Математические структуры и моделирование
2016. № 1(37). С. 103–111

УДК 004.056.5,004.658

ПРОБЛЕМА ГАРАНТИРОВАННОГО УНИЧТОЖЕНИЯ ИНФОРМАЦИИ В БАЗАХ ДАННЫХ: ПОДХОД НА ОСНОВЕ РАБОТЫ С ФАЙЛАМИ ДАННЫХ

Ю.В. Гольчевский
доцент, к.ф.-м.н., e-mail: yurag@syktsu.ru
В.Н. Тропников
студент, e-mail: tropnikovvn@syktsu.ru

Институт точных наук и информационных технологий, ФГБОУ ВО «Сыктывкарский государственный университет имени Питирима Сорокина»

Аннотация. Проведённое исследование посвящено изучению вопроса безопасности уничтожения информации ограниченного доступа в базах данных. Описаны причины появления остаточной информации при выполнении некоторых операций с данными. Представлены способы ее гарантированного уничтожения в базах данных. Опробован метод гарантированного уничтожения на основе прямой работы с файлами данных.

Ключевые слова: гарантированное уничтожение информации, база данных, остаточная информация, информационная безопасность.

Введение

Практически вся информация пользователей, обрабатываемая с помощью средств вычислительной техники, хранится на различных внешних накопителях. Регулярно файлы изменяются и удаляются. Однако с помощью специального программного обеспечения (ПО) или оборудования можно полностью или частично восстановить удалённые данные. Среди них может быть и информация ограниченного доступа. Восстановление возможно в силу того, что на носителе находится информация, оставшаяся от формально удалённых данных.

Проблема гарантированного уничтожения информации (ГУИ) обычно решается с помощью средств защиты информации (СЗИ). Например, для автоматизированных систем имеются требования по обеспечению ГУИ [1]. В разных странах используются различные методы и стандарты обеспечения ГУИ. Примерами могут служить российский ГОСТ и американские стандарты DoD 5220.22-M, NAVSO P-5239-26 и NIST 800-88 R1 [2–5], подходы, применяемые в Канаде и Австралии [6, 7].

При этом, преимущественно, речь идёт об удалении остаточной информации от целых файлов, а не частей файлов, как, например, кортежей в базах данных (БД). Особенность в том, что при удалении записей в БД происходит не удаление файла, а изменение, что не позволяет СЗИ затереть удаляемую

информацию в соответствии с требованиями руководящих документов. Существует ряд исследований в этой области, включая исследования концепции защищённых хранилищ данных, например [8–10], но проблема остаётся до сих пор открытой.

1. Особенности удаления информации в базах данных

Далее под БД будем понимать физическое отражение информации, содержащейся на носителе информации в файле данных. Под термином «страница базы данных» — сегмент носителя, на котором хранится БД. Чтобы произвести операции чтения или записи в БД, необходимо обратиться к соответствующей странице, которая содержит в себе множество кортежей исследуемой БД.

Особенность хранения информации в БД под управлением наиболее распространённых систем управления базами данных (СУБД) заключается в том, что они состоят из файлов двух типов: файлов данных и файлов журналов, в которых хранятся транзакции БД, что позволяет восстановить базу на определённый момент времени [8].

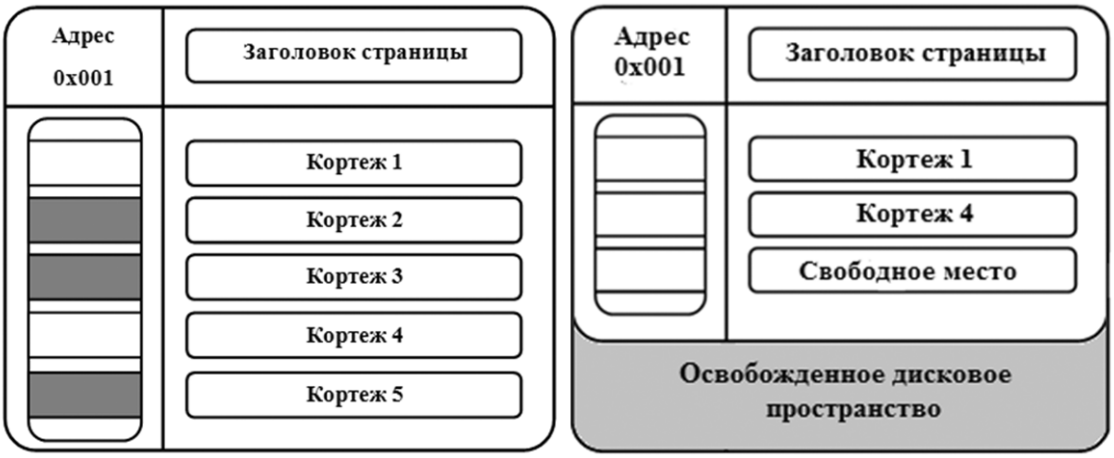
Из анализа команд SQL следует, что ГУИ требуется не только при удалении записей в БД, но и при выполнении ряда других операций, например, оптимизации и обновления.

