

Урезание логов и перевод баз 1С:Предприятие в модель SIMPLE

Назначение

Скрипт предназначен для **продуктивных (рабочих) баз 1С:Предприятие** на Microsoft SQL Server. Его задача — перевести все пользовательские базы в модель восстановления **SIMPLE** и освободить место, занимаемое файлами журналов транзакций (LDF).

Данный подход является стандартной практикой для продуктовых баз 1С, поскольку восстановление на момент времени (point-in-time recovery) в 1С связано с существенными сложностями и на практике применяется редко.

“ **Внимание!** Риски применения описаны в сноске в конце статьи.
Ознакомьтесь с ними перед запуском скрипта на продуктивном контуре.

Скрипт

sql

```
-----
-- СЛУЖЕБНЫЕ ПЕРЕМЕННЫЕ
DECLARE @db_name      NVARCHAR(100); -- имя базы данных
DECLARE @db_name_q    NVARCHAR(300); -- экранированное имя базы
DECLARE @sql          NVARCHAR(MAX); -- текст динамического SQL

DECLARE cursor_size_srv CURSOR FOR

-- Выбираем все базы, кроме системных
SELECT name AS DBName
```

```

FROM sys.databases
WHERE name NOT IN ('master','msdb','model','tempdb')
ORDER BY name;

-- Цикл по всем базам, попавшим в выборку
OPEN cursor_size_srv;
FETCH NEXT FROM cursor_size_srv INTO @db_name;

WHILE (@@FETCH_STATUS = 0)
BEGIN
    -- Экранируем имя базы (оборачиваем в [ ] и удваиваем )
    SET @db_name_q = QUOTENAME(@db_name);

    -- Формируем текст динамического SQL
    SET @sql = N'
        DECLARE @logname NVARCHAR(100);
        USE ' + @db_name_q + N';
        SELECT @logname = name FROM sys.database_files WHERE type = 1;
        ALTER DATABASE ' + @db_name_q + N' SET RECOVERY SIMPLE;
        DBCC SHRINKFILE (@logname, EMPTYFILE);';

    -- Выполняем
    EXEC (@sql);

    -- Следующая база данных
    FETCH NEXT FROM cursor_size_srv INTO @db_name;
END

CLOSE cursor_size_srv;
DEALLOCATE cursor_size_srv;
-----

```

Разбор логики работы

1. Объявление переменных и курсора

sql

```

DECLARE @db_name      NVARCHAR(100);
DECLARE @db_name_q    NVARCHAR(300);
DECLARE @sql          NVARCHAR(MAX);

DECLARE cursor_size_srv CURSOR FOR
SELECT name AS DBName
FROM sys.databases
WHERE name NOT IN ('master','msdb','model','tempdb')
ORDER BY name;

```

Курсор проходит по всем базам на экземпляре, исключая системные (`master`, `msdb`, `model`, `tempdb`). Это защищает от случайного изменения настроек системных баз.

Переменные:

- `@db_name` — имя текущей базы из курсора.
- `@db_name_q` — экранированное имя (в квадратных скобках).
- `@sql` — текст динамического SQL для выполнения.

2. Перебор баз в цикле

sql

```
OPEN cursor_size_srv;
FETCH NEXT FROM cursor_size_srv INTO @db_name;

WHILE (@@FETCH_STATUS = 0)
BEGIN
    ...
    FETCH NEXT FROM cursor_size_srv INTO @db_name;
END
```

Стандартный паттерн работы с курсором: открыть, получить первую строку, обрабатывать в цикле, пока `@@FETCH_STATUS = 0`.

3. Экранирование имени базы

sql

```
SET @db_name_q = QUOTENAME(@db_name);
```

Функция `QUOTENAME()` оборачивает имя в квадратные скобки и удваивает встречающиеся внутри `]`. Это делает скрипт устойчивым к любым именам баз:

- **Имена, начинающиеся с цифры** (например, `1C_Prod`) — без экранирования вызывают синтаксическую ошибку, так как не являются допустимыми обычными идентификаторами SQL Server.
- **Имена с пробелами и спецсимволами** (например, `База 1С Прод`).

Важно: `QUOTENAME()` вызывается **на уровне внешнего скрипта**, а не внутри строки `EXEC`. В строку динамического SQL подставляется уже готовое значение `@db_name_q`.

4. Формирование и выполнение динамического SQL

sql

```
SET @sql = N'  
    DECLARE @logname NVARCHAR(100);  
    USE ' + @db_name_q + N';  
    SELECT @logname = name FROM sys.database_files WHERE type = 1;  
    ALTER DATABASE ' + @db_name_q + N' SET RECOVERY SIMPLE;  
    DBCC SHRINKFILE (@logname, EMPTYFILE);';  
  
EXEC (@sql);
```

Ключевые шаги внутри динамического SQL:

- `USE [db_name]` — переключение контекста на текущую базу (нужно для `sys.database_files`).
- `SELECT @logname = name FROM sys.database_files WHERE type = 1` — получаем логическое имя файла журнала (`type = 1`).
- `ALTER DATABASE [db_name] SET RECOVERY SIMPLE` — переводим базу в простую модель восстановления.
- `DBCC SHRINKFILE (@logname, EMPTYFILE)` — очищаем файл журнала, перемещая все страницы в другой файл той же файловой группы.

