Sep 2019

|  |
| --- |
| BEA\_IFRS9\_ECL\_Customization\_Specification |
| prepared for MA Internal Use only |

Version: 0.02 (draft)

Date: 2019.10.10

# Version Control

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PROJECT NAME** | | **BEA IFRS9 For ECL Calculator** | | |
| **DOCUMENT STATUS** | | **V0.01 (10th Oct 2019)** | | |
| **Version** | **Date** | **Author(s)** | **Reviewer(s)** | **Summary of Changes** |
| 0.01 | 25 Sept 2019 | Larry |  | Initial Draft (TOC) |
| 0.02 | 10 Oct 2019 | Larry |  | Added final rating, PD, LGD, stage allocation and CRM allocation business logic |
| 0.03 | 23 Oct 2019 | Larry |  | Added CRM allocation code Logic |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Table of Contents

[Version Control 2](#_Toc21618168)

[Table of Contents 3](#_Toc21618169)

[Introduction 5](#_Toc21618170)

[1.1 Document Objective 5](#_Toc21618171)

[1.2 Reference Documents 5](#_Toc21618172)

[1.3 Document Scope 5](#_Toc21618173)

[1.4 Document Structure 5](#_Toc21618174)

[1.5 Abbreviations 5](#_Toc21618175)

[2 Simulation 7](#_Toc21618176)

[2.1 Business Requirement 7](#_Toc21618177)

[2.2 Data Requirement 7](#_Toc21618178)

[2.3 Proposed Design 7](#_Toc21618179)

[2.4 Procedure definition 8](#_Toc21618180)

[3 Borrowers,Final rating,PD 9](#_Toc21618181)

[3.1 Business Requirement 9](#_Toc21618182)

[3.2 Data Requirement 9](#_Toc21618183)

[3.3 Proposed Design 11](#_Toc21618184)

[3.4 Procedure definition 12](#_Toc21618185)

[4 Deals 12](#_Toc21618186)

[4.1 Business Requirement 12](#_Toc21618187)

[4.2 Data Requirement 13](#_Toc21618188)

[4.3 Proposed Design 13](#_Toc21618189)

[4.4 Procedure definition 14](#_Toc21618190)

[5 Summary of Risk Mitigants 14](#_Toc21618191)

[5.1 Business Requirement 14](#_Toc21618192)

[5.2 Data Requirement 14](#_Toc21618193)

[5.3 Proposed Design 16](#_Toc21618194)

[5.4 Procedure definition 16](#_Toc21618195)

[6 Risk Mitigants Contracts 16](#_Toc21618196)

[6.1 Business Requirement 16](#_Toc21618197)

[6.2 Data Requirement 17](#_Toc21618198)

[6.3 Proposed Design 18](#_Toc21618199)

[6.4 Procedure definition 18](#_Toc21618200)

[7 CRM Allocation Compute,LGD 18](#_Toc21618201)

[7.1 Business Requirement 18](#_Toc21618202)

[7.2 Data Requirement 19](#_Toc21618203)

[7.3 Proposed Design 20](#_Toc21618204)

[7.4 Procedure definition 21](#_Toc21618205)

[8 ECL Results 21](#_Toc21618206)

[8.1 Business Requirement 21](#_Toc21618207)

[8.2 Data Requirement 21](#_Toc21618208)

[8.3 Proposed Design 24](#_Toc21618209)

[8.4 Procedure definition 24](#_Toc21618210)

[9 Users, Roles, Access & Authentication 24](#_Toc21618211)

[9.1 Business Requirement 24](#_Toc21618212)

[9.2 Data Requirement 24](#_Toc21618213)

[9.3 Proposed Design 25](#_Toc21618214)

[9.4 Procedure definition 25](#_Toc21618215)

# Introduction

## Document Objective

The aims of this document is to detail migrate the customization from IFRS9 to ECL calculator in order to implement the customization work of the overall design as documented in the Functional Specification Design (“FSD”) as per signed off version for the BEA IFRS9 System Project.

