**Exp Name:** Cleaning Excel File Using Apache POI

Gouri Ramachandran

22MIA1074

**Aim**

To develop a Java application using the Apache POI library that cleans an Excel file by removing invalid data and duplicate entries, improving data quality and integrity.

**Algorithm**

1. File Initialization
   * Define input and output file paths
   * Create FileInputStream for reading the source Excel file
   * Initialize Apache POI Workbook objects for input and output
2. Data Processing
   * Iterate through each row and cell in the source worksheet
   * Apply cleaning operations:
     + Remove leading and trailing whitespace from text
     + Validate cell contents
     + Handle different data types (string, numeric, boolean)
   * Skip rows containing empty or invalid data
   * Track and remove duplicate entries
3. Output Generation
   * Write cleaned data to a new workbook
   * Save the processed data to the output file
   * Clean up resources and close file streams

**Program Implementation**

package com.gouri;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.util.ArrayList;

import java.util.HashSet;

import java.util.List;

import java.util.Set;

import org.apache.poi.ss.usermodel.Cell;

import org.apache.poi.ss.usermodel.CellType;

import org.apache.poi.ss.usermodel.Row;

import org.apache.poi.ss.usermodel.Sheet;

import org.apache.poi.ss.usermodel.Workbook;

import org.apache.poi.xssf.usermodel.XSSFWorkbook;

public class CleanExcelData {

public static void main(String[] args) {

// Specify input and output file paths

String inputFilePath = "/Users/gourir/Documents/bigdataframeworks/Table 2.xlsx"; // Input file

String outputFilePath = "/Users/ gourir/Documents/bigdataframeworks/Cleaned\_Table 2.xlsx"; // Output file

try (FileInputStream fis = new FileInputStream(inputFilePath);

Workbook workbook = new XSSFWorkbook(fis);

Workbook cleanedWorkbook = new XSSFWorkbook()) {

// Read the first sheet from the workbook

Sheet sheet = workbook.getSheetAt(0);

Sheet cleanedSheet = cleanedWorkbook.createSheet(sheet.getSheetName());

int cleanedRowIndex = 0;

for (Row row : sheet) {

Row cleanedRow = cleanedSheet.createRow(cleanedRowIndex);

boolean isValidRow = true;

List<Object> cleanedData = new ArrayList<>();

for (Cell cell : row) {

Cell cleanedCell = cleanedRow.createCell(cell.getColumnIndex());

switch (cell.getCellType()) {

case STRING:

String value = cell.getStringCellValue().trim(); // Remove whitespace

if (value.isEmpty()) {

isValidRow = false;

}

cleanedCell.setCellValue(value);

cleanedData.add(value);

break;

case NUMERIC:

cleanedCell.setCellValue(cell.getNumericCellValue());

cleanedData.add(cell.getNumericCellValue());

break;

case BOOLEAN:

cleanedCell.setCellValue(cell.getBooleanCellValue());

cleanedData.add(cell.getBooleanCellValue());

break;

default:

cleanedCell.setCellValue("N/A"); // Handle invalid data

isValidRow = false;

cleanedData.add("N/A");

}

}

// Check if the row contains missing data

if (cleanedData.contains(null) || cleanedData.contains("")) {

isValidRow = false;

}

// Increment only for valid rows

if (isValidRow) {

cleanedRowIndex++;

} else {

cleanedSheet.removeRow(cleanedRow); // Remove invalid row

}

}

// Remove duplicate rows

removeDuplicates(cleanedSheet);

// Write cleaned data to the new workbook

try (FileOutputStream fos = new FileOutputStream(outputFilePath)) {

cleanedWorkbook.write(fos);

}

System.out.println("Data cleaning completed. Cleaned data saved to " + outputFilePath);

} catch (Exception e) {

e.printStackTrace();

}

}

private static void removeDuplicates(Sheet sheet) {

Set<String> uniqueRows = new HashSet<>();

List<Integer> rowsToDelete = new ArrayList<>();

for (Row row : sheet) {

StringBuilder rowData = new StringBuilder();

for (Cell cell : row) {

if (cell.getCellType() == CellType.STRING) {

rowData.append(cell.getStringCellValue()).append("|");

} else if (cell.getCellType() == CellType.NUMERIC) {

rowData.append(cell.getNumericCellValue()).append("|");

}

}

if (!uniqueRows.add(rowData.toString())) {

rowsToDelete.add(row.getRowNum());

