34 Subsystem of ETL

«***Pentaho Kettle Solutions. Building Open Source ETL Solution s with Pentaho Data Integration***». Matt Casters. Roland Bouman. Jos van Dongen. WILEY 2010

I. Extraction
   1. Data Profiling System

      Три уровня выполнения Data Profiling:

      A. **Column Profile**: сбор статистических данных на уровне одного столбца;
      - Количество различных значений (Какое количество уникальных значений содержится в столбце)
      - Количество отсутствующих (null) и пустых значений (Сколь много записей, которые не имеют или имеют пустое значение)
      - Максимальное и минимальное значение (не только для численных но и для текстовых значений)
      - Числовые значения суммы, медианы, среднего и стандартного отклонения (Различные расчеты числовых значений и распределение значений)
      - Шаблоны строк и их длина, для проверки правильности хранения значений (Например, почтовые индексы Германии должны содержать пять цифр)
      - Количество слов, количество букв в верхнем и нижнем регистре (Какое кол-во слов в столбце, все слова в верхнем, нижнем или смешанном регистре?)
      - Частота подсчета (Какие верхние и нижние N элементов в столбце?)

      B. **Dependency Profile**: проверка зависимости в таблице между разными столбцами;

      Как пример, можно привести такие зависимости как: Принадлежность города Штату(Округу), принадлежность улице городу, соотношение с индексом.

      C. **Join Profile**: проверка зависимости между разными таблицами.

      Можно ли найти данные требуемые в одной таблице в другой. Каково соотношение всего количества одних данных и использование в других таблицах (как пример, количество клиентов всего и количество клиентов сделавших заказы)

   2. Change Data Capture System

      Подсистема, целью которой является загрузка данных из исходной системы в целевую без необходимости полного копирования всей информации повторно.

      В основном существуют две категории процессов CDC, «**intrusive**» и «**non-intrusive**». Первые могут повлиять на производительность системы из которой производится извлечение данных. 3 из 4 процессов CDC являются intrusive.

      A. Source Data–Based CDC

         Данный CDC полагается на наличие в исходной системе атрибутов, которые позволят процессу выбрать нужные записи.

         a. Наличие хотя бы одной метки Timestamps, но предпочтительнее две: метка времени вставки и времени обновления;
         b. Использование Database sequences, для определения новых вставленных элементов;

         Ограничения:
         - невозможность определить различие между вставкой и обновлением, если только не содержится две отметки timestamps;
         - обнаружение удаленных записей, если только они не были удалены логически;
         - обнаружение множественных изменений, когда запись обновилась несколько раз между текущей и предыдущей датой загрузки;
         - подходит только для пакетной загрузки данных

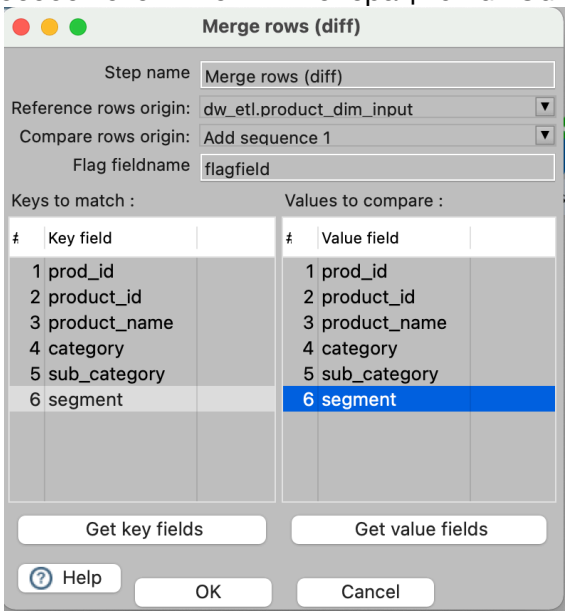
      B. Trigger-Based CDC

         Триггеры баз данных могут использоваться для запуска любого из операторов обработки данных (INSERT, UPDATE, DELETE). Эти триггеры могут сохранять данные изменения в промежуточных таблицах или для помещения данных в Staging область DW. Подобные решения требуют изменения

первоначального кода DB, а также могут серьезно замедлить работу транзакционной системы. Альтернативой использования триггеров в исходной DB является использование их на реплицированных данных. Это требует дополнительного места для хранения но является эффективным и **non-intrusive** поскольку основано на чтении изменений из логов DB.

