С.С. Федоров, Ю.А. Жомов, О.А. Нестерова

Тюменский государственный университет, г. Тюмень
УДК 004.056

АНАЛИЗ НОВЫХ ВОЗМОЖНОСТЕЙ ЗАЩИТЫ ДАННЫХ В MICROSOFT SQL SERVER 2019

Аннотация. В статье представлено описание результатов анализа эффективности и рисков применения новых функций Microsoft SQL Server 2019, таких как классификация данных и ускоренное восстановление данных.

Ключевые слова: ускоренное восстановление, классификация, Microsoft SQL Server, производительность, аудит.

Введение

Microsoft SQL Server 2019 предоставил ряд новых возможность в области информационной безопасности. В ходе данной работы были рассмотрены функции классификации данных и ускоренного восстановления. Отсутствие подробной документации и описания их влияния на производительность, привело к необходимости проведения анализа: изучение возможностей и особенностей функций; выявление уязвимостей; оценка влияния на производительность. Главным критерием оценки являлось время выполнения запросов. Все измерения проводились с процессором Intel i7-4770 с частотой 3.8 ГГц и 16 Гб ОЗУ DDR3 с частотой 1600 МГц (две AMD R538G1601U2S-UGO), диск samsung 870 qvo (MZ-77Q1T0BW), жестким диском Samsung 870 qvo (MZ-77Q1T0BW), ОС Windows 10 Pro x64. Измерение времени и выполнение запросов осуществлялось через программу SqlQueryStress.

Классификация данных

Эта функция помогает разметить данные для последующей реализации мер защиты и является одной из возможностей аудита доступа к данным [1]. Для классификации имеется возможность настроить способы определения по шаблону типа информации (банковская карта, номер телефона) и категории ее конфиденциальности. Однако по указанному типу информации

автоматически могут определиться только 2 категории конфиденциальности из 6 стандартных.

Обнаружение данных

На основе изучения файлов и создания различных таблиц, было выяснено, что функция обнаружения работает по следующему алгоритму – сверяет названия столбцов таблиц с заранее заданными правилами (шаблонами) для каждого типа информации. Стандартный набор правил поиска с типами информации и категориями конфиденциальности поставляется json-файлом вместе с MS SQL Management Studio. Файлы полностью идентичны в русской и английской локализациях. Большинство шаблонов поиска присутствуют только на английском языке. Документация по созданию своего файла с правилами классификации отсутствует, но такая возможность имеется. Пример файла пользовательского правила классификации показан на рисунке 1.

```
{
    "LastModifiedUtc": "0001-01-01T00:00:00",
    "Version": "2.0",
    "Labels": {
      "1866ca45-0000-0000-0000-04d407f147ad": {
        "DisplayName": "Проверочная категория",
        "Description": "Описание",
        "Rank": "Low",
        "Order": 100
      }
    },
    "InformationTypes": {
      "b40ad280-0000-0000-0000-2f1a08651fcf": {
        "DisplayName": "Данные 1",
        "Description": "Описание",
        "Order": 100,
        "RecommendedLabelId": "1866ca45-0000-0000-0000-04d407f147ad",
        "Keywords": [
          {
            "Pattern": "%test%"
          }
        ]
      }
    }
}
```

Рис. 1. Пример создания пользовательского правила классификации данных

Хранение в базе данных

Данные классификации сохраняются в виде метаданных столбцов. В файле базы данных метаданные с идентификаторами и названиями сохраняются в виде текста с 16-битной кодировкой Unicode и порядком байт от старшего к младшему. Пример представления данных классификации в метаданных столбцов на диске показан на рисунке 2.

