Building ETL Processes (Vendor)

Software Requirements Document

V 1.0

	Prepared By	Reviewed by	Approved By
Name	Avijit Bhuin		
Role	Intern		
Signature	Avijit Bhuin		
Date			

Contents:

1. Introduction

Welcome to the Software Requirements Document for the ETL (Extract, Transform, Load) Processes built using Talend as the primary ETL tool. This document outlines the design, scope, and functionalities of the Data Warehouse project developed for Cognizant Technology Solutions Ltd. The purpose of this document is to provide a comprehensive understanding of the project's objectives and how the ETL processes are leveraged to build a powerful and centralized data management system.

1.1 About this Document

This document serves as a definitive guide to the usage of Talend for implementing the ETL processes and constructing a robust Data Warehouse that caters to the specific business requirements of Cognizant Technology Solutions Ltd. It includes essential details about the system's architecture, table definitions, mappings, and functional requirements.

-Purpose & Scope of the Document

The primary purpose of this document is to clearly articulate the goals and functionality of the ETL-based Data Warehouse project. It aims to provide a reference for project stakeholders, developers, and users to understand the system's capabilities and limitations.

This document's scope covers the complete end-to-end process of building the data warehouse, creating a centralized repository, and implementing ETL processes using Talend. It describes the functionalities that enable Cognizant Technology Solutions Ltd. to generate various reports, forecast sales, and gain valuable insights for making data-driven decisions.

1.2 Intended Audience

The intended audience for this document includes the project team members involved in the development, testing, and maintenance of the ETL processes and the Data Warehouse. It also targets key stakeholders, project managers, business analysts, and other individuals who need to grasp the project's overall objectives and technical details.

Throughout the document, we will provide a clear and concise explanation of the system's components, Talend's role in the ETL processes, and the impact this system will have on Cognizant Technology Solutions Ltd. 's decision support and reporting capabilities.

2. Software System Overview

The Software System Overview provides a comprehensive understanding of the ETL-based Data Warehouse project developed for Cognizant Technology Solutions Ltd. This section outlines the project's objectives, scope, architecture, and key components, including the table definitions and mappings used in the system.

2.1 About the Software System:

In this section, we provide a comprehensive overview of the software system being developed for Cognizant Technology Solutions Ltd. The system's scope, exclusions, and perspective are discussed to outline the project's boundaries and objectives.

→Scope of the System

The scope of the software system encompasses three main modules, each serving a specific purpose to address Cognizant Technology Solutions Ltd. 's data management and reporting requirements:

-Build Data Warehouse

-Build a Centralized Repository

-Create ETL Processes

These are explained in details later on.

→Exclusions:

While the software system addresses critical aspects of data management and reporting for Cognizant Technology Solutions Ltd. , certain functionalities are explicitly excluded from the project's scope. The following items are not part of the current project:

1. Additional Functionality Beyond Defined Modules

The system will operate only within the confines of the three defined modules: Build Data Warehouse, Build a Centralized Repository, and Create ETL Processes. Any additional functionality outside these modules is considered out of scope for the current project.

2. Integration with Non-Specified Systems

The system will focus on integrating data from specific data sources identified during the project's planning phase. Integration with non-specified or newly introduced systems will be considered as out of scope for the current project.

→System Perspective:

The software system is envisioned as a comprehensive data management and reporting solution designed to enhance Cognizant Technology Solutions Ltd. 's decision support capabilities. The system's perspective is defined by the following key characteristics:

1. Data Warehouse as a Decision Support Platform

The Data Warehouse will serve as the central decision support platform for Cognizant Technology Solutions Ltd. It will facilitate efficient data storage, retrieval, and analysis, empowering decision-makers with timely and accurate information for strategic planning and business insights.

2. Data Integration and Standardization

The system will employ ETL processes to integrate data from multiple sources into the Data Warehouse. Data will be transformed and standardized to ensure consistency, making it suitable for analysis and reporting.

3. Scheduled and On-Demand Data Loads:

The ETL processes will be designed to support both scheduled data loads (daily, monthly, and yearly) and on-demand data loads. This flexibility ensures that data is readily available whenever required by end-users.

