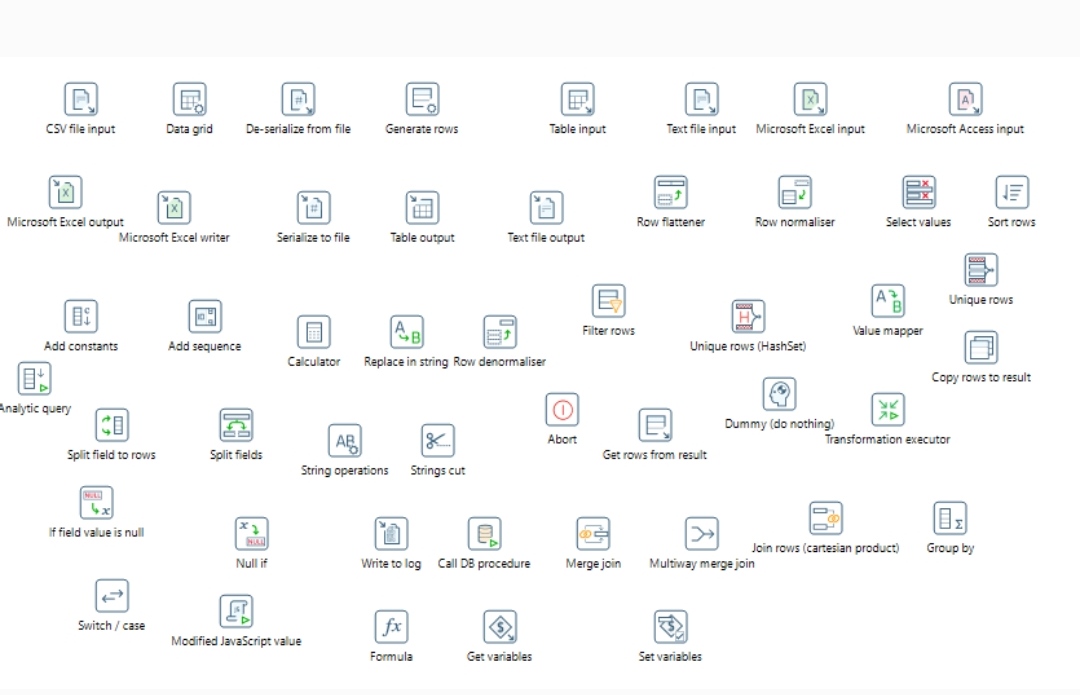
PENTAHO DATA INTEGRATION

-Task for interns

Prepared by

* Sony Shrestha

Important Steps for creating transformation and job



1. Create four databases named fc\_raw, fc\_rw, fc\_master, fc\_facts and QA.
2. Create a pentaho transformation that takes input from respective csv and excel files provided in mail and dump them into tables named:
3. fc\_rw.fc\_account\_master
4. fc\_rw.fc\_balance\_summary
5. fc\_rw.fc\_transaction\_base

Here,

In fc\_transaction\_base tables, value of column dc\_indicator(deposit,withdraw) should be transformed into dc\_indicator(D,C) 🡪 modified java script, value\_mapper

In fc\_account\_master table, value of column active\_flag(‘ACTIVE’, ‘INACTIVE’). should be transformed into active\_flag(1,0) 🡪 modified java script, value\_mapper

All of the above tables should contain following additional columns as well:

created\_by with default value of your name

created\_on with default value of current timestamp

Note: data type of columns in respective table should be checked properly.

1. Create a pentaho transformation to find avg\_monthy\_balance,std\_monthly\_balance.

The result should be dumped into table named fc\_facts.fc\_balance\_facts. This table should contain following fields:

account\_number

avg\_monthly\_balance

std\_monthly\_balance

created\_by

created\_on

* 1. avg\_monthly\_balance

Find avg(lcy\_amount) in monthly level

avg\_monthly\_balance=avg(above amount) for each account

* 1. std\_monthly\_balance

Find avg(lcy\_amount) in monthly\_level

std\_monthly\_balance=std(above amount) for each account

All null values whould be replaced with 0 in these facts.

