

Резервное копирование и восстановление MS SQL Server

Многие считают, что техника сейчас надежна и из-за своей лени никогда не делают резервных копий. Техника хороша, но на моих глазах умерло уже несколько винчестеров, украдено из офисов 5 компьютеров и даже полностью сгорел вместе с кабинетом один сервер. А кто из жителей великих башен города New York думал, что к ним в гости прилетят самолеты? Кто-то скажет, что такие слава жестоки в отношении беды других людей, но мы же тоже не застрахованы от терроризма и Россия тоже видела жестокие действия террористов. И как бы не было больно, закрывать на это глаза нельзя. Необходимо делать все, чтобы данные были сохранены в любой ситуации.

Политика

Нравится это выражение «политика безопасности». В нашей стране это пахнет чем-то нечистым и неприятным. Ну что поделаешь, если такое выражение придумали буржуи. В общем, не будем вникать в сложности терминологии, а просто поговорим простым языком о том, когда и как нужно делать резервное копирование.

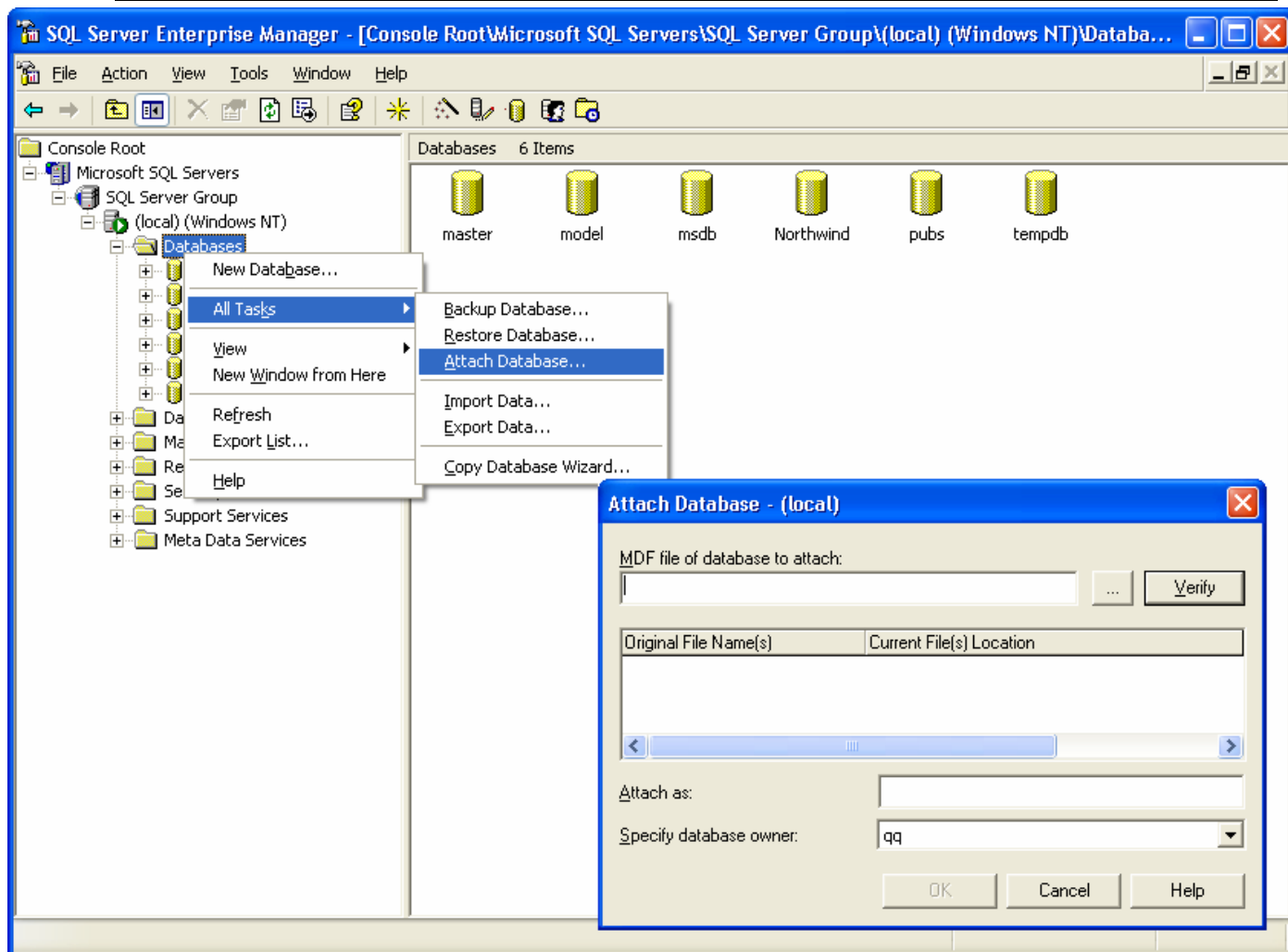
Все зависит от того, какие данные хранятся в базе, насколько страшной является их потеря и как сильно можно получить по заднему месту за простой в связи с восстановлением после сбоя. На простой можно наплевать, если будешь уверенным, что хотя бы 95% данных восстановится. Большинство боссов не будет бить твое заднее место, а наоборот, вылежит его языком до блеска за такую работу.

