My Datastage Notes

Pages

- Home
- Different Versions
- All Datastage Stages
- Configuration File
- Sequentional_Stage
- Dataset
- Transformer Stage
- Sort Stage
- Aggregator_Stage
- Join Stage
- Lookup_Stage
- Merge_Stage
- Filter_Stage
- Copy Stage
- Funnel_Stage
- Column Generator
- Surrogate_Key_Stage
- SCD
- Pivot_Enterprise_Stage
- Sequence_Activities
- Datastage Study Material/Interview Questions
- Datastage Errors and Resolution
- Datastage Scenarios and solutions
- Unix Shell Scripting
- SQL/Database
- Datawarehousing Concepts

Blog Archive

▼ 2014 (46)

► September (40)

Sunday, May 11, 2014

SCD Type 2

Slowly Changing Dimensions (SCDs) are dimensions that have data that changes slowly, rather than changing on a time-based, regular schedule.

Type 1

The Type 1 methodology overwrites old data with new data, and therefore does not track historical data at all.

Here is an example of a database table that keeps supplier information:

Supplier_Key	Supplier_Code	Supplier_Name	Supplier_State
123	ABC	Acme Supply Co	CA

In this example, Supplier_Code is the natural key and Supplier_Key is a surrogate key. Technically, the surrogate key is not necessary, since the table will be unique by the natural key (Supplier_Code). However, the joins will perform better on an integer than on a character string.

Now imagine that this supplier moves their headquarters to Illinois. The updated table would simply overwrite this record:

Supplier_Key	Supplier_Code	Supplier_Name	Supplier_State
123	ABC	Acme Supply Co	IL

Type 2

▼ May (4)
Resource Estimation tool
Local Containers and Shared Container
SCD Type 2
Dimension and Facts
► February (2)
► 2013 (39)

The Type 2 method tracks historical data by creating multiple records for a given natural key in the dimensional tables with separate surrogate keys and/or different version numbers. With Type 2, we have unlimited history preservation as a new record is inserted each time a change is made.

In the same example, if the supplier moves to Illinois, the table could look like this, with incremented version numbers to indicate the sequence of changes:

Supplier_Key	Supplier_Code	Supplier_Name	Supplier_State	Version
123	ABC	Acme Supply Co	CA	0
124	ABC	Acme Supply Co	IL	1

Another popular method for tuple versioning is to add effective date columns.

Supplier_Key	Supplier_Code	Supplier_Name	Supplier_State	Start_Date	End_Date
123	ABC	Acme Supply Co	CA	01-Jan-2000	21-Dec-2004
124	ABC	Acme Supply Co	IL	22-Dec-2004	

The null End_Date in row two indicates the current tuple version. In some cases, a standardized surrogate high date (e.g. 9999-12-31) may be used as an end date, so that the field can be included in an index, and so that null-value substitution is not required when querying.

How to Implement SCD using DataStage 8.1 –SCD stage?

Step 1: Create a datastage job with the below structure-

1. Source file that comes from the OLTP sources
2. Old dimesion refernce table link
3. The SCD stage
4. Target Fact Table
5. Dimesion Update/Insert link

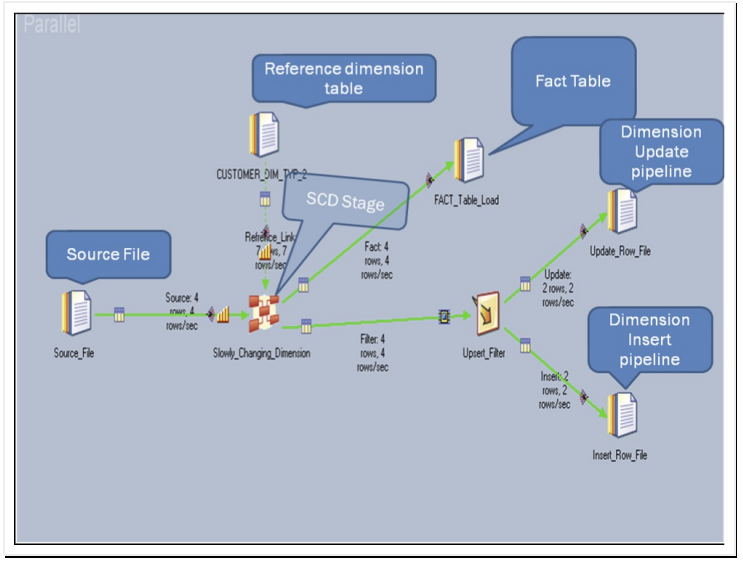

Figure 1

Step 2: To set up the SCD properties in the SCD stage ,open the stage and access the Fast Path