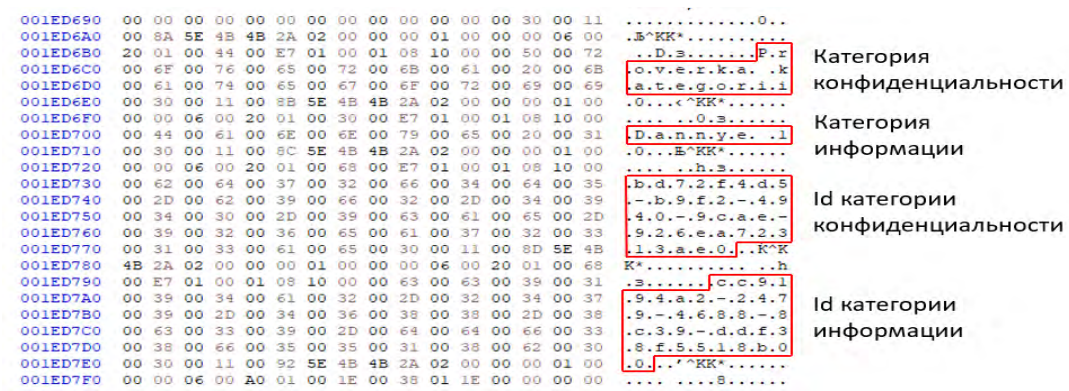

Рис. 2. Представление данных классификации в метаданных столбцов на диске

Хранение данных в таком виде может быть недостатком. Так, если у атакующего будет доступ к диску, то он сможет достаточно быстро найти файлы баз данных с конфиденциальными данными, например выполнив поиск по GUID типа информации. Поэтому авторы статьи рекомендуют менять GUID как в исходных, так и в пользовательских шаблонах. Кроме того, GUID не удаляется из файла после отключения функции классификации данных. Обязательно необходимо вручную удалить GUID из метаданных столбца.

Просмотр информации о конфиденциальности столбцов выполняется через представление «sys.sensitivity_classifications» [2], которое также является потенциальным источником информации для атакующего, поэтому следует контролировать, выдачу разращения «view any sensitivity classification».

Аудит и классификация

Изучение влияния на производительность проводилось при включенном аудите при чтении случайных строк из 10000, записанных в таблицу с шестью столбцами, половина которых были классифицированы с разными параметрами. Для оценки производительности при высокой нагрузке было принято решение выполнить поочередное обращение на чтение данных к классифицированному столбцу и неклассифицированному. В сумме выполнено 20 испытаний с отправкой по 50 тысяч запросов в одном пакете к случайным строкам. Запросы на чтение к классифицированному столбцу в среднем выполнялись на 3.4 секунды дольше, что составило около 17%. Время выполнения запросов к классифицированному и неклассифицированному столбцам в 20 испытаниях показано на рисунке 3.

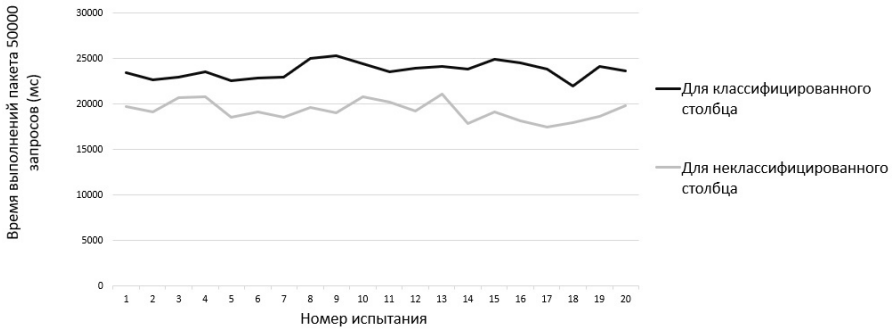

Рис. 3. Время выполнения пакета 50000 запросов к классифицированному и неклассифицированному столбцам в 20 испытаниях

При низкой загрузке – одном запросе раз в несколько миллисекунд – потеря времени составила около 1.6 мс (4%). Время выполнения одного запроса к классифицированному и неклассифицированному столбцам в 100 испытаниях показано на рисунке 4. Горизонтальными линиями показаны средние значения.