Так как я чаще всего по ходу своей работы использую MS SQL Server, то резервное копирование будем рассматривать на его примере. Тем более что здесь больше возможностей и выбор правильной политики может оказаться сложной задачей. В других базах данных дело обстоит чаще всего намного проще.

Настройки

Большинство серверов хранят все настройки прямо в базе данных в виде системных таблиц или в виде отдельной базы данных. В MS SQL Server вся служебная инфа хранится в базе данных Master. Здесь есть инфа о правах доступа, объектах базы данных, пользователях и многое другое. Многие специалисты рекомендуют делать резервную копию этой базы после каждого изменения каких-либо объектов метаданных. В принципе, это не сложно и не долго, потому что Master имеет не слишком большой размер, и его резервная копия будет делаться достаточно быстро и займет не очень много места.

Для повышения надежности, мы рекомендуем располагать файлы данных и журнала транзакций на разных физических дисках. Если диск с данными накроется, то по журналу транзакций можно восстановить данные.



Подключение базы данных.

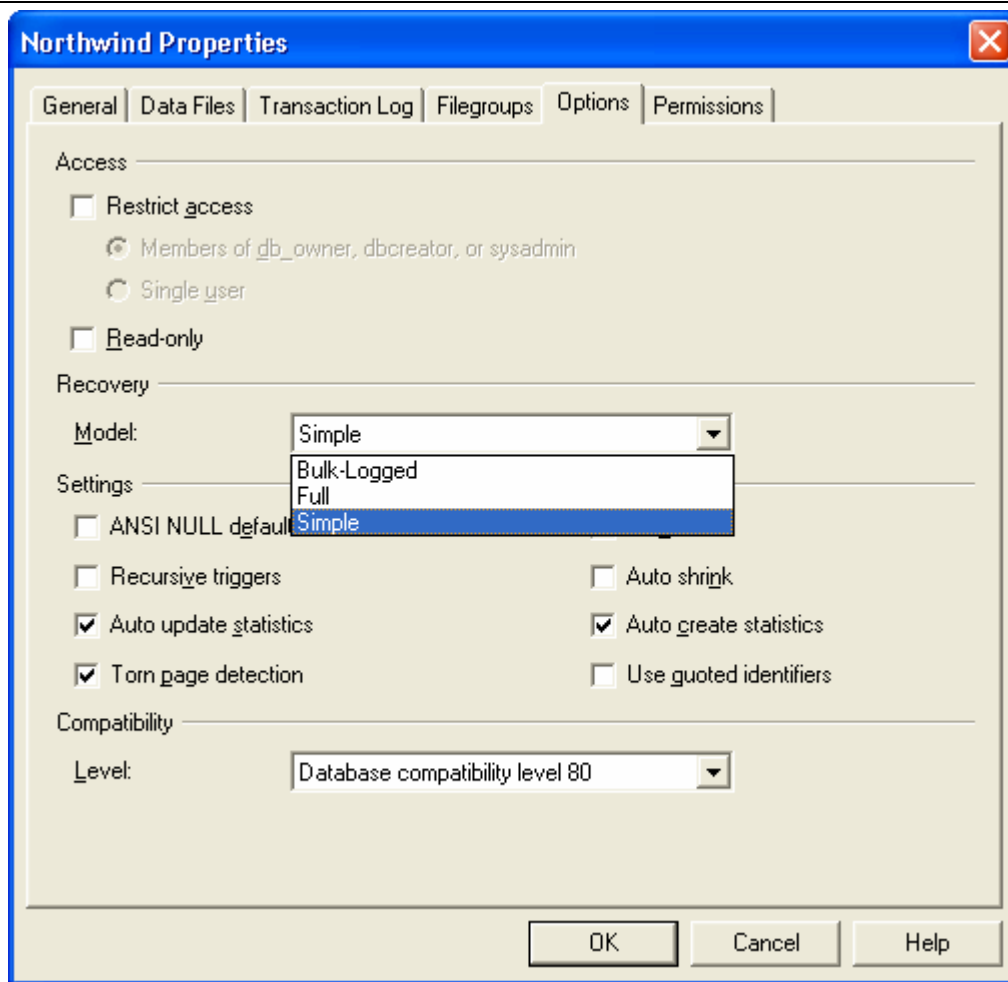
Просто выбери MDF файл и база будет подключена