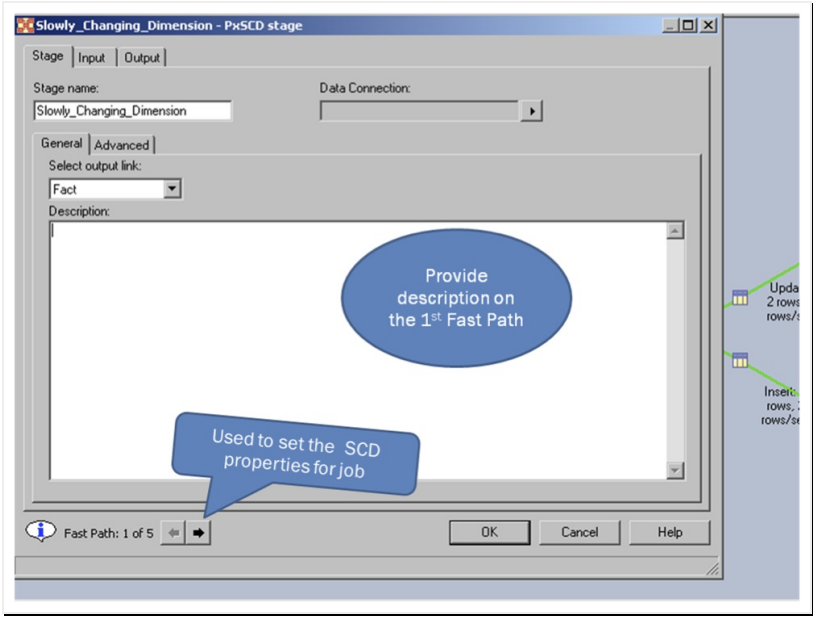

Figure 2

Step 3: The tab 2 of SCD stage is used specify the purpose of each of the pulled keys from the referenced dimension tables.

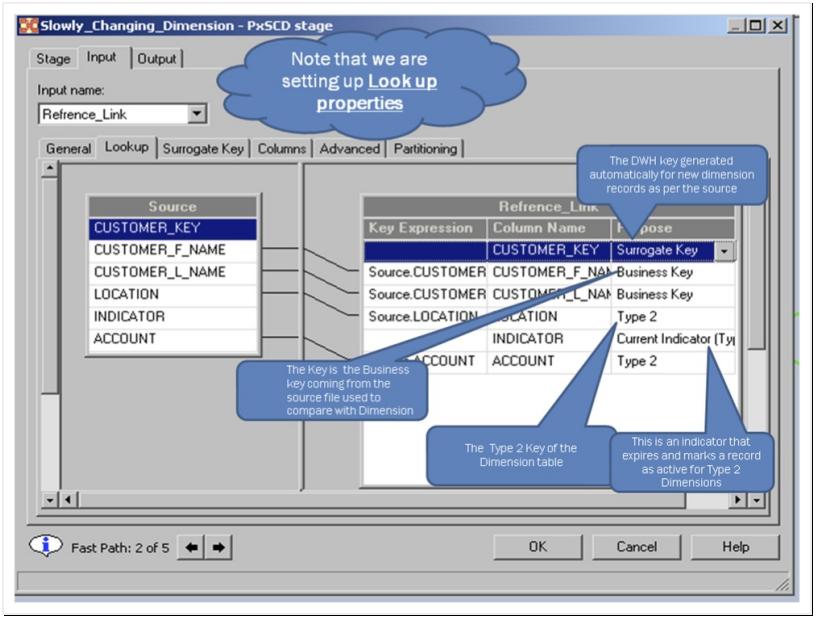

Figure 3

Step 4: Tab 3 is used to provide the seqence generator file/table name which is used to generate the new surrogate keys for the new or latest dimesion records.These are keys which also get passed to the fact tables for direct load.

