Topofminds

Top of Minds Report series

Using Informatica PowerCenter to automate your Data vault

Summary

With the on going success and sustained rapid growth of the Data Vault modelling standard, the need for robust and easy-to-use code for loading and extracting data is ever increasing. However, many companies struggle with non-reusable ETL solutions that cost too much to maintain and doesn't scale effectively. By the end of this white paper, you will know how and why Informatica PowerCenter is an excellent tool for developing a standardized set of reusable ETL code, ensuring a stable, scalable data warehouse solution that is easy to maintain over time.

Data Vault Overview

The core of the Data Vault modelling technique is the identification of the core business keys, such as Product or Customer. All business keys, and only the business keys, are stored in a specific table structure called a Hub. These business keys have certain relationships, such as that between an Order and a Customer. All business key relations, and only the relations, are stored in another table structure called a Link. All descriptive data, for instance the name, address and telephone of a Customer or the date and shipping information of an order is stored in a third table structure called a Satellite. With this structure, one of the most important features is that changes on data and historical versions of a row are tracked in the satellite, and only in the satellite. This image illustrates a typical small Data Vault model.

About the author

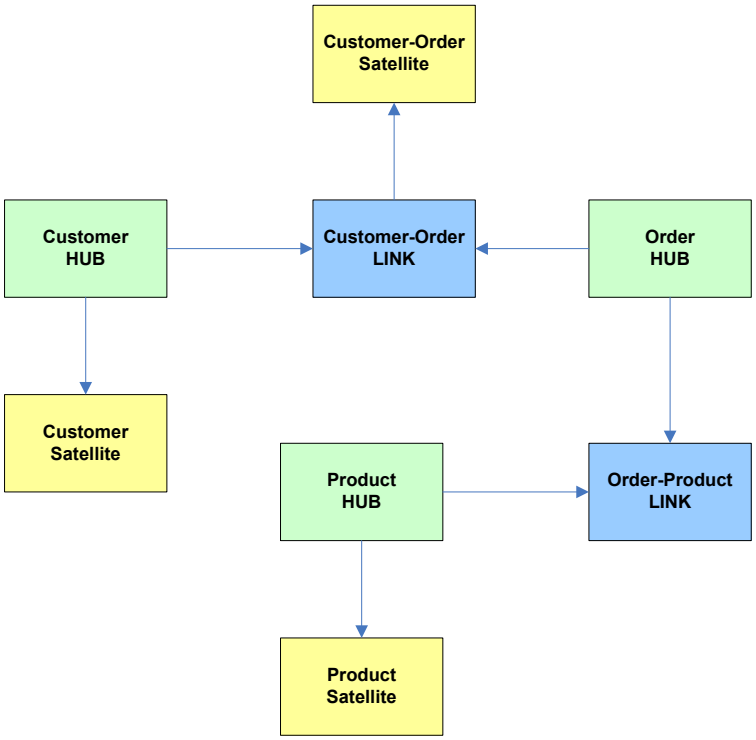
Therese Ahlstam is a senior specialist at Top of Minds AB. Therese is specialized in data warehousing with specific focus on large data volumes, efficient processing and best practice code. She has extensive experience in Bank & Finance, Retail and Gaming.

Comments on this white paper can be sent to: therese.ahlstam@topofminds.se.

About Top of Minds

Top of Minds is a specialized company that offers services in the data warehousing and business intelligence area. We are premier partner in the Nordic countries on the data warehouse development using an agile approach where we use Data Vault to increase the benefits of the projects we participate in. We are a company with great focus on competence, our expertise and our clients' expertise and we are constantly working to spread our knowledge.