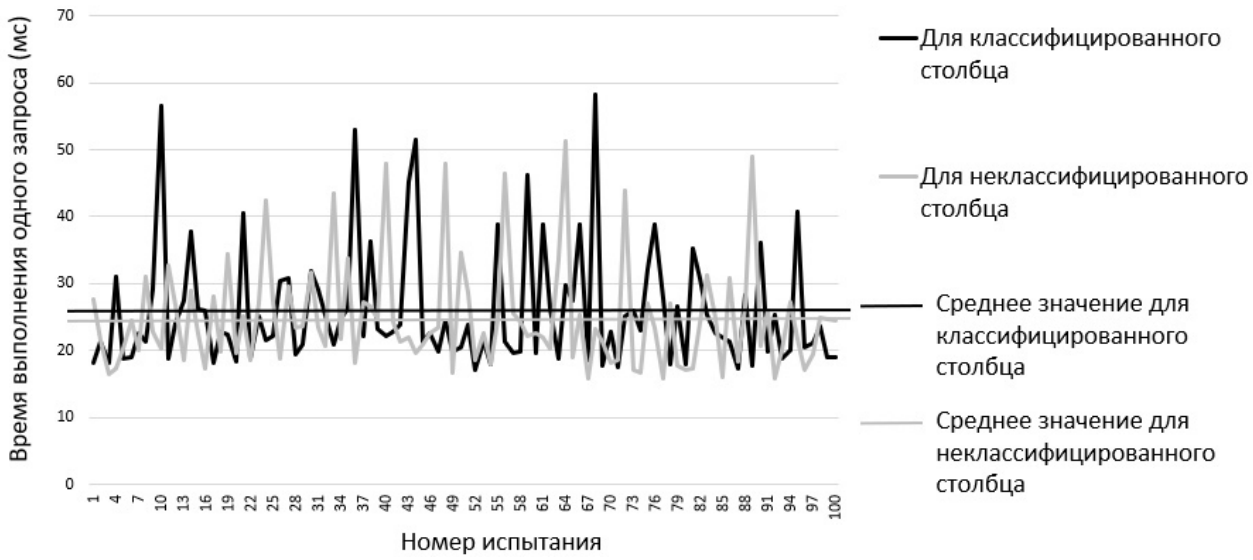

Рис. 4. Время выполнения одного запроса к классифицированному и неклассифицированному столбцам в 100 испытаниях

Проведено 110 испытаний, для оценки влияния количества запросов на производительность. В каждом, проводились измерения среднего времени среди 100 попыток выполнения пакета запросов поочередно для классифицированного и неклассифицированного столбца. Запросы в пакете отправлялись

одновременно, между пакетами был установлен таймаут в 200 миллисекунд. Каждый раз количество запросов увеличивалось на 25. Потеря производительности оценивалась как процент разницы времени выполнения одинакового количества запросов к классифицированному и неклассифицированному столбцам, график этой зависимости в одном испытании приведен на рисунке 5.

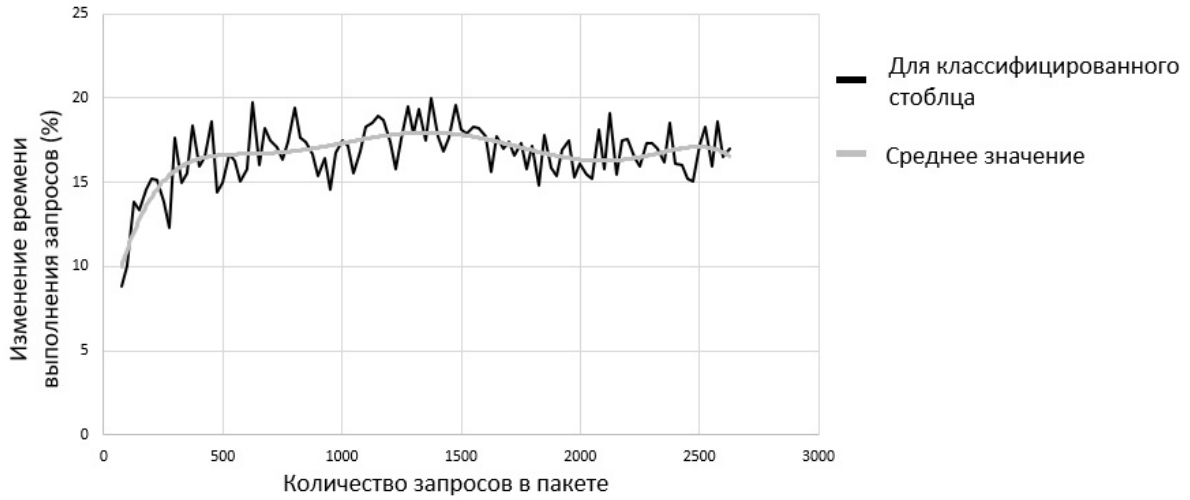

Рис. 5. График зависимости потери производительности от количества запросов к классифицированному и неклассифицированному столбцам в одном испытании

Как видно из графика при увеличении количества запросов до 500 увеличение времени выполнения пакета запросов растет на 15% – 20%.

