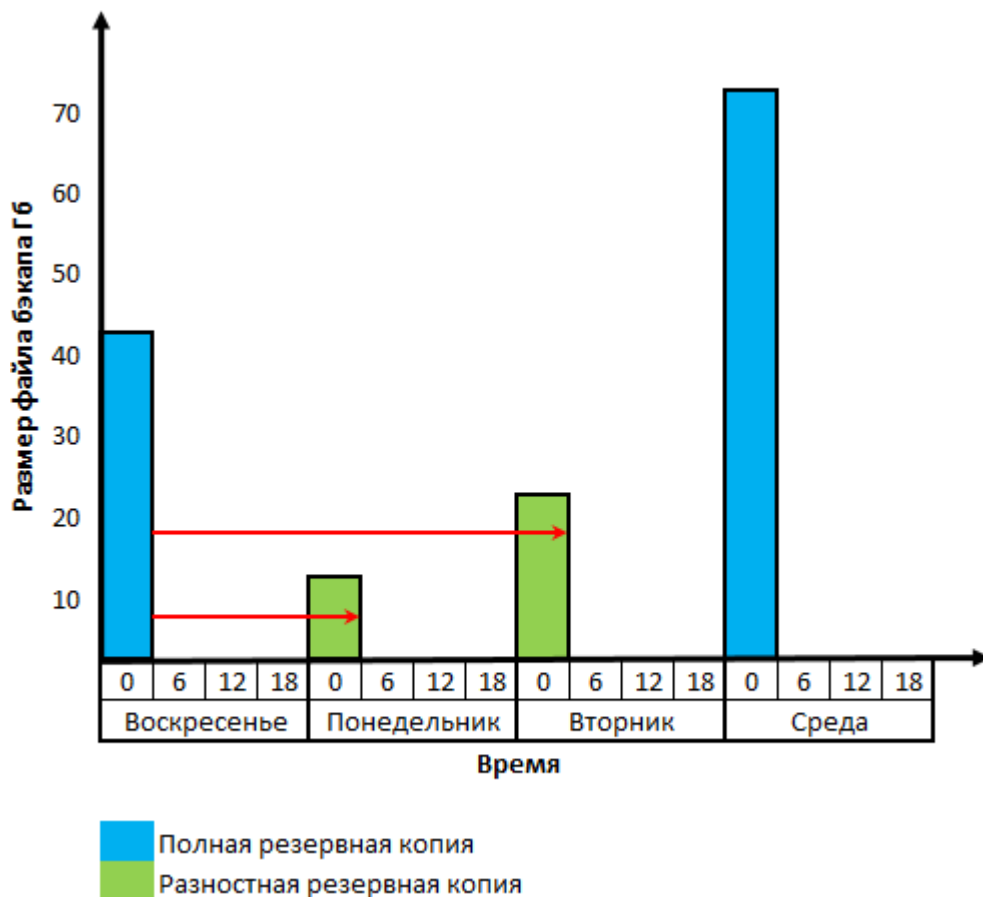
Разностная (дифференциальная) резервная копия (Differential backup)

Для того чтобы избежать вышеописанных проблем, существует разностная резервная копия, или дифференциальный бэкап. В этом случае в файл бэкапа будут помещены только те данные, которые были изменены с момента последнего полного бэкапа. Если сделать несколько разностных бэкапов, то более поздние бэкапы будут включать в себя данные предыдущих.

Так же, как и полный бэкап, разностный сохраняет данные тех транзакций, которые успешно завершились пока формировался сам бэкап.

Допустим, что помимо создания полных бэкапов два раза в неделю, было также настроено создание разностных бэкапов каждый день, за исключением тех дней, когда выполняется полный бэкап.

С помощью графика это можно представить следующим образом.



Красная линия показывает, за какой интервал времени хранит данные бэкап, расположенный справа от стрелки.

Например, бэкап вторника хранит в себе данные начиная с воскресенья (с момента окончания полного бэкапа), до вторника (до момента окончания разностного бэкапа).

На графике видно, что разностная копия с каждым днем увеличивается, т.к. включает в себя все данные, которые есть в предыдущих разностных копиях. В понедельник разностный бэкап составлял 10 ГБ, а во вторник – уже 20 ГБ, из которых 10 ГБ – данные, добавленные за воскресенье, и 10 ГБ – новые данные за понедельник. В четверг разностный бэкап будет составлять 10 ГБ, т.к. в среду был сделан полный бэкап.

За счет того, что разностные бэкапы хранят только измененные данные, они занимают гораздо меньше места и создаются намного быстрее полных бэкапов.

Теперь в рассматриваемом примере можно восстановить данные на начало каждого дня. Например, если потребуется восстановить данные на начало вторника, тогда необходимо будет сначала восстановить полный бэкап, созданный в воскресенье, а потом применить разностный бэкап, созданный во вторник.

