**TÀI LIỆU MÔ TẢ CHUYỂN ĐỔI STORE #33**

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 (chia theo step)**

Store tổng hợp và tính phí giao dịch IBFT nội địa, ghi vào bảng TCKT\_NAPAS\_IBFT, sau đó tổng hợp sang NAPAS\_FEE\_MONTH. Các bước chính (step) được suy ra từ các câu lệnh DML theo thứ tự thực thi:

1. Step 1: Ghi log bắt đầu (ERR\_EX).
2. Step 2: Insert tổng hợp giao dịch vào TCKT\_NAPAS\_IBFT (ISS-ACQ), chuẩn hóa SETT\_DATE/EDIT\_DATE; WITH\_BANK tính theo điều kiện Pcode2 & TGTT\_CONFIG.
3. Step 3: Reset các cột phí/điều chỉnh về 0 ở TCKT\_NAPAS\_IBFT.
4. Step 4: Tính DB\_TOTAL\_FEE = DB\_IR\_FEE + DB\_SV\_FEE + CD\_SV\_FEE.
5. Step 5: Tính DB\_TOTAL\_MONEY = DB\_TOTAL\_FEE + DB\_AMOUNT.
6. Step 6: Tính CD\_TOTAL\_MONEY = CD\_AMOUNT + CD\_IR\_FEE.
7. Step 7: Tính NAPAS\_FEE theo CASE nghiệp vụ.
8. Step 8: Đảo dấu các record cần bù trừ (âm).
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 bảng danh mục (thay GET\_FULL\_BANK\_NAME).
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 (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(expr, a, x, b, y, default) | CASE WHEN expr=a THEN x WHEN expr=b THEN y ELSE default END |
| SUBSTR(s, pos, len) | SUBSTRING(s, pos, len) |
| INSTR(s, sub) | INSTR(s, sub) (giữ nguyên) |
| TRIM(s) | TRIM(s) (giữ nguyên) |
| REPLACE(s, from, to) | REPLACE(s, from, to) (giữ nguyên) |
| LISTAGG(... WITHIN GROUP ...) | GROUP\_CONCAT(... ORDER BY ... SEPARATOR ...) |
| STANDARD\_HASH(str,'MD5') | MD5(str) |
| ROWNUM | ORDER BY ... LIMIT ... |
| /\*+ parallel \*/ | Bỏ hint (TiDB tự tối ưu) |
| 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 SQL thay thế (chạy trên TiDB) cho từng step**

Lưu ý: bỏ hint Oracle, thay hàm theo mục 1.2; dùng NOW()/CURDATE(), CASE/IFNULL/SUBSTRING; so sánh ngày bằng DATE()/CAST/STR\_TO\_DATE theo tham số.

**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');

**Step 2 – Insert tổng hợp vào TCKT\_NAPAS\_IBFT (ISS-ACQ, WITH\_BANK theo CASE)**

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 ở 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 3 – Reset phí/điều chỉnh**

UPDATE TCKT\_NAPAS\_IBFT  
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;

**Step 4 – Tính DB\_TOTAL\_FEE**

UPDATE TCKT\_NAPAS\_IBFT  
SET DB\_TOTAL\_FEE = DB\_IR\_FEE + DB\_SV\_FEE + CD\_SV\_FEE;

**Step 5 – Tính DB\_TOTAL\_MONEY**

UPDATE TCKT\_NAPAS\_IBFT  
SET DB\_TOTAL\_MONEY = DB\_TOTAL\_FEE + DB\_AMOUNT;

**Step 6 – Tính CD\_TOTAL\_MONEY**

UPDATE TCKT\_NAPAS\_IBFT  
SET CD\_TOTAL\_MONEY = CD\_AMOUNT + CD\_IR\_FEE;

**Step 7 – Tính NAPAS\_FEE (mẫu)**

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;

**Step 8 – Đảo dấu record bù trừ**

UPDATE TCKT\_NAPAS\_IBFT  
SET DB\_TOTAL\_TRAN = -DB\_TOTAL\_TRAN,  
 DB\_AMOUNT = -DB\_AMOUNT,  
 DB\_IR\_FEE = -DB\_IR\_FEE,  
 DB\_SV\_FEE = -DB\_SV\_FEE,  
 DB\_TOTAL\_FEE = -DB\_TOTAL\_FEE,  
 DB\_TOTAL\_MONEY = -DB\_TOTAL\_MONEY,  
 CD\_TOTAL\_TRAN = -CD\_TOTAL\_TRAN,  
 CD\_AMOUNT = -CD\_AMOUNT,  
 CD\_IR\_FEE = -CD\_IR\_FEE,  
 CD\_SV\_FEE = -CD\_SV\_FEE,  
 CD\_TOTAL\_MONEY = -CD\_TOTAL\_MONEY,  
 NAPAS\_FEE = -NAPAS\_FEE  
WHERE 0=1;

**Step 9 – Tính DEBIT/CREDIT (rút gọn)**

UPDATE TCKT\_NAPAS\_IBFT  
SET DEBIT = CASE  
 WHEN BANK\_ID = :bccard\_id AND SETTLEMENT\_CURRENCY = 704 THEN GREATEST(DB\_AMOUNT - CD\_AMOUNT, 0)  
 WHEN BANK\_ID IN (602907,605609,600005,600006,600007,980471,971100,971111)  
 THEN GREATEST(DB\_AMOUNT - CD\_AMOUNT, 0)  
 ELSE GREATEST(DB\_TOTAL\_MONEY - CD\_TOTAL\_MONEY, 0)  
 END,  
 CREDIT = CASE  
 WHEN BANK\_ID = :bccard\_id AND SETTLEMENT\_CURRENCY = 704 AND TRAN\_TYPE='POS'  
 THEN GREATEST(CD\_AMOUNT - DB\_AMOUNT, 0)  
 WHEN BANK\_ID IN (602907,605609,600005,600006,600007,980471,971100,971111)  
 THEN GREATEST(CD\_AMOUNT - DB\_AMOUNT, 0)  
 ELSE GREATEST(CD\_TOTAL\_MONEY - DB\_TOTAL\_MONEY, 0)  
 END;