}

}

// Remove rows in reverse order to avoid shifting issues

for (int i = rowsToDelete.size() - 1; i >= 0; i--) {

int rowIndex = rowsToDelete.get(i);

sheet.removeRow(sheet.getRow(rowIndex));

sheet.shiftRows(rowIndex + 1, sheet.getLastRowNum(), -1);

}

}

}

**Output**

Table 1

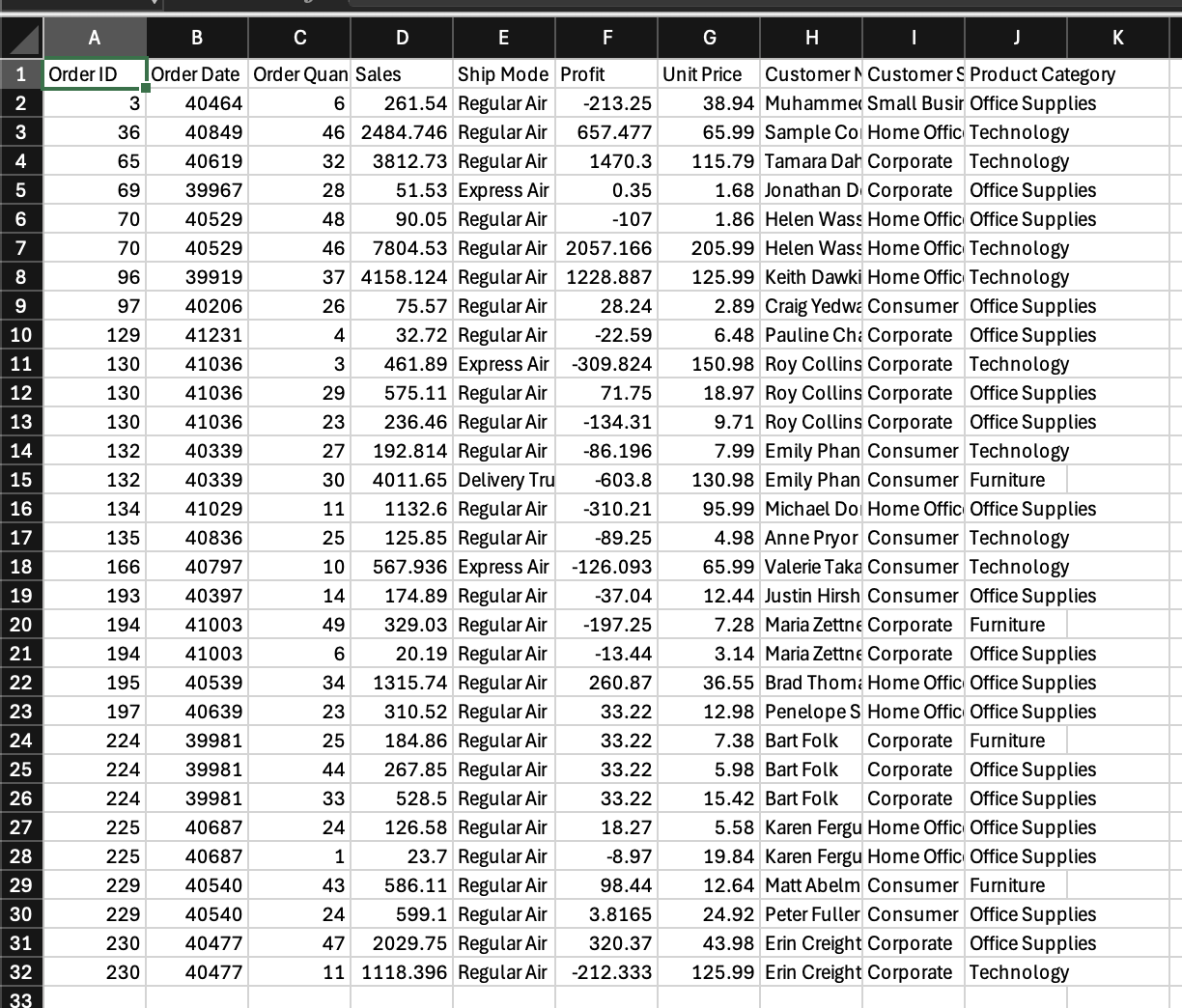
****

Table 2

A screenshot of a table

Description automatically generated

**Result**

The experiment successfully demonstrates the ability to clean an Excel file by removing invalid entries and duplicates using Apache POI in Java.