Если нет полного бэкапа, то разностный бэкап не имеет смысла, т.к. разностный бэкап хранит только дельту данных. Чтобы применить дельту, нужна опорная точка, в роли которой и выступает полный бэкап.

При использовании разностного бэкапа интервал потери данных сократился до одного дня. Этот результат, конечно, намного лучше, чем несколько дней из предыдущего примера, но все равно недостаточен с точки зрения надежности. В случае сбоя в вечернее время будут утеряны данные за целый рабочий день.

Резервная копия журнала транзакций (Log backup)

Перед тем как ознакомиться с данным разделом, рекомендуется повторить раздел «Общие сведения о файле данных и журнале транзакций» и освежить в памяти понятия **файл данных** и **файл журнала транзакции**.

Для того чтобы сократить интервал возможной потери данных до нескольких часов или даже нескольких минут, существует возможность создавать бэкап журнала транзакций (бэкап лога). В этом случае создается не копия самих данных, а копия журнала транзакций. Такой бэкап будет создаваться гораздо быстрее и иметь гораздо меньший размер, чем рассмотренные ранее бэкапы.

За счет малого объема и нетребовательности к ресурсам бэкап лога можно создавать каждые несколько минут. Обычно бэкап лога выполняют 1 раз в 15–30 минут.

Следует помнить, что файл журнала транзакций не хранит сами данные, он лишь содержит информацию о том, какие изменения были произведены в базе, что было и что стало в результате работы каждой транзакции.

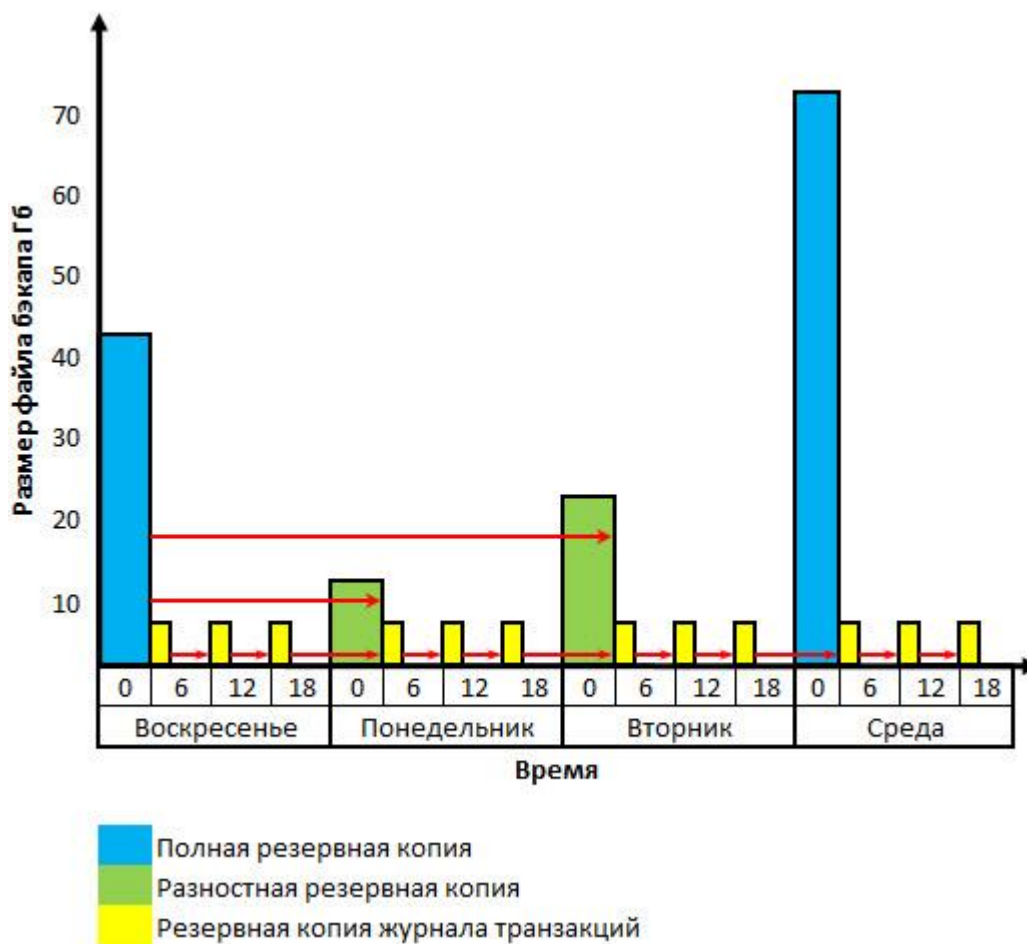
Используя эту информацию, можно восстановить состояние базы на определенный момент времени. Журнал не хранит информацию, которая не нужна для восстановления данных, например, там нет информации о том, какой запрос привел к изменению или какой план был у этого запроса.

