

Частное учреждение образования
«Колледж бизнеса и права»

УТВЕРЖДАЮ
Заведующий
методическим кабинетом
_____ Е.В. Фалей
«__» _____ 2017

Специальность: «Программное обеспечение информационных технологий»	Дисциплина: «Базы данных и системы управления базами данных»
Составлена на основании учебной программы, утвержденной директором Колледжа бизнеса и права 30.12.2016	

Лабораторная работа № 15
Инструкционно-технологическая карта

Тема: Создание индексов. Резервное копирование и восстановление БД

Цель работы:

- научиться работать с индексами: создание индекса (CREATE INDEX), удаление индекса (DROP INDEX), изменение индекса (ALTER INDEX), переименование индекса;
- научиться получать различные резервные копии базы данных с помощью инструкций языка T-SQL, а также с помощью визуальной среды;
- научиться восстанавливать базу данных с различных резервных копий базы данных с помощью инструкций языка T-SQL, а также с помощью визуальной среды.

Время выполнения: 2 часа

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Изучить раздел лекционного курса по теме «Описание данных на основе SQL.
2. Получить у преподавателя индивидуальное задание и выполнить лабораторную работу в соответствии с вариантом задания согласно описанной в разделе «Пример выполнения работы» методике настоящей инструкционно-технологической карты.
3. Ответить на контрольные вопросы.

1. ПРИМЕР ВЫПОЛНЕНИЯ РАБОТЫ

1. Создание индексов

Индекс для таблицы создается с помощью инструкции CREATE INDEX. Эта инструкция имеет следующий синтаксис:

CREATE [UNIQUE] [CLUSTERED |NONCLUSTERED] INDEX index_name

ON table_name (column1 [ASC DESC],...)

[INCLUDE (column_name [,...])]

[WITH

[FILLFACTOR=n]

[[,] PAD_INDEX = {ON | OFF}]

[[,] DROP_EXISTING = {ON | OFF}]

[[,] SORT_IN_TEMPDB = {ON | OFF}]

[[,] IGNORE_DUP_KEY = {ON | OFF}]

[[,] ALLOW_ROW_LOCKS = {ON | OFF}]

[[,] ALLOW_PAGE_LOCKS = {ON | OFF}],]

[[,] STATISTICS_NORECOMPUTE = {ON | OFF}]

[[,] ONLINE = {ON | OFF}]]

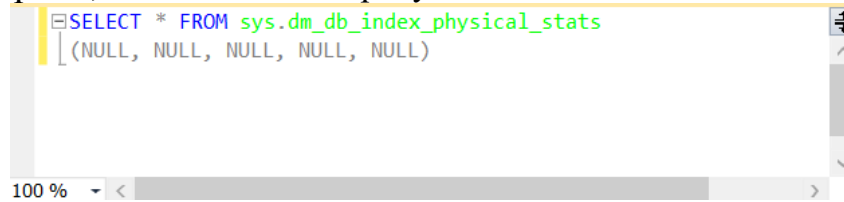
[ON file_group | «default»]

Параметр	Описание
index_name	задает имя создаваемого индекса
table_name	имя таблицы, в которой расположены индексируемые столбцы
column1	столбец (несколько столбцов), для которых создается индекс
UNIQUE	указывает, что проиндексированный столбец может содержать только однозначные (т. е. неповторяющиеся) значения. Если ключевое слово UNIQUE не указывается, то повторяющиеся значения в проиндексированном столбце (столбцах) разрешаются
CLUSTERED NONCLUSTERED	задает соответственно кластеризованный или некластеризованный индекс
ASC DESC	указывает, что индекс создается с возрастающим убывающим порядком значений столбца
FILLFACTOR=n	задает заполнение в процентах каждой страницы индекса во время его создания. Его значение можно устанавливать в диапазоне от 1 до 100. При значении n=10 каждая страница индекса заполняется на 100%, т.е. существующая страница листа не будет иметь свободного места для вставки новых строк. При значениях между 1 и 99 страницы листьев создаваемой структуры индекса будут содержать свободное место

В SQL Server поддерживается создание индексов для представлений. Такие представления называются индексируемыми представлениями.

Индекс может быть простым или составным. Простой индекс создается по одному столбцу, а составной индекс — по нескольким столбцам. Для составного индекса существуют определенные ограничения, связанные с его размером и количеством столбцов. Индекс может иметь максимум 900 байтов и не более 16 столбцов.

Чтобы получить информацию обо всех индексах, имеющихся в базу данных можно использовать функцию `sys.dm_db_index_physical_stats ()`. Эта функция имеет пять параметров и все они имеют значения по умолчанию. Можно выставить все параметры по умолчанию и не накладывать фильтр на столбцы и строки, как показано на рисунке ниже.

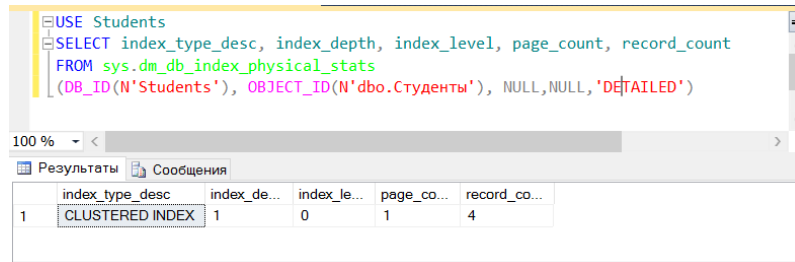


В результате функция возвращает 20 столбцов с данными для (почти) каждого уровня для каждого индекса по каждой таблице каждого раздела в каждой базе данных текущего экземпляра SQL Server.

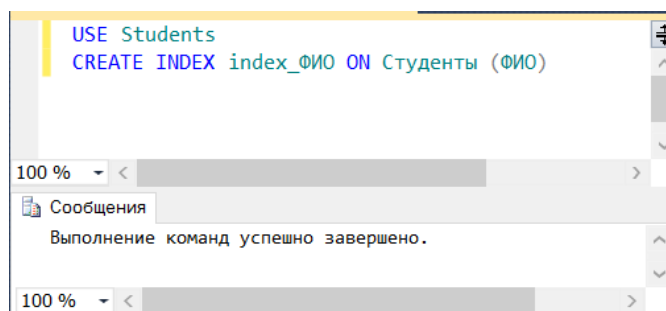
Ниже в таблице приведено описание пяти параметров для данной функции.

