1. Create a pentaho transformation that takes input from respective csv and excel files provided in mail and dump them into tables named:
2. account\_master
3. balance\_summary
4. transaction\_base

Here,

In transaction\_base tables, value of column dc\_indicator (deposit, withdraw) should be transformed into dc\_indicator (D,C)

In account\_master table, value of column active\_flag (‘ACTIVE’, ‘INACTIVE’). should be transformed into active\_flag (1,0)

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

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

account\_number

avg\_monthly\_balance

std\_monthly\_balance

* 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 should be replaced with 0 in these facts.

1. Create a pentaho transformation 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

* 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 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

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\_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

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 md\_clients from account\_master
4. This table should contain all customers in account master table.

id

customer\_code

1. Create a pentaho transformation to load data from tables in source database to destination database 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)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id | Source\_schema | Source\_table | Destination\_schema | Destination\_table |
| 1 |  | account\_master |  | account\_master |
| 2 |  | transaction\_base |  | transaction\_base |
| 3 |  | balance\_summary |  | balance\_summary |

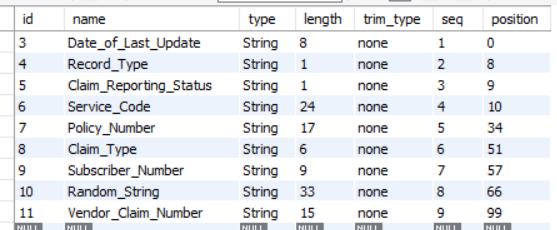
Config\_fields\_mapping with data

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

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



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Id | Name | type | length | trim\_type | Seq | position |
| 1 | employee\_id | String |  | both | 1 | 0 |
| 2 | full\_name | String |  | both | 2 |  |
| 3 | address | String |  | both | 3 |  |
| 4 | mobile\_number | Integer |  | both | 4 |  |
| 5 | email\_id | String |  | both | 5 |  |

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