

Для выбора оптической головки, наиболее подходящей для построения виброметра, был проведен сравнительный анализ конструкций трех типов головок: KRS-202A (Sony), KSS-572A (Sony) и SF-P151 (Sanyo). В головке KRS-202A для переключения между режимами чтения и записи используется контроллер лазерного диода, что усложняет электронную схему головки. Так как в виброметре достаточно использовать лазерное излучение низкой интенсивности, то предпочтительным является применение головок, предназначенных для работы только в режиме чтения и не имеющих контроллера лазерного диода. Этим условиям удовлетворяют головки KSS-572A и SF-P151. Простота электронной схемы головки имеет большое значение по той причине, что многие используемые в головках интегральные микросхемы не имеют подробного технического описания. Дополнительным достоинством головок, работающих только в режиме чтения, является снижение риска повреждения зрения излучением лазерного диода. В качестве фотодетекторов в головках KRS-202A и SF-P151 используются PDIC-модули (PDIC = PhotoDetectorIntegratedCircuit), содержащие интегрированные операционные усилители сигналов с фотодиодов, что позволяет исключить необходимость использования внешних усилителей. В оптико-электронной схеме головки KSS-572A используются всего два элемента: фокусирующая линза и LDH-модуль (LDH = Laser-Detector-Hologram). LDH-модуль представляет собой оптико-электронную микросхему, содержащую лазерный диод, зеркало, фотодетекторы и голографический оптический элемент (дифракционную решетку). В головке SF-P151 используются отдельно установленные лазерный диод и PDIC-модуль. Окончательно в качестве наиболее подходящей для построения виброметра была выбрана головка SF-P151, имеющая наиболее простую электрическую схему включения.

М.Ю. Бокий (УО «ГГТУ им. П.О. Сухого», Гомель)

Науч. рук. **Е.Г. Стародубцев**, канд. физ.-мат. наук, доцент

АЛГОРИТМ ОТСЛЕЖИВАНИЯ ИЗМЕНЕНИЙ ДАННЫХ В УНИВЕРСАЛЬНОМ ХРАНИЛИЩЕ ФИНАНСОВОЙ ОТЧЕТНОСТИ

Пользователи корпоративного хранилища данных нуждаются в отслеживании изменений значений некоторых атрибутов. Разработчик универсального хранилища должен реализовать возможность ведения истории для таких атрибутов. Такая функциональность предоставляется интерфейсами консолидации финансовой отчетности.

Сохранение истории данных выполняется путем создания нескольких записей для одного натурального ключа, разделяя старые и новые записи с помощью суррогатных ключей, ключей версий и наборов колонок аудита.

Для реализации процесса перезаписи одних атрибутов и сохранения истории изменения других необходимо сгенерировать хэш-код на основании определенного набора атрибутов, который затем может быть использован для идентификации обновленных колонок.

Набор колонок аудита будет использован для отслеживания сделанных изменений (см. таблицу).

Таблица – Набор колонок для реализации процесса

Тип колонки		Комментарий
Surrogate Key		Суррогатный ключ
Version Key		Значение возрастает на единицу с каждым обновлением, является частью первичного ключа таблицы.
Natural Key		Натуральный ключ
NonHistory		Атрибуты, для которых не нужно сохранять историю
History		Атрибуты, для которых нужно сохранять историю
Колонки аудита	Current Indicator	Текущий индикатор. Если это текущая версия записи - 'Y' иначе 'N'
	Source System Name	Название системы-источника, из которой данные поступили
	Record Checksum Code	Хэш-код на основе натурального ключа и History атрибутов
	Row Status Code	Статус записи: вставка = 'I', обновление = 'U', удаление = 'D'
Колонки аудита	Insert Timestamp	Дата\время (timestamp) вставки записи
	Update Timestamp	Timestamp обновления записи
	Effective Date	Действующая дата: текущая дата при вставке записи.
	Expired Date	Дата истечения. Значение по умолчанию – '31-12-9999'. При вставке новой версии для старой устанавливается текущая дата.

Алгоритм процесса представлен на рисунке 1.

