**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)... | UPDATE JOIN (Step 3.1) + INSERT NOT EXISTS (Step 3.2) |
| 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) |

**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 3.1 – MERGE (WHEN MATCHED) → UPDATE**

UPDATE SHCLOG\_SETT\_IBFT AS A  
JOIN V\_ISOMESSAGE\_TMP\_TURN\_0200 AS B  
 ON A.PAN = B.card\_no\_trim  
 AND A.ORIGTRACE = B.trace\_no\_num  
 AND A.TERMID = B.term\_id  
 AND DATE\_FORMAT(A.LOCAL\_DATE, '%m%d') = B.local\_mmdd  
 AND A.LOCAL\_TIME = B.local\_time\_num  
 AND A.ACQUIRER = B.acq\_id\_num  
SET  
 A.RESPCODE = CASE  
 WHEN A.AMOUNT <> ROUND(B.amount\_raw\_num/100, 0) AND A.RESPCODE = 0 THEN 116  
 ELSE A.RESPCODE  
 END,  
 A.TXNSRC = CASE  
 WHEN A.AMOUNT <> ROUND(B.amount\_raw\_num/100, 0) THEN 'RC=99'  
 ELSE A.TXNSRC  
 END,  
 A.CONTENT\_FUND = B.IBFT\_INFO;

**Step 3.2 – MERGE (WHEN NOT MATCHED) → INSERT (NOT EXISTS) + DEDUP**

INSERT INTO SHCLOG\_SETT\_IBFT (  
 PAN, ORIGTRACE, TERMID, LOCAL\_DATE, LOCAL\_TIME, ACQUIRER,  
 DATA\_ID, PPCODE, MSGTYPE, PCODE, AMOUNT, ACQ\_CURRENCY\_CODE, TRACE,  
 SETTLEMENT\_DATE, ISSUER, RESPCODE, MERCHANT\_TYPE, MERCHANT\_TYPE\_ORIG, AUTHNUM,  
 SETT\_CURRENCY\_CODE, ADD\_INFO, ACCTNUM, ISS\_CURRENCY\_CODE, 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  
)  
WITH base AS (  
 SELECT \*  
 FROM (  
 SELECT  
 card\_no\_trim, trace\_no\_num, term\_id, local\_mmdd, local\_time\_num, acq\_id\_num,  
 MTI, BEN\_ID, SERVICE\_CODE, ISS\_ID, PROC\_CODE, IBFT\_INFO,  
 ACQ\_COUNTRY, POS\_ENTRY\_CODE, POS\_CONDITION\_CODE, ADDRESPONSE, MVV,  
 F4, F5, F6, F49,  
 SETTLEMENT\_CODE, SETTLEMENT\_RATE, ISS\_CONV\_RATE, TCC,  
 REF\_NO, TNX\_STAMP, CARD\_ACCEPT\_NAME\_LOCATION, CARD\_ACCEPT\_ID\_CODE,  
 SETTLE\_DATE, ACCOUNT\_NO, DEST\_ACCOUNT, MCC, APPROVAL\_CODE, ADD\_INFO,  
 amount\_ins\_num, f4\_num, f5\_num, f6\_num, trace2\_num,  
 ROW\_NUMBER() OVER (  
 PARTITION BY card\_no\_trim, trace\_no\_num, term\_id, local\_mmdd, local\_time\_num, acq\_id\_num  
 ORDER BY TNX\_STAMP DESC  
 ) rn  
 FROM V\_ISOMESSAGE\_TMP\_TURN\_0200  
 ) t  
 WHERE t.rn = 1  
),  
src AS (  
 SELECT b.\*,  
 CASE CAST(b.acq\_id\_num AS UNSIGNED)  
 WHEN 970429 THEN 157979  
 WHEN 970400 THEN 161087  
 WHEN 970427 THEN 166888  
 WHEN 970441 THEN 180906  
 WHEN 970425 THEN 970459  
 WHEN 970431 THEN 452999  
 WHEN 970436 THEN 686868  
 WHEN 970419 THEN 818188  
 WHEN 970407 THEN 888899  
 WHEN 970434 THEN 888999  
 WHEN 970440 THEN 970468  
 WHEN 970418 THEN 970488  
 WHEN 970432 THEN 981957  
 WHEN 970405 THEN 970499  
 ELSE CAST(b.acq\_id\_num AS UNSIGNED)  
 END AS mapped\_acq\_id  
 FROM base b  
)  
SELECT  
 s.card\_no\_trim,  
 s.trace\_no\_num,  
 s.term\_id,  
 STR\_TO\_DATE(CONCAT(YEAR(CURDATE()), s.local\_mmdd), '%Y%m%d'),  
 s.local\_time\_num,  
 s.acq\_id\_num,  
  
 1,  
 CAST(s.PROC\_CODE AS UNSIGNED),  
 '210',  
 CAST(s.PROC\_CODE AS UNSIGNED),  
 s.amount\_ins\_num,  
 704,  
 s.trace2\_num,  
  
