Email\_engage\_aggr

Runs bteq :/infosrvr/app/edw/source/email\_engage\_aggr/bteq/email\_engage\_aggr\_crt\_prev\_mn.bteq

Runs datastage job : email\_engage\_aggr\_rule\_ref\_tgt\_p1\_optimized

Runs datastage job : email\_engage\_aggr\_measure\_ins\_stg\_p1\_optimized

Runs datastage job : email\_engage\_aggr\_rule\_ref\_tgt\_p1\_optimized

Runs datastage job : email\_engage\_aggr\_default\_ins\_tgt\_p1\_optimized

Runs datastage job : email\_engage\_aggr\_compare\_prev\_app\_tgt\_p1\_optimized

Runs datastage job : email\_engage\_aggr\_stats\_append\_tgt\_p1

1. Runs bteq :

/infosrvr/app/edw/source/email\_engage\_aggr/bteq/email\_engage\_aggr\_crt\_prev\_mn.bteq

This script is executed at the beginning of the monthly process to checks the max(d\_timestamp) of the previous month table and refresh data with the existing current month for comparison later on.

LOCK ${EDW\_OS\_PROCDB}.email\_engage\_prev\_mn\_proc FOR ACCESS

SELECT \*

FROM

(

SELECT MAX(d\_timestamp) AS d\_prev\_mn\_dt

FROM ${EDW\_OS\_PROCDB}.email\_engage\_prev\_mn\_proc

) eep

WHERE EXTRACT(MONTH FROM eep.d\_prev\_mn\_dt) = EXTRACT(MONTH FROM CAST('${EDW\_ORDER\_DATE}' AS DATE FORMAT 'YYYYMMDD') )

AND EXTRACT(YEAR FROM eep.d\_prev\_mn\_dt) = EXTRACT(YEAR FROM CAST('${EDW\_ORDER\_DATE}' AS DATE FORMAT 'YYYYMMDD') )

;

.IF ERRORCODE <> 0 THEN EXIT ERRORCODE;

.IF ACTIVITYCOUNT = 0 THEN .GOTO OK\_PROCEED ;

.LABEL ALREADY\_RAN

.REMARK "=== The month of MAX d\_timestamp in the previous month table is the same as current\_month.";

.REMARK "=== The monthly process has already run for this month. ===";

.QUIT 8;

.LABEL OK\_PROCEED

.REMARK "=== Delete the previous month table and insert the previous month data for comparison.";

LOCK ${EDW\_OS\_PROCDB}.email\_engage\_cur\_mn\_proc FOR ACCESS

DELETE

FROM ${EDW\_OS\_PROCDB}.email\_engage\_prev\_mn\_proc;

INSERT INTO ${EDW\_OS\_PROCDB}.email\_engage\_prev\_mn\_proc

SELECT \*

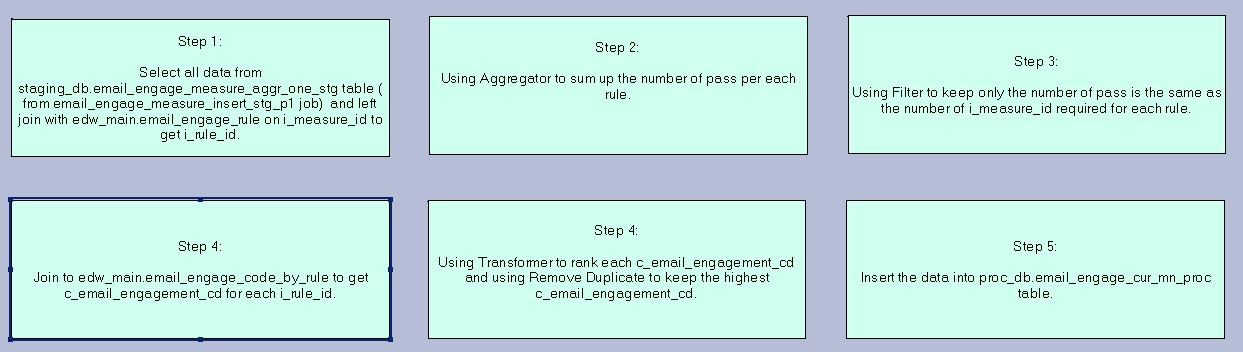
FROM ${EDW\_OS\_PROCDB}.email\_engage\_cur\_mn\_proc ;

1. Runs datastage job : email\_engage\_aggr\_rule\_ref\_tgt\_p1\_optimized

This job queries the aggregated staging table from the previous step left join on edw\_main.email\_engage\_rule, only keep the rule that passed all criteria , join to edw\_main.email\_engage\_code\_by\_rule to get c\_email\_engagement\_cd for the rule, and determine the level of email engagement for the guest (keep the highest).

