A STUDY ON ETL INFORMATICA TRANSFORMATIONS

N. THENMOZHI 1  A. BALAKRISHNAN 2

1 Asst Professor, N.K.R. Govt Arts College for Women, Namakkal, Tamilnadu, India
   Email: nthenmtu@yahoo.com

2 Asst Professor, Govt Arts College, Udumalpet, Coimbatore, Tamilnadu, India, balakrishnan.mca04@gmail.com

Abstract-

Transformation is one of the most critical areas of the ETL process. In terms of processing time for moving data between source to target and costly for the ETL process. Set of rules to transform the data between source and destination if measured data to the confirmed dimension using same units so they can joined later. Data quality of data quality paradigm transformation insists the correct, unambiguous, consistent and complete data. Various Transformations takes raw data from source to staging area and do if any cleaning, reformating and aggregation, etc that is required to get into the final format for reporting. This paper focus on how transformation can be done into ETL process to be implemented in informatica tool. It is easy to understand the working various transformations applied into informatica.

Keywords: ETL, Informatica, Extract, Transform, Load, Source, Destination (Target).

I. INTRODUCTION

What is ETL Process?

The process of extracting data from source systems and bringing it into the data warehouse is commonly called ETL, which stands for extraction, transformation, and loading.

Extract data from operational systems which are primary source for the data warehouse.

Transforming the data-it includes the cleaning, filtering, validating, applying business rules [1].

Loading the data into the data warehousing or applications that houses the data.

II. INFORMATICA TRANSFORMATIONS

A transformation is processes of repository objects that modifies or pass the data source to the target.informatica helps to transform the source data according to the requirements of target system to ensure quality of the data being data loaded into target.

III. TYPES OF TRANSFORMATIONS

There two types of transformation

1. Active transformation

   Active transformations can change the number of rows that pass through from source to destination. It eliminates the rows that do not meet conditions in transformation [2].

2. Passive transformation

   A passive transformation that does not change the number of rows that passes through it. It passes all rows though the transformations [2].

   Transformations can be connected or unconnected.

1. Connected transformation

   A connected transformation is connected to other transformations or directly to the target table in the mapping.

2. Unconnected transformation

   Unconnected transformations do not have to be connected to the data path or data flow. Most
unconnected transformations can be both connected and unconnected, such as store procedure or lookup transformations [3].

### IV. LIST OF TRANSFORMATIONS IN INFORMATICA.

- Source Qualifier Transformation
- Aggregator Transportations
- Application Source Qualifier Transformation
- Custom Transformation
- Data Masking Transformation
- Expression Transformation
- External Procedure Transformation
- Filter Transformation
- HTTP Transformation
- Input Transformation
- Java Transformation
- Joiner Transformation
- Lookup Transformation
- Normalizer Transformation
- Output Transformation
- Rank Transformation
- Reusable Transformation
- Router Transformation
- Sequence Generator Transformation
- Sorter Transformation
- Source Qualifier Transformation
- SQL Transformation
- Stored Procedure Transformation
- Transaction Control Transaction
- Union Transformation
- Unstructured Data Transformation
- Update Strategy Transformation
- XML Generator Transformation
- XML Parser Transformation
- XML Source Qualifier Transformation
- Advanced External Procedure Transformation
- External Transformation

There is lot of transformations can be applied into informatica some of the transformations most frequently used. Without transformations we would not be able to manipulate the data to fit our business need. The transformations into adding business value to the data warehouse.

### V. IMPLEMENTATION OF AGGREGATOR and SORTER TRANSFORMATIONS

**Implementation:**

Import Order, Items, Order_Items tables from the database

1. Calculate the total Order Amount for each Order
2. Create a target, which will show the total order amount in descending order
Below will be the structure of the completed mapping.

I. Import Source and Target Definition

1. Connect to the repository and open the project folder.
2. Import all the sources definitions Orders, Items, Order_Items from the database.
3. Create target table Tgt_OrderListing_x as shown below.

II. Source Qualifier and Aggregator Transformation

1. Create a Source Qualifier transformation and name it SQ_OrderListing_x.
2. Create an Aggregator transformation and group on the Order_id column.
3. Link ports ORDER_ID, DATE_ENTERED, CUSTOMER_ID, QUANTITY, PRICE, DISCOUNT into the Aggregator.
4. Add a new output port Order_Amount.
5. The expression for Order_Amount is SUM(PRICE * QTY – DISCOUNT)
6. Make QUANTITY, PRICE, DISCOUNT only input ports.
III. Create Sorter Transformation

1. To create the Sorter Transformation, use one of the following methods.
   - Select TRANSFORMATION | CREATE and select the Sorter transformation from the drop down. Enter the name as Srt_OrderListing_x or
   - Click on the icon from the Transformations toolbar and rename the transformation to SRT.OrderListing_x.

2. Drag the output ports from Aggregator transformation to Sorter transformation.

IV. Map the Target Columns

1. Link all ports from Sorter Transformation to target table.
2. Your mapping should look like the one as given below:

   ![Sorter Transformation mapping](image)

IV. Load the Target

1. Create a Workflow with the name wf_OrderList_x.
2. Create a session task with the name s_OrderList_x.
3. Run the Workflow.
4. Monitor the Workflow.
5. Verify the results for target table Tgt.OrderListing_x.

Your results should look something like this.

![Sorted orders](image)

Fig: Order_Amount field select descending options

3. Select the Ports tab in the Sorter transformation as shown below. Check the Key column of the Order_Amount port and select Descending from the Direction drop down as shown below.
VI. CONCLUSIONS

The Aggregator transformation calculates aggregates such as sums, minimum or maximum values across multiple groups of rows. The Aggregator transformation can apply expressions to its ports however those expressions will be applied to a group of rows unlike the Expression transformation which applies calculations on a row-by-row basis. Aggregate functions are created in output ports only. Function grouping requirements are set using the Aggregator GroupBy port. Available aggregate expressions: AVG, COUNT, FIRST, LAST, MAX, MEDIAN, MIN, PERCENTILE, STDDEV, SUM. The above example order amount should be summed Order_Amount is SUM (PRICE * QTY – DISCOUNT). Sorter is an active transformation which sorts the incoming data based on one or more key values (in an ascending, descending or mixed order). The Sorter transformation attribute, 'Distinct' provides a facility to remove duplicates from the input rows. The Order_Amount port should be sorted.

VII. REFERENCES.