C. Snapshot-Based CDC

Создание моментальных снимков. Используется когда временные метки, триггеры и репликации недоступны. Для реализации данного CDC необходимо выделить различия между двумя снимками. Один из возможных способов это выполнить операцию Full Outer Join.



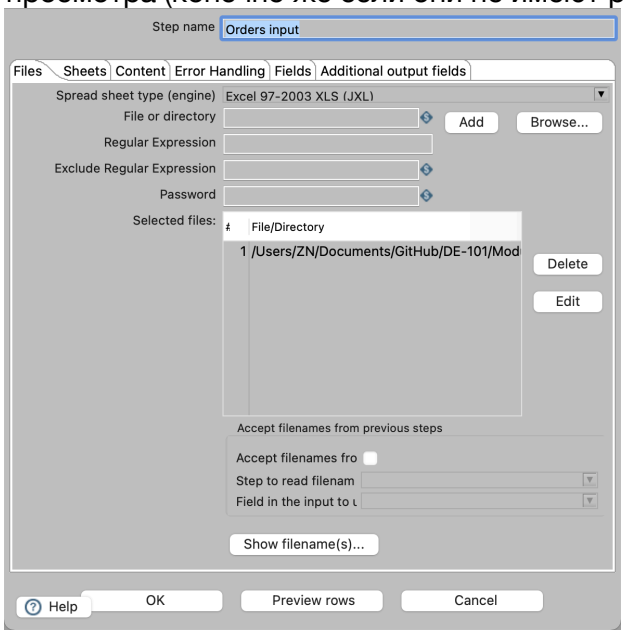
[инструмент «Merge Rows» из Pentaho]

D. Log-Based CDC ◒

3. Extraction System

A. File-Based Extraction

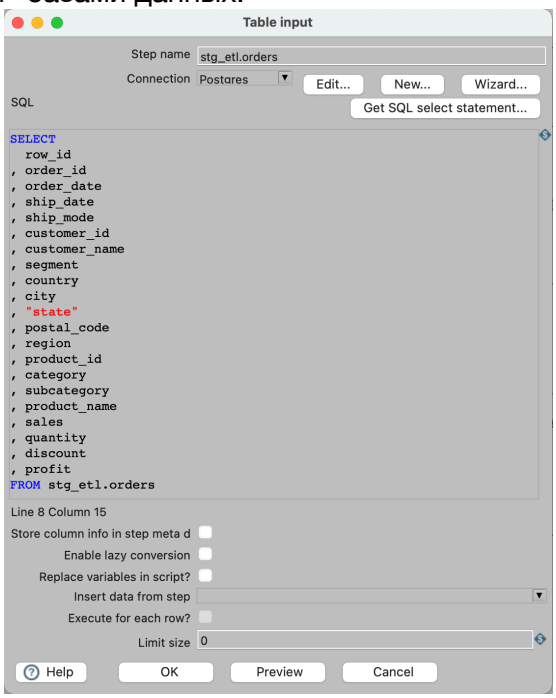
Pentaho DI обладает большим функционалом по извлечению данных из различных файлов. Файлы легко перемещать/передавать, они могут быть эффективно сжаты и любой простой редактор может быть использован для их просмотра (конечно же если они не имеют размер в несколько гигабайт)



[инструмент «Text file input» из Pentaho]

B. Database-Based Extraction

Pentaho DI позволяет взаимодействовать как с реляционными, так и с «no-sql» базами данных.



