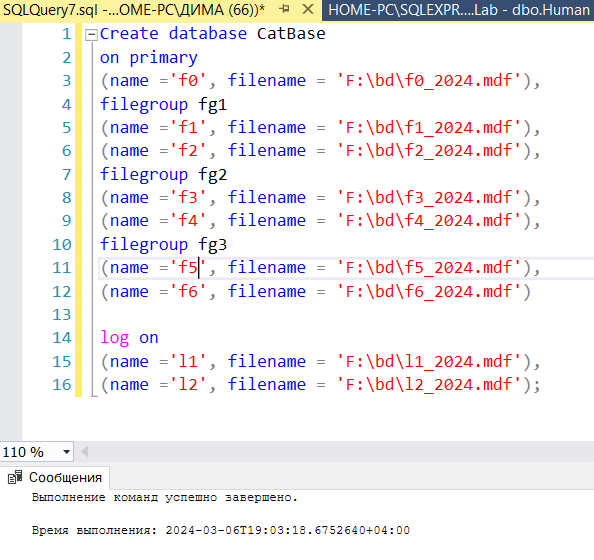
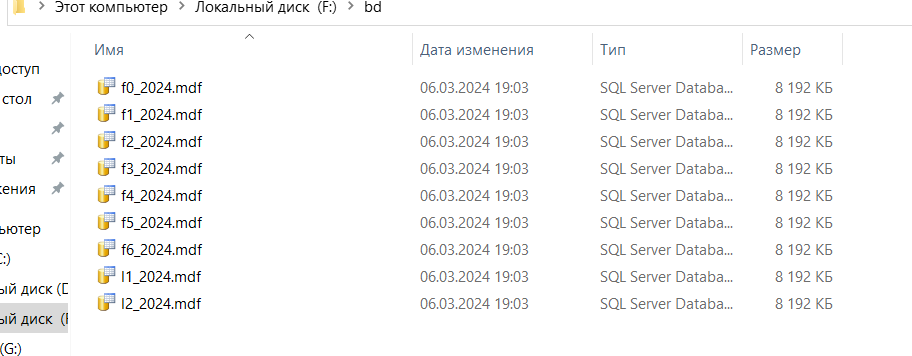
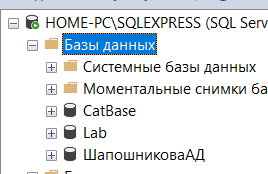
**Лабораторная работа 3. Резервное копирование и восстановление**

**Задание (выполняется посредством ввода команд на TransactSQL):**

1 Создать тестовую базу данных. База данных должна состоять из 6 файлов данных и 2 файлов журнала. Файлы данных должны быть распределены между 3 файловыми группами. Установить полную модель восстановления базы данных;

