**TÀI LIỆU MÔ TẢ CHUYỂN ĐỔI STORE – RPT.MERGE\_SHC\_SETT\_IBFT\_200**

Chuyển MERGE → UPDATE + INSERT | Nguồn: V\_ISOMESSAGE\_TMP\_TURN\_0200 | Chuẩn hoá font & định dạng SQL

**1. Tên store**

RPT.MERGE\_SHC\_SETT\_IBFT\_200 (Oracle → TiDB/Java).

**1.1 Phân tích store (giữ nguyên đánh số; tách MERGE thành 3.1 & 3.2)**

Step 1: Ghi log bắt đầu (ERR\_EX).

Step 2: Tiền xử lý/chuẩn bị tham số.

Step 3: Thực hiện MERGE dữ liệu IBFT 0200 vào SHCLOG\_SETT\_IBFT từ V\_ISOMESSAGE\_TMP\_TURN\_0200:

• Step 3.1 (WHEN MATCHED → UPDATE): Đồng bộ bản ghi đã tồn tại theo bộ khoá ghép (PAN, ORIGTRACE, TERMID, LOCAL\_DATE(mmdd), LOCAL\_TIME, ACQUIRER).

- Nếu chênh lệch số tiền so với nguồn thì đặt RESPCODE=116 khi RESPCODE hiện tại = 0; TXNSRC='RC=99'.

- Gán CONTENT\_FUND = IBFT\_INFO.

• Step 3.2 (WHEN NOT MATCHED → INSERT): Chèn bản ghi còn thiếu sau khi dedup nguồn 1 dòng/khóa (ưu tiên TNX\_STAMP mới nhất).

- Thay các hàm tự định nghĩa bằng SQL thuần: MAP\_IBFT\_ACQ\_ID → CASE; GET\_IBT\_BIN → JOIN IBFT\_BANK\_BINS; TO\_NUMBER\_BNV → CAST(... AS UNSIGNED).

Step 4: Ghi log kết thúc MERGE (Finish Merge).

Step 5: Gán STT = ROW\_NUMBER() + :iSTT cho các bản ghi STT IS NULL và ORIGRESPCODE=97.

Step 6: Ghi log kết thúc toàn store (End).

Step 7: Ghi log lỗi (Exception) khi phát sinh.

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

|  |  |
| --- | --- |
| Hàm/biểu thức Oracle | Thay thế (TiDB/Java) |
| MERGE INTO tgt USING src ON (key)... | Thay thế bằng INSERT ON DUPLICATE KEY UPDATE  Với Key dung tinh trung lặp là  UNIQUE KEY `uk\_ibft\_merge` (`PAN`,`ORIGTRACE`,`TERMID`,`LOCAL\_DATE`,`LOCAL\_TIME`,`ACQUIRER`) |
| SYSDATE | NOW() |
| TO\_DATE/TO\_CHAR/TRUNC | STR\_TO\_DATE/DATE\_FORMAT/CURDATE |
| NVL / DECODE | IFNULL / CASE WHEN ... THEN ... ELSE ... END |
| ROWNUM (dedup) | ROW\_NUMBER() OVER (PARTITION BY ... ORDER BY TNX\_STAMP DESC) |
| MAP\_IBFT\_ACQ\_ID(x) | CASE cố định trong SELECT/CTE |
| GET\_IBT\_BIN(x) | LEFT JOIN IBFT\_BANK\_BINS + CASE map 191919→970459, 970415→970489 |
| TO\_NUMBER\_BNV(x) | CAST(x AS UNSIGNED) |
| NP\_CONVERT\_LOCAL\_DATE | Thay thế bằng lệnh Case When:  CASE  WHEN (  CASE  WHEN B.SETTLE\_DATE = '0229' THEN  IFNULL(  STR\_TO\_DATE(  CONCAT(  (ROUND(FLOOR(YEAR(CURDATE()) / 4)) \* 4),  B.SETTLE\_DATE  ),  '%Y%m%d'  ),  STR\_TO\_DATE(  CONCAT(  (ROUND(FLOOR((YEAR(CURDATE()) - 1) / 4)) \* 4),  B.SETTLE\_DATE  ),  '%Y%m%d'  )  )  ELSE  IFNULL(  STR\_TO\_DATE(  CONCAT(YEAR(CURDATE()), B.SETTLE\_DATE),  '%Y%m%d'  ),  STR\_TO\_DATE(  CONCAT(YEAR(DATE\_SUB(CURDATE(), INTERVAL 1 YEAR)), B.SETTLE\_DATE),  '%Y%m%d'  )  )  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 MERGE SHCLOG\_SETT\_IBFT and ISOMESSAGE\_TMP\_TURN', 'MERGE\_SHC\_SETT\_IBFT\_200');