 STR\_TO\_DATE(CONCAT(YEAR(CURDATE()), LPAD(s.SETTLE\_DATE,4,'0')), '%Y%m%d'),  
 s.acq\_id\_num,  
 CASE  
 WHEN s.BEN\_ID = '971133' AND s.DEST\_ACCOUNT LIKE 'NPDC%' THEN 68  
 WHEN s.SERVICE\_CODE = 'QR\_PUSH' THEN 68  
 WHEN s.BEN\_ID = '971100' AND s.TCC = '99' THEN 68  
 WHEN TRIM(s.ISS\_ID) IN ('980471','980472') THEN 68  
 ELSE 0  
 END,  
 6011,  
 CAST(s.MCC AS UNSIGNED),  
 s.APPROVAL\_CODE,  
 704,  
 s.ADD\_INFO,  
 CONCAT(IFNULL(s.ACCOUNT\_NO,' '), '|', IFNULL(s.DEST\_ACCOUNT,'')),  
 704,  
  
 CASE  
 WHEN TRIM(s.ISS\_ID) = '980471' THEN 980471  
 WHEN TRIM(s.ISS\_ID) = '980475' THEN 980478  
 WHEN s.acq\_id\_num = 191919 THEN 970459  
 WHEN s.acq\_id\_num = 970415 THEN 970489  
 ELSE s.mapped\_acq\_id  
 END,  
  
 '97',  
 704,  
 NULL, NULL, NULL, NULL,  
  
 CASE  
 WHEN s.acq\_id\_num = 191919 THEN 970459  
 WHEN s.acq\_id\_num = 970415 THEN 970489  
 ELSE s.mapped\_acq\_id  
 END,  
  
 'IBT',  
  
 CASE  
 WHEN TRIM(s.ISS\_ID) = '980472' THEN 980471  
 WHEN TRIM(s.ISS\_ID) = '980474' THEN 980478  
 WHEN s.BEN\_ID IS NOT NULL AND s.PROC\_CODE IN ('912020','910020')  
 THEN CASE  
 WHEN CAST(i\_ben.MEMBER\_ID AS UNSIGNED) = 191919 THEN 970459  
 WHEN CAST(i\_ben.MEMBER\_ID AS UNSIGNED) = 970415 THEN 970489  
 ELSE CAST(i\_ben.MEMBER\_ID AS UNSIGNED)  
 END  
 ELSE CASE  
 WHEN CAST(i\_dest.MEMBER\_ID AS UNSIGNED) = 191919 THEN 970459  
 WHEN CAST(i\_dest.MEMBER\_ID AS UNSIGNED) = 970415 THEN 970489  
 ELSE CAST(i\_dest.MEMBER\_ID AS UNSIGNED)  
 END  
 END,  
  
 CASE  
 WHEN s.BEN\_ID IN (SELECT TGTT\_ID FROM TGTT\_20) THEN CAST(s.BEN\_ID AS UNSIGNED)  
 WHEN TRIM(s.ISS\_ID) IN ('980472','980474','980475') THEN  
 CASE  
 WHEN s.BEN\_ID IS NOT NULL AND s.PROC\_CODE IN ('912020','910020')  
 THEN CASE  
 WHEN CAST(i\_ben.MEMBER\_ID AS UNSIGNED) = 191919 THEN 970459  
 WHEN CAST(i\_ben.MEMBER\_ID AS UNSIGNED) = 970415 THEN 970489  
 ELSE CAST(i\_ben.MEMBER\_ID AS UNSIGNED)  
 END  
 ELSE CASE  
 WHEN CAST(i\_dest.MEMBER\_ID AS UNSIGNED) = 191919 THEN 970459  
 WHEN CAST(i\_dest.MEMBER\_ID AS UNSIGNED) = 970415 THEN 970489  
 ELSE CAST(i\_dest.MEMBER\_ID AS UNSIGNED)  
 END  
 END  
 ELSE CAST(s.BEN\_ID AS UNSIGNED)  
 END,  
  
 s.IBFT\_INFO,  
 'MTI=200',  
 s.ACQ\_COUNTRY, s.POS\_ENTRY\_CODE, s.POS\_CONDITION\_CODE, s.ADDRESPONSE, s.MVV,  
 s.f4\_num, s.f5\_num, s.f6\_num, s.F49,  
 s.SETTLEMENT\_CODE, s.SETTLEMENT\_RATE, s.ISS\_CONV\_RATE, s.TCC,  
 s.REF\_NO, DATE(s.TNX\_STAMP), DATE\_FORMAT(s.TNX\_STAMP, '%H%i%s'),  
 s.CARD\_ACCEPT\_NAME\_LOCATION, s.CARD\_ACCEPT\_ID\_CODE,  
 NULL, CAST(s.PROC\_CODE AS UNSIGNED),  
 s.ACCOUNT\_NO, s.DEST\_ACCOUNT  
FROM src s  
LEFT JOIN IBFT\_BANK\_BINS i\_ben  
 ON i\_ben.BIN = s.BEN\_ID  
LEFT JOIN IBFT\_BANK\_BINS i\_dest  
 ON i\_dest.BIN = SUBSTRING(s.DEST\_ACCOUNT,1,6)  
WHERE NOT EXISTS (  
 SELECT 1  
 FROM SHCLOG\_SETT\_IBFT A  
 WHERE A.PAN = s.card\_no\_trim  
 AND A.ORIGTRACE = s.trace\_no\_num  
 AND A.TERMID = s.term\_id  
 AND DATE\_FORMAT(A.LOCAL\_DATE,'%m%d') = s.local\_mmdd  
 AND A.LOCAL\_TIME = s.local\_time\_num  
 AND A.ACQUIRER = s.acq\_id\_num  
);

**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);