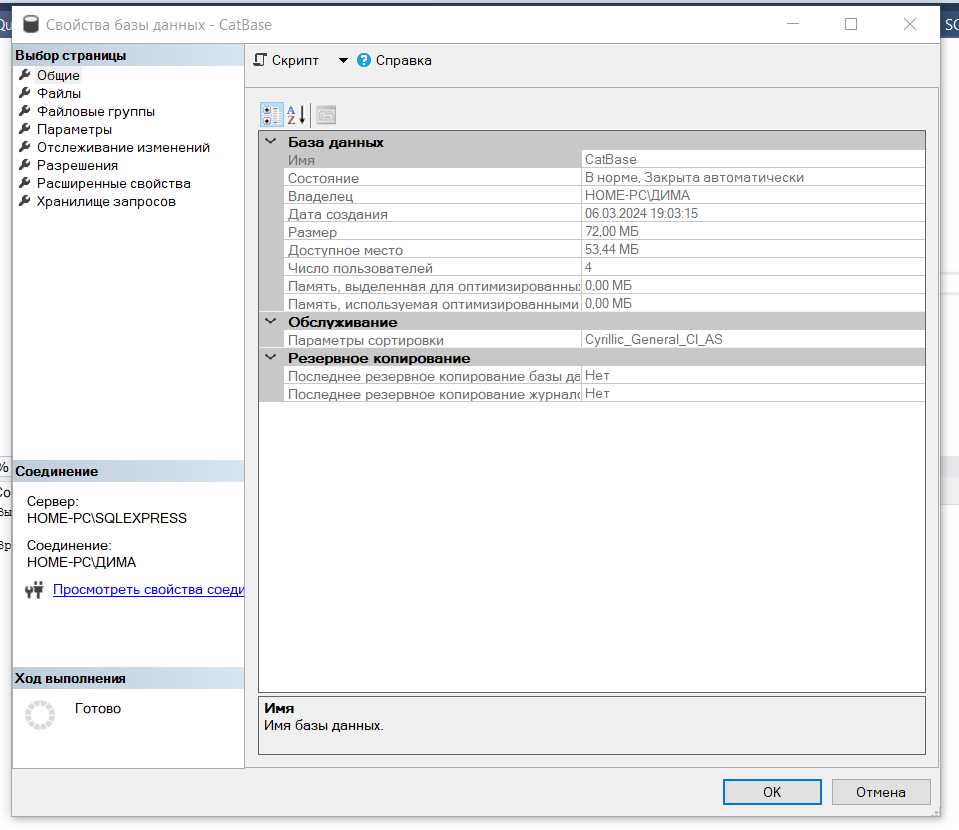




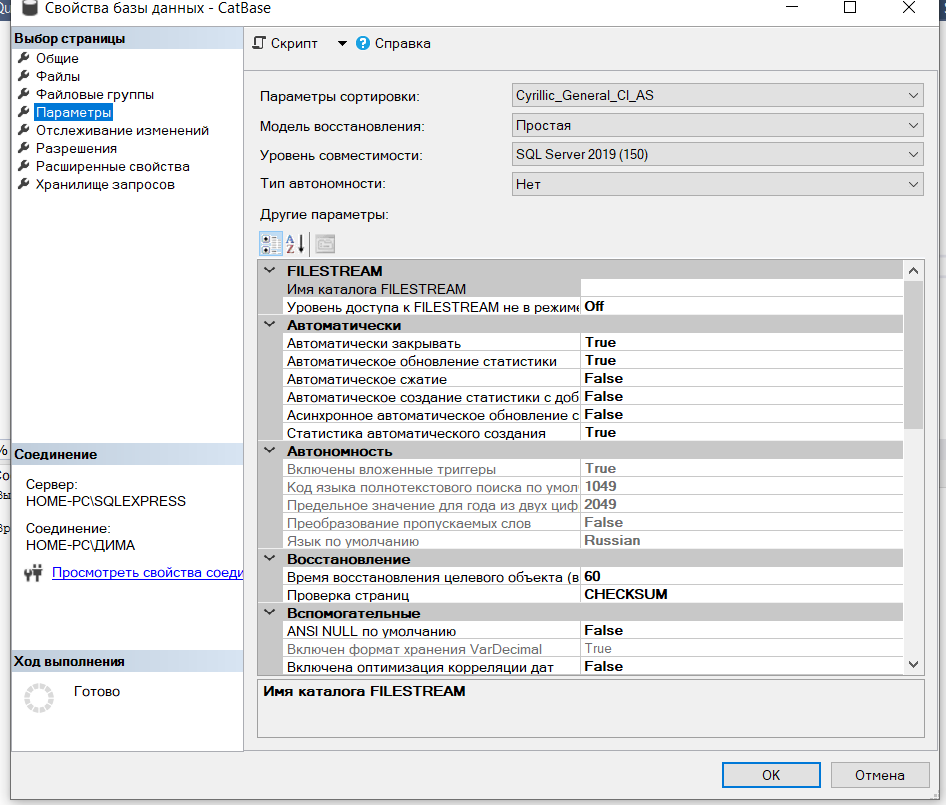


Модель восстановления — это свойство базы данных, которое управляет процессом регистрации транзакций, определяет, требуется ли для журнала транзакций резервное копирование, а также определяет, какие типы операций восстановления доступны.

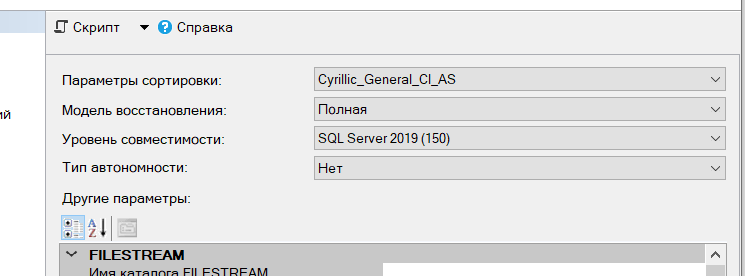
Чтобы установить Модель полного восстановления базы данных, нажимае правой кнопкой мышки по базе и выбираем "Свойства"



В Свойствах выбираем раздел "Параметры"



Находим "Модель восстановления", у нас установлена "Простая"



Меняем на полную.

Либо:

ALTER DATABASE CatBase SET RECOVERY FULL;

2 В базе данных создать тестовую таблицу с минимальным количеством столбцов (2-3). Таблица должна быть размещена в файлах только одной из файловых групп. Организовать циклическое заполнение таблицы 200000 записями с произвольными значениями полей.

create table Cat (

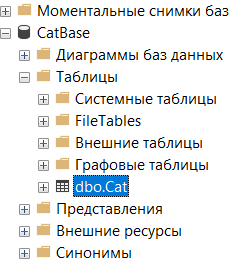
id int identity(1,1) primary key,

CatName varchar(50),

age int

)

on fg1;



declare @count int = 1;

while @count <= 200000

begin

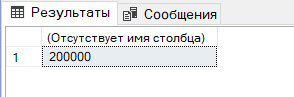
insert into Cat (CatName, age)

values ('Кличка' + CAST(@count as varchar), rand() \* 100);

set @count= @count+1;

end

select count(Id) from Cat



3 Создайте логическое устройство копирования на основе файла на жёстком диске; Название логического устройства и местоположения файла определите самостоятельно;

Процедура sp\_addumpdevice в SQL Server используется для добавления устройства резервного копирования (backup device). Устройство резервного копирования представляет собой логическое имя, на котором будут храниться резервные копии баз данных.

execute sp\_addumpdevice 'disk', 'CatBackup', 'F:\bd\backup.bak';

4 Создайте снимок тестовой базы данных;

Снимок базы данных полезен для восстановления данных к конкретному моменту времени.

create database CatBase\_snapshot

on

(name = 'f0', filename ='F:\bd\f0\_2024.ss'),

(name = 'f1', filename ='F:\bd\f1\_2024.ss'),

(name = 'f2', filename ='F:\bd\f2\_2024.ss'),

(name = 'f3', filename ='F:\bd\f3\_2024.ss'),

(name = 'f4', filename ='F:\bd\f4\_2024.ss'),

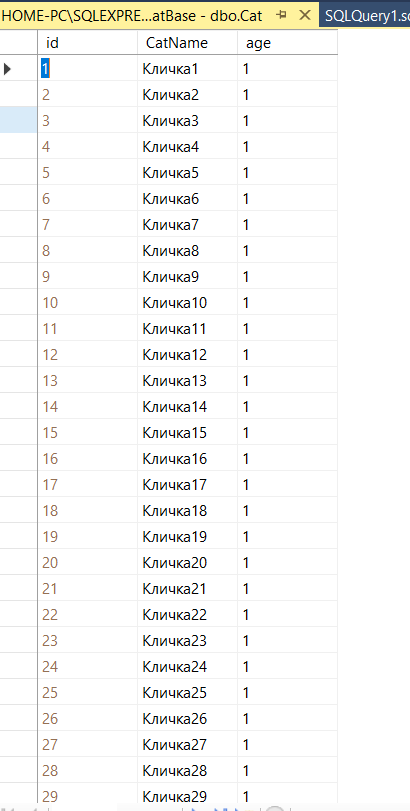
(name = 'f5', filename ='F:\bd\f5\_2024.ss'),

(name = 'f6', filename ='F:\bd\f6\_2024.ss')

as snapshot of CatBase;

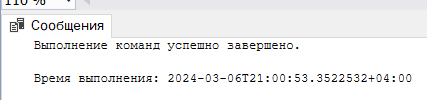
5 Смоделируйте порчу данных в тестовой таблице и восстановите БД из снимка; Убедитесь в успешности восстановления БД;