Номер параметра	Описание параметра	Пример использования параметра
1	идентификатор ID базы данных, который должен быть указан как число. Если указать NULL, что является значением по умолчанию, функция вернет информацию обо всех базах данных. Если ID не определено, SQL Server будет игнорировать следующие три параметра и предполагает, что они все принимают значения NULL	чтобы упростить получение ID базы данных можно задействовать функцию <code>db_id ()</code> . Например, <code>DB_ID(N'Students')</code>
2	это <code>object_id</code> , который также должен быть задан числом, а не именем. Если выставить NULL, то функция предоставит информацию обо всех объектах, и SQL Server назначит следующие два параметра, <code>index_id</code> и <code>partition_id</code> , также по умолчанию, что означает включение всех индексов и всех разделов	чтобы упростить получение ID объекта можно использовать функцию <code>object_id ()</code> , например, <code>OBJECT_ID(N'dbo.Студенты')</code>
3	позволяет устанавливать ID индекса для указанной таблицы, и значение по умолчанию NULL говорит о том, что пользователь получит на выходе все индексы.	—
4	указывает на номер раздела, и NULL означает, что нужна информация по всем разделам	—
5	это единственный параметр, для которого заданное по умолчанию значение NULL не предполагает возвращения максимума информации. Последний параметр указывает режим выборки, который SQL Server будет использовать при получении данных. Принимаемые параметры следующие: <code>DEFAULT</code> , <code>NULL</code> , <code>LIMITED</code> , <code>SAMPLED</code> или <code>DETAILED</code> . Установка значения NULL по умолчанию соответствует режиму <code>LIMITED</code>	—

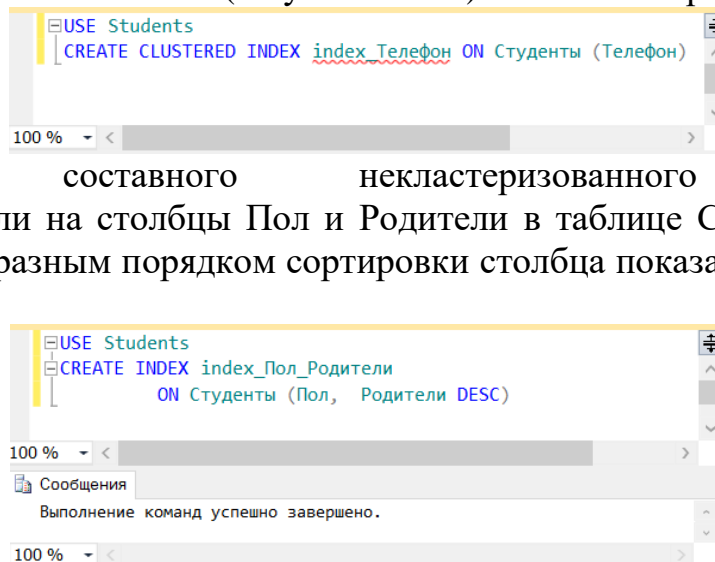
Для того, чтобы получить всю информацию об индексах в таблице Студенты базы данных Students необходимо выполнить код, представленный на рисунке ниже.



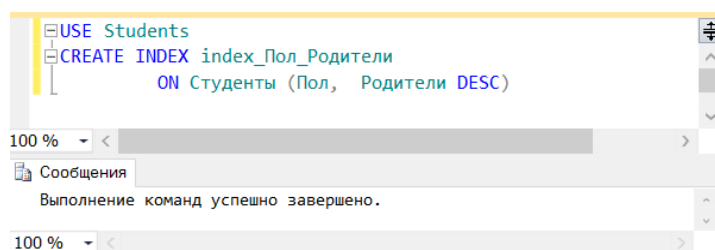
Создание простого некластеризованного (по умолчанию) индекса index_ФИО на столбец ФИО в таблице Студенты базы данных Students с порядком сортировки столбца по возрастанию ASC (по умолчанию) показано на рисунке ниже:



Создание простого кластеризованного индекса index_Телефон на столбец Телефон в таблице Студенты базы данных Students с порядком сортировки столбца по возрастанию ASC (по умолчанию) показано на рисунке ниже:



Создание составного некластеризованного индекса index_Пол_Родители на столбцы Пол и Родители в таблице Студенты базы данных Students с разным порядком сортировки столбца показано на рисунке ниже:



2. Изменение индексов

SQL Server является одной из немногих систем баз данных, которые поддерживают инструкцию ALTER INDEX. Эту инструкцию можно использовать для выполнения операций по обслуживанию индекса. Синтаксис инструкции ALTER INDEX очень сходен с синтаксисом инструкции CREATE INDEX.

Инструкция ALTER INDEX поддерживает три другие параметра:

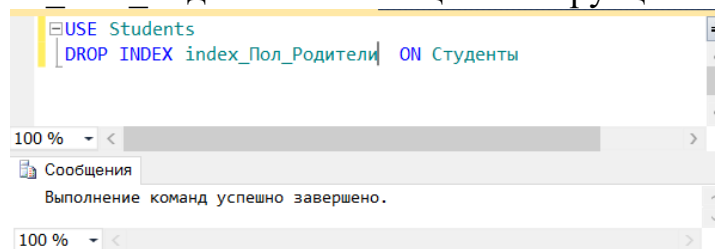
- ✓ параметр REBUILD, используемый для пересоздания индекса;

- ✓ параметр REORGANIZE, используемый для реорганизации страниц листьев индекса;
- ✓ параметр DISABLE, используемый для отключения индекса. Отключенный индекс недоступен для применения, пока он не будет снова включен. Кроме того, отключенный индекс не изменяется при внесении изменений в соответствующие данные. По этой причине, чтобы снова использовать отключенный индекс, его нужно полностью создать вновь. Для включения отключенного индекса применяется параметр REBUILD инструкции ALTER TABLE.

3. Удаление и переименование индексов

Для удаления индексов в текущей базе данных применяется инструкция DROP INDEX. Удаление кластеризованного индекса таблицы может быть очень ресурсоемкой операцией, т. к. потребуется пересоздать все некластеризованные индексы в связи с тем, что все некластеризованные индексы используют ключ индекса кластеризованного индекса, как указатель в своих страницах листьев.

В следующем примере показано удаление некластеризованного составного индекса index_Пол_Родители с помощью инструкции DROP INDEX.



Инструкция DROP INDEX имеет дополнительный параметр, MOVE TO, значение которого аналогично параметру ON инструкции CREATE INDEX. Иными словами, с помощью этого параметра можно указать, куда переместить строки данных, находящиеся в страницах листьев кластеризованного индекса. Данные перемещаются в новое место в виде кучи.

Индексы можно переименовывать с помощью системной процедуры sp_rename.

Замечание. Инструкцию DROP INDEX нельзя использовать для удаления индексов, которые создаются неявно системой для ограничений целостности, таких индексов, как PRIMARY KEY и UNIQUE. Чтобы удалить такие индексы, нужно удалить соответствующее ограничение.

