Inserting records in LANDINGDEV\_MIGRATION.SHARED\_SCHEMA.TEST\_CONTROL\_TABLE

It consists of 2 FQN per entry TABLE\_A AND TABLE\_B OR FILE\_A AND FILE\_B

* APPLICATION\_ID: is name of application being tested
* MASTER\_SEQUENCE: Master sequence name from application if any
* SOURCE\_TYPE: Table or file (if file is mentioned then file from source\_file\_location will be considered)
* SOURCE\_DATABASE: Source database(Landing database in case of file to stage the file)
* SOURCE\_SCHEMA: Source schema(Landing schema from database in case of file to stage the file)
* SOURCE\_TABLE: Source table / temp table name in case of file comparison
* TARGET\_TYPE : Table or file (if file is mentioned then file from target\_file\_location will be considered)
* TARGET\_DATABASE :Target database(Landing database in case of file to stage the file)
* TARGET\_SCHEMA: :Source schema(Landing schema from database in case of file to stage the file)
* TARGET\_TABLE : Source table / temp table name in case of file comparison
* IGNORE\_COLUMN: column to ignore during comparison i.e. timestamp column
* JOIN\_CONDITION: unique column from both side ($1 in case of Fixed with file with single column)
* NUMBER\_OF\_ROWS: Number of rows to be compared(sample size to return if case of mismatch)
* SOURCE\_FILE\_LOCATION: ADLS location of file
* TARGET\_FILE\_LOCATION: ADLS location of file
* COMPARISON\_OPTION
* INPUT\_FORMAT

Sample insert statement

INSERT INTO LANDINGDEV\_MIGRATION.SHARED\_SCHEMA.TEST\_CONTROL\_TABLE (APPLICATION\_ID, MASTER\_SEQUENCE, SOURCE\_TYPE, SOURCE\_DATABASE, SOURCE\_SCHEMA, SOURCE\_TABLE, TARGET\_TYPE, TARGET\_DATABASE, TARGET\_SCHEMA, TARGET\_TABLE, IGNORE\_COLUMN, JOIN\_CONDITION, NUMBER\_OF\_ROWS, SOURCE\_FILE\_LOCATION, TARGET\_FILE\_LOCATION, COMPARISON\_OPTION, INPUT\_FORMAT)

SELECT

'TEST\_AB\_TEST\_FILE' APPLICATION\_ID,

'TEST\_AB\_TEST\_FILE\_MASTER' MASTER\_SEQUENCE,

'FILE' SOURCE\_TYPE,

'EDRDEV' SOURCE\_DATABASE,

'EDRDEV\_MIGRATION' SOURCE\_SCHEMA,

'C' SOURCE\_TABLE,

'FILE' TARGET\_TYPE,

'EDRDEV\_MIGRATION' TARGET\_DATABASE,

'EDR' TARGET\_SCHEMA,

'C\_TEST' TARGET\_TABLE,

‘(0,0),(10,15)' IGNORE\_COLUMN,

'MEAL\_ORDER\_ID,LEGACY\_MEAL\_ORDER\_ID' JOIN\_CONDITION,

'10' NUMBER\_OF\_ROWS,

'steu2shdsnow01/test/MEAL\_ORDER\_FACT.csv' SOURCE\_FILE\_LOCATION,

'steu2shdsnow01/test/MEAL\_ORDER\_FACT.csv' TARGET\_FILE\_LOCATION,

NULL COMPARISON\_OPTION,

NULL INPUT\_FORMAT;

Inserting the file metadata for file comparison

* FILE\_CONTROLID: Unique file entry identifier autoincrement number
* APPLICATION\_ID: Application to which file belongs to
* FILE\_NAME: name of file path from container to file
* FILE\_TYPE: type of file i.e. extension/Delimiter/fixed width e.g. CSV, XLSX, mbi
* FILE\_LOCATION: adls location of file storage account
* DELIMITER: DILIMITER in case of required
* COLUMNS: DDL Json to create table for file load
* COLUMN\_POSITIONS: position or width of columns in case of fixed width files
* ARCHIVE: Archive flag for file
* ARCHIVE\_LOCATION: Archive location for file
* RETENTION\_PERIOD: retention period in days
* TABLE\_NAME: Source table name for file if available
* HEADER\_YES\_NO: Header exists or not
* HEADER\_STRING: Header string if required.