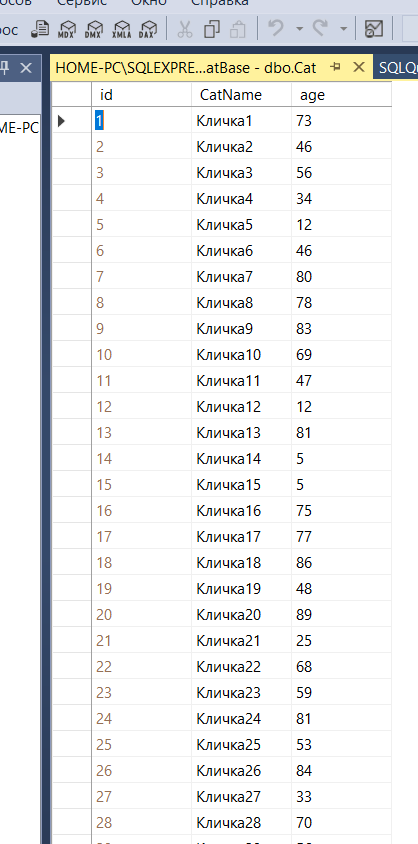
update Cat set age= 1



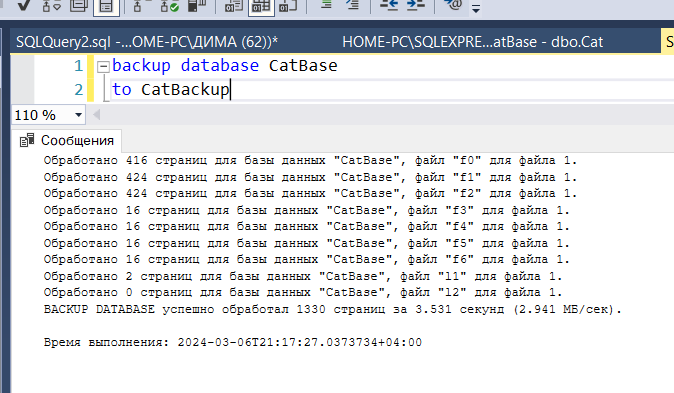
Восстанавливаем данные.

restore database CatBase from database\_snapshot='CatBase\_snapshot'





6 Создайте полную резервную копию тестовой БД в созданное логическое устройство копирования;

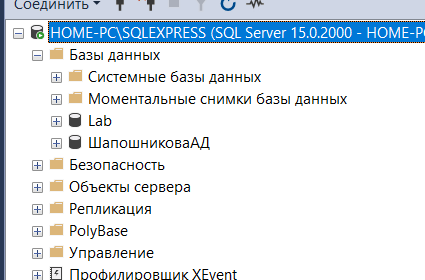


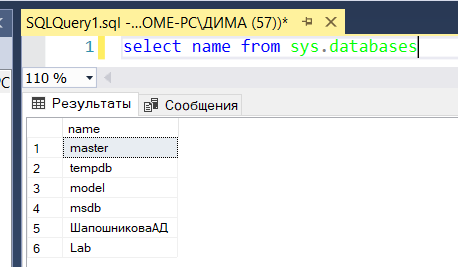
7 Смоделируйте потерю тестовой БД и запустите процесс восстановления из созданной полной резервной копии; Убедитесь в успешности восстановления БД;

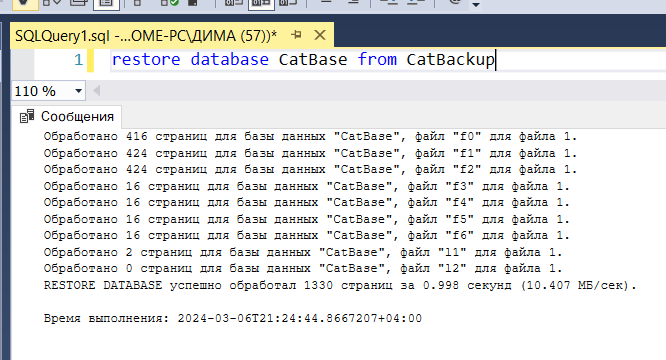
Сначала удалим снимок базы, затем саму базу.

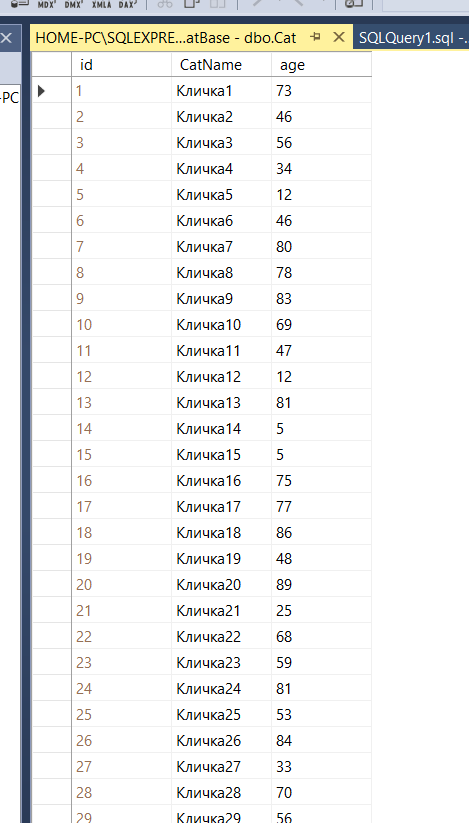
drop database Catbase\_snapshot

drop database CatBase









8 Внесите изменения в БД, добавив и изменив несколько строк (100-200). Создайте разностную резервную копию БД в созданном ранее логическом устройстве;

declare @i int = 0;

