Query 1 : (my answer CORRECT)

1. Count of records in Test\_Edw (Target for Stackline) being more than the source (Sales/Promotions/Content score/Rating Reviews) – Please refer to Variant # 1 Tab for counts reference. Could you help us understand why the count could be more than the source?

Result 1:

PROD\_RAW (SALES ) – 866259196

PROD\_RAW (PROMOTIONS) – 48095276

PROD\_RAW(CONTENTSCORE) – 854622829

PROD\_RAW(RATINGSREVIEWS) – 855769772

TOTAL\_COUNT\_OF\_RECORDS IS COMING AROUND (2624747073)\

Target : 881236622

Result 2:

-- ORIGINAL VIEW CODE

WITH STACKLINE\_RETAILERS AS

( SELECT RETAILERID,

RETAILERSKU,

WEEKID

FROM "PROD\_RAW"."STACKLINE"."SALES"

UNION

SELECT RETAILERID,

RETAILERSKU,

WEEKID

FROM "PROD\_RAW"."STACKLINE"."PROMOTIONS"

UNION

SELECT RETAILERID,

RETAILERSKU,

WEEKID

FROM "PROD\_RAW"."STACKLINE"."RATINGSREVIEWS"

UNION

SELECT RETAILERID,

RETAILERSKU,

WEEKID

FROM "PROD\_RAW"."STACKLINE"."CONTENTSCORE"

)

SELECT COUNT(\*) as SOURCE\_COUNT FROM STACKLINE\_RETAILERS AS R

Output : 881236622

3. select COUNT(\*) as "Target\_Count" from TEST\_EDW.CONSOLIDATED.EDW\_PRICING\_COMPETITOR\_SALES WHERE SRC\_SYS\_KEY = 'STACKLINE'

Output : 881236622

So, the count is getting varied in individual tables in prod\_raw (NEED TO CHECK)

QUERY 2: (APPROCH IS GOOD) AND I DIDN’T TAKE RETAILPRICE <> RP OF OTHER TABLE

Currently how do you consider the Retail Sale price for a unique record(Weekid/SKU/Retailer-id) should you encounter a price mismatch between Sales and Promotions table?  Please let us know – if you have encountered any examples please do share the PRC\_COMPTTR\_SLS\_KEY as well.

RETAILS SALES GETTING MAPPED WITH TOT\_PRC

I AM NOT ABLE TO SEE ANY COLUMN RETAILS SALES PRICE (IS THAT RETAILPRICE)

RES1 :

select "PROD\_RAW"."STACKLINE"."PROMOTIONS".RETAILPRICE AS PROMO\_RP,

"PROD\_RAW"."STACKLINE"."SALES".RETAILPRICE AS SALES\_RP

from "PROD\_RAW"."STACKLINE"."PROMOTIONS" INNER JOIN "PROD\_RAW"."STACKLINE"."SALES"

on "PROD\_RAW"."STACKLINE"."PROMOTIONS".WEEKID = "PROD\_RAW"."STACKLINE"."SALES".WEEKID

AND "PROD\_RAW"."STACKLINE"."PROMOTIONS".RETAILERID = "PROD\_RAW"."STACKLINE"."SALES".RETAILERID

AND "PROD\_RAW"."STACKLINE"."PROMOTIONS".RETAILERSKU = "PROD\_RAW"."STACKLINE"."SALES".RETAILERSKU

AND "PROD\_RAW"."STACKLINE"."PROMOTIONS".RETAILPRICE <> "PROD\_RAW"."STACKLINE"."SALES".RETAILPRICE

LIMIT 10

RES2:

with tb1 as (

select RETAILPRICE, SUM(RETAILPRICE) as sell1

from "PROD\_RAW"."STACKLINE"."PROMOTIONS"

group by RETAILPRICE

),

tb2 as (

select RETAILPRICE, SUM(RETAILPRICE) as sell2

from "PROD\_RAW"."STACKLINE"."SALES"

group by RETAILPRICE

)

select tb1.RETAILPRICE, tb1.sell1, (tb2.SELL2 - tb1.sell1) as profit

from tb1 left join tb2 on tb1.RETAILPRICE=tb2.RETAILPRICE

QUERY3 : MY Answer CORRECT

The date in the field ‘SRC\_RCRD\_CREATE\_DTE’ seems to be incorrect -  Please refer to ‘Data Record Samples’ tab for the date values.