INSERT INTO LANDINGDEV\_MIGRATION.EDR\_EDR.CTL\_FILE\_READ\_WRITE (FILE\_CONTROLID, APPLICATION\_ID, FILE\_NAME, FILE\_TYPE, FILE\_LOCATION, DELIMITER, COLUMNS, COLUMN\_POSITIONS, ARCHIVE, ARCHIVE\_LOCATION, RETENTION\_PERIOD, TABLE\_NAME, HEADER\_YES\_NO, HEADER\_STRING)

SELECT '99' FILE\_CONTROLID,

'TEST1' APPLICATION\_ID,

'DS\_CDH\_Wellness\_202303\_1.csv' FILE\_NAME,

'csv' FILE\_TYPE,

'steu2shdevmat01/test/' FILE\_LOCATION,

',' DELIMITER,

null COLUMNS,

null COLUMN\_POSITIONS,

null ARCHIVE,

null ARCHIVE\_LOCATION,

'90' RETENTION\_PERIOD,

'' TABLE\_NAME,

'NO' HEADER\_YES\_NO,

'' HEADER\_STRING

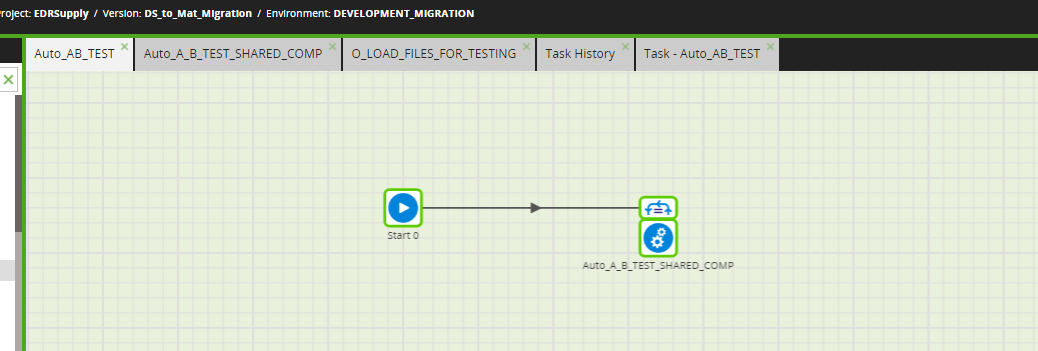
;

<https://10.31.8.7/#BCBSRI_EnterpriseData/EDRSupply/DS_to_Mat_Migration/Auto_AB_TEST>

Change filter to specific Application Id to filter records which need to be run

**JV\_IN\_APLLICATION\_ID\_FLTR: TEST\_999**

**Then run the job**



**One way to check if entries are current between control table and file metadata table**

SELECT K.$1::varchar,L.$1::varchar,M.$1::varchar,

K.SOURCE\_FILE\_LOCATION, L.FILE\_LOCATION||L.FILE\_NAME ,

K.TARGET\_FILE\_LOCATION, M.FILE\_LOCATION||M.FILE\_NAME ,

IFF(coalesce( K.$1::varchar,L.$1::varchar,M.$1::varchar) is null,'ENTRIES not correct','ENTRIES ARE JOINING WITH METADATA') FROM LANDINGDEV\_MIGRATION.SHARED\_SCHEMA.TEST\_CONTROL\_TABLE K

left join LANDINGDEV\_MIGRATION.EDR\_EDR.CTL\_FILE\_READ\_WRITE M on

K.TARGET\_FILE\_LOCATION=M.FILE\_LOCATION||M.FILE\_NAME

left join LANDINGDEV\_MIGRATION.EDR\_EDR.CTL\_FILE\_READ\_WRITE L on

K.SOURCE\_FILE\_LOCATION=L.FILE\_LOCATION||L.FILE\_NAME

where K.APPLICATION\_ID='HCBB-MED\_CLM';