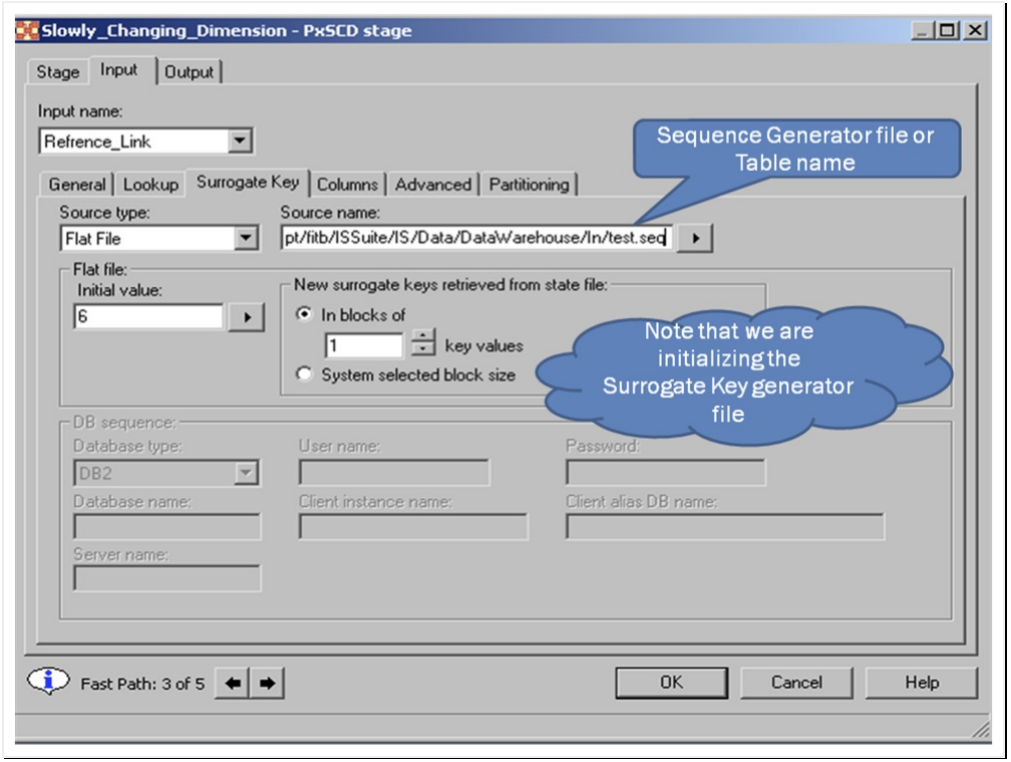

Figure 4

Step 5: The Tab 4 is used to set the properties for configuring the data population logic for the new and old dimension rows. The type of activies that we can configure as a part of this tab are:

1. Generation the new Surrogate key values to be passed to the dimension and fact table
2. Mapping the source columns with the source column
3. Setting up of the expired values for the old rows
4. Defining the values to mark the current active rows out of multiple type rows

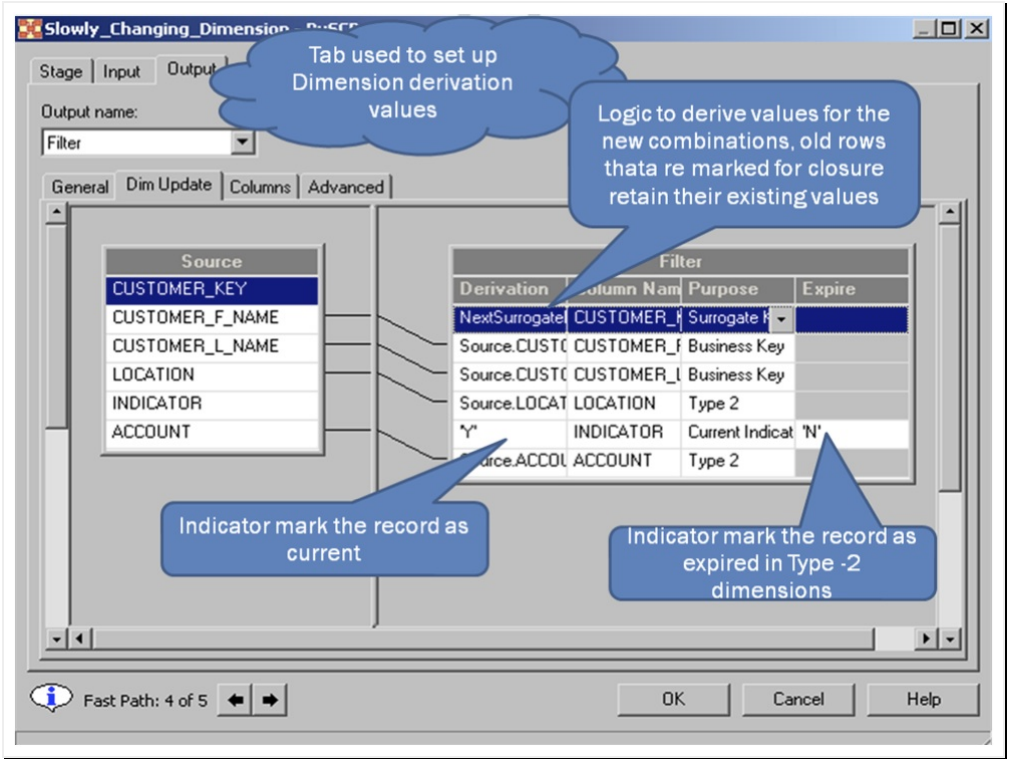

Figure 5

Step 6: Set the derivation logic for the fact as a part of the last tab.