## Reference Documents

This document should be based on the Design documents:

* Technical Specification Document (“TSD”)
* Functional Specification Document(“FSD”)

With the supplement details attached with this document:

* [Intentionally Blank]

## Document Scope

The scope of this document will cover the requirements addressed in the FSD during the business review discussions workshops with BEA user (“BEA”). Any requirements or discussions that are not addressed as above will consider as out of scope for this project.

## Document Structure

The structure of the document as follow:

* <<To be update>>

## Abbreviations

|  |  |  |
| --- | --- | --- |
| **Category** | **Code / Abbreviation** | **Description** |
| **Organization** | BEA | Bank of East Asia |
|  | IFRS | International Finance Reporting standard |
|  |  |  |
|  | BCBS | Basel Committee on Banking Supervision |
|  |  |  |
|  |  |  |
| **Risk System** | RFo | Risk Foundation datamart |
|  | RD | Reporting Date |
|  | WS | Workspace |
|  | RCo | Risk Confidence |
| **Documents** | TSD | Technical Specification Document |
|  | FSD | Functional Specification Document |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Simulation

## Business Requirement

## Data Requirement

**Pre-requisites**

Input of this customization:

* All transactions are loaded to PRICING\_SIMULATION

**Requirement**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **PRICING\_SIMULATION** | | | | | |
| **Table type** | Data | | | | |
| **Description** | All the simulations information are stored in non partitioned / regular table called PRICING\_SIMULATION with following structure. | | | | |
| **Column** | **Type** | **Description** | **Primary Key** | **Nullable** | **Example** |
| ID | number | primary key  Auto computed based on maximum number available/ sort of sequence | Y | N | 123456789 |
| DESCRIPTION | varchar2 | description | N | Y | this simulation is to .... |
| AS\_OF\_DATE | datetime | as of date | N | N | 2020/01/01/ |
| BORROWER\_NAME | varchar2 | borrower name  based on existing borrower or simulated borrower | N | N | Perter Park |
| EXISTING\_ENTITY\_CODE | varchar2 | Existing Borrower Code  if available in case of Edit | N | N |  |
| SIMULATED\_ENTITY\_CODE | varchar2 | Simulated Borrower Code | N | N | (Not Applicable to BEA at this stage) |
| USER\_NAME | varchar2 | created by  Automatically set from CD\_USERS.USER\_NAME field | N | N | Yao Ming |
| CREATED\_AT | datetime | create at  Automatically set when User creates new Simulation | N | N | 2019/02/24 |
| UPDATED\_AT | datetime | update at  Automatically set when the Simulation API completes Successfully or Error out | N | N | 2019/12/24 |
| STATUS | number | 1: 'In Progress'  2: 'Failed'  3: 'Success' | N | N | 1 |
| REMARK | varchar2 | remark | N | Y |  |

## Proposed Design

## Procedure definition

CREATE TABLE PRICING\_SIMULATION

1. Borrowers,Final rating,PD

## Business Requirement

Upon end user search request, search the table ENTITY and retrieve the results for borrowers by matching field mentioned in below table CONTAINS/LIKE operator. View returns only one record for a borrower.

Editor-borrower ratings can be adjusted

## Data Requirement

**Pre-requisites**

* Counterparty / Issuer / Guarantor’s INTERNAL and / or EXTERNAL rating are all loaded to ISSUER\_CREDIT\_RATINGS tables.
* BEA\_RATING\_MASTER configuration table is uploaded by BEA user with all the internal and external rating information provided with the corresponding, rating order, score, and also Moody’s equivalent rating, if applicable, provided as well.
* BEA\_OVERRIDE\_PD table is also uploaded by BEA user
* These information will then be loaded to RATING table.

