

Sprawozdanie

Arkadiusz Ciepliński 319028
12.04.2024

Cel projektu

Celem projektu jest utworzenie systemu, który na podstawie danych o transakcjach klienta otrzymywanych od banku tworzy plik XML o prawidłowej strukturze pliku JPK_WB zdefiniowanym przez Ministerstwo Finansów w Polsce.

Architektura technologiczna

Realizacja projektu zakłada użycie MS SQL oraz MS Business Intelligence do budowy aplikacji. Za pomocą MS Business Intelligence utworzone zostaną pakiety SSIS importujące dane do MS SQL. Walidacja danych, przetwarzanie i generowanie pliku XML będą realizowane przez odpowiednie procedury SQL.

Sposób realizacji

W celu zrealizowania celu projektu tworzony system wykonuje transfer, przetwarzanie i walidację otrzymywanych danych oraz generowanie raportu w ustalonym formacie. Dane są przesyłane do bazy danych, gdzie następuje ich walidacja. Jeśli walidacja przebiegnie poprawnie, dane są odpowiednio przetwarzane, a pliki z danymi są usuwane (nie są potrzebne, ponieważ dane znajdują się już wtedy w bazie danych). Jeśli podczas walidacji wykryto błąd, możliwe jest odczytanie z tabeli logów błędów przyczyny błędu. Na koniec wywoływana jest procedura tworząca plik XML.

Funkcjonalności

Poniżej wypisano poszczególne funkcjonalności tworzonego systemu

- Prawidłowe odczytanie danych wejściowych z dostarczonych zbiorów danych
- Walidacja danych wejściowych pod kątem zgodności ze specyfikacją JPK_WB, w tym formatów dat, kwot oraz identyfikatorów transakcji
- Przechowywanie informacji o błędach wykrytych w procesie walidacji
- Odpowiednie przetwarzanie danych
- Generowanie struktury pliku XML zgodnie ze specyfikacją JPK_WB, w tym sekcji nagłówkowej i szczegółowych wierszy dotyczących poszczególnych transakcji

Dane wejściowe

System przyjmuje na wejściu 2 pliki w formacie *txt*, w których kolejne kolumny są oddzielone tabulatorami:

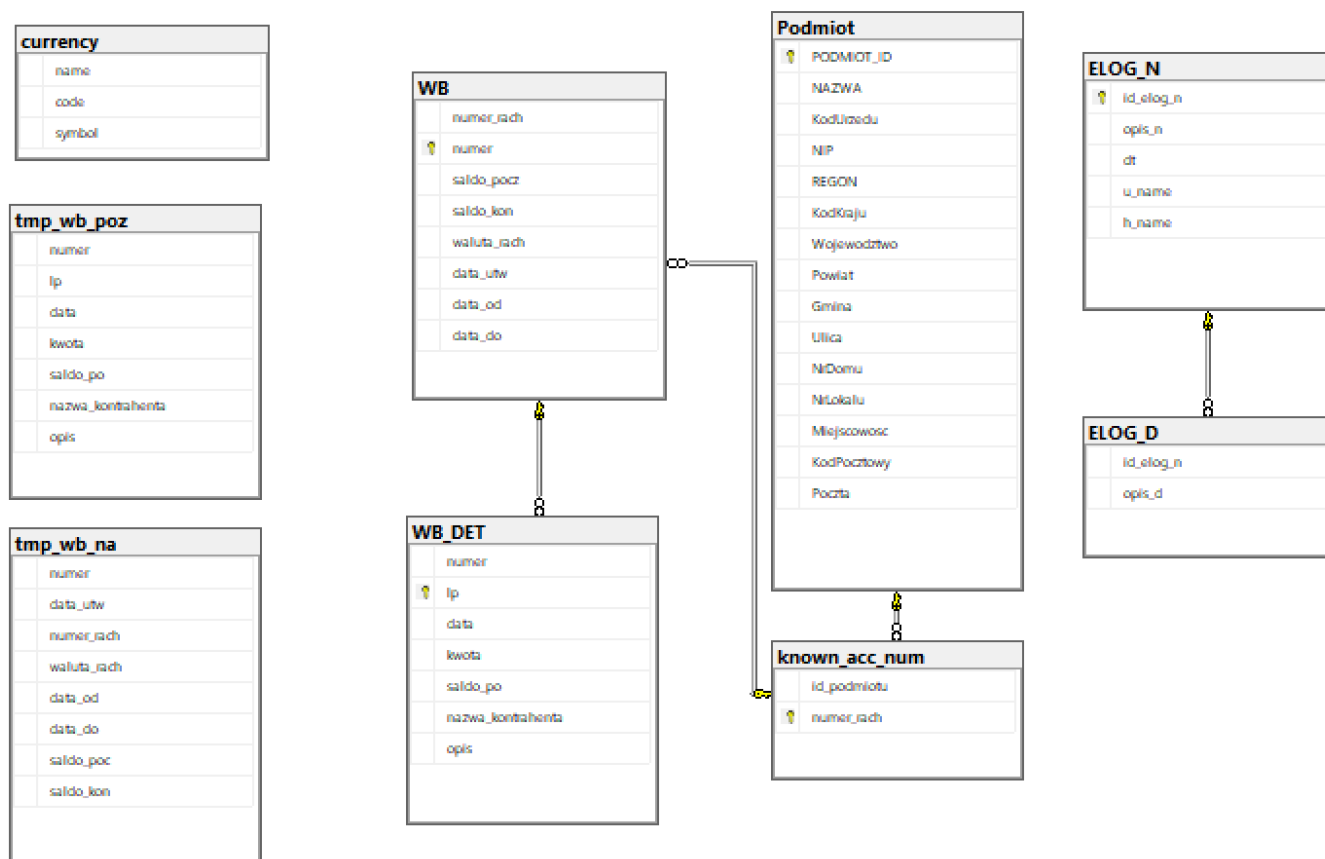
- Nagłówki wyciągów – zawierają ogólne informacje o dokumencie
Reprezentacja przykładowego pliku z nagłówkami w formacie tabeli:

