Informatica PowerCenter

Lesson 14: Creating a Type 2 Dimension/Version Data Mapping

June 17, 2014    Proprietary and Confidential    – 1 –

IGATE
Speed.Agility.Imagination

## 14.1. Introduction

➤ **Dimensions contains Textual attributes.**
  – The Type 1 Dimension mapping - Loads a slowly changing dimension table by inserting new dimensions and overwriting existing dimensions.
  – Ie. Rows containing changes to existing dimensions are updated in the target by overwriting the existing dimension.
  – In the Type 1 Dimension mapping, all rows contain current dimension data and historical data is lost.

June 17, 2014    Proprietary and Confidential    - 3 -

IGATE

Use this mapping when you do not you need to keep any previous versions / history of dimensions in the table.

## 14.1. Type 1 - Example

➤ For example, we have a fact table, FACT_TRANS, containing the sales data for various snack and beverage products and a dimension table,LU_PRODUCT, containing the product descriptions. Over time, however, one of these product descriptions as recorded in the transaction log, TRANS_LOG.TXT,changed, resulting in an SCD that must be processed.
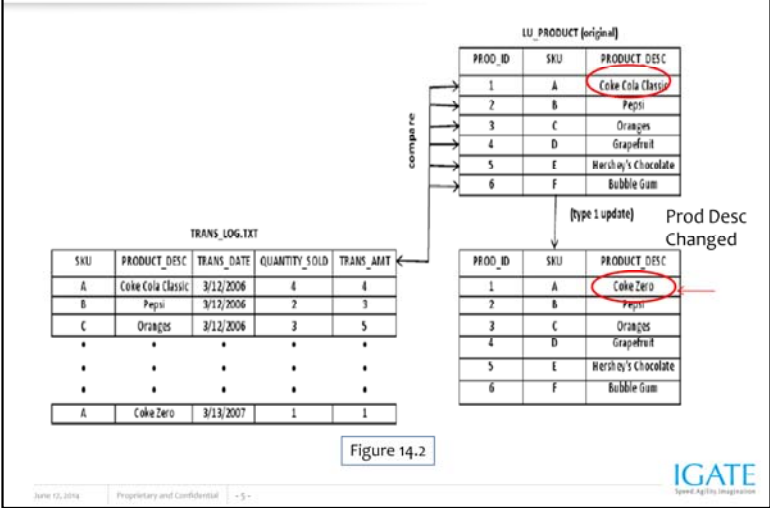
TRANS_LOG.TXT

| SKU | PRODUCT_DESC | TRANS_DATE | QUANTITY_SOLD | TRANS_AMT |
|-----|--------------|------------|---------------|-----------|
| A | Coke Cola Classic | 3/12/2006 | 4 | 4 |
| B | Pepsi | 3/12/2006 | 2 | 3 |
| C | Oranges | 3/12/2006 | 3 | 5 |
| . | . | . | . | . |
| . | . | . | . | . |
| . | . | . | . | . |
| A | Coke Zero | 3/13/2007 | 1 | 1 |

Figure 14.1

June 12, 2014    Proprietary and Confidential    - 4 -                contd.    IGATE

## 14.1. Type 1 - Example



Figure 14.2

## 14.1. Introduction

➢ The Type 2 Dimension/Version Data mapping filters source rows based on user-defined comparisons and inserts both new and changed dimensions into the target.

➢ Changes are tracked in the target table by versioning the primary key and creating a version number for each dimension in the table. When you use this option, the Designer creates two additional fields in the target:

➢ PM_PRIMARYKEY. The Integration Service generates a primary key for each row written to the target.

➢ PM_VERSION_NUMBER. The Integration Service generates a version number for each row written to the target

June 13, 2014    Proprietary and Confidential    - 6 -

IGATE
Speed. Agility. Imagination

In the Type 2 Dimension/Version Data target, the current version of a dimension has the highest version number and the highest incremented primary key of the dimension.
Use the Type 2 Dimension/Version Data mapping to update a slowly changing dimension table when you want to keep a full history of dimension data in the table. Version numbers and versioned primary keys track the order of changes to each dimension.