4. Centralized Data Access and Reporting:

With the creation of a centralized repository, Cognizant Technology Solutions Ltd. will gain a unified view of sales-related data from various agencies and geographies. The centralized data access will streamline reporting and analysis processes, fostering a data-driven decision-making culture.

2.2 System Architecture

The proposed system follows a well-defined architecture to ensure scalability, maintainability, and performance. The architecture consists of two main layers:

→Physical Architecture:

The physical architecture depicts the arrangement of system elements and interfaces. It includes the presentation layer, business logic layer, and data access layer. Talend, as the ETL tool, will be primarily operating in the data access layer, facilitating data movement and transformation.

→Logical Architecture:

The logical architecture defines the processes required to provide user services. It represents the functional components of the system and how they interact to achieve the project's objectives. Talend's workflows and data integration jobs will be integral parts of the logical architecture, orchestrating the ETL processes seamlessly.

2.3 Table Definitions & Mappings

The Data Warehouse will consist of several tables, each serving a specific purpose in the reporting and analysis process. These tables include:

- **Geography** (Type 1 Dimension): Contains geographical information.

- ***Vendors*** (Type 2 Dimension): Stores vendor-related data with historical tracking.

- **Parts** (Reference Table): Contains information about various parts.

- **Vendor_parts** (Type 2 Dimension): Connects vendors and parts with historical tracking.

- **Customer** (Type 1 Dimension): Contains customer-related data.

- **Part_sales_fact** (Fact Table): Holds sales-related metrics like net sales, product units, order count, discounts, average order value, and average order size.

Talend will be responsible for extracting data from source systems, transforming it into the appropriate format, and loading it into the relevant tables in the Data Warehouse based on these mappings.

3. ETL Tool – Talend

3.1 Overview of Talend

Talend is a powerful and widely-used ETL (Extract, Transform, Load) tool that enables organizations to efficiently integrate, process, and manage data from diverse sources. With its user-friendly graphical interface and robust functionality, Talend empowers developers and data engineers to create sophisticated data integration workflows, making it an ideal choice for building data warehouses and data management solutions.

Key features of Talend include:

-Data Connectivity:

Talend supports a wide range of data sources, including databases, flat files, cloud-based applications, and web services. Its extensive library of connectors facilitates seamless data extraction from various systems.

-Data Transformation:

Talend provides a rich set of data transformation components, enabling users to cleanse, enrich, and format data as required. These transformations play a crucial role in ensuring data quality and consistency.

-Data Loading and Synchronization:

Talend simplifies the process of loading data into target systems, including data warehouses, databases, and cloud platforms. It also supports incremental data loading, ensuring that only the changed or new data is processed during subsequent runs.

-Job Orchestration:

Talend allows users to design complex ETL workflows by visually connecting different components and defining the flow of data through the system. This graphical approach enhances the readability and maintainability of ETL jobs.

-Scalability and Performance:

Talend is designed to handle large volumes of data and can be deployed in distributed environments to achieve scalability and optimize performance.

3.2 How Talend Meets Project Requirements

Talend is well-suited to meet the specific requirements of the ETL-based Data Warehouse project for Cognizant Technology Solutions Ltd. Let's explore how Talend addresses each module's needs:

Build Data Warehouse:

Talend's data integration capabilities enable seamless extraction of sales data from various sources, transformation of data to conform to the Data Warehouse schema, and loading the processed data into the appropriate tables. Its support for batch processing ensures efficient handling of large datasets.

Build a Centralized Repository Module:

Talend's connectivity to multiple data sources allows for the smooth collection of data from agencies situated across different geographies. With Talend's data transformation and consolidation features, data from diverse sources can be standardized and integrated into a centralized repository for unified reporting.

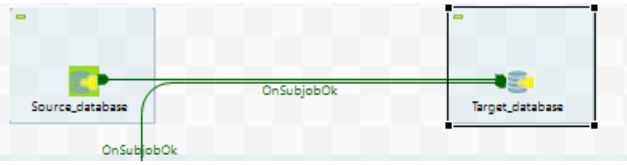
Create ETL Processes:

Talend excels at creating ETL workflows for daily, monthly, yearly, and on-demand data loads. With its scheduling capabilities, Talend jobs can be automated to run at specified intervals, ensuring that the Data Warehouse stays up-to-date with the latest information.