**Step 3 – MERGE → INSERT ON DUPLICATE KEY UPDATE**

INSERT INTO SHCLOG\_SETT\_IBFT (

DATA\_ID, PPCODE, MSGTYPE, PAN, PCODE, Amount, ACQ\_CURRENCY\_CODE, TRACE, LOCAL\_TIME, LOCAL\_DATE, SETTLEMENT\_DATE,

ACQUIRER, ISSUER, RESPCODE, MERCHANT\_TYPE, MERCHANT\_TYPE\_ORIG, AUTHNUM, SETT\_CURRENCY\_CODE, TERMID, ADD\_INFO, ACCTNUM,

ISS\_CURRENCY\_CODE, ORIGTRACE, ORIGISS, ORIGRESPCODE, CH\_CURRENCY\_CODE, ACQUIRER\_FE, ACQUIRER\_RP, ISSUER\_FE,

ISSUER\_RP, PCODE2, FROM\_SYS, BB\_BIN, BB\_BIN\_ORIG, CONTENT\_FUND, TXNSRC, ACQ\_COUNTRY, POS\_ENTRY\_CODE, POS\_CONDITION\_CODE,

ADDRESPONSE, MVV, F4, F5, F6, F49, SETTLEMENT\_CODE, SETTLEMENT\_RATE, ISS\_CONV\_RATE, TCC,

refnum, trandate, trantime, ACCEPTORNAME, TERMLOC, F15, pcode\_orig, ACCOUNT\_NO, DEST\_ACCOUNT

)

SELECT

1 AS DATA\_ID,

CAST(B.PROC\_CODE AS SIGNED) AS PPCODE,

'210' AS MSGTYPE,

TRIM(B.CARD\_NO) AS PAN,

CAST(B.PROC\_CODE AS SIGNED) AS PCODE,

CAST(COALESCE(LEFT(B.AMOUNT, LENGTH(B.AMOUNT)-2), '0') AS SIGNED) AS Amount,

704 AS ACQ\_CURRENCY\_CODE,

CAST(CONCAT('2', B.TRACE\_NO) AS SIGNED) AS TRACE,

CAST(B.LOCAL\_TIME AS SIGNED) AS LOCAL\_TIME,

-- NP\_CONVERT\_LOCAL\_DATE(B.LOCAL\_DATE, TRUNCATE(CURDATE(), 0))

CASE

WHEN (

CASE

WHEN B.LOCAL\_DATE = '0229' THEN

IFNULL(

STR\_TO\_DATE(

CONCAT(

(ROUND(FLOOR(YEAR(CURDATE()) / 4)) \* 4),

B.LOCAL\_DATE

),

'%Y%m%d'

),

STR\_TO\_DATE(

CONCAT(

(ROUND(FLOOR((YEAR(CURDATE()) - 1) / 4)) \* 4),

B.LOCAL\_DATE

),

'%Y%m%d'

)

)

ELSE

IFNULL(

STR\_TO\_DATE(

CONCAT(YEAR(CURDATE()), B.LOCAL\_DATE),

'%Y%m%d'

),

STR\_TO\_DATE(

CONCAT(YEAR(DATE\_SUB(CURDATE(), INTERVAL 1 YEAR)), B.LOCAL\_DATE),

'%Y%m%d'

)

)

END

) > CURDATE() THEN

CASE

WHEN B.LOCAL\_DATE = '0229' THEN

IFNULL(

STR\_TO\_DATE(

CONCAT(

(ROUND(FLOOR((YEAR(CURDATE()) - 1) / 4)) \* 4),

B.LOCAL\_DATE

),

'%Y%m%d'