Opinion1 :

Here in stackline dbt model code , we have this line stating

COALESCE(SA.EVENTDTS,P.EVENTDTS,CS.EVENTDTS,RR.EVENTDTS) AS      {{var('column\_SRC\_RCRD\_CREATE\_DTE')}},

The value getting stored in EVENTDTS @ SALES : 20210403000000000000

EVENTDTS DATATYPE IS VARCHAR(16777216)

SRC\_RCRD\_CREATE\_DTE DATATYPE IS TIMESTAMP

So my doubt is should we include

SA.LOADDTS,'2099-12-31'

Something like this here in EVENTDTS

OR

CURRENT\_TIMESTAMP::TIMESTAMP\_NTZ

OR

COALESCE(SA.EVENTDTS,P.EVENTDTS,CS.EVENTDTS,RR.EVENTDTS)

SA.EVENTDTS AS ::VARCHAR

TO DISPLAY SRC\_RCRD\_CREATE\_DTE IN TIMESTAMP (FORMAT GIVEN BY THEM) 10/06/2610 17:46:40

POINTS IN SCRUM:

desc table "PROD\_RAW"."STACKLINE"."SALES"

desc table "PROD\_RAW"."STACKLINE"."PROMOTIONS"

SELECT S.RETAILERID,

S.RETAILERSKU,

S.WEEKID,

CONCAT(COALESCE(S.RETAILERSKU::VARCHAR,''),

'~',COALESCE(S.RETAILERID::VARCHAR,''),

'~',COALESCE(S.WEEKID::VARCHAR,'')) AS SLS\_KEY,

S.RETAILPRICE AS SALES\_RETAILPRICE,

P.RETAILPRICE AS PROMOTIONS\_RETAILPRICE

FROM "PROD\_RAW"."STACKLINE"."SALES" S INNER JOIN "PROD\_RAW"."STACKLINE"."PROMOTIONS" P

ON S.RETAILERID = P.RETAILERID

AND S.RETAILERSKU = P.RETAILERSKU

AND S.WEEKID = P.WEEKID

WHERE S.RETAILPRICE <> P.RETAILPRICE

AND SLS\_KEY = 'B093Q66QQX~1~202210'

LIMIT 10

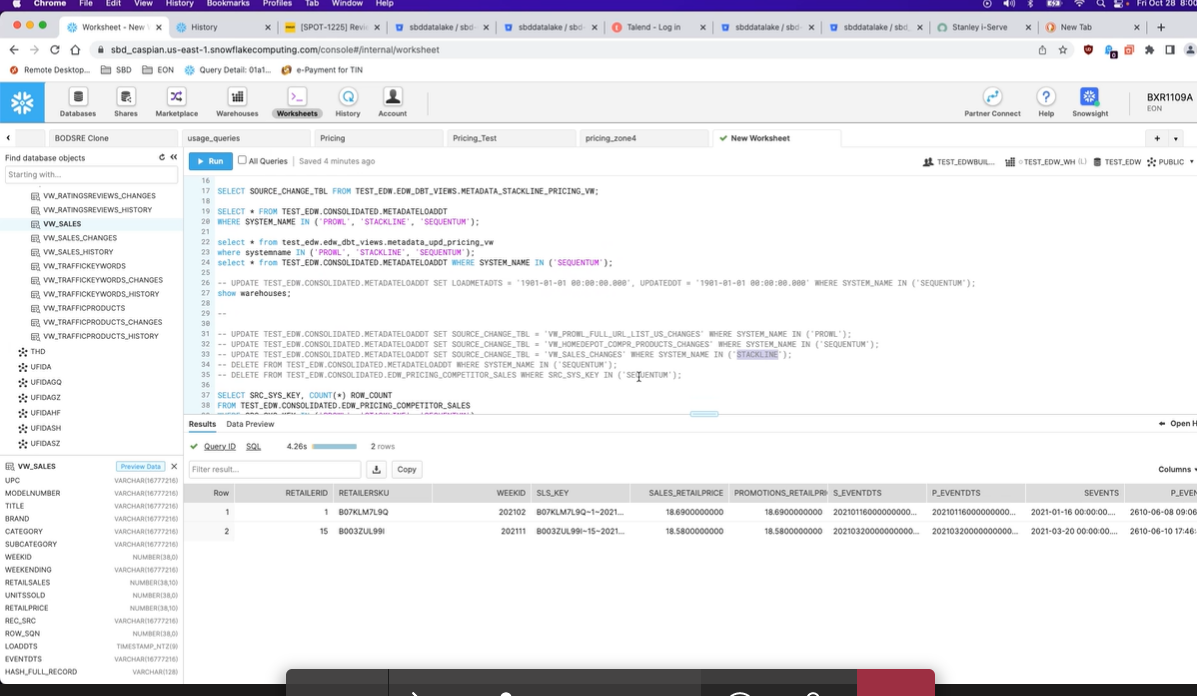
SELECT \*

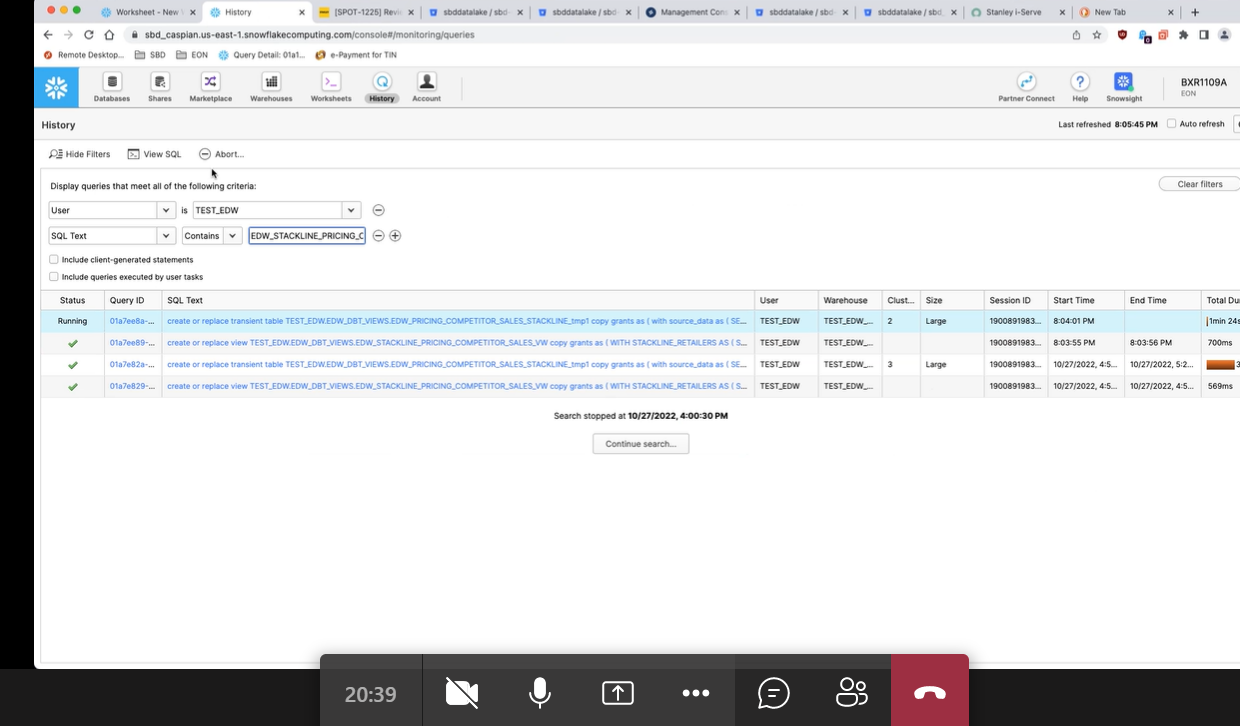
FROM TEST\_EDW.CONSOLIDATED.EDW\_PRICING\_COMPETITOR\_SALES

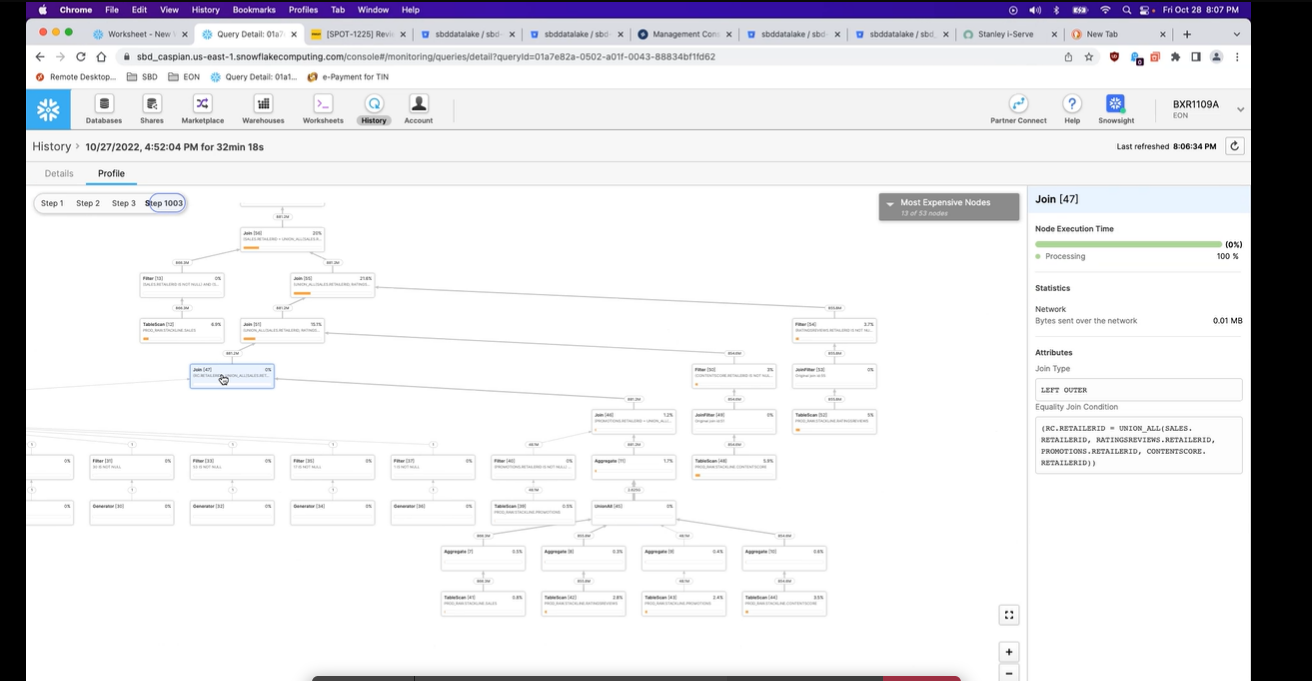
WHERE SRC\_SYS\_KEY = 'STACKLINE'

AND PRC\_COMPTTR\_SLS\_KEY = 'B093Q66QQX~1~202210';

once code is changed we need to delete the records from the source system and reload using talend TMC







select RCRD\_HASH\_KEY from TEST\_EDW.CONSOLIDATED.EDW\_PRICING\_COMPETITOR\_SALES

WHERE SRC\_SYS\_KEY = 'STACKLINE' AND RCRD\_HASH\_KEY NOT IN

(SELECT(md5(CONCAT(COALESCE(P.RETAILERSKU::VARCHAR,''),

'~',COALESCE(P.RETAILERID::VARCHAR,''),

'~',COALESCE(P.WEEKID::VARCHAR,''))))

FROM "PROD\_RAW"."STACKLINE"."PROMOTIONS" AS P)

UNION

select RCRD\_HASH\_KEY from TEST\_EDW.CONSOLIDATED.EDW\_PRICING\_COMPETITOR\_SALES

WHERE SRC\_SYS\_KEY = 'STACKLINE' AND RCRD\_HASH\_KEY NOT IN

(SELECT(md5(CONCAT(COALESCE(S.RETAILERSKU::VARCHAR,''),

'~',COALESCE(S.RETAILERID::VARCHAR,''),

'~',COALESCE(S.WEEKID::VARCHAR,''))))

