CREATE OR REPLACE TABLE F\_ORDER\_RETURN\_SILVER (

claim\_id STRING,

order\_id STRING,

product\_id STRING,

order\_date TIMESTAMP\_TZ,

return\_date TIMESTAMP\_TZ,

return\_reason STRING,

severity STRING,

under\_warranty BOOLEAN,

ingestion\_date DATE

);

CREATE OR REPLACE TABLE F\_ORDER\_RETURN\_REJECTS (

raw\_record OBJECT,

reason STRING,

logged\_at TIMESTAMP\_TZ DEFAULT CURRENT\_TIMESTAMP()

);

DROP STREAM IF EXISTS ARBORE\_WARRANTY\_CLAIMS\_STRM;

CREATE STREAM ARBORE\_WARRANTY\_CLAIMS\_STRM

ON TABLE ARBORE\_WARRANTY\_CLAIMS;

CREATE OR REPLACE PROCEDURE SP\_LOAD\_F\_ORDER\_RETURN\_SILVER\_FROM\_PARSED\_V1()

RETURNS STRING

LANGUAGE SQL

EXECUTE AS OWNER

AS

$$

BEGIN

CREATE OR REPLACE TEMP TABLE tmp\_parsed AS

SELECT

TRIM(claim\_id) AS claim\_id,

TRIM(order\_id) AS order\_id,

TRIM(product\_id) AS product\_id,

COALESCE(

TRY\_TO\_TIMESTAMP\_TZ(order\_date, 'MM/DD/YYYY'),

TRY\_TO\_TIMESTAMP\_TZ(order\_date, 'YYYY-MM-DD'),

TRY\_TO\_TIMESTAMP\_TZ(order\_date)

) AS order\_date\_ts,

COALESCE(

TRY\_TO\_TIMESTAMP\_TZ(return\_date, 'YYYY-MM-DD'),

TRY\_TO\_TIMESTAMP\_TZ(return\_date, 'MM/DD/YYYY'),

TRY\_TO\_TIMESTAMP\_TZ(return\_date)

) AS return\_date\_ts,

NULLIF(TRIM(return\_reason), '') AS return\_reason,

UPPER(TRIM(severity)) AS severity\_up,

UPPER(TRIM(under\_warranty)) AS uw\_str,

OBJECT\_CONSTRUCT(

'claim\_id', claim\_id,

'order\_id', order\_id,

'product\_id', product\_id,

'order\_date', order\_date,

'return\_date', return\_date,

'return\_reason', return\_reason,

'severity', severity,

'under\_warranty', under\_warranty

) AS raw\_full

FROM ARBORE\_WARRANTY\_CLAIMS\_STRM

WHERE METADATA$ACTION IN ('INSERT');

CREATE OR REPLACE TEMP TABLE tmp\_normalized AS

SELECT

claim\_id, order\_id, product\_id, order\_date\_ts, return\_date\_ts, return\_reason, raw\_full,

CASE

WHEN severity\_up IN ('MINOR','MAJOR','CRITICAL') THEN severity\_up

WHEN severity\_up IS NULL OR severity\_up = '' THEN NULL

ELSE 'OTHER'

END AS severity,

CASE

WHEN uw\_str IN ('1','Y','YES','TRUE') THEN TRUE

WHEN uw\_str IN ('0','N','NO','FALSE') THEN FALSE

ELSE NULL

END AS under\_warranty\_bool

FROM tmp\_parsed;

CREATE OR REPLACE TEMP TABLE tmp\_validated AS

SELECT

\*,

IFF(claim\_id IS NULL, 'claim\_id is NULL',

IFF(order\_id IS NULL, 'order\_id is NULL',

IFF(product\_id IS NULL, 'product\_id is NULL',

IFF(order\_date\_ts IS NULL, 'order\_date invalid',

IFF(return\_date\_ts IS NULL, 'return\_date invalid',

IFF(return\_date\_ts < order\_date\_ts, 'return\_date before order\_date',

IFF(under\_warranty\_bool IS NULL, 'under\_warranty undecodable', NULL)

)

)

)

)

)

) AS reject\_reason

FROM tmp\_normalized;

INSERT INTO F\_ORDER\_RETURN\_REJECTS (raw\_record, reason)

SELECT raw\_full, reject\_reason

FROM tmp\_validated

WHERE reject\_reason IS NOT NULL;

CREATE OR REPLACE TEMP TABLE tmp\_good AS

SELECT

claim\_id,

order\_id,

product\_id,

order\_date\_ts AS order\_date,

return\_date\_ts AS return\_date,

return\_reason,

severity,

under\_warranty\_bool AS under\_warranty,

CURRENT\_DATE() AS ingestion\_date

FROM tmp\_validated

WHERE reject\_reason IS NULL;

CREATE OR REPLACE TEMP TABLE tmp\_dedup AS

SELECT \*

FROM (

SELECT

g.\*,

ROW\_NUMBER() OVER (PARTITION BY claim\_id ORDER BY return\_date DESC NULLS LAST) AS rn

FROM tmp\_good g

)

WHERE rn = 1;

MERGE INTO F\_ORDER\_RETURN\_SILVER AS tgt

USING tmp\_dedup AS src

ON tgt.claim\_id = src.claim\_id

WHEN MATCHED THEN

UPDATE SET

tgt.order\_id = src.order\_id,

tgt.product\_id = src.product\_id,

tgt.order\_date = src.order\_date,

tgt.return\_date = src.return\_date,

tgt.return\_reason = src.return\_reason,

tgt.severity = src.severity,

tgt.under\_warranty = src.under\_warranty,

tgt.ingestion\_date = src.ingestion\_date

WHEN NOT MATCHED THEN

INSERT (

claim\_id, order\_id, product\_id, order\_date, return\_date,

return\_reason, severity, under\_warranty, ingestion\_date

)

VALUES (

src.claim\_id, src.order\_id, src.product\_id, src.order\_date, src.return\_date,

src.return\_reason, src.severity, src.under\_warranty, src.ingestion\_date

);

RETURN 'OK';

END;

$$;

CALL SP\_LOAD\_F\_ORDER\_RETURN\_SILVER\_FROM\_PARSED\_V1();

SELECT \* FROM F\_ORDER\_RETURN\_SILVER ORDER BY return\_date DESC LIMIT 20;

SELECT reason, COUNT(\*) FROM F\_ORDER\_RETURN\_REJECTS GROUP BY 1 ORDER BY 2 DESC;

select distinct product\_id from bronze\_orders\_parsed;