| numer | data_utw | numer_rach | waluta_rach | data_od | data_do | saldo_poc | saldo_kon |
|-------------|------------|------------------------------|-------------|------------|------------|-----------|-----------|
| 12345AB6789 | 07.05.2024 | PL61109010140000071219812874 | PLN | 01.04.2024 | 30.04.2024 | 1000.00 | 1500.00 |

- Pozycje wyciągów – zawierają szczegółowe informacje o każdej transakcji z wyciągu
Reprezentacja przykładowego pliku z pozycjami w formacie tabeli:

| numer | lp | data | kwota | saldo_po | nazwa_kontrahenta | opis |
|-------------|----|------------|----------|----------|----------------------------|------------------------------------|
| 12345AB6789 | 1 | 01.04.2024 | -500.00 | 500.00 | Telekomunikacja Polska S.A | Opłata za usługi telekomunikacyjne |
| 12345AB6789 | 2 | 03.04.2024 | 2000.00 | 2500.00 | XYZ Sp. z o.o. | Wpłata od XYZ Sp. z o.o. |
| 12345AB6789 | 3 | 15.04.2024 | -1000.00 | 1500.00 | ABCD | Przelew wychodzący na rzecz ABCD |
| 12345AB6789 | 4 | 20.04.2024 | -200.00 | 1300.00 | Office Supplies Co. | Zakup materiałów biurowych |
| 12345AB6789 | 5 | 27.04.2024 | 500.00 | 1800.00 | Jan Kowalski | Wpłata gotówkowa |

