[Snowflake] - Number of columns in file does not match that of the corresponding table

Last updated by | Jackie Huang | Jan 4, 2022 at 12:24 AM PST

Issue

When customer try to move/transfer data from different source into snowflake sink, it may meet issue like "Number of columns in file does not match that of the corresponding table" if input data's schema is like:

```
1. The schema of input data (The output of previous dataflow source or transformation) has more columns than s
2. The schema of input data (The output of previous dataflow source or transformation) has less columns than s
3. The column number of input data (The output of previous dataflow source or transformation) is the same like
```

Root Cause

Today we have one bug in dataflow snowflake sink, when customer try to directly write/insert source data into snowflake sink but columns are mismatched, the customer will meet up issue. Now we are working on fix, before the fix takes effective, please ask customer to use following way to work around.

Resolution

• If the column number of input data is different compared with snowflake sink table, the customer can use column mapping on sink + alter row transformation to finish move/transfer data, for example:

```
source(output(
        movieId as string,
        title as string,
        releaseDate as string,
        rated as string,
        screenedOn as string,
        ticketPrice as string,
        id as string
    ),
    allowSchemaDrift: true,
    validateSchema: false,
    ignoreNoFilesFound: false) ~> csvSource
csvSource select(mapColumn(
        RATED = rated,
        MOVIEID = movieId
    ),
    skipDuplicateMapInputs: true,
    skipDuplicateMapOutputs: true) ~> Select1
Select1 alterRow(upsertIf(true())) ~> AlterRow1
AlterRow1 sink(input(
        MOVIEID as string,
        TITLE as string,
        GENRES as string
    ),
    allowSchemaDrift: true,
    validateSchema: false,
    deletable:false,
    insertable:false,
    updateable:false,
    upsertable:true,
    keys:['MOVIEID'],
    format: 'table',
    mapColumn(
        TITLE = MOVIEID,
        MOVIEID = RATED
    ),
    skipDuplicateMapInputs: true,
    skipDuplicateMapOutputs: true) ~> sink1
```

It has 7 columns in source data

Optional: use select transformation to read partial data from source

Use alterRow with "upsert" to insert source data into snowflake sink.

Use column mapping on sink to write source data into expected columns. If no column mapping, it will map sink column by name.

Here it can map partial/full sink schema, for the column which is not mapped, it will be writed with null value

• If the column name or data type of input data is different compared with snowflake sink table, the customer can use "select" transformation to change column name + "derive" transformation to change data type before writing data into snowflake sink table.

```
source(output(
        movieId as string,
        title as string,
        releaseDate as string,
        rated as string,
        screenedOn as string,
        ticketPrice as string
    ),
    allowSchemaDrift: true,
    validateSchema: false,
    ignoreNoFilesFound: false) ~> ExcelSource
ExcelSource select(mapColumn(
        newMovieId = movieId,
        newTitle = title
    ),
    skipDuplicateMapInputs: true,
    skipDuplicateMapOutputs: true) ~> changeColumnName
changeColumnName derive(newMovieId = toDate(newMovieId)) ~> changeColumnType
```

Additional Information:

- Icm Reference: N/A
- Author: Zhangyi Yu
- Reviewer: Zhangyi Yu; Shawn Xiao