Runs query :

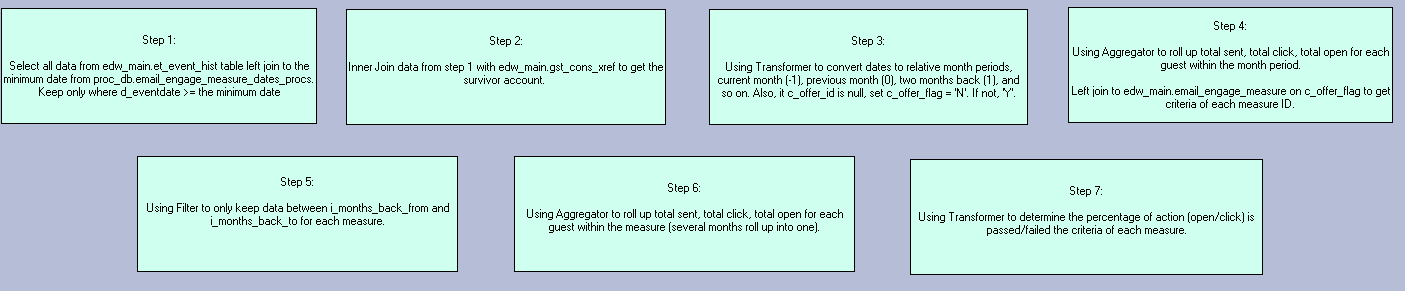




1. Runs datastage job : email\_engage\_aggr\_measure\_ins\_stg\_p1\_optimized

This job queries edw\_main.et\_event\_hist with the minimum report period from the previous step cross reference to gst\_cons\_xref , aggregate data based on edw\_main.email\_engage\_measure and determine passed or failed for each measurement.

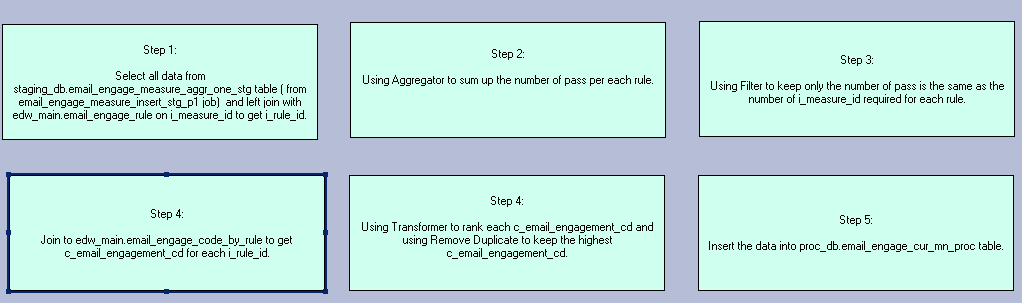
Run Query : 



1. Runs datastage job : email\_engage\_aggr\_rule\_ref\_tgt\_p1\_optimized

This job queries the aggregated staging table from the previous step left join on edw\_main.email\_engage\_rule, only keep the rule that passed all criteria , join to edw\_main.email\_engage\_code\_by\_rule to get c\_email\_engagement\_cd for the rule, and determine the level of email engagement for the guest (keep the highest).