Diagram bazy danych

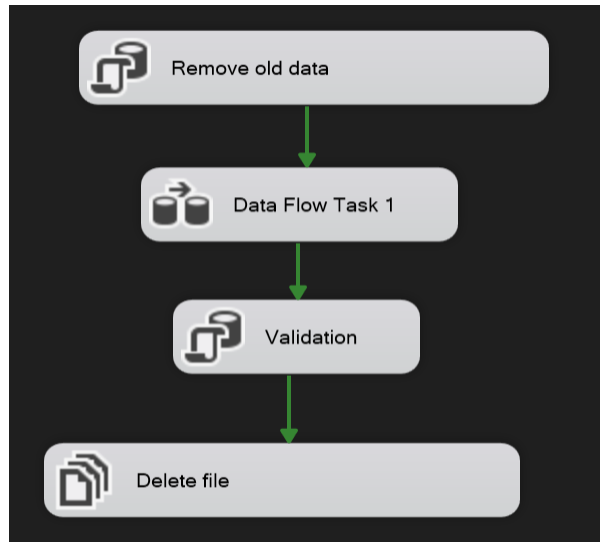


Pakiety SSIS

Utworzone zostały 2 pakiety SSIS. Jeden z nich obsługuje nagłówki a drugi pozycje wyciągów. Obydwa pakiety usuwają dane z tabel tymczasowych używanych do transferu danych, pobierają dane z plików, wprowadzają je do bazy danych, walidują pobrane dane i na koniec usuwają pliki tekstowe z danymi.

W przykładach przedstawiono pakiet obsługujący nagłówki, pakiet obsługujący pozycję działa analogicznie do tego pakietu.

Schemat pakietu SSIS:

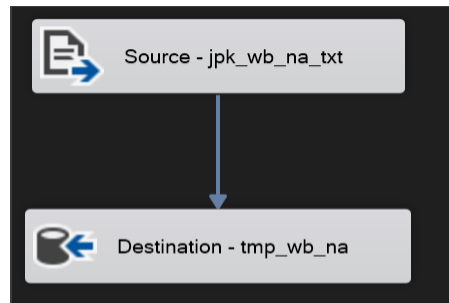


Szczegóły kolejnych zadań realizowanych przez pakiet

Usuwanie starych danych z tabeli tymczasowych

| General | |
|------------------------|--------------------------------------|
| Name | Remove old data |
| Description | Delete old data from temporary table |
| Options | |
| Timeout | 0 |
| CodePage | 1252 |
| TypeConversionMode | Allowed |
| Result Set | |
| ResultSet | None |
| SQL Statement | |
| ConnectionType | OLE DB |
| Connection | DestinationConnectionOLEDB |
| SQLSourceType | Direct input |
| SQLStatement | DELETE FROM [dbo].[tmp_wb_na] |
| IsQueryStoredProcedure | False |
| BypassPrepare | True |

Przepływ danych z plików do tabel



Connection Manager
Columns
Error Output

Flat file connection manager:

SourceConnectionFlatFile New...

☐ Retain null values from the source as null values in the data flow

Preview...

Connection Manager
Columns
Error Output

Available External...

- ☒ Name
- ☒ numer
- ☒ data_utw
- ☒ numer_rach
- ☒ waluta_rach
- ☒ data_od
- ☒ data_do
- ☒ saldo_poc

| External Column | Output Column |
|-----------------|---------------|
| numer | numer |
| data_utw | data_utw |
| numer_rach | numer_rach |
| waluta_rach | waluta_rach |
| data_od | data_od |
| data_do | data_do |
| saldo_poc | saldo_poc |
| saldo_kon | saldo_kon |

Connection Manager
Mappings
Error Output

Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command either by typing the query or by using Query Builder. For fast-load data access, set the table update options.

OLE DB connection manager:

DestinationConnectionOLEDB New...

Data access mode:

Table or view - fast load

Name of the table or the view:

[dbo].[tmp_wb_na] New...

☐ Keep identity ☒ Table lock

☐ Keep nulls ☒ Check constraints

Rows per batch:

Maximum insert commit size:

Walidacja

| | |
|-------------------|--|
| General | |
| Parameter Mapping | |
| Result Set | |
| Expressions | |

| | |
|------------------------|-----------------------------------|
| ▼ General | |
| Name | Validation |
| Description | Data validation |
| ▼ Options | |
| TimeOut | 0 |
| CodePage | 1252 |
| TypeConversionMode | Allowed |
| ▼ Result Set | |
| ResultSet | None |
| ▼ SQL Statement | |
| ConnectionType | OLE DB |
| Connection | DestinationConnectionOLEDB |
| SQLSourceType | Direct input |
| SQLStatement | EXEC tmp_na_check |
| IsQueryStoredProcedure | False |
| BypassPrepare | True |

Usuwanie plików z danymi

| | |
|----------------|--|
| General | |
| Expressions | |

| | |
|----------------------------|---|
| ▼ General | |
| Name | Delete file |
| Description | Delete source file after transfer data to db |
| ▼ Operation | |
| Operation | Delete file |
| ▼ Source Connection | |
| IsSourcePathVariable | False |
| SourceConnection | SourceConnectionFlatFile |

Walidacja danych

Walidację danych realizują procedury SQL – tmp_na_check sprawdzająca nagłówki oraz tmp_poz_check sprawdzająca pozycje.