while @i <=200

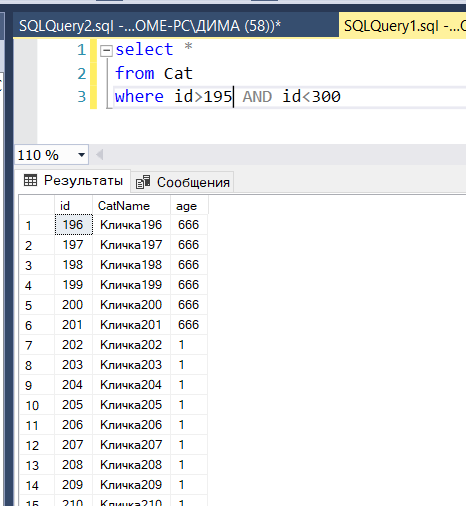
begin

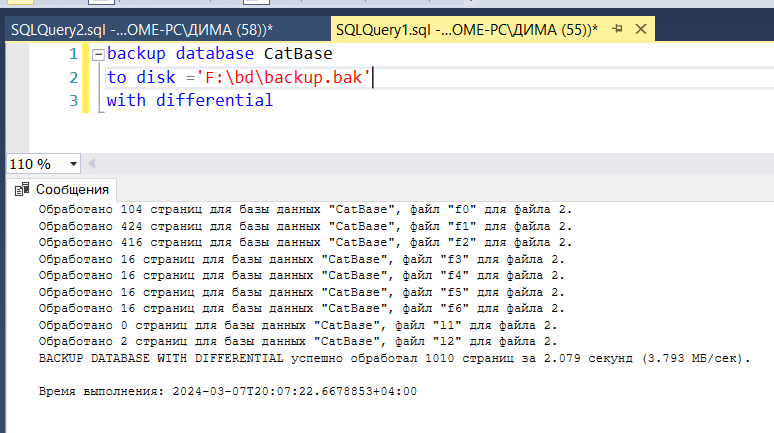
update Cat set age =666 where Id =@i+1;

set @i=@i+1;

end

Проверим, что код выполнился правильно и 200 строк были изменены.





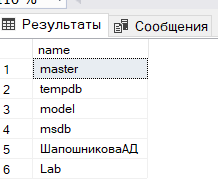
DIFFERENTIAL (разностная)

9 Смоделируйте потерю тестовой БД и запустите процесс восстановления из созданной полной резервной копии и разностной копии; Убедитесь в успешности восстановления БД;

Теряем базу котов:

drop database CatBase

Select name from sys.databases



Полная резервная копия содержит все данные и структуру БД на момент создания копии. Это означает, что при восстановлении БД из полной резервной копии мы получаем точную копию БД, включая все таблицы, записи и связи между ними.

- Разностная копия содержит только изменения, которые произошли после создания полной резервной копии. Это означает, что при восстановлении БД из разностной копии мы сначала восстанавливаем полную резервную копию, а затем применяем изменения из разностной копии, чтобы получить актуальное состояние БД.

restore database CatBase

from CatBackup

with norecovery

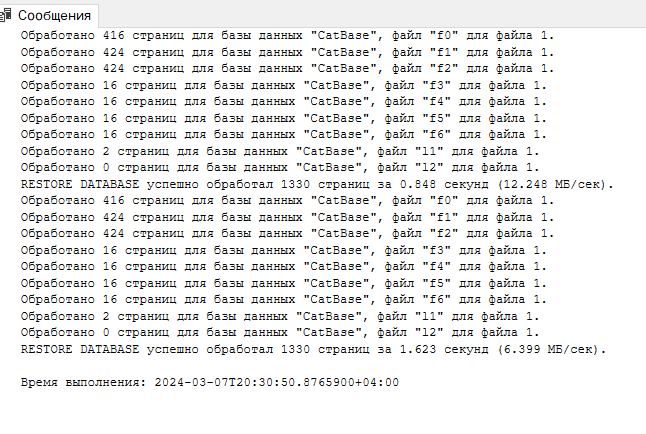
Схема RESTORE WITH NORECOVERY оставляет базу данных в нерабочем состоянии и не выполняет откат незафиксированных транзакций. Можно восстановить дополнительные журналы транзакций. Базу данных нельзя использовать, пока она не будет восстановлена.