FROM "PROD\_RAW"."STACKLINE"."SALES" AS S)

UNION

select RCRD\_HASH\_KEY from TEST\_EDW.CONSOLIDATED.EDW\_PRICING\_COMPETITOR\_SALES

WHERE SRC\_SYS\_KEY = 'STACKLINE' AND RCRD\_HASH\_KEY NOT IN

(SELECT(md5(CONCAT(COALESCE(R.RETAILERSKU::VARCHAR,''),

'~',COALESCE(R.RETAILERID::VARCHAR,''),

'~',COALESCE(R.WEEKID::VARCHAR,''))))

FROM "PROD\_RAW"."STACKLINE"."RATINGSREVIEWS" AS R)

UNION

select RCRD\_HASH\_KEY from TEST\_EDW.CONSOLIDATED.EDW\_PRICING\_COMPETITOR\_SALES

WHERE SRC\_SYS\_KEY = 'STACKLINE' AND RCRD\_HASH\_KEY NOT IN

(SELECT(md5(CONCAT(COALESCE(CS.RETAILERSKU::VARCHAR,''),

'~',COALESCE(CS.RETAILERID::VARCHAR,''),

'~',COALESCE(CS.WEEKID::VARCHAR,''))))

FROM "PROD\_RAW"."STACKLINE"."CONTENTSCORE" AS CS)

BALA’S QUERY:

-- new one

WITH STACKLINE\_RETAILERS AS

( SELECT RETAILERID,

RETAILERSKU,

WEEKID,

'#' as SLS\_PROMO\_TYPE

FROM "PROD\_RAW"."STACKLINE"."SALES"

-- WHERE WEEKID >= 202231

UNION

SELECT RETAILERID,

RETAILERSKU,

WEEKID,

PROMOTYPE as SLS\_PROMO\_TYPE

FROM "PROD\_RAW"."STACKLINE"."PROMOTIONS"

--WHERE WEEKID >= 202231

UNION

SELECT RETAILERID,

RETAILERSKU,

WEEKID,

'#' as SLS\_PROMO\_TYPE

FROM "PROD\_RAW"."STACKLINE"."RATINGSREVIEWS"

--WHERE WEEKID >= 202231

UNION

SELECT RETAILERID,

RETAILERSKU,

WEEKID,

'#' as SLS\_PROMO\_TYPE

FROM "PROD\_RAW"."STACKLINE"."CONTENTSCORE"

-- WHERE WEEKID >= 202230

)

SELECT SUM(SOURCE\_COUNT) AS PROD\_RAW\_SOURCE\_COUNT,

SUM(TARGET\_COUNT) AS TEST\_EDW\_TARGET\_COUNT

FROM (

SELECT COUNT(\*) as SOURCE\_COUNT, sum(0) as TARGET\_COUNT FROM STACKLINE\_RETAILERS AS R

UNION ALL

SELECT SUM(0) as SOURCE\_COUNT, COUNT(\*) as TARGET\_COUNT FROM test\_EDW.CONSOLIDATED.EDW\_PRICING\_COMPETITOR\_SALES

where SRC\_SYS\_KEY = 'STACKLINE'

and CURR\_RCRD\_FLAG = 'Y'

);

TARGET AND SOURCE COUNT QUERY (USING REC\_HASK\_KEY AND CONCAT(4 COLS) AS PK

select count(RCRD\_HASH\_KEY) as target from TEST\_EDW.CONSOLIDATED.EDW\_PRICING\_COMPETITOR\_SALES

WHERE SRC\_SYS\_KEY = 'STACKLINE'

UNION

(SELECT(count(md5(CONCAT(COALESCE(S.RETAILERSKU::VARCHAR,''),

'~',COALESCE(S.RETAILERID::VARCHAR,''),

'~',COALESCE(S.WEEKID::VARCHAR,''),

'~',COALESCE('#'::VARCHAR,'')))))

FROM "PROD\_RAW"."STACKLINE"."SALES" AS S WHERE WEEKID >= 202231)