В некоторых СУБД удаление кортежа подразумевает установку флага удаления, при этом данные на носителе не изменяются. Флагом удаления считают особую последовательность битов, сигнализирующую о том, что кортеж удалён. Схема происходящих изменений приведена в работе [8] на рисунке 1b. Подобные удалённые кортежи далее будем называть «остаточной информацией». Анализ такого поведения для четырёх СУБД приведён в работе [11].

При обновлении записи таблицы возможны два варианта. В первом старый кортеж перезаписывается новым, во втором — старый помечается на удаление, а новый записывается на другое место на носителе. Схемы обновления кортежей приведены на рисунках 1c и 1d в [8]. В обоих случаях возможно появление остаточной информации. Чаще в СУБД используется второй метод обновления. Среди причин — необходимость обеспечения одновременного доступа к БД нескольким пользователям.

Применение команды VACUUM (или аналогичной) позволяет оптимизировать освобождённое в результате работы команд удаления и обновления пространство на носителе информации. В результате её выполнения кортежи БД реорганизуются так, чтобы занимать минимально возможное число страниц. При этом кортежи, помеченные на удаление, перезаписываются. В результате дисковое пространство с освобождёнными страницами БД возвращается файловой системе. На рисунке 1а приведён пример страницы БД до выполнения команды оптимизации, а на 1б — после её выполнения.

Проблема ГУИ при выполнении данной операции заключается в том, что освобождаемое дисковое пространство может содержать старые страницы БД, хотя действия разных СУБД в этой ситуации несколько отличаются. В результате имеется возможность полного или частичного восстановления старых данных, имевшихся в БД.



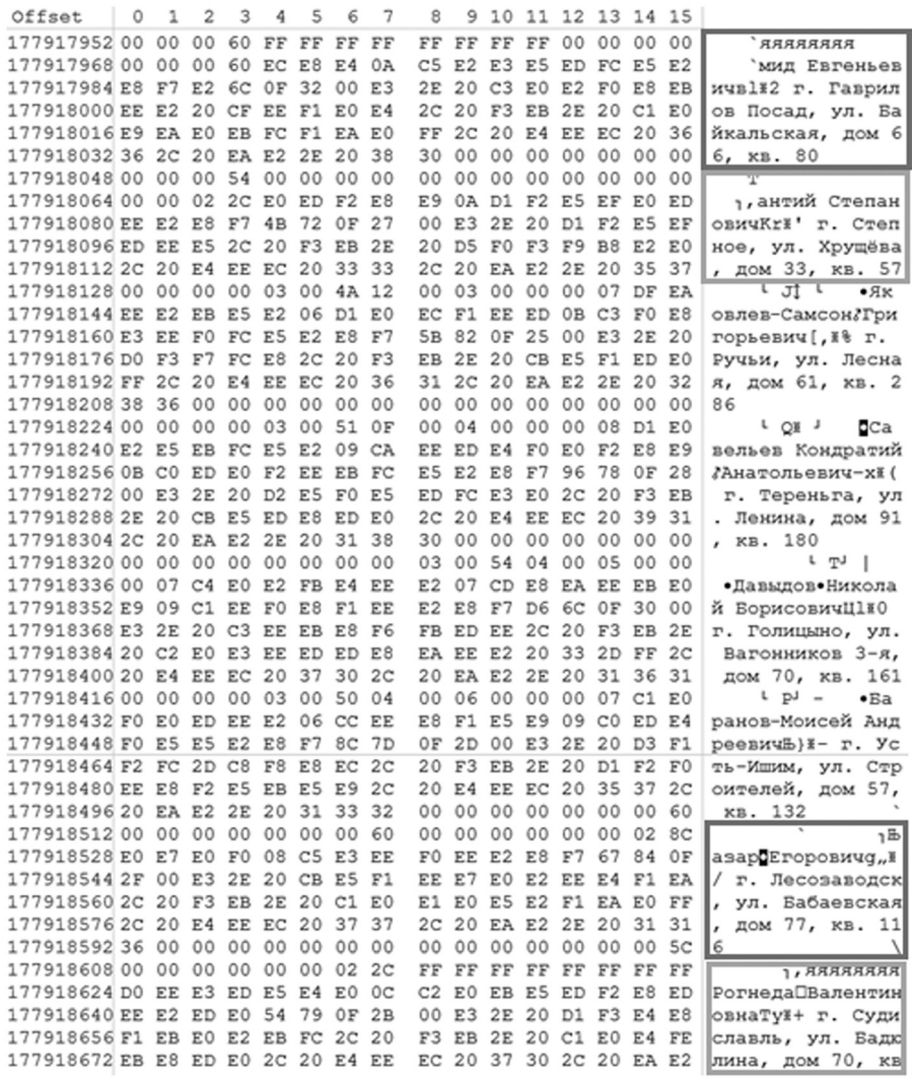
а б

Рис. 1. Страница БД до (а) и после (б) выполнения оптимизации

Продемонстрируем сказанное выше на примере использования БД в СУБД MySQL. Создадим БД с именем *Test*, которая содержит таблицу *Peoples*. Выполним операции, приводящие к возникновению остаточной информации. Например, сначала удалим некоторые четыре записи. Представление таблицы на жёстком диске после такого удаления отражено на рисунке 2. Прямоугольниками обведена удалённая информация. Видно, что фактически данные не удаляются с носителя. В заголовок записи заносится флаг удаления, исходная информация, хранящаяся в записи, практически полностью сохранилась. При этом не составит труда рассчитать критический объем информации, для которого возникающие потери практически не чувствительны.