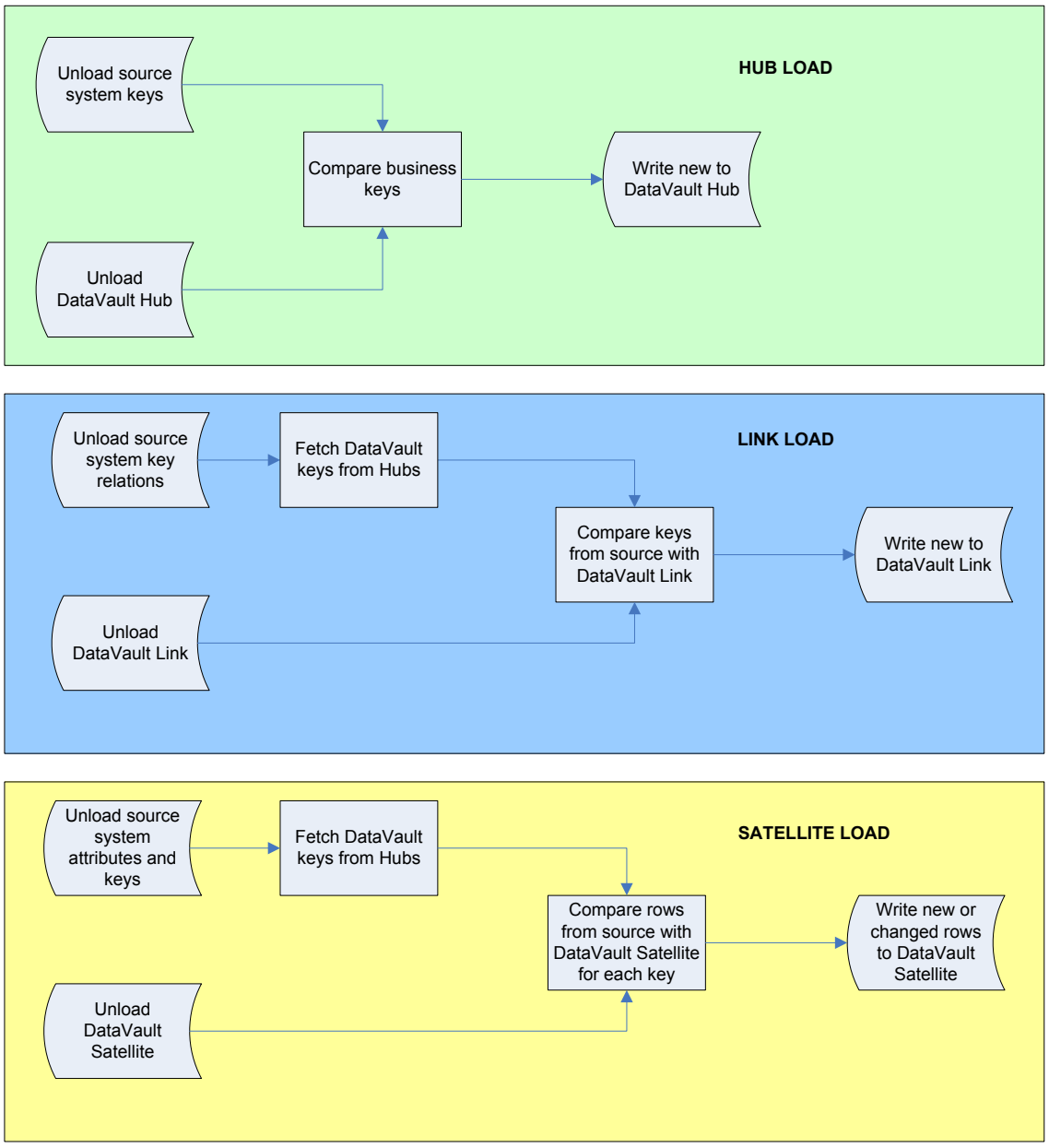
Visit www.topofminds.se for further details.



Thus, in the Data Vault all Hub and Link tables are fairly identical to each other. This is the key to developing reusable ETL code for your Data Vault.

Loading the Data Vault

When loading data into your Data Vault you would start with loading all new business keys into the Hubs, then load all new relations into the Links and finally load all changed descriptive data into the Satellites. A typical Hub, Link and Satellite load process is shown below.



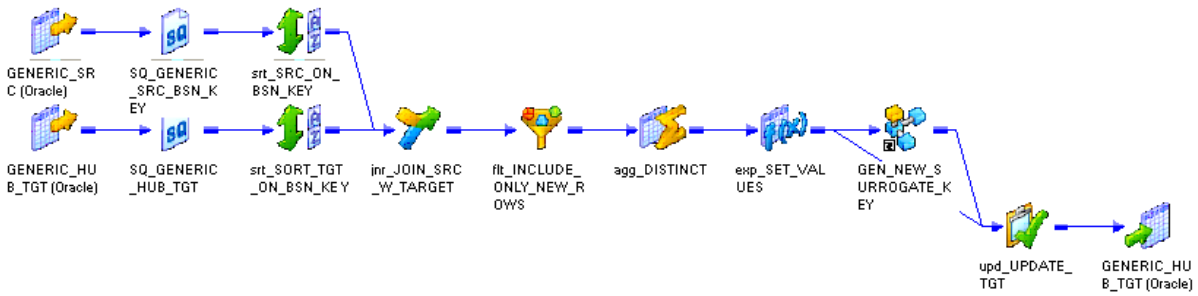
Automating the development process

The key to ETL automation in a Data Vault is the use of Informatica PowerCenter's Advanced Workflow feature. This feature allows you to run the same workflow over and over again, using a variation of different settings. These settings are controlled in a parameter file associated with the particular instance of the Workflow.

By using this feature you can develop one single Workflow for all your Hub tables, using parameters for the source system and Hub table name, business key name and surrogate key name.

In a similar way, you can develop one single Workflow for each variation of your Link tables, i.e. one for 2-key links, one for 3-key links etc.

The process of automating your Hub loads consists of developing one mapping with generic source and targets. In the Source Qualifier the override function can then be used to write a generic source query, which uses parameters instead of column names where applicable. By setting your target writes to Updates you can override the update statement to an insert statement, using parameters in a similar manner. An example Hub load Mapping is shown below.



