

Introduction

Edwin Weber
Weber Solutions
eacweber@gmail.com

Back end of Data Warehousing
MySQL, SQL Server, Oracle, PostgreSQL
PDI, SSIS, Oracle Warehouse Builder (long ago)

Project

Sint Antonius hospital in Utrecht, Open Source oriented
Chance to combine Kettle experience with a Data Vault
(new to me)
Practically at the same time: project SSIS and Data Vault
So I jumped on the Data Vault bandwagon

Data Vault ETL

- Many objects to load, standardized procedures
- This screams for a generic solution
- I don't want to:
 - manage too many Kettle objects
 - connect similar columns in mappings by hand
- Solution:
 - Generate Kettle objects?
 - Or take it one step further, there's only 1 parameterised hub load object. Don't need to know xml structure of PDI objects.

3

Goal

- Generic ETL to load a Data Vault
- Metadata driven
- No generation, 1 object for each Data Vault entity
 - Hub
 - Link
 - Hub satellite
 - Link satellite
 - Define the mappings, create the Data Vault tables: done!

4

Tools

- Ubuntu
- Pentaho Data Integration CE
- LibreOffice Calc
- MySQL 5.1
- Cookbook, doc generation by Roland Bouman
- (PostgreSQL 9.0, Oracle 11)

5

Deliverables

- Set of PDI jobs and transformations
- Configuration files:
kettle.properties
shared.xml
repositories.xml
- Excel sheet that contains the specifications
- Scripts to generate/populate the pdi_meta and data_vault databases (or schemas)

6

Design decisions

- Updateable views with generic column names
- (MySQL more lenient than PostgreSQL)
- Compare satellite attributes via string comparison
(concatenate all columns, with | (pipe) as delimiter)
- 'inject' the metadata using Kettle parameters
- Generate and use an error table for each Data Vault table. Kettle handles the errors. Helps to find DV design conflicts, tables should contain few to none records in production.

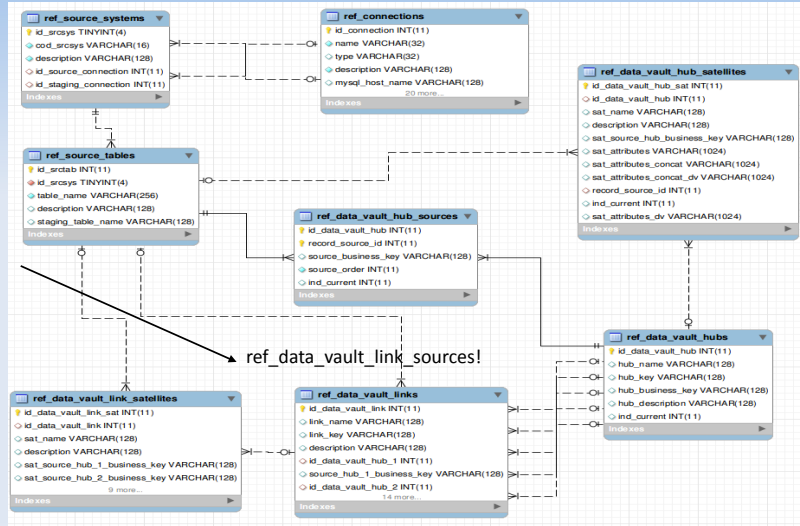
7

Prerequisites

- Data Vault designed and implemented in database
- Staging tables and loading procedures in place
(can also be generic, we use PDI Metadata Injection step for loading files)
- Mapping from source to Data Vault specified
(now in an Excel sheet)

8

Metadata tables



9

Design in LibreOffice (sources)

	A	B	C	D	E
1	id_srctsys	cod_srctsys	description	source_connection	staging_connection
2		1 sys	sysgen	antonius_dwh	
3		2 twin	twin		antonius_stg

Formula: $\sum = \text{IF}(\text{ISBLANK}(A8), "", \text{CONCATENATE}(A8, ", ", B8))$