4. Рекомендации по созданию и использованию индексов

Каждый индекс занимает определенный объем дискового пространства, следовательно, существует вероятность того, что общее количество страниц индекса базы данных может превысить количество страниц данных в базе. В отличие от получения выгоды при использовании индекса для выборки данных, вставка и удаление данных такой выгоды не предоставляют по причине необходимости обслуживания индекса. Чем больше индексов имеет таблица, тем больший требуется объем работы по их реорганизации. Общим правилом будет разумно выбирать индексы для частых запросов, а затем оценивать их использование.

Резервное копирование с помощью инструкций Transact-SQL

Все типы резервного копирования можно выполнять, используя только две инструкции языка Transact-SQL:

- ✓ BACKUP DATABASE;
- ✓ BACKUP LOG.

1. Типы устройств резервного копирования

Компонент Database Engine позволяет выполнять резервное копирование баз данных, журналов транзакции и файлов на следующие устройства резервного копирования:

- ✓ жесткие диски;
- ✓ накопители на магнитной ленте.

Жесткий диск является наиболее широко применяемым носителем для хранения резервных копий. Дисковые устройства для резервного копирования могут быть как локальными, установленными на компьютере сервера базы данных, так и удаленными, установленными на файловом сервере. Компонент Database Engine позволяет добавлять новую резервную копию в файл, который уже содержит резервные копии этой же или других баз данных. При добавлении нового резервного набора на носитель, уже содержащий резервную копию, ранее записанное содержимое носителя не затрагивается, а новая резервная копия добавляется на него после последней резервной копии. По умолчанию компонент Database Engine всегда добавляет новые резервные копии к уже существующему файлу резервных копий.

ВНИМАНИЕ! Никогда не сохраняйте файл резервной копии на тот же физический диск, на котором хранится база данных или ее журнал транзакций. В случае повреждения диска с базой данных будут повреждены не только файлы базы данных, но и файл резервной копии, вследствие чего восстановление базы данных будет невозможным.

Накопители на магнитной ленте обычно применяются для резервного копирования таким же образом, что и жесткие диски. Но для сохранения резервной копии на накопитель на магнитной ленте, он должен быть подключен к системе локально. Преимущество накопителей на магнитной ленте перед жесткими дисками состоит в простоте их администрирования и использования.

2. Инструкция BACKUP DATABASE

Инструкция **BACKUP DATABASE** применяется для выполнения полного резервного копирования или разностного копирования базы данных. Эта инструкция имеет следующий синтаксис:

```
BACKUP DATABASE {db_name | @variable}
TO device_list
[MIRROR TO device_list2]
[WITH | option_list]
```

Описание параметров и опций, присутствующих в инструкции **BACKUP DATABASE** приведено в таблице 1 к лабораторной работе.

3. Резервное копирование системных баз данных.

SQL Server содержит следующий список системных баз:

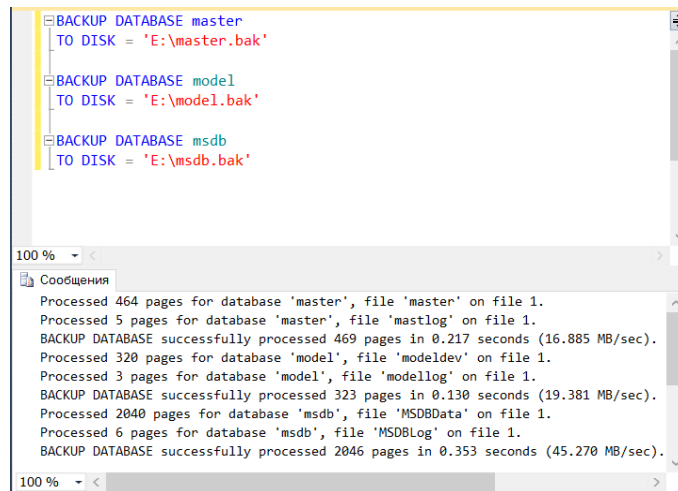
– **master** – содержит сведения обо всех базах данных на сервере. Резервное копирование необходимо делать каждый раз, когда создаются, удаляются или изменяются пользовательские базы данных;

– **model** – используется в качестве шаблона для создаваемых баз данных. Резервное копирование необходимо при изменении настройки самой базы **model**;

– **msdb** – содержит сведения о заданиях и для агента сервера MS SQL Server. копирование необходимо делать каждый раз при добавлении задания для агента сервера MS SQL Server

– **tempdb** – хранит временные данные, например, для транзакций. Уничтожается и создается при перезапуске экземпляра MS SQL Server. Резервное копирование делать нет смысла.

В следующем примере представлен код создания резервных копий системных баз данных.



В результате, на диск «Е» произведено успешное резервное копирование системных баз данных. Результат представлен на рисунке ниже.

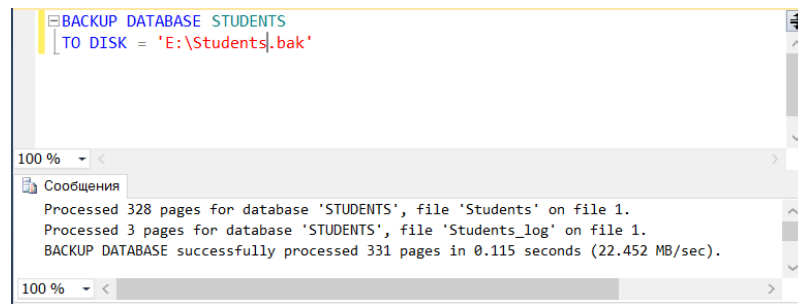
<input type="checkbox"/> master.bak	22.08.2016 15:54	Файл "BAK"	3 828 КБ
<input type="checkbox"/> model.bak	22.08.2016 15:54	Файл "BAK"	2 672 КБ
<input type="checkbox"/> msdb.bak	22.08.2016 15:54	Файл "BAK"	16 572 КБ

4. Полное резервное копирование

Полное резервное копирование включает в себя:

- резервное копирование данных в базе;
- резервное копирование изменений, возникающих во время резервного копирования
- резервное копирование транзакций, не зафиксированных в журнале транзакций.

Для выполнения полного резервного копирования базы данных Students, необходимо выполнить код, представленный на рисунке ниже.



В результате выполнения кода будет создан файл полного резервного копирования.