),

NULL

)

ELSE

IFNULL(

STR\_TO\_DATE(

CONCAT(YEAR(DATE\_SUB(CURDATE(), INTERVAL 1 YEAR)), B.LOCAL\_DATE),

'%Y%m%d'

),

NULL

)

END

ELSE

CASE

WHEN B.LOCAL\_DATE = '0229' THEN

IFNULL(

STR\_TO\_DATE(

CONCAT(

(ROUND(FLOOR(YEAR(CURDATE()) / 4)) \* 4),

B.LOCAL\_DATE

),

'%Y%m%d'

),

STR\_TO\_DATE(

CONCAT(

(ROUND(FLOOR((YEAR(CURDATE()) - 1) / 4)) \* 4),

B.LOCAL\_DATE

),

'%Y%m%d'

)

)

ELSE

IFNULL(

STR\_TO\_DATE(

CONCAT(YEAR(CURDATE()), B.LOCAL\_DATE),

'%Y%m%d'

),

STR\_TO\_DATE(

CONCAT(YEAR(DATE\_SUB(CURDATE(), INTERVAL 1 YEAR)), B.LOCAL\_DATE),

'%Y%m%d'

)

)

END

END AS LOCAL\_DATE,

-- NP\_CONVERT\_LOCAL\_DATE(B.SETTLE\_DATE, TRUNCATE(CURDATE(), 0))

CASE

WHEN (

CASE

WHEN B.SETTLE\_DATE = '0229' THEN

IFNULL(

STR\_TO\_DATE(

CONCAT(

(ROUND(FLOOR(YEAR(CURDATE()) / 4)) \* 4),

B.SETTLE\_DATE

),

'%Y%m%d'

),

STR\_TO\_DATE(

CONCAT(

(ROUND(FLOOR((YEAR(CURDATE()) - 1) / 4)) \* 4),

B.SETTLE\_DATE

),

'%Y%m%d'

)

)

ELSE

IFNULL(

STR\_TO\_DATE(

CONCAT(YEAR(CURDATE()), B.SETTLE\_DATE),

'%Y%m%d'

),

STR\_TO\_DATE(

CONCAT(YEAR(DATE\_SUB(CURDATE(), INTERVAL 1 YEAR)), B.SETTLE\_DATE),

'%Y%m%d'

)

)

END

) > CURDATE() THEN

CASE

WHEN B.SETTLE\_DATE = '0229' THEN

IFNULL(

STR\_TO\_DATE(

CONCAT(

(ROUND(FLOOR((YEAR(CURDATE()) - 1) / 4)) \* 4),

B.SETTLE\_DATE

),

'%Y%m%d'

),

NULL

)

ELSE

IFNULL(

STR\_TO\_DATE(

CONCAT(YEAR(DATE\_SUB(CURDATE(), INTERVAL 1 YEAR)), B.SETTLE\_DATE),

'%Y%m%d'

),

NULL

)

END

ELSE

CASE

WHEN B.SETTLE\_DATE = '0229' THEN

IFNULL(

STR\_TO\_DATE(

CONCAT(

(ROUND(FLOOR(YEAR(CURDATE()) / 4)) \* 4),

B.SETTLE\_DATE

),

'%Y%m%d'

),

STR\_TO\_DATE(

CONCAT(

(ROUND(FLOOR((YEAR(CURDATE()) - 1) / 4)) \* 4),

B.SETTLE\_DATE

),

'%Y%m%d'

)

)

ELSE

IFNULL(

STR\_TO\_DATE(

CONCAT(YEAR(CURDATE()), B.SETTLE\_DATE),

'%Y%m%d'

),

STR\_TO\_DATE(

CONCAT(YEAR(DATE\_SUB(CURDATE(), INTERVAL 1 YEAR)), B.SETTLE\_DATE),

'%Y%m%d'

)

)