Также была смоделирована ситуация, когда одна запись заменяется записью большего объёма, другая — меньшего. Такая процедура была повторена несколько раз. Анализ данных на носителе после этого показал, что в случае обновления записи на запись меньшего объёма может получаться остаточная информация. В случае обновления записи более длинной — получали фрагментацию записи. Действия были повторены для разных таблиц. На рисунке 3 приведён один пример представления записи таблицы до обновления и после. Прямоугольниками выделены исходная запись и запись, на которую её заменили. Видно, что при обновлении записи часть информации осталась без изменений.

Далее моделировалась ситуация выполнения оптимизации таблиц. Из таблицы были удалены несколько записей, после чего была выполнена оптимизация. Анализ результатов показал, что происходило перемещение неудалённых записей и сокращалось место на носителе информации, отводимое под БД. При этом удалённые записи физически перезаписывались. Степень гарантии невозможности восстановления зависит от числа таких перезаписей. Если данные уничтожаются методом однократной перезаписи (что обычно и происходит), то такой способ не удовлетворяет требованиям по обеспечению ГУИ в автомати-

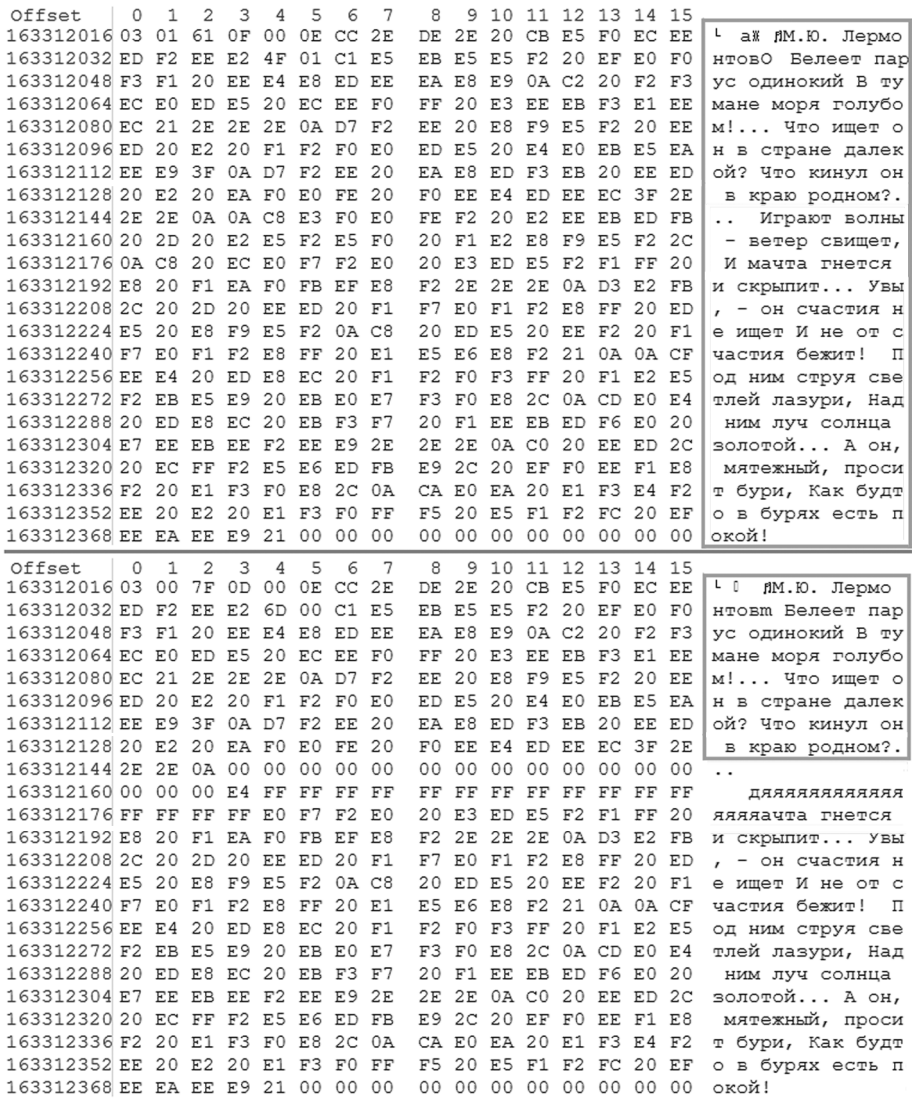


Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
177917952	00	00	00	60	FF	FF	FF	FF	FF	FF	FF	FF	00	00	00	00	`яяяяяяяя
177917968	00	00	00	60	EC	E8	E4	0A	C5	E2	E3	E5	ED	FC	E5	E2	`мид Евгеньев
177917984	E8	F7	E2	6C	0F	32	00	E3	2E	20	C3	E0	E2	F0	E8	EB	ичвl‼2 г. Гаврил
177918000	EE	E2	20	CF	EE	F1	E0	E4	2C	20	F3	EB	2E	20	C1	E0	ов Посад, ул. Ба
177918016	E9	EA	E0	EB	FC	F1	EA	E0	FF	2C	20	E4	EE	EC	20	36	йкальская, дом 6
177918032	36	2C	20	EA	E2	2E	20	38	30	00	00	00	00	00	00	00	6, кв. 80
177918048	00	00	00	54	00	00	00	00	00	00	00	00	00	00	00	00	T
177918064	00	00	02	2C	E0	ED	F2	E8	E9	0A	D1	F2	E5	EF	E0	ED	),антий Степан
177918080	EE	E2	E8	F7	4B	72	0F	27	00	E3	2E	20	D1	F2	E5	EF	овичKr‼' г. Степ
177918096	ED	EE	E5	2C	20	F3	EB	2E	20	D5	F0	F3	F9	B8	E2	E0	ное, ул. Хрущёва
177918112	2C	20	E4	EE	EC	20	33	33	2C	20	EA	E2	2E	20	35	37	, дом 33, кв. 57
177918128	00	00	00	00	03	00	4A	12	00	03	00	00	00	07	DF	EA	┴ J┤ ┴ •Як
177918144	EE	E2	EB	E5	E2	06	D1	E0	EC	F1	EE	ED	0B	C3	F0	E8	овлев-Самсон‽Гри
177918160	E3	EE	F0	FC	E5	E2	E8	F7	5B	82	0F	25	00	E3	2E	20	горьевич[,‼% г.