А вот [1] рекомендует не заморачиваться с этой ерундой. На MS SQL Server 2000 уже не раз проверено, если файлы базы данных, а лучше и журнала транзакций доступны и не были разрушены, то их легко скопировать на другую машину и просто подключить (Attach database) к другому серверу MS SQL Server. Для этой операции журнал транзакции желателен, но не есть обязательный. База данных будет работать, как будто родная.

В случае с другими серверами баз данных, мы не можем гарантировать такой шутки с сохранением системных данных. Каждый производитель в своем продукте использует свой способ хранения системной информации.

Методы резервирования

Теперь переходим к рассмотрению резервирования самих данных. Если системные данные чаще всего занимают не много места, изменяются достаточно редко и их можно резервировать полной копией, то для данных нужно выбирать что-то более эффективное. В MS SQL Server для ускорения и облегчения жизни есть три метода создания резервных копий:



*Выбор модели резервирования происходит
в свойствах базы на закладке Options*

- Simple это самая простая модель. При ее использовании, после каждой резервной копии высвобождается свободное место на диске, занимаемое журналом транзакций. Это значит, что файл журнала не будет расти до беспредельных размеров.

При использовании этой модели будет самая быстрая производительность при массовой загрузке данных в базу. С другой стороны нет никакой возможности восстановить изменения, которые были сделаны с момента последней резервной копии. В случае аварии просто восстанавливаем последнюю копию, а изменения, сделанные после этого бесследно теряются.

Резервные копии нужно хранить в отдельном кабинете или даже здании. Пожары и террористические акты еще никто не отменил. Если такой возможности нет, то несгораемый сейф твое спасение.

- Full полное резервирование. Самая мощная модель, при которой можно создавать промежуточные копии, например, резервировать журнал транзакций. Мощность модели заключается и в том, что базу можно восстановить на любой момент времени. Например, если данные были разрушены в какой-то момент времени, то мы всего лишь восстанавливаем данные на пять минут раньше страшного момента и все данные на родине. При этом журнал при резервировании не очищается и растет, пока явно не будет зарезервирован, поэтому будь готов к хранению на винте лишних данных. У меня есть базы, в которых журнал за месяц вырастает так, что становится больше чем файл данных. Помимо этого, база становится чувствительна к целостности журнала и если он нарушен, то восстановление последних изменений становится проблематичным.

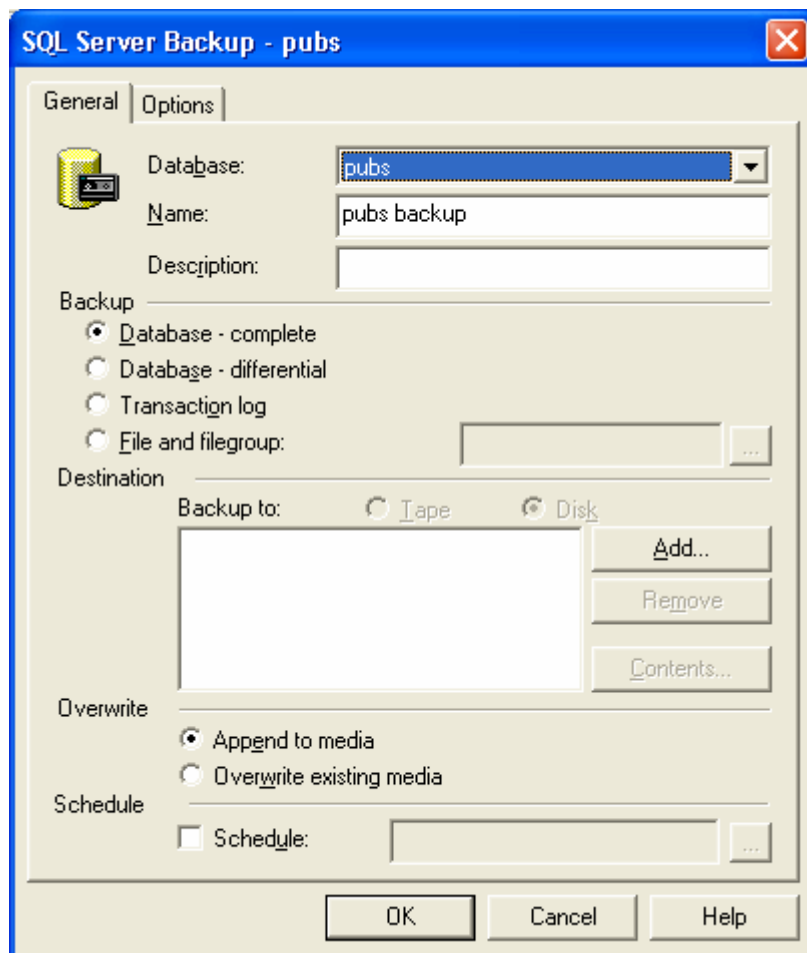
Самый главный недостаток – каждая операция подробно резервируется. Это значит, что при массовой загрузке каждая операция записи журналируется и скорость работы сервера оставляет желать лучшего.

