Table of Contents

[Design & Development 2](#_Toc162614214)

[Data Validation 4](#_Toc162614215)

[ETL Validation using python scripting 5](#_Toc162614216)

# Design & Development

|  |  |
| --- | --- |
| **Designer Window** | |
| Design Mapping      Transformations in this Mapping   1. Source Qualifier 2. Expression Transformation 3. Unconnected Look Up Transformation 4. Filter Transformation 5. Sorter Transformation   Filter Transformation | Create Transformation  Source Qualifier (SQL Overwrite can be used to join tables or filter records, to avoid transformations used in the Informatica – which improves performance)  Unconnected Lookup Transformation |
| Expression Transformation | Sorter Transformation |

|  |  |
| --- | --- |
| **Workflow Manager** | |
| Work flow | Configure Connections |

# Data Validation

|  |  |
| --- | --- |
| **Workflow Monitor** | |
| Run Properties | Session Logs |
| **Source Vs Target** | |
| Source | Target |

# ETL Validation using python scripting

|  |  |
| --- | --- |
| **Python scripting** | |
| Config Connection Details | Import Necessary Modules |
| Connect to source (Oracle DB) & load queries to DataFrame | Connect to Target (pgAdmin) & load queries to DataFrame |
| Perform Data Validation with Pandas and write the results to a file | |
| Data Validation results | |

Design & Development

* Import source and target details to the Designer
* Design Mapping from source to target with necessary transformation
* This is Cross DB ETL. Source is **Oracle** and Target is **Pgadmin**
* Lookup Transformation is used to fetch department name
* Filter transformation to limit the entry of data into target table
* Expression transformation used to replace null values for designation
* Sorter transformation is used to get distinct employee ids and to sort employees in Ascending order

Data Validation

* Check Run properties and session logs for the detailed report
* Check Count in Source and Target tables
* Check all the transformation logic are fulfilled such as (duplicates not loaded, Null values replaced)
* Check any data is truncated
* Using python scripts automate the testing of data validation and get results into a text file.

ETL validation using Python scripting

* Connect to Source: oracle DB from Config file
* Read Source queries and load to DataFrame
* Connect to Target: pgAdmin from config file
* Read Target Queries and load to DataFrame
* Validate data with the help of Pandas module in python
* Check for count, duplicates, null, truncation and transformation in source and target
* Write test results into a separate file