177918176	D0	F3	F7	FC	E8	2C	20	F3	EB	2E	20	CB	E5	F1	ED	E0	Ручьи, ул. Лесна
177918192	FF	2C	20	E4	EE	EC	20	36	31	2C	20	EA	E2	2E	20	32	я, дом 61, кв. 2
177918208	38	36	00	00	00	00	00	00	00	00	00	00	00	00	00	00	86
177918224	00	00	00	00	03	00	51	0F	00	04	00	00	00	08	D1	E0	┴ Q‼ ┘ ◘Са
177918240	E2	E5	EB	FC	E5	E2	09	CA	EE	ED	E4	F0	E0	F2	E8	E9	вельев Кондратий
177918256	0B	C0	ED	E0	F2	EE	EB	FC	E5	E2	E8	F7	96	78	0F	28	‽Анатольевич-x‼(
177918272	00	E3	2E	20	D2	E5	F0	E5	ED	FC	E3	E0	2C	20	F3	EB	г. Тереньга, ул
177918288	2E	20	CB	E5	ED	E8	ED	E0	2C	20	E4	EE	EC	20	39	31	. Ленина, дом 91
177918304	2C	20	EA	E2	2E	20	31	38	30	00	00	00	00	00	00	00	, кв. 180
177918320	00	00	00	00	00	00	00	00	03	00	54	04	00	05	00	00	┴ T┘ |
177918336	00	07	C4	E0	E2	FB	E4	EE	E2	07	CD	E8	EA	EE	EB	E0	•Давыдов•Никола
177918352	E9	09	C1	EE	F0	E8	F1	EE	E2	E8	F7	D6	6C	0F	30	00	й БорисовичЦl‼0
177918368	E3	2E	20	C3	EE	EB	E8	F6	FB	ED	EE	2C	20	F3	EB	2E	г. Голицыно, ул.
177918384	20	C2	E0	E3	EE	ED	ED	E8	EA	EE	E2	20	33	2D	FF	2C	Вагонников 3-я,
177918400	20	E4	EE	EC	20	37	30	2C	20	EA	E2	2E	20	31	36	31	дом 70, кв. 161
177918416	00	00	00	00	03	00	50	04	00	06	00	00	00	07	C1	E0	┴ P┘ - •Ба
177918432	F0	E0	ED	EE	E2	06	CC	EE	E8	F1	E5	E9	09	C0	ED	E4	ранов-Моисей Анд
177918448	F0	E5	E5	E2	E8	F7	8C	7D	0F	2D	00	E3	2E	20	D3	F1	реевичЬ)‼- г. Ус
177918464	F2	FC	2D	C8	F8	E8	EC	2C	20	F3	EB	2E	20	D1	F2	F0	ть-Ишим, ул. Стр
177918480	EE	E8	F2	E5	EB	E5	E9	2C	20	E4	EE	EC	20	35	37	2C	оителей, дом 57,
177918496	20	EA	E2	2E	20	31	33	32	00	00	00	00	00	00	00	60	кв. 132 `
177918512	00	00	00	00	00	00	00	60	00	00	00	00	00	00	02	8C	`),Ь
177918528	E0	E7	E0	F0	08	C5	E3	EE	F0	EE	E2	E8	F7	67	84	0F	азар◘Егоровичg„‼
177918544	2F	00	E3	2E	20	CB	E5	F1	EE	E7	E0	E2	EE	E4	F1	EA	/ г. Лесозаводск
177918560	2C	20	F3	EB	2E	20	C1	E0	E1	E0	E5	E2	F1	EA	E0	FF	, ул. Бабаевская
177918576	2C	20	E4	EE	EC	20	37	37	2C	20	EA	E2	2E	20	31	31	, дом 77, кв. 11
177918592	36	00	00	00	00	00	00	00	00	00	00	00	00	00	00	5C	6 \
177918608	00	00	00	00	00	00	02	2C	FF	FF	FF	FF	FF	FF	FF	FF	),яяяяяяяя
177918624	D0	EE	E3	ED	E5	E4	E0	0C	C2	E0	EB	E5	ED	F2	E8	ED	РогнедаВалентин
177918640	EE	E2	ED	E0	54	79	0F	2B	00	E3	2E	20	D1	F3	E4	E8	овнаTy‼+ г. Суди
177918656	F1	EB	E0	E2	EB	FC	2C	20	F3	EB	2E	20	C1	E0	E4	FE	славль, ул. Бадю
177918672	EB	E8	ED	E0	2C	20	E4	EE	EC	20	37	30	2C	20	EA	E2	лина, дом 70, кв