[инструмент «Table input» из Pentaho]

C. Web-Based Extraction

D. Stream-Based and Real-Time Extraction

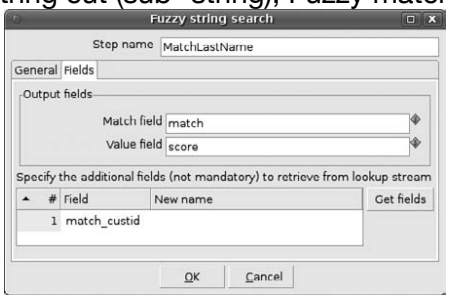
II. Cleaning and Conforming Data

«In 2003, The Data Warehouse Institute (TDWI) estimated that data quality problems cost U.S. businesses $600 billion each year. Things have probably gotten even worse since then.»

4. Data Cleaning and Quality Screen Handler System

Очистка данных это, как правило это не один, а группа шагов, которая начинается еще на этапе извлечения данных.

Стоит избегать преобразования данных на этапе их получения при помощи SQL запроса, т.к. подобное решение порождает сложности в его обслуживании, а также затрудняет выполнения проверки. Лучшем подходом будет загрузить данные «как есть», после чего уже выполнить преобразования имеющимися инструментами. Большинство шагов трансформации позволяет выявить потоки данных имеющих ошибки. Шаги в Pentaho DI предлагают множество различных вариантов очистки данных (Casing calculations, Return/remove digits, Replace in string, String cut (sub- string), Fuzzy match).



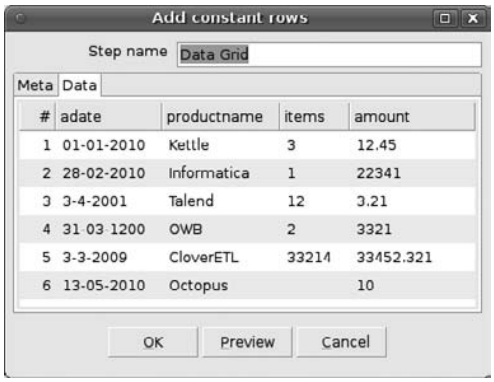
[Инструмент «Fuzzy match» из Pentaho]

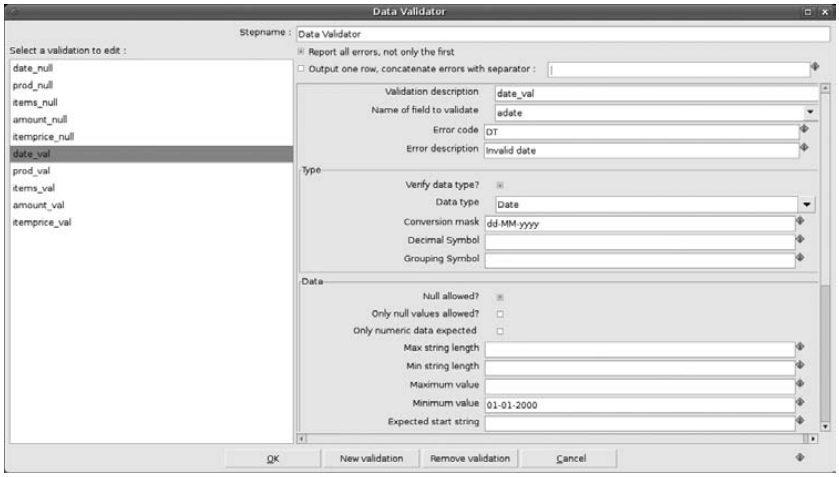
Для проверки качества данных часто используют:

- Проверку по справочным таблицам (Reference Tables)
- Проверку данных на соответствие тому или иному паттерну (формат телефонного номера, индекса, e-mail адреса, номер банковского счета, даты)

На шаге проверки данных в Pentaho DI могут быть выполнены такие функции например как:

- Проверка типов данных
- Объединение ошибок в одно поле с разделителем
- Параметризация значений
- Соответствие регулярным выражениям
- Подстановка значений и т.д.