**Requirement**

1. Overriding Rating with the Long Run average PD
2. Get the worst rating
3. Conversion to Moody’s equivalent rating
4. Backfilling PD

See “PACK\_BEA\_I9\_CUSTO.gen\_final\_issuer\_rating” for details

从视图V\_PRICING\_ENTITY视图结果中查找用户，视图V\_PRICING\_ENTITY的主表为ENTITY和ISSUER\_CREDIT\_RATINGS。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| V\_PRICING\_ENTITY | | | | | |
| **Description** |  | | | | |
| **Column** | **Type** | **Description** | **Primary Key** | **Nullable** | **Example** |
| AS\_OF\_DATE | datetime | as of date |  |  | 2020/01/01/ |
| ENTITY\_CODE | varchar2 | simulation\_id from pricing\_simulation table |  |  |  |
| ENTITY\_DESC | varchar2 | entity code |  |  |  |
| LONG\_NAME | varchar2 | Chinese name |  |  |  |
| ET\_CODE\_USER | varchar2 |  |  |  |  |
| COUNTRY\_NAME | varchar2 |  |  |  |  |
| CI\_ACCOUNT\_NUMBER | varchar2 |  |  |  |  |
| CI\_DOCUMENT\_NUMBER | varchar2 |  |  |  |  |
| CI\_DOCUMENT\_TYPE | varchar2 |  |  |  |  |
| CI\_ISSUE\_COUNTRY\_NAME | varchar2 |  |  |  |  |
| INDUSTRY\_SECTOR | varchar2 |  |  |  |  |
| TOTAL\_ASSETS | number |  |  |  |  |
| IMPORT\_SOURCE | varchar2 |  |  |  |  |
| STAGING\_SOURCE | varchar2 |  |  |  |  |
| MOODYS\_RATING | varchar2 |  |  |  |  |
| SP\_RATING | varchar2 |  |  |  |  |
| CAR\_RATING | varchar2 |  |  |  |  |
| INTERNAL\_RATING | varchar2 |  |  |  |  |
| FINAL\_EXTERNAL\_RATING | varchar2 |  |  |  |  |
| FLAG\_SIMULATED | number |  |  |  |  |
| REMARK | varchar2 |  |  |  |  |
| IMPORT\_SOURCE | varchar2 |  |  |  |  |

## Proposed Design

Borrower Screen User Actions

Always create New record when user Duplicate/Edit/create New borrower Record. Application should not modify the existing data loaded by ETL Batch. following explains the overall flow

New/Edit/Duplicate Action Flow

For New/ Duplicate, Create new record with following keys,

ENTITY.ENTITY\_CODE and ISSUER\_CREDIT\_RATINGS.ENTITY\_CODE = <provided by user>+'\_'+<simulation\_id>

PRICING\_SIMULATION.EXISTING\_ENTITY\_CODE = Null PRICING\_SIMULATION.SIMULATED\_ENTITY\_CODE = <provided by user>+'\_'+<simulation\_id

For Edit, Create new record with following keys if the selected ENTITY record is NOT SIMULATED entity. check the IMPORT\_SOURCE = 'PRICING\_SIMULATION' on the ENTITY record. Else Edit the same record,

ENTITY.ENTITY\_CODE and ISSUER\_CREDIT\_RATINGS.ENTITY\_CODE = <selected record entity\_code>+'\_'+<simulation\_id>

PRICING\_SIMULATION.EXISTING\_ENTITY\_CODE = <selected record entity\_code> PRICING\_SIMULATION.SIMULATED\_ENTITY\_CODE = <selected record entity\_code>+'\_'+<simulation\_id>

Derive various codes for names entered by user using lookup service.

ENTITY.INDUSTRY\_SECTOR = INDUSTRY\_SECTOR.CODE matching "industry\_sector" description selected by user from drop down

ENTITY.COUNTRY\_CODE = COUNTRY.COUNTRY\_CODE matching "country\_of\_operation" and "ci\_issue\_country\_name" selected by user from drop down

Set some default attributes

ENTITY / ISSUER\_CREDIT\_RATINGS.IMPORT\_SOURCE = 'PRICING\_SIMULATION' ENTITY / ISSUER\_CREDIT\_RATINGS.CHECK\_ERROR\_STATUS = 0 ENTITY / ISSUER\_CREDIT\_RATINGS.CHECK\_ERROR\_STATUS\_V = 0

Save Data into various RFO tables with changes done by user in the modal window.