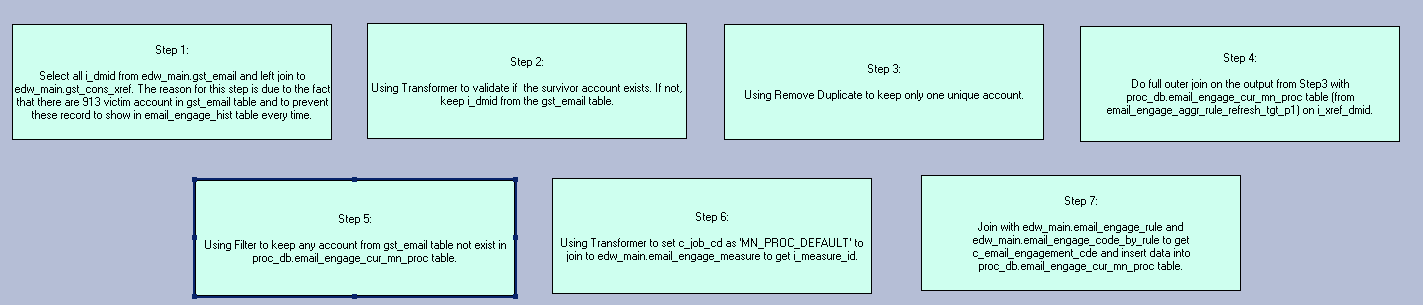
Runs Query :



1. Runs datastage job : email\_engage\_aggr\_default\_ins\_tgt\_p1\_optimized

This job identifies any guest with email but is not in the monthly email engagement aggregation table. Using c\_job\_cd = 'MN\_PROC\_DEFAULT' to get the c\_email\_engagement\_cd. There can be only one measure for this default rule.

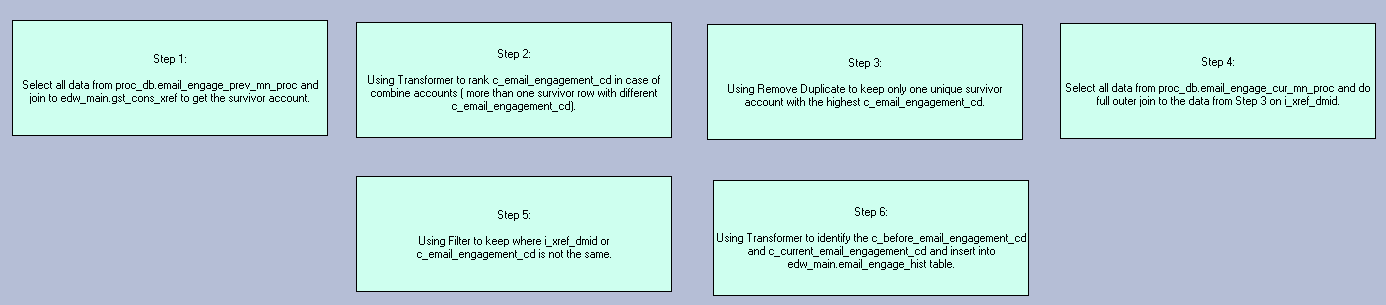
Runs Query :



1. Runs datastage job : email\_engage\_aggr\_compare\_prev\_app\_tgt\_p1\_optimized

This job is to compare c\_email\_engagement\_cd of a guest from the current with previous month. Insert the difference or not found into edw\_main.email\_engage\_hist table.

Runs Query :



1. Runs datastage job : email\_engage\_aggr\_stats\_append\_tgt\_p1

This process will generate the count of each c\_email\_engagement\_cd of the monthly tables and append the data into edw\_main.email\_engagement\_stat with c\_change\_cd as 'M'.

Two tasks perform beblow queries :

INSERT INTO #PS\_DB\_Name.$EDW\_OS\_EDWDB#.email\_engage\_stats

( i\_ins\_run\_id , c\_email\_engagement\_cd, i\_member\_cnt , c\_process\_cd)

VALUES( #PS\_STREAM\_INFO.EDW\_JOB\_RUN\_ID# , ORCHESTRATE.c\_email\_engagement\_cd , ORCHESTRATE.i\_member\_cnt , 'M');

LOCKING #PS\_DB\_Name.$EDW\_OS\_PROCDB#.email\_engage\_cur\_mn\_proc FOR ACCESS

select c\_email\_engagement\_cd, CAST(count(\*) AS INTEGER) i\_member\_cnt from #PS\_DB\_Name.$EDW\_OS\_PROCDB#.email\_engage\_cur\_mn\_proc

group by c\_email\_engagement\_cd;