

ПРЕИМУЩЕСТВА ИЗВЛЕЧЕНИЯ, ОБРАБОТКИ И ЗАГРУЗКИ ДАННЫХ СРЕДСТВАМИ PLUS 7 FORMIT В ЕДИНЫЕ ХРАНИЛИЩА ДАННЫХ

ADVANTAGES OF EXTRACTING, PROCESSING AND LOADING DATA WITH PLUS 7 FORMIT INTO IN DATA WAREHOUSING

E. Emelyanov

Summary. In IT systems data stored in many source systems. As rule databases, storages, systems on statistics and etc not related to each other. In data contains important information for business management, but to do this it is necessary to obtain data from diverse sources and present it in a form convenient for managers and analysts in the shortest possible time.

The purpose of the study is the efficiency of processing, extracting, and loading data using ETL methods, as well as 7 forms in visibility with the general functionality of external ETL tools.

The main functionality of ETL tools plus 7 formit is used for the transfer, highlighting the elements of the basic functionality, and considering possible shortcomings of foreign solutions in consideration with a visible analogue.

This study should clearly highlight the problems and shortcomings of processing results and loading data using ETL tools plus 7 formit.

Keywords: data warehouse, data base, source system, mapping, structured query language.

Емельянов Егор Гурьевич

аспирант, Технологический университет
имени дважды героя Советского Союза,
летчика-космонавта А.А. Леонова
did-qvi@mail.ru

Аннотация. В информационных системах данные хранятся в разных источниках. Часто это не связанные между собой БД, хранилища событий, системы статистики и т.п. В этих данных есть важная информация для управления бизнесом, но для этого необходимо достать данных из разнородных источников и представить их в удобном для менеджеров и аналитиков виде за минимально возможное время.

Целью исследования является обозначить преимущества обработки, извлечения и загрузки данных российским ETL средством плюс 7 формит в сравнении с общим функционалом зарубежных ETL средств.

Задачей является рассмотреть основной функционал ETL средства плюс 7 формит, выделить элементы основного функционала и указать на явные недостатки зарубежного решения в сравнении с российским аналогом.

Результатом данного исследования должно являться выделения явных преимуществ и нововведений в извлечении, обработке и загрузке данных с помощью российского ETL аналога плюс 7 формит.

Ключевые слова: хранилище данных, база данных, система источник, маппинг, язык структурированных запросов.

Инструменты ETL, часто используемые в сочетании с SQL, являются основополагающими элементами инженерии данных, предназначенными для решения сложных задач управления данными. Эти инструменты могут извлекать данные из многих источников, будь то традиционные реляционные базы данных, системы NoSQL или облачные платформы. Однако их настоящее мастерство проявляется на этапе трансформации. Здесь данные подвергаются тщательной очистке для устранения аномалий, обогащению для повышения их ценности и структурированию, чтобы сделать их пригодными для аналитических целей. Помимо этих основных функций, современные инструменты ETL решают проблемы, связанные с большими данными и аналитикой в реальном времени. Теперь они предлагают возможности потоковой обработки, позволяющие предприятиям обрабатывать данные в режиме реального времени, а также интеграцию машинного обучения для прогнозирования

тенденций и аномалий. Кроме того, с развитием облачных вычислений многие инструменты ETL теперь являются облачными, что обеспечивает масштабируемость, гибкость и экономическую эффективность. Их интеграция с современными решениями для хранения данных обеспечивает предприятиям бесперебойный конвейер данных от извлечения данных до генерации аналитической информации. В более широком контексте хранилищ данных и аналитики инструменты ETL являются не просто помощниками; они являются движущей силой, предоставляющей предприятиям возможность использовать истинный потенциал своих данных.

