В. А. Макухо

# АЛГОРИТМЫ ОБРАБОТКИ МЕДЛЕННО ИЗМЕНЯЮЩИХСЯ ДАННЫХ

*В работе рассматриваются причины использования алгоритмов обработки медленно изменяющихся данных и необходимости учета версионирования записей, производится сравнение подходов к поддержанию историчности на заданном временном промежутке. Делаются выводы о преимуществах использования вспомогательных технических колонок в используемой таблице.*

## Введение

Актуальность учета медленно изменяющихся данных заключается в необходимости обращения к истории изменения определенных атрибутов конкретных записей в объектах хранилища данных. Медленно изменяющиеся данные представлены в виде измерений, которые изменяются с течением времени относительно определенных атрибутов объекта.

Другими словами, реализация одного из типов медленно изменяющихся измерений должна позволить пользователям назначать правильное значение атрибута измерения на заданную дату. Принцип работы с подобными записями основывается на предварительном определении необходимого уровня историчности заданной сущности, с использованием ряда технических атрибутов, позволяющих отслеживать версионность записи.

В зависимости от уровня историчности определяется тип и наличие вспомогательных технических атрибутов.

При построении схемы по типу звезда, в хранилище данных таблицы измерений объединяются с таблицей фактов. Примером может служить информация о сотрудниках. Измерения позволяют отслеживать изменение данных сотрудника, используя различные атрибуты, такие как регион, город, почтовый индекс, должность, адрес, номер телефона, фамилия. Данные атрибуты объекта подразумевают возможное изменение с течением времени. Для решения проблемы учета изменения атрибутов существуют распространенные типы медленно изменяющихся измерений, которые могут быть реализованы при проектировании таблицы измерений в хранилище данных.

## I. Алгоритм обработки медленно изменяющихся измерений первого типа

Методология первого типа используется, когда нет необходимости хранить утратившие актуальность данные в таблице измерений. Этот метод заключается в перезаписи устаревших данных в таблице измерений новыми данными и используется, например, для поддержания записей в актуальном состоянии, или же для исправления ошибок данных в измерении. В отличие от второго типа, который допускает возможность отслеживать версии строк при изменении атрибутов, в данном типе при изменении значений для актуальной записи, актуальная запись помечается как устаревшая, при этом открывается новая запись, содержащая измененную актуальную информацию. Иными словами, после обновления записи последовательность выполненных изменений отследить невозможно. Представленный алгоритм обработки медленно изменяющихся данных первого типа подразумевает обеспечение хранения исключительно актуальных записей.

Метод первого типа используется, когда нет необходимости хранить исторические данные в таблице измерений путем замены устаревших данных актуальными в таблице измерений.

Процесс внедрения алгоритма обработки медленно изменяющихся данных первого типа включает в себя идентификацию новой записи и ее загрузка в таблицу измерений, выявление измененной записи и обновление таблицы.

## II. Алгоритм обработки медленно изменяющихся измерений второго типа

Алгоритм обработки медленно изменяющихся данных второго типа заключается в добавлении дополнительной записи. Когда значение в колонке, относительно которой отслеживается историчность записи, изменяется, предыдущая запись помечается как неактивная путем изменения значения столбца, предназначенного для отслеживания активной записи.

Каждая запись также содержит дату загрузки и конечную дату актуальности записи, с использованием которых можно вычислить период активности данной конкретной записи. Измерения подобного вида следует использовать в случае ожидания частых изменений в хранящихся данных, а также при необходимости построения аналитики с учетом изменения значений атрибутов.

В случае, когда конечная дата актуальности записи не определена, принято использовать максимально возможное значение даты для удобства последующих вычислений. Значение даты окончания устаревшей записи совпадает с датой начала действия актуальной записи.

Удобство использования алгоритма обработки медленно изменяющихся данных второго типа заключается в хранении полной истории версий, а также обеспечении необходимого доступа к данным заданного периода времени.

Хранение исторических данных позволяет производить создание исторических отчетов относительно заданных периодов времени, для корректной работы которых необходимо обращение не только к актуальным, но также и устаревшие данные для отчетов за более ранний период.

В качестве источника данных для подобных отчетов может быть использована таблица, содержащая актуальные, а также утратившие актуальность записи.

Данный алгоритм обработки данных позволяет использовать исторические данные в целевой таблице для построения исторических отчетов. Данные отчеты позволяют вести учет необходимых показателей, вычисления которых опираются на данные из созданного измерения, что позволяет отследить запись, актуальную для определенного периода, за который необходимо произвести расчёт или отследить изменения.

Рассматриваемый подход обеспечивает высокую точность расчётов необходимых показателей относительно требований. Преимущество использования данного подхода заключается в возможности хранения нескольких устаревших записей. Пример реализации алгоритма загрузки данных в медленно изменяющееся измерение второго типа приведен на рисунке 1.


Рис. 1 – Пример реализации алгоритма загрузки данных в медленно изменяющееся измерение второго типа

## III. Алгоритм обработки медленно изменяющихся измерений третьего типа

Алгоритм обработки медленно изменяющихся данных третьего типа заключается в добавлении дополнительных столбцов, хранящих предыдущее и текущее значение атрибутов с целью поддержания историчности.

Данный подход к построению таблиц измерений используется при необходимости изменения по конкретным заданным параметрам. В данном случае историчность сохраняется до предыдущего значения.

## IV. Выводы

На основе предоставленных данных можно сделать вывод, что хранение истории изменяющихся атрибутов позволяет производить аналитику историчных данных, что дает преимущество при создании отчетов за определенные периоды времени с учетом истории.

Применение алгоритмов обработки медленно изменяющихся данных позволяет создавать сущности с четко различимыми активными и историчными записями, что, в свою очередь, уменьшает нагрузку на систему при построении отчетов, а также при последующем анализе данных.

Нагрузка на систему снижается за счет уменьшения количества прямых обращений к хранилищу данных при построении элементов отчета, так как актуальная и историческая версии одной записи могут храниться в одной таблице.

### *Список литературы*


1. Kimball R. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling, Third Edition / R. Kimball, M.Ross // Published by John Wiley Sons, Inc., Indianapolis, Indiana Published simultaneously in Canada. – P. 54–59.
2. Create/Design/Implement SCD Type 1 Mapping in Informatica [Electronic resource] / Vijay Bhaskar, 2012. – Mode of access: `https://www.folkstalk.com/2012/03/createdesignimplementscd-type-1.html`. – Date of access: 19.10.2021.
3. Introduction to Slowly Changing Dimensions (SCD) Types [Electronic resource] / Whiteley S., 2014. – Mode of access: `https://https://adatis.co.uk/introductionto-slowly-changing-dimensions-scd-types/`. – Date of access: 19.10.2021.
4. Slowly Changing Dimensions (SCD) in Data Warehouse [Electronic resource] / Vithal S., 2021. – Mode of access: `https://dwgeek.com/slowly-changing-dimensionsscd.html/`. – Date of access: 19.10.2021
5. Slowly Changing Dimensions (SCD) Type 2 Implementation in Oracle Cloud Infrastructure (OCI) Data Integration [Electronic resource] / Duvuri A., 2020. – Mode of access: `https://blogs.oracle.com/dataintegration/post/slowlychanging-dimensions-scd-type-2-implementation-inoracle-cloud-infrastructure-oci-data-integration/`. – Date of access: 19.10.2021.



*Макухо Вероника Анатольевна*, магистрант кафедры информационных технологий автоматизированных систем БГУИР, nika.makuho@gmail.com.

*Научный руководитель: Ломако Александр Викторович*, кандидат технических наук, доцент, lavlot@bsuir.by