- Bulk-Logged это упрощенный вариант полной резервной копии. В этой модели при выполнении массовых операций (загрузка большого числа данных или создание индекса) в журнале сохраняется минимум необходимой инфы, а именно, только факт выполнения этой операции. Именно поэтому в модели Bulk-Logged массовые изменения выполняются быстрее. При этом если выполнена подобная операция, то восстановить данные на определенную точку времени станет уже невозможно.

Не забываем, что данные могут пропасть и от вторжения со стороны хакера, если он сотрет все содержимое базы или всего диска.

Простое резервирование

Теперь поговорим о том, когда и что резервировать. Если ты выбрал для себя простую модель, то при создании резервной копии сможешь выбрать создание полной резервной копии (Database complete) или дифференцированной (Database differential).



Выбор типа резервирования данных

Если выбрать Database complete, то создается полная копия всех страниц данных базы. Если база занимает пару сотен мегабайт, то на простейшем сервере эта операция не отнимет много времени. А если размер достигает нескольких гигабайт, то это уже напряжесто для сервака. Если попытаться резервировать во время работы пользователей, то производительность резко упадет, и полетят жалобы на вечные тормоза. В этом случае такие резервные копии нужно создавать только в не рабочее время. Я рекомендую это делать после окончания рабочего дня, а лучше ночью, потому

что по практике, в большинстве коммерческих контор понятия рабочего дня нет, и народ может работать до 24:00, пока работает метро и автобусы.

В случае использования Database differential, в резервной копии сохраняются только те страницы данных, в которых произошли изменения с момента выполнения последней полной резервной копии. Это значит, что полная копия у тебя уже должна быть, иначе с резервированием будут проблемы. Обрати внимание, что сохраняются страницы с измененными данными, а не журнал изменений. Это значит, что если в этой странице было 10 изменений, то в резервную копию попадет только последнее состояние, а все промежуточные хранятся в журнале транзакций.

Никогда не храни данные на том же сервере, где работает реальная база данных. Это бесполезно и не поможет в случае разрушения диска.

Если после последнего полного резервирования было произведено немного изменений, то намного выгоднее сделать дифференцированное. Результат – маленький файл резервной копии и достаточно быстрое выполнение. Если произошла массовая корректировка данных, то затраты на дифференцированную копию будут слишком большими.

Итак, для маленьких баз данных можно не заморачиваться и делать полную резервную копию хоть каждый час. Для этого скрипт резервирования можно поместить в планировщик задач и пока сервер работает можно продолжать читать любимый][. Если база большая, то ее резервирование можно проводить ежедневно по ночам и если есть возможность, то днем в обеденный перерыв. А в рабочее время иногда делать дифференцированное копирование.

Если ты ограничился только созданием полных копий, то я рекомендую завести под их хранение десяток носителей, например DVD-RW дисков. Семь носителей будут хранить неделькой, и хранить резервную копию каждого дня недели. Помимо этого я рекомендую сохранять на отдельных носителях резервные копии каждого последнего рабочего дня недели.

Модель Simple не позволяет восстанавливать данные на определенный период, но благодаря описанным выше рекомендациям, ты сможешь откатиться в прошлое.