Также измерена потеря производительности в зависимости от количества используемых классифицированных столбцов в одном запросе. Проведено 4 измерения, в первом запрос содержал 3 неклассифицированных столбца, во втором – 2 неклассифицированных и один классифицированный и так количество классифицированных увеличивалось до 3. В ходе измерений запросы выполнялись в 12 пакетах, в каждом пакете количество одновременных запросов увеличивалось на 50. График зависимости времени выполнения запросов от количества используемых классифицированных столбцов показан на рисунке 6.

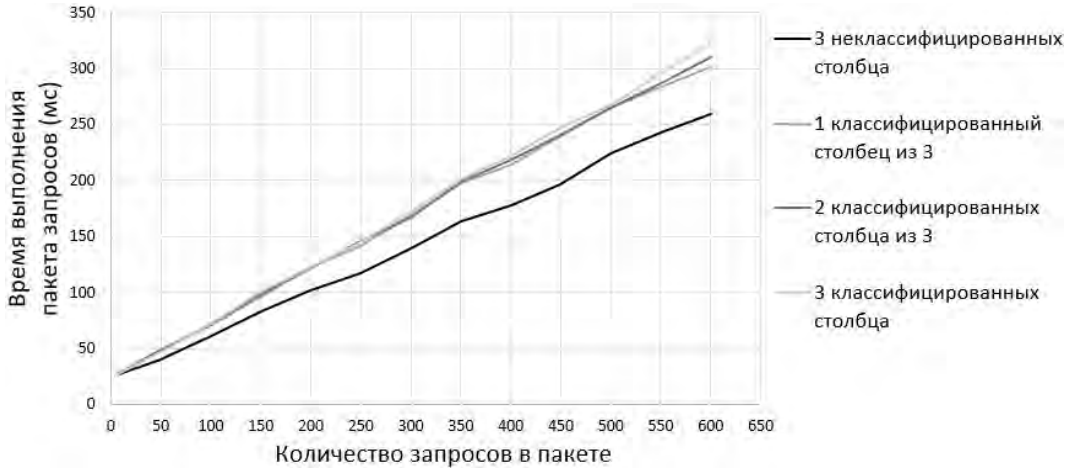

Рис. 6. График зависимости времени выполнения пакета запросов от количества используемых в них классифицированных столбцов

При включенном аудите скорость выполнения снижается при использовании хотя бы одного классифицированного столбца. Зависимости от количества используемых классифицированных столбцов выявлено не было.

Ускорение восстановления баз данных

В процессе обычного восстановления после сбоя система выполняет серию сканирований журнала транзакций и поэтому скорость напрямую зависела от количества изменений, выполненных транзакциями с момента самой ранней незавершенной транзакции или контрольной точки. При использовании ускоренного восстановления сканирование журнала начинается с последней контрольной точки вне зависимости от числа и длительности транзакций и количества внесенных изменений. Эта функция позволяет ограничить рост журнала транзакций [3].

В некоторых источниках [4] описывается, что при включении функции восстановления к строкам добавляется 14-байтовый указатель на старые данные, однако было обнаружено, что после включения для уже добавленных строк не происходит никаких изменений. Когда в строку таблицы вносятся изменения менее чем на 200 байт, предыдущая версия сохраняется внутри строки в виде дифференциальной копии, для экономии памяти. Но может привести и к обратному эффекту. Для демонстрации была создана таблица с пятью столбцами

и одной строкой. В первом столбце строки сохранено значение Field_1 и в последнем Field_*. Далее эти значения были изменены на Stolbec_1 и Field05. После этого было изучено содержимое памяти (рисунок 7). На рисунке видно, что сохранились только два текстовых значения: “Field” и “_*”. Для хранения изменений 7 байт понадобилась запись 60 байт служебной информации, при этом, хранение полной копии потребовало бы только 37 байт.