Значения колонок аудита для каждого действия:

1 Вставка записи в таблицу:

Current Indicator = 'Y'

Row Status Code = 'I'

Checksum Code = сгенерированный хэш-код

Insert Timestamp = текущий timestamp

Update Timestamp = текущий timestamp

Effective Date = текущая дата

ExpiredDate = '31-12-9999'

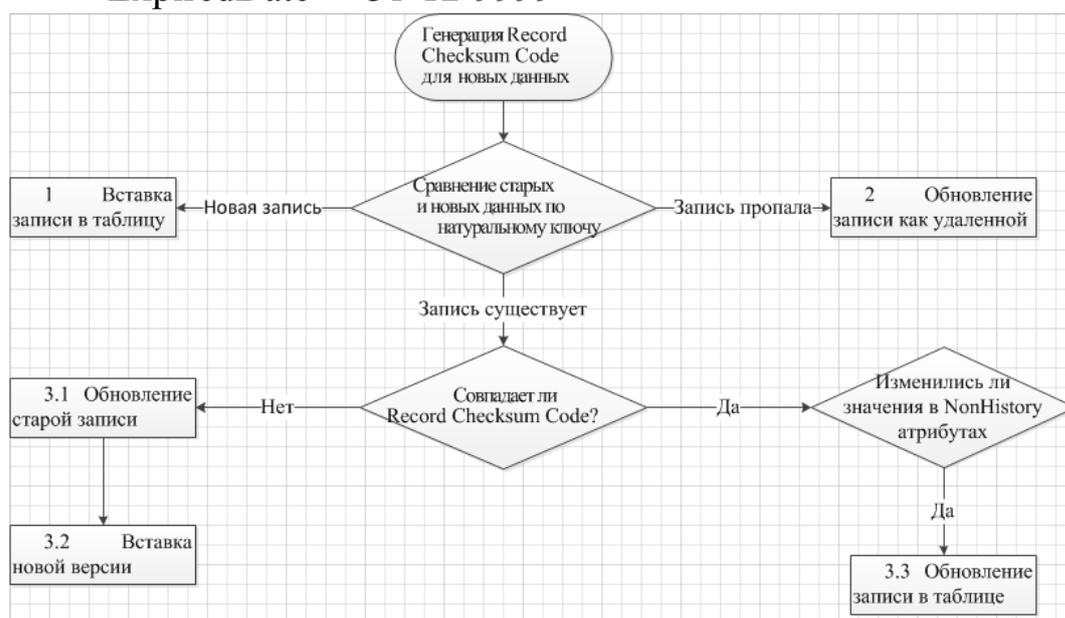


Рисунок 1 – Алгоритм отслеживания изменений данных

2 Обновление записи как удаленной:

Current Indicator = 'N'

Row Status Code = 'D'

Update Timestamp = текущий timestamp

ExpiredDate = текущая дата

3.1 Обновление старой записи:

Current Indicator = 'N'

Row Status Code = 'U'

Update Timestamp = текущий timestamp

ExpiredDate = текущая дата

3.2 Вставка новой версии:

Version key = version key старой записи + 1

Current Indicator = 'Y'

Row Status Code = 'I'

Checksum Code = сгенерированный checksum код

Update Timestamp = текущий timestamp

Insert Timestamp = текущий timestamp

Effective Date = текущая дата

ExpiredDate = '31-12-9999'

3.3 Обновление записи в таблице:

Применить изменения для обновления записи

Row Staus Code = 'U'

Update Timestamp = Текущий timestamp

Таким образом, предлагаемый алгоритм позволяет сохранять историю изменений значений атрибутов там, где это важно для аналитика, и в то же время не тратить ресурсы и не создавать излишних, ненужных данных там, где это не требуется. Использование контрольной суммы ускорит работу приложения, так как для каждой записи достаточно сравнивать только ее вместо множества атрибутов, требующих ведения истории. Набор колонок аудита предоставит аналитику исчерпывающую информацию, а также может быть использован для фильтрации данных различными интерфейсами.

И.А. Борихин (УО БрГТУ, Брест)

Науч. рук. **В.И. Хвещук**, канд. тех. наук, доцент

О МОДЕЛИРОВАНИИ ДЕЯТЕЛЬНОСТИ ИТ-СПЕЦИАЛИСТОВ

Одной из важных особенностей современных ИТ- изделий является то, что их производство носит единичный характер. Это требует от производителей каждый раз по-новому организовывать производственный цикл (подбор исполнителей, разработка модели производства, выбор средств для реализации и другие компоненты). В данной работе предложен модельный подход к организации планирования производства ИТ- изделий, который основан на использовании совокупности следующих моделей: модель ИТ- изделия; модель проекта; модель производства; модель ИТ – специалиста. Основные положения данного подхода следующие:

- Модель ИТ- изделия. Предложена типовая системная архитектура изделий. Она основана на классификации следующих компонентов: предметных областей для разработки изделий, видов разработки, архитектур изделий, элементов изделий.

- Модель ИТ- производства. Она представляет собой набор моделей технических процессов, необходимых для реализации заданного класса ИТ- изделий. Для каждого процесса определяется примерный набор работ и задач, а также спектр возможных средств для их реализации.