Бэкап лога доступен только в том случае, если у базы установлен **полный режим восстановления**. О режимах восстановления и о том, как они связаны с бэкапами, будет рассказано в отдельном разделе курса. Сейчас достаточно знать, что у всех баз 1С режим восстановления по умолчанию как раз полный, так что бэкап лога для них доступен.

При использовании полной модели восстановления размер журнала транзакций увеличивается по мере того, как в системе происходят изменения данных. При выполнении резервного копирования журнала (именно журнала, а не данных), он усекается, т.е. уменьшается в размере. Именно за счет этого размер бэкапа лога получается таким маленьким и создается так быстро.

Бэкап журнала транзакций содержит в себе данные, которые были изменены начиная с момента предыдущего бэкапа журнала транзакций, а не с момента полного или разностного бэкапа. Например, если делать бэкап журнала каждый час, то его размер будет примерно одинаков, т.к. бэкап журнала за второй час не будет включать в себя данные за первый час, а это значит, что размер бэкапа будет минимальным.

Допустим, было решено каждые 6 часов выполнять бэкап журнала транзакций, за исключением тех случаев, когда выполняется полный или разностный бэкап. Представим, что за 6 часов логи растут на 5 ГБ, тогда график примет следующий вид:



Из графика видно, что бэкап лога имеет примерно одинаковый размер, т.к. не хранит в себе данные предыдущих бэкапов. Например, бэкап журнала, созданный в 18.00 в понедельник, хранит в себе только измененные данные журнала начиная с 12.00 до 18.00 понедельника, т.е. с момента начала создания предыдущего бэкапа лога до момента создания бэкапа в 18.00.

Таким образом, имея бэкапы логов и полный бэкап базы, можно восстановить состояние системы на любой момент времени. Например, если во вторник в 18.30 произошел сбой, то, для того чтобы восстановить базу на ближайший к сбою момент времени, можно воспользоваться одним из двух вариантов.

Вариант 1

Необходимо сначала накатить полный бэкап, созданный в воскресенье, потом разностный бэкап, созданный во вторник, и потом последовательно накатить бэкапы журналов: сначала 6-часовой, потом 12-часовой и в конце 18-часовой.

Вариант 2

Допустим, что бэкапы разностных копий были повреждены или удалены. В этом случае все равно можно восстановить базу данных на требуемый момент времени, используя бэкапы логов.

Для этого необходимо сначала накатить полный бэкап, созданный в воскресенье, а потом последовательно накатить все бэкапы журналов, начиная с 6-часового бэкапа воскресенья и заканчивая 18-часовым бэкапом вторника. Этот вариант будет работать дольше, чем первый, но его также можно использовать.

Этот вариант возможен, т.к. журнал хранит список всех изменений, происходящих с базой: что было и что стало с данными, а это значит, что, последовательно применив все изменения из журнала, мы получим базу в том состоянии, в котором она была на момент создания бэкапа лога.

В любом из двух вариантов данные будут потеряны максимум за последние 30 минут.

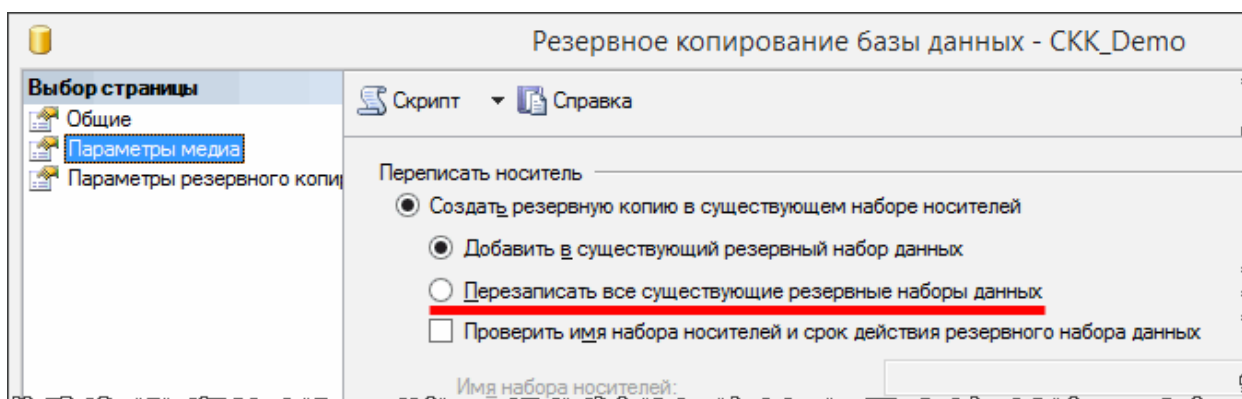
Последовательность наката бэкапов журнала очень важна. Например, если имеется 6-часовой бэкап, но 12-часовой бэкап лога будет поврежден или утерян, то мы не сможем накатить 18-часовой бэкап лога. В результате получится восстановить состояние базы только на 6.00 утра.