Рис. 7. Пример хранения изменений строки в памяти

Когда изменяется более 200 байт, то версии и другая служебная информация записываются в хранилище версий. При этом в файловую группу со строками таблицы добавляется 14-битная информация о версии с номером транзакции. Если изменение меньше 200 байт, то в конце предыдущей версии дописывается дифференциальная копия.

Информация о версии будет сохраняться до момента удаления или создания дифференциальной копии строки и ее последующего удаления. Полные копии из хранилища также как и дифференциальные удаляются, когда больше не потребуются. Однако, полные копии все еще остаются в самом файле хранилища, только со временем поверх них запишутся новые копии. Поэтому некоторое время в файле может сохраняться конфиденциальная информация, если она была отредактирована или удалена ранее. Если нежелательно допускать это, то необходимо или отключить ускоренное восстановление или время от времени провоцировать перезапись. Возможность работы функции

восстановления в другой файловой системе, где предусмотрено заполнение мусором таких областей в работе не рассматривалась.

Оценка времени выполнения ускоренного восстановления и влияния на выполнение запросов

Измерение времени восстановления базы данных с одной таблицей после сбоя проводилось путем изменения увеличивающегося числа случайно сгенерированных текстовых строк в таблицу с пятью текстовыми полями работа сервера неожиданно завершалась без фиксации транзакции и с помощью журнала просматривалось время восстановления. Сравнение времени восстановления базы данных после сбоя при ускорении восстановления и без ускорения показано на рисунке 8.

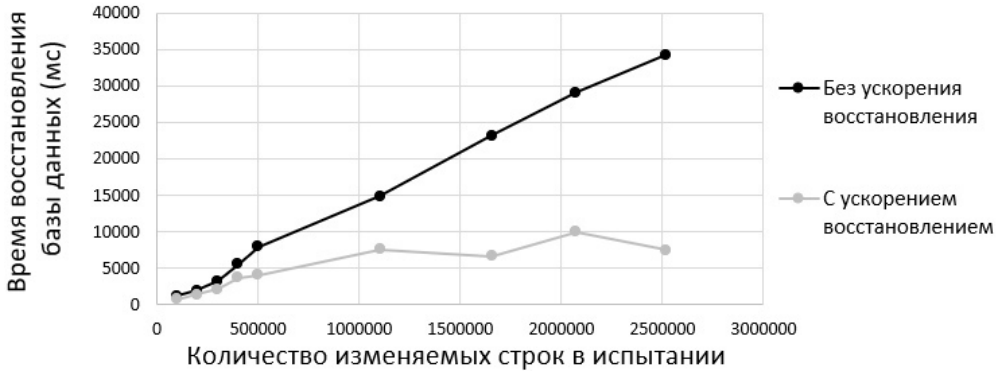

Рис. 8. Сравнение времени восстановления базы данных после сбоя при ускоренном восстановлении и без

Как видно из рисунка при включенной функции ускоренного восстановления время восстановления базы данных гораздо меньше, чем при выключенной, при работе с большим количеством строк. В случае небольшого числа строк ускоренное восстановление может занимать на несколько миллисекунд (до 30 мс) больше на одну транзакцию из-за длительного этапа анализа изменений. Поэтому, в случае если есть большое число небольших транзакций скорость работы может снизиться.

Несмотря на то, что при включении функции ускоренного восстановления к новым строкам добавляется идентификатор транзакции, скорость их добавления не снижается. Для сравнения времени обновления строк таблицы при

включенном ускоренном восстановлении было проведено 90 испытаний, в каждом из которых добавлялось на 200 строк больше. Результат показан на рисунке 9. Потеря в скорости заметна из-за записи служебной информации.

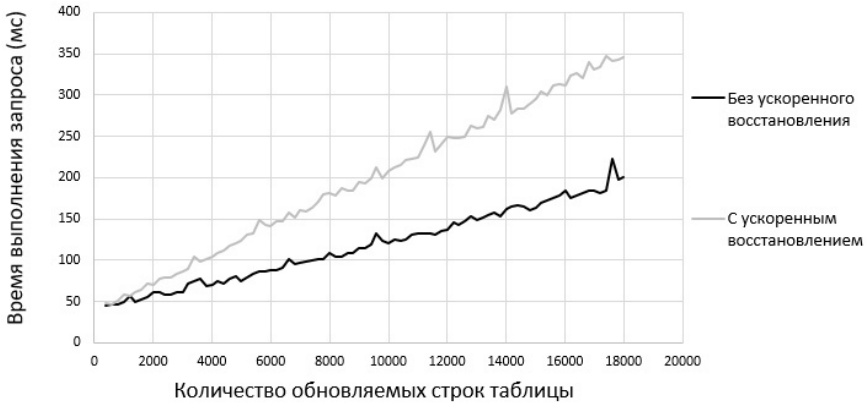

Рис. 9. Сравнение времени обновления строк таблицы при включенном ускоренном восстановлении