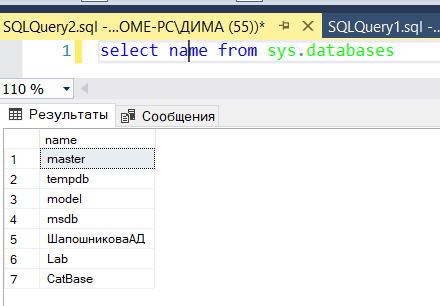
restore database CatBase

from CatBackup

with recovery

По умолчанию используется схема RESTORE WITH RECOVERY, которая выполняет откат незафиксированных транзакций и после завершения оставляет базу данных работоспособной. Дополнительные журналы транзакций не могут быть восстановлены.



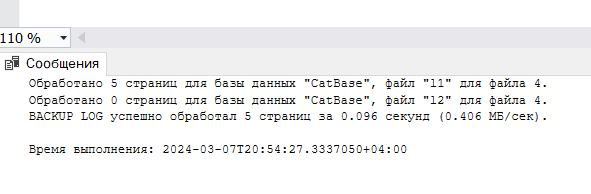


10 Убедитесь, что транзакции все зафиксированы. Создайте 2 резервных копий журнала транзакций тестовой БД на созданное ранее логическое устройство. Между двумя процессами резервного копирования выполните и зафиксируйте несколько простых транзакций, причём первую из них снабдите пометкой (with mark).

Создаем первую копию журнала транзакций.

backup log CatBase

to CatBackup



Выполним две транзакции, первая с пометкой 'one'

begin transaction One

with mark 'one'

update Cat set age=999 where id=1

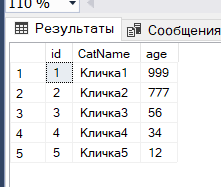
commit transaction

begin transaction

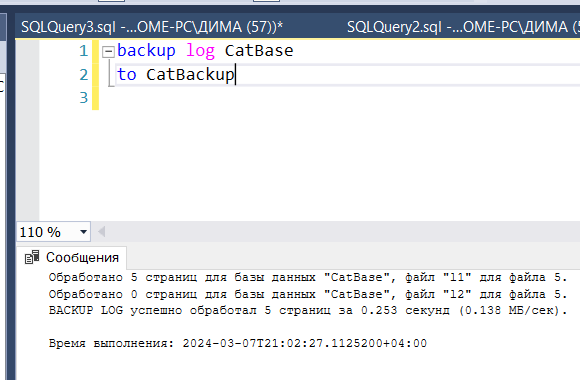
update Cat set age=777 where id =2

commit transaction

select top(5)\* from Cat

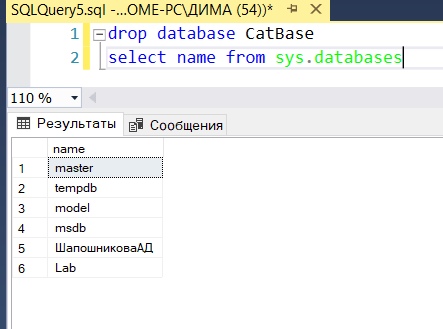


Вторая копия журнала транзакций.



11 Смоделируйте потерю тестовой БД и запустите процесс восстановления данных на момент сбоя. Убедитесь в успешности восстановления БД;

Потеряем кото-базу:



Восстановим ее:

код восстановления будет выглядеть так:

Восстановим базу данных из ее полной и разностной резервной копии

restore database CatBase

from CatBackup

with norecovery

restore database CatBase

from CatBackup

with norecovery

Восстановим журналы транзакций

restore log CatBase

from CatBackup

with norecovery

restore log CatBase

from CatBackup

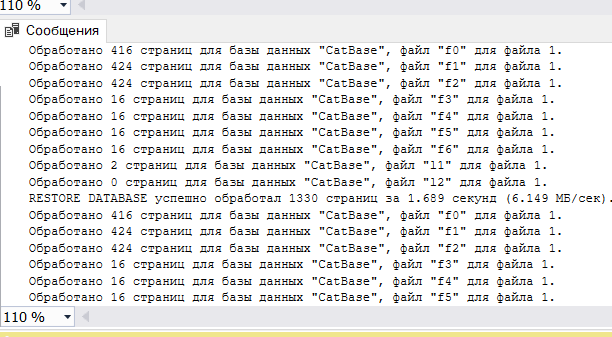
with norecovery

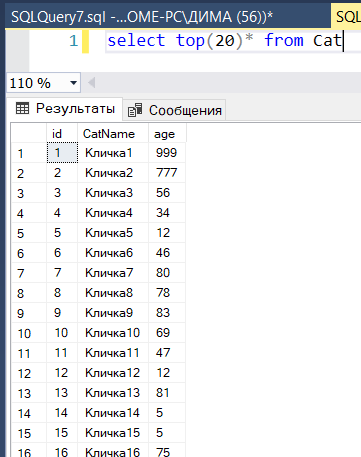
Наконец восстановим базу данных.