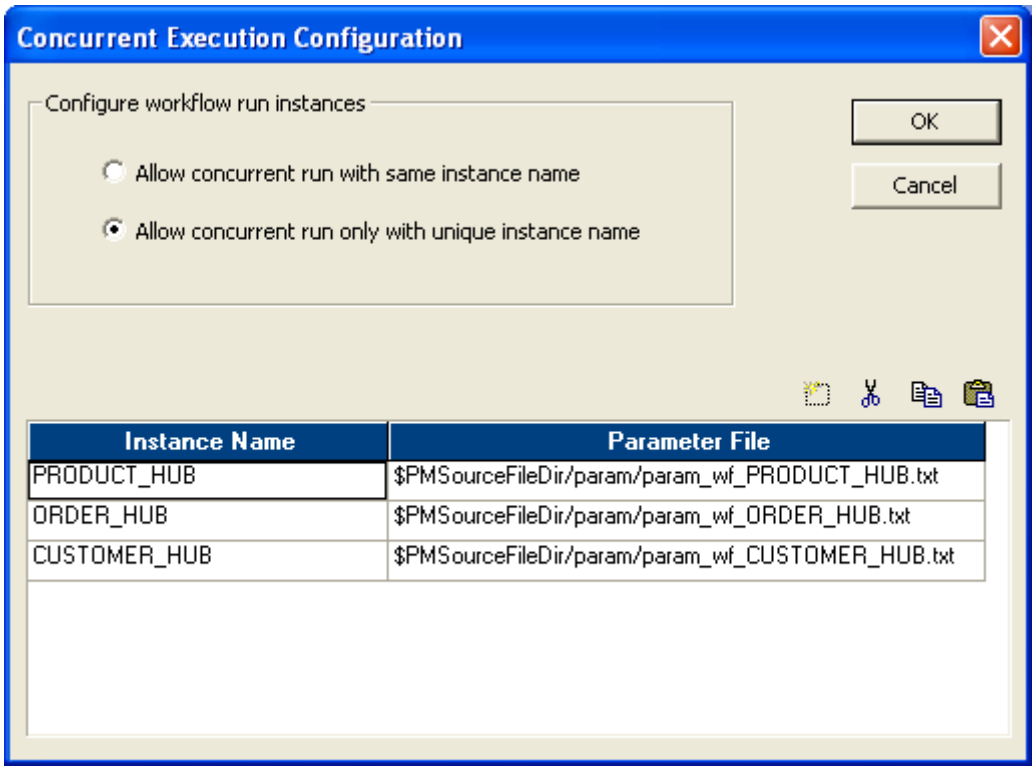
A typical Source Qualifier and Target update override example is shown below.

```
SELECT
srcHub.$$SRC_BSN_KEY_SOR as BSN_KEY
FROM
$$SRC_TBL srcHub

INSERT INTO $$TGT_TBL
($$TGT_SURR_KEY,$$TGT_BSN_KEY,LOAD_DT, SOURCE_ID)
VALUES
(   :TU.SQN,
   :TU.BSN_KEY,
   :TU.LOAD_DTS,
   :TU.SOURCE_ID)
```

With this mapping in place, the actual table and column names can be set via a parameter file, and by configuring concurrent execution via the Advanced Workflow feature the Workflow can be run using several different parameter files, as shown below.

By instantiating a workflow with the Advanced Workflow, the number of workflows will be held to a minimum. A drawback to this approach is the possibility to debug a mapping. When debugging a mapping, you cannot use a specific instance of a session, but rather the generic common settings of a specific session. Thus, to debug a certain instance of an Advanced Workflow with concurrent instances, ensure that the main session path for parameter files points to the specific instance you want to run (i.e. not the specific directory path entered for each instance under Concurrent Execution).



The process of loading your 2-key (or 3, or more) Link tables is much the same as loading your Hubs. For each table type (based on number of keys) a single mapping is developed, and by using Source Qualifier and Target Update overrides, and taking full advantage of the concurrent execution options in Advanced Workflows, the amount of actual code is reduced to a minimum.

Adding a new Hub or Link table to your ETL code base is then only a matter of creating a new parameter file. No new development, no deploys and no altering existing code.

Automating Satellite loads

For Satellite tables in a traditional Data Vault the possibilities of code reusability and automation are greatly reduced, simply due to the fact that Satellites hold all descriptive data and thus have a variable number of columns with different data types. However, at the very least your department should use a template mapping as a reference for all new mappings loading data into a Satellite. This mapping would include all the logical code and by using it as a guide for new development it ensures that all mappings loading to Satellite tables are similar and load data in a consistent matter, with identical solutions for error handling, delta row maintenance and general look-and-feel. A mapping template for Satellite loads is also useful for getting new developers up to speed fast.

For Satellite tables using a name-value pair table structure however, the code reusability factor is as high as that of Hubs and Links, since all Satellite tables will look the same. Other white papers in the ToM Report Series address these possibilities.

Conclusion

By having one single Mapping and Workflow for each table type the amount of code is greatly reduced and thus the number of possible error causes. The benefits of using Advanced Workflows in this manner include, but are not limited to:

- Increased operational stability
- Shorter time-to-market
- Less dependence on single individuals
- Easier and faster to get new colleagues up to speed
- Homogenized code base
- Shorter and more stable test phases