5. Особенность объявления @logname

Переменная `@logname` объявлена **внутри динамического SQL**, а не снаружи. Это сделано намеренно: при смене контекста через `USE` переменная из внешнего блока была бы недоступна для `DBCC SHRINKFILE`.

6. Типы переменных

- `@db_name_q NVARCHAR(300)` — с запасом под скобки и удвоение символов `1`.
 - `@sql NVARCHAR(MAX)` — снимает ограничение в 4000 символов, если имя базы длинное или добавляются дополнительные команды.
 - Строковые литералы помечены префиксом `N` для корректной работы с Unicode (кириллица в именах баз).
-

Рекомендации по применению

Перед запуском

1. **Сделайте полный бэкап** всех целевых баз. Это обязательное условие.
2. **Убедитесь, что модель SIMPLE допустима** для всех баз, попадающих в выборку.
3. **Запускайте в окно обслуживания** — `SHRINKFILE` может блокировать операции записи.
4. **Проверьте initial size** файла журнала. Если задано большое значение (например, 100 ГБ), шринк не уменьшит файл ниже этой отметки.

После перевода в SIMPLE

Если требуется физически уменьшить файл журнала до разумного размера, добавьте после `EMPTYFILE` команду:

sql

```
DBCC SHRINKFILE (@logname, 512); -- 512 МБ, подберите под задачу
```

Компенсация отсутствия PITR

После перевода в `SIMPLE` восстановление возможно только на момент последнего полного или дифференциального бэкапа. Необходимо:

- **Делать полные бэкапы часто** — с учётом допустимого окна потери данных.
- **Использовать дифференциальные бэкапы** каждые несколько часов для активных баз.
- **Регулярно проверять бэкапы восстановлением** в тестовую базу.

Проверка результата

sql

```
SELECT
    d.name AS DatabaseName,
    d.recovery_model_desc,
    f.name AS FileName,
    f.type_desc,
    f.size * 8 / 1024 AS SizeMB
FROM sys.databases d
```

```
JOIN sys.master_files f ON f.database_id = d.database_id  
ORDER BY d.name, f.type_desc;
```

Особенность 1С: журнал может не урезаться до минимума

Даже в модели **SIMPLE** размер журнала транзакций определяется **максимальным размером одной транзакции**. В 1С это особенно заметно при массивных операциях: перепроведение документов, закрытие месяца, реструктуризация базы. Если типичная операция занимает несколько гигабайт в логе, минимальный размер LDF не будет меньше этой величины.

Если журнал снова быстро растёт после урезания — проверьте, нет ли аномально длительных операций в 1С.

Сноска: риски применения

“ **Внимание! Перед запуском скрипта на продуктивном контуре ознакомьтесь с рисками.**

1. Потеря возможности восстановления на момент времени (PITR).

Перевод базы в модель **SIMPLE** делает невозможным восстановление на произвольный момент между полными бэкапами. При сбое вы сможете восстановиться только на момент последнего полного или дифференциального бэкапа. Вся работа, выполненная после него, будет потеряна.

2. Несовместимость с технологиями репликации.

Модель **SIMPLE** не поддерживается для групп доступности Always On (AG), лог-шиппинга (Log Shipping) и репликации транзакций. Если эти технологии настроены, их работа будет нарушена.

3. Невозможность точечного отката ошибок пользователей.

При ошибочных действиях пользователя (массовое удаление, некорректное перепроведение документов, закрытие смены ККМ) откатить изменения до нужного момента без PITR невозможно — придётся восстанавливать базу из последнего бэкапа и повторно вводить

данные.

4. Блокировка операций во время SHRINKFILE.

DBCC SHRINKFILE (... , EMPTYFILE) может длительное время ожидать завершения активных транзакций. На продуктовой базе это приведёт к блокировкам и замедлению работы пользователей.

5. Журнал может не уменьшиться.

Даже после перевода в SIMPLE и шринка файл журнала не станет меньше максимального размера одной транзакции. Для 1С характерны крупные транзакции (закрытие месяца, перепроведение, реструктуризация), поэтому ожидания «лога в 100 МБ» могут не оправдаться.

6. Ограничение по initial size.

Если в свойствах базы задан большой начальный размер файла журнала, шринк не уменьшит файл ниже этой отметки. Требуется предварительная корректировка начального размера.

7. Необходимость частых бэкапов.

После перехода в SIMPLE критически важно обеспечить регулярные полные и дифференциальные бэкапы. Без них при сбое возможна потеря существенного объёма данных.

8. Требования аудита и регуляторов.

Некоторые отрасли и внутренние регламенты требуют возможности восстановления на произвольный момент времени. Перевод в SIMPLE может нарушить эти требования.

Рекомендация. Перед запуском скрипта на продуктивном контуре согласуйте решение с ответственными за сохранность данных и убедитесь, что выбранная стратегия резервного копирования компенсирует отсутствие PITR.

Итог

Скрипт предназначен для продуктовых баз 1С:Предприятие и решает задачу перевода баз в модель SIMPLE с освобождением журналов транзакций. Благодаря QUOTENAME() он корректно работает с любыми именами баз, включая начинающиеся с цифры и содержащие пробелы. Перед запуском на рабочем контуре обязательно ознакомьтесь со сноской о рисках и обеспечьте регулярное резервное копирование.

Версия #4

Admin создал 2026-09-10 13:45:41 UTC

Admin обновил 2026-09-10 14:01:55 UTC