END

END AS SETTLEMENT\_DATE,

CAST(B.ACQ\_ID AS SIGNED) AS ACQUIRER,

CAST(B.ACQ\_ID AS SIGNED) AS ISSUER,

CASE

WHEN B.BEN\_ID = '971133' AND B.DEST\_ACCOUNT LIKE 'NPDC%' THEN 68

WHEN B.SERVICE\_CODE = 'QR\_PUSH' THEN 68

WHEN B.BEN\_ID = '971100' AND B.TCC = '99' THEN 68

WHEN TRIM(B.ISS\_ID) IN ('980471', '980472') THEN 68

ELSE 0

END AS RESPCODE,

6011 AS MERCHANT\_TYPE,

CAST(B.MCC AS SIGNED) AS MERCHANT\_TYPE\_ORIG,

B.APPROVAL\_CODE AS AUTHNUM,

704 AS SETT\_CURRENCY\_CODE,

B.TERM\_ID AS TERMID,

B.ADD\_INFO AS ADD\_INFO,

CONCAT(COALESCE(B.ACCOUNT\_NO, ' '), '|', COALESCE(B.DEST\_ACCOUNT, '')) AS ACCTNUM,

704 AS ISS\_CURRENCY\_CODE,

CAST(B.TRACE\_NO AS SIGNED) AS ORIGTRACE,

CAST(B.ACQ\_ID AS SIGNED) AS ORIGISS,

'97' AS ORIGRESPCODE,

704 AS CH\_CURRENCY\_CODE,

-- MAP\_IBFT\_ACQ\_ID(B.ACQ\_ID) for ACQUIRER\_FE

IFNULL(

CASE

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970429 THEN 157979

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970400 THEN 161087

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970427 THEN 166888

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970441 THEN 180906

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970425 THEN 970459

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970431 THEN 452999

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970436 THEN 686868

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970419 THEN 818188

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970407 THEN 888899

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970434 THEN 888999

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970440 THEN 970468

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970418 THEN 970488

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970432 THEN 981957

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970405 THEN 970499

ELSE CAST(B.ACQ\_ID AS SIGNED)

END,

NULL

) AS ACQUIRER\_FE,

-- MAP\_IBFT\_ACQ\_ID(B.ACQ\_ID) for ACQUIRER\_RP

IFNULL(

CASE

WHEN TRIM(B.ISS\_ID) = '980471' THEN 980471

WHEN TRIM(B.ISS\_ID) = '980475' THEN 980478

WHEN CAST(B.ACQ\_ID AS SIGNED) = 191919 THEN 970459

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970415 THEN 970489

ELSE

IFNULL(

CASE

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970429 THEN 157979

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970400 THEN 161087

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970427 THEN 166888

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970441 THEN 180906

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970425 THEN 970459

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970431 THEN 452999

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970436 THEN 686868

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970419 THEN 818188

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970407 THEN 888899

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970434 THEN 888999

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970440 THEN 970468

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970418 THEN 970488

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970432 THEN 981957

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970405 THEN 970499

ELSE CAST(B.ACQ\_ID AS SIGNED)

END,

NULL

)

END,

NULL

) AS ACQUIRER\_RP,

-- MAP\_IBFT\_ACQ\_ID(B.ACQ\_ID) for ISSUER\_FE

IFNULL(

CASE

WHEN CAST(B.ACQ\_ID AS SIGNED) = 191919 THEN 970459

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970415 THEN 970489

ELSE

IFNULL(

CASE

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970429 THEN 157979

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970400 THEN 161087

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970427 THEN 166888

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970441 THEN 180906

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970425 THEN 970459

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970431 THEN 452999

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970436 THEN 686868

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970419 THEN 818188

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970407 THEN 888899

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970434 THEN 888999

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970440 THEN 970468

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970418 THEN 970488

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970432 THEN 981957

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970405 THEN 970499

ELSE CAST(B.ACQ\_ID AS SIGNED)

END,

NULL

)

END,

NULL

) AS ISSUER\_FE,

-- MAP\_IBFT\_ACQ\_ID(B.ACQ\_ID) for ISSUER\_RP

IFNULL(

CASE

WHEN TRIM(B.ISS\_ID) = '980471' THEN 980471

WHEN TRIM(B.ISS\_ID) = '980475' THEN 980478

WHEN CAST(B.ACQ\_ID AS SIGNED) = 191919 THEN 970459

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970415 THEN 970489

ELSE

IFNULL(

CASE

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970429 THEN 157979

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970400 THEN 161087

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970427 THEN 166888

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970441 THEN 180906

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970425 THEN 970459

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970431 THEN 452999

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970436 THEN 686868

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970419 THEN 818188

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970407 THEN 888899

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970434 THEN 888999

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970440 THEN 970468

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970418 THEN 970488

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970432 THEN 981957

WHEN CAST(B.ACQ\_ID AS SIGNED) = 970405 THEN 970499

ELSE CAST(B.ACQ\_ID AS SIGNED)

