**Java Back-end Migration Developer Code Kata**

**Instructions**

This kata is designed for developers who are applying for a back-end migration developer position. The responsibility of this role will be to migrate data from one system to another, using methods such as APIs for reuse with different systems. By completing this exercise, the candidate is demonstrating an ability to migrate data held in the associated workbook (Import data.xlsx) to the database schema outlined below in **Step 1**. The data should be validated to ensure the integrity of the data matches the requirements outlined below in **Step 2**.

**Step 1. Creating the database table schemas:**

Here is the schema showing the 2 tables that you will need to import the data into:

-- Create the Customer table

CREATE TABLE Customer (

CustomerID INT PRIMARY KEY AUTO\_INCREMENT,

CustomerName VARCHAR(50) NOT NULL,

AddressLine1 VARCHAR(60) NOT NULL,

TownCity VARCHAR(60) NOT NULL,

County VARCHAR(60) NOT NULL,

PostCode VARCHAR(10) NOT NULL

);

-- Create the Supply table

CREATE TABLE Supply (

SupplyID INT PRIMARY KEY AUTO\_INCREMENT,

CustomerID INT NOT NULL,

MPAN VARCHAR(13) NOT NULL,

MSN VARCHAR(10) NOT NULL,

SSC VARCHAR(4) NOT NULL,

ProfileClass INT NOT NULL,

FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)

);

**Step 2. The requirements**

We need to import the data on the spreadsheet into the tables provided. We can’t always guarantee the data provided is in a good format for example, there could be spaces within the data at the start or the end that need to be stripped or an incorrect character in a field such as an alpha in a numeric field.

We need to validate that the data is in fact correct. Here is the list of validation required:

* CustomerName must be provided.
* AddressLine1 must be provided.
* TownCity must be provided.
* County must be provided.
* PostCode must be provided.
* MPAN must be provided and it is a 13-digit number which is then stored as a VARCHAR(13). We must ensure this is a number though upon import.
* MSN must be provided. This can be a number or a string and doesn’t require the same validation as MPAN.
* SSC is a 4-digit code and must be provided. The 0 at the start must remain on.
* Profile class is a single digit and must be provided.

Spaces must be stripped from the import.

If the data is in fact incorrect, then this needs to be outputted onto a new file. This should be one file containing all errors.

We should have 5 successful imports and 5 failures.

**Tip. Consider using tools such as OpenAPI to validate any APIs.**