In-Memory Technologies and Applications III

Implementation of ETL Workflow

Design Document

Version 1

12/1/2019

TABLE OF CONTENTS

Preface

1. Purpose of this document

This document is a Design Document document for use for In Memory Technology and application Project Implementation of ETL Workflow It provides guidance and template material.

2. Use of this document

This document addresses to the background, requirements and the design to be followed while working with ETL workflows. This document explains architecture, modeling, and implementation process of ETL workflows.

3. Introduction

Extract, transform and Load Processes typically reads data from one or more sources, transform raw data into useful datasets and loads it into target source from where external applications can consume it as required. The sources and targets of an ETL job could be relational databases or some flat files.

ETL Integration takes place in HANA native layer.

4. DATA DESIGN

Application reads data from non cooperative source i.e. external file of type csv.
This file consists of comma separated records with different fields which are first, mi, last, age, gender, birthday, street, city, state, zip, email, phone, dollar, ccnumber, pick(RED|BLUE|YELLOW|GREEN|WHITE), date.
All the transformations are performed such as data filters, data splits and prevent data redundancy.
All the null values are to be handled before storing data into the tables.
Implementation of normalization is done to store the data into a base table.

Data Type Format

Sr No	Column name	Data Type	Length	Unique	Nullable
1	FirstName	NVARCHAR	40	N	N
2	MiddleName	NVARCHAR	40	N	N
3	LastName	NVARCHAR	40	N	N
4	Age	INTEGER		N	N
5	Gender	NVARCHAR	10	N	N
6	DateOfBirth	DATE		N	N
7	Street	NVARCHAR	35	N	N
8	City	NVARCHAR	35	N	N
9	State	NVARCHAR	35	N	N
10	PostCode	NVARCHAR	5	N	N
11	Email	NVARCHAR	25	Y	N
12	Currency	DECIMAL	24,5	N	N
13	ContactNumber	NVARCHAR	20	Y	N
14	Color	NVARCHAR	10	N	Y
15	Date	DATE		N	N

System Design:

Extract :

- Extract relevant data
- Types of extracts: Extract applications (SQL): co-existence with other applications
- DB unload tools: faster than SQL-based extracts
- Extract data from external sources Application should be able to read data and extract data from two types of data: a. Non-Cooperative sources: Flat file of type csv b. Cooperative sources: DB Triggers

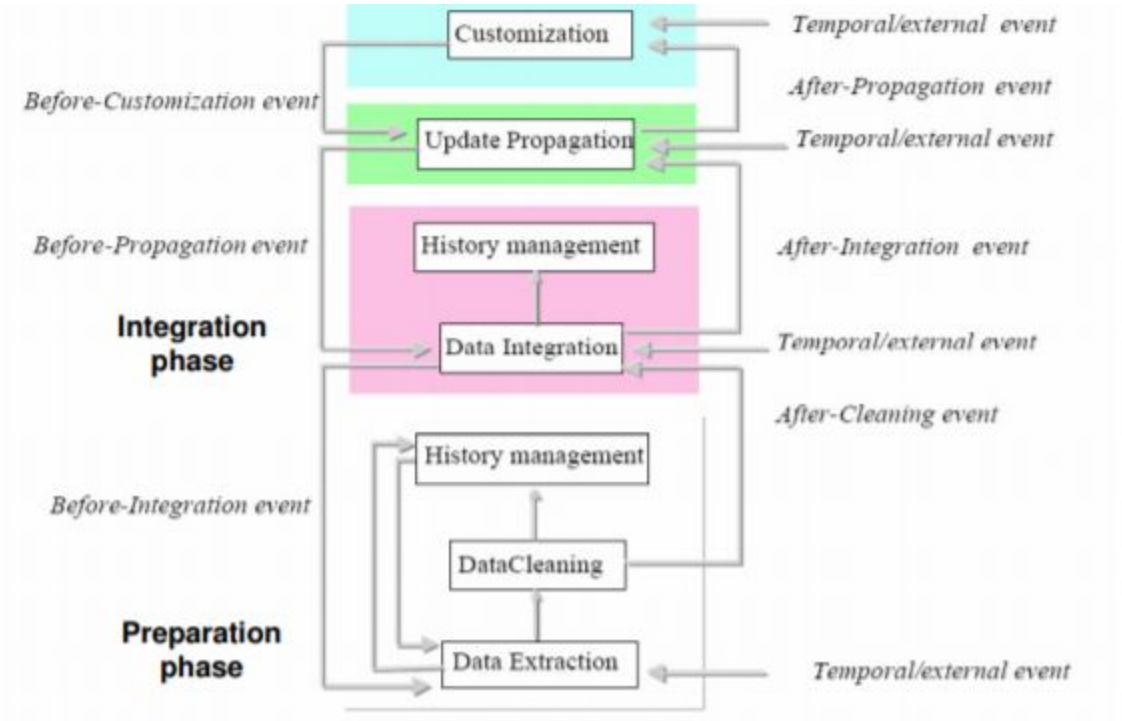
Transform :