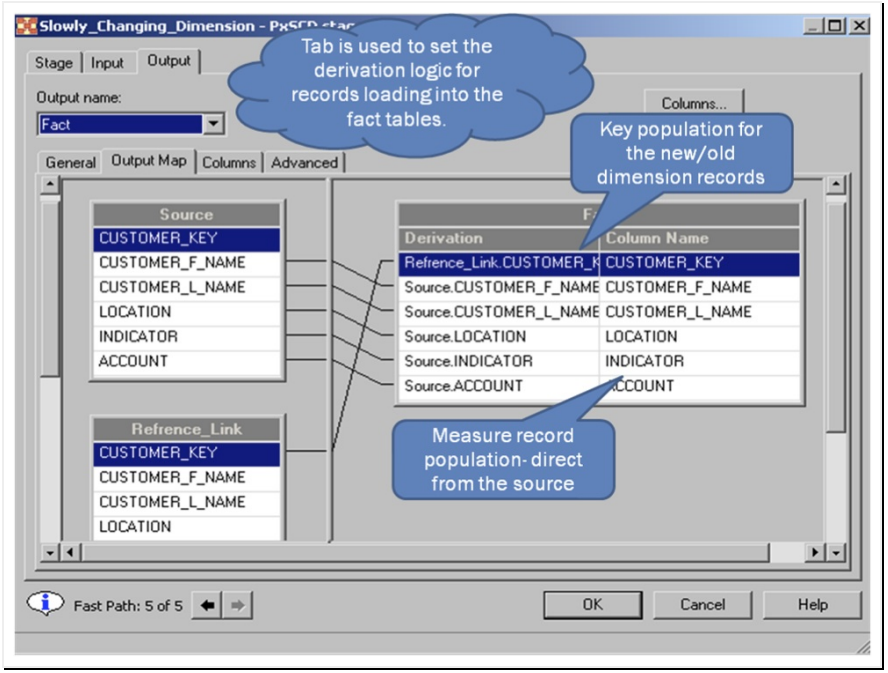

Figure 6

Step 7: Complete the remaining set up, run the job

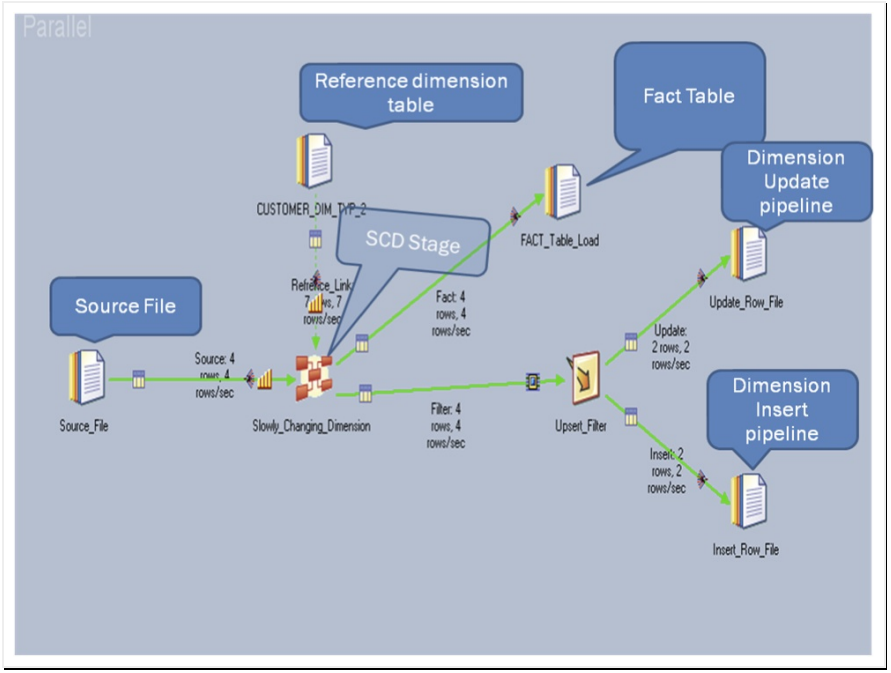

Figure 7

Posted by manohar at 3:35 AM

No comments:

Post a Comment

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)