Полные возможности

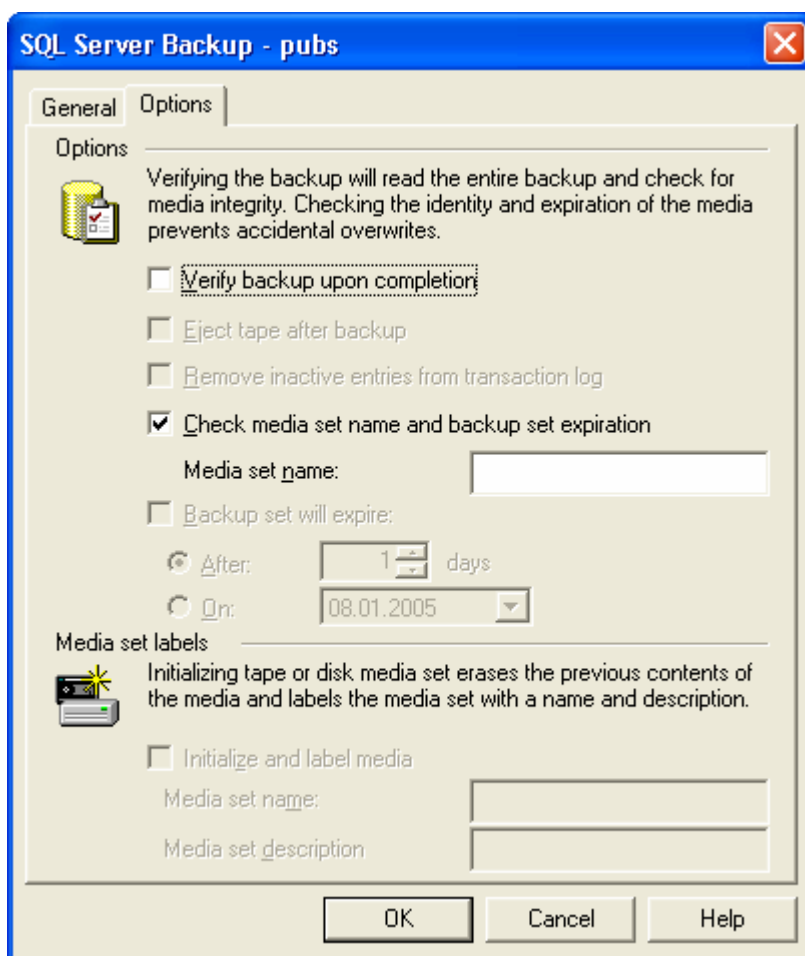
В модели Full и Bulk-Logged сервер позволяет делать еще резервирование журнала транзакций или отдельных файлов/файловых групп базы данных. Это мощная возможность, но нужно быть готовым к тому, что резервные копии будут занимать достаточно много места. Эти модели рекомендуются для крупных баз данных, с наиболее важными данными.

Если база очень большая, то ее резервирование будет происходить долго, даже при использовании дифференцированного режима. Намного быстрее зарезервировать только журнал изменений и по ним сервер сможет восстановить сами данные.

Преимущество модели Full (Bulk-Logged с ограничениями) в том, что данные могут быть восстановлены на любой момент времени. Если кто-то натворил бед, удалил данные или просто сделал что-то нелегальное, то можно восстановить базу на пять минут раньше на тестовом сервере и перенести потерянные данные в рабочий сервер. У меня уже были случаи, когда операторы базы наворачивали данные (например, случайно удаляли), а потом пытались списать это дело на меня. Возможность восстановления на определенный период спасала мой зад от утюга и паяльника, а кошелек от лишения зряплаты.

Возможность резервирования отдельных файлов и файловых групп тоже является достаточно мощной. В одной базе данных может быть множество таблиц и по практике наиболее часто изменяется не более 5 из них. Все остальные служат справочниками и редактируются раз в месяц, а то и вообще раз в год. Ты можешь расположить часто изменяемые таблицы в одну файловую группу, а не изменяемые таблицы в другую.

Теперь файловую группу с изменяемыми таблицами можно резервировать как угодно часто, и не тратить время и не нагружать процессор справочными таблицами.



Во время резервирования можно указать на закладке Options имя копии в поле Media set name. Не зная это имя восстановить данные нельзя, так что это простая защита данных.