Когда речь идет о бэкапах лога, возникает понятие цепочки журналов транзакций.

Цепочка журналов транзакций – это непрерывная последовательность резервных копий журнала транзакций. Резервные копии журналов должны строго следовать друг за другом. Если в цепочке не хватит хотя бы одного журнала, то цепочка прервется, а это значит, что восстановить данные из бэкапа лога можно будет только до момента обрыва цепочки.

Цепочка журналов начинается с полной резервной копии базы данных. Обычно новая цепочка журналов начинается, только когда создается первая резервная копия базы данных или после переключения модели восстановления с простой на полную или на модель с неполным протоколированием.

Существующая цепочка журналов прерывается, если выбрана перезапись существующих наборов резервных копий в момент создания полного бэкапа.



Чтобы восстановить базу данных на момент точки сбоя, нужна неповрежденная цепочка журналов. Непрерывная последовательность резервных копий журналов должна следовать до точки сбоя.

В приведенном выше примере цепочка прерывается на 12-часовом бэкапе лога, потому что он поврежден или утерян, а это значит, что нельзя использовать бэкап на 18 часов, т.к. в нем хранится только дельта данных по отношению к 12-часовому бэкапу. Из этого следует, что можно использовать только бэкапы журнала, созданные до 12 часов вторника.

При наличии непрерывной цепочки журналов транзакций можно восстановить состояние базы не только на момент создания последнего бэкапа лога, но и на любой момент времени в пределах цепочки журналов. Например, получен полный бэкап воскресенья на 0.00 и есть цепочка журналов транзакций начиная с 6.00 воскресенья до 18.00 вторника (включительно), как это показано на рисунке выше. В таком случае возможно восстановить базу на любой момент времени в промежутке между 0.00 воскресенья и 18.00 вторника.

Как восстановить базу на определенный момент времени, будет показано в соответствующем видеоуроке.

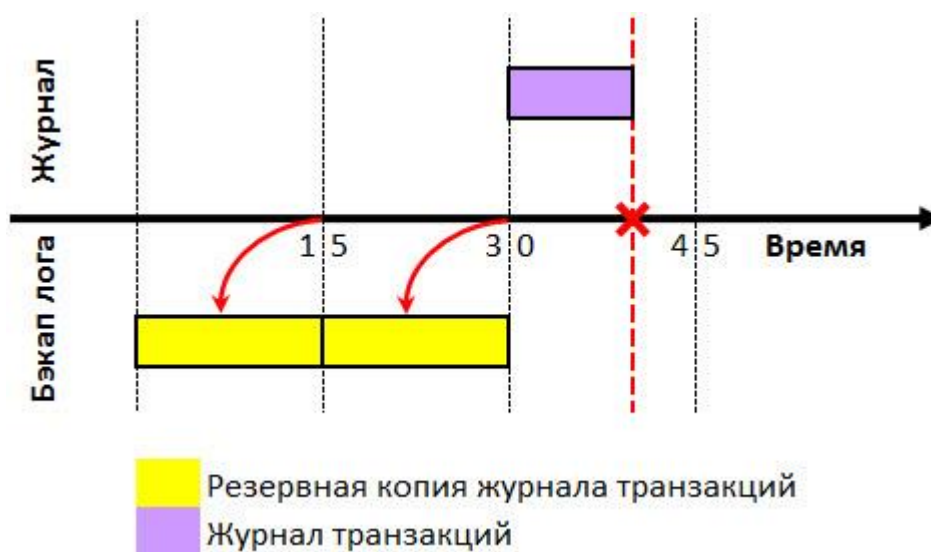
Следует понимать, что сам по себе бэкап лога бесполезен. Имея только бэкапы журнала, нельзя восстановить базу, для этого нужен хотя бы один полный бэкап, при этом цепочка журналов транзакций от полного бэкапа до последнего бэкапа журнала должна быть непрерывной.

Как видно из вышеописанного примера, бэкапы лога крайне важны и их обязательно нужно делать регулярно в течение дня. Если бэкапа лога нет, то систему можно будет восстановить только на момент создания последней разностной копии (при наличии предварительно сделанной полной резервной копии).

Резервная копия заключительного фрагмента журнала

Даже если выполнять бэкап лога каждые 15 минут, то данные за несколько минут перед сбоем все равно могут быть утеряны.