Рис. 2. Представление таблицы на носителе после удаления записей

зированных системах, предъявляемым в Российской Федерации для некоторых классов.

Исходя из анализа существующих работ, такая ситуация характерна и для других СУБД. Таким образом, исследованные операции обновления и удаления кортежей не являются безопасными, поскольку приводят к возникновению остаточной информации. Команду оптимизации однозначно назвать безопасной нельзя, поскольку используемый метод затирания не соответствует требованиям нормативных документов.



Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
163312016	03	01	61	0F	00	0E	CC	2E	DE	2E	20	CB	E5	F0	EC	EE	╘ a¥ ╠М.Ю. Лермо
163312032	ED	F2	EE	E2	4F	01	C1	E5	EB	E5	E5	F2	20	EF	E0	F0	нтовО Белеет пар
163312048	F3	F1	20	EE	E4	E8	ED	EE	EA	E8	E9	0A	C2	20	F2	F3	ус одинокий В ту
163312064	EC	E0	ED	E5	20	EC	EE	F0	FF	20	E3	EE	EB	F3	E1	EE	мане моря голубо
163312080	EC	21	2E	2E	2E	0A	D7	F2	EE	20	E8	F9	E5	F2	20	EE	м!... Что ищет о
163312096	ED	20	E2	20	F1	F2	F0	E0	ED	E5	20	E4	E0	EB	E5	EA	н в стране далек
163312112	EE	E9	3F	0A	D7	F2	EE	20	EA	E8	ED	F3	EB	20	EE	ED	ой? Что кинул он
163312128	20	E2	20	EA	F0	E0	FE	20	F0	EE	E4	ED	EE	EC	3F	2E	в краю родном?.
163312144	2E	2E	0A	0A	C8	E3	F0	E0	FE	F2	20	E2	EE	EB	ED	FB	.. Играют волны
163312160	20	2D	20	E2	E5	F2	E5	F0	20	F1	E2	E8	F9	E5	F2	2C	- ветер свищет,
163312176	0A	C8	20	EC	E0	F7	F2	E0	20	E3	ED	E5	F2	F1	FF	20	И мачта гнется
163312192	E8	20	F1	EA	F0	FB	EF	E8	F2	2E	2E	2E	0A	D3	E2	FB	и скрыпит... Увы
163312208	2C	20	2D	20	EE	ED	20	F1	F7	E0	F1	F2	E8	FF	20	ED	, - он счастия н
163312224	E5	20	E8	F9	E5	F2	0A	C8	20	ED	E5	20	EE	F2	20	F1	е ищет И не от с
163312240	F7	E0	F1	F2	E8	FF	20	E1	E5	E6	E8	F2	21	0A	0A	CF	частия бежит! П
163312256	EE	E4	20	ED	E8	EC	20	F1	F2	F0	F3	FF	20	F1	E2	E5	од ним струя све
163312272	F2	EB	E5	E9	20	EB	E0	E7	F3	F0	E8	2C	0A	CD	E0	E4	тлей лазури, Над
163312288	20	ED	E8	EC	20	EB	F3	F7	20	F1	EE	EB	ED	F6	E0	20	ним луч солнца
163312304	E7	EE	EB	EE	F2	EE	E9	2E	2E	2E	0A	C0	20	EE	ED	2C	золотой... А он,
163312320	20	EC	FF	F2	E5	E6	ED	FB	E9	2C	20	EF	F0	EE	F1	E8	мятежный, проси
163312336	F2	20	E1	F3	F0	E8	2C	0A	CA	E0	EA	20	E1	F3	E4	F2	т бури, Как будт
163312352	EE	20	E2	20	E1	F3	F0	FF	F5	20	E5	F1	F2	FC	20	EF	о в бурях есть п
163312368	EE	EA	EE	E9	21	00	00	00	00	00	00	00	00	00	00	00	окой!