	A	B	C	D	E
	source_system	table_name	table_description	staging_table_name	source_concat
	sysgen	ref_datum	ref_datum		sysgen.ref_datum
	sysgen	ref_dagdeel	ref_dagdeel		sysgen.ref_dagdeel
	sysgen	ref_specialisme_vertaling	ref_specialisme_vertaling		sysgen.ref_specialisme_vertaling
	twin	stg_TB_OPER_SESSIES	stg_TB_OPER_SESSIES	stg_TB_OPER_SESSIES	twin.stg_TB_OPER_SESSIES
	twin	stg_TB_SPECIALISMEN	stg_TB_SPECIALISMEN	stg_TB_SPECIALISMEN	twin.stg_TB_SPECIALISMEN
	twin	stg_TB_SPECIALISTMEN	stg_TB_SPECIALISTMEN	stg_TB_SPECIALISTMEN	twin.stg_TB_SPECIALISTMEN
	twin	stg_weekschemas	stg_weekschemas	stg_weekschemas	twin.stg_weekschemas

A	B
name	description
antonius_dwh	Database voor de Data Vault
antonius_stg	Staging database

10

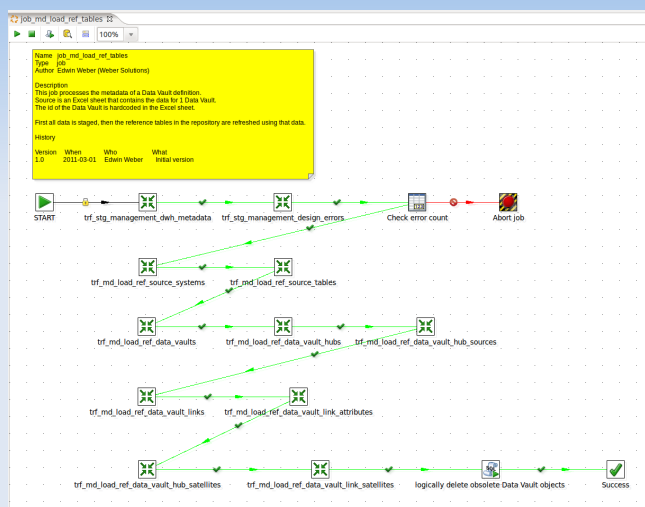
Design in LibreOffice (hub + sat)

A	B	C	D	E	F	G	H
hub_name	hub_description	hub_key	hub_business_key	hub_source	hub_source_business_key	hub_source_order	ind_current
hub_patient	hub_patient	hub_patient_id	patient_pin	twm.stg_TB_OPER_SESSIES	pat_nr	1	1
hub_ok_complex	hub_ok_complex	hub_ok_complex_id	ok_complex_code	twm.stg_TB_OPER_SESSIES	prof_atd_code	1	1
hub_ok_complex	hub_ok_complex	hub_ok_complex_id	ok_complex_code	twm.stg_weekschemas	ok_compl	2	1
hub_specialisme	hub_specialisme	hub_specialisme_id	specialisme_code	twm.stg_TB_SPECIALISMEN	spome_code	1	1
hub_specialisme	hub_specialisme	hub_specialisme_id	specialisme_code	twm.stg_TB_OPER_SESSIES	aanw_spectime_code	2	1
ref_datum	ref_datum	datum_id	datum	sysgen.ref_datum		1	0
ref_dagdeel	ref_dagdeel	dagdeel_id	dagdeel_kort	sysgen.ref_dagdeel		1	0

	A	B	C	D	E	F	G	H	I	J
1	sat_name	sat_key	sat_description	sat_hub	ind_current	source_concat	source_hub_business_key	attribute_number	attribute_source_column	attribute_target_column
2	sat_hub_specialisme_twn	sat_hub_specialisme_twn_id	sat_hub_specialisme_twn	hub_specialisme	1	twm.stg_TB_SPECIALISMEN	spome_code			
3								1	spome_ootschr	spome_ootschr
4								2	int_spome_code	int_spome_code
5								3	agb_spome_code	agb_spome_code
6								4	organisatie	organisatie
7								5	bcr_volgrr_verwerk	bcr_volgrr_verwerk
								6	loondest_ind	loondest_ind

11

Loading the metadata



12

'design errors'