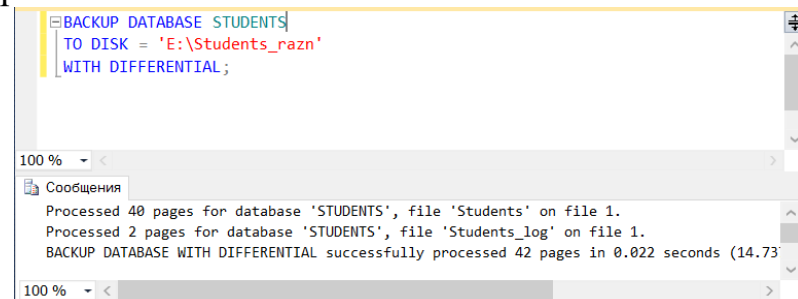
master.bak	22.08.2016 15:54	Файл "BAK"	3 828 КБ
model.bak	22.08.2016 15:54	Файл "BAK"	2 672 КБ
msdb.bak	22.08.2016 15:54	Файл "BAK"	16 572 КБ
Students.bak	22.08.2016 16:00	Файл "BAK"	2 736 КБ

5. Разностное резервное копирование

При создании разностного резервного копирования выполняются следующие действия:

- создание резервных копий баз данных, которые изменились с момента полного резервного копирования.
- создание резервных копий всех операций, выполняющихся во время разностного резервного копирования и всех транзакций не зафиксированных в журнале транзакций.

В следующем примере приведен код выполнения разностного резервного копирования.



Результат представлен на рисунке ниже.

master.bak	22.08.2016 15:54	Файл "BAK"	3 828 КБ
model.bak	22.08.2016 15:54	Файл "BAK"	2 672 КБ
msdb.bak	22.08.2016 15:54	Файл "BAK"	16 572 КБ
Students.bak	22.08.2016 16:00	Файл "BAK"	2 736 КБ
Students_razn	22.08.2016 16:09	Файл	432 КБ

6. Инструкция BACKUP LOG

Инструкция **BACKUP LOG** применяется для создания резервной копии журнала транзакций. Эта инструкция имеет следующий синтаксис:

```
BACKUP LOG {db_name | @variable}
TO device_list
[MIRROR TO device_list2]
[WITH option_list]
```


Значение параметров `db_name`, `@variable`, `device_list` и `device_list2` точно такое же, как и значение одноименных параметров инструкции `BACKUP DATABASE`. Параметр `option_list` имеет те же опции, как и одноименный параметр инструкции `BACKUP DATABASE`, но кроме этого он поддерживает специфические опции журнала транзакций:

✓ **NO_TRUNCATE** – резервное копирование выполняется без усеечения журнала транзакций, т. е. эта опция не удаляет подтвержденные зафиксированные транзакции из журнала. После выполнения инструкции с этой опцией система записывает все недавние операции с базой данных в журнал транзакций. Таким образом, опция `NO_TRUNCATE` позволяет восстановить данные вплоть до точки сбоя в базе данных;

✓ **NORECOVERY** создает резервную копию остатка журнала и оставляет базу данных в состоянии восстановления. Эта опция полезна, когда сбой происходит во вторичной базе данных или при сохранении остатка журнала перед выполнением операции восстановления;

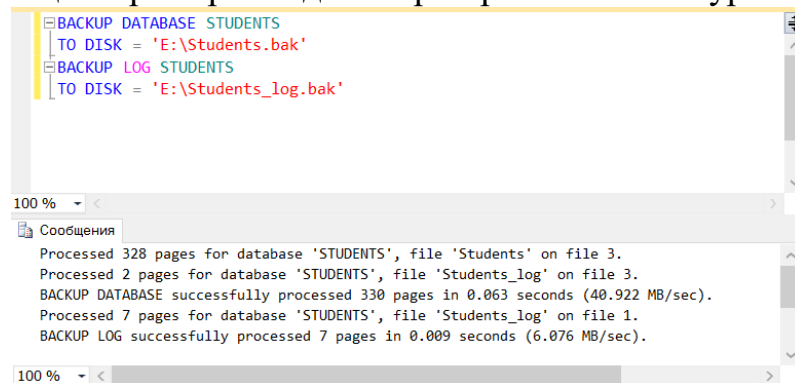
✓ **STANDBY** выполняет резервное копирование остатка журнала и оставляет базу данных в режиме «только для чтения» и в состоянии `STANDBY`.

7. Резервное копирование журнала транзакций базы данных






В процессе резервного копирования журнала транзакций базы данных выполняются следующие действия:

- создается копия журнала транзакций от последнего резервного копирования лога до конца текущего.
- очищаются части журнала транзакций до начала активной части и отбрасываются сведения в неактивной части.

В следующем примере создается резервная копия журнала транзакций.

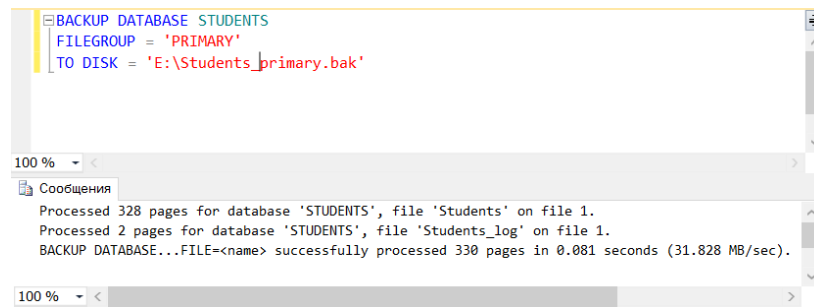


Результат представлен на рисунке ниже.

 master.bak	22.08.2016 15:54	Файл "BAK"	3 828 КБ
 model.bak	22.08.2016 15:54	Файл "BAK"	2 672 КБ
 msdb.bak	22.08.2016 15:54	Файл "BAK"	16 572 КБ
 Students.bak	22.08.2016 16:54	Файл "BAK"	5 564 КБ
 Students_log.bak	22.08.2016 16:54	Файл "BAK"	108 КБ

8. Резервное копирование файловых групп базы данных

В примере ниже приведен код резервного копирования файловых групп базы данных.