- Transform data into normalized/denormalized form as required.
- Data Type Conversions
- String manipulations
- Date-Time formats
- Removal of unnecessary attributes
- Handle inconsistent data formats
- Handle removal of duplicates
- Uniform treatment of NULL
- Build Primary key, foreign key
- Cleansing of data

Load :

- Load data into target
- Build aggregates
- The goal is fast loading that can be achieved by Delta Loading.

Delta Loading (Data warehouse refreshment process): The delta loading process is the loading process in its data flow which captures the differential changes held in the sources and propagates them through the hierarchy of data stores in the data warehouse.



Steps to implement ETL Process:

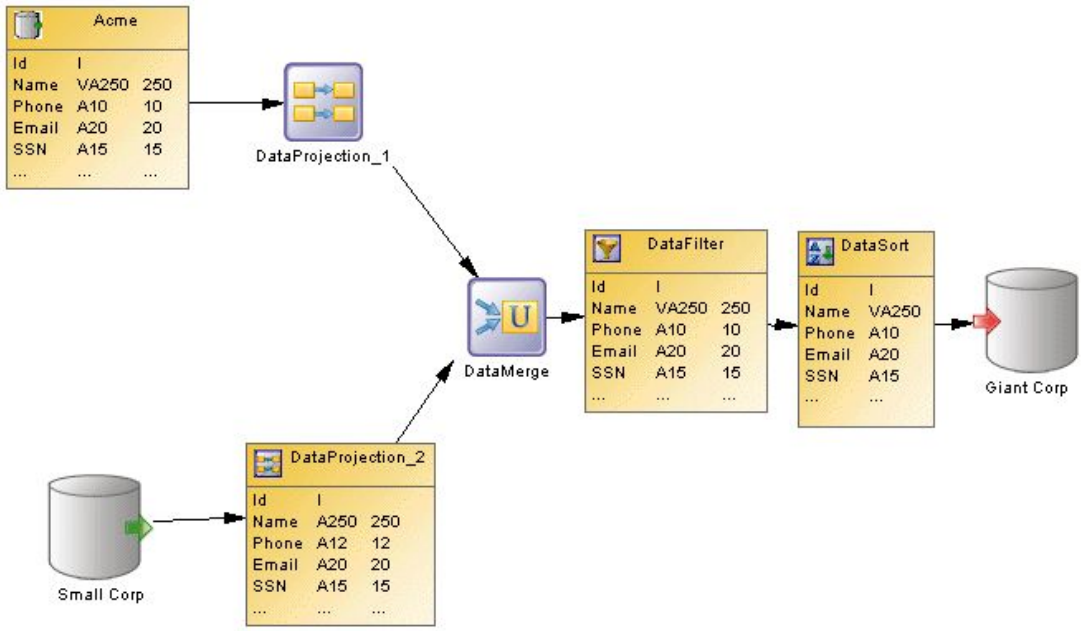
-Creating HANA Workflow
-Modeling ETL HANA Workflow
-Activating HANA Workflow
-Generating HANA Workflow
-Configuring HANA Workflow
-Configuring Generated XS Job
-Versioning HANA Workflow

Creating HANA Workflow:

- Necessary development environment - SAP HANA studio
- Package privileges are required for HANA Workflow
- Creation of HANA Workflow File. Workflow pattern is model template of workflow file. A new file with extension .hprworkflow is created in the SAP HANA package.

Modeling ETL HANA Workflow:

Modeling can be done using data movement, data transformation, and transformation control flow diagrams, and can import and export SAP Data Services project files.
In ETL process, data is extracted from different data sources i.e. Cooperative and non cooperative sources and then merged or sorted according to the required formats and finally loaded to target source.



Activating HANA Workflow:
HANA Workflow file is activated and HANA Workflow template is created. Runtime artifact (design time object) is generated.

Generating HANA Workflow: It automatically creates a set of all necessary XS and SQL artifacts in the package of the .hprworkflow file.

Configuring HANA Workflow: Enable HANA Workflow repository integration and XS Job scheduling.

Configuring Generated XS Job: Configuration and Activation of XS Job to ensure proper execution of HANA Workflow.

Versioning HANA Workflow: Create different versions of HANA workflow. When you create a new version or the first version of a HANA workflow template, new runtime artifacts are automatically created when it is generated.