

Sprawozdanie z wykonania projektu na przedmiot „Wykorzystanie MS SQL i pakietów MS Business Intelligence do budowy aplikacji”

Sebastian Pietrykowski, 319085

Temat: JPK_WB

1. Cel

Celem mojej pracy jest stworzenie aplikacji generującej wyciągi bankowe w formacie JPK_WB.

Zgodnie z prawem obowiązującym w Polsce wszyscy podatnicy prowadzący księgi podatkowe i wytwarzający dowody księgowe w formie elektronicznej są zobowiązani okazać struktury Jednolitego Pliku Kontrolnego (JPK) na żądanie organów podatkowych. Wyciąg bankowy jest dowodem księgowym stanowiącym podstawę zapisów w księgach rachunkowych. Raportowaniu podlegają wszelkie rachunki bankowe prowadzone przez podatnika. Obowiązek dostarczenia struktury JPK spoczywa na podatniku, może ona zostać wygenerowana przez bank w formie dokumentu księgowego przez system bankowości elektronicznej.

Zbiory danych w plikach tekstowych:

- Podmioty
- Nagłówki
- Pozycje
- Wiersze

Wygenerowany raport będzie miał format XML.

Aplikacja zostanie stworzona wykorzystując Microsoft Business Intelligence, w tym Microsoft SQL Server Management Studio oraz Microsoft Visual Studio z rozszerzeniem SQL Server Integration Services Projects.

2. Analiza

2.1. Wymagania funkcjonalne

1. System ma działać dla różnych firm działających na terenie Polski, które potrzebują generować plik JPK_WB
2. Raport jest generowany dla jednej firmy dla danego zbioru danych
3. System przyjmuje na wejściu 4 pliki tekstowe ze zbiorami danych (podmioty, nagłówki, pozycje, wiersze)
4. Format danych tekstowych zgodny ze schematem bazy danych dla tabel tymczasowych
5. Dane w każdym pliku muszą być identyfikowane przez identyfikator
6. Sprawdzanie poprawności powiązań między zbiorami danych
7. Sprawdzanie poprawności formatu danych

8. Sprawdzenie, czy saldo końcowe odpowiada saldu początkowemu po uwzględnieniu sumy obciążeń i sumy uznań
9. Sprawdzenie, czy pole ilość wierszy odpowiada ilości wierszy w zbiorze danych
10. Sprawdzenie, czy daty nie są w przyszłości i wzajemnie się nie wykluczają
11. Sprawdzenie, czy suma obciążeń i suma uznań odpowiadają sumom z wierszy

3. Opis algorytmu

1. Pliki źródłowe tekstowe w formacie Unicode UTF-8
2. Pakiety przesyłające dane z pliku tekstowego do bazy danych w formacie .dtsx
3. Tabele tymczasowe mają strukturę identyczną z plikami źródłowymi, wszystkie pola są przechowywane w formie tekstowej, tabele tymczasowe nie zachowują relacji między sobą
4. Weryfikacja poprawności następuje w procedurach Microsoft SQL
5. Procedura MS SQL przenosi dane z tabel tymczasowych do tabel docelowych, w których pola mają prawidłowy typ danych oraz zachowane są relacje między tabelami
6. Procedura MS SQL tworzy raport w formacie XML, akceptowany przez Ministerstwo Finansów

Przykładowy plik źródłowy

| id_dokumentu | id_podmiotu | numer_rachunku | saldo_poczatkowe | saldo_koncowe | liczba_wierszy | suma_obciazen | suma_uznan |
|--------------|-------------|----------------------------|------------------|---------------|----------------|---------------|------------|
| 1 | 1 | 44114020205082626899431966 | 100,43 | 240,00 | 3 | 0,00 | 139,57 |

Utworzenie schematu bazy danych

Aby utworzyć odpowiednie tabele i wczytać procedury należy uruchomić skrypt *wb.sql*

Diagram bazy danych – tabele tymczasowe



```
-- utwórz tabelę tmp_wb_na
EXEC dbo.remove_table @tab_name = 'tmp_wb_na'
GO
CREATE TABLE dbo.tmp_wb_na
(
  id_dokumentu      NVARCHAR(15)    NOT NULL
, kod_formularza    NVARCHAR(10)    NOT NULL
, kod_systemowy     NVARCHAR(15)    NOT NULL
, wersja_schemy     NVARCHAR(5)     NOT NULL
, wariant_formularza NVARCHAR(5)     NOT NULL
, cel_zlozenia      NVARCHAR(5)     NOT NULL
, data_wytworzenia_jpk NCHAR(19)    NOT NULL -- RRRR-MM-DDTHH:MM:SS,
, data_od           NCHAR(10)       NOT NULL -- RRRR-MM-DD
, data_do           NCHAR(10)       NOT NULL -- RRRR-MM-DD
, domyslny_kod_waluty NCHAR(3)      NOT NULL -- 3 litery (ISO 4217)
, kod_urzedu        NCHAR(4)       NOT NULL -- 4 cyfry
)
GO
```

```

-- utwórz tabelę tmp_wb_pod
EXEC dbo.remove_table @tab_name = 'tmp_wb_pod'
GO
CREATE TABLE dbo.tmp_wb_pod
(
    id_podmiotu      NVARCHAR(15)    NOT NULL
,   nip              NCHAR(10)      NOT NULL -- 10 cyfr
,   pelna_nazwa      NVARCHAR(255)   NOT NULL
,   regon            NCHAR(9)        NOT NULL -- 9 cyfr
,   kod_kraju        NVARCHAR(50)    NOT NULL
,   wojewodztwo      NVARCHAR(50)    NOT NULL
,   powiat           NVARCHAR(50)    NOT NULL
,   gmina            NVARCHAR(50)    NOT NULL
,   ulica            NVARCHAR(100)   NOT NULL
,   nr_domu          NVARCHAR(10)    NOT NULL
,   nr_lokalu        NVARCHAR(5)     NOT NULL
,   miejscowosc      NVARCHAR(100)   NOT NULL
,   kod_pocztowy     NVARCHAR(10)    NOT NULL
,   poczta           NVARCHAR(100)   NOT NULL
)
GO

-- utwórz tabelę tmp_wb_poz
EXEC dbo.remove_table @tab_name = 'tmp_wb_poz'
GO
CREATE TABLE dbo.tmp_wb_poz
(
    id_dokumentu     NVARCHAR(15)    NOT NULL
,   id_podmiotu      NVARCHAR(15)    NOT NULL
,   numer_rachunku   NCHAR(26)       NOT NULL -- IBAN (26 cyfr)
,   saldo_poczkowe   NVARCHAR(20)    NOT NULL
,   saldo_koncowe    NVARCHAR(20)    NOT NULL
,   liczba_wierszy   NVARCHAR(10)    NOT NULL
,   suma_obciazen    NVARCHAR(20)    NOT NULL
,   suma_uznan       NVARCHAR(20)    NOT NULL
)
GO

-- utwórz tabelę tmp_wb_wiersz
EXEC dbo.remove_table @tab_name = 'tmp_wb_wiersz'
GO
CREATE TABLE dbo.tmp_wb_wiersz
(
    id_dokumentu     NVARCHAR(15)    NOT NULL
,   numer_wiersza    NVARCHAR(15)    NOT NULL
,   data_operacji    NCHAR(10)       NOT NULL -- RRRR-MM-DD
,   nazwa_podmiotu   NVARCHAR(255)   NOT NULL
,   opis_operacji    NVARCHAR(255)   NOT NULL
,   kwota_operacji   NVARCHAR(20)    NOT NULL
,   saldo_operacji   NVARCHAR(20)    NOT NULL
)
GO

```

Tabele *tmp_wb_pod*, *tmp_wb_na*, *tmp_wb_poz*, *tmp_wb_wiersz* nie są powiązane z innymi tabelami. Służą do transferu danych z plików tekstowych do tych tabel.