Insert or update multiple records.

in case user selected the ENTITY which is NOT simulated

find if the simulated record available for the same EXISTING\_ENTITY\_CODE , SIMULATION\_ID and AS\_OF\_DATE then update the same record.

Else create new simulated record

Insert/Update ENTITY (all fields as provided in the table above)

Insert/Update ISSUER\_CREDIT\_RATINGS (all fields as provided in the table above)

note then we insert one record for each rating provided by user. e.g. if user provided moodys\_rating, sp\_rating, internal\_rating, car\_rating. we insert/update 4 records in this table.

Post Save PL/SQL Call PACK\_PRICING.UPDATE\_ENTITY following is Pseudo code

Purpose : anything BEA wants to update after entity record is saved.

Pseudo Code TBD by Moody's Partner team

Input Parameters:

p\_simulation\_id (PRICING\_SIMULATION.ID)

p\_as\_of\_date (PRICING\_SIMULATION.AS\_OF\_DATE)

p\_entity\_code (PRICING\_SIMULATION.EXISTING\_ENTITY\_CODE)

p\_simulation\_entity\_code (PRICING\_SIMULATION.SIMULATED\_ENTITY\_CODE)

p\_user\_name (PRICING.USER\_NAME)

Return Successfully if not error Else Show Error on the Screen for User to take screen shot and communicate back to Admin team.

Next Button to Go to Deals Screen

## Procedure definition

CREATE VIEW V\_PRICING\_ENTITY

CREATE PACK\_PRICING.UPDATE\_ENTITY

1. Deals

## Business Requirement

anything BEA wants to update before deals are loaded.

Deal record does not exist to create a new simulated deal record

## Data Requirement

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **PRICING\_DEALS\_EXT** | | | | | |
| **Table type** | Admin | | | | |
| **Description** | This table will be a non-partition table that accumulate and stored the ECL user &&BU Dept information | | | | |
| **Column** | **Type** | **Description** | **Primary Key** | **Nullable** | **Example** |
| User\_id | Number | User\_id corresponding to CD\_USERS table | Y | N | 3 |
| user\_name | varchar2 |  | N | N |  |
| bu\_dept\_codes | varchar2 |  | N | N |  |
| default\_bu\_dept\_code | varchar2 |  | N | N |  |
| role | varchar2 |  | N | N | 123456789 |
| bu\_dept\_codes | varchar2 |  | N | N | 153507.00 |

## Proposed Design

The newly simulated deal record is stored in the PRICING\_DEALS\_EXT table

PRICING\_DEALS\_EXT table is updated by PACK\_PRICING.PREPARE\_DEALS and PACK\_PRICING.UPDATE\_DEAL

procedures.

Call PACK\_PRICING.PREPARE\_DEALS

Input Parameters:

p\_simulation\_id (PRICING\_SIMULATION.ID) p\_as\_of\_date (PRICING\_SIMULATION.AS\_OF\_DATE)

p\_entity\_code (PRICING\_SIMULATION.EXISTING\_ENTITY\_CODE)

p\_simulation\_entity\_code (PRICING\_SIMULATION.SIMULATED\_ENTITY\_CODE)

p\_user\_name (PRICING.USER\_NAME) Return Successfully if not error Else Show Error on the Screen for User to take screen shot and communicate back to Admin team.

Call PACK\_PRICING.UPDATE\_DEAL

Input Parameters:

p\_simulation\_id (PRICING\_SIMULATION.ID)

p\_as\_of\_date (PRICING\_SIMULATION.AS\_OF\_DATE)

p\_entity\_code (PRICING\_SIMULATION.EXISTING\_ENTITY\_CODE)

p\_simulation\_entity\_code (PRICING\_SIMULATION.SIMULATED\_ENTITY\_CODE)

p\_user\_name (PRICING.USER\_NAME)

p\_instrument\_type (V\_PRICING\_DEALS.INSTRUMENT\_TYPE)

p\_contract\_reference (V\_PRICING\_DEALS.CONTRACT\_REFERENCE)

Return Successfully if not error Else show full error for User to further rectify with Admin

## Procedure definition

CREATE TABLE PRICING\_DEALS\_EXT

CREATE PACK\_PRICING.PREPARE\_DEALS

CREATE PACK\_PRICING.UPDATE\_DEAL

1. Summary of Risk Mitigants

## Business Requirement

## Data Requirement

**Pre-requisites**

Input of this customization:

V\_PRICING\_CRM.AS\_OF\_DATE = CONTEXTS.PK\_RD (converted to Date format) where

CONTEXTS.WORKSPACE = 'PRICING' and POSITION = 0

Match the Borrower deals based on BU Dept code of the user to fetch the relevant

GUAR\_CONTRACT\_REF values and GUAR\_TABLE\_NAME from CONTRACT\_GUARANTEE

table.

**Requirement**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| V\_PRICING\_CRM | | | | | |
| **TABLE TYPE** | VIEW | | | | |
| **Description** | Forming a view of transaction records for UI teams to read directly in SQL | | | | |
| **Column** | **Type** | **Description** | **Primary Key** | **Nullable** | **Example** |
| as\_of\_date |  |  |  |  |  |
| crm\_type |  |  |  |  |  |
| simulated |  |  |  |  |  |
| reference |  |  |  |  |  |
| contract\_type |  |  |  |  |  |
| contract\_type\_code |  |  |  |  |  |
| crm\_value\_in\_currency |  |  |  |  |  |
| currency |  |  |  |  |  |
| value |  |  |  |  |  |
| guarantee\_percentage |  |  |  |  |  |
| joint |  |  |  |  |  |
| remark |  |  |  |  |  |
| link\_to\_borrower |  |  |  |  |  |
| import\_source |  |  |  |  |  |
| beneficiary |  |  |  |  |  |
| remaining\_collateral |  |  |  |  |  |
| as\_of\_date |  |  |  |  |  |
| crm\_type |  |  |  |  |  |
| simulated |  |  |  |  |  |
| reference |  |  |  |  |  |
| contract\_type |  |  |  |  |  |
| contract\_type\_code |  |  |  |  |  |
| crm\_value\_in\_currency |  |  |  |  |  |
| currency |  |  |  |  |  |
| value |  |  |  |  |  |
| guarantee\_percentage |  |  |  |  |  |
| joint |  |  |  |  |  |
| remark |  |  |  |  |  |
| link\_to\_borrower |  |  |  |  |  |
| import\_source |  |  |  |  |  |
| beneficiary |  |  |  |  |  |
| remaining\_collateral |  |  |  |  |  |

## Proposed Design

## Procedure definition

CREATE VIEW V\_PRICING\_CRM

1. Risk Mitigants Contracts

## Business Requirement

## Data Requirement

**Pre-requisites**

Input of this customization:

**Requirement**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| V\_PRICING\_DEALS | | | | | |
| **TABLE TYPE** | VIEW | | | | |
| **Description** | Forming a view of transaction records for UI teams to read directly in SQL | | | | |
| **Column** | **Type** | **Description** | **Primary Key** | **Nullable** | **Example** |
| as\_of\_date |  |  |  |  |  |
| instrument\_type |  |  |  |  |  |
| simulated |  |  |  |  |  |
| contract\_reference |  |  |  |  |  |
| contract\_type |  |  |  |  |  |
| contract\_type\_code |  |  |  |  |  |
| trade\_date |  |  |  |  |  |
| maturity\_date |  |  |  |  |  |
| principal\_amount |  |  |  |  |  |
| principal\_amount\_in\_currency |  |  |  |  |  |
| currency |  |  |  |  |  |
| outstanding |  |  |  |  |  |
| outstanding\_in\_currency |  |  |  |  |  |
| original\_rating |  |  |  |  |  |
| current\_rating |  |  |  |  |  |
| simulated\_rating |  |  |  |  |  |
| existing\_stage |  |  |  |  |  |
| simulated\_stage |  |  |  |  |  |
| remark |  |  |  |  |  |
| counterparty\_code |  |  |  |  |  |
| basel\_asset\_class\_code |  |  |  |  |  |
| default\_ind |  |  |  |  |  |
| past\_due\_days |  |  |  |  |  |
| bu\_dept\_code |  |  |  |  |  |
| company\_code |  |  |  |  |  |
| import\_source |  |  |  |  |  |

## Proposed Design

V\_PRICING\_DEALS.AS\_OF\_DATE = CONTEXTS.PK\_RD (converted to Date format) where

CONTEXTS.WORKSPACE = 'PRICING' and POSITION = 0

V\_PRICING\_DEALS.COUNTERPARTY\_CODE IN (PRICING\_SIMULATION.

ENTITY\_CODE, PRICING\_SIMULATION.SIMULATED\_ENTITY\_CODE)

## Procedure definition

CREATE VIEW V\_PRICING\_DEALS

1. CRM Allocation Compute,LGD

## Business Requirement

## Data Requirement

**Pre-requisites**

* BEA\_STAGE\_ALLOCATION – This is the configuration table that BEA would upload to the system.
* All transactions are loaded to LOANDEPO / FACILITY / REPO / SECURITY\_POSITIONS.
* BEA\_UNSECURED\_LGD is uploaded by BEA user
* BEA\_CRM\_RECOVERY\_RATE is uploaded by BEA user
* All transactions are loaded to LOANDEPO / FACILITY / REPO / SECURITY\_POSITIONS
* All CRM information are loaded to COLLATERAL / GUARANTEE / CONTRACT\_GUARANTEE

Input of this customization:

Input Parameters:

* p\_simulation\_id (PRICING\_SIMULATION.ID)
* p\_as\_of\_date (PRICING\_SIMULATION.AS\_OF\_DATE)
* p\_entity\_code (PRICING\_SIMULATION.EXISTING\_ENTITY\_CODE)
* p\_simulation\_entity\_code (PRICING\_SIMULATION.SIMULATED\_ENTITY\_CODE)
* MOODY'S ANALYTICS $pageProperty.from("group")
* BEA Real-time ECL Technical Design Moody's Analytics Confidential Information - Do Not Redistribute 29
* p\_user\_name (PRICING.USER\_NAME

**Requirement**

For non-retail exposure, it will base on the BEA\_UNSECURED\_LGD, to get the unsecured LGD by the basel asset class, company code, as well as contract type. While for the retail exposure, it is expected that there is another customization to fill up the RETAIL\_POOL before this LGD patching could be applied to the retail exposure. Once these retail pool are in place, the unsecured LGD could be retried by the basel asset class and together with the retail pool

Please note that the LGD provided in this BEA\_UNSECURED\_LGD is the actual LGD percentage, so that it is required to divided by 100 to the final result.

For collateral / guarantee, it will base on the BEA\_CRM\_RECOVERY\_RATE, to get the recovery rate by the location and the code of the crm. For any guarantee which have contract type start with 08-00-xx-xx, those will be mapped into GUARNATEE table, the rest of the collateral type will be mapped into COLLATERAEL table.

Please note that the recovery rate is provided for the crm, so that the LGD should be retrieved by 1 – recovery rate.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PRICING\_T\_CDR | | | | | |
| **Table type** | Data | | | | |
| **Description** |  | | | | |
| **Column** | **Type** | **Description** | **Primary Key** | **Nullable** | **Example** |
| as\_of\_date | varchar2 |  | Y | N | 2020/01/01/ |
| simulation\_id | number |  | N | N | 123456789 |
| collateral\_reference | varchar2 |  | N | N |  |
| id | number |  | N | N |  |
| ead | number |  | N | N | 123456789 |
| allocated\_amount | number |  | N | N | 153507.00 |
| remark | varchar2 |  |  |  |  |
| Note : PRICING\_T\_CDR table will have many fields in addition to above. Refer to the COLLATERAL\_INPUT.JSON file structure provided with Real-Time ECL Documentation. | | | | | |

## Proposed Design

The final result for the exposure side will be updated to:

The LGD on the following table: LOANDEPO/FACILITY/REPO/ SECURITY

The final result for the crm side will be updated to:

The LGD on the following table: COLLATERAL / GUARANTEE

The ATTRIBUTE\_1 to store the macro-economic index on the COLLATERAL table

The NDIM1 to store the Coefficient between the collateral and the marco-economic index on the COLLATERAL table.

The NDIM2 to store the recovery day required to sell off the collateral to recovery on the COLLATERAL table.

This is expected to be executed at Post-Transaction process.

According to contract Type And Asset class, contract LIKE 'NB%'，NOT LIKE 'NB%'

There are 26 asset classes in total

1. If the current and original rating are NOT the same, then convert the INTERNAL one to Moody's equvilant one first. But this does NOT need to update LOANDEPO.
2. Then to calculate the number of notch down, it can be just simply by getting the last digit of CURRENT\_RATING\_ORDER minus the last digit of ORIGINAL\_RATING\_ORDER
3. If there is a rule matched with the BEA\_STAGE\_ALLOCATION (rule order =6 for this case), then update the LOANDEPO.DIM20 to STAGE\_2

Create PRICING\_T\_CDR table to store the CRM Allocation Compute results.

Computing logic can refer to “ PACK\_IFRS9\_CRMALLOC.Launch\_I9PCA\_Process ”

## Procedure definition

CREATE TABLE PRICING\_T\_CDR

PACK\_PRICING. CRM\_ALLOCATION

1. ECL Results

## Business Requirement

Once user request the Simulation to Moody's ECL API, the Front end should get the job\_id to furthermonitor the simulation request.use the GET request to monitor the status and once the status is OK retrieve the ECL results with Cashflows (in same JSON) using separate GET API.

## Data Requirement

**Pre-requisites**

Input of this customization:

* ECL API response JSON is split and stored into two separate tables.
* ECL results stored into PRICING\_ECL\_RESULT table
* Cashflow results stored in the PRICING\_CASHFLOWS\_RESULT table.