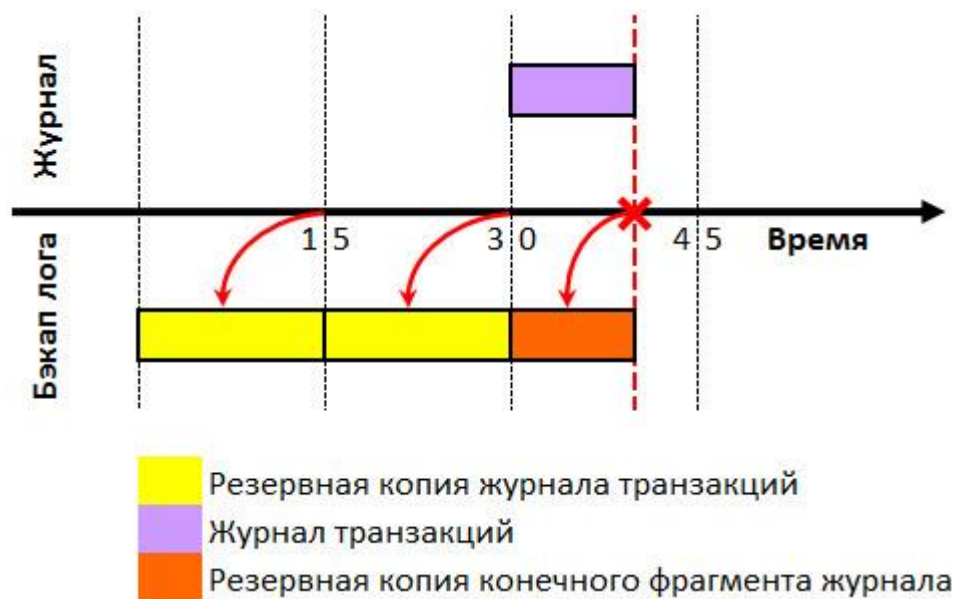
Допустим, бэкап лога выполняется каждые 15 минут, при этом сбой случился на 40-й минуте. Эту ситуацию можно представить следующим образом.



Черной пунктирной линией показаны 15-минутные отрезки, а красной пунктирной линией показан момент времени, когда произошел сбой. Видно, что в 15 минут был создан бэкап журнала, который включает в себя информацию с начала часа по 15-ю минуту. В 30 минут был создан еще один бэкап, который включает в себя журнал с 15-й по 30-ю минуту. После последнего бэкапа лога журнал продолжал фиксировать изменения, и в 40 минут произошел сбой.

Если использовать только полный бэкап и бэкапы логов, то можно восстановить базу данных на момент 30-й минуты указанного часа. Данные, введенные с 30-й по 40-ю минуту, будут утеряны. Чтобы не потерять данные даже за эти 10 минут, следует использовать резервную копию заключительного фрагмента журнала.

Резервная копия заключительного (конечного) фрагмента журнала – это бэкап лога базы уже после того, как произошел какой-то сбой, например, диск с файлами данных базы вышел из строя. В бэкап конечного фрагмента входят данные журнала, которые не попали в предыдущий бэкап лога.



Если в результате сбоя сам журнал не пострадал (он был на другом диске), то можно восстановить данные на момент сбоя, т.е. на 40-ю минуту. Это значит, что никакие данные не будут утеряны, кроме транзакций, которые были открыты на момент сбоя – по ним будет сделан откат.

Если в результате сбоя файл журнала был поврежден, тогда восстановить данные на момент сбоя уже не получится. В этом случае придется взять бэкапы логов и восстановить базу на 30-ю минуту. Именно поэтому, а также по ряду других причин, рекомендуется располагать файлы журналов на надежном носителе с дублированием, например, использовать RAID10.

Чтобы восстановить базу на время, максимально близкое к моменту сбоя, необходимо выполнить следующую последовательность действий:

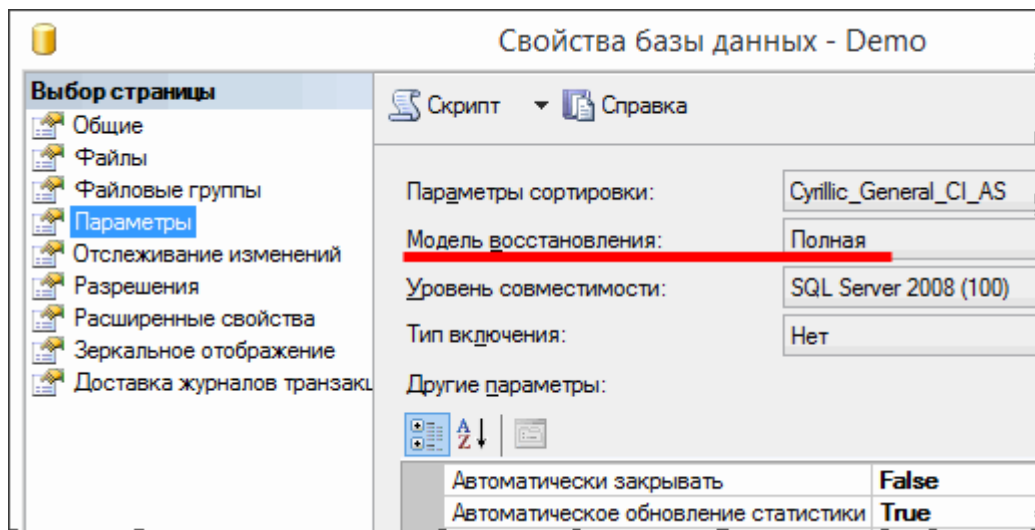
- Сделать резервную копию заключительного фрагмента журнала
- Накатить полный бэкап, ближайший по времени к моменту сбоя
- Накатить разностный бэкап, ближайший по времени к моменту сбоя
- Накатить поочередно бэкапы логов с момента разностного бэкапа до момента сбоя
- Накатить бэкап заключительного фрагмента журнала.