END,

NULL

)

END,

NULL

) AS ISSUER\_RP,

CASE

WHEN B.TCC = '99' THEN 930000

WHEN B.TCC = '95' THEN 950000

WHEN B.SERVICE\_CODE = 'QR\_PUSH' THEN 890000

WHEN B.TCC = '97' THEN 720000

WHEN B.TCC = '98' THEN 730000

ELSE 910000

END AS PCODE2,

'IBT' AS FROM\_SYS,

-- GET\_IBT\_BIN(B.BEN\_ID) or GET\_IBT\_BIN(LEFT(B.DEST\_ACCOUNT, 6)) for BB\_BIN

CASE

WHEN TRIM(B.ISS\_ID) = '980472' THEN 980471

WHEN TRIM(B.ISS\_ID) = '980474' THEN 980478

WHEN B.BEN\_ID IS NOT NULL AND B.PROC\_CODE IN ('912020', '910020')

THEN IFNULL(

CASE

WHEN (SELECT COUNT(\*) FROM IBFT\_BANK\_BINS WHERE BIN = B.BEN\_ID) = 1 THEN

CASE

WHEN (SELECT CAST(MEMBER\_ID AS SIGNED) FROM IBFT\_BANK\_BINS WHERE BIN = B.BEN\_ID) = 191919 THEN 970459

WHEN (SELECT CAST(MEMBER\_ID AS SIGNED) FROM IBFT\_BANK\_BINS WHERE BIN = B.BEN\_ID) = 970415 THEN 970489

ELSE (SELECT CAST(MEMBER\_ID AS SIGNED) FROM IBFT\_BANK\_BINS WHERE BIN = B.BEN\_ID)

END

ELSE NULL

END,

NULL

)

ELSE

IFNULL(

CASE

WHEN (SELECT COUNT(\*) FROM IBFT\_BANK\_BINS WHERE BIN = LEFT(B.DEST\_ACCOUNT, 6)) = 1 THEN

CASE

WHEN (SELECT CAST(MEMBER\_ID AS SIGNED) FROM IBFT\_BANK\_BINS WHERE BIN = LEFT(B.DEST\_ACCOUNT, 6)) = 191919 THEN 970459

WHEN (SELECT CAST(MEMBER\_ID AS SIGNED) FROM IBFT\_BANK\_BINS WHERE BIN = LEFT(B.DEST\_ACCOUNT, 6)) = 970415 THEN 970489

ELSE (SELECT CAST(MEMBER\_ID AS SIGNED) FROM IBFT\_BANK\_BINS WHERE BIN = LEFT(B.DEST\_ACCOUNT, 6))

END

ELSE NULL

END,

NULL

)

END AS BB\_BIN,

-- GET\_IBT\_BIN for BB\_BIN\_ORIG

CASE

WHEN B.BEN\_ID IN (SELECT TGTT\_ID FROM TGTT\_20) THEN IFNULL(CAST(B.BEN\_ID AS SIGNED), NULL) -- to\_number\_bnv(B.BEN\_ID)

WHEN TRIM(B.ISS\_ID) IN ('980472', '980474', '980475')

THEN CASE

WHEN B.BEN\_ID IS NOT NULL AND B.PROC\_CODE IN ('912020', '910020')

THEN IFNULL(

CASE

WHEN (SELECT COUNT(\*) FROM IBFT\_BANK\_BINS WHERE BIN = B.BEN\_ID) = 1 THEN

CASE

WHEN (SELECT CAST(MEMBER\_ID AS SIGNED) FROM IBFT\_BANK\_BINS WHERE BIN = B.BEN\_ID) = 191919 THEN 970459

WHEN (SELECT CAST(MEMBER\_ID AS SIGNED) FROM IBFT\_BANK\_BINS WHERE BIN = B.BEN\_ID) = 970415 THEN 970489

ELSE (SELECT CAST(MEMBER\_ID AS SIGNED) FROM IBFT\_BANK\_BINS WHERE BIN = B.BEN\_ID)