This query will join the two table for given entries

A screenshot of a computer

Description automatically generated

MERGE INTO LANDINGDEV\_MIGRATION.EDR\_EDR.CTL\_FILE\_READ\_WRITE T USING (

SELECT '99' FILE\_CONTROLID,

'TEST1' APPLICATION\_ID,

'DS\_CDH\_Wellness\_202303\_1.csv' FILE\_NAME,

'CSV' FILE\_TYPE,

'steu2shdevmat01/test/' FILE\_LOCATION,

',' DELIMITER,

null COLUMNS,

null COLUMN\_POSITIONS,

null ARCHIVE,

null ARCHIVE\_LOCATION,

'90' RETENTION\_PERIOD,

'' TABLE\_NAME,

'NO' HEADER\_YES\_NO,

'' HEADER\_STRING) S

ON T.FILE\_NAME=S.FILE\_NAME AND T.FILE\_LOCATION=S.FILE\_LOCATION

WHEN MATCHED THEN

UPDATE SET

T.FILE\_CONTROLID=S.FILE\_CONTROLID

, T.APPLICATION\_ID=S.APPLICATION\_ID

, T.FILE\_NAME=S.FILE\_NAME

, T.FILE\_TYPE=S.FILE\_TYPE

, T.FILE\_LOCATION=S.FILE\_LOCATION

, T.DELIMITER=S.DELIMITER

, T.COLUMNS=S.COLUMNS

, T.COLUMN\_POSITIONS=S.COLUMN\_POSITIONS

, T.ARCHIVE=S.ARCHIVE

, T.ARCHIVE\_LOCATION=S.ARCHIVE\_LOCATION

, T.RETENTION\_PERIOD=S.RETENTION\_PERIOD

, T.TABLE\_NAME=S.TABLE\_NAME

, T.HEADER\_YES\_NO=S.HEADER\_YES\_NO

, T.HEADER\_STRING=S.HEADER\_STRING

WHEN NOT MATCHED THEN

INSERT

(FILE\_CONTROLID, APPLICATION\_ID, FILE\_NAME, FILE\_TYPE, FILE\_LOCATION, DELIMITER, COLUMNS, COLUMN\_POSITIONS, ARCHIVE, ARCHIVE\_LOCATION, RETENTION\_PERIOD, TABLE\_NAME, HEADER\_YES\_NO, HEADER\_STRING) values ( S.FILE\_CONTROLID

,S.APPLICATION\_ID

,S.FILE\_NAME

,S.FILE\_TYPE

,S.FILE\_LOCATION

,S.DELIMITER

,S.COLUMNS

,S.COLUMN\_POSITIONS

,S.ARCHIVE

,S.ARCHIVE\_LOCATION

,S.RETENTION\_PERIOD

,S.TABLE\_NAME

,S.HEADER\_YES\_NO

,S.HEADER\_STRING)

;

Below job can be used to make entries in TEST\_CONTROL\_CONFIG

This has been created to replace insert statements,(In future running same will validate entries being made with file metadata table)

<https://10.31.8.7/#BCBSRI_EnterpriseData/EDRSupply/DS_to_Mat_Migration/O_CREATE_CONTROL_ENTRY>

---csv query

INSERT INTO LANDINGDEV\_MIGRATION.EDR\_EDR.CTL\_FILE\_READ\_WRITE (FILE\_CONTROLID, APPLICATION\_ID, FILE\_NAME, FILE\_TYPE, FILE\_LOCATION, DELIMITER, COLUMNS, COLUMN\_POSITIONS, ARCHIVE, ARCHIVE\_LOCATION, RETENTION\_PERIOD, TABLE\_NAME, HEADER\_YES\_NO, HEADER\_STRING, COLUMN\_WIDTH, TABLE\_COLUMNS)

SELECT

'80' FILE\_CONTROLID,

'TEST1' APPLICATION\_ID,

'BCBSRI\_CDH\_Rx\_File\_202020.csv' FILE\_NAME,

'csv' FILE\_TYPE,

'steu2shdsnow01/datahub/demand/publish/CDH-London-health/pharmacy/' FILE\_LOCATION,

',' DELIMITER,

OBJECT\_AGG(COLUMN\_NAME,OBJECT\_CONSTRUCT\_KEEP\_NULL(\*)) COLUMNS,listagg( POSITION,',' ) within GROUP (order by POSITION ) COLUMN\_POSITIONS,

'NO' ARCHIVE,

'steu2shdsnow01/edr/omnichannel\_inbound/archive/' ARCHIVE\_LOCATION,

'90' RETENTION\_PERIOD,

'MEAL\_ORDER\_FACT' TABLE\_NAME,

'NO' HEADER\_YES\_NO,

'' HEADER\_STRING

from (

SELECT COLUMN\_NAME

,POSITION

,DATA\_TYPE

,IFF(PRECISION<36,PRECISION::number(38,0)+2,PRECISION) AS PRECISION

,SCALE

FROM (

SELECT T.TABLE\_CATALOG DB,t.TABLE\_SCHEMA SCHEMA,t.TABLE\_NAME ,c.COLUMN\_NAME,c.ORDINAL\_POSITION POSITION,c.DATA\_TYPE,coalesce(c.DATETIME\_PRECISION

