**Vip\_dc**

1. Truncate table

\*\*\*

TRUNCATE TABLE [dbo].[stg\_VIP\_DC\_EMAIL]

\*\*\*

1. Select and Insert Email Data, then store in temp table

\*\*\*

SELECT TOP 1 WITH TIES ID

, BRAND COLLATE SQL\_Latin1\_General\_CP1\_CI\_AS AS BRAND

, VIPCODE COLLATE SQL\_Latin1\_General\_CP1\_CI\_AS AS VIPCODE

, EMAIL COLLATE SQL\_Latin1\_General\_CP1\_CI\_AS AS EMAIL

, FIXED\_TYPO\_EMAIL COLLATE SQL\_Latin1\_General\_CP1\_CI\_AS AS FIXED\_TYPO\_EMAIL

, EMAIL\_STATUS COLLATE SQL\_Latin1\_General\_CP1\_CI\_AS AS EMAIL\_STATUS

, DOMAIN COLLATE SQL\_Latin1\_General\_CP1\_CI\_AS AS DOMAIN

, FREE\_DOMAIN COLLATE SQL\_Latin1\_General\_CP1\_CI\_AS AS FREE\_DOMAIN

, STATUS\_CODE COLLATE SQL\_Latin1\_General\_CP1\_CI\_AS AS STATUS\_CODE

, BOUNCED\_REASON COLLATE SQL\_Latin1\_General\_CP1\_CI\_AS AS BOUNCED\_REASON

, CREATED\_DATE

, MODIFIED\_DATE

FROM OneCX.VIP\_DC\_EMAIL

WHERE CREATED\_DATE >= DATEADD(DAY, DATEDIFF(DAY, 0, GETDATE()), - 30)

OR MODIFIED\_DATE >= DATEADD(DAY, DATEDIFF(DAY, 0, GETDATE()), - 30)

ORDER BY ROW\_NUMBER() OVER (PARTITION BY BRAND, VIPCODE ORDER BY MODIFIED\_DATE DESC, CREATED\_DATE DESC, ID DESC)

\*\*\*

1. Execute stored data

\*\*\*

EXEC [dbo].[sp\_Sync\_VIP\_DC\_EMAIL]

\*\*\*

**Bonus\_redeem**

1. Truncate table
2. Converting data to string
3. Locating of the destination table where data is inserted
4. EXEC stored table into the destination table

**Ixs\_customer\_item\_xref**

1. Truncate table
2. Insert data into table

**Onecx\_ec\_data\_sync**

1. Execute stored data
2. Select and Insert Data, then store in temp table
3. Union 2 created source tables
4. Execute stored data

**Ixs\_dim\_orc\_prod\_hier:**

All same

**SocialMemberRaw:**

1. DELETE FROM, WHERE

\*\*\*

DELETE

FROM [dbo].[SocialMemberRaw]

WHERE [ID] >= 1000000000

AND (

([Brand] IN ('BES') AND ([ModifiedDate] >= '2019-12-04' OR [CreatedDate] >= '2019-12-04'))

OR

([Brand] IN ('BPI', 'DGB', 'SLP', 'SFD') AND ([ModifiedDate] >= '2020-03-10' OR [CreatedDate] >= '2020-03-10'))

OR

([Brand] IN ('IPS') AND ([ModifiedDate] >= '2020-04-15' OR [CreatedDate] >= '2020-04-15'))

OR

([Brand] IN ('NAR') AND ([ModifiedDate] >= '2020-04-27' OR [CreatedDate] >= '2020-04-27'))

OR

([Brand] IN ('ZOT', 'PBD') AND ([ModifiedDate] >= '2020-06-29' OR [CreatedDate] >= '2020-06-29'))

OR

([Brand] IN ('CPC', 'ETS', 'ZAS') AND ([ModifiedDate] >= '2020-07-21' OR [CreatedDate] >= '2020-07-21'))

OR

([Brand] IN ('LMR') AND ([ModifiedDate] >= '2020-09-28' OR [CreatedDate] >= '2020-09-28'))

OR

([Brand] IN ('CPB') AND ([ModifiedDate] >= '2021-03-09' OR [CreatedDate] >= '2021-03-09'))

OR

([Brand] IN ('SSD','DKE') AND ([ModifiedDate] >= '2021-07-13' OR [CreatedDate] >= '2021-07-13'))

)

\*\*\*

1. Data conversion
2. DB destination (same)

**Nonvipcoo:**

1. DELETE FROM, WHERE (same)
2. Derived Column: Replace the existing column by the expression generated.

\*\*\*

Derived Column Transformation Editor Breakdown

1. Derived Column Name:

• ID: This is the name of the column being created or updated.

2. Derived Column:

• Replace ‘ID’: This indicates that the existing ID column will be replaced with the new value generated by the expression.

3. Expression:

• ID + 1000000000: This is the expression used to calculate the new value for the ID column. It adds 1000000000 to the current value of the ID column.

4. Data Type:

• four-byte signed integer [DT\_I4]: This is the data type of the ID column after the transformation. It specifies that the ID column will be a 32-bit integer.

\*\*\*

1. Data conversion (same)
2. DB destination (same)

**CRM\_VIP:**

1. All same
2. ISNULL (step 4.2)

\*\*\*

It checks for null values in a single column: XF\_DESCRIPTION\_X. If the value is null, the ISNULL function will likely return an empty string (depending on the specific database implementation). If the value is not null, it will return the actual value in the column. The concatenation operator (||) then joins all the non-null values together into a single string.

\*\*\*

**FOCUS\_ORDER:**

1. Truncate

\*\*\*

truncate table [dbo].[stg\_F\_OrderHeader\_ECOMM]

\*\*\*

1. Execute

\*\*\*

Execute exec [dbo].[sp\_Sync\_F\_OrderHeader\_Ecomm] job

\*\*\*

**FOCUS\_Dimension:**

All same

**Dimensions:**

All Same

**Facts:**

1. ISNULL + Concatenate

\*\*\*

ISNULL(ID) || ISNULL(Brand) || ISNULL(VIPcode) || ISNULL(Name) || ISNULL(Grade) || ISNULL(JoinDate) || ISNULL(CreatedDate)

\*\*\*

**Summary:**

1. **TRUNCATE**
2. **SELECT … FROM …**
   1. **COLLATE**
   2. **CAST**
3. **WHERE** （In, Datediff)
   1. STRING\_SPLIT

- String\_to\_array function.

1. **DATA CONVERSION**

- Tested via changing data type from integer to string.

1. **DATA DESTINATION**

- Built-in function in Airflow-BigQuery Insert Job operator. No need for testing.

1. **UNION FROM 2 SOURCE**

**-** UNION from 2 table, then select non-null rows

1. **DELETE FROM**
2. **DERIVED TABLE (alter column value directly)**

**-** Update function: Directly set {column} = ({column} + 10000000)

1. **ISNULL + CONCATENATE**

**-** Update function: Directly concat null column with non-null column to test both functions.

1. **EXECUTE**