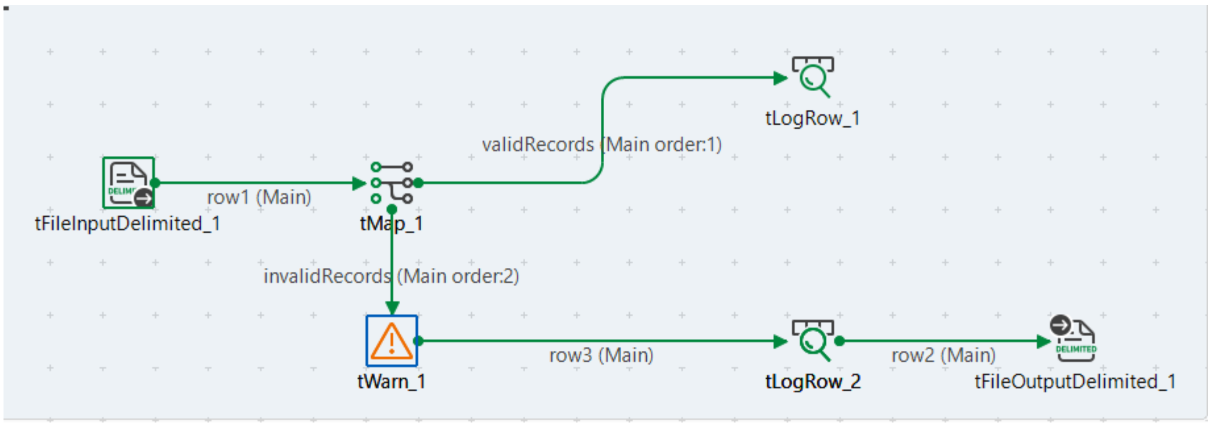
Exercise on Data Quality Validation:

Objective:

- **We want to validate and transform records from customer_data.csv, checking for valid phone numbers, emails, and full names. We will route valid and invalid records using tMap and use of tWarn component.**

Workflow:



Step 1: Prepare Input Data (tFileInputDelimited)

1. **Drag and Drop tFileInputDelimited:**
   - **From the Palette, drag tFileInputDelimited to the design workspace.**
   - **In the Basic Settings, set the File Name to customer_data.csv (the file you prepared earlier).**
2. **Set the Schema:**
   - **Define the schema with the following columns:**
     - **customer_id (Integer)**
     - **full_name (String)**
     - **email (String)**
     - **phone (String)**

Step 2: Add tMap Component

1. **Drag and Drop tMap:**

- From the Palette, drag tMap to the workspace.

2. **Link tFileInputDelimited to tMap:**
   - Connect tFileInputDelimited to tMap by dragging a link from tFileInputDelimited to tMap.

Step 3: Configure the Input and Output in tMap

1. **Input Columns:**
   - In tMap, you will see row1 (the input row) on the left. The columns from the CSV file (e.g., customer_id, full_name, email, phone) will be available under row1.
2. **Output Columns:**
   - Create two output flows in tMap:
     - validRecords (for valid records)
     - invalidRecords (for invalid records)

Step 4: Define the Mapping Logic

1. **Map Full Name with Transformation:**
   - For the validRecords.full_name column:
     - In the Expression column of validRecords.full_name, write the following transformation logic to convert the first letter to uppercase and the rest to lowercase:

```
StringHandling.UPCASE(StringHandling.LEFT(StringHandling.TRIM(row1.full_
name), 1)) +
row1.full_name.substring(1).toLowerCase()
```

This will ensure that the full_name is properly capitalized.

2. **Define the Condition for Valid Records:**
   - In the Expression Filter for validRecords, add the following condition to validate the phone, email, and full_name:
   - row1.phone.matches("\\d{10}") && row1.email.contains("@") && row1.email.endsWith(".com") && row1.full_name.split(" ").length >= 2
   - This condition ensures:

        - The phone is 10 digits long.
        - The email contains "@" and ends with ".com".
        - The full_name has at least two words (split by space).

3. Invalid Records Routing:
    - For the invalidRecords route, no expression is required in the Expression Filter because tMap will automatically route records that fail the condition to the invalidRecords output.
    - However, if you'd like to manually flag the reason for invalid records, you can add a custom field to invalidRecords to store the error message (e.g., error_message column).

Step 5: Set Output for Valid and Invalid Records

1. Valid Records:
    - The validRecords output will contain rows where all conditions are met. You can connect this to a tFileOutputDelimited to save valid records to a file.
2. Invalid Records:
    - The invalidRecords output will contain rows where any condition is violated (phone, email, or full name). You can connect this to another tFileOutputDelimited to store invalid records separately.

---

4. tLogRow (Connect to both valid and invalid)

Purpose: Print the results in the console

- Component: tLogRow
- Settings:
    - Connect both validRecords to tLogRow_1 and invalidRecords to tLogRow_2(Note: Before tLogRow_2 you need to add tWarn_1(steps given below)
    - Mode: Table

---

5. tFileOutputDelimited

Purpose: Save invalid records to a CSV file

- **Component: tFileOutputDelimited**
- **Settings:**
  - **File Path: e.g., C:/TalendOutput/invalid_customers.csv**
  - **Field Separator: ,**
  - **Include Header: Yes**
  - **Connect from invalidRecords output of tMap**

6. Set Up tWarn to Trigger Alerts

You can use tWarn to generate a warning when invalid records are found. This will not stop the job but can be used to notify you about the issue.

1. **Add tWarn Component:**
   - **Drag tWarn from the Palette to the workspace.**
   - **Connect tMap (after processing invalid records) to tWarn.**
2. **Configure tWarn:**
   - **In the Basic Settings of tWarn, set the Warning Message that should be triggered when invalid records are found. For example:**

Warn Message:

"Warning: Invalid records detected!"

Code:42

Priority:Warning

FYI..

Validrecords.full_name expression:
StringHandling.UPCASE(StringHandling.LEFT(StringHandling.TRIM(row1.full_name), 1)) +

row1.full_name.substring(1).toLowerCase()

Explanation of the Updated Expression:

- **StringHandling.UPCASE(StringHandling.LEFT(StringHandling.TRIM(row1.full_name), 1))**:
    - This part of the expression capitalizes the first character of the full_name.
    - It first trims any extra spaces and takes the first character using LEFT, then converts it to uppercase using UPCASE.
- **row1.full_name.substring(1).toLowerCase()**:
    - This part extracts the rest of the full_name (excluding the first character) and converts it to lowercase using toLowerCase(), which is a standard Java string metho

Valid expression:

```
row1.phone.matches("\\d{10}") && row1.email.contains("@") &&
row1.email.endsWith(".com") && row1.full_name.split(" ").length >= 2
```

Invalid expression:

```
!(row1.phone.matches("\\d{10}") &&
 row1.email.contains("@") &&
 row1.email.endsWith(".com") &&
 row1.full_name.split(" ").length >= 2)
```