,NUMERIC\_PRECISION,c.CHARACTER\_MAXIMUM\_LENGTH

) PRECISION,coalesce(NUMERIC\_SCALE,'0') SCALE

FROM LANDINGDEV\_MIGRATION.information\_schema.tables t inner join LANDINGDEV\_MIGRATION.information\_schema.columns c on c.table\_schema = t.table\_schema and c.table\_name = t.table\_name where t.table\_type = 'BASE TABLE' and t.table\_schema='SHARED\_SCHEMA' and t.TABLE\_NAME = 'BCBSRI\_CDH\_RX\_FILE'

)

)

;

Result queries

SELECT top 4 \* FROM LANDINGDEV\_MIGRATION.SHARED\_SCHEMA.AB\_TEST\_SUMMARY order by 1 desc;

SELECT \* FROM LANDINGDEV\_MIGRATION.SHARED\_SCHEMA.AB\_TEST\_ROW\_REJECT\_DETAIL order by 1 desc;

INSERT INTO LANDINGDEV\_MIGRATION.EDR\_EDR.CTL\_FILE\_READ\_WRITE (FILE\_CONTROLID, APPLICATION\_ID, FILE\_NAME, FILE\_TYPE, FILE\_LOCATION, DELIMITER, COLUMNS, COLUMN\_POSITIONS, ARCHIVE, ARCHIVE\_LOCATION, RETENTION\_PERIOD, TABLE\_NAME, HEADER\_YES\_NO, HEADER\_STRING)

SELECT NVL((SELECT MAX(FILE\_CONTROLID) FROM LANDINGDEV\_MIGRATION.EDR\_EDR.CTL\_FILE\_READ\_WRITE),'-1')+1 FILE\_CONTROLID,

'TEST1' APPLICATION\_ID,

'DS\_LAB\_M01\_LANDMARK\_O\_MTHLY\_LABMEM\_DATA\_20240304\_041420\_01.TXT' FILE\_NAME,

'FixedWidth' FILE\_TYPE,

'steu2shdevmat01/test/HCBBLabData/' FILE\_LOCATION,

'~' DELIMITER,---here changed dilimiter

null COLUMNS,

null COLUMN\_POSITIONS,

null ARCHIVE,

null ARCHIVE\_LOCATION,

'' RETENTION\_PERIOD,

'' TABLE\_NAME,

'NO' HEADER\_YES\_NO,

'' HEADER\_STRING

UNION ALL

SELECT NVL((SELECT MAX(FILE\_CONTROLID) FROM LANDINGDEV\_MIGRATION.EDR\_EDR.CTL\_FILE\_READ\_WRITE),'-1')+2 FILE\_CONTROLID,

'TEST1' APPLICATION\_ID,

'LAB\_D01\_LANDMARK\_O\_MTHLY\_LABMEM\_DATA\_20240304\_041420\_01.txt' FILE\_NAME,

'FixedWidth' FILE\_TYPE,

'steu2shdevmat01/datahub/demand/publish/HCBB/LAB\_DATA/' FILE\_LOCATION,

'~' DELIMITER,---here changed dilimiter

null COLUMNS,

null COLUMN\_POSITIONS,

null ARCHIVE,

null ARCHIVE\_LOCATION,

'90' RETENTION\_PERIOD,

'' TABLE\_NAME,

'YES' HEADER\_YES\_NO,

'' HEADER\_STRING;

----validation query for entries

SELECT K.$1::varchar,L.$1::varchar,M.$1::varchar,

K.SOURCE\_FILE\_LOCATION, L.FILE\_LOCATION||L.FILE\_NAME ,

K.TARGET\_FILE\_LOCATION, M.FILE\_LOCATION||M.FILE\_NAME ,

IFF(coalesce( K.$1::varchar,L.$1::varchar,M.$1::varchar) is null,'ENTRIES not correct','ENTRIES ARE JOINING WITH METADATA') FROM LANDINGDEV\_MIGRATION.SHARED\_SCHEMA.TEST\_CONTROL\_TABLE K

left join LANDINGDEV\_MIGRATION.EDR\_EDR.CTL\_FILE\_READ\_WRITE M on

K.TARGET\_FILE\_LOCATION=M.FILE\_LOCATION||M.FILE\_NAME

left join LANDINGDEV\_MIGRATION.EDR\_EDR.CTL\_FILE\_READ\_WRITE L on

K.SOURCE\_FILE\_LOCATION=L.FILE\_LOCATION||L.FILE\_NAME

where K.APPLICATION\_ID='CDH\_WELLNESS';