В этой цепочке разностных бэкапов может и не быть, главное, чтобы были полный бэкап и непрерывная цепочка журналов с момента полного бэкапа.

Отметим, что выполнять всю последовательность действий вручную совершенно не обязательно: MS SQL Server самостоятельно сортирует и восстанавливает бэкапы в нужной последовательности.

Модели восстановления

У всех баз данных MS SQL Server есть свойство «Модель восстановления».



Модель восстановления – это свойство базы данных, которое управляет процессом регистрации транзакций, определяет, требуется ли для журнала транзакций резервное копирование и какие типы операций восстановления доступны.

Если говорить проще, то данное свойство определяет поведение журнала транзакций, а именно – за какой период будут храниться данные в журнале и какие события в нем будут фиксироваться. Модель восстановления играет важную роль в процессе восстановления базы из бэкапа.

Модель восстановления может принимать одно из трех значений: полная, простая и с неполным протоколированием. Модель восстановления можно изменить в свойствах базы в любой момент времени с любого на любое другое значение.

Рассмотрим каждую из моделей восстановления подробнее.

Полная (Full)

При полной модели восстановления журнал транзакций хранит максимальную информацию обо всех транзакциях. При этом хранятся не только текущие активные транзакции, но и все изменения всех предыдущих транзакций.

Если, например, бэкап журнала не выполнялся в течение месяца, тогда файл журнала транзакций содержит информацию обо всех изменениях в базе, которые произошли за этот месяц. Следовательно, при полной модели восстановления файл журнала постоянно увеличивается до тех пор, пока не будет сделан бэкап лога.

Если не делать бэкап лога длительное время, файл журнала может расти неограниченно и со временем занять все свободное место на диске (если не установлено ограничение на размер журнала).

В момент создания копии журнала он усекается и в нем остаются только данные об активных на текущий момент транзакциях. Усечение журнала не означает уменьшение размера файла журнала на диске, просто место внутри этого файла будет освобождено. При выполнении полного или разностного бэкапа усечение журнала не происходит.

Чтобы сократить объем, занимаемый файлом журнала на диске, необходимо выполнить сжатие журнала, т.е. выполнить команду [DBCC SHRINKFILE](#). Следует понимать, что усечение журнала – это освобождение места внутри физического файла, а сжатие файла журнала – это уменьшение размера файла журнала на диске.

Особо отметим, что не следует путать полную модель восстановления и полный бэкап – это разные понятия.

Простая (Simple)

В данном случае в журнале транзакций будет храниться минимальный набор данных. Когда файл журнала транзакций будет заполнен на 70%, система начнет писать данные в начало файла, затирая самые ранние данные. При такой модели восстановления файл журнала транзакций будет занимать минимальный объем на диске, но не будет хранить никакой истории транзакций. Так как история транзакций не хранится, то нет смысла делать бэкапы логов.

Таким образом, если для базы данных выбрана простая модель восстановления, то бэкап журнала транзакций для нее становится недоступным.

С неполным протоколированием (Bulk logged)

Дополнение к полной модели восстановления и отличается от нее лишь тем, что в журнал транзакций пишется минимальная информация при массовых операциях по изменению данных.

В данной модели восстановления при массовом изменении данных журнал транзакций не будет разрастаться так сильно, как при полной модели. Для баз данных, работающих на платформе 1С:Предприятие, данная модель не имеет большого практического смысла, поэтому далее будут рассматриваться только первые две модели восстановления.

Влияние модели восстановления на резервное копирование

Модель восстановления напрямую влияет на процесс восстановления данных из бэкапов.

Полная модель восстановления

Если база данных использует полную модель восстановления, то с этой базы можно делать все 3 вида резервных копий: полную, разностную и журнала транзакций. Само название «Полная модель восстановления» говорит о том, что можно использовать максимальные возможности резервного копирования. При наличии полного бэкапа и цепочки журналов транзакций можно восстановить состояние базы данных на любой момент времени с момента полного бэкапа до момента создания последнего бэкапа лога в цепочке.

Допустим, есть следующие бэкапы:

- Полный бэкап за воскресенье 0.00
- Бэкап лога за воскресенье 6.00
- Бэкап лога за воскресенье 12.00
- Бэкап лога за воскресенье 18.00
- Разностный бэкап за понедельник 0.00
- Бэкап лога за понедельник 6.00
- Бэкап лога за понедельник 12.00
- Бэкап лога за понедельник 18.00
- Разностный бэкап за вторник 0.00

Время показывает момент завершения бэкапа.