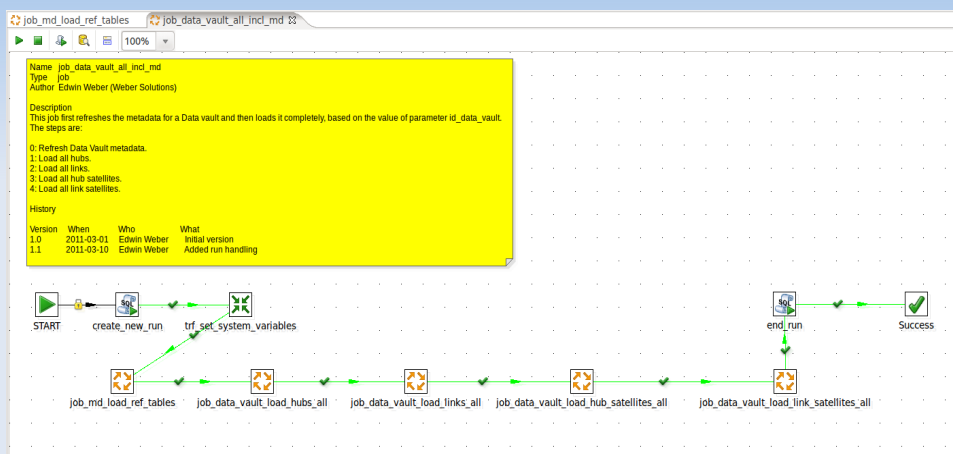
Checks to avoid debugging:

(compares design metadata with Data Vault DB information_schema)

- hubs, links, satellites that don't exist in the DV
- key columns that do not exist in the DV
- missing connection data (source db)
- missing attribute columns

13

A complete run



14

Spec: loading a hub

Load a hub, specified by:

- name
 - key column
 - business key column
 - source table
 - source table business key column
- (can be expression, e.g. concatenate for composite key)

15

The Kettle objects: job hub

The screenshot shows a Kettle job design with the following steps: START, create_hub_lookup_view, create_hub_updateable_view, trf_data_vault_set_connection_parameters, trf_data_vault_hub_generic, and Success. Two 'Execute SQL Script' dialog boxes are overlaid on the job design.

Execute SQL Script ... (Left)

- Job entry name: create_hub_lookup_view
- Connection: data_vault
- SQL from file:
- SQL filename:
- Send SQL as single statement:
- Use variable substitution:
- SQL Script:

```
CREATE OR REPLACE VIEW vw_$(hub_name)_ltp AS
SELECT $(hub_business_key) as hub_business_key_dv
, $(hub_key) as hub_key_dv
, 1 as exists_ltp_dv
FROM $(hub_name)
```
- Line 1 Column 0

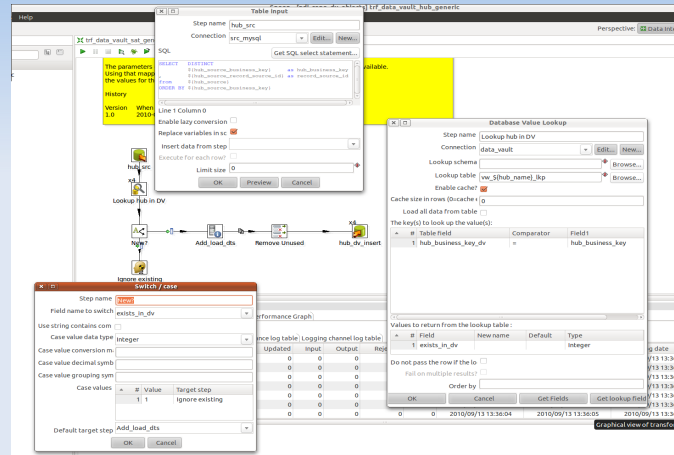
Execute SQL Script ... (Right)

- Job entry name: create_hub_updateable_view
- Connection: data_vault
- SQL from file:
- SQL filename:
- Send SQL as single statement:
- Use variable substitution:
- SQL Script:

```
CREATE OR REPLACE VIEW vw_$(hub_name)_upd AS
SELECT $(hub_business_key) as hub_business_key
, load_offs
, record_source_id
FROM $(hub_name)
```
- Line 1 Column 0

16

The Kettle objects: trf hub



17

Spec: loading a link

Load a link, specified by:

- name
- key column
- for each hub (maximum 10, can be a ref-table)
 - hub name
 - column name for the hub key in the link (roles!)
 - column in the source table → business key of hub
- link 'attributes' (part of key, no hub, maximum 5)
- source table

18

The Kettle objects: job link

Name: job_data_vault_load_link
Type: job
Author: Edwin Weber (Weber Solutions)

Description:
 This job loads a link in a Data Vault.
 Based on the parameter values the steps are:
 1. Create (or replace) a lookup view for the link. This gives the generic link transformation a lookup functionality with fixed column names.
 2. Create (or replace) an updateable view for the link. This gives the generic link transformation an updateinsert functionality with fixed column names.
 3. Create the error table (if not already present).
 4. Set the connection parameters. This makes it possible to source data from different databases in 1 Data Vault run.
 5. Execute the generic transformation.

History:

Version	When	Who	What
1.0	2011-03-01	Edwin Weber	Initial version
1.1	2011-03-10	Edwin Weber	Added creation of error table

The flowchart shows the following sequence of transformations:
 START → Link attributes? → create_link_lookup_view_incl_attr → create_link_updateable_view_incl_attr → create_error_table (attr) → create_link_updateable_view_err (attr) → trf_data_vault_set_connection_parameters → trf_data_vault_link_generic → Update et_id_run in errors → Success.
 A parallel path for 'no attributes' exists: Success 2 → No attributes? → create_link_lookup_view_no_attr → create_link_updateable_view_no_attr → create_error_table (no attr) → create_link_updateable_view_err (no attr).

The Kettle objects: trf link

Name: trf_data_vault_link_generic_no_attributes
Type: transformation
Author: Edwin Weber (Weber Solutions)

Description:
 This transformation loads a Data Vault link that only links hubs and has no 'from' hub key attributes.
 A maximum of 10 hubs is currently supported.
 For every link 10 hubs are joined in this transformation.
 In case the link links 4 hubs, the first 4 joins will be with those hubs, the 6 remaining joins will lookup the dummy hub.
 This result in relevant hub_keys and irrelevant ones. The latter must be removed from the flow, because the link table has no target column for them. That's why there are 9 'Remove Unused ... hub' steps.

History:

Version	When	Who	What
1.0	2011-03-01	Edwin Weber	Initial version
1.1	2011-03-10	Edwin Weber	Added error handling

Log the handling of the Data Vault object

The flowchart for the transformation includes:
 Get System Info → Log transformation → Ink_src → Lookup hub 1-10 → Lookup link → Already exists? → Add_load_ds → How may rows? → Ink_dv_insert → Link_dv_errors.
 There are 9 'Remove Unused ... hub' objects (1-9) that branch off from the 'Lookup link' step. An arrow points to 'Remove Unused 1 hub (peg-legged link)'.

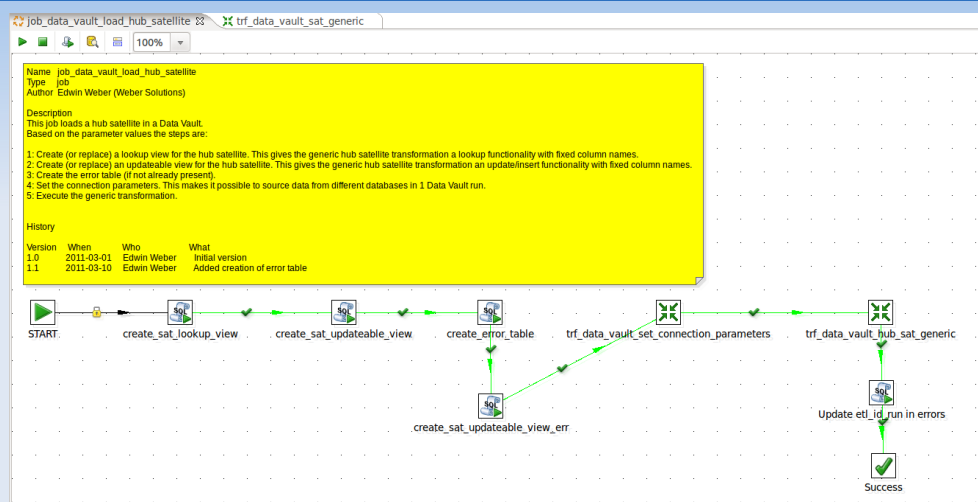
Spec: loading a hub satellite

Load a hub satellite, specified by:

- name
- key column
- hub name
- column in the source table → business key of hub
- for each attribute (maximum 200)
 - source column
 - target column
- source table

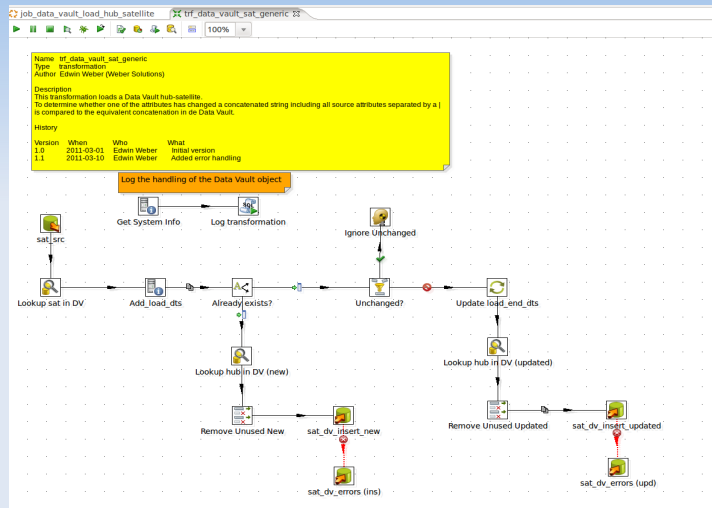
21

The Kettle objects: job hub sat



22

The Kettle objects: trf hub sat



23

Spec: loading a link satellite

Load a link satellite, specified by:

- name
- key column
- link name
- for each hub of the link:
column in the source table → business key of hub
- for each key attribute: source column
- for each attribute: source column → target column
- source table

24

Executing in a loop ..

The screenshot displays the Informatica Designer interface for a job named 'job_data_vault_load_hubs'. The job is configured to execute in a loop. The 'Executing a job...' dialog shows the job entry name and job specification. The SQL code is as follows:

```

select
  hubs.hub_name
, hubs.hub_key
, hubs.hub_business_key
, ifnull(srctab.staging_table_name, srctab.table_name) as hub_source
, src.record_source_id as hub_source_business_key
, src.record_source_id as hub_source_record_source_id
, ifnull(srccsys.id_staging_connection, srccsys.id_source_connection) as id_connection
, 'trf_data_vault_hub_generic' as hub_connection_to_execute
from
  ref_data_vault_hubs hubs
inner join
  ref_data_vault_hubs_sources src
on
  hubs.id_data_vault_hub = src.id_data_vault_hub
inner join
  ref_source_tables srctab
on
  src.record_source_id = srctab.id_srctab
inner join
  ref_source_systems srccsys
on
  srctab.id_srccsys = srccsys.id_srccsys
where
  hubs.ind_current = 1
and
  src.ind_current = 1
and
  mod(hubs.id_data_vault_hub,4) = ${mod_4_value}
and
  hubs.id_data_vault = ${id_data_vault}
order by
  hubs.hub_name
, src.source_order
    
```

25

.. and parallel

The screenshot displays the Informatica Designer interface for a job named 'job_data_vault_load_hubs_all'. The job is configured to execute in parallel. The 'Executing a job...' dialog shows the job entry name and job specification. The dialog also shows the parameter values for the job:

#	Parameter	Stream column name	Value
1	mod_4_value		0

26

Logging

Default PDI logging enabled (e.g. errors)

N times 'generic job' is not so informative, so the jobs log:

- hub name
- link name
- hub satellite name
- link satellite name
- number of rows as start/end
- start/end time

27

Some points of interest

- Easy to make mistake in design sheet
- Generic → a bit harder to maintain and debug
- Application/tool to maintain metadata?
- Doc&#x24;@%tation (internals, checklists)

28

Availability of the code

- Free, because that's fair. I make a living with stuff that other people give away for free.
- Two flavours for now, MySQL and PostgreSQL. Oracle is 'under construction'.
- It's not on SourceForge, just mail me some Belgium beer and you get the code.