**TÀI LIỆU MÔ TẢ CHUYỂN ĐỔI STORE – 33.NAPAS\_MASTER\_VIEW\_DOMESTIC\_IBFT.prc**

Oracle → TiDB/Java | Chuẩn hoá bước & SQL | Font: Times New Roman (SQL: Courier New)

**1. Tên store**

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

**1.1 Phân tích store (giữ nguyên đánh số; hiển thị logic 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) – tổng hợp phiên IBFT vào bảng tạm TCKT\_SESSION\_DOMESTIC.
3. Step 3: Reset các cột phí/điều chỉnh về 0 ở TCKT\_SESSION\_DOMESTIC (nếu có).
4. Step 4: Xoá dữ liệu cũ ở TCKT\_NAPAS\_IBFT (theo khoảng ngày/điều kiện batch) để tránh trùng.
5. Step 5: Insert từ SHCLOG\_SETT\_IBFT 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 = CD\_AMOUNT + CD\_IR\_FEE.
7. Step 7: Tính NAPAS\_FEE theo rule nghiệp vụ (CASE).
8. Step 8: Đảo dấu các record cần bù trừ (âm) theo rule.
9. Step 9: Tính DEBIT/CREDIT theo rule (bao gồm case đặc biệt BANK\_ID=BCCARD và TRAN\_TYPE='POS').
10. Step 10–12: Insert cộng gộp bổ sung (Grand total/nhóm) vào TCKT\_NAPAS\_IBFT.
11. Step 13: Xác định OD\_BY theo nhóm/loại giao dịch.
12. Step 14: Cập nhật BANK\_NAME bằng JOIN danh mục (thay GET\_FULL\_BANK\_NAME, rút gọn 'Ngân hàng' → 'NH').
13. Step 15: Set SERVICE\_TYPE = 'IBFT'.
14. Step 16: Xóa dữ liệu cũ NAPAS\_FEE\_MONTH (DATA\_TYPE='DOMESTIC', SERVICE\_TYPE='IBFT').
15. Step 17: Insert dữ liệu sang NAPAS\_FEE\_MONTH.
16. Step 18: Sao lưu sang NAPAS\_FEE\_MONTH\_AUTO\_BACKUP.
17. Step 19: Ghi log kết thúc (ERR\_EX).
18. Step 20: Ghi log lỗi (ERR\_EX) trong khối EXCEPTION.

**1.2 Bảng thay thế hàm Oracle → TiDB/Java**

|  |  |
| --- | --- |
| Oracle (store gốc) | TiDB/Java thay thế |
| SYSDATE | NOW() |
| TO\_DATE(str,'dd/MM/yyyy') | STR\_TO\_DATE(str,'%d/%m/%Y') |
| TO\_CHAR(dt,'fmt') | DATE\_FORMAT(dt,'fmt MySQL') |
| 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) |
| 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 theo từng step**

**Step 1 – Log bắt đầu**

INSERT INTO ERR\_EX(ERR\_TIME, ERR\_CODE, ERR\_DETAIL, ERR\_MODULE)  
VALUES (NOW(), 0, 'Start NAPAS\_MASTER\_VIEW\_DOMESTIC\_IBFT', 'STORE\_33');

**Step 2 – Gọi thủ tục con tổng hợp phiên IBFT**

-- Cách A: nếu giữ procedure ở DB  
CALL INSERT\_TCKT\_SESSION\_DOMESTIC\_IBFT(:pQRY\_FROM\_DATE, :pQRY\_TO\_DATE, :pUser);  
  
-- Cách B: chuyển lên Java, chạy các INSERT/UPDATE tương đương để populate TCKT\_SESSION\_DOMESTIC (tài liệu riêng).

**Step 4 – Xoá dữ liệu cũ ở TCKT\_NAPAS\_IBFT**

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

**Step 5 – Insert từ SHCLOG\_SETT\_IBFT → 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 bằng JOIN danh mục  
 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 SHCLOG\_SETT\_IBFT A  
WHERE DATE(A.LOCAL\_DATE) BETWEEN :sett\_from AND :sett\_to;

**Step 6–9 – Các bước tính toán (mẫu, điều chỉnh theo store gốc)**

-- Step 6: CD\_TOTAL\_MONEY  
UPDATE TCKT\_NAPAS\_IBFT  
SET CD\_TOTAL\_MONEY = CD\_AMOUNT + CD\_IR\_FEE  
WHERE DATE(SETT\_DATE) BETWEEN :sett\_from AND :sett\_to;  
  
-- Step 7: NAPAS\_FEE (ví dụ khung)  
UPDATE TCKT\_NAPAS\_IBFT  
SET NAPAS\_FEE = CASE  
 WHEN RESPCODE = 0 THEN IFNULL(DB\_SV\_FEE,0) + IFNULL(CD\_SV\_FEE,0) - IFNULL(NP\_ADJ\_FEE,0)  
 ELSE 0  
END  
WHERE DATE(SETT\_DATE) BETWEEN :sett\_from AND :sett\_to;  
  
-- Step 8: Đảo dấu record cần bù trừ (điền rule cụ thể)  
-- UPDATE ...  
  
-- Step 9: DEBIT/CREDIT (bao gồm case BANK\_ID=BCCARD & TRAN\_TYPE='POS')  
-- UPDATE ...;

**Step 10–12 – Insert cộng gộp bổ sung (Grand total/nhóm)**

-- INSERT ... SELECT ... GROUP BY ... (điền theo rule nhóm của store gốc)

**Step 13 – Xác định OD\_BY**

-- UPDATE TCKT\_NAPAS\_IBFT SET OD\_BY = ... WHERE ...

**Step 14 – Cập nhật BANK\_NAME (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  
 AND DATE(t.SETT\_DATE) BETWEEN :sett\_from AND :sett\_to;

**Step 15 – Set SERVICE\_TYPE**

UPDATE TCKT\_NAPAS\_IBFT  
SET SERVICE\_TYPE = 'IBFT'  
WHERE DATE(SETT\_DATE) BETWEEN :sett\_from AND :sett\_to;

**Step 16 – Xoá dữ liệu cũ NAPAS\_FEE\_MONTH**

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';

**Step 17 – Insert vào NAPAS\_FEE\_MONTH**

INSERT INTO NAPAS\_FEE\_MONTH (...columns...)  
SELECT ... FROM TCKT\_NAPAS\_IBFT  
WHERE DATE(SETT\_DATE) BETWEEN :sett\_from AND :sett\_to;

**Step 18 – Sao lưu sang NAPAS\_FEE\_MONTH\_AUTO\_BACKUP**

INSERT INTO NAPAS\_FEE\_MONTH\_AUTO\_BACKUP (...columns...)  
SELECT ... 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';

**Step 19 – Log kết thúc**

INSERT INTO ERR\_EX(ERR\_TIME, ERR\_CODE, ERR\_DETAIL, ERR\_MODULE)  
VALUES (NOW(), 0, 'End NAPAS\_MASTER\_VIEW\_DOMESTIC\_IBFT', 'STORE\_33');

**Step 20 – Log lỗi (Exception)**

-- Khi có lỗi, ứng dụng/Java ghi:  
INSERT INTO ERR\_EX(ERR\_TIME, ERR\_CODE, ERR\_DETAIL, ERR\_MODULE, CRITICAL\_LEVEL)  
VALUES (NOW(), :ecode, :emesg, 'STORE\_33', 2);