[«Data Validator step» из Pentaho]

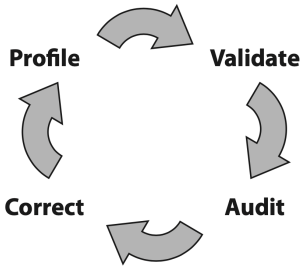
5. Error Event Handler

   При обработке можно выделить следующие классы ошибок

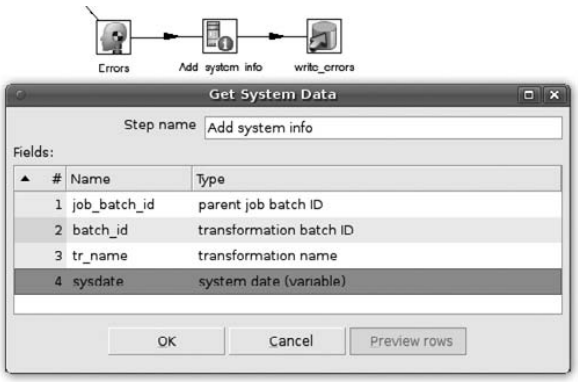
   - Ошибки процесса
   - Ошибки проверки данных
   - Ошибки фильтрации
   - Ошибки общего шага

6. Audit Dimension Assembler

   Жизненный цикл качества данных процесса



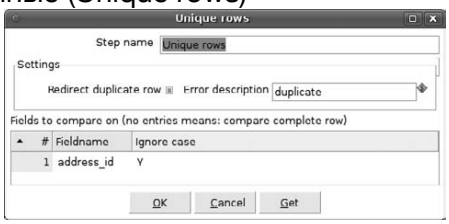
Полученные данные об ошибках могут быть дополнены системной информацией о текущем batch преобразовании при помощи шага «**Get System Info**», что позволит сохранить данные об ошибках в отдельной таблице.



7. Deduplication

Можно выделить два типа дубликатов:

- точные (Unique rows)



[Инструмент «Unique rows» из Pentaho]

- неточные (Unique rows (HashSet))

Нет точного и однозначного алгоритма по выявлению и исправлению неточных дубликатов. Один из множества возможных способов это выполнение проверок на соответствие по другим атрибутам одного и того же объекта, а также поиск и сравнение по нечеткому соответствию.

На устранение дубликатов затрачивается значительное количество вычислительной мощности, т.к. требует выполнение проверки каждой записи ко всем имеющимся.

8. Data Conformer

Система обобщающая результаты подсистем дедупликации и других подсистем качества данных. Цель системы - согласовать все входящие данные на предмет соответствия друг-другу.

III. Data Delivery

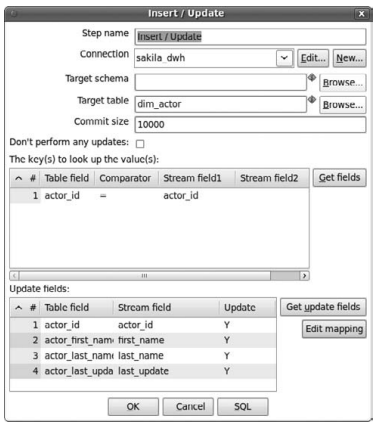
Данные подсистемы решают множества задач добавления записей в целевую систему.

9. Slowly Changing Dimension Processor

Принципы SDC являются одними из основополагающих аспектов для таблиц измерений в DWH. Реализованные принципы SCD позволяют учесть изменения происходящие с объектами DWH в долгосрочном периоде.