## 14.2. Handling Keys

- In a Type 2 Dimension/Version Data mapping, the Integration Service generates a new primary key value for each new dimension it inserts into the target.
- An Expression transformation increments key values by 1,000 for new dimensions.
- When updating an existing dimension, the Integration Service increments the existing primary key by 1.

June 17, 2014   Proprietary and Confidential   - 7 -

IGATE

## 14.3. Example

➢ For example, the Integration Service inserts the following new row with a key value of 65,000 since this is the sixty-fifth dimension in the table.

➢ PM_PRIMARYKEY      ITEM      STYLES

     65000          Sandal          5

Contd…

June 17, 2014     Proprietary and Confidential     - 8 -

IGATE
Speed. Agility. Imagination

➤ The next time you run the workflow containing the session, the same item has a different number of styles.

➤ The Integration Service creates a new row with updated style information and increases the existing key by 1 to create a new key of 65,001.

➤ Both rows exist in the target, but the row with the higher key version contains current dimension data.

➤ 

| PM_PRIMARYKEY | ITEM | STYLES |
|---|---|---|
| 65000 | Sandal | 5 |
| 65001 | Sandal | 14 |

Contd...

IGATE

June 17, 2014    Proprietary and Confidential    - 9 -

➤ When you run the workflow again, the Integration Service again increments the key.
➤ The highest key version contains current dimension data.
➤ The target keeps a full history of the item as well as the order in which the versions occurred.
➤ PM_PRIMARYKEY             ITEM              STYLES

| PM_PRIMARYKEY | ITEM | STYLES |
|---|---|---|
| 65000 | Sandal | 5 |
| 65001 | Sandal | 14 |
| 65002 | Sandal | 17 |

IGATE

June 17, 2014     Proprietary and Confidential    ~10~

For example, in the data below, the versions are 0, 1, and 2. The highest version number contains the current dimension data.

| PM_PRIMARYKEY | ITEM | STYLES | PM_VERSION_NUMBER |
|---|---|---|---|
| 65000 | Sandal | 5 | 0 |
| 65001 | Sandal | 14 | 1 |
| 65002 | Sandal | 17 | 2 |

## 14.5. Understanding the Mapping

➤ Selects all rows
➤ Caches the existing target as a lookup table
➤ Compares logical key columns in the source against corresponding columns in the target lookup table
➤ Compares source columns against corresponding target columns if key columns match
➤ Flags new rows and changed rows
➤ Creates two data flows: one for new rows, one for changed rows
➤ Generates a primary key and version number for new rows
➤ Inserts new rows to the target
➤ Increments the primary key and version number for changed rows
➤ Inserts changed rows in the target

June 17, 2014     Proprietary and Confidential    « 12 »

IGATE
Speed.Agility.Imagination

The Type 2 Dimension/Version Data mapping uses a Lookup and an Expression transformation to compare source data against existing target data. When you step through the Slowly Changing Dimensions Wizard, you enter the lookup conditions (source key columns) and source columns that you want the Integration Service to compare against the existing target.

For each source row :
• without a matching primary key in the target, the Expression transformation marks the row new.
• with a matching primary key in the target, the Expression compares user-defined source and target columns. If those columns do not match, the Expression marks the row changed.
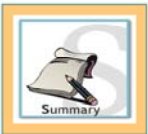
The mapping then splits into two separate data flows.
• The first data flow uses the Filter transformation, FIL_InsertNewRecord, to filter out existing rows. The Filter transformation passes only new rows to the UPD_ForceInserts Update Strategy transformation. UPD_ForceInserts inserts new rows to the target. A Sequence Generator creates a primary key for each row. The Expression transformation, EXP_KeyProcessing_InsertNew, increases the increment between keys by 1,000 and creates a version number of 0 for each new row.

• In the second data flow, the FIL_InsertChangedRecord Filter transformation allows only changed rows to pass to the Update Strategy transformation, UPD_ChangedInserts. UPD_ChangedInserts inserts changed rows to the target. The Expression transformation, EXP_KeyProcessing_InsertChanged, increments both the existing key and the existing version number by one.

## Summary

➢ This Lesson gives knowledge about Creating a Type 2 Dimension/Version Data Mapping

IGATE