3.3 ETL Workflow Diagrams:

NOTE: For the input and output and the mapping file please refer to the annexure section below.

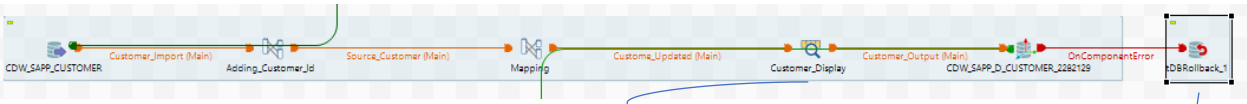
Before starting the job, we need to connect to our databases, so we're using two tdbconnection to do so:



Before Moving to any table, we are establishing a secure connection to our source database and our Target database. This part is to ensure that before we start any operation the databases should be ready. Once they're connected, we can move to a table.

Let's understand each table one by one first:

Customer Table:



INPUT FORMAT:

FIRST_NAME	STRING (40)
MIDDLE_NAME	STRING (40)
LAST_NAME	STRING (40)
SSN	STRING (9)
APT_NO	STRING (7)
STREET_NAME	STRING (30)
CUST_CITY	STRING (30)
CUST_STATE	STRING (30)
CUST_COUNTRY	STRING (30)
CUST_ZIP	STRING (7)
CUST_PHONE	NUMBER (10)
CUST_EMAIL	STRING (40)

This display is for the sake of development only and does not serve anything towards the operation on the data.

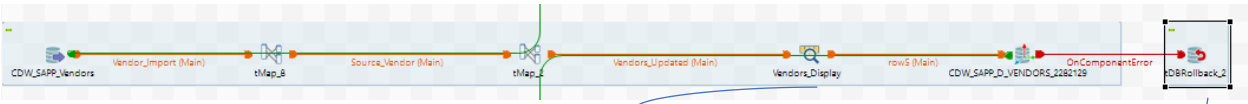
OPERATING ON THE DATA AS PER RREQUIREMENT

OUTPUT FORMAT:

CUST_ID	INT (10)
CUST_F_NAME	VARCHAR (40)
CUST_M_NAME	VARCHAR (40)
CUST_L_NAME	VARCHAR (40)
CUST_SSN	INT (9)
CUST_STREET	VARCHAR (38)
CUST_CITY	VARCHAR (30)
CUST_STATE	VARCHAR (30)
CUST_COUNTRY	VARCHAR (30)
CUST_ZIP	INT (7)
CUST_PHONE	VARCHAR (12)
CUST_EMAIL	VARCHAR (40)

In case of any difficulty or error in the component, it will rollback the database and will not update anything.

Vendor's Table:



INPUT FORMAT:

VENDOR_NAME	STRING (40)
Vendor_SSN	STRING (9)
Vendor_STREET_NAME	STRING (40)
Vendor_CITY	STRING (20)
Vendor_STATE	STRING (20)
Vendor_Country	STRING (20)
Vendor_zip	STRING (7)
Vendor_email	STRING (30)
Vendor_phone	STRING (10)

This display is for the sake of development only and does not serve anything towards the operation on the data.

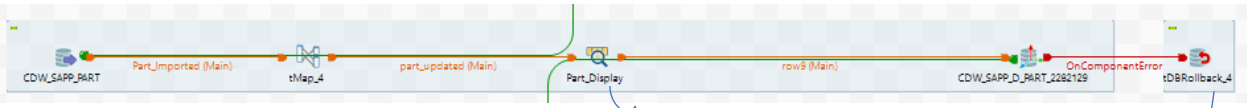
OPERATING ON THE DATA AS PER RREQUIREMENT

OUTPUT FORMAT:

VENDOR_ID	INT (10)
VENDOR_NAME	VARCHAR (40)
Vendor_SSN	INT (9)
Vendor_STREET	VARCHAR (38)
Vendor_CITY	VARCHAR (30)
Vendor_STATE	VARCHAR (30)
Vendor_COUNTRY	VARCHAR (30)
Vendor_ZIP	INT (7)
Vendor_PHONE	VARCHAR (12)
Vendor_EMAIL	VARCHAR (40)

In case of any difficulty or error in the component, it will roll back the database and will not update

Parts Table:



This display is for the sake of development only and does not serve anything towards the operation on the data.