Современные зарубежные ETL средства обладают следующими преимуществами:

1. Консолидированное представление данных. ETL обеспечивает консолидированное представление данных для углубленного анализа и отчет-

ности. Управление многочисленными наборами данных требует времени и координации и может привести к неэффективности и задержкам. ETL объединяет базы данных и различные формы данных в единое, унифицированное представление. Процесс интеграции данных улучшает качество данных и экономит время, необходимое для перемещения, категоризации или стандартизации данных. Это облегчает анализ, визуализацию и осмысление больших массивов данных.

2. Точный анализ данных. ETL обеспечивает более точный анализ данных для соответствия нормативным и регулятивным стандартам. Вы можете интегрировать инструменты ETL с инструментами обеспечения качества данных для профилирования, аудита и очистки данных, обеспечивая их достоверность.
3. Автоматизация задач. ETL автоматизирует повторяющиеся задачи обработки данных для эффективного анализа. Инструменты ETL автоматизируют процесс миграции данных, и вы можете настроить их на периодическую интеграцию изменений данных или даже во время выполнения. В результате инженеры по обработке данных могут больше времени уделять инновациям и меньше — решению таких утомительных задач, как перемещение и форматирование данных.

Однако несмотря на преимущества, можно выделить и следующие значительные недостатки:

1. Масштабирование. Как было указано выше, перед транспортировкой данных их необходимо обработать. Для этого данные предварительно материализуются и индексируются в кэше. Фактически кэш представляет из себя файл, элементы которого разбиты на массивы значения которого содержат первичный ключ, указатели на дочерний связанный объект и номер главного элемента массива. Такой подход удобен с точки зрения горизонтального расширения т.к. в этом случае будут увеличиваться элементы внутри самого массива, а само кол-во массивов будет статично на последующих этапах обработки данных. Обратная ситуация с вертикальным расширением, при котором увеличивается как кол-во входящих массивов, так и операций по загрузке и выгрузке данных из кэша тем самым значительно увеличивая общий объем памяти выделяемого для кэша, для обработки данных, и приводя к резкому снижению времени раскрытия и обработки элементов внутри каждого массива.
2. Переполнение кэша. Кэш является промежуточной областью хранения данных элементы которого зачищаются, в случае если в дальнейшей обработке их участие исключается. Однако это правило не работает в случае, например, если

связи 1 к 100000000 т.е. 1 элемент массива связан с 100000000 элементами других массивов. Как правило данная проблема решается расширением общего объема кэша путем дополнительных вычислительных ресурсов сервера. В случае если ответ от сервера не был получен или же превышено время ожидания ответа, то трансформации, выполняемые в рамках текущей транзакции, будут отменены с соответствующей ошибкой о переполнении кэша. На практике данную проблему решают путем создания несколько кэшей доступ к которым осуществляется последовательно. Однако такой подход увеличивает время трансформации во много раз т.к. остальные процессы простаивают по причине ожидания завершения основного процесса и выделения ресурса для их работы.

В современном российском решении плюс 7 формит, недостатки зарубежных средств были устранены следующим образом:

1. Масштабирование. Подход с вертикальным расширением масштабируемости был пересмотрен. Внутри каждого блока допустимы только двухуровневые массивы данных при этом процессы, рассматривающие первый и второй уровень вложенности, являются независимыми. Итоги обработки процессов фиксируются в отдельном файле на сервере и подлежат удалению по итогу вставки или обновления новых строк родительским процессом в целевой объект конечной таблицы базы данных приемника. В случае недостаточности в использовании двумерных массивов один из дочерних процессов организует связанный список, в котором выстраивается цепочка из правил обработки и извлечения данных из смежных массивов. Подход с горизонтальным расширением остался почти неизменным. Данные также предварительно материализуются и индексируются в кэше, но доступ на чтение этих данных, и их изменение, имеет только родительский процесс, инициировавший заполнение кэша. Это сделано с целью исключения блокировок объекта на чтение или запись процессами, которые могут просматривать или ожидать его изменения другими родительскими процессами.
2. Переполнение кэша. Общий подход к кэшированию элементов массива, в современных ETL средствах, мог приводить к его переполнению что в свою очередь влияло на простой основных процессов обработки и загрузки данных. Общий подход был переработан и вместо индексации и наполнения кэша был разработан оркестратор, задача которого координировать работы дочерних процессов. Работа оркестратора заключается в том, чтобы распределять допустимые ресурсы