UNION

(SELECT(count(md5(CONCAT(COALESCE(P.RETAILERSKU::VARCHAR,''),

'~',COALESCE(P.RETAILERID::VARCHAR,''),

'~',COALESCE(P.WEEKID::VARCHAR,''),

'~',COALESCE(P.PROMOTYPE::VARCHAR,'')))))

FROM "PROD\_RAW"."STACKLINE"."PROMOTIONS" AS P WHERE WEEKID >= 202231)

UNION

(SELECT(count(md5(CONCAT(COALESCE(CS.RETAILERSKU::VARCHAR,''),

'~',COALESCE(CS.RETAILERID::VARCHAR,''),

'~',COALESCE(CS.WEEKID::VARCHAR,''),

'~',COALESCE('#'::VARCHAR,'')))))

FROM "PROD\_RAW"."STACKLINE"."CONTENTSCORE" AS CS WHERE WEEKID >= 202231)

UNION

(SELECT(count(md5(CONCAT(COALESCE(R.RETAILERSKU::VARCHAR,''),

'~',COALESCE(R.RETAILERID::VARCHAR,''),

'~',COALESCE(R.WEEKID::VARCHAR,''),

'~',COALESCE('#'::VARCHAR,'')))))

FROM "PROD\_RAW"."STACKLINE"."RATINGSREVIEWS" AS R WHERE WEEKID >= 202231)

METHOD 3:

QUERY TO FIND SOURCE AND TARGET (WITHOUT SOURCE COUNT COUNT(MD5()))

with output as (

select RCRD\_HASH\_KEY from TEST\_EDW.CONSOLIDATED.EDW\_PRICING\_COMPETITOR\_SALES

WHERE SRC\_SYS\_KEY = 'STACKLINE' AND RCRD\_HASH\_KEY IN

(SELECT(md5(CONCAT(COALESCE(S.RETAILERSKU::VARCHAR,''),

'~',COALESCE(S.RETAILERID::VARCHAR,''),

'~',COALESCE(S.WEEKID::VARCHAR,''),

'~',COALESCE('#'::VARCHAR,''))))

FROM "PROD\_RAW"."STACKLINE"."SALES" AS S)

UNION

select RCRD\_HASH\_KEY from TEST\_EDW.CONSOLIDATED.EDW\_PRICING\_COMPETITOR\_SALES

WHERE SRC\_SYS\_KEY = 'STACKLINE' AND RCRD\_HASH\_KEY IN

(SELECT(md5(CONCAT(COALESCE(P.RETAILERSKU::VARCHAR,''),

'~',COALESCE(P.RETAILERID::VARCHAR,''),

'~',COALESCE(P.WEEKID::VARCHAR,''),

'~',COALESCE(P.PROMOTYPE::VARCHAR,''))))

FROM "PROD\_RAW"."STACKLINE"."PROMOTIONS" AS P)

UNION

select RCRD\_HASH\_KEY from TEST\_EDW.CONSOLIDATED.EDW\_PRICING\_COMPETITOR\_SALES

WHERE SRC\_SYS\_KEY = 'STACKLINE' AND RCRD\_HASH\_KEY IN

(SELECT(md5(CONCAT(COALESCE(R.RETAILERSKU::VARCHAR,''),

'~',COALESCE(R.RETAILERID::VARCHAR,''),

'~',COALESCE(R.WEEKID::VARCHAR,''),

'~',COALESCE('#'::VARCHAR,''))))

FROM "PROD\_RAW"."STACKLINE"."RATINGSREVIEWS" AS R)

UNION

select RCRD\_HASH\_KEY from TEST\_EDW.CONSOLIDATED.EDW\_PRICING\_COMPETITOR\_SALES

WHERE SRC\_SYS\_KEY = 'STACKLINE' AND RCRD\_HASH\_KEY IN

(SELECT(md5(CONCAT(COALESCE(CS.RETAILERSKU::VARCHAR,''),

'~',COALESCE(CS.RETAILERID::VARCHAR,''),

'~',COALESCE(CS.WEEKID::VARCHAR,''),

'~',COALESCE('#'::VARCHAR,''))))