INPUT FORMAT:

PART_Id	STRING (10)
PART_name	STRING (40)
No_of_part	Number (10)
Price	Number (10)

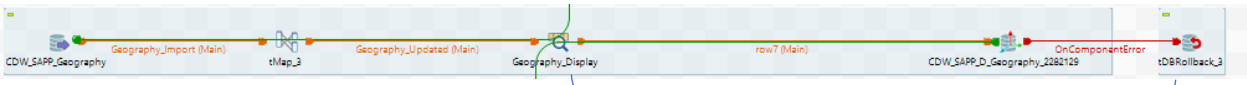
OPERATING ON THE DATA AS PER RREQUIREMENT

OUTPUT FORMAT:

PART_ID	INT (40)
PART_NAME	VARCHAR (30)
NO_OF_PART	INT (10)
Price	INT (10)

In case of any difficulty or error in the component, it will roll back the database and will not update anything.

Geography Table:



This display is for the sake of development only and does not serve anything towards the operation on the data.

INPUT FORMAT:

Geography_Id	STRING (10)
Geography_name	STRING (40)

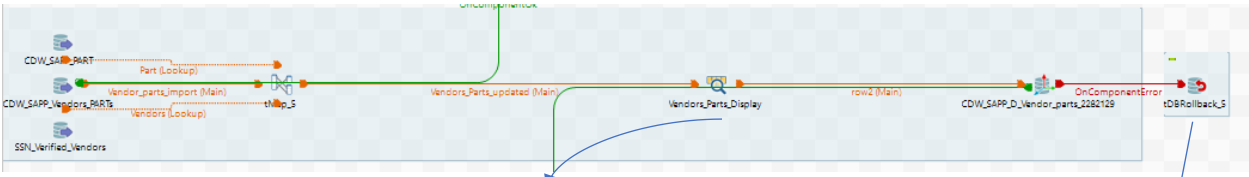
OPERATING ON THE DATA AS PER RREQUIREMENT

OUTPUT FORMAT:

GEOGRAPHY_ID	INT (40)
GEOGRAPHY_NAME	VARCHAR (30)

In case of any difficulty or error in the component, it will roll back the database and will not update anything.

Vendor Parts Table:



This display is for the sake of development only and does not serve anything towards the operation on the data.

INPUT FORMAT:

Vendor Parts Table:

Vendor_Ssn	STRING (9)
PART_Id	STRING (10)

Part Table:

PART_ID	INT (40)
PART_NAME	VARCHAR (30)
NO_OF_PART	INT (10)
Price	INT (10)

Vendors Table:

VENDOR_ID	INT (10)
VENDOR_NAME	VARCHAR (40)
Vendor_SSN	INT (9)
Vendor_STREET	VARCHAR (38)
Vendor_CITY	VARCHAR (30)
Vendor_STATE	VARCHAR (30)
Vendor_COUNTRY	VARCHAR (30)
Vendor_ZIP	INT (7)
Vendor_PHONE	VARCHAR (12)
Vendor_EMAIL	VARCHAR (40)

OPERATING ON THE DATA AS PER RREQUIREMENT

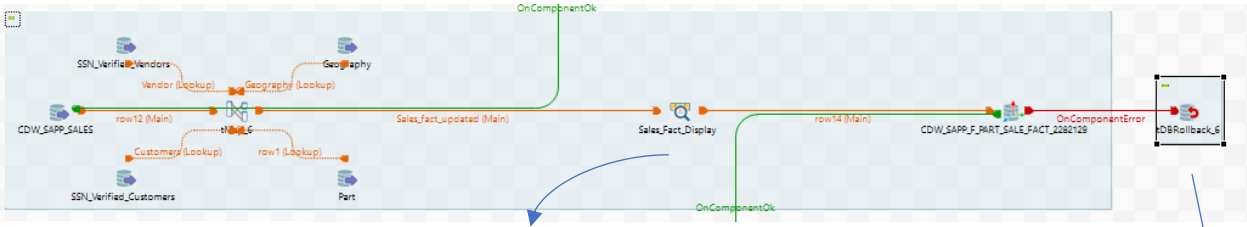
OUTPUT FORMAT:

Vendor_id	INT (10)
Vendor_SSN	INT (9)
PART_ID	INT (10)
PART_NAME	VARCHAR (40)

In case of any difficulty or error in the component, it will roll back the database and will not update anything.

