Data Warehousing Concept Using ETL Process
For SCD Type-3
K. Srikanth¹, N.V.E.S. Murthy², J. Anitha³

¹Andhra University, CSE, M.Tech (Ph.D), Vishakhapatnam, India
²Andhra University, CSE, Professor, Vishakhapatnam, India
³Andhra University, CSE, M.Tech (Ph.D), Vishakhapatnam, India

Abstract: In this approach, only the information about a previous value of a dimension is written into the database. An 'old' or 'previous' column is created which stores the immediate previous attribute. The problem with this approach is over years, if the product price continuously changes, then the complete history may not be stored, only the latest change will be stored. Slowly Changing Dimension Type 3 preserves only few history versions of data, most of the time 'Current' and Previous versions. The 'Previous' version value will be stored into the additional columns with in the same dimension record. In this article let's discuss the step by step implementation of SCD Type 3 using Informatica Power Center.

Keywords–ETL; Metadata; Mapping; Transformation.

1. INTRODUCTION

1. Introduction to Slowly Changing Dimensions:
One of the standard issues in Data warehousing is how to handle the changes when they take place in a particular field or an attribute in certain database? In some situations, data may need to be updated, while other data needs to be tracked historically [1]. These different tracking requirements are known as certain dimension change types. Although there are several different change types, the following are the three most common ones that relate to dimension ETL:

They are SCD Type1, SCD Type 2 and SCD Type 3. In this paper we study about SCD Type 3 and SCD Type was studied in Data Warehouse concepts with informatica and SCD Type 2 was studied in Informatica with ETL.

1.1 SCD Type 3, Slowly Changing Dimension Use, Example, Advantage, Disadvantage:
In Type 3 Slowly Changing Dimension, there will be two columns to indicate the particular attribute of interest, one indicating the original value, and one indicating the current value[2]. There will also be a column that indicates when the current value becomes active. In our example, recall we originally have the following table:

<table>
<thead>
<tr>
<th>Empno</th>
<th>Name</th>
<th>Sal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1111</td>
<td>Srikanth</td>
<td>500</td>
</tr>
</tbody>
</table>

To accommodate Type 3 Slowly Changing Dimension, we will now have the following columns:

<table>
<thead>
<tr>
<th>Empno</th>
<th>Name</th>
<th>Sal</th>
<th>Previous Sal</th>
<th>Effective Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1111</td>
<td>Srikanth</td>
<td>500</td>
<td>2000</td>
<td>01-Feb-2012</td>
</tr>
</tbody>
</table>

I. Advantages:
- This does not increase the size of the table, since new information is updated.
- This allows us to keep the immediate past of the history, which is required as in Update Emp Table set Sal is 2500 and assign the Empno.

Ex: first write the Query in update salary
SQL> update emp set sal=2500 where Empno=1111

II. Disadvantages:
- Type 3 will not be able to keep all history where an attribute is changed more than once.

III. When to use Type 3:
- Type III slowly changing dimension should only be used when it is necessary for the data warehouse to track historical changes[4], and when such changes will only occur for a finite number of time.

2. Implementation:

2.1 Source:

Table 1: Oracle SQL Query Through EMP Table

<table>
<thead>
<tr>
<th>Empno</th>
<th>Name</th>
<th>Sal</th>
<th>Previous Sal</th>
<th>Effective Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1111</td>
<td>Srikanth</td>
<td>500</td>
<td>2000</td>
<td>01-Feb-2012</td>
</tr>
</tbody>
</table>

Create the source Table and dimension tables in the database using Table 1.

Open the mapping designer tool, source analyzer and either create or import the source definition.

Go to the Warehouse designer or Target designer and import the target definition[2].

Go to the mapping designer tab and create new mapping.

Drag the source into the mapping[7].

2.2 Understand the Source and Dimension Table:

For our demonstration purpose, let's consider the EMP Dimension. Here we will keep previous version of SAL into its corresponding PREV columns. Below are the detailed structure of both source and dimension table.

1) 2.2.1 Source Table:

2) In our source table, we have all the columns required for the dimension table attributes.[5], So no other tables other than Dimension table will be involved in the mapping. Below is the structure of Figure 1. our source table.

![Figure 1: Source Table through EMP](image)

**Figure 1: Source Table through EMP**

I. Key Points:

1. Source table will have only one day's data. Change Data Capture is not in scope.
2. Data is uniquely identified using EMPNO.
3. All attribute required by Dimension Table is available in the staging table.[6].

3) 2.2.2 2.2.2 Dimension Table:

Here is the structure of Figure 2. our Dimension table.