1. Create a pentaho transaformation to find avg\_monthy\_deposit, std\_monthly\_deposit, avg\_monthly\_withdraw, std\_monthly\_withdraw.

account\_number

avg\_monthly\_deposit

std\_monthly\_deposit

avg\_monthly\_withdraw

std\_monthly\_withdraw

created\_by

created\_on

* 1. avg\_monthly\_deposit

Find sum(lcy\_amount) in monthly level where dc\_indicator= ‘C’

avg\_monthly\_deposit=avg(above amount) for each account

* 1. std\_monthly\_deposit

Find sum(lcy\_amount) in monthly level where dc\_indicator= ‘C’

std\_monthly\_deposit=std(above amount) for each account

* 1. avg\_monthly\_withdraw

Find sum(lcy\_amount) in monthly level where dc\_indicator= ‘D’

avg\_monthly\_deposit=avg(above amount) for each account

* 1. std\_monthly\_withdraw

Find sum(lcy\_amount) in monthly level where dc\_indicator= ‘D’

std\_monthly\_deposit=std(above amount) for each account

All null values should be replaced with 0.

1. Create a pentaho transformation that generates balance of today and last three days for which entry is present.

The result should be dumped into table named fc\_facts.fc\_balance\_last\_3\_days.

Resulting table should contain following fields:

account\_number

tran\_date

balance

balance\_before\_1\_day

balance\_before\_2\_days

balance\_before\_3\_days

created\_by

creted\_on

1. Write a SQL script that generate balance of today and last 3 days for which entry is present.
2. Write a SQL script to compare result generated from point 5 and 6 using column name status for balance of each day.
3. Result generated from point 7 should be dumped into table named QA.qa\_fc\_balance\_last\_3\_days.

This table should contain following fields:

account\_number

customer\_code

balance\_qa

balance

balance\_before\_1\_day\_qa (result from point 6)

balance\_before\_1\_day (result from point 5)

balance\_before\_1\_day\_status

balance\_before\_2\_days\_qa (result from point 6)

balance\_before\_2\_days (result from point 5)

balance\_before\_2\_days\_status

balance\_before\_3\_days\_qa (result from point 6)

balance\_before\_3\_days (result from point 5)

balance\_before\_3\_days\_status

created\_by

created\_on

1. Write a SQL procedure for point 8.
2. Create a pentaho transformation to call procedure created in point 9.
3. Create a master table named fc\_master.fc\_clients from fc\_rw.fc\_account\_master
4. This table should contain all customers in account master table.

id

customer\_code

active\_flag

created\_by

created\_on

modified\_by

modified\_on

1. In fc\_rw, we have created three tables.
2. Create same tables in fc\_raw with some of the column names changed (only table structure)
3. Create two tables in fc\_master as:

fc\_master.fc\_tables with data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id | Source\_schema | Source\_table | Destination\_schema | Destination\_table |
| 1 | fc\_rw | fc\_account\_master | fc\_raw | fc\_account\_master |
| 2 | fc\_rw | fc\_transaction\_base | fc\_raw | fc\_transaction\_base |
| 3 | fc\_rw | fc\_balance\_summary | fc\_raw | fc\_balance\_summary |

fc\_master.fc\_fields\_mapping with data

|  |  |  |  |
| --- | --- | --- | --- |
| id | Table\_id | Source\_field | Destination\_field |
| 1 | 1 | account\_number | acc\_num |
| 1 | 1 | customer\_code | cust\_code |
|  | (table\_id from fc\_tables table) | column name of table in source (fc\_rw) | column name of table in destination (fc\_raw) |
| ……. | ……….. | …………. | …….. |

Basically, second table describes the mapping of field between source and destination (add more entries for all columns in fc\_account\_master, fc\_transaction\_base, fc\_balance\_summary).

1. Create a pentaho transformation to load data from tables in fc\_raw to fc\_rw based on mappings specified in table.

Note:

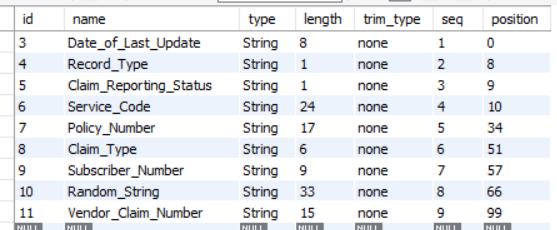
Here, source\_schema, destination\_schema, source\_table, destination\_table should be dynamic (config driven).

Fields mapping (field name in source table should be dynamic)

1. Create a pentaho job that executes all above transformations one after another.
2. If execution is successful, send success mail.
3. If execution is unsuccessful, send error message stating that execution of pentaho job was not successful.

Task A

1. Accept .fixed file as input
2. This is a fixed file, whose fields are not delimited
3. Fields of this file are defined in table dpp\_etl.reformat\_canonical\_map based on position and length



1. Prepare PDI transformation that will generate .conversion file equivalent to .fixed file delimited by |#^#|
2. Note: Fields should be config driven