Customer Data Cleanup Pipeline Full Student Guide

Objective

This hands-on exercise will guide students through building a data cleaning and integration pipeline using Talend Studio.
Students will load customer records from a CSV and orders from MySQL, join them, validate records, and output clean/invalid rows separately.

1. Create MySQL Database and Table

Open MySQL Workbench or CLI and run the following commands:

```
CREATE DATABASE talend_demo;
USE talend_demo;

CREATE TABLE orders (
   order_id INT,
   cust_id INT,
   order_date DATE,
   amount DECIMAL(10,2)
);

INSERT INTO orders VALUES
(1, 101, '2024-01-01', 120.50),
(2, 102, '2024-02-10', 220.00),
(3, 103, '2024-03-20', 90.75);
```

2. Create customers.csv File

Use Notepad or Excel to save the following content as customers.csv:

```
cust_id,first_name,last_name,email,phone
101,John,Doe,john.doe@example.com,9876543210
102,Jane,,jane.doe@sample,98765ABCD
103,Bob,Smith,bob.smith@gmail.com,8765432109
```

3. Build the Talend Job

Your job should include these components:

```
tFileInputDelimited  tMap  valid_customers  tDBOutput
                          invalid_customers  tFileOutputDelimited
```

tDBInput (orders table) Lookup in tMap

Configure each component as follows:

tFileInputDelimited:
- File: customers.csv
- Field Separator: ,
- Header: 1
- Schema: cust_id (Integer), first_name (String), last_name (String), email (String), phone (String)

tDBInput:
- Property Type: Built-In
- Host: localhost
- DB: talend_demo
- Query: SELECT order_id, cust_id, order_date, amount FROM orders
- Schema: order_id, cust_id, order_date, amount

tMap:
- Connect CSV as Row Main
- Connect DBInput as Row Lookup
- Join on: row1.cust_id = row2.cust_id

Create two outputs:
 valid_customers
 invalid_customers

Expression Filter for valid_customers:
row1.email.matches("^[A-Za-z0-9+_.-]+@[A-Za-z0-9.-]+$") &&
row1.phone.matches("\\d{10}")

Expression Filter for invalid_customers:
!row1.email.matches("^[A-Za-z0-9+_.-]+@[A-Za-z0-9.-]+$") ||
!row1.phone.matches("\\d{10}")

Fields mapped to both outputs:
cust_id, first_name, last_name, email, phone, total_amount (row2.amount)

tDBOutput:
- Target Table: valid_customers
- Action on Table: Create if not exists
- Sync schema with valid_customers

tFileOutputDelimited:
- File path: invalid_customers.csv
- Field separator: ,
- Include Header: Checked
- Sync schema with invalid_customers

4. Add tLogRow (Optional)

You can also connect valid_customers and/or invalid_customers to tLogRow for testing.
Set Output Mode: Table.

5. Run and Verify Results

- Run the job
- Verify valid_customers are inserted into the MySQL table
- Check invalid_customers.csv for failed rows

How to create Data Base connection and save it ::

In talend studio→ MetaData→DB connections→ Create the DB Connection Here::

Field	Correct Value
DB Version	MySQL 8
String of Connection	jdbc:mysql://localhost:3306/talend_demo
Login	root (or your MySQL user)
Password	your password
Server	localhost
Port	3306
Database	talend_demo
Additional Parameters	noDatetimeStringSync=true&enabledTLSProtocols=TLSv1.2,TLSv1.1,TLSv1