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
163312016	03	00	7F	0D	00	0E	CC	2E	DE	2E	20	CB	E5	F0	EC	EE	╘ ⌂ ╠М.Ю. Лермо
163312032	ED	F2	EE	E2	6D	00	C1	E5	EB	E5	E5	F2	20	EF	E0	F0	нтовm Белеет пар
163312048	F3	F1	20	EE	E4	E8	ED	EE	EA	E8	E9	0A	C2	20	F2	F3	ус одинокий В ту
163312064	EC	E0	ED	E5	20	EC	EE	F0	FF	20	E3	EE	EB	F3	E1	EE	мане моря голубо
163312080	EC	21	2E	2E	2E	0A	D7	F2	EE	20	E8	F9	E5	F2	20	EE	м!... Что ищет о
163312096	ED	20	E2	20	F1	F2	F0	E0	ED	E5	20	E4	E0	EB	E5	EA	н в стране далек
163312112	EE	E9	3F	0A	D7	F2	EE	20	EA	E8	ED	F3	EB	20	EE	ED	ой? Что кинул он
163312128	20	E2	20	EA	F0	E0	FE	20	F0	EE	E4	ED	EE	EC	3F	2E	в краю родном?.
163312144	2E	2E	0A	00	00	00	00	00	00	00	00	00	00	00	00	00	..
163312160	00	00	00	E4	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	дяяяяяяяяяяяя
163312176	FF	FF	FF	FF	E0	F7	F2	E0	20	E3	ED	E5	F2	F1	FF	20	яяяячта гнется
163312192	E8	20	F1	EA	F0	FB	EF	E8	F2	2E	2E	2E	0A	D3	E2	FB	и скрыпит... Увы
163312208	2C	20	2D	20	EE	ED	20	F1	F7	E0	F1	F2	E8	FF	20	ED	, - он счастия н
163312224	E5	20	E8	F9	E5	F2	0A	C8	20	ED	E5	20	EE	F2	20	F1	е ищет И не от с
163312240	F7	E0	F1	F2	E8	FF	20	E1	E5	E6	E8	F2	21	0A	0A	CF	частия бежит! П
163312256	EE	E4	20	ED	E8	EC	20	F1	F2	F0	F3	FF	20	F1	E2	E5	од ним струя све
163312272	F2	EB	E5	E9	20	EB	E0	E7	F3	F0	E8	2C	0A	CD	E0	E4	тлей лазури, Над
163312288	20	ED	E8	EC	20	EB	F3	F7	20	F1	EE	EB	ED	F6	E0	20	ним луч солнца
163312304	E7	EE	EB	EE	F2	EE	E9	2E	2E	2E	0A	C0	20	EE	ED	2C	золотой... А он,
163312320	20	EC	FF	F2	E5	E6	ED	FB	E9	2C	20	EF	F0	EE	F1	E8	мятежный, проси
163312336	F2	20	E1	F3	F0	E8	2C	0A	CA	E0	EA	20	E1	F3	E4	F2	т бури, Как будт
163312352	EE	20	E2	20	E1	F3	F0	FF	F5	20	E5	F1	F2	FC	20	EF	о в бурях есть п
163312368	EE	EA	EE	E9	21	00	00	00	00	00	00	00	00	00	00	00	окой!

Рис. 3. Представление записи на носителе до и после обновления

2. Требования к гарантированному уничтожению информации

В Российской Федерации к автоматизированным системам, производящим обработку информации ограниченного доступа, предъявляется ряд требований по защите, в том числе и требования по ГУИ. Согласно [1] для автоматизированных систем некоторых классов очистка осуществляется путём однократной, а для других — двукратной записи в любую освобождаемую область, использованную для хранения защищаемой информации. Эти требования обычно реализуются с помощью СЗИ, например, таких как Secret Net и Dallas Lock, СГУ-2, программы TERRIER.

Чтобы обеспечить ГУИ в БД, необходимо выполнить два условия. Первое — не должна появляться остаточная информация на страницах БД, второе —

не должна появляться остаточная информация в файловой системе. Поскольку перечисленные выше программные средства не предоставляет возможности затирания внутри файла, то реализация требований только с их помощью невозможна.

При совместном использовании средств СУБД и СЗИ можно достичь желаемого результата. Например, после выполнения операций, при которых появляется остаточная информация, можно выполнять команду оптимизации БД, а затем затирать свободное пространство с помощью СЗИ. Однако, во-первых, этот метод нельзя считать безопасным, так как операция оптимизации таблиц БД может не соответствовать требованиям по ГУИ в Российской Федерации, а во-вторых, в это время теряется доступ к БД.