restore database CatBase

from CatBackup

with recovery





12 Смоделируйте потерю тестовой БД и запустите процесс восстановления данных на момент выполнения транзакции, снабжённой меткой. Убедитесь в успешности восстановления БД;

drop database CatBase

Восстанавливаем.

restore database CatBase

from CatBackup

with norecovery

restore database CatBase

from CatBackup

with norecovery

restore log CatBase

from CatBackup

with norecovery

restore log CatBase

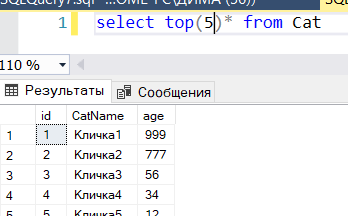
from CatBackup

with norecovery,stopatmark='one'

restore database CatBase

from CatBackup

with recovery



**Вопросы к отчёту:**

* Способы резервного копирования. Полная резервная копия;

Полное резервное копирование - это резервное копирование всей базы данных. Выполняется резервное копирование всех групп файлов и файлов данных, которые являются частью этой базы данных.

В зависимости от размера баз данных этот процесс может занимать очень много времени, и если для вас важен вопрос времени, то вы можете предусмотреть разностное резервное копирование.

После запуска копирования нельзя приостановить его – оно продолжает выполняться, пока не будет получена резервная копия всей базы данных.

* Способы резервного копирования. Разностная резервная копия;

Разностное резервное копирование позволяет выполнять резервное копирование только той информации, которая изменилась с момента последнего резервного копирования.

Поскольку создается резервная копия только части данных, это происходит быстрее и занимает меньше места, чем полная резервная копия.

Недостатком является то, что восстановление с разностных копий происходит сложнее и занимает больше времени, чем восстановление с полной резервной копии. Для восстановления с разностной копии требуется восстановление полной резервной копии и всех разностных копий, созданных с момента последнего полного резервного копирования.

* Способы резервного копирования. Резервная копия файла или файловой группы;

Резервное копирование файла данных позволяет вам получать резервную копию отдельного файла из группы файлов.

Резервное копирование файла данных может оказаться полезным, если у вас недостаточно времени для ежесуточного резервного копирования всей группы файлов.

Резервное копирование группы файлов предусматривает резервное копирование всех файлов данных, связанных с отдельным файлом в базе данных.

Можно использовать этот тип резервного копирования для группы файлов, связанной с определенным отделом или рабочей группой.

* Способы резервного копирования. Резервная копия файла журнала транзакций;

Резервное копирование журнала транзакций позволяет получать резервные копии журнала транзакций.

При запуске SQL Server после отказа системы журнал транзакций используется для повторного исполнения транзакций, которые были фиксированы, но не записаны на диск, и отката транзакций, которые не были фиксированы на момент аварии системы. Такой подход гарантирует точность данных.

Поскольку изменения, которые вносятся в базу данных, не сразу записываются на диск, журнал транзакций является единственным средством, с помощью которого можно воспроизвести транзакции в случае отказа системы.

* Модели восстановления БД: полная;

В полной модели восстановления все изменения данных сохраняются в журнале транзакций.

В этой модели выполняются полные резервные копии базы данных и регулярные резервные копии журналов транзакций.

Процесс восстановления данных в полной модели включает следующие шаги:

1. Восстановление последней полной резервной копии базы данных.

2. Применение последовательности резервных копий журналов транзакций, чтобы восстановить базу данных до нужного момента времени.

3. Восстановление базы данных до состояния, соответствующего выбранной точке восстановления.

* Модели восстановления БД: с неполным протоколированием;

Основные особенности модели восстановления с неполным протоколированием:

1. Журнал транзакций не используется для сохранения изменений данных.

2. Выполняются только резервные копии базы данных, без резервных копий журналов транзакций.

3. Восстановление базы данных осуществляется путем восстановления последней полной резервной копии.

Модель восстановления с неполным протоколированием улучшает производительность операций с загрузкой больших объемов данных посредством сокращения количества журнализируемой информации.

* Модели восстановления БД: простая;

В данной модели журнал транзакций используется для записи только минимальной информации, необходимой для обеспечения целостности базы данных.

Основные особенности модели восстановления с простым восстановлением:

1. Журнал транзакций используется только для сохранения информации о начале и завершении каждой транзакции.

2. Точка восстановления не может быть установлена между резервными копиями базы данных.

3. Точка восстановления находится на момент окончания последней резервной копии базы данных.

.