END

ELSE NULL

END,

NULL

)

ELSE

IFNULL(

CASE

WHEN (SELECT COUNT(\*) FROM IBFT\_BANK\_BINS WHERE BIN = LEFT(B.DEST\_ACCOUNT, 6)) = 1 THEN

CASE

WHEN (SELECT CAST(MEMBER\_ID AS SIGNED) FROM IBFT\_BANK\_BINS WHERE BIN = LEFT(B.DEST\_ACCOUNT, 6)) = 191919 THEN 970459

WHEN (SELECT CAST(MEMBER\_ID AS SIGNED) FROM IBFT\_BANK\_BINS WHERE BIN = LEFT(B.DEST\_ACCOUNT, 6)) = 970415 THEN 970489

ELSE (SELECT CAST(MEMBER\_ID AS SIGNED) FROM IBFT\_BANK\_BINS WHERE BIN = LEFT(B.DEST\_ACCOUNT, 6))

END

ELSE NULL

END,

NULL

)

END

ELSE IFNULL(CAST(B.BEN\_ID AS SIGNED), NULL) -- to\_number\_bnv(B.BEN\_ID)

END AS BB\_BIN\_ORIG,

B.IBFT\_INFO AS CONTENT\_FUND,

'MTI=200' AS TXNSRC,

B.ACQ\_COUNTRY,

B.POS\_ENTRY\_CODE,

B.POS\_CONDITION\_CODE,

B.ADDRESPONSE,

B.MVV,

CAST(COALESCE(LEFT(B.F4, LENGTH(B.F4)-2), '0') AS SIGNED) AS F4,

CAST(COALESCE(LEFT(B.F5, LENGTH(B.F5)-2), '0') AS SIGNED) AS F5,

CAST(COALESCE(LEFT(B.F6, LENGTH(B.F6)-2), '0') AS SIGNED) AS F6,

B.F49,

B.SETTLEMENT\_CODE,

B.SETTLEMENT\_RATE,

B.ISS\_CONV\_RATE,

B.TCC,

B.REF\_NO,

TRUNCATE(B.Tnx\_Stamp, 0) AS trandate,

DATE\_FORMAT(B.tnx\_stamp, '%H%i%s') AS trantime,

B.CARD\_ACCEPT\_NAME\_LOCATION AS ACCEPTORNAME,

B.CARD\_ACCEPT\_ID\_CODE AS TERMLOC,

-- NP\_CONVERT\_LOCAL\_DATE(B.SETTLE\_DATE, TRUNCATE(CURDATE(), 0)) for F15

CASE

WHEN (

CASE

WHEN B.SETTLE\_DATE = '0229' THEN

IFNULL(

STR\_TO\_DATE(

CONCAT(

(ROUND(FLOOR(YEAR(CURDATE()) / 4)) \* 4),

B.SETTLE\_DATE

),

'%Y%m%d'

),

STR\_TO\_DATE(

CONCAT(

(ROUND(FLOOR((YEAR(CURDATE()) - 1) / 4)) \* 4),

B.SETTLE\_DATE

),

'%Y%m%d'

)

)

ELSE

IFNULL(

STR\_TO\_DATE(

CONCAT(YEAR(CURDATE()), B.SETTLE\_DATE),

'%Y%m%d'

),

STR\_TO\_DATE(

CONCAT(YEAR(DATE\_SUB(CURDATE(), INTERVAL 1 YEAR)), B.SETTLE\_DATE),

'%Y%m%d'

)

)

END

) > CURDATE() THEN

CASE

WHEN B.SETTLE\_DATE = '0229' THEN

IFNULL(

STR\_TO\_DATE(

CONCAT(

(ROUND(FLOOR((YEAR(CURDATE()) - 1) / 4)) \* 4),

B.SETTLE\_DATE

),

'%Y%m%d'

),

NULL

)