3. Обеспечение гарантированного уничтожения остаточной информации в базах данных

Для обеспечения ГУИ в БД можно предложить два способа. Первый — расширение функционала СУБД. Для этого необходимо изменить работу команд СУБД, в результате выполнения которых может появляться остаточная информация. Второй — создание программы, которая будет работать непосредственно с файлами таблиц и затирать в них остаточную информацию согласно предъявляемым требованиям.

3.1. Метод расширения функционала СУБД

Возможные методы расширения функционала СУБД достаточно подробно описаны в работе [8]. Во-первых, требуется реализовать интерфейс определения пользователем настроек безопасности БД, в том числе и настроек стирания данных. Для этого требуется расширить функционал команды создания таблицы. Для определения используемых циклов перезаписи можно ввести команды, описанные в работе [8]. Во-вторых, необходимо реализовать алгоритм перезаписи данных для команд, приводящих к возможности появления остаточной информации, а также выбор требуемого алгоритма пользователем. В-третьих, необходимо определить процесс выбора пользователем создания обычной или защищённой таблицы, к которой будут применяться описанные алгоритмы затирания удаляемой информации.

Главное преимущество реализации ГУИ данным способом — доступность информации, содержащейся в БД. Затирание остаточной информации происходит сразу, нет необходимости останавливать работу СУБД и проводить дополнительное обслуживание. Основной недостаток — значительное падение производительности СУБД при выполнении многократной перезаписи.

3.2. Метод работы с файлами данных

При непосредственной работе с файлами данных уничтожение остаточной информации в БД производится не средствами СУБД, а сторонней программой.

В ходе исследования была разработана программа, осуществляющая очистку файлов данных таблиц MySQL. На рисунке 4 приведена блок-схема алгоритма.

Сначала считываются номера секторов, которые занимает файл таблицы, затем выполняется чтение файла. Данные первого сектора, принадлежащего файлу, заносятся в кэш программы и производится поиск остаточной информации. Если она обнаружена — производится затирание в соответствии с выбранным алгоритмом, иначе в кэш загружается следующий сектор, в котором расположен файл. Алгоритм возвращается к третьему шагу. Это повторяется, пока не будут проверены все секторы файла таблицы.

Остановимся на нескольких проблемах, возникающих при таком подходе. Основную сложность составляет алгоритм поиска остаточной информации. Возникает проблема — если запись таблицы располагается более чем в одном секторе, возникает необходимость считывать её заголовок, чтобы это обнаружить. Иначе очистка остаточной информации будет произведена неправильно и БД будет повреждена. Вторая проблема, требующая проработки, — отслеживание дискового пространства, освобождаемого при оптимизации таблиц БД. Ещё одна особенность состоит в том, что при прямой работе с файлами данных могут появляться расхождения с журналами транзакций. Восстановление состояния БД в соответствии с журналами будет невозможно. То есть для последующей корректной работы пользователя с БД недостаточно только изменять сами данные, но и возникает необходимость соответствующим образом модифицировать и файлы журналов.

В отличие от предыдущего, данный метод не приводит к падению производительности СУБД. Также положительной стороной является ситуативность его использования. Основной недостаток — временная потеря доступа к данным. Стоит заметить, что при работе с большой БД очистка может занять достаточно продолжительное время. Также недостатком можно считать и невозможность корректной работы пользователя с БД при отсутствии корректировки журналов транзакций.

Анализ результатов работы данной программы позволяет сделать вывод, что такой метод может эффективно применяться для уничтожения остаточной информации.

Заключение

Описанные методы гарантированного уничтожения остаточной информации отличаются противоположностью своих достоинств и недостатков. Первый метод увеличивает время выполнения запросов удаления и обновления БД, при этом к информации всегда имеется доступ. Во втором — не происходит ухудшения производительности, но для обеспечения безопасности БД требуется приостанавливать доступ к данным. Каждый из методов может применяться для обеспечения ГУИ, выбор зависит от условий эксплуатации БД.

На практике протестирован метод ГУИ на основе прямой работы с файлами данных при помощи разработанной программы. Полученный результат показал,