* Общие сведения об операциях, допускающих неполное протоколирование;

Можно выполнять определенные операции, которые не записываются в журнал транзакций. Такие операции называются непротоколируемыми. Выполняя массовую операцию, как непротоколируемую операцию, повышается производительность этой операции.

• SELECT INTO

• BULK COPY и программа массового копирования (BCP)

• CREATE INDEX

• Определенные текстовые операции

Чтобы активизировать непротоколируемые массовые операции данных, надо задать для базы режим воспроизведения BULK\_LOGGED.

SELECT INTO

Оператор SELECT INTO используется для создания новой таблицы в базе данных. Его можно использовать только для создания данных, но не для обновления данных. Этот процесс создания можно легко повторить; поэтому операции SELECT INTO вполне подходят для выполнения как непротоколируемые операции.

BULK COPY и программа BCP

Чтобы операции BULK COPY и BCP можно было выполнять как непротоколируемые операции, они должны отве-чать следующим требованиям.

• Для параметра базы данных select into/bulkcopy должно быть задано значение TRUE.

• Целевая таблица не может иметь никаких индексов

• Целевая таблица не может реплицироваться, поскольку для транзакционной репликации используются записи журнала транзакций.

• Чтобы задать блокировку на уровне таблиц, должна быть указана подсказка TABLOCK.

Эти ограничения позволяют выполнять операции массового копирования с большей скоростью при экономии места в журнале транзакций. Однако при необходимости восстановления базы данных из резервной копии эти непротоколируемые операции придется повторить.

CREATE INDEX

Операция CREATE INDEX в базах данных используется для создания индексов на столбцах таблицы, что позволяет ускорить выполнение запросов, особенно при поиске и сортировке данных

Текстовые операции

К текстовым операциям, которые можно выполнять как непротоколируемые операции, относят-ся WRITETEXT и UPDATETEXT.

\*\*WRITETEXT\*\*: Команда WRITETEXT используется для записи данных в столбец типа text, ntext или image. С помощью этой команды можно добавить новые данные в существующее значение столбца или перезаписать его полностью.

\*UPDATETEXT\*\*: Команда UPDATETEXT используется для обновления данных в столбце типа text, ntext или image. Она позволяет изменять часть существующих данных в столбце без полной перезаписи.

* Восстановление БД из снимка;

Восстановление базы данных (БД) из снимка (snapshot) - это процесс восстановления данных БД до определенного момента времени, когда был создан снимок.

Важно отметить, что при восстановлении БД из снимка все изменения данных после создания снимка будут потеряны.

* Восстановление БД из полной копии и разностной;

Полная резервная копия содержит все данные и структуру БД на момент создания копии. Это означает, что при восстановлении БД из полной резервной копии мы получаем точную копию БД, включая все таблицы, записи и связи между ними. Но какие либо изменения, произошедшие после создания копии не сохранятся

В зависимости от размера баз данных этот процесс может занимать очень много времени.

После запуска копирования нельзя приостановить его – оно продолжает выполняться, пока не будет получена резервная копия всей базы данных.

- Разностная копия содержит только изменения, которые произошли после создания полной резервной копии. Это означает, что при восстановлении БД из разностной копии мы сначала восстанавливаем полную резервную копию, а затем применяем изменения из разностной копии, чтобы получить актуальное состояние БД.

Схема RESTORE WITH NORECOVERY оставляет базу данных в нерабочем состоянии и не выполняет откат незафиксированных транзакций. Можно восстановить дополнительные журналы транзакций. Базу данных нельзя использовать, пока она не будет восстановлена.

По умолчанию используется схема RESTORE WITH RECOVERY, которая выполняет откат незафиксированных транзакций и после завершения оставляет базу данных работоспособной. Дополнительные журналы транзакций не могут быть восстановлены.

Процесс восстановления из полной и разностных копий позволяет уменьшить объем хранимых данных и сэкономить место на диске,

* Восстановление БД на определённый момент времени;

Восстановление базы данных на определенный момент времени - это процесс восстановления данных базы данных до конкретного момента времени, когда была создана резервная копия. Этот процесс позволяет вернуть базу данных к состоянию, которое она имела на момент создания резервной копии, игнорируя все изменения, произошедшие после этого момента.

Восстановление на момент времени всегда производится из резервной копии журналов. В каждой инструкции RESTORE LOG последовательности восстановления необходимо указать целевое время или транзакцию в идентичном предложении STOPAT. В качестве предварительного условия для восстановления на момент времени необходимо сначала восстановить полную резервную копию базы данных, чья конечная точка предшествует моменту времени восстановления