ELSE

IFNULL(

STR\_TO\_DATE(

CONCAT(YEAR(DATE\_SUB(CURDATE(), INTERVAL 1 YEAR)), B.SETTLE\_DATE),

'%Y%m%d'

),

NULL

)

END

ELSE

CASE

WHEN B.SETTLE\_DATE = '0229' THEN

IFNULL(

STR\_TO\_DATE(

CONCAT(

(ROUND(FLOOR(YEAR(CURDATE()) / 4)) \* 4),

B.SETTLE\_DATE

),

'%Y%m%d'

),

STR\_TO\_DATE(

CONCAT(

(ROUND(FLOOR((YEAR(CURDATE()) - 1) / 4)) \* 4),

B.SETTLE\_DATE

),

'%Y%m%d'

)

)

ELSE

IFNULL(

STR\_TO\_DATE(

CONCAT(YEAR(CURDATE()), B.SETTLE\_DATE),

'%Y%m%d'

),

STR\_TO\_DATE(

CONCAT(YEAR(DATE\_SUB(CURDATE(), INTERVAL 1 YEAR)), B.SETTLE\_DATE),

'%Y%m%d'

)

)

END

END AS F15,

CAST(B.PROC\_CODE AS SIGNED) AS pcode\_orig,

B.ACCOUNT\_NO,

B.DEST\_ACCOUNT

FROM ISOMESSAGE\_TMP\_TURN B

WHERE B.CARD\_NO IS NOT NULL

AND B.MTI = '0200'

AND (COALESCE(CAST(B.BEN\_ID AS SIGNED), 0) <> 0 AND COALESCE(CAST(B.TRACE\_NO AS SIGNED), 0) <> 0)

ON DUPLICATE KEY UPDATE

RESPCODE = CASE

WHEN SHCLOG\_SETT\_IBFT.AMOUNT <> ROUND(B.AMOUNT/100, 0) AND SHCLOG\_SETT\_IBFT.RESPCODE = 0 THEN 116

ELSE SHCLOG\_SETT\_IBFT.RESPCODE

END,

TXNSRC = CASE

WHEN SHCLOG\_SETT\_IBFT.AMOUNT <> ROUND(B.AMOUNT/100, 0) THEN 'RC=99'

ELSE CAST(SHCLOG\_SETT\_IBFT.TXNSRC AS CHAR)

END,

CONTENT\_FUND = B.IBFT\_INFO;

**Step 4 – Log kết thúc MERGE**

INSERT INTO ERR\_EX(ERR\_TIME, ERR\_CODE, ERR\_DETAIL, ERR\_MODULE)  
VALUES (NOW(), 0, 'Finish Merge SHCLOG\_SETT\_IBFT and ISOMESSAGE\_TMP\_TURN', 'MERGE\_SHC\_SETT\_IBFT\_200');  
-- TiDB autocommit; nếu dùng transaction, commit ở tầng ứng dụng.

**Step 5 – Gán STT = ROW\_NUMBER() + :iSTT khi STT IS NULL & ORIGRESPCODE = 97**

UPDATE SHCLOG\_SETT\_IBFT AS a  
JOIN (  
 SELECT  
 tidb\_id,  
 ROW\_NUMBER() OVER (ORDER BY TRANDATE, TRANTIME, TRACE, id) + :iSTT AS new\_stt  
 FROM SHCLOG\_SETT\_IBFT  
 WHERE STT IS NULL  
 AND ORIGRESPCODE = 97  
) AS b  
ON a.tidb\_id = b.tidb\_id  
SET a.STT = b.new\_stt;

**Step 6 – Log kết thúc toàn store**

INSERT INTO ERR\_EX(ERR\_TIME, ERR\_CODE, ERR\_DETAIL, ERR\_MODULE)  
VALUES (NOW(), 0, 'End MERGE\_SHC\_SETT\_IBFT\_200', 'RPT.MERGE\_SHC\_SETT\_IBFT\_200');

**Step 7 – Log lỗi (Exception)**

INSERT INTO ERR\_EX(ERR\_TIME, ERR\_CODE, ERR\_DETAIL, ERR\_MODULE, CRITICAL\_LEVEL)  
VALUES (NOW(), :ecode, :emesg, 'RPT.MERGE\_SHC\_SETT\_IBFT\_200', 2);