Procedura tmp_na_check sprawdza:

- czy plik z nagłówkami nie jest pusty
- czy numer wyciągu jest unikalny
- czy daty określające okres rozpatrywany przez wyciąg są zgodne z chronologią (początek okresu musi być datą wcześniejszą od końca)
- czy data utworzenia wyciągu jest datą późniejszą od daty końcowej okresu rozpatrywanego przez wyciąg
- czy podane daty nie są datami z przyszłości
- czy podana waluta znajduje się w słowniku walut
- czy nagłówek posiada pozycje

Procedura tmp_poz_check sprawdza:

- czy w procedurze tmp_na_check nie wykryto błędu
- czy każda pozycja ma swój nagłówek
- czy data pozycji wyciągu zawiera się w okresie rozpatrywanym przez wyciąg

Przykładowy fragment części skryptu odpowiedzialnej za walidację:

```
SELECT @totalRows = COUNT(*) FROM tmp_wb_na

SELECT @uniqueNum = COUNT(DISTINCT numer) FROM tmp_wb_na
IF @uniqueNum < @totalRows
BEGIN
    SET @en = @en + 'Bank statement number must be unique !!!'
    INSERT INTO ELOG_N(opis_n) VALUES (@en)
    SET @id_en = SCOPE_IDENTITY()

    INSERT INTO ELOG_D(id_eleg_n, opis_d)
    SELECT DISTINCT @id_en, t.numer
    FROM tmp_wb_na t
    WHERE t.numer IN (
        SELECT numer
        FROM tmp_wb_na
        GROUP BY numer
        HAVING COUNT(*) > 1
    )

    RAISERROR(@en, 16, 4)
    SET @err = 1
    RETURN -1
    RETURN -1
END
```

Informowanie o błędach walidacji

Tabele ELOG_N oraz ELOG_D przechowują odpowiednio nagłówki oraz szczegóły wykrytych błędów podczas walidacji. Poniżej przedstawione przykład informacji zawartych w tabelach w przypadku wykrycia błędów w danych wejściowych.

Tabela ELOG_N:

| | id_eleg_n | opis_n | dt | u_name | h_name |
|---|-----------|---|-------------------------|--------|--------|
| 1 | 1 | Error in procedure tmp_na_check / Every date must be before current date!!! | 2024-04-12 17:23:52.693 | dbo | ARKPC |

Tabela ELOG_D:

| | id_eleg_n | opis_d |
|---|-----------|--|
| 1 | 1 | Invalid future data in row: data.utw. 07.05.2025, data_od: 01.04.2023, data_do: 30.04.2023 |

Generowanie pliku XML

System generuje plik JPK_WB w formacie *xml* zgodnie z jego zdefiniowaną strukturą. Procedura JPK_WB_1 realizuje generowanie pliku na podstawie danych zawartych w utworzonych w tym celu tabelach tymczasowych.