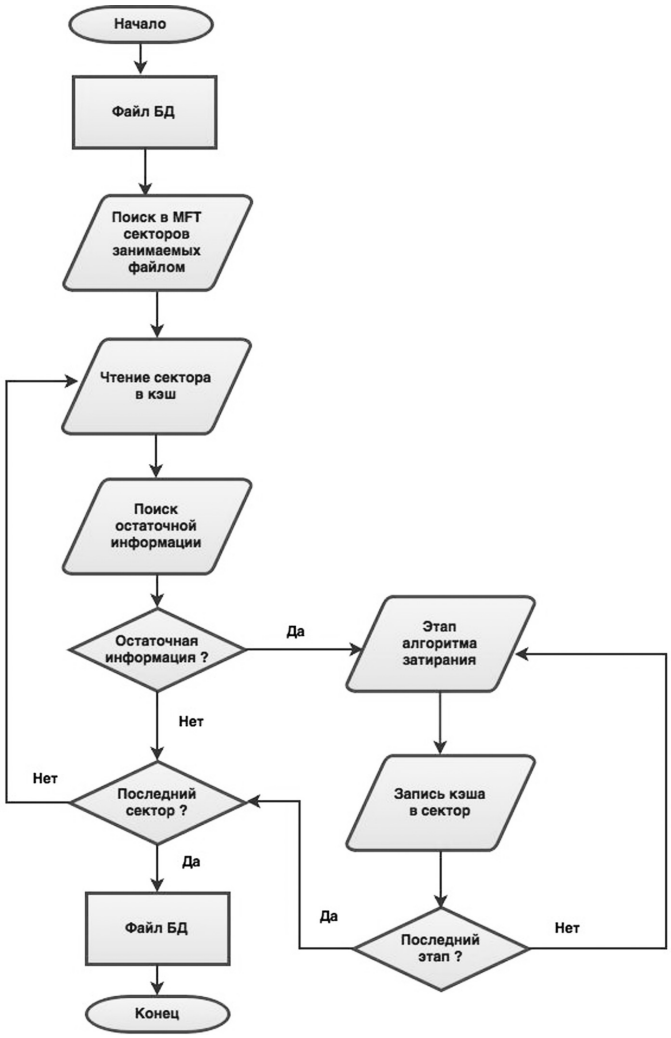


Рис. 4. Блок-схема реализованного алгоритма ГУИ

что его можно использовать для уничтожения остаточной информации.

Хотелось бы отметить, что данный вопрос необходимо решать и на организационном уровне. Например, стоит предъявить необходимые требования к разработчикам СУБД по обеспечению ГУИ, либо доработать требования к соответствующим СЗИ.

Литература

1. Руководящий документ. Автоматизированные системы. Защита от несанкционированного доступа к информации. Классификация автоматизированных систем и требования по защите информации (утв. решением Гостехкомиссии России от 30.03.1992). URL: `http://base.consultant.ru/cons/cgi/online.cgi?req=doc;base=EXP;n=575668` (дата обращения: 28.05.2015).
2. ГОСТ Р 50739-95. «Средства вычислительной техники. Защита от несанкционированного доступа к информации. Общие технические требования». Портал Фе-

дерального агентства по техническому регулированию и метрологии. URL: http://protect.gost.ru/document.aspx?control=7&id=134268 (дата обращения: 28.05.2015).
3. DoD 5220.22-M. URL: http://www.fas.org/sgp/library/nispom/nispom2006.pdf (дата обращения: 28.05.2015).
4. US Naval Information Systems Management Center: NAVSO P-5239-26. URL: http://fas.org/irp/doddir/navy/5239_26.htm (дата обращения: 28.05.2015).
5. NIST Special Publication 800-88 Revision 1. Guidelines for Media Sanitization. URL: http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-88r1.pdf (дата обращения: 28.05.2015).
6. Government of Canada. IT Security Guidance. Clearing And Declassifying Electronic Data Storage Devices (ITSG-06). URL: https://www.cse-cst.gc.ca/en/system/files/pdf_documents/itsg06-eng.pdf (дата обращения: 28.05.2015).
7. Australian Government ICT Security Manual 2014 – Controls. URL: http://www.asd.gov.au/publications/Information_Security_Manual_2014_Controls.pdf (дата обращения: 28.05.2015).
8. Grebhahn A., Schäler M., Köppen V. Secure Deletion: Towards Tailor-Made Privacy in Database Systems. URL: http://www.iti.cs.uni-magdeburg.de/iti_db/publikationen/ps/auto/GSK13.pdf (дата обращения: 11.06.2015).
9. Miklau G., Neil Levine B., Stahlberg P. Securing history: Privacy and accountability in database systems // 3rd Biennial Conference on Innovative Data Systems Research (CIDR), Asilomar, 2007. P. 387–396.
10. Diesburg S.M., Wang A.A. A Survey Of Confidential Data Storage And Deletion Methods // ACM Computing Surveys. 2010. Vol. 43, № 1. P. 2:1–2:27.
11. Stahlberg P. Threats to Privacy in the Forensic Analysis of Database Systems. UMassAmherst Center for Forensics. URL: http://forensics.umass.edu/pubs/stahlberg07forensicDB.pdf (дата обращения: 07.06.2015).

THE PROBLEM OF SECURE DELETION IN DATABASES: THE METHOD BASED ON THE DIRECT OPERATION WITH DATA FILES

Yu.V. Golchevskiy
Ph.D. (Phys.-Math.), Associate Professor, e-mail: yurag@syktsu.ru
V.N. Tropnikov
Student, e-mail: tropnikovvn@syktsu.ru

Syktyvkar State University n.a. Pitirim Sorokin

Abstract. The problem of secure deletion of confidential information in databases was investigated. The causes of the residual information appearance as a result of some operations with data were described. The ways of data secure deletion in databases were presented. The method of secure deletion on the basis of direct operation with data files was tested.

Keywords: secure deletion, database, residual information, information security.