

# ПРОТОКОЛИРОВАНИЕ ДЕЙСТВИЙ ПОЛЬЗОВАТЕЛЕЙ СРЕДСТВАМИ МОДУЛЯ «БАСТИОН-2 – АУДИТ»

А.Н. Крутов

Самарский государственный аэрокосмический университет имени академика С.П. Королёва (национальный исследовательский университет) (СГАУ), Самара, Россия

В работе представлено описание принципиальных схем решения задачи по протоколированию действий пользователей в информационных системах, а также решения, реализованные в модуле «Бастион-2 – Аудит».

**Ключевые слова:** протоколирование, базы данных, персональные данные.

## Введение

Необходимость анализа проводимых пользователем действий является неотъемлемой частью эксплуатации практически любой информационной системы. Особенно это характерно для систем, в которых хранятся персональные данные работников, так как в этом случае в дополнение к существующим требованиям к системе, сформулированные заказчиком, добавляются требования федерального закона № 152 о персональных данных. Как известно, в этом законе одним из обязательных условий эксплуатации подобных систем регламентируется необходимость ведения истории изменения персональных данных.

## 1. Краткие сведения об АПК «Бастион-2»

Модуль «Бастион-2 – Аудит» является составной частью аппаратно-программного комплекса «Бастион-2». Данный аппаратно-программный комплекс позволяет объединить системы безопасности различных производителей в области видеонаблюдения, контроля доступа и пожарной сигнализации в единую систему, удобным образом настраиваемую и надежно функционирующую. По своей сути аппаратно-программный комплекс «Бастион-2» является качественным развитием АПК «Бастион» с дополнительными возможностями. К ним относятся поддержка СУБД Oracle, существенные доработки ядра системы и поддержка большего спектра устройств и систем безопасности. Кроме этого, АПК «Бастион-2» разрабатывался как масштабируемый продукт, способный к стабильной работе на объектах разного масштаба – от небольших офисов до крупных предприятий, с развитой филиальной сетью. Все вышесказанное послужило основанием для создания модуля «Бастион-2 – Аудит», основной функцией которого являлось бы ведение истории изменений основных справочников системы с выводом соответствующих отчетов.

## 2. Принципиальные схемы построения системы протоколирования

При разработке принципиальной схемы решения задачи о протоколировании действий пользователя, как правило, принимают один из трех нижеописанных способов решения задачи:

1. При трехзвенной организации приложения модуль протоколирования действий располагается на сервере приложений [1].
2. Протоколирование действий пользователя использует журнал транзакций базы данных [2].
3. Протоколирование действий осуществляется с использованием дополнительных таблиц, триггеров и хранимых процедур [3].

При разработке модуля «Бастион-2 – Аудит» первый способ решения был неприменим, т.к. большинство модулей системы построены по двухзвенной архитектуре. Кроме этого, реализация модуля протоколирования на среднем звене способна нарушить целостность критически важной для бизнеса информации, что делает необходимым аудит изменений на уровне базы данных [1]. Второй способ решения задачи так же не мог быть применим, т.к. при этом следует хранить журнал транзакций за весь период эксплуатации системы, что делает практически невозможным ее использование на бесплатных версиях СУБД Oracle. Все вышеперечисленное обосновывает выбор именно третьего способа реализации модуля протоколирования. Так, для хранения исторических данных создаются дополнительные таблицы, данные в которые попадают автоматически с помощью триггеров при выполнении DML-операций. С помощью специально разработанных хранимых процедур данные из этих таблиц обрабатываются и выводятся в отчеты пользователей в наиболее удобном виде. Для минимизации задержек при выполнении DML операций триггеры по поддержке историчности не содержат какого-либо анализа или предварительной обработки данных, а просто копируют данные в таблицы-протоколы. Полный анализ данных с указанием характера проведенных изменений над объектами данных осуществляется в модуле «Бастион-2 – Аудит» непосредственно при формировании отчетов. При этом весь анализ данных реализован на уровне базы данных в виде пакета хранимых процедур.

### 3. Сохранение исторических данных и формирование отчетов в модуле «Бастион-2 – Аудит»

Для протоколирования изменений для каждой основной таблицы %TABLE\_NAME% создается отдельная лог-таблица %TABLENAME%\_HISTORY. Данные в исторические таблицы заносятся триггерами при выполнении операций изменения и удаления данных. Ниже в таблицах 1 и 2 представлены структуры основных и соответствующих исторических таблиц модуля.

Табл. 1. Структура основной таблицы (Table1)

| Название поля        | Тип   | Краткое описание                           |
|----------------------|-------|--|
| <b>Table1_ID</b>     | Число | Первичный ключ                             |
| <b>TableFields</b>   |       | Информационные поля                        |
| <b>Ver_StartDate</b> | Дата  | Дата/время начала действия версии записи   |
| <b>OperNo</b>        | Число | Ссылка на оператора                        |
| <b>OperationMode</b> | Число | Код операции, приведшей к изменению данных |
| <b>RecVersion</b>    | Число | Номер версии записи                        |

Табл. 2. Структура историчной таблицы (Table1\_history)

| Название поля            | Тип   | Краткое описание                            |
|--------------------------|-------|---|
| <b>Table1_history_ID</b> | Число | Первичный ключ                              |
| <b>TableFields_Saved</b> |       | Информационные поля                         |
| <b>Ver_StartDate</b>     | Дата  | Дата/время начала действия версии записи    |
| <b>Ver_EndDate</b>       | Дата  | Дата/время окончания действия версии записи |
| <b>OperNo</b>            | Число | Ссылка на оператора                         |
| <b>OperationMode</b>     | Число | Код операции, приведшей к изменению данных  |
| <b>RecVersion</b>        | Число | Номер версии записи                         |

В представленных выше таблицах значение атрибута OperationMode может принимать следующие возможные значения – ручной ввод (1), импорт данных (2), сканирование документа (3) и репликация (4). При удалении записи из основной таблицы ее последняя версия переносится в историчную таблицу, а из основной удаляется. Для формирования отчетов по изменениям данных используется представление V\_Table1\_Full вида:

```

select TableFields, RecVersion, Ver_StartDate,
       sysdate Ver_EndDate
from Table1
union all
select TableFields_Saved, RecVersion, Ver_StartDate, Ver_EndDate
from Table1_history

```

В процессе непосредственного формирования отчета в соответствующей хранимой процедуре проводится последовательный анализ всех произведенных изменений с основными таблицами системы и выявляются поля, в которых произошли изменения. Вся полученная при этом информация сохраняется во временных таблицах для последующего отображения в приложении.

В результате проведенной работы был разработан модуль, использующий лишь стандартные средства баз данных и способный, при необходимости, с минимальными доработками работать практически на любой современной клиент-серверной СУБД.

## Литература

1. Лашманов, А., Слепенков, М. Аудит по журналу транзакций // Открытые системы. СУБД. – 2012. – №1. – С. 40. – ISSN 1028-7493.
2. Китаев, Е.Л., Кузьмичев, Д.Л., Слепенков М.И., Богданова, В.М. Мониторинг изменений в базах данных на основе анализа транзакций. // Препринты ИПМ им. М.В. Келдыша – 2012. – № 2. – С. 1-14. – ISSN 2071-2898.
3. Прилипко, А.Г. Хранение истории изменения данных в реляционной базе данных // Труды научно-исследовательского института системных исследований Российской академии наук. – 2012. – Т.2., № 1. – С. 67-72. – ISSN 2225-7349.