Три основных типа SCD:

- Тип 1. Обновление в исходной системе приводит к соответствующему изменению в целевой системе (p.229);

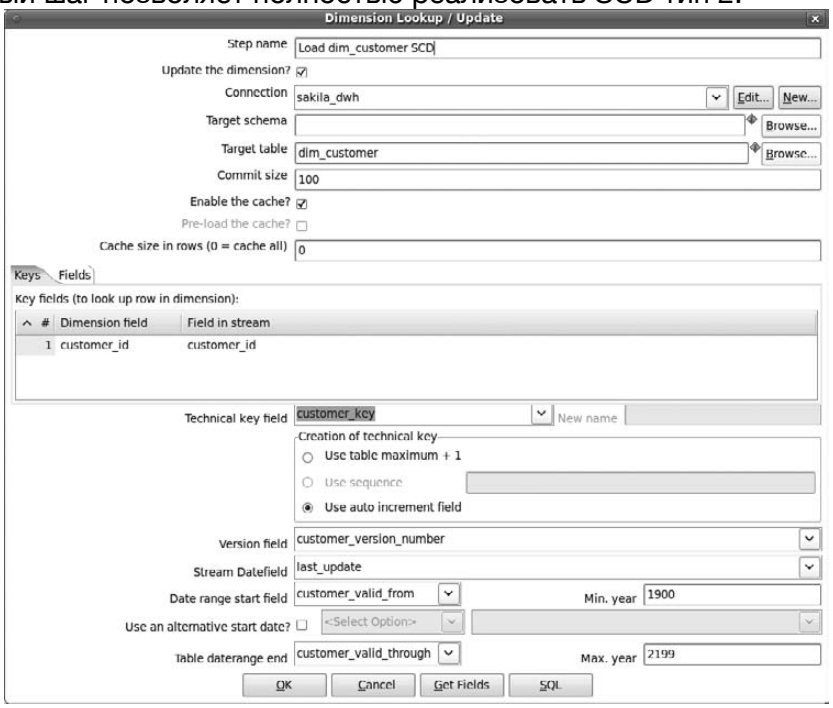


Если нужно только вставить новые строки, то стоит использовать просто шаг [Table output], т.к. он не будет сначала искать необходимую строку.

- Тип 2. Обновление в исходной системе приводит к вставкам в целевую систему нескольких версий строк измерений у каждой из которых есть своя отметка времени (timestamp), что позволяет определить версию строки на тот или иной период времени;

Dimension lookup / update
(p.232 | 280 pdf.)
a. можно использовать для добавления/обновления данных в таблице измерений
b. можно использовать в качестве шага для получения суррогатного ключа. Особенно полезно при загрузке таблиц фактов и называется режимом поиска.

Данный шаг позволяет полностью реализовать SCD тип 2.



[Инструмент «Dimension lookup» из Pentaho]

- Тип 3. Обновление в исходной системе хранится в разных столбцах одной строки.

Pentaho DI не поддерживает напрямую данный шаг, но его можно реализовать при помощи имеющихся шагов. Можно выполнить шаг «Database Lookup» и шаг «Update» сохранив одновременно предыдущее и текущее значение. Для динамического добавления столбцов можно написать «Job» и использовать шаг

«Columns exist in table» проверив нужно ли изменение таблицы и добавив соответствующий скрипт SQL DDL.

10. Surrogate Key Creation System
11. Hierarchy Dimension Builder
12. Special Dimension Builder
13. Fact Table Loader
14. Surrogate Key Pipeline
15. Multi-Valued Dimension Bridge Table Builder
16. Late-Arriving Data Handler
17. Dimension Manager System
18. Fact Table Provider System
19. Aggregate Builder
20. Multidimensional (OLAP) Cube Builder
21. Data Integration Manager

IV. Managing the ETL Environment
    22. Job Scheduler
    23. Backup System
    24. Recovery and Restart System
    25. Version Control System, and Subsystem
    26. Version Migration System from development to test to production
    27. Workflow Monitor
    28. Sort System
    29. Lineage and Dependency Analyzer
    30. Problem Escalation System
    31. Parallelizing/Pipelining System
    32. Security System
    33. Compliance Reporter
    34. Metadata Repository Manager