Частота резервирования

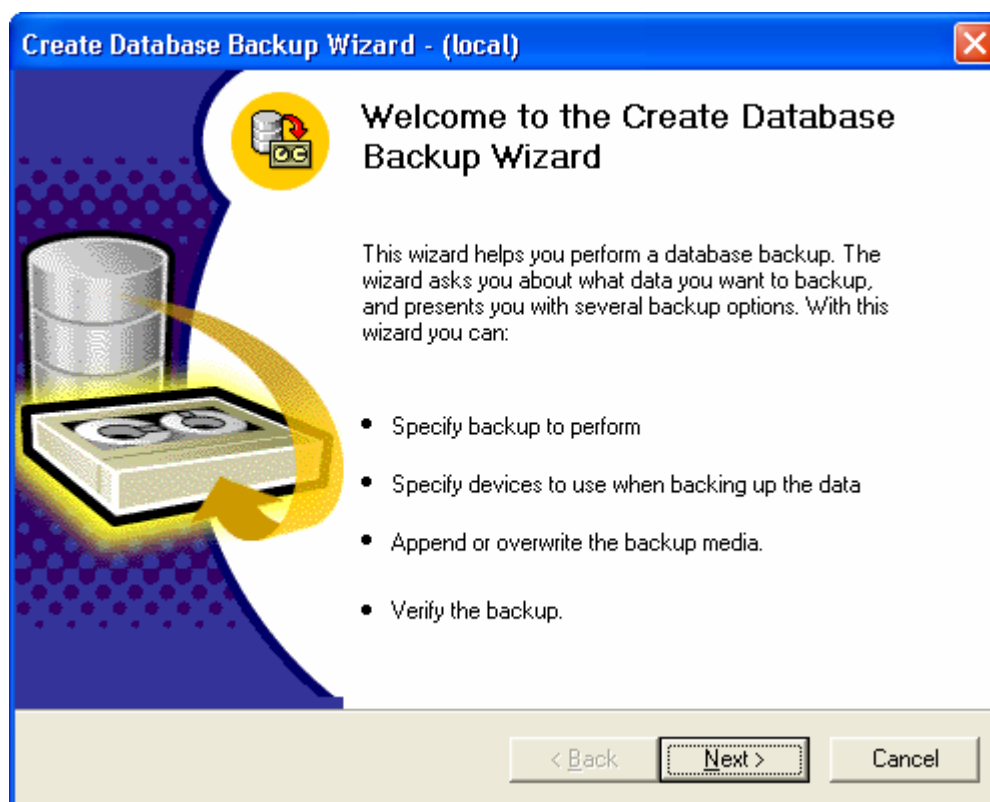
Вечная проблема – как часто нужно делать резервную копию? Все зависит от количества данных, частоты их обновления и важности. Рассмотрим основные варианты сочетания количества, частоты обновления и важности.

Если данные являются справочными, изменяются редко и в случае потери не составляет труда воспроизвести ручное восстановление, то можно делать полную резервную копию раз в день. Если данные изменяются часто, но их немного, то возможна полная даже копия каждый час, потому что малое количество данных резервируется быстро.

Когда база данных достаточно большая, но изменения проходят не часто, то мы уже говорили о том, что полная копия должна делаться в не рабочее время. В рабочие периоды, например, ежечасно, можно делать дифференцированное резервирование или копирование журнала транзакций.

Если данных много и изменения происходят постоянно, то тут уже нужна модель Full или Bulk-Logged. Полную резервную копию делаем в нерабочее время, а в рабочее время (пару раз в день или даже каждые два часа) можно делать резервирование журнала.

Когда в базе данных очень часто происходят массовые загрузки или изменения и данных мало, то используй модель Simple. Если данных много, то лучше юзать Bulk-Logged, а не Full. Это позволит увеличить производительность и уменьшить журнал транзакций.



В MS SQL Server можно резервировать с помощью мастера, простого визуального диалога или SQL командой.

В не зависимости от выбранной частоты резервирования, я всегда делаю внеплановые копии в следующих случаях:

- по возможности полная резервная копия до и после массового обновления данных или загрузки.

- по возможности полная резервная копия после изменения структуры данных или добавления/удаления индексов.

- обязательно полная резервная копия перед праздниками, когда офис уходит на несколько дней в заслуженную попойку.

- перед и после осуществления репликации данных.