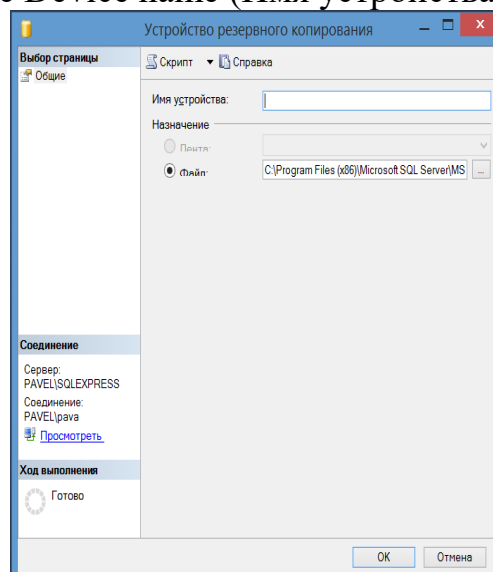
Результат представлен на рисунке ниже.

master.bak	22.08.2016 15:54	Файл "BAK"	3 828 КБ
model.bak	22.08.2016 15:54	Файл "BAK"	2 672 КБ
msdb.bak	22.08.2016 15:54	Файл "BAK"	16 572 КБ
Students.bak	22.08.2016 16:54	Файл "BAK"	5 564 КБ
Students_log.bak	22.08.2016 16:54	Файл "BAK"	108 КБ
Students_primary.bak	22.08.2016 16:58	Файл "BAK"	2 736 КБ

Резервное копирование с помощью интегрированной среды Management Studio

Прежде чем начать выполнять резервное копирование базы данных или журнала транзакций, необходимо указать (или создать) устройства хранения резервной копии. Среда SQL Server Management Studio позволяет одним и тем же образом создавать дисковые устройства и устройства на магнитной ленте. Чтобы создать устройство любого из этих типов, необходимо:

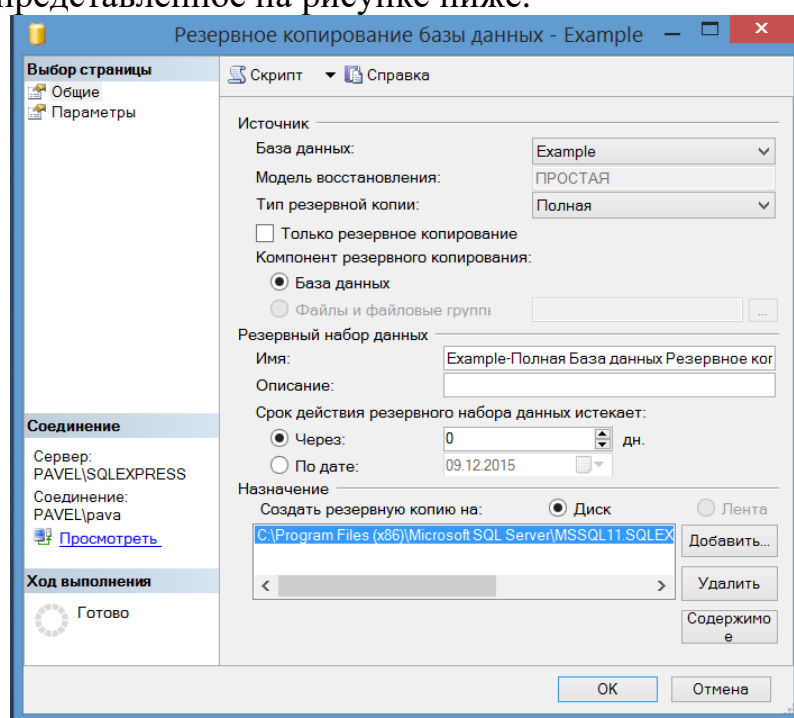
1. развернуть в обозревателе объектов узел Server Objects (Объекты сервера) и щелкните правой кнопкой папку «Backup Devices» («Устройства резервного копирования»);
2. в контекстном меню необходимо выбрать пункт New Backup Device (Создать устройство резервного копирования...);
3. в открывшемся диалоговом окне Backup Device (Устройство резервного копирования), представленном на рисунке ниже, необходимо выбрать дисковое устройство или устройство на магнитной ленте, установив переключатель File (Файл) или Tape (Лента), соответственно, и введя имя для этого устройства в поле Device name (Имя устройства).



В случае если выбран диск (переключатель File), то можно указать расположение файла резервной копии, щелкнув кнопку с тремя точками, расположенную справа от поля для отображения размещения существующих устройств, и выбрав требуемую папку в окне файловой иерархии. В случае, если переключатель Tape (Лента) недоступен для выбора, это означает, что на локальном компьютере отсутствует устройство на магнитных лентах.

После указания устройства для хранения резервной копии можно выполнять резервное копирование базы данных. Для этого необходимо:

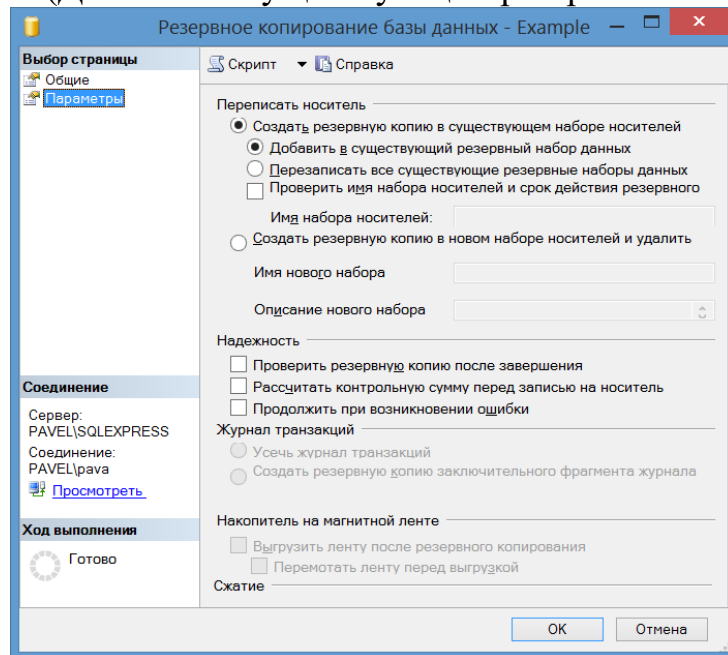
1. развернуть в обозревателе объектов узел сервера базы данных, в нем разверните папку «Databases» («Базы данных»);
2. затем необходимо щелкнуть правой кнопкой требуемую базу данных и в контекстном меню выбрать последовательность команд Tasks (Задачи) | Back Up (Создать резервную копию). Откроется диалоговое окно Back Up Database, представленное на рисунке ниже.



На странице General диалогового окна, представленного на рисунке выше, необходимо выбрать в раскрывающемся списке Backup type (Тип резервной копии) тип резервного копирования (полное — Full, разностное — Differential или копирование журнала транзакций — Transaction Log), в поле Name необходимо ввести имя резервного набора и, необязательно, ввести описание этого набора в поле Description (Описание). В этой же области можно указать срок действия резервной копии, заполнив соответствующие поля. В области Destination (Назначение) необходимо выбрать существующее устройство, нажав кнопку Add (Добавить). (Кнопка Remove (Удалить) используется для удаления устройств резервного копирования из списка устройств, выбранных для хранения резервной копии.)

Чтобы добавить создаваемую резервную копию к уже существующей копии на выбранном устройстве, на странице Options (Параметры), представ-

ленной на рисунке ниже, необходимо выбрать переключатель Append to the existing backup set (Добавить в существующий резервный набор данных).