Poniżej przedstawiono fragmenty skryptu tworzące tabele tymczasowe:

```
SELECT
    ,      numer                AS numer_wyciagu
    ,      i.numer_rach         AS numer_rachunku
    ,      dbo.SAFT_GET_AMT(saldo_pocz) AS saldo_pocztkowe
    ,      dbo.SAFT_GET_AMT(saldo_kon)  AS saldo_koncowe
    ,      waluta_rach          AS domyslne_kod_waluty
    ,      dbo.SAFT_DATE(data_utw)      AS data_wytworzenia_jpk
    ,      dbo.SAFT_DATE(data_od)       AS data_od
    ,      dbo.SAFT_DATE(data_do)       AS data_do
    ,      dbo.SAFT_GET_AMT(dbo.PobierzSumeObciazen(@numer)) AS suma_obciazen
    ,      dbo.SAFT_GET_AMT(dbo.PobierzSumeUznan(@numer))  AS suma_uznan
    ,      dbo.LiczbaWierszyDlaNumeru(@numer) AS liczba_wierszy
    INTO #TI
    FROM WB i (NOLOCK)
    join known_acc_num k (NOLOCK) ON (k.numer_rach = i.numer_rach)
    join Podmiot c (NOLOCK) ON (c.PODMIOT_ID = k.id_podmiotu)
    WHERE (i.numer = @numer)
    ORDER BY i.numer
```

```
SELECT
    ,      p.numer            AS numer_wyciagu
    ,      p.lp              AS numer_wiersza
    ,      dbo.SAFT_DATE(p.data) AS data_operacji
    ,      dbo.SAFT_GET_AMT(p.kwota) AS kwota_operacji
    ,      dbo.SAFT_GET_AMT(p.saldo_po) AS saldo_operacji
    ,      p.nazwa_kontrahenta AS nazwa_podmiotu
    ,      p.opis            AS opis_operacji
    INTO #TIT
    FROM WB_DET p (NOLOCK)
    join WB f ON (p.numer = f.numer)
    join known_acc_num k (NOLOCK) ON (k.numer_rach = f.numer_rach)
    join Podmiot c (NOLOCK) ON (c.PODMIOT_ID = k.id_podmiotu)
    WHERE (f.numer = @numer)
    ORDER BY f.numer
```

Poniżej przedstawiono fragment generujący jedną z sekcji finalnego pliku XML:

```
(SELECT numer_wiersza AS [tns:NumerWiersza]
    , data_operacji AS [tns:DataOperacji]
    , nazwa_podmiotu AS [tns:NazwaPodmiotu]
    , opis_operacji AS [tns:OpisOperacji]
    , kwota_operacji AS [tns:KwotaOperacji]
    , saldo_operacji AS [tns:SaldoOperacji]
    FROM #TIT
    FOR XML PATH('tns:WyciagWiersz'), TYPE
),
```