Для измерения влияния ускоренного восстановления на скорость удаления данных было проведено 100 испытаний, в каждом из которых количество удаляемых строк увеличивалось на 200.

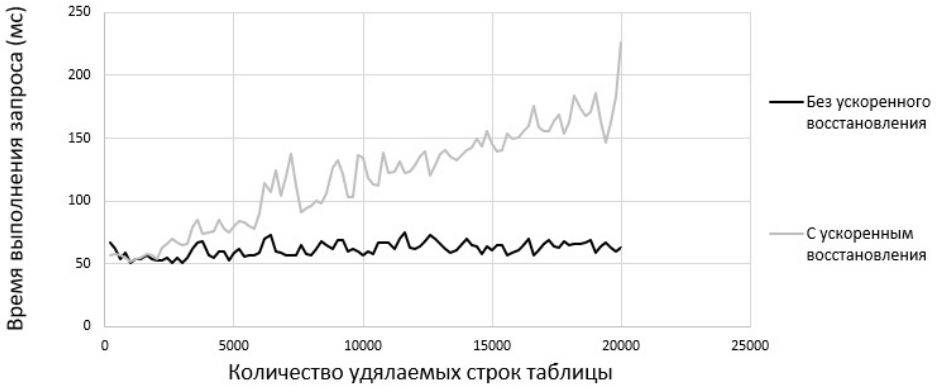

Рис. 10. Сравнение скорости удаления

В результате было обнаружено что ускоренное восстановление сильнее всего влияет на скорость удаления. Так, например при удалении 20000 строк скорость оказалась ниже в 3.5 раза.

Ошибка выполнения при ускоренном восстановлении

При выполнении следующей последовательности команд, возникает ошибка выполнения, приводящая к повреждению базы данных: включить функции ускоренного восстановления; начать транзакцию; изменить строки таблицы; отменить транзакцию; отключить функцию ускоренного

восстановления; начать транзакцию; запустить запрос на удаление ранее измененных строк из таблицы; отменить транзакцию. При изменении 1000 и более строк ошибка происходит всегда. Последовательность команд и сообщение об ошибке показаны на рисунке 11.

