ETL Workflow Modeling

Abhishek Tiwari

Citation: *A. Tiwari*, ”ETL Workflow Modeling”, Abhishek Tiwari, 2018.
doi:10.59350/2m8sv-krh84

Published on: April 14, 2018

Developing Extract–transform–load (ETL) workflow is a time-consuming activity yet a very important component of data warehousing process. The process to develop ETL workflow is often ad-hoc, complex, trial and error based. It has been suggested that formal modeling of ETL process can alleviate most of these pain points. Generally speaking, formal modeling can reduce implementation time and save money by adopting structural patterns and best-practices when implementing ETL workflows.

Why

Modeling of ETL workflow is important for several reasons. First and foremost, modeling ETL process helps in designing an efficient, robust and evolvable ETL. It enables data warehouse teams to ask questions like how good is the current or proposed ETL workflow design, is the workflow resilient to occasional failures, what part of the workflow can be parallelized, are there any variants of ETL workflow, and if so is one variant is better than other. Second, modeling ETL workflow plays an important role to optimize the data warehousing. When we talk about optimizing the ETL workflow we are mainly concerned with fast and efficient execution plan i.e. the sequence of the ETL operations.

An ETL workflow is responsible for the extraction of data from the source systems, their cleaning, transformation, and loading into the target data warehouse. There are existing formal methods to model the schema of source systems or databases such as entity-relationship diagram (ERD). Similarly, destination data warehouse can follow well-accepted standard data models such as star schema and snowflake schema. For databases, we have a well established relational algebra, but there is no equivalent algebra for ETL workflows. Contrary to source and target areas, models of ETL workflow are still in the fancy stage.

What's out there

Currently, there are few approaches to model the ETL workflow. An ETL workflow can be modeled using (1) mapping expressions and guidelines, (2) conceptual constructs, (3) entity mapping, and (4) UML notations [1][2][3][4]. Out of these approaches, conceptual modeling approaches (2 and 3) hits a good balance between simplicity, usability, and extensibility. These models provide mapping operators. When representing ETL process using these modes, mapping operators can be applied to entities (entity transformation such union, join, intersection, difference, etc.) or attributes of entities (attribute transformations such as add, subtract, data type conversions, etc.).

[1] Data Mapping Diagrams for Data Warehouse Design with UML

[2] Query-based Data Warehousing Tool

[3] Conceptual Modeling for ETL Processes

[4] A comprehensive method for data warehouse design

10.59350/2m8sv-krh84

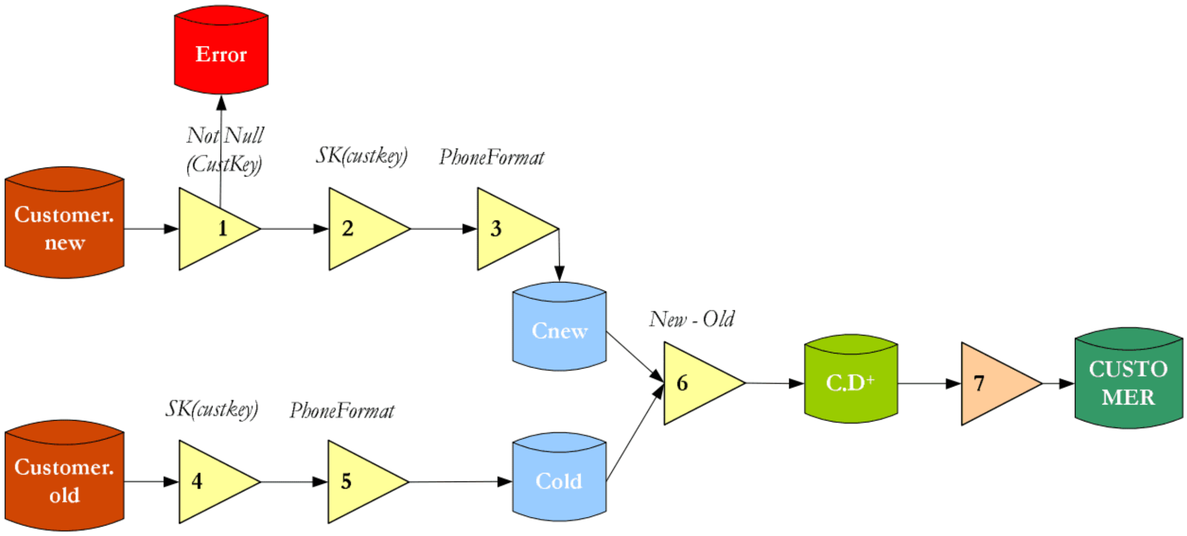

Figure 1: An example ETL process modeled using conceptual constructs. This particular ETL pattern can be classified as wishbone - a variation of butterfly pattern. Image credits Vassiliadis et al.

Butterfly pattern

Often ETL workflows defined by conceptual constructs can be classified as structural patterns. The butterfly pattern is most common pattern observed across a large number ETL and data warehouse implementations. There other patterns such as fan-out, fan-in, wishbone, linear, tree, primary but the can be simply considered as variations or classes of the butterfly.

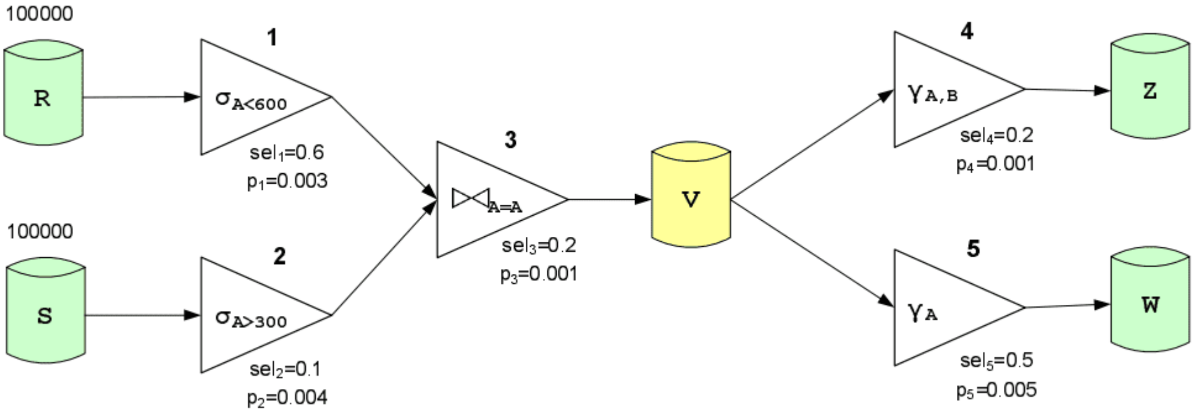

Figure 2: An example of butterfly pattern. Image credits Vassiliadis et al.

Closing thoughts

Although these ETL modeling techniques are designed for traditional relational databases (i.e. source dependent), they can be applied to model the modern ETL workflow and data pipelines. In the upcoming blog post, I will discuss how you can apply conceptual modeling to architect modern data pipelines.