- Tabela *tmp_wb_pod* przechowuje dane podmiotu, którego dotyczy Jednolity Plik Kontrolny.
- Tabela *tmp_wb_na* przechowuje dane nagłówku w Jednolitym Pliku Kontrolnym.
- Tabela *tmp_wb_poz* przechowuje dane kontrolne dotyczące wierszy.
- Tabela *tmp_wb_wiersz* przechowuje kolejne wiersze wyciągu bankowego.

Tabele *WB_ELOG_N* i *WB_ELOG_D* służą do zapisu błędów występujących w trakcie walidacji danych. *WB_ELOG_N* zapisuje nagłówki błędów, a *WB_ELOG_D* – szczegóły tych błędów.

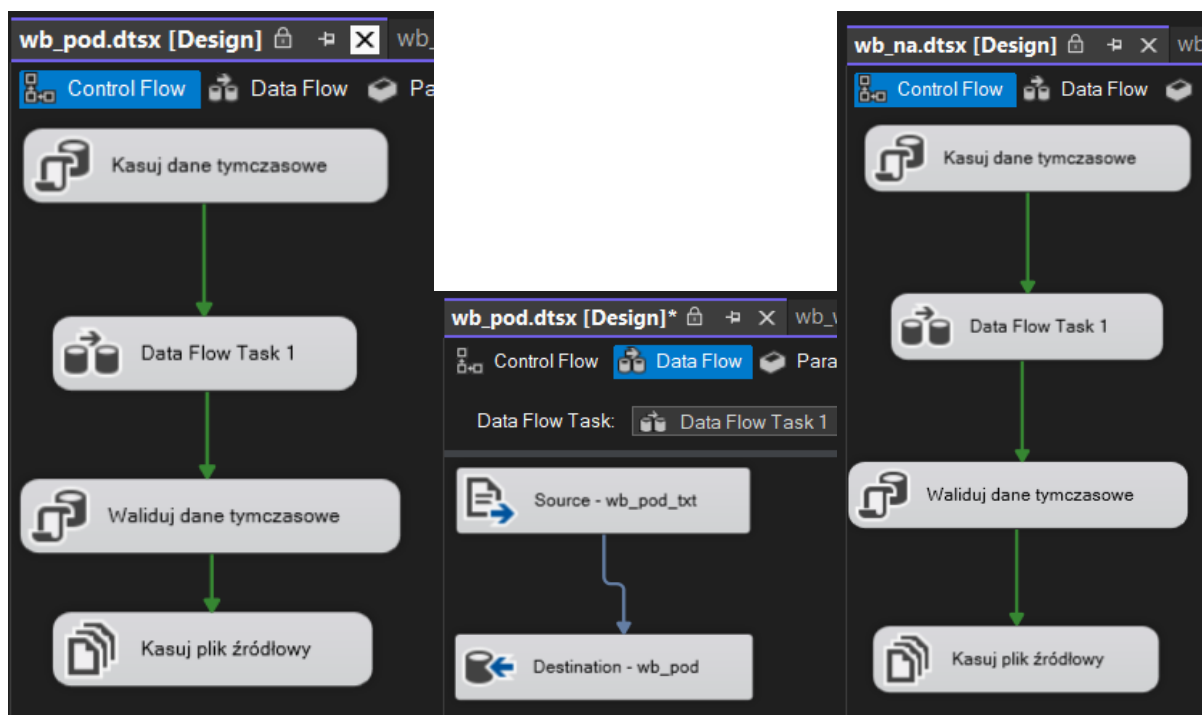
Transfer odbywa się poprzez uruchomienie pliku Data Transformation Services Package XML File Format z rozszerzeniem .dtsx, który importuje dane z pliku tekstowego zapisanego w formacie Unicode, jednocześnie przeprowadzając walidację. Dane do pliku tekstowego pozyskują poprzez eksport danych z Excela.

Wczytywanie pakietów

Pakiety należy wczytać w następującej kolejności:

1. *wb_pod.dtsx*
2. *wb_na.dtsx*
3. *wb_poz.dtsx*
4. *wb_wiersz.dtsx*

Można do tego użyć komendy: *dtexec /F plik.dtsx*

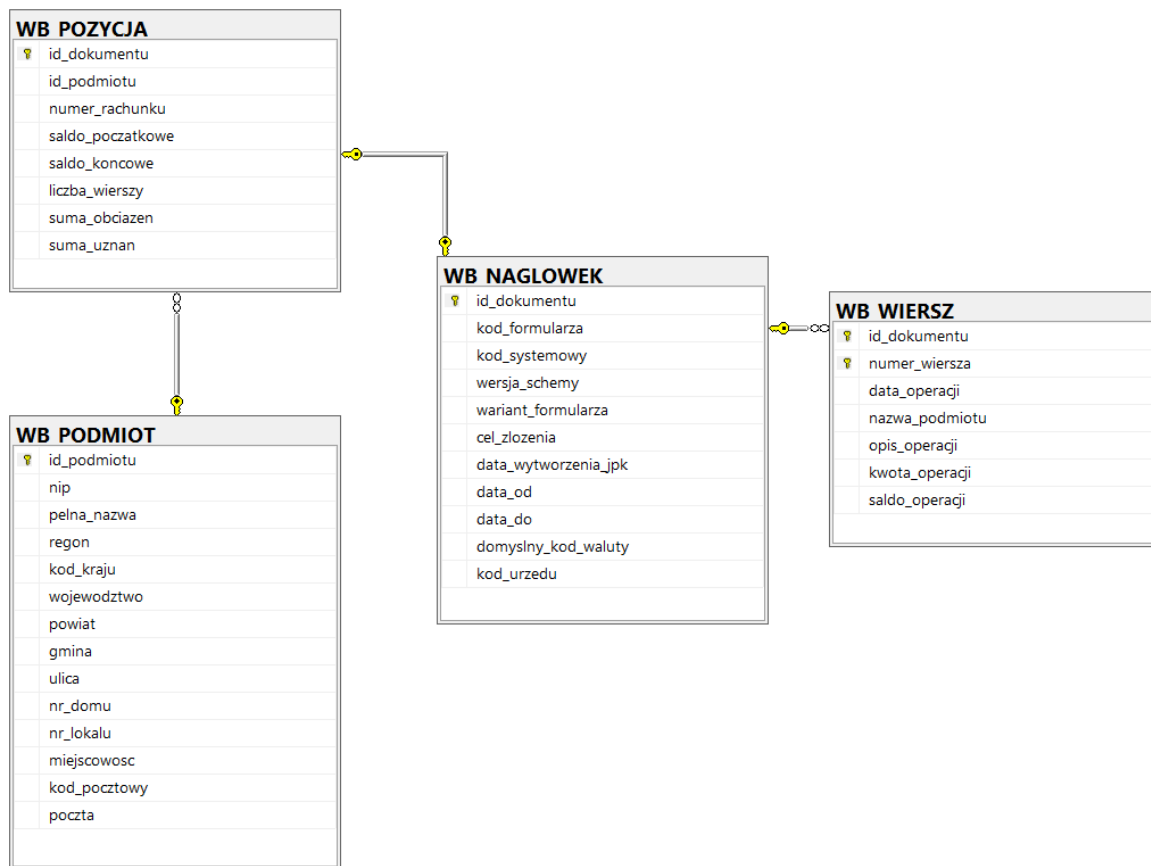




Pliki służące do transferu danych mają podobną strukturę:

