

**TÀI LIỆU THIẾT KẾ CHI TIẾT**

**BACKEND MODULE: SEAOPS\_JOB\_SCHEDULE**

**Version 2.9**

**Hanoi, May 2024**

**Mục Lục**

[1. BẢNG THEO DÕI HIỆU LỰC VÀ THAY ĐỔI 9](#_Toc169019017)

[2. KSSCT 12](#_Toc169019018)

[2.1. Tình trạng tiến trình hiện hữu trên API KSSCT 12](#_Toc169019019)

[2.2. Tiến trình tạo dữ liệu bảng slip mark 12](#_Toc169019020)

[2.2.1. Mô tả nghiệp vụ 12](#_Toc169019021)

[2.2.2. Biểu đồ tuần tự 13](#_Toc169019022)

[2.2.3. Quy tắc gộp và dữ liệu theo nghiệp vụ 16](#_Toc169019023)

[2.2.4. Bảng TBL\_BI\_TRANSACTION\_ACCOUNT 17](#_Toc169019024)

[2.2.5. Bảng TBL\_SLIP\_MARK 20](#_Toc169019025)

[2.3. Tiến trình đẩy file upload sang API OCR 26](#_Toc169019026)

[2.3.1. Mô tả nghiệp vụ 26](#_Toc169019027)

[2.3.2. Biểu đồ tuần tự 26](#_Toc169019028)

[2.3.3. Bảng TBL\_SLIP\_UPLOAD 31](#_Toc169019029)

[2.4. Lấy trạng thái file đc xử lý trên OCR 33](#_Toc169019030)

[2.4.1. Mô tả nghiệp vụ 33](#_Toc169019031)

[2.4.2. Biểu đồ tuần tự 34](#_Toc169019032)

[2.4.3. Sử dụng lại bảng TBL\_SLIP\_UPLOAD tham khảo mục 2.3.3 34](#_Toc169019033)

[2.5. Dữ liệu bảng bi\_inputter\_date 34](#_Toc169019034)

[2.5.1. Mô tả nghiệp vụ 34](#_Toc169019035)

[2.5.2. Biểu đồ tuần tự 34](#_Toc169019036)

[2.5.3. Quy tắc gộp dữ liệu 36](#_Toc169019037)

[2.5.3.1. Tổng hợp dữ có định khoản 36](#_Toc169019038)

[2.5.3.2. Tổng hợp dữ liệu ko định khoản 37](#_Toc169019039)

[2.6. Job đồng bộ dữ liệu báo cáo năng suất 38](#_Toc169019040)

[2.6.1. Mô tả nghiệp vụ 38](#_Toc169019041)

[2.6.2. Quy tắc lấy dữ liệu 38](#_Toc169019042)

[3. GSSGN 40](#_Toc169019043)

[3.1. Tình trạng tiến trình hiện hữu trên API GSSGN 40](#_Toc169019044)

[3.2. Tiến trình tạo Thông tin tài sản bảo đảm hình thành trong tương lai 43](#_Toc169019045)

[3.2.1. Mô tả nghiệp vụ 43](#_Toc169019046)

[3.2.2. Biểu đồ tuần tự 44](#_Toc169019047)

[3.2.2.1. (1) Lock dữ liệu bảng TBL\_BI\_COLLATERAL: 45](#_Toc169019048)

[3.2.2.2. (2) Tổng hợp dữ liệu tạo bản ghi tbl\_collateral\_future 45](#_Toc169019049)

[3.2.2.3. (3) Insert dữ liệu tổng hợp vào tbl\_collateral\_future 46](#_Toc169019050)

[3.2.2.4. (4) Update trạng thái đồng bộ bảng TBL\_BI\_COLLATERAL 47](#_Toc169019051)

[3.2.2.5. (5) Exception khi chạy job 47](#_Toc169019052)

[3.2.3. Bảng tbl\_collateral\_future 47](#_Toc169019053)

[3.3. Tìm các ghi thay đổi giá trị trong bảng tbl\_bi\_collateral\_tmp sau đó merge sang tbl\_bi\_collateral 52](#_Toc169019054)

[3.3.1. Mô tả nghiệp vụ 52](#_Toc169019055)

[3.3.2. Biểu đồ tuần tự 53](#_Toc169019056)

[3.3.2.1. (1) Lock dữ liệu bảng TBL\_BI\_COLLATERAL\_TMP: 55](#_Toc169019057)

[3.3.2.2. (2) Mapping dữ liệu 55](#_Toc169019058)

[3.3.2.3. (3) Insert / Update dữ liệu merge tbl\_bi\_collateral và tmp 63](#_Toc169019059)

[3.3.2.4. (4) Update lại trạng thái đồng bộ bảng TBL\_BI\_COLLATERAL\_TMP 63](#_Toc169019060)

[3.3.2.5. (5) Exception khi chạy job 63](#_Toc169019061)

[3.3.3. Bảng tbl\_bi\_collateral 64](#_Toc169019062)

[3.4. Tự động dánh giá dựa trên điều kiện quan hệ tín dụng 68](#_Toc169019063)

[3.4.1. Mô tả nghiệp vụ 68](#_Toc169019064)

[3.4.2. Biểu đồ tuần tự 68](#_Toc169019065)

[3.4.3. Quy tắc xử lý dữ liệu 68](#_Toc169019066)

[3.4.3.1. Bước 1 : Lấy dữ liệu kỳ đánh giá thỏa mãn điều kiện đánh giá : 68](#_Toc169019067)

[3.4.3.2. Bước 2 : xử lý dữ liệu ( từng Row) 69](#_Toc169019068)

[3.4.3.2.1. B2.1 Lấy dữ liệu kỳ trước : 69](#_Toc169019069)

[3.4.3.2.2. B2.2 Truy vấn thông tin ngày quá hạn (NO\_DAY\_OVERDUE ) và check vi phạm quy tắc nghị quyết 69](#_Toc169019070)

[3.4.3.2.3. B2.3 Tính toán kết quả tuân thủ 70](#_Toc169019071)

[3.4.3.2.4. B2.4 Check điều kiện kết hợp TBL\_CONDITION\_CREDIT\_RELATION.TYPE 71](#_Toc169019072)

[3.4.4. Mô tả DB bảng TBL\_PERIOD\_EVALUATION 71](#_Toc169019073)

[3.5. Tự động đánh giá dựa trên DKNQ doanh thu 81](#_Toc169019074)

[3.5.1. Mô tả nghiệp vụ 81](#_Toc169019075)

[3.5.2. Biểu đồ tuần tự 81](#_Toc169019076)

[3.5.3. Quy tắc xử lý dữ liệu 81](#_Toc169019077)

[3.5.3.1. Lấy dữ liệu thỏa mãn điều kiện đánh giá 81](#_Toc169019078)

[3.5.3.2. Xử lý dữ liệu ( quy tắc theo từng row) 82](#_Toc169019079)

[3.5.3.2.1. Bước 1 Tính phạm vi thời gian lấy dữ liệu (from\_date & to\_date) 82](#_Toc169019080)

[3.5.3.2.2. Tính doanh số giải ngân, doanh số thu gốc và lãi, doanh số phát hành LC, doanh số phát hành bảo lãnh 82](#_Toc169019081)

[3.5.3.3. Tính toán và đánh giá theo các quy tắc 93](#_Toc169019082)

[3.5.3.3.1. Đánh giá điều kiện TOI 93](#_Toc169019083)

[3.5.3.3.2. Đánh giá điều kiện doanh thu tối thiểu 93](#_Toc169019084)

[3.5.3.3.3. Đánh giá theo các quy tắc khác của DKNQ 93](#_Toc169019085)

[3.5.3.3.4. Đánh giá tổng hợp các điều kiện trên 96](#_Toc169019086)

[3.5.4. Mô tả bảng 97](#_Toc169019087)

[3.5.4.1. Bảng TBL\_PERIOD\_EVALUATION tham khảo mục 3.4.4 97](#_Toc169019088)

[3.5.4.2. Bảng TBL\_REVENUE\_CONFIGURATION 97](#_Toc169019089)

[3.5.4.3. Bảng TBL\_REVENUE\_CONDITION 100](#_Toc169019090)

[3.5.4.4. Bảng TBL\_REVENUE\_PERIOD\_RATE 104](#_Toc169019091)

[3.5.4.5. Bảng TBL\_CONFIG\_INPUT\_TOI 105](#_Toc169019092)

[3.5.4.6. Bảng TBL\_BI\_DISBURSEMENT\_HISTORY 105](#_Toc169019093)

[3.5.4.7. Bảng TBL\_BI\_CONTRACT\_DEBT 106](#_Toc169019094)

[3.5.4.8. Bảng TBL\_BI\_GUARANTEE\_SALES 107](#_Toc169019095)

[3.5.4.9. Bảng TBL\_BI\_LC\_RELEASE\_SALES 108](#_Toc169019096)

[3.5.4.10. Bảng TBL\_BI\_DISBURSEMENT\_SALES 108](#_Toc169019097)

[3.5.4.11. Bảng TBL\_BI\_CREDIT\_SALES 109](#_Toc169019098)

[3.6. Job tự động Đánh giá tự động ĐKSP, ĐKNQ tài chính, BHNT, định kỳ, tài sản bảo đảm, mục đích sử dụng vốn, pháp lý, khác 109](#_Toc169019099)

[3.6.1. Mô tả nghiệp vụ 109](#_Toc169019100)

[3.6.2. Biểu đồ tuần tự 110](#_Toc169019101)

[3.6.3. Quy tắc xử lý dữ liệu 110](#_Toc169019102)

[3.6.3.1. Lấy dữ liệu thỏa mãn điều kiện 110](#_Toc169019103)

[3.6.3.2. Xử lý dữ liệu 110](#_Toc169019104)

[3.7. Job đánh giá tự động ĐKNQ bán chéo 111](#_Toc169019105)

[3.7.1. Mô tả nghiệp vụ 111](#_Toc169019106)

[3.7.2. Biểu đồ tuần tự 111](#_Toc169019107)

[3.7.3. Quy tắc xử lý dữ liệu 111](#_Toc169019108)

[3.7.3.1. Lấy dữ liệu thỏa mãn điều kiện 111](#_Toc169019109)

[3.7.3.2. Xử lý dữ liệu 112](#_Toc169019110)

[3.8. Job sinh kỳ tự động tổng hợp 114](#_Toc169019111)

[3.8.1. Mô tả nghiệp vụ 114](#_Toc169019112)

[3.8.2. Biểu đồ tuần tự 114](#_Toc169019113)

[3.8.3. Quy tắc xử lý dữ liệu 114](#_Toc169019114)

[3.8.3.1. Lấy dữ liệu thỏa mãn điều kiện 114](#_Toc169019115)

[3.8.3.2. Xử lý dữ liệu 115](#_Toc169019116)

[3.8.3.2.1. Sinh kỳ tự động không theo ngày kiểm tra chứng từ 115](#_Toc169019117)

[3.8.3.2.1.1. Trường hợp không chọn lưu ý đơn vị nghiệp vụ, không chọn tự động sinh kỳ 115](#_Toc169019118)

[3.8.3.2.1.2. Có chon tự động sinh kỳ 116](#_Toc169019119)

[3.8.3.2.1.3. Có lưu ý đơn vị kinh doanh 117](#_Toc169019120)

[3.8.3.2.2. Sinh kỳ tự động dự vào ngày kiểm tra chứng từ 118](#_Toc169019121)

[3.8.3.2.2.1. Trương hợp GROUP\_CONTRACT\_CONDITION = PERIODIC 118](#_Toc169019122)

[3.8.3.2.2.2. Trường hợp GROUP\_CONTRACT\_CONDITION = LOAN\_PRODUCT\_CONDITION 119](#_Toc169019123)

[3.8.3.2.2.3. Trường hợp GROUP\_CONTRACT\_CONDITION = CREDIT\_RELATION 120](#_Toc169019124)

[3.9. Ngừng theo dõi Bảo hiểm tài sản bảo đảm 123](#_Toc169019125)

[3.9.1. Mô tả nghiệp vụ 123](#_Toc169019126)

[3.9.2. Biểu đồ tuần tự 123](#_Toc169019127)

[3.9.3. Mô tả nghiệp vụ 123](#_Toc169019128)

[3.10. Sinh kỳ tự động do thay đổi thông tin TSBD 124](#_Toc169019129)

[3.10.1. Mô tả nghiệp vụ 124](#_Toc169019130)

[3.10.2. Biểu đồ tuần tự 124](#_Toc169019131)

[3.10.3. Quy tắc xử lý dữ liệu 124](#_Toc169019132)

[3.10.3.1. Lấy dữ liệu thỏa mãn điều kiện 124](#_Toc169019133)

[3.10.3.2. Xử lý dữ liệu 124](#_Toc169019134)

[3.10.3.2.1. Sinh kỳ TSBD 125](#_Toc169019135)

[3.10.3.2.2. Sinh kỳ bảo hiểm TSBD ( period\_condition\_ins = ENOUGH) 127](#_Toc169019136)

[3.11. Job sinh kỳ đầu tiên DKNQ sản phẩm vay 128](#_Toc169019137)

[3.11.1. Mô tả nghiệp vụ 128](#_Toc169019138)

[3.11.2. Biểu đồ tuần tự 128](#_Toc169019139)

[3.11.3. Quy tắc xử lý dữ liệu 128](#_Toc169019140)

[3.11.3.1. Lấy dữ liệu thỏa mãn điều kiện 128](#_Toc169019141)

[3.11.3.2. Xử lý dữ liệu 128](#_Toc169019142)

[3.12. Job sinh kỳ dựa theo MDSDV 130](#_Toc169019143)

[3.12.1. Mô tả nghiệp vụ 130](#_Toc169019144)

[3.12.2. Biểu đồ tuần tự 130](#_Toc169019145)

[3.12.3. Quy tắc xử lý dữ liệu 130](#_Toc169019146)

[3.12.3.1. Lấy dữ liệu thỏa mãn điều kiện 131](#_Toc169019147)

[3.12.3.2. Xử lý dữ liệu 132](#_Toc169019148)

[3.13. Job sinh kỳ dự theo tần suất kiểm tra định kỳ 134](#_Toc169019149)

[3.13.1. Mô tả nghiệp vụ 134](#_Toc169019150)

[3.13.2. Biểu đồ tuần tự 135](#_Toc169019151)

[3.13.3. Quy tắc xử lý dữ liệu 135](#_Toc169019152)

[3.13.3.1. Lấy dữ liệu thỏa mãn điều kiện 135](#_Toc169019153)

[3.13.3.2. Xử lý dữ liệu 135](#_Toc169019154)

[3.14. Áp chế tài ĐKNQ 145](#_Toc169019155)

[3.14.1. Mô tả nghiệp vụ 145](#_Toc169019156)

[3.14.2. Biểu đồ tuần tự 145](#_Toc169019157)

[3.14.3. Quy tắc xử lý dữ liệu 145](#_Toc169019158)

[3.14.3.1. Lấy dữ liệu thỏa mãn điều kiện 145](#_Toc169019159)

[3.14.3.2. Xử lý dữ liệu 147](#_Toc169019160)

[3.15. Áp chế tài vi phạm bảo hiểm tài sản bảo đảm 152](#_Toc169019161)

[3.15.1. Mô tả nghiệp vụ 152](#_Toc169019162)

[3.15.2. Biểu đồ tuần tự 152](#_Toc169019163)

[3.15.3. Quy tắc xử lý dữ liệu 152](#_Toc169019164)

[3.15.3.1. Lấy dữ liệu thỏa mãn điều kiện 152](#_Toc169019165)

[3.15.3.2. Xử lý dữ liệu 152](#_Toc169019166)

[3.15.3.2.1. Lấy dữ liệu tài sản bảo đảm 153](#_Toc169019167)

[3.15.3.2.2. Lấy dữ liệu những hợp đồng giải ngân tương ứng 153](#_Toc169019168)

[3.15.3.2.3. Lấy dữ liệu cấu hình chế tài TSBD tương ứng 153](#_Toc169019169)

[3.15.3.2.3.1. Tạo dữ liệu áp chế tài 154](#_Toc169019170)

[3.16. Báo cáo KPI 159](#_Toc169019171)

[3.16.1. Mô tả nghiệp vụ 159](#_Toc169019172)

[3.16.2. Biểu đồ tuần tự 159](#_Toc169019173)

[3.16.3. Quy tắc xử lý dữ liệu 159](#_Toc169019174)

[3.16.3.1. Xử lý và cập nhật dữ liệu báo cáo KPI theo từng đơn vị 160](#_Toc169019175)

[3.16.3.1.1. Tính và lưu dữ liệu KPI với trường hợp không có dữ liệu KPI trong năm 160](#_Toc169019176)

[3.16.3.1.2. Tính và lưu dữ liệu KPI với trường hợp có dữ liệu KPI trong năm 162](#_Toc169019177)

# BẢNG THEO DÕI HIỆU LỰC VÀ THAY ĐỔI

A – Hiệu lực của Tài liệu

|  |  |  |
| --- | --- | --- |
| Đóng dấu | Ngày đóng dấu | Người đóng dấu |
|  |  |  |

B – Theo dõi thay đổi của Tài liệu

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Ngày | Phiên bản | Hình thức \* | Mô tả | Tác giả |
| 1 | 17/10/2023 | 1.0 | Create | Viết mới tài liệu | Bùi Đức Toan, Bùi Thanh Tùng |
| 2 | 19/10/2023 | 1.1 | Update | Thêm đặc tả các tiến trình [mục 2.3](#_Tiến_trình_đẩy), [mục 3.2](#_Tiến_trình_tạo_1) | Bùi Đức Toan, Bùi Thanh Tùng |
| 3 | 23/10/2023 | 1.2 | Update | Thêm đặc tả các tiến trình [mục 2.4](#_Lấy_trạng_thái), [mục 3.3](#_Tìm_các_ghi) | Bùi Đức Toan, Bùi Thanh Tùng |
| 4 | 09/11/2023 | 1.3 | Update | Sửa lại quy tắc gộp và xử lý dữ liệu [mục 2.2.3](#_Quy_tắc_gộp) | Bùi Đức Toan |
| 5 | 09/11/2023 | 1.4 | Update | Sửa lại quy tắc gộp và xử lý dữ liệu [mục 3.2](#_Tiến_trình_tạo_1), [mục3.3](#_Tìm_các_ghi) | Bùi Thanh Tùng |
| 6 | 15/12/2023 | 1.5 | Update | Thêm mới job đồng bộ BI\_INPUTTER\_DATE  [Mục 2.5](#_Dữ_liệu_bảng) | Bùi Đức Toan |
| 7 | 15/01/2024 | 1.6 | Update | [Chỉnh sửa mô tả quy tắc merge dữ liệu từ tbl\_bi\_collateral sang tbl\_collateral\_future](#_(2)_Tổng_hợp) | Bùi Đức Toan |
| 8 | 19/02/2024 | 1.7 | Update | [Thêm đặc tả tiến trình Tự động đánh giá dự trên điều kiện QHTT](#_Tự_động_dánh) | Bùi Đức Toan |
| 9 | 20/02/2024 | 1.8 | Update | [Thêm đặc tả tiến trình tổng hợp dữ liệu báo cáo năng suất](#_Job_đồng_bộ) | Bùi Đức Toan |
| 10 | 05/03/2024 | 1.9 | Update | Đánh giá DKNQ Doanh thu, DKNQ khác | Bùi Đức Toan |
| 11 | 20/03/2024 | 2.1 | Update | Đánh giá ĐKNQ bán chéo | Bùi Đức Toan |
| 12 | 29/03/2024 | 2.2 | Update | * [JOB sinh kỳ tự động tổng hợp](#_Job_sinh_kỳ) * [Ngừng theo dõi Bảo hiểm tài sản bảo đảm](#_Ngừng_theo_dõi) | Bùi Đức Toan |
| 13 | 04/04/2024 | 2.3 | Update | * [Sinh kỳ tự dộng do thay đổi dữ liệu TSBĐ](#_Sinh_kỳ_tự) | Bùi Đức Toan |
| 14 | 10/04/2024 | 2.4 | Update | * [Sinh kỳ đầu tiên của hợp đồng vay theo DKNQ sản phẩm vay](#_Job_sinh_kỳ_1) | Bùi Đức Toan |
| 15 | 06/05/2024 | 2.5 | Update | * [Sinh kỳ tự động dựa theo dữ liệu lịch sử giải ngân](#_Job_sinh_kỳ_2) | Bùi Đức Toan |
| 16 | 22/05/2024 | 2.6 | Update | * [Sinh kỳ tự động (Tần suất kiểm tra định kỳ)](#_Job_sinh_kỳ_3) | Bùi Đức Toan |
| 17 | 28/05/2024 | 2.7 | Update | * [Áp chế tài vi phạm điều kiện nghị quyết](#_Áp_chế_tài) | Bùi Đức Toan |
| 18 | 06/06/2024 | 2.8 | Update | * [Áp chế tài vi phạm bảo hiểm TSBD](#_Áp_chế_tài_1) | Bùi Đức Toan |
| 19 | 11/06/2024 | 2.9 | Update | * [Báo cáo KPI](#_Báo_cáo_KPI) | Bùi Đức Toan |
| 20 | 09/08/2024 | 2.10 | Update | * [Job sinh kỳ dựa theo Điều kiện sản phẩm Khác](#_Job_sinh_kỳ_4) * [Job sinh kỳ dựa theo Điều kiện sản phẩm Bán chéo](#_Job_sinh_kỳ_5) * [Job sinh kỳ dựa theo Điều kiện sản phẩm Quan hệ tín dụng](#_Job_sinh_kỳ_6) | Lê Văn Lợi |
| 21 | 14/08/2024 | 2.11 | Update | * [Job sinh kỳ dựa theo khoản giải tỏa tiền tạm ứng](#_Job_sinh_kỳ_7) * [Job sinh kỳ dựa theo Điều kiện Doanh thu](#_Job_sinh_kỳ_8) * [Job sinh kỳ dựa theo Điều kiện sản phẩm Bảo hiểm nhân thọ](#_Job_sinh_kỳ_9) * [Job sinh kỳ dựa theo Điều kiện sản phẩm Pháp lý](#_Job_sinh_kỳ_10) * [Job sinh kỳ dựa theo Điều kiện sản phẩm Tài chính](#_Job_sinh_kỳ_11) * [Job sinh kỳ dự theo tần suất kiểm tra định kỳ](#_Job_sinh_kỳ_3) * [Job sinh kỳ dựa theo MDSDV](#_Job_sinh_kỳ_2) * [Job sinh kỳ tự động tổng hợp](#_Job_sinh_kỳ) * [Tự động đánh giá dựa trên DKNQ doanh thu](#_Tự_động_đánh) |  |

Chú thích: (*\*) A (Add) – thêm mới, M (Modify) – sửa đổi, D (Delete) – xóa*

# KSSCT

## Tình trạng tiến trình hiện hữu trên API KSSCT

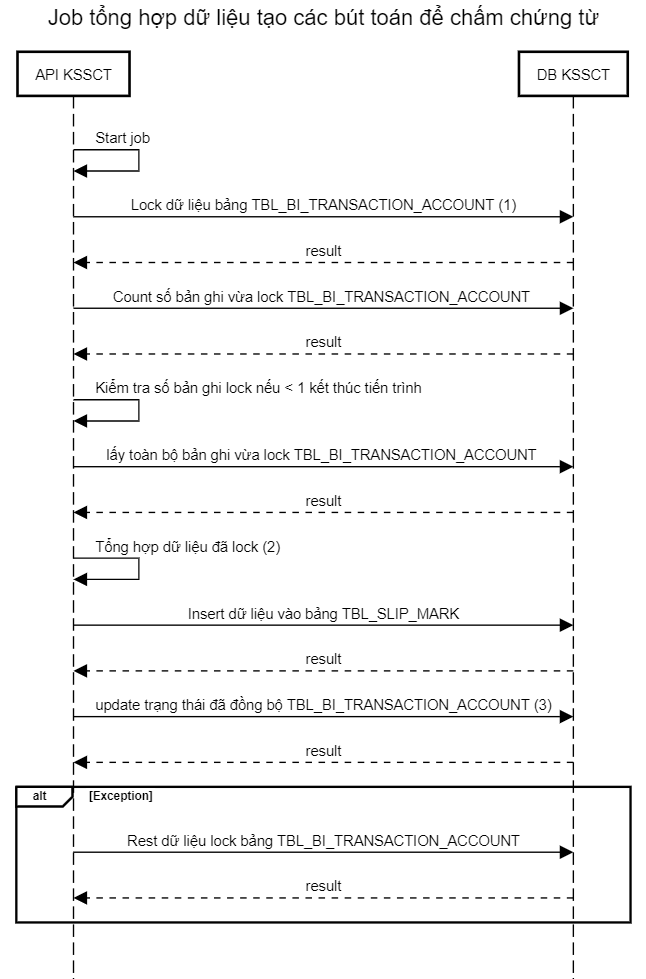
|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Mô tả tiến trình** | **Trạng thái trên KSSCT** | **Đánh giá chuyển đổi** |
| 1 | Tạo tham số cho nghiệp vụ chứng từ có định khoản | Active | Có thể off do BI đã tự đồng bộ |
| 2 | Tạo tham số cho nghiệp vụ chứng từ không định khoản | Active | Có thể off do BI đã tự đồng bộ |
| 3 | Tạo tham số cho nghiệp vụ lấy user BI | Active | Có thể off do BI đã tự đồng bộ |
| 4 | Đồng bộ dữ liệu BI có định khoản | Active | Có thể off do BI đã tự đồng bộ |
| 5 | Đồng bộ dữ liệu BI không định khoản | Active | Có thể off do BI đã tự đồng bộ |
| 6 | Reset dữ liệu đồng bộ chứng từ lỗi | Active |  |
| 7 | [Tạo dữ liệu bảng slip mark](#_Tiến_trình_tạo) | Active |  |
| 8 | Đồng bộ dữ liệu user từ BI | Active | Có thể off do BI đã tự đồng bộ |
| 9 | Tạo dữ liệu cho bảng BiInputterDate | Active |  |
| 10 | [Đẩy file lên hệ thống OCR](#_Tiến_trình_đẩy) | Active |  |
| 11 | Lấy trạng thái file được xử lý trên OCR | Active |  |
| 12 | Lấy dữ liệu chi tiết file upload | Active |  |
| 13 | Lấy dữ liệu chi tiết chứng từ | Active |  |
| 14 | Chấm tự động | Active |  |
| 15 | Đồng bộ lại dữ liệu loại chứng từ | Active |  |
| 16 | Lấy DS phòng ban BI | Active | Có thể off do BI đã tự đồng bộ |
| 17 | Tổng hợp dữ liệu báo cáo năng suất | Active | Đã gửi mô tả chưa thấy thực hiện |

## Tiến trình tạo dữ liệu bảng slip mark

### Mô tả nghiệp vụ

Job tạo dữ liệu bảng SlipMark ( Bảng Group lại các bút toán theo điều kiện để tạo bản ghi mới phục vụ mục đích châm chứng từ)

### Biểu đồ tuần tự



**Hình ảnh 1: Biểu đồ tuần tự mô tả job tạo dữ liệu bảng SlipMark**

\*) Chú thích về job

(1) Lock dữ liệu bảng TBL\_BI\_TRANSACTION\_ACCOUNT :

+ B1 : Sinh ra 1 batchId kiểu String random (16 ký tự)

+ B2 : Update thông tin bảng TBL\_BI\_TRANSACTION\_ACCOUNT :

|  |  |
| --- | --- |
| **Tên trường** | **Giá trị** |
| **Thông tin trường Update** | |
| BATCH\_ID | BatchId sinh random từ B1 |
| PRIORITY | PRIORITY + 1 |
| SYNC | 'P' |
| LOCKED\_DATE | sysdate |
| **Điều kiện update** | |
| DAYID ngày đồng bộ nằm trong vòng 3 ngày so với hiện tại | |
| SYNC = 'PRE' | |
| BATCH\_ID is null | |

(2) Tổng hợp dữ liệu đã lock :

+ B1: Lấy dữ liệu đã lock theo batchId từ bảng TBL\_BI\_TRANSACTION\_ACCOUNT

+ B2: Group đữ liệu theo các trường thông tin sau ( sử dụng java Collectors để group) : TRAN\_REF, AUTHORISER, INPUTER, DATE\_TIME, COMPANY\_USER, COMPANY\_ACCOUNTING, BUT\_TOAN\_LUONG, TRANS\_REF\_DATE\_TIME)

+ B3: Loại bỏ trans\_ref đặc biệt ( hardcode : 000042809673)

+ B4: Khởi tạo dữ liệu insert vào bảng **TBL\_SLIP\_MARK** :

(3) Update trạng thái đồng bộ bảng **TBL\_BI\_TRANSACTION\_ACCOUNT**

|  |  |
| --- | --- |
| **Tên trường** | **Giá trị** |
| **Thông tin trường Update** | |
| SYNC | 'P' |
| **Điều kiện update** | |
| DAYID ngày đồng bộ nằm trong vòng 3 ngày so với hiện tại | |
| BATCH\_ID = batchId sinh ra khi lock bản ghi | |
| TRANS\_REF not in ( Danh sách trans\_ref khi group dữ liệu bị lỗi) | |

(4) **Exception khi chạy job**

Reset những bản ghi đã lock trong bảng **TBL\_BI\_TRANSACTION\_ACCOUNT**

|  |  |
| --- | --- |
| **Tên trường** | **Giá trị** |
| **Thông tin trường Update** | |
| BATCH\_ID | null |
| SYNC | 'PRE' |
| LOCKED\_DATE | null |
| **Điều kiện update** | |
| DAYID ngày đồng bộ nằm trong vòng 3 ngày so với hiện tại | |
| SYNC = 'P' | |
| BATCH\_ID is not null | |
| LOCKED\_DATE quá 5 phút so với thời điểm hiện tại | |

### Quy tắc gộp và dữ liệu theo nghiệp vụ

Group dữ liệu theo TRANS\_REF\_DATE\_TIME và lầy mặc định 1 bản ghi đảm bảo các tiêu chí sau :

|  |  |  |
| --- | --- | --- |
| **STT** | **Trương** | **Điều kiện** |
| 1 | TRAN\_REF | Not Null |
| 2 | AUTHORISER | Not Null |
| 3 | INPUTTER | Not Null |
| 4 | DATE\_TIME | Not Null |
| 5 | CONPANY\_USER | Not Null |
| 6 | BUT\_TOAN\_LUONG | Not Null |
| 7 | COMPANY\_ACCOUNTING | Not Null |
| 10 | TRANS\_REF\_ORIGIN | Nếu trong danh sách các bản ghi đc Group có chứa TRANS\_REF\_ORIGIN đc bắt đầu bằng : "AZ-" , "SS" , "ON" , "LD" ,"PDLD" , "FT" , "CHG" thì lấy bản ghi đó làm giá trị |

Quy tắc xử lý dữ liệu : So sánh trường dữ liệu TRANS\_REF\_DATE\_TIME nếu đã tồn tại thì bỏ qua . Nếu chưa tồn tại thì thực hiện insert bản ghi mới với những trường thông tin có giá trị sau :

|  |  |  |
| --- | --- | --- |
| **STT** | **TBL\_SLIP\_MARK** | **Mapping với dữ liệu bảng TBL\_BI\_TRANSACTION\_ACCOUNT** |
| 1 | TRAN\_REF | TRAN\_REF |
| 2 | AUTHORISER | * Nếu trans\_ref bắt đầu bằng “LD” lấy giá trị cứng = “DEFAULT\_LD\_AUTHORISER” * Các trường hợp còn lại lấy theo dữ liệu bảng TBL\_BI\_TRANSACTION\_ACCOUNT |
| 3 | INPUTTER | INPUTTER |
| 4 | DATE\_TIME | DATE\_TIME |
| 5 | CO\_CODE | CONPANY\_USER |
| 6 | BUT\_TOAN\_LUONG | BUT\_TOAN\_LUONG |
| 7 | COMPANY\_ACCOUNTING | COMPANY\_ACCOUNTING |
| 8 | STATUS | ‘NEW’ |
| 9 | SLIP\_FROM | ‘BI’ |
| 10 | TRANS\_REF\_ORIGIN | TRANS\_REF\_ORIGIN |
| 11 | VALUE\_DATE | Parse Date dựa trên thông tin trường DATE\_TIME theo format : yyyy-MM-dd HH:mm:ss" |
| 12 | DEPARTMENT\_CODE | * Nếu TBL\_BI\_TRANSACTION\_ACCOUNT .CONPANY\_USER = “VN0010001” lấy thông tin trường TBL\_BI\_TRANSACTION\_ACCOUNT . DEPARTMENT\_CODE * Các trường hợp còn lại để NULL |
| 15 | CREATE\_INPUTTER\_STATUS | * Trường hợp tạo mới giá trị = “NEW” * Trường hợp UPDATE giữ nguyên giá trị cũ |
| 16 | CREATE\_INPUTTER\_PRIORITY | * Trường hợp tạo mới giá trị = 0 * Trường hợp UPDATE giữ nguyên giá trị cũ |
| 17 | CREATED\_DATE | * Trường hợp tạo mới giá trị = SYSDATE * Trường hợp UPDATE giữ nguyên giá trị cũ |
| 18 | CREATED\_USER | * Trường hợp tạo mới giá trị = “SYSTEM” * Trường hợp UPDATE giữ nguyên giá trị cũ |

### Bảng TBL\_BI\_TRANSACTION\_ACCOUNT

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Bảng | | **TBL\_BI\_TRANSACTION\_ACCOUNT** | | | | |
| Mô tả | | Bảng lưu dữ liệu có định khoản gốc đồng bộ từ SB | | | | |
| Index | | index BI\_TRANS\_REF\_INDEX  on TBL\_BI\_TRANSACTION\_ACCOUNT (TRANS\_REF)  index BI\_TRANS\_REF\_DATE\_TIME  on TBL\_BI\_TRANSACTION\_ACCOUNT (TRANS\_REF\_DATE\_TIME)  index BI\_BATCH\_ID\_INDEX  on TBL\_BI\_TRANSACTION\_ACCOUNT (BATCH\_ID) | | | | |
| Chi tiết bảng dữ liệu | | | | | | |
| **STT** | **Tên trường** | | **Kiểu dữ liệu** | **Bắt buộc** | **Mô tả** | **SeaOps cập nhật** |
| 1 | ID | | VARCHAR2(36) | PK | ID bản ghi, là chuỗi UUID |  |
| 2 | TRANS\_REF\_ORIGIN | | VARCHAR2(150 char) | Y | Số bút toán gốc |  |
| 3 | TRANS\_REF\_DATE\_TIME | | VARCHAR2(150 char) | N | Số bút toán và thời gian |  |
| 4 | TRANS\_REF | | VARCHAR2(50 char) | N | Số bút toán |  |
| 5 | ACCT\_DEBIT | | VARCHAR2(150 char) | N | Số tài khoản ghi nợ |  |
| 6 | ACCT\_CREDIT | | VARCHAR2(150 char) | N | Số tài khoản ghi có |  |
| 7 | AMOUNT | | VARCHAR2(255 char) | N | Số tiền |  |
| 8 | CCY | | VARCHAR2(10 char) | N | Đơn vị tiền tệ |  |
| 9 | AMOUNT\_LCY | | VARCHAR2(255 char) | N | Số tiền quy đổi VND |  |
| 10 | NARRATIVE\_ALL | | VARCHAR2(4000 char) | N | Nội dung giao dịch |  |
| 11 | COMPANY\_USER | | VARCHAR2(150 char) | N | Đơn vị quản lý user |  |
| 12 | INPUTTER | | VARCHAR2(150 char) | N | User hạch toán |  |
| 13 | AUTHORISER | | VARCHAR2(150 char) | N | User duyệt |  |
| 14 | DATE\_TIME | | VARCHAR2(150 char) | N | Thời gian duyệt |  |
| 15 | CUSTOMER\_ID | | VARCHAR2(150 char) | N | Mã khách hàng |  |
| 16 | SHORT\_NAME | | VARCHAR2(255 char) | N | Tên khách hàng |  |
| 17 | LEGACY\_ID | | VARCHAR2(150 char) | N | Số sổ |  |
| 18 | COMPANY\_ACCOUNTING | | VARCHAR2(150 char) | N | Đơn vị hạch toán |  |
| 19 | BUT\_TOAN\_LUONG | | VARCHAR2(150 char) | N | Bút toán lương |  |
| 20 | CREDIT\_ACCT\_NAME | | VARCHAR2(255 char) | N | Tên tài khoản có |  |
| 21 | DEBIT\_ACCT\_NAME | | VARCHAR2(255 char) | N | Tên tài khoản nợ |  |
| 22 | DEPARTMENT\_CODE | | VARCHAR2(150 char) | N | Department Code |  |
| 23 | SYNC | | VARCHAR2(150 char) | N | Trạng thái đồng bộ | x |
| 24 | BATCH\_ID | | VARCHAR2(150 char) | N | ID lô xử lý dữ liệu bằng job tự động, là chuỗi UUID | x |
| 25 | UNIT\_DATE\_ID | | VARCHAR2(150 char) | N | Unit date id |  |
| 26 | LOCKED\_DATE | | TIMESTAMP(6) | N | Thời gian job tự động khóa bản ghi | x |
| 27 | PRIORITY | | NUMBER | N | Thứ tự ưu tiên xử lý tự động | x |
| 28 | CREATED\_USER | | VARCHAR2(150 char) | N | User tạo |  |
| 29 | CREATED\_DATE | | TIMESTAMP(6) | N | Ngày tạo |  |
| 30 | UPDATED\_USER | | VARCHAR2(150 char) | N | User cập nhật |  |
| 31 | UPDATED\_DATE | | TIMESTAMP(6) | N | Ngày cập nhật |  |

### Bảng TBL\_SLIP\_MARK

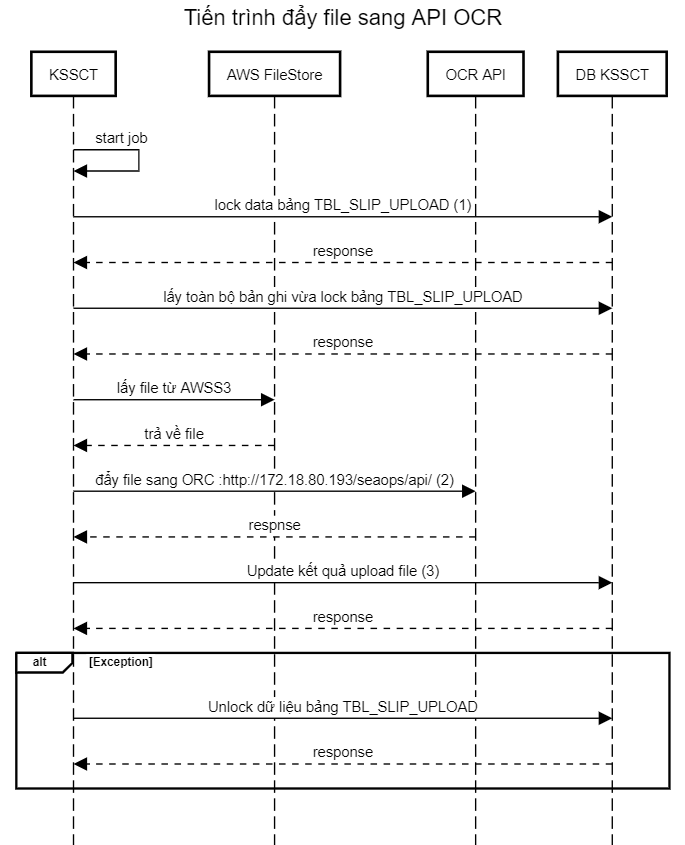
| Bảng | | **TBL\_SLIP\_MARK** | | | |
| --- | --- | --- | --- | --- | --- |
| Mô tả | | Bảng lưu dữ liệu rút gọn của bảng [TBL\_BI\_TRANSACTION\_ACCOUNT](#_Bảng_TBL_BI_TRANSACTION_ACCOUNT) (Dữ liệu giao dịch có định khoản gốc đồng bộ từ SB). | | | |
| Unique constrain | | constraint IDX\_SM\_TRANS\_REF\_DATE\_TIME  unique | | | |
| Index | | index IDX\_SM\_CREATE\_INPUTTER\_KEY  on TBL\_SLIP\_MARK (CREATE\_INPUTTER\_KEY) | | | |
| Chi tiết bảng dữ liệu | | | | | |
| **STT** | **Tên trường** | | **Kiểu dữ liệu** | **Bắt buộc** | **Mô tả** |
| 1 | ID | | NUMBER(9) | PK | ID bản ghi, là số tự tăng |
| 2 | AUTHORISER | | VARCHAR(150) | N | User duyệt |
| 3 | BATCH\_ID | | VARCHAR2(150 char) | N | ID lô xử lý dữ liệu bằng job tự động, là chuỗi UUID |
| 4 | BUSINESS | | VARCHAR2(50 char) | N | Mã nghiệp vụ |
| 5 | BUT\_TOAN\_LUONG | | VARCHAR2(150 char) | N | Số bút toán lương |
| 6 | CO\_CODE | | VARCHAR2(150 char) | N | Mã chi nhánh quản lý User |
| 7 | COMPANY\_ACCOUNTING | | VARCHAR2(150 char) | N | Mã chi nhánh hạch toán |
| 8 | CREATE\_INPUTTER\_KEY | | VARCHAR2(150 char) | N | Key lưu danh sách user hạch toán trên hệ thống, là chuỗi UUID |
| 9 | CREATE\_INPUTTER\_PRIORITY | | NUMBER | N | Thứ tự ưu tiên xử lý tự động lưu danh sách user hạch toán trên hệ thống |
| 10 | CREATE\_INPUTTER\_STATUS | | VARCHAR2(50 char) | N | Trạng thái xử lý tự động lưu danh sách user hạch toán trên hệ thống |
| 11 | CREATED\_DATE | | TIMESTAMP(6) | N | Thời gian tạo |
| 12 | CREATED\_USER | | VARCHAR2(150 char) | N | User tạo |
| 13 | DATE\_TIME | | VARCHAR2(150 char) | N | Thời gian duyệt |
| 14 | DEPARTMENT\_CODE | | VARCHAR2(150 char) | N | Mã phòng ban |
| 15 | FEE | | VARCHAR2(150 char) | N | Phí |
| 16 | INPUTTER | | VARCHAR2(150 char) | N | Mã user hạch toán |
| 17 | LOCKED\_DATE | | VARCHAR2(150 char) | N | Thời gian job tự động khóa bản ghi |
| 18 | MARK\_DATE | | TIMESTAMP(6) | N | Ngày chấm chứng từ |
| 19 | MARK\_NOTE | | VARCHAR2(1000 char) | N | Ghi chú chấm chứng từ |
| 20 | MARK\_USER | | VARCHAR2(150 char) | N | Mã user chấm chứng từ |
| 21 | NOTE | | VARCHAR2(1000 char) | N | Ghi chú |
| 22 | PRIORITY | | NUMBER | N | Thứ tự ưu tiên xử lý tự động |
| 23 | SLIP\_FROM | | VARCHAR2(150 char) | N | Nguồn dữ liệu BI |
| 24 | SLIP\_IMAGE | | VARCHAR2(4000 char) | N | Đường dẫn ảnh chứng từ |
| 25 | STATUS | | VARCHAR2(50 char) | N | Trạng thái chứng từ |
| 26 | THEMATIC | | VARCHAR2(150 char) | N | Chuyên đề |
| 27 | TOTAL\_ERROR | | NUMBER(10) | N | Tổng số lỗi của bút toán |
| 28 | TRANS\_REF | | VARCHAR2(50 char) | N | Số bút toán |
| 29 | TRANS\_REF\_DATE\_TIME | | VARCHAR2(150 char) | N | Số bút toán và thời gian |
| 30 | TRANS\_REF\_ORIGIN | | VARCHAR2(150 char) | Y | Số bút toán gốc |
| 31 | UPDATED\_USER | | VARCHAR2(150 char) | N | User cập nhật |
| 32 | UPDATED\_DATE | | TIMESTAMP(6) | N | Ngày cập nhật |
| 33 | VALUE\_DATE | | TIMESTAMP(6) | N | Ngày thực hiện giao dịch |

## Tiến trình đẩy file upload sang API OCR

### Mô tả nghiệp vụ

Tiến trình đẩy những file người dùng đã scan và upload lên hệ thống sang API OCR

### Biểu đồ tuần tự



\*) Chú thích về job

(1) lock data bảng TBL\_SLIP\_UPLOAD :

+ B1 : Sinh ra 1 batchId kiểu String random (16 ký tự)

+ B2 : Update thông tin bảng TBL\_SLIP\_UPLOAD:

|  |  |
| --- | --- |
| **Tên trường** | **Giá trị** |
| **Thông tin trường Update** | |
| BATCH\_ID | BatchId sinh random từ B1 |
| PRIORITY | PRIORITY + 1 |
| STATUS\_OCR | ' PUSH\_FILE\_TO\_OCR\_PROCESSING ' |
| LOCKED\_DATE | sysdate |
| **Điều kiện update** | |
| STATUS\_OCR = 'NEW' | |
| BATCH\_ID is null | |
| Lấy 500 bản ghi sắp xếp theo PRIORITY | |

(2) Đẩy file sang API ORC:

Thực hiện đẩy tuần tự từng bản ghi List File lấy ra từ bảng TBL\_SLIP\_UPLOAD

Thông tin API upload file OCR :

|  |  |
| --- | --- |
| **Request** | |
| URL | http://172.18.80.193/seaops/api/ |
| Method | POST |
| Headers | Content-Type: multipart/form-data |
| Body | |  |  | | --- | --- | | id | Ramdom String 16 Byte | | file | File lấy về từ AWSS3 | |
| **Response Success** | |
| {  "request\_id": "",  "time": 1697697437,  "code": "SUCCESS",  " output ": {  "id": "",  "file": "1697697437",  "file\_name": "SUCCESS",  "status": "SUCCESS",  "created": "SUCCESS",  "updated\_at": "SUCCESS",  "num\_page": 5,  "num\_processed\_page": 5,  "percent": 80,  " file\_format ": {  "id": "",  "name": "",  "short\_name": "",  "type": ""  },  "ocr\_type ": {  "id": "",  "name": "",  "short\_name": "",  "type": ""  }  }  } | |

(3) Update kết quả upload file vào bảng TBL\_SLIP\_UPLOAD:

Trường hợp đẩy file thành công :

|  |  |
| --- | --- |
| **Tên trường** | **Giá trị** |
| **Thông tin trường Update** | |
| BATCH\_ID | NULL |
| STATUS\_OCR | 'PUSH\_FILE\_TO\_OCR\_SUCCESS' |
| LOCKED\_DATE | NULL |
| **Điều kiện update** | |
| ID = id bản ghi lấy ra để đẩy file | |

Trường hợp đẩy file Thất bại :

|  |  |
| --- | --- |
| **Tên trường** | **Giá trị** |
| **Thông tin trường Update** | |
| BATCH\_ID | NULL |
| STATUS\_OCR | 'NEW' |
| LOCKED\_DATE | NULL |
| **Điều kiện update** | |
| ID = id bản ghi lấy ra để đẩy file | |

(4) Exception xảy ra :

Reset các bản ghi đang bị lock trong batch này:

|  |  |
| --- | --- |
| **Tên trường** | **Giá trị** |
| **Thông tin trường Update** | |
| BATCH\_ID | NULL |
| STATUS\_OCR | 'NEW' |
| LOCKED\_DATE | NULL |
| **Điều kiện update** | |
| BATCH\_ID = batchId sinh ra để lock trong Thread này | |

### Bảng TBL\_SLIP\_UPLOAD

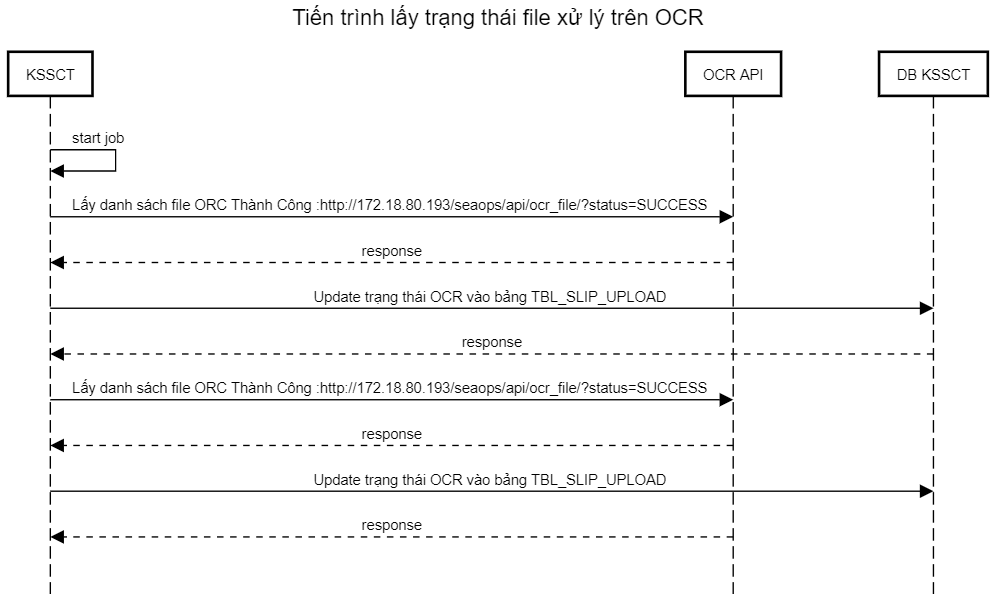
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Bảng | | **TBL\_BI\_SLIP\_UPLOAD** | | | | | |
| Mô tả | | Bảng lưu dữ liệu file upload từ người dùng | | | | | |
| Index | |  | | | | | |
| Chi tiết bảng dữ liệu | | | | | | | |
| **STT** | **Tên cột** | | **Kiểu dữ liệu** | **Bắt buộc** | **Khóa** | **Mô tả** |
| 1 | ID | | NUMBER(19,0) | X | PK | Id bản ghi |
| 2 | PRIORITY | | NUMBER(10,0) | X |  | Giá trị mặc định: 0 |
| 3 | LOCKED\_DATE | | TIMESTAMP (6) |  |  | Ngày khóa bản ghi của job tự động |
| 4 | BATCH\_ID | | VARCHAR2(255 CHAR) |  |  | Id lô xử lý dữ liệu của job tự động |
| 5 | CREATED\_DATE | | TIMESTAMP (6) | X |  | Ngày upload |
| 6 | CREATED\_USER | | VARCHAR2(255 CHAR) | X |  | User upload |
| 7 | MINIO\_BUCKET  \_NAME | | VARCHAR2(255 CHAR) | X |  | MinIO bucket name của Seaops |
| 8 | MINIO\_FILE  \_NAME | | VARCHAR2(255 CHAR) | X |  | MinIO file name của Seaops |
| 9 | NOTE | | VARCHAR2(255 CHAR) |  |  | Ghi chú |
| 10 | NUM\_PAGE | | NUMBER(10,0) |  |  | Số trang |
| 11 | NUM\_PROCESSED  \_PAGE | | NUMBER(10,0) |  |  |  |
| 12 | OCR\_FILE\_ID | | VARCHAR2(255 CHAR) |  |  | Id file OCR |
| 13 | OCR\_RESPONSE | | VARCHAR2(255 CHAR) |  |  | Kết quả OCR file |
| 14 | OCR\_TYPE | | VARCHAR2(255 CHAR) |  |  | Loại file OCR |
| 15 | PERCENT | | FLOAT(24) |  |  | Tỉ lệ OCR thành công |
| 16 | STATUS\_OCR | | VARCHAR2(255 CHAR) |  |  | Trạng thái OCR  ‘NEW’: Đang chờ đẩy file  ‘PUSH\_FILE\_TO\_OCR  \_PROCESSING’: Đang đẩy file  ‘PUSH\_FILE\_TO\_OCR  \_SUCCESS’: Đẩy file thành công  ‘PUSH\_FILE\_TO\_OCR  \_ERROR’: Đẩy file thất bại  OCR\_SUCCESS: OCR thành công  OCR\_ERROR: OCR thất bại  GET\_FILE\_DETAIL  \_PROCESSING: Đang lấy chi tiết file  GET\_FILE\_DETAIL\_SUCCESS: Lấy chi tiết file thành công  GET\_FILE\_DETAIL\_ERROR: Lấy chi tiết file thất bại |
| 17 | UPDATED\_DATE | | TIMESTAMP (6) |  |  | Ngày cập nhật |
| 18 | UPDATED\_USER | | VARCHAR2(255 CHAR) |  |  | Người cập nhật |
| 19 | UPLOAD\_FILE\_ID | | VARCHAR2(255 CHAR) | X |  | Mã số tập CT |
| 20 | CO\_CODE | | VARCHAR2(255 CHAR) |  |  | Mã chi nhánh của User upload |

## Lấy trạng thái file đc xử lý trên OCR

### Mô tả nghiệp vụ

Tiến trình phục vụ mục đích lấy trạng thái OCR của file về, phục vụ mục đích xác định thông tin file đã OCR thành công chưa để lấy thông tin OCR chi tiết

### Biểu đồ tuần tự



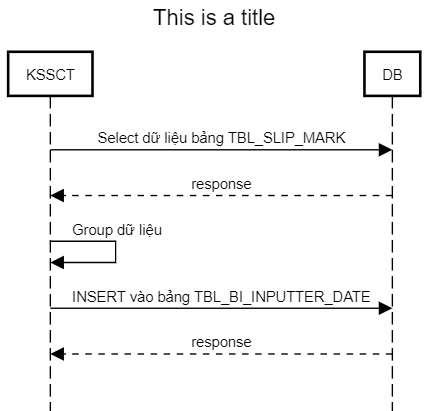
### Sử dụng lại bảng TBL\_SLIP\_UPLOAD tham khảo mục 2.3.3

## Dữ liệu bảng bi\_inputter\_date

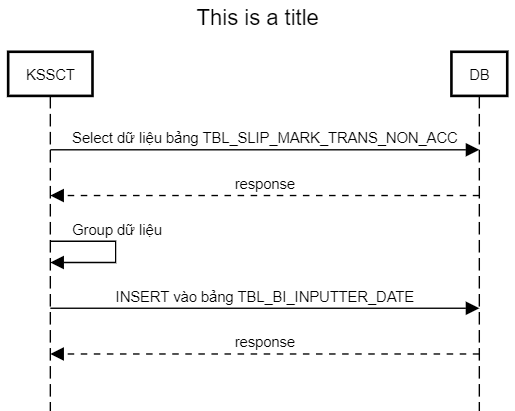
### Mô tả nghiệp vụ

Job Group lại các thông tin liên quan đến chi nhánh/ user hạch toán/ ngày hạch toán để phục vụ mục đích show lên các trường trong màn điều kiện tìm kiếm bút toán.

### Biểu đồ tuần tự



Biểu đồ tuần tự job tổng hợp Param Inputter Date ( Có định khoản)



Biểu đồ tuần tự job tổng hợp Param Inputter Date ( Không định khoản)

### Quy tắc gộp dữ liệu

#### Tổng hợp dữ có định khoản

SELECT dữ liệu từ bảng **TBL\_SLIP\_MARK** bảng group by theo INPUTTER, DEPARTMENT\_CODE, COMPANY\_ACCOUNTING, CO\_CODE, DATE\_TIME ( Các trường này đóng vai trò là KEY dữ liệu ).

|  |  |  |
| --- | --- | --- |
| **STT** | **TBL\_BI\_INPUTTER\_DATE** | **Mapping với dữ liệu bảng TBL\_SLIP\_MARK** |
| 1 | ID | Ramdom UUID |
| 3 | INPUTTER | INPUTTER |
| 4 | V\_DATE | DATE\_TIME lấy YYYY-MM-DD |
| 5 | COMPANY\_USER | CO\_CODE |
| 6 | DEPARTMENT\_CODE | DEPARTMENT\_CODE |
| 7 | COMPANY\_ACCOUNTING | COMPANY\_ACCOUNTING |
| 8 | TYPE | ‘TRANS\_ACC’ |
| 9 | CREATED\_USER | ‘SYSTEM’ |
| 10 | CREATED\_DATE | SYSDATE() |
| 11 | COMBINE\_KEY | String JOIN : INPUTTER + V\_DATE + COMPANY\_USER + COMPANY\_ACCOUNTING + DEPARTMENT\_CODE + TYPE |

Dữ liệu liệu nếu trùng bộ KEY (INPUTTER, DEPARTMENT\_CODE, COMPANY\_ACCOUNTING, COMPANY\_USER, V\_DATE, TYPE) ko cần update chỉ INSERT những dữ liệu mới

#### Tổng hợp dữ liệu ko định khoản

SELECT dữ liệu từ bảng **TBL\_SLIP\_MARK\_TRANS\_NON\_ACC** bảng group by theo INPUTTER, VALUE\_DATE ( lấy theo Date ko lấy Time) , CO\_CODE, BUSINESS, DEPARTMENT\_CODE, COMPANY\_USER ( Các trường này đóng vai trò là KEY dữ liệu ).

|  |  |  |
| --- | --- | --- |
| **STT** | **TBL\_BI\_INPUTTER\_DATE** | **Mapping với dữ liệu bảng TBL\_SLIP\_MARK\_TRANS\_NON\_ACC** |
| 1 | ID | Ramdom UUID |
| 2 | BUSINESS\_TYPE | BUSINESS |
| 3 | INPUTTER | INPUTTER |
| 4 | V\_DATE | VALUE\_DATE convert sang String YYYY-MM-DD |
| 5 | COMPANY\_USER | COMPANY\_USER |
| 6 | DEPARTMENT\_CODE | DEPARTMENT\_CODE |
| 7 | COMPANY\_ACCOUNTING | CO\_CODE |
| 8 | TYPE | ‘TRANS\_NON\_ACC’ |
| 9 | CREATED\_USER | ‘SYSTEM’ |
| 10 | CREATED\_DATE | SYSDATE() |
| 11 | COMBINE\_KEY | String JOIN : INPUTTER + V\_DATE + COMPANY\_USER + COMPANY\_ACCOUNTING + DEPARTMENT\_CODE + BUSINESS\_TYPE + TYPE |

Dữ liệu liệu nếu trùng bộ KEY (INPUTTER, DEPARTMENT\_CODE, COMPANY\_ACCOUNTING, COMPANY\_USER, V\_DATE, BUSINESS\_TYPE, TYPE) ko cần update chỉ INSERT những dữ liệu mới

## Job đồng bộ dữ liệu báo cáo năng suất

### Mô tả nghiệp vụ

Job tính tổng số các bút toán đã chấm theo từng ngày, từng user chấm, từng đơn vị vào 1 bảng để truy xuất báo cáo năng xuất của user chấm chứng từ

### Quy tắc lấy dữ liệu

Dữ liệu được tổng hợp theo ngày T-1 như câu truy vấn dưới đây :

WITH slipMarkData AS (

SELECT

sm.MARK\_USER mark\_user,

sm.COMPANY\_ACCOUNTING co\_code,

AD\_UNIT.UNIT\_NAME unit\_name,

TO\_CHAR( sm.MARK\_DATE, 'DD-MM-YYYY' ) mark\_date,

1 number\_slip

FROM

TBL\_SLIP\_MARK sm

JOIN AD\_UNIT ON sm.COMPANY\_ACCOUNTING = AD\_UNIT.UNIT\_CODE

WHERE

TRUNC( sm.MARK\_DATE ) = TRUNC( SYSDATE - INTERVAL '1' DAY )

AND sm.MARK\_USER IS NOT NULL

AND sm.SLIP\_FROM = 'BI'

),

slipMarkNonAccData AS (

SELECT

sm.MARK\_USER mark\_user,

sm.CO\_CODE co\_code,

AD\_UNIT.UNIT\_NAME unit\_name,

TO\_CHAR( sm.MARK\_DATE, 'DD-MM-YYYY' ) mark\_date,

1 number\_slip

FROM

STG\_SEAOPS.TBL\_SLIP\_MARK\_TRANS\_NON\_ACC sm

JOIN AD\_UNIT ON sm.CO\_CODE = AD\_UNIT.UNIT\_CODE

WHERE

TRUNC( sm.MARK\_DATE ) = TRUNC( SYSDATE - INTERVAL '1' DAY )

AND sm.MARK\_USER IS NOT NULL

),

slipMarkNonTransRefData AS (

SELECT

sm.MARK\_USER mark\_user,

sm.CO\_CODE co\_code,

AD\_UNIT.UNIT\_NAME unit\_name,

TO\_CHAR( sm.CREATED\_DATE, 'DD-MM-YYYY' ) mark\_date,

sm.TOTAL number\_slip

FROM

TBL\_SLIP\_MARK\_NON\_TRANS\_REF sm

JOIN AD\_UNIT ON sm.CO\_CODE = AD\_UNIT.UNIT\_CODE

WHERE

TRUNC( sm.CREATED\_DATE ) = TRUNC( SYSDATE - INTERVAL '1' DAY )

AND sm.MARK\_USER IS NOT NULL

),

union\_datas AS ( SELECT \* FROM slipMarkData UNION ALL SELECT \* FROM slipMarkNonAccData UNION ALL SELECT \* FROM slipMarkNonTransRefData ),

group\_datas AS (

SELECT

mark\_user,

co\_code AS unit\_code,

unit\_name,

mark\_date,

sum( number\_slip ) total\_trans\_ref

FROM

union\_datas ud

GROUP BY

mark\_user,

co\_code,

mark\_date,

unit\_name

ORDER BY

mark\_user

) SELECT

( gds.mark\_user || unit\_code || mark\_date ) AS COMPOSE\_KEY,

gds.\*

FROM

group\_datas gds

Dữ liệu lấy ra từ câu truy vấn sẽ được insert vào bảng **TBL\_REPORT\_PRODUCTIVE :**

Mapping dữ liệu :

|  |  |  |
| --- | --- | --- |
| **Trường trong bảng**  **TBL\_REPORT\_PRODUCTIVE** | **Data truy vấn** | **Ý nghĩa** |
| ID | StringRandom | ID của bản ghi |
| BATCH\_ID | null |  |
| CREATED\_DATE | sysdate | Ngày tạo |
| CREATED\_USER | ‘SYSTEM’ |  |
| MARK\_DATE | MARK\_DATE | Ngày chấm CT |
| MARK\_USER | MARK\_USER | User chấm CT |
| TOTAL\_TRANS\_REF | TOTAL\_TRANS\_REF | Tổng số bút toán |
| UNIT\_CODE | UNIT\_CODE | Mã Chi nhánh hạch toán |
| UNIT\_NAME | UNIT\_NAME | Tên Chi nhánh hạch toán |
| YEAR\_MONTH | MARK\_DATE (format : MM-YYYY) | Tháng chấm |
| COMPOSE\_KEY | COMPOSE\_KEY |  |

Trường hợp trùng COMPOSE\_KEY update lại bản ghi.

# GSSGN

## Tình trạng tiến trình hiện hữu trên API GSSGN

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Mô tả tiến trình** | **Loại** | **Trạng thái trên GSSGN** | **Đánh giá chuyển đổi** |
| 1 | Tạo kỳ đánh giá  launchJobCreatePeriodEvaluation | Job nghiệp vụ | Active |  |
| 2 | Tạo kỳ đánh giá Mục đích sử dụng vốn từ dữ liệu lần giải ngân đồng bộ từ BI  launchJobCreatePeriodFromLDEvaluation | Job nghiệp vụ | Active |  |
| 3 | [Tạo thông tin tài sản bảo đảm hình thành trong tương lai](#_Tiến_trình_tạo)  launchJobCreateCollateralFuture | Job nghiệp vụ | Active |  |
| 4 | [Tìm các ghi thay đổi giá trị trong bảng tbl\_bi\_collateral\_tmp sau đó merge sang](#_Tìm_các_ghi)  [tbl\_bi\_collateral](#_Tìm_các_ghi)  launchJobMergingCollateralData | Job nghiệp vụ | Active |  |
| 5 | Tạo kỳ đánh giá của hợp đồng theo danh mục tần suất kiểm tra định kỳ  launchJobCreatePeriodFromTSKTDK | Job nghiệp vụ | Active |  |
| 6 | Tạo kỳ đánh giá của hợp đồng theo danh mục điều kiện sản phẩm vay  launchJobCreatePeriodFromDKSP | Job nghiệp vụ | Active |  |
| 7 | Tạo kỳ đánh giá bảo hiểm tài sản đảm bảo  launchJobCreatePeriodFromBHTSBD | Job nghiệp vụ | Active |  |
| 8 | Đánh giá tự động điều kiện doanh thu  launchJobAutoEvaluationRevenue | Job nghiệp vụ | Active |  |
| 9 | Đánh giá tự động điều kiện bán chéo  launchJobAutoEvaluationCrossSelling | Job nghiệp vụ | Active |  |
| 10 | Đánh giá tự động điều kiện quan hệ tín dụng  launchJobAutoEvaluationCreditRelation | Job nghiệp vụ | Active |  |
| 11 | Đánh giá tự động ĐKSP, ĐKNQ tài chính, BHNT, định kỳ, tài sản bảo đảm, mục đích sử dụng vốn, pháp lý, khác  launchJobAutoEvaluationOther | Job nghiệp vụ | Active |  |
| 12 | Đánh giá hợp đồng hết hạn  launchJobOverdueContract | Job nghiệp vụ | Active |  |
| 13 | Cập nhật hợp đồng TSKTDK  launchJobUpdateContractTSKTDK | Job nghiệp vụ | Active |  |
| 14 | Tổng hợp Hợp đồng quá hạn  launchJobGetContractNoOverdue | Job nghiệp vụ | Active |  |
| 15 | Tạo Param Kpi  launchJobCreateKpiParam | Job nghiệp vụ | Active |  |
| 16 | Áp chế tài vi phạm điều kiện nghị quyết  launchJobCreateApplySanctionConditionReport | Job nghiệp vụ | Active |  |
| 17 | Áp chế tài vi phạm bảo hiểm tài sản bảo đảm  launchJobCreateApplySanctionInsuranceReport | Job nghiệp vụ | Active |  |
| 18 | Tổng hợp dữ liệu báo cáo KPI  Từ bảng đánh giá tbl\_period\_evaluation  launchJobCreateKpiReport | Job nghiệp vụ | Active |  |
| 19 | Job đồng bộ kiểm tra value\_date\_min ở contract\_ref chưa có sẽ lấy ở disbursement\_history  launchJobSyncValueDateMin | Job nghiệp vụ | Active |  |
| 20 | Mdsdv ngừng theo dõi  mdsdvStopFollow | Job nghiệp vụ | Active |  |
| 21 | Bhtsbd ngừng theo dõi  bhtsbdStopFollow | Job nghiệp vụ | Active |  |
| 22 | Đồng bộ bảng dữ liệu  tbl\_bi\_contract\_ref\_define  launchJobGetContractRefDefineData | Job đồng bộ | Active |  |
| 23 | Đồng bộ bảng dữ liệu  TBL\_BI\_CREDIT\_SALES  launchJobGetCreditSalesData | Job đồng bộ | Active |  |
| 24 | Đồng bộ bảng dữ liệu  TBL\_BI\_DISBURSEMENT\_HISTORY  launchJobGetDisbursementHistory | Job đồng bộ | Active |  |
| 25 | Đồng bộ bảng dữ liệu  tbl\_follow\_additions\_doc  launchJobGetPendingRLOS | Job đồng bộ | Active |  |
| 26 | Đồng bộ bảng dữ liệu  TBL\_BI\_LOAN\_PRODUCT  launchJobGetLoanProduct | Job đồng bộ | Active |  |
| 27 | Đồng bộ bảng dữ liệu  TBL\_BI\_COLLATERAL\_TYPE  launchJobGetCollateralType | Job đồng bộ | Active |  |
| 28 | Đồng bộ bảng dữ liệu  TBL\_BI\_COLLATERAL\_TMP  launchJobGetCollateral | Job đồng bộ | Active |  |
| 29 | Đồng bộ bảng dữ liệu  TBL\_BI\_CONTRACT\_DEBT  launchJobGetContractDebtData | Job đồng bộ | Active |  |
| 30 | Đồng bộ bảng dữ liệu  TBL\_BI\_LC\_RELEASE\_SALES  launchJobGetLCReleaseSalesData | Job đồng bộ | Active |  |
| 31 | Đồng bộ bảng dữ liệu  TBL\_BI\_DISBURSEMENT\_SALES  launchJobGetDisbursementSalesData | Job đồng bộ | Active |  |
| 32 | Đồng bộ bảng dữ liệu  TBL\_BI\_GUARANTEE\_SALES  launchJobGetGuaranteeSalesData | Job đồng bộ | Active |  |

## Tiến trình tạo Thông tin tài sản bảo đảm hình thành trong tương lai

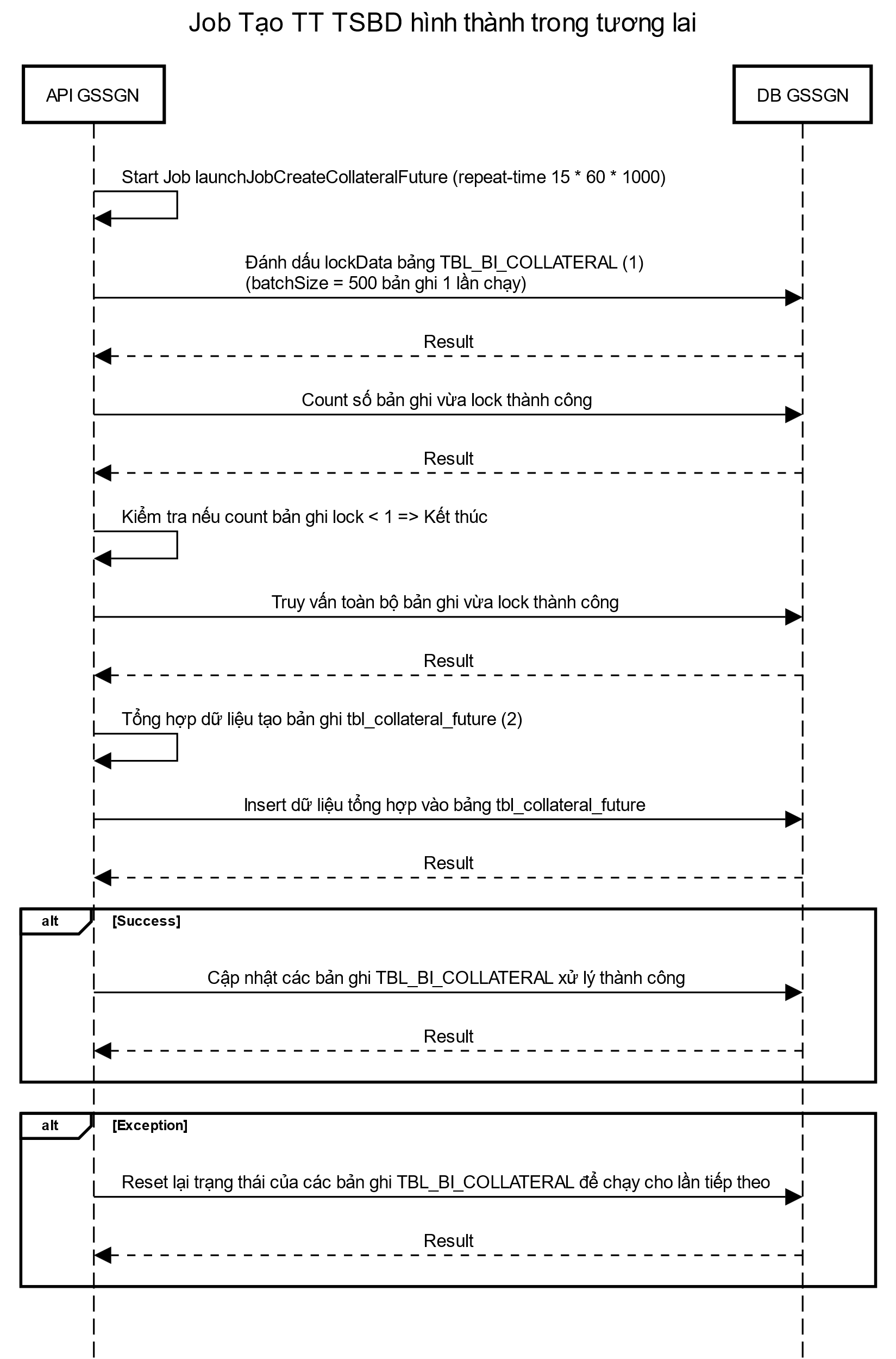
* launchJobCreateCollateralFuture - Job nghiệp vụ

### Mô tả nghiệp vụ

Job tạo dữ liệu bảng tbl\_collateral\_future (Bảng dữ liệu cập nhật thông tin tài sản đảm bảo hình thành trong tương lai) từ data đồng bộ BI bảng tbl\_bi\_ collateral.

tbl\_bi\_ collateral -> tbl\_collateral\_future

### Biểu đồ tuần tự



**Hình ảnh 1: Biểu đồ tuần tự**

(\*) Chú thích về job

#### (1) Lock dữ liệu bảng TBL\_BI\_COLLATERAL:

+ B1 : Sinh ra 1 batchId kiểu String random (16 ký tự)

+ B2 : Update thông tin bảng TBL\_BI\_COLLATERAL:

|  |  |
| --- | --- |
| **Tên trường** | **Giá trị** |
| **Thông tin trường Update** | |
| CREATE\_FUTURE\_BATCH\_ID | BatchId sinh random từ B1 |
| **Điều kiện update** | |
| CREATE\_FUTURE\_BATCH\_ID is null | |
| COLLATERAL\_FUTURE\_KEY is not null | |

#### (2) Tổng hợp dữ liệu tạo bản ghi tbl\_collateral\_future

*(\*) Note:**tbl\_bi\_collateral.collateral\_future\_key = tbl\_collateral\_future.id*

Từ danh sách các bản ghi *tbl\_bi\_collateral* đã lock thành công tạo từng bản ghi *tbl\_collateral\_future*

Bước 1 : Copy thông tin từ *tbl\_bi\_collateral* sang *tbl\_collateral\_future* theo quy tắc sau :

Theo quy tắc copy **toàn bộ các trường giống nhau** ngoại trừ các trường sau:

+ *tbl\_collateral\_future.id:* ***tbl\_bi\_collateral*.collateral\_future\_key**

+ record\_status: **OPEN**

+ created\_date: **NOW()**

+ updated\_date: **NOW()**

Bước 2 : Check thông tin dữ liệu đã tồn tại trước ở bảng *tbl\_collateral\_future* chưa

(\*) Gọi itemExit là bản ghi *tbl\_collateral\_future* đã tồn tại trước đó với điều kiện

*tbl\_collateral\_future.id = tbl\_bi\_collateral.collateral\_future\_key*

Check điều kiện itemExit :

Nếu itemExit tồn tại thì lấy copy dữ liệu cũ từ itemExit với những trường sau :

. date\_delivery\_qsd,

. los\_code,

. date\_on\_report,

. col\_status,

. status,

. risk\_level,

. sanction,

. receive\_file,

. missing\_file, note,

. record\_status,

. reason,

. created\_user,

. created\_date,

. updated\_user,

. seab\_reval\_no

Nếu itemExit không tồn tại đẻ nguyên các giá trị cũ

Bước 3 : Update lại 1 số thông tin cần tính toán sau khi đã compares và merge dữ liệu

+ priority:

. record\_status = APPROVED => priority = 2

. record\_status != APPROVED => priority = 1

+ maturity\_date

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| |-> Nếu DateDeliveryQSD != null && DateDeliveryBDS != null | | | | |  |  |  |
| |---> Nếu -> DateDeliveryBDS > DateDeliveryQSD: => MaturityDate = DateDeliveryBDS | | | | | | |  |
| |---> Nếu -> DateDeliveryBDS < DateDeliveryQSD: => MaturityDate = DateDeliveryQSD | | | | | | |  |
| | |  |  |  |  |  |  |  |
| |-> Nếu DateDeliveryQSD != null && DateDeliveryBDS == null: => MaturityDate = DateDeliveryQSD | | | | | | | |
| |-> Nếu DateDeliveryQSD == null && DateDeliveryBDS != null: => MaturityDate = DateDeliveryBDS | | | | | | | |
| ------------------------------------------------------------------------------------------------ | | | | | |  |  |
| |-> Nếu (DateDeliveryQSD == null && DateDeliveryBDS == null) | | | | |  |  |  |
| |---> Nếu (NotaryDate != null && RegDate != null) | | | |  |  |  |  |
| |------> Nếu -> RegDate > NotaryDate: => MaturityDate = RegDate | | | | | |  |  |
| |------> Nếu -> RegDate < NotaryDate: => MaturityDate = NotaryDate | | | | | |  |  |
| | |  |  |  |  |  |  |  |
| |---> Nếu NotaryDate != null && RegDate == null: => MaturityDate = NotaryDate | | | | | | |  |
| |---> Nếu NotaryDate == null && RegDate != null: => MaturityDate = RegDate | | | | | |  |  |

#### (3) Insert dữ liệu tổng hợp vào tbl\_collateral\_future

Cấu trúc bảng [tbl\_collateral\_future](#_Bảng_tbl_collateral_future)

#### (4) Update trạng thái đồng bộ bảng TBL\_BI\_COLLATERAL

|  |  |
| --- | --- |
| **Tên trường** | **Giá trị** |
| **Thông tin trường Update** | |
| COLLATERAL\_FUTURE\_KEY | null |
| CREATE\_FUTURE\_BATCH\_ID | null |
| **Điều kiện update** | |
| CREATE\_FUTURE\_BATCH\_ID = batchId sinh ra khi lock bản ghi | |

#### (5) Exception khi chạy job

Reset những bản ghi đã lock trong bảng TBL\_BI\_COLLATERAL

|  |  |
| --- | --- |
| **Tên trường** | **Giá trị** |
| **Thông tin trường Update** | |
| CREATE\_FUTURE\_BATCH\_ID | null |
| **Điều kiện update** | |
| CREATE\_FUTURE\_BATCH\_ID = batchId sinh ra khi lock bản ghi | |

### Bảng tbl\_collateral\_future

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Bắt buộc** | **Khóa** | **Mô tả** |
| 1 | ID | VARCHAR2(36) | x | pk | Khóa bản ghi |
| 2 | COL\_STATUS | VARCHAR2(255 CHAR) |  |  | Trạng thái TSĐB   * *ON //Tài sản đang trong trạng thái hình thành trong tương lai* * *OFF // Tài sản đã hình thành TSĐB* * *OUT\_OFF\_GSSGN // Tài sản không nằm trong phạm vi kiểm soát của GSSGN* |
| 3 | COLL\_STATUS\_CODE | VARCHAR2(255 CHAR) |  |  | Trạng thái thế chấp: gồm các giá trị: 1/2/3/4/5 |
| 4 | COLL\_TYPE | VARCHAR2(255 CHAR) |  |  | Loại tài sản đảm bảo (link đến bảng loại tài sản đảm bảo) TBL\_BI\_COLLATERAL\_TYPE.COLLA\_TYPE\_CODE |
| 5 | COLLATERAL\_CODE | VARCHAR2(255 CHAR) |  |  | Mã tài sản đảm bảo |
| 6 | CONTRACT\_MD | VARCHAR2(255 CHAR) |  |  | Mã MD ngoại bảng giá trị tài sản,  link bảng TBL\_BI\_GUARANTEE\_SALES.CONTRACT\_MD |
| 7 | CONTRACT\_REF | VARCHAR2(1500 CHAR) |  |  | Danh sách Mã REF, link bảng STG\_SEAOPS.TBL\_BI\_CONTRACT\_REF\_DEFINE.REF\_ID |
| 8 | CREATED\_DATE | TIMESTAMP(6) |  |  | Thời gian tạo bản ghi  Định dạng *DD-MMM-YY hh.mm.ss.fffffffff tt* |
| 9 | CREATED\_USER | VARCHAR2(255 CHAR) |  |  | User tạo bản ghi |
| 10 | CURRENCY | VARCHAR2(255 CHAR) |  |  | Loại tiền tệ |
| 11 | CUSTOMER\_ID | VARCHAR2(255 CHAR) |  |  | Mã khách hàng |
| 12 | DATE\_ON\_REPORT | VARCHAR2(255 CHAR) |  |  | Ngày trên biên bản |
| 13 | DESCRIPTION | VARCHAR2(4000 CHAR) |  |  | Mô tả |
| 14 | EXECUTION\_VALUE | VARCHAR2(255 CHAR) |  |  | Giá trị tài sản |
| 15 | INSUR\_ISSUE\_DATE | VARCHAR2(255 CHAR) |  |  | Ngày hiệu lực bảo hiểm. Định dạng: YYYY-MM-DD HH:mm:ss |
| 16 | INSUR\_PROVIDER | VARCHAR2(255 CHAR) |  |  | Đơn vị bảo hiểm |
| 17 | INSURANCE\_AMOUNT | VARCHAR2(255 CHAR) |  |  | Giá trị bảo hiểm tài sản |
| 18 | INSURANCE\_REQ | VARCHAR2(255 CHAR) |  |  | TS có bảo hiểm (Yes/No) |
| 19 | INSURANCE\_TYPE | VARCHAR2(255 CHAR) |  |  | Loại bảo hiểm |
| 20 | IS\_EXPIRY\_DATE | VARCHAR2(255 CHAR) |  |  | Ngày hết hạn, Định dạng: YYYY-MM-DD HH:mm:ss |
| 21 | LOS\_CODE | VARCHAR2(50 CHAR) |  |  | Mã LOS |
| 22 | MISSING\_FILE | VARCHAR2(4000 CHAR) |  |  | Hồ sơ thiếu |
| 23 | NOMINAL\_VALUE | VARCHAR2(255 CHAR) |  |  | Số tiền theo MD |
| 24 | NOTARY\_DATE | VARCHAR2(255 CHAR) |  |  | Ngày bàn giao GCN theo CT, Định dạng: YYYY-MM-DD HH:mm:ss |
| 25 | NOTE | VARCHAR2(4000 CHAR) |  |  | Ghi chú |
| 26 | RE\_EXEC\_VALUE | VARCHAR2(255 CHAR) |  |  | Giá trị vay tối đa trên TS |
| 27 | REASON | VARCHAR2(4000 CHAR) |  |  | Lý do từ chối |
| 28 | RECEIVE\_FILE | VARCHAR2(4000 CHAR) |  |  | Hồ sơ nhận |
| 29 | RECORD\_STATUS | VARCHAR2(255 CHAR) |  |  | Trạng thái   * *OPEN "Chờ đánh giá"* * *PENDING "Chờ duyệt"* * *APPROVED "Đã duyệt"* * *REJECTED "Từ chối"* * *SAVED "Đã lưu"* * *RETURN "Trả về"* |
| 30 | REG\_DATE | VARCHAR2(255 CHAR) |  |  | Ngày bàn giao BĐS theo CT, Định dạng: YYYY-MM-DD HH:mm:ss |
| 31 | REVAL\_VALUE | VARCHAR2(255 CHAR) |  |  | Giá trị định giá lại |
| 32 | RISK\_LEVEL | VARCHAR2(255 CHAR) |  |  | Mức độ rủi ro   * *LOW "Thấp"* * *MEDIUM "Trung bình"* * *HIGH "Cao"* |
| 33 | SANCTION | VARCHAR2(4000 CHAR) |  |  | Chế tài áp dụng |
| 34 | SEAB\_COL\_SOURCE | VARCHAR2(255 CHAR) |  |  | Trạng thái hình thành tài sản đảm bảo  *1 - đã hình thành*  *2 - chưa hình thành* |
| 35 | SEAB\_REVAL\_COMP | VARCHAR2(255 CHAR) |  |  | Đơn vị định giá lại |
| 36 | STATUS | VARCHAR2(255 CHAR) |  |  | Trạng thái   * WARNING "Nhắc nhở" * VIOLATE "Vi phạm" * RISK "Rủi ro" |
| 37 | UPDATED\_DATE | TIMESTAMP(6) |  |  | Ngày cập nhật |
| 38 | UPDATED\_USER | VARCHAR2(255 CHAR) |  |  | User cập nhật |
| 39 | VALUE\_DATE | VARCHAR2(255 CHAR) |  |  | Ngày hiệu lực  YYYY-mm-dd HH:MM:SS |
| 40 | VER\_DATE | VARCHAR2(255 CHAR) |  |  | Ngày định giá lại  YYYY-mm-dd HH:MM:SS |
| 41 | DATE\_DELIVERY\_BDS | VARCHAR2(255 CHAR) |  |  | Ngày bàn giao BĐS theo chứng từ  DD-MM-YYYY HH:MM:SS |
| 42 | DATE\_DELIVERY\_QSD | VARCHAR2(255 CHAR) |  |  | Ngày bàn giao GCN QSD theo chứng từ  DD-MM-YYYY HH:MM:SS |
| 43 | ROLE\_UPDATE | VARCHAR2(255 CHAR) |  |  | Vai trò user cập nhật tài liệu |
| 44 | CUSTOMER\_NAME | VARCHAR2(255 CHAR) |  |  | Tên khách hàng |
| 45 | MATURITY\_DATE | VARCHAR2(255 CHAR) |  |  | Ngày đến hạn  YYYY-MM-DD HH:MM:SS |
| 46 | COLLATERAL\_ID | VARCHAR2(255 CHAR) |  |  | ID TSĐB, link đến bảng TSĐB  TBL\_BI\_COLLATERAL.COLLATERAL\_ID |
| 47 | CO\_CODE | VARCHAR2(255 CHAR) |  |  | Mã chi nhánh |
| 48 | SEAB\_REVAL\_NO | VARCHAR2(255 CHAR) |  |  | Số chứng thư định giá |
| 49 | PRIORITY | NUMBER |  |  | Thứ tự ưu tiên |

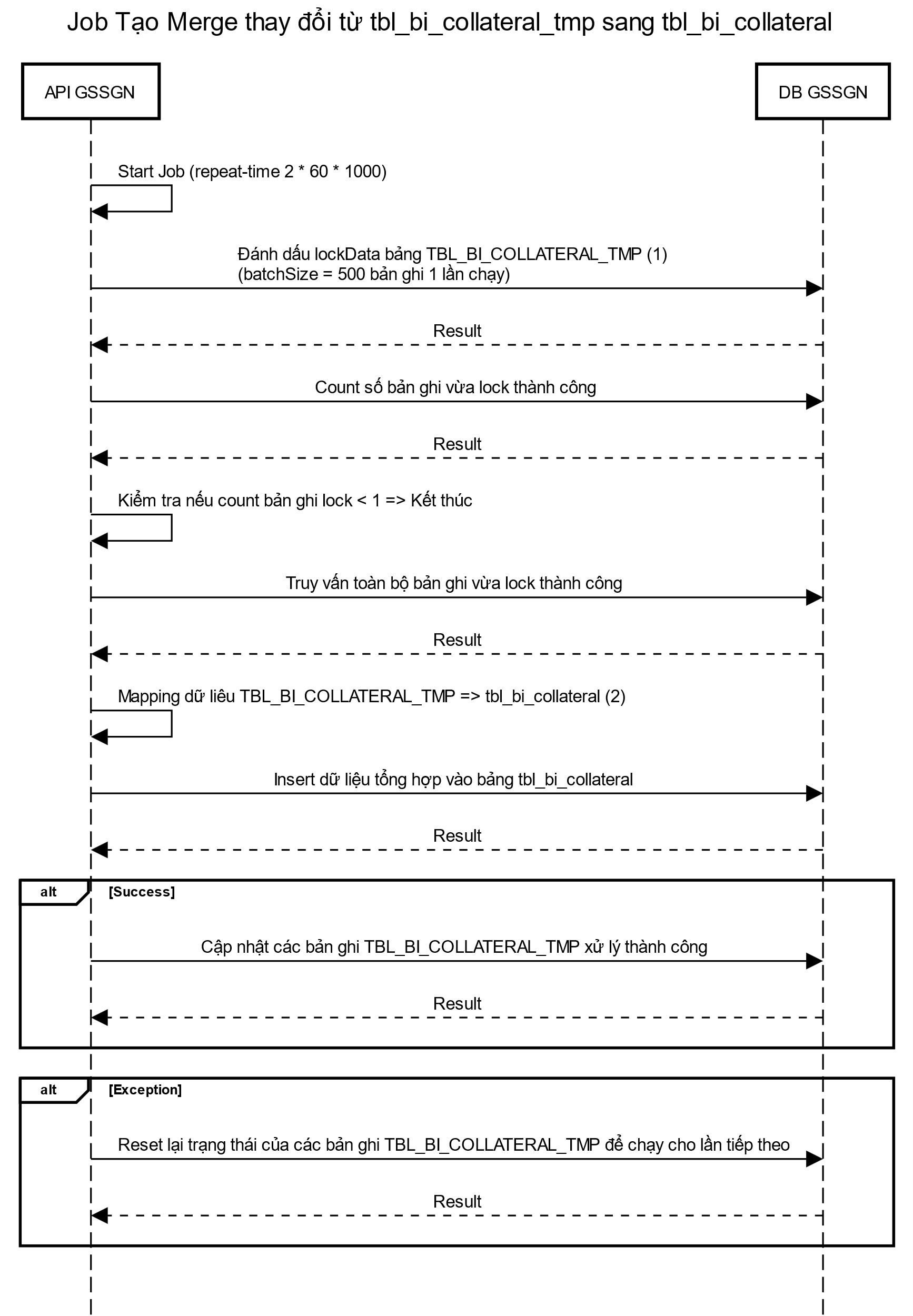
## Tìm các ghi thay đổi giá trị trong bảng tbl\_bi\_collateral\_tmp sau đó merge sang tbl\_bi\_collateral

launchJobMergingCollateralData – Job Nghiệp vụ

### Mô tả nghiệp vụ

Tìm các ghi thay đổi giá trị trong bảng tbl\_bi\_collateral\_tmp sau đó merge sang tbl\_bi\_collateral

### Biểu đồ tuần tự



(\*) Chú thích về job

#### (1) Lock dữ liệu bảng TBL\_BI\_COLLATERAL\_TMP:

+ B1 : Sinh ra 1 batchId kiểu String random (16 ký tự)

+ B2 : Update thông tin bảng TBL\_BI\_COLLATERAL\_TMP:

|  |  |
| --- | --- |
| **Tên trường** | **Giá trị** |
| **Thông tin trường Update** | |
| LOCKED\_KEY | BatchId sinh random từ B1 |
| MAPPING\_STATUS | PROCESSING |
| LOCKED\_DATE | NOW() |
| PRIORITY | PRIORITY + 1 |
| **Điều kiện update** | |
| DATA\_STATUS = ‘ACTIVE’ | |
| LOCKED\_KEY is null | |
| MAPPING\_STATUS in ('NEW', 'RENEW') | |

#### (2) Mapping dữ liệu

Từ LOCKED\_KEY truy vấn danh sách các bản ghi tbl\_bi\_collateral\_tmp đã lock thành công

Với mỗi bản ghi tbl\_bi\_collateral\_tmp hiện tại lấy ra được xử lý theo nghiệp vụ dưới đây:

**B1:** Kiểm tra trạng thái merge của Bản ghi tmp NGÀY TRƯỚC ĐÓ

**B1.1**

- Kiểm tra bản ghi tmp NGÀY TRƯỚC ĐÓ dựa vào Bản ghi tmp hiện tại theo điều kiện:

(COLLATERAL\_ID || DATE\_SYNC\_DATA) = (collateral\_id || (date\_sync\_data – 1))

- Nếu bản ghi tmp NGÀY TRƯỚC ĐÓ tồn tại VÀ mapping\_status <> ‘SUCCESS’

Thực hiện cập nhật lại các trạng thái của bản ghi tmp hiện tại đang xử lý:

|  |  |
| --- | --- |
| **Tên trường** | **Giá trị** |
| **Thông tin trường Update** | |
| mapping\_status | *RENEW* |
| locked\_date | NULL |
| locked\_key | NULL |
| updated\_date | NOW() |
| updated\_user | SYSTEM |

- Kết thúc tiến trình mapping

**B2:** Set điều kiện sinh kỳ cho TSDB cho bản ghi tmp hiện tại

Thực hiện cập nhật giá trị 2 trường period\_condition, period\_condition\_ins theo quy tắc chi tiết dưới đây

**B2.1:**

**-** Kiểm tra thỏa mãn một trong các case sau

Case1: contract\_md IS NULL OR contract\_md = ‘’

Case2: coll\_status\_code NOT IN ("1", "2")

Case3: execution\_value IS NULL OR execution\_value <= 0

Case4: collateral\_code IN ("500", "450")

Case5: value\_date IS NULL OR value\_date = ‘’

Case6: nominal\_value IS NULL OR nominal\_value <= 0

- Nếu thỏa mãn một trong các điệu kiện trên, thực hiện cập nhật tmp và Kết thúc B2.

|  |  |
| --- | --- |
| **Tên trường** | **Giá trị** |
| **Thông tin trường Update** | |
| period\_condition | NOT\_ENOUGH |
| period\_condition\_ins | NOT\_ENOUGH |

- Nếu KHÔNG thỏa mãn một trong các điều kiện thực hiện chuyển sang B2.2

**B2.2:** Cập nhật giá trị trường period\_condition

|  |  |
| --- | --- |
| **Tên trường** | **Giá trị** |
| **Thông tin trường Update** | |
| period\_condition | ENOUGH |

**B2.3:**

- Nếu insurance\_req = ‘YES’ thực hiện cập nhật period\_condition\_ins và kết thúc B2

|  |  |
| --- | --- |
| **Tên trường** | **Giá trị** |
| **Thông tin trường Update** | |
| period\_condition\_ins | ENOUGH |

- Nếu insurance\_req <> ‘YES’ chuyển sang B2.4

**B2.4:**

- Nếu coll\_type = ‘421’ thực hiện tiếp tục kiểm tra seab\_col\_source theo 2 case dưới

+ seab\_col\_source = ‘1’

|  |  |
| --- | --- |
| **Tên trường** | **Giá trị** |
| **Thông tin trường Update** | |
| period\_condition\_ins | ENOUGH |

+ seab\_col\_source <> ‘1’

|  |  |
| --- | --- |
| **Tên trường** | **Giá trị** |
| **Thông tin trường Update** | |
| period\_condition\_ins | NOT\_ENOUGH |

- Nếu coll\_type <> ‘421’ chuyển sang B2.5

**B2.5:** Cập nhật period\_condition\_ins theo 2 case dưới và chuyển sang B3

**-** Nếu coll\_type IN (401", "402", "403", "404", "405", "406", "407", "531",

"541", "551", "552", "553", "554", "555", "556", "557", "582")

|  |  |
| --- | --- |
| **Tên trường** | **Giá trị** |
| **Thông tin trường Update** | |
| period\_condition\_ins | ENOUGH |

**-** Nếu coll\_type NOT IN (401", "402", "403", "404", "405", "406", "407", "531",

"541", "551", "552", "553", "554", "555", "556", "557", "582")

|  |  |
| --- | --- |
| **Tên trường** | **Giá trị** |
| **Thông tin trường Update** | |
| period\_condition\_ins | NOT\_ENOUGH |

**B3:** Nếu có thông tin của tài sản bảo đảm và tài sản bảo đảm đủ điều kiện sinh kỳ nhưng chưa được sinh kỳ thì không được ghi đè thực hiện chuyển trạng thái bản ghi thành RENEW để merge lại sau

**B3.1:** Lấy bản ghi tbl\_bi\_collateral theo điều kiện

tbl\_bi\_collateral.collateral\_id = tmp.collateral\_id

**B3.2:**

- Nếu B3.1 tồn tại bản ghi và thỏa mãn tất cả các điều kiện

DK1: period\_condition = ‘ENOUGH’

DK2: period\_status <> ‘SUCCESS’

Thực hiện cập nhật lại các trạng thái của bản ghi tmp hiện tại đang xử lý và Kết thúc quá trình mapping:

|  |  |
| --- | --- |
| **Tên trường** | **Giá trị** |
| **Thông tin trường Update** | |
| mapping\_status | *RENEW* |
| locked\_date | NULL |
| locked\_key | NULL |
| updated\_date | NOW() |
| updated\_user | SYSTEM |

- Nếu B3.1 không tồn tại bản ghi chuyển sang B4

**B4:** Tạo mới bản ghi hoặc merge update dữ liệu tbl\_bi\_collateral

**B4.1:** Nếu B3.1 không tồn tại bản ghi tbl\_bi\_collateral thực hiện sinh mới tbl\_bi\_collateral từ tmp theo quy tắc => Kết thúc B4.

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên trường** | **Giá trị** |
| 1 | ID | UUID |
| 2 | COLL\_TYPE | tmp.COLL\_TYPE |
| 3 | DESCRIPTION | tmp.DESCRIPTION |
| 4 | CURRENCY | tmp.CURRENCY |
| 5 | COUNTRY | tmp.COUNTRY |
| 6 | NOMINAL\_VALUE | tmp.NOMINAL\_VALUE |
| 7 | EXECUTION\_VALUE | tmp.EXECUTION\_VALUE |
| 8 | VALUE\_DATE | tmp.VALUE\_DATE |
| 9 | COLLATERAL\_CODE | tmp.COLLATERAL\_CODE |
| 10 | INSUR\_PROVIDER | tmp.INSUR\_PROVIDER |
| 11 | INSURANCE\_AMOUNT | tmp.INSURANCE\_AMOUNT |
| 12 | INSURANCE\_TYPE | tmp.INSURANCE\_TYPE |
| 13 | INSUR\_ISSUE\_DATE | tmp.INSUR\_ISSUE\_DATE |
| 14 | IS\_EXPIRY\_DATE | tmp.IS\_EXPIRY\_DATE |
| 15 | SEAB\_COL\_SOURCE | tmp.SEAB\_COL\_SOURCE |
| 16 | COLL\_STATUS\_CODE | tmp.COLL\_STATUS\_CODE |
| 17 | CONTRACT\_MD | tmp.CONTRACT\_MD |
| 18 | INSURANCE\_REQ | tmp.INSURANCE\_REQ |
| 19 | CONTRACT\_REF | tmp.CONTRACT\_REF |
| 20 | VER\_DATE | tmp.VER\_DATE |
| 21 | SEAB\_REVAL\_COMP | tmp.SEAB\_REVAL\_COMP |
| 22 | MARKET\_VALUE | tmp.MARKET\_VALUE |
| 23 | REVAL\_VALUE | tmp.REVAL\_VALUE |
| 24 | RE\_EXEC\_VALUE | tmp.RE\_EXEC\_VALUE |
| 25 | REG\_DATE | tmp.REG\_DATE |
| 26 | NOTARY\_DATE | tmp.NOTARY\_DATE |
| 27 | COLLATERAL\_ID | tmp.COLLATERAL\_ID |
| 28 | CUSTOMER\_ID | tmp.CUSTOMER\_ID |
| 29 | CUSTOMER\_NAME | tmp.CUSTOMER\_NAME |
| 30 | CO\_CODE | tmp.CO\_CODE |
| 31 | SEAB\_REVAL\_NO | tmp.SEAB\_REVAL\_NO |
| 32 | DATE\_SYNC\_DATA | tmp.DATE\_SYNC\_DATA |
| 33 | DATA\_STATUS | ‘ACTIVE’ |
| 34 | LOCKED\_DATE | tmp.LOCKED\_DATE |
| 35 | LOCKED\_KEY | tmp.LOCKED\_KEY |
| 36 | PRIORITY | 0 |
| 37 | PERIOD\_CONDITION | tmp.PERIOD\_CONDITION |
| 38 | PERIOD\_CONDITION\_INS | tmp.PERIOD\_CONDITION\_INS |
| 39 | HASH\_CODE | MD5(tmp.value\_date + ',' + tmp.insur\_issue\_date + ',' + tmp.is\_expiry\_date + "," + tmp.ver\_date) |
| 40 | STATE | ‘CHANGE’ |
| 41 | PERIOD\_STATUS | ‘NEW’ |
| 44 | CREATED\_USER | ‘SYSTEM’ |
| 45 | CREATED\_DATE | NOW() |

**B4.2:** Nếu B3.1 tồn tại bản ghi tbl\_bi\_collateral thực hiện update tbl\_bi\_collateral từ tmp theo quy tắc

**B4.2.1:** Nếu điều kiện sinh kỳ thay đổi cũng merg

(tmp.hash\_code <> tbl\_bi\_collateral.hash\_code OR tmp.period\_condition <> tbl\_bi\_collateral. period\_condition)

AND tmp.mark\_renew <> ‘NO’

Thực hiện update lại thông tin tbl\_bi\_collateral

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên trường** | **Giá trị** |
| 2 | COLL\_TYPE | tmp.COLL\_TYPE |
| 3 | DESCRIPTION | tmp.DESCRIPTION |
| 4 | CURRENCY | tmp.CURRENCY |
| 5 | COUNTRY | tmp.COUNTRY |
| 6 | NOMINAL\_VALUE | tmp.NOMINAL\_VALUE |
| 7 | EXECUTION\_VALUE | tmp.EXECUTION\_VALUE |
| 8 | VALUE\_DATE | tmp.VALUE\_DATE |
| 9 | COLLATERAL\_CODE | tmp.COLLATERAL\_CODE |
| 10 | INSUR\_PROVIDER | tmp.INSUR\_PROVIDER |
| 11 | INSURANCE\_AMOUNT | tmp.INSURANCE\_AMOUNT |
| 12 | INSURANCE\_TYPE | tmp.INSURANCE\_TYPE |
| 13 | INSUR\_ISSUE\_DATE | tmp.INSUR\_ISSUE\_DATE |
| 14 | IS\_EXPIRY\_DATE | tmp.IS\_EXPIRY\_DATE |
| 15 | SEAB\_COL\_SOURCE | tmp.SEAB\_COL\_SOURCE |
| 16 | COLL\_STATUS\_CODE | tmp.COLL\_STATUS\_CODE |
| 17 | CONTRACT\_MD | tmp.CONTRACT\_MD |
| 18 | INSURANCE\_REQ | tmp.INSURANCE\_REQ |
| 19 | CONTRACT\_REF | tmp.CONTRACT\_REF |
| 20 | VER\_DATE | tmp.VER\_DATE |
| 21 | SEAB\_REVAL\_COMP | tmp.SEAB\_REVAL\_COMP |
| 22 | MARKET\_VALUE | tmp.MARKET\_VALUE |
| 23 | REVAL\_VALUE | tmp.REVAL\_VALUE |
| 24 | RE\_EXEC\_VALUE | tmp.RE\_EXEC\_VALUE |
| 25 | REG\_DATE | tmp.REG\_DATE |
| 26 | NOTARY\_DATE | tmp.NOTARY\_DATE |
| 27 | COLLATERAL\_ID | tmp.COLLATERAL\_ID |
| 28 | CUSTOMER\_ID | tmp.CUSTOMER\_ID |
| 29 | CUSTOMER\_NAME | tmp.CUSTOMER\_NAME |
| 30 | CO\_CODE | tmp.CO\_CODE |
| 31 | SEAB\_REVAL\_NO | tmp.SEAB\_REVAL\_NO |
| 32 | DATE\_SYNC\_DATA | tmp.DATE\_SYNC\_DATA |
| 33 | DATA\_STATUS | ‘ACTIVE’ |
| 34 | LOCKED\_DATE | tmp.LOCKED\_DATE |
| 35 | LOCKED\_KEY | tmp.LOCKED\_KEY |
| 36 | PRIORITY | 0 |
| 37 | PERIOD\_CONDITION | tmp.PERIOD\_CONDITION |
| 38 | PERIOD\_CONDITION\_INS | tmp.PERIOD\_CONDITION\_INS |
| 39 | HASH\_CODE | MD5(tmp.value\_date + ',' + tmp.insur\_issue\_date + ',' + tmp.is\_expiry\_date + "," + tmp.ver\_date) |
| 40 | STATE | ‘CHANGE’ |
| 41 | PERIOD\_STATUS | ‘RENEW’ |
| 42 | locked\_date | NULL |
| 43 | locked\_key | NULL |
| 44 | updated\_date | NOW() |
| 45 | updated\_user | ‘SYSTEM’ |

**B4.2.2:** Cập nhật tbl\_bi\_collateral.collateral\_future\_key với điều kiện

seab\_col\_source = ‘2’ AND period\_condition = ‘ENOUGH’ AND collateral\_code = ‘420’

|  |  |
| --- | --- |
| **Tên trường** | **Giá trị** |
| **Thông tin trường Update** | |
| collateral\_future\_key | MD5(collateral\_id) |

**B4.2.3:** Cập nhật tbl\_bi\_collateral 3 trường sau

|  |  |
| --- | --- |
| **Tên trường** | **Giá trị** |
| **Thông tin trường Update** | |
| DATE\_SYNC\_DATA | tmp. DATE\_SYNC\_DATA |
| updated\_date | NOW() |
| updated\_user | ‘SYSTEM’ |

**B5:** Đánh dấu ktra ngừng theo dõi với các mã TSBD ko đủ ĐK sinh kỳ

Nếu thỏa mãn điều kiện sau thì thực hiện update tbl\_bi\_collateral

Điều kiện:

tbl\_bi\_collateral.period\_condition = ‘NOT\_ENOUGH’

AND tbl\_bi\_collateral.period\_condition <> tmp.period\_condition

Update:

|  |  |
| --- | --- |
| **Tên trường** | **Giá trị** |
| **Thông tin trường Update** | |
| state | ‘STOP\_FOLLOW’ |
| PRIORITY | 0 |
| locked\_date | NULL |
| locked\_key | NULL |

**B6:** Sau quá trình mapping cập nhật trạng thái đồng bộ tham chiếu [1.3.2.4](#_(4)_Update_lại)

#### (3) Insert / Update dữ liệu merge tbl\_bi\_collateral và tmp

Cấu trúc bảng [mapping tbl\_bi\_collateral](#_Bảng_tbl_bi_collateral)

Sau quá trình mapping dữ liệu sẽ tồn tại 2 bản ghi cần lưu thay đổi

- tbl\_bi\_collateral: Bản ghi mới Hoặc Bản ghi được update dữ liệu

Chỉ thực hiện lưu thay đổi nếu thỏa mãn điều kiện

tbl\_bi\_collateral.state = ‘CHANGE’ OR tbl\_bi\_collateral.state = ‘STOP\_FOLLOW’

- tbl\_bi\_collateral\_tmp: Bản ghi được update dữ liệu

Thực hiện update vào DB bản ghi thay đổi

#### (4) Update lại trạng thái đồng bộ bảng TBL\_BI\_COLLATERAL\_TMP

|  |  |
| --- | --- |
| **Tên trường** | **Giá trị** |
| **Thông tin trường Update** | |
| mapping\_status | *SUCCESS* |
| updated\_date | NOW() |
| updated\_user | SYSTEM |
| **Điều kiện update** | |
| LOCKED\_KEY = BatchId sinh random từ B1 | |

#### (5) Exception khi chạy job

|  |  |
| --- | --- |
| **Tên trường** | **Giá trị** |
| **Thông tin trường Update** | |
| LOCKED\_KEY | *null* |
| MAPPING\_STATUS | RENEW |
| LOCKED\_DATE | null |
| **Điều kiện update** | |
| MAPPING\_STATUS = 'PROCESSING' | |
| ((cast(:current\_date as date) - cast(LOCKED\_DATE as date)) \* 24 \* 60 > 5 | |

### Bảng tbl\_bi\_collateral

**TBL\_BI\_COLLATERAL**

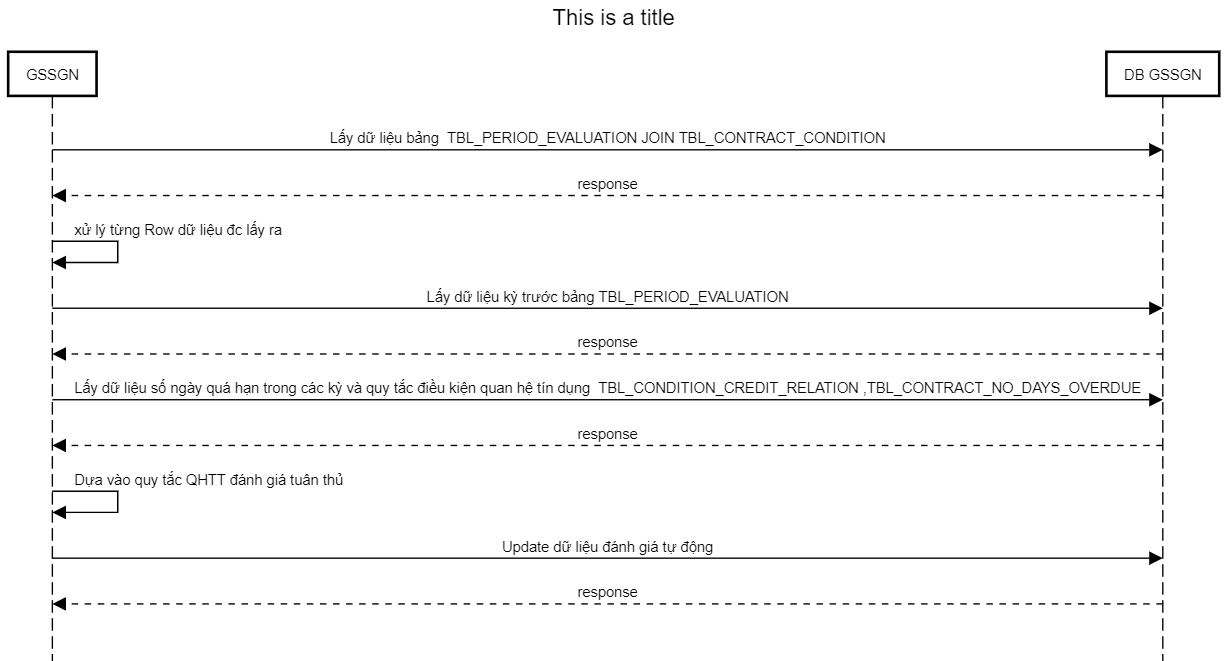
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Bắt buộc** | **Khóa** | **Mô tả** |
| 1 | ID | VARCHAR2(36) | x | x | Khóa bản ghi |
| 2 | COLL\_TYPE | VARCHAR2(255 CHAR) |  |  | Loại tài sản đảm bảo (link đến bảng loại tài sản đảm bảo) TBL\_BI\_COLLATERAL\_TYPE  .COLLA\_TYPE\_CODE |
| 3 | DESCRIPTION | VARCHAR2(4000 CHAR) |  |  | Mô tả |
| 4 | CURRENCY | VARCHAR2(255 |  |  | Loại tiền tệ |
| 5 | COUNTRY | VARCHAR2(255) |  |  | Quốc gia |
| 6 | NOMINAL\_VALUE | VARCHAR2(255) |  |  | Số tiền theo MD |
| 7 | EXECUTION\_VALUE | VARCHAR2(255) |  |  | Giá trị tài sản |
| 8 | VALUE\_DATE | VARCHAR2(255) |  |  | Ngày hiệu lực |
| 9 | COLLATERAL\_CODE | VARCHAR2(255) |  |  | Mã tài sản đảm bảo |
| 10 | INSUR\_PROVIDER | VARCHAR2(255) |  |  | Đơn vị bảo hiểm |
| 11 | INSURANCE\_AMOUNT | VARCHAR2(255) |  |  | Giá trị bảo hiểm tài sản |
| 12 | INSURANCE\_TYPE | VARCHAR2(255) |  |  | Loại bảo hiểm |
| 13 | INSUR\_ISSUE\_DATE | VARCHAR2(255) |  |  | Ngày hiệu lực bảo hiểm  Định dạng: YYYY-MM-DD HH:mm:ss |
| 14 | IS\_EXPIRY\_DATE | VARCHAR2(255) |  |  | Ngày hết hạn  Định dạng: YYYY-MM-DD HH:mm:ss |
| 15 | SEAB\_COL\_SOURCE | VARCHAR2(255) |  |  | Trạng thái hình thành tài sản đảm bảo  *1 - đã hình thành*  *2 - chưa hình thành* |
| 16 | COLL\_STATUS\_CODE | VARCHAR2(255) |  |  | Trạng thái thế chấp: gồm các giá trị: 1/2/3/4/5 |
| 17 | CONTRACT\_MD | VARCHAR2(255) |  |  | Mã hợp đồng Tài sản |
| 18 | INSURANCE\_REQ | VARCHAR2(255) |  |  | TS có bảo hiểm (Yes/No) |
| 19 | CONTRACT\_REF | VARCHAR2(4000) |  |  | Danh sách Mã REF |
| 20 | VER\_DATE | VARCHAR2(255) |  |  | Ngày định giá lại, Định dạng: YYYY-MM-DD HH:mm:ss |
| 21 | SEAB\_REVAL\_COMP | VARCHAR2(255) |  |  | Đơn vị định giá lại |
| 22 | MARKET\_VALUE | VARCHAR2(255) |  |  | Giá thị trường |
| 23 | REVAL\_VALUE | VARCHAR2(255) |  |  | Giá trị định giá lại |
| 24 | RE\_EXEC\_VALUE | VARCHAR2(255) |  |  | Giá trị vay tối đa trên TS |
| 25 | REG\_DATE | VARCHAR2(255) |  |  | Ngày bàn giao BĐS theo CT, Định dạng: YYYY-MM-DD HH:mm:ss |
| 26 | NOTARY\_DATE | VARCHAR2(255) |  |  | Ngày bàn giao GCN theo CT, Định dạng: YYYY-MM-DD HH:mm:ss |
| 27 | COLLATERAL\_ID | VARCHAR2(255) |  |  | Mã TSĐB  **Dùng làm Key nghiệp vụ** |
| 28 | CUSTOMER\_ID | VARCHAR2(255) |  |  | Mã khách hàng |
| 29 | CUSTOMER\_NAME | VARCHAR2(255 BYTE) |  |  | Tên khách hàng |
| 30 | CO\_CODE | VARCHAR2(255) |  |  | Mã CN/PGD |
| 31 | SEAB\_REVAL\_NO | VARCHAR2(255) |  |  | Số chứng thư định giá |
| 32 | DATE\_SYNC\_DATA | VARCHAR2(255) |  |  | Ngày đồng bộ dữ liệu: YYYY-MM-DD |
| 33 | DATA\_STATUS | VARCHAR2(50) | x |  | Trạng thái:   * PRE dữ liệu chưa thể sử dụng * ACTIVE dữ liệu có thể sử dụng |
| 34 | LOCKED\_DATE | TIMESTAMP(6) |  |  | Thời gian khóa bản ghi khi xử lý nghiệp vụ  Định dạng *DD-MMM-YY hh.mm.ss.fffffffff tt* |
| 35 | LOCKED\_KEY | VARCHAR2(40) |  |  | ID khóa bản ghi khi xử lý nghiệp vụ |
| 36 | PRIORITY | NUMBER |  |  | Mức độ ưu tiên |
| 37 | PERIOD\_CONDITION | VARCHAR2(255) |  |  | Trạng thái đánh giá TSĐB đủ điều kiện sinh kỳ hay không   * ENOUGH, * NOT\_ENOUGH |
| 38 | PERIOD\_CONDITION\_INS | VARCHAR2(20 BYTE) | x |  | Trạng thái đánh giá BHTSĐB đủ điều kiện sinh kỳ hay không   * ENOUGH, * NOT\_ENOUGH |
| 39 | HASH\_CODE | VARCHAR2(255) |  |  | Mã check các trường thay đổi |
| 40 | STATE | VARCHAR2(20 BYTE) |  |  | Đánh giá xem bản ghi có thay đổi so với ngày trước đó hay không   * CHANGE, * NOT\_CHANGE, * STOP\_FOLLOW |
| 41 | PERIOD\_STATUS | VARCHAR2(20 BYTE) |  |  | Trạng thái sinh kỳ   * *NEW,* * *RENEW,* * *PROCESSING,* * *SUCCESS,* * *ERROR* |
| 42 | CREATE\_FUTURE\_BATCH\_ID | VARCHAR2(255 BYTE) |  |  | ID job chạy dữ liệu (sinh random string) |
| 43 | COLLATERAL\_FUTURE\_KEY | VARCHAR2(255 BYTE) |  |  | Mã hóa MD5 của trường COLLATERAL\_ID |
| 44 | CREATED\_USER | VARCHAR2(255 CHAR) |  |  | Username người tạo |
| 45 | CREATED\_DATE | TIMESTAMP(6) |  |  | Thời gian tạo  Định dạng *DD-MMM-YY hh.mm.ss.fffffffff tt* |
| 46 | UPDATED\_USER | VARCHAR2(255 CHAR) |  |  | Username người cập nhật |
| 47 | UPDATED\_DATE | TIMESTAMP(6) |  |  | Thời gian cập nhật  Định dạng *DD-MMM-YY hh.mm.ss.fffffffff tt* |

## Tự động dánh giá dựa trên điều kiện quan hệ tín dụng

### Mô tả nghiệp vụ

Sau khi duyệt điều kiện quan hệ tín dụng của Hợp đồng sẽ sinh ra kỳ đánh giá. Job này thực hiện tác vụ đánh giá tuân thủ điều kiện QHTT của hợp đồng

### Biểu đồ tuần tự



### Quy tắc xử lý dữ liệu

#### Bước 1 : Lấy dữ liệu kỳ đánh giá thỏa mãn điều kiện đánh giá :

SELECT

\*

FROM

TBL\_PERIOD\_EVALUATION pe

JOIN TBL\_CONTRACT\_CONDITION cc ON cc.id = pe.CONTRACT\_CONDITION\_ID

WHERE pe.EVALUATION\_DATE IS NOT NULL

AND pe.AUTO\_EVALUATION\_RESULT IS NULL

AND pe.GROUP\_CONTRACT\_CONDITION = 'CREDIT\_RELATION'

AND cc.STATUS = 'APPROVED'

AND ( (pe.REGISTER\_DATE IS NOT NULL AND TRUNC(pe.REGISTER\_DATE) < TRUNC (sysdate)) OR (pe.REGISTER\_DATE IS NULL AND TRUNC(pe.EVALUATION\_DATE) < TRUNC (sysdate)) );

#### Bước 2 : xử lý dữ liệu ( từng Row)

##### B2.1 Lấy dữ liệu kỳ trước :

SELECT

\*

FROM

TBL\_PERIOD\_EVALUATION pe

JOIN TBL\_PERIOD\_EVALUATION pe1 ON pe.CONTRACT\_CONDITION\_ID = pe1.CONTRACT\_CONDITION\_ID AND pe.PERIOD - 1 = pe1.PERIOD

WHERE AND pe.FREQUENCY = 1 AND pe.FREQUENCY\_UNIT = 'M' AND pe.CONTRACT\_CONDITION\_ID = :contractConditionId;

Nếu có dữ liệu kỳ trước và thông tin AUTO\_EVALUATION\_RESULT kỳ trước bằng 'NOT\_FOLLOW' đánh dấu lại.

##### B2.2 Truy vấn thông tin ngày quá hạn (NO\_DAY\_OVERDUE ) và check vi phạm quy tắc nghị quyết

- Lấy dữ liệu cấu hình điều kiện nghị quyết Quan Hệ tín dụng ( dữ liệu này có thể là số nhiều :

SELECT \* TBL\_CONDITION\_CREDIT\_RELATION cr WHERE cr.CONDITION\_ID = :contractConditionId;

Lấy dữ liệu bảng TBL\_CONTRACT\_NO\_DAYS\_OVERDUE theo điều kiện TBL\_PERIOD\_EVALUATION.CONTRACT\_ID = TBL\_CONTRACT\_NO\_DAYS\_OVERDUE.CONTRACT\_ID

và quy tắc được cấu hình trong TBL\_CONTRACT\_CONDITION như sau :

TBL\_CONDITION\_CREDIT\_RELATION.operator = 'GREATER' so sánh TBL\_CONDITION\_CREDIT\_RELATION.OVERDUE\_DATE < TBL\_CONTRACT\_NO\_DAYS\_OVERDUE.NO\_DAYS\_OVERDUE ==> Ko Tuân thủ

TBL\_CONDITION\_CREDIT\_RELATION.operator = 'LESS' so sánh TBL\_CONDITION\_CREDIT\_RELATION.OVERDUE\_DATE > TBL\_CONTRACT\_NO\_DAYS\_OVERDUE.NO\_DAYS\_OVERDUE ==> Ko Tuân thủ

TBL\_CONDITION\_CREDIT\_RELATION.operator = 'GREATER\_EQUAL' so sánh TBL\_CONDITION\_CREDIT\_RELATION.OVERDUE\_DATE <= TBL\_CONTRACT\_NO\_DAYS\_OVERDUE.NO\_DAYS\_OVERDUE ==> Ko Tuân thủ

TBL\_CONDITION\_CREDIT\_RELATION.operator = 'LESS\_EQUAL' so sánh TBL\_CONDITION\_CREDIT\_RELATION.OVERDUE\_DATE >= TBL\_CONTRACT\_NO\_DAYS\_OVERDUE.NO\_DAYS\_OVERDUE ==> Ko Tuân thủ

##### B2.3 Tính toán kết quả tuân thủ

Nếu ko tìm đc thông tin NO\_DAY\_OVERDUE : Đánh dấu AUTO\_EVALUATION\_RESULT = 'FOLLOW'

Nếu có dữ liệu thông tin NO\_DAY\_OVERDUE :

( TBL\_CONDITION\_CREDIT\_RELATION.PERIOD là thông tin tổng số tháng cần check ko tuân thủ )

+ TH1 TBL\_CONDITION\_CREDIT\_RELATION.CONTINUOUS = 1 ==> check liên tiếp số tháng không tuân thủ

VD : TBL\_CONDITION\_CREDIT\_RELATION.PERIOD = 2 ;TBL\_CONDITION\_CREDIT\_RELATION.operator = 'GREATER'

Kỳ hiện tại là kỳ 2024-01 ==> dữ liệu kỳ 2024-01 và 2023-12 (2 kỳ liên tiếp) phải đảm bảo : TBL\_CONDITION\_CREDIT\_RELATION.OVERDUE\_DATE < TBL\_CONTRACT\_NO\_DAYS\_OVERDUE.NO\_DAYS\_OVERDUE (dk ko tuân thủ)

==> Đánh dấu AUTO\_EVALUATION\_RESULT = 'NOT\_FOLLOW' ngược lại Đánh dấu AUTO\_EVALUATION\_RESULT = 'FOLLOW'

+ TH2 TBL\_CONDITION\_CREDIT\_RELATION.CONTINUOUS = 0 ==> check tổng số tháng không tuân thủ

VD : TBL\_CONDITION\_CREDIT\_RELATION.operator = 'GREATER'

Tổng số kỳ thỏa màn điều kiện TBL\_CONDITION\_CREDIT\_RELATION.OVERDUE\_DATE < TBL\_CONTRACT\_NO\_DAYS\_OVERDUE.NO\_DAYS\_OVERDUE (dk ko tuân thủ) >= TBL\_CONDITION\_CREDIT\_RELATION.PERIOD

==> Đánh dấu AUTO\_EVALUATION\_RESULT = 'NOT\_FOLLOW' ngược lại Đánh dấu AUTO\_EVALUATION\_RESULT = 'FOLLOW'

##### B2.4 Check điều kiện kết hợp TBL\_CONDITION\_CREDIT\_RELATION.TYPE

+ nếu TBL\_CONDITION\_CREDIT\_RELATION.TYPE = 'MATCH\_ANY' ==> nếu bất kỳ bản ghi TBL\_CONDITION\_CREDIT\_RELATION nào ko tuân thủ

==> update TBL\_PERIOD\_EVALUATION.AUTO\_EVALUATION\_RESULT = 'NOT\_FOLLOW' ngược lại TBL\_PERIOD\_EVALUATION.AUTO\_EVALUATION\_RESULT = 'FOLLOW'. Đông thời update TBL\_PERIOD\_EVALUATION.EVALUATION\_RESULT = AUTO\_EVALUATION\_RESULT nếu EVALUATION\_RESULT is null

+ nếu TBL\_CONDITION\_CREDIT\_RELATION.TYPE = 'MATCH\_ALL' ==> nếu tất cả các bản ghi TBL\_CONDITION\_CREDIT\_RELATION ko tuân thủ

==> update TBL\_PERIOD\_EVALUATION.AUTO\_EVALUATION\_RESULT = 'NOT\_FOLLOW' ngược lại TBL\_PERIOD\_EVALUATION.AUTO\_EVALUATION\_RESULT = 'FOLLOW'. Đông thời update TBL\_PERIOD\_EVALUATION.EVALUATION\_RESULT = AUTO\_EVALUATION\_RESULT nếu EVALUATION\_RESULT is null

//TODO : Tạm thời thiếu thông tin trường AUTO\_EVALUATION\_DATA do trường này lưu dữ liệu JsonString ( sẽ tìm phương án lưu trực tiếp dữ liệu cần sử dụng )

### Mô tả DB bảng TBL\_PERIOD\_EVALUATION

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Bắt buộc** | **Khóa** | **Mô tả** |
| 1 | ID | VARCHAR2(255 CHAR) | x | pk | ID bảng |
| 2 | CO\_CODE | VARCHAR2(255 CHAR) |  |  | Mã đơn vị |
| 3 | CONTROL\_METHOD | VARCHAR2(255 CHAR) |  |  | Phương thức kiểm soát  Link trường *TBL\_CONTROL\_METHOD.CODE* |
| 4 | CUSTOMER\_ID | VARCHAR2(255 CHAR) |  |  | ID khách hàng |
| 5 | REPORT\_DATE | TIMESTAMP(6) |  |  | Ngày biên bản |
| 6 | BUSINESS\_UNIT\_NOTE | NUMBER(1,0) | x |  | Lưu ý đơn vị kinh doanh |
| 7 | VALUE\_DATE\_MIN | TIMESTAMP(6) |  |  | Ngày giải ngân đầu tiên |
| 8 | AUTO\_PERIOD | NUMBER(1,0) | x |  | Tự động sinh kỳ (0/1) |
| 9 | CREATE\_NEXT\_PERIOD\_STATUS | VARCHAR2(255 CHAR) | x |  | Trạng thái kỳ đánh giá   * *NEW,* * *RENEW,* * *PROCESSING,* * *SUCCESS,* * *ERROR* |
| 10 | PRIORITY | NUMBER | x |  | Thú tự, default 0 |
| 11 | LOCKED\_DATE | TIMESTAMP(6) |  |  | Thời gian khóa bản ghi khi xử lý nghiệp vụ |
| 12 | LOCKED\_KEY | VARCHAR2(40 CHAR) |  |  | ID khóa bản ghi khi xử lý nghiệp vụ |
| 13 | CONTRACT\_CONDITION\_ID | VARCHAR2(255 CHAR) |  |  | ID bảng TBL\_CONTRACT\_CONDITION.ID |
| 14 | CONTRACT\_ID | VARCHAR2(255 CHAR) |  |  | Mã hợp đồng REF, link trường STG\_SEAOPS.TBL\_BI\_CONTRACT\_REF\_DEFINE.REF\_ID |
| 15 | CREATED\_DATE | TIMESTAMP(6) |  |  | Ngày tạo |
| 16 | CREATED\_USER | VARCHAR2(255 CHAR) |  |  | User tạo |
| 17 | EVALUATION\_DATE | TIMESTAMP(6) |  |  | Ngày đánh giá |
| 18 | EVALUATION\_RESULT | VARCHAR2(255 CHAR) |  |  | Kết quả đánh giá cuối cùng   * FOLLOW("Tuân thủ"), * NOT\_FOLLOW("Không tuân thủ"), * NOT\_ENOUGH\_EVALUATION\_DATA("Không đủ dữ liệu đánh giá"); |
| 19 | FIRST\_DATE\_CHOSEN\_PERIOD | TIMESTAMP(6) |  |  | Ngày kiểm tra đầu tiên |
| 20 | FREQUENCY | NUMBER(10,0) | x |  | Tần suất |
| 21 | FREQUENCY\_UNIT | VARCHAR2(255 CHAR) |  |  | Đơn vị tần suất   * *D ngày* * *M tháng* * *LAST\_DAY ngày cuối cùng của tháng* |
| 22 | GROUP\_CONTRACT\_CONDITION | VARCHAR2(255 CHAR) | x |  | Nhóm điều kiện nghị quyết   * *CROSS\_SELLING // bán chéo* * *REVALUE\_COLLATERAL // Định giá lại TSĐB* * *CREDIT\_RELATION // Quan hệ tín dụng* * *PERIODIC // Định kỳ* * *MDSDV // MĐSDV* * *LIFE\_INSURANCE // bảo hiểm nhân thọ* * *FINANCIAL // tài chính* * *LAW // pháp lý* * *SECUR\_PROP\_INSURANCE // bảo hiểm TSĐB* * *OTHER// khác* * *REVENUE // Doanh thu* * *LOAN\_PRODUCT\_CONDITION // Điều kiện sản phẩm vay* |
| 23 | PERIOD | NUMBER(10,0) | x |  | Kỳ số n |
| 24 | UPDATED\_DATE | TIMESTAMP(6) |  |  | Ngày cập nhật dữ liệu |
| 25 | UPDATED\_USER | VARCHAR2(255 CHAR) |  |  | User cập nhật |
| 26 | MANUAL\_EVALUATION\_RESULT | VARCHAR2(20 CHAR) |  |  | Kết quả đánh giá bằng tay   * *FOLLOW("Tuân thủ")* * *NOT\_FOLLOW("Không tuân thủ")* * *NOT\_ENOUGH\_EVALUATION\_DATA("Không đủ dữ liệu đánh giá")* |
| 27 | AUTO\_EVALUATION\_RESULT | VARCHAR2(50 CHAR) |  |  | Kết quả đánh giá tự động   * *FOLLOW("Tuân thủ")* * *NOT\_FOLLOW("Không tuân thủ")* * *NOT\_ENOUGH\_EVALUATION\_DATA("Không đủ dữ liệu đánh giá")* |
| 28 | LOAN\_CODE | VARCHAR2(100 CHAR) |  |  | Mã khoản vay trong lịch sử giải ngân TBL\_BI\_DISBURSEMENT\_HISTORY.CONTRACT |
| 29 | COMPANY\_NAME\_VN | VARCHAR2(255 CHAR) |  |  | Tên chi nhánh  STG\_SEAOPS.TBL\_BI\_CONTRACT\_REF\_DEFINE.COMPANY\_NAME\_VN |
| 30 | SHORT\_NAME | VARCHAR2(255 CHAR) |  |  | Tên khách hàng  STG\_SEAOPS.TBL\_BI\_CONTRACT\_REF\_DEFINE.SHORT\_NAME |
| 31 | LOAN\_PRODUCT\_ID | VARCHAR2(255 CHAR) |  |  | Mã sản phẩm cho vay. Link đến bảng sản phẩm TBL\_BI\_LOAN\_PRODUCT.PRODUCT\_CODE |
| 32 | CONTENT | VARCHAR2(4000 CHAR) |  |  | Nội dung, lấy từ bảng TBL\_CONTRACT\_CONDITION.CONTENT |
| 33 | REASON | VARCHAR2(4000 CHAR) |  |  | Lý do từ chối kỳ đánh giá |
| 34 | CONTRACT\_CONDITION\_STATUS | VARCHAR2(20 CHAR) |  |  | Trạng thái điều kiện nghị quyết   * *SAVED "Đã lưu" // Đã nhập* * *PENDING "Chờ duyệt"* * *APPROVED "Đã duyệt"* * *STOP\_FOLLOW "Ngừng theo dõi"* * *RETURN "Trả về"* * *REJECTED "Từ chối"* * *OVERDUE\_1 "Ngừng theo dõi" // QUÁ HẠN theo case 1* * *OVERDUE\_2 "Qua han 90 ngay" // QUÁ HẠN theo case 2* |
| 35 | CONTRACT\_STATUS | VARCHAR2(20 CHAR) |  |  | Trạng thái hợp đồng.   * *SAVED "Đã lưu" // Đã nhập* * *PENDING "Chờ duyệt"* * *APPROVED "Đã duyệt"* * *STOP\_FOLLOW "Ngừng theo dõi"* * *RETURN "Trả về"* * *REJECTED "Từ chối"* * *OVERDUE\_1 "Ngừng theo dõi" // QUÁ HẠN theo case 1* * *OVERDUE\_2 "Qua han 90 ngay" // QUÁ HẠN theo case 2* |
| 36 | VALUATION\_USER | VARCHAR2(255 CHAR) |  |  | User đánh giá  (Hiện tại đã bỏ trong code) |
| 37 | VALUATION\_DATE | TIMESTAMP(6) |  |  | Ngày đánh giá |
| 38 | VALUATION\_UNIT | VARCHAR2(255 CHAR) |  |  | Đơn vị đánh giá |
| 39 | NOTE | VARCHAR2(4000 CHAR) |  |  | Ghi chú |
| 40 | STATUS | VARCHAR2(255 CHAR) |  |  | Trạng thái, từ bảng STG\_SEAOPS.TBL\_BI\_CONTRACT\_REF\_DEFINE   * *SAVED "Đã lưu" // Đã nhập* * *PENDING "Chờ duyệt"* * *APPROVED "Đã duyệt"* * *STOP\_FOLLOW "Ngừng theo dõi"* * *RETURN "Trả về"* * *REJECTED "Từ chối"* * *OVERDUE\_1 "Ngừng theo dõi" // QUÁ HẠN theo case 1* * *OVERDUE\_2 "Qua han 90 ngay" // QUÁ HẠN theo case 2* |
| 41 | LOS\_QLSV | VARCHAR2(255 CHAR) |  |  | Mã LOS |
| 42 | CUSTOMER\_NAME | VARCHAR2(255 CHAR) |  |  | Tên khách hàng, từ bảng TBL\_BI\_COLLATERAL.CUSTOMER\_NAME |
| 43 | IS\_CHEAT | NUMBER(1,0) |  |  | Dấu hiệu gian lận (0/1) |
| 44 | INVOICE\_NO | VARCHAR2(255 CHAR) |  |  | Số hợp đồng bảo hiểm/số hồ sơ bảo hiểm |
| 45 | FILE\_RECEIVE | VARCHAR2(4000 CHAR) |  |  | Hồ sơ nhận |
| 46 | FILE\_MISSING | VARCHAR2(4000 CHAR) |  |  | Hồ sơ thiếu |
| 47 | VALUATION\_AMOUNT | NUMBER(19,0) |  |  | Giá trị định giá lại |
| 48 | MARKET\_AMOUNT | NUMBER(19,0) |  |  | Giá thị trường |
| 49 | MAX\_LOAN\_AMOUNT | NUMBER(19,0) |  |  | Giá trị vay tối đa |
| 50 | RISK\_LEVEL | VARCHAR2(255 CHAR) |  |  | Mức độ rủi ro   * *LOW "Thấp"* * *MEDIUM "Trung bình"* * *HIGH "Cao"* |
| 51 | REGISTER\_DATE | TIMESTAMP(6) |  |  | Ngày gia hạn |
| 52 | FORCE\_CREATE\_NEXT | NUMBER(1,0) |  |  | Bắt buộc tạo kỳ tiếp theo (0/1) |
| 53 | REVOLVING | VARCHAR2(255 CHAR) |  |  | Loại khoản vay. STG\_SEAOPS.TBL\_BI\_CONTRACT\_REF\_DEFINE.REVOLVING |
| 54 | DESCRIPTION | VARCHAR2(4000 CHAR) |  |  | Mô tả TBL\_CONTRACT\_CONDITION.DESCRIPTION |
| 55 | COLLATERAL\_ID | VARCHAR2(255 CHAR) |  |  | Mã chi tiết tài sản đảm bảo  TBL\_BI\_COLLATERAL.COLLATERAL\_ID |
| 56 | COLLATERAL\_TYPE | VARCHAR2(255 CHAR) |  |  | Loại tài sản đảm bảo  TBL\_BI\_COLLATERAL\_TYPE.COLLA\_TYPE\_CODE |
| 57 | REASON\_RETURN | VARCHAR2(4000 CHAR) |  |  | Lý do từ chối |
| 58 | CONTRACT\_CONDITION\_REASON | VARCHAR2(4000 CHAR) |  |  | Lý do trả về đơn vị kinh doanh  TBL\_CONTRACT\_CONDITION.REASON |
| 59 | DOCUMENT\_NAME | VARCHAR2(4000 CHAR) |  |  | Tên chứng từ yêu cầu  TBL\_CONTRACT\_CONDITION.DOCUMENT\_NAME |
| 60 | MDSDV\_IMAGE | NUMBER(1,0) |  |  | Ảnh mục đích sử dụng vốn |
| 61 | AUTO\_EVALUATION\_BATCH\_ID | VARCHAR2(255 BYTE) |  |  | Trạng thái tự động đánh giá   * NULL/STOP\_EVALUATION |
| 62 | PERIODIC\_TYPE | VARCHAR2(100 CHAR) |  |  | Loại tần suất kiểm tra định kỳ. Giá trị TBL\_PERIODIC\_TEST\_FREQUENCY.CUSTOMER\_TYPE + # + SHORT\_TERM/MIDDLE\_LONG\_TERM |
| 63 | LOAN\_PRODUCT\_CODE | VARCHAR2(100 CHAR) |  |  | Mã sản phẩm vay. TBL\_LOAN\_PRODUCT\_CONDITION.CODE |
| 64 | CERT\_NUMBER\_VALUATION | VARCHAR2(255 CHAR) |  |  | Số chứng từ định giá lại |
| 65 | INSURANCE\_AMOUNT | VARCHAR2(255 CHAR) |  |  | Số tiền bảo hiểm |
| 66 | INSURANCE\_COMPANY | VARCHAR2(255 CHAR) |  |  | Công ty bảo hiểm |
| 67 | INSURANCE\_END\_DATE | TIMESTAMP(6) |  |  | Ngày kết thúc bảo hiểm |
| 68 | INSURANCE\_NOTE | VARCHAR2(4000 CHAR) |  |  | Ghi chú bảo hiểm |
| 69 | INSURANCE\_START\_DATE | TIMESTAMP(6) |  |  | Ngày bắt đầu bảo hiểm |
| 70 | INSURANCE\_STATUS | VARCHAR2(3 CHAR) |  |  | Trạng thái bảo hiểm (YES/NO) |
| 71 | INSURANCE\_TYPE | VARCHAR2(255 CHAR) |  |  | Loại bảo hiểm |
| 72 | VALUATION\_NOTE | VARCHAR2(4000 CHAR) |  |  | Ghi chú định giá lại |
| 73 | ROLE\_USER | VARCHAR2(50 CHAR) |  |  | Quyền người dùng thực hiện |
| 74 | REAL\_AMOUNT\_TRANSFERRED | VARCHAR2(255 CHAR) |  |  | Số tiền thực chuyển |
| 75 | COLLATERAL\_CONTRACT\_REF | VARCHAR2(4000 CHAR) |  |  | Danh sách hợp đồng theo mã tài sản đảm bảo bảng TBL\_BI\_COLLATERAL.CONTRACT\_REF |
| 76 | APPROVED\_USER | VARCHAR2(255 CHAR) |  |  | User phê duyệt |
| 77 | REPORT\_BATCH\_ID | VARCHAR2(255 CHAR) |  |  | BatchId chạy báo cáo áp chế tài |
| 78 | REPORT\_NOTE | VARCHAR2(2000 CHAR) |  |  | Ghi chú báo cáo |
| 79 | AUTO\_EVALUATION\_DATA | CLOB |  |  | Dữ liệu đánh giá tự động |
| 80 | CUSTOMER\_CLASS | VARCHAR2(255 CHAR) |  |  | Phân khúc khách hàng  STG\_SEAOPS.TBL\_BI\_CONTRACT\_REF\_DEFINE.CUSTOMER\_CLASS |
| 81 | EVALUATED\_USER | VARCHAR2(255 CHAR) |  |  | User đánh giá |
| 82 | REVENUE\_MONEY\_1 | VARCHAR2(255 CHAR) |  |  | Giá trị sổ sách từ bảng TBL\_REVENUE\_CONFIGURATION.MONEY\_ON\_DOCUMENT  trong trường hợp config điều kiện doanh thu theo RULE\_NUMBER\_1 |
| 83 | REVENUE\_MONEY\_2 | VARCHAR2(255 CHAR) |  |  | Giá trị sổ sách từ bảng TBL\_REVENUE\_CONFIGURATION.MONEY\_ON\_DOCUMENT  trong trường hợp config điều kiện doanh thu theo RULE\_NUMBER\_2 |
| 84 | CHECK\_PERIOD\_REPORT | VARCHAR2(255 CHAR) |  |  |  |
| 85 | JOB\_EVALUATION\_REPORT\_DATE | TIMESTAMP(6) |  |  | Ngày đánh giá tự động |
| 86 | DATE\_SYNC\_DATA | VARCHAR2(255 CHAR) |  |  | ngày đồng bộ dữ liệu  TBL\_BI\_COLLATERAL.DATE\_SYNC\_DATA |
| 87 | PRE\_EVALUATION\_DATE | TIMESTAMP(6) |  |  | Ngày đánh giá của kỳ trước đó |
| 88 | CONTRACT\_IDS\_PERIOD | VARCHAR2(255 CHAR) |  |  | Lưu các hợp đồng đã sinh kỳ định kỳ tại thời điểm duyệt đánh giá bằng tay |
| 89 | STF\_LOCKED\_KEY | VARCHAR2(40 BYTE) |  |  | Key lock bản ghi để thực hiện nghiệp vụ dừng follow |
| 90 | STF\_LOCKED\_DATE | TIMESTAMP(6) |  |  | Thời gian lock bản ghi để thực hiện nghiệp vụ dừng follow |
| 91 | STF\_PRIORITY | NUMBER |  |  | Thứ tự thực hiện stop follow |

## Tự động đánh giá dựa trên DKNQ doanh thu

### Mô tả nghiệp vụ

Job này nhằm mục đích tự động đánh giá các hạng mục điều kiện nghị quyết về doanh thu được phê duyệt trên hợp đồng theo dõi của GSSGN

### Biểu đồ tuần tự

N/A

### Quy tắc xử lý dữ liệu

#### Lấy dữ liệu thỏa mãn điều kiện đánh giá

SELECT

\*

FROM

TBL\_PERIOD\_EVALUATION pe

JOIN TBL\_CONTRACT\_CONDITION cc ON cc.ID = pe.CONTRACT\_CONDITION\_ID

WHERE

pe.EVALUATION\_DATE IS NOT NULL

AND pe.AUTO\_EVALUATION\_RESULT IS NULL

AND pe.GROUP\_CONTRACT\_CONDITION = 'REVENUE'

AND cc.STATUS = 'APPROVED'

AND (

( pe.REGISTER\_DATE IS NOT NULL AND TRUNC( pe.REGISTER\_DATE ) < TRUNC( sysdate ) )

OR ( pe.REGISTER\_DATE IS NULL AND TRUNC( pe.EVALUATION\_DATE ) < TRUNC( sysdate ) )

);

#### Xử lý dữ liệu ( quy tắc theo từng row)

##### Bước 1 Tính phạm vi thời gian lấy dữ liệu (from\_date & to\_date)

\*) Tính from\_date

TH1: Nếu chưa có ngày đánh giá kỳ trước (pre\_evaluation\_date is null)

* Nếu first\_date\_chosen\_period is not null:

from\_date = first\_date\_chosen\_period + (period - 2)\*frequency (nếu frequency\_unit = D đơn vị + tính ngày ngược lại đơn vị + tính bằng tháng )

* Nếu first\_date\_chosen\_period is null:

from\_date = value\_date\_min + (period - 2)\*frequency (nếu frequency\_unit = D đơn vị + tính ngày ngược lại đơn vị + tính bằng tháng )

TH2: Nếu đã có ngày đánh giá kỳ trước (pre\_evaluation\_date is not null)

* Nếu period > 1:

from\_date = pre\_evaluation\_date + 1 ( DAYS)

* Nếu period <= 1:

from\_date = value\_date\_min

\*) Tính to\_date

to\_date = evalution\_date

##### Tính doanh số giải ngân, doanh số thu gốc và lãi, doanh số phát hành LC, doanh số phát hành bảo lãnh

\*) Doanh số giải ngân ( Bảng STG\_SEAOPS.TBL\_BI\_DISBURSEMENT\_SALES )

======================================================

disbursedSales = SELECT

SUM( DISTINCT LOC\_AMT )

FROM

STG\_SEAOPS.TBL\_BI\_DISBURSEMENT\_SALES

WHERE

CONTRACT\_REF =: contract\_ref

AND TRANS\_TYPE IN ( '08', '09', '12' )

AND ( TO\_DATE( VALUE\_DATE, 'DD-MM-YYYY' ) BETWEEN : from\_date AND : to\_date )

GROUP BY

CONTRACT,

DAYID,

TRANS\_TYPE;

======================================================

\*) Doanh số thu gốc và lãi ( Bảng STG\_SEAOPS.TBL\_BI\_DISBURSEMENT\_SALES )

======================================================

revenuePrincipalAndInterest =

SELECT

SUM( DISTINCT LOC\_AMT )

FROM

STG\_SEAOPS.TBL\_BI\_DISBURSEMENT\_SALES

WHERE

CONTRACT\_REF =: contract\_ref

AND TRANS\_TYPE IN ( '01', '02', '03' )

AND ( TO\_DATE( VALUE\_DATE, 'DD-MM-YYYY' ) BETWEEN : from\_date AND : to\_date )

GROUP BY

CONTRACT,

DAYID,

TRANS\_TYPE;

======================================================

\*) Doanh số LC ( Bảng STG\_SEAOPS. TBL\_BI\_LC\_RELEASE\_SALES )

======================================================

lcSales =

SELECT

SUM( LC\_AMOUNT\_LCY )

FROM

TBL\_BI\_LC\_RELEASE\_SALES

WHERE

( TO\_DATE( ISSUE\_DATE, 'DD-MM-YYYY' ) BETWEEN : from\_date AND : to\_date )

AND CUSTOMER = : customerId

GROUP BY

CUSTOMER;

======================================================

\*) Doanh số bảo lãnh ( Bảng STG\_SEAOPS. TBL\_BI\_LC\_RELEASE\_SALES )

======================================================

issueGuarantee = SELECT

SUM( AMOUNT )

FROM

TBL\_BI\_GUARANTEE\_SALES

WHERE

( TO\_DATE( PAST\_SCHED\_DATE, 'DD-MM-YYYY' ) BETWEEN : from\_date AND : to\_date )

AND CUSTOMER\_ID = : customerId

GROUP BY

CUSTOMER\_ID;

=======================================================

issueGuaranteeAdvance = SELECT

SUM( AMOUNT )

FROM

TBL\_BI\_GUARANTEE\_SALES

WHERE

( TO\_DATE( PAST\_SCHED\_DATE, 'DD-MM-YYYY' ) BETWEEN : from\_date AND : to\_date )

AND CUSTOMER\_ID = : customerId AND CATEGORY = '28030'

GROUP BY

CUSTOMER\_ID;

=======================================================

issueGuaranteeContract = SELECT

SUM( AMOUNT )

FROM

TBL\_BI\_GUARANTEE\_SALES

WHERE

( TO\_DATE( PAST\_SCHED\_DATE, 'DD-MM-YYYY' ) BETWEEN : from\_date AND : to\_date )

AND CUSTOMER\_ID = : customerId AND CATEGORY = '28010'

GROUP BY

CUSTOMER\_ID;

=======================================================

issueGuaranteeReciprocal = SELECT

SUM( AMOUNT )

FROM

TBL\_BI\_GUARANTEE\_SALES

WHERE

( TO\_DATE( PAST\_SCHED\_DATE, 'DD-MM-YYYY' ) BETWEEN : from\_date AND : to\_date )

AND CUSTOMER\_ID = : customerId AND CATEGORY = '28009'

GROUP BY

CUSTOMER\_ID;

=======================================================

issueGuaranteePayment = SELECT

SUM( AMOUNT )

FROM

TBL\_BI\_GUARANTEE\_SALES

WHERE

( TO\_DATE( PAST\_SCHED\_DATE, 'DD-MM-YYYY' ) BETWEEN : from\_date AND : to\_date )

AND CUSTOMER\_ID = : customerId AND CATEGORY = '28040'

GROUP BY

CUSTOMER\_ID;

=======================================================

issueGuaranteeLoan = SELECT

SUM( AMOUNT )

FROM

TBL\_BI\_GUARANTEE\_SALES

WHERE

( TO\_DATE( PAST\_SCHED\_DATE, 'DD-MM-YYYY' ) BETWEEN : from\_date AND : to\_date )

AND CUSTOMER\_ID = : customerId AND CATEGORY = '28042'

GROUP BY

CUSTOMER\_ID;

=======================================================

\*) Dư nợ hiện tại ( Bảng STG\_SEAOPS. TBL\_BI\_CONTRACT\_DEBT)

=======================================================

Debt = SELECT

SUM( CAST( DEBT AS DECIMAL ) )

FROM

TBL\_BI\_CONTRACT\_DEBT

WHERE

DAYID = : to\_date

AND CUSTOMER\_ID = : customerId

GROUP BY

CUSTOMER\_ID;

Lấy tổng dư nơ còn lại của KH trong với dayId = ngày thực hiện đánh giá (to\_date mục 3.5.3.2.1)

=======================================================

\*) Doanh số ghi có ( Bảng STG\_SEAOPS.TBL\_BI\_CREDIT\_SALES)

=======================================================

\*) Tổng Doanh số

creditSale = SELECT

SUM( DISTINCT CAST( AMOUNT\_LCY AS DECIMAL ) )

FROM

TBL\_BI\_CREDIT\_SALES

WHERE

( DAYID BETWEEN : from\_date AND : to\_date )

AND CUSTOMER\_ID = : customerId

GROUP BY

CUSTOMER\_ID,OUT\_REFERENCE;

=======================================================

\*) Tổng doanh số FT

creditSaleFT = SELECT

SUM( DISTINCT CAST( AMOUNT\_LCY AS DECIMAL ) )

FROM

TBL\_BI\_CREDIT\_SALES

WHERE

( DAYID BETWEEN : from\_date AND : to\_date )

AND CUSTOMER\_ID = : customerId AND OUT\_REFERENCE like 'FT%'

GROUP BY

CUSTOMER\_ID,OUT\_REFERENCE;

=======================================================

\*) Tổng doanh số TT

creditSaleTT = SELECT

SUM( DISTINCT CAST( AMOUNT\_LCY AS DECIMAL ) )

FROM

TBL\_BI\_CREDIT\_SALES

WHERE

( DAYID BETWEEN : from\_date AND : to\_date )

AND CUSTOMER\_ID = : customerId AND OUT\_REFERENCE like 'TT%'

GROUP BY

CUSTOMER\_ID,OUT\_REFERENCE;

=======================================================

\*) Tổng doanh số TF

creditSaleTF = SELECT

SUM( DISTINCT CAST( AMOUNT\_LCY AS DECIMAL ) )

FROM

TBL\_BI\_CREDIT\_SALES

WHERE

( DAYID BETWEEN : from\_date AND : to\_date )

AND CUSTOMER\_ID = : customerId AND OUT\_REFERENCE like 'TF%'

GROUP BY

CUSTOMER\_ID,OUT\_REFERENCE;

\*) Quy tắc tính doanh số theo kỳ

minCreditSales =

* Với đơn vị tần suất là tháng ( frequency\_unit = M ) :

+ Chia quãng thời gian giữa from\_date và to\_date ra thành các quãng từng tháng và tính tổng doanh số theo tùng quãng đó. VD : from\_date = 15/6/2023 -> to\_date = 14/9/2023 🡺 chia ra dc 3 row :

|  |  |  |
| --- | --- | --- |
| STT | Value\_date | value |
| 1 | 15/06/2023 - 14/07/2023 | Tổng theo quãng tương ứng trong bảng TBL\_BI\_CREDIT\_SALES |
| 2 | 15/07/2023 - 14/08/2023 | Tổng theo quãng tương ứng trong bảng TBL\_BI\_CREDIT\_SALES |
| 3 | 15/08/2023 - 14/09/2023 | Tổng theo quãng tương ứng trong bảng TBL\_BI\_CREDIT\_SALES |

* Với đơn vị tần suất là ngày ( frequency\_unit = D ) : tính tổng doanh số theo từng ngày và lưu lại dưới dạng list sau :

|  |  |  |
| --- | --- | --- |
| STT | Value\_date | value |
| 1 | 15/06/2023 | Tổng theo ngày tương ứng trong bảng TBL\_BI\_CREDIT\_SALES |
| 2 | 16/06/2023 | Tổng theo ngày tương ứng trong bảng TBL\_BI\_CREDIT\_SALES |
| n | 15/06/2023 + (n -1) days | Tổng theo ngày tương ứng trong bảng TBL\_BI\_CREDIT\_SALES |

=======================================================

\*) Số tiền giải ngân ( Bảng STG\_SEAOPS. TBL\_BI\_DISBURSEMENT\_HISTORY)

=======================================================

disbursementAmount = SELECT

SUM( BALANCE )

FROM

TBL\_BI\_DISBURSEMENT\_HISTORY

WHERE

CONTRACT\_REF = : contractId --( TBL\_PERIOD\_EVALUATION.COMTRACT\_ID)

GROUP BY

CONTRACT\_REF;

=======================================================

\*) Điều kiện cầu hình TOI ( Bảng GSSGN. TBL\_CONFIG\_INPUT\_TOI)

=======================================================

toiValue = SELECT

TOI\_VALUE

FROM

TBL\_CONFIG\_INPUT\_TOI

WHERE

CONTRACT\_ID = : contractId --( TBL\_PERIOD\_EVALUATION.COMTRACT\_ID)

AND CUSTOMER\_ID = : customerId --( TBL\_PERIOD\_EVALUATION.CUSTOMER\_ID)

AND PERIOD = : period; --( TBL\_PERIOD\_EVALUATION.PERIOD)

=======================================================

\*) Cấu hình điều kiện nghị quyết ( Bảng GSSGN. TBL\_CONFIG\_INPUT\_TOI)

=======================================================

revenueCondition = SELECT

ID,

AUTO\_PERIOD,

BUSINESS\_UNIT\_NOTE,

CONTENT,

CONTRACT\_ID,

CONTROL\_METHOD,

CREATED\_DATE,

CREATED\_USER,

DESCRIPTION,

DOCUMENT\_NAME,

FIRST\_DATE\_CHOSEN\_PERIOD,

FREQUENCY,

FREQUENCY\_UNIT,

NEW\_CUSTOMER,

RULE\_MATCH,

RULE\_NUMBER,

SANCTION,

SANCTION\_TYPE,

STATUS,

TOI,

TOI\_MATCH,

TOI\_VALUE,

TYPE\_PERIOD\_RATE,

UPDATED\_DATE,

UPDATED\_USER,

START\_PERIOD,

GROUP\_CONTRACT\_CONDITION,

REASON,

MIN\_APPROVAL\_VALUE,

IS\_MIN\_APPROVAL\_VALUE,

LOAN\_PRODUCT\_ID,

CUSTOMER\_ID,

CLONE\_ID,

AUTO\_EVALUATION\_RESULT,

CUSTOMER\_CLASS,

CO\_CODE,

COMPANY\_NAME\_VN,

VERSION\_CONFIG\_ID,

ORIGIN\_ID

FROM

TBL\_REVENUE\_CONDITION

WHERE

ID = : contractConditionId;--( TBL\_PERIOD\_EVALUATION.CONTRACT\_CONDITION\_id)

=======================================================

\*) Cấu hình doanh thu ( Bảng GSSGN. TBL\_REVENUE\_CONFIGURATION)

=======================================================

subRevenueConfigurations = SELECT

ID,

ADVANCE,

APPROVAL\_RATE,

CONTRACT,

CREATED\_DATE,

CREATED\_USER,

CREDIT\_REVENUE,

DEBT\_REVENUE,

DISBURMENT\_REVENUE,

FT\_ENTRY,

GUARANTEE\_REVENUE,

LC\_REVENUE,

LOAN,

MONEY\_ON\_DOCUMENT,

ORIGINAL\_INTEREST\_REVENUE,

OTHER\_NUMBER,

PAYMENT,

RECIPROCAL,

REVENUE\_CONDITION\_ID,

RULE\_NUMBER,

TF\_ENTRY,

TT\_ENTRY,

UPDATED\_DATE,

UPDATED\_USER,

DISBURSEMENT\_MONEY,

DOCUMENT\_ADDITIONAL,

AUTO\_EVALUATION\_RESULT,

VERSION\_CONFIG\_ID

FROM

TBL\_REVENUE\_CONFIGURATION

WHERE

REVENUE\_CONDITION\_ID = : contractConditionId;--( TBL\_PERIOD\_EVALUATION.CONTRACT\_CONDITION\_ID)

=======================================================

\*) Cấu hình doanh thu ( Bảng GSSGN. TBL\_REVENUE\_CONFIGURATION)

=======================================================

subRevenueConfigurations = SELECT

ID,

ADVANCE,

APPROVAL\_RATE,

CONTRACT,

CREATED\_DATE,

CREATED\_USER,

CREDIT\_REVENUE,

DEBT\_REVENUE,

DISBURMENT\_REVENUE,

FT\_ENTRY,

GUARANTEE\_REVENUE,

LC\_REVENUE,

LOAN,

MONEY\_ON\_DOCUMENT,

ORIGINAL\_INTEREST\_REVENUE,

OTHER\_NUMBER,

PAYMENT,

RECIPROCAL,

REVENUE\_CONDITION\_ID,

RULE\_NUMBER,

TF\_ENTRY,

TT\_ENTRY,

UPDATED\_DATE,

UPDATED\_USER,

DISBURSEMENT\_MONEY,

DOCUMENT\_ADDITIONAL,

AUTO\_EVALUATION\_RESULT,

VERSION\_CONFIG\_ID

FROM

TBL\_REVENUE\_CONFIGURATION

WHERE

REVENUE\_CONDITION\_ID = : contractConditionId;--( TBL\_PERIOD\_EVALUATION.CONTRACT\_CONDITION\_ID)

=======================================================

\*) Cấu hình tỉ lệ doanh thu theo kỳ ( Bảng GSSGN. TBL\_REVENUE\_CONFIGURATION)

=======================================================

periodRate = SELECT

TBL\_REVENUE\_PERIOD\_RATE.PERIOD,

TBL\_REVENUE\_PERIOD\_RATE.RATE,

TBL\_REVENUE\_PERIOD\_RATE.TYPE

FROM

TBL\_REVENUE\_PERIOD\_RATE

WHERE

( PERIOD =: period AND TYPE =: typePeriodRate ) --(typePeriodRate = TBL\_REVENUE\_CONDITION.TYPE\_PERIOD\_RATE , period = TBL\_PERIOD\_EVALUATION.PERIOD);

Nếu ko có dữ liệu periodRate tiếp tục truy vấn theo câu tiếp theo :

WITH max\_period AS ( SELECT MAX( PERIOD ) AS PERIOD, TYPE FROM TBL\_REVENUE\_PERIOD\_RATE GROUP BY TYPE );

SELECT

TBL\_REVENUE\_PERIOD\_RATE.PERIOD,

TBL\_REVENUE\_PERIOD\_RATE.RATE,

TBL\_REVENUE\_PERIOD\_RATE.TYPE

FROM

TBL\_REVENUE\_PERIOD\_RATE

JOIN max\_period ON max\_period.PERIOD = TBL\_REVENUE\_PERIOD\_RATE.PERIOD

AND max\_period.TYPE = TBL\_REVENUE\_PERIOD\_RATE.TYPE

WHERE

TBL\_REVENUE\_PERIOD\_RATE.TYPE =: typePeriodRate;

=======================================================

\*) Doanh số LC đã thanh toán ( Trường dữ liệu mới được đồng bộ về bảng dư nợ Bảng STG\_SEAOPS. TBL\_BI\_CONTRACT\_DEBT)

=======================================================

=======================================================

\*) Hạn mức thấu chi (Bảng TBL\_BI\_CONTRACT\_REF\_DEFINE)

=======================================================

SELECT CONTRACT\_AMT FROM TBL\_BI\_CONTRACT\_REF\_DEFINE WHERE CONTRACT\_ID = :contractId; --- Lấy từ dữ liệu đầu vào bảng TBL\_PERIOD\_EVALUATION

Sau khi xử lý ta lấy được dữ liệu sau :

|  |  |  |
| --- | --- | --- |
| **Key\_name** | **Kiểu dữ liệu** | **Mô tả** |
| disbursedSales | Number | Tổng doanh số giải ngân |
| revenuePrincipalAndInterest | Number | Tổng doanh số thu gốc, lãi |
| lcSales | Number | Tổng doanh số phát hanh LC |
| issueGuarantee | Number | Tổng doanh số phát hành bảo lãnh |
| issueGuaranteeAdvance | Number | Tổng doanh số phát hành bảo lãnh 28030 |
| issueGuaranteeContract | Number | Tổng doanh số phát hành bảo lãnh 28010 |
| issueGuaranteeReciprocal | Number | Tổng doanh số phát hành bảo lãnh 28009 |
| issueGuaranteePayment | Number | Tổng doanh số phát hành bảo lãnh 28040 |
| issueGuaranteeLoan | Number | Tổng doanh số phát hành bảo lãnh 28042 |
| debt | Number | Tổng dư nợ hiện tại |
| creditSale | Number | Tổng doanh số ghi có |
| creditSaleFT | Number | Tổng doanh số ghi có loại FT |
| creditSaleTT | Number | Tổng doanh số ghi có loại TT |
| creditSaleTF | Number | Tổng doanh số ghi có loại TF |
| minCreditSales | List<Object> | Danh sách doanh số theo tần suất kiểm tra |
| disbursementAmount | Number | Tổng số tiền giải ngân |
| configInputTOI | Object | Cấu hình TOI |
| revenueCondition | Object | Cấu hình DKNQ doanh thu |
| subRevenueConfigurations | List<Object> | Cấu hình doanh thu |
| periodRate | Object | Danh mục tỉ lệ theo kỳ DK doanh thu |

#### Tính toán và đánh giá theo các quy tắc

Sử dụng dữ liệu Mục 3.5.3.2.2 trên để tính toán

##### Đánh giá điều kiện TOI

* Đầu vào :

+ Dữ liệu cấu hình TOI (configInputTOI)

+ Dữ liệu DKNQ doanh thu (revenueCondition)

* Cách đánh giá :

Nếu revenueCondition.toi\_value <= configInputTOI.toi\_value :

Đánh dấu toiResult = FOLLOW ngược lại toiResult = NOT\_FOLLOW

##### Đánh giá điều kiện doanh thu tối thiểu

* Đầu vào :

+ Danh sách doanh số theo tần suất kiểm tra (minCreditSales)

+ Dữ liệu DKNQ doanh thu (revenueCondition)

* Cách đánh giá :

Nếu trong danh sách doanh số theo tần suất kiểm tra có tồn tại 1 bản ghi có doanh thu nhỏ hơn doanh thu tối thiếu cấu hình trong điều kiện nghị quyết ( revenueCondition.min\_approval\_value <= minCreditSales.get(i).value ) :

Đánh dấu minCreditSaleResult = NOT\_FOLLOW ngược lại minCreditSaleResult = FOLLOW

##### Đánh giá theo các quy tắc khác của DKNQ

* Đầu vào :

+ Dữ liệu cấu hình doanh thu (subRevenueConfigurations)

+ Danh mục tỉ lệ theo kỳ DK doanh thu (periodRate)

+ Dữ liệu DKNQ doanh thu (revenueCondition)

* Cách đánh giá :

Bước 1 : Chạy từng dòng subRevenueConfigurations để so sánh dữ liệu

+ Tính valueColumnC = Tổng các trưởng thông tị và điều kiện sau :

subRevenueConfigurations.other\_number

disbursedSales ( nếu subRevenueConfigurations.disburment\_revenue = true)

revenuePrincipalAndInterest ( nếu subRevenueConfigurations. original\_interest\_revenue = true)

lcSales ( nếu subRevenueConfigurations. lc\_revenue= true)

issueGuaranteeAdvance ( nếu subRevenueConfigurations. guarantee\_revenue = true và subRevenueConfigurations.advance)

issueGuaranteeContract ( nếu subRevenueConfigurations. guarantee\_revenue = true và subRevenueConfigurations. contract)

issueGuaranteeReciprocal ( nếu subRevenueConfigurations. guarantee\_revenue = true và subRevenueConfigurations. reciprocal)

issueGuaranteePayment ( nếu subRevenueConfigurations. guarantee\_revenue = true và subRevenueConfigurations. payment)

issueGuaranteeLoan ( nếu subRevenueConfigurations. guarantee\_revenue = true và subRevenueConfigurations. loan)

disbursementAmount ( nếu subRevenueConfigurations. disbursement\_money = true)

debt ( nếu subRevenueConfigurations. debt\_revenue = true)

+ Tính valueColumnD = Tổng các trưởng thông tị và điều kiện sau :

creditSaleFT ( nếu subRevenueConfigurations. credit\_revenue = true và subRevenueConfigurations. ft\_entry)

creditSaleTT ( nếu subRevenueConfigurations. credit\_revenue = true và subRevenueConfigurations. tt\_entry)

creditSaleTF ( nếu subRevenueConfigurations. credit\_revenue = true và subRevenueConfigurations. tf\_entry)

+ Tính amountTrans = subRevenueConfigurations. money\_on\_document

+ Tính amountFTTT = creditSaleFT + creditSaleTT +

creditSaleTF

Nếu xét các bản ghi trong danh sách subRevenueConfigurations điều kiện sau :

subRevenueConfigurations[i].approval\_rate is null OR ( subRevenueConfigurations[i].rule\_number = ‘RULE\_NUMBER\_3’ AND subRevenueConfigurations[i].money\_on\_document is null)

Đánh dấu trạng thái NOT\_FOLLOW ( gán subRevenueConfigurations[i].auto\_evaluation\_reusult = ‘NOT\_FOLLOW’ )

Nếu không rơi vào điều kiện trên tiếp tục xử lý :

amountBSCT = CASE WHEN subRevenueConfigurations[i].rule\_number = ‘RULE\_NUMBER\_1’ THEN subRevenueConfigurations[i].revenus\_money\_1 WHEN subRevenueConfigurations[i].rule\_number = ‘RULE\_NUMBER\_2’ THEN subRevenueConfigurations[i].revenus\_money\_2 ELSE null END

TH1 : subRevenueConfigurations[i].rule\_number = ‘RULE\_NUMBER\_2’ OR subRevenueConfigurations[i].rule\_number = ‘RULE\_NUMBER\_1’

totalValueLeft = valueColumnC\*( periodRate.rate/100)\*( subRevenueConfigurations[i].approval\_rate\*100)

subRevenueConfigurations[i].auto\_evaluation\_reusult = CASE WHEN ‘BSCT’ = TBL\_PERIOD\_EVALUATION.CONTROL\_METHOD AND amountBSCT is not null AND amountBSCT\*( periodRate.rate/100)\*( subRevenueConfigurations[i].approval\_rate\*100) <= valueColumnD THEN ‘FOLLOW’ WHEN ‘BSCT’ = TBL\_PERIOD\_EVALUATION.CONTROL\_METHOD AND amountBSCT is not null AND amountBSCT\*( periodRate.rate/100)\*( subRevenueConfigurations[i].approval\_rate\*100) > valueColumnD THEN ‘NOT\_FOLLOW’ WHEN ‘BSCT’ = TBL\_PERIOD\_EVALUATION.CONTROL\_METHOD AND amountBSCT is null THEN ‘NOT\_FOLLOW’ WHEN ‘BSCT’ <> TBL\_PERIOD\_EVALUATION.CONTROL\_METHOD AND totalValueLeft <= valueColumnD THEN ‘FOLLOW’ ELSE ‘NOT\_FOLLOW’ END

TH2 : subRevenueConfigurations[i].rule\_number <> ‘RULE\_NUMBER\_2’ AND subRevenueConfigurations[i].rule\_number <> ‘RULE\_NUMBER\_1’

totalValueLeft = amountTrans\*( periodRate.rate/100)\*( subRevenueConfigurations[i].approval\_rate\*100)

subRevenueConfigurations[i].auto\_evaluation\_reusult = CASE WHEN totalValueLeft <= amountFTTT THEN ‘FOLLOW’ ELSE ‘NOT\_FOLLOW’ END.

Bước 2 Đánh giá tổng thể trạng thái tuân thủ

totalFollow = tổng số bản ghi được đánh dấu subRevenueConfigurations. auto\_evaluation\_reusult = ‘FOLLOW’ được tính toán từ bước 2

revenueResult = CASE WHEN (revenueCondition.rule\_match = ‘MATCH\_ANY’ AND totalFollow > 0) OR (revenueCondition.rule\_match <> ‘MATCH\_ANY’ AND totalFollow = COUNT(subRevenueConfigurations) ) THEN ‘FOLLOW’ ELSE ‘NOT\_FOLLOW’ END

##### Đánh giá tổng hợp các điều kiện trên

1. Trường hợp cả 3 toiResult , minCreditSaleResult , revenueResult đều ko có dữ liệu ( NULL) và EVALUATION\_DATE < sysdate :

UPDATE TBL\_PERIOD\_EVALUATION

SET auto\_evaluation\_result =

CASE

WHEN evaluation\_result IS NULL THEN

'NOT\_FOLLOW' ELSE 'FOLLOW'

END,

evaluation\_result =

CASE

WHEN evaluation\_result IS NULL THEN

'NOT\_FOLLOW' ELSE evaluation\_result

END;

1. Trường hợp còn lại :

revenueResult is null 🡺 set revenueResult = ‘FOLLOW’

+ Nếu minCreditSaleResult = ‘NOT\_FOLLOW’ 🡺 Đánh dấu kết quả chấm tự động autoEvaluationResult = ‘NOT\_FOLLOW’.

+ Nếu toiResult is null 🡺 Đánh dấu kết quả chấm tự động autoEvaluationResult = revenueResult.

+ Nếu toiResult is not null AND revenueCondition.toi\_match = ‘MATCH\_ALL’ AND toiResult = ‘FOLLOW’ AND revenueResult = ‘FOLLOW’

🡺 Đánh dấu kết quả chấm tự động autoEvaluationResult = ‘FOLLOW’

+ Nếu toiResult is not null AND revenueCondition.toi\_match = ‘MATCH\_ALL’ AND (toiResult <> ‘FOLLOW’ OR revenueResult <> ‘FOLLOW’)

🡺 Đánh dấu kết quả chấm tự động autoEvaluationResult = ‘NOT\_FOLLOW’

+ Nếu toiResult is not null AND revenueCondition.toi\_match <> ‘MATCH\_ALL’ AND (toiResult = ‘FOLLOW’ OR revenueResult = ‘FOLLOW’)

🡺 Đánh dấu kết quả chấm tự động autoEvaluationResult = ‘ FOLLOW’

+ Nếu toiResult is not null AND revenueCondition.toi\_match = ‘MATCH\_ALL’ AND toiResult <> ‘FOLLOW’ AND revenueResult <> ‘FOLLOW’

🡺 Đánh dấu kết quả chấm tự động autoEvaluationResult = ‘NOT\_FOLLOW’

Thực hiện cập nhật vào DB với kỳ đang đánh giá

+ Cập nhật bảng kỳ đánh giá :

UPDATE TBL\_PERIOD\_EVALUATION

SET auto\_evaluation\_result =

:autoEvaluationResult,

evaluation\_result =

CASE

WHEN evaluation\_result IS NULL THEN

:autoEvaluationResult ELSE evaluation\_result

END;

Cập nhật bảng điều kiện nghị quyết Doanh thu

UPDATE tbl\_revenue\_condition

SET auto\_evaluation\_result =

: autoEvaluationResult;

Cập nhật bảng cấu hình doanh thu tbl\_revenue\_configuration ( Cập nhật tất cả các trạng thái auto\_evaluation\_result tương ứng trong danh sách subRevenueConfigurations được tính toán và gán lại từ bước 1 ).

### Mô tả bảng

#### Bảng TBL\_PERIOD\_EVALUATION tham khảo mục 3.4.4

#### Bảng TBL\_REVENUE\_CONFIGURATION

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Bắt buộc** | **Khóa** | **Mô tả** |
| 1 | ID | VARCHAR2(255 CHAR) | x | pk | ID |
| 2 | UPDATED\_USER | VARCHAR2(255 CHAR) |  |  | User cập nhật |
| 3 | UPDATED\_DATE | TIMESTAMP(6) |  |  | Thời gian cập nhật |
| 4 | TT\_ENTRY | NUMBER(1,0) |  |  | Bút toán TT (0/1) |
| 5 | TF\_ENTRY | NUMBER(1,0) |  |  | Bút toán TF (0/1) |
| 6 | RULE\_NUMBER | VARCHAR2(255 CHAR) | x |  | Quy tắc   * *RULE\_NUMBER\_1* * *RULE\_NUMBER\_2* * *RULE\_NUMBER\_3* |
| 7 | REVENUE\_CONDITION\_ID | VARCHAR2(255 CHAR) | x |  | ID bảng TBL\_REVENUE\_CONDITION |
| 8 | RECIPROCAL | NUMBER(1,0) |  |  | Tiêu chí đối ứng (0/1) |
| 9 | PAYMENT | NUMBER(1,0) |  |  | Tiêu chí thanh toán (0/1) |
| 10 | OTHER\_NUMBER | NUMBER(19,2) |  |  | Số khác |
| 11 | ORIGINAL\_INTEREST\_REVENUE | NUMBER(1,0) |  |  | Tiêu chí doanh thu gốc lãi (0/1) |
| 12 | MONEY\_ON\_DOCUMENT | NUMBER(19,2) |  |  | Số tiền trên chứng từ đơn vị kinh doanh bổ sung |
| 13 | LOAN | NUMBER(1,0) |  |  | Tiêu chí vay vốn (0/1) |
| 14 | LC\_REVENUE | NUMBER(1,0) |  |  | Tiêu chí doanh số LC (0/1) |
| 15 | GUARANTEE\_REVENUE | NUMBER(1,0) |  |  | Tiêu chí doanh số bảo lãnh |
| 16 | FT\_ENTRY | NUMBER(1,0) |  |  | Bút toàn FT (0/1) |
| 17 | DISBURMENT\_REVENUE | NUMBER(1,0) |  |  | Tiêu chí doanh số giải ngân (0/1) |
| 18 | DEBT\_REVENUE | NUMBER(1,0) |  |  | Tiêu chí dư nợ (0/1) |
| 19 | CREDIT\_REVENUE | NUMBER(1,0) |  |  | Tiêu chí doanh số ghi nợ (0/1) |
| 20 | CREATED\_USER | VARCHAR2(255 CHAR) |  |  | Người tạo |
| 21 | CREATED\_DATE | TIMESTAMP(6) |  |  | Thời gian tạo |
| 22 | CONTRACT | NUMBER(1,0) |  |  | Tiêu chí hợp đồng (0/1) |
| 23 | APPROVAL\_RATE | FLOAT |  |  | Tỉ lệ phê duyệt |
| 24 | ADVANCE | NUMBER(1,0) |  |  | Tiêu chí tạm ứng (0/1) |
| 25 | DOCUMENT\_ADDITIONAL | NUMBER(19,2) |  |  | Check box số tiền giải ngân |
| 26 | DISBURSEMENT\_MONEY | NUMBER(1,0) |  |  | Số tiền giải ngân (0/1) |
| 27 | AUTO\_EVALUATION\_RESULT | VARCHAR2(50 CHAR) |  |  | Kết quả đánh giá tự động   * *FOLLOW("Tuân thủ")* * *NOT\_FOLLOW("Không tuân thủ")* * *NOT\_ENOUGH\_EVALUATION\_DATA("Không đủ dữ liệu đánh giá")* |
| 28 | VERSION\_CONFIG\_ID | VARCHAR2(40 CHAR) |  |  | Trường VERSION\_CONFIG\_ID ở bảng **TBL\_REVENUE\_CONDITION** |

#### Bảng TBL\_REVENUE\_CONDITION

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Bắt buộc** | **Khóa chính** | **Mô tả** |
| 1 | ID | VARCHAR2(255 CHAR) | x | pk | Khóa bản ghi |
| 2 | STATUS | VARCHAR2(255 CHAR) | x |  | Trạng thái   * *SAVED "Đã lưu"* * *PENDING "Chờ duyệt"* * *APPROVED "Đã duyệt"* * *STOP\_FOLLOW "Ngừng theo dõi"* * *RETURN "Trả về"* * *REJECTED "Từ chối"* * *OVERDUE\_1 "Ngừng theo dõi" // QUÁ HẠN theo case 1* * *OVERDUE\_2 "Qua han 90 ngay" // QUÁ HẠN theo case 2* |
| 3 | SANCTION | VARCHAR2(4000 CHAR) |  |  | Chế tài |
| 4 | FREQUENCY\_UNIT | VARCHAR2(255 CHAR) |  |  | Đơn vị tần suất   * *D ngày* * *M tháng* * *LAST\_DAY ngày cuối cùng của tháng* |
| 5 | FREQUENCY | NUMBER(10,0) |  |  | Tần suất thực hiện |
| 6 | FIRST\_DATE\_CHOSEN\_PERIOD | TIMESTAMP(6) |  |  | Ngày bắt đầu chọn kỳ kiểm tra |
| 7 | DOCUMENT\_NAME | VARCHAR2(4000 CHAR) |  |  | Tên chứng từ yêu cầu |
| 8 | DESCRIPTION | VARCHAR2(4000 CHAR) |  |  | Mô tả |
| 9 | CUSTOMER\_ID | VARCHAR2(255 CHAR) |  |  | Mã khách hàng |
| 10 | CONTROL\_METHOD | VARCHAR2(255 CHAR) | x |  | Phương thức kiểm soát, link đến bảng TBL\_CONTROL\_METHOD.CODE |
| 11 | CONTRACT\_ID | VARCHAR2(255 CHAR) | x |  | Mã hợp đồng. Link đến bảng hợp đồng vay vốn STG\_SEAOPS.TBL\_BI\_CONTRACT\_REF\_DEFINE.REF\_ID/ID |
| 12 | CONTENT | VARCHAR2(4000 CHAR) |  |  | Nội dung |
| 13 | BUSINESS\_UNIT\_NOTE | NUMBER(1,0) | x |  | Lưu ý đơn vị kinh doanh. Default 0 |
| 14 | AUTO\_PERIOD | NUMBER(1,0) | x |  | Tự động sinh kỳ. Default 0 |
| 15 | TOI | NUMBER(1,0) |  |  | Cấu hình theo TOI hay không |
| 16 | TOI\_MATCH | VARCHAR2(255 CHAR) |  |  | Quy tắc so sánh   * *MATCH\_ALL* * *MATCH\_ANY* |
| 17 | RULE\_NUMBER | VARCHAR2(255 CHAR) |  |  | Quy tắc   * *RULE\_NUMBER\_1* * *RULE\_NUMBER\_2* * *RULE\_NUMBER\_3* |
| 18 | RULE\_MATCH | VARCHAR2(255 CHAR) |  |  | Quy tắc so sánh   * *MATCH\_ALL* * *MATCH\_ANY* |
| 19 | TOI\_VALUE | VARCHAR2(255 CHAR) |  |  | Giá trị TOI |
| 20 | TYPE\_PERIOD\_RATE | NUMBER(10,0) |  |  | Loại cấu hình tỉ lệ theo kỳ điều kiện doanh thu |
| 21 | MIN\_APPROVAL\_VALUE | NUMBER(19,0) |  |  | Giá trị tối thiểu phê duyệt |
| 22 | IS\_MIN\_APPROVAL\_VALUE | NUMBER(1,0) |  |  | Có phải là trạng thái giá trị tối thiểu phê duyệt (0/1) |
| 23 | VERSION\_CONFIG\_ID | VARCHAR2(40 CHAR) |  |  | Random UUID |
| 24 | START\_PERIOD | NUMBER(19,2) |  |  | Kỳ bắt đầu |
| 25 | SANCTION\_TYPE | VARCHAR2(255 CHAR) |  |  | Loại chế tài áp dụng   * *INCRE\_DECRE\_INTEREST\_RATE tăng giảm lãi suất* * *PENALTY\_FEE phí phạt* * *OTHER khác* |
| 26 | REASON | VARCHAR2(4000 CHAR) |  |  | Lý do từ chối |
| 27 | GROUP\_CONTRACT\_CONDITION | VARCHAR2(255 CHAR) | x |  | Nhóm điều kiện nghị quyết   * *CROSS\_SELLING // bán chéo* * *REVALUE\_COLLATERAL // Định giá lại TSĐB* * *CREDIT\_RELATION // Quan hệ tín dụng* * *PERIODIC // Định kỳ* * *MDSDV // MĐSDV* * *LIFE\_INSURANCE // bảo hiểm nhân thọ* * *FINANCIAL // tài chính* * *LAW // pháp lý* * *SECUR\_PROP\_INSURANCE // bảo hiểm TSĐB* * *OTHER// khác* * *REVENUE // Doanh thu* * *LOAN\_PRODUCT\_CONDITION // Điều kiện sản phẩm vay* |
| 28 | LOAN\_PRODUCT\_ID | VARCHAR2(255 CHAR) |  |  | Id sản phẩm vay, link đến bảng sản phẩm  TBL\_BI\_LOAN\_PRODUCT.PRODUCT\_CODE |
| 29 | CLONE\_ID | VARCHAR2(255 CHAR) |  |  | Id clone |
| 30 | CUSTOMER\_CLASS | VARCHAR2(255 CHAR) |  |  | Phân khúc |
| 31 | CO\_CODE | VARCHAR2(255 CHAR) |  |  | Mã CN/PGD |
| 32 | COMPANY\_NAME\_VN | VARCHAR2(255 CHAR) |  |  | Tên CN/PGD |
| 33 | NEW\_CUSTOMER | NUMBER(1,0) |  |  | Khách hàng mới (0/1) |
| 34 | AUTO\_EVALUATION\_RESULT | VARCHAR2(50 CHAR) |  |  | Kết quả đánh giá tự động   * *FOLLOW("Tuân thủ")* * *NOT\_FOLLOW("Không tuân thủ")* * *NOT\_ENOUGH\_EVALUATION\_DATA("Không đủ dữ liệu đánh giá")* |
| 35 | ORIGIN\_ID | VARCHAR2(255 CHAR) |  |  | Id bản ghi clone từ clone |
| 36 | CREATED\_USER | VARCHAR2(255 CHAR) |  |  | Username người tạo |
| 37 | CREATED\_DATE | TIMESTAMP(6) |  |  | Thời gian tạo |
| 38 | UPDATED\_USER | VARCHAR2(255 CHAR) |  |  | Username người cập nhật |
| 39 | UPDATED\_DATE | TIMESTAMP(6) |  |  | Thời gian cập nhật |

#### Bảng TBL\_REVENUE\_PERIOD\_RATE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Bắt buộc** | **Khóa** | **Mô tả** |
| 1 | ID | VARCHAR2(255 CHAR) | x | pk | Id bản ghi |
| 2 | TYPE | NUMBER(10,0) | x |  | Loại tỉ lệ |
| 3 | PERIOD | NUMBER(10,0) |  |  | Kỳ đánh giá |
| 4 | RATE | NUMBER(10,0) |  |  | Phần trăm |
| 5 | CREATED\_USER | VARCHAR2(255 CHAR) |  |  | Username người tạo |
| 6 | CREATED\_DATE | TIMESTAMP(6) |  |  | Thời gian tạo |
| 7 | UPDATED\_USER | VARCHAR2(255 CHAR) |  |  | Username người cập nhật |
| 8 | UPDATED\_DATE | TIMESTAMP(6) |  |  | Thời gian cập nhật |

#### Bảng TBL\_CONFIG\_INPUT\_TOI

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Bắt buộc** | **Khóa** | **Mô tả** |
| 1 | ID | VARCHAR2(255 CHAR) | x | pk | Id bản ghi |
| 2 | CUSTOMER\_ID | VARCHAR2(255 CHAR) | x |  | Mã khách hàng |
| 3 | CONTRACT\_ID | VARCHAR2(255 CHAR) | x |  | Mã hợp đồng, link đến bảng STG\_SEAOPS.TBL\_BI\_CONTRACT\_REF\_DEFINE.REF\_ID |
| 4 | TOI\_VALUE | VARCHAR2(255 CHAR) | x |  | Giá trị TOI theo kỳ (%) |
| 5 | PERIOD | NUMBER(10,0) |  |  | Kỳ |
| 6 | CREATED\_USER | VARCHAR2(255 CHAR) |  |  | Username người tạo |
| 7 | CREATED\_DATE | TIMESTAMP(6) |  |  | Thời gian tạo |
| 8 | UPDATED\_USER | VARCHAR2(255 CHAR) |  |  | Username người cập nhật |
| 9 | UPDATED\_DATE | TIMESTAMP(6) |  |  | Thời gian cập nhật |

#### Bảng TBL\_BI\_DISBURSEMENT\_HISTORY

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bảng | | **TBL\_BI\_DISBURSEMENT\_HISTORY** | | |
| Mô tả | | Bảng lưu dữ liệu lịch sử giải ngân | | |
| Index | | index DISBUR\_HIS\_CONTRACT\_REF on TBL\_BI\_DISBURSEMENT\_HISTORY (CONTRACT\_REF) | | |
| Chi tiết bảng dữ liệu | | | | |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Bắt buộc** | **Mô tả** |
| 1 | ID | VARCHAR(36) | PK | ID bản ghi, là chuỗi UUID |
| 2 | CONTRACT | VARCHAR(255) | Y | Mã khoản vay |
| 3 | CONTRACT\_REF | VARCHAR(255) | N | Mã REF |
| 4 | BALANCE | VARCHAR(255) | N | Số tiền giải ngân |
| 5 | VALUE\_DATE | VARCHAR(255) | N | Ngày giải ngân |
| 6 | MATURITY\_DATE | VARCHAR(255) | N | Ngày đáo hạn |
| 7 | LOANS\_TERM | VARCHAR(255) | Y | Kỳ hạn |
| 8 | LOANS\_TERM\_EXT | VARCHAR(255) | N | Kỳ hạn |
| 9 | SEAB\_LOS\_ID | VARCHAR(255) | Y | Mã LOS |
| 10 | CREATED\_USER | VARCHAR(255) | N | User tạo |
| 11 | CREATED\_DATE | TIMESTAMP(6) | N | Ngày đẩy dữ liệu |
| 12 | UPDATED\_USER | VARCHAR(255) | N | User cập nhật |
| 13 | UPDATED\_DATE | TIMESTAMP(6) | N | Ngày cập nhật |
| 14 | DATA\_STATUS | VARCHAR(50) | Y | Trạng thái dữ liệu |
| 15 | BI\_PARAM\_ID | VARCHAR(36) | N |  |
| 16 | EVALUATE\_DATE | TIMESTAMP(6) | N |  |
| 17 | EVALUATE\_STATUS | VARCHAR(255) | N |  |
| 18 | LOCKED\_KEY | VARCHAR(40) | N |  |
| 19 | LOCKED\_DATE | TIMESTAMP(6) | N |  |
| 20 | PRIORITY | NUMBER | Y, default 0 |  |
| 21 | CREATE\_PERIOD\_STATUS | VARCHAR(20) | Y |  |
| 22 | CREATED\_PERIODIC | NUMBER | Y, default 0 |  |
| 23 | NOTE | VARCHAR(1000) | N |  |

#### Bảng TBL\_BI\_CONTRACT\_DEBT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bảng | | **TBL\_BI\_CONTRACT\_DEBT** | | |
| Mô tả | | Bảng lưu dữ liệu dư nợ | | |
| Index | | index BI\_CONTRACT\_DEBT\_CONTRACT\_REF  on TBL\_BI\_CONTRACT\_DEBT (CONTRACT\_REF) | | |
| Chi tiết bảng dữ liệu | | | | |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Bắt buộc** | **Mô tả** |
| 1 | ID | VARCHAR(36) | PK | ID bản ghi, là chuỗi UUID |
| 2 | CONTRACT | VARCHAR(255) | Y | Mã LD |
| 3 | CONTRACT\_REF | VARCHAR(255) | Y | Mã Hợp đồng |
| 4 | DEBT | VARCHAR(255) | N | Gốc còn lại |
| 5 | INTEREST\_RATE | VARCHAR(255) |  | Lãi suất |
| 6 | NO\_DAYS\_OVERDUE | VARCHAR(20) | N | Số ngày quá hạn |
| 7 | CUSTOMER\_ID | VARCHAR(100) | Y | Mã KH |
| 8 | CUSTOMER\_CLASS | VARCHAR(20) | N | Phân khúc KH |
| 9 | DATE\_SYNC\_DATA | VARCHAR(20) | Y | Ngày đồng bộ (YYYY-MM-DD) |
| 10 | DATA\_STATUS | VARCHAR(50) | Y |  |
| 11 | CREATED\_USER | VARCHAR(255) | N |  |
| 12 | CREATED\_DATE | TIMESTAMP(6) | N |  |
| 13 | UPDATED\_USER | VARCHAR(255) | N |  |
| 14 | UPDATED\_DATE | TIMESTAMP(6) | N |  |
| 15 | BI\_PARAM\_ID | VARCHAR(36) | N |  |
| 16 | DATA\_SYNTHESIS\_BATCH\_ID | VARCHAR(255) | N |  |

#### Bảng TBL\_BI\_GUARANTEE\_SALES

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bảng | | **TBL\_BI\_GUARANTEE\_SALES** | | |
| Mô tả | | Bảng lưu dữ liệu doanh số PHBL | | |
| Index | | index BI\_GUARANTEE\_SALES\_CUSTOMER\_ID  on TBL\_BI\_GUARANTEE\_SALES (CUSTOMER\_ID) | | |
| Chi tiết bảng dữ liệu | | | | |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Bắt buộc** | **Mô tả** |
| 1 | ID | VARCHAR(36) | PK | ID bản ghi, là chuỗi UUID |
| 2 | CONTRACT\_MD | VARCHAR(255) | N | Số HĐ - Tài sản |
| 3 | PAST\_SCHED\_TYPE | VARCHAR(255) | N | Loại giao dịch |
| 4 | PAST\_SCHED\_NAME | VARCHAR(255) | N | Tên giao dịch |
| 5 | PAST\_SCHED\_DATE | VARCHAR(255) | N | Ngày phát sinh giao dịch |
| 6 | AMOUNT | VARCHAR(255) | N | Số tiền |
| 7 | CUSTOMER\_ID | VARCHAR(255) | N | Mã KH |
| 8 | CATEGORY | VARCHAR(255) | N | Mã nghiệp vụ |
| 9 | CATEGORY\_NAME\_VN | VARCHAR(255) | N | Loại nghiệp vụ |
| 10 | SHORT\_NAME | VARCHAR(255) | N | Tên KH |
| 11 | YEAR\_MONTH | TIMESTAMP(6) | N | Kỳ báo cáo (YYYYMM) |
| 12 | DATA\_STATUS | VARCHAR(50) | N | Trạng thái dữ liệu |
| 13 | CREATED\_USER | VARCHAR(255) | N | User cập nhật dữ liệu |
| 14 | CREATED\_DATE | TIMESTAMP(6) | N | Ngày đồng bộ dữ liệu |
| 15 | UPDATED\_USER | VARCHAR(255) | N | User cập nhật dữ liệu |
| 16 | UPDATED\_DATE | TIMESTAMP(6) | N | Ngày cập nhật dữ liệu |
| 17 | BI\_PARAM\_ID | VARCHAR(36) |  |  |

#### Bảng TBL\_BI\_LC\_RELEASE\_SALES

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bảng | | **TBL\_BI\_LC\_RELEASE\_SALES** | | |
| Mô tả | | Bảng lưu dữ liệu doanh số PH\_LC | | |
| Index | | index BI\_LC\_RELEASE\_SALES\_CONTRACT\_ID  on TBL\_BI\_LC\_RELEASE\_SALES (CONTRACT\_ID) | | |
| Chi tiết bảng dữ liệu | | | | |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Bắt buộc** | **Mô tả** |
| 1 | ID | VARCHAR(36) | PK | ID bản ghi, là chuỗi UUID |
| 2 | CUSTOMER | VARCHAR(255) | N | Mã KH |
| 3 | ISSUE\_DATE | VARCHAR(255) | N | Ngày phát sinh giao dịch |
| 4 | SHORT\_NAME | VARCHAR(255) | N | Tên KH |
| 5 | CONTRACT\_ID | VARCHAR(255) | N | Số HĐ |
| 6 | LC\_TYPE | VARCHAR(255) | N | Loại LC |
| 7 | LC\_AMOUNT | VARCHAR(255) | N | Số tiền |
| 8 | LC\_AMOUNT\_LCY | VARCHAR(255) | N | Số tiền quy đổi |
| 9 | CREATED\_USER | VARCHAR(255) | N | Ngày cập nhật dữ liệu |
| 10 | CREATED\_DATE | TIMESTAMP(6) | N | Ngày cập nhật dữ liệu |
| 11 | UPDATED\_USER | VARCHAR(255) | N | User cập nhật |
| 12 | UPDATED\_DATE | TIMESTAMP(6) | N | Ngày cập nhật dữ liệu |
| 13 | DATA\_STATUS | VARCHAR(50) | Y | Trạng thái dữ liệu |
| 14 | BI\_PARAM\_ID | VARCHAR(36) | N |  |

#### Bảng TBL\_BI\_DISBURSEMENT\_SALES

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bảng | | **TBL\_BI\_DISBURSEMENT\_SALES** | | |
| Mô tả | | Bảng lưu dữ liệu doanh số giải ngân | | |
| Index | | index BI\_DISBUR\_SALES\_CONTRACT\_REF  on TBL\_BI\_DISBURSEMENT\_SALES(CONTRACT\_REF) | | |
| Chi tiết bảng dữ liệu | | | | |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Bắt buộc** | **Mô tả** |
| 1 | ID | VARCHAR(36) | PK | ID bản ghi, là chuỗi UUID |
| 2 | DAY\_ID | VARCHAR(255) | N | Ngày báo cáo |
| 3 | CONTRACT | VARCHAR(255) | N | Mã HĐ |
| 4 | CUSTOMER\_CODE | VARCHAR(255) | N | Mã KH |
| 5 | MAIN\_CONTRACT | VARCHAR(255) | N | HĐ chính/ Số khế ước nhận nợ |
| 6 | CONTRACT\_REF | VARCHAR(255) | N | Số HĐ/ Số tham chiếu |
| 7 | VALUE\_DATE | VARCHAR(255) | N | Ngày phát sinh |
| 8 | LOC\_AMT | VARCHAR(255) | N | Số tiền được quy đổi về VNĐ |
| 9 | TRANS\_TYPE | VARCHAR(255) | N |  |
| 10 | TRANS\_NAME | VARCHAR(255) | N | Nội dung giao dịch |
| 11 | LOS\_ID | TIMESTAMP(6) | N | Mã LOS |
| 12 | DATE\_SYNC | VARCHAR(255) | N | Ngày đồng bộ (YYYY-MM-DD) |

#### Bảng TBL\_BI\_CREDIT\_SALES

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bảng | | **TBL\_BI\_CREDIT\_SALES** | | |
| Mô tả | | Bảng lưu dữ liệu doanh số ghi có | | |
| Index | | index BI\_CREDIT\_SALES\_CUSTOMER\_ID  on TBL\_BI\_CREDIT\_SALES(CUSTOMER\_ID) | | |
| Chi tiết bảng dữ liệu | | | | |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Bắt buộc** | **Mô tả** |
| 1 | ID | VARCHAR(36) | PK | ID bản ghi, là chuỗi UUID |
| 2 | CUSTOMER\_ID | VARCHAR(255) | Y | Mã KH |
| 3 | FULL\_NAME | VARCHAR(255) | N | Tên KH |
| 4 | AMOUNT | VARCHAR(255) | N | Tổng số tiền |
| 5 | AMOUNT\_LCY | VARCHAR(255) | N | Tổng số tiền quy đổi |
| 6 | CURRENCY | VARCHAR(255) | N | Loại tiền |
| 7 | OUT\_REFERENCE | VARCHAR(255) | Y | Tháng báo cáo |
| 8 | YEARMONTH | VARCHAR(255) | N | Số bút toán |
| 9 | DATA\_STATUS | VARCHAR(255) | Y | Trạng thái dữ liệu |
| 10 | BI\_PARAM\_ID | VARCHAR(36) | N |  |
| 11 | DATE\_SYNC\_DATA | VARCHAR(255) | N | Ngày đồng bộ (YYYY-MM-DD) |
| 12 | CREATED\_USER | VARCHAR(255) | N | User tạo |
| 13 | CREATED\_DATE | TIMESTAMP(6) | N | Ngày đẩy dữ liệu |
| 14 | UPDATED\_USER | VARCHAR(255) | N | User cập nhật |
| 15 | UPDATED\_DATE | TIMESTAMP(6) | N | Ngày cập nhật |

## Job tự động Đánh giá tự động ĐKSP, ĐKNQ tài chính, BHNT, định kỳ, tài sản bảo đảm, mục đích sử dụng vốn, pháp lý, khác

### Mô tả nghiệp vụ

Job nhằm mục đích update trạng thái ngừng theo dõi cho các loại DKNQ KSP, ĐKNQ tài chính, BHNT, định kỳ, tài sản bảo đảm, mục đích sử dụng vốn, pháp lý, khác khi đến ngày đánh giá mà vẫn chưa được CBNV đánh giá

### Biểu đồ tuần tự

N/A

### Quy tắc xử lý dữ liệu

#### Lấy dữ liệu thỏa mãn điều kiện

SELECT

\*

FROM

TBL\_PERIOD\_EVALUATION pe

LEFT JOIN TBL\_CONTRACT\_CONDITION cc ON cc.id = pe.CONTRACT\_CONDITION\_ID

LEFT JOIN TBL\_LOAN\_PRODUCT\_CONDITION lc ON lc.id = pe.CONTRACT\_CONDITION\_ID

WHERE

pe.EVALUATION\_DATE IS NOT NULL

AND ( pe.AUTO\_EVALUATION\_RESULT IS NULL OR ( pe.AUTO\_EVALUATION\_RESULT = 'FOLLOW' AND pe.GROUP\_CONTRACT\_CONDITION = 'SECUR\_PROP\_INSURANCE' ) )

AND pe.EVALUATION\_RESULT IS NULL

AND pe.GROUP\_CONTRACT\_CONDITION NOT IN ( 'CROSS\_SELLING', 'CREDIT\_RELATION', 'REVENUE' ) ( pe.STATUS IS NULL OR pe.STATUS <> 'APPROVED' )

AND (

pe.CONTRACT\_CONDITION\_ID IS NULL

OR ( pe.GROUP\_CONTRACT\_CONDITION <> 'LOAN\_PRODUCT\_CONDITION' AND cc.STATUS = 'APPROVED' )

OR ( pe.GROUP\_CONTRACT\_CONDITION = 'LOAN\_PRODUCT\_CONDITION' AND lc.STATUS = 'ACTIVE' )

)

AND (

( pe.REGISTER\_DATE IS NOT NULL AND TRUNC( pe.REGISTER\_DATE ) < TRUNC( SYSDATE ) )

OR ( pe.REGISTER\_DATE IS NULL AND TRUNC( pe.EVALUATION\_DATE ) < TRUNC( SYSDATE ) )

);

#### Xử lý dữ liệu

Trường hợp GROUP\_CONTRACT\_CONDITION = 'SECUR\_PROP\_INSURANCE'

UPDATE TBL\_PERIOD\_EVALUATION

SET AUTO\_EVALUATION\_RESULT = 'NOT\_FOLLOW',

EVALUATION\_RESULT= 'NOT\_FOLLOW',

report\_note =

CASE

WHEN 'STOP\_CHECK\_REPORT' = report\_batch\_id THEN

report\_note ELSE NULL

END,

report\_batch\_id =

CASE

WHEN report\_batch\_id IS NOT NULL

AND 'STOP\_CHECK\_REPORT' <> report\_batch\_id THEN

'RE\_CHECK'

WHEN 'STOP\_CHECK\_REPORT' = report\_batch\_id THEN

report\_batch\_id ELSE NULL END;

Các trường hợp còn lại update :

UPDATE TBL\_PERIOD\_EVALUATION

SET AUTO\_EVALUATION\_RESULT = 'NOT\_FOLLOW',

EVALUATION\_RESULT= 'NOT\_FOLLOW';

## Job đánh giá tự động ĐKNQ bán chéo

### Mô tả nghiệp vụ

Job tự động đánh giá Tuân thủ/ Không tuân thủ các kỳ theo dõi dự trên điều kiện nghị quyết bán chéo.

### Biểu đồ tuần tự

### Quy tắc xử lý dữ liệu

#### Lấy dữ liệu thỏa mãn điều kiện

SELECT

\*

FROM

TBL\_PERIOD\_EVALUATION pe

JOIN TBL\_CONTRACT\_CONDITION cc ON cc.id = pe.CONTRACT\_CONDITION\_ID

WHERE

pe.EVALUATION\_DATE IS NOT NULL

AND pe.AUTO\_EVALUATION\_RESULT IS NULL OR pe.AUTO\_EVALUATION\_RESULT = 'NOT\_ENOUGH\_EVALUATION\_DATA'

AND pe.GROUP\_CONTRACT\_CONDITION = 'CROSS\_SELLING'

AND ( pe.STATUS IS NULL OR.STATUS = 'APPROVED' )

AND (

( pe.REGISTER\_DATE IS NOT NULL AND TRUNC( pe.REGISTER\_DATE ) < TRUNC( SYSDATE ) )

OR ( pe.REGISTER\_DATE IS NULL AND TRUNC( pe.EVALUATION\_DATE ) < TRUNC( SYSDATE ) )

);

#### Xử lý dữ liệu

Dữ liệu lấy ra tại bước 3.7.3.1 sẽ đc xử lý từng row theo quy tắc sau :

Bước 1: Lấy cấu hình điều kiện nghị quyết tương ứng với hợp đồng.

crossSellings = SELECT \* FROM TBL\_CROSS\_SELLING WHERE CONDITION\_ID = :conditionId;

Bước 2: Truy vấn dữ liệu bán chéo của KH

+ PKG\_SEAOPS.FN\_BAN\_CHEO\_DVCT

Input :

- P\_CUSTOMER\_ID = customerId

- P\_FROM\_DATE = SYSDATE

- P\_TO\_DATE = SYSDATE

Output:

- P\_CT\_TN

- P\_CT\_QT

+ PKG\_SEAOPS.FN\_BAN\_CHEO\_HUYDONG

Input :

- P\_CUSTOMER\_ID = customerId

Output:

- P\_TERM

- P\_NON\_TERM

+ PKG\_SEAOPS.FN\_BAN\_CHEO\_THE

Input :

- P\_CUSTOMER\_ID = customerId

Output:

- result

- P\_CREDIT

- P\_DEBIT

- P\_S24

- P\_VISA

- P\_MASTER

+ PKG\_SEAOPS.FN\_BAN\_CHEO\_TK

Input :

- P\_CUSTOMER\_ID = customerId

Output:

- P\_TKTT

- P\_TKSD\_V3

- P\_TKSD

- P\_TKSD\_V4

+ PKG\_SEAOPS.FN\_BAN\_CHEO\_TTTM

Input :

- P\_CUSTOMER\_ID = customerId

Output:

- P\_LC

- P\_THUXEPVON

- P\_BAOLANH

+ PKG\_SEAOPS.FN\_DV\_NGAN\_HANG

Input :

- P\_CUSTOMER\_ID = customerId

Output:

- P\_SEANET

- P\_SMS

- P\_SEAPAY

Bước 3 : Đánh giá từ những dữ liệu trên

\*) Nếu tất cả các function trên đều ko trả ra dữ liệu =>

UPDATE TBL\_PERIOD\_EVALUATION SET AUTO\_EVALUATION\_RESULT = 'NOT\_ENOUGH\_EVALUATION\_DATA' WHERE id = :id;

Tính tổng số kết quả trả ra từ các function có giá trị = YES ( totalUser)

Chạy từng dòng dữ liệu cấu hình ĐKNQ bán chéo (crossSellings) để so sánh với kết quả bán chéo từ BI

+ TH1 : crossSellings.PRODUCT\_TYPE = 'QUANTITY'

totalUser >= crossSellings.NUMBER\_PRODUCT => gán crossSellings. AUTO\_EVALUATION\_RESULT = FOLLOW

totalUser < crossSellings.NUMBER\_PRODUCT => gán crossSellings. AUTO\_EVALUATION\_RESULT = NOT\_FOLLOW

+ TH2 : crossSellings.PRODUCT\_TYPE <> 'QUANTITY'

crossSellings.PRODUCT là tên trường kết quả bán chéo từ BI trả về nếu kết quả là YES => gán crossSellings. AUTO\_EVALUATION\_RESULT = FOLLOW; nếu kết quả <> YES => gán crossSellings. AUTO\_EVALUATION\_RESULT = NOT\_FOLLOW

( VD crossSellings.PRODUCT = P\_SEANET và kết quả trả về giá trị trường P\_SEANET từ function PKG\_SEAOPS.FN\_DV\_NGAN\_HANG = YES => gán crossSellings. AUTO\_EVALUATION\_RESULT = FOLLOW …)

Tính tổng số kết quả ko tuân thủ khi đánh giá tùng dòng điều kiện nghị quyết crossSellings (notFollowsItem)

IF notFollowsItem > 0 => autoEvaluationResult = FOLLOW ELSE autoEvaluationResult = NOT\_FOLLOW

Cập nhật dữ liệu theo kết quả theo autoEvaluationResult vừa đánh giá trên

UPDATE TBL\_PERIOD\_EVALUATION

SET auto\_evaluation\_result =

:autoEvaluationResult,

evaluation\_result =

CASE

WHEN evaluation\_result IS NULL THEN

:autoEvaluationResult ELSE evaluation\_result

END;

## Job sinh kỳ tự động tổng hợp

### Mô tả nghiệp vụ

Job tự động sinh kỳ cho các loại DKNQ khác với BHTSBD, DKSP, LD, TSKTDK

### Biểu đồ tuần tự

### Quy tắc xử lý dữ liệu

#### Lấy dữ liệu thỏa mãn điều kiện

SELECT

\*

FROM

TBL\_PERIOD\_EVALUATION

WHERE

PERIOD <> 0

AND GROUP\_CONTRACT\_CONDITION NOT IN ( 'REVALUE\_COLLATERAL', 'SECUR\_PROP\_INSURANCE' )

AND ( CONTRACT\_STATUS IS NULL OR CONTRACT\_STATUS NOT IN ( 'OVERDUE\_1', 'OVERDUE\_2' ) )

AND ( CONTRACT\_CONDITION\_STATUS IS NULL OR CONTRACT\_CONDITION\_STATUS <> 'STOP\_FOLLOW' )

AND LOCKED\_KEY IS NULL

AND CREATE\_NEXT\_PERIOD\_STATUS IN ( 'NEW', 'RENEW' )

AND ( FORCE\_CREATE\_NEXT = 1 OR CAST( EVALUATION\_DATE AS DATE ) < CAST( SYSDATE ) );

#### Xử lý dữ liệu

##### Sinh kỳ tự động không theo ngày kiểm tra chứng từ

Lọc các bản ghi có điều kiện : GROUP\_CONTRACT\_CONDITION IN ( 'REVENUE','LIFE\_INSURANCE','CROSS\_SELLING') OR (GROUP\_CONTRACT\_CONDITION IN ('OTHER','FINANCIAL','CREDIT\_RELATION') AND CONTROL\_METHOD = 'HT').

###### Trường hợp không chọn lưu ý đơn vị nghiệp vụ, không chọn tự động sinh kỳ

Điều kiện : (AUTO\_PERIOD IS NULL OR AUTO\_PERIOD <> 1) AND (BUSINESS\_UNIT\_NOTE IS NULL OR BUSINESS\_UNIT\_NOTE <> 1).

+ Nếu FREQUENCY = 0 🡺 UPDATE bản ghi hiện tại : Không thực hiện sinh kỳ mới

+ Nếu FREQUENCY <> 0 AND EVALUATION\_RESULT = FOLLOW thực hiện insert thêm 1 bản ghi kỳ mới theo quy tắc

Copy các trường thông tin từ bản ghi kỳ cũ sang kỳ mới và update lại các thông tin sau :

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên trường** | **Giá trị** |
| 1 | ID | RandomString |
| 2 | PERIOD | PERIOD +1 |
| 3 | CREATE\_NEXT\_PERIOD\_STATUS | 'NEW' |
| 4 | CREATED\_DATE | SYSDATE |
| 5 | CREATED\_USER | 'SYSTEM\_USER' |
| 6 | STATUS | NULL |
| 7 | EVALUATION\_DATE | evalDate = EVALUATION\_DATE ( dữ liệu bản ghi gốc) + FREQUENCY ( đon vị là MONTHS nếu FREQUENCY\_UNIT <> DAY , ngược lại đơn vị là DAYS)  + Nếu FREQUENCY\_UNIT = LAST\_DAY 🡺 Lấy giá trị bẳng ngày cuối tháng của evalDate ngược lại lấy nguyên giá trị evalDate |
| 8 | PRE\_EVALUATION\_DATE | EVALUATION\_DATE |
| 9 |  |  |

P/S các trường hợp trên đều update lại bản ghi gốc với dữ liệu sau :

SET CREATE\_NEXT\_PERIOD\_STATUS = SUCCESS.

###### Có chon tự động sinh kỳ

Điều kiện : AUTO\_PERIOD IS NOT NULL AND AUTO\_PERIOD = 1;

+ Nếu FREQUENCY = 0 🡺 UPDATE bản ghi hiện tại : Không thực hiện sinh kỳ mới

+ Nếu FREQUENCY <> 0 thực hiện insert thêm 1 bản ghi kỳ mới theo quy tắc

Copy các trường thông tin từ bản ghi kỳ cũ sang kỳ mới và update lại các thông tin sau :

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên trường** | **Giá trị** |
| 1 | ID | RandomString |
| 2 | PERIOD | PERIOD +1 |
| 3 | CREATE\_NEXT\_PERIOD\_STATUS | 'NEW' |
| 4 | CREATED\_DATE | SYSDATE |
| 5 | CREATED\_USER | 'SYSTEM\_USER' |
| 6 | STATUS | NULL |
| 7 | EVALUATION\_DATE | evalDate = EVALUATION\_DATE ( dữ liệu bản ghi gốc) + FREQUENCY ( đon vị là MONTHS nếu FREQUENCY\_UNIT <> DAY , ngược lại đơn vị là DAYS)  + Nếu FREQUENCY\_UNIT = LAST\_DAY 🡺 Lấy giá trị bẳng ngày cuối tháng của evalDate ngược lại lấy nguyên giá trị evalDate |
| 8 | PRE\_EVALUATION\_DATE | EVALUATION\_DATE |
| 9 |  |  |

P/S các trường hợp trên đều update lại bản ghi gốc với dữ liệu sau :

SET CREATE\_NEXT\_PERIOD\_STATUS = SUCCESS.

###### Có lưu ý đơn vị kinh doanh

Điều kiện : BUSINESS\_UNIT\_NOTE IS NULL AND BUSINESS\_UNIT\_NOTE = 1

+ Nếu EVALUATION\_RESULT = NOT\_FOLLOW AND STATUS = APPROVED 🡺 Không thực hiện sinh kỳ mới và update lại bản ghi gốc với dữ liệu sau :

SET CREATE\_NEXT\_PERIOD\_STATUS = SUCCESS.

+ Nếu (EVALUATION\_RESULT = NOT\_FOLLOW AND GROUP\_CONTRACT\_CONDITION IN ( 'REVENUE', CREDIT\_RELATION ') ) OR EVALUATION\_RESULT = FOLLOW thực hiện insert thêm 1 bản ghi kỳ mới theo quy tắc

Copy các trường thông tin từ bản ghi kỳ cũ sang kỳ mới và update lại các thông tin sau :

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên trường** | **Giá trị** |
| 1 | ID | RandomString |
| 2 | PERIOD | PERIOD +1 |
| 3 | CREATE\_NEXT\_PERIOD\_STATUS | 'NEW' |
| 4 | CREATED\_DATE | SYSDATE |
| 5 | CREATED\_USER | 'SYSTEM\_USER' |
| 6 | STATUS | NULL |
| 7 | EVALUATION\_DATE | evalDate = EVALUATION\_DATE ( dữ liệu bản ghi gốc) + FREQUENCY ( đon vị là MONTHS nếu FREQUENCY\_UNIT <> DAY , ngược lại đơn vị là DAYS)  + Nếu FREQUENCY\_UNIT = LAST\_DAY 🡺 Lấy giá trị bẳng ngày cuối tháng của evalDate ngược lại lấy nguyên giá trị evalDate |
| 8 | PRE\_EVALUATION\_DATE | NULL |
| 9 |  |  |

+ Nếu không rơi vào 2 trường hợp trên 🡪 update lại bản ghi gốc với dữ liệu sau :

SET CREATE\_NEXT\_PERIOD\_STATUS = RENEW

##### Sinh kỳ tự động dự vào ngày kiểm tra chứng từ

###### Trương hợp GROUP\_CONTRACT\_CONDITION = PERIODIC

Lấy dữ liệu tần suất kiểm traOM TBL\_PERIODIC\_TEST\_FREQUENCY WHERE CUSTOMER\_TYPE = NV

định kỳ : frequency = SELECT \* FROM TBL\_PERIODIC\_TEST\_FREQUENCY WHERE CUSTOMER\_TYPE = NVL(SUBSTR(PERIODIC\_TYPE, 0, INSTR(PERIODIC\_TYPE, '#')-1), PERIODIC\_TYPE)= CUSTOMER\_TYPE;

+ Nếu AUTO\_PERIOD = 1 OR BUSINESS\_UNIT\_NOTE = 1 OR frequency IS NULL 🡪 update lại bản ghi gốc với dữ liệu sau :

SET CREATE\_NEXT\_PERIOD\_STATUS = RENEW

+ Nếu : (PERIODIC\_TYPE LIKE '%MIDDLE\_LONG\_TERM' AND ( frequency .MIDDLE\_LONG\_TERM = 0 OR EVALUATION\_RESULT = NOT\_FOLLOW ) ) OR (PERIODIC\_TYPE LIKE '%SHORT\_TERM' AND ( frequency . SHORT\_TERM = 0 OR EVALUATION\_RESULT = NOT\_FOLLOW ) ) 🡪 Không thực hiện sinh kỳ mới và update lại bản ghi gốc với dữ liệu sau :

SET CREATE\_NEXT\_PERIOD\_STATUS = SUCCESS.

+ Các trường hợp còn lại thực hiện insert thêm 1 bản ghi kỳ mới theo quy tắc

Copy các trường thông tin từ bản ghi kỳ cũ sang kỳ mới và update lại các thông tin sau :

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên trường** | **Giá trị** |
| 1 | ID | RandomString |
| 2 | PERIOD | PERIOD +1 |
| 3 | CREATE\_NEXT\_PERIOD\_STATUS | 'NEW' |
| 4 | CREATED\_DATE | SYSDATE |
| 5 | CREATED\_USER | 'SYSTEM\_USER' |
| 6 | STATUS | NULL |
| 7 | EVALUATION\_DATE | evalDate = REPORT\_DATE ( dữ liệu bản ghi gốc) + FREQUENCY ( đon vị là MONTHS nếu FREQUENCY\_UNIT <> DAY , ngược lại đơn vị là DAYS)  + Nếu FREQUENCY\_UNIT = LAST\_DAY 🡺 Lấy giá trị bẳng ngày cuối tháng của evalDate ngược lại lấy nguyên giá trị evalDate |
| 8 | PRE\_EVALUATION\_DATE | REPORT \_DATE |
| 9 | FREQUENCY | CASE WHEN PERIODIC\_TYPE LIKE '%MIDDLE\_LONG\_TERM' THEN frequency .MIDDLE\_LONG\_TERM ELSE frequency . SHORT\_TERM END |
|  | FREQUENCY\_UNIT | 'M' |

update lại bản ghi gốc với dữ liệu sau :

SET CREATE\_NEXT\_PERIOD\_STATUS = SUCCESS.

###### Trường hợp GROUP\_CONTRACT\_CONDITION = LOAN\_PRODUCT\_CONDITION

Lấy dữ liệu điều hiện sản phẩm : loanCondition = SELECT \* FROM TBL\_LOAN\_PRODUCT\_CONDITION WHERE CODE = LOAN\_PRODUCT\_CODE; ( LOAN\_PRODUCT\_CODE lấy từ dữ liệu gốc)

+ Nếu AUTO\_PERIOD = 1 OR BUSINESS\_UNIT\_NOTE = 1 OR loanCondition IS NULL 🡪 update lại bản ghi gốc với dữ liệu sau :

SET CREATE\_NEXT\_PERIOD\_STATUS = RENEW

+ Nếu : loanCondition. FREQUENCY = 0 OR EVALUATION\_RESULT = NOT\_FOLLOW ) ) 🡪 Không thực hiện sinh kỳ mới và update lại bản ghi gốc với dữ liệu sau :

SET CREATE\_NEXT\_PERIOD\_STATUS = SUCCESS.

+ Các trường hợp còn lại thực hiện insert thêm 1 bản ghi kỳ mới theo quy tắc

Copy các trường thông tin từ bản ghi kỳ cũ sang kỳ mới và update lại các thông tin sau :

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên trường** | **Giá trị** |
| 1 | ID | RandomString |
| 2 | PERIOD | PERIOD +1 |
| 3 | CREATE\_NEXT\_PERIOD\_STATUS | 'NEW' |
| 4 | CREATED\_DATE | SYSDATE |
| 5 | CREATED\_USER | 'SYSTEM\_USER' |
| 6 | STATUS | NULL |
| 7 | EVALUATION\_DATE | evalDate = REPORT\_DATE ( dữ liệu bản ghi gốc) + FREQUENCY ( đon vị là MONTHS nếu FREQUENCY\_UNIT <> DAY , ngược lại đơn vị là DAYS)  + Nếu FREQUENCY\_UNIT = LAST\_DAY 🡺 Lấy giá trị bẳng ngày cuối tháng của evalDate ngược lại lấy nguyên giá trị evalDate |
| 8 | PRE\_EVALUATION\_DATE | REPORT \_DATE |
| 9 | FREQUENCY | loanCondition. FREQUENCY |
|  | FREQUENCY\_UNIT | loanCondition. FREQUENCY\_UNIT |

update lại bản ghi gốc với dữ liệu sau :

SET CREATE\_NEXT\_PERIOD\_STATUS = SUCCESS.

###### Trường hợp GROUP\_CONTRACT\_CONDITION = CREDIT\_RELATION

Trường hợp không chọn lưu ý đơn vị nghiệp vụ, không chọn tự động sinh kỳ

Điều kiện : (AUTO\_PERIOD IS NULL OR AUTO\_PERIOD <> 1) AND (BUSINESS\_UNIT\_NOTE IS NULL OR BUSINESS\_UNIT\_NOTE <> 1).

+ Nếu FREQUENCY = 0 🡺 UPDATE bản ghi hiện tại : Không thực hiện sinh kỳ mới và update lại bản ghi gốc với dữ liệu sau :

SET CREATE\_NEXT\_PERIOD\_STATUS = SUCCESS.

+ Nếu REPORT\_DATE is null OR STATUS <> APPROVED 🡪 update lại bản ghi gốc với dữ liệu sau :

SET CREATE\_NEXT\_PERIOD\_STATUS = RENEW

+ Nếu FREQUENCY <> 0 AND EVALUATION\_RESULT = FOLLOW AND REPORT\_DATE is not null OR STATUS = APPROVED thực hiện insert thêm 1 bản ghi kỳ mới theo quy tắc

Copy các trường thông tin từ bản ghi kỳ cũ sang kỳ mới và update lại các thông tin sau :

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên trường** | **Giá trị** |
| 1 | ID | RandomString |
| 2 | PERIOD | PERIOD +1 |
| 3 | CREATE\_NEXT\_PERIOD\_STATUS | 'NEW' |
| 4 | CREATED\_DATE | SYSDATE |
| 5 | CREATED\_USER | 'SYSTEM\_USER' |
| 6 | STATUS | NULL |
| 7 | EVALUATION\_DATE | evalDate = REPORT\_DATE ( dữ liệu bản ghi gốc) + FREQUENCY ( đon vị là MONTHS nếu FREQUENCY\_UNIT <> DAY , ngược lại đơn vị là DAYS)  + Nếu FREQUENCY\_UNIT = LAST\_DAY 🡺 Lấy giá trị bẳng ngày cuối tháng của evalDate ngược lại lấy nguyên giá trị evalDate |
| 8 | PRE\_EVALUATION\_DATE | REPORT \_DATE |
| 9 |  |  |

Update lại bản ghi gốc với dữ liệu sau :

SET CREATE\_NEXT\_PERIOD\_STATUS = SUCCESS.

Có chon tự động sinh kỳ

Điều kiện : AUTO\_PERIOD IS NOT NULL AND AUTO\_PERIOD = 1;

+ Nếu FREQUENCY = 0 🡺 UPDATE bản ghi hiện tại : Không thực hiện sinh kỳ mới và update lại bản ghi gốc với dữ liệu sau :

SET CREATE\_NEXT\_PERIOD\_STATUS = SUCCESS.

+ Nếu REPORT\_DATE is null OR STATUS <> APPROVED 🡪 update lại bản ghi gốc với dữ liệu sau :

SET CREATE\_NEXT\_PERIOD\_STATUS = RENEW

+ Nếu FREQUENCY <> 0 AND EVALUATION\_RESULT = FOLLOW AND REPORT\_DATE is not null OR STATUS = APPROVED thực hiện insert thêm 1 bản ghi kỳ mới theo quy tắc

Copy các trường thông tin từ bản ghi kỳ cũ sang kỳ mới và update lại các thông tin sau :

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên trường** | **Giá trị** |
| 1 | ID | RandomString |
| 2 | PERIOD | PERIOD +1 |
| 3 | CREATE\_NEXT\_PERIOD\_STATUS | 'NEW' |
| 4 | CREATED\_DATE | SYSDATE |
| 5 | CREATED\_USER | 'SYSTEM\_USER' |
| 6 | STATUS | NULL |
| 7 | EVALUATION\_DATE | evalDate = REPORT\_DATE ( dữ liệu bản ghi gốc) + FREQUENCY ( đon vị là MONTHS nếu FREQUENCY\_UNIT <> DAY , ngược lại đơn vị là DAYS)  + Nếu FREQUENCY\_UNIT = LAST\_DAY 🡺 Lấy giá trị bẳng ngày cuối tháng của evalDate ngược lại lấy nguyên giá trị evalDate |
| 8 | PRE\_EVALUATION\_DATE | REPORT \_DATE |
| 9 |  |  |

Update lại bản ghi gốc với dữ liệu sau :

SET CREATE\_NEXT\_PERIOD\_STATUS = SUCCESS.

Có lưu ý đơn vị kinh doanh

Điều kiện : BUSINESS\_UNIT\_NOTE IS NULL AND BUSINESS\_UNIT\_NOTE = 1

+ Nếu REPORT\_DATE is null OR STATUS <> APPROVED 🡪 update lại bản ghi gốc với dữ liệu sau :

SET CREATE\_NEXT\_PERIOD\_STATUS = RENEW.

+ Nếu (EVALUATION\_RESULT = NOT\_FOLLOW AND GROUP\_CONTRACT\_CONDITION = ' CREDIT\_RELATION ' ) OR EVALUATION\_RESULT = FOLLOW thực hiện insert thêm 1 bản ghi kỳ mới theo quy tắc

Copy các trường thông tin từ bản ghi kỳ cũ sang kỳ mới và update lại các thông tin sau :

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên trường** | **Giá trị** |
| 1 | ID | RandomString |
| 2 | PERIOD | PERIOD +1 |
| 3 | CREATE\_NEXT\_PERIOD\_STATUS | 'NEW' |
| 4 | CREATED\_DATE | SYSDATE |
| 5 | CREATED\_USER | 'SYSTEM\_USER' |
| 6 | STATUS | NULL |
| 7 | EVALUATION\_DATE | evalDate = REPORT\_DATE ( dữ liệu bản ghi gốc) + FREQUENCY ( đon vị là MONTHS nếu FREQUENCY\_UNIT <> DAY , ngược lại đơn vị là DAYS)  + Nếu FREQUENCY\_UNIT = LAST\_DAY 🡺 Lấy giá trị bẳng ngày cuối tháng của evalDate ngược lại lấy nguyên giá trị evalDate |
| 8 | PRE\_EVALUATION\_DATE | NULL |
| 9 |  |  |

+ Nếu không rơi vào 2 trường hợp trên 🡪 update lại bản ghi gốc với dữ liệu sau :

SET CREATE\_NEXT\_PERIOD\_STATUS = SUCCESS

###### Trường hợp GROUP\_CONTRACT\_CONDITION = MDSDV

Tương tự Sinh kỳ tự động không theo ngày kiểm tra chứng từ.

Khác cách tính

|  |  |
| --- | --- |
| EVALUATION\_DATE | evalDate = REPORT\_DATE ( ngày trên biên bản, dữ liệu bản ghi gốc) + FREQUENCY ( đon vị là MONTHS nếu FREQUENCY\_UNIT <> DAY , ngược lại đơn vị là DAYS)  + Nếu FREQUENCY\_UNIT = LAST\_DAY 🡺 Lấy giá trị bẳng ngày cuối tháng của evalDate ngược lại lấy nguyên giá trị evalDate |

## Ngừng theo dõi Bảo hiểm tài sản bảo đảm

### Mô tả nghiệp vụ

Tiến trình đánh dấu ngừng theo dõi bảo hiểm tài sản bảo đảm khi có sự thay đổi thông tin TSBD được đánh dấu lúc đồng bộ về từ BI

### Biểu đồ tuần tự

### Mô tả nghiệp vụ

Update DB theo câu SQL sau :

UPDATE TBL\_PERIOD\_EVALUATION

SET STATUS = 'STOP\_FOLLOW',

REPORT\_NOTE = 'Ngừng theo dõi TSBD'

WHERE

GROUP\_CONTRACT\_CONDITION IN ( 'REVALUE\_COLLATERAL', 'SECUR\_PROP\_INSURANCE' )

AND COLLATERAL\_ID IN ( SELECT DISTINCT COLLATERAL\_ID FROM TBL\_BI\_COLLATERAL WHERE STATE = 'STOP\_FOLLOW' );

Lưu ý : Job này thực hiện sau khi các job đồng bộ và merge TSBD chạy xong

## Sinh kỳ tự động do thay đổi thông tin TSBD

### Mô tả nghiệp vụ

### Biểu đồ tuần tự

### Quy tắc xử lý dữ liệu

#### Lấy dữ liệu thỏa mãn điều kiện

SELECT

\*

FROM

TBL\_BI\_COLLATERAL

WHERE

WHERE

TBL\_BI\_COLLATERAL.STATE = 'CHANGE'

AND TBL\_BI\_COLLATERAL.PERIOD\_STATUS IN ( 'NEW', 'RENEW' )

AND TBL\_BI\_COLLATERAL.DATA\_STATUS = 'ACTIVE'

AND TBL\_BI\_COLLATERAL.PERIOD\_CONDITION = 'ENOUGH';

#### Xử lý dữ liệu

\*) Dữ liệu được lấy phía trên sẽ được xử lý từng dòng theo quy tắc sau :

B1 : Lấy tất cả danh sách kỳ đã sinh trước đó :

oldPeriods = SELECT

\*

FROM

TBL\_PERIOD\_EVALUATION

WHERE

TBL\_BI\_COLLATERAL.COLLATERAL\_ID = TBL\_PERIOD\_EVALUATION.COLLATERAL\_ID

AND TBL\_PERIOD\_EVALUATION.GROUP\_CONTRACT\_CONDITION IN ( 'REVALUE\_COLLATERAL', 'SECUR\_PROP\_INSURANCE' );

B2 : Lấy danh mục tần suất định giá TSBD

Frequency = SELECT \* FROM tbl\_val\_collateral\_frequency WHERE collateral\_type = TBL\_BI\_COLLATERAL.COLL\_TYPE;

B3 : Lấy điều kiện nghị quyết

Conditions = SELECT

\*

FROM

tbl\_contract\_condition

WHERE

GROUP\_CONTRACT\_CONDITION = 'REVALUE\_COLLATERAL'

AND (STATUS IS NULL OR STATUS <>'STOP\_FOLLOW';

Và điều kiện contract\_id có tồn tại trong thông tin trường TBL\_BI\_COLLATERAL.contract\_ref ( Trường TBL\_BI\_COLLATERAL.contract\_ref lưu trữ thông tin nhiều contract\_ref khác nhau ngăn cách bằng ký tự space).

B4 : xử lý dữ liệu

##### Sinh kỳ TSBD

+ Nếu Frequency null 🡺 Update bản ghi TBL\_BI\_COLLATERAL :

UPDATE TBL\_BI\_COLLATERAL SET period\_status = 'SUCCESS' AND state = 'NOT\_CHANGE' AND updated\_user = 'SYSTEM\_USER' AND update\_date = SYSDATE;

+ Nếu Frequency not null

Lấy oldTsbd = kỳ đánh giá TSBD gần nhất ( EVALUTION\_DATE = MAX(EVALUTION\_DATE) AND GROUP\_CONTRACT\_CONDITION = 'REVALUE\_COLLATERAL' ) từ danh sách kỳ lấy tại bước 1.

\*) Xác định ngày đánh giá của kỳ mới và DKNQ tương ứng

startDate = CASE WHEN collateral.ver\_date is not null AND collateral.ver\_date < SYSDATE AND collateral.ver\_date > collateral.value\_date THEN

collateral.ver\_date ELSE collateral.value\_date END

min\_date = startDate + Frequency.FREQENCY ( đon vị là MONTHS nếu Frequency.FREQUENCY\_UNIT <> DAY , ngược lại đơn vị là DAYS)

+ Nếu FREQUENCY\_UNIT = LAST\_DAY 🡺 Lấy giá trị bẳng ngày cuối tháng của startDate ngược lại lấy nguyên giá trị startDate

For condition IN Conditions LOOP ( chạy từng dòng dữ liệu DKNQ để tìm ra DKNQ phù hơp) :

Tmp = startDate + condition.FREQENCY ( đon vị là MONTHS nếu condition.FREQUENCY\_UNIT <> DAY , ngược lại đơn vị là DAYS)

+ Nếu FREQUENCY\_UNIT = LAST\_DAY 🡺 Lấy giá trị bẳng ngày cuối tháng của startDate ngược lại lấy nguyên giá trị startDate

Nếu Tmp < min\_date gán lại min\_date = Tmp. Gán minCondition = condition

Khởi tạo bản ghi kỳ mới theo dữ liệu dưới bảng sau :

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên trường** | **Giá trị** |
| 1 | ID | RandomString |
| 2 | PERIOD | 1 |
| 3 | FREQUENCY | Frequency.FREQENCY |
| 4 | FREQUENCY\_UNIT | Frequency.FREQENCY\_UNIT |
| 5 | AUTO\_PERIOD | false |
| 6 | BUSINESS\_UNIT\_NOTE | false |
| 7 | COLLATERAL\_ID | Collateral. COLLATERAL\_ID |
| 8 | COLLATERAL\_TYPE | Collateral. COLLATERAL\_TYPE |
| 9 | CUSTOMER\_NAME | Collateral. CUSTOMER\_NAME |
| 10 | CUSTOMER\_ID | Collateral. CUSTOMER\_ID |
| 11 | CO\_CODE | Collateral. CO\_CODE |
| 12 | CONTRACT\_CONDITION\_ID | minCondition. CONTRACT\_CONDITION\_ID |
| 13 | CONTRACT\_CONDITION\_STATUS | minCondition. CONTRACT\_CONDITION\_STATUS |
| 14 | CONTRACT\_ID | minCondition. CONTRACT\_ID |
| 15 | CUSTOMER\_CLASS | minCondition. CUSTOMER\_CLASS |
| 16 | CONTROL\_METHOD | minCondition. CONTROL\_METHOD |
| 17 | CREATE\_NEXT\_PERIOD\_STATUS | 'NEW' |
| 18 | CREATED\_DATE | SYSDATE |
| 19 | CREATED\_USER | 'SYSTEM\_USER' |
| 20 | STATUS | NULL |
| 21 | EVALUATION\_DATE | Min\_date |
| 22 | AUTO\_EVALUATION\_RESULT | CASE WHEN Min\_date < SYSDATE THEN NOT\_FOLLOW ELSE null END |
| 23 | EVALUATION\_RESULT | CASE WHEN Min\_date < SYSDATE THEN NOT\_FOLLOW ELSE null END |
| 24 | GROUP\_CONTRACT\_CONDITION | REVALUE\_COLLATERAL |

##### Sinh kỳ bảo hiểm TSBD ( period\_condition\_ins = ENOUGH)

Lấy oldTsbd = kỳ đánh giá BHTSBD gần nhất ( EVALUTION\_DATE = MAX(EVALUTION\_DATE) AND GROUP\_CONTRACT\_CONDITION = ' SECUR\_PROP\_INSURANCE' ) từ danh sách kỳ lấy tại bước 1.

min\_date = CASE WHEN collateral. is\_expiry\_date is not null THEN TO\_DATE (collateral. is\_expiry\_date ) WHEN collateral. insur\_issue\_date is not null THEN TO\_DATE (collateral. insur\_issue\_date) ELSE TO\_DATE (collateral. value\_date) END

Khởi tạo bản ghi kỳ mới theo dữ liệu dưới bảng sau :

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên trường** | **Giá trị** |
| 1 | ID | RandomString |
| 2 | PERIOD | 1 |
| 3 | FREQUENCY | 0 |
| 4 | FREQUENCY\_UNIT | null |
| 5 | AUTO\_PERIOD | false |
| 6 | BUSINESS\_UNIT\_NOTE | false |
| 7 | COLLATERAL\_ID | Collateral. COLLATERAL\_ID |
| 8 | COLLATERAL\_TYPE | Collateral. COLLATERAL\_TYPE |
| 9 | CUSTOMER\_NAME | Collateral. CUSTOMER\_NAME |
| 10 | CUSTOMER\_ID | Collateral. CUSTOMER\_ID |
| 11 | CO\_CODE | Collateral. CO\_CODE |
| 12 | COLLATERAL\_CONTRACT\_REF | Collateral. COLLATERAL\_CONTRACT\_REF |
| 13 | DATE\_SYNC\_DATA | Collateral.DAYID |
| 14 | PRIORITY | 0 |
| 17 | CREATE\_NEXT\_PERIOD\_STATUS | 'NEW' |
| 18 | CREATED\_DATE | SYSDATE |
| 19 | CREATED\_USER | 'SYSTEM\_USER' |
| 20 | STATUS | NULL |
| 21 | EVALUATION\_DATE | Min\_date |
| 22 | AUTO\_EVALUATION\_RESULT | CASE WHEN Min\_date < SYSDATE THEN NOT\_FOLLOW ELSE null END |
| 23 | EVALUATION\_RESULT | CASE WHEN Min\_date < SYSDATE THEN NOT\_FOLLOW ELSE null END |
| 24 | GROUP\_CONTRACT\_CONDITION | SECUR\_PROP\_INSURANCE |

## Job sinh kỳ đầu tiên DKNQ sản phẩm vay

### Mô tả nghiệp vụ

Job tự động quét các hợp đồng vay chưa được sinh kỳ đầu tiên theo DKNQ sản phẩm vay để tạo kỳ theo dõi đầu tiên

### Biểu đồ tuần tự

### Quy tắc xử lý dữ liệu

#### Lấy dữ liệu thỏa mãn điều kiện

SELECT

\*

FROM

STG\_SEAOPS.TBL\_BI\_CONTRACT\_REF\_DEFINE

WHERE

DATA\_STATUS = 'ACTIVE'

AND (

STATUS IS NULL

OR STATUS NOT IN ( 'OVERDUE\_1', 'OVERDUE\_2' )

AND CPS\_DKSPV = 'N'

AND CREATE\_PERIOD\_STATUS IN ( 'NEW', 'RENEW' );

\*) *Khi dữ liệu không bị miss những ngày quá khứ có thể thêm điều kiện quét theo DAYID ( Job này sẽ chạy sau các job : Đồng bộ hợp đồng vay, Đồng bộ ngày giải ngân gần nhất, Đồng bộ dư nợ, Ngừng theo dõi hợp đồng )*

#### Xử lý dữ liệu

\*) *Xử lý từng row dữ liệu lấy đc trong danh sách trên (* contract *)*

B1 : Lấy danh mục DKNQ sản phẩm vay

loanConditions = SELECT

\*

FROM

TBL\_LOAN\_PRODUCT\_CONDITION

WHERE

PRODUCT\_TYPE = contract.LOAN\_PRODUCT\_ID

AND STATUS = 'ACTIVE';

Nếu loanConditions không có dữ liệu 🡺 bỏ qua ko xử lý.

Trường hợp có dữ liệu loanConditions ( dữ liệu này có thể có nhiều row) 🡪 Tạo kỳ theo từng row DKNQ sản phẩm ( loanCondition ):

+ Check hợp đồng đã được tạo kỳ bằng tay hay ko :

oldPeriod = SELECT

\*

FROM

TBL\_PERIOD\_EVALUATION

WHERE

CONTRACT\_ID = contract.REF\_ID

AND LOAN\_PRODUCT\_CODE = loanCondition.CODE;

oldPeriod is not null 🡪 UPDATE dữ liệu bảng hợp đồng vay :

UPDATE STG\_SEAOPS.TBL\_BI\_CONTRACT\_REF\_DEFINE

SET CPS\_DKSPV = 'Y'

AND CREATE\_PERIOD\_STATUS = 'RENEW'

AND UPDATED\_DATE = SYSDATE

AND UPDATED\_USER = 'SYSTEM\_USER';

oldPeriod is null 🡪 Tạo bản ghi kỳ theo dõi mới insert vào bảng TBL\_PERIOD\_EVALUATION

theo quy tắc sau :

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên trường** | **Giá trị** |
| 1 | ID | RandomString |
| 2 | PERIOD | 1 |
| 3 | FREQUENCY | loanCondition.FREQENCY |
| 4 | FREQUENCY\_UNIT | loanCondition.FREQENCY\_UNIT |
| 5 | AUTO\_PERIOD | false |
| 6 | BUSINESS\_UNIT\_NOTE | false |
|  | CONTRACT\_ID | contract.REF\_ID |
|  | CONTRACT\_STATUS | contract.STATUS |
|  | CUSTOMER\_ID | contract.CUSTOMER\_ID |
| 15 | CUSTOMER\_CLASS | contract. CUSTOMER\_CLASS |
| 11 | CO\_CODE | contract.CO\_CODE |
|  | COMPANY\_NAME\_VN | contract. COMPANY\_NAME\_VN |
|  | SHORT\_NAME | contract. SHORT\_NAME |
|  | LOAN\_PRODUCT\_ID | contract. LOAN\_PRODUCT\_ID |
| 12 | LOAN\_PRODUCT\_ID | loanCondition. CODE |
| 17 | CREATE\_NEXT\_PERIOD\_STATUS | 'NEW' |
| 18 | CREATED\_DATE | SYSDATE |
| 19 | CREATED\_USER | 'SYSTEM\_USER' |
| 21 | EVALUATION\_DATE | evalDate = contract.VALUE\_DATE\_MIN ( dữ liệu bản ghi gốc) + loanCondition.FREQUENCY ( đon vị là MONTHS nếu loanCondition.FREQUENCY\_UNIT <> DAY , ngược lại đơn vị là DAYS)  + Nếu loanCondition.FREQUENCY\_UNIT = LAST\_DAY 🡺 Lấy giá trị bẳng ngày cuối tháng của evalDate ngược lại lấy nguyên giá trị evalDate |
| 22 | AUTO\_EVALUATION\_RESULT | CASE WHEN evalDate < SYSDATE THEN NOT\_FOLLOW ELSE null END |
| 23 | EVALUATION\_RESULT | CASE WHEN evalDate < SYSDATE THEN NOT\_FOLLOW ELSE null END |
| 24 | GROUP\_CONTRACT\_CONDITION | LOAN\_PRODUCT\_CONDITION |

Sau khi sinh kỳ UPDATE dữ liệu bảng hợp đồng vay :

UPDATE STG\_SEAOPS.TBL\_BI\_CONTRACT\_REF\_DEFINE

SET CPS\_DKSPV = 'Y'

AND CREATE\_PERIOD\_STATUS = 'RENEW'

AND UPDATED\_DATE = SYSDATE

AND UPDATED\_USER = 'SYSTEM\_USER';

## Job sinh kỳ dựa theo MDSDV

### Mô tả nghiệp vụ

Job tự động sinh kỳ dựa theo dữ liệu lịch sử giản ngân và điều kiện nghị quyết MDSDV hoặc danh mục MDSDV nếu chưa khai báo DKNQ.

### Biểu đồ tuần tự

### Quy tắc xử lý dữ liệu

#### Lấy dữ liệu thỏa mãn điều kiện

~~SELECT~~

~~\*~~

~~FROM~~

~~STG\_SEAOPS.TBL\_BI\_DISBURSEMENT\_HISTORY~~

~~JOIN STG\_SEAOPS.TBL\_BI\_CONTRACT\_REF\_DEFINE ON STG\_SEAOPS.TBL\_BI\_CONTRACT\_REF\_DEFINE.REF\_ID = STG\_SEAOPS.TBL\_BI\_DISBURSEMENT\_HISTORY.CONTRACT\_REF~~

~~WHERE~~

~~DATA\_STATUS = 'ACTIVE'~~

~~and CREATE\_PERIOD\_STATUS in ('NEW', 'RENEW')~~

**SELECT**

*DH*.CONTRACT\_REF *ref\_id*, *DH*.LOAN\_CODE *loan\_code*, *CRD*.VALUE\_DATE\_MIN *value\_date\_min\_ref*, *DH*.VALUE\_DATE *value\_date*, *DH*.PRODUCT\_CODE *product\_code*, *DH*. MATURITY\_DATE *maturity\_date*

**FROM**

TBL\_BI\_DISBURSEMENT\_HISTORY *DH*

**JOIN** TBL\_BI\_CONTRACT\_REF\_DEFINE *CRD*

**ON** *CRD*.REF\_ID = *DH*.CONTRACT\_REF

**WHERE**

*DH*.DATA\_STATUS = **'ACTIVE'**

**and** *DH*.CREATE\_PERIOD\_STATUS in (‘NEW’,’RENEW’)

**UNION** **ALL**

**SELECT**

*DH*.CONTRACT\_REF *ref\_id*, *dh*.contract\_md *loan\_code*, *CRD*.VALUE\_DATE\_MIN *value\_date\_min\_ref*, *dh*.past\_sched\_date *value\_date*, *DH*.PRODUCT\_CODE *product\_code*, *DH*. MATURITY\_DATE *maturity\_date*

**FROM**

TBL\_BI\_GUARANTEE\_SALES *DH*

**JOIN** TBL\_BI\_CONTRACT\_REF\_DEFINE *CRD*

**ON** *CRD*.REF\_ID = *DH*.CONTRACT\_REF

**WHERE**

*DH*.DATA\_STATUS = **'ACTIVE'**

**and** *DH*.CREATE\_PERIOD\_STATUS in (‘NEW’,’RENEW’)

**UNION** **ALL**

**SELECT**

*DH*.CONTRACT\_REF *ref\_id*, *dh*.contract\_id *loan\_code*, *CRD*.VALUE\_DATE\_MIN *value\_date\_min\_ref*, *dh*.issue\_date *value\_date*, *DH*.PRODUCT\_CODE *product\_code*, *DH*. MATURITY\_DATE *maturity\_date*

**FROM**

TBL\_BI\_LC\_RELEASE\_SALES *DH*

**JOIN** TBL\_BI\_CONTRACT\_REF\_DEFINE\_BIPLUS *CRD*

**ON** *CRD*.REF\_ID = *DH*.CONTRACT\_REF

**WHERE**

*DH*.DATA\_STATUS = **'ACTIVE'**

**and** *DH*.CREATE\_PERIOD\_STATUS in (‘NEW’,’RENEW’);

#### Xử lý dữ liệu

Dữ liệu lấy ta từ bước trên được xử lý theo từng Row (contract) theo quy tắc sau :

B1 : Lấy dữ liệu tương ứng làm điều kiện đánh giá

Với từng khoản giải ngân, lấy ĐKNQ, ĐKSP MĐSDV của hợp đồng cùng product\_code với khoản giải ngân, Danh mục MĐSDV

Ưu tiên lấy thông tin từ ĐKNQ để sinh hạng mục.

Kiểm tra đã có hạng mục MĐSDV gắn với id của khoản giải ngân chưa, nếu có rồi thì update lại thông tin ngày đến hạn kiểm tra, tần suất. Nếu chưa có thì insert mới.

* Dữ liệu điều kiện nghị quyết MDSDV đã duyệt tương ứng với hợp đồng:

condition = SELECT

\*

FROM

TBL\_CONTRACT\_CONDITION

WHERE

(GROUP\_CONDITION\_TYPE is null or GROUP\_CONDITION\_TYPE = DKNQ)

AND GROUP\_CONTRACT\_CONDITION = 'MDSDV'

AND STATUS = 'APPROVED'

AND CONTRACT\_ID = : contract\_id; -- contract\_id là CONTRACT\_REF lấy được từ dữ liệu gốc

* Dữ liệu điều kiện sản phẩm MDSDV đã duyệt tương ứng với hợp đồng:

condition = SELECT

\*

FROM

TBL\_CONTRACT\_CONDITION

WHERE

(GROUP\_CONDITION\_TYPE = DKSP)

AND LOAN\_PRODUCT\_ID = :product\_code – product\_code từ khoản giải ngân lấy từ 3 bảng

AND GROUP\_CONTRACT\_CONDITION = 'MDSDV'

AND STATUS = 'APPROVED'

AND CONTRACT\_ID = : contract\_id; -- contract\_id là CONTRACT\_REF lấy được từ dữ liệu gốc

* Dữ liệu danh mục MDSDV :

useCapital= SELECT

\*

FROM

tbl\_frck\_purpose\_use\_capital

WHERE

PRODUCT\_TYPE = : product\_type -- product\_type là LOAN\_PRODUCT\_ID lấy được từ dữ liệu gốc

B2 : Tạo dữ liệu kỳ đánh giá

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên trường** | **Giá trị** |
| 1 | ID | RandomString |
| 2 | PERIOD | CASE WHEN condition.START\_PERIOD is not null THEN condition.START\_PERIOD ELSE 1 END |
| 3 | FREQUENCY | CASE WHEN condition.FREQENCY is not null THEN condition.FREQENCY ELSE 0 END |
| 4 | FREQUENCY\_UNIT | condition.FREQENCY\_UNIT |
| 5 | AUTO\_PERIOD | CASE WHEN condition. AUTO\_PERIOD is not null THEN condition. AUTO\_PERIOD ELSE false END |
| 6 | BUSINESS\_UNIT\_NOTE | CASE WHEN condition. BUSINESS\_UNIT\_NOTE is not null THEN condition. BUSINESS\_UNIT\_NOTE ELSE false END |
| 7 | CONTENT | condition.CONTENT |
| 8 | CONTROL\_MENTHOD | condition. CONTROL\_MENTHOD |
| 9 | CONTRACT\_CONDITION\_ID | condition. CONTRACT\_CONDITION\_ID |
| 10 | CONTRACT\_CONDITION\_STATUS | condition. CONTRACT\_CONDITION\_STATUS |
| 11 | FIRST\_DATE\_CHOSEN\_PERIOD | condition. FIRST\_DATE\_CHOSEN\_PERIOD |
| 12 | DESCRIPTION | condition. DESCRIPTION |
| 13 | CONTRACT\_ID | contract.REF\_ID |
| 14 | CONTRACT\_STATUS | contract.STATUS |
| 15 | CUSTOMER\_ID | contract.CUSTOMER\_ID |
| 16 | CUSTOMER\_CLASS | contract. CUSTOMER\_CLASS |
| 17 | CO\_CODE | contract.CO\_CODE |
| 18 | COMPANY\_NAME\_VN | contract. COMPANY\_NAME\_VN |
| 19 | SHORT\_NAME | contract. SHORT\_NAME |
| 20 | LOAN\_CODE | contract. CONTRACT |
| 21 | REVOLVING | contract. REVOLVING |
| 22 | VALUE\_DATE\_MIN | contract.VALUE\_DATE |
| 23 | LOAN\_PRODUCT\_ID | contract. LOAN\_PRODUCT\_ID |
| 24 |  |  |
| 25 |  |  |
| 26 | CREATE\_NEXT\_PERIOD\_STATUS | 'NEW' |
| 27 | CREATED\_DATE | SYSDATE |
| 28 | CREATED\_USER | 'SYSTEM\_USER' |
| 29 | EVALUATION\_DATE | evalDate = contract.VALUE\_DATE( dữ liệu bản ghi gốc) + condition.CHECK\_TERM (DAYS)  + Nếu condition.CHECK\_TERM is null 🡺 evalDate = contract.VALUE\_DATE( dữ liệu bản ghi gốc) + useCapital.DEADLINE (DAYS) |
| 30 | AUTO\_EVALUATION\_RESULT | CASE WHEN evalDate < SYSDATE THEN NOT\_FOLLOW ELSE null END |
| 31 | EVALUATION\_RESULT | CASE WHEN evalDate < SYSDATE THEN NOT\_FOLLOW ELSE null END |
| 32 | GROUP\_CONTRACT\_CONDITION | MDSDV |
| 33 | GROUP\_CONDITION\_TYPE | DKNQ : Nếu condition có GROUP\_CONDITION\_TYPE = null hoặc DKNQ  DKSP: Nếu condition có GROUP\_CONDITION\_TYPE = DKSP  QDC: Nếu condition null |

~~Lưu dữ liệu vừa khởi tạo vào bảng TBL\_PERIOD\_EVALUATION đồng thời update bảng STG\_SEAOPS.TBL\_BI\_DISBURSEMENT\_HISTORY SET CREATE\_PERIOD\_STATUS = ‘SUCCESS’~~

Update 3 bảng STG\_SEAOPS.TBL\_BI\_DISBURSEMENT\_HISTORY; STG\_SEAOPS.TBL\_BI\_LC\_RELEASE\_SALES, STG\_SEAOPS.TBL\_BI\_GUARANTEE\_SALES với CREATE\_PERIOD\_STATUS = ‘SUCCESS’

## Job sinh kỳ dự theo tần suất kiểm tra định kỳ

### Mô tả nghiệp vụ

Job tự động tạo định kỳ theo dõi theo khách hàng và chi nhánh. Khi có phát sinh hợp đồng vay mới sẽ chạy thông tin để tạo kỳ đánh giá

### Biểu đồ tuần tự

### Quy tắc xử lý dữ liệu

#### Lấy dữ liệu thỏa mãn điều kiện

SELECT

CUSTOMER\_ID, CO\_CODE

from

STG\_SEAOPS.TBL\_BI\_CONTRACT\_REF\_DEFINE

where

DATA\_STATUS = 'ACTIVE'

and (

STATUS is null

or STATUS not in ('OVERDUE\_1', 'OVERDUE\_2')

)

and CPS\_TSKTDK = 'N'

and LOCKED\_KEY is null

and CREATE\_PERIOD\_STATUS in ('NEW', 'RENEW') GROUP BY CUSTOMER\_ID, CO\_CODE;

#### Xử lý dữ liệu

Dữ liệu được xử lý theo từng row ( lấy customer\_id và co\_code làm key lấy ra nhưng hợp đồng, nghị quyệt, danh mục tương ứng)

B1 : Lấy chi tiết dữ liệu hợp đồng liên quan

\*) Lấy dữ liệu hợp đồng tương ứng

listContract = SELECT \* FROM STG\_SEAOPS.TBL\_BI\_CONTRACT\_REF\_DEFINE WHERE CUSTOMER\_ID = :customerId AND CO\_CODE = :coCode;

\*) Lấy dữ liệu kỳ cũ gần nhất

oldPeriod = SELECT

\*

FROM

TBL\_PERIOD\_EVALUATION

WHERE

GROUP\_CONTRACT\_CONDITION = 'PERIODIC'

AND CUSTOMER\_ID = : customerId

AND CO\_CODE = : coCode

AND CAREATE\_DATE = (

SELECT

MAX(CAREATE\_DATE)

FROM

TBL\_PERIOD\_EVALUATION

WHERE

GROUP\_CONTRACT\_CONDITION = 'PERIODIC'

AND CUSTOMER\_ID = : customerId

AND CO\_CODE = : coCode

);

B2 : Tính ngày đánh giá nhỏ nhất của các hợp đồng

\*) Lấy dữ liệu kỳ cũ đã đc duyệt gần nhất

periodApproved = SELECT

\*

FROM

TBL\_PERIOD\_EVALUATION

WHERE

GROUP\_CONTRACT\_CONDITION = 'PERIODIC'

AND CUSTOMER\_ID = : customerId

AND CO\_CODE = : coCode

AND STATUS = 'APPROVED'

AND PERRIOD = (

SELECT

MAX(PERRIOD)

FROM

TBL\_PERIOD\_EVALUATION

WHERE

GROUP\_CONTRACT\_CONDITION = 'PERIODIC'

AND CUSTOMER\_ID = : customerId

AND CO\_CODE = : coCode

AND STATUS = 'APPROVED'

);

* Đặt các biến min như sau :

minContract object

minCondition object

minDate dateTime

minFreUnit String

minFre Interger

periodicType String

* Gán giá trin ban đầu

|  |  |  |
| --- | --- | --- |
| STT | Tên biến | Giá trị |
| 1 | minContract | Null |
| 2 | minCondition | Null |
| 3 | minDate | + Nếu oldPeriod <> null : oldPeriod.EVALUATION\_DATE  + Nếu oldPeriod = null && periodApproved <> null : periodApproved.REPORT\_DATE + periodApproved.FREQUENCY ( đon vị là MONTHS nếu periodApproved.FREQUENCY\_UNIT <> DAY , ngược lại đơn vị là DAYS)  + Còn lại lấy giá trin NULL |
| 4 | minFreUnit | CASE WHEN oldPeriod is not null THEN oldPeriod. PREQUENCY\_UNIT WHEN  periodApproved is not null THEN periodApproved.PREQUENCY\_UNIT ELSE NULL END |
| 5 | minFre | CASE WHEN oldPeriod is not null THEN oldPeriod. PREQUENCY WHEN  periodApproved is not null THEN periodApproved.PREQUENCY ELSE NULL END |
| 6 | periodicType | CASE WHEN oldPeriod is not null THEN oldPeriod. PERIODIC\_TYPE WHEN  periodApproved is not null THEN periodApproved. PERIODIC\_TYPE ELSE NULL END |

\*) Chạy từng row dữ liệu hợp đồng (listContract ) để tính toán lại ngày nhỏ nhất

contract = listContract.get(i);

* Lấy điều kiện nghị quyết của hợp đồng

condition = SELECT \* FROM TBL\_CONTRACT\_CONDITION WHERE GROUP\_CONTRACT\_CONDITION = 'PERIODIC' and STATUS = 'APPROVED' and CONTRACT\_ID = :refId; ( refId = contract.REF\_ID)

* Lấy danh mục tần suât kiểm tra định kỳ tương ứng

cusType = CASE WHEN 'KHCN' = contract.CUSTOMER\_CLASS THEN 'PERSONAL' ELSE 'ENTERPRISE' END

frequency = SELECT

\*

FROM

tbl\_periodic\_test\_frequency

WHERE

CUSTOMER\_TYPE = : custType

Frequency is null => bỏ qua ko xử lý

* Lấy dữ liệu loại kỳ hạn từ lịch sử giải ngân

terms = SELECT

loans\_term\_ext

FROM

STG\_SEAOPS.TBL\_BI\_DISBURSEMENT\_HISTORY

WHERE

CONTRACT\_REF = : refID

AND loans\_term\_ext is not null

terms is null => bỏ qua ko xử lý

Nếu terms có tồn tại giá trị SHORT\_TERM =>

tmpTerm = SHORT\_TERM

val ( biến trung gian ) = frequency.SHORT\_TERM

Nếu terms không tồn tại giá trị SHORT\_TERM =>

tmpTerm = MIDDLE\_LONG\_TERM

val = frequency.MIDDLE\_LONG\_TERM

periodicType is null OR val < minFre 🡺 periodicType = cusType + '#' + tmpTerm (1)

Tính ngày đánh theo danh mục TSKTDK :

tmp ( biến trung gian ) = contract.VALUE\_DATE\_MIN ( dữ liệu bản ghi gốc) + val ( đon vị là MONTHS )

* Nếu minDate is null hoặc tmp < minDate gán lại các giá trị min khởi tạo ban đầu :

|  |  |  |
| --- | --- | --- |
| STT | Tên biến | Giá trị |
| 1 | minContract | contract |
| 2 | minCondition | NULL |
| 3 | minDate | tmp |
| 4 | minFreUnit | 'M' |
| 5 | minFre | val |
| 6 | periodicType | Lấy dữ liệu mục đánh dấu (1) |

Tính ngày đánh giá theo ĐKNQ :

condition <> null thực hiện tính ngày theo condition

tmp ( biến trung gian ) = contract.VALUE\_DATE\_MIN ( dữ liệu bản ghi gốc) + condition.FREQUENCY ( đon vị là MONTHS nếu condition.FREQUENCY\_UNIT <> DAY , ngược lại đơn vị là DAYS)

+ Nếu condition.FREQUENCY\_UNIT = LAST\_DAY 🡺 Lấy giá trị bẳng ngày cuối tháng của tmp ngược lại lấy nguyên giá trị tmp

* Nếu minDate is null hoặc tmp < minDate gán lại các giá trị min khởi tạo ban đầu :

|  |  |  |
| --- | --- | --- |
| STT | Tên biến | Giá trị |
| 1 | minContract | contract |
| 2 | minCondition | condition |
| 3 | minDate | tmp |
| 4 | minFreUnit | condition. FREQUENCY\_UNIT |
| 5 | minFre | condition. FREQUENCY |
| 6 | periodicType | Lấy dữ liệu mục đánh dấu (1) |

B3 : Set giá trị cho kỳ mới ( bảng TBL\_PERIOD\_EVALUATION )

TH1 oldPeriod <> null :

Giữ nguyên các giá trị ban đầu và set lại các giá trị sau :

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên trường** | **Giá trị** |
| 3 | FREQUENCY | minFre |
| 4 | FREQUENCY\_UNIT | minFreUnit |
|  | EVALUATION\_DATE | minDate |
|  | PERIODIC\_TYPE | periodicType |
| 8 | CONTROL\_MENTHOD | minCondition. CONTROL\_MENTHOD |
| 9 | CONTRACT\_CONDITION\_ID | minCondition. CONTRACT\_CONDITION\_ID |
| 10 | CONTRACT\_CONDITION\_STATUS | minCondition. CONTRACT\_CONDITION\_STATUS |
| 13 | CONTRACT\_ID | minContract .REF\_ID |
| 14 | CONTRACT\_STATUS | minContract .STATUS |
| 23 | LOAN\_PRODUCT\_ID | minContract . LOAN\_PRODUCT\_ID |

TH1 oldPeriod = null 🡺 Tạo kỳ mới :

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên trường** | **Giá trị** |
| 1 | ID | RandomString |
| 2 | PERIOD | 1 |
| 3 | FREQUENCY | minFre |
| 4 | FREQUENCY\_UNIT | minFreUnit |
| 5 | AUTO\_PERIOD | false |
| 6 | BUSINESS\_UNIT\_NOTE | false |
| 7 | CONTROL\_MENTHOD | minCondition. CONTROL\_MENTHOD |
| 8 | CONTRACT\_CONDITION\_ID | minCondition. CONTRACT\_CONDITION\_ID |
| 9 | CONTRACT\_CONDITION\_STATUS | minCondition. CONTRACT\_CONDITION\_STATUS |
| 10 | CONTRACT\_ID | minContract.REF\_ID |
| 11 | CONTRACT\_STATUS | minContract.STATUS |
| 12 | CUSTOMER\_ID | minContract.CUSTOMER\_ID |
| 13 | CUSTOMER\_CLASS | minContract. CUSTOMER\_CLASS |
| 14 | CO\_CODE | minContract.CO\_CODE |
| 15 | COMPANY\_NAME\_VN | minContract. COMPANY\_NAME\_VN |
| 16 | SHORT\_NAME | minContract. SHORT\_NAME |
| 17 | VALUE\_DATE\_MIN | minContract.VALUE\_DATE |
| 18 | LOAN\_PRODUCT\_ID | minContract. LOAN\_PRODUCT\_ID |
| 19 | PERIODIC\_TYPE | periodicType |
| 20 | CREATE\_NEXT\_PERIOD\_STATUS | 'NEW' |
| 21 | CREATED\_DATE | SYSDATE |
| 22 | CREATED\_USER | 'SYSTEM\_USER' |
| 23 | EVALUATION\_DATE | minDate |
| 24 | AUTO\_EVALUATION\_RESULT | CASE WHEN minDate < SYSDATE THEN NOT\_FOLLOW ELSE null END |
| 25 | EVALUATION\_RESULT | CASE WHEN minDate < SYSDATE THEN NOT\_FOLLOW ELSE null END |
| 26 | GROUP\_CONTRACT\_CONDITION | PERIODIC |

* INSERT hoặc UPDATE dữ liệu 2 trường hợp trên vào bảng TBL\_PERIOD\_EVALUATION

Đồng thời update bảng STG\_SEAOPS.TBL\_BI\_CONTRACT\_REF\_DEFINE :

UPDATE

STG\_SEAOPS.TBL\_BI\_CONTRACT\_REF\_DEFINE

SET

cps\_tsktdk = 'Y' WHRER CUSTOMER\_ID = : customerId

AND CO\_CODE = : coCode;

### Quy tắc xử lý dữ liệu mới

#### Lấy dữ liệu thỏa mãn điều kiện

SELECT

CUSTOMER\_ID, CO\_CODE

from

STG\_SEAOPS.TBL\_BI\_CONTRACT\_REF\_DEFINE

where

DATA\_STATUS = 'ACTIVE'

and (

STATUS is null

or STATUS not in ('OVERDUE\_1', 'OVERDUE\_2')

)

and CPS\_TSKTDK = 'N'

and LOCKED\_KEY is null

and CREATE\_PERIOD\_STATUS in ('NEW', 'RENEW') GROUP BY CUSTOMER\_ID, CO\_CODE

#### Xử lý dữ liệu

Dữ liệu được xử lý theo từng row ( lấy customer\_id và co\_code làm key lấy ra nhưng hợp đồng, nghị quyệt, danh mục tương ứng)

B1 : Lấy chi tiết dữ liệu hợp đồng liên quan

\*) Lấy dữ liệu hợp đồng tương ứng

listContract = SELECT \* FROM STG\_SEAOPS.TBL\_BI\_CONTRACT\_REF\_DEFINE WHERE CUSTOMER\_ID = :customerId AND CO\_CODE = :coCode;

\*) Lấy dữ liệu kỳ cũ gần nhất

oldPeriod = SELECT

\*

FROM

TBL\_PERIOD\_EVALUATION

WHERE

GROUP\_CONTRACT\_CONDITION = 'PERIODIC'

AND CUSTOMER\_ID = : customerId

AND CO\_CODE = : coCode

AND CAREATE\_DATE = (

SELECT

MAX(CAREATE\_DATE)

FROM

TBL\_PERIOD\_EVALUATION

WHERE

GROUP\_CONTRACT\_CONDITION = 'PERIODIC'

AND CUSTOMER\_ID = : customerId

AND CO\_CODE = : coCode

);

\*) Lấy dữ liệu kỳ cũ đã đc duyệt gần nhất

periodApproved = SELECT

\*

FROM

TBL\_PERIOD\_EVALUATION

WHERE

GROUP\_CONTRACT\_CONDITION = 'PERIODIC'

AND CUSTOMER\_ID = : customerId

AND CO\_CODE = : coCode

AND STATUS = 'APPROVED'

AND PERRIOD = (

SELECT

MAX(PERRIOD)

FROM

TBL\_PERIOD\_EVALUATION

WHERE

GROUP\_CONTRACT\_CONDITION = 'PERIODIC'

AND CUSTOMER\_ID = : customerId

AND CO\_CODE = : coCode

AND STATUS = 'APPROVED'

);

B2 : Tính ngày đánh giá nhỏ nhất của các hợp đồng

Xử lý từng hợp đồng, tính tần suất nhỏ nhất của hợp đồng đó:

- Kiểm tra xem hợp đồng đó có ĐKNQ Định kỳ không:

condition = SELECT \* FROM TBL\_CONTRACT\_CONDITION WHERE GROUP\_CONTRACT\_CONDITION = 'PERIODIC' and STATUS = 'APPROVED' and CONTRACT\_CONDITION\_TYPE = (NULL or DKNQ) and CONTRACT\_ID = :refId; ( refId = contract.REF\_ID)

- Nếu không có ĐKNQ Định kỳ thì kiểm tra ĐKSP

condition = SELECT \* FROM TBL\_CONTRACT\_CONDITION WHERE GROUP\_CONTRACT\_CONDITION = 'PERIODIC' and STATUS = 'APPROVED' and CONTRACT\_CONDITION\_TYPE = DKSP and CONTRACT\_ID = :refId; ( refId = contract.REF\_ID)

- Nếu không có ĐKNQ, ĐKSP, thì kiểm tra Danh mục tần suất định kỳ

cusType = CASE WHEN 'KHCN' = contract.CUSTOMER\_CLASS THEN 'PERSONAL' ELSE 'ENTERPRISE' END

frequency = SELECT

\*

FROM

tbl\_periodic\_test\_frequency

WHERE

CUSTOMER\_TYPE = : custType

Lấy trong bảng lịch sử giải ngân, nếu tồn tại ít nhất 1 bản ghi có kỳ là SHORT\_TERM thì lấy tần suất SHORT\_TERM trong danh mục

So sánh giữa các hợp đồng:

- Nếu periodApproved = null , sinh kỳ 1, lấy ngày giải ngân đầu tiên của từng hợp đồng + tần suất nhỏ nhất của hợp đồng đó ra ngày đến hạn đánh giá, lấy ngày nhỏ nhất làm EVALUATION\_DATE

- Nếu periodApproved ! = null , sinh kỳ 1, lấy periodApproved.REPORT\_DATE + tần suất nhỏ nhất của hợp đồng đó ra ngày đến hạn đánh giá, lấy ngày nhỏ nhất làm EVALUATION\_DATE

minFre: tần suất nhỏ nhất

minFreUnit: đơn vị của tuần suất nhỏ nhất

minDate: ngày đến hạn đánh giá nhỏ nhất

minCondition: điều kiện sinh ra tần suất nhỏ nhất (có trường hợp danh mục sinh ra tần suất nhỏ nhất)

minContract: hợp đồng sinh ra ngày đến hạn nhỏ nhất

B3 : Set giá trị cho kỳ mới ( bảng TBL\_PERIOD\_EVALUATION )

TH1 oldPeriod <> null :

Giữ nguyên các giá trị ban đầu và set lại các giá trị sau :

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên trường** | **Giá trị** |
| 1 | FREQUENCY | minFre |
| 2 | FREQUENCY\_UNIT | minFreUnit |
| 3 | EVALUATION\_DATE | minDate |
| 4 | PERIODIC\_TYPE | periodicType |
| 5 | CONTROL\_MENTHOD | minCondition. CONTROL\_MENTHOD |
| 6 | CONTRACT\_CONDITION\_ID | minCondition. CONTRACT\_CONDITION\_ID |
| 7 | CONTRACT\_CONDITION\_STATUS | minCondition. CONTRACT\_CONDITION\_STATUS |
| 8 | CONTRACT\_ID | minContract .REF\_ID |
| 9 | CONTRACT\_STATUS | minContract .STATUS |
| 10 | LOAN\_PRODUCT\_ID | minContract . LOAN\_PRODUCT\_ID |

TH1 oldPeriod = null 🡺 Tạo kỳ mới :

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên trường** | **Giá trị** |
| 1 | ID | RandomString |
| 2 | PERIOD | 1 |
| 3 | FREQUENCY | minFre |
| 4 | FREQUENCY\_UNIT | minFreUnit |
| 5 | AUTO\_PERIOD | false |
| 6 | BUSINESS\_UNIT\_NOTE | false |
| 7 | CONTROL\_MENTHOD | minCondition. CONTROL\_MENTHOD |
| 8 | CONTRACT\_CONDITION\_ID | minCondition. CONTRACT\_CONDITION\_ID |
| 9 | CONTRACT\_CONDITION\_STATUS | minCondition. CONTRACT\_CONDITION\_STATUS |
| 10 | CONTRACT\_ID | minContract.REF\_ID |
| 11 | CONTRACT\_STATUS | minContract.STATUS |
| 12 | CUSTOMER\_ID | minContract.CUSTOMER\_ID |
| 13 | CUSTOMER\_CLASS | minContract. CUSTOMER\_CLASS |
| 14 | CO\_CODE | minContract.CO\_CODE |
| 15 | COMPANY\_NAME\_VN | minContract. COMPANY\_NAME\_VN |
| 16 | SHORT\_NAME | minContract. SHORT\_NAME |
| 17 | VALUE\_DATE\_MIN | minContract.VALUE\_DATE |
| 18 | LOAN\_PRODUCT\_ID | minContract. LOAN\_PRODUCT\_ID |
| 19 | PERIODIC\_TYPE | periodicType |
| 20 | CREATE\_NEXT\_PERIOD\_STATUS | 'NEW' |
| 21 | CREATED\_DATE | SYSDATE |
| 22 | CREATED\_USER | 'SYSTEM\_USER' |
| 23 | EVALUATION\_DATE | minDate |
| 24 | AUTO\_EVALUATION\_RESULT | CASE WHEN minDate < SYSDATE THEN NOT\_FOLLOW ELSE null END |
| 25 | EVALUATION\_RESULT | CASE WHEN minDate < SYSDATE THEN NOT\_FOLLOW ELSE null END |
| 26 | GROUP\_CONTRACT\_CONDITION | PERIODIC |

* INSERT hoặc UPDATE dữ liệu 2 trường hợp trên vào bảng TBL\_PERIOD\_EVALUATION

Đồng thời update bảng STG\_SEAOPS.TBL\_BI\_CONTRACT\_REF\_DEFINE :

UPDATE

STG\_SEAOPS.TBL\_BI\_CONTRACT\_REF\_DEFINE

SET

cps\_tsktdk = 'Y' WHRER CUSTOMER\_ID = : customerId

AND CO\_CODE = : coCode

## Áp chế tài ĐKNQ

### Mô tả nghiệp vụ

Tiến trình áp chế tài xử lý những kỳ theo dõi vi phạm điều kiện nghị quyết theo cấu hình chế tài.

### Biểu đồ tuần tự

### Quy tắc xử lý dữ liệu

#### Lấy dữ liệu thỏa mãn điều kiện

SELECT

pe.\*,

CASE WHEN pe.group\_contract\_condition = 'LOAN\_PRODUCT\_CONDITION' THEN lp.sanction WHEN pe.group\_contract\_condition = 'REVENUE' THEN rc.sanction ELSE cd.sanction END as sanction,

CASE WHEN pe.group\_contract\_condition = 'LOAN\_PRODUCT\_CONDITION' THEN 'OTHER' WHEN pe.group\_contract\_condition = 'REVENUE' THEN rc.sanctionType ELSE cd.sanctionType END as sanctionType,

CASE WHEN pe.group\_contract\_condition = 'LOAN\_PRODUCT\_CONDITION' THEN pe.LOAN\_PRODUCT\_CODE || pe.CONTRACT\_ID WHEN pe.group\_contract\_condition = 'REVENUE'

AND rc.origin\_id is not null THEN rc.origin\_id WHEN pe.group\_contract\_condition = 'REVENUE'

AND rc.origin\_id is null THEN rc.ID WHEN cd.origin\_id is not null THEN cd.origin\_id ELSE cd.ID END as conditionId

from

TBL\_PERIOD\_EVALUATION pe

left join TBL\_LOAN\_PRODUCT\_CONDITION lp on lp.CODE = pe.LOAN\_PRODUCT\_CODE

AND pe.group\_contract\_condition = 'LOAN\_PRODUCT\_CONDITION'

left join TBL\_REVENUE\_CONDITION rc on rc.ID = pe.contract\_condition\_id

AND pe.group\_contract\_condition = 'REVENUE'

left join TBL\_CONTRACT\_CONDITION cd on cd.ID = pe.contract\_condition\_id

AND pe.group\_contract\_condition not in (

'REVENUE', 'LOAN\_PRODUCT\_CONDITION'

)

where

pe.GROUP\_CONTRACT\_CONDITION <> 'SECUR\_PROP\_INSURANCE'

and pe.EVALUATION\_RESULT is not null

and (

(

pe.CONTRACT\_CONDITION\_ID is not null

and pe.GROUP\_CONTRACT\_CONDITION <> 'LOAN\_PRODUCT\_CONDITION'

)

or (

lp.SANCTION is not null

and pe.GROUP\_CONTRACT\_CONDITION = 'LOAN\_PRODUCT\_CONDITION'

)

)

and (

pe.CONTRACT\_CONDITION\_STATUS is null

or pe.CONTRACT\_CONDITION\_STATUS not in ('STOP\_FOLLOW', 'OVERDUE\_1')

)

and (

pe.STATUS is null

or pe.STATUS not in ('STOP\_FOLLOW', 'OVERDUE\_1')

)

and pe.EVALUATION\_DATE < to\_date(sysdate, 'DD-MM-YYYY');

#### Xử lý dữ liệu

Dữ liệu xử lý theo từng row lấy ra được ( period )

Check nếu period.saction is null 🡺 UPDATE TBL\_PERIOD\_EVALUATION Set period.report\_note = 'Điều kiện nghị quyết không có áp chế tài' WHERE ID= period.ID; Đồng thời ko xử lý bước tiếp theo.

applySanctionReports : Biến lưu list dữ liệu áp chế tài ( kiểu dữ liệu lấy theo bảng tbl\_apply\_sanction\_report ). Mỗi khi xử lý 1 row dữ sẽ add them dữ liệu vào List này để tạo mới dữ liệu chế tài.

B1 : Tạo các biến điều kiện

* Tìm chế tài kỳ trước ( previous )

checkPrevious = SELECT

\*

FROM

tbl\_apply\_sanction\_report

WHERE

condition\_id = peroid.condition\_id

AND status\_report = 'ACTIVE'

ORDER BY

create\_user

limit

1

checkExits = SELECT

\*

FROM

applySanctionReports

WHERE

condition\_id = peroid.condition\_id

AND status\_report = 'ACTIVE'

ORDER BY

create\_user

limit

1

+ Nếu checkExits not null null 🡺 previous = checkExits

+ Ngược lại previous = checkPrevious

+ Nếu period. group\_contract\_condition = 'LOAN\_PRODUCT\_CONDITION AND previous is not null AND previous. Frequency = 0 🡺 previous = null

* Xác định loại chế tài vi phạm ( sactionViolationType )

+ Nếu period.sanctionType = 'INCRE\_DECRE\_INTEREST\_RATE' AND period. evaluation\_result = 'FOLLOW' 🡺 sactionViolationType = 'DECRE\_CONDITION' ( Giảm do khắc phục DKNQ )

+ Nếu period.sanctionType = 'INCRE\_DECRE\_INTEREST\_RATE' AND period. evaluation\_result = 'NOT\_FOLLOW' 🡺 sactionViolationType = INCRE\_CONDITION' ( Tăng do khắc phục DKNQ )

+ Nếu period.sanctionType = 'PENALTY\_FEE' AND period. evaluation\_result = 'NOT\_FOLLOW' 🡺 sactionViolationType = 'PENALTY\_FEE' ( Áp phí phạt do vi phạm DKNQ )

+ Nếu period.sanctionType = 'OTHER' AND period. evaluation\_result = 'FOLLOW' 🡺 sactionViolationType = 'OTHER' ( Áp chế tài khác do vi phạm DKNQ )

+ Trường hợp còn lại 🡺 sactionViolationType = null

* Xác định thông tin hợp đồng vay ( contractRefDefine)

contractRefDefine = SELECT

\*

FROM

stg\_seaops.tbl\_bi\_contract\_ref\_define

WHERE

ref\_id = period.contract\_id

* Xác định thông tin dư nợ hợp đồng ( lstLd)

lstLd = SELECT

\*

FROM

stg\_seaops.tbl\_bi\_contract\_debt

WHERE

contract\_ref = period.contract\_id

AND DEBT > 0

AND (

period.group\_contract\_condition <> 'MDSDV'

OR upper(CONTRACT) = upper (period.loan\_code)

AND DAYID = (

SELECT

MAX(DAYID)

FROM

stg\_seaops.tbl\_bi\_contract\_debt

WHERE

contract\_ref = period.contract\_id

AND DEBT > 0

)

* Xác định đơn vị ( adUnit )

adUnit = SELECT

\*

FROM

KSSCT.ad\_unit

WHERE

unit\_code = period.CO\_CODE;

* Xác định rule tính tháng áp chế tài

rule= SELECT

\*

FROM

tbl\_apply\_sanction\_report

WHERE

condition\_id = peroid.condition\_id AND rating\_status = peroid.evaluation\_result

AND status\_report = 'ACTIVE'

limit

1

B2 : Xác định trạng thái chế tài và khởi tạo object lưu chế tài ( processStatus và applySanctionReport )

+ Nếu sactionViolationType is null 🡺 processStatus = "Không áp dụng do không có loại chế tài vi phạm thoải mãn"

+ Nếu sactionViolationType NOT IN ('OTHER','PENALTY\_FEE') AND previous is not null AND previous.evaluation\_result = period.evaluation\_result 🡺 processStatus = "Đã áp dụng cho kỳ trước"

+ Nếu sactionViolationType NOT IN ('OTHER','PENALTY\_FEE') AND previous is null AND 'FOLLOW' = period.evaluation\_result 🡺 processStatus = "Đã áp dụng cho kỳ trước"

+ Nếu contractRefDefine is null 🡺 "Không tồn tại mã REF ở hợp đồng giải ngân"

+ lstLD is null 🡺 "Không truy vấn được thông tin LD theo REF"

Nếu không vi phạm các trường hợp trên xử lý FOR LOOP dữ liệu lstLD (objectLD) khởi tạo đối tượng applySanctionReport gán vào danh sách applySanctionReports và gán lại processStatus theo bảng maping sau:

applySanctionReport :

|  |  |  |
| --- | --- | --- |
| STT | Tên trường (applySanctionReport) | Giá trị |
| 1 | ID | Random UUID |
| 2 | CONDITION\_ID | period.CONDITION\_ID |
| 3 | AREA\_CODE | adUnit.AREA\_CODE |
| 4 | CO\_CODE | contractRefDefine. CO\_CODE |
| 5 | CO\_NAME | contractRefDefine. COMPANY\_NAME\_VN |
| 6 | CUSTOMER\_ID | period. CUSTOMER\_ID |
| 7 | CUSTOMER\_NAME | contractRefDefine.SHORT\_NAME |
| 8 | CUSTOMER\_CLASS | objectLD.CUSTOMER\_CLASS |
| 9 | CONTRACT\_MD | Null |
| 10 | INSURANCE\_REQ | Null |
| 11 | COLLATERAL\_ID | Null |
| 12 | COLLATERAL\_TYPE | Null |
| 13 | SEAB\_COL\_SOURCE | Null |
| 14 | CONTRACT\_REF | period. CONTRACT\_REF |
| 15 | CONTRACT\_LD | objectLD.CONTRACT |
| 16 | VALUE\_DATE\_MIN | period. VALUE\_DATE\_MIN |
| 17 | NO\_DAYS\_OVERDUE | objectLD.NO\_DAYS\_OVERDUE |
| 18 | LOAN\_PRODUCT\_ID |  |
| 19 | CONTRACT\_DEBT | objectLD.CONTREACT\_DEBT |
| 20 | EVALUATION\_DATE | period. EVALUATION\_DATE |
| 21 | PERIOD | period. PERIOD |
| 22 | APPLY\_MONTH | period. EVALUATION\_DATE + rule ( Unit Months) ( String format : “yyyy-MM-dd” |
| 23 | SANCTION | period.saction |
| 24 | RATE | objectLD.INTEREST\_RATE |
| 25 | APPLY\_DATE |  |
| 26 | NOTE |  |
| 27 | APPLY\_STATUS | 'NO' |
| 28 | SANCTION\_VIOLATION\_TYPE | sactionViolationType |
| 29 | GROUP\_CONTRACT\_CONDITION | period. GROUP\_CONTRACT\_CONDITION |
| 30 | CONTENT | period. CONTENT |
| 31 | STATUS\_REPORT | CASE WHEN previous is null OR sactionViolationType <> INCRE\_CONDITION' OR APPLY\_MONTH is null THEN 'ACTIVE' WHEN to\_date( previous. APPLY\_MONTH ) < to\_date( APPLY\_MONTH ) THEN 'DELETE' ELSE 'ACTIVE' END |
| 32 | EVALUATION\_RESULT | period. EVALUATION\_RESULT |
| 33 | CREATED\_USER |  |
| 34 | CREATED\_DATE | SYSDATE |
| 35 | UPDATED\_USER | 'PERIOD\_' || period.PERIOD || '\_' || TO\_CHAR(sysdate ,'yyyy-MM-dd HH:mm:ss') |
| 36 | UPDATED\_DATE |  |
| 37 | FREQUENCY | period. FREQUENCY |
| 38 | FREQUENCY\_UNIT | period. FREQUENCY\_UNIT |

add applySanctionReport vào danh sách applySanctionReports

processStatus = CASE WHEN STATUS\_REPORT <> 'ACTIVE' THEN 'DELETE' ELSE 'SUCCESS' END;

B3 : Xác định Report\_note cho kỳ đánh giá và update kỳ đánh giá

reportNote = CASE WHEN processStatus = 'DELETE' THEN 'Áp dụng với trạng thái DELETE' WHEN processStatus <> 'DELETE' AND processStatus <> 'SUCCESS' THEN processSuccess ELSE 'Áp dụng với trạng thái ACTIVE' END.

UPDATE TBL\_PERIOD\_EVALUATION Set period.report\_note = : reportNote WHERE ID= period.ID;

B4 : Tạo các bản ghi báo cáo áp chế tài

Sau khi For Loop dữ liệu các kỳ đánh giá sẽ tạo được 1 list các bản ghi báo cáo áp chế tài trong biến applySanctionReports 🡺 Insert list dữ liệu này vào bảng tbl\_apply\_sanction\_report ( dữ liệu đã được xác định tại bước 2 )

## Áp chế tài vi phạm bảo hiểm tài sản bảo đảm

### Mô tả nghiệp vụ

### Biểu đồ tuần tự

### Quy tắc xử lý dữ liệu

#### Lấy dữ liệu thỏa mãn điều kiện

SELECT

\*

from

TBL\_PERIOD\_EVALUATION pe

where

pe.GROUP\_CONTRACT\_CONDITION = 'SECUR\_PROP\_INSURANCE'

and pe.EVALUATION\_RESULT is not null

and (

pe.STATUS is null

or pe.STATUS <> 'STOP\_FOLLOW'

)

and (

pe.REPORT\_BATCH\_ID is null

or pe.REPORT\_BATCH\_ID = 'RE\_CHECK'

)

#### Xử lý dữ liệu

Xử lý từng dòng dữ liệu lấy được tại mục trên ( period )

* Thông tin trường period. REPORT\_NOTE được lưu dưới dạng String có thể là String thuần (VD : 'Không có dữ liệu tài sản bảo đảm'), có thể là dạng StringJSON được build từ object HashMap của java {key : contract\_ref, value : ${{note}} } (VD : {"AAAA":"Không áp dụng do là kỳ trước ko bị tăng lãi suất","BBBB":"Đã áp dụng cho kỳ trước","CCCC":"Không có dữ liệu hợp đồng giải ngân"} )

Trường hợp dữ liệu được lấy ra để áp chế tài thông tin trường period. REPORT\_NOTE hoặc null hoặc có dữ liệu dạng StringJSON 🡺 convert thông tin trường period. REPORT\_NOTE sang 1 dạng dữ liệu nào đó của plsql để khai thác ( tạm đặt tên biến là reportNotes )

* Từ reportNotes lấy ra list giá trị 'key' gán bằng : listRefHasCheck ( danh sách những ref\_id đã xử lý )
* Lấy list REF\_ID từ trường period.COLLATERAL\_CONTRACT\_REF ( Trường period.COLLATERAL\_CONTRACT\_REF lưu trữ thông tin nhiều REF\_ID khác nhau ngăn cách bằng ký tự space) gán cho biến contractRefs

Nếu trong list contractRefs tất cả các ref\_id đều tồn tại trong list listRefHasCheck ngưng xử lý và update lại dữ liệu đánh giá 🡺 UPDATE

TBL\_PERIOD\_EVALUATION

SET

REPORT\_BATCH\_ID = 'SUCCESS'

WHERE

ID = period.ID;

##### Lấy dữ liệu tài sản bảo đảm

collateral = SELECT \* FROM TBL\_BI\_COLLATERAL WHERE collateral\_id = period.collateral\_id;

Nếu collateral is null 🡺 UPDATE

TBL\_PERIOD\_EVALUATION

SET

REPORT\_NOTE = 'Không có dữ liệu tài sản bảo đảm',

REPORT\_BATCH\_ID = 'STOP\_CHECK\_REPORT'

WHERE

ID = period.ID

##### Lấy dữ liệu những hợp đồng giải ngân tương ứng

contractRefDefines = SELECT \* FROM STG\_SEAOPS.TBL\_BI\_CONTRACT\_REF\_DEFINE

WHERE REF\_ID IN (: contractRefs );

Nếu contractRefDefines is empty 🡺 UPDATE

TBL\_PERIOD\_EVALUATION

SET

REPORT\_NOTE = 'Không có dữ liệu hợp đồng giải ngân',

REPORT\_BATCH\_ID = 'STOP\_CHECK\_REPORT'

WHERE

ID = period.ID

##### Lấy dữ liệu cấu hình chế tài TSBD tương ứng

insuranceSanctions = SELECT

\*

FROM

tbl\_insurance\_sanction

WHERE

COLLATERAL\_TYPES = period.COLLATERAL\_TYPES

AND LOAN\_TYPE = contractRefDefine.LOAN\_PRODUCT\_ID

AND STATUS = 'ACTIVE'

Nếu insuranceSanctions is empty 🡺 UPDATE

TBL\_PERIOD\_EVALUATION

SET

REPORT\_NOTE = 'Không có chế tài bảo hiểm tài sản bảo đảm',

REPORT\_BATCH\_ID = 'STOP\_CHECK\_REPORT'

WHERE

ID = period.ID

###### Tạo dữ liệu áp chế tài

\*) B1 : Tạo các biến ngoài thông tin HD giải ngân

* applySanctionReports : Biến lưu list dữ liệu áp chế tài ( kiểu dữ liệu lấy theo bảng tbl\_apply\_sanction\_report ). Mỗi khi xử lý 1 row dữ sẽ add them dữ liệu vào List này để tạo mới dữ liệu chế tài.
* Tạo biến kết quả đánh giá result = reportNotes nếu reportNotes không có dữ liệu 🡺 khởi tạo biến để lưu trữ dưới dạng tương tự HashMap trong java.
* Tạo biến đếm số lần áp chế tài của các REF numCheckRef =

Thông tin trường period. CHECK\_PERIOD\_REPORT được lưu dưới dạng StringJSON được build từ object HashMap của java {key : contract\_ref, value : ${{number}} } (VD : {"AAAA":1,"BBBB":2,"CCCC":0) 🡺 convert thông tin trường period. REPORT\_NOTE sang 1 dạng dữ liệu nào đó của plsql để khai thác nếu period. CHECK\_PERIOD\_REPORT không có dữ liệu 🡺 khởi tạo biến để lưu trữ dưới dạng tương tự HashMap trong java.

\*) B2 : FOR LOOP list contractRefs ( refId ) tạo các biến phụ thuộc, đánh giá chế tài và gán dữ liệu vào danh sách applySanctionReports

* Tìm chế tài kỳ trước ( previous )

checkPrevious = SELECT

\*

FROM

tbl\_apply\_sanction\_report

WHERE

collateral\_id = peroid.collateral\_id

AND CONTRACT\_REF = refId

AND status\_report = 'ACTIVE'

ORDER BY DESC

create\_user

limit

1

checkExits = SELECT

\*

FROM

applySanctionReports

WHERE

collateral\_id = peroid.collateral\_id

AND CONTRACT\_REF = refId

AND status\_report = 'ACTIVE'

ORDER BY DESC

create\_user

limit

1

+ Nếu checkExits not null null 🡺 previous = checkExits

+ Ngược lại previous = checkPrevious

* Đánh giá dữ liệu chế tài qua thông tin previous:

+ Nếu previous is null AND period.auto\_evaluation\_result = 'FOLLOW' 🡺 result.put(ref, "Không áp dụng do là kỳ trước ko bị tăng lãi suất")

+ Nếu previous is not null AND period.auto\_evaluation\_result = previous.evaluation\_result 🡺 result.put(ref, "Đã áp dụng cho kỳ trước")

* Tìm thông tin hợp đồng tương ứng refId

contract = SELECT \* FROM STG\_SEAOPS.TBL\_BI\_CONTRACT\_REF\_DEFINE

WHERE REF\_ID = refId;

* Đánh giá dữ liệu chế tài qua thông tin contract:

+ Nếu contract is null 🡺 result.put(ref, "Không có dữ liệu hợp đồng giải ngân")

+ Nếu contract.data\_status in ('INACTIVE','OVERDUE\_1','OVERDUE\_2') 🡺 result.put(ref, "Hợp đồng không thuộc phạm vi theo dõi")

* Tính ngày báo cáo (valueDateMin ) và check đã đến ngày chạy báo cáo chưa

sactionType = CASE WHEN period.AUTO\_EVALUATION\_RESULT = 'FLLOW' THEN 'DECRE\_INSURANCE' ELSE 'INCRE\_INSURANCE' END

interestAdjustEnum = CASE WHEN sactionType = 'INCRE\_INSURANCE' THEN 'INCREASE' ELSE 'DECREASE' END

monthValue = CASE WHEN (

numCheckMap is null

OR numCheckMap.get(refId) is null

)

AND previous is not null

AND period.DATE\_SYNC\_DATA is not null THEN period.DATE\_SYNC\_DATA WHEN (

numCheckMap is null

OR numCheckMap.get(refId) is null

)

AND previous is not null

AND period.DATE\_SYNC\_DATA is null THEN collateral.DATE\_SYNC\_DATA ELSE period.EVALUATION\_DATE

reportDateType 🡺

+ Nếu interestAdjustEnum = 'INCREASE' 🡪 reportDateType = 'LAST\_DAY'

+ Nếu interestAdjustEnum = 'DECREASE' và DAY OF MONTH của monthValue < 25 🡪 reportDateType = 'BEFORE\_25'

+ Nếu interestAdjustEnum = 'DECREASE' và DAY OF MONTH của monthValue >= 25 🡪 reportDateType = 'FROM\_25'

Kiểm tra đến ngày chạy báo cáo chưa 🡺

+ Nếu monthValue >= Ngày đầu tháng của thời gian hiện tại 🡪 bỏ qua đối tượng hiện tại và xử lý đối tượng tiếp theo trong vòng FOR

+ Nếu reportDateType = 'BEFORE\_25' và monthValue >= Ngày 25 của tháng hiện tại 🡪 bỏ qua đối tượng hiện tại và xử lý đối tượng tiếp theo trong vòng FOR

+ Nếu reportDateType = 'FROM\_25' và monthValue < Ngày 25 của tháng hiện tại 🡪 bỏ qua đối tượng hiện tại và xử lý đối tượng tiếp theo trong vòng FOR

+ Nếu reportDateType = 'LAST\_DAY' và monthValue <> Ngày cuối tháng của thời gian hiện tại 🡪 bỏ qua đối tượng hiện tại và xử lý đối tượng tiếp theo trong vòng FOR

* Lọc đữ liệu cấu hình chế tài thỏa mãn điều kiện

insuranceSanction = SELECT \* FROM insuranceSanctions WHERE START\_DATE >= monthValue AND (END\_DATE is null OR (END\_DATE > monthValue ) ) AND INTEREST\_ADJUST = interestAdjustEnum AND REPORT\_DATE = reportDateType;

Nếu insuranceSanction is null 🡺 result.put(ref, " Không có dữ liệu áp chế tài")

* Truy vấn danh sách dư nợ theo refId ( lstLD )

lstLd = SELECT

\*

FROM

stg\_seaops.tbl\_bi\_contract\_debt

WHERE

CONTRACT\_REF = refID

AND DEBT > 0

AND DAYID = (

SELECT

MAX(DAYID)

FROM

stg\_seaops.tbl\_bi\_contract\_debt

WHERE

CONTRACT\_REF = refID

AND DEBT > 0

)

Nếu lstLd is empty 🡺 result.put(ref, "Không truy vấn được thông tin LD theo REF")

* Xác định đơn vị ( adUnit )

adUnit = SELECT

\*

FROM

KSSCT.ad\_unit

WHERE

unit\_code = period.CO\_CODE

* Xác định số lần đánh giá của REF

timeRun = CASE WHEN numCheckMap is null

OR numCheckMap.get(refId) is null THEN 1 ELSE numCheckMap.get(refId) + 1 END

B3 : Xác định trạng thái chế tài và khởi tạo object lưu chế tài ( result và applySanctionReport )

Xử lý FOR LOOP dữ liệu lstLD (objectLD) khởi tạo đối tượng applySanctionReport gán vào danh sách applySanctionReports và gán lại processStatus theo bảng maping sau:

applySanctionReport :

|  |  |  |
| --- | --- | --- |
| STT | Tên trường (applySanctionReport) | Giá trị |
| 1 | ID | Random UUID |
| 2 | CONDITION\_ID | period.CONDITION\_ID |
| 3 | AREA\_CODE | adUnit.AREA\_CODE |
| 4 | CO\_CODE | contract. CO\_CODE |
| 5 | CO\_NAME | contract. COMPANY\_NAME\_VN |
| 6 | CUSTOMER\_ID | period. CUSTOMER\_ID |
| 7 | CUSTOMER\_NAME | contract.SHORT\_NAME |
| 8 | CUSTOMER\_CLASS | objectLD.CUSTOMER\_CLASS |
| 9 | CONTRACT\_MD | collateral.CONTRACT\_MD |
| 10 | INSURANCE\_REQ | collateral.INSURANCE\_REQ |
| 11 | COLLATERAL\_ID | collateral.COLLATERAL\_ID |
| 12 | COLLATERAL\_TYPE | collateral.COLLATERAL\_TYPE |
| 13 | SEAB\_COL\_SOURCE | collateral.SEAB\_COL\_SOURCE |
| 14 | CONTRACT\_REF | refId |
| 15 | CONTRACT\_LD | objectLD.CONTRACT |
| 16 | VALUE\_DATE\_MIN | period. VALUE\_DATE\_MIN |
| 17 | NO\_DAYS\_OVERDUE | objectLD.NO\_DAYS\_OVERDUE |
| 18 | LOAN\_PRODUCT\_ID |  |
| 19 | CONTRACT\_DEBT | objectLD.CONTREACT\_DEBT |
| 20 | EVALUATION\_DATE | period. EVALUATION\_DATE |
| 21 | PERIOD | period. PERIOD |
| 22 | APPLY\_MONTH | monthValue + insuranceSanction.APPLY\_MONTH ( Unit Months) ( String format : “yyyy-MM-dd” |
| 23 | SANCTION | period.saction |
| 24 | RATE | objectLD.INTEREST\_RATE |
| 25 | APPLY\_DATE |  |
| 26 | NOTE |  |
| 27 | APPLY\_STATUS | 'NO' |
| 28 | SANCTION\_VIOLATION\_TYPE | sactionViolationType |
| 29 | GROUP\_CONTRACT\_CONDITION | period. GROUP\_CONTRACT\_CONDITION |
| 30 | CONTENT | period. CONTENT |
| 31 | STATUS\_REPORT | CASE WHEN previous is null OR previous.EVALUTION\_RESULT = 'NOT\_FOLLOW' OR previous.APPLY\_MONTH is null OR APPLY\_MONTH is null THEN 'ACTIVE' WHEN to\_date( previous. APPLY\_MONTH ) < to\_date( APPLY\_MONTH ) THEN 'DELETE' ELSE 'ACTIVE' END |
| 32 | EVALUATION\_RESULT | period. EVALUATION\_RESULT |
| 33 | CREATED\_USER |  |
| 34 | CREATED\_DATE | SYSDATE |
| 35 | UPDATED\_USER | 'PERIOD\_' || period.PERIOD || '\_TIME\_' || timeRun |
| 36 | UPDATED\_DATE |  |
| 37 | FREQUENCY | period. FREQUENCY |
| 38 | FREQUENCY\_UNIT | period. FREQUENCY\_UNIT |

add applySanctionReport vào danh sách applySanctionReports

Nếu STATUS\_REPORT = 'DELETE' 🡪 result.put(ref, "Áp dụng chế tài với trạng thái DELETE") ngược lại 🡪 result.put(ref, "Áp dụng chế tài với trạng thái ACTIVE")

B3 : Xác định Report\_note và trạng thái đánh giá cho kỳ đánh giá và update kỳ đánh giá

lstRefHasCheck = số phần tử của HashMap result

* Xác định lại giá trị checkPeriodReport

for (String ref : lstRefHasCheck ) {

if (numCheckMap.get(ref) == null) {

numCheckMap.put(ref, 1);

} else {

numCheckMap.put(ref, numCheckMap.get(ref) + 1);

}

}

checkPeriodReport = numCheckMap convert thành JsonString

* Xác định reportNote

reportNote = result convert thành JsonString

* Xác định reportBatchId

reportBatchId = 'SUCCESS' nếu số phần tử trong list lstRefHasCheck = số phần tử contractRefs ngược lại lấy giá trị 'RE\_CHECK'

UPDATE TBL\_PERIOD\_EVALUATION SET period.report\_note = : reportNote, CHECK\_PERIOD\_REPORT = : checkPeriodReport ,REPORT\_BATCH\_ID = : reportBatchId WHERE ID= period.ID;

B5 : Tạo các bản ghi báo cáo áp chế tài

Sau khi For Loop dữ liệu các kỳ đánh giá sẽ tạo được 1 list các bản ghi báo cáo áp chế tài trong biến applySanctionReports 🡺 Insert list dữ liệu này vào bảng tbl\_apply\_sanction\_report ( dữ liệu đã được xác định tại bước 3 )

## Báo cáo KPI

### Mô tả nghiệp vụ

### Biểu đồ tuần tự

### Quy tắc xử lý dữ liệu

Dữ liệu báo cáo sẽ chạy theo tháng ngày đầu tiên của tháng sẽ lấy toàn bộ dữ liệu của tháng trước param pFromDate là ngày đầu tiên của tháng trước, param pToDate là ngày đầu tiên của tháng hiện tại

VD : pFromDate = 2024-05-01 ; pToDate = 2024-06-01

Lấy danh sách đơn vị ( coCodes )

adUnit =

SELECT \* FROM KSSCT.ad\_unit

WHERE STATUS = 'ACTIVE' GROUP BY unit\_code;

#### Xử lý và cập nhật dữ liệu báo cáo KPI theo từng đơn vị

p\_co\_code = adUnit.UNIT\_CODE

p\_area\_code = adUnit.AREA\_CODE

* Lấy tổng số bản ghi kỳ đánh giá

totalRecords = SELECT COUNT(\*)

from TBL\_PERIOD\_EVALUATION

where CO\_CODE = :p\_co\_code

and cast(EVALUATION\_DATE as date) >= to\_date(:p\_from\_date, 'YYYY-MM-DD HH24:MI:SS')

and cast(EVALUATION\_DATE as date) < to\_date(:p\_to\_date, 'YYYY-MM-DD HH24:MI:SS')

and (STATUS is null or STATUS <> 'STOP\_FOLLOW');

* Lấy tổng só bản ghi đánh giá tuân thủ

periodFollow = SELECT COUNT(\*)

from TBL\_PERIOD\_EVALUATION

where CO\_CODE = :p\_co\_code

and cast(EVALUATION\_DATE as date) >= to\_date(:p\_from\_date, 'YYYY-MM-DD HH24:MI:SS')

and cast(EVALUATION\_DATE as date) < to\_date(:p\_to\_date, 'YYYY-MM-DD HH24:MI:SS')

and (STATUS is null or STATUS <> 'STOP\_FOLLOW')

and evaluation\_result = 'FOLLOW';

##### Tính và lưu dữ liệu KPI với trường hợp không có dữ liệu KPI trong năm

\*) Trường hợp này xảy ra khi ngày chạy tạo dữ liệu báo cáo là Tháng 2 ( Lấy dữ liệu tháng 1 để tạo báo cáo ) 🡺 dữ liệu KPI trong năm chưa có

kpiReport = SELECT \*

from TBL\_KPI\_REPORT

where CO\_CODE = :p\_co\_code

and (MONTH\_YEAR = (EXTRACT(YEAR FROM SYSDATE) -1 || '-12-01'))

Mapping thông tin các trường bảng TBL\_KPI\_REPORT và insert vào DB như sau :

|  |  |  |
| --- | --- | --- |
| STT | Tên trường | Giá trị |
| 1 | ID | Random UUID |
| 2 | TOTAL\_RECORD | totalRecords |
| 3 | FOLLOW | periodFollow |
| 4 | NOT\_FOLLOW | totalRecords -  periodFollow |
| 5 | VIOLATION\_RATE | totalRecords = 0 🡺 0  totalRecords <> = 🡺  (totalRecords -  periodFollow) \* 100 / totalRecords – Làm tròn 2 chữ số sau dấu phẩy |
| 6 | VIOLATION\_POINT | totalRecords = 0 🡺 10  totalRecords <> = 🡺  0 <= VIOLATION\_RATE <= 10 🡪 10  10 < VIOLATION\_RATE <= 20 🡪 9  20 < VIOLATION\_RATE <= 30 🡪 8  30 < VIOLATION\_RATE <= 40 🡪 7  40 < VIOLATION\_RATE <= 50 🡪 6  50 < VIOLATION\_RATE <= 60 🡪 5  60 < VIOLATION\_RATE <= 70 🡪 4  70 < VIOLATION\_RATE <= 80 🡪 3  80 < VIOLATION\_RATE <= 90 🡪 2  90 < VIOLATION\_RATE <= 100 🡪 1 |
| 7 | FIX\_RATE | totalRecords = 0 🡺 0  totalRecords <> = 🡺  100 - VIOLATION\_RATE |
| 8 | FIX\_POINT | totalRecords = 0 🡺 12  totalRecords <> = 🡺  0 <= FIX\_RATE <= 10 🡪 1  10 < FIX\_RATE <= 20 🡪 2  20 < FIX\_RATE <= 30 🡪 3  30 < FIX\_RATE <= 40 🡪 4  40 < FIX\_RATE <= 50 🡪 5  50 < FIX\_RATE <= 60 🡪 6.5  60 < FIX\_RATE <= 70 🡪 8  70 < FIX\_RATE <= 80 🡪 9.5  80 < FIX\_RATE <= 90 🡪 11  90 < FIX\_RATE <= 100 🡪 12 |
| 9 | MONTH\_POINT | kpiReport is null 🡪  VIOLATION\_POINT/2 – Làm tròn 2 chữ số sau dấu phẩy  kpiReport is not null 🡪  VIOLATION\_POINT/2 + kpiReport.FIX\_POINT/2 – Làm tròn 2 chữ số sau dấu phẩy |
| 10 | ACCUMULATED\_POINT | MONTH\_POINT |
| 11 | CREATED\_TIME | sysdate |
| 12 | CO\_CODE | p\_co\_code |
| 13 | CREATED\_USER | SYSTEM\_USER |
| 14 | MONTH\_YEAR | p\_from\_date |
| 15 | AREA\_CODE | p\_area\_code |

Lưu dữ liệu vừa khởi tạo vào bảng TBL\_KPI\_REPORT

##### Tính và lưu dữ liệu KPI với trường hợp có dữ liệu KPI trong năm

* KPI tháng trước nữa

kpiReport = SELECT \* FORM TBL\_KPI\_REPORT WHERE CO\_CODE = :p\_co\_code

AND ( MONTH\_YEAR = EXTRACT(YEAR FROM SYSDATE) || '-' ||(CASE WHEN (EXTRACT(MONTH FROM SYSDATE) -2) < 10 THEN '0' || TO\_CHAR(EXTRACT(MONTH FROM SYSDATE) -2) ELSE TO\_CHAR(EXTRACT(MONTH FROM SYSDATE) -2) END ) || '-01' );

* Tổng Month\_Point KPI trong năm đã tính

totalPoint = ( SELECT SUM (MONTH\_POINT) FROM TBL\_KPI\_REPORT WHERE CO\_CODE = :p\_co\_code AND EXTRACT(YEAR FROM SYSDATE) = EXTRACT(YEAR FROM to\_date(MONTH\_YEAR,'YYYY-MM-DD')) GROUP BY CO\_CODE ) ;

* Số tháng đã được tính KPI

totalMonth = SELECT EXTRACT(MONTH FROM SYSDATE) -2) FROM DUAL;

|  |  |  |
| --- | --- | --- |
| STT | Tên trường | Giá trị |
| 1 | ID | Random UUID |
| 2 | TOTAL\_RECORD | totalRecords |
| 3 | FOLLOW | periodFollow |
| 4 | NOT\_FOLLOW | totalRecords -  periodFollow |
| 5 | VIOLATION\_RATE | totalRecords = 0 🡺 0  totalRecords <> = 🡺  (totalRecords -  periodFollow) \* 100 / totalRecords – Làm tròn 2 chữ số sau dấu phẩy |
| 6 | VIOLATION\_POINT | totalRecords = 0 🡺 10  totalRecords <> = 🡺  0 <= VIOLATION\_RATE <= 10 🡪 10  10 < VIOLATION\_RATE <= 20 🡪 9  20 < VIOLATION\_RATE <= 30 🡪 8  30 < VIOLATION\_RATE <= 40 🡪 7  40 < VIOLATION\_RATE <= 50 🡪 6  50 < VIOLATION\_RATE <= 60 🡪 5  60 < VIOLATION\_RATE <= 70 🡪 4  70 < VIOLATION\_RATE <= 80 🡪 3  80 < VIOLATION\_RATE <= 90 🡪 2  90 < VIOLATION\_RATE <= 100 🡪 1 |
| 7 | FIX\_RATE | totalRecords = 0 🡺 0  totalRecords <> = 🡺  100 - VIOLATION\_RATE |
| 8 | FIX\_POINT | totalRecords = 0 🡺 12  totalRecords <> = 🡺  0 <= FIX\_RATE <= 10 🡪 1  10 < FIX\_RATE <= 20 🡪 2  20 < FIX\_RATE <= 30 🡪 3  30 < FIX\_RATE <= 40 🡪 4  40 < FIX\_RATE <= 50 🡪 5  50 < FIX\_RATE <= 60 🡪 6.5  60 < FIX\_RATE <= 70 🡪 8  70 < FIX\_RATE <= 80 🡪 9.5  80 < FIX\_RATE <= 90 🡪 11  90 < FIX\_RATE <= 100 🡪 12 |
| 9 | MONTH\_POINT | VIOLATION\_POINT/2 + kpiReport.FIX\_POINT/2 – Làm tròn 2 chữ số sau dấu phẩy |
| 10 | ACCUMULATED\_POINT | (totalPoint + MONTH\_POINT) / (totalMonth + 1) 2 – Làm tròn 2 chữ số sau dấu phẩy |
| 11 | CREATED\_TIME | sysdate |
| 12 | CO\_CODE | p\_co\_code |
| 13 | CREATED\_USER | SYSTEM\_USER |
| 14 | MONTH\_YEAR | p\_from\_date |
| 15 | AREA\_CODE | p\_area\_code |

Lưu dữ liệu vừa khởi tạo vào bảng TBL\_KPI\_REPORT

## Job sinh kỳ dựa theo Điều kiện sản phẩm Khác

### Mô tả nghiệp vụ

Job tự động sinh kỳ dựa theo dữ liệu lịch sử giản ngân và điều kiện sản phẩm khác.

### Biểu đồ tuần tự

### Quy tắc xử lý dữ liệu

#### Lấy dữ liệu thỏa mãn điều kiện

**SELECT**

*DH*.CONTRACT\_REF *ref\_id*, *DH*.LOAN\_CODE *loan\_code*, *CRD*.VALUE\_DATE\_MIN *value\_date\_min\_ref*, *DH*.VALUE\_DATE *value\_date*, *DH*.PRODUCT\_CODE *product\_code*, *DH*. MATURITY\_DATE *maturity\_date*

**FROM**

TBL\_BI\_DISBURSEMENT\_HISTORY *DH*

**JOIN** TBL\_BI\_CONTRACT\_REF\_DEFINE *CRD*

**ON** *CRD*.REF\_ID = *DH*.CONTRACT\_REF

**JOIN** TBL\_CONTRACT\_CONDITION *CD*

**ON** *DH*.CONTRACT\_REF = *CD*.CONTRACT\_ID

**WHERE**

*DH*.DATA\_STATUS = **'ACTIVE'**

**and** *DH*.CREATE\_OTHER\_PERIOD\_STATUS = **null**

**and** *CD*.GROUP\_CONTRACT\_CONDITION = **'OTHER'**

**and** *CD*.GROUP\_CONDITION\_TYPE = ‘ĐKSP’

**UNION** **ALL**

**SELECT**

*DH*.CONTRACT\_REF *ref\_id*, *dh*.contract\_md *loan\_code*, *CRD*.VALUE\_DATE\_MIN *value\_date\_min\_ref*, *dh*.past\_sched\_date *value\_date*, *DH*.PRODUCT\_CODE *product\_code*, *DH*. MATURITY\_DATE *maturity\_date*

**FROM**

TBL\_BI\_GUARANTEE\_SALES *DH*

**JOIN** TBL\_BI\_CONTRACT\_REF\_DEFINE *CRD*

**ON** *CRD*.REF\_ID = *DH*.CONTRACT\_REF

**JOIN** TBL\_CONTRACT\_CONDITION\_BIPLUS *CD*

**ON** *DH*.CONTRACT\_REF = *CD*.CONTRACT\_ID

**WHERE**

*DH*.DATA\_STATUS = **'ACTIVE'**

**and** *DH*.CREATE\_OTHER\_PERIOD\_STATUS = **null**

**and** *CD*.GROUP\_CONTRACT\_CONDITION = **'OTHER'**

**and** *CD*.GROUP\_CONDITION\_TYPE = **'ĐKSP'**

**UNION** **ALL**

**SELECT**

*DH*.CONTRACT\_REF *ref\_id*, *dh*.contract\_id *loan\_code*, *CRD*.VALUE\_DATE\_MIN *value\_date\_min\_ref*, *dh*.issue\_date *value\_date*, *DH*.PRODUCT\_CODE *product\_code*, *DH*. MATURITY\_DATE *maturity\_date*

**FROM**

TBL\_BI\_LC\_RELEASE\_SALES *DH*

**JOIN** TBL\_BI\_CONTRACT\_REF\_DEFINE\_BIPLUS *CRD*

**ON** *CRD*.REF\_ID = *DH*.CONTRACT\_REF

**JOIN** TBL\_CONTRACT\_CONDITION\_BIPLUS *CD*

**ON** *DH*.CONTRACT\_REF = *CD*.CONTRACT\_ID

**WHERE**

*DH*.DATA\_STATUS = **'ACTIVE'**

**and** *DH*.CREATE\_OTHER\_PERIOD\_STATUS = **null**

**and** *CD*.GROUP\_CONTRACT\_CONDITION = **'OTHER'**

**and** *CD*.GROUP\_CONDITION\_TYPE = **'ĐKSP'**;

Lấy id hợp đồng, id khoản giải ngân, ngày giải ngân hợp đồng, ngày giải ngân khoản giải ngân, product\_code khoản giải ngân, ngày đến hạn của khoản giải ngân

#### Xử lý dữ liệu

Dữ liệu lấy ta từ bước trên được xử lý theo từng Row (contract) theo quy tắc sau :

B1 : Lấy dữ liệu tương ứng làm điều kiện đánh giá

* Dữ liệu điều kiện sản phẩm Khác (nhóm ĐKSP) đã duyệt, tương ứng với hợp đồng:

condition = SELECT

\*

FROM

TBL\_CONTRACT\_CONDITION

WHERE

GROUP\_CONTRACT\_CONDITION = 'OTHER'

GROUP\_CONDITION\_TYPE = 'ĐKSP'

AND STATUS = 'APPROVED'

AND CONTRACT\_ID = ref\_id; // ref\_id lấy từ dữ liệu select đầu vào

B2 : Dựa vào thông tin của condition bước 1

- Nếu condition.METHOD\_OF\_PERIOD = METHOD\_PERIOD\_F1:

Kiểm tra đã có hạng mục sinh ra bởi Điều kiện và Hợp đồng. Nếu đã có rồi thì không sinh hạng mục nữa.

**SELECT** \* **FROM** TBL\_PERIOD\_EVALUATION **WHERE** CONTRACT\_CONDITION\_ID = condition.ID **AND** CONTRACT\_ID = ref\_id; // Tồn tại bản ghi thì không sinh hạng mục

Nếu chưa, thực hiện sinh hạng mục, tính toán ngày đến hạn kiểm tra

EVALUATION\_DATE = value\_date\_min\_ref + condition.deadline \* condition.deadline\_unit

condition.deadline là số

condition.deadline\_unit = D(ngày) hoặc M(tháng) hoặc LAST\_DAY (cùng cách tính đơn vị tháng, lấy ngày cuối cùng của tháng)

Ví dụ: value\_date\_min\_ref = 23/07/2024

condition.deadline = 1, condition.deadline\_unit = D, EVALUATION\_DATE = 24/07/2024

condition.deadline = 1, condition.deadline\_unit = M, EVALUATION\_DATE = 23/08/2024

condition.deadline = 1, condition.deadline\_unit = LAST\_DAY, EVALUATION\_DATE = 31/08/2024

- Nếu condition.METHOD\_OF\_PERIOD = METHOD\_PERIOD\_F2:

Kiểm tra product\_code của điều kiện và khoản giải ngân trùng nhau không, không trùng thì không sinh hạng mục:

Nếu condition. LOAN\_PRODUCT\_ID != product\_code (product\_code lấy từ dữ liệu đầu vào ban đầu) 🡺 Không sinh hạng mục

Ngược lại, thực hiện sinh hạng mục, tính toán ngày đến hạn kiểm tra

EVALUATION\_DATE = value\_date + condition.deadline \* condition.deadline\_unit

- Nếu condition.METHOD\_OF\_PERIOD = METHOD\_PERIOD\_F3:

Kiểm tra product\_code của điều kiện và khoản giải ngân trùng nhau không, không trùng thì không sinh hạng mục:

Nếu condition. LOAN\_PRODUCT\_ID != product\_code (product\_code lấy từ dữ liệu đầu vào ban đầu) 🡺 Không sinh hạng mục

Kiểm tra khoản giải ngân này có phải là khoản giải ngân đầu tiên không? Nếu không phải khoản giải ngân đầu tiên, không sinh hạng mục:

**SELECT** **min**(*VALUE\_DATE*) min\_value\_date **FROM** (

**SELECT** VALUE\_DATE *VALUE\_DATE*

**FROM**

STG\_SEAOPS.TBL\_BI\_DISBURSEMENT\_HISTORY *DH*

**WHERE** *DH*.CONTRACT\_REF = ref\_id **AND** *DH*.PRODUCT\_CODE = product\_code

**UNION** **ALL**

**SELECT** ISSUE\_DATE *VALUE\_DATE*

**FROM**

STG\_SEAOPS.TBL\_BI\_LC\_RELEASE\_SALES *LC*

**WHERE** *LC*.CONTRACT\_REF = ref\_id **AND** *LC*.PRODUCT\_CODE = product\_code

**UNION** **ALL**

**SELECT** PAST\_SCHED\_DATE *VALUE\_DATE*

**FROM**

STG\_SEAOPS.TBL\_BI\_GUARANTEE\_SALES *GS*

**WHERE** *GS*.CONTRACT\_REF = ref\_id **AND** *GS*.PRODUCT\_CODE = product\_code

); // nếu min\_value\_date < value\_date dữ liệu đầu vào thì không sinh hạng mục

Kiểm tra đã sinh hạng mục với product\_code của khoản giải ngân này chưa. Nếu sinh hạng mục rồi, thì không sinh hạng mục nữa.

**SELECT** \* **FROM** TBL\_PERIOD\_EVALUATION **WHERE** CONTRACT\_CONDITION\_ID = **condition**.ID

**AND** CONTRACT\_ID = ref\_id

**AND** LOAN\_PRODUCT\_ID = product\_code; // Tồn tại bản ghi thì không sinh hạng mục

Tính ngày đến hạn kiểm tra:

EVALUATION\_DATE = min\_value\_date + condition.deadline \* condition.deadline\_unit

- Nếu condition.METHOD\_OF\_PERIOD = METHOD\_PERIOD\_F4:

Tính ngày đến hạn kiểm tra:

EVALUATION\_DATE = maturity\_date - condition.deadline \* condition.deadline\_unit

B2 : Tạo dữ liệu kỳ đánh giá

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên trường** | **Giá trị** |
| 1 | ID | RandomString |
| 2 | PERIOD | CASE WHEN condition.START\_PERIOD is not null THEN condition.START\_PERIOD ELSE 1 END |
| 3 | FREQUENCY | CASE WHEN condition.FREQUENCY is not null THEN condition.FREQUENCY ELSE 0 END |
| 4 | FREQUENCY\_UNIT | condition.FREQUENCY\_UNIT |
| 5 | AUTO\_PERIOD | CASE WHEN condition. AUTO\_PERIOD is not null THEN condition. AUTO\_PERIOD ELSE false END |
| 6 | BUSINESS\_UNIT\_NOTE | CASE WHEN condition. BUSINESS\_UNIT\_NOTE is not null THEN condition. BUSINESS\_UNIT\_NOTE ELSE false END |
| 7 | CONTENT | condition.CONTENT |
| 8 | CONTROL\_METHOD | condition. CONTROL\_METHOD |
| 9 | CONTRACT\_CONDITION\_ID | condition. CONTRACT\_CONDITION\_ID |
| 10 | CONTRACT\_CONDITION\_STATUS | condition. CONTRACT\_CONDITION\_STATUS |
| 11 | FIRST\_DATE\_CHOSEN\_PERIOD | condition. FIRST\_DATE\_CHOSEN\_PERIOD |
| 12 | DESCRIPTION | condition. DESCRIPTION |
| 13 | CONTRACT\_ID | contract.REF\_ID |
| 14 | CONTRACT\_STATUS | contract.STATUS |
| 15 | CUSTOMER\_ID | contract.CUSTOMER\_ID |
| 16 | CUSTOMER\_CLASS | contract. CUSTOMER\_CLASS |
| 17 | CO\_CODE | contract.CO\_CODE |
| 18 | COMPANY\_NAME\_VN | contract. COMPANY\_NAME\_VN |
| 19 | SHORT\_NAME | contract. SHORT\_NAME |
| 20 | LOAN\_CODE | contract. CONTRACT |
| 21 | REVOLVING | contract. REVOLVING |
| 22 | VALUE\_DATE\_MIN | contract.VALUE\_DATE |
| 23 | LOAN\_PRODUCT\_ID | contract. LOAN\_PRODUCT\_ID |
| 24 | CREATE\_NEXT\_PERIOD\_STATUS | 'NEW' |
| 25 | CREATED\_DATE | SYSDATE |
| 26 | CREATED\_USER | 'SYSTEM\_USER' |
| 27 | EVALUATION\_DATE | EVALUATION\_DATE đã tính ở trên |
| 28 | AUTO\_EVALUATION\_RESULT | CASE WHEN evalDate < SYSDATE THEN NOT\_FOLLOW ELSE null END |
| 29 | EVALUATION\_RESULT | CASE WHEN evalDate < SYSDATE THEN NOT\_FOLLOW ELSE null END |
| 30 | GROUP\_CONTRACT\_CONDITION | OTHER |
| 31 | GROUP\_CONDITION\_TYPE | ĐKSP |

Lưu dữ liệu vừa khởi tạo vào bảng TBL\_PERIOD\_EVALUATION đồng thời

update bảng

STG\_SEAOPS.TBL\_BI\_DISBURSEMENT\_HISTORY, STG\_SEAOPS.TBL\_BI\_LC\_RELEASE\_SALES, STG\_SEAOPS.TBL\_BI\_GUARANTEE\_SALES

SET

CREATE\_OTHER\_PERIOD\_STATUS = ‘SUCCESS’

## Job sinh kỳ dựa theo Điều kiện sản phẩm Bán chéo

### Mô tả nghiệp vụ

Job tự động sinh kỳ dựa theo dữ liệu lịch sử giản ngân và điều kiện sản phẩm Bán chéo.

### Biểu đồ tuần tự

### Quy tắc xử lý dữ liệu

#### Lấy dữ liệu thỏa mãn điều kiện

**SELECT**

*DH*.CONTRACT\_REF *ref\_id*, *DH*.LOAN\_CODE *loan\_code*, *CRD*.VALUE\_DATE\_MIN *value\_date\_min\_ref*, *DH*.VALUE\_DATE *value\_date*, *DH*.PRODUCT\_CODE *product\_code*, *DH*. MATURITY\_DATE *maturity\_date*

**FROM**

TBL\_BI\_DISBURSEMENT\_HISTORY *DH*

**JOIN** TBL\_BI\_CONTRACT\_REF\_DEFINE *CRD*

**ON** *CRD*.REF\_ID = *DH*.CONTRACT\_REF

**JOIN** TBL\_CONTRACT\_CONDITION *CD*

**ON** *DH*.CONTRACT\_REF = *CD*.CONTRACT\_ID

**WHERE**

*DH*.DATA\_STATUS = **'ACTIVE'**

**and** *DH*.CREATE\_CROSS\_SELLING\_PERIOD\_STATUS = **null**

**and** *CD*.GROUP\_CONTRACT\_CONDITION = **'OTHER'**

**and** *CD*.GROUP\_CONDITION\_TYPE = ‘ĐKSP’

**UNION** **ALL**

**SELECT**

*DH*.CONTRACT\_REF *ref\_id*, *dh*.contract\_md *loan\_code*, *CRD*.VALUE\_DATE\_MIN *value\_date\_min\_ref*, *dh*.past\_sched\_date *value\_date*, *DH*.PRODUCT\_CODE *product\_code*, *DH*. MATURITY\_DATE *maturity\_date*

**FROM**

TBL\_BI\_GUARANTEE\_SALES *DH*

**JOIN** TBL\_BI\_CONTRACT\_REF\_DEFINE *CRD*

**ON** *CRD*.REF\_ID = *DH*.CONTRACT\_REF

**JOIN** TBL\_CONTRACT\_CONDITION\_BIPLUS *CD*

**ON** *DH*.CONTRACT\_REF = *CD*.CONTRACT\_ID

**WHERE**

*DH*.DATA\_STATUS = **'ACTIVE'**

**and** *DH*.CREATE\_CROSS\_SELLING\_PERIOD\_STATUS = **null**

**and** *CD*.GROUP\_CONTRACT\_CONDITION = **'OTHER'**

**and** *CD*.GROUP\_CONDITION\_TYPE = **'ĐKSP'**

**UNION** **ALL**

**SELECT**

*DH*.CONTRACT\_REF *ref\_id*, *dh*.contract\_id *loan\_code*, *CRD*.VALUE\_DATE\_MIN *value\_date\_min\_ref*, *dh*.issue\_date *value\_date*, *DH*.PRODUCT\_CODE *product\_code*, *DH*. MATURITY\_DATE *maturity\_date*

**FROM**

TBL\_BI\_LC\_RELEASE\_SALES *DH*

**JOIN** TBL\_BI\_CONTRACT\_REF\_DEFINE\_BIPLUS *CRD*

**ON** *CRD*.REF\_ID = *DH*.CONTRACT\_REF

**JOIN** TBL\_CONTRACT\_CONDITION\_BIPLUS *CD*

**ON** *DH*.CONTRACT\_REF = *CD*.CONTRACT\_ID

**WHERE**

*DH*.DATA\_STATUS = **'ACTIVE'**

**and** *DH*.CREATE\_CROSS\_SELLING\_PERIOD\_STATUS = **null**

**and** *CD*.GROUP\_CONTRACT\_CONDITION = **'OTHER'**

**and** *CD*.GROUP\_CONDITION\_TYPE = **'ĐKSP'**;

Lấy id hợp đồng, id khoản giải ngân, ngày giải ngân hợp đồng, ngày giải ngân khoản giải ngân, product\_code khoản giải ngân, ngày đến hạn của khoản giải ngân

#### Xử lý dữ liệu

Xử lý tương tự Job sinh kỳ theo điều kiện sản phẩm Khác

Lưu dữ liệu vừa khởi tạo vào bảng TBL\_PERIOD\_EVALUATION đồng thời

update bảng

STG\_SEAOPS.TBL\_BI\_DISBURSEMENT\_HISTORY, STG\_SEAOPS.TBL\_BI\_LC\_RELEASE\_SALES, STG\_SEAOPS.TBL\_BI\_GUARANTEE\_SALES

SET

CREATE\_CROSS\_SELLING\_PERIOD\_STATUS = ‘SUCCESS’

## Job sinh kỳ dựa theo Điều kiện sản phẩm Quan hệ tín dụng

### Mô tả nghiệp vụ

Job tự động sinh kỳ dựa theo dữ liệu lịch sử giản ngân và điều kiện sản phẩm Quan hệ tín dụng.

### Biểu đồ tuần tự

### Quy tắc xử lý dữ liệu

#### Lấy dữ liệu thỏa mãn điều kiện

**SELECT**

*DH*.CONTRACT\_REF *ref\_id*, *DH*.LOAN\_CODE *loan\_code*, *CRD*.VALUE\_DATE\_MIN *value\_date\_min\_ref*, *DH*.VALUE\_DATE *value\_date*, *DH*.PRODUCT\_CODE *product\_code*, *DH*. MATURITY\_DATE *maturity\_date*

**FROM**

TBL\_BI\_DISBURSEMENT\_HISTORY *DH*

**JOIN** TBL\_BI\_CONTRACT\_REF\_DEFINE *CRD*

**ON** *CRD*.REF\_ID = *DH*.CONTRACT\_REF

**JOIN** TBL\_CONTRACT\_CONDITION *CD*

**ON** *DH*.CONTRACT\_REF = *CD*.CONTRACT\_ID

**WHERE**

*DH*.DATA\_STATUS = **'ACTIVE'**

**and** *DH*.CREATE\_ CREDIT\_RELATION\_PERIOD\_STATUS = **null**

**and** *CD*.GROUP\_CONTRACT\_CONDITION = **'OTHER'**

**and** *CD*.GROUP\_CONDITION\_TYPE = ‘ĐKSP’

**UNION** **ALL**

**SELECT**

*DH*.CONTRACT\_REF *ref\_id*, *dh*.contract\_md *loan\_code*, *CRD*.VALUE\_DATE\_MIN *value\_date\_min\_ref*, *dh*.past\_sched\_date *value\_date*, *DH*.PRODUCT\_CODE *product\_code*, *DH*. MATURITY\_DATE *maturity\_date*

**FROM**

TBL\_BI\_GUARANTEE\_SALES *DH*

**JOIN** TBL\_BI\_CONTRACT\_REF\_DEFINE *CRD*

**ON** *CRD*.REF\_ID = *DH*.CONTRACT\_REF

**JOIN** TBL\_CONTRACT\_CONDITION\_BIPLUS *CD*

**ON** *DH*.CONTRACT\_REF = *CD*.CONTRACT\_ID

**WHERE**

*DH*.DATA\_STATUS = **'ACTIVE'**

**and** *DH*.CREATE\_CREDIT\_RELATION\_PERIOD\_STATUS = **null**

**and** *CD*.GROUP\_CONTRACT\_CONDITION = **'OTHER'**

**and** *CD*.GROUP\_CONDITION\_TYPE = **'ĐKSP'**

**UNION** **ALL**

**SELECT**

*DH*.CONTRACT\_REF *ref\_id*, *dh*.contract\_id *loan\_code*, *CRD*.VALUE\_DATE\_MIN *value\_date\_min\_ref*, *dh*.issue\_date *value\_date*, *DH*.PRODUCT\_CODE *product\_code*, *DH*. MATURITY\_DATE *maturity\_date*

**FROM**

TBL\_BI\_LC\_RELEASE\_SALES *DH*

**JOIN** TBL\_BI\_CONTRACT\_REF\_DEFINE\_BIPLUS *CRD*

**ON** *CRD*.REF\_ID = *DH*.CONTRACT\_REF

**JOIN** TBL\_CONTRACT\_CONDITION\_BIPLUS *CD*

**ON** *DH*.CONTRACT\_REF = *CD*.CONTRACT\_ID

**WHERE**

*DH*.DATA\_STATUS = **'ACTIVE'**

**and** *DH*.CREATE\_CREDIT\_RELATION\_PERIOD\_STATUS = **null**

**and** *CD*.GROUP\_CONTRACT\_CONDITION = **'OTHER'**

**and** *CD*.GROUP\_CONDITION\_TYPE = **'ĐKSP'**;

Lấy id hợp đồng, id khoản giải ngân, ngày giải ngân hợp đồng, ngày giải ngân khoản giải ngân, product\_code khoản giải ngân, ngày đến hạn của khoản giải ngân

#### Xử lý dữ liệu

Xử lý tương tự Job sinh kỳ theo điều kiện sản phẩm Khác

Lưu dữ liệu vừa khởi tạo vào bảng TBL\_PERIOD\_EVALUATION đồng thời

update bảng

STG\_SEAOPS.TBL\_BI\_DISBURSEMENT\_HISTORY, STG\_SEAOPS.TBL\_BI\_LC\_RELEASE\_SALES, STG\_SEAOPS.TBL\_BI\_GUARANTEE\_SALES

SET

CREATE\_CREDIT\_RELATION\_PERIOD\_STATUS = ‘SUCCESS’

## Job sinh kỳ dựa theo Điều kiện sản phẩm Tài chính

### Mô tả nghiệp vụ

Job tự động sinh kỳ dựa theo dữ liệu lịch sử giản ngân và điều kiện sản phẩm Tài chính.

### Biểu đồ tuần tự

### Quy tắc xử lý dữ liệu

#### Lấy dữ liệu thỏa mãn điều kiện

**SELECT**

*DH*.CONTRACT\_REF *ref\_id*, *DH*.LOAN\_CODE *loan\_code*, *CRD*.VALUE\_DATE\_MIN *value\_date\_min\_ref*, *DH*.VALUE\_DATE *value\_date*, *DH*.PRODUCT\_CODE *product\_code*, *DH*. MATURITY\_DATE *maturity\_date*

**FROM**

TBL\_BI\_DISBURSEMENT\_HISTORY *DH*

**JOIN** TBL\_BI\_CONTRACT\_REF\_DEFINE *CRD*

**ON** *CRD*.REF\_ID = *DH*.CONTRACT\_REF

**JOIN** TBL\_CONTRACT\_CONDITION *CD*

**ON** *DH*.CONTRACT\_REF = *CD*.CONTRACT\_ID

**WHERE**

*DH*.DATA\_STATUS = **'ACTIVE'**

**and** *DH*.CREATE\_FINANCIAL\_PERIOD\_STATUS = **null**

**and** *CD*.GROUP\_CONTRACT\_CONDITION = **'FINANCIAL'**

**and** *CD*.GROUP\_CONDITION\_TYPE = ‘ĐKSP’

**UNION** **ALL**

**SELECT**

*DH*.CONTRACT\_REF *ref\_id*, *dh*.contract\_md *loan\_code*, *CRD*.VALUE\_DATE\_MIN *value\_date\_min\_ref*, *dh*.past\_sched\_date *value\_date*, *DH*.PRODUCT\_CODE *product\_code*, *DH*. MATURITY\_DATE *maturity\_date*

**FROM**

TBL\_BI\_GUARANTEE\_SALES *DH*

**JOIN** TBL\_BI\_CONTRACT\_REF\_DEFINE *CRD*

**ON** *CRD*.REF\_ID = *DH*.CONTRACT\_REF

**JOIN** TBL\_CONTRACT\_CONDITION\_BIPLUS *CD*

**ON** *DH*.CONTRACT\_REF = *CD*.CONTRACT\_ID

**WHERE**

*DH*.DATA\_STATUS = **'ACTIVE'**

**and** *DH*.CREATE\_FINANCIAL\_PERIOD\_STATUS = **null**

**and** *CD*.GROUP\_CONTRACT\_CONDITION = **'FINANCIAL'**

**and** *CD*.GROUP\_CONDITION\_TYPE = **'ĐKSP'**

**UNION** **ALL**

**SELECT**

*DH*.CONTRACT\_REF *ref\_id*, *dh*.contract\_id *loan\_code*, *CRD*.VALUE\_DATE\_MIN *value\_date\_min\_ref*, *dh*.issue\_date *value\_date*, *DH*.PRODUCT\_CODE *product\_code*, *DH*. MATURITY\_DATE *maturity\_date*

**FROM**

TBL\_BI\_LC\_RELEASE\_SALES *DH*

**JOIN** TBL\_BI\_CONTRACT\_REF\_DEFINE\_BIPLUS *CRD*

**ON** *CRD*.REF\_ID = *DH*.CONTRACT\_REF

**JOIN** TBL\_CONTRACT\_CONDITION\_BIPLUS *CD*

**ON** *DH*.CONTRACT\_REF = *CD*.CONTRACT\_ID

**WHERE**

*DH*.DATA\_STATUS = **'ACTIVE'**

**and** *DH*.CREATE\_FINANCIAL\_PERIOD\_STATUS = **null**

**and** *CD*.GROUP\_CONTRACT\_CONDITION = **'FINANCIAL'**

**and** *CD*.GROUP\_CONDITION\_TYPE = **'ĐKSP'**;

Lấy id hợp đồng, id khoản giải ngân, ngày giải ngân hợp đồng, ngày giải ngân khoản giải ngân, product\_code khoản giải ngân, ngày đến hạn của khoản giải ngân

#### Xử lý dữ liệu

Xử lý tương tự Job sinh kỳ theo điều kiện sản phẩm Khác

Lưu dữ liệu vừa khởi tạo vào bảng TBL\_PERIOD\_EVALUATION đồng thời

update bảng

STG\_SEAOPS.TBL\_BI\_DISBURSEMENT\_HISTORY, STG\_SEAOPS.TBL\_BI\_LC\_RELEASE\_SALES, STG\_SEAOPS.TBL\_BI\_GUARANTEE\_SALES

SET

CREATE\_FINANCIAL\_PERIOD\_STATUS = ‘SUCCESS’

## Job sinh kỳ dựa theo Điều kiện sản phẩm Pháp lý

### Mô tả nghiệp vụ

Job tự động sinh kỳ dựa theo dữ liệu lịch sử giản ngân và điều kiện sản phẩm Pháp lý.

### Biểu đồ tuần tự

### Quy tắc xử lý dữ liệu

#### Lấy dữ liệu thỏa mãn điều kiện

**SELECT**

*DH*.CONTRACT\_REF *ref\_id*, *DH*.LOAN\_CODE *loan\_code*, *CRD*.VALUE\_DATE\_MIN *value\_date\_min\_ref*, *DH*.VALUE\_DATE *value\_date*, *DH*.PRODUCT\_CODE *product\_code*, *DH*. MATURITY\_DATE *maturity\_date*

**FROM**

TBL\_BI\_DISBURSEMENT\_HISTORY *DH*

**JOIN** TBL\_BI\_CONTRACT\_REF\_DEFINE *CRD*

**ON** *CRD*.REF\_ID = *DH*.CONTRACT\_REF

**JOIN** TBL\_CONTRACT\_CONDITION *CD*

**ON** *DH*.CONTRACT\_REF = *CD*.CONTRACT\_ID

**WHERE**

*DH*.DATA\_STATUS = **'ACTIVE'**

**and** *DH*.CREATE\_LAW\_PERIOD\_STATUS = **null**

**and** *CD*.GROUP\_CONTRACT\_CONDITION = **'LAW'**

**and** *CD*.GROUP\_CONDITION\_TYPE = ‘ĐKSP’

**UNION** **ALL**

**SELECT**

*DH*.CONTRACT\_REF *ref\_id*, *dh*.contract\_md *loan\_code*, *CRD*.VALUE\_DATE\_MIN *value\_date\_min\_ref*, *dh*.past\_sched\_date *value\_date*, *DH*.PRODUCT\_CODE *product\_code*, *DH*. MATURITY\_DATE *maturity\_date*

**FROM**

TBL\_BI\_GUARANTEE\_SALES *DH*

**JOIN** TBL\_BI\_CONTRACT\_REF\_DEFINE *CRD*

**ON** *CRD*.REF\_ID = *DH*.CONTRACT\_REF

**JOIN** TBL\_CONTRACT\_CONDITION\_BIPLUS *CD*

**ON** *DH*.CONTRACT\_REF = *CD*.CONTRACT\_ID

**WHERE**

*DH*.DATA\_STATUS = **'ACTIVE'**

**and** *DH*.CREATE\_LAW\_PERIOD\_STATUS = **null**

**and** *CD*.GROUP\_CONTRACT\_CONDITION = **'LAW'**

**and** *CD*.GROUP\_CONDITION\_TYPE = **'ĐKSP'**

**UNION** **ALL**

**SELECT**

*DH*.CONTRACT\_REF *ref\_id*, *dh*.contract\_id *loan\_code*, *CRD*.VALUE\_DATE\_MIN *value\_date\_min\_ref*, *dh*.issue\_date *value\_date*, *DH*.PRODUCT\_CODE *product\_code*, *DH*. MATURITY\_DATE *maturity\_date*

**FROM**

TBL\_BI\_LC\_RELEASE\_SALES *DH*

**JOIN** TBL\_BI\_CONTRACT\_REF\_DEFINE\_BIPLUS *CRD*

**ON** *CRD*.REF\_ID = *DH*.CONTRACT\_REF

**JOIN** TBL\_CONTRACT\_CONDITION\_BIPLUS *CD*

**ON** *DH*.CONTRACT\_REF = *CD*.CONTRACT\_ID

**WHERE**

*DH*.DATA\_STATUS = **'ACTIVE'**

**and** *DH*.CREATE\_LAW\_PERIOD\_STATUS = **null**

**and** *CD*.GROUP\_CONTRACT\_CONDITION = **'LAW'**

**and** *CD*.GROUP\_CONDITION\_TYPE = **'ĐKSP'**;

Lấy id hợp đồng, id khoản giải ngân, ngày giải ngân hợp đồng, ngày giải ngân khoản giải ngân, product\_code khoản giải ngân, ngày đến hạn của khoản giải ngân

#### Xử lý dữ liệu

Xử lý tương tự Job sinh kỳ theo điều kiện sản phẩm Khác

Lưu dữ liệu vừa khởi tạo vào bảng TBL\_PERIOD\_EVALUATION đồng thời

update bảng

STG\_SEAOPS.TBL\_BI\_DISBURSEMENT\_HISTORY, STG\_SEAOPS.TBL\_BI\_LC\_RELEASE\_SALES, STG\_SEAOPS.TBL\_BI\_GUARANTEE\_SALES

SET

CREATE\_LAW\_PERIOD\_STATUS = ‘SUCCESS’

## Job sinh kỳ dựa theo Điều kiện sản phẩm Bảo hiểm nhân thọ

### Mô tả nghiệp vụ

Job tự động sinh kỳ dựa theo dữ liệu lịch sử giản ngân và điều kiện sản phẩm Bảo hiểm nhân thọ.

### Biểu đồ tuần tự

### Quy tắc xử lý dữ liệu

#### Lấy dữ liệu thỏa mãn điều kiện

**SELECT**

*DH*.CONTRACT\_REF *ref\_id*, *DH*.LOAN\_CODE *loan\_code*, *CRD*.VALUE\_DATE\_MIN *value\_date\_min\_ref*, *DH*.VALUE\_DATE *value\_date*, *DH*.PRODUCT\_CODE *product\_code*, *DH*. MATURITY\_DATE *maturity\_date*

**FROM**

TBL\_BI\_DISBURSEMENT\_HISTORY *DH*

**JOIN** TBL\_BI\_CONTRACT\_REF\_DEFINE *CRD*

**ON** *CRD*.REF\_ID = *DH*.CONTRACT\_REF

**JOIN** TBL\_CONTRACT\_CONDITION *CD*

**ON** *DH*.CONTRACT\_REF = *CD*.CONTRACT\_ID

**WHERE**

*DH*.DATA\_STATUS = **'ACTIVE'**

**and** *DH*.CREATE\_LIFE\_INSURANCE\_PERIOD\_STATUS = **null**

**and** *CD*.GROUP\_CONTRACT\_CONDITION = **'LIFE\_INSURANCE'**

**and** *CD*.GROUP\_CONDITION\_TYPE = ‘ĐKSP’

**UNION** **ALL**

**SELECT**

*DH*.CONTRACT\_REF *ref\_id*, *dh*.contract\_md *loan\_code*, *CRD*.VALUE\_DATE\_MIN *value\_date\_min\_ref*, *dh*.past\_sched\_date *value\_date*, *DH*.PRODUCT\_CODE *product\_code*, *DH*. MATURITY\_DATE *maturity\_date*

**FROM**

TBL\_BI\_GUARANTEE\_SALES *DH*

**JOIN** TBL\_BI\_CONTRACT\_REF\_DEFINE *CRD*

**ON** *CRD*.REF\_ID = *DH*.CONTRACT\_REF

**JOIN** TBL\_CONTRACT\_CONDITION\_BIPLUS *CD*

**ON** *DH*.CONTRACT\_REF = *CD*.CONTRACT\_ID

**WHERE**

*DH*.DATA\_STATUS = **'ACTIVE'**

**and** *DH*.CREATE\_LIFE\_INSURANCE\_PERIOD\_STATUS = **null**

**and** *CD*.GROUP\_CONTRACT\_CONDITION = **'LIFE\_INSURANCE'**

**and** *CD*.GROUP\_CONDITION\_TYPE = **'ĐKSP'**

**UNION** **ALL**

**SELECT**

*DH*.CONTRACT\_REF *ref\_id*, *dh*.contract\_id *loan\_code*, *CRD*.VALUE\_DATE\_MIN *value\_date\_min\_ref*, *dh*.issue\_date *value\_date*, *DH*.PRODUCT\_CODE *product\_code*, *DH*. MATURITY\_DATE *maturity\_date*

**FROM**

TBL\_BI\_LC\_RELEASE\_SALES *DH*

**JOIN** TBL\_BI\_CONTRACT\_REF\_DEFINE\_BIPLUS *CRD*

**ON** *CRD*.REF\_ID = *DH*.CONTRACT\_REF

**JOIN** TBL\_CONTRACT\_CONDITION\_BIPLUS *CD*

**ON** *DH*.CONTRACT\_REF = *CD*.CONTRACT\_ID

**WHERE**

*DH*.DATA\_STATUS = **'ACTIVE'**

**and** *DH*.CREATE\_LIFE\_INSURANCE\_PERIOD\_STATUS = **null**

**and** *CD*.GROUP\_CONTRACT\_CONDITION = **'LIFE\_INSURANCE'**

**and** *CD*.GROUP\_CONDITION\_TYPE = **'ĐKSP'**;

Lấy id hợp đồng, id khoản giải ngân, ngày giải ngân hợp đồng, ngày giải ngân khoản giải ngân, product\_code khoản giải ngân, ngày đến hạn của khoản giải ngân

#### Xử lý dữ liệu

Xử lý tương tự Job sinh kỳ theo điều kiện sản phẩm Khác

Lưu dữ liệu vừa khởi tạo vào bảng TBL\_PERIOD\_EVALUATION đồng thời

update bảng

STG\_SEAOPS.TBL\_BI\_DISBURSEMENT\_HISTORY, STG\_SEAOPS.TBL\_BI\_LC\_RELEASE\_SALES, STG\_SEAOPS.TBL\_BI\_GUARANTEE\_SALES

SET

CREATE\_LFFE\_INSURANCE\_PERIOD\_STATUS = ‘SUCCESS’

## Job sinh kỳ dựa theo Điều kiện Doanh thu

### Mô tả nghiệp vụ

Job tự động sinh kỳ dựa theo dữ liệu lịch sử giản ngân và điều kiện doanh thu (ĐK Sản phẩm và ĐK Nghị quyết)

### Biểu đồ tuần tự

### Quy tắc xử lý dữ liệu

#### Lấy dữ liệu thỏa mãn điều kiện

**SELECT**

*DH*.CONTRACT\_REF *ref\_id*, *DH*.LOAN\_CODE *loan\_code*, *CRD*.VALUE\_DATE\_MIN *value\_date\_min\_ref*, *DH*.VALUE\_DATE *value\_date*, *DH*.PRODUCT\_CODE *product\_code*, *DH*. MATURITY\_DATE *maturity\_date*

**FROM**

TBL\_BI\_DISBURSEMENT\_HISTORY *DH*

**JOIN** TBL\_BI\_CONTRACT\_REF\_DEFINE *CRD*

**ON** *CRD*.REF\_ID = *DH*.CONTRACT\_REF

**JOIN** TBL\_REVENUE\_CONDITION *CD*

**ON** *DH*.CONTRACT\_REF = *CD*.CONTRACT\_ID

**WHERE**

*DH*.DATA\_STATUS = **'ACTIVE'**

**and** *DH*.CREATE\_REVENUE\_PERIOD\_STATUS = **null**

**and** *CD*.GROUP\_CONTRACT\_CONDITION = **'REVENUE'**

**UNION** **ALL**

**SELECT**

*DH*.CONTRACT\_REF *ref\_id*, *dh*.contract\_md *loan\_code*, *CRD*.VALUE\_DATE\_MIN *value\_date\_min\_ref*, *dh*.past\_sched\_date *value\_date*, *DH*.PRODUCT\_CODE *product\_code*, *DH*. MATURITY\_DATE *maturity\_date*

**FROM**

TBL\_BI\_GUARANTEE\_SALES *DH*

**JOIN** TBL\_BI\_CONTRACT\_REF\_DEFINE *CRD*

**ON** *CRD*.REF\_ID = *DH*.CONTRACT\_REF

**JOIN** TBL\_REVENUE\_CONDITION *CD*

**ON** *DH*.CONTRACT\_REF = *CD*.CONTRACT\_ID

**WHERE**

*DH*.DATA\_STATUS = **'ACTIVE'**

**and** *DH*.CREATE\_REVENUE\_PERIOD\_STATUS = **null**

**and** *CD*.GROUP\_CONTRACT\_CONDITION = **'REVENUE'**

**UNION** **ALL**

**SELECT**

*DH*.CONTRACT\_REF *ref\_id*, *dh*.contract\_id *loan\_code*, *CRD*.VALUE\_DATE\_MIN *value\_date\_min\_ref*, *dh*.issue\_date *value\_date*, *DH*.PRODUCT\_CODE *product\_code*, *DH*. MATURITY\_DATE *maturity\_date*

**FROM**

TBL\_BI\_LC\_RELEASE\_SALES *DH*

**JOIN** TBL\_BI\_CONTRACT\_REF\_DEFINE\_BIPLUS *CRD*

**ON** *CRD*.REF\_ID = *DH*.CONTRACT\_REF

**JOIN** TBL\_REVENUE\_CONDITION *CD*

**ON** *DH*.CONTRACT\_REF = *CD*.CONTRACT\_ID

**WHERE**

*DH*.DATA\_STATUS = **'ACTIVE'**

**and** *DH*.CREATE\_REVENUE\_PERIOD\_STATUS = **null**

**and** *CD*.GROUP\_CONTRACT\_CONDITION = **'REVENUE'**

**and** *CD*.GROUP\_CONDITION\_TYPE = **'ĐKSP'**;

Lấy id hợp đồng, id khoản giải ngân, ngày giải ngân hợp đồng, ngày giải ngân khoản giải ngân, product\_code khoản giải ngân, ngày đến hạn của khoản giải ngân

#### Xử lý dữ liệu

Dữ liệu lấy ta từ bước trên được xử lý theo từng Row (contract) theo quy tắc sau :

B1 : Lấy dữ liệu tương ứng làm điều kiện đánh giá

* Dữ liệu điều kiện Doanh thu đã duyệt, tương ứng với hợp đồng:

condition = SELECT

\*

FROM

TBL\_CONTRACT\_CONDITION

WHERE

GROUP\_CONTRACT\_CONDITION = 'REVENUE'

AND STATUS = 'APPROVED'

AND CONTRACT\_ID = ref\_id; // ref\_id lấy từ dữ liệu select đầu vào

B2 : Dựa vào thông tin của condition bước 1

- Nếu condition.METHOD\_OF\_PERIOD = METHOD\_PERIOD\_F1:

Kiểm tra đã có hạng mục sinh ra bởi Điều kiện và Hợp đồng. Nếu đã có rồi thì không sinh hạng mục nữa.

**SELECT** \* **FROM** TBL\_PERIOD\_EVALUATION **WHERE** CONTRACT\_CONDITION\_ID = condition.ID **AND** CONTRACT\_ID = ref\_id; // Tồn tại bản ghi thì không sinh hạng mục

Nếu chưa, thực hiện sinh hạng mục, tính toán ngày đến hạn kiểm tra

EVALUATION\_DATE = value\_date\_min\_ref + condition.deadline \* condition.deadline\_unit

condition.deadline là số

condition.deadline\_unit = D(ngày) hoặc M(tháng) hoặc LAST\_DAY (cùng cách tính đơn vị tháng, lấy ngày cuối cùng của tháng)

Ví dụ: value\_date\_min\_ref = 23/07/2024

condition.deadline = 1, condition.deadline\_unit = D, EVALUATION\_DATE = 24/07/2024

condition.deadline = 1, condition.deadline\_unit = M, EVALUATION\_DATE = 23/08/2024

condition.deadline = 1, condition.deadline\_unit = LAST\_DAY, EVALUATION\_DATE = 31/08/2024

- Nếu condition.METHOD\_OF\_PERIOD = METHOD\_PERIOD\_F2:

Nếu condition.group\_condition\_type = DKSP, thì cần kiểm tra product\_code của khoản giải ngân và điều kiện.

Kiểm tra product\_code của điều kiện và khoản giải ngân trùng nhau không, không trùng thì không sinh hạng mục:

Nếu condition. LOAN\_PRODUCT\_ID != product\_code (product\_code lấy từ dữ liệu đầu vào ban đầu) 🡺 Không sinh hạng mục

Ngược lại (condition.group\_condition\_type != DKSP, hoặc LOAN\_PRODUCT\_ID = product\_code), thực hiện sinh hạng mục, tính toán ngày đến hạn kiểm tra

EVALUATION\_DATE = value\_date + condition.deadline \* condition.deadline\_unit

- Nếu condition.METHOD\_OF\_PERIOD = METHOD\_PERIOD\_F3:

Nếu condition.group\_condition\_type = DKSP, thì cần kiểm tra product\_code của khoản giải ngân và điều kiện.

Kiểm tra product\_code của điều kiện và khoản giải ngân trùng nhau không, không trùng thì không sinh hạng mục:

Nếu condition. LOAN\_PRODUCT\_ID != product\_code (product\_code lấy từ dữ liệu đầu vào ban đầu) 🡺 Không sinh hạng mục

Kiểm tra khoản giải ngân này có phải là khoản giải ngân đầu tiên không? Nếu không phải khoản giải ngân đầu tiên, không sinh hạng mục:

**SELECT** **min**(*VALUE\_DATE*) min\_value\_date **FROM** (

**SELECT** VALUE\_DATE *VALUE\_DATE*

**FROM**

STG\_SEAOPS.TBL\_BI\_DISBURSEMENT\_HISTORY *DH*

**WHERE** *DH*.CONTRACT\_REF = ref\_id **AND** *DH*.PRODUCT\_CODE = product\_code

**UNION** **ALL**

**SELECT** ISSUE\_DATE *VALUE\_DATE*

**FROM**

STG\_SEAOPS.TBL\_BI\_LC\_RELEASE\_SALES *LC*

**WHERE** *LC*.CONTRACT\_REF = ref\_id **AND** *LC*.PRODUCT\_CODE = product\_code

**UNION** **ALL**

**SELECT** PAST\_SCHED\_DATE *VALUE\_DATE*

**FROM**

STG\_SEAOPS.TBL\_BI\_GUARANTEE\_SALES *GS*

**WHERE** *GS*.CONTRACT\_REF = ref\_id **AND** *GS*.PRODUCT\_CODE = product\_code

); // nếu min\_value\_date < value\_date dữ liệu đầu vào thì không sinh hạng mục

Kiểm tra đã sinh hạng mục với product\_code của khoản giải ngân này chưa. Nếu sinh hạng mục rồi, thì không sinh hạng mục nữa.

**SELECT** \* **FROM** TBL\_PERIOD\_EVALUATION **WHERE** CONTRACT\_CONDITION\_ID = **condition**.ID

**AND** CONTRACT\_ID = ref\_id

**AND** LOAN\_PRODUCT\_ID = product\_code; // Tồn tại bản ghi thì không sinh hạng mục

Tính ngày đến hạn kiểm tra:

EVALUATION\_DATE = min\_value\_date + condition.deadline \* condition.deadline\_unit

- Nếu condition.METHOD\_OF\_PERIOD = METHOD\_PERIOD\_F4:

Tính ngày đến hạn kiểm tra:

EVALUATION\_DATE = maturity\_date - condition.deadline \* condition.deadline\_unit

B2 : Tạo dữ liệu kỳ đánh giá

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên trường** | **Giá trị** |
| 1 | ID | RandomString |
| 2 | PERIOD | CASE WHEN condition.START\_PERIOD is not null THEN condition.START\_PERIOD ELSE 1 END |
| 3 | FREQUENCY | CASE WHEN condition.FREQUENCY is not null THEN condition.FREQUENCY ELSE 0 END |
| 4 | FREQUENCY\_UNIT | condition.FREQUENCY\_UNIT |
| 5 | AUTO\_PERIOD | CASE WHEN condition. AUTO\_PERIOD is not null THEN condition. AUTO\_PERIOD ELSE false END |
| 6 | BUSINESS\_UNIT\_NOTE | CASE WHEN condition. BUSINESS\_UNIT\_NOTE is not null THEN condition. BUSINESS\_UNIT\_NOTE ELSE false END |
| 7 | CONTENT | condition.CONTENT |
| 8 | CONTROL\_METHOD | condition. CONTROL\_METHOD |
| 9 | CONTRACT\_CONDITION\_ID | condition. CONTRACT\_CONDITION\_ID |
| 10 | CONTRACT\_CONDITION\_STATUS | condition. CONTRACT\_CONDITION\_STATUS |
| 11 | FIRST\_DATE\_CHOSEN\_PERIOD | condition. FIRST\_DATE\_CHOSEN\_PERIOD |
| 12 | DESCRIPTION | condition. DESCRIPTION |
| 13 | CONTRACT\_ID | contract.REF\_ID |
| 14 | CONTRACT\_STATUS | contract.STATUS |
| 15 | CUSTOMER\_ID | contract.CUSTOMER\_ID |
| 16 | CUSTOMER\_CLASS | contract. CUSTOMER\_CLASS |
| 17 | CO\_CODE | contract.CO\_CODE |
| 18 | COMPANY\_NAME\_VN | contract. COMPANY\_NAME\_VN |
| 19 | SHORT\_NAME | contract. SHORT\_NAME |
| 20 | LOAN\_CODE | contract. CONTRACT |
| 21 | REVOLVING | contract. REVOLVING |
| 22 | VALUE\_DATE\_MIN | contract.VALUE\_DATE |
| 23 | LOAN\_PRODUCT\_ID | contract. LOAN\_PRODUCT\_ID |
| 24 | CREATE\_NEXT\_PERIOD\_STATUS | 'NEW' |
| 25 | CREATED\_DATE | SYSDATE |
| 26 | CREATED\_USER | 'SYSTEM\_USER' |
| 27 | EVALUATION\_DATE | EVALUATION\_DATE đã tính ở trên |
| 28 | AUTO\_EVALUATION\_RESULT | CASE WHEN evalDate < SYSDATE THEN NOT\_FOLLOW ELSE null END |
| 29 | EVALUATION\_RESULT | CASE WHEN evalDate < SYSDATE THEN NOT\_FOLLOW ELSE null END |
| 30 | GROUP\_CONTRACT\_CONDITION | REVENUE |
| 31 | GROUP\_CONDITION\_TYPE | condition.GROUP\_CONDITION\_TYPE |
| 32 | REVENUE\_MONEY\_1 |  |
| 33 | REVENUE\_MONEY\_2 |  |

Lưu dữ liệu vừa khởi tạo vào bảng TBL\_PERIOD\_EVALUATION đồng thời

update bảng

STG\_SEAOPS.TBL\_BI\_DISBURSEMENT\_HISTORY, STG\_SEAOPS.TBL\_BI\_LC\_RELEASE\_SALES, STG\_SEAOPS.TBL\_BI\_GUARANTEE\_SALES

SET

CREATE\_REVENUE\_PERIOD\_STATUS = ‘SUCCESS’

Nếu ControlMethod = BSCT, lấy RuleNumber, MoneyOnDocument trong bảng REVENUE\_CONFIGURATION theo revenueConditionId = condition.Id

RevenueMoney1 = MoneyOnDocument nếu RuleNumber = RULE\_NUMBER\_1

RevenueMoney2 = MoneyOnDocument nếu RuleNumber = RULE\_NUMBER\_2

## Job sinh kỳ dựa theo khoản giải tỏa tiền tạm ứng

### Mô tả nghiệp vụ

Job tự động sinh kỳ dựa theo dữ liệu lịch sử giải tỏa tiền tạm ứng và điều kiện nghị quyết GTTTU hoặc danh mục ĐKSP hạng mục GTTTU nếu chưa khai báo DKNQ GTTTU.

Nếu có ĐKNQ sẽ ưu tiên lấy thông tin từ ĐKNQ

### Biểu đồ tuần tự

### Quy tắc xử lý dữ liệu

#### Lấy dữ liệu thỏa mãn điều kiện

Lấy dữ liệu từ bảng mới lịch sử giải tỏa tiền tạm ứng

sub\_MD = SELECT \* FROM TBL\_ADVANCE\_PAYMENT\_RELEASE WHERE CREATE\_PERIOD\_STATUS = NULL OR CREATE\_PERIOD\_STATUS IN (‘NEW’,’RENEW’);

#### Xử lý dữ liệu

Dữ liệu lấy ta từ bước trên được xử lý theo từng Row (sub\_MD) theo quy tắc sau :

B1 : Lấy dữ liệu tương ứng làm điều kiện đánh giá

Với từng khoản giải tỏa TTU, kiểm tra trong bảng lịch sử giải ngân bảo lãnh có MD tương ứng.

Nếu không có dữ liệu, không xử lý tiếp, update CREATE\_PERIOD\_STATUS = SUCCESS

MD = SELECT \* FROM TBL\_BI\_GUARANTEE\_SALES WHERE CONTRACT\_MD = sub\_MD.CONTRACT\_MD;

Lấy điều kiện nghị quyết GTTTU:

condition = SELECT

\*

FROM

TBL\_CONTRACT\_CONDITION

WHERE

GROUP\_CONTRACT\_CONDITION = 'GTTTU'

AND STATUS = 'APPROVED'

AND CONTRACT\_ID = MD.CONTRACT\_REF; ---- ref\_id lấy từ MD ở trên

Nếu không có ĐKNQ GTTTU, lấy danh mục ĐKSP hạng mục GTTTU:

productCondition = SELECT

\*

FROM

TBL\_LOAN\_PRODUCT\_CONDITION

WHERE

CATEGORY = ‘ADVANCE\_RELEASE\_CONDITION’

AND PRODUCT\_TYPE = MD.PRODUCT\_CODE --- Lấy các bản ghi có cùng product\_code với MD

Nếu không có ĐKNQ và danh mục, thì không xử lý tiếp, update CREATE\_PERIOD\_STATUS = SUCCESS

B2 : Tạo dữ liệu kỳ đánh giá

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên trường** | **Giá trị** |
| 1 | ID | RandomString |
| 2 | PERIOD | CASE WHEN condition.START\_PERIOD is not null THEN condition.START\_PERIOD ELSE 1 END |
| 3 | FREQUENCY | CASE WHEN condition.FREQUENCY is not null THEN condition. FREQUENCY ELSE 0 END |
| 4 | FREQUENCY\_UNIT | condition. FREQUENCY \_UNIT |
| 5 | AUTO\_PERIOD | CASE WHEN condition. AUTO\_PERIOD is not null THEN condition. AUTO\_PERIOD ELSE false END |
| 6 | BUSINESS\_UNIT\_NOTE | CASE WHEN condition. BUSINESS\_UNIT\_NOTE is not null THEN condition. BUSINESS\_UNIT\_NOTE ELSE false END |
| 7 | CONTENT | condition.CONTENT |
| 8 | CONTROL\_MENTHOD | condition. CONTROL\_MENTHOD |
| 9 | CONTRACT\_CONDITION\_ID | condition. CONTRACT\_CONDITION\_ID |
| 10 | CONTRACT\_CONDITION\_STATUS | condition. CONTRACT\_CONDITION\_STATUS |
| 11 | FIRST\_DATE\_CHOSEN\_PERIOD | condition. FIRST\_DATE\_CHOSEN\_PERIOD |
| 12 | DESCRIPTION | condition. DESCRIPTION |
| 13 | CONTRACT\_ID | contract.REF\_ID |
| 14 | CONTRACT\_STATUS | contract.STATUS |
| 15 | CUSTOMER\_ID | contract.CUSTOMER\_ID |
| 16 | CUSTOMER\_CLASS | contract. CUSTOMER\_CLASS |
| 17 | CO\_CODE | contract.CO\_CODE |
| 18 | COMPANY\_NAME\_VN | contract. COMPANY\_NAME\_VN |
| 19 | SHORT\_NAME | contract. SHORT\_NAME |
| 20 | LOAN\_CODE | contract. CONTRACT |
| 21 | REVOLVING | contract. REVOLVING |
| 22 | VALUE\_DATE\_MIN | contract.VALUE\_DATE |
| 23 | LOAN\_PRODUCT\_ID | contract. LOAN\_PRODUCT\_ID |
| 24 |  |  |
| 25 |  |  |
| 26 | CREATE\_NEXT\_PERIOD\_STATUS | 'NEW' |
| 27 | CREATED\_DATE | SYSDATE |
| 28 | CREATED\_USER | 'SYSTEM\_USER' |
| 29 | EVALUATION\_DATE | evalDate = sub\_MD. DATE\_TIME\_AUTHORISE ( dữ liệu bản ghi gốc) + condition.CHECK\_TERM (DAYS)  + Nếu condition is null 🡺 evalDate = sub\_MD. DATE\_TIME\_AUTHORISE ( dữ liệu bản ghi gốc) + productCondition.DEADLINE \* productCondition.DEADLINE\_UNIT |
| 30 | AUTO\_EVALUATION\_RESULT | CASE WHEN evalDate < SYSDATE THEN NOT\_FOLLOW ELSE null END |
| 31 | EVALUATION\_RESULT | CASE WHEN evalDate < SYSDATE THEN NOT\_FOLLOW ELSE null END |
| 32 | GROUP\_CONTRACT\_CONDITION | ADVANCE\_PAYMENT\_RELEASE |
| 33 | GROUP\_CONDITION\_TYPE | DKNQ : Nếu condition có GROUP\_CONDITION\_TYPE = null hoặc DKNQ  DKSP: Nếu condition có GROUP\_CONDITION\_TYPE = DKSP  QDC: Nếu condition null |

Lưu dữ liệu vừa khởi tạo vào bảng TBL\_PERIOD\_EVALUATION đồng thời update bảng TBL\_ADVANCE\_PAYMENT\_RELEASE SET CREATE\_PERIOD\_STATUS = ‘SUCCESS’