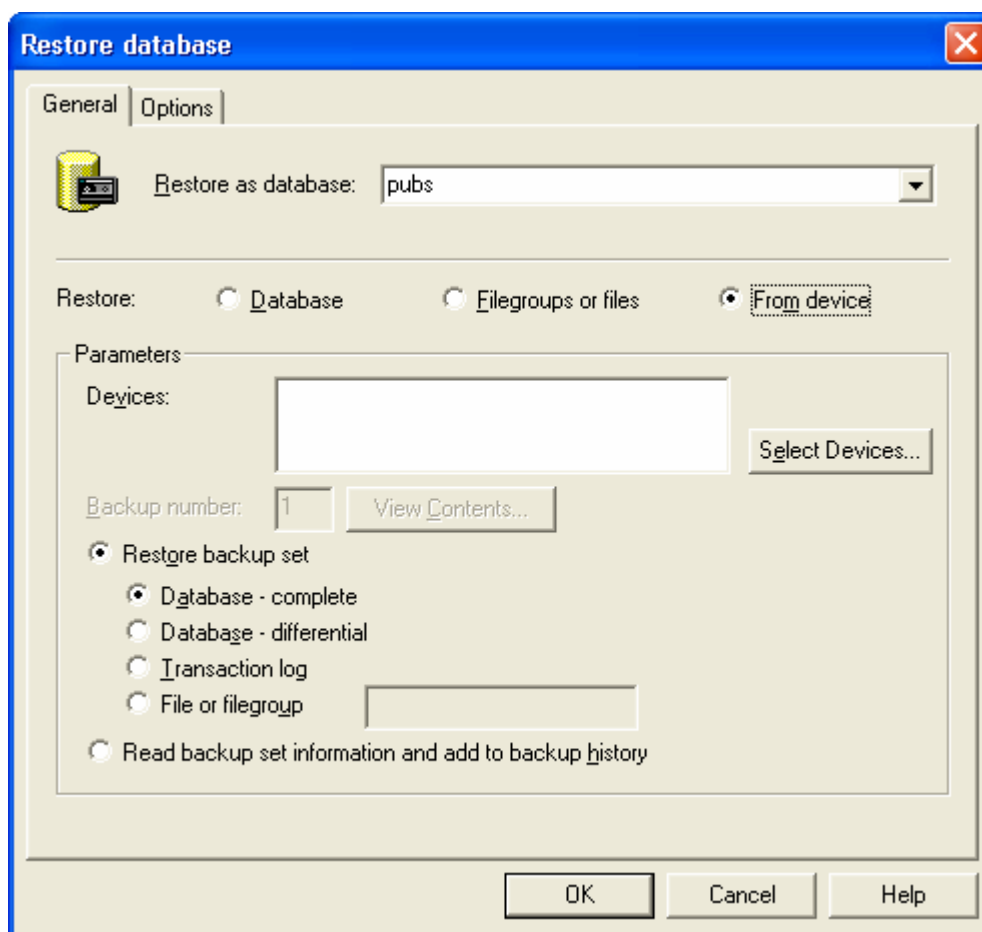
Резервное копирование/
восстановление данных – хороший
способ перенести данные с одного
сервера на другой.

Восстановление данных

Восстановление данных зависит от метода резервирования. Если делалась полная копия, то достаточно просто ее восстановить. Если делалось дифференцированное резервирование, то сначала нужно восстановить последнюю полную копию, а потом надо восстановить дифференцированную.

Для полной модели с использованием резервирования журналов опять же сначала нужно восстановить последнюю полную копию. После этого, необходимо

последовательно восстановить резервные копии журнала транзакций. Необходимо четко следовать последовательности, иначе будут проблемы.



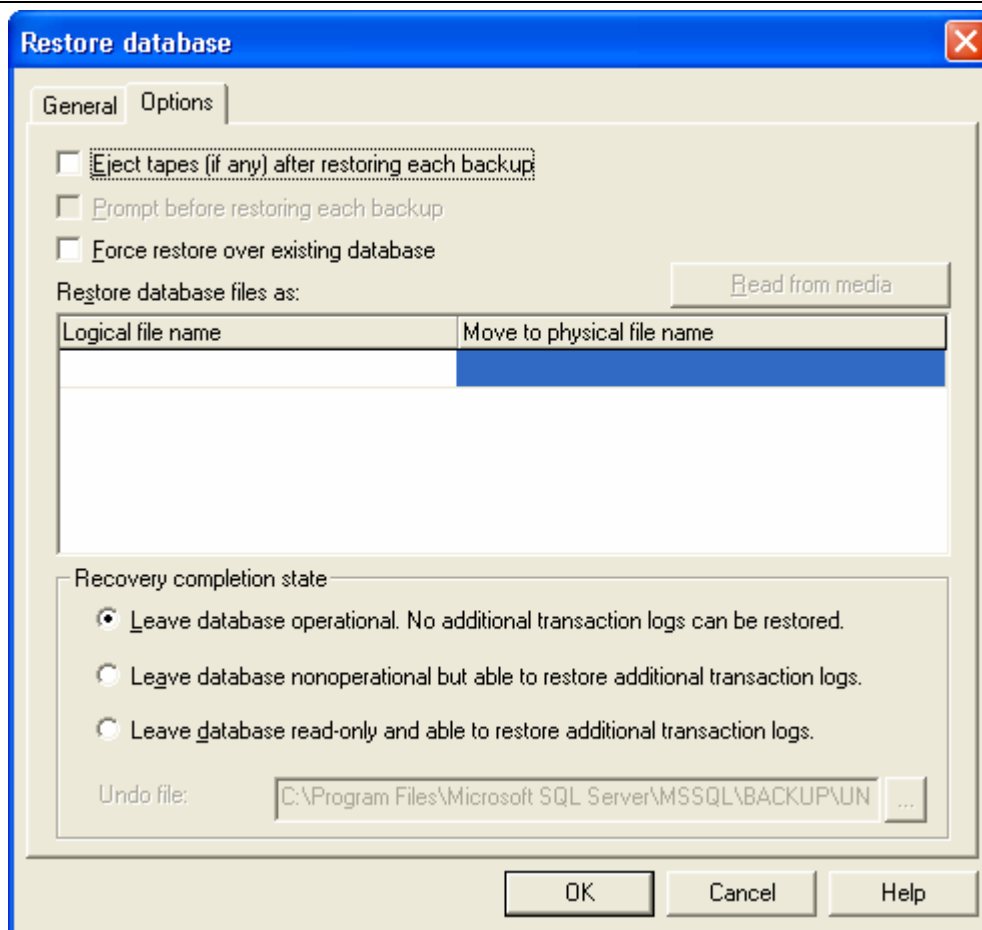
Окно восстановления базы данных.