Если в этой же области выбрать переключатель Overwrite all existing backup sets (Перезаписать все существующие резервные наборы данных), то все существующие на выбранных устройствах резервные копии будут перезаписаны.

Чтобы выполнить проверку сохраненной резервной копии, в области Reliability (Надежность) необходимо установить флажок Verify backup when finished (Проверить резервную копию после завершения). На странице Options (Параметры) можно также задать сохранение резервной копии на новый набор носителей. Для этого в области Overwrite media (Переписать носитель) необходимо установить переключатель Back up to a new media set, and erase all existing backup sets (Создать резервную копию в новом наборе носителей и удалить), а затем ввести в соответствующие поля имя и описание набора носителей.

Для создания и проверки разностной резервной копии и резервной копии журнала транзакций в выпадающем списке Backup type (Тип резервной копии) на странице General (Общие), представленной на рисунке выше необходимо выбрать требуемый тип резервного копирования, а затем выполнить те же самые действия, что и для полного резервного копирования базы данных.

3. После задания всех требуемых параметров резервного копирования, для создания резервной копии необходимо нажать кнопку ОК. Чтобы просмотреть имя, физическое расположение и тип устройств резервного копирования, необходимо развернуть последовательно узел сервера, папку «Server Objects» (Объекты сервера), папку «Backup Devices» (Устройства резервного копирования), а затем выбрать требуемый файл.

Восстановления баз данных и журналов транзакций с помощью инструкций Transact-SQL

Все операции восстановления с резервной копии можно выполнять, используя только две следующие инструкции языка Transact-SQL:

- ✓ **RESTORE DATABASE**, которая используется для восстановления с резервной копии базы данных;
- ✓ **RESTORE LOG**, которая применяется для восстановления резервной копии журнала транзакций.

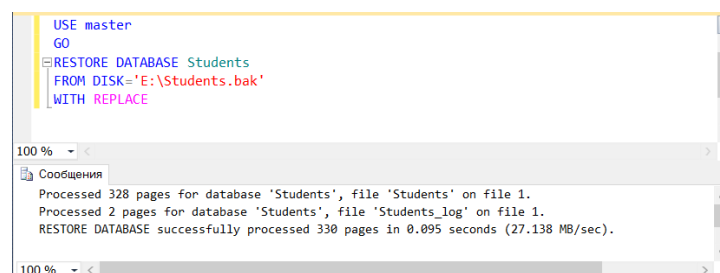
Синтаксис инструкции RESTORE DATABASE выглядит таким образом:

```
RESTORE DATABASE {db_name | @variable}
[FROM device_list]
[WITH option_list]
```

Параметр	Описание
db_name @variable	имя восстанавливаемой базы данных. Имя можно задать и с помощью переменной @variable
device_list	устройство (или устройства), на котором находится резервная копия базы данных. Этот параметр может быть списком имен дисковых файлов или магнитных лент
option_list	указываются опции для разных типов резервного копирования: <ul style="list-style-type: none"> ✓ RECOVERY/NORECOVERY; ✓ CHECKSUM/NO_CHECKSUM; ✓ REPLACE; ✓ PARTIAL; ✓ STOPAT; ✓ STOPATMARK; ✓ STOPBEFOREMARK. Подробное описание параметров приведено в таблице 2 в приложении в лабораторной работе.

Инструкция RESTORE DATABASE также применяется для восстановления базы данных с разностной резервной копии. Синтаксис и опции для восстановления с разностной резервной копии такие же, как и для восстановления с полной резервной копии базы данных. При восстановлении с разностной резервной копии компонент Database Engine восстанавливает только те данные, которые изменились после последнего полного резервного копирования. Следовательно, прежде чем восстанавливать разностную резервную копию, необходимо выполнить восстановление полной резервной копии.

В следующем примере показано восстановление из полной резервной копии.



Для восстановления базы данных из разностной резервной копии в начале восстанавливается полная копия (например, как это сделано на рисунке выше в предыдущем шаге), а далее восстанавливается разностная копия.

```
USE master
RESTORE DATABASE Students
FROM DISK = 'E:\Students_razn'
WITH FILE = 1, RECOVERY
GO
```

Для восстановления журнала транзакций в начале следует восстановить базу данных из полной резервной копии, затем накатить на базу последовательно резервные копии журнала транзакций. Для того, чтоб все отработало корректно, необходимо вернуться к разностному бэкапу, и после него накатить журнал транзакций:

```
USE master
RESTORE DATABASE Students
FROM DISK = 'E:\Students.bak'
WITH FILE = 1, NORECOVERY, REPLACE

RESTORE DATABASE Students
FROM DISK = 'E:\Students_razn'
WITH FILE = 1, NORECOVERY, REPLACE

RESTORE LOG Students
FROM DISK = 'E:\Students_log.bak'
WITH FILE = 1, NORECOVERY
```

Для восстановления файловых групп необходимо выполнить код, представленный на рисунке ниже:

```
RESTORE DATABASE Students
FILEGROUP = 'PRIMARY'
FROM DISK = 'E:\Students_primary.bak'
WITH PARTIAL, RECOVERY, REPLACE
```

Сообщения

Processed 328 pages for database 'Students', file 'Students' on file 1.
 Processed 2 pages for database 'Students', file 'Students_log' on file 1.
 RESTORE DATABASE ... FILE=<name> successfully processed 330 pages in 0.110 seconds (23.437 MB/sec).

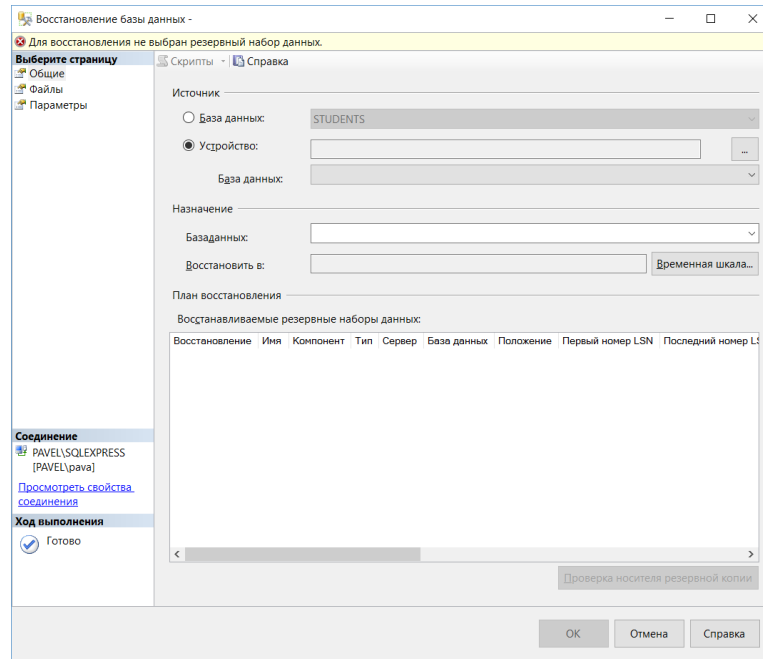
Восстановление системных баз данных

Если экземпляр SQL сервера доступен, то системные базы восстанавливаются согласно приведенной таблице:

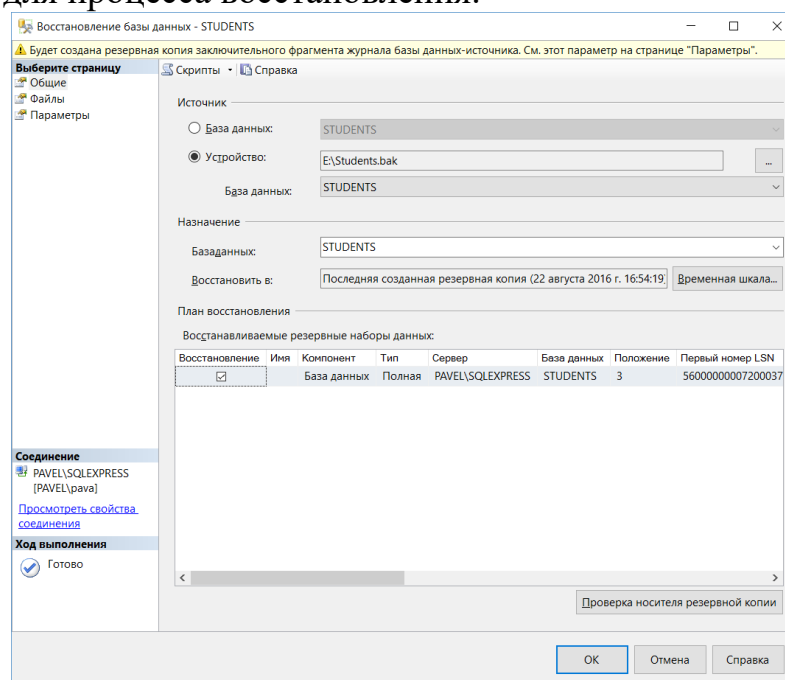
Системная база данных	Способ восстановления
master	Необходимо запустить экземпляр сервера в однопользовательском режиме. Восстановление базы осуществляется так же, как полное восстановление пользовательской базы данных. После восстановления следует перезапустить экземпляр SQL сервера
msdb	Восстановление базы осуществляется так же, как полное восстановление пользовательской базы данных.
model	Восстановление базы осуществляется так же, как полное восстановление пользовательской базы данных

Восстановление баз данных и журналов транзакций с помощью интегрированной среды Management Studio

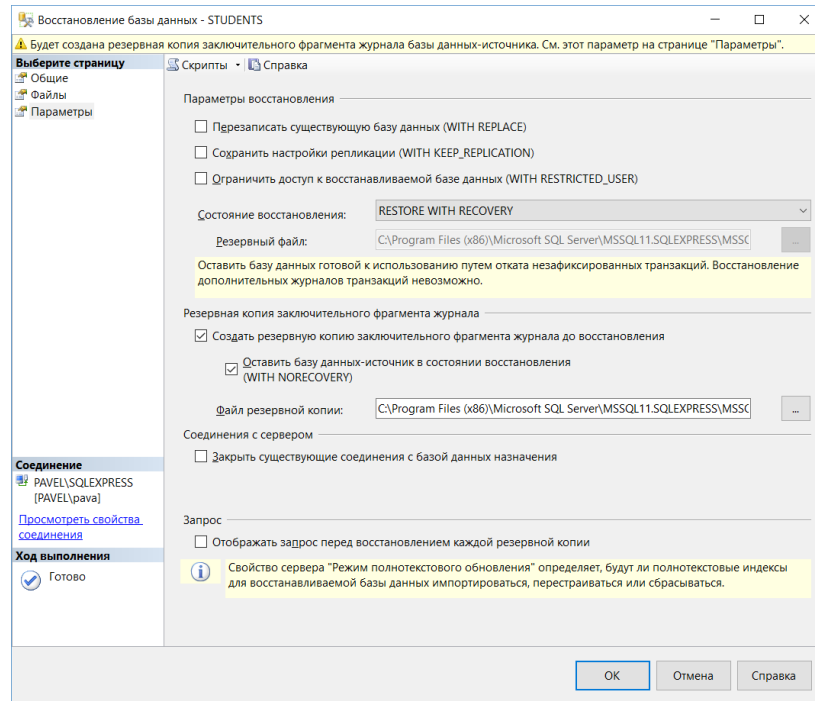
Чтобы восстановить полную резервную копию базы данных, необходимо развернуть папку «Databases» («Базы данных»), щелкнуть правой кнопкой требуемую базу данных и в контекстном меню выберите последовательность команд Tasks (Задачи)| Restore (Восстановить)| Database (База данных...). Откроется диалоговое окно Restore Database, представленное на рисунке ниже.



На странице General этого диалогового окна необходимо выбрать базу данных, в которую (поле To database (Назначение|База данных)) и из которой (поле From device (Источник|Устройство)) необходимо выполнить восстановление. После этого отметить тот набор резервной копии, который нужно использовать для процесса восстановления.



Чтобы для восстановления базы можно было выбрать необходимые опции, в диалоговом окне **Restore Database** (Восстановление базы данных) необходимо перейти на страницу **Options** (Параметры), как представлено на рисунке ниже.



В верхней части окна в разделе **Параметры восстановления** необходимо выбрать один или несколько типов восстановления, установив в соответствующем поле флажок, а ниже в поле **Состояние восстановления** необходимо указать один из трех режимов восстановления, выбрав соответствующее значение из выпадающего списка. Выбор первого **RESTORE WITH RECOVERY** из этих значений позволяет выполнить накат всех подтвержденных и откат всех неподтвержденных транзакций. После применения этой опции база данных будет находиться в согласованном состоянии и готова к использованию. Эта опция эквивалентна выполнению инструкции **RESTORE DATABASE** с опцией **RECOVERY**.

Выбор второго **RESTORE WITH NORECOVERY** из этих значений позволяет не выполнять откат неподтвержденных транзакций, поскольку будет осуществляться восстановление с помощью последующих резервных копий. После применения этой опции база данных будет недоступна для использования, поскольку нужно будет восстановить дополнительные резервные копии журнала транзакций. Эта опция эквивалентна выполнению инструкции **RESTORE DATABASE** с опцией **NORECOVERY**.