Note that these two tables are imported from the target table and not from the source table. This helps Automatically filter the invalid Vendor_ssn when we perform a inner join between them.

Sales Fact Table:



INPUT FORMAT:

Sales Table:

Cust_ssn	STRING (9)
Vendor_ssn	STRING (9)
Geography_id	STRING (10)
Part_id	STRING (10)

This display is for the sake of development only and does not serve anything towards the operation on the data.

OUTPUT FORMAT:

Vendor_id	INT (10)
CUST_ID	INT (10)
CUST_SSN	INT (9)
Vendor_id	INT (10)
vendor_name	VARCHAR (40)
vendor_SSN	INT (9)
Geography_id	INT (10)
Geography_name	VARCHAR (40)
Part_id	INT (10)
Part_name	VARCHAR (40)
no_of_parts	INT (10)
price	INT (10)
total_price	INT (10)

In case of any difficulty or error in the component, it will roll back the database and will not update anything.

Part Table:

PART_ID	INT (40)
PART_NAME	VARCHAR (30)
NO_OF_PART	INT (10)
Price	INT (10)

Vendors Table:

VENDOR_ID	INT (10)
VENDOR_NAME	VARCHAR (40)
Vendor_SSN	INT (9)
Vendor_STREET	VARCHAR (38)
Vendor_CITY	VARCHAR (30)
Vendor_STATE	VARCHAR (30)
Vendor_COUNTRY	VARCHAR (30)
Vendor_ZIP	INT (7)
Vendor_PHONE	VARCHAR (12)
Vendor_EMAIL	VARCHAR (40)

Customer Table:

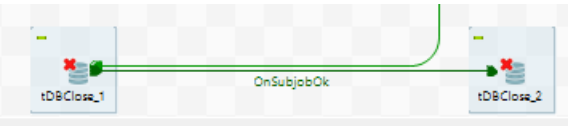
CUST_ID	INT (10)
CUST_F_NAME	VARCHAR (40)
CUST_M_NAME	VARCHAR (40)
CUST_L_NAME	VARCHAR (40)
CUST_SSN	INT (9)
CUST_STREET	VARCHAR (38)
CUST_CITY	VARCHAR (30)
CUST_STATE	VARCHAR (30)
CUST_COUNTRY	VARCHAR (30)
CUST_ZIP	INT (7)
CUST_PHONE	VARCHAR (12)
CUST_EMAIL	VARCHAR (40)

Geography Table:

GEOGRAPHY_ID	INT (40)
GEOGRAPHY_NAME	VARCHAR (30)

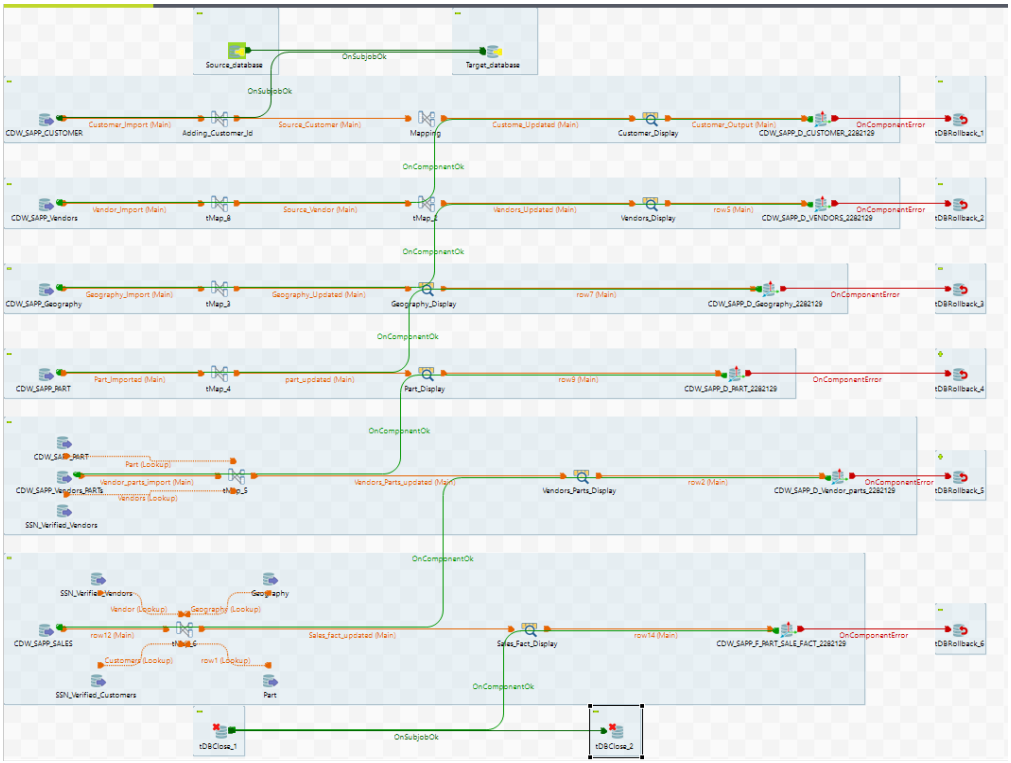
Note that these four tables are imported from the target table and not from the source table. This helps Automatically filter the invalid Vendor_ssn and the invalid custome_ ssn when we perform a inner join between them.