При восстановлении журнала, на закладке Options нужно оставлять возможность восстановить другие журналы, если они есть. Для этого выбираем переключатель Leave database nonoperational but able to restore additional logs или Leave database read-only and able to restore additional logs. Если восстанавливаешь последний журнал, то выбирай Leave database optional, иначе база будет недоступна для изменений. После восстановления с этой опцией, больше журналов восстанавливать нельзя.

Сбой системы

Все вышеописанные рекомендации относятся только к восстановлению данных в нормальной работе или при сбое и использовании модели Simple. Такое восстановление позволяет вернуть базу данных на момент последнего резервирования. Все изменения, сделанные после этого теряются, и их вернуть не удастся.

В работе администратора самое главное не паниковать. У тебя должна быть холодная голова, а в штанах железные шары. Сразу после катастрофы необходимо правильно оценить обстановку и сделать правильные выводы.



Закладка Options окна восстановления данных.

В случае непредвиденной ситуации и если используется модель Full или Bulk-Logged, можно попытаться восстановить все данные, и вернуть состояние вплоть до момента сбоя. Продвинутые админы держат файлы данных и журнала на разных дисках, а оба сразу накрываются достаточно редко. Если накрылся диск с журналом, то делаем полную копию данных и восстанавливаем на новом винте. А если полетел диск с данными, то прежде чем начать процесс восстановления данных, необходимо попытаться зарезервировать журнал, в котором хранится информация о последних изменениях данных. Только после этого можно приступить к восстановлению работы сервера на новом винте и данные будут восстановлены полностью.

Оригинальность сестра хакера

Недавно увидел отличное решение с точки зрения резервирования. Хотя это не совсем резервное копирование, но решает задачу именно резервирования данных. Итак, в современных базах данных есть возможность репликации данных. Это мощная штука, принцип работы которой очень хорошо увидеть на примере.

Допустим, что у тебя есть два удаленных офиса, в каждом из которых происходит массовая нагрузка на сервер данных. Можно расположить один сервер в главном офисе, а второй будет обращаться к базе через интернет. Но это достаточно большая нагрузка на сеть с точки зрения трафика и скорость доступа не высока. Намного эффективнее расположить два сервера в каждом из офисов, а потом сервера данных будут синхронизироваться. Процесс синхронизации как раз и называется репликацией.

От крушения данных из-за проблем с винчестером спасает использование RAID с зеркалированием. Но от других проблем это не спасет.

А что если в эту систему поставить еще один сервак, который тоже будет участвовать в репликации, но не будет использоваться в работе. Получится реальный сервер, который будет железной резервной копией. В случае проблем, можно в считанные секунды превратить резервный сервер в рабочий, и он будет содержать реально работающие данные. Правда свежесть этих данных зависит от настроек репликации (как восстановление от настроек резервирования). Зато у тебя всегда под рукой будет сервер, пусть даже не

Итого

От потери данных никто не застрахован, а они регулярно теряются и причин для этого много: выход из строя оборудования, воровство техники, стихийные бедствия, пожары, терроризм. Как видишь, причин очень много. Конечно же, вероятность возникновения некоторых из них слишком мала, но она есть. А если сложить все вместе, то общая вероятность потери данных становится достаточно высокой, чтобы задуматься об этом.

Так почему же большинство из нас не задумывается о возможных проблемах, пока сами не встретимся с бедой? Виновником как всегда является простая человеческая лень, с которой надо бороться.

Необходимо делать все, чтобы данные ни в коем случае не были утеряны бесследно и затраты на восстановление были минимальны. Мы постарались показать тебе самое интересное из теории резервирования и восстановления. Практика и нюансы зависят от конкретной СУБД, а это уже нюансы.