между дочерними процессами тем самым снижая время простоя родительского процесса. Сами данные записываются в промежуточные файлы на сервере и сортируются средствами ETL в необходимой для джобов последовательности. В случае выявления коллизий, или же других конфликтных ситуаций, оркестратор принимает решение по своему усмотрению, но есть функционал по переопределению правил, на основании которых оркестратор принимает решения по разрешению конфликтов.

3. Двухстороннее кэширование. В отличие от зарубежных аналогов, в плюс 7 формит допустимо кэширование не только на стороне источника, но и на стороне базы данных приемника. Это позволяет сократить время трансформаций данных путем обращений к метаданным ETL с даты последней загрузки данных в БД источника.

В сравнении с зарубежными ETL средств, российский аналог имеет следующие преимущества:

1. Возможность захвата данных в режиме реального времени. Обработка данных в режиме реального времени позволяет системе принимать структурированные и неструктурированные данные из диапазона потоковых источников. К ним относятся IoT, подключенные устройства, каналы социальных сетей, данные датчиков и мобильные приложения. Система обмена сообщениями гарантирует, что данные запечатлены точно. Преобразование данных выполняется с использованием механизмов обработки данных в реальном времени.
2. Возможность захвата данных по изменениям. ETL используется для воспроизведения и автоматической синхронизации из различных исходных баз данных в облачное хранилище данных. Чтобы сэкономить время и повысить эффективность, используйте захват данных изменений, чтобы автоматизировать процесс, чтобы обновить только измененные наборы данных.

3. Обновленный подход к извлечению, преобразованию, и загрузки данных. Данные собираются из одного или нескольких источников данных и затем удерживаются во временном хранилище предварительно отсортированными в соответствии с правилами, по которым предполагаются преобразования. Во время извлечения применяются правила проверки на соответствие требованиям конечных типов данных. Данные, которые не выполняют проверку, отклоняются и не загружаются до начала следующего шага. На этапе преобразования данные обрабатываются для того, чтобы его значения и структура соответствовали предполагаемому варианту использования. Цель преобразования состоит в том, чтобы сделать все данные в соответствии с единой схемой, прежде чем она переходит к последнему шагу. Типичные преобразования включают агрегаторы, маскировку данных, выражение, столяру, фильтр, поиск, ранжирование, маршрутизатор, объединение, нормализатор, и веб-сервис. Это помогает нормализовать, стандартизировать и фильтровать данные. Это также делает данные для потребления для аналитики, бизнес-функций и других последующих действий. Наконец, фаза загрузки перемещает преобразованные данные в постоянную целевую систему. Это может быть целевая база данных, хранилище данных, хранилище данных, центр данных или озеро данных-локально, или в облаке. Как только все данные загружены, процесс будет завершен.

Выводы

В рамках данной статьи рассмотрены основные преимущества ETL средства плюс 7 формит в сравнении с современными зарубежными аналогами. По результатам проведенного анализа можно сделать вывод о том, что российским аналог ETL средства превосходит зарубежные аналоги в качестве, быстродействии и надежности при работе с большим количеством данных с различными источниками их обработки.

ЛИТЕРАТУРА

1. Сбор, объединение и преобразование данных с помощью Power Query. 2022. [с. 128–256].
2. Справочная документация по эксплуатации программного обеспечения informatica [Электронный ресурс]. Режим доступа: <https://docs.informatica.com>.
3. Клеппман М.А. Высоконагруженные приложения. Питер, 2022. [с. 350–526].

© Емельянов Егор Гурьевич (did-qvi@mail.ru)

Журнал «Современная наука: актуальные проблемы теории и практики»