**Step 10–12 – Insert nhóm/tổng hợp bổ sung**

INSERT INTO TCKT\_NAPAS\_IBFT(MSGTYPE\_DETAIL, OD\_BY, SETT\_DATE, EDIT\_DATE, SETTLEMENT\_CURRENCY, RESPCODE,  
 GROUP\_TRAN, PCODE, TRAN\_TYPE, SERVICE\_CODE, GROUP\_ROLE, BANK\_ID, WITH\_BANK,  
 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,  
 DEBIT, CREDIT, ADJ\_FEE, NP\_ADJ\_FEE, BC\_NP\_ADJ, BC\_NP\_SUM, BC\_CL\_ADJ, STEP, SUB\_BANK, LIQUIDITY)  
SELECT MSGTYPE\_DETAIL, NULL, SETT\_DATE, EDIT\_DATE, SETTLEMENT\_CURRENCY, NULL,  
 GROUP\_TRAN, NULL, TRAN\_TYPE, SERVICE\_CODE, NULL, NULL, WITH\_BANK,  
 SUM(DB\_TOTAL\_TRAN), SUM(DB\_AMOUNT), SUM(DB\_IR\_FEE), SUM(DB\_SV\_FEE),  
 SUM(DB\_TOTAL\_FEE), SUM(DB\_TOTAL\_MONEY),  
 SUM(CD\_TOTAL\_TRAN), SUM(CD\_AMOUNT), SUM(CD\_IR\_FEE), SUM(CD\_SV\_FEE),  
 SUM(CD\_TOTAL\_MONEY), SUM(NAPAS\_FEE),  
 SUM(DEBIT), SUM(CREDIT), SUM(ADJ\_FEE), SUM(NP\_ADJ\_FEE), SUM(BC\_NP\_ADJ), SUM(BC\_NP\_SUM), SUM(BC\_CL\_ADJ),  
 'TOTAL', NULL, 'N'  
FROM TCKT\_NAPAS\_IBFT  
WHERE SETT\_DATE BETWEEN :sett\_from AND :sett\_to  
GROUP BY MSGTYPE\_DETAIL, SETT\_DATE, EDIT\_DATE, SETTLEMENT\_CURRENCY, GROUP\_TRAN, TRAN\_TYPE, SERVICE\_CODE, WITH\_BANK;

**Step 13 – Xác định OD\_BY**

UPDATE TCKT\_NAPAS\_IBFT  
SET OD\_BY = CASE  
 WHEN GROUP\_TRAN='IBFT' AND TRAN\_TYPE='SSP\_ON' THEN 'AD'  
 WHEN TRAN\_TYPE='Non IBFT' THEN 'AA'  
 ELSE OD\_BY  
END;

**Step 14 – Cập nhật BANK\_NAME bằng JOIN (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;

**Step 15 – SERVICE\_TYPE**

UPDATE TCKT\_NAPAS\_IBFT SET SERVICE\_TYPE='IBFT';

**Step 16 – Xóa 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 – Ghi NAPAS\_FEE\_MONTH**

INSERT INTO NAPAS\_FEE\_MONTH(  
 MSGTYPE\_DETAIL, NAPAS\_TIME, USER\_SETT, EDIT\_NOTE, 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,  
 DEBIT, CREDIT, OD\_BY, ADJ\_FEE, NP\_ADJ\_FEE, MERCHANT\_TYPE, STEP, SETTLEMENT\_DATE\_FROM, SETTLEMENT\_DATE\_TO,  
 DATA\_TYPE, SERVICE\_TYPE  
)  
SELECT MSGTYPE\_DETAIL, NOW(), :pUser, NULL, 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,  
 DEBIT, CREDIT, OD\_BY, ADJ\_FEE, NP\_ADJ\_FEE, MERCHANT\_TYPE, STEP, :sett\_from, :sett\_to,  
 'DOMESTIC', 'IBFT'  
FROM TCKT\_NAPAS\_IBFT;

**Step 18 – Sao lưu AUTO\_BACKUP**

INSERT INTO NAPAS\_FEE\_MONTH\_AUTO\_BACKUP(  
 MSGTYPE\_DETAIL, NAPAS\_SETT\_TIME, USER\_SETT, EDIT\_NOTE, 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,  
 DEBIT, CREDIT, OD\_BY, ADJ\_FEE, NP\_ADJ\_FEE, MERCHANT\_TYPE, STEP,  
 SETTLEMENT\_DATE\_FROM, SETTLEMENT\_DATE\_TO, DATA\_TYPE, SERVICE\_TYPE  
)  
SELECT MSGTYPE\_DETAIL, NOW(), :pUser, NULL, 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,  
 DEBIT, CREDIT, OD\_BY, ADJ\_FEE, NP\_ADJ\_FEE, MERCHANT\_TYPE, STEP,  
 :sett\_from, :sett\_to, 'DOMESTIC', 'IBFT'  
FROM NAPAS\_FEE\_MONTH;

**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');

**Step 20 – Log lỗi (trong EXCEPTION)**

INSERT INTO ERR\_EX(ERR\_TIME, ERR\_CODE, ERR\_DETAIL, ERR\_MODULE, CRITICAL\_LEVEL)  
VALUES (NOW(), :ecode, :emesg, 'NAPAS\_MASTER\_VIEW\_DOMESTIC\_IBFT', 2);