Wygenerowany plik XML

Poniżej przedstawiono wygenerowany na podstawie testowych danych plik JPK_WB w formacie XML:

```
<tns:JPK xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:kck="http://crd.gov.pl/xml/schematy/dziedzinowe">
  <tns:Naglowek xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:kck="http://crd.gov.pl/xml/schematy/dzie
    <tns:KodFormularza wersjaSchemy="1-0" kodSystemowy="JPK_WB (1)">JPK_WB</tns:KodFormularza>
    <tns:WariantFormularza>1</tns:WariantFormularza>
    <tns:CelZlozenia>1</tns:CelZlozenia>
    <tns:DataWytworzeniaJPK>2024-04-12T17:36:16.487</tns:DataWytworzeniaJPK>
    <tns:DataOd>2023-04-01T00:00:00</tns:DataOd>
    <tns:DataDo>2023-04-30T00:00:00</tns:DataDo>
    <tns:DomylnyKodWaluty>PLN</tns:DomylnyKodWaluty>
    <tns:KodUrzedu>026</tns:KodUrzedu>
  </tns:Naglowek>
  <tns:Podmiot1 xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:kck="http://crd.gov.pl/xml/schematy/dzie
    <tns:IdentyfikatorPodmiotu xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:kck="http://crd.gov.pl/x
      <etd:NIP>1234567890</etd:NIP>
      <etd:PeInaNazwa>firma XXX</etd:PeInaNazwa>
      <etd:REGON>012345678</etd:REGON>
    </tns:IdentyfikatorPodmiotu>
    <tns:AdresPodmiotu xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:kck="http://crd.gov.pl/xml/schem
      <etd:KodKraju>PL</etd:KodKraju>
      <etd:Wojewodztwo>Mazowieckie</etd:Wojewodztwo>
      <etd:Powiat>warszawski</etd:Powiat>
      <etd:Gmina>Warszawa</etd:Gmina>
      <etd:Ulica>Marszałkowska</etd:Ulica>
      <etd:NrDomu>100</etd:NrDomu>
      <etd:NrLokalu>12A</etd:NrLokalu>
      <etd:Miejscowosc>Warszawa</etd:Miejscowosc>
      <etd:KodPocztowy>00-001</etd:KodPocztowy>
      <etd:Poczta>Warszawa</etd:Poczta>
    </tns:AdresPodmiotu>
  </tns:Podmiot1>
  <tns:NumerRachunku xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:kck="http://crd.gov.pl/xml/schematy
    <tns:NumerRachunku>PL61109010140000071219812874</tns:NumerRachunku>
  </tns:NumerRachunku>
  <tns:Salda xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:kck="http://crd.gov.pl/xml/schematy/dziedzi
    <tns:SaldoPocztakowe>1000.00</tns:SaldoPocztakowe>
    <tns:SaldoKoncowe>1500.00</tns:SaldoKoncowe>
  </tns:Salda>
  <tns:WyciagWiersz xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:kck="http://crd.gov.pl/xml/schematy/
    <tns:NumerWiersza>1</tns:NumerWiersza>
    <tns>DataOperacji>2024-04-01</tns>DataOperacji>
    <tns:NazwaPodmiotu>Telekomunikacja Polska S.A.</tns:NazwaPodmiotu>
    <tns:OpisOperacji>Opłata za usługi telekomunikacyjne</tns:OpisOperacji>
    <tns:KwotaOperacji>-500.00</tns:KwotaOperacji>
    <tns:SaldoOperacji>500.00</tns:SaldoOperacji>
  </tns:WyciagWiersz>
  <tns:WyciagWiersz xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:kck="http://crd.gov.pl/xml/schematy/
    <tns:NumerWiersza>2</tns:NumerWiersza>
    <tns>DataOperacji>2024-04-03</tns>DataOperacji>
    <tns:NazwaPodmiotu>XYZ Sp. z o.o.</tns:NazwaPodmiotu>
    <tns:OpisOperacji>Wpłata od XYZ Sp. z o.o.</tns:OpisOperacji>
    <tns:KwotaOperacji>2000.00</tns:KwotaOperacji>
    <tns:SaldoOperacji>2500.00</tns:SaldoOperacji>
  </tns:WyciagWiersz>
  <tns:WyciagWiersz xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:kck="http://crd.gov.pl/xml/schematy/
    <tns:NumerWiersza>3</tns:NumerWiersza>
    <tns>DataOperacji>2024-04-15</tns>DataOperacji>
    <tns:NazwaPodmiotu>ABCD</tns:NazwaPodmiotu>
    <tns:OpisOperacji>Przelew wychodzący na rzecz ABCD</tns:OpisOperacji>
    <tns:KwotaOperacji>-1000.00</tns:KwotaOperacji>
    <tns:SaldoOperacji>1500.00</tns:SaldoOperacji>
  </tns:WyciagWiersz>
  <tns:WyciagWiersz xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:kck="http://crd.gov.pl/xml/schematy/
    <tns:NumerWiersza>4</tns:NumerWiersza>
    <tns>DataOperacji>2024-04-20</tns>DataOperacji>
    <tns:NazwaPodmiotu>Office Supplies Co.</tns:NazwaPodmiotu>
    <tns:OpisOperacji>Zakup materiałów biurowych</tns:OpisOperacji>
    <tns:KwotaOperacji>-200.00</tns:KwotaOperacji>
    <tns:SaldoOperacji>1300.00</tns:SaldoOperacji>
  </tns:WyciagWiersz>
  <tns:WyciagWiersz xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:kck="http://crd.gov.pl/xml/schematy/
    <tns:NumerWiersza>5</tns:NumerWiersza>
    <tns>DataOperacji>2024-04-27</tns>DataOperacji>
    <tns:NazwaPodmiotu>Jan Kowalski</tns:NazwaPodmiotu>
    <tns:OpisOperacji>Wpłata gotówkowa</tns:OpisOperacji>
    <tns:KwotaOperacji>500.00</tns:KwotaOperacji>
    <tns:SaldoOperacji>1800.00</tns:SaldoOperacji>
  </tns:WyciagWiersz>
  <tns:WyciagCtrl xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:kck="http://crd.gov.pl/xml/schematy/dz
    <tns:LiczbaWierszy>5</tns:LiczbaWierszy>
    <tns:SumaObciazon>-1700.00</tns:SumaObciazon>
    <tns:SumaUznan>2500.00</tns:SumaUznan>
  </tns:WyciagCtrl>
</tns:JPK>
```