В этом случае можно восстановить состояние системы на любой момент времени начиная с воскресенья 0.00 и заканчивая понедельником до 18.00, а также на момент времени 0.00 вторника. Чтобы восстановить данные на 14.00 понедельника, нужно будет:

- Сначала развернуть полный бэкап за воскресенье
- Сделать накат разностного бэкапа за понедельник
- Сделать накат трех бэкапов лога за 6, 12 и 18 часов.

Таким образом, если бэкапы логов выполняются достаточно часто, то риск потери данных будет минимальным.

Рассмотрим еще один пример.

Допустим, есть тот же набор бэкапов, но два из них повреждены:

- Полный бэкап за воскресенье 0.00
- Бэкап лога за воскресенье 6.00
- Бэкап лога за воскресенье 12.00
- **Бэкап лога за воскресенье 18.00 (файл бэкапа поврежден)**
- Разностный бэкап за понедельник 0.00
- Бэкап лога за понедельник 6.00
- **Бэкап лога за понедельник 12.00 (файл бэкапа поврежден)**
- Бэкап лога за понедельник 18.00
- Разностный бэкап за вторник 0.00

В таком случае можно восстановить состояние системы на следующих интервалах времени:

- С 0.00 воскресенья до 12.00 воскресенья на любой момент времени
 - Накатить полный бэкап, потом 2 бэкапа лога
- С 0.00 понедельника до 6.00 понедельника на любой момент времени
 - Накатить полный бэкап, потом разностный бэкап и в конце бэкап лога
- С 0.00 понедельника до 0.00 понедельника на момент создания бэкапа
 - Накатить полный бэкап, потом разностный бэкап за понедельник и разностный бэкап за вторник.

Из-за того, что цепочка журналов прервалась, восстановить данные в промежутке между поврежденным бэкапом и следующим за ним полным или разностным бэкапом невозможно. В данном примере нельзя восстановить систему на момент в интервале с 12.00 воскресенья до 0.00 понедельника и с 6.00 понедельника до 0.00 вторника.

Простая модель восстановления

Если база данных использует простую модель восстановления, то бэкап журнала для нее недоступен, т.к. в этом случае журнал не хранит всю историю транзакций. В данном случае доступны только полные и разностные резервные копии. Это значит, что данные можно восстановить не на любой момент времени, а только на момент создания полного или разностного бэкапа.

Допустим, есть следующие бэкапы:

- Полный бэкап за воскресенье 0.00
- Разностный бэкап за понедельник 0.00
- Разностный бэкап за вторник 0.00

В данном случае нет возможности восстановить базу данных на любой момент времени. При наличии таких бэкапов базу можно восстановить только на начало любого из трех дней.

Это значит, что есть риск потерять данные за один рабочий день, т.к. разностный бэкап в течение дня обычно не выполняется, чтобы не создавать излишнюю нагрузку на оборудование.

Какую модель восстановления выбрать?

Для рабочих баз следует использовать полную модель восстановления, чтобы иметь возможность выполнять бэкапы логов и при необходимости восстановить базу данных на момент сбоя. Минус данной модели только в том, что размер файла журнала растет, пока не будет сделан бэкап лога.

Следует отметить, что простая модель восстановления, дает некоторое преимущество в скорости на [операциях массовой вставки](#). В этом случае, пишется меньше данных на диск в журнал транзакций и за счет этого получается ускорение. Однако если диск с журналом транзакций не являлся «узким» местом, то это ускорение будет практически не заметным.

В любом случае для рабочих баз нужно выбирать полную модель.

Для баз разработчиков, различных технических и тестовых баз, например, баз сервисов и ЦУП, можно использовать простую модель восстановления, т.к. там обычно не требуется восстанавливать данные на какой-то определенный момент времени. Файл журнала для этих баз расти практически не будет.

Модель	Преимущества	Недостатки	Применение
Полная	Восстановление на любой момент времени	Растет журнал лога, необходимо периодически делать его бэкап	Для рабочих баз
Простая	Журнал практически не растет	Нельзя восстановить на любой момент времени	Для тестовых и технических баз

Стратегия резервного копирования

У системного администратора обязательно должна быть стратегия резервного копирования: какие бэкапы когда создавать, сколько будут храниться копии бэкапов, где они будут храниться и т.д.

Не рекомендуется хранить бэкапы на том же физическом диске, на котором располагается сама база данных.

В большинстве случаев подходит следующая стратегия резервного копирования:

Создание бэкапов

Полный бэкап – 1 раз в неделю на выходных. Конечно, можно делать полный бэкап каждый час, но это сильно загрузит оборудование и бэкапы будут занимать слишком много места на диске.

Разностный бэкап – 1 раз в день в ночное время.