1. *Kasuj dane tymczasowe* – usuwa dane z tabeli tymczasowej.
2. *Data Flow Task 1* – przesyła dane z pliku źródłowego do tabeli w bazie danych.
3. *Waliduj dane tymczasowe* – wywołuje procedurę sprawdzającą poprawność wczytywanych danych. W przypadku niepowodzenia zwraca błąd, o którym informacje są zapisywane w tabelach z błędami.
4. (Tylko dla *wb_wiersz.dtsx*) *Przenieś dane do tabeli docelowych* – przenosi dane z tabeli tymczasowych do tabeli docelowych.
5. *Kasuj plik źródłowy* – usuwa plik z danymi źródłowymi.

Diagram bazy danych – tabele docelowe



Dane z tabel tymczasowych są konwertowane do odpowiednich typów i zapisywane w tabelach docelowych odpowiadających tabelom tymczasowym.

```
-- utwórz tabelę PODMIOT do przechowywania podmiotów
IF NOT EXISTS ( SELECT 1 FROM sysobjects o WHERE o.[name] = 'WB_PODMIOT'
                AND (OBJECTPROPERTY(o.[ID], 'IsUserTable') = 1)
              )
BEGIN
    CREATE TABLE dbo.WB_PODMIOT
    (
        id_podmiotu      INT NOT NULL IDENTITY CONSTRAINT PK_WB_PODMIOT PRIMARY KEY
        , nip             NCHAR(10)      NOT NULL -- 10 cyfr
        , pelna_nazwa     NVARCHAR(255)   NOT NULL
        , regon           NCHAR(9)        NOT NULL -- 9 cyfr
        , kod_kraju       NVARCHAR(50)    NOT NULL
        , wojewodztwo     NVARCHAR(50)    NOT NULL
        , powiat          NVARCHAR(50)    NOT NULL
        , gmina           NVARCHAR(50)    NOT NULL
        , ulica           NVARCHAR(100)   NOT NULL
        , nr_domu         NVARCHAR(10)    NOT NULL
        , nr_lokalu       NVARCHAR(5)     NOT NULL
        , miejscowosc     NVARCHAR(100)   NOT NULL
        , kod_pocztowy    NVARCHAR(10)    NOT NULL
        , poczta          NVARCHAR(100)   NOT NULL
    )
END
GO
```

```

-- utwórz tabelę NAGLOWEK do przechowywania nagłówków wyciągu bankowego
IF NOT EXISTS ( SELECT 1 FROM sysobjects o WHERE o.[name] = 'WB_NAGLOWEK'
AND (OBJECTPROPERTY(o.[ID], 'IsUserTable') = 1)
)
BEGIN
CREATE TABLE dbo.WB_NAGLOWEK
(
id_dokumentu INT NOT NULL IDENTITY CONSTRAINT PK_WB_NAGLOWEK PRIMARY KEY
, kod_formularza NVARCHAR(10) NOT NULL
, kod_systemowy NVARCHAR(15) NOT NULL
, wersja_schemy NVARCHAR(5) NOT NULL
, wariant_formularza TINYINT NOT NULL
, cel_zlozenia TINYINT NOT NULL
, data_wytworzenia_jpk DATETIME NOT NULL -- RRRR-MM-DDTHH:MM:SS, np. 2024-04-03T12:12:33
, data_od DATE NOT NULL -- RRRR-MM-DD
, data_do DATE NOT NULL -- RRRR-MM-DD
, domyslny_kod_waluty NCHAR(3) NOT NULL -- 3 litery (ISO 4217)
, kod_urzedu NCHAR(4) NOT NULL -- 4 cyfry
)
END
GO

-- utwórz tabelę POZYCJA do przechowywania szczegółów wyciągu bankowego
IF NOT EXISTS ( SELECT 1 FROM sysobjects o WHERE o.[name] = 'WB_POZYCJA'
AND (OBJECTPROPERTY(o.[ID], 'IsUserTable') = 1)
)
BEGIN
CREATE TABLE dbo.WB_POZYCJA
(
id_dokumentu INT NOT NULL
CONSTRAINT PK_WB_POZYCJA PRIMARY KEY
CONSTRAINT FK_WB_POZYCJA_NAGLOWEK FOREIGN KEY
REFERENCES WB_NAGLOWEK(id_dokumentu)
, id_podmiotu INT NOT NULL CONSTRAINT FK_WB_POZYCJA__PODMIOT
FOREIGN KEY REFERENCES WB_PODMIOT(id_podmiotu)
, numer_rachunku NCHAR(26) NOT NULL -- IBAN (26 cyfr)
, saldo_poczatkowe MONEY NOT NULL
, saldo_koncowe MONEY NOT NULL
, liczba_wierszy INT NOT NULL
, suma_obciazen MONEY NOT NULL
, suma_uznan MONEY NOT NULL
)
END
GO

-- utwórz tabelę WIERSZ do przechowywania wierszów wyciągu bankowego
IF NOT EXISTS ( SELECT 1 FROM sysobjects o WHERE o.[name] = 'WB_WIERSZ'
AND (OBJECTPROPERTY(o.[ID], 'IsUserTable') = 1)
)
BEGIN
CREATE TABLE dbo.WB_WIERSZ
(
id_dokumentu INT NOT NULL CONSTRAINT FK_WB_WIERSZ__NAGLOWEK
FOREIGN KEY REFERENCES WB_NAGLOWEK(id_dokumentu)
, numer_wiersza INT NOT NULL
CONSTRAINT PK_WB_WIERSZ PRIMARY KEY(id_dokumentu, numer_wiersza)
, data_operacji DATE NOT NULL -- RRRR-MM-DD
, nazwa_podmiotu NVARCHAR(255) NOT NULL
, opis_operacji NVARCHAR(255) NOT NULL
, kwota_operacji MONEY NOT NULL
, saldo_operacji MONEY NOT NULL
)
END
GO