Before exiting the job, we need to close our connection to our database, so we're using two "tdbclose" to do so:



Before exiting the job, we are closing the connection to the database.

NOW HERE IS A OVERVIEW OF THE WHOLE TALEND JOB:



As you can see every table is divided into different sub jobs and every sub job is interconnected with a “On Component OK” component and it’s also connected to a “tdbrollback” component that rolls back the database to the previous save point in case there is some kind of error in any of the sub jobs.

After the job is done we need to build it and then use it along with tack scheduler in order to automate this process every day or every week or every month or according to customer demands.

4. Annexure:

4.1 Terms & Conditions:

-Licensing Information:

The ETL project using Talend is subject to specific licensing terms and conditions. These terms govern the usage, distribution, and modification of the project deliverables. The licensing information is outlined as follows:

1. The ETL project's source code and documentation are the intellectual property of Cognizant Technology Solutions Ltd. and are protected under applicable copyright laws.

2. The project's deliverables, including Talend job designs, ETL workflows, and documentation, are solely for internal use by Cognizant Technology Solutions Ltd. and may not be distributed or shared with external parties without prior written consent.

3. Any modifications or enhancements to the ETL project must comply with Cognizant Technology Solutions Ltd. 's change management process and be approved by the designated project authorities.

4. Cognizant Technology Solutions Ltd. shall not hold Talend responsible for any issues or challenges arising from the usage of the Talend software, as per the terms and conditions set forth by Talend's licensing agreement.

-Data Privacy and Security Policies:

As part of the ETL project's implementation, data privacy and security policies are paramount to safeguard sensitive information. The following policies are to be adhered to:

1. Data Encryption: All sensitive data transmitted between systems and during ETL processes must be encrypted to prevent unauthorized access.

2. Access Controls: Access to the Data Warehouse and related systems shall be granted based on the principle of least privilege, ensuring that only authorized personnel can access sensitive data.

3. Data Anonymization: Personally identifiable information (PII) and other sensitive data must be anonymized or pseudonymized when not required for specific reporting or analysis.

4. Data Retention: Data retention policies shall be defined to manage the storage and archival of data in compliance with legal and regulatory requirements.

5. Audit Logging: Comprehensive audit logs shall be maintained to track data access, changes, and user activities for accountability and troubleshooting purposes.

6. Disaster Recovery: Robust data backup and disaster recovery mechanisms shall be established to ensure data availability and continuity in case of unforeseen events.

4.2 Appendix:

-Source File:

All the data used as source to test this project has been provided in this file, separated in different sheets for each table.

In Case this file is not opening try this link: Source_Files_Used.xlsx (Please use your Cognizant account to view)

-Mapping Document:

In this file all the mapping has been documented. For each table we have a separate worksheet.

In Case this file is not opening try this link: Mapping_Document.xlsx (Please use your Cognizant account to view)

-Target File:

All the generated output is here.

In Case this file is not opening try this link: Target_File_Generated.xlsx (Please use your Cognizant account to view)

Project:

THANK YOU