FROM "PROD\_RAW"."STACKLINE"."CONTENTSCORE" AS CS)

)

select count(\*) as Target\_Count from output - 91722991

|  |
| --- |
| SELECT SUM(SOURCE\_COUNT) AS PROD\_RAW\_SOURCE\_COUNT, |
| SUM(TARGET\_COUNT) AS TEST\_EDW\_TARGET\_COUNT |
| FROM ( |
| SELECT COUNT(\*) as SOURCE\_COUNT, sum(0) as TARGET\_COUNT FROM  output AS R |
| UNION ALL |
| SELECT SUM(0) as SOURCE\_COUNT, COUNT(\*) as TARGET\_COUNT FROM test\_EDW.CONSOLIDATED.EDW\_PRICING\_COMPETITOR\_SALES |
| where SRC\_SYS\_KEY = 'STACKLINE' |
| and CURR\_RCRD\_FLAG = 'Y' |

FINAL CODE

with output as (

select RCRD\_HASH\_KEY from TEST\_EDW.CONSOLIDATED.EDW\_PRICING\_COMPETITOR\_SALES

WHERE SRC\_SYS\_KEY = 'STACKLINE' AND RCRD\_HASH\_KEY IN

(SELECT(md5(CONCAT(COALESCE(S.RETAILERSKU::VARCHAR,''),

'~',COALESCE(S.RETAILERID::VARCHAR,''),

'~',COALESCE(S.WEEKID::VARCHAR,''),

'~',COALESCE('#'::VARCHAR,''))))

FROM "PROD\_RAW"."STACKLINE"."SALES" AS S)

UNION

select RCRD\_HASH\_KEY from TEST\_EDW.CONSOLIDATED.EDW\_PRICING\_COMPETITOR\_SALES

WHERE SRC\_SYS\_KEY = 'STACKLINE' AND RCRD\_HASH\_KEY IN

(SELECT(md5(CONCAT(COALESCE(P.RETAILERSKU::VARCHAR,''),

'~',COALESCE(P.RETAILERID::VARCHAR,''),

'~',COALESCE(P.WEEKID::VARCHAR,''),

'~',COALESCE(P.PROMOTYPE::VARCHAR,''))))

FROM "PROD\_RAW"."STACKLINE"."PROMOTIONS" AS P)

UNION

select RCRD\_HASH\_KEY from TEST\_EDW.CONSOLIDATED.EDW\_PRICING\_COMPETITOR\_SALES

WHERE SRC\_SYS\_KEY = 'STACKLINE' AND RCRD\_HASH\_KEY IN

(SELECT(md5(CONCAT(COALESCE(R.RETAILERSKU::VARCHAR,''),

'~',COALESCE(R.RETAILERID::VARCHAR,''),

'~',COALESCE(R.WEEKID::VARCHAR,''),

'~',COALESCE('#'::VARCHAR,''))))

FROM "PROD\_RAW"."STACKLINE"."RATINGSREVIEWS" AS R)

UNION

select RCRD\_HASH\_KEY from TEST\_EDW.CONSOLIDATED.EDW\_PRICING\_COMPETITOR\_SALES

WHERE SRC\_SYS\_KEY = 'STACKLINE' AND RCRD\_HASH\_KEY IN

(SELECT(md5(CONCAT(COALESCE(CS.RETAILERSKU::VARCHAR,''),

'~',COALESCE(CS.RETAILERID::VARCHAR,''),

'~',COALESCE(CS.WEEKID::VARCHAR,''),

'~',COALESCE('#'::VARCHAR,''))))

FROM "PROD\_RAW"."STACKLINE"."CONTENTSCORE" AS CS)

)

SELECT COUNT(\*) as SOURCE\_COUNT, sum(0) as TARGET\_COUNT FROM output AS R

UNION ALL

SELECT SUM(0) as SOURCE\_COUNT, COUNT(\*) as TARGET\_COUNT FROM test\_EDW.CONSOLIDATED.EDW\_PRICING\_COMPETITOR\_SALES

where SRC\_SYS\_KEY = 'STACKLINE'

and CURR\_RCRD\_FLAG = 'Y'