```


4. Opis procedur

- Procedury pomocnicze:
 - remove_table – Usuń tabelę, jeżeli istnieje
 - create_empty_proc – Utwórz procedurę o zadanej nazwie jeżeli nie istnieje
 - create_empty_fun – Utwórz funkcję o zadanej nazwie jeżeli nie istnieje
 - text_to_money – Konwersja z typu tekstowego na typ money
 - text_to_date – Konwersja z typu tekstowego na typ date
 - text_to_datetime – Konwersja z typu tekstowego na typ datetime
- Procedury właściwe:
 - tmp_wb_pod_check – Sprawdź poprawność podmiotu w tabeli tymczasowej
 - Sprawdź, czy jest dokładnie 1 podmiot
 - Sprawdź czy id podmiotów unikalne
 - Sprawdź czy nip poprawny

```
-- Sprawdź czy poprawny nip
IF EXISTS (SELECT 1 FROM dbo.tmp_wb_pod t WHERE t.nip NOT LIKE REPLICATE('[0-9]', 10))
BEGIN
    SET @elog_n = @elog_n + 'Niepoprawny nip'
    INSERT INTO WB_ELOG_N(opis_n) VALUES (@elog_n)

    SET @id_en = SCOPE_IDENTITY()
    INSERT INTO WB_ELOG_D(id_elog_n, opis_d)
        SELECT DISTINCT @id_en, t.nip
        FROM tmp_wb_pod t WHERE t.nip NOT LIKE REPLICATE('[0-9]', 10)

    RAISERROR(@elog_n, 16, 4)
    SET @error = 1
    RETURN -1
END
```
 - Sprawdź czy pelna_nazwa nie jest pusta

```
-- Sprawdź czy pelna_nazwa nie jest pusta
IF EXISTS (SELECT 1 FROM dbo.tmp_wb_pod t WHERE t.pelna_nazwa NOT LIKE '%_%')
BEGIN
    SET @elog_n = @elog_n + 'Pusta pelna_nazwa'
    INSERT INTO WB_ELOG_N(opis_n) VALUES (@elog_n)

    SET @id_en = SCOPE_IDENTITY()
    INSERT INTO WB_ELOG_D(id_elog_n, opis_d)
        SELECT DISTINCT @id_en, t.pelna_nazwa
        FROM tmp_wb_pod t WHERE t.pelna_nazwa NOT LIKE '%_%'

    RAISERROR(@elog_n, 16, 4)
    SET @error = 1
    RETURN -1
END
```

- Sprawdź czy poprawny region

```
-- Sprawdź czy poprawny region
IF EXISTS (SELECT 1 FROM dbo.tmp_wb_pod t WHERE t.region NOT LIKE REPLICATE('[0-9]', 9))
BEGIN
    SET @elog_n = @elog_n + 'Niepoprawny region'
    INSERT INTO WB_ELOG_N(opis_n) VALUES (@elog_n)

    SET @id_en = SCOPE_IDENTITY()
    INSERT INTO WB_ELOG_D(id_elog_n, opis_d)
        SELECT DISTINCT @id_en, t.region
        FROM tmp_wb_pod t WHERE t.nip NOT LIKE REPLICATE('[0-9]', 9)

    RAISERROR(@elog_n, 16, 4)
    SET @error = 1
    RETURN -1
END
```

- Sprawdź czy kod_kraju nie jest pusty
- Sprawdź czy wojewodztwo nie jest puste
- Sprawdź czy powiat nie jest pusty
- Sprawdź czy gmina nie jest pusta
- Sprawdź czy ulica nie jest pusta
- Sprawdź czy nr_domu nie jest pusty
- Sprawdź czy miejscowosc nie jest pusta
- Sprawdź czy poprawny kod_pocztowy

```
-- Sprawdź czy poprawny kod_pocztowy
IF EXISTS (SELECT 1 FROM dbo.tmp_wb_pod t
    WHERE t.kod_pocztowy NOT LIKE REPLICATE('[0-9]', 2) + '-' + REPLICATE('[0-9]', 3))
BEGIN
    SET @elog_n = @elog_n + 'Niepoprawny kod pocztowy'
    INSERT INTO WB_ELOG_N(opis_n) VALUES (@elog_n)

    SET @id_en = SCOPE_IDENTITY()
    INSERT INTO WB_ELOG_D(id_elog_n, opis_d)
        SELECT DISTINCT @id_en, t.kod_pocztowy
        FROM tmp_wb_pod t
        WHERE t.kod_pocztowy NOT LIKE REPLICATE('[0-9]', 2) + '-' + REPLICATE('[0-9]', 3)

    RAISERROR(@elog_n, 16, 4)
    SET @error = 1
    RETURN -1
END
```

- Sprawdź czy poczta nie jest pusta

- tmp_wb_na_check – Sprawdź poprawność nagłówka w tabeli tymczasowej

- Sprawdź czy jest dokładnie 1 nagłówek
- Sprawdź czy id nagłówków unikalne
- Sprawdź czy kod_formularza prawidłowy

```
-- Sprawdź czy kod_formularza prawidłowy
IF EXISTS (SELECT 1 FROM tmp_wb_na t WHERE t.kod_formularza NOT LIKE 'JPK_WB')
BEGIN
    SET @elog_n = @elog_n + 'Kod formularza nie wskazuje na wyciąg bankowy / '
    INSERT INTO WB_ELOG_N(opis_n) VALUES (@elog_n)

    SET @id_en = SCOPE_IDENTITY()
    INSERT INTO WB_ELOG_D(id_elog_n, opis_d) VALUES (@id_en, 'Kod formularza różny od JPK_WB')

    RAISERROR(@elog_n, 16, 4)
    SET @error = 1
    RETURN -1
END
```

- Sprawdź czy kod_systemowy poprawny
- Sprawdź czy wersja_schemy poprawna
- Sprawdź czy wariant_formularza poprawny
- Sprawdź czy cel_zlozenia poprawny

- Sprawdź czy data_do nie jest w przyszłości

```
-- Sprawdź czy data_do nie jest w przyszłości
DECLARE @date_now nchar(10)
SET @date_now = CONVERT(nchar(10), GETDATE(), 102)

IF EXISTS (
    SELECT 1 FROM tmp_wb_na t
    WHERE CONVERT(nchar(10), REPLACE(t.data_do, '-', '.'), 102) >= @date_now
)
BEGIN
    SET @elog_n = @elog_n + 'data_do jest w przyszłości'
    INSERT INTO WB_ELOG_N(opis_n) VALUES (@elog_n)

    SET @id_en = SCOPE_IDENTITY()
    INSERT INTO WB_ELOG_D(id_eelog_n, opis_d)
        SELECT DISTINCT @id_en, t.data_do
        FROM tmp_wb_na t WHERE CONVERT(nchar(10), t.data_do, 102) >= @date_now

    RAISERROR(@elog_n, 16, 4)
    SET @error = 1
    RETURN -1
END
```

- Sprawdź czy data_od nie jest późniejsza niż data_do

```
-- Sprawdź czy data_od nie jest późniejsza niż data_do
IF EXISTS (SELECT 1 FROM tmp_wb_na t WHERE t.data_od > t.data_do)
BEGIN
    SET @elog_n = @elog_n + 'data_od jest później niż data_do'
    INSERT INTO WB_ELOG_N(opis_n) VALUES (@elog_n)

    SET @id_en = SCOPE_IDENTITY()
    INSERT INTO WB_ELOG_D(id_eelog_n, opis_d)
        SELECT DISTINCT @id_en, t.data_od + ' ' + t.data_do
        FROM tmp_wb_na t WHERE t.data_wytworzenia_jpk >= @datetime_now

    RAISERROR(@elog_n, 16, 4)
    SET @error = 1
    RETURN -1
END
```

- Sprawdź czy domyslny_kod_waluty jest prawidłowy
- Sprawdź czy kod_urzedu jest prawidłowy
- tmp_wb_poz_check – Sprawdź poprawność pozycji w tabeli tymczasowej
 - Sprawdź czy jest dokładnie 1 wiersz
 - Sprawdź czy id_dokumentu unikalne

- Sprawdź czy każde id_dokumentu z tmp_wb_pod istnieje w tmp_wb_na

```
-- Sprawdź czy każde id_dokumentu z tmp_wb_pod istnieje w tmp_wb_na
```

```
DECLARE @wrong_id_dokumentu_count int
SELECT @wrong_id_dokumentu_count = COUNT(*)
FROM tmp_wb_poz p
WHERE NOT EXISTS
( SELECT 1
  FROM tmp_wb_na n
  WHERE p.id_dokumentu = n.id_dokumentu
)

IF @wrong_id_dokumentu_count > 0
BEGIN
  SET @elog_n = @elog_n + 'Pozycja bez nagłówka'
  INSERT INTO WB_ELOG_N(opis_n) VALUES (@elog_n)

  SET @id_en = SCOPE_IDENTITY()
  INSERT INTO WB_ELOG_D(id_elog_n, opis_d)
  SELECT @id_en, p.id_dokumentu
  FROM tmp_wb_poz p
  WHERE NOT EXISTS
  ( SELECT 1
    FROM tmp_wb_na n
    WHERE p.id_dokumentu = n.id_dokumentu
  )

  RAISERROR(@elog_n, 16, 4)
  SET @error = 1
  RETURN -1
END
```

- Sprawdź czy każde id_dokumentu z tmp_wb_na istnieje w tmp_wb_poz
- Sprawdź czy każde id_podmiotu z tmp_wb_poz istnieje w tmp_wb_pod
- Sprawdź czy numer_rachunku poprawny
- Sprawdź czy saldo_początkowe jest kwotą
- Sprawdź czy saldo_koncowe jest kwotą
- Sprawdź czy liczba wierszy jest liczbą całkowitą większą od 0

```
-- Sprawdź czy liczba wierszy jest liczbą całkowitą większą od 0
```

```
IF EXISTS (
  SELECT 1 FROM tmp_wb_poz t
  WHERE ISNUMERIC(t.liczba_wierszy) <> 1 OR CAST(t.liczba_wierszy AS INT) < 1
)
BEGIN
  SET @elog_n = @elog_n + 'Liczba wierszy nie jest liczbą całkowitą większą od 0'
  INSERT INTO WB_ELOG_N(opis_n) VALUES (@elog_n)

  SET @id_en = SCOPE_IDENTITY()
  INSERT INTO WB_ELOG_D(id_elog_n, opis_d)
  SELECT DISTINCT @id_en, t.saldo_koncowe
  FROM tmp_wb_poz t
  WHERE ISNUMERIC(t.liczba_wierszy) <> 1 OR CAST(t.liczba_wierszy AS INT) < 1

  RAISERROR(@elog_n, 16, 4)
  SET @error = 1
  RETURN -1
END
```

- Sprawdź czy suma_obciazen jest kwotą

```
-- Sprawdź czy suma_obciazen jest kwotą
IF EXISTS (SELECT 1 FROM tmp_wb_poz t WHERE TRY_CONVERT(MONEY, t.suma_obciazen) IS NULL)
BEGIN
    SET @elog_n = @elog_n + 'Suma obciążeń nie jest kwotą'
    INSERT INTO WB_ELOG_N(opis_n) VALUES (@elog_n)

    SET @id_en = SCOPE_IDENTITY()
    INSERT INTO WB_ELOG_D(id_elog_n, opis_d)
        SELECT DISTINCT @id_en, t.suma_obciazen
        FROM tmp_wb_poz t WHERE TRY_CONVERT(MONEY, t.suma_obciazen) IS NULL

    RAISERROR(@elog_n, 16, 4)
    SET @error = 1
    RETURN -1
END
```

- Sprawdź czy suma_uznan jest kwotą

- Sprawdź czy saldo_koncowe jest prawidłowe

```
-- Sprawdź czy saldo_koncowe jest prawidłowe
IF EXISTS (
    SELECT 1 FROM tmp_wb_poz t
    WHERE CAST(REPLACE(t.saldo_koncowe, ',', '.') AS MONEY)
        <> CAST(REPLACE(t.saldo_poczkowe, ',', '.') AS MONEY)
        - CAST(REPLACE(t.suma_obciazen, ',', '.') AS MONEY)
        + CAST(REPLACE(t.suma_uznan, ',', '.') AS MONEY)
)
BEGIN
    SET @elog_n = @elog_n + 'Saldo końcowe nie jest prawidłowe'
    INSERT INTO WB_ELOG_N(opis_n) VALUES (@elog_n)

    SET @id_en = SCOPE_IDENTITY()
    INSERT INTO WB_ELOG_D(id_elog_n, opis_d)
        SELECT DISTINCT @id_en, t.suma_obciazen
        FROM tmp_wb_poz t
        WHERE CAST(REPLACE(t.saldo_koncowe, ',', '.') AS MONEY)
            <> CAST(REPLACE(t.saldo_poczkowe, ',', '.') AS MONEY)
            - CAST(REPLACE(t.suma_obciazen, ',', '.') AS MONEY)
            + CAST(REPLACE(t.suma_uznan, ',', '.') AS MONEY)

    RAISERROR(@elog_n, 16, 4)
    SET @error = 1
    RETURN -1
END
```

- tmp_wb_poz_wiersz – Sprawdź poprawność wierszy w tabeli tymczasowej

- Sprawdź czy każde id_dokumentu z tmp_wb_wiersz istnieje w tmp_wb_na
- Sprawdź czy liczba_wierszy z tmp_wb_poz jest równa liczbie wierszy z tmp_wb_wiersz
- Sprawdź czy data operacji jest datą
- Sprawdź czy nazwa_podmiotu nie jest pusta
- Sprawdź czy opis_operacji nie jest pusty
- Sprawdź czy kwota_operacji jest kwotą
- Sprawdź czy saldo_operacji jest kwotą
- Sprawdź czy kwota_operacji odpowiada saldo_operacji
- Sprawdź czy różnica kwot z tmp_wb_poz odpowiada sumie kwot z tmp_wb_wiersz

- wb_move_data – przenieś dane z tabeli tymczasowych do tabeli docelowych

- Wyczyść tabele docelowe
- Dodaj dane do tabeli WB_PODMIOT
- Dodaj dane do tabeli WB_NAGLOWEK
- Dodaj dane do tabeli WB_POZYCJA

- Dodaj dane do tabeli WB_WIERSZ
- generuj_raport – generuje raport w formacie XML

```

;WITH XMLNAMESPACES
(
    N'http://jpk.mf.gov.pl/wzor/2016/03/09/03092/' AS tns,
    N'http://crd.gov.pl/xml/schematy/dziedziczne/mf/2016/01/25/eD/DefinicjeTypy/' AS etd
)
SELECT @xml = (
    SELECT
    (
        SELECT
            n.kod_formularza AS [tns:KodFormularza],
            n.kod_systemowy AS [tns:WariantFormularza],
            n.cel_zlozenia AS [tns:CelZlozenia],
            dbo.SAFT_DATETIME(n.data_wytworzenia_jpk) AS [tns:DataWytworzeniaJPK],
            dbo.SAFT_DATE(n.data_od) AS [tns:DataOd],
            dbo.SAFT_DATE(n.data_do) AS [tns:DataDo],
            n.domyslny_kod_waluty AS [tns:DomyslnyKodWaluty],
            n.kod_urzedu AS [tns:KodUrzedu]
        FOR XML PATH('tns:Naglowek'), TYPE
    ),
    (
        SELECT
        (
            SELECT
                pod.nip AS [etd:NIP],
                pod.pelna_nazwa AS [etd:PełnaNazwa],
                pod.regon AS [etd:REGON]
            FOR XML PATH('tns:IdentyfikatorPodmiotu'), TYPE
        ),
        (
            SELECT
                pod.kod_kraju AS [etd:KodKraju],
                pod.wojewodztwo AS [etd:Wojewodztwo],
                pod.powiat AS [etd:Powiat],
                pod.gmina AS [etd:Gmina],
                pod.ulica AS [etd:Ulica],
                pod.nr_domu AS [etd:NrDomu],
                pod.nr_lokalu AS [etd:NrLokalu],
                pod.miejscowosc AS [etd:Miejscowosc],
                pod.kod_pocztowy AS [etd:KodPocztowy],
                pod.powiat AS [etd:Poczta]
            FOR XML PATH('tns:AdresPodmiotu'), TYPE
        )
        FOR XML PATH('tns:Podmiot1'), TYPE
    ),
    poz. numer_rachunku AS [tns:NumerRachunku]
    ,
    (
        SELECT
            dbo.SAFT_GET_AMT(poz.saldo_pocztkowe) AS [tns:SaldoPocztkowe],
            dbo.SAFT_GET_AMT(poz.saldo_koncowe) AS [tns:SaldoKoncowe]
        FOR XML PATH('tns:Salda'), TYPE
    ),
    (
        SELECT
            w.numer_wiersza AS [tns:NumerWiersza],
            dbo.SAFT_DATE(w.data_operacji) AS [tns:DataOperacji],
            w.nazwa_podmiotu AS [tns:NazwaPodmiotu],
            w.opis_operacji AS [tns:OpisOperacji],
            dbo.SAFT_GET_AMT(w.kwota_operacji) AS [tns:KwotaOperacji],
            dbo.SAFT_GET_AMT(w.saldo_operacji) AS [tns:SaldoOperacji]
        FROM WB_WIERSZ w
        JOIN WB_NAGLOWEK ON (w.id_dokumentu = n.id_dokumentu)
        FOR XML PATH('tns:WyciagWiersz'), TYPE
    ),
    (
        SELECT
            poz.liczba_wierszy AS [tns:LiczbaWierszy],
            dbo.SAFT_GET_AMT(poz.suma_obciazen) AS [tns:SumaObciazen],
            dbo.SAFT_GET_AMT(poz.suma_uznan) AS [tns:SumaUznan]
        FOR XML PATH('tns:WyciagCtrl'), TYPE
    )
    FOR XML PATH('tns:JPK'), TYPE
)
FROM WB_POZYCJA poz
JOIN WB_NAGLOWEK n ON (n.id_dokumentu = poz.id_dokumentu)
JOIN WB_PODMIOT pod ON (pod.id_podmiotu = poz.id_podmiotu)

SELECT @xml

```

```
<tns:JPK xmlns:etd="http://crd.gov.pl/xml/schematy/dziedzinowe/mf/201
<tns:Naglowek xmlns:etd="http://crd.gov.pl/xml/schematy/dziedzinowe
  <tns:KodFormularza>JPK_WB</tns:KodFormularza>
  <tns:WariantFormularza>JPK_WB (1)</tns:WariantFormularza>
  <tns:CelZlozenia>1</tns:CelZlozenia>
  <tns:DataWytworzeniaJPK>2024-04-03</tns:DataWytworzeniaJPK>
  <tns:DataOd>2024-03-01</tns:DataOd>
  <tns:DataDo>2024-03-31</tns:DataDo>
  <tns:DomyslnyKodWaluty>PLN</tns:DomyslnyKodWaluty>
  <tns:KodUrzedu>0271</tns:KodUrzedu>
</tns:Naglowek>
<tns:Podmiot1 xmlns:etd="http://crd.gov.pl/xml/schematy/dziedzinowe
  <tns:IdentyfikatorPodmiotu xmlns:etd="http://crd.gov.pl/xml/schem
    <etd:NIP>1577568942</etd:NIP>
    <etd:PeInaNazwa>Microsoft Corporation</etd:PeInaNazwa>
    <etd:REGON>493996463</etd:REGON>
  </tns:IdentyfikatorPodmiotu>
  <tns:AdresPodmiotu xmlns:etd="http://crd.gov.pl/xml/schematy/dzie
    <etd:KodKraju>POLSKA</etd:KodKraju>
    <etd:Wojewodztwo>mazowieckie</etd:Wojewodztwo>
    <etd:Powiat>m.st. Warszawa</etd:Powiat>
    <etd:Gmina>Warszawa</etd:Gmina>
    <etd:Ulica>al. Jerozolimskie</etd:Ulica>
    <etd:NrDomu>195A</etd:NrDomu>
    <etd:NrLokalu />
    <etd:Miejscowosc>Warszawa</etd:Miejscowosc>
    <etd:KodPocztowy>02-222</etd:KodPocztowy>
    <etd:Poczta>m.st. Warszawa</etd:Poczta>
  </tns:AdresPodmiotu>
</tns:Podmiot1>
<tns:NumerRachunku>44114020205082626899431966</tns:NumerRachunku>
<tns:Salda xmlns:etd="http://crd.gov.pl/xml/schematy/dziedzinowe/mf
  <tns:SaldoPocztowe>100.43</tns:SaldoPocztowe>
  <tns:SaldoKoncowe>240.00</tns:SaldoKoncowe>
</tns:Salda>
<tns:WyciagWiersz xmlns:etd="http://crd.gov.pl/xml/schematy/dziedzi
  <tns:NumerWiersza>1</tns:NumerWiersza>
  <tns:DataOperacji>2024-03-03</tns:DataOperacji>
  <tns:NazwaPodmiotu>Budowa</tns:NazwaPodmiotu>
  <tns:OpisOperacji>Za wykonanie usługi</tns:OpisOperacji>
  <tns:KwotaOperacji>100.34</tns:KwotaOperacji>
  <tns:SaldoOperacji>100.34</tns:SaldoOperacji>
</tns:WyciagWiersz>
<tns:WyciagWiersz xmlns:etd="http://crd.gov.pl/xml/schematy/dziedzi
  <tns:NumerWiersza>2</tns:NumerWiersza>
  <tns:DataOperacji>2024-03-07</tns:DataOperacji>
  <tns:NazwaPodmiotu>Kanalizacja</tns:NazwaPodmiotu>
  <tns:OpisOperacji>.</tns:OpisOperacji>
  <tns:KwotaOperacji>19.00</tns:KwotaOperacji>
  <tns:SaldoOperacji>19.00</tns:SaldoOperacji>
</tns:WyciagWiersz>
<tns:WyciagWiersz xmlns:etd="http://crd.gov.pl/xml/schematy/dziedzi
  <tns:NumerWiersza>3</tns:NumerWiersza>
  <tns:DataOperacji>2024-03-24</tns:DataOperacji>
  <tns:NazwaPodmiotu>Remontownia</tns:NazwaPodmiotu>
  <tns:OpisOperacji>Dziękuję za pomyślną współpracę</tns:OpisOperacji>
  <tns:KwotaOperacji>20.23</tns:KwotaOperacji>
  <tns:SaldoOperacji>20.23</tns:SaldoOperacji>
</tns:WyciagWiersz>
<tns:WyciagCtrl xmlns:etd="http://crd.gov.pl/xml/schematy/dziedzinowe/m
  <tns:LiczbaWierszy>3</tns:LiczbaWierszy>
  <tns:SumaObciazen>0.00</tns:SumaObciazen>
  <tns:SumaUznan>139.57</tns:SumaUznan>
</tns:WyciagCtrl>
</tns:JPK>
```


5. Testy poprawności

Przykład pomyślnego wykonania transferu danych:

```
DTEXec: The package execution returned DTSER_SUCCESS (0).
```

- Zły kod_formularza

```
Error: 2024-05-12 14:19:36.21
Code: 0xC002F210
Source: Waliduj dane tymczasowe Execute SQL Task
Description: Executing the query "EXEC tmp_wb_na_check" failed with the following error: "Błąd w procedurze: tmp_wb_na_check / Kod formularza nie wskazuje na wyciąg bankowy / ". Possible failure reasons: Problems with the query, "ResultSet" property not set correctly, parameters not set correctly, or connection not established correctly.
End Error
DTEXec: The package execution returned DTSER_FAILURE (1).
Started: 14:19:35
```

- Zły kod_systemowy

```
Error: 2024-05-12 14:21:03.09
Code: 0xC002F210
Source: Waliduj dane tymczasowe Execute SQL Task
Description: Executing the query "EXEC tmp_wb_na_check" failed with the following error: "Błąd w procedurze: tmp_wb_na_check / Kod systemowy niepoprawny". Possible failure reasons: Problems with the query, "ResultSet" property not set correctly, parameters not set correctly, or connection not established correctly.
End Error
DTEXec: The package execution returned DTSER_FAILURE (1).
```

- Zła data_od

```
Error: 2024-05-12 14:23:01.06
Code: 0xC002F210
Source: Waliduj dane tymczasowe Execute SQL Task
Description: Executing the query "EXEC tmp_wb_na_check" failed with the following error: "Błąd w procedurze: tmp_wb_na_check / data_od jest później niż data_do". Possible failure reasons: Problems with the query, "ResultSet" property not set correctly, parameters not set correctly, or connection not established correctly.
End Error
DTEXec: The package execution returned DTSER_FAILURE (1).
```

- Zła data_do

```
Error: 2024-05-12 14:23:50.86
Code: 0xC002F210
Source: Waliduj dane tymczasowe Execute SQL Task
Description: Executing the query "EXEC tmp_wb_na_check" failed with the following error: "Błąd w procedurze: tmp_wb_na_check / data_do jest w przyszłości". Possible failure reasons: Problems with the query, "ResultSet" property not set correctly, parameters not set correctly, or connection not established correctly.
End Error
DTEXec: The package execution returned DTSER_FAILURE (1).
```

- Zły kod_urzedu

```
Error: 2024-05-12 14:24:40.47
Code: 0xC002F210
Source: Waliduj dane tymczasowe Execute SQL Task
Description: Executing the query "EXEC tmp_wb_na_check" failed with the following error: "Błąd w procedurze: tmp_wb_na_check / Kod urzędu nie jest 4-cyfrowy". Possible failure reasons: Problems with the query, "ResultSet" property not set correctly, parameters not set correctly, or connection not established correctly.
End Error
DTEXec: The package execution returned DTSER_FAILURE (1).
```

- Zły nip

```
Error: 2024-05-12 14:26:26.39
Code: 0xC002F210
Source: Waliduj dane tymczasowe Execute SQL Task
Description: Executing the query "EXEC tmp_wb_pod_check" failed with the following error: "Błąd w procedurze: tmp_wb_pod_check / Niepoprawny nip". Possible failure reasons: Problems with the query, "ResultSet" property not set correctly, parameters not set correctly, or connection not established correctly.
End Error
DTEXec: The package execution returned DTSER_FAILURE (1).
```

- Pusta pelna_nazwa

```
Error: 2024-05-12 14:27:02.17
Code: 0xC002F210
Source: Waliduj dane tymczasowe Execute SQL Task
Description: Executing the query "EXEC tmp_wb_pod_check" failed with the following error: "Błąd w procedurze: tmp_wb_pod_check / Pusta pelna_nazwa". Possible failure reasons: Problems with the query, "ResultSet" property not set correctly, parameters not set correctly, or connection not established correctly.
End Error
DTEXec: The package execution returned DTSER_FAILURE (1).
```


- Zły region

```
Error: 2024-05-12 14:27:43.88
Code: 0xC002F210
Source: Waliduj dane tymczasowe Execute SQL Task
Description: Executing the query "EXEC tmp_wb_pod_check" failed with the following error: "Błąd w procedurze: tmp_wb_pod_check / Niepoprawny region". Possible failure reasons: Problems with the query, "ResultSet" property not set correctly, parameters not set correctly, or connection not established correctly.
End Error
DTEXec: The package execution returned DTSEr_FAILURE (1).
```

- Pusta gmina

```
Error: 2024-05-12 14:28:26.83
Code: 0xC002F210
Source: Waliduj dane tymczasowe Execute SQL Task
Description: Executing the query "EXEC tmp_wb_pod_check" failed with the following error: "Błąd w procedurze: tmp_wb_pod_check / Pusta gmina". Possible failure reasons: Problems with the query, "ResultSet" property not set correctly, parameters not set correctly, or connection not established correctly.
End Error
DTEXec: The package execution returned DTSEr_FAILURE (1).
```

- Zła suma_uznan

```
Error: 2024-05-12 14:10:02.68
Code: 0xC002F210
Source: Waliduj dane tymczasowe Execute SQL Task
Description: Executing the query "EXEC tmp_wb_poz_check" failed with the following error: "Błąd w procedurze: tmp_wb_poz_check / Saldo końcowe nie jest prawidłowe". Possible failure reasons: Problems with the query, "ResultSet" property not set correctly, parameters not set correctly, or connection not established correctly.
End Error
DTEXec: The package execution returned DTSEr_FAILURE (1).
```

- Zła suma_obciazen

```
Error: 2024-05-12 14:11:24.62
Code: 0xC002F210
Source: Waliduj dane tymczasowe Execute SQL Task
Description: Executing the query "EXEC tmp_wb_poz_check" failed with the following error: "Błąd w procedurze: tmp_wb_poz_check / Saldo końcowe nie jest prawidłowe". Possible failure reasons: Problems with the query, "ResultSet" property not set correctly, parameters not set correctly, or connection not established correctly.
End Error
DTEXec: The package execution returned DTSEr_FAILURE (1).
```

- Złe id_dokumentu

```
Error: 2024-05-12 14:13:43.49
Code: 0xC002F210
Source: Waliduj dane tymczasowe Execute SQL Task
Description: Executing the query "EXEC tmp_wb_poz_check" failed with the following error: "Błąd w procedurze: tmp_wb_poz_check / Pozycja bez nagłówka". Possible failure reasons: Problems with the query, "ResultSet" property not set correctly, parameters not set correctly, or connection not established correctly.
End Error
DTEXec: The package execution returned DTSEr_FAILURE (1).
```

- Złe id_podmiotu

```
Error: 2024-05-12 14:14:44.73
Code: 0xC002F210
Source: Waliduj dane tymczasowe Execute SQL Task
Description: Executing the query "EXEC tmp_wb_poz_check" failed with the following error: "Błąd w procedurze: tmp_wb_poz_check / Pozycja bez podmiotu". Possible failure reasons: Problems with the query, "ResultSet" property not set correctly, parameters not set correctly, or connection not established correctly.
End Error
DTEXec: The package execution returned DTSEr_FAILURE (1).
```

- Zły numer_rachunku

```
Error: 2024-05-12 14:16:01.03
Code: 0xC002F210
Source: Waliduj dane tymczasowe Execute SQL Task
Description: Executing the query "EXEC tmp_wb_poz_check" failed with the following error: "Błąd w procedurze: tmp_wb_poz_check / Numer rachunku nie jest 26-cyfrowy". Possible failure reasons: Problems with the query, "ResultSet" property not set correctly, parameters not set correctly, or connection not established correctly.
End Error
DTEXec: The package execution returned DTSEr_FAILURE (1).
```

- Zła liczba_wierszy w poz

```
Error: 2024-05-12 14:18:19.58
Code: 0xC002F210
Source: Waliduj dane tymczasowe Execute SQL Task
Description: Executing the query "EXEC tmp_wb_wiersz_check" failed with the following error: "Błąd w procedurze: tmp_wb_wiersz_check / Liczba wierszy z poz nie zgadza się z liczbą wierszy z wier". Possible failure reasons: Problems with the query, "ResultSet" property not set correctly, parameters not set correctly, or connection not established correctly.
End Error
```

- Zła liczba wierszy w wiersz

```
Error: 2024-05-12 14:31:03.59
Code: 0xC002F210
Source: Waliduj dane tymczasowe Execute SQL Task
Description: Executing the query "EXEC tmp_wb_wiersz_check" failed with the following error: "Błąd w procedurze: tmp_wb_wiersz_check / Wiersz bez nagłówka". Possible failure reasons: Problems with the query, "ResultSet" property not set correctly, parameters not set correctly, or connection not established correctly.
End Error
DTEXec: The package execution returned DTSER_FAILURE (1).
```

- Zła data operacji

```
Error: 2024-05-12 14:33:25.80
Code: 0xC002F210
Source: Waliduj dane tymczasowe Execute SQL Task
Description: Executing the query "EXEC tmp_wb_wiersz_check" failed with the following error: "Błąd w procedurze: tmp_wb_wiersz_check / Niepoprawny format daty". Possible failure reasons: Problems with the query, "ResultSet" property not set correctly, parameters not set correctly, or connection not established correctly.
End Error
DTEXec: The package execution returned DTSER_FAILURE (1).
```

- Pusta nazwa_podmiotu

```
Error: 2024-05-12 14:34:15.82
Code: 0xC002F210
Source: Waliduj dane tymczasowe Execute SQL Task
Description: Executing the query "EXEC tmp_wb_wiersz_check" failed with the following error: "Błąd w procedurze: tmp_wb_wiersz_check / Pusta nazwa podmiotu". Possible failure reasons: Problems with the query, "ResultSet" property not set correctly, parameters not set correctly, or connection not established correctly.
End Error
DTEXec: The package execution returned DTSER_FAILURE (1).
```

- Złe saldo_operacji

```
Error: 2024-05-12 14:35:17.58
Code: 0xC002F210
Source: Waliduj dane tymczasowe Execute SQL Task
Description: Executing the query "EXEC tmp_wb_wiersz_check" failed with the following error: "Błąd w procedurze: tmp_wb_wiersz_check / Saldo operacji nie odpowiada kwocie". Possible failure reasons: Problems with the query, "ResultSet" property not set correctly, parameters not set correctly, or connection not established correctly.
End Error
DTEXec: The package execution returned DTSER_FAILURE (1).
```

- Zła kwota_operacji

```
Error: 2024-05-12 14:36:05.29
Code: 0xC002F210
Source: Waliduj dane tymczasowe Execute SQL Task
Description: Executing the query "EXEC tmp_wb_wiersz_check" failed with the following error: "Błąd w procedurze: tmp_wb_wiersz_check / Saldo operacji nie odpowiada kwocie". Possible failure reasons: Problems with the query, "ResultSet" property not set correctly, parameters not set correctly, or connection not established correctly.
End Error
DTEXec: The package execution returned DTSER_FAILURE (1).
```

6. Wnioski

Użycie pakietu DTSX jest bardzo użyteczne, ponieważ pozwala zautomatyzować transfer danych z pliku tekstowego do bazy danych przy użyciu komendy w wierszu poleceń. Miałem jednak duży problem z instalacją tego pakietu w Microsoft SQL Server Management Studio i Microsoft Visual Studio.

Język Microsoft SQL okazał się nie najlepszym narzędziem do walidacji danych tekstowych. W zapytaniu z LIKE brakuje wielu podstawowych operacji, które znajdują się w REGEXie. Najbardziej zabrakło mi operatora ? z REGEXa, który stosuje się do zaznaczania opcjonalnego ciągu znaków – w Microsoft SQL nie ma takiej możliwości. Ponadto żeby powtórzyć dany ciąg należało użyć funkcji REPLICATE, co według mnie jest nadmiarowe w porównaniu do operatora {} z REGEXa. Warto jednak zaznaczyć, że w pewnych przypadkach LIKE rozwiązano lepiej niż w REGEXie, np. sprawdzenie daty za pomocą funkcji ISDATE() czy sprawdzenie liczby za pomocą ISNUMERIC() albo sprawdzenie kwoty pieniędzy za pomocą ISMONEY().

Podejście z przesyłaniem danych źródłowych do tabeli tymczasowych pozwoliło na walidację danych w całości w Microsoft SQL.

Spotkałem się z sytuacją gdy funkcja działała nieprawidłowo z powodu przestania `nvarchar` o długości większej niż zadeklarowana w argumencie deklaracji tej funkcji. Uważam, że w takiej sytuacji program powinien sam powiadomić o takiej sytuacji.

Przydatną funkcją okazało się generowanie pliku XML bezpośrednio w Microsoft SQL. Kod odpowiadający za to jest czytelny i intuicyjny. Nie spodobało mi się jednak to że każdy element agregujący ma identyczną przestrzeń nazw zadeklarowaną przed generacją pliku, nawet jeżeli ten element nie używa wszystkich jej zbiorów.

Uważam, że cel ćwiczenia został spełniony. Program generuje poprawny raport XML otrzymując na wejściu pliki tekstowe z danymi. Wszystkie założenia zostały spełnione.

7. Dalsze możliwości rozwoju

- Wprowadzenie wielowalutowości.
- Wprowadzenie szczegółowej walidacji danych adresowych. Program mógłby sprawdzić czy wprowadzono istniejące województwo, powiat itd. Oprócz tego mógłby sprawdzić czy taki adres rzeczywiście istnieje i poinformować o konieczności ręcznego sprawdzenia.
- Sprawdzanie czy dana firma istnieje w spisie krajowym poprzez porównanie REGON, NIP i nazwy podmiotu.
- Przygotowanie do implementacji nowych wersji standardu JPK_WB.