![Figure 2: Dimension Table Through EMP](image)

**Figure 2: Dimension Table Through EMP**

I. Key Points:

1. EMPKEY is the surrogate key.
2. EMPNO is the Natural key, hence the unique record identifier.
3. Previous versions are kept in SAL columns.

3. Mapping Building and Configuration

3.1 Step 1:

Let's start the mapping building process. Figure 3. For that pull the M_T3 source definition into the mapping designer.

![Figure 3: Mapping Designer Through M_T3](image)

**Figure 3: Mapping Designer Through M_T3**

3.2 Step 2:

Now using a LookUp Transformation fetch the existing Customer columns from the dimension table LKPTRANS. Figure 4. This lookup will give NULL values if the customer is not already existing in the Dimension tables.

LookUp Condition: MPNO=EMPNO1

Return Columns: EMPKEY, SAL

![Figure 4: Mapping Designer Through LKPTRANS](image)

**Figure 4: Mapping Designer Through LKPTRANS**

3.3 Step 3:

Using an Expression Transformation, identify the records for Insert and Update using below expression. Additionally, map the columns from the LookUp Transformation to the Expression as shown below[8]. With this we get both the previous and current values of the EMPNO. Figure 5. Map the columns from the Expression Transformation to a Router Transformation and create two groups (INSERT, UPDATE) in Router Transformation using the below expression. The mapping will look like shown in the image.

- INSERT: IIF(ISNULL(EMPKEY), 'T', 'F')
- UPDATE: IIF(NOT ISNULL(EMPKEY) AND SAL ! = SAL1), 'T', 'F')

Additionally create two output ports.

- INSERT: TRUE
- UPDATE: TRUE

![Figure 5: Mapping Designer Through EXPTRANS](image)

**Figure 5: Mapping Designer Through EXPTRANS**
3.4 Step 4: Records coming from the ‘UPDATE Group will update the customer Dimension with Current customer attributes and the ‘PREV’ attributes.[9]. Figure 6. Add an Update Strategy Transformation before the target instance and set it as UPS,UP2. Below is the structure of the mapping.

![Image of Update Strategy Transformation](image1)

Figure 6: Update Strategy Transformation Through PreviousSal

3.5 We are done with the mapping building and below is the structure of the completed mapping. Figure 7.

![Image of Slowly Changing Dimensions](image2)

Figure 7: Slowly Changing Dimensions (SCDs) Flow

4. Workflow and Session Creation

4.1 There is not any specific properties required to be given during the session configuration.

![Image of Run Workflow Designer](image3)

Figure 8: Run Workflow Designer

4.2 Below is a sample data set taken from the Dimension table EMP_T3. See the highlighted values. Figure 8.

```sql
SQL> update emp set sal=500 where empno=1111;
1 row updated.
```

![Image of Oracle SQL Query](image4)

Table 2: Oracle SQL Query Through EMP Target Table

4.3 Once load the target data after write oracle queries in update the values Using connect the employee table1. Table 2, Table 3. Below oracle table update display the new type 3 complete updated data.

![Image of Oracle SQL Query Table](image5)

Table 3: Oracle SQL Query Through EMP Target Table for GUI

<table>
<thead>
<tr>
<th>Source Data</th>
<th>Target Data</th>
<th>Updated Target Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>Table 2</td>
<td>Table 3</td>
</tr>
</tbody>
</table>

5. CONCLUSIONS

Extraction-Transformation-Loading (ETL) tools are pieces of software responsible for the extraction of data from several sources. SCD Type 2, Slowly Changing Dimension Type 3 preserves only few history versions of data, most of the time ‘Current’ and Previous’ versions. The ‘Previous’ version value will be stored into the additional columns within the same dimension record. In this article, let’s discuss the step by step implementation of SCD Type 3 using Informatica PowerCenter. The number of records we store in SCD Type 3 do not increase exponentially as we do not insert a record for each and every historical record. Hence we may not need the performance improvement techniques used in the SCD Type 2 Tutorial. Know more about SCDs at Slowly Changing Dimensions Concepts. The new incoming record (changed/modified data set) replaces the existing old record in target. Comprehensive ETL criteria were identified. Testing procedures were developed. And this work was applied commercial ETL tools. The study covered all major aspects of ETL usage and can be used to effectively compare and evaluate various ETL tools. We can implementation on SCD TYPE-3 based on SCD TYPE-1, SCD TYPE-2 and new fields like Previoussal, Effective Dates.

REFERENCES

Journal Papers:


Books:

Theses:

Proceedings Papers: