**TÀI LIỆU MÔ TẢ CHUYỂN ĐỔI STORE #33 (BẢN CHỈNH)**

RPT.NAPAS\_MASTER\_VIEW\_DOMESTIC\_IBFT → TiDB/Java

**1. Tên store**

RPT.NAPAS\_MASTER\_VIEW\_DOMESTIC\_IBFT (file #33.NAPAS\_MASTER\_VIEW\_DOMESTIC\_IBFT.prc).

**1.1 Phân tích store (chuẩn hoá theo store gốc)**

1. Step 1: Ghi log bắt đầu (ERR\_EX).
2. Step 2: Gọi thủ tục con INSERT\_TCKT\_SESSION\_DOMESTIC\_IBFT(pQRY\_FROM\_DATE, pQRY\_TO\_DATE, pUser) để tổng hợp dữ liệu IBFT vào bảng tạm TCKT\_SESSION\_DOMESTIC.
3. Step 3: Reset các cột phí/điều chỉnh về 0 ở TCKT\_SESSION\_DOMESTIC (nếu có nhánh này trong store gốc).
4. Step 4: Xoá dữ liệu cũ ở bảng đích (ví dụ TCKT\_NAPAS\_IBFT) theo khoảng ngày / tham số chạy để tránh trùng.
5. Step 5: Insert từ bảng tạm TCKT\_SESSION\_DOMESTIC sang TCKT\_NAPAS\_IBFT (ISS-ACQ); chuẩn hoá SETT\_DATE/EDIT\_DATE; WITH\_BANK theo điều kiện Pcode2 & TGTT\_CONFIG.
6. Step 6: Tính CD\_TOTAL\_MONEY / DB\_TOTAL\_MONEY / NAPAS\_FEE / DEBIT / CREDIT… (các bước tính toán tiếp theo tuỳ store gốc, đánh lại số step cho phù hợp).
7. Step x: Cập nhật BANK\_NAME bằng JOIN danh mục (thay GET\_FULL\_BANK\_NAME).
8. Step y: Xoá-ghi NAPAS\_FEE\_MONTH & AUTO\_BACKUP (nếu thuộc phạm vi store này).
9. Step z-1: Ghi log kết thúc (ERR\_EX).
10. Step z-2: Ghi log lỗi (ERR\_EX) trong khối EXCEPTION.

**1.2 Bảng thay thế hàm Oracle → TiDB/Java (áp dụng cho store này)**

|  |  |
| --- | --- |
| Hàm/biểu thức Oracle | Thay thế (TiDB/Java) |
| TO\_DATE(str,'dd/MM/yyyy') | STR\_TO\_DATE(str, '%d/%m/%Y') (TiDB) |
| TO\_CHAR(dt,'fmt') | DATE\_FORMAT(dt, 'fmt MySQL') |
| SYSDATE | NOW() |
| TRUNC(SYSDATE) | CURDATE() |
| TRUNC(dt) | DATE(dt) hoặc DATE\_FORMAT(dt,'%Y-%m-%d') |
| NVL(a,b) | IFNULL(a,b) |
| DECODE(...) | CASE WHEN ... THEN ... ELSE ... END |
| SUBSTR(s, pos, len) | SUBSTRING(s, pos, len) |
| LISTAGG(... WITHIN GROUP ...) | GROUP\_CONCAT(... ORDER BY ... SEPARATOR ...) |
| STANDARD\_HASH(str,'MD5') | MD5(str) |
| ROWNUM | ORDER BY ... LIMIT ... (khi cần); hoặc dùng ROW\_NUMBER() OVER() trong CTE khi cập nhật theo thứ hạng |
| GET\_FULL\_BANK\_NAME(bank\_id) | UPDATE ... JOIN BANK\_DICT b ON t.BANK\_ID=b.BANK\_ID SET t.BANK\_NAME = REPLACE(b.FULL\_NAME,'Ngân hàng','NH') |
| GET\_QRC\_WITH(...) | CASE WHEN Pcode2 IN (720000,730000,890000) THEN (CASE WHEN NOT EXISTS(SELECT 1 FROM TGTT\_CONFIG t WHERE t.TGTT\_ID=A.BB\_BIN) THEN A.BB\_BIN ELSE 0 END) ELSE 0 END |

**1.3 Câu lệnh thay thế cho từng step**

Trong phần này, Step 2 giữ nguyên ý nghĩa “gọi store con”, nhưng khi chuyển sang Java/Spring bạn có hai cách: (A) tiếp tục gọi stored procedure TiDB tương đương; (B) chuyển toàn bộ logic của thủ tục con thành SQL/Java. Dưới đây là khuyến nghị:

**Step 1 – Log bắt đầu (SQL TiDB)**

INSERT INTO ERR\_EX(ERR\_TIME, ERR\_CODE, ERR\_DETAIL, ERR\_MODULE)  
VALUES (NOW(), '0', 'Start', 'NAPAS\_MASTER\_VIEW\_DOMESTIC\_IBFT');

**Step 2 – Gọi thủ tục con `INSERT\_TCKT\_SESSION\_DOMESTIC\_IBFT`**

-- Cách A: nếu vẫn để logic trong DB (TiDB hỗ trợ procedure cơ bản)  
CALL INSERT\_TCKT\_SESSION\_DOMESTIC\_IBFT(:pQRY\_FROM\_DATE, :pQRY\_TO\_DATE, :pUser);  
  
-- Cách B (khuyến nghị “chuyển lên Java”): triển khai service Java chạy các INSERT/UPDATE  
-- để đổ dữ liệu IBFT vào TCKT\_SESSION\_DOMESTIC. (Đính kèm tài liệu thủ tục con riêng)

**Step 3 – Reset phí/điều chỉnh (nếu store gốc có)**

UPDATE TCKT\_SESSION\_DOMESTIC  
SET DB\_IR\_FEE = 0, DB\_SV\_FEE = 0, DB\_TOTAL\_FEE = 0, DB\_TOTAL\_MONEY = 0,  
 CD\_IR\_FEE = 0, CD\_SV\_FEE = 0, CD\_TOTAL\_MONEY = 0, NAPAS\_FEE = 0,  
 ADJ\_FEE = 0, BC\_NP\_ADJ = 0, BC\_NP\_SUM = 0  
WHERE SETT\_DATE BETWEEN :sett\_from AND :sett\_to;

**Step 4 – Xoá dữ liệu cũ ở bảng đích (tránh trùng)**

DELETE FROM TCKT\_NAPAS\_IBFT  
WHERE SETT\_DATE BETWEEN :sett\_from AND :sett\_to  
 AND SERVICE\_TYPE='IBFT';

**Step 5 – Insert từ TCKT\_SESSION\_DOMESTIC sang TCKT\_NAPAS\_IBFT (ISS-ACQ)**

INSERT INTO TCKT\_NAPAS\_IBFT(  
 MSGTYPE\_DETAIL, SUB\_BANK, SETT\_DATE, EDIT\_DATE, SETTLEMENT\_CURRENCY, RESPCODE,  
 GROUP\_TRAN, PCODE, TRAN\_TYPE, SERVICE\_CODE, GROUP\_ROLE, BANK\_ID, WITH\_BANK, BANK\_NAME,  
 DB\_TOTAL\_TRAN, DB\_AMOUNT, DB\_IR\_FEE, DB\_SV\_FEE, DB\_TOTAL\_FEE, DB\_TOTAL\_MONEY,  
 CD\_TOTAL\_TRAN, CD\_AMOUNT, CD\_IR\_FEE, CD\_SV\_FEE, CD\_TOTAL\_MONEY, NAPAS\_FEE,  
 ADJ\_FEE, NP\_ADJ\_FEE, MERCHANT\_TYPE, BC\_NP\_SUM, BC\_CL\_ADJ, STEP, FEE\_TYPE, PART\_FE, LIQUIDITY  
)  
SELECT  
 A.MSGTYPE\_DETAIL,  
 CASE WHEN A.ISSUER\_RP = 970426 AND SUBSTRING(TRIM(A.PAN),1,8)='97046416' THEN 970464 ELSE NULL END AS SUB\_BANK,  
 CASE  
 WHEN A.Respcode = 0 AND A.SETTLEMENT\_DATE < :sett\_from THEN :sett\_from  
 WHEN A.Respcode = 0 AND A.SETTLEMENT\_DATE > :sett\_to THEN :sett\_to  
 WHEN A.Respcode = 0 AND A.SETTLEMENT\_DATE BETWEEN :sett\_from AND :sett\_to THEN A.SETTLEMENT\_DATE  
 ELSE NULL  
 END AS SETT\_DATE,  
 CASE  
 WHEN A.Respcode = 0 THEN NULL  
 ELSE CASE WHEN DATE(A.Edit\_Date) < :sett\_from THEN :sett\_from  
 WHEN DATE(A.Edit\_Date) > :sett\_to THEN :sett\_to  
 ELSE DATE(A.Edit\_Date) END  
 END AS EDIT\_DATE,  
 A.SETTLEMENT\_CURRENCY, A.RESPCODE, A.GROUP\_TRAN, A.PCODE, A.TRAN\_TYPE, A.SERVICE\_CODE,  
 A.GROUP\_ROLE, A.BANK\_ID,  
 CASE  
 WHEN A.Pcode2 IN (720000,730000,890000) THEN  
 CASE WHEN NOT EXISTS (SELECT 1 FROM TGTT\_CONFIG t WHERE t.TGTT\_ID = A.BB\_BIN) THEN A.BB\_BIN ELSE 0 END  
 ELSE 0  
 END AS WITH\_BANK,  
 NULL AS BANK\_NAME, -- sẽ cập nhật Step 14  
 A.DB\_TOTAL\_TRAN, A.DB\_AMOUNT, A.DB\_IR\_FEE, A.DB\_SV\_FEE, 0 AS DB\_TOTAL\_FEE, 0 AS DB\_TOTAL\_MONEY,  
 A.CD\_TOTAL\_TRAN, A.CD\_AMOUNT, A.CD\_IR\_FEE, A.CD\_SV\_FEE, 0 AS CD\_TOTAL\_MONEY,  
 0 AS NAPAS\_FEE, A.ADJ\_FEE, A.NP\_ADJ\_FEE, A.MERCHANT\_TYPE, A.BC\_NP\_SUM, A.BC\_CL\_ADJ,  
 A.STEP, A.FEE\_TYPE, A.PART\_FE, 'N' AS LIQUIDITY  
FROM TCKT\_SESSION\_DOMESTIC A  
WHERE A.SETT\_DATE BETWEEN :sett\_from AND :sett\_to;

**Các bước tính toán tiếp theo (ví dụ Step 6+): NAPAS\_FEE, DEBIT/CREDIT…**

-- Ví dụ mẫu (điều chỉnh theo store gốc):  
UPDATE TCKT\_NAPAS\_IBFT  
SET DB\_TOTAL\_FEE = DB\_IR\_FEE + DB\_SV\_FEE + CD\_SV\_FEE,  
 DB\_TOTAL\_MONEY = DB\_TOTAL\_FEE + DB\_AMOUNT,  
 CD\_TOTAL\_MONEY = CD\_AMOUNT + CD\_IR\_FEE;  
  
UPDATE TCKT\_NAPAS\_IBFT  
SET NAPAS\_FEE = CASE  
 WHEN GROUP\_TRAN = 'IBFT' AND RESPCODE = 0 THEN DB\_SV\_FEE + CD\_SV\_FEE - IFNULL(NP\_ADJ\_FEE,0)  
 WHEN RESPCODE <> 0 THEN 0  
 ELSE DB\_SV\_FEE + CD\_SV\_FEE - IFNULL(NP\_ADJ\_FEE,0)  
END;  
  
UPDATE TCKT\_NAPAS\_IBFT  
SET DEBIT = GREATEST(DB\_TOTAL\_MONEY - CD\_TOTAL\_MONEY, 0),  
 CREDIT = GREATEST(CD\_TOTAL\_MONEY - DB\_TOTAL\_MONEY, 0);

**Cập nhật BANK\_NAME bằng JOIN danh mục (thay GET\_FULL\_BANK\_NAME)**

UPDATE TCKT\_NAPAS\_IBFT t  
JOIN BANK\_DICT b ON t.BANK\_ID = b.BANK\_ID  
SET t.BANK\_NAME = REPLACE(b.FULL\_NAME,'Ngân hàng','NH')  
WHERE t.BANK\_ID IS NOT NULL;

**Hậu kỳ NAPAS\_FEE\_MONTH (nếu thuộc store gốc)**

DELETE FROM NAPAS\_FEE\_MONTH  
WHERE NAPAS\_TIME BETWEEN :sett\_from AND DATE\_ADD(:sett\_to, INTERVAL 1 DAY) - INTERVAL 1 SECOND  
 AND DATA\_TYPE='DOMESTIC' AND SERVICE\_TYPE='IBFT';  
  
INSERT INTO NAPAS\_FEE\_MONTH(...)  
SELECT ... FROM TCKT\_NAPAS\_IBFT;

**Log kết thúc / Log lỗi**

INSERT INTO ERR\_EX(ERR\_TIME, ERR\_CODE, ERR\_DETAIL, ERR\_MODULE) VALUES (NOW(),'0','End','NAPAS\_MASTER\_VIEW\_DOMESTIC\_IBFT');  
-- Khi lỗi trong ứng dụng:  
-- INSERT INTO ERR\_EX(ERR\_TIME, ERR\_CODE, ERR\_DETAIL, ERR\_MODULE, CRITICAL\_LEVEL) VALUES (NOW(), :ecode, :emesg, 'NAPAS\_MASTER\_VIEW\_DOMESTIC\_IBFT', 2);