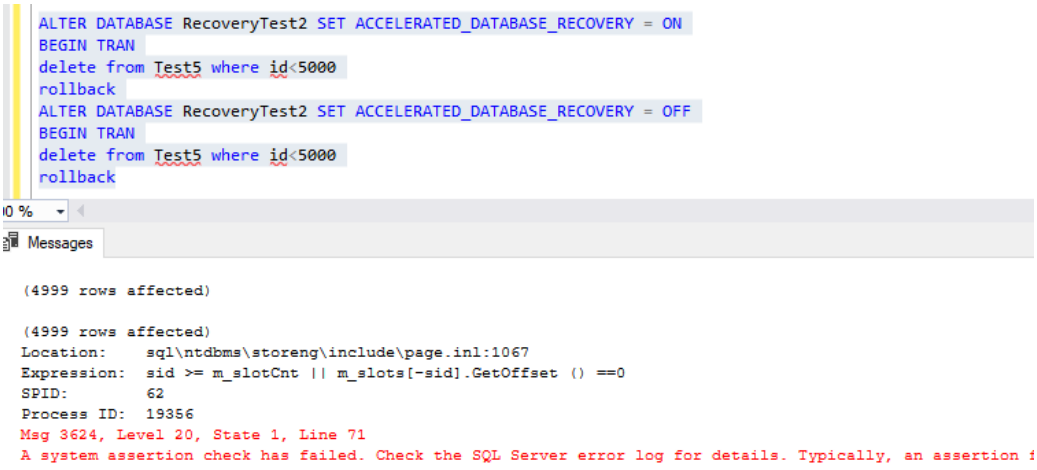

Рис. 11. Последовательность команд, приводящая к возникновению ошибки

После возникновения ошибки база данных оказывается в режиме Suspected, обращение к ней невозможно. Ошибка может вызвать переполнение оперативной памяти после перезагрузки MS SQL сервера, который попытается загрузить в нее все файлы базы данных, которые могут превышать ее размер из-за чего может начать использоваться файл подкачки, что может замедлить работу системы, либо для других приложений не останется памяти. Причина заключается в том, что часть данных переносится из одной страницы на другую, и на предыдущую записывается ссылка на новую, однако записывается сразу несколько ссылок на одну и ту же запись. Содержимое страницы файла базы данных после возникновения ошибки показано на рисунке 12.

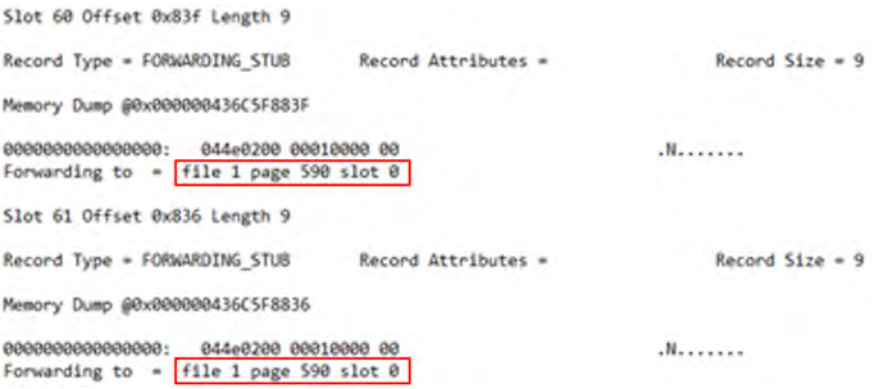

Рис. 12. Содержимое страницы файла базы данных после ошибки

Как видно из рисунка обе 60 и 61 записи ссылаются на одну и туже запись с данными на странице 590. Для устранения ошибки база данных может быть восстановлена только из резервной копии. Средствами SQL-сервера по устранению ошибок базы данных исправить данную ошибку не получилось.

Выводы

Классификация данных полезна при изучении неизвестной базы данных с большим числом столбцов. Возможно создать пользовательские правила классификации. Но имеет несовершенный механизм работы. Функция классификации замедляет время обработки запросов до 17%, увеличивает объем метаданных, при этом не дает много больше информации аудита, возникает уязвимость обнаружения конфиденциальных данных по меткам. Поэтому необходимо заменять GUID типа информации в исходных и пользовательских шаблонах, а при отключении классификации и аудита вручную удалять метки из метаданных столбцов. Ускоренное восстановление полезно, при частых сбоях или отменах длительных транзакций, когда критически важно быстро восстановить работу системы, однако, его использование может оказаться неоправданным в случаях, когда происходят частые удаления строк. Возможна утечка конфиденциальной информации через не удаленные полные копии строк в хранилище версий. Была обнаружена ошибка в работе данной функции, приводящая к нарушению целостности базы данных.

СПИСОК ЛИТЕРАТУРЫ

1. Gorman K., Hirt A., Noderer D. Introducing Microsoft SQL Server 2019. – Packt Publishing, Birmingham SQL. – С. 46-51.

2. Документация по SQL [Электронный ресурс]: sys.sensitivity_ classifications (Transact-SQL) – URL: https://docs.microsoft.com/ru-ru/sql/relational-databases/system-catalog-views/sys-sensitivity-classifications-transact-sql (дата обращения: 15.04.2021).

3. Документация по SQL [Электронный ресурс]: Ускоренное восстановление базы данных – URL: https://docs.microsoft.com/ru-ru/sql/relational-databases/accelerated-database-recovery-concepts (дата обращения: 30.04.2021).

4. Gill F. Accelerated Database Recovery – The Future of the Transaction Log [Электронный ресурс]. – URL: https://www.concurrency.com/blog/may-2019/accelerated-database-recovery-the-future-of-the (дата обращения: 30.03.2021).