**Requirement**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PRICING\_ECL\_RESULT | | | | | |
| **Table type** | Data | | | | |
| **Description** |  | | | | |
| **Column** | **Type** | **Description** | **Primary Key** | **Nullable** | **Example** |
| simulation\_id | number |  | Y | N |  |
| as\_of\_date | datetime |  | N | N |  |
| contract\_reference | varchar2 |  | N | N |  |
| ci\_account\_number | varchar2 |  | N | N |  |
| ead | number |  | N | N |  |
| allocated\_collateral | number |  | N | N |  |
| guarantee\_effect | number |  |  |  |  |
| basel\_asset\_class\_code | varchar2 |  |  |  |  |
| bu\_dept\_code | varchar2 |  |  |  |  |
| company\_code | varchar2 |  |  |  |  |
| existing\_stage | varchar2 |  |  |  |  |
| simulated\_stage | varchar2 |  |  |  |  |
| existing\_ecl | number |  |  |  |  |
| simulated\_ecl | number |  |  |  |  |
| ecl\_delta | number |  |  |  |  |
| scenario\_id | varchar2 |  |  |  |  |
| eir\_spread | number |  |  |  |  |
| one\_year\_ecl | number |  |  |  |  |
| lifetime\_ecl | number |  |  |  |  |
| job\_id | number |  |  |  |  |
| error\_message | number |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PRICING\_CASHFLOWS\_RESULT | | | | | |
| **Table type** | Data | | | | |
| **Description** | 存储调用ECL API后获取的JSON结果中CASHFLOWS部分 | | | | |
| **Column** | **Type** | **Description** | **Primary Key** | **Nullable** | **Example** |
| simulation\_id | number |  | Y | N |  |
| as\_of\_date | datetime |  | N | N |  |
| contract\_reference | varchar2 |  | N | N |  |
| table\_name | varchar2 |  | N | N |  |
| model\_id | varchar2 |  | N | N |  |
| accrual\_factor | number |  | N | N |  |
| amount | number |  |  |  |  |
| amount\_type | varchar2 |  |  |  |  |
| begin\_date | date |  |  |  |  |
| end\_date | date |  |  |  |  |
| cf\_type | varchar2 |  |  |  |  |
| discount\_factor | number |  |  |  |  |
| exchange\_rate | number |  |  |  |  |
| reset\_date | date |  |  |  |  |
| payment\_date | date |  |  |  |  |
| outstanding\_balance | number |  |  |  |  |
| rate | number |  |  |  |  |
| rate\_type | varchar2 |  |  |  |  |
| amount\_quantity | varchar2 |  |  |  |  |
| leg\_type | varchar2 |  |  |  |  |
| income\_type | varchar2 |  |  |  |  |
| is\_real | varchar2 |  |  |  |  |
| cap | number |  |  |  |  |
| floor | number |  |  |  |  |
| market\_spread | number |  |  |  |  |
| scenario\_id | varchar2 |  |  |  |  |
| job\_id | number |  |  |  |  |
| is\_real | number |  |  |  |  |
| cap | datetime |  |  |  |  |
| floor | varchar2 |  |  |  |  |
| market\_spread | varchar2 |  |  |  |  |
| scenario\_id | varchar2 |  |  |  |  |
| job\_id | number | ECL\_Realtime\_Response. jobId |  |  |  |

## Proposed Design

This procedure must be called once the successful result is retrieved from ECL API

Ead,ci\_account\_number, allocated\_collateral,guarantee\_effect ,basel\_asset\_class\_code，bu\_dept\_code,company\_code , existing\_stage, existing\_ecl

Updated by PL/SQL Procedure PACK\_PRICING.PREPARE\_ECL\_RESULT

## Procedure definition

CREATE TABLE PRICING\_ECL\_RESULT

CREATE TABLE PRICING\_CASHFLOWS\_RESULT

PACK\_PRICING. PREPARE\_ECL\_RESULT

1. Users, Roles, Access & Authentication

## Business Requirement

All the users are defined under CD\_USERS table of Moody's Data Mart. In addition to this table the

following custom table for ECL Application will be created with additional information of List of BU Dept

Code User can access and role he is assigned too.

## Data Requirement

**Pre-requisites**

Input of this customization:

**Requirement**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **PRICING\_USERS** | | | | | |
| **Table type** | Admin | | | | |
| **Description** | This table will be a non-partition table that accumulate and stored the ECL user &&BU Dept information | | | | |
| **Column** | **Type** | **Description** | **Primary Key** | **Nullable** | **Example** |
| User\_id | Number | User\_id corresponding to CD\_USERS table | N | N | 3 |
| bu\_dept\_codes | varchar2 | BU Dept code, can be multiple, comma-separated. bu\_dept\_code corresponding to BEA\_BU\_DEPT table | N | N |  |
| default\_bu\_dept\_code | varchar2 | BU Dept code, default BU Dept when new user is created | N | N |  |
| role | varchar2 |  | N | N | 123456789 |
| creation\_date | date |  |  |  |  |
|  |  |  |  |  |  |

## Proposed Design

UI team directly adds / edits / deletes data to the table

## Procedure definition

CREATE TABLE PRICING\_USERS