Бэкап лога – каждые 15–30 минут. Бэкап лога можно делать и чаще, но обычно это не имеет большого смысла, т.к. в случае сбоя можно будет сделать бэкап конечного фрагмента журнала. Даже если бэкап конечного фрагмента сделать не удастся (что бывает крайне редко), то потеря данных за 15-30 минут обычно не является большой проблемой: данные за потерянные 15-30 минут можно быстро внести вручную.

Если базы небольшие и достаточно места на дисках, тогда можно отказаться от разностных бэкапов. В этом случае можно делать полные бэкапы каждый день и бэкапы лога каждые 15-30 минут.

Местоположение бэкапов

Бэкапы должны храниться на физическом диске, отличном от диска, на котором расположена база данных. Рекомендуется хранить файлы бэкапов минимум в двух разных местах.

Обычно одна копия хранится территориально там же, где расположена база данных, но на другом диске. Это нужно для того, чтобы в случае сбоя можно было быстро произвести восстановление и не тратить время на копирование файлов бэкапа из другого места.

Вторая копия должна храниться территориально в другом месте, например, на удаленном сервере или в облачном хранилище. Вторая копия необходима на случай повреждения или утери первой копии.

При разработке стратегии резервного копирования нужно исходить из того факта, что весь мир настроен против вас. Уборщицы, американские шпионы, метеориты и дикие животные – все они только и ждут за углом, чтобы уничтожить ваши данные. Не давайте им шанса.

При выборе носителя для бэкапов следует помнить о том, что чем выше у него скорость передачи данных, тем быстрее восстановится база из бэкапа. Хранить полный бэкап базы в 1Тб на диске с интерфейсом USB 2.0 – не очень хорошая идея.

Чтобы снизить стоимость хранения, можно для локальной копии бэкапа использовать быстрый диск, но хранить там бэкап только за последнюю неделю, а на удаленном сервере использовать более медленный диск и хранить там бэкапы за последние несколько недель. В подавляющем большинстве случаев, даже если случится сбой, то для восстановления данных будет достаточно локальной копии бэкапа.

Период хранения бэкапов

Если полный бэкап создается каждую неделю, то обычно нет смысла хранить бэкапы старше 2-3 недель. В некоторых случаях может потребоваться хранить полные бэкапы за несколько лет, например, по требованию бухгалтерии.

Инструкция по восстановлению

Обязательно должна быть четкая и понятная инструкция, описывающая порядок действий в случае сбоя. Даже если систему обслуживает опытный специалист, который все знает и все умеет, в стрессовой ситуации он тоже может что-то забыть. Это может привести к ошибкам или просто увеличить время простоя. Наличие инструкции имеет еще один неочевидный плюс: если специалист, обслуживающий систему, уходит в отпуск, он должен передать план действий в случае сбоя своему заместителю. В первую очередь это в интересах самого специалиста, т.к. вряд ли кому-то понравится вызов на работу ранним утром во время отпуска.

Инструкция должна быть максимально простой, но подробной. Возможно, стоит записать видеоинструкцию: это может занять меньше времени, чем написание ее бумажного варианта.

Запомните главное: даже самая плохая инструкция из 10 строк гораздо лучше, чем вообще никакой.

Регулярные проверки

Необходимо периодически выполнять плановую проверку восстановления из бэкапов. Например, можно восстанавливать бэкап рабочей базы в копию. Проверки полезны по следующим причинам:

- **Оценка времени на восстановление.** Вы должны точно знать сколько времени займет восстановление базы. Например, если полгода назад база восстанавливалась за 10 минут, то это вовсе не значит, что сейчас она будет восстанавливаться за то же время. Необходимо проверить разные сценарии восстановления, например, сколько времени займет восстановление из локальной копии бэкапа, а сколько – из копии на удаленном сервере. Скорость соединения с удаленным сервером здесь будет играть важную роль.
- **Отработка нестандартных ситуаций.** Что делать, если полный бэкап оказался поврежден? Как взять бэкап с удаленного сервера, если нет соединения с интернетом или соединение очень медленное? Что делать, если есть полный бэкап и бэкапы логов, но разностный бэкап оказался поврежден? Ответы на все эти вопросы нужно обязательно получить, проверив все на практике. Все возможные нестандартные ситуации, которые удалось придумать, и пути их решения нужно обязательно отразить в инструкции. Потратив один раз время на создание инструкции, в будущем вы сэкономите гораздо больше и времени, и нервных клеток, а также гарантируете себе спокойный отпуск без неожиданностей.

- **Регулярная практика.** Если случится сбой, думать будет уже некогда, в этом случае нужно на автомате выполнить уже отработанные действия, сверяясь с инструкцией при необходимости. Регулярные проверки как раз дают возможность эти действия отработать. Как говорится: кто готов, тому не надо готовиться.

Для того чтобы не забывать регулярно выполнять плановую проверку, лучше всего настроить автоматическое напоминание. Например, можно делать проверку бэкапов 1 раз в каждый первый понедельник месяца.