При выборе третьего **RESTORE WITH STANDBY** из этих значений необходимо указать файл, который будет использоваться для отката восстановлений. Эта опция эквивалентна выполнению инструкции **RESTORE DATABASE** с опцией **STANDBY**.

Процесс восстановления базы данных из разностной резервной копии эквивалентен процессу восстановления из полной резервной копии базы данных. В этом случае в диалоговом окне **Database Backup** (Восстановление базы

данных) в качестве типа резервной копии нужно выбрать разностную резервную копию (Differential). Единственным отличием между восстановлением с помощью полной резервной копии базы данных и восстановлением с помощью разностной резервной копии является то, что для второго типа восстановления можно выбрать только первое значение из выпадающего списка в поле Состояние восстановления в нижней половине страницы Options (Параметры), т. е. значение RESTORE WITH RECOVERY.

2. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Дайте определение понятию *индексированное представление*.
2. Поясните понятие простого и составного индекса.
3. Назовите системную функцию, с помощью которой можно получить информацию о существующих индексах. Поясните ее параметры.
4. Назовите команды создания, удаления и изменения индекса.

ДОМАШНЕЕ ЗАДАНИЕ

[1], страницы 435-463

ЛИТЕРАТУРА

1. Петкович, Д. Microsoft SQL Server 2012. Руководство для начинающих: пер. с английского / Д. Петкович. – СПб.: БХВ-Петербург, 2013. – 816 с.: ил.
2. Сеть разработчиков Microsoft [Электронный ресурс]. – Режим доступа: <https://msdn.microsoft.com/ru-ru/library>

Преподаватель

С.В. Банцевич

Рассмотрено на заседании цикловой
комиссии программного обеспечения
информационных технологий №10
Протокол № от « » _____ 2017
Председатель ЦК С.В. Банцевич

Таблица 1 Описание параметров и опций инструкции BACKUP DATABASE

Параметр опция		Значение
db_name @variable		имя базы данных, для которой выполняется резервное копирование (оно может быть задано в переменной @variable)
device_list		указывается одно или несколько имен устройств, где будет храниться копия резервной копии (может быть списком)
option_list	DIFFERENTIAL	задает разностное резервное копирование
	NOSKIP/SKIP	SKIP отменяет проверку срока завершения и имени набора резервного копирования, которая обычно выполняется с помощью инструкции BACKUP DATABASE, чтобы предотвратить запись поверх существующих резервных наборов. Опция NOSKIP, используемая по умолчанию, указывает инструкции BACKUP, что должна быть выполнена проверка даты завершения и имен всех наборов резервных копий, прежде чем выполнять их перезапись.
	NOINIT/INIT	INIT указывает, что нужно перезаписать все существующие данные на носителе, за исключением заголовка носителя, если таковой имеется. Если при указании этого параметра на носителе имеется резервный набор, срок действия которого не истек, операция резервного копирования завершается неудачей. В таком случае, для перезаписи устройства резервного копирования применяется комбинация опций SKIP и INIT. Если указана опция NOINIT, которая используется по умолчанию, создаваемая резервная копия добавляется к уже имеющимся на носителе резервным наборам.
	NOFORMAT/FORMAT	применяется для инициализации носителя информации. Опция NOFORMAT, которая является опцией по умолчанию, означает, что операция резервного копирования сохранит существующий заголовок носителей и резервные наборы данных на томах носителей, используемых для текущей операции резервного копирования.
	UNLOAD/NOUNLOAD	применяются только при использовании в качестве носителей резервной копии магнитной ленты. Опция UNLOAD указывает, что после завершения операции резервного копирования необходимо выполнить перемотку магнитной ленты и снять ее с устройства. Эта опция применяется в начале сеанса резервного копирования по умолчанию. Опция NOUNLOAD указывает, что после завершения резервного копирования магнитная лента в устройстве оставляется.
	MEDIANAME MEDIADescription	и указывают описание и имя набора носителей, соответственно
	BLOCKSIZE	применяется для указания физического размера блока в байтах. Поддерживаются блоки размером в 512, 1024, 2048, 4096, 8192, 16 384, 32 768 и 65 536 (64 Кбайт) байт. Размер блока по умолчанию для накопителей на магнитной ленте — 65 536 байт, и 512 байт для всех других устройств.
	COMPRESSION	сжатие файла резервного копирования

Таблица 2 Описание значений параметра option_list инструкции RESTORE DATABASE

Значение параметра option_list	Описание
RECOVERY/NORECOVERY	<p>Опция RECOVERY указывает выполнить накат всех подтвержденных (зафиксированных) транзакций, а для всех неподтвержденных (незафиксированных) транзакций выполнить откат. После выполнения инструкции с этой опцией база данных будет находиться в согласованном состоянии и будет готова для использования. Эта опция является опцией по умолчанию.</p> <p>Когда указана опция NORECOVERY, не выполняется откат неподтвержденных (незафиксированных) транзакций, поскольку выполнение восстановления будет продолжаться с других резервных копий. После выполнения инструкции с опцией NORECOVERY база данных будет недоступной для использования.</p>
CHECKSUM/NO_CHECKSUM	<p>Опция CHECKSUM инициирует проверку как контрольной суммы резервной копии, так и контрольных сумм страниц, если эти суммы имеются. При отсутствии контрольных сумм инструкции восстановления RESTORE выполняются без их проверки. Опция NO_CHECKSUM в операции восстановления явно отключает проверку контрольных сумм.</p>
REPLACE	<p>Опция REPLACE указывает на замену существующей базы данных данными из резервной копии другой базы данных. В этом случае сначала удаляется существующая база данных, а различия в именах файлов базы данных и имени базы данных игнорируются. (Если опция REPLACE не указывается, то система базы данных выполняет безопасную проверку, которая гарантирует, что существующая база данных не будет заменена, если имена файлов базы данных или имя самой базы данных отличаются от соответствующих имен в наборе резервной копии.)</p>
PARTIAL	<p>Опция PARTIAL задает операцию частичного восстановления. Эта опция позволяет восстановить часть базы данных, состоящую из ее основной файловой группы и одной или нескольких вторичных файловых групп, которые указываются в дополнительной опции FILEGROUP. (Опцию PARTIAL нельзя применять в инструкции RESTORE LOG.)</p>
STOPAT	<p>Опция STOPAT позволяет восстановить базу данных до состояния, в котором она находилась в конкретный момент времени перед сбоем, указав требуемую точку во времени. Сервер восстанавливает все подтвержденные (зафиксированные) транзакции, которые были записаны в журнал транзакций до указанного момента времени.</p>
STOPATMARK	<p>Опции STOPATMARK и STOPBEFOREMARK задают восстановление до определенной метки.</p>
STOPBEFOREMARK	

